Implementasi Enkripsi Pesan Menggunakan ChaCha20-Poly1305 sebagai Enkripsi Terotentikasi

Disusun oleh :

Jonathan Ferdinand Mayon	**1203230087**
Yonathan Hari Dharmawan	**1203230050**
Zakaria Arrozi	**1203230089**

PROGRAM STUDI

INFORMATIKA TELKOM

UNIVERSITY SURABAYA

SURABAYA

2025

DAFTAR ISI

Pendahuluan

Keamanan informasi merupakan salah satu aspek yang sangat penting dalam perkembangan teknologi informasi dan komunikasi saat ini. Pertukaran data melalui jaringan komputer, terutama melalui internet, telah menjadi bagian yang tidak terpisahkan dari berbagai aktivitas, seperti komunikasi pribadi, transaksi keuangan, sistem pemerintahan, hingga layanan berbasis cloud. Namun, kemudahan dalam pertukaran data tersebut juga diiringi dengan meningkatnya ancaman keamanan, seperti penyadapan data, pencurian informasi, serta manipulasi pesan oleh pihak yang tidak berwenang.

Dalam konteks tersebut, kriptografi memegang peranan penting sebagai solusi untuk melindungi data. Kriptografi memungkinkan pesan yang dikirimkan diubah ke dalam bentuk yang tidak dapat dipahami oleh pihak lain selain penerima yang sah. Akan tetapi, kebutuhan keamanan modern tidak hanya terbatas pada menjaga kerahasiaan pesan (*confidentiality*), tetapi juga mencakup jaminan bahwa pesan tidak mengalami perubahan (*integrity*) serta memastikan keaslian sumber pesan (*authentication*).

Algoritma enkripsi konvensional yang hanya berfokus pada kerahasiaan data memiliki keterbatasan karena tidak mampu mendeteksi adanya perubahan data selama proses transmisi. Oleh karena itu, dikembangkan konsep *Authenticated Encryption*, yaitu metode enkripsi yang menggabungkan proses enkripsi dan autentikasi dalam satu mekanisme terpadu. Salah satu implementasi dari konsep ini adalah *Authenticated Encryption with Associated Data* (AEAD), yang tidak hanya melindungi plaintext, tetapi juga mampu mengautentikasi data tambahan yang tidak dienkripsi.

ChaCha20-Poly1305 merupakan algoritma kriptografi modern yang termasuk dalam skema AEAD. Algoritma ini menggabungkan ChaCha20 sebagai algoritma enkripsi dan Poly1305 sebagai algoritma autentikasi. ChaCha20-Poly1305 dikenal memiliki tingkat keamanan yang tinggi, performa yang baik, serta efisiensi yang unggul pada berbagai platform, termasuk perangkat dengan sumber daya terbatas. Karena keunggulan tersebut, ChaCha20-Poly1305 telah diadopsi secara luas dalam berbagai protokol keamanan, seperti TLS dan QUIC.

Gambaran Umum

Kriptografi

Kriptografi adalah teknik pengamanan informasi dengan cara mengubah data asli (plaintext) menjadi data tersandi (ciphertext) menggunakan algoritma dan kunci tertentu. Tujuan utama kriptografi adalah menjaga kerahasiaan, integritas, autentikasi, dan non-repudiation.

ChaCha20

ChaCha20 adalah algoritma stream cipher yang dirancang oleh Daniel J. Bernstein. Algoritma ini dikenal memiliki performa tinggi dan tingkat keamanan yang baik, terutama pada perangkat dengan keterbatasan sumber daya.

Poly1305

Poly1305 merupakan algoritma Message Authentication Code (MAC) yang berfungsi untuk memastikan integritas dan autentikasi data. Algoritma ini dapat mendeteksi perubahan data yang tidak sah.

Authenticated Encryption with Associated Data (AEAD)

AEAD adalah metode enkripsi yang menggabungkan enkripsi dan autentikasi dalam satu proses. Selain plaintext, AEAD juga mendukung Associated Authenticated Data (AAD), yaitu data tambahan yang tidak dienkripsi tetapi tetap diautentikasi.

Metode Pelaksanaan

Tools :

- Bahasa pemrograman Python
- Library cryptography
- Modul os

Tahapan alur sistem :

1. Input Pesan :Pengguna memasukkan pesan dalam bentuk teks melalui terminal. Pesan ini berfungsi sebagai plaintext yang akan diamankan.
2. Konversi Data : Pesan yang masih berbentuk string dikonversi ke dalam format byte agar dapat diproses oleh algoritma kriptografi.
3. Penentuan AAD (Associated Authenticated Data) : Sistem menetapkan AAD sebagai data tambahan yang tidak dienkripsi, namun tetap diautentikasi untuk menjamin integritasnya.
4. Pembuatan Key dan Nonce : Sistem menghasilkan kunci rahasia (key) secara acak dan nonce unik menggunakan generator bilangan acak yang aman.
5. Proses Enkripsi : Plaintext, nonce, dan AAD diproses menggunakan algoritma ChaCha20-Poly1305 sehingga menghasilkan ciphertext yang telah dilengkapi dengan authentication tag.
6. Proses Dekripsi : Ciphertext didekripsi kembali menggunakan key, nonce, dan AAD yang sama untuk memastikan bahwa pesan dapat dikembalikan ke bentuk semula tanpa perubahan.

Implementasi Program

Input Data

```
pesan = input("Masukkan Pesan : ")
data = pesan.encode()
```

Program diawali dengan meminta pengguna untuk memasukkan pesan melalui terminal. Pesan yang dimasukkan masih berupa data bertipe string. Agar dapat diproses oleh algoritma kriptografi, pesan tersebut dikonversi ke dalam format byte menggunakan fungsi encode(). Proses ini penting karena algoritma ChaCha20-Poly1305 hanya dapat bekerja dengan data bertipe byte.

Associated Authenticated Data (AAD)

```
aad = b"authenticated but unencrypted data"
```

Associated Authenticated Data (AAD) merupakan data tambahan yang tidak dienkripsi, namun tetap dilibatkan dalam proses autentikasi. Pada implementasi ini, AAD didefinisikan secara statis sebagai sebuah nilai byte. Peran AAD adalah untuk memastikan bahwa data tambahan tidak mengalami perubahan selama proses transmisi. Apabila AAD diubah, maka proses dekripsi akan gagal meskipun ciphertext dan key yang digunakan benar.

Pembuatan Key dan Nonce

```
key = ChaCha20Poly1305.generate_key()
print("key:", key)
nonce = os.urandom(12)
print("nonce:", nonce)
```

Key dan nonce merupakan komponen penting dalam proses enkripsi. Key dihasilkan secara acak menggunakan fungsi generate_key() dengan panjang 256-bit, sesuai dengan standar ChaCha20-Poly1305. Sementara itu, nonce dibuat menggunakan fungsi os.urandom(12) yang menghasilkan nilai acak sepanjang 12 byte. Nonce harus bersifat unik untuk setiap proses enkripsi agar keamanan sistem tetap terjaga.

Inisialisasi Algoritma ChaCha20-Poly1305

```
chacha = ChaCha20Poly1305(key)
```

Setelah key dihasilkan, sistem melakukan inisialisasi objek ChaCha20-Poly1305 menggunakan key tersebut. Objek ini selanjutnya digunakan untuk melakukan proses enkripsi dan dekripsi pesan.

Proses Enkripsi

```
ciphertext = chacha.encrypt(nonce, data, aad)

print("=== ENKRIPSI ===")
print("Plaintext :", data)
print("AAD       :", aad)
print("Ciphertext:", ciphertext)
```

Proses enkripsi dilakukan dengan memanggil fungsi encrypt() yang menerima parameter nonce, plaintext, dan AAD. Hasil dari proses ini adalah ciphertext yang sudah mencakup data terenkripsi beserta authentication tag. Authentication tag berfungsi untuk mendeteksi adanya perubahan data selama proses penyimpanan atau pengiriman.

Proses Dekripsi

```
decrypted = chacha.decrypt(nonce, ciphertext, aad)

print("\n=== DEKRIPSI ===")
print("Hasil     :", decrypted)
```

Tahap terakhir adalah proses dekripsi, yang dilakukan menggunakan fungsi decrypt() dengan parameter key, nonce, ciphertext, dan AAD yang sama seperti saat proses enkripsi. Jika seluruh parameter sesuai dan data tidak mengalami perubahan, maka ciphertext berhasil dikembalikan ke bentuk plaintext semula. Proses ini sekaligus membuktikan bahwa mekanisme enkripsi dan autentikasi berjalan dengan baik.

Hasil dan Pembahasan

Hasil pengujian menunjukkan bahwa sistem berhasil melakukan enkripsi dan dekripsi pesan menggunakan algoritma ChaCha20-Poly1305. Pesan asli (plaintext) berhasil diubah menjadi ciphertext yang tidak dapat dibaca, dan dapat dikembalikan ke bentuk semula melalui proses dekripsi dengan parameter yang sesuai.

Perbedaan ciphertext pada setiap proses enkripsi, meskipun pesan yang digunakan sama, menunjukkan bahwa penggunaan nonce acak berjalan dengan baik. Selain itu, mekanisme autentikasi Poly1305 mampu menjamin integritas data, karena perubahan pada ciphertext, nonce, atau AAD menyebabkan proses dekripsi gagal.

Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma ChaCha20-Poly1305 sebagai metode enkripsi terotentikasi memberikan solusi keamanan yang komprehensif bagi perlindungan data modern. Melalui mekanisme Authenticated Encryption with Associated Data (AEAD), algoritma ini tidak hanya berfokus pada kerahasiaan pesan melalui proses enkripsi, tetapi juga secara simultan menjamin integritas dan autentikasi data. Hal ini merupakan keunggulan signifikan dibandingkan algoritma enkripsi tradisional, karena sistem mampu mendeteksi adanya upaya manipulasi atau perubahan data sekecil apa pun selama proses transmisi melalui penggunaan authentication tag yang dihasilkan oleh Poly1305. Kegagalan autentikasi pada saat dekripsi secara otomatis mencegah data yang telah dimodifikasi untuk diproses lebih lanjut, sehingga memberikan perlindungan terhadap serangan bit-flipping maupun chosen-ciphertext attacks.