LAPORAN KRIPTOGRAFI TUGAS BESAR 2

Universitas Teknologi Digital

Disusun Oleh :

Mochammad Rival Sopyan	20123006
Fernanda Syah Putra	20123019
Muhamad Rifky Raihan	20123021

PROGRAM STUDI S1 INFORMATIKA

UNIVERSITAS TEKNOLOGI DIGITAL

TAHUN AJARAN 2024/2025

1. Pendahuluan

Keamanan data menjadi aspek penting dalam pertukaran informasi digital. Untuk melindungi kerahasiaan, integritas, dan autentikasi data, digunakan berbagai algoritma kriptografi modern. Praktikum ini bertujuan mempelajari dan mengimplementasikan empat algoritma utama, yaitu **AES**, **RSA**, **Hash Functions** (MD5, SHA-1, SHA-256), dan **Digital Signature**, menggunakan bahasa Python. Melalui eksperimen ini, mahasiswa dapat memahami cara kerja, karakteristik, serta tingkat keamanan masing-masing algoritma dalam melindungi data pada sistem informasi modern.

2. Landasan Teori

2.1 AES (Advanced Encryption Standard)

Jenis: Simetris (kunci sama untuk enkripsi & dekripsi).

Konsep utama:

AES bekerja dengan membagi data menjadi blok 128 bit dan mengenkripsi setiap blok menggunakan kunci 128, 192, atau 256 bit. Algoritma ini terdiri dari beberapa putaran (round) transformasi yang melibatkan substitusi, permutasi, dan operasi matematika di medan GF(2^8).

2.2 RSA (Rivest–Shamir–Adleman)

Jenis: Asimetris (kunci publik & kunci privat berbeda)

Konsep utama: RSA didasarkan pada kesulitan faktorisasi bilangan besar menjadi dua bilangan prima. Pesan dienkripsi menggunakan kunci publik (n, e) dan hanya bisa didekripsi dengan kunci privat (d).

2.3 Hash Functions

Jenis: One-way (tidak dapat dibalik)

Konsep utama: Hash function mengubah input (teks atau file) menjadi nilai hash tetap (misal 128-bit atau 256-bit). Tujuannya bukan untuk enkripsi, tapi untuk integritas data memastikan data tidak diubah.

Contoh algoritma: MD5, SHA-1, SHA-256.

2.4 Digital Signature

Jenis: Asimetris (menggunakan RSA, DSA, atau ECDSA)

Konsep utama: Digital signature digunakan untuk menjamin keaslian (authenticity) dan integritas (integrity) pesan. Pengirim menandatangani pesan dengan kunci privat, penerima memverifikasi dengan kunci publik.

3. Implementasi & Eksperimen

Tujuan: menunjukkan hasil nyata dari kode yang dijalankan.

3.1 AES

Output:

Generated AES-256 key (base64):

uhnTbZ36bF8wQ5J1/QFtgmCcOznJs7eYzCF6Vndwhio=

Ciphertext (base64):

btLeEhj6syQ3UmACsU8noDARv2AaElpyaXoQyd8wV0lEuknP1C69InLDPTdazL+y8a xUe+hB0/ZrYSlcSA2a2Q==

Recovered: Pesan rahasia. Ini contoh AES-CBC.

Analisis:

- AES-256 menghasilkan ciphertext acak dan tidak dapat dibaca tanpa kunci.
- Proses dekripsi berhasil, menandakan simetri kunci berjalan dengan baik.

3.2 Hash Functions

Output:

Original : b'Hello, dunia!'

Modified : b'Hella, dunia!'

MD5 : 719d9c3f73911356beca20d13a2b8c00 / f6cd467a3ef91c73ba04f42ebf7c8f92

SHA1 : 804e665ede7b7ffbc3debd1355cea0614d9b245e / ba353df510b522cbc456701697cc505d7dfd8892

SHA256 : 8978cc925e8981a26fcaa44f577b8e425c4f0be359006d98e31678f63506b859 / f987bdd525e004d73c18d5b3c18a81609e6db47afceca8573fa349769817495d

Analisis:

- Perubahan satu huruf (“o” → “a”) menyebabkan seluruh hash berubah total → menunjukkan **avalanche effect.**
- MD5 dan SHA-1 sudah tidak direkomendasikan untuk keamanan tinggi (rawan collision), tapi tetap berguna untuk pembelajaran.

- SHA-256 lebih kuat karena panjang hash lebih besar dan resistansi kolisi lebih tinggi.

3.3 RSA

Output:

Generated rsa_private.pem and rsa_public.pem

Analisis:

- RSA menghasilkan sepasang kunci: rsa_private.pem (privat) dan rsa_public.pem (publik).
- Dapat digunakan untuk enkripsi, dekripsi, atau tanda tangan digital.
- Ukuran kunci (misal 2048 bit) memengaruhi tingkat keamanan dan kecepatan.

3.4 Digital Signature

Output:

Generated keys: sig_private.pem, sig_public.pem

Message: b'Pesan untuk ditandatangani'

Signature (len): 256

Verify result: True

Analisis:

- Hasil Verify result: True menunjukkan tanda tangan valid dan pesan tidak dimodifikasi.
- Panjang signature 256 byte menunjukkan penggunaan RSA 2048-bit.
- Membuktikan integritas dan autentikasi pesan.

4. Analisis Keamanan

Algoritma	Jenis	Kelebihan	Kelemahan	Aplikasi
AES	Simetris	Cepat, kuat, standar global	Kunci harus dijaga rahasia	Enkripsi file/data
RSA	Asimetris	Aman untuk pertukaran kunci	Lambat untuk data besar	SSL/TLS, tanda tangan digital
Hash (SHA-256)	One-way	Integritas kuat, cepat	Tidak bisa didekripsi	Verifikasi file, password

Algoritma	Jenis	Kelebihan	Kelemahan	Aplikasi
Digital Signature	Kombinasi	Menjamin keaslian & integritas	Membutuhkan manajemen kunci	Sertifikat digital, transaksi online

Kesimpulan utama: AES cocok untuk enkripsi data massal, RSA untuk distribusi kunci, Hash untuk verifikasi, dan Digital Signature untuk autentikasi.

5. Kesimpulan

- Implementasi algoritma kriptografi modern berhasil dijalankan dengan benar.
- Masing-masing algoritma memiliki fungsi berbeda namun saling melengkapi.
- AES menunjukkan efisiensi tinggi, RSA menunjukkan keamanan berbasis kunci publik, hash menunjukkan efek avalanche, dan digital signature membuktikan integritas data.
- Penggabungan (hybrid encryption) memperlihatkan penerapan nyata kriptografi modern.