TUGAS 1
RESUME ARTIKEL

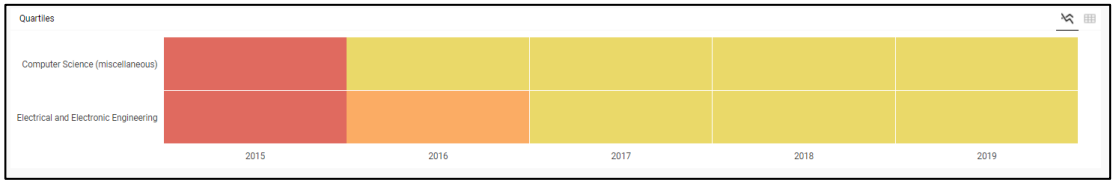
Files cryptography based on one-time pad algorithm

Disusun Oleh:
Dyan Azka Ingkafi
24060118130139

PROGRAM STUDI S1 INFORMATIKA
DEPARTEMEN ILMU KOMPUTER/INFORMATIKA
UNIVERSITAS DIPONEGORO
2021

INFORMASI ARTIKEL

1. Judul Artikel : Files cryptography based on one-time pad algorithm
2. Jurnal : International Journal of Electrical and Computer Engineering (IJECE)
3. Volume / Issue (Nomor) : 11 / 3
4. Halaman : 8
5. Scopus Quartile :



RESUME

Dewasa ini, enkripsi berperan penting di berbagai aspek, seperti militer, rahasia dagang, dan citra satelit. Enkripsi memungkinkan seseorang untuk menyembunyikan makna informasi atau pesan sehingga hanya mereka yang mengetahui metode rahasia yang dapat membacanya. Enkripsi saat ini dicapai dengan memanfaatkan algoritma yang memiliki kunci untuk mengenkripsi dan mendekripsi data. Secara prinsip, semakin panjang kunci maka semakin kompleks untuk memecahkan kode tersebut. Sebagai contoh, setiap unit biner data memiliki estimasi 0 atau 1. Sehingga, kunci 8-bit akan memiliki 256 potensial kunci sedangkan kunci dengan 56-bit akan memiliki 72 kuadriliun potensial kunci. Dengan semakin berkembangnya teknologi, sandi yang menggunakan kunci dengan panjang tersebut semakin mudah dipecahkan. Salah satu peningkatan luar biasa dalam penelitian kriptografi adalah penyajian kunci sandi asimetris yaitu algoritma yang memanfaatkan dua kunci secara matematis untuk mengenkripsi pesan yang sama. Oleh karena itu, pembentukan protokol baru yang dikenal sebagai *secure socket layer* (SSL), membuka jalan baru untuk transaksi *online* seperti proses pembelian, pembayaran tagihan *online,* dan perbankan*.*

One-time pad algorithm berasal dari sandi sebelumnya yang disebut *Vernam Cipher*. *Vernam Cipher* adalah *cipher* yang menggabungkan pesan dengan *keystream* yang dibaca dari pita kertas atau *pad*. *Pad* satu kali biasanya diterapkan dengan menggunakan tambahan modular (XOR) untuk menggabungkan elemen teks biasa dengan elemen aliran utama. Manfaat menggunakan operasi XOR adalah dapat dikembalikan hanya dengan mengimplementasikan operasi yang sama. Berikut adalah formula yang mengilustrasikan proses enkripsi dan dekripsi pada algoritma Vernam.

Enkripsi: $P \oplus k = C$;

Dekripsi: $C \oplus K = P$,

Dimana $\oplus$ menunjukkan operasi XOR, P mewakili teks biasa atau *plain text*, K mewakili kunci, dan C mewakili teks sandi atau *cipher text* . *Advanced Encryption Standard* (AES) adalah cabang dari *block cipher* Rijndael yang dibuat oleh dua kriptografer Belgia. Rijndael adalah sekelompok *cipher* dengan berbagai ukuran kunci dan blok. Algoritma yang digambarkan oleh AES adalah algoritma kunci-simetris, yang mana kunci serupa digunakan untuk mengenkripsi dan mendekripsi data. Kelemahan utama dalam menggunakan *one-time pad* adalah kunci enkripsi memiliki panjang yang sama dengan pesan yang ingin dienkripsi. Dengan demikian, pada artikel ini, kunci acak dibuat sebelum menerapkan algoritma Vernam kemudian dikompresi dengan pesan yang akan dienkripsi.

Pada tahap implementasi, sistem dibuat menggunakan bahasa pemrograman VB.Net. Langkah pertama dalam metodologi penelitian ini yaitu kunci acak enkripsi dibuat untuk mengenkripsi teks biasa menggunakan algoritma Vernam, kemudian kunci dengan versi terenkripsi disimpan dalam satu *file.* Kedua, algoritma AES digunakan untuk mengenkripsi kata sandi, yang dianggap sebagai titik pemisah antara teks terenkripsi dan kunci enkripsi. Fase enkripsi mencakup dua algoritma (Vernam dan AES) dan menggunakan teknik

steganografi sederhana untuk menyembunyikan data kunci kriptografi. Ketiga, ukuran *file* menjadi dua kali lipat karena kunci enkripsi dan data terenkripsi disimpan dalam file yang sama sehingga diimplementasikan kompresi data menggunakan algoritma Huffman.

Dengan demikian, metodologi ini telah mengatasi manajemen kunci enkripsi pada algoritma Vernam dengan mengontrol tipe data pada kunci enkripsi. Algoritma Huffman digunakan untuk mengurangi ukuran *file* keluaran. *File* keluaran dilindungi dengan kata sandi yang dienkripsi oleh algoritma AES, sehingga meningkatkan kesulitan dalam memecahkan *file* keluaran yang dienkripsi. Berbagai jenis *file*, seperti (.txt, .pdf, .doc, .bmp, .mp4, .exe), dilakukan untuk percobaan ini dan berhasil tanpa kehilangan informasi apapun. Lebih lanjut, hasil dari penelitian ini mendemonstrasikan bahwa waktu yang dihabiskan untuk enkripsi dan dekripsi *file,* yang mana dikompresi dengan kunci kriptografi yang dihasilkan dari *integer* memakan waktu lebih kecil dibandingkan kunci kriptografi yang dihasilkan dari tabel ASCII. Sementara itu, ukuran *file* berpengaruh kecil terhadap waktu enkripsi data tanpa kompresi. Sejak kriptosistem berperan penting dalam banyak aplikasi, tugas di masa yang akan datang adalah bagaimana mengeksplorasi perancangan sistem kriptografi yang aman pada kunci panjang enkripsi *one-time-pad.*

Research Gap

Metodologi pada penelitian ini, kompresi data dilakukan pada langkah terakhir. Hal ini dinilai tidak efisien oleh penelitian sebelumnya yang dilakukan oleh B. Carpentieri yang berjudul "*Efficient compression and encryption for digital data transmission*,". Menurut Carpentieri (2018), sama sekali tidak efisien untuk dilakukan enkripsi terlebih dahulu kemudian kompresi. Ukuran *file* pasti akan semakin besar dan dalam beberapa kasus *output* yang dihasilkan akan jauh lebih besar dari *input* aslinya. Dapat disimpulkan bahwa terjadi kontroversi antara penelitian Alsmadi dengan Carpentieri dalam hal kapan waktu yang efisien untuk dilakukan kompresi data. Oleh karena itu, diperlukan adanya penelitian lebih lanjut untuk mendapatkan hasil yang maksimal.