Shuffle Cipher

Karlsen Adiyasa Bachtiar (13519001)[1], Mochammad Fatchur Rochman (13519009)[2], Yudi Alfayat (13519051)[3]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519001@std.stei.itb.ac.id[1], 13519009@std.stei.itb.ac.id[2], 13519051@std.stei.itb.ac.id[3]

***Abstrak*—Pada makalah ini diajukan sebuah usulan pengembangan *block cipher* yang merupakan modifikasi dari algoritma *AES*-128 (*Advanced Encryption Standard*) yang bernama *Shuffle Cipher*. Modifikasi yang dilakukan adalah pada mekanisme *shift rows* algoritma *AES*-128 menjadi mekanisme *shuffle matrix* yang diharapkan dapat meningkatkan performa dari algoritma tersebut.**

Kata Kunci—block cipher; AES; shuffle matrix

I. PENDAHULUAN

Pada era informasi ini, pertukaran informasi dapat dilakukan dengan mudah melalui internet, kemudahan dalam mempertukarkan informasi tersebut juga memiliki sisi yang lain yaitu kemudahan untuk informasi tersebut bocor ke pihak yang tidak diinginkan jika proses pertukaran informasi tersebut tidak terjaga keamanannya. Oleh karena hal itu, diperlukan protokol keamanan yang dapat menjamin keamanan dan kerahasiaan dari informasi saat informasi tersebut dipertukarkan melalui internet. Salah satu protokol keamanan yang digunakan untuk menjaga keamanan dan kerahasiaan dari suatu informasi adalah dengan melakukan enkripsi menggunakan algoritma kriptografi.

Block cipher merupakan algoritma kriptografi simetrik yang mengenkripsi plainteks dengan cara membagi plainteks menjadi blok-blok bit dengan panjang yang sama, sehingga panjang blok cipherteks sama dengan panjang plainteks-nya, dalam proses enkripsi-nya *block cipher* menerapkan prinsip *confusion* dan *diffusion* sehingga dapat memastikan keamanan dari informasi jauh lebih baik dari algoritma kriptografi konvensional yang sudah sangat mudah dipecahkan dengan berbagai metode kriptanalisis yang memanfaatkan hubungan pola-pola statistik yang muncul antara cipherteks, kunci, dan plainteks.

Dalam makalah ini akan membahas algoritma *block cipher* yang bernama *Shuffle Cipher*. Algoritma ini berdasarkan pada *AES*-128 yang dilakukan modifikasi pada mekanisme *shift rows* menjadi mekanisme *shuffle matrix*.

II. DASAR TEORI

A. AES

AES (Advanced Encryption Standard) merupakan salah satu algoritma kriptografi simetri yang berbasis *block cipher*. Algoritma ini secara *de facto* memiliki 2 jenis varian yang didasarkan pada panjang kunci yang digunakannya yaitu *AES-128* untuk panjang kunci 128 bit dan *AES-256* untuk panjang kunci 256 bit. Algoritma yang digunakan pada *AES* adalah algoritma yang diusulkan oleh Vincent Rijmen dan Joan Daemen yaitu Algoritma Rijndael.

Algoritma Rijndael yang beroperasi pada blok 128-bit dengan kunci 128-bit dan 10 putaran (*AES-128*) adalah sebagai berikut (di luar proses pembangkitan *round key*):

1) *AddRoundKey*: melakukan *initial round* yaitu melakukan XOR antara *state* awal (plainteks) dengan *cipher key*.
2) Putaran sebanyak *Nr-1* kali, setiap putaran melakukan proses:
    a) *SubBytes*:

        melakukan substitusi *byte* dengan menggunakan tabel substitusi (*S-Box*).

    b) *ShiftRows*

        melakukan pergeseran baris-baris *array state* secara *wrapping*.

    c) *MixColumns*

        melakukan pengacakan data di masing-masing kolom *array state*.

    d) *AddRoundKey*

        melakukan XOR antara *state* sekarang dengan *round key*.

3) *Final round*, pada putaran terakhir melakukan proses:
    a) *SubBytes*

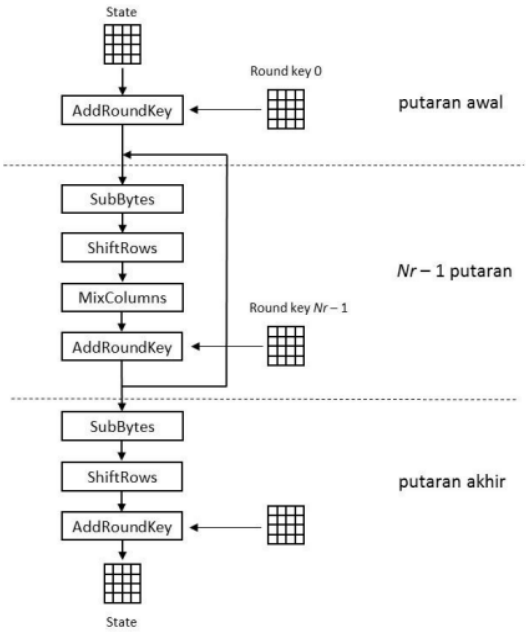
        melakukan substitusi *byte* dengan menggunakan tabel substitusi (*S-Box*).

    b) *ShiftRows*

        melakukan pergeseran baris-baris *array state* secara *wrapping*.

    c) *AddRoundKey*

        melakukan XOR antara *state* sekarang dengan *round key*.



Gambar 1. Algoritma *AES* [1]

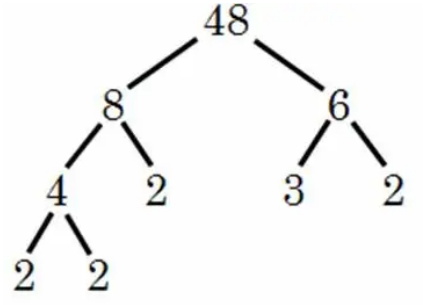
B. *Prinsip Confusion dan Diffusion*

Confusion adalah prinsip yang bertujuan untuk menyembunyikan hubungan statistik antara cipherteks, dan kunci sehingga sulit untuk mencari pola-pola statistik yang muncul di dalam cipherteks. Implementasi dari prinsip *confusion* dapat dilakukan dengan menggunakan teknik substitusi nirlanjar(*non-linear*). Pada *AES*, teknik substitusi direalisasikan menggunakan *S-Box*.

Diffusion adalah prinsip yang bertujuan untuk menyembunyikan hubungan statistik antara ciphertext dengan plaintext, prinsip ini memastikan pola-pola seperti bit yang redundan tidak akan muncul pada ciphertext yang dihasilkan. Implementasi dari prinsip *di* dapat dilakukan dengan menggunakan teknik permutasi/transposisi secara berulang. Pada *AES*, *diffusion* direalisasikan dengan teknik transposisi secara berulang yaitu pergeseran *byte*.

C. *Pohon Faktor*

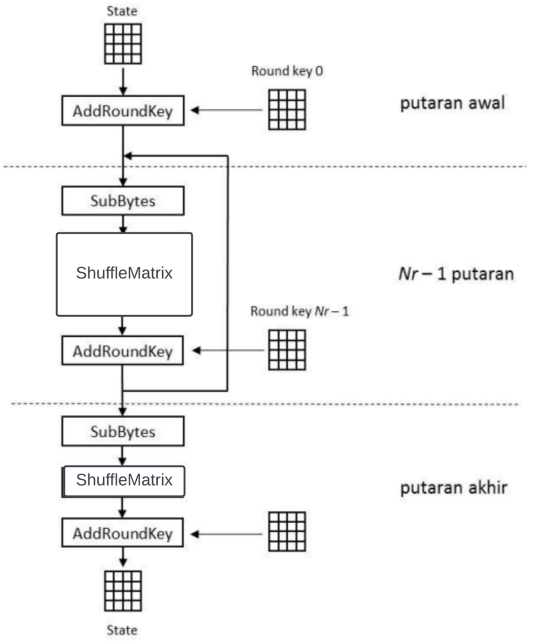
Pohon Faktor adalah diagram pohon yang mendekomposisikan suatu bilangan menjadi faktor-faktor prima dari bilangan tersebut.



Gambar 2. Rancangan *Block Cipher* [3]

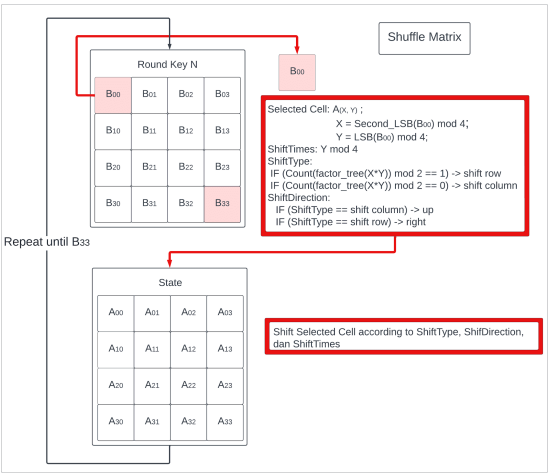
III. RANCANGAN BLOCK CIPHER

Secara umum, rancangan proses algoritma mirip dengan proses algoritma AES.



Gambar 3. Rancangan *Block Cipher*

Algoritma ini memiliki ukuran blok 128 bit, panjang kunci 128 bit, jumlah putaran 16, dan mengganti proses “shift rows dan mixColumn” menjadi “shuffle matrix” dengan alur pada gambar di bawah ini.



Gambar 4. *Shuffle Matrix*

Berikut alur dari proses “shuffle matrix”:

1. Setiap cell pada Round Key akan menentukan satu buah shift pada matrix state. Penentuan cell, shift type (row atau column), dan shift direction (kanan/atas untuk encrypt dan kiri/bawah untuk decrypt) ditentukan berdasarkan nilai dari cell Round Key.
2. Setelah menentukan cell state yang dipilih, shift type, shift direction, dan shiftTimes, akan dilakukan shift pada matrix state sesuai dengan yang ditentukan.
3. Langkah 1 akan dilakukan untuk setiap cell pada Round Key sehingga langkah 1 akan dilakukan dari B00 hingga B33 sehingga akan ada 16 kali pergeseran pada setiap putaran

A. Encrypt

1) *AddRoundKey*: melakukan *initial round* yaitu melakukan XOR antara *state* awal (plainteks) dengan *cipher key*.

2) Putaran sebanyak *Nr-1* kali, setiap putaran melakukan proses:

a) *SubBytes*:

melakukan substitusi *byte* dengan menggunakan tabel substitusi (*S-Box*).

b) *ShuffleMatrix*

Misal cell pertama dari Round Key adalah B_{00} (0x2D):

- Cell State yang dipilih adalah A_{21} karena second LSB(2) mod 4 adalah 2 dan LSB(D) mod 4 adalah 1
- X = second LSB = 2 ; Y = LSB = D
- Shift type yang dipilih adalah shift column karena jumlah pohon faktor dari X*Y adalah 2 yaitu 2 dan 13 sehingga menghasilkan genap jika dilakukan operasi mod 2
- Shift times adalah seberapa jauh matriks diputar, untuk kasus ini kolom akan dilakukan shift sebanyak 1 cell
- Shift direction yang dipilih adalah atas karena shift type nya merupakan shift column
- Ulangi hingga B_{33}

c) *AddRoundKey*

melakukan XOR antara *state* sekarang dengan *round key*.

3) *Final round*, pada putaran terakhir melakukan proses:

a) *SubBytes*

melakukan substitusi *byte* dengan menggunakan tabel substitusi (*S-Box*).

b) *ShuffleMatrix*

sama seperti langkah 2b

c) *AddRoundKey*

melakukan XOR antara *state* sekarang dengan *round key*.

B. Decrypt

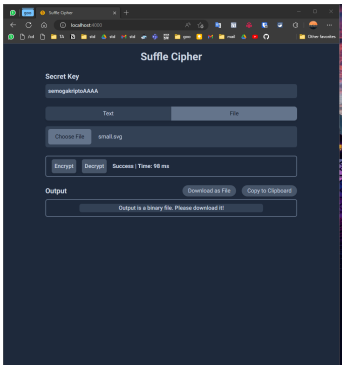
Dekripsi dilakukan dengan membalikkan semua proses Encypt dan shift direction diantara kiri (untuk shift row) / bawah (untuk shift column)

IV. EKSPERIMEN DAN PEMBAHASAN HASIL

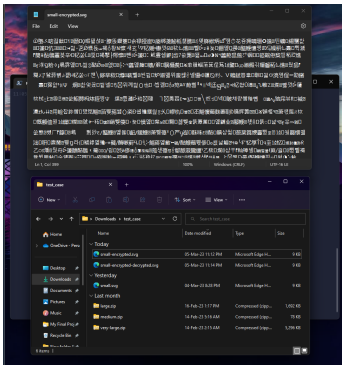
Berikut adalah beberapa eksperimen yang dilakukan dalam pengujian Shuffle Cipher. *file-file test case yang digunakan sudah terdapat pada repository.

A. Analisis File Small

File masukkan berjudul “small.svg” dan memiliki ukuran sebesar 9 kb.

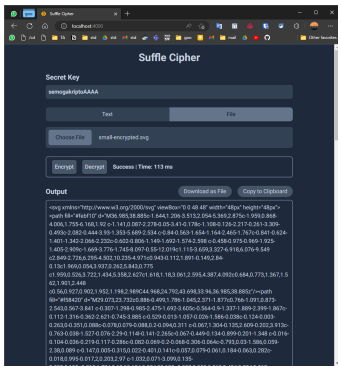


Gambar 5. Proses Enkripsi pada Web untuk *File Small*

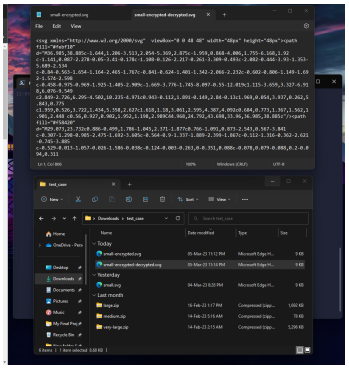


Gambar 6. File Hasil Enkripsi untuk *File Small*

Berhasil dienkripsi dengan proses enkripsi selama 98 milisecond.



Gambar 7. Proses Dekripsi pada Web untuk *File Small*

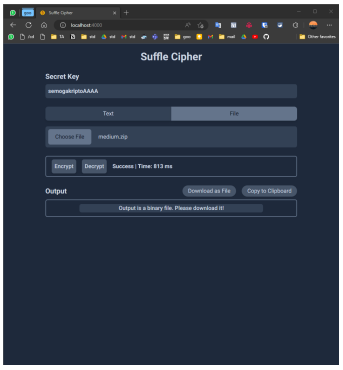


Gambar 8. File Hasil Dekripsi untuk *File Small*

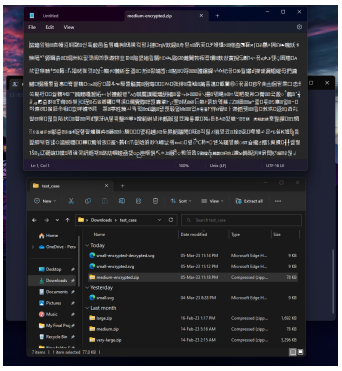
Berhasil didekripsi dengan proses dekripsi selama 113 milisecond.

B. Analisis File Medium

File masukkan berjudul “medium.zip” dan memiliki ukuran sebesar 78 kb.

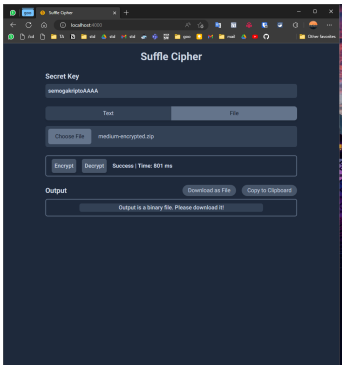


Gambar 9. Proses Enkripsi pada Web untuk *File Medium*

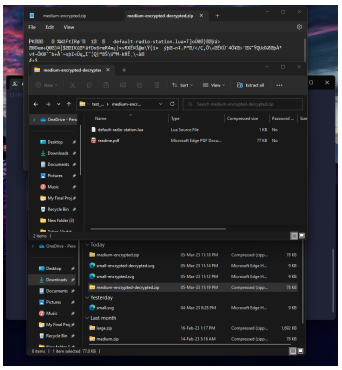


Gambar 10. File Hasil Enkripsi untuk *File Medium*

Berhasil dienkripsi dengan proses enkripsi selama 813 milisecond.



Gambar 11. Proses Dekripsi pada Web untuk *File Medium*

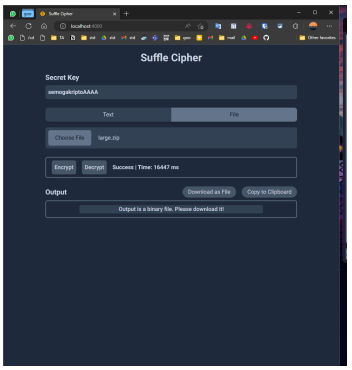


Gambar 12. File Hasil Dekripsi untuk *File Medium*

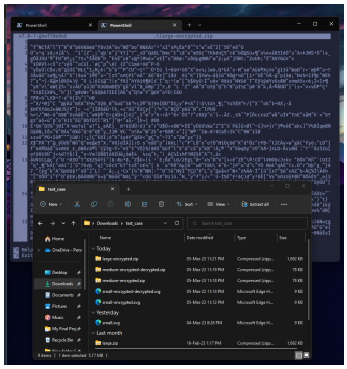
Berhasil didekripsi dengan proses dekripsi selama 801 milisecond.

C. Analisis File Large

File masukkan berjudul “large.zip” dan memiliki ukuran sebesar 1692 kb.

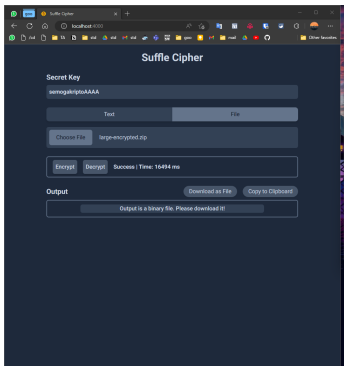


Gambar 13. Proses Enkripsi pada Web untuk *File Large*

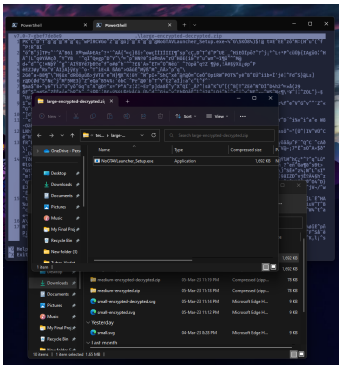


Gambar 14. File Hasil Enkripsi untuk *File Large*

Berhasil dienkripsi dengan proses enkripsi selama 16447 milisecond



Gambar 15. Proses Dekripsi pada Web untuk *File Large*

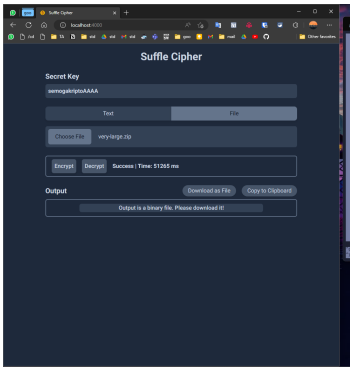


Gambar 16. File Hasil Dekripsi untuk *File Large*

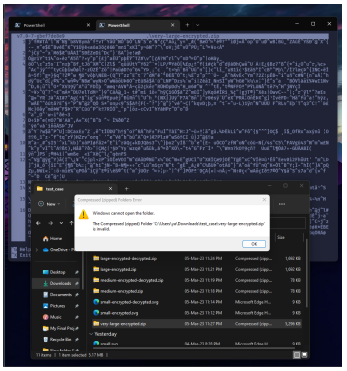
Berhasil didekripsi dengan proses dekripsi selama 16494 milisecond.

D. Analisis File Very Large

File masukkan berjudul “veryLarge.zip” dan memiliki ukuran sebesar 5296 kb.

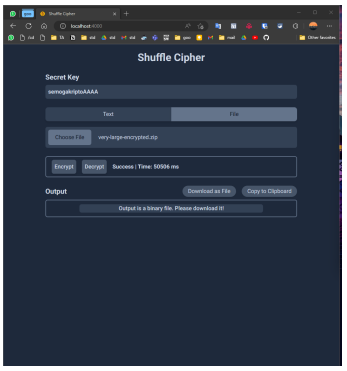


Gambar 17. Proses Enkripsi pada Web untuk *File Very Large*

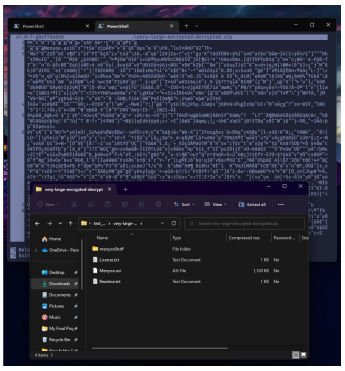


Gambar 18. File Hasil Enkripsi untuk *File Very Large*

Berhasil dienkripsi dengan proses enkripsi selama 51265 milisecond



Gambar 19. Proses Dekripsi pada Web untuk *File Very Large*

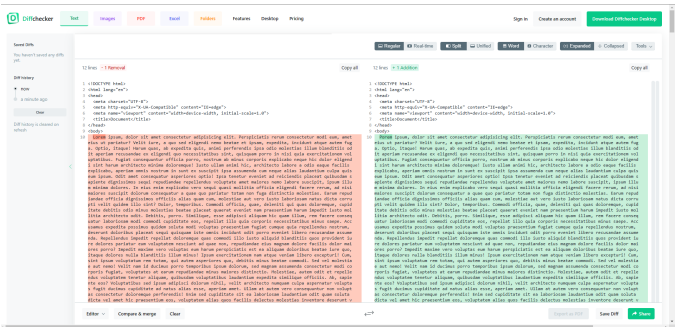


Gambar 20. File Hasil Dekripsi untuk *File Very Large*

Berhasil didekripsi dengan proses dekripsi selama 50506 milisecond.

E. Analisis Efek Longsoran

File masukkan berjudul “plainttext.” dan memiliki ukuran sebesar 73.9 kb.



Gambar 21. *Analisis Efek Longsoran* pada “plainttext.”

File masukkan berjudul “plainttext.” dan memiliki ukuran sebesar 73.9 kb.

Gambar 22. *Analisis Efek Longsoran* pada “plainttext.”

F. Analisis Keamanan

Algoritma ini memiliki keamanan yang cukup baik karena memiliki confusion (S-box) dan diffusion (shuffle matrix) walaupun algoritma initidak memiliki avalanche effect atau efek longsoran yang baik

V. KESIMPULAN DAN SARAN

Algoritma Shuffle Cipher ini memodifikasi AES dan cukup kuat karena memiliki bagian confusion dan diffusion walaupun tidak memiliki efek longsoran yang baik. Selain itu, kecepatan eksekusi algoritma ini masih dapat dikembangkan lebih lanjut agar semakin efisien.

KATA PENUTUP

Puji syukur penulis panjatkan pada hadirat Tuhan Yang Maha Esa. Atas rahmat dan kemudahan yang diberikan kepada penulis, sehingga makalah yang berjudul “Shuffle Cipher” dapat diselesaikan. Penulis ingin menyampaikan rasa syukur dan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., selaku Dosen Mata Kuliah IF4020 Kriptografi yang telah memberikan banyak masukan, referensi, dan dukungan.

REFERENCES

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/13-Block-Cipher-Bagian1-2023.pdf

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/17-Beberapa-block-cipher-bagian3-2023.pdf

[3] https://mackenziefretty.wordpress.com/2012/06/19/factor-tree/