Taiji Block Cipher

Dendy Suprihady - 13514070
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514070@std.stei.itb.ac.id

Kode sumber tersedia di: https://github.com/dendyliu1306/TaijiChiper

***Abstrak*— Chiper blok adalah teknik yang banyak digunakan saat ini dalam algoritma kriptografi modern yang terstandarisasi. Keunggulan chiper blok yang melakukan operasi pada blok bit atau blok byte memungkinkan perancang algoritma untuk merancang algoritma yang rumit dan sulit dipecahkan. *Taiji* chiper merupakan salah satu algoritma chiper blok yang mengimplementasikan jaringan feistel dan memiliki 16 byte kunci yang beroperasi pada 16 byte blok data. Nama *taiji* terinsipirasi dari terminologi kosmologi dari negara Cina yang memiliki arti keesaan yang mendasari munculnya konsep *Yin-and-Yang*.**

Keywords—Kriptografi; Block cipher; Feistel; Taiji; Yin-and-Yang.

I. PENDAHULUAN

Kriptografi merupakan sebuah seni atau ilmu yang digunakan untuk mengamankan suatu data atau pesan yang tidak ingin diketahui maknanya selain oleh sang penerima pesan. Kriptografi sudah dikenal sejak jaman sebelum masehi seperti misalnya pada jaman mesin kuro yang menggunakan simbol untuk sebuah pesan, jaman Yunani dan romawi kuno yang melahirkan salah satu algoritma terkenal yaitu *Caesar Chiper*, dan pada jaman arab kuno yang meinumbalkan banyak kemajuan pada bidang kriptografi.

Kriptografi juga memilki peranan penting dalam sejarah kehidupan manusia seperti pada jaman renaisan, peristiwa hukuman mati kepada ratu Mary dari Skotlandia akibat terpecahkannya pesan yang berisi rencana pembunuhan ratu inggris, dan yang paling terkenal adalah pada jaman perang dunia kedua dimana Alan Turing berhasil memecahkan Enigma chiper sehingga memperpendek perang dunia ke-2 [1].

Pada saat ini kriptografi masih sangat berperan dalam menyediakan pelayanan kemanan dalam pertukaran pesan. Terutama pada penggunaan internet yang semakin meluas dan tingkat ketergantungan yang tinggi terhadap penggunaan internet untuk mengirim dan menerima informasi. Perkembangan pada era internet ini ternyata tidak hanya memberikan akses informasi yang cepat dan mudah tetapi juga kemudahan untuk mencuri atau menyadap suatu informasi yang ditransimisikan melalui internet. Munculnya era kriptografi modern ini dikarenakan kebutuhan pengamanan pesan yang jauh lebih kompleks dan dapat diterapkan pada pesan yang ditukarkan melalui internet.

Blok chiper merupakan salah satu teknik yang banyak diterapkan pada algoritma kriptografi modern seperti AES (*Advanced Encryption Standard)* dan DES (*Data Encryption Standard)* yang menjadi standarisasi untuk beberapa proses enkripsi suatu data. Teknik chiper blok merupakan teknik operasi yang mengoperasikan chiper untuk setiap blok bit atau blok byte daari sebuah pesan, pengoperasian yang dilakukan untuk setiap blok dari suatu data ini memungkinkan para perancang algoritma untuk merancang chiper yang lebih kompleks dan susah dipecahkan.

Taiji Chiper merupakan algoritma chiper blok yang terinspirasi dari algoritma DES dan AES, dimana algoritma ini menggunakan dan mengimprovisasi beberapa teknik yang digunakan dalam kedua algoritma tersebut, misalnya:

1) Pengunaan jaringan Feistel seperti pada DES.
2) Pengunaan fungsi ekspansi dan kompresi seperti pada algoritma DES namun dibuat lebih kompleks.
3) Beberapa fungsi dari algoritma AES yang diimprovisasi menjadi lebih kompleks
4) Penggunaan *round constants* sepert pada AES untuk *key-scheduling*

Penjelasan algoritma secara mendetil akan dibahas dalam makalah ini, untuk selanjutnya, makalah akan disusun dengan susunan sebagai berikut. Bagian dua akan membahas dasar teori yang relevan dengan pekerjaan yang dilakukan. Bagian tiga akan membahas rancangan detail dari algoritma Taiji Chiper. Bagian empa akan membahas hasil eksperimen penggunaan algoritma Taiji Chiper. Bagian kelima akan merupakan kesimpulan dari hasil eksperimen.

II. DASAR TEORI

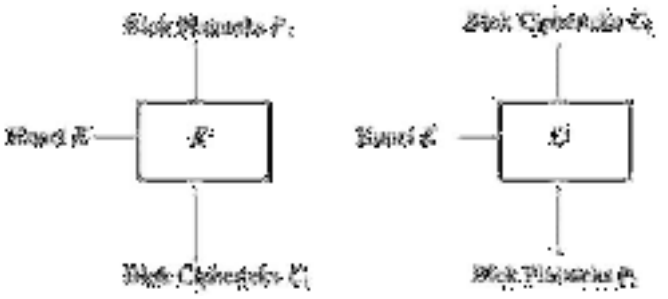
A. Block Chiper

Block cipher adalah sebuah teknik operasi yang digunakan dalam kriptografi modern dimana pada *block chiper*, pesan dibagi menjadi sekumpulan blok n-bit yang kemudian akan dipetakan dengan fungsi chiper menjadi sebuah blok chiper, hal ini diulang untuk semua blok dari data yang ada, hal ini berbeda dengan jenis teknik operasi kriptografi modern yang lain yaitu *stream chiper* yang melakukan operasi untuk 1 bit.

Dalam *block chiper* terdapat beberapa mode untuk pengoperasian pada tiap blok, yaitu:

1) Electronic Codebook (ECB)

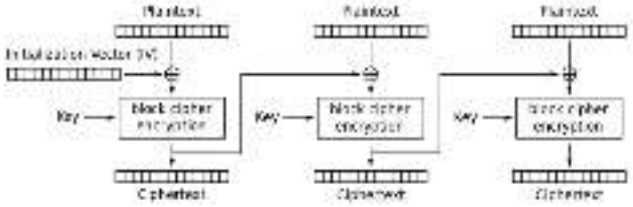
Pada mode ECB , setiap blok dioperasikan secara independent sehingga setiap blok dapat bisa dioperasikan sebara tidak terurut.



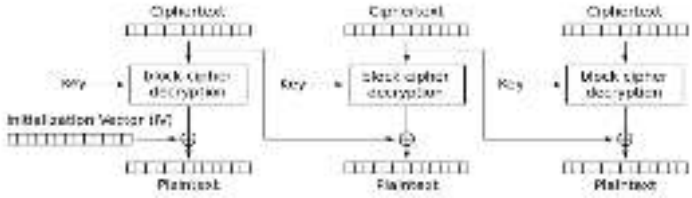
Gambar 1. Skema enkripsi dan dekripsi dari mode operasi ECB [2].

2) Cipher Block Chaining (CBC)

Mode operasi CBC setiap hasil enkripsi blok plain teks bergantung pada hasil enkripsi blok sebelumnya [2].



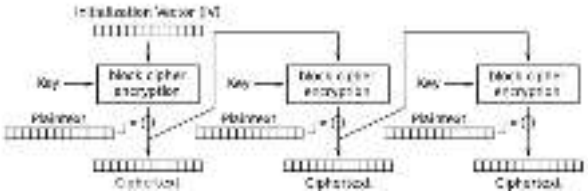
Gambar 2. Skema enkripsi mode operasi CBC [5].



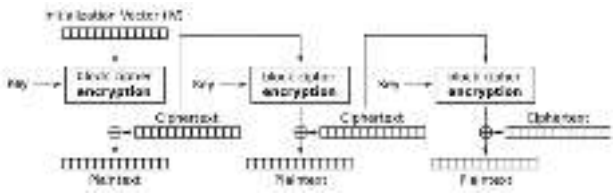
Gambar 3. Skema dekripsi mode operasi CBC [5].

3) Cipher Feedback (CFB)

Mode operasi CFB melakukan enkripsi dan dekripsi yang bergantung dari blok sebelumyna sama seperti CBC namun perbedaanya adalah CFB dapat mengenkripsi data unti yang lebih kecil dari ukuran blok.



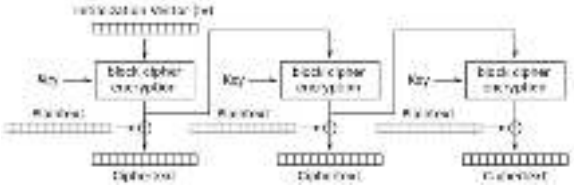
Gambar 4. Skema enkripsi dari mode operasi CFB [5].



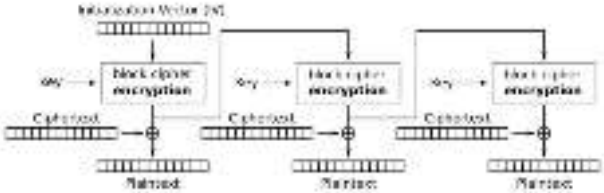
Gambar 5. Skema dekripsi dari mode operasi CFB [5].

4) Output Feedback (OFB)

Mode operasi OFB kurang lebih sama dengan mode operasi CFB, yang berbeda adalah hasil enkripsi yang dipakai sebagai posisi paling kanan di antrian [2].



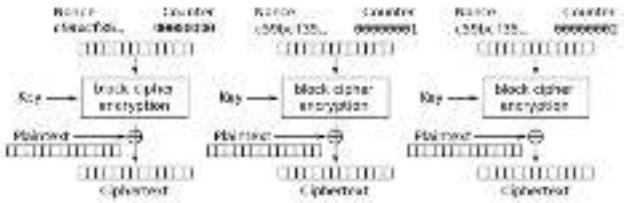
Gambar 6. Skema dekripsi dari mode operasi OFB [5].



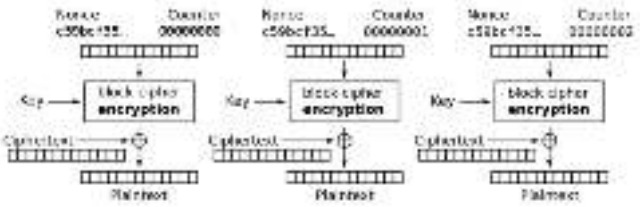
Gambar 7. Skema dekripsi dari mode operasi OFB [5].

5) Counter Mode (CM)

Mode CM tidak melakukan perantaian atau *feedback* untuk memproses blok selanjutnya, namun menggunakan suatu nilai *counter* yang nilainya dinaikan ketika akan memproses blok selanjutnya.



Gambar 8. Skema enkripsi mode operasi CM [5].



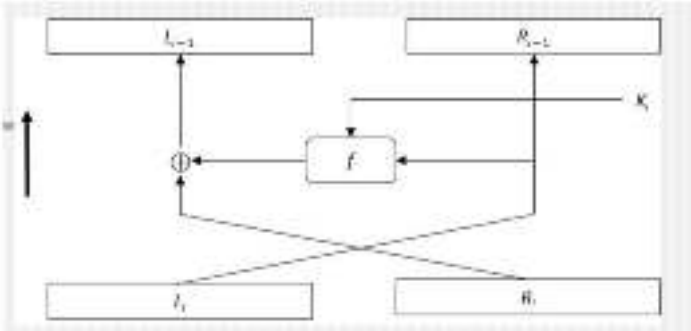
Gambar 9. Skema dekripsi mode operasi CM [5].

B. Confusion dan Diffusion

Prinsip yang diperkenalkan oleh Claud Shannon dalam makalah klasiknya, *Communication theory of secrecy system* pada tahun 1949 [2]. Prinsip ini bertujuan untuk mengatasi serangan statistik. *Confusion* merupakan prinsip yang bertujuan untuk menyembunyikan keterhubungan antara plain teks, chiper teks, dan kunci, salah satu caranya adalah menggunakan algoritma subtitusi yang kompleks. *Diffusion* adalah prinsip yang bertujuan untuk membuat pengaruh perubahan 1 bit menjadi sangat besar terhadap chiper teks yang terbentuk, salah satu caranya adalah dengan menggunakan permutasi yang kompleks.

C. Jaringan Feistel

Jaringan Feistel adalah sebuah skema yang menerapkan mekanisme chiper berulang (*iterated chiper)* dan memungkinkan kita untuk melakukan enkripsi dan dekripsi dengan algoritma chiper yang sama *(reversible)*



Gambar 10. Skema jaringan feistel [2].

III. RANCANGAN ALGORITMA

A. Ikhtisar Algoritma

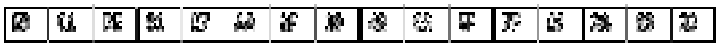
Algoritma Taiji adalah sebuah algoritma blok chiper yang menggunakan jaringan feistel dan beroperasi dalam blok *byte*. Algoritma ini terinspirasi dari DES dan AES yang dimodifikasi dengan fungsi yang lebih kompleks. Spesifikasi dasar dari Taiji chiper, yakni:

1) Panjang kunci 128 bit / 16 byte
2) Panjang blok data 128 bit / 16 byte
3) Menggunakan jaringan feistel dengan 13 putaran.
4) Beroperasi dalam *byte.*

B. Variable

Bagian ini mendekripsikan *variable* yang digunakan dalam algoritma Taiji.

1) Initialization Vector (IV)



Gambar 11. Blok initialisasi

2) Round Constant (Rcon)

Konstanta matriks berukuran 2 x 4 yang digunakan dalam pembangkitan *round keys* dari kunci eksternal.

Tabel 1. *Round constant*

67	69	DD	1D
41	F4	15	57

3) Subtitution Box (S-Box)

Taiji chiper memiliki 4 kotak subtitusi.

Gambar 12. Kotak Subtitusi-0

Gambar 13. Kotak Subtitusi-1

Gambar 14. Kotak Subtitusi-2

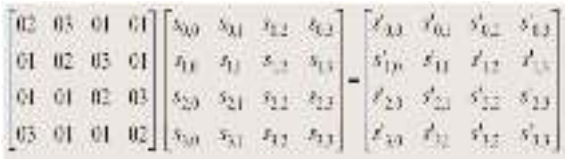
Gambar 15. Kotak Subtitusi-3

C. Fungsi Transformasi

Bagian ini menjelaskan fungsi – fungsi transformasi yang digunakan dalam algoritma Taiji pada bagian *Round Function* dan *Key Scheduling*.

1) MixCol

Fungsi transformasi yang juga digunakan oleh algoritma AES. Fungsi mengkalikan matriks *state* dengan suatu matriks pengacak.



Gambar 16. Transformasi *MixCol* AES [4]

2) *ShiftRow*

Fungsi transformasi ini melakukan pergeseran secara siklik untuk setiap baris dari matriks *state*. Aturan pergeseran adalah setiap baris dari i = (0..N), dimana N adalah jumlah baris, akan dilakukan pergeseran ke kanan sebesar nilai desimal dari *byte* pada posisi:

$$r, c = (i - 1)\ mod\ N, (i - 1)\ mod\ N$$

3) *SubExp*

Fungsi transformasi ini adalah fungsi transformasi kompleks yaxng melakukan subtitusi dan ekspansi dari sebuah larik dengan ukuran (1 x 4) menjadi sebuah matriks berukuran (4 x 4). Aturan fungsi transformasi adalah sebagai berikut:

- Misal masukan larik *state* adalah:

S_0	S_1	S_2	S_3

- Maka matriks hasil transformasi adalah:

Tabel 2. Fungsi transformasi *SubExp*

$S'_{0,0} = Sub(S_0, 0)$	$S'_{0,1} = Sub(S_1, 1) + S'_{0,0}$	$S'_{0,2} = Sub(S_0, 1)$	$S'_{0,3} = Sub(S_1, 2) + S'_{0,2}$
$S'_{1,0} = Sub(S_2, 2) + S'_{0,0}$	$S'_{1,1} = Sub(S_3, 3) + S'_{0,0}$	$S'_{1,2} = Sub(S_2, 3) + S'_{0,2}$	$S'_{1,3} = Sub(S_3, 0) + S'_{0,2}$
$S'_{2,0} = Sub(S_0, 2)$	$S'_{2,1} = Sub(S_1, 3) + S'_{2,0}$	$S'_{2,2} = Sub(S_0, 3)$	$S'_{2,3} = Sub(S_1, 0) + S'_{2,2}$
$S'_{3,0} = Sub(S_2, 0) + S'_{2,0}$	$S'_{3,1} = Sub(S_3, 1) + S'_{2,0}$	$S'_{3,2} = Sub(S_2, 1) + S'_{2,2}$	$S'_{3,3} = Sub(S_3, 2) + S'_{2,2}$

Fungsi *Sub(val, i)* adalah fungsi yang melakukan subtitusi *val* dengan S-Box$_i$. Misal *Sub(2D, 2)* akan mensubtitusim nilai 2D (format *hexadecimal)* menggunakan S-Box$_2$ yang akan menghasilkan nilai hasil subtitusi yaitu 8E.

4) *SubComp*

Fungsi transformasi yang akan melakukan subtitusi dan kompresi pada sebuah matriks *state* berukuran (4 x 4) menjadi sebuah larik berukuran (1 x 4). Aturan fungsi transformasi adalah sebagai berikut:

- Misal masukan matriks *statei* adalah:

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0} =$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0} =$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

- Maka larik hasil trasnformasi adalah:

Tabel 3. Fungsi transformasi *SubComp*

$S'_0 = Sub(S'_{0,0}, 0) \oplus Sub(S'_{0,1}, 1) \oplus Sub(S'_{1,0}, 2) \oplus Sub(S'_{1,1}, 3)$
$S'_1 = Sub(S'_{0,2}, 1) \oplus Sub(S'_{0,3}, 2) \oplus Sub(S'_{1,2}, 3) \oplus Sub(S'_{1,3}, 0)$
$S'_2 = Sub(S'_{2,0}, 2) \oplus Sub(S'_{2,1}, 3) \oplus Sub(S'_{3,0}, 0) \oplus Sub(S'_{3,1}, 1)$
$S'_3 = Sub(S'_{2,2}, 3) \oplus Sub(S'_{2,3}, 1) \oplus Sub(S'_{3,2}, 1) \oplus Sub(S'_{3,3}, 2)$

C. Key Scheduling

Dari kunci eksternal sepanjang 128 bit (16 *byte*) akan dibangkitkan 26 *round keys* sepanjang 128 bit (16 *byte*) untuk 13 putaran, setiap putaran dibutuhkan 2 *round keys*. Proses pembangkitan kunci dibagi menjadi 2 bagian, pembangkitan 13 kunci pertama (Rk^l) dan pembangkitan 13 kunci kedua (Rk^r).

- Pembangkitan Rk^l

Kunci eksternal *K* sepanjang 16 *byte* akan disalin ke dalam matriks *W* berukuran (4 x 4) dengan aturan:

Tabel 4. Matriks W^l pada pembangkitan Rk^l

$Sub(K_0, 0)$	$Sub(K_1, 0)$	$Sub(K_2, 0)$	$Sub(K_3, 0)$
$Sub(K_4, 1)$	$Sub(K_5, 1)$	$Sub(K_6, 1)$	$Sub(K_7, 1)$
$Sub(K_8, 2)$	$Sub(K_9, 2)$	$Sub(K_{10}, 2)$	$Sub(K_{11}, 2)$
$Sub(K_{12}, 3)$	$Sub(K_{13}, 3)$	$Sub(K_{14}, 3)$	$Sub(K_{15}, 3)$

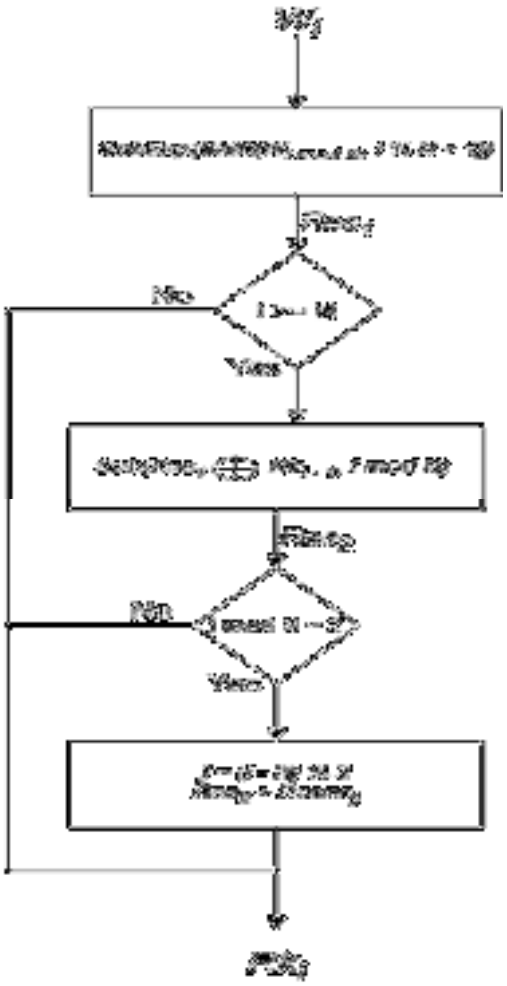
- Pembangkitan Rk^r

Kunci eksternal *K* sepanjang 16 *byte* akan disalin ke dalam matriks *W* berukuran (4 x 4) dengan aturan:

Tabel 5. Matriks W^r pada pembangkitan Rk^r

$Sub(K_0, 3)$	$Sub(K_1, 3)$	$Sub(K_2, 3)$	$Sub(K_3, 3)$
$Sub(K_4, 2)$	$Sub(K_5, 2)$	$Sub(K_6, 2)$	$Sub(K_7, 2)$
$Sub(K_8, 1)$	$Sub(K_9, 1)$	$Sub(K_{10}, 1)$	$Sub(K_{11}, 1)$
$Sub(K_{12}, 0)$	$Sub(K_{13}, 0)$	$Sub(K_{14}, 0)$	$Sub(K_{15}, 0)$

Setelah matriks W dibentuk maka untuk setiap i = 0 … 11 dilakukan:



Gambar 17. Skema *key scheduling*

Untuk *final round* atau i = 12, $Rk_{12} = MixCol(V)$ dimana V adalah hasil operasi xor dari semua nilai Rk_0 ... Rk_{11}. Proses ini dilakukan 2 kali untuk W^l dan W^r.

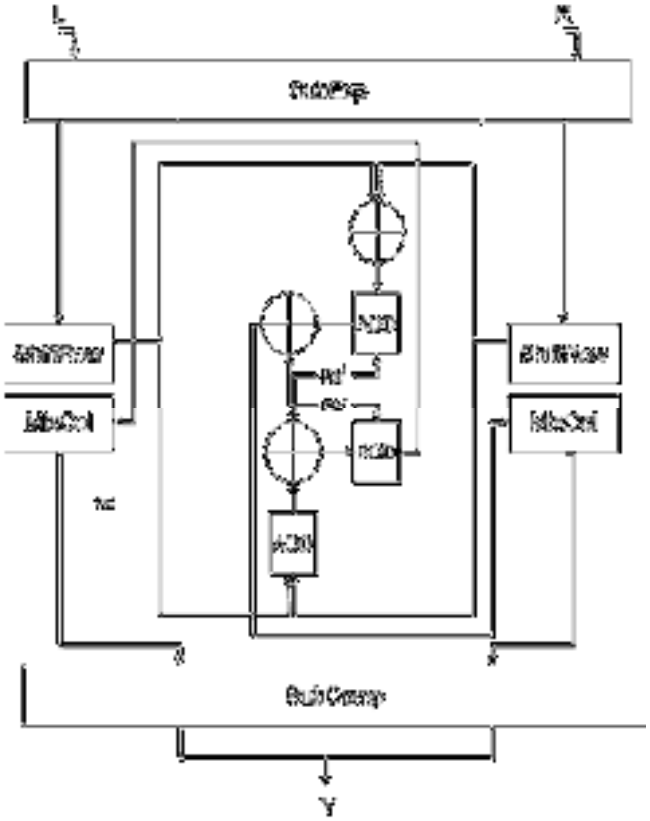
Keterangan:

- N = 4
- $W = W^l$, $Rcons = Rcons[0]$ untuk Rk^l
- $W = W^r$, $Rcons = Rcons[1]$ untuk Rk^r.
- *Shift(A, n)* = fungsi menggeser vector A kekanan secara siklik sejauh n.
- *Sub(A, n)* = fungsi untuk melakukan subtitusi semua nilai pada vector A dengan menggunakan S-Box_n.
- *Rcons* adalah matriks *round constants* yang berukuran (2 x 4), Rcons[i] menyatakan baris ke-i pada matriks.
- Operasi a % b pada skema adalah *floor(*a div b).

Proses *key scheduling* akan menghasilkan 13 Rk^l dan 13 Rk^r yang dimana akan dipakai sebagai *round keys* untuk 13 ronde putaran. Tiap putaran ke-i akan menggunakan 2 kunci yaitu Rk^l_i dan Rk^r_i.

D. Round Function

Taiji chiper menggunakan jaringan feistel dengan 13 putaran yang dimana memiliki *round function* yang akan dioperasikan untuk enkripsi dan dekripsi pada setiap ronde putaran. *Round function* dari taiji chiper adalah sebagai berikut:



Gambar 18. *Taiji Round Function*

Keterangan proses:

1) Ukuran blok yang diproses adalah 16 *byte*.
2) Ketika memasuki jaringan feistel maka blok data akan dibagi menjadi 2 blok dengan ukuran 8 *byte* untuk setiap blok (lihat **Gambar 10** untuk lebih detil).
3) Salah satu blok 8 *byte* yang akan memasuki *round function* akan dibagi menjadi 2 blok *L* dan *R* dengan ukuran 4 *byte.*
4) Blok *L* dan *R* dengan ukuran 4 *byte* masing – masing akan ditransformasi dengan fungsi *SubExp* menjadi matriks dengan ukuran (4 x 4) atau total 16 *byte*.
5) Kedua matriks akan ditransformasi dengan fungsi *ShiftRow* menghasilkan matriks *ML* dan *MR*.
6) $ML' = ((ML + MR) \oplus Rk^l) + Rk^r$
   $MR' = ((ML \oplus MR) + Rk^l) \oplus Rk^r$
7) Matriks *ML'* dan *MR'* ditransformasi dengan fungsi *MixCol.*
8) Hasil transformasi *MixCol* akan ditransformasi lagi dengan fungsi *SubComp* yang bertujuan untuk mengkompresi matriks berukuran (4 x 4) menjadi sebuah blok dengan ukuran 4 *byte*.
9) Hasil transformasi *SubComp* akan digabung membentuk blok *Y* dengan ukuran 8 *byte*.

IV. Hasil dan Simulasi

Untuk implementasi algoritma Taiji chiper digunakan bahasa pemrograman *python*. Mode blok chiperyang akan disimulasikan adalah mode ECB dan CBC. Untuk mode CBC akan digunakan *initialization vector* yang terlampir pada **Gambar 11.**

Tabel 6. Hasil ujicoba Taiji Chiper mode ECB

Kunci
kriptografitaiji
Plainteks
A brilliant mathematician and cryptographer Alan was to become the founder of modern-day computer science and artificial intelligence; designing a machine at Bletchley Park to break secret Enigma encrypted messages used by the Nazi German war machine to protect sensitive commercial, diplomatic and military communications during World War 2.
Chiperteks Mode ECB (Dalam *Hexadecimal)*
2c867fb60770ef1a54ec0ea8aa773b3e332933ee2f87d3bef746a717c696418f8057e48a03d91719c3d3b0397b666860b8bfa58ccdb0695489fabcc76932e07d849be96d72ecff833535445afd52d8c4360f6a60134625dc191e460ce4a9f0c438c477a9538919fd30904980873e8549f0aec526a7b436fba50bafd924e2b9a9f1e65043e583a7890da58ff4b5b923a9d209aedc86f52287ece2a9f4aa69314f0bb95ef8787a11eeacaaa0d7c00c416abf92fdf84adbcfee3a86856c971a6163d1e59fe8cdb47585a7fd5720a08a254eba694736c456a80a99a3afc96772a634ad77c1ddff128552e0bcfc7c16b1a43cc7ae7cafc026e8d73e4dc12bb38c9accd911fda9c6df21bfdbb3b318cfcc30158fc42d9469096ebccc7d73617e0ca02cf8fbd6b8733f958b357daaecb051ff511aaad39eb2dd7b17261df93bf5e4ea42b4dc1d0b44aac72a5c5f940483820bcceb6142d7a5422c9685d5fcef9a209b3c
Chiperteks Mode CBC (Dalam *Hexadecimal)*
62156905e67056256a8cbdc4196b55c60c99f9061e7f1f90f8b717f4a295b1657b97359f828be1c15a6cecce6ac3f67b267730afd62bd8916433f60d9f5fc3965569c481ef8d59d223e79e847cc29a2944eda9297e00ea0eb39bc268a68b4156709163c34023e9f4d9fd6771f75dfb12b22f7325b34c6bf37f2e733d57d6ada8378681786f9f17f6bcc9321884a8f1c44e15e42fbd26722250a175bf0d68805b243fffbfafbd607f9fa5877ad0a3387cad8a75268bee5e3ddc3a8d1357281aaa61b70cc8db8970917118ce22eae0e6839fd0d19d17da8fc0ddfe8936f915ee9626c3bc0fb799cd3bf36bc46df3ce0a1b334ba707976f2c24f4c652af2df9c87c6ed8e266a3f4dba295d2e853327b7d85c5b8b2bf858bb03169736e6814fe5630037e69b3874a33eebe0553c0a3f478667fe730547742b05300451d6bf4247f199a3c1594db8851d86fc15acd7edd79d84b139244d8844e68bc42c0f4b375f3f0

Dari hasil enkripsi 2 mode diatas dapat dilihat perbedaan yang sangat signifikan antara kedua hasil enkripsi. Pada Taiji chiper jika jumlah *byte* data yang akan diproses bukan merupakan kelipatan 16 maka akan dilakukan *null padding* yakni menambah *byte* dengan nilai 0000 0000.

V. Analisis Keamanan

A) Analisis Confusion dan Diffusion

Prinsip *confusion* dan *diffusion* merupakan prinsip dasar yang dapat mengukur seberapa rumit dan sulitnya suatu algoritma chiper untuk dipecahkan.

Pada Taiji chiper prinsip *confusion* diterapkan dengan menggunakan subtitusi kompleks yang dilakukan oleh fungsi transformasi *SubExp* dan *SubComp* yang juga menggunakan 4 kotak S-Box berbeda seperti pada diterangkan pada bab III. Untuk prinsip *diffusion*, diterapkan algoritma permutasi kompleks dengan menggunakan fungsi transformasi *ShiftRow* dan *MixCol*. Hasil eksperimen untuk melakukan uji *confusion dan diffusion* adalah sebagai berikut:

- Data ujicoba

Gambar 19. Citra lena *grayscale*

Menggunakan citra lena *grayscale* dengan ukuran 48077 *byte* yang akan dienkripsi menggunakan kunci = 'kriptografitaiji' sepanjang 16 *byte*.

- Hasil ujicoba

Tabel 7. Hasil ujicoba pergantian 1 bit kunci

Kunci	Jumlah Kesamaan *Byte*	Persentase Perubahan Chiperteks
kriptografitai**k**i	166 / 48077	99.65
kript**p**grafitaiji	208 / 48077	99.57
kripto**h**rafitaiji	202 / 48077	99.58
lriptografitaiji	196 / 48077	99.59
kriptogr**b**fitaiji	182 / 48077	99.62
Rata – Rata	190	99.60

Tabel 8. Hasil ujicoba pergantian 1 bit plainteks

Posisi Pergantian Bit	Jumlah Kesamaan *Byte*	Persentase Perubahan Chiperteks
20560	189 / 48077	99.61
2539	206 / 48077	99.57
18532	171 / 48077	99.64
34701	150 / 48077	99.69
25542	200 / 48077	99.58
Rata – Rata	183.2	99.62

Dari hasil ujicoba dapat dilihat bahwa pergantian 1-bit baik pada kunci maupun pada plainteks akan memberikan chiperteks yang jauh berbeda, dengan persentase perubahan rata-rata sebesar > 99.60%. Hal ini mengindikasikan bahwa prinsip *confusion* dan *diffusion* telah terimplementasi dengan baik.

B) Analisis Brute Force Attack

Taiji chiper menggunakan kunci sepanjang 128-bit atau 16 byte dimana untuk penyerangan dengan metode *brute force* maka kombinasi kunci yang mungkin adalah sebanyak

$$2^{128} = 3{,}4\,x\,10^{38}\ kemungkinan$$

Misal kita memiliki computer tercepat yang dapat mencoba 1 juta kemungkinan dalam 1 detik, maka dibutuhkan 5,4 x 10^{24} tahun untuk dapat menemukan kunci yang tepat.

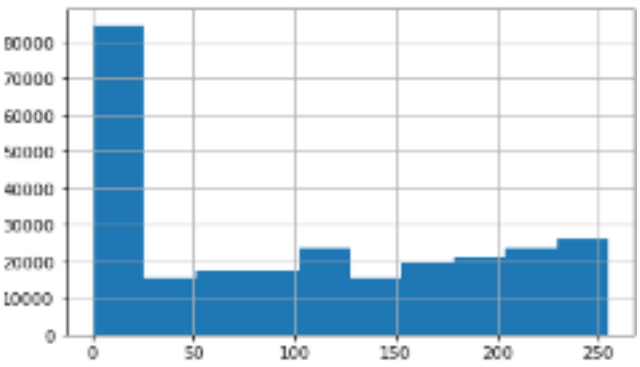
C) Analisis Statistik

Untuk analisis statistik akan dicoba analisis frekuensi byte dari suatu plainteks yang akan dibandingkan dengan analisis frekuensi dari hasil enkripsinya. Untuk ujicoba akan digunakan 2 mode operasi blok chiper yaitu ECB dan CBC.

- Data ujicoba

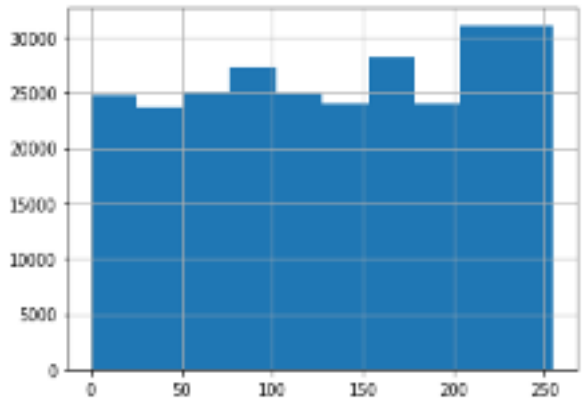
Gambar 20. Citra analisis frekuensi.

Menggunakan kunci = 'kriptografitaiji' dan sebuah citra *grayscale* yang memiliki histogram tidak merata dan terkonsentrasi pada nilai pixel yang gelap.

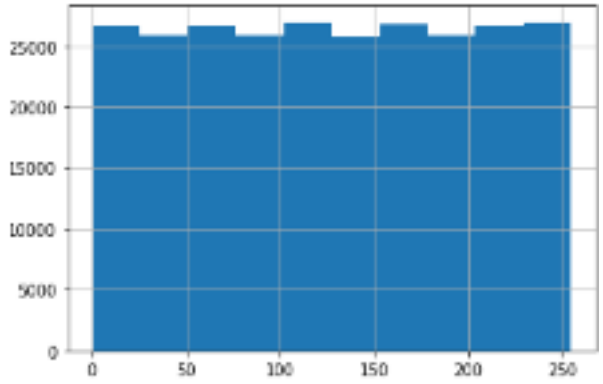


Gambar 21. Histogram dari frkuensi nilai *byte* plainteks

- Hasil ujicoba



Gambar 22. Histogram dari frkuensi nilai *byte* chiper ECB



Gambar 23. Histogram dari frkuensi nilai *byte* chiper CBC

Dari hasil ujicoba diatas dapat dilihat bahwa meskipun plainteks memiliki konsentrasi frekuensi yang lebih tinggi pada suatu nilai *byte* namun hasil dari enkripsi plainteks tersebut akan menghasil distribusi frekuensi yang merata sehingga analisis statistik akan sulit dilakukan.

VI. KESIMPULAN DAN SARAN

A) Kesimpulan

Algoritma Taiji Chiper sudah memberikan hasil yang baik pada beberapa analisis keamanan seperti prinsip *confusion* dan *diffusion,* penangan terhadap serangan *brute force*, dan juga penanganan terhadap analisa statistik. Hal ini diperoleh karena penggunaan fungsi transformasi yang kompleks seperti *SubExp* dan *SubComp* yang dimana menggunakan 4 S-Box untuk melakukan subtitusi sekaligus melakukan ekspansi dan kompresi.

B) Saran

Dalam paper ini belum dicantumkan beberapa hal penting seperti waktu eksekusi dan pemakaian memori dari algoritma. Beberapa optimisasi juga masih bisa dilakukan untuk mengurangi biaya komputasi.

REFERENSI

[1] Munir Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi: Pengantar Kriptografi

[2] Munir Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi: Kriptografi Modern

[3] Munir Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi: Review Algoritma Kriptografi Modern

[4] Munir Rinaldi. 2019. Slide Kuliah IF4020 Kriptografi: Advanced Encryption Standard (AES)

[5] NIST Computer Security Division's (CSD) *Security* Technology Group (STD), 2013, Block Chiper modes