BAB II
TINJAUAN PUSTAKA

2.1 Kajian Terkait

Aziiz & Pakereng (2020) melakukan penelitian terkait perancangan teknik kriptografi block cipher berbasis pola batik ceplok yogyakarta. Penelitian tersebut merancang kriptografi block cipher 64 bit dengan 10 putaran, dimana setiap putaran terdapat 4 proses. Pada setiap putaran terdapat 4 pola untuk proses plaintext dan 4 pola untuk proses kunci. Di proses kedua dan keempat ditransformasikan dengan tabel S-BOX untuk mendapatkan ciphertext yang lebih acak. Pengujian dilakukan menggunakan Avalanche Effect dan nilai Korelasi dimana rata-rata perubahan karakter mencapai 47,656%.

Prihanto & Pakereng (2019) melakukan penelitian terkait perancangan teknik kriptografi block cipher berbasis pola tarian sajojo papua. Penelitian tersebut merancang kriptografi block cipher 64 bit dengan 10 putaran, dimana setiap putaran terdapat 4 proses. Pada setiap putaran terdapat 4 pola untuk proses plaintext dan 4 pola untuk proses kunci. Di proses kedua dan keempat ditransformasikan dengan tabel S-BOX untuk mendapatkan ciphertext yang lebih acak. Pengujian dilakukan menggunakan Avalanche Effect dan nilai Korelasi dimana rata-rata perubahan karakter mencapai 49,69%.

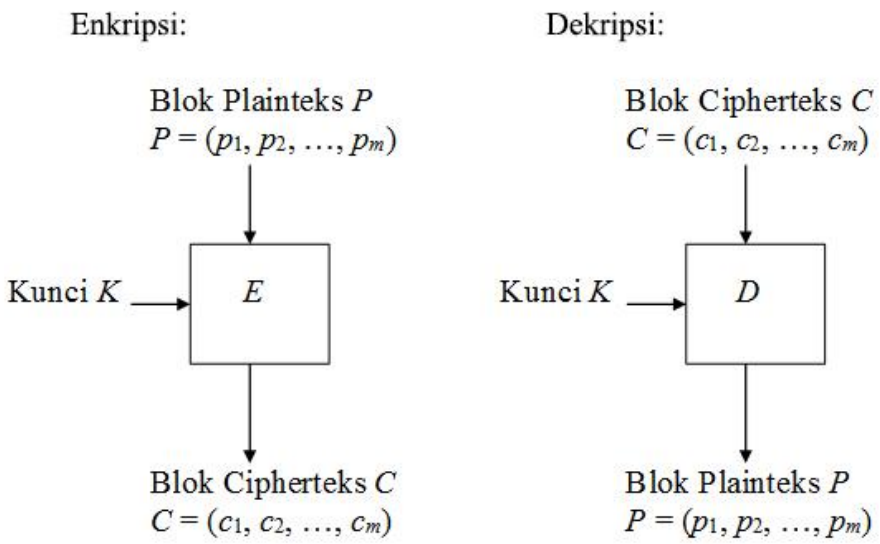
Fauzi & Wellem (2021) melakukan penelitian terkait perancangan teknik kriptografi block cipher berbasis pola dribbling practice. Dalam perancangan block cipher ini, diterapkan prinsip-prinsip dalam desain block cipher seperti, kotak substitusi (substitution box), operasi XOR, jaringan fiestel, dan transposisi kunci. Hasil pengujian block cipher menghasilkan nilai AE sebesar 54.687%.

Mahmoud dkk. (2013) dalam penelitian yang berjudul "Dynamic AES-128 with Key-Dependent S-box" melakukan penelitian yang bertujuan untuk merancang AES-128 dengan S-box dinamis yang bergantung pada kunci rahasia. Penelitian tersebut merancang dan mengimplementasikan AES-128 dengan S-box dinamis. Penelitian menyajikan pendekatan baru untuk menghasilkan AES dinamis, dimana struktur S-box pada dasarnya akan berubah untuk setiap perubahan kunci

rahasia. Parameter dari S-box yang baru dibuat memiliki karakteristik yang sama dengan yang ada di algoritma asli AES. Penelitian juga melakukan percobaan untuk menguji kualitas dari hasil perancangan dengan membandingkan AES S-box dinamis yang dihasilkan dengan AES S-box standar. Hasil perbandingan performa mengarah kepada peningkatan keamanan untuk AES S-box dinamis.

2.2 Block Cipher

Block Cipher adalah algoritma dalam teknik kriptografi yang beroperasi dengan membagi bit-bit plainteks menjadi blok dengan panjang yang sama. Block Cipher bekerja dengan menggabungkan perhitungan operasi sederhana seperti XOR atau substitusi yang dilakukan dalam beberapa putaran untuk meningkatkan keamanan pesan.



Gambar 2. 1 Skema enkripsi dan dekripsi block cipher

Gambar diatas menunjukkan skema proses enkripsi dan dekripsi pada algoritma block cipher. Blok plaintext (P) yang berukuran panjang m bit dinyatakan sebagai

$$P = (\mathrm{p1}, \mathrm{p2}, \ldots \mathrm{pm}) \tag{2.1}$$

Blok ciphertext (C) dinyatakan sebagai

$$C = (c1, c2, \dots cm) \quad (2.2)$$

Kunci (K) dinyatakan sebagai

$$K = (k1, k2, \dots km) \quad (2.3)$$

Proses enkripsi adalah

$$EK(P) = C \quad (2.4)$$

Dan proses dekripsi adalah

$$DK(C) = P \quad (2.5)$$

Suatu rancangan kriptografi harus melalui uji kriptosistem yaitu kondisi kriptografi harus memenuhi lima-tupel (five-tuple) (P,C,K,E,D) dengan kondisi (Stinson & Paterson, 2019) :

1. P adalah himpunan berhingga dari Plaintext
2. C adalah himpunan berhingga dari Ciphertext
3. K merupakan ruang kunci (keyspace), adalah himpunan berhingga dari kunci
4. Untuk setiap $k \in K$ terdapat aturan enkripsi $ek \in E$ dan berkorespodensi dengan aturan dekripsi $dk \in D$. Setiap $ek: P \rightarrow C$ dan $dk: C \rightarrow P$ adalah fungsi sedemikian hingga $dk\big(ek(x)\big) = x$ untuk setiap plaintext $x \in P$

Di dalam proses Enkripsi maupun Dekripsi Block cipher menggunakan operasi XOR dimana output yang dihasilkan dari proses enkripsi akan susah ditebak, karena apabila kita melihat dasar dari XOR seperti berikut :

- 0 XOR 0 = 0
- 0 XOR 1 = 1

- 1 XOR 0 = 1
- 1 XOR 1 = 0

Operasi XOR diatas menunjukkan untuk mendapatkan hasil output 0 maka input nya tidak dapat diketahui, bisa jadi input yang diberikan adalah 1 atau 0. Dasar tersebut digunakan untuk melakukan kriptografi block cipher.

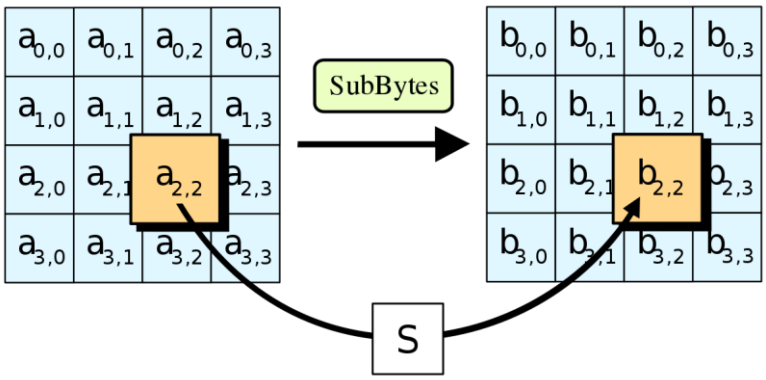
2.2.1 Confusion dan Difusion

Dua prinsip dasar pada desain block cipher adalah confusion dan diffusion. Confusion menyembunyikan hubungan antara plaintext dan ciphertext. Confusion dapat dihasilkan melalu transformasi substitutsi. Diffusion berarti penyebaran pengaruh suatu bit plaintext atau kunci untuk menyembunyikan karakteristik statistik plaintext. Dengan kata lain, perbedaan satu bit masukan plaintext atau kunci menyebabkan perubahan yang signifikan. Cara sederhana untuk menghasilkan diffusion adalah permutasi plaintext pada level tertentu (byte/bit). Operasi-operasi sederhana, seperti substitusi dan permutasi, bila dilakukan berkali-kali pada suatu blok plaintext dapat menghasilkan confusion dan diffusion yang baik. Hal inilah yang mendasari desain iterated block cipher pada suatu block cipher.

Shannon C. E. (1949) pertama kali memperkenalkan kedua prinsip ini melalui karyanya yang berjudul “Communication Theory of Secrecy System”. Tujuan utama diperkenalkan prinsip ini adalah mempersulit kriptanalis dalam memecahkan cipherteks melalui metode analisis statistik. Confusion dalam konteks ini berarti proses perancangan plainteks, kunci, dan cipherteks yang memiliki hubungan yang dibuat serumit mungkin. Prinsip ini ditujukan agar kriptanalis kesulitan untuk menemukan pola - pola statistik yang mungkin muncul. Difusion merupakan prinsip yang digunakan untuk menyebarkan pengaruh dari bit plainteks atau kunci pada cipherteks. Pada block cipher yang memiliki prinsip ini, perubahan beberapa bit pada plainteks akan membuat perubahan yang tidak dapat diduga pada cipherteks.

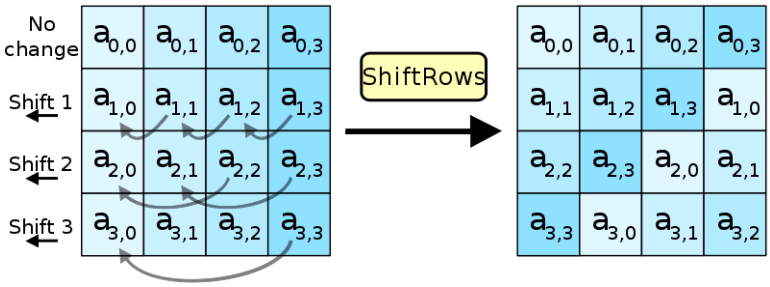
2.2.2 Advance Encryption Standard (AES)

Advance Encryption Standard (AES) adalah blok cipher kunci privat yang memproses blok data 128 bit dengan panjang kunci 128, 192, atau 256 bit. Operasi algoritma AES dilakukan pada array 4 kali 4 byte yang disebut state. State awal adalah teks biasa dan state terakhir adalah teks tersandi. AES-128 menerapkan fungsi putaran 10 kali, AES-192 menerapkan fungsi putaran 12 kali, dan AES-256 menerapkan fungsi putaran 14 kali. Setiap putaran AES membutuhkan satu kunci hasil dari generasi kunci yang menggunakan 2 transformasi yaitu subtitusi dan transformasi. Pada proses enkripsi AES mengunakan 4 transformasi dasar dengan urutan trasformasi *subbytes, shiftrows, mixcolumns,* dan *addroundkey*. Transformasi adalah operasi linier dan non-linier yang dapat dibalik untuk memungkinkan dekripsi menggunakan inversnya. Setiap transformasi memengaruhi semua byte State.



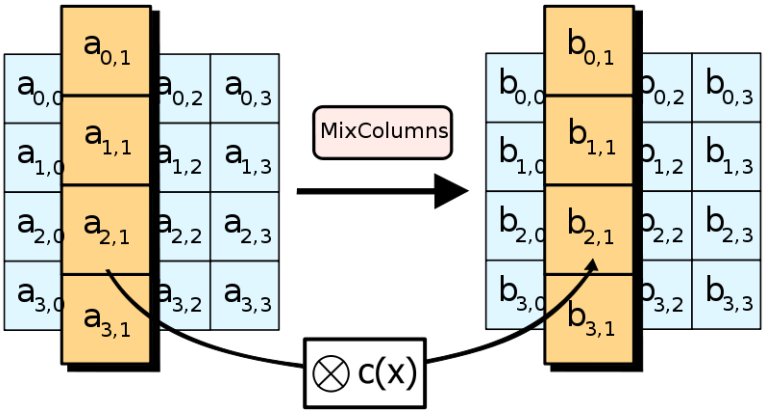
Gambar 2. 2 Transformasi subbytes

Transformasi SubBytes adalah substitusi byte nonlinier yang beroperasi pada setiap byte State menggunakan tabel (S-box). Jumlah tabel dihitung dengan inversi medan hingga diikuti dengan transformasi affine sehingga menghasilkan tabel yang disebut S-box.



Gambar 2. 3 Transformasi shiftrows

Transformasi ShiftRows adalah operasi perpindahan melingkar, yang memutar baris-baris state dengan jumlah byte (offset) yang berbeda. Offset sama dengan indeks baris: baris kedua digeser satu byte ke kiri, baris ketiga - dua byte ke kiri, baris keempat - tiga byte ke kiri, dan baris pertama - empat byte ke kiri.



Gambar 2. 4 Transformasi mixcolumns

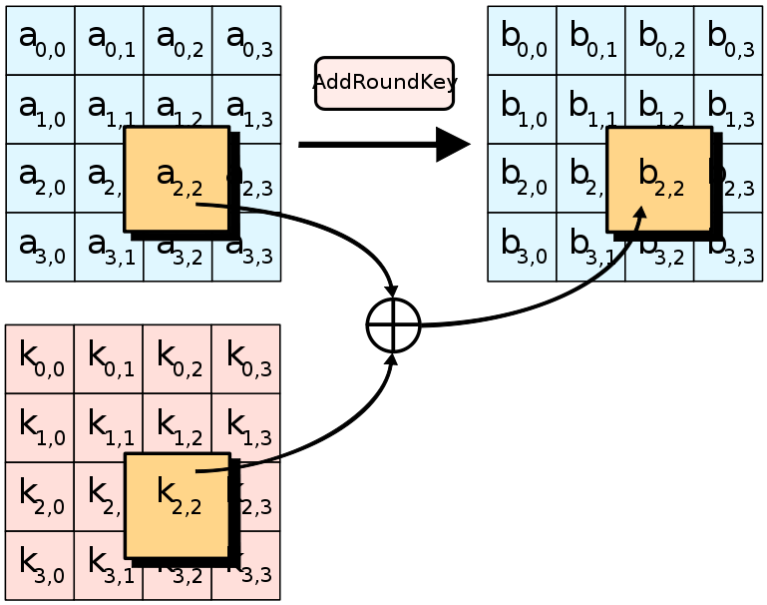
Transformasi MixColumns mencampur byte di setiap kolom dengan mengalikan State dengan matrix tetap.

$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{bmatrix}.$$

Gambar 2. 5 Matrix tetap dalam langkah transformasi mixcolumns

Dalam langkah MixColumns, empat byte dalam tiap kolom state digabung dengan transformasi linear terbalikkan. Fungsi MixColumns menerima empat byte

input dan mengeluarkan empat byte yang tiap byte inputnya saling memengaruhi. Fungsi ini memberikan penghamburan dalam penyandian.



Gambar 2. 6 Transformasi addroundkey

Transformasi AddRoundKey adalah operasi XOR yang menambahkan kunci bulat ke State di setiap putaran. Kunci bulat dihasilkan selama proses ekspansi kunci. Kunci bulat awal sama dengan kunci rahasia (Rothke, 2007).

2.2.3 Substitution Box (S-Box)

Substitution box atau disingkat S-box adalah suatu komponen kriptografi yang sering digunakan untuk melakukan substitusi, baik substitusi bit maupun substitusi byte. S-box sering digunakan untuk memenuhi properti confussion dari Shannon yang mengharuskan sebuah block cipher untuk meminimalisasi hubungan antara kunci dan ciphertext. S-box umumnya berbentuk matriks yang berfungsi seperti lookup table. Salah satu algoritma terkenal yang menggunakan S-box dalam algoritma enkripsinya adalah AES atau Rijndael.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Gambar 2. 7 S-box standard AES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Gambar 2. 8 Invers s-box standard AES

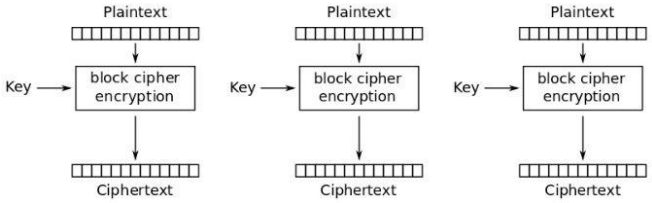
S-box memiliki tugas meminimalkan kerentanan algoritma terhadap metode kriptanalisis linear dan diferensial dan serangan aljabar. Selain persyaratan kompleksitas, fungsi S-box harus dapat dibalik serta tidak memiliki poin tetap $S(a) = a$ atau poin tetap pelengkap $S(a) = \bar{a}$, S-box juga harus dapat dieksekusi dengan cepat dan mudah untuk diimplementasikan. S-box berisi permutasi dari semua kemungkinan nilai 256 8-bit. Setiap byte state dipetakan menjadi byte baru dengan cara berikut: Paling kiri 4 bit dari byte digunakan sebagai nilai baris dan 4 bit paling kanan digunakan sebagai nilai kolom. Nilai baris dan kolom berfungsi sebagai indeks ke dalam S-box untuk memilih nilai keluaran 8-bit

yang unik. Selama enkripsi setiap nilai state diganti dengan nilai S-box yang sesuai pada tabel satu. Pada proses dekripsi, S-box harus digunakan secara terbalik. Nilai state pada proses dekripsi diganti dengan invers dari S-box seperti yang digambarkan pada tabel dua (Rothke, 2007).

2.2.4 Mode Operasi Block Cipher

Dalam kriptografi, block cipher adalah algoritma deterministik yang beroperasi pada kelompok bit dengan panjang tetap, yang disebut block, dengan transformasi yang tidak bervariasi yang ditentukan oleh kunci simetris. Cipher blok beroperasi sebagai komponen dasar penting dalam desain banyak protokol kriptografi dan banyak digunakan untuk mengimplementasikan enkripsi data massal. Terdapat lima mode operasi yang direkomendasikan NIST dalam block cipher, yaitu Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), dan Counter (CTR). Berikut merupakan penjelasan mengenai lima mode operasi yang telah disebutkan (Dworkin, 2005):

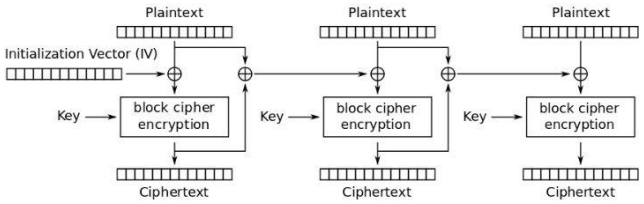
1. Electronic Code Book (ECB)



Gambar 2. 9 Operasi block cipher dengan ECB

Mode ini merupakan mode yang paling sederhana. Pesan dibagi menjadi blok - blok, kemudian setiap blok di enkripsi secara independen. Kekurangan dari metode ini adalah kurangnya prinsip difusion, hal ini dikarenakan ECB melakukan enkripsi blok plainteks menjadi blok cipher yang identik, tidak ada pola dari pesan yang disembunyikan.

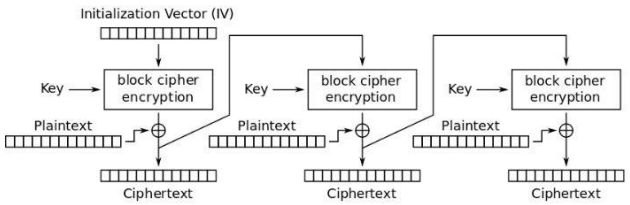
2. Cipher Block Chaining (CBC)



Gambar 2. 10 Operasi block cipher dengan CBC

Pada mode CBC, setiap blok di enkripsi dengan memanfaatkan hasil enkripsi dari block cipher sebelumnya, sedangkan block cipher paling awal di enkripsi dengan Initialization Vector. Untuk setiap block cipher hasil enkripsi di operasikan XOR dengan block cipher selanjutnya. Kekurangan utama pada mode ini adalah enkripsi yang berurutan (tidak dapat diparalelkan) dan juga pesan harus di padding sesuai dengan ukuran block cipher.

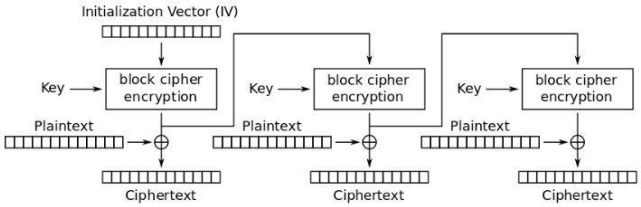
3. Cipher Feedback (CFB)



Gambar 2. 11 Operasi block cipher dengan CFB

Berbeda dari mode CBC yang menggunakan blok plaintext sebagai masukan dari fungsi enkripsi. Enkripsi pada mode CFB dilakukan dengan menggunakan ciphertext dari blok sebelumnya, kemudian dilakukan operasi XOR dengan plainteks untuk menghasilkan cipherteks.

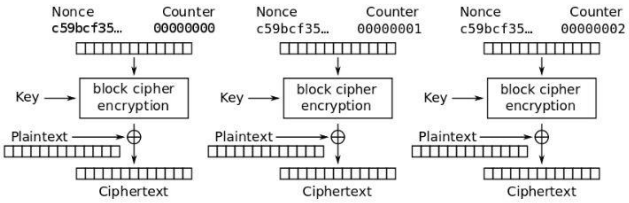
4. Output Feedback (OFB)



Gambar 2. 12 Operasi block cipher dengan OFB

Mirip dengan metode CFB, namun yang menjadi masukan untuk operasi enkripsi blok pada mode OFB adalah hasil output dari enkripsi blok sebelumnya.

5. Counter (CTR)



Gambar 2. 13 Operasi block cipher dengan CTR

Mode counter menggunakan blok yang merupakan kombinasi dari nonce dan counter sebagai masukan untuk enkripsi. Nonce merupakan satu Initialization Vector yang acak dan dipakai untuk setiap blok selanjutnya. Pada tahap selanjutnya nilai counter bertambah.

2.3 Koefisien Korelasi

Analisis korelasi adalah suatu metode yang digunakan untuk menggambarkan hubungan antara dua variabel kuantitatif tanpa mempersoalkan apakah suatu variabel tertentu tergantung kepada variabel lain. Korelasi pearson adalah salah satu jenis pengujian korelasi yang digunakan untuk mengetahui derajat keeratan hubungan dua variabel yang berskala interval atau rasio, di mana dengan pengujian ini akan mengembalikan nilai koefisien korelasi. Hubungan keeratan dua variabel dalam koefisien korelasi pearson memiliki nilai berkisar antara -1, 0 dan 1. Nilai -1 artinya terdapat korelasi negatif yang sempurna, 0 artinya tidak ada korelasi

dan nilai 1 berarti ada korelasi positif yang sempurna. Rasio koefisien korelasi yang berkisar antara -1, 0 dan 1 tersebut menggambarkan bahwa apabila semakin mendekati nilai 1 atau -1 maka hubungan makin erat, sedangkan jika semakin mendekati 0 maka hubungan semakin lemah (Sugiyono, 2013).

Perhitungan nilai koefisien korelasi didapatkan dari nilai kovarian dibagi dengan standar deviasi. Dua variabel yang digunakan dalam hal ini adalah variabel plaintext (P) dan ciphertext (C). Rumus untuk menghitung koefisien korelasi (R) dapat dilihat pada persamaan dibawah (Pearson, 1896)

$$r(p,c) = \frac{\sum_{i=1}^{n}(p_i-\mu(p))(c_i-\mu(c))}{\sigma(p)\sigma(c)} \quad (2.6)$$

Plaintext (P) yang berukuran panjang n karakter dinyatakan sebagai

$$p = (\mathrm{p1}, \mathrm{p2}, \ldots \mathrm{pn}) \quad (2.7)$$

Ciphertext (C) dinyatakan sebagai

$$C = (c1, c2, \ldots cn) \quad (2.8)$$

Perhitungan rata-rata pada plaintext ($\mu(p)$) dinyatakan sebagai

$$\mu(p) = \frac{1}{n}\sum_{i=1}^{n} p_i \quad (2.9)$$

Perhitungan rata-rata pada plaintext ($\mu(c)$) dinyatakan sebagai

$$\mu(c) = \frac{1}{n}\sum_{i=1}^{n} c_i \quad (2.10)$$

Perhitungan untuk standar deviasi plaintext ($\sigma(p)$) dinyatakan sebagai

$$\sigma(p) = \sqrt{\sum_{i=1}^{n}(p_i - \mu(p))^2} \quad (2.11)$$

Perhitungan untuk standar deviasi ciphertext ($\sigma(c)$) dinyatakan sebagai

$$\sigma(c) = \sqrt{\sum_{i=1}^{n}(c_i - \mu(c))^2} \quad (2.12)$$

Koefisien korelasi ini disebut koefisien korelasi Pearson karena diperkenalkan pertama kali oleh Karl Pearson pada tahun 1896. Analisis korelasi pearson sampai dengan saat ini masih banyak digunakan terutama dalam bidang matematika dan statistik. Tujuan dilakukannya analisis korelasi terhadap algoritma kriptografi pada penelitian ini supaya enkripsi yang dihasilkan dapat semakin acak.

2.4 Avalance Effect

Dalam kriptografi avalance effect adalah sifat yang mengubah sebagian besar keluaran meski masukan diubah sedikit saja. Avalance effect dibutuhkan dalam algoritma kriptografi untuk menambah efek keacakan pada enkripsi. Teknik kriptografi berkualitas tinggi akan mengubah drastis teks tersandi (cipher) hanya dengan mengubah sebagian kecil pada teks asal (plaintext) atau kunci (Fiestel, 1973).

$$Avalance\ effect = \frac{jumlah\ bit\ yang\ berubah\ dalam\ ciphertext}{jumlah\ bit\ dalam\ ciphertext}\ x\ 100\% \qquad (2.13)$$

Kinerja algoritma kriptografi dievaluasi menggunakan avalance effect dengan melihat perubahan satu bit dari plaintext atau satu bit dari kunci harus menghasilkan perubahan dalam banyak bit teks sandi. Kriteria suatu algoritma kriptografi berkualitas tinggi apabila ketika satu bit masukan diubah, tiap bit keluaran memiliki memiliki 50% peluang untuk berubah (Dewangan dkk., 2012)

2.5 Notasi Big-O

Algoritma adalah prosedur selangkah demi selangkah untuk mencari solusi suatu masalah dalam waktu yang terbatas. Pada umumnya algoritma mentransformasi obyek masukan menjadi obyek keluaran dengan waktu eksekusi yang merupakan fungsi dari obyek masukkan. Dalam arti makin besar obyek masukkan makin lama waktu yang dibutuhkan untuk menghasilkan obyek keluaran. Kompleksitas algoritma diukur berdasarkan kinerjanya dengan menghitung waktu eksekusi suatu algoritma. Waktu eksekusi algoritma dapat diklasifikasikan menjadi tiga kelompok besar, yaitu best-case (kasus terbaik), average-case (kasus rerata) dan worst-case (kasus terjelek). Pada pemrograman yang dimaksud dengan kasus terbaik, kasus terjelek dan kasus rerata suatu algoritma adalah besar kecilnya atau

banyak sedikitnya sumber-sumber yang digunakan oleh suatu algoritma. Makin sedikit makin baik; makin banyak makin jelek. Biasanya sumber-sumber yang paling dipertimbangkan tak hanya waktu eksekusi tetapi bisa juga besar memori, catu-daya dan sumber-sumber lain (Goldreich, 2008).

Notasi O-Besar adalah cara yang digunakan untuk menguraikan laju pertumbuhan suatu fungsi yang tidak lain adalah time complexity suatu algoritma. Berikut ini adalah sejumlah contoh nilai O-Besar yang dapat ditemui (Greene and Knuth, 1982).

O(1): konstan. Algoritma dengan O-Besar O(1) dieksekusi di kecepatan yang sama tidak tergantung pada data masukannya.

O(log n): logaritmik. Algoritma yang didasarkan pada pohon biner kerap mempunyai efisiensi O(log n).

O(n): linear. Pencarian linear untuk menemukan suatu elemen dalam suatu array tak urut

O(n log n): loglinear, quasilinear atau linearithmik. Algoritma pengurutan yang baik kerap mempunyai order O(n log n).

$O(n^2)$: kuadratik. Cukup efisien karena masih tetap dalam rentang waktu polinomial.

$O(2^n)$: eksponensial. Efisiensi non-polinomial yang paling penting adalah exponential time. Banyak masalah penting yang hanya dapat diselesaikan oleh algoritma dengan efisiensi seperti ini.

2.6 Brute Force

Brute force dalam kriptografi adalah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci. Penyerang secara sistematis memeriksa semua kemungkinan kata sandi dan frasa sandi sampai yang benar ditemukan, atau dikenal sebagai exhaustive search. Aplikasi pencarian brute force menggunakan teknik pemecahan masalah umum

dengan menghitung semua kandidat dan memeriksa masing-masing. Brute force bekerja dengan mengkalkulasikan semua kemungkinan kombinasi yang dapat membuat sebuah kata sandi dan mencobanya untuk melihat apakah itu adalah kunci yang tepat. Seiring panjang kunci bertambah, jumlah waktu, rata-rata, untuk menemukan kunci yang benar meningkat secara eksponensial (Deschall, 2005)