T-4

POLYNOMIAL FUNCTIONS DAN IMPLEMENTASINYA DALAM ALGORITMA *ADVANCED ENCRYPTION* STANDARD PADA *DATABASE ACCOUNTING*

[1]Agus Winarno, [2]Eko Tulus Budi Cahyanto, [3]Mulyadi

Sekolah Tinggi Sandi Negara, Jl. Raya H.Usa, Ciseeng, Bogor
[1]vivincord@gmail.com, [2]ekotulus.stsn9@gmail.com , [3]moviex.40809@gmail.com

Abstrak

Kebutuhan akan tersampaikannya suatu informasi dengan cepat dan aman merupakan latar belakang dari perkembangan teknologi proteksi komunikasi data. Proteksi komunikasi data dalam penerapannya merupakan implementasi dari ilmu kriptografi. Kriptografi adalah suatu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data serta autentikasi data [A.Menezes, P. van Oorschot, dan S. Vanstone- Handbook of Applied Cryptography]. System kunci dalam kriptografi terdiri dari dua jenis yaitu system kunci simetris dan system kunci asimetris. System kunci simetris adalah suatu system kriptografi yang dalam proses penyandiannya menggunakan kunci yang sama. Pada paper ini kami akan mengkaji tentang kriptografi system kunci simetris yaitu *Advanced Encryption Standard* (AES) yang menerapkan implementasi dari galois field dan polynomial functions. Fungsi-fungsi yang digunakan sangatlah sederhana namun dapat memberi layanan proteksi data sesuai standar yang berlaku [NIST]. Dalam implementasinya, AES dapat diterapkan dalam berbagai system proteksi komunikasi dan keamanan data salah satunya diterapkan pada *database accounting*. Alasan suatu database perlu dilindindungi keamanannya dikarenakan apabila database itu tidak dienkripsi maka dengan mudah database itu akan dicuri informasinya sehingga secara langsung ataupun tidak langsung akan merugikan perorangan atau instansi pemilik database tersebut.

Kata Kunci : Kriptografi, Galois field, Polynomial functions, Advanced Encryption Standard, Database accounting

1. PENDAHULUAN

Perkembangan teknologi yang semakin pesat mendorong orang untuk berlomba-lomba mengembangkan teknologi seusuai bidangnya baik untuk kepentingan dirinya sendiri maupun untuk kepentingan masyarakat luas. Bidang telekomunikasi merupakan bidang yang perkembangannya sangatlah signifikan juga sangat berpengaruh besar terhadap kehidupan manusia di zaman sekarang. Kebutuhan akan tersampaikannya suatu informasi dengan cepat, utuh, dan aman merupakan tuntutan yang harus terpenuhi dalam dunia pertelekomunikasian yang harus terpenuhi dalam mentransmisikan data

Makalah dipresentasikan dalam Seminar Nasional Matematika dan Pendidikan Matematika dengan tema *" Kontribusi Pendidikan Matematika dan Matematika dalam Membangun Karakter Guru dan Siswa"* pada tanggal 10 November 2012 di Jurusan Pendidikan Matematika FMIPA UNY

maupun informasinya. Namun yang perlu ditekankan di sini adalah aspek keamanan dari data ataupun informasi yang hendak disampaikan. Suatu data atau informasi memiliki kualifikasi biasa, rahasia dan sangat rahasia. Suatu data yang bersifat rahasia dan sangat rahasia haruslah benar-benar dijaga keamanannya baik dalam pentransmisian maupun dalam penyimpanan data, agar tidak diketahui oleh orang yang tidak berkepentingan. Karena, apabila data tersebut diketahui oleh orang yang tidak berhak, dikhawatirkan akan membahayakan maupun merugikan pemilik data tersebut.

Banyak cara untuk mengamankan suatu data ataupun informasi yang tidak ingin diketahui oleh pihak yang tidak berhak, salah satunya menggunakan ilmu kriptografi. Kriptografi adalah suatu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data serta autentikasi data [A.Menezes, P. van Oorschot, dan S. Vanstone- Handbook of Applied Cryptography]. Namun terkadang kita salah mengartikan antara istilah kriptografi dengan kriptologi. Kriptologi merupakan gabungan antara kriptografi dan kriptanalisis yang mana merupakan salah satu cabang dari ilmu matematika. Dasar keilmuan dari kriptologi sebagian besar adalah matematika yang antara lain mencakup Teori Probabilitas (*Probability Theory)*, Teori Informasi (*Information Theory*), Teori Kompleksitas (*Complexity Theory*), Teori Bilangan (*Number Theory*), Aljabar Abstrak (*Abstract Algebra*) dan Field Hingga (*Finite Field*), Graph dan Kombinatorika, dan Statistika.

Dengan ilmu kriptografi, suatu data yang hendak ditransmisikan maupun dijaga keamanannya akan dienkripsi terlebih dahulu supaya tidak bisa diketahui isinya dengan mudah oleh orang yang tidak berhak untuk mengetahui isi dari data tersebut. Enkripsi merupakan proses transformasi *plaintext* menjadi *cipher text* untuk merahasiakan berita dengan system kriptografi yang bertujuan agar pihak-pihak lain yang tidak memiliki hak tidak bisa membaca ataupun mengerti isi dari berita maupun informasi tersebut. Ketika hendak dibaca atau mengetahui isi dari data yang sudah dienkripsi maka harus dilakukan proses dekripsi untuk membuka sandi yang ada pada data tersebut.

Di zaman modern ini sudah banyak algoritma yang diciptakan untuk mengamankan suatu data baik yang sudah terpublikasikan secara umum maupun masih bersifat rahasia oleh instansi tertentu. *Advanced Encryption Standard* (AES) merupakan salah satu algoritma kriptografi yang pernah ada dan masih dipergunakan karena masih dipercaya keamanannya dalam enkripsi suatu data atau informasi. AES merupakan algoritma kriptografi system kunci simetris yang termasuk dalam klasifikasi *block cipher*. Algoritma AES salah satunya diimplementasikan dalam *database accounting* untuk menjaga keamanan dan kerahasian data yang terdapat dalam database tersebut.

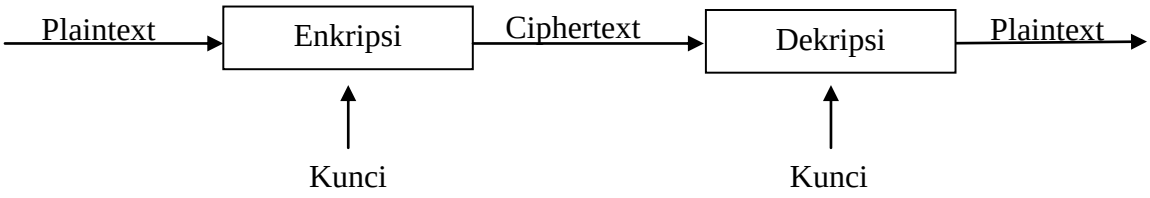
2. DASAR TEORI

2.1 Kriptografi

Kriptografi adalah suatu ilmu atau seni mengamankan pesan, dan dilakukan oleh cryptographer. Sedangkan cryptanalis adalah suatu ilmu dan seni membuka (breaking)

ciphertext dan orang yang melakukannya disebut *cryptanalyst*. Ditinjau dari terminologinya, kata kriptografi berasal dari bahasa Yunani yaitu *kryptos*, 'menyembunyikan', dan *graphein*, 'menulis', sehingga dapat didefinisikan sebagai ilmu yang mengubah informasi dari keadaan/bentuk normal (dapat dipahami) menjadi bentuk yang tidak dapat dipahami.

Algoritma Kriptografi selalu terdiri dari dua bagian, yaitu enkripsi dan dekripsi. Enkripsi (*encryption*) merupakan proses yang dilakukan untuk mengubah pesan yang tidak disandikan (*plaintext*) ke dalam bentuk yang tidak dapat dibaca (*ciphertext*). Sedangkan dekripsi (*decryption*) adalah proses kebalikannya. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Suatu sistem yang memiliki algoritma kriptografi, ditambah seluruh kemungkinan *plaintext*, *ciphertext* dan kunci-kuncinya disebut disebut dengan kriptosistem (*cryptosystem* atau *cryptographic system*). Secara sederhana proses kriptografi dapat digambarkan sebagai berikut:



Gambar 1. Proses Enkripsi/Dekripsi Sederhana

2.2 Advanced Encryption Standard (AES)

Advanced Encryption Standard merupakan algoritma pengganti dari *Data Encryption Standard* (DES) yang masa berlakunya sudah selesai dikarenakan faktor keamanan. Pada bulan Maret tahun 2001 ditetapkanlah algoritma baru Rijndael sebagai AES oleh *National Institute of Standards and Technology* (NIST). Algoritma Rijndael ini sendiri terpilih sebagai algoritma AES setelah mengalahkan 5 finalis lainnya yang diseleksi oleh NIST. Algoritma AES ini dipilih sebagai algoritma pengganti DES didasarkan pada tiga kriteria utama yaitu : keamanan, harga, dan karakteristik algoritma beserta implementasinya. Keamanan merupakan faktor utama dalam kriteria ini. Supaya algoritma ini tahan terhadap semua jenis serangan yang telah diketahui maupun belum diketahui. Disamping itu algoritma ini haruslah bebas digunakan tanpa harus membayar royalti. Dalam pengaplikasian dalam hardware maupun software, algoritma AES harus efisien dan cepat apabila dijalankan dalam berbagai *platform* 8 bit hingga 64 bit.

2.2.1 Panjang Kunci dan Ukuran Block

Panjang kunci algoritma Rijndael memiliki panjang kunci antara 128 bit sampai dengan 256 bit. Namun dalam penerapannya AES menetapkan panjang kunci yang dibutuhkan adalah 128 bit, 192 bit dan 256 bit sehingga kemudian dikenal dengan sebutan AES-128, AES-192, dan AES-256 walaupun pada penggunaanya paling banyak

menggunakan AES-128 dan AES-256 dikarenakan AES-192 sangatlah jarang digunakan. Untuk lebih jelasnya dapat dilihat pada tabel dibawah ini :

	Panjang Kunci (Nk words)	Ukuran Blok (Nb words)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Catatan : 1 word = 32 bit

Tabel 1. Tabel jumlah round berdasarkan panjang kunci.

Algoritma AES memiliki panjang kunci minimal 128 bit yang memberikan ketahanan terhadap serangan *exhaustive key search*. Dengan panjang kunci yang paling minimal ini mampu menghasilkan kemungkinan kunci sebanyak $3{,}4 \times 10^{38}$ kemungkinan. Sehingga apabila menggunakan metode *brute force attack* untuk mencoba semua kemungkian kunci dengan komputer tercepat yang mampu mencoba 1 juta kunci setiap detiknya maka akan membutuhkan waktu $5{,}4 \times 10^{24}$ tahun untuk mencoba semua kemungkian kuncinya. Dengan waktu yang bisa dianggap lama inilah diharapkan walaupun nanti AES mampu untuk dipecahkan dan diketahui kunci yang digunakan namun data maupun informasi yang dienkripsi dengan kunci tersebut sudah tidak berlaku lagi.

2.2.2 Algoritma Rijndael

Algoritma Rijndael menggunkan permutasi dan substitusi dan sejumlah putaran. Untuk setiap putarannya, Rijndael menggunakan kunci yang berbeda. Kunci pada setiap putaran algoritma ini disebut dengan *round key*. Dikarenakan algoritma Rijndael beroprasi dalam byte sehingga memungkinkan untuk diimplementasikan menjadi algoritma yang efisien ke dalam *software* maupun *hardware*[DAE04]. Algoritma Rijndael yang beroperasi pada blok 128-bit dengan panjang kunci 128-bit adalah sebagai berikut:

1. *AddRoundKey*(*initial round*) : melakukan XOR antara *state* awal (plainteks) dengan *cipher key*.
2. Putaran sebanyak *Nr*-1 kali. Sedangkan proses yang dilakukan pada setiap putarannya adalah sebagai berikut :

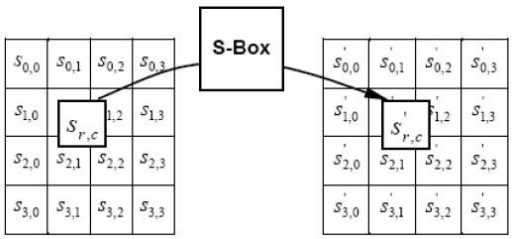
a. Transformasi *SubBytes()*
Transformasi substitusi *byte* non linear yang dioperasikan secara independen pada setiap *byte* dengan menggunakan tabel substitusi *s-box* dimana s-box tersebut juga memiliki invers yang digunakan untuk proses *deskripsi* nantinya. Tabel s-box untuk transformasi *SubBytes* AES dapat dilihat pada table 2 sedangkan pengaruh transformasi *SubBytes()* dapat diilustrasikan pada gambar 2.

b. Transformasi *ShiftRows()*

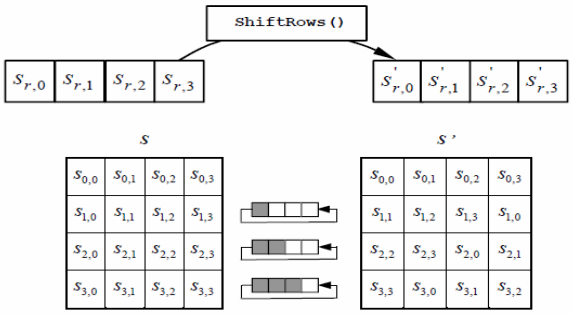
Pada transformasi shiftRows(), *state byte* pada tiga baris terakhir secara siklik digeser dengan jumlah pergeseran yang berbeda (*offset)*. Hal ini memiliki pengaruh terhadap pergerakan *byte* dari posisi rendah di dalam baris (yaitu nilai rendah dari c di dalam baris yang diketahui). Ilustrasi dari transformasi *ShiftRows()* dapat dilihat pada gambar 3.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabel 2. Tabel S-box transformasi SubBytes AES



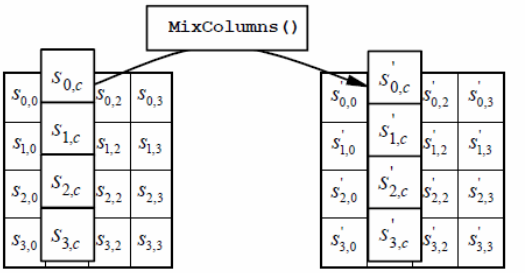
Gambar 2. Ilustrasi transformasi SubBytes() pada setiap byte state



Gambar 3. Ilustrasi transformasi ShiftRows()

c. Transformasi *MixColumns()*

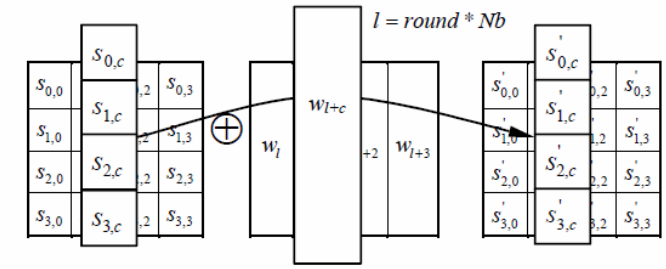
Transformasi MixColumns() beroperasi pada state kolom per kolom, dengan memperlakukan setiap kolom sebagai 4 buah polynomial. Kolom tersebut dianggap sebagai polynomial pada GF(2^8) dan dikalikan modulo $x^4 + 1$dengan polynomial tetap a(x). Untuk ilustrasi dari transformasi *MixColumns* dapat dilihat pada gambar 4.



Gambar 3. Ilustrasi transformasi MixColumns()

d. Transformasi *AddRoundKey()*

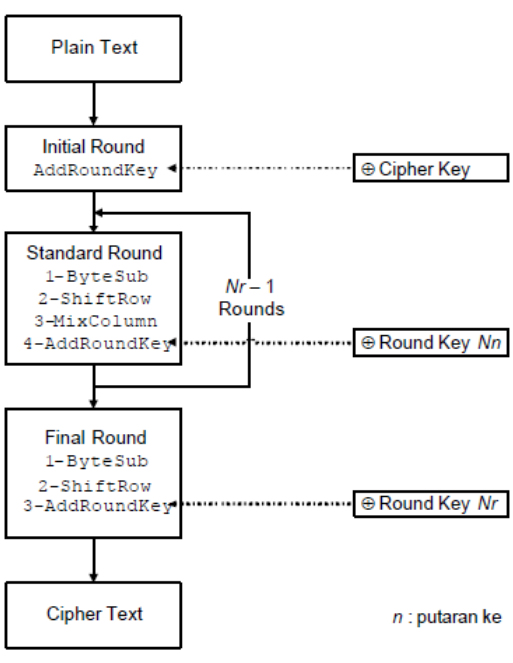
Pada transformasi *AddRoundKey()* sebuh kunci round ditambahkan kepada state dengan operasi *bitwise* sederhana XOR. Setiap kunci *round* terdiri dari Nb *word* dari hasil key schedule. Dengan demikian masing-masing ditambahkan ke dalam kolom dari state. Applikasi dari transformasi *AddRoundKey()* pada Nr round dari proses penyandian terjadi ketika $1 \leq round \leq$ Nr. Untuk lebih jelasnya dapat dilihat pada gambar 4 dimana $l = round * Nb$ alamat *byte* dalam *word* dari *key schedule*.



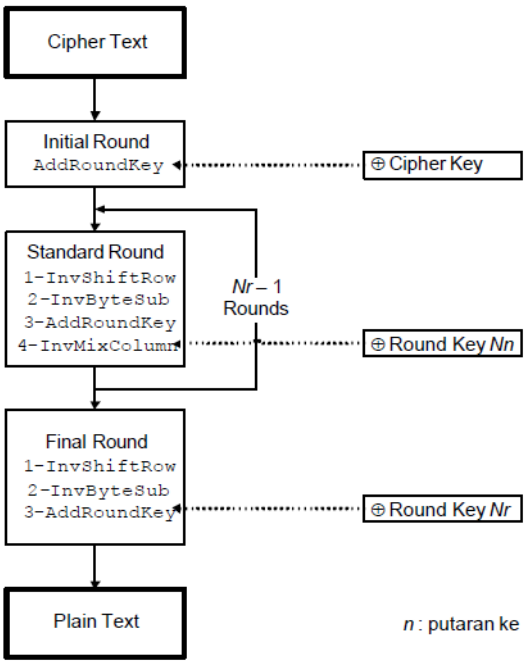
Gambar 4. Ilustrasi transformasi AddRoundKey()

2.2.3 Proses Enkripsi dan Deskripsi

Berikut ini adalah *flowchart* dari proses enkripsi dan dekripsi algoritma AES. Pada gambar 5 merupakan gambaran dari proses enkripsi algoritma AES sedangkan pada gambar 6 adalah proses dekripsi dari algoritma AES. Proses dekripsi dengan enkripsi memiliki perbedaan yang sangat besar yaitu terdapat proses invers dari tahapan transformasi *SubBytes()*, transformasi *ShiftRows()* dan transformasi *MixColumns()* menjadi transformasi *InvSubBytes()*, transformasi *InvShiftRows()* dan transformasi *InvMixColumns()*. Sehingga S-box algoritma AES yang digunakan untuk proses enkripsi pun berbeda dan menggunakan S-box invers dari algoritma deskripsi AES. Dan adapun S-box algoritma deskripsi AES dapat dilihat pada tabel 5.



Gambar 5. Diagram proses enkripsi AES



Gambar 6. Diagram proses deskripsi AES

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1x	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2x	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3x	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4x	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5x	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6x	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7x	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8x	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9x	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
ax	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
bx	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
cx	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
dx	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
ex	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
fx	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Tabel 3. Tabel InvS-box transformasi InvSubBytes AES

2.3 Fungsi Polinomial

Fungsi polynomial adalah suatu fungsi matematika yang melibatkan penjumlahan perkalian pangkat dengan satu atau lebih variable yang koefisien. Bentuk umum fungsi polynomial adalah sebagai berikut :

$$a_n x^n + \ldots + a_2 x^2 + a_1 x + a_0$$

Dimana pengkat tertinggi menunjukkan orde atau derajat dari fungsi tersebut. Dalam penerapan pada algoritma *Rijndael*, fungsi polynomial dapat direpresentasikan dalam opersasi penjumlahan, operasi perkalian, dan operasi perkalian dengan nilai x untuk menghasilkan algoritma yang kuat terhadap berbagai macam *attack* baik yang sudah diketahui maupun belum diketahui pada masa itu. Sebagai contoh yang paling sederhana adalah untuk merepresentasikan sebuah *byte* dengan nilai heksadesimal 57

(memiliki representasi biner 01010111) maka representasi nilai tersebut dalam polynomial adalah :

$$x^6 + x^4 + x^2 + x + 1$$

3. PEMBAHASAN POLINOMIAL PADA AES

Dalam proses penyandian dalam algoritma AES tidak terlepas dari fungsi polinomial dan galois field. Fungsi polinomial yang digunakan dalam AES ini mempengunakan koefisien berupa bilangan biner {1,0} dan hanya bekerja pada finite galois field(2^8). kita dapat membuat bentuk umum polinomial AES seperti dibawah ini:

$n = n_7x^7 + n_6x^6 + \ldots + n_1x + n_0$, dimana $n = (n_1,n_2,n_3,n_4,n_5,n_6,n_7)$, $n_i \in \{0,1\}$

contoh:

ada sebuah biner 10010101, dapat direpresentasikan dalam polinomial seperti:

$n = 1.x^7 + 0.x^6 + 0.x^5 + 1.x^4 + 0.x^3 + 1.x^2 + 0.x + 1.$

$n = x^7 + x^4 + x^2 + 1$

operasi-operasi yang berlaku pada algoritma AES antara lain penjumlahan(+) dan perkalian(*) group.

3.1. Operasi Penjumlahan Polinomial di GF (2^8)

Oprerasi penjumlahan polinomial di GF(2^8) yang dilakukan pada dua buah poliomial yang ada dalam komponen AES sama seperti operasi penjumlahan polinomial biasa. Namun penjumlahan koefesien dalam polinomial menggunakan operasi penjumlahan modulo dua atau sering disebut operasi X-OR yang dinotasikan $\oplus$, sehingga dalam penerapannya $1 \oplus 1 = 2 \bmod 2 = 0$ dan dan $1 \oplus 0 = 1$.

Contoh operasi penjumlahan polinomial dua buah *sub byte*:

Jika A = 11001110 , B =10110111 maka A + b = …?

$A = 11001110 \rightarrow 1.x^7 + 1.x^6 + 0.x^5 + 0.x^4 + 1.x^3 + 1.x^2 + 1.x + 0$

$B = 10110111 \rightarrow 1.x^7 + 0.x^6 + 1.x^5 + 1.x^4 + 0.x^3 + 1.x^2 + 1.x + 1$

$A+B = (1\oplus1).x^7 + (1\oplus0).x^6 + (0\oplus1).x^5 + (0\oplus1).x^4 + (1\oplus0)x^3 + (1\oplus1)x^2 + (1\oplus1).x + (0\oplus1)$

$= 0.x^7 + 1.x^6 + 1.x^5 + 1.x^4 + 1.x^3 + 0.x^2 + 0.x + 1$

Jadi hasil penjumlahan $A + B = x^6 + x^5 + x^4 + x^3 + 1$ atau dalam biner bisa ditulis 01111001.

Yang menjadi catatan adalah degree atau pangkat tertinggi dari proses penjumlahan ini tidak akan lebih dari x^7 .

3.2. Operasi Perkalian Polinomial di GF(2^8)

Operasi perkalian yang terjadi tidak jauh berbeda dengan operasi perkalian yang terjadi pada polinomial biasa, akan tetapi setiap perkalian yang terjadi antara dua elemen memungkinkan terjadinya degree yang lebih tinggi dari x^7 sehingga untuk tetap membuat semua hasil perkalian dari elemen ini tetap dalam galois field (2^8) maka setiap hasil perkalian yang menghasilkan pangkat yang lebih tinggi dari X^7 maka harus dimodulus dengan polinomial primitive yang ada di degree x^8 . Polinomial yang digunakan dalam operasi modulasi ini adalah $m(x) = x^8+x^4+x^3+x+1$.

Contoh :

A = $x^7 + x^4 + x^2 + 1$ dan B = $x^2 + 1$, maka A*B=…?

$x^7 + x^4 + x^2 + 1$

$x^2 + 1 \quad x^7 + x^4 + x^2 + 1$

$x^9 + x^6 + x^4 + x^2$

$x^9 + x^7 + x^6 + 1$

karena degree hasil perkalian lebih dari x^7 maka hasil perkalian tersebut harus dimodulus dengan $x^8+x^4+x^3+x+1 \overline{)\, x^9 + x^7 + x^6 + 1}$ dengan hasil $x+1$

$x^9 + x^5 + x^4 + x^2 + x$

$$x^7 + x^6 + x^5 + x^4 + x^2 + x + 1$$

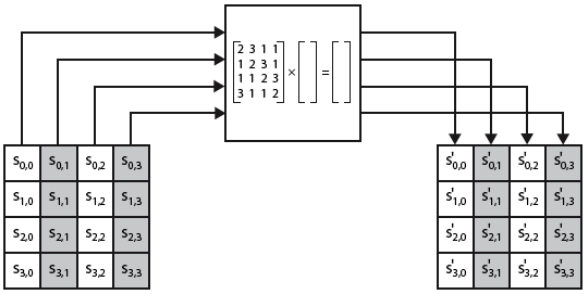
jadi hasil perkalian dari A*B $= x^7 + x^6 + x^5 + x^4 + x^2 + x + 1$.

3.3. Proses-proses AES yang Menggunakan Operasi Penjumlahan dan Perkalian Polinomial

a. MixColumns()

MixColumns() adalah sutu proses dalam algoritma AES yang mengambil masing-masing kolom dari input blok yang akan dioperasikan dengan suatu matrix konstan untuk menghasilkan kolom output yang panjangnya sama. Matrix konstan yang digunakan adalah $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$

Berikut ini adalah ilustrasi proses mix column dari algoritma AES,



Contoh:

Misal kolom ke-3 dari input mix column ini bernilai dalam heksadesimal $\begin{pmatrix} 1A \\ 1B \\ 2A \\ 2B \end{pmatrix}$

sehingga perhitungan polinomialnya mudah sehingga kita bisa menghitung outpunya sepperti berikut

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \text{x} \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 2.1A \oplus 3.1B \oplus 1.2A \oplus 1.2B \\ 1.1A \oplus 2.1B \oplus 3.2A \oplus 1.2B \\ 1.1A \oplus 1.1B \oplus 2.2A \oplus 3.2B \\ 3.1A \oplus 1.1B \oplus 1.2A \oplus 2.2B \end{pmatrix}$$

Perhitungan diatas dijabarkan dalam notasi polinomial seperti dibawah ini:

≫ $2.1A \oplus 3.1B \oplus 1.2A \oplus 1.2B \rightarrow x.(x^4+x^3+x) \oplus (x+1).(x^4+x^3+x+1) \oplus 1.(x^5+x^3+x) \oplus 1.h(x^5+x^3+x+1) = x^5+x^4+x^2 \oplus x^5+x^3+x^2+1 \oplus x^5+x^3+x \oplus x^4+x^3+x+1 = x^5+x^3+1 = 29$

≫ $1.1A \oplus 2.1B \oplus 3.2A \oplus 1.2B \rightarrow 1.(x^4+x^3+x) \oplus x.(x^4+x^3+x+1) \oplus (x+1).(x^5+x^3+x) \oplus 1.(x^5+x^3+x+1) = x^4+x^3+x \oplus x^5+x^4+x^2+x \oplus x^6+x^5+x^4+x^3+x^2+x \oplus x^5+x^3+x+1 = x^6+x^5+x^4+x^3+1 = 79$

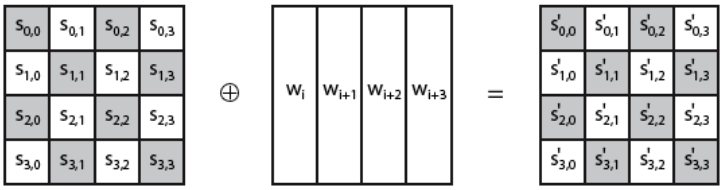
≫ $1.1A \oplus 1.1B \oplus 2.2A \oplus 3.2A \rightarrow x^4+x^3+x \oplus x^4+x^3+x+1 \oplus x^6+x^4+x^2 \oplus x^6+x^5+x^4+x^3+x^2+x = x^5+x^3+x+1 = 2B$

≫ $3.1A \oplus 1.1B \oplus 1.2A \oplus 2.2B \rightarrow x^5+x^3+x^2+x \oplus x^4+x^3+x+1 \oplus x^5+x^3+x \oplus x^6+x^4+x^2+x = x^6+x^3+1 = 49$

Dan didapat hasil output $\begin{pmatrix} 29 \\ 79 \\ 2B \\ 49 \end{pmatrix}$

b. AddRoundKey()

AddRoundKey() adalah proses penjumlahan modulo dua antara input S dengan kunci sepanjang 128 bit menjadi s'.



Sehingga bentuk persamaan kolom pertama dapat ditulis:

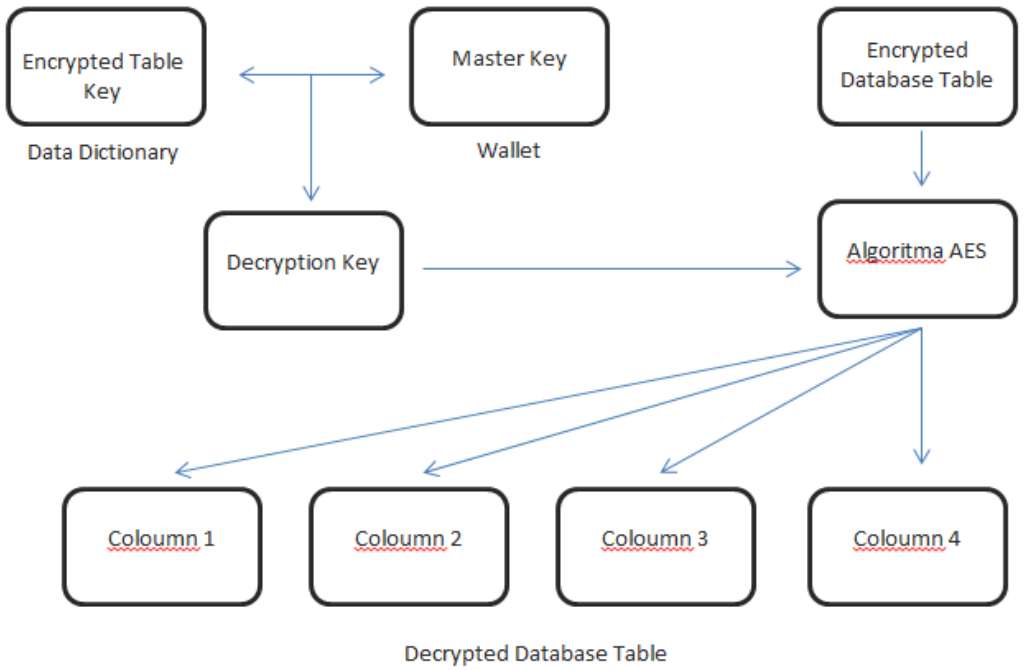
$S'_{i,0} = S_{i,0} \oplus w_i = (S_{i7,0}x^7 + S_{i6,0}x^6 + S_{i5,0}x^5 + \ldots + S_{i1,0}x + 1) \oplus (w_{i7}\,x^7 + w_{i6}\,x^7 + w_{i5}\,x^5 + \ldots + w_{i1}\,x + 1)$, dimana $0 \leq i \leq 3$

Dan dapat dibuat persamaan umum seperti berikut

$S'_{i,j} = S_{i,} \oplus w_i = (S_{i7,j}x^7 + S_{i6,j}x^6 + S_{i5,j}x^5 + \ldots + S_{i1,j}x + 1) \oplus (w_{i7}\,x^7 + w_{i6}\,x^7 + w_{i5}\,x^5 + \ldots + w_{i1}\,x + 1)$, dimana $0 \leq i \leq 3$ adalah baris, $0 \leq j \leq 3$ adalah kolom.

4. IMPLEMENTASI

Algoritma AES banyak digunakan untuk enkripsi database khususnya *database accounting*. Tujuan dari dilakukannya enkripsi pada *database* adalah untuk memberikan keamanan pada informasi yang terdapat di dalam *database* tersebut terhadap pihak yang tidak memiliki hak. Dikarenakan apabila informasi tersebut jatuh ataupun diketahui oleh orang yang tidak berhak dikhawatirkan dapat memberikan dampak negatif terhadap pemilik *database* tersebut baik dalam jangka panjang maupun jangka pendek. Enkripsi *database* dilakukan pada masing-masing kolom pada *database* tersebut dengan algoritma AES sedangkan *encryption key* untuk membukanya disimpan pada lokasi yang aman, disebut dengan *wallet* dan dapat berupa file pada *database server*. Untuk mengambil *master key* yang ada pada *wallet*, user harus melakukan login untuk otentikasi bahwa yang hendak melakukan proses dekripsi adalah *server* yang sah. *Table key* yang telah dienkripsi tersebut diletakkan di sebuah *data dictionary*. Dan yang perlu diketahui di sini adalah proses enkripsi sudah berjalan secara otomatis sehingga disebut dengan *enkripsi transparent* sehingga tidak perlu dikhawatirkan akan terjadi kealpaan pada user yang lalai untuk melakukan *enkripsi* pada *database*-nya setelah *database* itu didekripsi. Ketika seorang *user* memasukkan data ke sebuah kolom yang didefinisikan sebagai terenkripsi, sistem *database* mengambil *master key* dari *wallet*, sehingga mendapatkan *decryption key* untuk tabel tersebut dan menggunakannya sebagai nilai input untuk membuka data yang dienkripsi dengan algoritma AES pada database seperti gambar dibawah ini.



Gambar 7. Diagram proses dekripsi database accounting dengan AES

5. KESIMPULAN

Advance Encription Standard (AES) merupakan algoritma yang dipilih dari lima finalis oleh *National Institute of Standards and Technology* (NIST) pengganti Data Encryption Standard (DES) didasarkan pada tiga kriteria utama yaitu : keamanan, harga, dan karakteristik algoritma beserta implementasinya, dimana AES merupakan algoritma yang *multiplatform*. AES merupakan algoritma kriptografi system kunci simetris yang termasuk dalam klasifikasi *block cipher*. Algoritma AES memiliki panjang kunci minimal 128 bit. Dengan panjang kunci ini dihasilkan kemungkinan kunci sebanyak 3,4 x 10^{38} kemungkinan. Apabila menggunakan metode *brute force attack* untuk mencoba semua kemungkian kunci dengan komputer yang mampu mencoba 1 juta kunci setiap detiknya maka akan membutuhkan waktu 5,4 x 10^{24} tahun untuk mencoba semua kemungkian kuncinya. Dengan waktu yang lama dalam proses *anagramming* kunci inilah diharapkan walaupun nantinya AES mampu untuk dipecahkan namun data ataupun informasi yang dienkripsi dengan kunci tersebut sudah tidak berlaku lagi.

AES merupakan algoritma kriptografi yang menggunakan fungsi polinomial dalam tahapan proses enkripsi dan dekripsinya. Algoritma AES juga banyak diimplementasikan pada *database accounting* dengan tujuan untuk mengamankan data atau informasi yang terdapat didalam database, sehingga walaupun data tersebut diketahui atau bahkan dicuri oleh pihak yang tidak berhak namun pihak tersebut tidak bisa mengetahui isinya.

DAFTAR PUSTAKA

Hafman, Sari Agustini. Windarta, Susila, 2009. *"Teknik Block Cipher"*. Sekolah Tinggi Sandi Negara: Bogor

Lembaga Sandi Negara, 2007. *"Jelajah Kriptologi".* Lembaga Sandi Negara: Bogor

Zairul, Reza Andhika, 2006. *"Enkripsi Database Menggunakan Transparent Data Encryption*". Institut Teknologi Bandung: Bandung

Chung, Raphael. Phan, Wei, 2002. *"Mini-Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students".* Swinburne Sarawak Institute of Technology: Malaysia

Stinson, D.R. 2008, *"Cryptography Theory and Practice Third Edition*", Chapman & Hall/CRC: Florida

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, 1997. "*Handbook of Applied Cryptography*", CRC Press, Boca Raton