TESLA Vol. 8 No. 2, 97 – 101 (Oktober 2006)
Jurnal Teknik Elektro

ALGORITMA KRIPTOGRAFI AES RIJNDAEL

Didi Surian[*)]

Abstract
This paper discusses about AES Rijndael, the cryptography algorithm. The security factor in delivering important information has became awareness for some people. Many ways have been issued to secure the information. One of those ways is by applying encryption using some cryptography algorithms. The algorithm itself has been improved year by year to overcome irresponsible people to get the information easily. The algorithm Rijndael were born in 1997 and chosen as new standard in cryptography world in 2000. This algorithm has been chosen not just for its security, but also because the flexibility and efficiency implementation in many platform.

Keywords*: cryptography, encryption, decryption, rijndael, algorithm, plaintext, cipher text, cracker, AES*

PENDAHULUAN

Kriptografi adalah ilmu yang mempelajari mengenai bagaimana cara mengamankan suatu informasi. Pengamanan ini dilakukan dengan mengenkrip informasi tersebut dengan suatu kunci khusus. Informasi ini sebelum dienkrip dinamakan *plaintext*. Setelah dienkrip dengan suatu kunci dinamakan *ciphertext*. Keamanan suatu informasi agar tidak jatuh ke tangan orang-orang yang tidak berkepentingan sangatlah penting agar tidak disalahgunakan. Informasi ini dapat berupa *password*, nomor kartu kredit, ataupun informasi pribadi lainnya.

Usaha untuk menjaga kerahasiaan suatu informasi telah ada sejak jaman dahulu. Julius Caesar, kaisar Romawi, telah menggunakan metoda enkripsi sederhana dengan cara menggeser setiap karakter dalam pesannya dengan nilai tertentu. Cara ini cukup aman pada saat itu namun tidaklah mungkin dipakai saat ini karena dengan kemampuan komputasi komputer sekarang, akan sangat mudah dipecahkan.

Berbagai algoritma kriptografi telah diciptakan oleh para ahli kriptografi, namun berbagai usaha dilakukan oleh *cracker* untuk memecahkannya tidak sedikit yang membawa keberhasilan. Hal ini mendorong para kriptografi untuk menciptakan algoritma-algoritma yang lebih aman.

ADVANCED ENCRYPTION STANDARD

Hingga tahun 1990-an, algoritma kriptografi yang banyak dipakai adalah *Data Encryption Standard* (DES). Algoritma ini dipakai oleh *National Institute of Standards and Technology (*NIST) sebagai standar enkripsi data Federal Amerika Serikat.

DES termasuk dalam algoritma enkripsi yang sifatnya *cipher block*, yang berarti DES mengubah data masukan menjadi blok-blok 64-bit dan kemudian menggunakan kunci enkripsi sebesar 56-bit. Setelah mengalami proses enkripsi maka akan menghasilkan output blok 64-bit.

Seiring dengan perkembangan teknologi, kunci DES yang sebesar 56-bit dianggap sudah tidak memadai lagi. Pada tahun 1998, 70 ribu komputer di Internet berhasil membobol satu kunci DES dalam waktu 96 hari. Tahun 1999 kejadian yang sama terjadi lagi dalam waktu lebih cepat yaitu hanya dalam waktu 22 hari. Pada tanggal 16 Juni 1998, sebuah mesin seharga

[*)] Staf Pengajar Jurusan Teknik Elektro, Universitas Tarumanagara

250 ribu dolar dapat dengan mudah memecahkan 25% kunci DES dalam waktu kira-kira 2,3 hari atau diperkirakan dapat memecahkan kunci DES dalam waktu 4,5 hari.

Adanya kenyataan bahwa algoritma kriptografi DES tidak lagi aman, maka NIST mulai memikirkan sebuah algoritma kriptografi lain sebagai pengganti DES. Untuk itu diadakan kontes Internasional dimana pesertanya adalah ahli kriptografi dari seluruh dunia.Adapun diadakan secara terbuka dimaksudkan agar algoritma yang baru bukan dari produk badan pemerintah yang dapat dengan sengaja menanamkan *backdoor* pada algoritmanya. *Backdoor* ini dicurigai membuat *plaintext* dapat langsung dibaca tanpa harus menggunakan kunci.

Pada tahun 1997 kontes pemilihan suatu standar algoritma kriptografi baru pengganti DES dimulai dan diikuti oleh 21 peserta dari seluruh dunia. Algoritma yang akan dipilih selain harus memenuhi beberapa kriteria, yaitu

- Faktor keamanan, yang berarti algoritma tersebut harus tidak mudah dipecahkan oleh *cracker*, bersifat acak atau tidak mudah diterka outputnya, dan tidak berdasar algoritma matematika tertentu.
- Faktor biaya, dimana diperhitungkan kecepatan prosesing pada baik pada *hardware* dan *software*, dan besarnya memory yang dipakai.
- Faktor karakteristik implementasi, yakni meliputi kesederhanaan algoritma yang digunakan, kemudahan dan keamanan dalam implementasi di *hardware* dan *software*.

Algoritma ini akan dinamakan *Advanced Encryption Standard* (AES).

Setelah melewati tahap seleksi yang ketat, pada tahun 1999 hanya tinggal 5 calon yaitu algoritma *Serpent* (Ross Anderson-University of Cambridge, Eli Biham-Technion, Lars Knudsen-University of California San Diego), *MARS* (IBM Amerika), *Twofish* (Bruce Schneier, John Kelsey, dan Niels Ferguson-Counterpane Internet Security Inc, Doug Whiting-Hi/fn Inc, David Wagner-University of California Berkeley, Chris Hall-Princeton University), *Rijndael* (Dr. Vincent Rijmen-Katholieke Universiteit Leuven dan Dr. Joan Daemen-Proton World International), dan *RC6* (RSA Amerika).

Setahun kemudian pada tahun 2000, algoritma *Rijndael* terpilih sebagai algoritma kriptografi yang selain aman juga efisien dalam implementasinya dan dinobatkan sebagai AES. Nama *Rijndael* sendiri berasal dari gabungan nama penemunya.

DESKRIPSI ALGORITMA RIJNDAEL

Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan *cipher block*. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu.

Rijndael mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun *Rijndael* mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

Tabel 1. Jumlah proses berdasarkan bit blok dan kunci

Panjang Kunci (Nk) Dalam words	Ukuran Blok Data (Nb) Dalam words	Jumlah Proses (Nr)
4	4	10
6	4	12
8	4	14

Blok-blok data masukan dan kunci dioperasikan dalam bentuk *array*. Setiap anggota *array* sebelum menghasilkan keluaran *ciphertext* dinamakan dengan *state*. Setiap *state* akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, *AddRoundKey, SubBytes, ShiftRows*, dan *MixColumns*. Kecuali tahap *MixColumns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap *MixColumns* tidak akan dilakukan pada tahap terakhir. Proses dekripsi adalah kebalikkan dari dekripsi.

Karena terjadi beberapa tahap dalam proses enkripsi, maka diperlukan *subkey-subkey* yang akan dipakai pada tiap tahap. Pengembangan jumlah kunci yang akan dipakai diperlukan karena kebutuhan *subkey-subkey* yang akan dipakai dapat mencapai ribuan bit, sedangkan kunci yang disediakan secara *default* hanya 128-256 bit. Jumla total kunci yang diperlukan sebagai *subkey* adalah sebanyak *Nb(Nr+1),* dimana *Nb* adalah besarnya blok data dalam satuan *word*. Sedangkan *Nr* adalah jumlah tahapan yang harus dilalui dalam satuan *word*. Sebagai contoh, bilamana digunakan 128 bit (4 word) blok data dan 128 bit (4 word) kunci maka akan dilakukan 10 kali proses (lihat Tabel 1). Dengan demikian dari rumus didapatkan 4(10+1)=44 word=1408 bit kunci. Untuk melakukan pengembangan jumlah kunci yang akan dipakai dari kunci utama maka dilakukan *key schedule*.

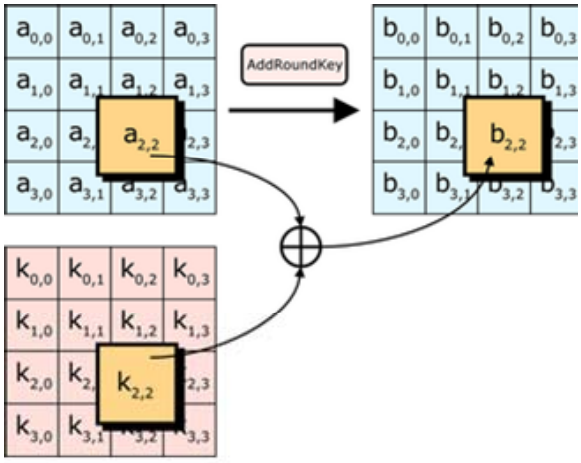
Key Schedule

Proses *key schedule* diperlukan untuk mendapatkan *subkey-subkey* dari kunci utama agar cukup untuk melakukan enkripsi dan dekripsi. Proses ini terdiri dari beberapa operasi, yaitu:

- Operasi *Rotate*, yaitu operasi perputaran 8 bit pada 32 bit dari kunci.
- Operasi *SubBytes*, pada operasi ini 8 bit dari *subkey* disubstitusikan dengan nilai dari S-Box.
- Operasi *Rcon*, operasi ini dapat diterjemahkan sebagai operasi pangkat 2 nilai tertentu dari *user*. Operasi ini menggunakan nilai-nilai dalam *Galois field*. Nilai-nilai dari *Rcon* kemudian akan di-XOR dengan hasil operasi *SubBytes.*
- Operasi XOR dengan w[i-Nk] yaitu word yang berada pada Nk sebelumnya.

Add Round Key

Pada proses ini *subkey* digabungkan dengan *state*. Proses penggabungan ini menggunakan operasi XOR untuk setiap byte dari *subkey* dengan byte yang bersangkutan dari *state*. Untuk setiap tahap, *subkey* dibangkitkan dari kunci utama dengan menggunakan proses *key schedule*. Setiap *subkey* berukuran sama dengan *state* yang bersangkutan. Proses *AddRoundKey* diperlihatkan pada Gambar 1.



Gambar 1. Proses *add round key*

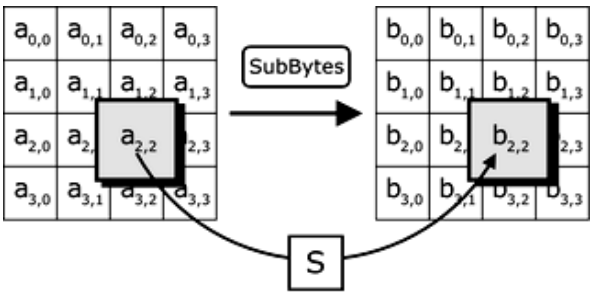
SubBytes

Proses *SubBytes* adalah operasi yang akan melakukan substitusi tidak linear dengan cara mengganti setiap *byte state* dengan *byte* pada sebuah tabel yang dinamakan tabel S-Box.

Sebuah tabel S-Box terdiri dari 16x16 baris dan kolom dengan masing-masing berukuran 1 *byte*. Tabel S-Box diperlihatkan pada Gambar 2 sedangkan proses *SubBytes* diperlihatkan pada Gambar 3.

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

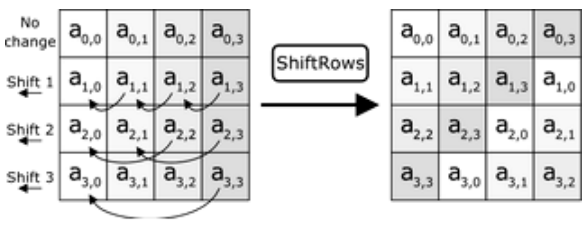
Gambar 2. S-Box.



Gambar 3. Proses *sub bytes*

Shift Rows

Proses *Shift Rows* akan beroperasi pada tiap baris dari tabel *state*. Proses iniakan bekerja dengan cara memutar *byte-byte* pada 3 baris terakhir (baris 1, 2, dan 3) dengan jumlah perputaran yang berbeda-beda. Baris 1 akan diputar sebanyak 1 kali, baris 2 akan diputar sebanyak 2 kali, dan baris 3 akan diputar sebanyak 3 kali. Sedangkan baris 0 tidak akan diputar. Proses *ShiftRows* diperlihatkan pada Gambar 4.



Gambar 4. Proses *shift rows.*

MixColumns

Proses *MixColumns* akan beroperasi pada tiap kolom dari tabel *state*. Operasi ini menggabungkan 4 *bytes* dari setiap kolom tabel *state* dan menggunakan transformasi linier

Operasi *Mix Columns* memperlakukan setiap kolom sebagai polinomial 4 suku dalam *Galois field* dan kemudian dikalikan dengan c(x) modulo (x4+1), dimana $c(x)=3x^3+x^2+x+2$. Kebalikkan dari polinomial ini adalah $c(x)=11x^3+13x^2+9x+14$. Operasi *MixColumns* juga dapat dipandang sebagai perkalian matrix.

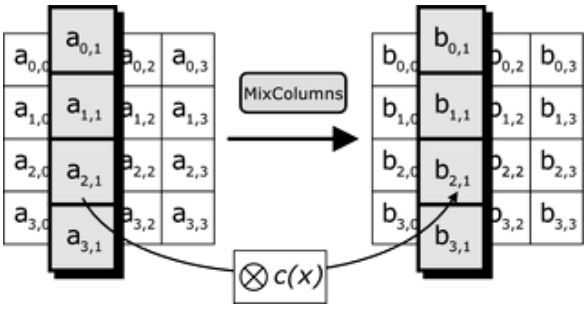
Langkah *MixColumns* dapat ditunjukkan dengan mengalikan 4 bilangan di dalam *Galois field* oleh matrix berikut ini.

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Atau bila dijabarkan:

$r_0=2a_0+a_3+a_2+3a_1$
$r_1=2a_1+a_0+a_3+3a_2$
$r_2=2a_2+a_1+a_0+3a_3$
$a_3=2a_3+a_2+a_1+3a_0$

Operasi penjumlahan di atas dilakukan dengan operasi XOR, sedangkan operasi perkalian dilakukan dalam *Galois field*.



Gambar 5. Proses *mix columns*

PENUTUP

Algoritma kriptografi AES *Rijndael* adalah algoritma kriptografi yang cukup

handal hingga saat ini. Pada tahun 2006, *National Security Agency* (NSA) pernah menyatakan bahwa AES cukup aman digunakan untuk mengamankan data-data pemerintah Amerika Serikat yang bukan tergolong sangat rahasia.

Hingga tahun 2006 serangan terbaik terhadap algoritma *Rijndael* hanya berhasil menembus putaran ke-7 untuk kunci 128 bit, putaran ke-8 untuk kunci 192 bit, dan putaran ke-9 untuk kunci 256 bit. Dengan melihat jumlah putaran yang berhasil ditembus, tidaklah tidak mungkin suatu hari algoritma ini dapat dengan mudah ditembus. Namun demikian algoritma *Rijndael* masih dipandang algoritma yang cukup handal.

Referensi

http://en.wikipedia.org/wiki/Rijndael_encryption_algorithm.html diakses 12 Maret 2006 10:45:59

http://en.wikipedia.org/wiki/Rijndael_key_schedule.html diakses 13 Maret 2006 09:45:49

http://www.samiam.org/galois.html diakses 17 Maret 2006 11:06:52

http://www.samiam.org/key-schedule.html diakses 17 Maret 2006 11:17:17

http://en.wikipedia.org/wiki/Rijndael_S-box.html diakses 13 Maret 2006 09:04:56

http://en.wikipedia.org/wiki/Rijndael_mix_columns.html diakses 13 Maret 2006 09:38:38

Y. Kurniawan. *Kriptografi: Keamanan Internet dan Jaringan Komunikasi*. April 2004. Informatika Bandung