RAPID ENCRYPTION ALGORITHM: SOLUSI KEAMANAN *VIDEO CONFERENCE*

Kholif Faiz Ma'ruf[1], Mora H.[2], Zaenal S.[3]
Lembaga Sandi Negara
Jl. Harsono R.M. No.70, Ragunan, Pasar Minggu, Jakarta Selatan 12550
Telp. (021) 7805814 ext. 2603, Faks. (021) 78844104
E-mail: kholif.faiz@gmail.com, morahertanto@yahoo.co.id, z41on7@yahoo.com

ABSTRAK

Teknologi video conference telah menjadi suatu kebutuhan bagi instansi pemerintah maupun swasta. Pemanfaatan video conference untuk rapat tertutup membutuhkan keamanan yang menjamin kerahasiaan dan keutuhan data. Enkripsi merupakan alternatif solusi keamanan yang dapat diterapkan pada video conference. Namun tidak semua algoritma enkripsi dapat diimplementasikan secara efisien pada aplikasi video conference. Hal ini dikarenakan video conference bersifat data streaming sehingga membutuhkan algoritma enkripsi yang memiliki fitur high speed of encryption, low memory usage, simple but secure, easy implementation dan low error propagation. Penulis mendesain Rapid Encryption Algorithm (REAL) sebagai solusi keamanan untuk video conference. Pengujian dilakukan dengan melakukan perbandingan terhadap beberapa algoritma enkripsi standard. Pengujian yang dilakukan meliputi pengujian keamanan, kecepatan, memory usage, kemudahan implementasi, dan error propagation. Di akhir paper disimpulan bahwa Rapid Encryption Algorithm (REAL) dapat diimplementasikan sebagai solusi keamanan untuk video conference yang memiliki fitur high speed of encryption, low memory usage, simple but secure, easy implementation dan low error propagation.

Kata Kunci: Rapid Encryption Algorithm (REAL), secure, video conference

1. PENDAHULUAN

Perkembangan Teknologi Informasi dan Komunikasi (TIK) serta kejahatan digital dewasa ini semakin menigkat. Salah satu contoh kejahatannya adalah penyadapan pada *video conference*. Instansi pemerintah dan swasta membutuhkan *video conference* untuk melaksanakan rapat tertutup jarak jauh yang bersifat rahasia. Tanpa disadari, *video conference* konvensional dapat dengan mudah disadap oleh pihak lain. Untuk mengamankan informasi yang bersifat rahasia tersebut, diperlukan suatu pengamanan. Salah satu solusinya adalah menggunakan teknik enkripsi.

Pada kenyataannya, tidak semua algoritma enkripsi dapat digunakan untuk mengamankan *video conference*. Algoritma enkripsi yang dapat digunakan pada *video conference* haruslah cepat, *low error propagation*, konsisten, *low memory usage*, *easy implementation,* tetapi tetap kuat secara kriptografis. Desain algoritma Enkripsi yang kompleks memiliki tingkat keamanan dan kekuatan kriptografis yang baik, namun akan mengurangi kecepatan transmisi *video conference*. Sedangkan desain algoritma enkripsi yang *simple* akan meningkatkan kecepatan *video conference,* namun akan mengurangi tingkat keamanan dan kekuatan kriptografisnya.

Dalam paper ini penulis mendisain *Rapid Encryption Algorithm* (REAL) yang dapat digunakan sebagai solusi untuk membangun *secure video conference*. Disertai pula perbandingan REAL terhadap algoritma enkripsi *standard* seperti AES dan VEA dalam hal kecepatan enkripsi, *memory usage*, kemudahan implementasi dan tingkat *error propagation.*

2. VIDEO CONFERENCE

2.1 Pengertian Umum

Video conference merupakan teknologi yang sedang berkembang saat ini dan banyak diminati oleh berbagai kalangan masyarakat. Informasi yang berupa suara maupun gambar pada *video conference* ditransmisikan melalui media jaringan telepon (*digital* maupun analog), LAN/ internet, atau satelit. Media tersebut merupakan media publik dimana semua orang dapat mengakses informasi yang ditransmisikan. Oleh karena itu, informasi strategis yang bersifat rahasia jika ditransmisikan melalui media publik perlu diamankan terlebih dahulu.

2.2 Pemanfaatan Kriptografi pada *Video Conference*

Aspek pengamanan yang dibutuhkan oleh *video conference* adalah otentikasi, kerahasiaan, integritas dan *non-repudiation*.

a. Otentikasi
   Setiap *user* atau mesin yang terlibat dalam sebuah *call conference* harus dapat diotentikasi bahwa *user*/mesin tersebut adalah sebagaimana yang diakuinya. Tanpa otentikasi maka siapapun atau mesin apapun bisa berpura-pura menjadi orang lain dan ikut dalam sebuah *call conference*. Tidak hanya *endpoint* yang harus diotentikasi, *gateway, gatekeeper*, dan MCU juga harus diotentikasi. Otentikasi dapat terjadi pada level *end user* atau hanya pada level fisik.
b. Kerahasiaan
   Untuk melindungi data dari pihak ketiga yang tidak berkepentingan maka data yang ditransmisikan harus dijamin kerahasiaannya.

Kriptografi menyediakan aspek pengamanan berupa kerahasiaan dengan melakukan enkripsi pada data yang akan dilindungi. Hal ini berarti, jika data jatuh ke pihak ketiga yang tidak berwenang maka pihak tersebut tidak akan bisa melihat informasi yang dikirim tanpa mengetahui algoritma enkripsi dan kunci yang digunakan.

c. Integritas
Integritas data berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain ke dalam data yang sebenarnya. Kriptografi menyediakan hal tersebut melalui fungsi *hash*. Pada *video conference*, semua data *signaling* dan media *stream* yang ditransimisikan harus utuh (sama dengan aslinya).

d. *Non-repudiation*
Non-repudiation adalah suatu mekanisme yang memastikan bahwa seseorang yang ikut dalam sebuah *call conference* tidak bisa menyangkalinya. *Non-repudiation* bersifat dua arah, artinya di sisi yang lain *service provider* juga tidak bisa menuntut seseorang untuk membayar biaya *call* yang ternyata tidak pernah dilakukan oleh orang tersebut.

3. KRIPTOGRAFI

Kriptografi adalah ilmu mengenai teknik-teknik matematika yang dihubungkan dengan aspek-aspek pengamanan informasi seperti kerahasiaan (*confidentiality*), keutuhan data (*data integrity*), otentikasi entitas (*entity authentication*), dan keaslian data (*data origin authentication*).

3.1 Tujuan Kriptografi

Tujuan kriptografi terbagi menjadi empat yaitu:

a. *Privacy/confidentiality* adalah layanan untuk menjaga kandungan atau isi informasi dari pihak yang tidak berhak. Terdapat banyak pendekatan-pendekatan yang dilakukan untuk mewujudkan kerahasiaan tersebut, dimulai dari pengamanan atau perlindungan secara fisik hingga ke dalam bentuk algoritma berbasis matemetika yang membuat data menjadi tidak terbaca.

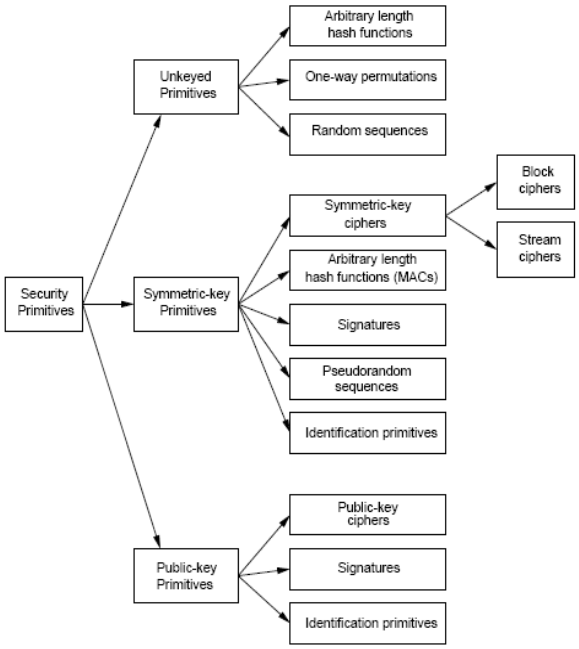
b. *Data integrity* adalah layanan untuk mengetahui dan mencegah kegiatan perubahan ataupun pemodifikasian data oleh pihak yang tidak berhak. Kemampuan yang harus dimiliki untuk menjamin keutuhan data adalah adanya teknik untuk dapat mendeteksi adanya manipulasi data yang dilakukan oleh pihak yang tidak berhak. Manipulasi data terdiri dari penyisipan, penghapusan, dan penggantian.

c. *Authentication* adalah layanan yang berhubungan dengan *identification*. Layanan ini mendeteksi keaslian entitas (pengirim/penerima) dan keaslian informasi (data). Pihak-pihak yang berkomunikasi saling diidentifikasi satu sama lain. Sama halnya dengan informasi yang dipertukarkan, juga diidentifikasi mengenai keaslian, tanggal pembuatan, kandungan isi, waktu pengiriman, dan lain sebagainya. Karena alasan-alasan itulah *authentication* di dalam kriptografi dibagi menjadi dua yaitu *entity authentication* dan *data origin authentication*. Secara implisit *data origin authentication* menyediakan keutuhan data (dalam artian, jika data berubah, maka sumber data tersebut juga berubah).

d. *Non-repudiation* adalah layanan yang berfungsi sebagai anti penyangkalan suatu entitas ataupun seseorang bahwa entitas tersebut telah melakukan suatu kegiatan ataupun aktivitas tertentu. Ketika seseorang menyangkal melakukan suatu aktivitas, dibutuhkan suatu cara untuk membuktikan kebenarannya. Cara tersebut dapat melibatkan pihak ketiga untuk melakukan pembuktian.

3.2 Taksonomi Kriptografi

Kriptografi tidak hanya berkecimpung pada empat hal diatas. Selain hal tersebut, kriptografi dapat digunakan untuk mencegah dan melindungi segala aktivitas kecurangan ataupun usaha-usaha perusakan. Hal tersebut dapat digambarkan dengan taksonomi kriptografi yang digambarkan pada Gambar 1.



Gambar 1. Taksonomi kriptografi primitif

Sistem kripto simetris yang menyediakan keamanan secara praktis terbagi dalam dua kategori yaitu *stream cipher* dan *block cipher*. Menurut Rueppel, perbedaan *stream cipher* dan *block cipher* adalah:

a. *Stream cipher* membagi teks terang ke dalam karakter-karakter dan menyandi masing-masing karakter per satuan waktu dengan suatu fungsi waktu bervariasi, yang ketergantungan waktunya diatur berdasarkan kondisi (*state*) internal dari algoritma sistem sandi tersebut
b. *Block cipher* membagi teks terang kedalam blok-blok dengan ukuran yang ditentukan, kemudian memproses masing-masing blok secara terpisah untuk menghasilkan blok-blok teks sandi. Umumnya ukuran yang digunakan untuk blok teks terang sama dengan ukuran blok teks sandi. Sedangkan ukuran blok kunci yang digunakan untuk menyandi setiap blok teks terang dapat sama atau berbeda ukurannya

4. RAPID ENCRYPTION ALGORITHM (REAL)

4.1 Deskripsi REAL

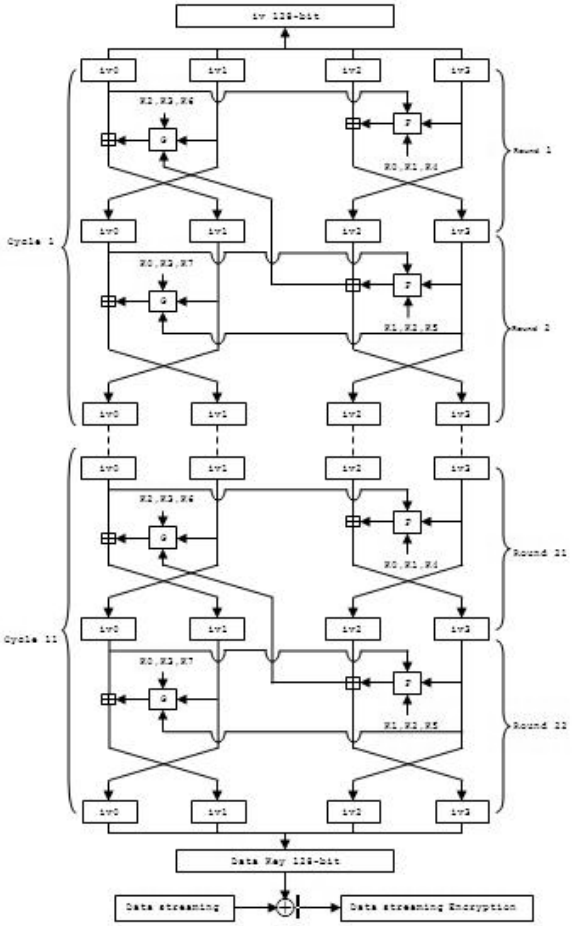
REAL (*Rapid Encryption Algorithm*) merupakan algoritma *stream cipher based on block cipher.* REAL mengenkripsi 128-bit data, serta memiliki *input* kunci sebesar 256-bit. Awalnya *input initialitation vector* (iv) dibagi menjadi empat buah subblok, kemudian masing-masing subblok akan diproses sebagai *input* untuk fungsi *f* dan fungsi *g* secara *feistel*. Proses ini diulang terus-menerus sampai 11-*cycle*. Satu *cycle* terdiri dari dua *round*. *Output feistel* ini kemudian di XOR dengan *data streaming video conference.*

4.2 Struktur dan Desain REAL

REAL menggunakan nilai *delta* `9e3779b9` untuk mencegah *output* yang buruk (menghasilkan difusi yang baik) terhadap *input*-an yang ekstrim. Penulis mendesain REAL agar dapat menghasilkan output 128-bit *data key* sekali waktu agar proses enkripsi menjadi lebih cepat.

REAL menggunakan dua buah *feistel network* yang disusun secara paralel. REAL memiliki 11-*cycle* (22-*round*) dengan setiap *cycle* terdiri dari dua *round*. Setiap *round* REAL terdiri dari dua buah fungsi sederhana yang digunakan secara berulang yaitu fungsi *f* dan fungsi *g*.

Input iv dari algoritma REAL akan dibagi menjadi empat buah subblok yang masing-masing subblok berisi 32-bit teks terang (v_0, v_1, v_2, v_3) yang terurut dari kiri ke kanan. Pada setiap *round i* (i = 0, 1, 2,…, 22*)*, dua subblok paling kiri (v_0, v_1) akan menjadi *input* pada *feistel* yang pertama. Sedangkan untuk dua buah subblok paling kanan (v_2, v_3) akan menjadi *input* pada *feistel* yang kedua. Selain itu, blok v_1 juga digunakan sebagai input *feistel* fungsi *f* pada round ke *l*, dimana $l=i+1$ (untuk setiap *i* yang ganjil). Lalu o*utput xor* dari *round-l* akan menjadi input *feistel* fungsi *g* pada *round* ke *i* (untuk setiap *i* yang ganjil). Proses ini berulang terus menerus sampai pada *round* terakhir. Dan akhirnya *output* dari *round* terakhir akan di *concate* menjadi 128-bit *output data key*. Kemudian *data key* tersebut akan di XOR dengan *data streaming video conference.* Untuk lebih jelasnya, alur pemrosesan dan struktur dari algoritma REAL dapat dilihat pada Gambar 2.



Gambar 2. Struktur algoritma REAL

4.3 Proses Enkripsi

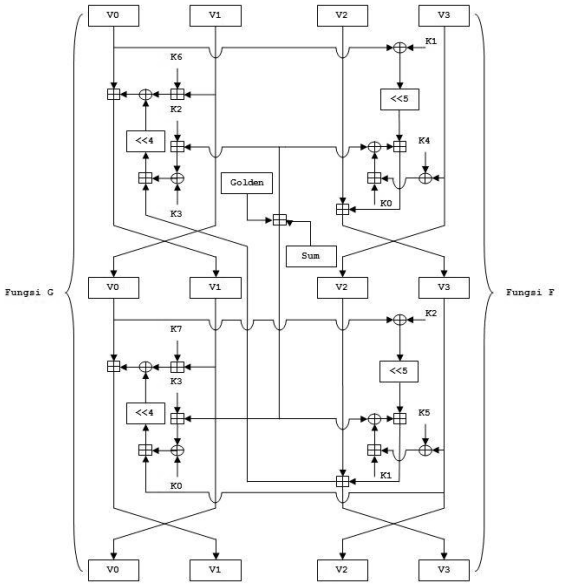
Proses enkripsi diawali dengan *input* iv sebanyak 128-bit. Kemudian 128-bit iv tersebut dibagi menjadi empat buah subblok yaitu blok sisi kiri (v_0,v_1) masing-masing sebanyak 32-bit dan blok sisi kanan (v_2,v_3) masing-masing sebanyak 32-bit. Setiap subblok iv tersebut akan dioperasikan dengan fungsi *f* dan fungsi *g* disetiap *round*-nya. Struktur penyandian REAL untuk satu *cycle* (dua *round*) dapat dilihat pada Gambar 3.

Hasil proses *feistel* satu blok iv (128-bit) menjadi blok *data key* (128-bit) dalam satu *cycle* REAL adalah dengan menggabungkan ($v_0||v_1||v_2||v_3$). Pada *cycle* berikutnya dilanjutkan proses seperti diatas sampai dengan 11-*cycle* (22-*round*). Hasil dari 11-*cycle Data key* inilah yang akan di XOR dengan *data streaming* untuk menghasilkan *data streaming encryption.*

4.4 Proses Dekripsi

Proses dekripsi pada prinsipnya sama halnya seperti pada proses enkripsi. Namun hal yang

berbeda adalah penggunaan *data key* sebagai input iv, serta operator *substraction* sebagai operator pembalik terhadap *addition.*



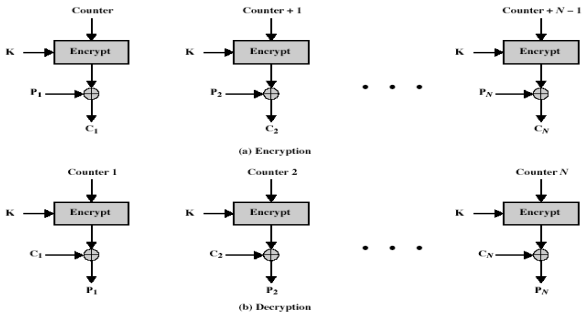
Gambar 3. Fungsi *f* dan *g* pada satu *cycle* REAL

4.5 Mode Operasi

Mode operasi yang digunakan adalah mode operasi *Counter* (CTR). Mode operasi ini cocok digunakan untuk data yang bersifat *streaming* karena dapat melakukan *generating data key* secara parallel bersamaan dengan mengalirnya *data streaming.* Dengan catatan bahwa *counter* (dalam hal ini adalah iv) yang digunakan selalu berbeda untuk menghindari hasil *data key* yang sama. Gambar 4 memperlihatkan alur mode operasi *counter.*

5. DESAIN RASIONAL

REAL menggunakan 22-*round* untuk mendapatkan kecepatan enkripsi yang baik namun tetap memiliki tingkat difusi yang cukup baik pula. REAL memiliki 256-bit kunci yang cukup untuk memenuhi keamanan 128-bit *security level* dalam melindungi 128-bit iv. *Key schedule*-nya di desain sangat sederhana agar algoritma ini dapat berjalan lebih cepat.



Gambar 4. Mode operasi *counter*

5.1 Estimasi Tingkat Keamanan

Pada tulisan ini, penulis mengestimasi tingkat keamanan algoritma REAL sampai pada tingkat difusinya menggunakan uji SAC dan uji BIC, hasilnya adalah algoritma ini memiliki tingkat keacakan yang baik. Jika diuji dengan uji SAC, 11-*cycle* REAL memiliki tingkat difusi sebesar 49.19% - 50.8%, dengan tingkat difusi yang terbaik adalah 50%. Jika diuji dengan uji BIC, 11-*cycle* REAL memiliki tingkat difusi sebesar 0 – 0,01 dengan tingkat difusi yang terbaik adalah 0.

Performa algoritma juga harus dapat diestimasi untuk mengetahui tingkat kecepatan dan seberapa besar penggunaan memorinya. REAL memiliki kecepatan penyandian rata-rata sebesar 85 MB/s untuk menyandi data file berukuran 1 MB sampai 100 MB pada prosesor AMD 2,2 GHz dan RAM sebesar 926 MB. dan membutuhkan memori minimal sebesar 58 Byte pada implementasi C.

5.2 Ukuran Kunci

Menurut Blaze, M. et al, 1996, bahwa untuk melindungi informasi dalam 20 tahun mendatang terhadap kekuatan komputer yang besar, panjang kunci (untuk kriptografi simetrik) sedikitnya adalah 80-bit. Sedangkan untuk melindungi informasi dalam jangka waktu 50 tahun mendatang, panjang kunci yang disarankan adalah sama dengan atau lebih besar dari 128-bit.

Menurut NESSIE *report* (2004), untuk mendapatkan *high security level,* ukuran kunci yang diperlukan setidaknya sepanjang 256-bit dengan ukuran blok setidaknya 128-bit. REAL memiliki ukuran kunci sepanjang 256-bit yang cukup untuk memenuhi ukuran standar penggunaan kunci hingga 38 tahun mendatang.

5.3 Ukuran Blok Pembangkit Kunci

REAL memiliki ukuran blok pembangkit kunci sebesar 128-bit ini tidak akan mengurangi kecepatan enkripsinya. Hal ini dikarenakan REAL terdiri dari dua buah *feistel network* paralel dengan hanya mengoperasikan 32-bit subblok teks terang ke dalam fungsi *f* dan fungsi *g.*

REAL memiliki ukuran blok pembangkit kunci sebesar 128-bit, sedangkan ukuran operasi sub-blok pada REAL adalah sebesar 32-bit. Hal ini dapat memudahkan dan mempercepat proses perhitungan pada prosesor terutama prosesor 32-bit, karena dapat menghemat penggunaan memori yang dibutuhkan untuk operasional.

5.4 Jumlah Round

REAL memiliki 11-*cycle* dengan setiap *cycle* REAL terdiri dari dua *round*. Tujuan penggunaan 22-*round* adalah untuk menjamin kecepatan dengan tetap mempertimbangkan tingkat difusi. Tabel 1 merupakan tabel perbandingan jumlah *round* REAL terhadap kecepatan enkripsi dan tingkat difusinya.

Tabel 1. Perbandingan jumlah *round* REAL

No	*Cycle*	SAC		Kecepatan (MBps)	Keterangan
		Max	min		
1	1	100 %	0 %	686,6255	Tidak lulus
2	6	100 %	0 %	147,3161	Tidak lulus
3	7	100 %	0 %	126,64	Tidak lulus
4	8	100 %	0 %	104,2875	Tidak lulus
5	9	100 %	0 %	92,10435	Tidak lulus
6	10	50,83 %	43,62 %	89,36995	Tidak lulus
7	**11**	**50,80 %**	**49,20 %**	**84,98466992**	**Lulus**
8	12	50,73 %	49,13 %	76,28354	Lulus
9	13	50,84 %	49,20 %	71,27433	Lulus
10	14	50,81 %	49,23 %	66,63476	Lulus
11	15	50,79 %	49,27 %	62,47178	Lulus
12	16	50,84 %	49,27 %	59,2006	Lulus
13	32	50,82 %	49,23 %	30,78003	Lulus

Dari tabel di atas dapat dilihat bahwa 11-*cycle* REAL memiliki tingkat kecepatan yang paling baik dengan tingkat difusi yang cukup acak.

5.5 Algoritma pembangkit subkunci (*key schedule*)

Key schedule REAL didesain sesederhana mungkin agar algoritma ini dapat berjalan cepat. REAL memiliki 256-bit kunci yang terpecah menjadi delapan subkunci (k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7), dengan masing-masing subkunci terdiri dari 32-bit. Penjadwalan kuncinya adalah sebagai berikut:

Tabel 2. *Key schedule* REAL

Untuk round ganjil:	$k_0, k_1, k_2, k_3, k_4, k_6$
Untuk round genap:	$k_0, k_1, k_2, k_3, k_5, k_7$

Dengan demikian, pada setiap *round* akan terdapat enam buah subkunci berbeda yang memiliki panjang total kunci di setiap *round*-nya adalah 192-bit kunci yang digunakan secara berulang.

REAL tidak memiliki algoritma pembangkit subkunci yang cukup kompleks. Tujuan utama menyederhanakan algoritma pembangkit subkunci ini adalah untuk meningkatkan kecepatannya serta memperkecil biaya implementasi *software*.

5.6 Fungsi REAL

REAL memiliki dua buah fungsi *round* yaitu fungsi *f* dan fungsi *g*. masing-masing fungsi ini bekerja saling melengkapi terhadap satu *cycle* REAL untuk dapat menghasilkan *output* yang mempunyai nilai SAC yang cukup baik. Kedua fungsi ini memiliki operasi yang sama namun susunan yang berbeda. Untuk setiap *round i,* sebuah *fungsi f* dan fungsi *g* didefinisikan sebagai berikut:

$$f(k,v,sum):\begin{cases} v_2 \mathrel{+}= k_4 \oplus v_3 + k_0 \oplus sum + (v_0 \oplus k_1) \ll 5 \\ v_3 \mathrel{+}= k_5 \oplus v_2 + k_1 \oplus sum + (v_1 \oplus k_2) \gg 5 \end{cases} \quad (1)$$

$$g(k,v,sum):\begin{cases} v_0 \mathrel{+}= k_6 + v_1 \oplus (k_2 + sum \oplus k_3 + v_3) \ll 4 \\ v_1 \mathrel{+}= k_7 + v_0 \oplus (k_3 + sum \oplus k_0 + v_2) \gg 4 \end{cases} \quad (2)$$

Nilai *sum* di-*update* menggunakan rumus

$$sum = sum + golden \quad (3)$$

Bilangan *golden* itu sendiri berasal dari *golden* number, digunakan *golden*=$(\sqrt{5}\ -1)*2^{31}$. Suatu bilangan *golden* ganda yang berbeda digunakan pada setiap *round*-nya sehingga tidak ada bit dari proses enkripsi yang tidak berubah secara teratur.

Fungsi *f* dan fungsi *g* di atas digunakan karena merupakan fungsi dengan operasi yang sederhana sehingga akan membutuhkan waktu yang cepat untuk dieksekusi prosesor. *Golden number* digunakan sebagai konstanta awal nilai *sum* untuk menghindari nilai yang buruk pada saat inputan yang digunakan teks terang maupun teks sandi bernilai ekstrim.

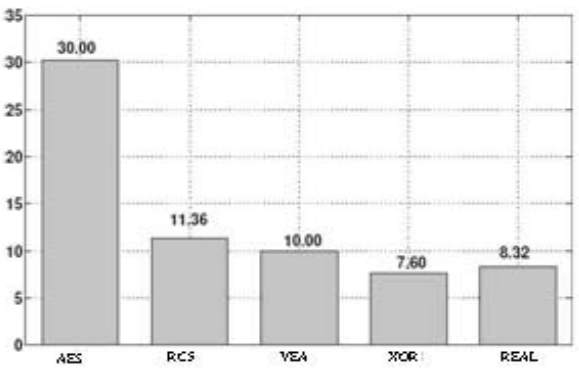
6. ANALISIS HASIL UJI REAL TERHADAP IMPLEMENTASI VIDEO CONFERENCE

REAL di desain khusus untuk implementasi *video conference.* Oleh karena itu pada *chapter* ini akan dibuktikan mengenai kecepatan, *memory usage*, *eficiency of implementation* dan *error propagation*. Pengujian dilakukan dengan melakukan perbandingan antara REAL terhadap algoritma AES, RC5 dan VEA.

6.1 Kecepatan

Perbandingan waktu kecepatan enkripsi REAL terhadap AES, RC5, VEA dan XOR biasa dapat terlihat pada gambar 5. Terlihat bahwa rata-rata waktu kecepatan enkripsi REAL per *frame* sekitar 8,32*ms*. Waktu enkripsi ini lebih kecil dibandingkan dengan waktu enkripsi AES (30,00*ms*), RC5 (11.36*ms*) dan VEA (10,00*ms*). Namun jika dibandingkan dengan enkripsi menggunakan operasi XOR biasa, REAL memiliki waktu enkripsi lebih besar dimana XOR biasa memiliki waktu 7,60*ms.*

Walaupun demikian, kekuatan kriptografis REAL lebih baik dibandingkan dengan XOR biasa. Ini yang membuat REAL lebih cocok digunakan sebagai algoritma enkripsi *video conference* yang bersifat *real time.*



Gambar 5. Perbandingan kecepatan enkripsi

6.2 Memori

Memori yang dibutuhkan untuk implementasi REAL adalah 58 *byte*. Jumlah memori ini sangat kecil untuk implementasi *video conference.* Dengan kebutuhan memori yang kecil, maka biaya implementasi akan semakin murah. Hal ini terkait dengan efisiensi implentasi.

6.3 Efisiensi Implementasi

Seperti pada gambar 6, Implementasi algoritma REAL pada *video conference* dilakukan dengan cara mengenkripsi *RTP payload.* Dengan cara ini maka hanya *payload data* yang akan terenkripsi.

Gambar 6. Paket enkripsi *video conference*

Efisiensi implementasi pada Java dan C++ menunjukkan bahwa algoritma REAL dapat di coding hanya dalam 12 baris saja. Sedangkan untuk implementasi REAL pada hardware, penulis melakukan simulasi *coding* kedalam *tiny* mikroprosesor menggunakan *CodeVisionAVR C Compiler V1.23.6a Standard*. Hasilnya menunjukkan bahwa hanya dengan menggunakan mikroprosesor jenis *tiny* saja REAL dapat diimplementasikan dengan baik. Pada gambar 7 merupakan *code* implementasi REAL menggunakan C++ .

```
void encrypt(unsigned long* v, unsigned long* k)  {
     unsigned long iv[4], output, datastream, sum=0, i;
     unsigned long golden=0x9e3779b9;      // constant //
     unsigned long k[8];            // cache key //
     for(i=0;i<11;i++){          // basic cycle start //
         sum+=golden;
         iv[2]+=k[4]^iv[3]+k[0]^sum+(iv[0]^k[1])<<5;
          iv[3]+=k[5]^iv[2]+k[1]^sum+(iv[1]^k[2])>>5;
          iv[0]+=k[6]+iv[1]^(k[2]+sum^k[3]+iv[3])<<4;
          iv[1]+=k[7]+iv[0]^(k[3]+sum^k[0]+iv[2])>>4;
         }                              // end cycle //
     output=iv^datastream;              // enciphered //
}
```

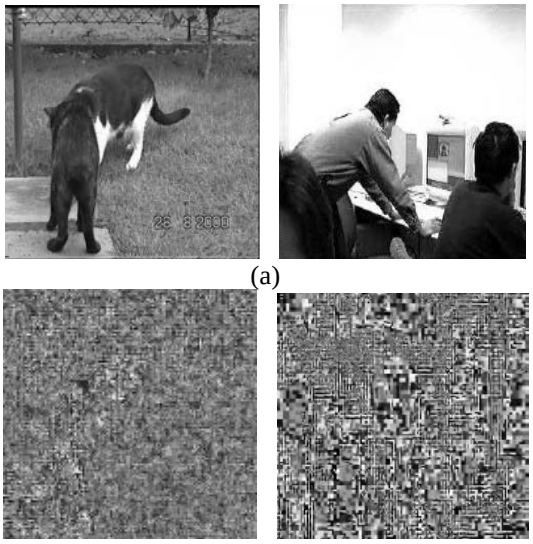
Gambar 7. Algoritma REAL dalam bahasa C++

6.4 *Error Propagation*

Pada algoritma REAL memiliki propagasi *error* yang kecil sebesar 128-bit. Hal ini dapat terjadi karena REAL menggunakan mode operasi CTR, sehingga jika terdapat satu bit *error* pada iv ataupun pada *data streaming encryption*-nya, maka yang akan mengalami *error* hanyalah sebanyak satu blok saja sebesar 128-bit.

6.5 Distribusi Keacakan Video Frame

Korespondensi antara frame video *plain* terhadap hasil frame video *cipher* dapat terlihat pada gambar 8. Hasilnya dapat disimpulkan bahwa detail blok-blok gambar video teracak secara sempurna melalui proses enkripsi.



(a)

(b)

Gambar 8. (a) Distribusi keacakan frame video *plain*
(b) Distribusi keacakan frame video *cipher*

7. KESIMPULAN

Dari hasil penelitian dapat disimpulkan bahwa REAL dapat menjadi alternatif solusi keamanan *video conference*. REAL merupakan algoritma enkripsi yang di disain khusus untuk *secure video conference* yang memiliki fitur kecepatan enkripsi per *frame* sebesar 8,32*ms*, membutuhkan memori minimum 58 *byte*, *easy software and hardware implementation*, *low error propagation* dan telah terbukti kekuatan kriptografisnya.

PUSTAKA

Ahmed, T., Mehaoua, A., Boutaba, R., Iraqi, Y. (2005). *Adaptive packet video streaming over IP networks: a cross-layer approach*. IEEE Journal on Selected Areas in Communications Volume 23

Andem, R. Virkam (2003). *A Cryptanalysis of The Tiny Encryption Algorithm.* A thesis from Department of Computer Science in the Graduate School of The University of Alabama.

I. Agi I and L. Gong (1996). *An empirical study of MPEG video transmission*. In Proc. of the Internet Society Symposium on Network and Distributed Systems Security.

Ibrahim, Subariah. Maarof, Mohd A dan Idris, Norbik B. *Avalanche Analysis od Extended Feistel Network.* Malaysia.

L. Qiao and K. Nahrstedt (1997). *A new algorithm for MPEG video encryption*. In Proc. of First International Conference on Imaging Science System and Technology.

Ma'ruf, F. Kholif (2008). *Desain Algoritma Block Cipher Vast Encryption Algorithm*. Tugas Akhir, Bogor: Sekolah Tinggi Sandi negara.

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

Schneier, Bruce (1996). *Applied Cryptography Second Edition Protocols Algorithms and Source Codes in C*. John Wiley & Sons, Inc. New York.

Schneier, Bruce., D.Whiting, *Rapid Software Encryption: Designing Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processo*r, Counterpane systems, Stac Electronics.

T. B. Maples and G. A. Spanos (1995). *Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-Time Video*. In Proc. of Fourth International Workshop on Multimedia Software Development '96.

W. Zeng and S. Lei (2002). *Efficient frequency domain selective scrambling of digital video*. In Proc. of the IEEE Transactions on Multimedia.

Xiao, Lu dan Heys, Howard M. (2003). *Hardware Performance Characterization of Block Cipher Structures.* RSA Conference 2003, San Francisco, LNCS 2612, Springer-Verlag.

Yan, J., Katrinis, K., May, M.; Plattner, B. (2006). *Media- and TCP-Friendly Congestion. Control for Scalable Video Streams*". IEEE Transactions on Multimedia, Volume 8.

Yucel, Melek D and Selcuk Kavut. *On Some Cryptographic Properties Of Rijndael.* Electrical and Electronics Engineering Dept. Middle East Technical University.