JITE, 2 (1) Juli 2018 ISSN 2549-6255 (Print) ISSN 2549-6247 (Online)

JITE

(Journal Of Informatics And Telecommunication Engineering)

Available online http://ojs.uma.ac.id/index.php/jite

Perbandingan Algoritma Blowfish Dan Twofish Untuk Kriptografi File Gambar

Comparison Of Blowfish And Twofish Algorithms For Image File Cryptography

Muhathir[1)]*

[1)] Prodi Informatika, Fakultas Teknik Universitas Medan Area, Indonesia

*Coresponding Email: muhathir@staff.uma.co.id

Abstrak

Masalah keamanan dan kerahasiaan data dan informasi merupakan suatu hal yang sangat penting. Salah satu cara untuk menjaga data tersebut yaitu dengan teknik enkripsi dan dekripsi atau juga sering disebut dengan kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga keamanan data dengan cara mengubahnya menjadi suatu bentuk yang tidak dapat dikenali lagi. Kriptografi dapat diterapkan pada berbagai jenis file salah satunya adalah gambar. Algoritma blowfish dan twofish yang digunakan untuk membahas tingkat kecepatan masing-masing algoritma. Hasil penelitian menunjukkan bahwa rata-rata perbandingan kecepatan dari algoritma Blowfish dan algoritma Twofish dalam satuan milidetik adalah 4355:4267.

Kata Kunci: Kriptografi, Blowfish, Twofish.

Abstract

The issue of security and confidentiality of data and information is very important. One way to keep the data is by encryption and decryption techniques or also often called cryptography. Cryptography is the science and art of keeping data secure by turning it into an unrecognizable form. Cryptography can be applied to different types of files one of which is an image. Blowfish and twofish algorithms are used to discuss the level of each algorithm's density. The results showed that the average speed ratio of the blowfish algorithm and the twofish algorithm in milliseconds was 4355: 4267.

Keywords : Cryptography, Blowfish, Twofish.

How to Cite: Muhathir. (2018). Perbandingan Algoritma Blowfish Dan Twofish Untuk Kriptografi File Gambar. JITE (Journal Of Informatics And Telecommunication Engineering). 2 (1): 23-32

PENDAHULUAN

Kriptografi merupakan pembagian kriptologi di mana algoritma enkripsi / dekripsi dirancang, untuk menjamin keamanan dan otentikasi data". Kriptografi dapat diklasifikasikan sebagai algoritma kunci simetris dan algoritma kunci asimetris[1][2], kunci simetris atau juga dikenal sebagai kunci private yaitu sebuah kunci tunggal yang dugunakan untuk mengenkripsi dan mendekripsi teks biasa, sedangkah kunci asimetris yaitu dua kunci berbeda yang digunakan, satu untuk enkripsi disebut kunci publik dan dekripsi dilakukan oleh kunci lain yang disebut juga sebagai kunci private[3][4].

Kriptografi juga bisa digunakan untuk mengamankan data seperti file text, gambar, suara dan lain sebagainya. Untuk kasus ini penulis akan menyandikan file gambar, yang pada dasarnya banyak file gambar digunakan dalam berbagai bidang seperti kemanan, medis, ilmu, teknik, seni dan lain sebagainya yang digunakan sebagai informasi yang berharga dan dapat menjadi bersifat pribadi sehingga dibutuhkan kriptografi file gambar.

Blowfish dan Twofish keduanya dirancang oleh orang yang sama, yaitu Bruce Schneier. Namun keduanya berasal dari masa yang berbeda. Blowfish merupakan algoritma yang lebih tua, ia dirancang pada tahun 1993. Tujuan perancangannya adalah untuk menggantikan algoritma DES yang sudah sangat tua (sejak 1977). Algoritma DES sendiri merupakan algoritma standar kriptografi yang ditetapkan oleh NIST – sebuah lembaga yang mengatur tentang standar-standar. Antara DES dan blowfish memiliki banyak kesamaan. Kesamaan mendasar adalah keduanya memiliki panjang blok yang sama, yaitu 64 bit. Namun tentu blowfish karena jauh lebih muda, memiliki banyak kelebihan dibandingkan dengan DES. Blowfish sangat terkenal di dunia kriptografi, alasan utamanya adalah karena lisensinya yang bebas dan gratis. Bahkan komunitas open source menghargai blowfish dengan mempercayai blowfish menjadi salah satu *Open Cryptography Interface* (OCI) pada kernel Linux versi 2.5 keatas.

Sedangkan twofish adalah suksesor dari blowfish. Pada tahun 1997, dengan kemajuan teknologi prosesor yang sangat cepat, maka bukanlah ahal yang sulit untuk menjebol algoritma kriptografi dengan panjang kunci 64 bit. Untuk itu NIST membuka sayembara untuk umum. Semua pihak boleh mensubmit algoritmanya, namun tentu dengan syarat-syarat kualitas minimum, seperti panjang blok minimum 128 bit. Twofish merupakan salah satu peserta, dan berhasil meraih posisi 5 besar, dan bahkan secara tidak resmi

mendapatkan posisi 2 besar. Lima besar algoritma yang lolos semuanya memiliki tingkat keamanan yang hampir seimbang, sehingga penilaian disampingkan menjadi performansi kecepatan. Twofish dan rijndael bersaing memperebutkan posisi teratas. Twofish unggul di nomor kunci 256 bit, namun kalah di nomor kunci 128 bit. Akhirnya rijndael yang keluar sebagai pemenang dan berubah nama menjadi AES, suksesor dari DES.

Twofish dan Blowfish keduanya memiliki keunggulan tersendiri dibandingkan rivalnya (DES dan AES). Dan diantara keduanya pun masing - masing memiliki keunggulan dan kelemahan masingmasing. Untuk itu dirasakan perlu adanya sebuah studi pembandingan kedua algoritma tersebut secara mendalam. Mulai dari teknik-teknik yang dipakai oleh keduanya, hingga performansi keduanya

Pada kesempatan ini penulis ingin membandingkan tingkat kecepatan algoritma blowfish dan twofish dalam proses mengenkripsi dan dekripsi *file* gambar.

METODE PENELITIAN

Kriptografi adalah teknik mengirim data dari pengirim ke penerima. Ini adalah proses konversi data menjadi kode rahasia untuk pengiriman melalui jaringan publik. Teks asli diubah menjadi setara kode disebut "*cipher teks* ". Ada dua proses yang terjadi di teknologi kriptografi. Yaitu Enkripsi dan Dekripsi. Enkripsi adalah proses mengaburkan Informasi untuk membuatnya tidak terbaca tanpa penetahuan khusus. Dengan kata lain itu juga digambarkan sebagai konversi pesan yang dapat dibaca manusia, yang dikenal sebagai baik teks biasa atau teks yang jelas, dalam cipher teks yang orang yang tidak berhak tidak dapat dengan mudah mengerti. Hal ini biasanya dilakukan untuk kerahasiaan, dan biasanya untuk informasi komunikasi rahasia. Enkripsi juga digunakan untuk otentikasi. ada dua jenis algoritma enkripsi yaitu simetris dan asimetris[3]. kunci simetris atau juga dikenal sebagai kunci private yaitu sebuah kunci tunggal yang dugunakan untuk mengenkripsi dan mendekripsi teks biasa, sedangkah kunci asimetris yaitu dua kunci berbeda yang digunakan, satu untuk enkripsi disebut kunci publik dan dekripsi dilakukan oleh kunci lain yang disebut juga sebagai kunci private[3] Dekripsi adalah proses mengembalikan informasi dari format tidak terbaca dengan format yang dapat dibacaakhir, kunci digunakan untuk membuka kode dan mengembalikan data asli. [4].

Algoritma blowfish

Blowfish adalah algoritma kriptografi kunci simetrik cipher blok dengan panjang blok tetap sepanjang 64 bit. Blowfish menerapkan teknik kunci yang berukuran sembarang. Ukuran kunci yang dapat diterima oleh blowfish adalah antara 32 bit hingga 448 bit, dengan ukuran default sebesar 128 bit. Blowfish memanfaatkan teknik pemanipulasian bit (subbab 3.1), kotak permutasi (subbab 3.2), jaringan feistel (subbab 3.3), dan teknik pemutaran ulang dan pergiliran kunci (subbab 3.4) yang dilakukan sebanyak 16 kali. Algoritma utama terbagi menjadi dua subalgoritma utama, yaitu bagian ekspansi kunci dan bagian enkripsi-dekripsi data. Pengekspansian kunci dilakukan pada saat awal dengan masukan sebuah kunci dengan panjang 32 bit hingga 448 bit, dan keluaran adalah sebuah array subkunci dengan total 4168 byte. Bagian enkripsi-dekripsi data terjadi dengan memanfaatkan perulangan 16 kali terhadap jaringan feistel. Setiap perulangan terdiri dari permutasi dengan masukan adalah kunci, dan substitusi data. Semua operasi dilakukan dengan memanfaatkan operator Xor dan penambahan. Operator penambahan dilakukan terhadap empat array lookup yang dilakukan setiap putarannya[5].

Blowfish menggunakan subkunci berukuran besar. Kunci-kunsi tersebut harus dikomputasikan pada saat awal, sebelum pengkomputasian enkripsi dan dekripsi data. Langkah-langkahnya adalah sebagai berikut:

1. Terdapat kotak permutasi (P-box) yang terdiri dari 18 buah 32 bit sub kunci: P1, P2, P3, ... P18. P-box ini telah ditetapkan sejak awal, 4 buah P-box awal adalah sebagai berikut:

   P1 = 0x243f6a88

   P2 = 0x85a308d3

   P3 = 0x13198a2e

   P4 = 0x03707344

2. P1 di Xor dengan 32 bit awal kunci, P2 di Xor dengan 32 bit berikutnya dari kunci, dan teruskan hingga seluruh panjang kunci telah terxorkan (kemungkinan sampai P14, 14x32 = 448, panjang maksimal kunci).

3. Terdapat 64 bit dengan isi kosong, bit-bit tersebut dimasukkan ke langkah 2.

4. Gantikan P1 dan P2 dengan keluaran dari langkah 3.

5. Enkripsikan keluaran langkah 3 dengan langkah 2 kembali, namun kali ini dengan subkunci yang berbeda (sebab langkah 2 menghasilkan sub kunci baru).

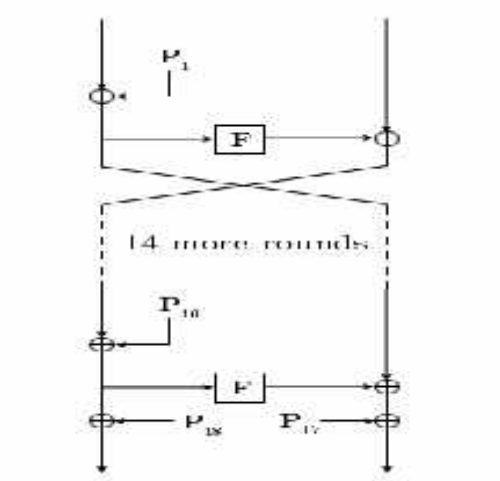
6. Gantikan P3 dan P4 dengan keluaran dari langkah 5.

7. Lakukan seterusnya hingga seluruh P-box teracak sempurna.

8. Total keseluruhan, terdapat 521 iterasi untuk menghasilkan subkunci-subkucni yang dibutuhkan[6].

Aplikasi hendaknya menyimpannya daripada menghasilkan ulang subkunci-subkunci tersebut.

Proses enkripsi-dekripsi data pada algoritma blowfish adalah sebagai berikut:



Gambar 1 – Proses enkripsi Blowfish

1. Masukan dari proses ini adalah 64 bit data yang diinisialkan “x”.

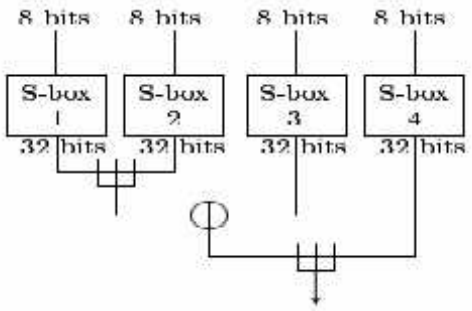
2. Bagi x menjadi 2 buah bagian sama besar, xL (x kiri) sepanjang 32 bit, dan xR (x kanan) sepanjang 32 bit.

3. Lakukan iterasi sebanyak i=1 hingga i=16: xL = xL XOR P[i]; xR = F(xL) XOR xR; Swap(xL, xR);

4. Fungsi F adalah sebagai berikut: bagi xL menjadi 4 buah 8 bit a, b, c, dan d. F(xL) = ((S[1,a] + S2[2,b] mod 232) XOR S[3,c]) + S[4,d] mod 232.

5. Langkah terakhir adalah: Swap(xL, xR); xR = xR XOR P[17]; xL = xL XOR P[18]; gabungkan xL dan xR menjadi 64 bit return hasil gabungan

6. Pada proses dekripsi langkah-langkahnya sama persis dengan proses enkripsi, namun hanya saja P-box digunakan dengan urutan yang terbalik.



Gambar 2 – Proses ekspansi dan filter (fungsi F) pada Blowfish

Algoritma twofish

Twofish merupakan algoritma kriptografi kunci simetrik cipher blok dengan panjang setiap blok adalah tetap 128 bit. Sedangkan kunci yang dapat diterima adalah: 128, 192, atau 256 bit. Seperti halnya blowfish, twofish juga memiliki dua tahapan utama, yaitu tahap pembangkitan kunci dan tahap algoritma utama[4].

Jumlah kunci internal yang harus dibangkitkan adalah sejumlah 40 kunci masing-masing 32 bit (K0 hingga K39). Dan juga dibutuhkan pembangkitan 4 buah kotak substitusi dari yang bergantung pada kunci. Twofish dapat menerima kunci sepanjang 128, 192, dan 256 bit (N). Kemudian terdefinisi k=N/64. Kunci M terdiri dari 8k byte, m0, ..., m8k-1. Byte-

byte tersebut pertama-tama diubah menjadi 2k buah yang masing-masing terdiri dari 32 bit.

$$M_i = \sum_{j=0}^{3} m_{(4i+j)} \cdot 2^{8j} \qquad i = 0, \ldots, 2k-1$$

Hasil fungsi di atas kemudian digolongkan menjadi dua buah, ganjil dan genap.

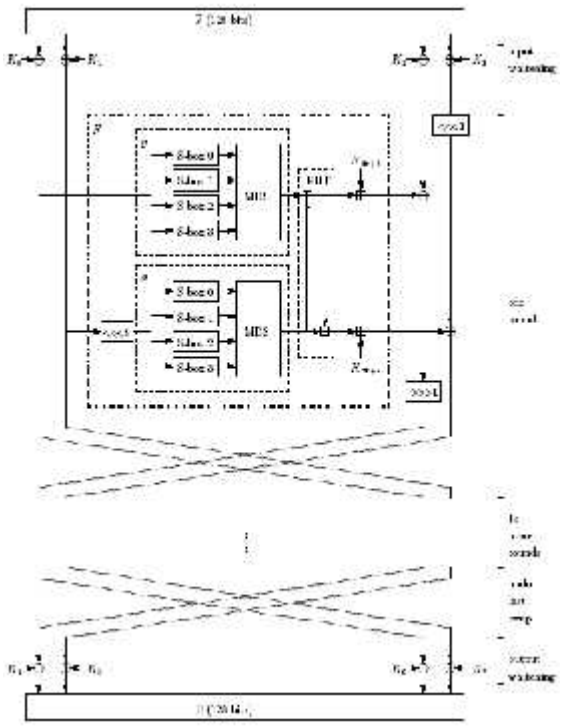
Me = (M0, M2, ..., M2k-2)

Mo = (M1, M3, ..., M2k-1)

Selanjutnya adalah kotak S. Langkah pertama adalah dengan mengelompokkan kunci menjadi masing-masing 8. Kemudian kelompok kunci tersebut dikalikan dengan matriks 4x8 yang diturunkan dari RS. Setiap hasil sepanjang 4 byte duartikan sebagai satu buah 32 bit, menghasilkan kotak S.

$$\mathrm{RS} = \begin{pmatrix} 01 & A4 & 55 & 87 & 5A & 58 & DB & 9E \\ A4 & 56 & 82 & F3 & 1E & C6 & 68 & E5 \\ 02 & A1 & FC & C1 & 47 & AE & 3D & 19 \\ A4 & 55 & 87 & 5A & 58 & DB & 9E & 03 \end{pmatrix}$$

Hasil keluaran tahap ini adalah 2 buah matriks, matriks M genap dan matriks M ganjil, dan sebuah matriks kotak substitusi.



Gambar 3 – Algoritma twofish

Langkah-langkah algoritma twofish adalah sebagai berikut:

1. Masukan satu blok plain teks adalah 128 bit. Satu blok tersebut dibagi menjadi 4 buah subblok yang masing-masing sepanjang 32 bit (A, B, C, dan D).

2. Masing-masing subblok tersebut diputihkan dengan mengxorkan dengan kunci K0, K1, K2, dan K3.

Langkah-langkah 1 putaran dalah sebagai berikut:

1. dua buah 32 bit yang kiri (A dan B) merupakan input dari fungsi g (yang merupakan bagian dari fungsi f), yang salah satunya (B) di geser ke kiri sejauh 8 bit dahulu.

2. Fungsi g memiliki 4 buah kotak substitusi yang dibangkitkan oleh kunci.

3. Keluaran fungsi kotak substitusi dilakukan percampuran linear menggunakan kotak Most Distance Separable (subbab 3.7).

4. Keluaran fungsi g dimasukkan ke fungsi transformasi pseudo-Hadamard, kemudian ditambahkan dengan 2 buah 32 bit dari kunci.

5. Dua buah 32 bit hasil kemudian di xorkan dengan C dan D. Hasil xor dengan C digeser ke kanan sejauh 1 bit. Dan untuk D sebelum dixorkan digeser ke kiri sejauh 1 bit.

6. dua buah 32 bit kiri dan kanan dipertukarkan (A dan B dipertukarkan dengan C dan D)[7].

Langkah diatas dilakukan hingga 16 kali putaran. Kemudian langkah-langkah selanjutnya:

Hasil keluaran setelah diputar 16 kali, ditukar lagi (A dan B diperukarkan dengan C dan D).

Hasil dari pertukaran tersebut di xorkan dengan empat buah 32 bit dari kunci menghasilkan cipher teks.

Fungsi F

Fungsi F adalah permutasi yang bergantung pada kunci dengan nilai 64 bit. Fungsi ini menerima 3 argumen, dua buah 32 bit R0 dan R1, dan nomor putaran untuk menentukan subkunci mana yang dipakai. R0 akan diserahkan ke fungsi g yang akan mengembalikan T0. R1 akan digeser sejauh 8 bit yang kemudian di berikan juga ke fungsi g yang akan mengembalikan T1. Hasil T0 dan T1 kemudian dikombinasikan ulang menggunakan transformasi pseudo-Hadamard, yang kemudian ditambahkan dengan dua buah 32 bit dari kunci.

T0 = g(R0);

T1 = g(shiftLeft(R1,8));

F0 = (T0+T1+K2r+8) mod 232;

F1 = (T0+2T1+K2r+9) mod 232;

F0 dan F1 adalah hasil dari F, yang masingmasing sepanjang 32 bit. Hasil keluaran ini nantinya akan dipertukarkan dan dimasukkan kembali ke putaran selanjutnya.

Fungsi G

Fungsi g merupakan jantung dari keseluruhan algoritma twofish. 32 bit masukan X dari fungsi F dipecah menajdi 4 buah yang masingmasing sepanjang 8 bit. Setiap 8 bit kemudian diproses dengan kotak S yang bersesuaian. Setiap kotak S bersifat bijektif, yaitu menerima 8 bit dan mengeluarkan 8 bit pula. 4 buah 8 bit hasil keluaran kemudian dikalikan dengan matriks Most Distance Separable (MDS)

4x4. Hasil pengalian kemudian diartikan sebagai 32 bit, yang merupakan keluaran dari fungsi g, yang kemudian akan dikembalikan kembali ke fungsi F.

Matriks MDS yang setiap elemennya ditampilkan sebagai heksadesimal adalah sebagai berikut:

$$\text{MDS} = \begin{pmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{pmatrix}$$

Bitmap (*.bmp)*

Citra bitmap adalah susunan bit-bit warna untuk tiap pixel yang membentuk pola tertentu. Pola-pola warna ini menyajikan informasi yang dapat dipahami sesuai dengan persepsi indera penglihatan manusia. Format file ini merupakan format grafis yang fleksibel untuk platform Windows sehingga dapat dibaca oleh program grafis manapun. Format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit sampai 24 bit

HASIL DAN PEMBAHASAN

Implementasi program

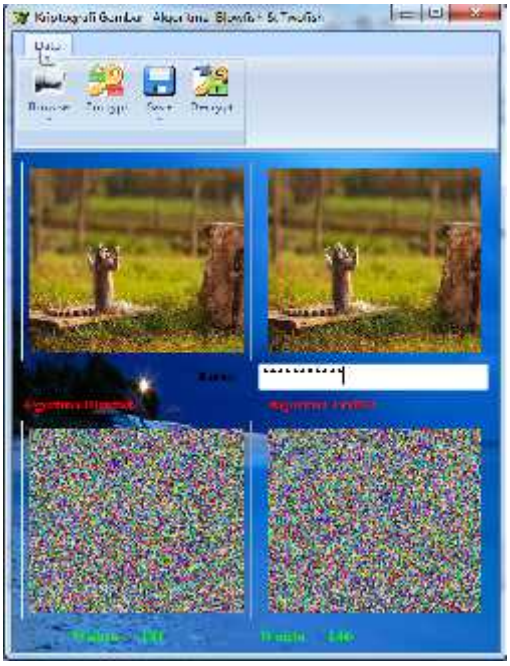
Pada halaman ini terdapat 4 tombol, yaitu : Browse, Encrypt, Save dan Decrypt

Gambar 4. Halaman utama Aplikasi

Untuk melakukan enkripsi/dekripsi file gambar sebelumnya user harus memilih gambar yang akan di enkripsi/dekripsi dan menginput kata kunci sebagai kunci file yg akan di enkripsi/dekripsi.

Pengujian aplikasi

Pada tahapan ini dilakukan pengujian aplikai untuk mengenkripsi file gambar dan melihat kecepatan waktu proses dari algoritma blowfish dan twofish. Berikut hasil pengujian enkripsi dan dekripsi



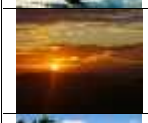
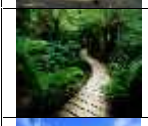
Gambar 5. Proses Enkripsi

Gambar 6. Proses Dekripsi

Hasil Pengujian terhadap Ukuran File dan Waktu Proses

Hasil pengujian proses enkripsi dan dekripsi pada berbagai ukuran pixel gambar dapat dilihat pada tabel 1 dan table 2.

Tabel 1. Hasil Proses Enkripsi

Citra	Ukuran	Kecepatan proses Enkripsi (millisecond)	
		Blowfish	Twofish
	256 x 192	26	18
	400 x 400	88	50
	640 x 480	192	162
	1200 x 900	923	892
	1600 x 1000	1704	1375
	1600 x 1200	2297	2242
	3344 x 2224	25261	25133

Tabel 2. Hasil Proses Dekripsi

Citra	Ukuran	Kecepatan proses Dekripsi (millisecond)	
		Blowfish	Twofish
	256 x 192	32	19
	400 x 400	93	70
	640 x 480	219	161
	1200 x 900	976	948
	1600 x 1000	1763	1696
	1600 x 1200	2251	2193

	3344 x 2224	25191	23953

Dari hasil diatas dilihat bahwa besarnya ukuran *file* mempengaruhi waktu atau lamanya proses enkrisi dan dekripsi.

SIMPULAN

Berdasarkan pengujian apalikasi diatas dapat diambil kesimpulan bahwa algoritma Twofish lebih cepat dibandingkan dari algoritma Blowfish untuk proses enkripsi begitu juga untuk proses dekripsi. Rata-rata perbandingan kecepatan dari algoritma blowfish dan algoritma twofish dalam satuan milidetik adalah 4355:4267. Kecepatan proses enkripsi/dekripsi masing-masing algoritma tergantung pada besarnya ukuran file, semakin besar file yang diproses semakin banyak waktu yang dibutuhkan untuk proses enkripsi/dekripsi.

DAFTAR PUSTAKA

[1] Ebrahim Mansoor,dkk. (2013). Symmetric Algorithm Survey: A Comparative Analysis International Journal of Computer Applications (0975 – 8887) Volume 61– No.20.

[2] Lalit Singh, dkk. (2013) Comparative Performance Analysis of Cryptographic Algorithms. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 3, Issue 11,

[3] Pallavi H.Dixit, dkk. (2013) Comparative Implementation of Cryptographic Algorithms on ARM Platform. International Journal of Innovative Research in Science, Engineering and Technology. Vol. 2, Issue 10

[4] K.Durgadevi. (2011) Implementation Of Secure Master Using Modified Twofish Algorithm In Fpga Devices. International Journal of Engineering Research and Applications (IJERA), Vol. 1, Issue 3, pp.507-512

[5] Octamanullah Mohamad. Perbandingan Algoritma Kriptografi Kunci Simetrik BlowFish dan TwoFish

[6] Ashwak ALabaichi (2013) Randomness Analysis of 128 bits Blowfish Block Cipher on ECB mode. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 10,

[7] Purnima Gehlot. (2013) Implementation of Modified Twofish Algorithm using 128 and 192-bit keys on VHDL. International Journal of Computer Applications (0975 – 8887) Volume 70– No.13