Analisis Serangan dengan *Selective Plaintext* pada Sebuah Algoritma Enkripsi Citra Berbasis *Chaos*

Rinaldi Munir [1)]

[1)] Program Studi Informatika, Sekolah Teknik Elektro dan Informatika (STEI), ITB
Jl. Ganesha 10, Bandung 40132
email : rinaldi-m@stei.itb.ac.id

ABSTRAK

Enkripsi citra sederhana dengan teknik meng-XOR-kan pixel-pixel plain-image dengan kunci tidak aman terhadap serangan known-plaintext attack. Dengan memilih sejumlah plain-image dan cipher-image yang berkoresponden, kunci dapat ditemukan dengan mudah. Varian known-plaintext attack semacam ini dinamakan selective plaintext attack. Makalah ini memaparkan analisis serangan selective-plaintext terhadap sebuah usulan algoritma enkripsi citra berbasis chaos. Algoritma tersebut menggabungkan teknik permutasi (mengguankan Arnold Cat Map) dan teknik substitusi (menggunakan operasi XOR). Berdasarkan serangkaian eksperimen yang telah dilakukan dapat disimpulkan bahwa mengacak pixel-pixel sebelum operasi XOR dapat membuat algoritma aman terhadap serangan selective-plaintext.

Kata Kunci:

Enkripsi citra, chaos, selective-plaintext attack.

1. Pendahuluan

Keamanan informasi merupakan topik yang penting saat ini. Informasi dalam bentuk data multimedia seperti citra digital perlu dilindungi dari pengaksesan yang ilegal, baik pada citra yang ditransmisikan maupun pada citra yang tersimpan. Masalah keamanan ini dapat ditangani dengan mengenkripsi citra tersebut. Enkripsi citra adalah proses menyandikan citra ke bentuk visual lain yang tidak bermakna lagi. Penelitian tentang enkripsi citra berbasis *chaos* merupakan topik yang menarik dan hingga saat ini sudah banyak algoritma enkripsi yang sudah diusulkan.

Kebanyakan algoritma enkripsi citra dalam ranah spasial melakukan pengubahan nilai *pixel* dengan menggunakan kunci rahasia. Salah satu operasi yang umum ditemukan pada kebanyakan algoritma enkripsi citra dalam ranah spasial adalah operasi *exclusive-or* (XOR). Nilai *pixel* di-XOR-kan dengan kunci yang dibangkitkan dari sebuah pembangkit bilangan acak.

Operasi XOR digunakan dalam enkripsi citra karena relatif cepat dan mudah diimplementasikan. Namun, enkripsi dengan operasi XOR memiliki kelemahan, yaitu rawan terhadap serangan menggunakan *selective plaintext*. Penelitian di dalam [1, 2, 3] menyatakan bahwa *plain-image* dapat ditemukan melalui serangan *selective plaintext* meskipun pihak lawan tidak mengetahui kunci dan algoritma yang digunakan. Hongmei di dalam [4] mendemonstrasikan melalui *k* buah *plain-image* dan *k* buah *cipher-image* yang dienkripsi dengan algoritma dan kunci yang sama, ternyata kriptanalis dapat mendeduksi *plain-image* dari *cipher-image* dan beberapa *selective plain-image*.

Makalah ini mempresentasikan analisis serangan *selective plaintext* pada algoritma enkripsi citra yang diusulkan di dalam [5]. Tujuan analisis ini adalah untuk mengetahui apakah serangan tersebut dapat berhasil menemukan *plain-image* dari *cipher-image* yang diperoleh dari algoritma tersebut. Melalui serangkaian eksperimen seperti yang dilakukan oleh Hongmei di dalam [4], akan dibuktikan keamanan algoritma tersebut terhadap serangan *selective plaintext*.

2. Dasar Teori

2.1 Operator XOR dan Sifat-Sifatnya

Operator XOR adalah operator boolean yang sering ditemukan di dalam algoritma enkripsi yang beroperasi dalam mode bit. Nilai boolean *true* dapat disamakan dengan 1 dan nilai *false* sama dengan 0. Aturan operasi bit dengan XOR adalah sebagai berikut:

$0 \oplus 0 = 0; \quad 0 \oplus 1 = 1; \quad 1 \oplus 0 = 1; \quad 1 \oplus 1 = 0.$

Misalkan *a, b,* dan *c* adalah peubah boolean, maka sifat-sifat ini dipenuhi oleh operator XOR:

(i) $a \oplus a = 0$
(ii) $a \oplus 0 = a$
(iii) $a \oplus b = b \oplus a$ (Hukum Komutatif)
(iv) $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ (Hukum Asosiatif)

Jika dua rangkaian bit dioperasikan dengan XOR, maka operasinya dikerjakan dengan meng-XOR-kan setiap bit-bit yang berkoresponden pada dua rangkaian bit tersebut.

Algoritma enkripsi yang sederhana meng-XOR-kan plainteks (*P*) dengan kunci (*K*) dan menghasilkan cipherteks *C* dengan persamaan:

$$C = P \oplus K \qquad (1)$$

Karena meng-XOR-kan plainteks dua kali berturut-turut menghasilkan nilai semula, maka dekripsi dilakukan dengan persamaan:

$$P = C \oplus K \qquad (2)$$

Bukti: $C \oplus K = (P \oplus K) \oplus K$ (substitusi dengan (1))
$= P \oplus (K \oplus K)$ (Hukum (iii))
$= P \oplus 0$ (Hukum (i))
$= P$ (Terbukti)

Serangan yang lazim ditemukan pada algoritma enkripsi sederhana dengan XOR adalah *known-plaintext attack* dan *ciphertext-only attack*. Pada *known-plaintext attack*, jika *K* adalah kunci yang sama digunakan untuk mengenkripsi bermacam-macam plainteks, maka jika sebuah cipherteks *C* dan plainteks yang berkoresponden *P* diketahui, maka kunci *K* dapat ditemukan dengan meng-XOR-kan *P* dan *C* sebagai berikut:

$$\begin{aligned} P \oplus C &= P \oplus (P \oplus K) \\ &= (P \oplus P) \oplus K \\ &= 0 \oplus K \\ &= K \end{aligned}$$

Kunci *K* yang dihasilkan ini dapat digunakan untuk mendekripsi plainteks yang lain dengan persamaan (2).

Pada *ciphertext-only attack*, kriptanalis memiliki dua cipherteks C_1 dan C_2 yang keduanya dienkripsi dengan kunci *K* yang sama. Dengan meng-XOR-kan kedua cipherteks ini diperoleh dua plainteks yang ter-XOR satu sama lain:

$$\begin{aligned} C_1 \oplus C_2 &= (P_1 \oplus K) \oplus (P_2 \oplus K) \\ &= (P_1 \oplus P_2) \oplus (K \oplus \mathrm{K}) \\ &= (P_1 \oplus P_2) \oplus 0 \\ &= (P_1 \oplus P_2) \end{aligned}$$

Jika salah satu P_1 atau P_2 dapat diterka, maka kunci *K* dapat ditemukan.

2.2 Arnold Cat Map

Arnold Cat Map (*ACM*) adalah fungsi *chaos* 2-D yang mentransformasikan koordinat (*x*, *y*) di dalam citra yang berukuran $N \times N$ ke koordinat baru (*x*', *y*'). Persamaan iterasinya adalah

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod(N) \qquad (3)$$

yang dalam hal ini *p* dan *q* adalah *integer* positif sembarang. Dimensi matriks *ACM* harus 1 untuk menjamin hasil transformasinya berada dalam area citra. *ACM* diiterasikan sebanyak *m* kali dan setiap iterasi menghasilkan citra yang acak. Citra yang sudah diacak oleh *ACM* dapat direkonstruksi menjadi citra semula dengan menggunakan parameter nilai yang sama (*p*, *q*, dan *m*). Persamaan *invers ACM* adalah

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \bmod(N) \qquad (4)$$

Nilai *p*, *q*, dan jumlah iterasi *m* dapat dianggap sebagai kunci yang harus tetap dijaga kerahasiaannya.

2.3 Logistic Map

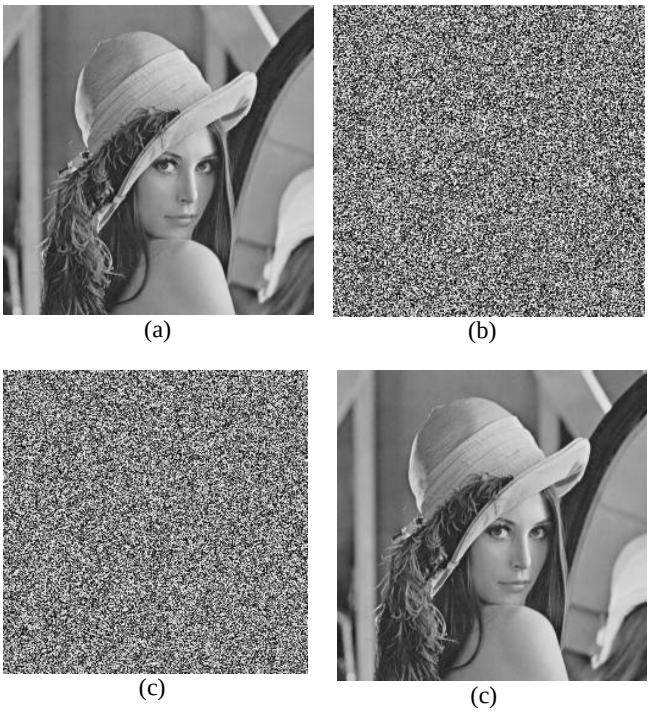
Logistic Map adalah fungsi *chaos* satu dimensi yang didefinsikan sebagai

$$x_{i+1} = \mu\, x_i\, (1 - x_i) \qquad (5)$$

yang dalam hal ini $0 < x_i < 1$ dan $0 < \mu \le 4$. Untuk memulai iterasi *Logistic Map* diperlukan nilai awal x_0. Nilai awal *chaos*, x_0, dan parameter μ berperan sebagai kunci rahasia. Agar bisa menjadi *keystream*, maka nilai-nilai acak x_i ditransformasikan menjadi *integer*. Cara transformasi yang sederhana adala dengan mengambil bagian desimal dari bilangan riil, membuang angka nol yang tidak signifikan, lalu mengekstrak sejumlah digit *integer*.

3. Algoritma Enkripsi Sederhana dengan XOR

Misalkan $P = \{p_{ij}\}$adalah citra *plain-image* yang berukuran $m \times n$, $K = \{k_{ij}\}$ adalah matriks kunci yang berukuran $m \times n$, maka enkripsi citra sederhana dengan operator XOR menghasilan citra *cipher-image* $C = \{c_{ij}\}$ sedemikian sehingga $c_{ij} = p_{ij} \oplus k_{ij}$. Dengan kata lain, setiap *pixel* di dalam *plain-image* dienkripsi dengan elemen kunci yang berbeda.



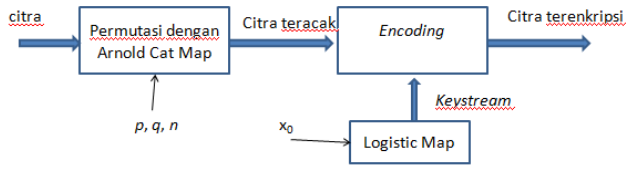
(a) (b) (c) (c)

Gambar 1. Citra 'Lenna'; (a) *plain-image*; (b) *cipher-image*; (c) kunci yang digunakan; (d) *plain-image* hasil dekripsi

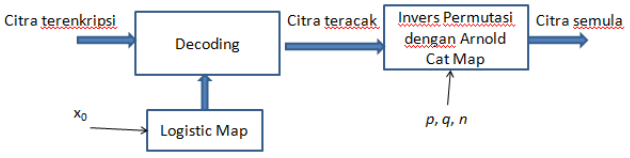
Gambar 1 memperlihatkan enkripsi terhadap citra 'Lenna' dengan algoritma XOR sederhana. Gambar 1(a) adalah *plain-image* dari citra 'Lenna', Gambar 1(b) adalah *cipher-image* dari citra 'Lenna'. Matriks *K* yang digunakan adalah sekumpulan bilangan bulat acak yang nilainya dari 0 sampai 255. Representasi citra dari matriks *K* diperlihatkan pada Gambar 1(c). Hasil dekripsi terhadap *cipher-image* dengan kunci *K* yang sama menghasilkan kembali citra *plain-image* semula (Gambar 1(c)).

4. Tinjauan Singkat Algoritma Enkripsi Citra Berbasis *Chaos* yang Diusulkan

Sebuah algoritma enkripsi citra berbasis *chaos* telah dipresentasikan di dalam [5]. Algoritma tersebut menggabungkan teknik permutasi dan substitusi. Dua buah *chaotic map* digunakan pada masing-masing teknik yaitu *Arnold Cat Map* (*ACM*) dan *Logistic Map*. Garis besar algoritma enkripsinya adalah mengacak terlebih dahulu *pixel-pixel* dengan *Arnold Cat Map*, selanjutnya, *pixel-pixel* tersebut diubah nilainya melalui operasi *XOR* dengan *keystream* yang dibangkitkan dari *Logistic Map*. Algoritma dekripsi merupakan kebalikan dari algoritma menkripsi. Gambar 2 memperlihatkan diagram enkripsi dan dekripsi di dalam algoritma tersebut.



(a)



(b)

Gambar 2. (a) Diagram enkripsi; (b) diagram dekripsi dari algoritma enkripsi citra yang diusulkan di dalam [5]

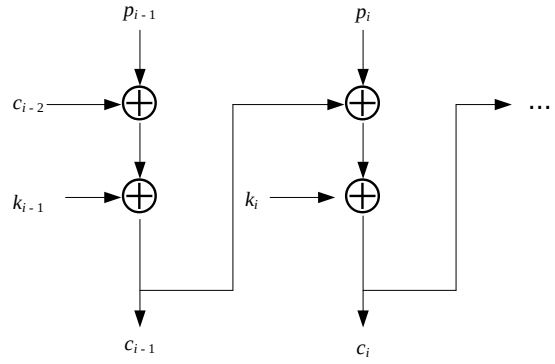
Proses *encoding* pada tahap enkripsi dilakukan dengan persamaan berikut:

$$c_i = (p_i \oplus c_{i-1}) \oplus k_i \qquad (6)$$

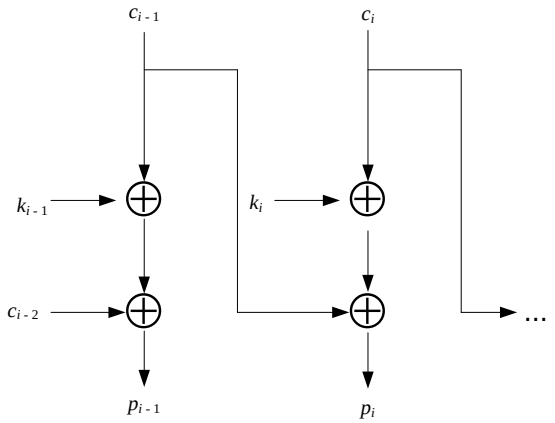
sedangkan proses *decoding* pada tahap dekripsi menggunakan persamaan:

$$p_i = (c_i \oplus k_1) \oplus c_{i-1} \qquad (7)$$

Gambar 3 memperlihatkan masing-masing diagram *encoding* dan *decoding* pada tahap enkripsi dan dekripsi yang dimaksudkan.



(a)



(b)

Gambar 3. (a) Diagram *encoding*; (b) diagram *decoding* [5]

5. Eksperimen *Selective Plaintext Attack*

Selective plaintext attack termasuk ke dalam *known-plaintext attack*, yang dalam hal ini penyerang dapat memilih sejumlah plainteks dan cipherteks yang berkoresponden untuk mendeduksi kunci.

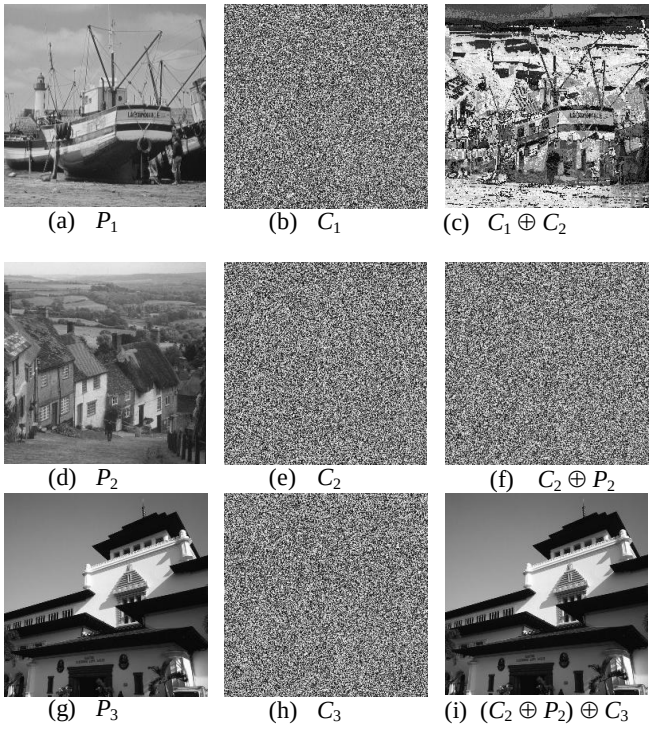
Misalkan anda mempunyai koleksi foto di dalam sebuah basis data. Jika setiap foto dienkripsi dengan kunci yang berbeda dan dengan algoritma enkripsi yang berbeda (untuk memaksimalkan keamanan) maka anda harus mengingat semua kunci.dan algoritma yang digunakan. Cara seperti ini ini jelas tidak praktis. Cara yang lebih mangkus (efisien) adalah mengenkripsinya dengan algoritma yang sama dan kunci yang sama.

Misalkan di dalam basis data foto tersebut disimpan sejumlah *cipher-image* C_1, C_2, …, C_n yang merupakan versi terenkripsi dari *plain-image* P_1, P_2, …, P_n. Semua citra dienkripsi dengan algoritma yang sama dan kunci yang sama. Misalkan kriptanalis dapat mencuri atau menerka sejumlah *plain-image* dari *cipher-image* yang berkoresponden. Dengan memilih beberapa *plain-image*,

kriptanalis berharap dapat mendeduksi kunci rahasia (atau kunci yang ekivalen), selanjutnya kunci tersebut digunakan untuk mendekripsi *cipher-image* yang lain. Tinjau eksperimen serangan *selective-plaintext*, masing-masing pada algoritma enkripsi sederhana dengan XOR dan algoritma enkripsi berbasis *chaos* yang diusulkan. Tanpa kehilangan generalisasi, percobaan hanya dilakukan pada citra *grayscale* saja. Dalamksperimen ini digunakan tiga buah citra *grayscale* yaitu citra 'kapal' (P_1), citra 'desa' (P_2), dan citra 'gedung sate' (P_3).

5.1 Serangan pada Algoritma Enkripsi XOR Sederhana

Gambar 4 memperlihatkan serangan dengan *selective plaintext*. Tiga buah *plain-image* P_1, P_2, dan P_3 telah dienkripsi dengan algoritma XOR sederhana menghasilkan tiga *cipher-image* C_1, C_2, dan C_3. Ketiga buah *plain-image* dienkripsi dengan kunci K yang sama (yang tidak diketahui oleh kriptanalis).



(a) P_1 (b) C_1 (c) $C_1 \oplus C_2$

(d) P_2 (e) C_2 (f) $C_2 \oplus P_2$

(g) P_3 (h) C_3 (i) $(C_2 \oplus P_2) \oplus C_3$

Gambar 4. Serangan dengan *selective plaintext* pada algoritma enkripsi XOR sederhana

Gambar 4(c) adalah dua buah *cipher-image* yang ter-XOR satu sama lain ($C_1 \oplus C_2$). Belum ada informasi kunci yang dapat diperoleh dari $C_1 \oplus C_2$. Gambar 4(f) adalah hasil peng-XOR-an C_2 dan P_2 (yaitu $C_2 \oplus P_2$). Jika hasil $C_2 \oplus P_2$ di-XOR-kan dengan C_3 maka C_3 berhasil didekripsi menjadi *plain-image* (P_3), sehingga $C_2 \oplus P_2$ dapat dianggap sebagai kunci rahasia yang ekivalen. Penjelasannya adalah sebagai berikut:

$$\begin{aligned} C_2 \oplus P_2 &= (P_2 \oplus K) \oplus P_2 \\ &= (P_2 \oplus P_2) \oplus K \\ &= 0 \oplus K \\ &= K \end{aligned}$$

Jadi, kunci rahasia (atau yang ekivalen dengan) K berhasil ditemukan. Untuk membuktikan bahwa K adalah kunci yang benar, maka selanjutnya $K = C_2 \oplus P_2$ digunakan untuk mendekripsi salah satu *cipher-image* yaitu C_3:

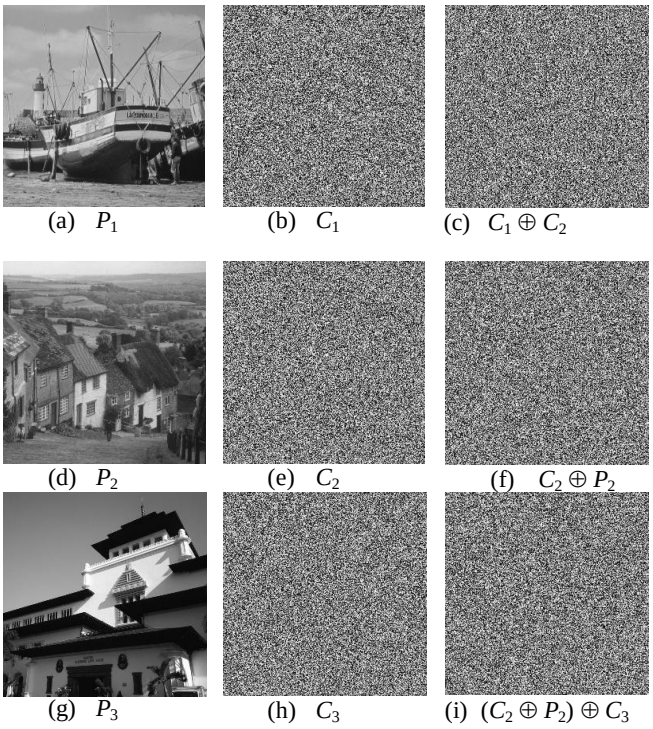
$$\begin{aligned} (C_2 \oplus P_2) \oplus C_3 &= K \oplus C_3 \\ &= K \oplus (P_3 \oplus K) \\ &= (K \oplus K) \oplus \mathrm{P}_3 \\ &= 0 \oplus P_3 \\ &= P_3 \end{aligned}$$

Hasil yang terakhir ini benar sehingga kunci rahasia K yang benar sudah ditemukan. Selanjutnya K dapat digunakan untuk mendekripsi semua *cipher-image* di dalam basis data foto.

Dari eksperimen ini dapat disimpulkan bahwa algoritma enkripsi citra dengan XOR sederhana tidak aman terhadap *selective plaintext attack.*

5.2 Serangan pada Algoritma Enkripsi Berbasis Chaos yang Diusulkan

Sama seperti eksperimen sebelumnya, tiga buah *plain-image* P_1, P_2, dan P_3 telah dienkripsi dengan algoritma berbasis *chaos* yang telah dijelaskan pada Bagian 4. Hasilnya adalah tiga buah *cipher-image* C_1, C_2, dan C_3. Ketiga buah *plain-image* tersebut dienkripsi dengan kunci K yang sama (yang tidak diketahui oleh kriptanalis).



(a) P_1 (b) C_1 (c) $C_1 \oplus C_2$

(d) P_2 (e) C_2 (f) $C_2 \oplus P_2$

(g) P_3 (h) C_3 (i) $(C_2 \oplus P_2) \oplus C_3$

Gambar 5. Serangan dengan *selective plaintext* pada algoritma enkripsi berbasis *chaos* yang diusulkan

Hasil eksperimen yang ditunjukkan pada Gambar 5 memperlihatkan bahwa *selective plaintext attack* tidak berhasil mendeduksi kunci dari $C_2 \oplus P_2$. Karena $C_2 \oplus P_2$ tidak menghasilkan kunci rahasia maka hasil dekripsi C_3 dengan $C_2 \oplus P_2$ tidak berhasil mengembalikan *plain-image*. Penjelasannya adalah sebagai berikut: Ingatlah bahwa sebelum di-XOR-kan dengan kunci *K*, *plain-image* diacak terlebih dahulu dengan *Arnold Cat Map* (*ACM*). Dengan demikian,

$$\begin{aligned} C_2 \oplus P_2 &= (ACM(P_2) \oplus K) \oplus P_2 \\ &= P_2' \oplus K \oplus P_2 \\ &= X \end{aligned}$$

yang dalam hal ini P_2' adalah hasil pengacakan dengan *ACM* dan *X* adalah matriks acak lain yang tidak sama dengan kunci.

Selanjutnya,

$$\begin{aligned} (C_2 \oplus P_2) \oplus C_3 &= X \oplus C_3 \\ &= X \oplus (ACM(P_3) \oplus K) \\ &= X \oplus P_3' \oplus K \\ &= Y \end{aligned}$$

yang dalam hal ini P_3' adalah hasil pengacakan dengan *ACM* dan *Y* adalah citra acak lain yang tidak sama dengan salah satu *plain-image* yang dipilih.

Eksperimen ini menunjukkan bahwa algoritma enkripsi citra berbasis *chaos* yang diusulkan di dalam [5] aman terhadap *selective plaintext attack*. Pengacakan *pixel-pixel* citra sebelum dilakukan operasi XOR dengan kunci telah meningkatkan tingkat keamanan algoritma dari serangan semacam ini yang bertujuan mendeduksi kunci.

6. Kesimpulan

Di dalam makalah ini telah dipresentasikan analisis serangan *selective plaintext* terhadap sebuah algoritma enkripsi citra berbasis *chaos* yang mengkombinasikan teknik permutasi dan teknik substitusi. Sebagai pembandingnya adalah serangan *selective plaintext* pada algoritma enkripsi sederhana dengan *XOR*. Tujuan serangan *selective plaintext* adalah untuk mendeduksi kunci dari sejumlah *plain-image* dan *cipher-image*. Hasil eksperimen menunjukkan bahwa algoritma enkripsi citra berbasis *chaos* tersebut aman terhadap serangan *selective plaintext*, karena pengacakan *pixel-pixel* citra sebelum dilakukan operasi XOR telah meningkatkan tingkat keamanan algoritma dari pendeduksian kunci.

7. *ACKNOWLEDGMENT*

Penelitian yang dipublikasikan di dalam makalah ini sepenuhnya didukung oleh dana **Riset dan Inovasi KK 2012** (Program Riset ITB 2012).

REFERENSI

[1] Fangjun Huang, *Information Security Research Based on Discrete Chaotic Theory*, Huazhong University of Science and Technology, Wuhan, 2005.

[2] Shujun Li, Xuan Zheng, *On the Security of an image encryption method*, in Proceeding 2002 International Conference Image Processing, 2002, vol. 2, pp. 925-928.

[3] Jiasheng Liu, *Study on Chaos-based Image Encryption Technology*, Anhui University, 2007, pp. 58 – 60

[4] Tang Hongmei, Han Liying, He Yu, Wang Xia, *An Improved Compound Image Encryption Scheme*, in Proc. 2010 International Conference and Communication Technologies in Agriculture Engineering, 2010.

[5] Rinaldi Munir, Algoritma Enkripsi *Citra Digital Berbasis Chaos dengan Penggabungan Teknik Permutasi dan Teknik Substitusi Menggunakan Arnold Cat Map dan Logistic Map*, dalam Prosdiding Konferensi Nasional Pendidikan Teknik Informatika (SENAPATI) 2012, Universitas Pendidikan Ganesha (Undiksha), Singaraja, Bali