Modifikasi Model Enkripsi *Encryption With Covertext and Reordering* menggunakan Fungsi Random dan Tabel Permutasi

H. Murti[1], E. Lestariningsih[2] , E. Supriyanto[3] dan E. Ardhianto[4]
[1,3]*Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Stikubank Semarang*
[2,4]*Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Stikubank Semarang*
Jl.Tri Lomba Juang No. 1, Semarang
E-mail : harimurti@edu.unisbank.ac.id[1], endang_lestariningsih@edu.unisbank.ac.id[2],
edysupri4@edu.unisbank.ac.id[3], ekaardhianto@edu.unisbank.ac.id[4]

Abstract*—For limited entities, documents can be confidential and important, so special security mechanisms are needed. Encryption with Covertext and Reordering (ECR) is a text-based document security model. ECR uses a Random key to generate the ciphertext. The ECR Random key is generated manually. This study aims to increase the level of document security by using the ECR encryption model. The Random key permutation table is proposed in the change of the ECR encryption model. A random value will be selected automatically using the random function from the permutation table. In this study as a measure is entropy, which is the value of the security level of encrypted documents. The experimental results show that embedding the permutation table of Random keys gives better entropy values, which implies a better level of security. The use of a Random key permutation table also makes it easier to use ECR to secure documents.

***Abstrak*—Bagi entitas terbatas dokumen dapat bersifat rahasia dan penting, sehingga diperlukan mekanisme keamanan khhusus. *Encryption with Covertext and Reordering* (ECR) adalah salahsatu model keamanan dokumen berbasis teks. ECR menggunakan kunci acak untuk menghasilkan ciphertext. Kunci acak ECR dibuat secara manual. Penelitian ini bertujuan meningkatkan tingkat keamanan dokumen dengan menggunakan model enkripsi ECR. Kunci acak tabel permutasi diusulkan dalam perubahan model enkripsi ECR. Nilai acak akan dipilih secara otomatis menggunakan fungsi acak dari tabel permutasi. Dalam penelitian ini sebagai ukuran adalah entropi, yang merupakan nilai tingkat keamanan dokumen terenkripsi. Hasil eksperimen menunjukkan bahwa penyematan tabel permutasi dari kunci acak memberikan nilai entropi yang lebih baik, yang menyiratkan tingkat keamanan yang lebih baik. Penggunaan tabel permutasi kunci acak juga mempermudah penggunaan ECR untuk mengamankan dokumen.**

***Kata Kunci*— kunci acak, pengacakan, pengamanan informasi, table permutasi.**

I. PENDAHULUAN

Dokumen merupakan tulisan penting yang berisi informasi. Tulisan dalam dokumen secara umum berbentuk teks. Beberapa entitas memandang dokumen merupakan aset yang penting dan rahasia. Sehingga sebuah mekanisme pengamanan dokumen diperlukan untuk menjaga kerahasiaannya. Salah satu ilmu yang bertujuan untuk mengamankan dokumen dikenal sebagai kriptografi [1]. Kriptografi bertujuan mengamankan dokumen dengan cara membuat dokumen tersebut menjadi sulit untuk diartikan [1] [2]. Kriptografi mengacak informasi menggunakan kunci sehingga pihak ketiga tidak dapat mengakses informasi tanpa kunci tersebut [3].

Salah satu model chiper adalah ECR (*Encryption with Covertext and Reordering*). ECR menyajikan pendekatan gabungan steganografi berbasis teks yang bekerja pada teknik enkripsi menggunakan Ex-OR dan proses menyusunan ulang menggunakan *Random key* [4]. Pendekatan penggunaan Ex-Or memberikan keuntungan dalam mempercepat proses enkripsi dekripsi [4]. Parameter penting dalam ECR adalah *Covertext* dan *Random key*. *Covertext* dalam ECR berguna sebagai kunci dalam proses enkripsi dekripsi. *Random key* digunakan untuk menggabungkan *enciphertext* dengan *Covertext*. *Random key* dibuat secara *human generated*. *Random key* berisi empat angka "1" dan empat angka "0". Penempatan angka dalam *Random key* disusun sedemikian rupa sehingga bersifat acak.

Penggunaan *Pseudorandom Number Generation* (PRNG) untuk membangkitkan nilai acak diaplikasikan pada proses pengamanan dokumen [5]. PRNG juga digunakan untuk menanamkan pesan rahasia kedalam *Covertext* [6]. Penggunaan kondisi acak pada permainan sudoku juga menjadi sebuah solusi dalam menggenerate nilai yang acak [7]. Penggunaan angka random menggunakan *Linear Congruential Generator* (LGC) juga diterapkan untuk penyematan karakter kedalam piksel gambar [8]. Angka random juga dihasilkan menggunakan posisi karakter pada fonetik *keyboard* dalam huruf benggali [9].

Dalam ECR, *Random key* tersusun atas angka 1 dan 0. *Random key* sebaiknya adalah yang benar benar acak. Hal ini akan menjadi sulit ketika keacakan *Random key* harus ditentukan oleh manusia. Pada penelitian ini sebuah pendekatan penggunaan tabel permutasi *Random key* dan

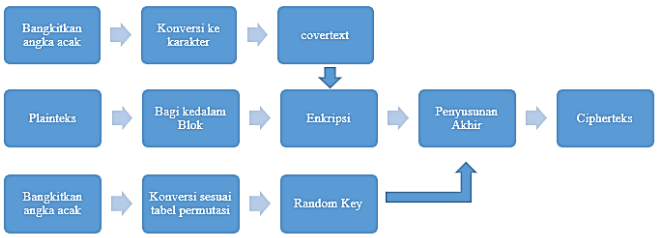
fungsi *random* diusulkan untuk menjadikan sebuah kondisi acak. Sebagai perbandingan hasil digunakan perhitungan entropy.

II. METODE PENELITIAN

A. Usulan Modifikasi Metode ECR

Pada bagian ini akan dijelaskan usulan modifikasi model ECR. Proses ECR diawali dengan membagi *Plaintext* menjadi beberapa blok dengan ukuran 4 karakter setiap blok. Langkah selanjutnya ialah menentukan *Random key* dan *Covertext* untuk setiap blok. Proses enkripsi dilakukan menggunakan operator x-or kepada *Plaintext* dan *Covertext* pada masing masing blok. Proses *reordering* ialah melakukan penyusunan *ciphertext* akhir berdasar nilai *Random key* antara *enciphertext* dan *Covertext*.

Bentuk modifikasi yang diusulkan ialah pada bagian penentuan *Covertext* dan *Random key*. Pada mulanya, pembangkitan dilakukan secara *human generated*. Penelitian ini menggunakan fungsi *random* dan penggunaan tabel permutasi untuk menentukan keduanya. *Covertext* ditentukan dengan diawali penentuan 4 angka acak antara 32 hingga 256. Angka tersebut ialah angka desimal yang menyatakan *printable character*. Proses enkrpsi tetap dilakukan sesuai aturan ECR. *Random key* ditentukan dengan menggunakan fungsi *random* pada angka indeks tabel permutasi. Angka *random* yang didapat dikonversikan menjadi *Random key* sesuai pada tabel. Proses *reordering* dilakukan sesuai dengan proses ECR, hingga menghasilkan *ciphertext* akhir. Gambar 1 menunjukkan bagan ECR yang telah dimodifikasi.



Gambar. 1. Usulan Modifikasi Model Enkripsi ECR.

B. Desain Tabel Permutasi

Random key dalam ECR berperan pada tahap *reordering*. *Random key* digunakan sebagai acuan dalam penyusunan *ciphertext* akhir dari *Covertext* dan *enciphertext*. *Random key* terdiri dari angka "0" dan "1". *Random key* memiliki panjang 8 angka. Untuk menyusun *Random key* diperlukan angka "0" dan "1" masing masing 4 buah yang disusun secara bebas. Dalam penelitian ini, *Random key* disusun pada tabel dengan 2 parameter yaitu: parameter indeks sebagai nomor urut dan parameter *Random key* yang menyatakan *Random key* yang digunakan. Penyusunan *Random key* pada tabel menggunakan persamaan 1.

$$P = \frac{n!}{k_1! \, x \, k_2!} \qquad (1)$$

(1)

Panjang *Random key* dinyatakan sebagai n. Jumlah digit 1 dan digit 2 dinyatakan sebagai k1 dan k2. Menggunakan persamaan 1, maka nilai permutasi (P) yang didapatkan ialah 70 susunan *Random key*. Tabel 1 menunjukkan tabel permutasi *Random key* yang digunakan.

Tabel 1. Tabel Permutasi *Random key*

Indek	Angka Desimal	*Random key*							
1	15	0	0	0	0	1	1	1	1
2	23	0	0	0	1	0	1	1	1
3	27	0	0	0	1	1	0	1	1
4	29	0	0	0	1	1	1	0	1
5	30	0	0	0	1	1	1	1	0
6	39	0	0	1	0	0	1	1	1
7	43	0	0	1	0	1	0	1	1
8	45	0	0	1	0	1	1	0	1
9	46	0	0	1	0	1	1	1	0
10	51	0	0	1	1	0	0	1	1
11	53	0	0	1	1	0	1	0	1
12	54	0	0	1	1	0	1	1	0
13	57	0	0	1	1	1	0	0	1
14	58	0	0	1	1	1	0	1	0
15	60	0	0	1	1	1	1	0	0
16	71	0	1	0	0	0	1	1	1
17	75	0	1	0	0	1	0	1	1
18	77	0	1	0	0	1	1	0	1
19	78	0	1	0	0	1	1	1	0
20	83	0	1	0	1	0	0	1	1
21	85	0	1	0	1	0	1	0	1
22	86	0	1	0	1	0	1	1	0
23	89	0	1	0	1	1	0	0	1
24	90	0	1	0	1	1	0	1	0
25	92	0	1	0	1	1	1	0	0
26	99	0	1	1	0	0	0	1	1
27	101	0	1	1	0	0	1	0	1
28	102	0	1	1	0	0	1	1	0
29	105	0	1	1	0	1	0	0	1
30	106	0	1	1	0	1	0	1	0
31	108	0	1	1	0	1	1	0	0
32	113	0	1	1	1	0	0	0	1
33	114	0	1	1	1	0	0	1	0
34	116	0	1	1	1	0	1	0	0
35	120	0	1	1	1	1	0	0	0
36	135	1	0	0	0	0	1	1	1
37	139	1	0	0	0	1	0	1	1
38	141	1	0	0	0	1	1	0	1
39	142	1	0	0	0	1	1	1	0
40	147	1	0	0	1	0	0	1	1
41	149	1	0	0	1	0	1	0	1
42	150	1	0	0	1	0	1	1	0
43	153	1	0	0	1	1	0	0	1
44	154	1	0	0	1	1	0	1	0
45	156	1	0	0	1	1	1	0	0
46	163	1	0	1	0	0	0	1	1
47	165	1	0	1	0	0	1	0	1
48	166	1	0	1	0	0	1	1	0

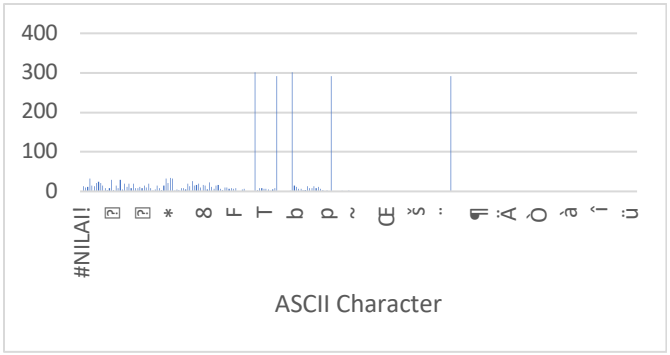
49	169	1	0	1	0	1	0	0	1
50	170	1	0	1	0	1	0	1	0
51	172	1	0	1	0	1	1	0	0
52	177	1	0	1	1	0	0	0	1
53	178	1	0	1	1	0	0	1	0
54	180	1	0	1	1	0	1	0	0
55	184	1	0	1	1	1	0	0	0
56	195	1	1	0	0	0	0	1	1
57	197	1	1	0	0	0	1	0	1
58	198	1	1	0	0	0	1	1	0
59	201	1	1	0	0	1	0	0	1
60	202	1	1	0	0	1	0	1	0
61	204	1	1	0	0	1	1	0	0
62	209	1	1	0	1	0	0	0	1
63	210	1	1	0	1	0	0	1	0
64	212	1	1	0	1	0	1	0	0
65	216	1	1	0	1	1	0	0	0
66	225	1	1	1	0	0	0	0	1
67	226	1	1	1	0	0	0	1	0
68	228	1	1	1	0	0	1	0	0
69	232	1	1	1	0	1	0	0	0
70	240	1	1	1	1	0	0	0	0

III. HASIL DAN PEMBAHASAN

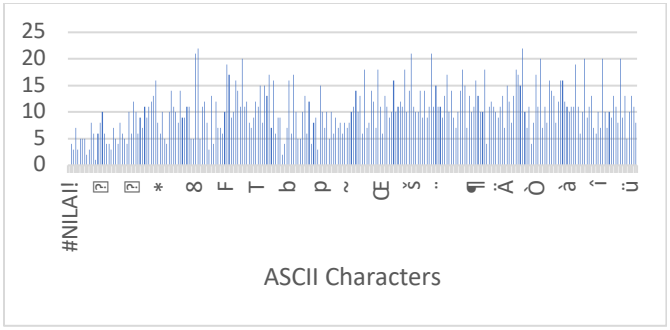
Ekperimen penelitian ini menggunakan data dari *https://haveibeenpwned.com/Passwords*. Data yang digunakan ialah karakter *password* yang kemudian terbagi menjadi 291 blok. Eksperimen yang dilakukan ada dua macam. Eksperimen pertama melakukan pengamanan data menggunakan *Covertext* dan *Random key* yang sama untuk setiap blok. Ekperimen kedua menggunakan fungsi random untuk *Covertext* dan tabel permutasi pada *Random key*. Proses enkripsi dekripsi ECR dilakukan secara bolak balik untuk membuktikan bahwa tidak ada perubahan antara *Plaintext* sebelum di enkripsi dan *Plaintext* setelah didekripsi.

Pada proses enkripsi, *Plaintext* akan diubah menjadi ciphertext. Pada tahap ini dilakukan penghitungan frekuensi karakter yang membentuk *ciphertext*. Pada ekperimen pertama, *Covertext* dan *Random key* di tentukan secara human generated. Gambar 2 memperlihatkan bahwa terdapat 5 karakter dengan jumlah yang menonjol diantara karakter *ciphertext* lainnya. Karakter tersebut merupakan representasi dari 4 karakter *Covertext* dan 1 karakter *Random key*. Hal ini karena penggunaan *Covertext* dan *Random key* yang diimplementasikan secara identik untuk setiap blok. Gambar 2 juga menunjukkan bahwa distribusi karakter pada *ciphertext* tidak merata. Dapat dilihat bahwa terdapat karakter yang tidak digunakan. Hal ini akan menjadi celah bagi kriptanalis untuk membuka dokumen yang dienkripsikan.

Pada eksperimen kedua, penggunaan fungsi random dan tabel permutasi digunakan. *Covertext* dipilih secara random dari printable karakter ASCII. *Random key* dipilih dengan menggunakan fungsi random pada nomor indeks tabel permutasi, yang kemudian dikonveriskan pada angka *Random key* sesuai indek. Hasil yang didapatkan ialah bahwa sebaran karakter pada *ciphertext* lebih merata. Tidak ada penggunan karakter yang terlalu menonjol. Gambar 3 memperlihatkan sebaran karakter pada *ciphertext* dari eksperimen ini. Dengan sebaran karakter yang lebih merata, hal ini akan menjadikan kriptanalis menjadi lebih sulit untuk menebak isi dokumen sebenarnya.



Gambar. 2. Histogram karakter cipherteks hasil ECR.



Gambar. 3. Histogram karakter cipherteks hasil ECR modifikasi

Entropi digunakan untuk mengukur keacakan dari sebuah informasi [10]. Sebuah hasil enkripsi akan menjadi lebih aman jika memiliki nilai entropi yang lebih tinggi. Semakin tinggi nilai entropi dan semakin ideal nilai entropi, mana untuk membobol sistem enkripsi akan semakin sulit [11]. Dalam penelitian ini akan membandingkan dua buah ciphertext yang akan diukur nilai entropinya. *Ciphertext* pertama ialah *ciphertext* dari hasil ECR yang menggunakan *Random key* yang sama untuk semua blok, ciphertext kedua ialah yang mengimplementasikan penggunan fungsi random dan tabel permutasi. perhitungan nilai entropi menggunakan perangkat *cryptool*. Tabel 2 memperlihatkan hasil perhitungan entropi kedua file.

Tabel 2. Entropi hasil enkripsi ECR dan Modifikasi ECR

Nama File	Nilai Entropi	Entropi Maksimum	%
ciphertext1.txt	4.34	6.61	65.66
ciphertext2.txt	6.45	6.61	97.58

Nilai entropi *ciphertext* hasil eksperimen kedua menunjukkan nilai 6.45 yang lebih baik dari *ciphertext*

sebelumnya yaitu 4.34. Terjadi peningkatan lebih dari 30%. Hal ini merupakan efek dari penggunaan bentuk acak pada *Covertext* dan *Random key* yang digunakan. Semakin acak *ciphertext* maka akan semakin baik model *cipher*nya. Demikian pula dapat dikatakan bahwa tingkat keamanan pada *ciphertext* kedua adalah lebih baik

IV. KESIMPULAN

Berdasarkan hasil penelitian, dapat ditarik simpulan bahwa penggunaan bentuk random akan memberikan efek pada peningkatan level keamanan dokumen. ECR modifikasi yang diusulkan akan memberikan tingkat pengamanan yang lebih baik dari versi sebelumnya. Meski nilai entropi sudah menjadi lebih baik, celah celah lain dalam ECR mungkin masih ada. Sehingga bentuk cara lain untuk meningkatkan level keamanan perlu difikirkan secara kontinyu dan menjadi fokus penelitian kedepan.

DAFTAR PUSTAKA

[1] E. Ardhianto, A. Trisetyarso, W. Suparta, B. S. Abbas dan C. H. Kang, "Design Securing Online Payment Transactions Using Stegblock Through Network Layers," dalam *INCITEST 2020*, Bandung, 2020.

[2] E. Ardhianto, H. L. H. S. Warnars, B. Soewito, F. L. Gaol dan E. Abdurachman, "Improvement of Steganography Technique: A Survey," dalam *1st International Multidisciplinary Conference on Education, Technology, and Engineering (IMCETE 2019)*, Serang, Banten, Indonesia, 2020.

[3] V. S. Babu dan H. K. J, "A Study on Combined Cryptography and Steganography," *International Journal of Research Studies in Computer Science and Engineering,* vol. 2, no. 5, pp. 45-49, 2015.

[4] S. Kataria, K. Singh, T. Kumar dan M. S. Nehra, "ECR(Encryption with Cover Text and Reordering) based Text Steganography," dalam *IEEE Second International Conference on Image Information Processing*, Waknaghat, Shimla, Himachal Pradesh, INDIA, 2013.

[5] M. Y. Elmahi, T. M. Wahbi dan M. H. Sayed, "Text Steganography Using Compression and Random Number Generators," *International Journal of Computer Applications Technology and Research,* vol. 6, no. 6, pp. 259-263, 2017.

[6] M. Y. Elmahi dan T. M. Wahbi, "Multi-Level Steganography Aided with Compression," dalam *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, Sudan, 2019.

[7] A. Majumder, S. Changder dan N. C. Debnath, "A New Text Steganography Method Based on Sudoku Puzzle Generation," dalam *International Conference on Emerging Trends in Information Technology (ICETIT 2019)*, New Delhi, India, 2020.

[8] M. Elveny, R. Syah, I. Jaya dan I. Affandi, "Implementation of Linear Congruential Generator (LCG) Algorithm, Most Significant Bit (MSB) and Fibonacci Code in Compression and Security Messages Using Images," dalam *4th International Conference on Computing and Applied Informatics 2019 (ICCAI 2019)*, Medan, Indonesia, 2020.

[9] M. Khairullah, "A novel steganography method using transliteration of Bengali text," *Journal of King Saud University –Computer and Information Sciences,* vol. 31, no. 3, pp. 348-366, 2019.

[10] P. Patil, P. Narayankar, N. D. G dan M. S. M, "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish," dalam *International Conference on Information Security & Privacy (ICISP2015)*, Nagpur, INDIA, 2016.

[11] S. Rajesh, V. Paul, V. G. Menon dan M. . R. Khosravi, "A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices," *Symmetry ,* vol. 11, no. 2, pp. 1-21, 2019.