NARATIF(Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika)
Vol. 03 No. 02 Desember 2021
P-ISSN: 2656-7377 || E-ISSN: 2714-8467

PENGGUNAAN *POLY LETTER KEY WORD RADIX CRYPTOGRAPHY* DALAM PENGAMANAN PESAN

Eza Budi Perkasa[1], Elly Yanuarti[2]
Program Studi Teknik Informatika[1], Program Studi Sistem Informasi[2]
Institut Sains dan Bisnis Atma Luhur[1,2]
ezabudiperkasa@atmaluhur.ac.id[1], elly@atmaluhur.ac.id[2]

Abstrak

Penelitian ini mengenalkan sebuah algoritma kriptografi yang diberi nama *Poly Letter Key Word Radix Cryptography* yang didasarkan dari basis bilangan (*radix*). Algoritma kriptografi ini merupakan pengembangan dari algoritma sebelumnya, yaitu (*Mono Letter Key*) *Word Radix Cryptography*. Dalam makalah ini, *Poly Letter Key Word Radix Cryptography* akan diterapkan dalam pengenkripsian pesan. *Poly Letter Key Word Radix Cryptography* memiliki enam tahapan: Pemindaian kata, enkode, pengubahan basis, penghapusan nol, dekode, dan pengubahan kapitalisasi. Tidak seperti algoritma kriptografi lainnya, panjang dari pesan yang telah diacak dapat tidak sama dan tidak tetap dibandingkan pesan semula.
Kata kunci : Kriptografi, *Poly Letter Key Word Radix Cryptography*, basis bilangan, algoritma

Abstract

This research introduces a cryptography algorithm called Poly Letter Key Word Radix Cryptography which is based on numerical base (radix). The cryptography algorithm is an improvement of the previous algorithm, namely (Mono Letter Key) Word Radix Cryptography. In this paper, Poly Letter Key Word Radix Cryptography will be applied in encrypting a message. Poly Letter Key Word Radix Cryptography has six steps: Word scanning, encode, base converting, zero removal, decode, and capitalization changing. Unlike other cryptography algorithms, length of the enciphered message can be either unequal or inconstant compared to the original message.
Keywords : Cryptography, Poly Letter Key Word Radix Cryptography, numerical base, algorithm

I. PENDAHULUAN

Keamanan dan kerahasiaan data merupakan salah satu dari beberapa aspek penting dari suatu sistem. Dari segi keamanan, umumnya kriptografi merupakan hal utama yang dibahas. Kriptografi merupakan ilmu yang mempelajari cara menjaga data atau pesan tetap aman ketika dikirimkan antarpihak yang berwenang tanpa campur tangan dari pihak lainnya [1]. Tujuan dari kriptografi adalah untuk memberikan layanan keamanan, termasuk keamanan dalam penjagaan kerahasiaan dari pesan. Pesan yang telah dikirim tidak boleh jatuh ke tangan orang yang tidak berhak ataupun tidak berkepentingan [2].

Berdasarkan studi peneliti, teknik-teknik pengamanan pesan dapat dibagi menjadi beberapa kelompok. Kelompok pertama, algoritma kriptografi klasik, merupakan teknik substitusi pesan sederhana dengan mengganti setiap karakter pada pesan dengan karakter lainnya dengan memanfaatkan aritmatika modulo. Jika pesan tersebut merupakan *plain text* (teks bermakna) yang akan diubah menjadi *cipher text* (teks acak yang tak bermakna), maka proses tersebut adalah proses enkripsi. Adapun proses sebaliknya (pengubahan *cipher text* ke *plain text*) dinamakan dekripsi [3]. Algoritma lainnya yang tergolong dalam kelompok algoritma kriptografi modern menggunakan teknik perhitungan yang lebih kompleks. Contoh dari penerapan algoritma kriptografi ini dapat dilihat pada [4]. Baik algoritma kriptografi klasik ataupun modern, keduanya memiliki kelemahan, yaitu panjang *cipher text* biasanya sama dengan *plain text* sehingga dapat diserang menggunakan teknik-teknik dalam kriptanalisis. Oleh karena itu, digunakan teknik lainnya, yaitu fungsi *hash*. Fungsi tersebut merupakan fungsi satu arah yang berarti pesan yang sudah di-*hash* tidak dapat dikembalikan seperti semula. Panjang *hash* ini selalu tetap (tergantung dari algoritma yang digunakan) berapapun panjang pesan semula. Fungsi *hash* ini pun memiliki kekurangan: Terdapat kemungkinan dua pesan yang berbeda yang memiliki *hash* yang sama. Akibatnya, pesan yang sampai di tangan penerima bisa jadi berbeda dengan pesan yang dikirimkan sebelumnya.

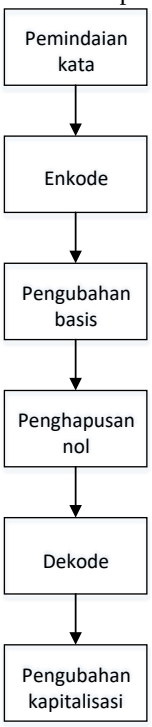
Penelitian ini akan memperkenalkan sebuah algoritma kriptografi yang bernama *Poly Letter Key Word Radix Cryptography* (selanjutnya disingkat WRC-PLK). Algoritma tersebut merupakan pengembangan algoritma sebelumnya yang dinamakan (*Mono Letter Key*) *Word Radix Cryptography* (selanjutnya disingkat WRC(-MLK)) yang dapat dilihat di [5]. Penjelasan lebih lanjut mengenai WRC-PLK serta perbandingannya dengan WRC(-MLK) dan algoritma kriptografi lain yang merupakan analoginya dibahas pada Bab II. Adapun contoh penerapan algoritma tersebut dapat dilihat pada Bab III.

II. TENTANG *POLY LETTER KEY WORD RADIX CRYPTOGRAPHY*

WRC-PLK adalah algoritma kriptografi yang didasarkan dari basis (*radix*) bilangan, seperti halnya WRC(-MLK). Perbedaannya, WRC-PLK menggunakan sistem bilangan basis campuran (*mixed radix*). Dalam sistem bilangan tersebut, sebuah bilangan desimal dapat diubah ke basis lainnya dengan membagi secara berulang-ulang bilangan tersebut dengan nilai basisnya, dimulai dari nilai terkanan hingga pembagian tersebut menghasilkan 0. Angka-angka dari bilangan hasil pengonversian merupakan sisa dari pembagian. Bilangan hasil pengonversian tersebut dibaca dari sisa pembagian terakhir

ke sisa pembagian pertama. Sebagai contoh, untuk mengonversi 2021 ke basis (3, 4, 5) dapat dilakukan dengan membagi 2021 dengan 5, 4, dan 3 secara berulang-ulang:
2021 / 5 = 404 sisa 1
404 / 4 = 101 sisa 0
101 / 3 = 33 sisa 2
33 / 5 = 6 sisa 3
6 / 4 = 1 sisa 2
1 / 3 = 0 sisa 1
Dari pembagian berulang tersebut, dapat dilihat bahwa 2021 sama dengan 123201 dalam basis (3, 4, 5).

Seperti halnya WRC(-MLK), WRC-PLK juga mengguakan tahap pengonversian bilangan dari suatu basis ke basis lainnya dengan sistem numerasi bijektif. Uraian lebih lanjut mengenai sistem numerasi tersebut dapat dilihat di [6]. Alur dari WRC-PLK secara keseluruhan sama dengan WRC(-MLK) dengan sedikit perubahan proses komputasi mengingat *key* yang digunakan terdiri dari beberapa huruf. Alur tersebut dapat diilustrasikan seperti terlihat pada Gambar 1.



Gambar 1. Alur *Word Radix Cryptography* [5]

1. Pemindaian Kata

Definisi "kata" dalam WRC-PLK sama dengan WRC(-MLK), yaitu sederetan karakter yang diakhiri dengan karakter bukan huruf. Pemindaian kata dilakukan dengan membaca karakter demi karakter hingga akhir kalimat. Jika ditemukan karakter selain huruf, maka kata yang telah dipindai sebelumnya dimasukkan ke dalam variabel. Karakter yang dibaca dan dikonkatenasi bersifat *case insensitive* sehingga huruf kapital maupun nonkapital dianggap sama.

2. Enkode

Tahap enkode adalah tahap untuk mengubah pesan menjadi nilai numerik. Setiap kata memiliki nilai numerik yang unik. Nilai numerik kata untuk *plain text* adalah n_p yang dapat dirumuskan:

$$n_p = \sum_{i=0}^{L-1} n_C(C_{L-i}) \times 26^i \qquad (1)$$

dengan n_C merupakan nomor urut atau nilai numerik karakter dalam abjad (A = 1, B = 2, C = 3, …, Z = 26), L merupakan panjang atau jumlah karakter suatu kata, dan C_{L-i} merupakan karakter ke-(L-i) dalam kata. Di lain pihak, nilai numerik kata untuk *cipher text*, n_c, memiliki rumus yang hampir sama dengan nilai numerik kata untuk *plain text*. Perbedaannya terletak pada konstanta 26 yang diganti dengan hasil kali dari n_K yang merupakan nilai numerik dari setiap karakter pada *key*.

$$n_c = \sum_{i=0}^{L-1} n_C(C_{L-i}) \times \prod_{j=1}^{i} n_K(K_j) \qquad (2)$$

3. Pengubahan Basis

Seperti WRC(-MLK), pengubahan basis dalam WRC-PLK dilakukan dengan pembagian berulang nilai numerik kata dengan n_K (dimulai dari karakter *key* terkanan) untuk proses enkripsi atau 26 untuk proses dekripsi. Dalam

pemrograman, setiap "digit" dalam hasil pengubahan basis dimasukkan ke sebuah variabel *array* yang urutan elemennya dibalik.

4. Penghapusan Nol

Seperti diuraikan pada [6], sistem numerasi bijektif tidak mengenal angka 0. Untuk menghapus nol dalam bilangan hasil pengubahan basis, angka 0 pada daftar (*array*) digit diganti dengan n_K karakter *key* yang sesuai pada posisi angka 0 tersebut pada proses enkripsi atau 26 pada proses dekripsi dan angka sebelumnya dikurangi 1. Jika angka pertama pada daftar digit sama dengan 0, maka angka tersebut dapat dihapus dari daftar.

5. Dekode

Tahap dekode dapat dilakukan dengan syarat daftar digit hasil dari penghapusan nol tidak memiliki elemen tunggal yang nilainya 0. Dalam tahap ini, setiap angka pada daftar digit diganti menjadi huruf yang nomor urutnya adalah angka bersangkutan (1 = A, 2 = B, 3 = C, …, 26 = Z).

6. Pengubahan Kapitalisasi

Tahap pengubahan kapitalisasi ini merupakan tahap opsional yang dapat dilakukan jika kata yang telah dienkripsi/dekripsi ingin "diseimbangkan" kapitalisasinya dengan kata semula. Untuk melakukan tahap ini, pertama-tama kata hasil dipecah menjadi blok-blok yang masingmasing berisi substring yang kapitalisasinya berselang-seling. Kapitalisasi hanya diubah pada blok substring kapital saja atau nonkapital saja. Pada proses enkripsi, nomor karakter *cipher text* yang diubah kapitalisasinya adalah I_{sc} sampai I_{fc}. Kedua nilai tersebut dapat dirumuskan secara matematis sebagai berikut.

$$I_{sc} = \left\lfloor \frac{I_{sp}-1}{L_p} \times L_c \right\rfloor + 1 \qquad (3)$$

$$I_{fc} = \left\lfloor \frac{I_{fp}}{L_p} \times L_c \right\rfloor \qquad (4)$$

Sedangkan untuk dekripsi, rumus yang digunakan hampir sama. Perbedaannya, fungsi *floor* diganti dengan fungsi *ceiling* dan indeks *c* diganti dengan *p* atau sebaliknya.

$$I_{sp} = \left\lceil \frac{I_{sc}-1}{L_c} \times L_p \right\rceil + 1 \qquad (5)$$

$$I_{fp} = \left\lceil \frac{I_{fc}}{L_c} \times L_p \right\rceil \qquad (6)$$

Pada Persamaan (3) hingga (6), I_{sp}, I_{fp}, L_p, dan L_c berturut-turut adalah nomor karakter awal *plain text*, nomor karakter akhir *plain text*, panjang kata *plain text*, dan panjang kata *cipher text*. Persamaan (3) dan (5) adalah koreksi dari rumus penentuan nilai I_{sc} dan I_{sp} sebelumnya pada [6] untuk menghindari kapitalisasi yang masih "tidak seimbang" ketika diterapkan pada kata yang telah dienkripsi/dekripsi.

III. CONTOH PENERAPAN

Bab ini akan mendemonstrasikan contoh penerapan WRC-PLK dalam pengenkripsian sebuah pesan. Pesan yang dienkripsi adalah "SemInar Nasional peneliTian dan pengAbdian MASyarakat" dengan *key* EZA (n_K = (5, 26, 1)).

TABEL I
PROSES ENKRIPSI PESAN

Plain text **SemInar Nasional peneliTian dan pengAbdian MASyarakat**					
Pemindaian Kata	**Enkode**	**Pengubahan Basis**	**Penghapusan Nol**	**Dekode**	**Pengubahan Kapitalisasi**
SemInar	5934915004	20 0 3 23 0 1 22 0 2 0 0 1 18 0	19 1 3 22 1 1 21 1 1 25 1 1 17 1	SACVAAUAA YAAQA	SAcvaaUAayaaqa
(spasi)	0	0	0	(spasi)	(spasi)
Nasional	112984390310	3 0 0 5 0 2 24 0 3 8 0 4 10 0 3 12 0	1 1 5 4 1 2 23 1 3 7 1 4 9 1 3 11 1	AAEDABWAC GADIACKA	AAedabwacgadiacka
(spasi)	0	0	0	(spasi)	(spasi)
peneliTian	88030331154780	18 0 1 4 0 4 14 0 3 7 0 3 6 0 3 0 0 1 14 0	17 1 1 3 1 4 13 1 3 6 1 3 5 1 2 25 1 1 13 1	QAACADMA CFACEABYA AMA	qaacadmacfacEAbyaama
(spasi)	0	0	0	(spasi)	(spasi)
dan	2744	21 0 0 14 0	19 1 5 13 1	SAEMA	saema
(spasi)	0	0	0	(spasi)	(spasi)

pengAbdian	88030814811148	18 0 1 4 0 4 16 0 1 19 0 3 25 0 1 15 0 4 14 0	17 1 1 3 1 4 15 1 1 18 1 3 24 1 1 14 1 4 13 1	QAACADOA ARACXAANA DMA	qaacadoaARacxaanadma
(spasi)	0	0	0	(spasi)	(spasi)
MASyarakat	70952722311010	14 0 3 12 0 4 21 0 1 0 0 2 3 0 2 21 0 0 20 0	13 1 3 11 1 4 19 1 5 25 1 2 2 1 2 19 1 5 19 1	MACKADSAE YABBABSAESA	MACKADsaeyabbabsaesa
Cipher text **SAcvaaUAayaaqa AAedabwacgadiacka qaacadmacfacEAbyaama saema qaacadoaARacxaanadma MACKADsaeyabbabsaesa**					

Berikut diberikan contoh tahap pengenkripsian untuk salah satu kata pada pesan tersebut. Kata yang dimaksud adalah "SemInar"

1. Enkode
   Panjang kata "SemInar" adalah 7, sehingga nilai numerik dari kata tersebut adalah

$$n_p = \sum_{i=0}^{6} n_C(C_{7-i}) \times 26^i$$

TABEL II
PROSES ENKODE

Karakter	Nilai Karakter	Bobot	Nilai × Bobot
S	19	26^6	5869399744
e	5	26^5	59406880
m	13	26^4	5940688
I	9	26^3	158184
n	14	26^2	9464
a	1	26^1	26
r	18	26^0	18
Nilai numerik kata			**5934915004**

2. Pengubahan basis
   Basis yang berkesesuaian dengan *key* EZA adalah (5, 26, 1). Pembagian berulang untuk nilai kata yang didapat dari Tabel II adalah

```
5934915004 /   1  = 5934915004 sisa  0
5934915004 /  26  =  228265961 sisa 18
 228265961 /   5  =   45653192 sisa  1
  45653192 /   1  =   45653192 sisa  0
  45653192 /  26  =    1755892 sisa  0
   1755892 /   5  =     351178 sisa  2
    351178 /   1  =     351178 sisa  0
    351178 /  26  =      13506 sisa 22
     13506 /   5  =       2701 sisa  1
      2701 /   1  =       2701 sisa  0
      2701 /  26  =        103 sisa 23
       103 /   5  =         20 sisa  3
        20 /   1  =         20 sisa  0
        20 /  26  =          0 sisa 20
```

   Dari pembagian berulang tersebut, diperoleh nilai kata yang telah diubah basisnya, yaitu 20 0 3 23 0 1 22 0 2 0 0 1 18 0.
3. Penghapusan nol
   Nilai kata yang telah diubah basisnya memiliki elemen dengan nilai nol. Oleh karena itu, nilai nol tersebut dapat dihapus sebagai berikut.

~~20~~ ~~0~~ 3 ~~23~~ ~~0~~ 1 ~~22~~ ~~0~~ ~~2~~ ~~0~~ ~~0~~ 1 ~~18~~ ~~0~~

19 1 22 1 21 1 1 ~~26~~ 25 1 17 1

Berdasarkan proses tersebut, diperoleh nilai tanpa elemen nol, yaitu 19 1 3 22 1 1 21 1 1 25 1 1 17 1

4. Dekode
Dari penghapusan nol sebelumnya, setiap elemen nilai dapat diubah menjadi huruf yang berkesesuaian seperti terlihat pada Tabel III.

TABEL III
PROSES DEKODE

Nilai	19	1	3	22	1	1	21	1	1	25	1	1	17	1
Huruf	S	A	C	V	A	A	U	A	A	Y	A	A	Q	A

Berdasarkan huruf yang didapat pada Tabel III, didapat *cipher text* "SACVAAUAAYAAQA" dengan panjang 14. *Cipher text* tersebut dapat diubah kapitalisasinya agar "seimbang" dengan *plain text* semula seperti diuraikan di tahap selanjutnya.

5. Pengubahan kapitalisasi
Plain text semula dapat dikelompokkan menjadi beberapa blok seperti terlihat pada Tabel IV.

TABEL IV
BLOK *PLAIN TEXT*

Nomor	1	2	3	4	5	6	7
Huruf	S	e	m	I	n	a	r
Blok	I	II		III	IV		

Berdasarkan pembagian blok *plain text* tersebut, dapat ditentukan batasan-batasan blok *cipher text* sebagai berikut.

a. Untuk blok I, $I_{sp} = 1$ dan $I_{fp} = 1$ sehingga $I_{sc} = \left\lfloor \frac{1-1}{7} \times 14 \right\rfloor + 1 = \lfloor 0 \rfloor + 1 = 0 + 1 = 1$ dan $I_{fc} = \left\lfloor \frac{1}{7} \times 14 \right\rfloor = \lfloor 2 \rfloor = 2$
b. Untuk blok II, $I_{sp} = 2$ dan $I_{fp} = 3$ sehingga $I_{sc} = \left\lfloor \frac{2-1}{7} \times 14 \right\rfloor + 1 = \lfloor 2 \rfloor + 1 = 2 + 1 = 3$ dan $I_{fc} = \left\lfloor \frac{3}{7} \times 14 \right\rfloor = \lfloor 6 \rfloor = 6$
c. Untuk blok III, $I_{sp} = 4$ dan $I_{fp} = 4$ sehingga $I_{sc} = \left\lfloor \frac{4-1}{7} \times 14 \right\rfloor + 1 = \lfloor 6 \rfloor + 1 = 6 + 1 = 7$ dan $I_{fc} = \left\lfloor \frac{4}{7} \times 14 \right\rfloor = \lfloor 8 \rfloor = 8$
d. Untuk blok IV, $I_{sp} = 5$ dan $I_{fp} = 7$ sehingga $I_{sc} = \left\lfloor \frac{5-1}{7} \times 14 \right\rfloor + 1 = \lfloor 8 \rfloor + 1 = 8 + 1 = 9$ dan $I_{fc} = \left\lfloor \frac{7}{7} \times 14 \right\rfloor = \lfloor 14 \rfloor = 14$

TABEL V
BLOK *CIPHER TEXT*

Nomor	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Blok	I		II				III		IV					
Huruf	S	A	c	v	a	a	U	A	a	y	a	a	q	a

Dari pembagian blok pada Tabel V, diperoleh *cipher text* yang kapitalisasinya sudah "seimbang" dengan *plain text* semula, yaitu "SAcvaaUAayaaqa".

IV. KESIMPULAN

Makalah ini memperkenalkan sebuah algoritma kriptografi yang diberi nama *Poly Letter Key Word Radix Cryptography*. Pada penelitian ini, algoritma tersebut diterapkan dalam pengenkripsian pesan. Algoritma kriptografi tersebut merupakan pengembangan dari algoritma *(Mono Letter Key) Word Radix Cryptography*. Penelitian selanjutnya dapat menggunakan pendekatan yang berbeda, misalnya dengan menggunakan nilai ASCII untuk setiap karakter (termasuk karakter khusus).

REFERENSI

[1] E. Handayani, W. L. Pratitis, A. Nur, S. A. Mashuri, dan B. Nugroho, "Perancangan Aplikasi Kriptografi Berbasis Web Dengan Algoritma Double Caesar Cipher Menggunakan Tabel ASCII," dalam *Seminar Nasional Teknologi Informasi dan Multimedia 2017*, 2017, hal. 241-246.
[2] C. A. Sari, E. H. Rachmawanto, D. W. Utomo, dan R. R. Sani, "Penyembunyian Data Untuk Seluruh Ekstensi File Menggunakan Kriptografi *Vernam Cipher* dan *Bit Shiffting*," *Journal of Applied Intelligent System*, vol. 1, hal. 179-190, Okt. 2016.
[3] R. K. Hondro. "Aplikasi Enkripsi dan Dekripsi SMS Dengan Algoritma Zig Zag Cipher Pada Mobile Phone Berbasis Android," *Pelita Informatika Budi Darma*, vol. X, Jul. 2015.

[4] K. Aryasa dan Y. T. Paulus, "Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java," *Citec Journal*, vol. 1, hal. 57-66, Nov. 2013.
[5] E. B. Perkasa, R. Sulaiman, D. Y. Sylfania, dan F. P. Juniawan, "*Word Radix Cryptography*: Pengenalan dan Contoh Penggunaannya," dalam *Seminar Nasional Telekomunikasi dan Informatika*, 2018, hal. 45-49.
[6] H. H. Hansen, C. Kupke, J. Rutten, dan J. Winter, "A Final Coalgebra for *k*-regular Sequences," dalam *Horizons of the Mind. A Tribute to Prakash Panangaden*, 2014, hal. 363-383.