Enkripsi Teks dengan Algoritma Affine Cipher

Text Encryption using Affine Cipher Algorithm

Andysah Putera Utama Siahaan

Fakultas Ilmu Komputer, Universitas Pembangunan Panca Budi, Medan, Indonesia

e-mail: andiesiahaan@gmail.com

Abstrak

Affine Cipher adalah algoritma kriptografi yang dikembangkan dari metode Caesar Cipher. Algoritma ini merupakan contoh dari kriptografi pertukaran monoalfabet. Affine Cipher melakukan proses enkripsi dengan cara pergeseran karakter dengan substansial matematis. Perbedaan yang mendasar dari algoritma ini adalah pergeseran dilakukan dengan cara melakukan perkalian terhadap suatu bilangan yang relatif prima dengan bilangan yang digunakan pada saat proses dekripsi. Seluruh proses bergantung pada kunci dan modulo yang bekerja. Kunci yang digunakan pada algoritma ini adalah dua buah bilangan prima dan satu buah bilangan integer sebagai penggeser. Hasil yang diperoleh adalah penggunaan algoritma Affine Cipher dalam melakukan proses enkripsi dan dekripsi. Penggunaan metode ini sangat membantu mengamankan teks yang akan dikirimkan kepada orang lain atau pada jaringan komputer.

Kata kunci: Kriptografi, Affine Cipher, Enkripsi, Dekripsi

Abstract

Affine Cipher is a cryptographic algorithm developed from Caesar Cipher method. This algorithm is an example of a monoalphabetic exchange cryptography. Affine Cipher performs the encryption process by shifting characters substantially mathematically. The fundamental difference of this algorithm is the shift is done by multiplication of a number that is relatively prime to decryption numbers. The whole process depends on the key and the working module. The keys used in this algorithm are two primes and one integer as a slider. This research will explain the use of Affine Cipher algorithm in the process of encryption and decryption. Use of this method is helpful in securing text that will be sent to others or on computer networks.

Keywords: Cryptography, Affice Cipher, Encryption, Decryption

1. PENDAHULUAN

Kriptografi adalah ilmu yang menerapkan matematika dan logika kompleks untuk merancang proses enkripsi dan dekripsinya [6][7]. Setiap saat teknik yang digunakan untuk memecahkan ciphertext juga berkembang dengan pesat. Dengan ini, adalah sangat dibutuhkan teknik-teknik yang mendukung kekuatan untuk memperkuat keamanan dari ciphertext. Untuk melakukan hal ini, teknik yang digunakan membutuhkan penerapan metode baru atau metode yang lebih kreatif. Jadi kriptografi juga merupakan suatu seni dalam melindungi keamanan informasi. Proses enkripsi melakukan pertukaran karakter dengan karakter yang lain. Tetapi, agar pertukaran ini tidak dapat diketahui oleh orang lain, ini membutuhkan kunci untuk menentukan karakter mana yang akan menjadi ciphertext tersebut.

Affine Cipher adalah salah satu teknik kriptografi yang melakukan pertukaran karakter secara monoalfabet dimana setiap satu karakter pada plaintext akan ditukarkan dengan karakter sesuai dengan rumus yang digunakan [3]. Algoritma ini memiliki kunci yang berupa bilangan prima sebagai penentu letak karakter yang digunakan pada tabel ASCII [4][5]. Penggunaan tabel secara manual atau yang diciptakan sendiri juga dapat memperkuat keamanan pada algoritma ini. Algoritma ini memiliki tiga bagian utama yaitu, pembangkitan kunci, proses enkripsi dan proses

dekripsi. Pada kajian ini akan dicoba bagaimana cara menerapkan algoritma ini ke dalam proses enkripsi dan dekripsi pada teks. Penggunaan algoritma ini diharapkan dapat meningkatkan keamanan dari kriptografi substitusi seperti Caesar dan Vigenère Cipher.

2. METODE PENELITIAN

Algoritma Affine Cipher memiliki kunci dalam proses enkripsi dan dekripsi. Kunci tersebut terdiri dari 2 buah bilangan prima yaitu x dan y . Penjelasan berikut mengasumsikan penggunaan karakter ASCII sebanyak 256 karakter ($m = 256$). x harus dipilih untuk menjadi relatif prima terhadap m . Misalnya, 9 dan 57 tidak memiliki faktor yang sama, jadi 9 adalah nilai digunakan sebagai x . Angka 57 tidak memiliki faktor yang sama juga sehingga 57 dapat digunakan untuk nilai b . Angka 9 merupakan nilai yang digunakan pada proses enkripsi sedangkan 57 adalah nilai yang dipakai untuk proses dekripsi. Pergeseran dapat dilakukan sebanyak $s > 0$ [1][2]. Pergeseran adalah angka yang digunakan untuk memperkuat keamanan pada Affine Cipher. Adapun rumus enkripsi yang digunakan adalah sebagai berikut:

$$c = (x * p) + s \bmod m \quad (1)$$

dimana:

c = karakter cipher
 x = bilangan prima enkripsi
 p = karakter plain
 s = pergeseran
 m = jumlah tabel karakter

Rumus dekripsi adalah sebagai berikut

$$p = y * (c - s) \bmod m \quad (2)$$

dimana:

p = karakter plain
 y = bilangan prima dekripsi
 c = karakter cipher
 s = pergeseran
 m = jumlah tabel karakter

Metode ini mempunyai tiga bagian penting dalam prosesnya. Berikut ini adalah potongan kode program yang ditulis dengan menggunakan Microsoft Visual C# yang digunakan untuk tiap-tiap bagian.

```
public static int GCD(int m, int n)
{
    int r = n % m;
    while (r != 0)
    {
        r = m % n;
        m = n;
        n = r;
    }
    return m;
}
```

Kode program sebelumnya adalah fungsi untuk mencari dua buah bilangan prima yang relatif prima. GCD adalah Great Common Divisor yang dalam bahasa Indonesia adalah Faktor Persekutuan Terbesar. Fungsi ini untuk menghasilkan nilai pembagi terbesar dari dua bilangan bulat. Pembagi terbesar adalah bilangan bulat paling besar yang membagi bilangan pertama dan bilangan kedua tanpa ada sisa.

```
public void GetKey()
{
    int x = R.Next(1, 100);
    int y = 0;
    int MOD = 256;

    do
    {
        x++;
    } while (GCD(x, MOD) != 1);

    do
    {
        y++;
    } while ((x * y) % MOD != 1);
}
```

Fungsi *GetKey()* bertujuan untuk menentukan pasangan kunci x dan y. Nilai x pertama sekali diinisialisasi dengan nilai acak dengan batasan 1 hingga 100. Batasan ini dapat diatur sesuai dengan besar bilangan prima yang diinginkan. Semakin besar bilangan prima, semakin kuat tingkat keamanannya. Nilai y diinisialisasi pertama sekali dengan nol, kemudian akan dicari dengan cara menambahkan nilai tersebut dengan 1 dan akan diuji apakah nilai tersebut setelah dikalikan dengan x akan mempunyai nilai relatif prima.

```
public void Encrypt()
{
    Log = "";
    PT = txtPT.Text;
    CT = "";

    int Key = Convert.ToInt16(txtKey1.Text);
    int Shift = Convert.ToInt16(txtShift.Text);

    for (int i = 0; i < PT.Length; i++)
    {
        int PC = Convert.ToInt16(PT[i]);
        int CC = ((Key * PC) + Shift) % MOD;

        CT += Convert.ToChar(CC);
    }
    txtCT.Text = CT;
}
```

Fungsi *Encrypt()* berfungsi untuk melakukan proses enkripsi. Nilai x dan s akan dimasukkan ke dalam variabel Key dan Shift. Untuk setiap panjang string, ini akan dilakukan proses enkripsi berdasarkan persamaan 1. Shift merupakan nilai pergeseran s pada persamaan sebelumnya. Hasil tiap karakter ciphertext akan dimasukkan ke variabel penampung CT yang

bertipe string yang pertama sekali diinisialisasi dengan CT=""'. Setelah proses pengulangan selesai, CT merupakan ciphertext.

```
public void Decrypt()
{
    Log = "";
    CT = txtCT.Text;
    PT = "";

    int Key = Convert.ToInt16(txtKey2.Text);
    int Shift = Convert.ToInt16(txtShift.Text);

    for (int i = 0; i < CT.Length; i++)
    {
        int CC = Convert.ToInt16(CT[i]);
        int PC = (Key1 * (CC - Key2)) % MOD;

        PT += Convert.ToChar(PC);
    }
    txtDT.Text = PT;
}
```

Fungsi *Decrypt()* berfungsi untuk melakukan proses dekripsi. Nilai y dan s akan dimasukkan ke dalam variabel Key dan Shift. Untuk setiap panjang string, ini akan dilakukan proses dekripsi berdasarkan persamaan 2. Hasil tiap karakter hasil dekripsi akan dimasukkan ke variabel penampung PT yang bertipe string yang pertama sekali diinisialisasi dengan PT=""'. Setelah proses pengulangan selesai, PT merupakan hasil dekripsi.

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan ditampilkan hasil dari pengujian untuk sebuah string. Plaintext yang digunakan adalah "KEYSHA SIAHAAN". Perhitungan berikut akan menjelaskan proses enkripsi dari kalimat tersebut.

Tabel 1 Daftar kode ASCII yang digunakan pada plaintext

K	E	Y	S	H	A		S	I	A	H	A	A	N
75	69	89	83	72	65	32	83	73	65	72	65	65	78

Sebelum dilakukan proses enkripsi, ada beberapa parameter yang digunakan sebagai penentu proses enkripsi ini. Nilai y tidak diikutsertakan pada proses ini. Berikut adalah parameter dan perhitungan yang dilakukan untuk mendapatkan ciphertext.

Variabel yang digunakan:

x = 9
s = 8
m = 256

Perhitungan:

$$\begin{aligned} p_1 &= 75 \\ c_1 &= (x * p) + s \bmod m \\ &= (9 * 75) + 8 \bmod 256 \\ &= 683 \bmod 256 \\ &= 171 \end{aligned}$$

$$\begin{aligned} p_2 &= 69 \\ c_2 &= (x * p) + s \bmod m \\ &= (9 * 69) + 8 \bmod 256 \\ &= 629 \bmod 256 \\ &= 117 \end{aligned}$$

$$\begin{aligned} p_3 &= 89 \\ c_3 &= (x * p) + s \bmod m \\ &= (9 * 89) + 8 \bmod 256 \\ &= 809 \bmod 256 \\ &= 41 \end{aligned}$$

$$\begin{aligned} p_4 &= 83 \\ c_4 &= (x * p) + s \bmod m \\ &= (9 * 83) + 8 \bmod 256 \\ &= 755 \bmod 256 \\ &= 243 \end{aligned}$$

$$\begin{aligned} p_5 &= 72 \\ c_5 &= (x * p) + s \bmod m \\ &= (9 * 72) + 8 \bmod 256 \\ &= 656 \bmod 256 \\ &= 144 \end{aligned}$$

$$\begin{aligned} p_6 &= 65 \\ c_6 &= (x * p) + s \bmod m \\ &= (9 * 65) + 8 \bmod 256 \\ &= 593 \bmod 256 \\ &= 81 \end{aligned}$$

$$\begin{aligned} p_7 &= 32 \\ c_7 &= (x * p) + s \bmod m \\ &= (9 * 32) + 8 \bmod 256 \\ &= 296 \bmod 256 \\ &= 40 \end{aligned}$$

$$\begin{aligned} p_8 &= 83 \\ c_8 &= (x * p) + s \bmod m \\ &= (9 * 83) + 8 \bmod 256 \\ &= 755 \bmod 256 \\ &= 243 \end{aligned}$$

$$\begin{aligned} p_9 &= 73 \\ c_9 &= (x * p) + s \bmod m \\ &= (9 * 73) + 8 \bmod 256 \\ &= 665 \bmod 256 \\ &= 153 \end{aligned}$$

$$\begin{aligned} p_{10} &= 65 \\ c_{10} &= (x * p) + s \bmod m \\ &= (9 * 65) + 8 \bmod 256 \\ &= 593 \bmod 256 \\ &= 81 \end{aligned}$$

$$\begin{aligned} p_{11} &= 72 \\ c_{11} &= (x * p) + s \bmod m \\ &= (9 * 72) + 8 \bmod 256 \\ &= 656 \bmod 256 \\ &= 144 \end{aligned}$$

$$\begin{aligned} p_{12} &= 65 \\ c_{12} &= (x * p) + s \bmod m \\ &= (9 * 65) + 8 \bmod 256 \\ &= 593 \bmod 256 \\ &= 81 \end{aligned}$$

$$\begin{aligned} p_{13} &= 65 \\ c_{13} &= (x * p) + s \bmod m \\ &= (9 * 65) + 8 \bmod 256 \\ &= 593 \bmod 256 \\ &= 81 \end{aligned}$$

$$\begin{aligned} p_{14} &= 78 \\ c_{14} &= (x * p) + s \bmod m \\ &= (9 * 78) + 8 \bmod 256 \\ &= 710 \bmod 256 \\ &= 198 \end{aligned}$$

Tabel 2 Hasil enkripsi

	K	E	Y	S	H	A		S	I	A	H	A	A	N
PT	75	69	89	83	72	65	32	83	73	65	72	65	65	78
CT	171	117	41	243	144	81	40	243	153	81	144	81	81	198
	«	u	)	ó	?	Q	(	ó	™	Q	?	Q	Q	Æ

Tabel 2 adalah hasil enkripsi dari plaintext sebelumnya. Ciphertext dapat dilihat pada baris terakhir dalam tabel. Karakter-karakter ini mempunyai tampilan seperti simbol-simbol karena ASCII code pada angka tersebut tidak lagi sesuai dengan deretan alfabet yang ada. Karakter yang berulang akan mempunyai ciphertext yang sama. Seperti pada contoh di atas, ada empat buah karakter “A” sehingga ada nilai 81 sebanyak empat kali pada hasil enkripsi.

Setelah memperoleh hasil enkripsi di perhitungan sebelumnya, ini akan dicoba kembali untuk melakukan proses dekripsi. Jika algoritma dan persamaan yang digunakan sudah benar, maka hasil dekripsi akan kembali seperti plaintext awal. Tabel 3 adalah ciphertext dan kode ASCII yang di gunakan pada proses dekripsi.

Tabel 3 Daftar kode ASCII yang digunakan pada ciphertext

«	u	)	ó	•	Q	(	ó	™	Q	•	Q	Q	Æ
171	117	41	243	144	81	40	243	153	81	144	81	81	198

Pada proses dekripsi, ketentuan juga sama seperti pada proses enkripsi. Ada beberapa parameter yang digunakan untuk melaksanakan proses ini. Tetapi pada dekripsi, nilai x tidak digunakan lagi, yang dipakai adalah nilai y. Berikut adalah parameter dan perhitungan yang dilakukan untuk mendapatkan plaintext.

Variabel yang digunakan:

$$\begin{aligned}y &= 57 \\s &= 8 \\m &= 256\end{aligned}$$

Perhitungan:

$$\begin{aligned}c_1 &= 171 & c_2 &= 117 \\p_1 &= y * (c - s) \bmod m & p_2 &= y * (c - s) \bmod m \\&= 57 * (171 - 8) \bmod 256 & &= 57 * (117 - 8) \bmod 256 \\&= 9291 \bmod 256 & &= 6213 \bmod 256 \\&= 75 & &= 69 \\c_3 &= 41 & c_4 &= 243 \\p_3 &= y * (c - s) \bmod m & p_4 &= y * (c - s) \bmod m \\&= 57 * (41 - 8) \bmod 256 & &= 57 * (243 - 8) \bmod 256 \\&= 181 \bmod 256 & &= 13395 \bmod 256 \\&= 89 & &= 83 \\c_5 &= 144 & c_6 &= 81 \\p_5 &= y * (c - s) \bmod m & p_6 &= y * (c - s) \bmod m \\&= 57 * (144 - 8) \bmod 256 & &= 57 * (81 - 8) \bmod 256 \\&= 7752 \bmod 256 & &= 4161 \bmod 256 \\&= 72 & &= 65 \\c_7 &= 40 & c_8 &= 243 \\p_7 &= y * (c - s) \bmod m & p_8 &= y * (c - s) \bmod m \\&= 57 * (40 - 8) \bmod 256 & &= 57 * (243 - 8) \bmod 256\end{aligned}$$

$$\begin{aligned}
 &= 1824 \bmod 256 \\
 &= 32 \\
 c_9 &= 153 \\
 p_9 &= y * (c - s) \bmod m \\
 &= 57 * (153 - 8) \bmod 256 \\
 &= 8265 \bmod 256 \\
 &= 73 \\
 c_{11} &= 144 \\
 p_{11} &= y * (c - s) \bmod m \\
 &= 57 * (144 - 8) \bmod 256 \\
 &= 7752 \bmod 256 \\
 &= 72 \\
 c_{13} &= 81 \\
 p_{13} &= y * (c - s) \bmod m \\
 &= 57 * (81 - 8) \bmod 256 \\
 &= 4161 \bmod 256 \\
 &= 65 \\
 &= 13395 \bmod 256 \\
 &= 83 \\
 c_{10} &= 81 \\
 p_{10} &= y * (c - s) \bmod m \\
 &= 57 * (81 - 8) \bmod 256 \\
 &= 4161 \bmod 256 \\
 &= 65 \\
 c_{12} &= 81 \\
 p_{12} &= y * (c - s) \bmod m \\
 &= 57 * (81 - 8) \bmod 256 \\
 &= 4161 \bmod 256 \\
 &= 65 \\
 c_{14} &= 198 \\
 p_{14} &= y * (c - s) \bmod m \\
 &= 57 * (198 - 8) \bmod 256 \\
 &= 10830 \bmod 256 \\
 &= 78
 \end{aligned}$$

Tabel 4 Hasil dekripsi

	«	u	)	ó	•	Q	(	ó	™	Q	•	Q	Q	Æ
CT	171	117	41	243	144	81	40	243	153	81	144	81	81	198
PT	75	69	89	83	72	65	32	83	73	65	72	65	65	78
	K	E	Y	S	H	A		S	I	A	H	A	A	N

Tabel 4 adalah hasil dekripsi dari ciphertext. Plaintext berada pada baris terakhir dalam tabel. Karakter-karakter tersebut akan kembali seperti awal yaitu “KEYSHA ALISHA”. Seperti pada proses enkripsi, karakter yang berulang akan mempunyai plaintext yang sama. Seperti pada contoh di atas, ada empat buah karakter “Q” sehingga ada nilai 65 sebanyak empat kali pada hasil dekripsi.

4. KESIMPULAN

Affine Cipher mempunyai banyak kelebihan dibandingkan pendahulunya Caesar Cipher. Algoritma ini menggunakan dua buah bilangan prima untuk melakukan proses modulo. Jenis ini termasuk kunci asimetrik dimana kunci yang digunakan untuk melakukan enkripsi berbeda dengan kunci pada saat dekripsi. Ini berfungsi untuk meningkatkan keamanan sehingga jika ada pihak-pihak yang ingin mencoba untuk memecahkan ciphertext, mereka harus mengetahui tiga buah pasangan bilangan yang digunakan yaitu kunci enkripsi, kunci dekripsi dan pergeseran karakter. Tetapi algoritma ini memiliki kelemahan. Huruf yang sering muncul pada suatu dokumen akan memiliki ciphertext yang sama, sehingga mudah ditebak dengan serangan berdasarkan kemunculan karakter. Selain itu, Affine Cipher tidak terlalu aman, karena kunci dapat dipecahkan dengan teknik exhaustive search, yang mencari pasangan bilangan prima yang digunakan.

Referensi

- [1] "Affine Cipher," Crypto Corner, [Online]. Available: <http://crypto.interactive-maths.com/affine-cipher.html>. [Accessed 1 September 2017].
- [2] "The Role of Cryptography in Information Security," Infosec Institute, 11 June 2012. [Online]. Available: <http://resources.infosecinstitute.com/role-of-cryptography/#gref>. [Accessed 1 September 2017].
- [3] S. A. Babu, "Modification Affine Ciphers Algorithm For Cryptographic Password," *International Journal of Research In Science & Engineering*, vol. 3, no. 2, pp. 346-351, 2017.
- [4] P. Sundarayya, M. ... Prasad and K. Satyam, "Some Technique Algorithms of Extension of Affine Cipher Cryptosystem Using Residue Modulo Prime Number," *Open Journal of Applied & Theoretical Mathematics*, vol. 2, no. 4, pp. 88-98, 2016.
- [5] D. A. Hammood and M. A. Naji, " Implementation And Enhancement Affine Cipher Of Database," *Journal of Engineering and Sustainable Development*, vol. 20, no. 4, pp. 264-276, 2016.
- [6] A. P. U. Siahaan and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *International Journal of Security and Its Application*, vol. 10, no. 8, pp. 173-180, 2016.
- [7] B. Oktaviana and A. P. U. Siahaan, "Three-Pass Protocol Implementation in Caesar Cipher Classic Cryptography," *IOSR Journal of Computer Engineering*, vol. 18, no. 4, pp. 26-29, 2016.