Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

Penerapan Multiply With Carry Generator pada Proses Pembangkitan Kunci Algoritma Beaufort Cipher

Taronisokhi Zebua

Fakultas Ilmu Komputer, Teknologi Informasi, Universitas Budi Darma, Medan
Jln. Sisingamangaraja No. 338, Siti Rejo I, Kec. Medan Kota, Kota Medan, Sumatera Utara, Indonesia
Email: taronizeb@gmail.com
Email Penulis Korespondensi: taronizeb@gmail.com
Submitted: **15/01/2023**; Accepted: **23/01/2023**; Published: **29/01/2023**

Abstrak−Kekuatan algoritma yang digunakan dalam pengamanan data tidak hanya tergantung pada rumitnya algoritma yang digunakan, namun terletak pada keacakan dan kerumitan peemcahan kunci yang digunakan. Pengulangan karakter yang sama dalam pembentukan kunci yang digunakan oleh sebuah algoritma pengamanan data sangat rentang terhadap penyerangan, karena dapat memudahkan para penyerang untuk memecahkan algoritma itu sendiri. Salah satu algoritma pengamanan data yang umum digunakan dalah algoritma beufort cipher. Jumlah karakter kunci yang dibutuhkan pada algoritma ini berbanding lurus atau sama dengan jumlah banyaknya karakter teks atau data yang diamankan. Penggunaan jumlah karakter kunci yang seperti ini memungkinan terjadinya pengulangan karakter yang sama, sehingga memudahkan penyerang untuk mengetahui pola dari kata kunci yang digunakan. Berdasarkan hal tersebut, maka pembangkitan kunci yang acak sangatlah diperlukan untuk mengoptimalkan ketahanan sebuah algoritma yang digunakan. Multiply with carry generator merupakan sebuah teknik pembangkitan bilangan acak semu yang memiliki keunggulan dalam kecepatan dan panjang siklus pengacakan sehingga memiliki variasi seed yang lebih banyak terhadap bilangan acak yang dihasilkan. Penelitian ini menguraikan bagaimana penerapan algoritma pembangkitan bilangan acak ini untuk menghasilkan kata kunci yang digunakan pada algoritma beaufort cipher, sehingga kunci yang digunakan dapat lebih optimal dan rumit untuk dipecahkan oleh penyerang.

Kata Kunci: Kriptografi; Beaufort; MWCG; Kunci; Keamanan

Abstract−The strength of the algorithm used in data security does not only depend on the complexity of the algorithm used, but lies in the randomness and complexity of the key solving used. The repetition of the same character in the formation of a key used by a data security algorithm is very vulnerable to attack, because it can make it easier for attackers to break the algorithm itself. One of the commonly used data security algorithms is the comfort cipher algorithm. The number of key characters needed in this algorithm is directly proportional to or equal to the number of characters of text or data that is secured. Using a number of key characters like this allows the repetition of the same characters, making it easier for attackers to find out the pattern of the keywords used. Based on this, random key generation is necessary to optimize the robustness of the algorithm used. Multiply with carry generator is a random number generation technique that has advantages in speed and length of randomization cycle so that it has more seed variations to the generated random numbers. This study describes how to apply this random number generation method to generate keywords used in the beaufort cipher algorithm, so that the keys used can be more optimal and more complicated for attackers to crack.

Keywords: Cryptography; Beaufort; MWCG; Key; Security

1. PENDAHULUAN

Salah satu teknik yang dapat digunakan upaya pengamanan dan penjagaan terhadap kerahasiaan data adalah teknik kriptografi. Teknik ini memiliki banyak algoritma yang dapat digunakan untuk mengimplementasikannya dalam berbagai jenis data yang bersifat penting dan rahasia. Hal ini dilakukan dengan tujuan agar kerahasiaan data tetap terlindungi dan tidak mudah disalah gunakan oleh pihak-pihak yang tidak berkepentingan. Secara umum, dalam penerapan algoritma teknik kriptografi memerlukan kunci sebagai salah satu elemen penting yang harus ada dalam mewujudkan tujuannya, karena dengan pemanfaatan kunci maka kekuatan sebuah algoritma yang digunakan dalam pengamanan data penting menjadi lebih optimal.

Ketahanan dan kekuatan sebuah algoritma kriptografi dalam mengamankan data dapat diukur dari kunci yang digunakan dalam mengamankan data dan sekaligus menjadi ukuran kekuatan dari penerapan teknik kriptografi itu sendiri [1]. Tidak semua algoritma kriptografi yang ada saat ini menerapkan struktur pembangkitan kunci yang kuat, sehingga pola-pola yang digunakan untuk membangkitkan kunci sangat mudah diketahui oleh para penyerang. Hal inilah yang mendorong lahirnya dan dikembangkannya berbagai teknik untuk memperbaiki struktur pembangkitan kunci algoritma teknik kriptografi. Salah satu upaya yang banyak digunakan adalah meningkatkan keacakan karakter kunci yang digunakan, sehingga dapat menyembunyikan pola-pola yang digunakan dalam pembangkitan kunci. Pembangkitan kunci secara acak juga dapat mengomptimalkan ketahanan algoritma terhadap upaya penyeranganan dar pihak yang tidak bertanggung jawab [2].

Beufort cipher merupakan salah satu algoritma kriptografi klasik yang dapat digunakan untuk mengamankan data rahasi. Kunci yang digunakan oleh algoritma ini dalam proses penyandian data memiliki struktur yang masih sederhana, dimana jumlah kunci yang digunakan harus sama dengan jumlah karakater data yang diamankan, karena pada algoritma ini masing-masing karakter data yang diamankan harus memiliki karakter kunci sebagai pasangannya. Bila jumlah data asli lebih panjang dari kata kunci yang disediakan oleh pengguna, maka dapat memungkinkan pengguna algoritma ini mengulangi penggunaan kata kunci secara periodik hingga jumlahnya sama dengan jumlah karakter data asli [3].

Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

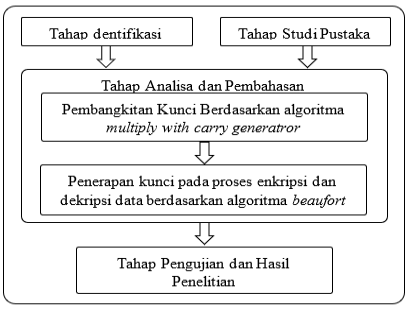
Multiply With Carry Generator (MWCG) merupakan salah satu algoritma yang digunakan untuk membangkitkan sejumlah bilangan secara acak. Kecepatan dan panjang siklus pengacakan yang dihasilkan menjadi keunggulan dari algoritma ini [4]. Algoritma pembangkitan bilangan acak MWCG ini telah banyak digunakan pada aplikasi pemrograman karena beberapa variabel yang dimiliki dapat mendeklarasikan modulus-modulus dari multiply with carry dan nilai acaknya.

Penelitian ini menguraikan bagaimana penerapan algoritma multiply with carry generator untuk membangkitkan kunci yang digunakan dalam proses pengamanan data rahasia berdasarkan algoritma beaufort cipher. Modifikasi pembangkitan kunci ini diharapkan dapat meminimalkan pengulangan karakter kunci sehingga dapat mempersulit para penyerang untuk mengetahui pola-pola pembentukan kunci dengan mudah. Pembangkitan kunci ini juga dapat memudahkan pengguna untuk mengingat kata kunci yang digunakan baik dalam proses enkripsi maupun dekripsi.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini berfokus pada penguraian tahap-tahap pembangkitan kunci pada algoritma beaufort cipher, dimana metode yang digunakan untuk membangkitkan kunci adalah algoritma multiply with carry generator. Adapun tahap-tahap yang dilakukan dalam menyelesaikan penelitian ini disajikan pada gambar di bawah ini.



Gambar 1. Tahap Pelaksanaan Penelitian

Gambar 1 di atas mendeskripsikan tahap-tahap yang dilakukan dalam menyelesaikan penelitian ini. Tahapan penelitian diawali dengan melakukan identifikasi terhadap masalah kekuatan kunci yang digunakan oleh sebuah algoritma kriptografi yang didukung oleh berbagai referensi yang relevan baik yang bersumber dari jurnal maupun buku. Tahap berikutnya adalah melakukan analisa dan pembahasan mengenai penerapan algoritma multiply with carry generator untuk membangkitkan kunci yang digunakan pada proses enkripsi maupun dekripsi data rahasia berdasarkan algoritma beuafort cipher. Pengujian keacakan bilangan yang dihasilkan merupakan tahap akhir dalam penelitian ini untuk melihat sejauh mana keacakan tersebut dapat menyembunyikan pola-pola pembangkitannya.

2.2 Kriptografi

Teknik kriptografi merupakan salah satu teknik yang dapat digunakan untuk mengamankan data penting atau data yang bersifat rahasia. Teknik kriptografi mengamankan data dengan cara merubah karakter data penting menjadi karakter yang baru (menyandikan). Artinya bahwa hasil akhir dari penerapan teknik kriptografi adalah sandi-sandi atau simbol-simbol baru yang tidak lagi sama dengan data aslinya. Teknik kriptografi melakukan proses enkripsi untuk menyandikan data asli, sehingga dihasilkan data sandi (cipher), sedangkan untuk mengembalikan data tersandi (cipher) menjadi data asli (plain), maka dilakukan proses dekripsi.

Secara umum, teknik kriptografi bertujuan untuk menjaga kerahasiaan data asli (plaintext), menjaga integritas data, otentikasi dan nirpenyangkalan data asli dari tindakan pihak lain yang tidak berhak mengakses data tersebut melalui berbagai algoritma yang dimilikinya [5], [6]. Elemen-elemen penting dalam penerapan algoritma kriptografi antara lain data yang diamankan, algoritma dan kunci. Kunci yang digunakan dalam mengamankan data bersifat rahasia dan harus dijaga, karena memiliki peran penting dalam mendukung keoptimalan algoritma yang digunakan dalam mengamankan data [7], [8].

2.3 Beaufort Cipher

Beaufort cipher merupakan algoritma yang dikembangkan dari algoritma vegenere cipher. Algoritma ini mengamankan data dengan teknik subtitusi sederhana mirip seperti algoritma vegenere cipher. Perbedaan kedua algoritma ini adalah pada formula untuk melakukan proses enkripsi dan dekripsi. Berdasarkan algoritma ini, cipher didapatkan berdasarkan operasi pengurangan antara nilai karakter plaintext dengan nilai karakter kunci dan dilakukan berdasarkan prosedur algoritmanya. Banyaknya kunci yang digunakan pada algoritma ini harus berbanding lurus atau sama dengan jumlah banyaknya karakter plaintext [9], [10]. Umumnya, jumlah kata kunci

Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

yang panjang atau sama banyaknya dengan jumlah plain merupakan salah satu kelemahan utama dari algoritma ini. Pengguna sangat susah menghafal kunci yang terlalu panjang, sehingga memungkinkan pengguna melakukan penggunaan kata kunci secara periodik hingga jumlahnya sama dengan plain. Kelemahan ini juga sangat memudahkan para penyerang, karena dengan pengulangan kata kunci yang sama, maka penyerang dapat dengan mudah mempelajari pola kunci yang digunakan pada algoritma ini, Adapun formulasi yang digunakan dalam proses penerapan algoritma ini [3] [11] [12] adalah :

Formulasi untuk melakukan enkripsi Ci = (Ki – Pi) mod 26 (1)

Formulasi untuk melakukan dekripsi Pi = (Ki – Ci) mod 26 (2)

Keterangan :
Pi = Plaintext
Ci = Ciphertext
Ki = Key (Kunci)
Mod 26 = nilai modulus yang gunakan.

Saat ini, nilai modulus yang digunakan tidak lagi hanya sebanyak 26, namun disesuaikan dengan jumlah karakter yang dibaca oleh komputer (nilai ASCII, yaitu 256).

2.4 Multiply With Carry Generator

Deretan bilangan acak dapat dihasilkan dengan melakukan proses pembangkitan berdasarkan algoritma pengacakan bilangan acak. Bilangan yang dihasilkan selama proses pengacakan berjalan adalah acak dan akan berhenti bila telah sampai pada nilai yang menjadi batas akhir atau batas berhentinya proses pengacakan [4] [13]. Algoritma Multiple With Carry Generator (MWCG) merupakan salah satu algoritma yang biasa digunakan untuk membangkitkan bilangan acak semu. Siklus yang terjadi dalam proses pembangkitan bilangan acak dapat disembunyikan dan dilakukan dengan cepat [14] [15]. Inilah yang menjadi salah satu kelebihan atau keunggulan dari algoritma ini. Adapun formula yang digunakan untuk menghasilkan bilangan acak berdasarkan algoritma ini adalah [4] [16] [17]:

$$X_n = (a\, X_{n-1} + C_{n-1}) \bmod b \quad (3)$$

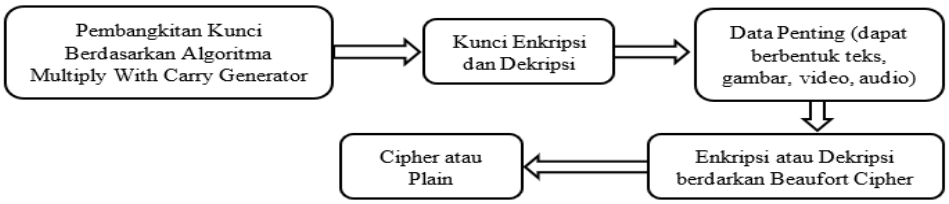
$$C_n = \left[\frac{a\, X_{n-1} + C_{n-1}}{b}\right] \quad (4)$$

Keterangan :
X_n = Bilangan acak ke-n
X_{n-1} = Bialangan acak ke n-1
a = Konstanta pengali
C_n = Konstanta kenaikan ke-n
C_{n-1} = Konstanta kenaikan ke n-1
b = Konstanta modulus dan pembagi

Syarat yang harus terpenuhi agar siklus keacakan yang lebih baik dari algoritma ini adalah $C_0 < b$ dan ab-1 adalah bilangan prima.

3. HASIL DAN PEMBAHASAN

Penelitian ini forkus untuk menguraikan bagaimana penerapan metode multiply with carry generator untuk membangkitkan bilangan acak yang digunakan sebagai kunci pada proses enkripsi dan dekripsi data penting berdasarkan algoritma beaufort cipher. Penelitian ini didasari oleh identifikasi masalah keamanan data, lalu melakukan studi literatur tentang penerapan teknik kriptografi serta implementasi dan pengujian. Adapun skema yang dulakukan dalam menyelesaikan penelitian ini, ditunjukkan pada bagan di bawah ini.



Gambar 2. Skema Pembangkitan Kunci

Saat ini algoritma kriptografi yang telah dikembangkan untuk menjaga keamanan data memang telah banyak, namun elemen terpenting dari penerapannya adalah kekuatan kunci yang digunakan. Kekuatan kunci menjadi salah satu elemen penting yang dapat mempertahankan algoritma yang digunakan dari tindakan penyerangan.

Salah satu permasalahan dalam struktur pembentukan kunci algoritma kriptografi adalah kata kunci yang digunakan tidak acak dan terjadi perulangan penggunaan kata kunci. Kelemahan ini menjadikan para penyerang dapat dengan mudah mengetahui pola pembentukan kunci yang bermuara pada mudahnya pemecahan algoritma.

Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

Pengacakan kata kunci yang digunakan dalam algoritma kriptografi menjadi salah satu solusi yang dapat digunakan untuk mengoptimalkan ketahanan algoritma yang digunakan dalam mengamankan data.

3.1 Pembangkitan Kunci Beaufort Cipher

Berdasarkan formulasi penerapan algoritma beaufort cipher dalam mengamankan data penting, maka jumlah kunci yang digunakan dalam proses enkripsi dan dekripsi harus berbanding lurus atau sama dengan jumlah karakter plain yang akan dienkripsi maupun dekripsi. Artinya bahwa masing-masing karakter plain harus memiliki pasangan kunci agar dapat diproses berdasarkan algoritma ini.

Sebagai contoh misalnya data asli yang diamankan berupa data teks dengan jumlah karakter 100 karakter, maka untuk menyandikan plaintext ini dibutuhkan 100 karakter kunci. Umumnya, jumlah kata kunci yang banyak akan sangat sulit dihafal oleh pengguna. Inilah yang menyebabkan pengguna dapat menggunakan kata kunci yang pendek, namun melakukan pengulangan kunci pendek tersebut agar jumlahnya mencapai 100 kata kunci.

Plaintext : Universitas_Budidarma_Medan (27 words)
Key : Saya12345Saya12345Saya1234 (27 words)

Berdasarkan contoh di atas, maka terlihat adanya pengulangan penggunaan kata kunci secara periodik, sehingga ini sangat mudah dipecahkan bila penyerang mengetahui pola perulangan kata kunci tersebut.

3.1.1 Pembangkitan Kunci Berdasarkan Algoritma Multiply With Carry Generator

Proses pembangkitan kunci berdasarkan algoritma ini bertujuan untuk menghasilkan sejumlah bilangan acak yang akan digunakan sebagai kunci dalam proses enkripsi maupun dekripsi data rahasia berdasarkan algoritma beuafort cipher.

Plaintext : Universitas_Budidarma_Medan (27 words)

Berdarkan jumlah plaintext di atas, maka akan dibangkitkan 27 bilangan acak sebagai kunci berdasarkan algoritma multiply with carry generator dengan a = 28, b = 700, X_0 = 33, C_0 = 17. Bila hasil proses perhitungan Xn dan Cn bernilai desimal, maka akan dilakukan pembulatan.

$X_1 = ((28 * 33) + 17) \bmod 700$
$X_1 = 241$
$C_1 = \left\lfloor \frac{(28 * 33) + 17}{700} \right\rfloor$
$C_1 = 3$, maka nilai desimal kunci untuk karakter pertama dari plaintext atau ciphertext adalah 3 (nilai C_1)

$X_2 = ((28 * 241) + 3) \bmod 700$
$X_2 = 451$
$C_2 = \left\lfloor \frac{(28 * 241) + 3}{700} \right\rfloor$
$C_2 = 23$, maka nilai desimal kunci untuk karakter kedua dari plaintext atau ciphertext adalah 23 (nilai C_2)

$X_3 = ((28 * 451) + 23) \bmod 700$
$X_3 = 54$
$C_3 = \left\lfloor \frac{(28 * 451) + 23}{700} \right\rfloor$
$C_3 = 42$

$X_4 = ((28 * 54) + 42) \bmod 700$
$X_4 = 166$
$C_4 = \left\lfloor \frac{(28 * 54) + 42}{700} \right\rfloor$
$C_4 = 5$

$X_5 = ((28 * 166) + 5) \bmod 700$
$X_5 = 455$
$C_5 = \left\lfloor \frac{(28 * 166) + 5}{700} \right\rfloor$
$C_5 = 16$

Proses pembangkitan nilai acak ini dilakukan hingga C_{27} dengan cara yang sama seperti di atas, sehingga diperoleh 27 buah nilai bilangan acak yang digunakan sebagai kunci dalam proses enkripsi atau dekripsi berdasarkan algoritma beautfor cipher. Adapun 27 nilai bilangan acak tersebut adalah 3, 23, 42, 5, 16, 42, 14, 51, 4, 55, 27, 43, 27, 30, 60, 54, 65, 44, 64, 15, 39, 36, 39, 47, 8, 22, 31.

3.1.2 Proses Enkripsi Berdasarkan Algoritma Beaufort Cipher

Kunci yang digunakan pada proses ini adalah bilangan acak yang telah dibangkitkan berdasarkan algoritma multiply with carry generator, sedangkan algoritma yang digunakan untuk melakukan proses enkripsi dan dekripsi adalah beuafort cipher. Sebelum proses enkripsi atau dekripsi dilakukan, maka karakter plaintext harus dirubah ke dalam bentuk nilai decimal sesuai tabel ASCII.

Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

Tabel 1. Nilai Desimal Karakter Plaintext

Karakter Plaintext	Nilai Decimal	Karakter Plaintext	Nilai Decimal
U	85	d	100
n	110	i	105
i	105	d	100
v	118	a	97
e	101	r	114
r	114	m	109
s	115	a	97
i	105	_	95
t	116	M	77
a	97	e	101
s	115	d	100
_	95	a	97
B	66	n	110
u	117		

Barisan bilangan acak yang dijadikan kunci adalah bilangan acak yang telah didapatkan dari pembangkitan bilangan acak berdasarkan algoritma multiply with carry generator sebelumnya, yaitu 3, 23, 42, 5, 16, 42, 14, 51, 4, 55, 27, 43, 27, 30, 60, 54, 65, 44, 64, 15, 39, 36, 39, 47, 8, 22, 31

Berdasarkan formulasi proses enkripsi dari algoritma beaufort cipher, maka yang dilakukan adalah operasi pengurangan antara nilai desimal kunci dengan nilai desimal setiap karakter plaintext kemudian dimoduluskan dengan 256.

Tabel 2. Proses dan Hasil Enkripsi

Karakter Plaintext (Pi)	Nilai Decimal Pi	Nilai Kunci (Ki)	Hasil Enkripsi Ci=(Ki-Pi) Mod 256	Karakter Ciphertext (Ci)	Karakter Plaintext (Pi)	Nilai Decimal Pi	Nilai Kunci (Ki)	Hasil Enkripsi Ci=(Ki-Pi) Mod 256	Karakter Ciphertext (Ci)
U	85	3	174	®	d	100	60	216	Ø
n	110	23	169	¨	i	105	54	205	Ì
i	105	42	193	Á	d	100	65	221	Ü
v	118	5	143	•	a	97	44	203	Ë
e	101	16	171	ª	r	114	64	206	Í
r	114	42	184	¸	m	109	15	162	¢
s	115	14	155	š	a	97	39	198	Å
i	105	51	202	Ê	_	95	36	197	Å
t	116	4	144	•	M	77	39	218	Ú
a	97	55	214	Õ	e	101	47	202	É
s	115	27	168	¨	d	100	8	164	£
_	95	43	204	Ë	a	97	22	181	µ
B	66	27	217	Ø	n	110	31	177	°
u	117	30	169	©					

Berdasarkan proses pada tabel 2 di atas, maka diperoleh ciphertext adalah :

Plaintext : Universitas_Budidarma_Medan

Ciphertext : ®¨Á• ª¸š Ê• Õ¨ËØ©ØÌÜËÍ¢ÅÅÚÉ£µ°

Proses dekripsi atau pengembalian ciphertext menjadi plaintext (teks asli) dilakukan dengan operasi pengurangan antara nilai kunci dengan desimal cipher. Proses pembangkitan kunci pada proses dekripsi ini sama dengan proses pembangkitan kunci pada proses enkripsi.

Tabel 3. Proses dan Hasil Dekripsi

Karakter Ciphertext (Ci)	Nilai Decimal	Nilai Kunci (Ki)	Hasil Enkripsi Ci=(Ki-Pi) Mod 256	Karakter Plaintext (Pi)	Karakter Ciphertext (Ci)	Nilai Decimal Ci	Nilai Kunci (Ki)	Hasil Enkripsi Ci=(Ki-Pi) Mod 256	Karakter Plaintext (Pi)
®	174	3	85	U	Ø	216	60	100	d
¨	169	23	110	n	Ì	205	54	105	i
Á	193	42	105	i	Ü	221	65	100	d
•	143	5	118	v	Ë	203	44	97	a
ª	171	16	101	e	Í	206	64	114	r

Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

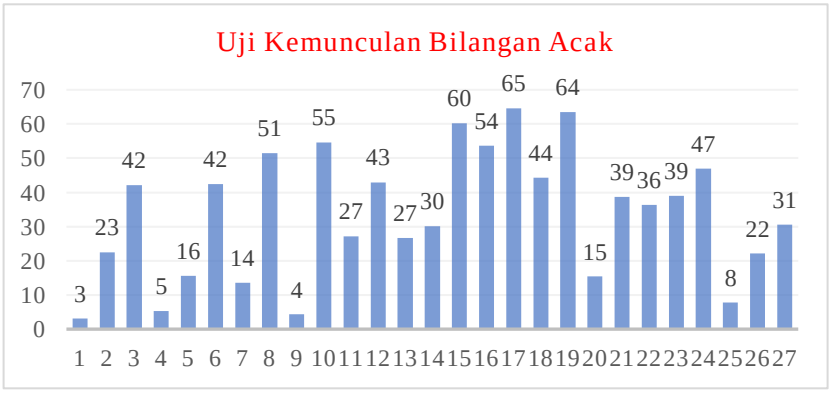
Karakter Ciphertext (Ci)	Nilai Decimal	Nilai Kunci (Ki)	Hasil Enkripsi Ci=(Ki-Pi) Mod 256	Karakter Plaintext (Pi)
¸	184	42	114	r
š	155	14	115	s
Ê	202	51	105	i
•	144	4	116	t
Õ	214	55	97	a
¨	168	27	115	s
Ë	204	43	95	_
Ø	217	27	66	B
©	169	30	117	u

Karakter Ciphertext (Ci)	Nilai Decimal Ci	Nilai Kunci (Ki)	Hasil Enkripsi Ci=(Ki-Pi) Mod 256	Karakter Plaintext (Pi)
¢	162	15	109	m
Å	198	39	97	a
Â	197	36	95	_
Ú	218	39	77	M
É	202	47	101	e
£	164	8	100	d
µ	181	22	97	a
°	177	31	110	n

Berdasarkan proses pada tabel 3 di atas, maka diperoleh plaintext adalah :
Ciphertext : ®¨Á• ª¸š Ê• Õ¨ËØ©ØÌÜËÍ¢ÅÅÚÉ£µ°
Plaintext : Universitas_Budidarma_Medan

3.2 Pengujian Keacakan (Run Test) Kunci

Pengujian yang dilakukan dalam penelitian ini difokuskan terhadap keacakan nilai kunci yang muncul apakah memiliki periodik dan angka yang sama atau tidak. Berdasarkan hasil pengujian yang dilakukan, diketahui bahwa penyebaran angka acak yang dihasilkan tidak memperlihatkan pola kunci. Hasil uji keacakan ini dapat diketahui melalui grafik di bawah ini.



Gambar 3. Grafik Kemunculan Bilangan Acak Kunci

Berdasarkan grafik pada gambar 3 di atas, terlihat bahwa kemunculan angka yang sama tidak dapat ditebak baik iterasi kemunculan maupun periodiknya. Misalnya bilangan 42, 27 dan 39 pada grafik di atas muncul dua periode, namun tidak dapat diprediksi kapan angka itu muncul.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan di atas, maka dapat disimpulkan bahwa dalam penerapan algoritma teknik kriptografi, kunci memiliki peran penting dalam mendukung ketahanan algoritma yang digunakan. Serumit apapun algoritma yang digunakan dalam mengamankan data, akan menjadi mudah diserang bila kunci yang digunakan tidak memiliki struktur yang baik dan kuat terhadap serangan. Pembangkitan bilangan acak berdasarkan algoritma multiply with carry generator dapat mengoptimalkan kekuatan kunci pada algoritma beautfort cipher. Hal ini dapat diketahui dari nilai-nilai kunci yang dibangkitkan cukup acak dan tidak memperlihatkan pola-pola kemunculan nilai yang sama. Berdasarkan pengujian yang dilakukan, diketahui bahwa kemunculan bilangan yang sama tidak dapat ditebak iterasi dan jarak kemunculannya. Pemanfaatan algoritma pembangkitan kunci secara acak juga dapat memudahkan dan mempercepat pengguna untuk menghasilkan kunci yang jumlahnya banyak tanpa mengingat kata kunci tersebut secara keseluruhan.

REFERENCES

[1] R. K. Hondro, “Modifikasi Platform Kunci Algoritma Playfair Untuk Meningkatkan Nilai Confusion Pada Ciphertext,” Journal of Computer System and Informatics (JoSYC), vol. 1, no. 2, pp. 76-82, 2020.

[2] M. Diana dan T. Zebua, “Optimalisasi Beufort Cipher Menggunakan Pembangkit Kunci RC4 dalam Penyandian SMS,” Jurnal Sains Komputer & Informatika (J-SAKTI), vol. 2, no. 1, pp. 12-22, 2018.

Journal of Information System Research (JOSH)
Volume 4, No. 2, Januari 2023, pp 607−613
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v4i2.2928

[3] M. Fadlan, S. Sinawati, A. Indriani dan E. D. Bintari, "Pengamanan Data Teks Melalui Perpaduan Algoritma Beaufort Cipher dan Caesar Cipher," Jurnal Teknik Informatika, vol. 12, no. 2, pp. 149-158, 2019.

[4] R. A. Saputra, I. T. Awalda dan B. Pramono, "Implementasi Algoritma Multiply With Carry Generator (MWCG) dalam Pengacakan Soal Ujian Semester Berbasis Web pada SMKN 1 Kendari," Jurnal Informatika Universitas Palembang, vol. 7, no. 1, pp. 60-67, 2022.

[5] E. Haryadi dan S. M. Ladjamuddin, "Teknik Keamanan Pesan Menggunakan Kriptografi dengan Algoritma Vernam Cipher," Incomtech, vol. 6, no. 1, pp. 40-47, 2017.

[6] H. dan M. A. Rony, "Implementasi Keamanan Database Dengan Menggunakan Metode Edvanced Encription Standard pada Sekolah SMK Islam Al Hikmah Jakarta Berbasis Dekstop," SKANIKA, vol. 8, no. 3, pp. 1137-1142, 2018.

[7] S. "Implemenmtasi Algoritma Kriptografi RSA (Rivest Shamir Adleman) untuk Keamanan Data Rekam Medis Pasien," INTECOMS, vol. 4, no. 1, pp. 104-114, 2021.

[8] D. H. Pane, "Implementasi Kriptografi Keamanan Data Resi pada PT JNE Perbaungan Menggunakan Metode Markle Hellman," Journal of Information System, Computer Science and Information Technology, vol. 1, no. 1, pp. 6-10, 2020.

[9] E. Ndruru dan T. Zebua, "Pembangkitan Kunci Beaufort Cipher dengan Teknik Blum-blum Shub pada Pengamanan Citra Digital," Buletin of Information Technology, vol. 2, no. 2, pp. 149-154, 2022.

[10] A. Lingga, "Analisa Implementasi Aplikasi Keamanan File Audio WAV dengan Menerapkan Algoritma Beaufort Cipher dan ROT 13," Jurnal Pelita Informatika, vol. 8, no. 1, pp. 33-40, 2019.

[11] A. Rachmadsyah, A. Perdana dan A. Budiman, "Kombinasi Algoritma Beaufort Cipher dan Vigenere Cipher untuk Pengamanan Pesan Teks Berbasis Mobile Aplication," Jurnal Minfo Polgan, vol. 9, no. 2, pp. 12-17, 2020.

[12] M. D. Irawan, "Implementasi Kriptografi Vigenere Cipher dengan PHP," E-Journal, vol. 1, no. 1, pp. 127-134, 2017.

[13] K. Paraditasari dan A. D. Wowor, "Desain Pembangkit Kunci Block Cipher Berbasis CSPRNG Chaso Menggunakan Fungsi Trigonometri," Jurnal Ilmiah Penelitian dan Pembelajaran Informatika, vol. 06, no. 02, p. 400–405, 2021.

[14] HandWiki, "HandWiki," Multiply-with-carry, 06 March 2021. [Online]. Available: https://handwiki.org/wiki/Multiply-with-carry. [Diakses 14 December 2022].

[15] Y. P. Mulya, R. Latuconsina dan A. S. R. Ansori, "Implementasi Algoritma Multiply With Carry pada Game Panahan," e-Proceeding of Engineering, vol. 7, no. 1, pp. 40-44, 2020.

[16] M. Goresky dan A. Klapper, "ACM Digital Libary," 01 October 2018. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/945511.945514. [Diakses 21 December 2022].

[17] W. F. Encyclopedia, "Wikipedia Free Encyclopedia," Multiply-with-carry pseudorandom number generator, 4 July 2022. [Online]. Available: https://en.wikipedia.org/wiki/Multiply-with-carry_pseudorandom_number_generator. [Diakses 11 December 2022].