JITSI: Jurnal Informatika Dan Teknologi Sistem Informasi
https://jurnal.interaksisaintek.com/index.php/JITSI
ISSN : 3123 - 6650 (Media Online)
Vol. 1 No. 1 (November 2024) Hal 40-50

IMPLEMENTASI LAGGED FIBONACCI GENERATOR PADA PROSES PEMBANGKITAN KUNCI ALGORITMA ONE TIME PAD (OTP)

Taronisokhi Zebua [1],

[1]Sistem Informasi, Fakultas Ilmu Komputer & Teknologi Informasi Universitas Budi darma , Medan, Indonesia
Email: [1]taronisokhi@gmail.com,

ABSTRACT

The effectiveness of the key used in a data security algorithm is one of the essential aspects that must be considered when implementing data security techniques. Many data security algorithms, particularly classical cryptographic algorithms, still employ key generation techniques that are not sufficiently optimized to ensure data security. As a result, secured data or information can be easily accessed by unauthorized parties.
Key resistance to attacks is a fundamental requirement in cryptographic techniques for protecting data against various forms of attacks. Resistance to data attacks does not depend solely on the complexity of the cryptographic algorithm used but is also significantly influenced by the optimization of the key to support the objectives of data security.
One technique that can be used to generate more effective keys is random key generation, in which the keys are designed to be unpredictable so that the patterns of the keys used to secure data are difficult for attackers to identify. One approach that can be employed is random number generation, which can be used to produce keys with unpredictable patterns and no easily identifiable relationships. The Lagged Fibonacci Generator algorithm is one of the variants of random number generation algorithms. The application of this algorithm in generating keys for cryptographic algorithms is expected to improve the effectiveness of the algorithms used to encrypt and decrypt confidential data or information.

Keyword: *Cryptography, Data, Information, Key, Random, LFG.*

ABSTRAK

Keoptimalan kunci yang digunakan pada sebuah algoritma keamanan data merupakan salah satu poin penting yang harus diperhatikan dalam penerapan teknik keamanan data. Ada banyak algoritma teknik keamanan data terutama algoritma teknik kriptografi yang masih tergolong klasik masih menggunakan teknik pembangkitan kunci yang belum optimal dengan baik untuk menjaga keamanan data, sehingga data atau informasi yang diamankan sangat dengan mudah dapat didapatkan oleh pihak-pihak yang tidak berhak.

Ketahanan kunci terhadap tindakan penyerangan data merupakan salah satu keharusan dalam penerapan teknik kriptografi untuk mengamankan data dari berbagai serangan. Ketahanan terhadap serangan data tidak hanya tergantung pada rumitnya algoritma kriptografi yang digunakan, namun sangat ditentukan oleh penggunaan kunci yang lebih optimal dalam mendukung tujuan teknik keamanan data.

Salah satu teknik yang dapat digunakan untuk menghasilkan kunci yang lebih optimal adalah melakukan pembangkitan kunci secara acak dan sulit ditebak, sehingga pola dari kunci yang digunakan dalam mengamankan data tidaklah mudah diketahui oleh penyerang. Salah satu teknik yang dapat digunakan adalah teknik pembangkitan angka secara acak (*random number*

generation). Teknik ini dapat dimanfaatkan untuk menghasilkan kunci-kunci yang saling tidak memiliki koresponden dengan pola yang susah ditebak. Algoritma *lagged fibonacci generator* merupakan salah satu dari variasi algoritma pembangkitan angka secara acak. Pemanfaatan algoritma ini dalam pembangkitan kunci yang digunakan pada sebuah algoritma teknik kriptografi diharapkan dapat lebih mengoptimlkan keampuhan dari algoritma yang digunakan untuk melakukan proses enkripsi dan dekripsi data atau informasi yang rahasia.

Kata Kunci: *Kriptografi, Data, Informasi, Kunci, Acak, LFG*

1. PENDAHULUAN

Peningkatan pemanfaataan media online dan media-media sosial dalam kegiatan berdistribusi data dan informasi yang sifatnya rahasia saat ini telah memicu perhatian terhadap penanganan dan usaha pengoptimalan keamanan data tersebut. Data penting atau informasi yang bersifat rahasia yang didistribusikan melalui media online sangat rentang terhadap tindakan pencurian atau tindakan lain yang dapat merugikan pihak pemilik data. Tentu hal ini dilakukan oleh pihak-pihak yang tidak diberikan akses untuk data tersebut (penyerang). Berdasarkan penelian terdahulu, mengatakan bahawa dibutuhkan sebuah tindakan pengamanan terhadap data penting atau informasi yang bersifat rahasia agar data tersebut tetap terjaga serta terhindar dari tindakan-tindakan pencurian data tersebut [1].

Teknik kriprografi merupakan salah satu teknik yang dapat digunakan dalam mengamankan data atau informasi yang bersifat penting dan rahasia. Teknik ini mengamankan data penting dengan menyandikan data penting menjadi data yang sulit dipahami oleh orang lain, sehingga dapat meminimalkan tindakan-tindakan para pihak yang tidak berhak untuk itu. Teknik ini dapat diimplementasikan dengan memanfaatkan berbagai algoritma yang dimiliki oleh teknik ini, sehingga kerahasiaan, integritas dan otentikasi data dapat tetap terjaga. Berdarkan penelitian terdahulu, mengatakan bahwa teknik kriptografi menjadi salah satu teknik yang dapat dimanfaatkan untuk mengatasi masalah keamanan data, sehingga data tetap terjadi dan terhindar dari tindakan manipulasi data yang dilakukan oleh pihak lain [1] [2].

Teknik kriptografi ini memiliki berbagai algoritma untuk mengimplementasikannya dalam mengamankan data. Salah satunya adalah algoritma One Time Pad (OTP). Selain elemen algoritma, maka elemen penting yang diperlukan dalam penerapan teknik kriptografi ini adalah kunci. Peranan kunci dalam penerapan algoritma kriptografi sangat penting, karena dengan pemanfaatan kunci yang kuat, maka keoptimalan keamanan terhadap data yang diamankan tentu lebih optimal pula. Kunci ini harus dijaga kerahasiaannya sehingga tidak mudah didapatkan oleh orang lain yang tidak memiliki hak untuk mengakses data. Namun dari beberapa algoritma teknik kriptografi khususnya yang masih tergolong klasik, masih menggunakan komposisi pembentukan kunci yang kurang optimal, sehingga masih mudah dipecahkan atau diketahui oleh para penyerang. Penelitian terdahulu mengatakan bahwa kunci dalam algoritma kriptografi sangat memiliki pengaruh yang kuat untuk mewujudkan tujuan pengamanan data [3].

Salah satu algoritma kriptografi yang tergolong klasik adalah algoritma One Time Pad (OTP). Algoritma ini mengamankan data penting dengan kunci yang memiliki panjang sama dengan jumlah panjang data yang diamankan. Umumnya algoritma ini dikatakan bahwa termasuk salah satu algoritma yang rumit dipecahkan, namun dilihat dari sisi kunci yang memiliki jumlah yang sama dengan jumlah datanya, maka akan sangat sulit dibentuk dan diingat oleh pengguna. Selain hal itu, kunci yang panjang dapat mengakibatkan penggunaan

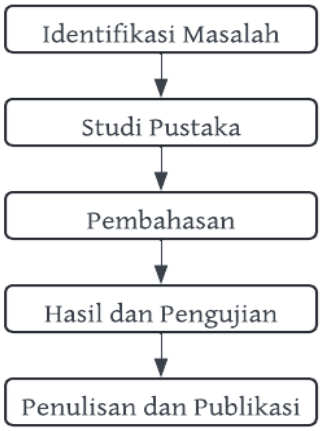
kata kunci yang berulang secara periodik untuk mencapai jumlah yang sama dengan jumlah data yang akan diamanakan. Penelitian terdahulu mengatakan bahwa salah satu kelemahan dari algoritma ini adalah nilai kunci yang dibangkitkan pada proses enkripsi akan sulit mendapatkan nilai dan periodik yang sama pada saat pembangkitan kunci pada proses dekripsi [4].
Pembangkitan bilangan acak merupakan salah satu teknik yang umum digunakan untuk mendapakan deret bilangan yang acak. Teknik ini juga banyak digunakan dalam mendukung kinerja algoritma kriptografi. Salah satu algoritma yang dapat digunakan dalam menghasilkan bilangan acak adalah algoritma lagged fibonacci generator. Algoritma ini merupakan generalisasi dari bentuk barisan Fibonacci yang sangat acak dengan memanfaatkan beberapa nilai awal untuk menghasilkan rangkaian nilai bilangan acak [5] [6].

Penelitian ini menguraikan bagaimana prosedur yang dilakukan untuk membangkitkan kunci dari algoritma one time pad berdasarkan pembangkit bilangan acak lagged fibonacci generator yang akan diimplementasikan dalam proses enkripsi dan dekripsi data teks, sehingga diharapkan kinerja dari algorima one time pad lebih meningkat dalam menjamin kemanan data atau informasi penting.

.

2. METODE

Pelaksanaan penelitian ini meliputi beberapa fase yang dilakukan untuk mencapai tujuan yang diharapkan. Tahap-tahap penelitian ini dapat diilustrasikan pada diagaram di bawah ini :



Gambar 1. Diagram Tahapan Penelitian

1. Identifikasi Masalah
   Tahap ini merupakan kegiatan untuk mengidentifikasi permasalahan dan solusi yang dapat mengoptimalkan keamanan data yang dilakukan berdasarkan algoritma *one time pad*.
2. Studi Pustaka
   Tahap ini merupakan tahap untuk mengumpulkan dan mempelajari berbagai teori yang menjadi referensi bagi peneliti untuk menyelesaikan penelitian ini.
3. Pembahasan
   Tahap ini merupakan kegiatan yang dilakukan untuk menguraikan bagaimana mengimplementasikan pembangkitan bilangan acak yang digunakan sebagai kunci dalam mengamankan data berdasarkan algoritma *one time pad*.
4. Hasil dan Pengujian

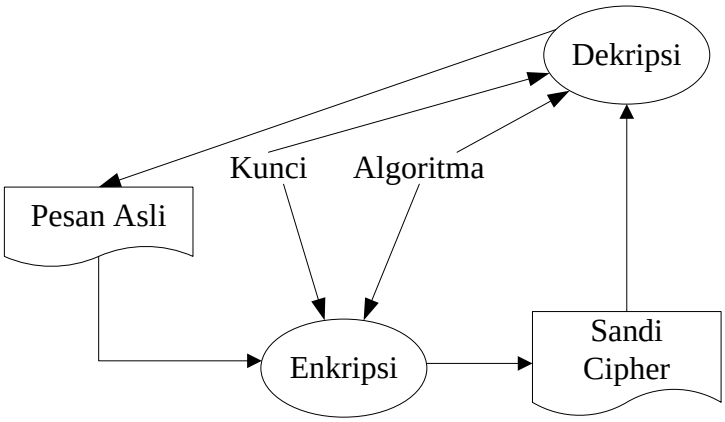
Tahap ini merupakan tahap untuk menguji tingkat keoptimalan kunci yang dibangkitkan dan telah diterapkan pada algoritma *one time pad* dalam mengamanakan pesan teks

5. Penulisan dan Publikasi Penelitian
   Tahap ini merupakan tahap akhir untuk menyiapkan bahan atau draf penelitian yang akan dipublikasi.

3. HASIL DAN PEMBAHASAN

Permasalah keamanan data memang menjadi salah satu masalah penting yang harus diperhatikan pada perkembangan teknologi saat ini. Sangat banyak kasus yang terjadi saat ini terkait dengan keamanan data penting, misalnya pencurian data, manipulasi data, pemalsuan data dan tindakan lain yang menyebabkan kerugian sendiri bagi pemilik maupun penerima data.

Teknik kriptografi merupakan salah satu teknik pengamanan data yang sifatnya penting. Teknik kriptografi bekerja dengan melakukan mengubah data asli menjadi sandi-sandi yang tidak dapat lagi mendeskripsikan pola dan makna asli dari data yang sebenarnya. Pemanfaatan teknik ini dapat meminimalisir kemudahan bagi orang lain yang tidak berhak mengakses data penting atau rahasia untuk melakukan tindakan-tindakan pencurian, manipulasi maupun penggunaan informasi tersebut untuk tujuan tertentu yang dapat merugikan pemilik data.



Gambar 4.1. Skema Pengamanan Pesan Berdasarkan Teknik Kriptogafi

Algoritma kriptografi yang bersifat umum (*public*), telah memberikan keleluasaan bagi para pengembang maupun para penyerang untuk mencari celah pemecahan dari berbagai algoritma kriptografi yang umum digunakan dalam mengamankan data, termasuk algoritma *one time pad*. Hal ini jugalah yang menyebabkan lahirnya pemutakhiran atau perbaikan terhadap berbagai algoritma kriptografi agar tetap berfungsi sebagai salah satu alternatif algoritma yang dapat dihandalkan untuk mengamankan data. Pengembangan yang dilakukan bertujuan untuk mengoptimalkan kinerja sebuah algoritma untuk mencapai tujuan utama dari kriptografi yaitu *confusion* dan *diffusion*.

a. Pembangkitan Kunci Lagged Fibonacci Generator

Algoritma *Lagged Fibonacci Generator* (LFG) adalah salah satu metode untuk menghasilkan bilangan acak pseudorandom. Prinsip dasar dari LFG adalah menggunakan urutan *fibonacci* yang dimodifikasi dengan memperkenalkan lag atau keterlambatan dalam penghitungan.

1. Modulus (m)
   Biasanya merupakan bilangan besar yang digunakan untuk memastikan bahwa hasil akhir berada dalam rentang tertentu (pada penelitian ini menggunakan modulus 256, karena rentang karakter yang digunakan antara 0 – 255).
2. Operasi (op)
   Operasi yang digunakan, bisa berupa penambahan (+), pengurangan (-), perkalian (*), atau operasi XOR ($\oplus$).
3. Lags (k dan j)
   Dua angka yang menentukan jarak antara angka-angka sebelumnya dalam urutan.
4. State array (S)
   Sebuah *array* yang menyimpan nilai awal yang digunakan untuk memulai proses. *Array* ini biasanya diinisialisasi dengan nilai-nilai acak atau konstan.

Langkah-langkah algoritma *lagged fibonacci generator* adalah :

1. Inisialisasi
   a. Tentukan parameter: modulus (m), operasi (op), lags (k dan j), dan state array (S) yang berisi setidaknya max(k, j) elemen.
   b. Misalkan nilai-nilai awal dari S adalah [S0, S1, ..., S(max(k,j)-1)].
2. Perhitungan Iteratif
   Untuk setiap iterasi i dimulai dari max(k, j) hingga iterasi yang diinginkan:
   a. Hitung nilai baru S_i berdasarkan operasi yang telah dipilih
      1) Operasi penambahan
         $S_i = (S_i - k + S_{i-j}) \bmod m$
      2) Operasi perkalian
         $S_i = (S_i - k + S_{i-j}) \bmod m$
   b. Simpan hasilnya dalam array S
3. Pengambilan bilangan acak
   Nilai Si yang dihasilkan pada setiap iterasi dianggap sebagai bilangan acak yang dihasilkan oleh algoritma.

Sebagai salah satu contoh pembangkitan bilangan acak berdasarkan algoritma *lagged fibonacci generator* adalah :

Diketahui :

M = 256

Op (operasi) = + (penambahan)

k = 3 dan j = 5

State Array awal S = [64, 65, 66, 67, 68]

Berdasarkan nilai-nilai yang telah diketahui di atas, maka langkah selanjutnya adalah menentukan nilai S5, S6, S7, S8, S9.

$S_5 = (S_{5-3} + S_{5-5}) \bmod 256 = (S_2 + S_0) \bmod 256 = (66 + 64) \bmod 256 = 130$

$S_6 = (S_{6-3} + S_{6-5}) \bmod 256 = (S_3 + S_1) \bmod 256 = (67 + 65) \bmod 256 = 132$

$S_7 = (S_{7-3} + S_{7-5}) \bmod 256 = (S_4 + S_2) \bmod 256 = (68 + 66) \bmod 256 = 134$

$S_8 = (S_{8-3} + S_{8-5}) \bmod 256 = (S_5 + S_3) \bmod 256 = (130 + 67) \bmod 256 = 197$

$S_9 = (S_{9-3} + S_{9-5}) \bmod 256 = (S_6 + S_4) \bmod 256 = (132 + 68) \bmod 256 = 200$

Berdasarkan proses pembangkitan di atas, maka diperoleh bilangan acak yaitu 130, 132, 134, 197, 200.

Proses di atas, dapat dilanjutkan untuk mendapatkan nilai array S berikutnya sampai nilai yang dibutuhkan terpenuhi.

b. Algoritma One Time Pad

One-Time Pad (OTP) adalah teknik enkripsi dimana setiap karakter atau bit dari *plaintext* dikombinasikan dengan karakter atau bit dari kunci yang acak untuk menghasilkan *ciphertext*. Keamanan OTP terjamin selama kunci memenuhi beberapa kondisi penting:

1. Kunci harus benar-benar acak: Tidak boleh ada pola atau prediktabilitas dalam kunci.
2. Kunci harus seukuran dengan *plaintext*: Panjang kunci sama dengan panjang pesan yang akan dienkripsi.
3. Kunci hanya digunakan sekali: Tidak boleh ada pengulangan kunci untuk pesan yang berbeda.
4. Kunci harus dirahasiakan: Hanya pengirim dan penerima yang mengetahui kunci.

Langkah-langkah Algoritma *One-Time Pad*

1. Persiapan:
   a. *Plaintext* (Pesan Asli): Pesan yang akan dienkripsi.
   b. Kunci (*Key*): Deretan karakter atau bit yang acak, seukuran dengan *plaintext*.
2. Enkripsi:
   a. Metode Penambahan Modular (Untuk Huruf):
      1) Konversi setiap huruf *plaintext* dan kunci ke dalam angka (A=0, B=1, ..., Z=25).
      2) Tambahkan angka *plaintext* dengan angka kunci secara modulo 26.
      3) Konversi kembali hasil penjumlahan ke huruf untuk mendapatkan *ciphertext*.
   b. Metode XOR (Untuk Data Biner):
      Operasikan setiap bit *plaintext* dengan bit kunci menggunakan operasi XOR.

3. Dekripsi:
   a. Metode Penambahan Modular:
      1) Konversi *ciphertext* dan kunci ke angka.
      2) Kurangkan angka *ciphertext* dengan angka kunci secara modulo 26.
      3) Konversi kembali hasil pengurangan ke huruf untuk mendapatkan *plaintext* asli.
   b. Metode XOR
      Operasikan setiap bit *ciphertext* dengan bit kunci menggunakan operasi XOR untuk mendapatkan *plaintext*.

Dalam perkembangan penggunaannya, nilai modulus yang digunakan dalam operasi *one time pad* ini adalah 256 yang disesuikan dengan jumlah karakter ASCII.
Sebagai contoh penerapan algoritma *one time pad* adalah :

Plaintext	=	B	E	S	O	K	L	A	R	I
Decimal	=	66	69	83	79	75	76	65	82	73
Key	=	L	A	R	I	P	A	G	I	X
Decimal	=	76	65	82	73	80	65	71	73	88

Formula yang digunakan untuk melakukan enkripsi :
Ci = (Pi + Ki) mod 256, sehingga :

Plaintext	=	66	69	83	79	75	76	65	82	73
Key	=	76	65	82	73	80	65	71	73	88
(Pi+Ki) Mod 256	=	142	134	165	152	155	141	136	155	161
Charakter	=	Ž	†	¥	˜	›		ˆ	›	¡

Sehingga *cipher* yang dihasilkan dari plaintext BESOKLARI adalah Ž†¥˜›ˆ›¡

Proses dekripsi dilakukan dengan formula Pi = (Ci – Ki) mod 256

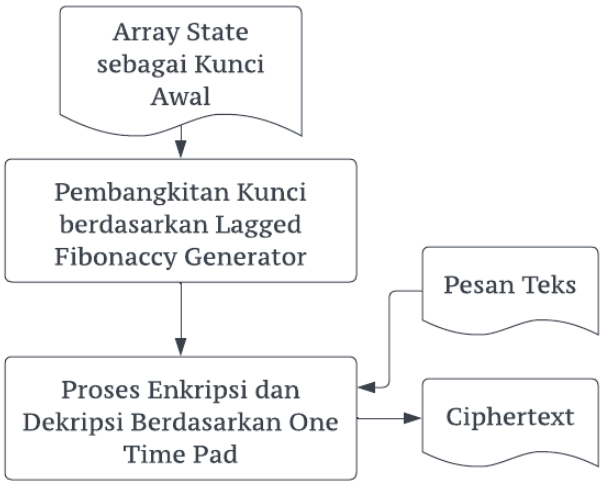
Ciphertext	=	Ž	†	¥	˜	›		ˆ	›	¡
Decimal		142	134	165	152	155	141	136	155	161
Key	=	76	65	82	73	80	65	71	73	88
(Pi+Ki) Mod 256	=	66	69	83	79	75	76	65	82	73
Charakter	=	B	E	S	O	K	L	A	R	I

Sehingga didapatkan *plaintext* semula yaitu BESOKLARI

Berdasarkan proses di atas, maka dapat diketahui bahwa kunci yang digunakan pada algoritma *one time pad* ini adalah kunci yang sangat panjang karena masing-masing *plaintext* yang dienkripsi harus memiliki pasangan kunci untuk mengenkripsi maupun dekripsi. Penggunaan kunci yang seperti ini juga akan menyulitkan para pengguna untuk mengingat kunci yang sangat panjang. Selain itu, dalam penggunaan kunci seperti ini ada kalanya akan menggunakan kata yang diulang untuk memenuhi kebutuhan kunci dari *plantext*, sehingga kunci yang digunakan sangat tidak efektif.

c. Kombinasi One Time Pad dan Lagged Fibonacci Generator

Pengkombinasian *one time pad* dengan *lagged fibonacci generator* dalam penelitian ini bertujuan untuk mengoptimalkan kekuatan kunci dari algoritma *one time pad* yang digunakan dalam proses enkripsi maupun dekripsi data rahasia. Bilangan yang dibangkitkan berdasarkan *lagged fibonacci generator* yang digunakan sebagai kunci oleh *one time pad*, akan lebih acak dan tidak perlu ingat oleh pengguna. *Array state* awal sebagai input pada *lagged fibonaccy* akan digunakan sebagai kunci awal dan nilai kunci ini yang harus diingat oleh pengguna. Diagram proses yang dilakukan untuk mengkombinasikan keduanya dapat dilihat pada diagaram berikut ini :



Gambar 3.1. Skema Kombinasi *Lagged Fibonaccy* dan *One Time Pad*

Berdasarkan diagram di atas, dapat diketahui *input*, proses dan *output* dalam kombinasi keduanya sebagai berikut :

1. *Input*

Input yang dibutuhkan terdiri dari dua yaitu *array state* yang dibutuhkan sebagai kunci awal untuk membangkitkan barisan bilangan kunci, kemudian pesan yang akan dienkripsi atau didekripsi.

2. Proses
   Proses yang dilakukan adalah :
   a. Proses pembangkitan barisan bilangan kunci berdasarkan *lagged fibonacci*
   b. Proses enkripsi *plaintext* berdasarkan *one time pad*
   c. Proses dekripsi *ciphertext* berdasarkan *one time pad*
3. *Output*
   a. Barisan kunci hasil *lagged fibonacci*
   b. *Ciphertext* (dari proses enkripsi) dan *plaintext* (dari proses dekripsi)

Sebagai contoh dalam mengimplementasikan pengkombinasian keduanya, akan diuraikan pada contoh berikut ini :

Pesan Teks (Plaintext) : BUDIDARMA MEDAN
Array State (kunci awal) : univBD
Plaintext dan array state di atas akan dirubah menjadi nilai decimal sebagai berikut :

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Plain	B	U	D	I	D	A	R	M	A		M	E	D	A	N
Dec	66	85	68	73	68	65	82	77	65	32	77	69	68	65	78
Array Sate	u	n	i	v	B	D									
Dec	117	110	105	118	66	68									

1. Proses pembangkitan kunci
   Tetapkan nilai yang dibutuhkan oleh lagged fibonanci
   M = 256
   Op (operasi) = + (penambahan)
   k = 3 dan j = 5
   State Array awal S = [117, 110, 105, 118, 66, 68] (diambil dari *state* awal)
   Jumlah karakter plaintext adalah 15 karakter, berarti membutuhkan kunci sebanyak 15 nilai. Selanjutnya, akan dilakukan pembangkitan nilai-nilai kunci tersebut, dimana 6 karakter kunci awal (*array state*) telah diketahui nilainya, sehingga yang dilakukan selanjutnya adalah membangkitkan nilai kunci ke-7 sampai nilai kunci ke-15.
   Berdasarkan nilai-nilai yang telah diketahui di atas, maka langkah selanjutnya adalah menentukan nilai S6, S7, S8, S9, S10,…,S14
   $S_6 = (S_{6-3} + S_{6-5}) \bmod 256 = (S_3 + S_1) \bmod 256 = (73 + 85) \bmod 256 = 158$
   $S_7 = (S_{7-3} + S_{7-5}) \bmod 256 = (S_4 + S_2) \bmod 256 = (68 + 68) \bmod 256 = 136$
   $S_8 = (S_{8-3} + S_{8-5}) \bmod 256 = (S_5 + S_3) \bmod 256 = (65 + 73) \bmod 256 = 138$
   $S_9 = (S_{9-3} + S_{9-5}) \bmod 256 = (S_6 + S_4) \bmod 256 = (82 + 68) \bmod 256 = 150$
   $S_{10} = (S_{10-3} + S_{10-5}) \bmod 256 = (S_7 + S_5) \bmod 256 = (77 + 65) \bmod 256 = 142$
   $S_{11} = (S_{11-3} + S_{11-5}) \bmod 256 = (S_8 + S_6) \bmod 256 = (65 + 82) \bmod 256 = 147$
   $S_{12} = (S_{12-3} + S_{12-5}) \bmod 256 = (S_9 + S_7) \bmod 256 = (32 + 77) \bmod 256 = 109$
   $S_{13} = (S_{13-3} + S_{13-5}) \bmod 256 = (S_{10} + S_8) \bmod 256 = (77 + 65) \bmod 256 = 142$
   $S_{14} = (S_{14-3} + S_{14-5}) \bmod 256 = (S_{11} + S_9) \bmod 256 = (69 + 32) \bmod 256 = 101$
   Sehingga dihasilkan deret bilangan yang telah dibangkitkan adalah :

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Array	u	n	i	v	B	D	ž	ˆ	Š	–	Ž	“	m	Ž	e

Sate															
Dec	117	110	105	118	66	68	158	136	138	150	142	147	109	142	101

Teks yang diwarnai merah merupakan nilai kunci yang telah dibangkitkan dari lagged *fibonacci generator*. Nilai kunci inilah awal dan nilai kunci yang telah dibangkitkan inilah yang digunakan untuk melakukan proses enkripsi dan dekripsi berdasarkan *one time pad*. Sehingga masing-masing karakter *plaintext* telah memiliki pasangan kunci masing-masing.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Plain	B	U	D	I	D	A	R	M	A		M	E	D	A	N
Dec	66	85	68	73	68	65	82	77	65	32	77	69	68	65	78
Array Sate	u	n	i	v	B	D	ž	ˆ	Š	–	Ž	“	m	Ž	e
Dec	117	110	105	118	66	68	158	136	138	150	142	147	109	142	101

2. Proses enkripsi

Proses enkripsi *plaintext* dilakukan berdasarkan formula *one time pad*, yaitu menambahkan nilai karakter plain dengan nilai kuncinya kemudian dimodulus dengan 256.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Plain	B	U	D	I	D	A	R	M	A		M	E	D	A	N
Dec	66	85	68	73	68	65	82	77	65	32	77	69	68	65	78
Kunci	u	n	i	v	B	D	ž	ˆ	Š	–	Ž	“	m	Ž	e
Dec	117	110	105	118	66	68	158	136	138	150	142	147	109	142	101
(Pi+Ki) mod 256	183	195	173	191	134	133	240	213	203	182	219	216	177	207	179
Cipher	·	Ã		¿	†	…	ð	Õ	Ë	¶	Û	Ø	±	Ï	³

3. Proses dekripsi

Proses dekripsi dilakukan dengan mengurangi nilai decimal dari masing-masing karakter ciphertext dengan kuncinya masing-masing.

Proses pembangkitan kunci pada proses dekripsi, dilakukan dengan cara yang sama seperti yang dilakukan pada proses enkripsi.

Cipher	·	Ã		¿	†	…	ð	Õ	Ë	¶	Û	Ø	±	Ï	³
Dec	183	195	173	191	134	133	240	213	203	182	219	216	177	207	179
Kunci	u	n	i	v	B	D	ž	ˆ	Š	–	Ž	“	m	Ž	e
Dec	117	110	105	118	66	68	158	136	138	150	142	147	109	142	101
(Pi-Ki) mod 256	66	85	68	73	68	65	82	77	65	32	77	69	68	65	78
Plain	B	U	D	I	D	A	R	M	A		M	E	D	A	N

Dengan demikian, maka didapakan kembali plaintext seperti teks asli awal.

4.1. Kesimpulan

Berdasarkan penelitian dengan judul "Implementasi Lagged Fibonacci Generator pada Proses Pembangkitan Kunci Algoritma One Time Pad (OTP)", dapat disimpulkan bahwa:

1. Algoritma One Time Pad (OTP) merupakan salah satu algoritma kriptografi simetris yang dapat digunakan untuk mengamankan data atau informasi melalui proses enkripsi dan dekripsi menggunakan kunci. Keamanan OTP sangat bergantung pada kualitas kunci yang digunakan, terutama dalam hal kerahasiaan, keacakan, panjang kunci, dan penggunaan kunci yang hanya satu kali.
2. Implementasi algoritma Lagged Fibonacci Generator (LFG) pada proses pembangkitan kunci dapat digunakan untuk menghasilkan rangkaian bilangan yang dimanfaatkan sebagai dasar pembentukan kunci OTP. Pendekatan ini bertujuan menyediakan kunci dengan pola yang lebih sulit diprediksi dibandingkan penggunaan kunci yang dibentuk secara sederhana atau berulang.
4. Penggabungan algoritma LFG dengan OTP dapat membantu proses pembangkitan kunci menjadi lebih sistematis dan mendukung proses enkripsi serta dekripsi data. Dengan menggunakan kunci yang sama pada kedua proses, data yang telah dienkripsi dapat dikembalikan ke bentuk semula oleh penerima yang memiliki kunci yang sesuai.
5. Keamanan penerapan OTP tidak hanya ditentukan oleh algoritma pembangkitan kunci, tetapi juga oleh kualitas keacakan kunci, kerahasiaan kunci, panjang kunci yang sesuai dengan panjang pesan, serta penerapan prinsip penggunaan kunci satu kali. Oleh karena itu, penggunaan LFG perlu dievaluasi berdasarkan karakteristik keacakan dan ketahanannya terhadap prediksi.

4.2. Saran

Berdasarkan kesimpulan tersebut, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Pengujian kualitas kunci: Penelitian selanjutnya disarankan melakukan pengujian terhadap kualitas rangkaian kunci yang dihasilkan oleh LFG, termasuk distribusi nilai, pola pengulangan, dan tingkat prediktabilitasnya.
2. Peningkatan keamanan pembangkitan kunci: Penggunaan LFG standar perlu dipertimbangkan kembali untuk kebutuhan keamanan nyata. Sebaiknya digunakan pembangkit bilangan acak yang aman secara kriptografis (Cryptographically Secure Pseudorandom Number Generator atau CSPRNG), terutama jika kunci digunakan untuk melindungi data yang bersifat rahasia.
3. Pengujian proses enkripsi dan dekripsi: Penelitian selanjutnya dapat melakukan pengujian terhadap berbagai jenis data, seperti teks, dokumen, dan berkas digital, untuk memastikan bahwa proses enkripsi dan dekripsi berjalan dengan benar serta menghasilkan data yang identik dengan data asli setelah dekripsi.
4. Perbandingan algoritma: Disarankan melakukan perbandingan LFG dengan metode pembangkitan kunci lainnya untuk mengetahui perbedaan kualitas keacakan, waktu pembangkitan kunci, dan efisiensi proses pengamanan data.
5. Pengelolaan kunci: Sistem yang dikembangkan perlu memperhatikan mekanisme penyimpanan, distribusi, dan perlindungan kunci agar kunci tidak diketahui oleh pihak yang tidak berwenang. Kunci OTP juga harus memiliki panjang yang sama dengan pesan, benar-benar acak, dirahasiakan, dan tidak pernah digunakan kembali.

DAFTAR PUSTAKA

[1] D. A. Fauzan, A. Fathurrozi dan S. , "Perancangan Aplikasi Pengamanan Data Menggunakan Algoritma RSA (Rivest Shamir Adleman) dan AES (Advanced Encryption Standard) Berbasis Web," *Journal of Information and Information Security (JIFORTY)*, vol. 4, no. 1, pp. 91-104, 2023.

[2] R. Rivaldi dan S. , "Implementasi Pengamanan Data Arsitektur Menggunakan Metode Kriptografi Dengan Algoritma Rivest Code 4(RC4) PADA PT. Naviri Indah Cemerlang," *SKANIKA*, vol. 4, no. 2, pp. 139-143, 2021.

[3] M. Sianturi dan T. Zebua, "Modifikasi Pembangkit Kunci Algoritma Vigenere Cipher Berdasarkan Pembangkit Bilangan Acak CSPRNG Untuk Mengamankan Citra Digital," *JUSSI: Jurnal Sains Dan Teknologi Informasi*, vol. 2, no. 4, pp. 136-142, 2023.

[4] H. Alam, A. K. Habibi dan H. Widya, "Penggunaan Algoritma Vegenere Cipher dan One Time Pad untuk Keamanan Pesan Teks," dalam *SEMNASTEK - UISU*, Medan, 2022.

[5] I. Sunni, "Aplikasi Lagged Fibonacci Generator dan Bilangan Irrasional dalam Stream Cipher," Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung, 2011.

[6] Junianto dan T. , "Penerapan Algoritme Lagged Fibonacci Generator (LFG) Pada Edugame Puzzle Pengenalan Hewan Berbasis Android," Jurusan Teknik Informatika, STMIK MDP, Palembang.

[7] N. M. S. Sianturi, N. B. Nugroho dan W. R. Maya, "Implementasi Kriptografi Untuk Pengamanan Data Aset Perusahaan Pada PT.PLN (Persero) Dengan Menggunakan Metode Algoritma AES 192," *Jurnal CyberTech*, 2020.

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

[9] T. Zebua, "Penerapan Multiply With Carry Generator pada Proses Pembangkitan Kunci Algoritma Beaufort Cipher," *Journal of Information System Research (JOSH)*, vol. 4, no. 2, p. 607−613, 2023.

[10] M. Z. Solihin dan K. A. M., "Implementasi Kriptografi Menggunakan Metode Algoritma RSA Pada Aplikasi Pengamanan Data Berbasis Java Desktop Untuk Ud Tirta Soeper Teloer," dalam *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, Jakarta, 2022.

[11] M. K. Harahap dan N. Khairina, "Analisis Algoritma One Time Pad Dengan Algoritma Cipher Transposisi Sebagai Pengamanan Pesan Teks," *Sinkron Jurnal & PenelitianTeknik Informatika*, vol. 1, no. 2, pp. 58-62, 2017.

[12] S. A. Saputri dan N. , "Penerapan Algoritma One Time Pad Pada Pesan Rahasia Menggunakan QR Code Berbasis Android," *Jurnal FTIK*, vol. 1, no. 1, pp. 1193-1206.

[13] O. J. Harmaja dan A. Einstein, "Aplikasi Crazy Clock Game Dalam Bahasa Mandarin Bagi Anak-Anak Dengan Metode Lagged Fibonacci Generator," *Jurnal Penelitian Teknik Informatika Universitas Prima Indonesia (UNPRI) Medan (Jutikomp)*, vol. 1, no. 2, pp. 1-4, 2018.

[14] Wikipedia, "Wikipedia," 07 November 2022. [Online]. Available: https://en.wikipedia.org/wiki/Lagged_Fibonacci_generator. [Diakses 16 February 2024].