SNESTIK

Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika

https://ejurnal.itats.ac.id/snestik dan https://snestik.itats.ac.id

Informasi Pelaksanaan :
SNESTIK IV - Surabaya, 27 April 2024
Ruang Seminar Gedung A, Kampus Institut Teknologi Adhi Tama Surabaya

Informasi Artikel:
DOI : 10.31284/p.snestik.2024.5604

Prosiding ISSN 2775-5126

Fakultas Teknik Elektro dan Teknologi Informasi-Institut Teknologi Adhi Tama Surabaya
Gedung A-ITATS, Jl. Arief Rachman Hakim 100 Surabaya 60117 Telp. (031) 5945043
Email : snestik@itats.ac.id

Implementasi Multi Enkripsi Algoritma Stream Cipher dan Affine Cipher untuk Pengamanan Data Customer

Agam Jaya, Gusti Eka Yuliastuti*

Teknik Informatika, Institut Teknologi Adhi Tama Surabaya
e-mail: gustiekay@itats.ac.id

ABSTRACT

Encryption is one way to protect data from various fields and levels, but problems often occur, such as the length of execution time and the amount of resources needed, especially in strong encryption. To overcome this, the authors try to innovate by combining two encryption methods, namely stream cipher and affine cipher. The combination offers a more effective encryption process with a fairly fast execution time but is still difficult to penetrate because it uses two encryption algorithms. In the stream cipher method, the data will be randomized using an arbitrary key and then processed again using the affine cipher method, which relies on mathematical formulas to perform data scrambling operations, thereby increasing the complexity and security of encryption. This method was tested using the Avalanche effect, producing an average value of 39.4%. In conclusion, the multi-encryption application developed by the author has a security level that is not too strong but has a fairly fast execution time.

Keywords*: encryption, Stream Cipher, Affine Cipher*

ABSTRAK

Salah satu cara untuk melindungi data dari berbagai bidang dan tingkatan adalah dengan menggunakan enkripsi, namun seringkali ada masalah seperti lamanya durasi eksekusi dan besarnya sumber daya yang dibutuhkan terutama pada enkripsi yang kuat. Untuk mengatasi hal ini penulis mencoba membuat inovasi dengan menggabungkan dua metode enkripsi yaitu stream cipher dan affine cipher. Kombinasi ini menawarkan proses enkripsi yang lebih efektif dengan waktu eksekusi yang cukup cepat namun tetap sulit ditembus karena menggunakan dua algoritma enkripsi. Pada metode stream cipher data akan diacak

menggunakan kunci yang diubah-ubah kemudian diproses lagi menggunakan metode affine cipher yang mengandalkan rumus-rumus matematika dalam melakukan operasi pengacakan data sehingga meningkatkan kompleksitas dan keamanan enkripsi. Pengujian metode ini dibuktikan melalui pengujian Avalanche Effect yang menghasilkan nilai rata-rata dibawah 50% dimana ini menunjukkan bahwa penerapan multi enkripsi yang dikembangkan penulis memiliki tingkat keamanan yang tidak terlalu kuat tetapi memiliki execution time yang cukup cepat. Implementasi multi enkripsi ini telah diaplikasikan pada database MySQL yang berisi data pelanggan dari jayashoesnation dan berhasil dijalankan tanpa adanya kendala sedikitpun. Meskipun demikian masih terdapat kemungkinan dilakukan penyempurnaan pada metode stream cipher untuk meningkatkan efisiensi serta menangani potensi permasalahan yang muncul pada proses dekripsi.

Kata kunci : enkripsi, Stream Cipher, Affine Cipher

PENDAHULUAN

Masalah keamanan dan kerahasiaan data tersebut adalah satu aspek material Informasi (Munandar et al., n.d.). Sistem informasi tidak dapat dipisahkan dari keamanan data, beberapa informasi yang di maksud secara umum saja oleh karena itu bagi kelompok masyarakat tertentu keamanan data sangat diperlukan untuk mencegah Informasi pengungkapan kepada pihak ketiga cenderung acuh tak acuh Kebocoran dapat dihindari dengan merancang sistem keamanan yang berfungsi melindungi sistem informasi. Berbagai jenis umum masalah dengan hal-hal keamanan informasi meliputi penyadapan pasif, menguping aktif, penipuan, dll. Dari Pada kenyataannya, pencurian data bisa terjadi dalam bentuk pembacaan data file teks oleh pihak ketiga otorisasi, manipulasi data file teks, kerusakan data karena koneksi fisik yang buruk atau keamanan data (Putra and Ginting, 2017).

Di era digital saat ini, perusahaan-perusahaan memiliki akses terhadap banyak data pelanggan yang sangat sensitif dan rahasia. Data pelanggan seperti nama, alamat dan nomor telepon dapat menjadi target empuk bagi para peretas atau pihak yang tidak bertanggung jawab. Oleh karena itu, perlindungan data pelanggan menjadi sangat penting bagi perusahaan agar data tersebut tidak jatuh ke tangan yang salah dan digunakan untuk tujuan yang merugikan. Penerapan metode pengamanan seperti multi enkripsi algoritma Stream Cipher dan Affine Cipher dapat membantu perusahaan untuk melindungi data pelanggan dari serangan siber dan memastikan privasi pelanggan terjaga.

Masalah utama dalam proses enkripsi dan dekripsi adalah menentukan algoritma yang cocok dan efisien untuk digunakan dalam proses enkripsi dan dekripsi. Proses enkripsi membutuhkan algoritma yang dapat mengenkripsi data dengan aman, dan proses dekripsi membutuhkan algoritma yang dapat mendekripsi data dengan balik. Berikut adalah beberapa contoh metode enkripsi yang berbeda untuk membuat pesan rahasia. Ada enkripsi klasik, enkripsi modern, enkripsi dengan pertukaran kunci, enkripsi dengan fungsi hash, dan sebagainya. Beberapa algoritma berikut termasuk dalam kategori kriptografi tradisional, termasuk *Caesar, Affine, Monoalphabetic, Polyalphabetic, Vigenere, Beaufort , Playfair, Transposition, Rail Fence, Gronsfeld,* dan lain-lain.

Pada web jayashoesnation customer bisa melihat mengenai data-data mereka tanpa melalui sensor yaitu berupa proses enkripsi namun sebenarnya pada database sistem data-data tersebut tampak seperti huruf-huruf acak yang tidak berarti karena sudah melalui proses enkripsi. sehingga data pada database akan selalu terenkripsi tetapi pada halaman web data tersebut akan terlihat bentuk aslinya tetapi dengan syarat hanya user teridentifikasi saja yang bisa melihat datanya. Untuk proses enkripsi yang digunakan menggunakan gabungan metode enkripsi stream cipher dengan affine cipher dimana enkripsi stream cipher adalah Teknik enkripsi yang bekerja byte demi byte untuk mengubah teks biasa menjadi kode yang tidak dapat dibaca oleh siapapun tanpa menggunakan kunci yang sesuai. Kemudian enkripsi affine cipher adalah jenis enkripsi substitusi monoalfabetik dimana setiap huruf dalam alfabet dipetakan ke persamaannya dalam

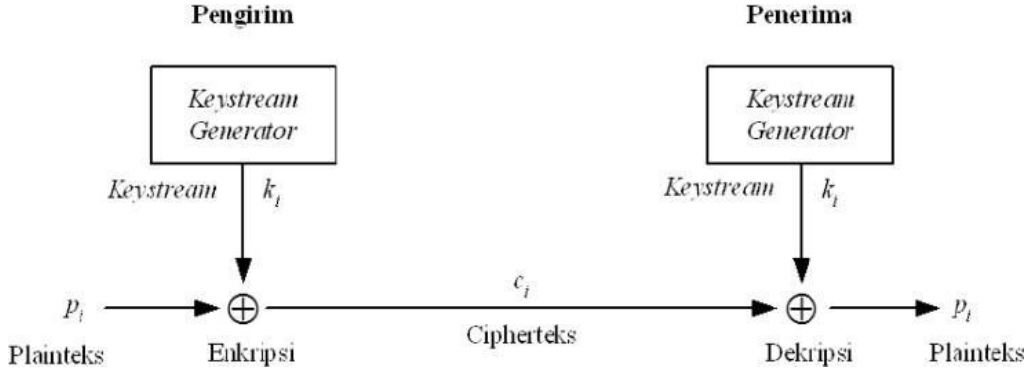
nilai angka lalu dienkripsi 4 menggunakan fungsi matematika sederhana dan dikonversi kembali menjadi huruf. Dengan menggunakan gabungan kedua jenis enkripsi ini, data di halaman web tersebut menjadi sangat sulit untuk ditembus.

TINJAUAN PUSTAKA

Stream Cipher

Stream cipher merupakan cipher yang hampir sama dengan Caesar Cipher (cipher yang menggeser urutan alphabet sehingga urutan alfabetnya berubah), tetapi cipher ini mempunyai kunci yang unik, yaitu menggunakan karakter sebelumnya sebagai kunci. fungsi maltemaltikal nyal terdiri dari:
C = (P+K) mod n
P = (C+K) mod n
C = Plaintext
K = kunci
N = Jumlah Karakter

Stream cipher adalah algoritma sandi yang mengenkripsikan data persatuan data, seperti bit, byte, nible atau perlima bit (saat data yang dienkripsi berupa data boudout). Setiap mengenkripsi satu kesatuan data digunakan kunci yang merupakan hasil pembangkitan dari kunci sebelumnya.



Gambar 1. Konsep stream Cipher

.

Pada Gambar 1 memperlihatkan konsep stream cipher, dimana pembangkit kunci menghasilkan bit kunci ki yang kemudian di-XOR-kan dengan plaintext pi menghasilkan bit ciphertext ci. Di sisi penerima pembangkit kunci yang sama akan meng-XOR-kan bit ciphertext ci dengan kunci ki yang sama untuk menghasilkan plaintext pi awal.

Affine Cipher

Affine Cipher adalah perluasan dari Caesar cipher. Affine Cipher mengacu pada algoritma klasik, yang merupakan algoritma pengkodean yang ada sebelum era digital modern. Semua orang yang Anda temui di rumah dapat melihat ke atas dan ke bawah. Cipher substitution adalah proses penggantian karakter dalam plaintext. Sedangkan cipher permutation adalah proses pertukaran huruf yang terdapat dalam sebuah string (Prasetyo and Ariyani, 2018).

Affine Cipher adalah teknologi kriptografi yang menukar karakter dengan cara monoalphabetic dimana setiap karakter dalam plaintext akan ditukar dengan karakter sesuai dengan rumus yang digunakan. Algoritma ini memiliki satu kunci berupa bilangan prima sebagai

penentu posisi karakter yang digunakan pada karakter yang digunakan. Selain itu, gunakan tabel secara manual atau buat secara otomatis untuk mengumpulkan algoritma pencarian kunci. Algoritma dalam memiliki tiga bagian utama, rahasia pembangkitan kunci bagian, proses kriptografer dan proses deskripsi. Pada penelitian ini, kami akan mencoba menerapkan algoritma ini pada proses enkripsi dan dekripsi pada teks. Diharapkan penggunaan algoritma ini akan meningkatkan keamanan cipher pengganti seperti Caesar dan Vigenere Cipher (Siahaan, 2017).

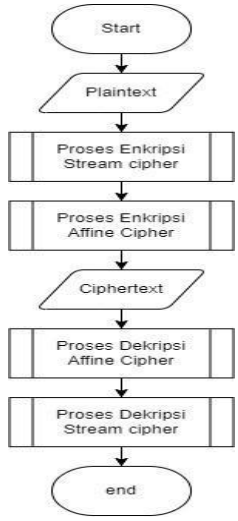
Avalanche effect

Avalanche Effect adalah rasio jumlah bit ciphertext yang berubah akibat perubahan plaintext terhadap jumlah total bit. Jika bit berubah setengah (50%) dari jumlah bit dalam ciphertext, menjadi sulit untuk dipecahkan. Pengujian ini menguji avalanche effect (transformasi plaintext) pada proses enkripsi, melihat transformasi bit-bit pada plaintext sehingga dapat diperoleh hasil persentase yang menentukan baik tidaknya suatu algoritma stream cipher(Aminudin et al., 2018). Hasil dari pengujian menggunakan avalanche effect bisa dibilang baik apabila persentase hasil pengujian tersebut memiliki nilai akhir diatas 50% sehingga apabila hasilnya masih dibawah 50% maka bisa dinyatakan bahwa metode enkripsi yang diuji menggunakan avalanche effect masih tergolong lemah (Muslih Muslih and Lekso Budi Handoko, 2022). pengujian AE dihitung menggunakan rumus AE dalam Persamaan 1.

$$Avalanche\ Effect = \frac{jumlah\ bit\ berubah}{jumlah\ bit\ total} * 100\% \quad (1)$$

METODE

Gambaran Umum Sistem



Gambar 2. Flowchart Gambaran Umum Sistem

Flowchart pada Gambar 2 menggambarkan alur proses enkripsi, yang diawali dengan input plaintext, kemudian dilanjutkan dengan multi enkripsi yang menghasilkan ciphertext. Dan mendekripsi kembali dengan multi deskripsi.

HASIL DAN PEMBAHASAN

Proses pengujian enkripsi dekripsi

1. Pengujian Enkripsi *Stream Cipher*

Bagian ini menjelaskan cara kerja stream cipher. Artinya ini menguraikan cara mengubah dari pesan asli menjadi rahasia atau terenkripsi dan cara mengubah kembali ke normal atau didekripsi.

Rumus dan contoh perhitungan dari metode kriptografi Stream Cipher :

Algoritma enkripsi stream cipher :

Ci = (Pi + Ki) mod 71

Keterangan :

Ci = Ciphertext halsil enkrip

Pi = Plalintext yalng alkaln dienkrip

Ki = Kunci untuk proses enkrip

Mod = Sisal Bagi/Modulus

Tabel 1 Index 71

Karakter	+	-	=	?	!	#	.	,	spasi
Urutan	0	1	2	3	4	5	6	7	8

Karakter	0	1	2	3	4	5	6	7	8	9
Urutan	9	10	11	12	13	14	15	16	17	18

Karakter	a	b	c	d	e	f	g	h	i	j	k	l	m
Urutan	19	20	21	22	23	24	25	26	27	28	29	30	31

Karakter	n	o	p	q	r	s	t	u	v	w	x	y	z
Urutan	32	33	34	35	36	37	38	39	40	41	42	43	44

Karakter	A	B	C	D	E	F	G	H	I	J	K	L	M
Urutan	45	46	47	48	49	50	51	52	53	54	55	56	57

Karakter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Urutan	58	59	60	61	62	63	64	65	66	67	68	69	70

Plalintext : PROPOSAL SKRIPSI

Kunci : KERJAlKAlN

Solusi : XOR menggunakan index 71

Tabel 2 Proses Enkripsi Stream Cipher

Plaintext	Key	Ciphertext
P = 60	K = 55	C1 = (P + K) mod 71 = (60 + 55) mod 71 = 115 mod 71 = 44(z)
R = 62	R = 62	C2 = (R + E) mod 71 = (62 + 49) mod 71 = 111 mod 71 = 40(v)
O = 59	R = 62	C3 = (O + R) mod 71 = (59 + 62) mod 71 = 121 mod 71 = 50(F)
P = 60	J = 54	C4 = (P + J) mod 71 = (60 + 54) mod 71 = 114 mod 71 = 43(y)
O = 59	A = 45	C5 = (O + A) mod 71 = (59 + 45) mod 71 = 104 mod 71 = 33(o)
S = 63	K = 55	C6 = (S + K) mod 71 = (63 + 55) mod 71 = 118 mod 71 = 47(C)
A = 45	A = 45	C7 = (A + A) mod 71 = (45 + 45) mod 71 = 90 mod 71 = 19(a)
L = 56	N = 58	C8 = (A + N) mod 71 = (56 + 58) mod 71 = 114 mod 71 = 43(y)
SPASI = 8	K = 55	C9 = (SPASI+K) mod 71 = (8 + 55) mod 71 = 63 mod 71 = 63(S)
S = 63	E = 49	C10 = (S + E) mod 71 = (63 + 49) mod 71 = 112 mod 71 = 41(w)
K = 55	R = 62	C11 = (K+R) mod 71 = (55 + 62) mod 71 = 117 mod 71 = 46(B)
R = 62	J = 54	C12 = (R + J) mod 71 = (62 + 54) mod 71 = 116 mod 71 = 45(A)
I = 53	A = 45	C13 = (I + A) mod 71 = (53 + 45) mod 71 = 98 mod 71 = 27 (i)
P = 60	K = 55	C14 = (P + K) mod 71 = (60 + 55) mod 71 = 115 mod 71 = 44(z)
S = 63	A = 45	C15 = (S + A) mod 71 = (63 + 45) mod 71 = 108 mod 71 = 37 (s)
I = 53	N = 58	C16 = (I + N) mod 71 = (53 + 58) mod 71 = 111 mod 71 = 40 = v
Ciphertext 1 = zvFyoCaySwBAizsv		

2. Pengujian Enkripsi Affine Cipher

Sebanyak 71 karakter digunakan. Dengan demikian, modulus yang digunakan dalam perhitungan akan menjadi 71. Modulus tersebut digunakan agar ciphertext yang berupa angka dapat diubah kembali menjadi karakter. Rumus dan contoh perhitungan dari metode kriptografi Affine Cipher:

Algoritma enkripsi Affine Cipher :

(a.Pi+b) mod 71

Tabel 3 Index 71

Karakter	0	1	2	3	4	5	6	7	8	9
Urutan	9	10	11	12	13	14	15	16	17	18

Karakter	a	b	c	d	e	f	g	h	i	j	k	l	m
Urutan	19	20	21	22	23	24	25	26	27	28	29	30	31

Karakter	n	o	p	q	r	s	t	u	v	w	x	y	z
Urutan	32	33	34	35	36	37	38	39	40	41	42	43	44

Karakter	A	B	C	D	E	F	G	H	I	J	K	L	M
Urutan	45	46	47	48	49	50	51	52	53	54	55	56	57

Karakter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Urutan	58	59	60	61	62	63	64	65	66	67	68	69	70

Key:

a = 3 (harus relatif prima dengan karakter 71)

b = 7 (harus berada dalam karakter 71)

Contoh:

plaintext = zvFyoCaySwBAizsv

a=3,b=7

Tabel 4 Proses Enkripsi Affine Cipher

Proses enkripsi:
C1 = (a.z+b) mod 71 = (3.44+7) mod 71 = 139 mod 71 = 68 (X)
C2 = (a.v+b) mod 71 = (3.40+7) mod 71 = 127 mod 71 = 56 (L)
C3 = (a.F+b) mod 71 = (3.50+7) mod 71 = 157 mod 71 = 15 (6)
C4 = (a.y+b) mod 71 = (3.43+7) mod 71 = 136 mod 71 = 65 (U)
C5 = (a.o+b) mod 71 = (3.33+7) mod 71 = 106 mod 71 = 35 (q)
C6 = (a.C+b) mod 71 = (3.47+7) mod 71 = 148 mod 71 = 6 (.)
C7 = (a.a+b) mod 71 = (3.19+7) mod 71 = 64 mod 71 = 64 (T)
C8 = (a.y+b) mod 71 = (3.43+7) mod 71 = 136 mod 71 = 65 (U)
C9 = (a.S+b) mod 71 = (3.63+7) mod 71 = 196 mod 71 = 54 (J)
C10 = (a.w+b) mod 71 = (3.41+7) mod 71 = 130 mod 71 = 59 (O)
C11 = (a.B+b) mod 71 = (3.46+7) mod 71 = 145 mod 71 = 3 (?)
C12 = (a.A+b) mod 71 = (3.45+7) mod 71 = 142 mod 71 = 0 (+)
C13 = (a.i+b) mod 71 = (3.27+7) mod 71 = 88 mod 71 = 17 (8)
C14 = (a.z+b) mod 71 = (3.44+7) mod 71 = 139 mod 71 = 68 (X)
C15 = (a.s+b) mod 71 = (3.37+7) mod 71 = 118 mod 71 = 47 (C)
C16 = (a.v+b) mod 71 = (3.40+7) mod 71 = 127 mod 71 = 56 (L)
Hasil Ciphertext 2 = XL6Uq.TUJO?+8XCL

3. Pengujian Dekripsi Affine Cipher

Pengujian ini mengembalikan karakter dari hasil enkripsi affine cipher. Dibawah ini adalah hasil pengujian saya menggunakan metode Affine Cipher.

Algoritma dekripsi Affine cipher :

a-1(Ci-b) mod 71

a-1 = nilai dapat dicari jika bilangan berapa Jika dikalikan dengan a, lalu di-modulus-kan dengan 71 sehingga hasilnya menjadi 1

Ciphertext : XL6Uq.TUJO?+8XCL

Kunci a = 3

Kunci b = 7

Tabel 5 Index 71

Karakter	0	1	2	3	4	5	6	7	8	9
Urutan	9	10	11	12	13	14	15	16	17	18

Karakter	a	b	c	d	e	f	g	h	i	j	k	l	m
Urutan	19	20	21	22	23	24	25	26	27	28	29	30	31

Karakter	n	o	p	q	r	s	t	u	v	w	x	y	z
Urutan	32	33	34	35	36	37	38	39	40	41	42	43	44

Karakter	A	B	C	D	E	F	G	H	I	J	K	L	M
Urutan	45	46	47	48	49	50	51	52	53	54	55	56	57

Karakter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Urutan	58	59	60	61	62	63	64	65	66	67	68	69	70

Mencari invers modulo :

a. a-1 = 1 (mod 71)

3.24 = 72 mod 71

= 1

Tabel 6 Proses dekripsi Affine Cipher

Proses dekripsi:
P1 = a^{-1} (X - b) mod 71 = 24(68-7) mod 71 = 1464 mod 71= 44 (z)
P2 = a^{-1} (L - b) mod 71 = 24(56-7) mod 71 = 1176 mod 71 = 40 (v)
P3 = a^{-1} (6 - b) mod 71 = 24(15-7) mod 71 = 192 mod 71 = 50 (F)
P4 = a^{-1} (U - b) mod 71 = 24(65-7) mod 71 = 1392 mod 71 = 43 (y)
P5 = a^{-1} (q - b) mod 71 = 24(35-7) mod 71 = 672 mod 71 = 33 (o)
P6 = a^{-1} ((.) - b) mod 71 = 24(6-7) mod 71 = -24 mod 71 = 47 (C)
P7 = a^{-1} (T - b) mod 71 = 24(64-7) mod 71 = 1368 mod 71 = 19 (a)

P8 = a^{-1} (U - b) mod 71 = 24(65-7) mod 71 = 1392 mod 71 = 43 (y)
P9 = a^{-1} (J - b) mod 71 = 24(54-7) mod 71 = 1128 mod 71 = 63 (S)
P10 = a^{-1} (O - b) mod 71 = 24(59-7) mod 71 = 1248 mod 71 = 41 (w)
P11 = a-1 ((?) - b) mod 71 = 24(3-7) mod 71 = -96 mod 71 = 3 (?)
P12 = a^{-1} ((+) - b) mod 71 = 24(0-7) mod 71 = -168 mod 71 = 45 (A)
P13 = a^{-1} (8 - b) mod 71 = 24(17-7) mod 71 = 240 mod 71 = 27(i)
P14 = a^{-1} (X - b) mod 71 = 24(68-7) mod 71 = 1464 mod 71 = 44 (z)
P15 = a^{-1} (C- b) mod 71 = 24(47-7) mod 71 = -24 mod 71 = 37 (s)
P16 = a^{-1} (L - b) mod 71 = 24(56-7) mod 71 = 408 mod 71 = 40 (v)
Hasil Plaintext 1 = zvFyoCaySwBAizsv

4. Pengujian Dekripsi Stream Cipher

Pengujian ini menggunakan lima contoh hasil dekripsi. Salah satunya menggunakan plaintext "PROPOSAL SKRIPSI" dan kunci "KERJAKAN". Di bawah ini adalah hasil pengujian saya menggunakan metode Stream Cipher.

Algoritma dekripsi stream cipher:

Pi = (Ci-Ki) mod 26

Keterangan:

Pi = Plaintext yang akan dienkrip

Ci = Ciphertext hasil enkrip

Ki = Kunci untuk proses enkrip

Mod = Sisal Bagi/Modulus

Tabel 7 Index 71

Karakter	0	1	2	3	4	5	6	7	8	9
Urutan	9	10	11	12	13	14	15	16	17	18

Karakter	a	b	c	d	e	f	g	h	i	j	k	l	m
Urutan	19	20	21	22	23	24	25	26	27	28	29	30	31

Karakter	n	o	p	q	r	s	t	u	v	w	x	y	z
Urutan	32	33	34	35	36	37	38	39	40	41	42	43	44

Karakter	A	B	C	D	E	F	G	H	I	J	K	L	M
Urutan	45	46	47	48	49	50	51	52	53	54	55	56	57

Karakter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Urutan	58	59	60	61	62	63	64	65	66	67	68	69	70

Ciphertext: zvFyoCaySwBAizsv

Kunci: KERJAKAN KERJAKA

Plaintext: PROPOSAL SKRIPSI

Tabel 8 Proses Dekripsi Stream Cipher

Ciphertext	Key	Plaintext
z = 44	K = 55	C1 = (z+ K) mod 71 = (44 - 55) mod 71 = -11 mod 71 = 60(P)
v = 40	R = 62	C2 = (v + E) mod 71 = (40 - 49) mod 71 = -9 mod 71 = 62(R)
F = 50	R = 62	C3 = (F + R) mod 71 = (50 - 62) mod 71 = -12 mod 71 = 59(O)
y = 43	J = 54	C4 = (y + J) mod 71 = (43 - 54) mod 71 = -11 mod 71 = 60(P)
o = 33	A = 45	C5 = (o + A) mod 71 = (33 - 45) mod 71 = -12 mod 71 = 59(O)
C = 47	K = 55	C6 = (C + K) mod 71 = (47 - 55) mod 71 = -8 mod 71 = 63(S)
a = 19	A = 45	C7 = (a + A) mod 71 = (19 - 45) mod 71 = -26 mod 71 = 45(A)
y = 43	N = 58	C8 = (y + N) mod 71 = (43 - 58) mod 71 = -15 mod 71 = 56(L)
S = 63	K = 55	C9 = (S + K) mod 71 = (63 - 55) mod 71 = 8 mod 71 = 8(SPASI)
w = 41	E = 49	C10 = (w + E) mod 71 = (41 - 49) mod 71 = -8 mod 71= 63(S)
B = 46	R = 62	C11 = (B + R) mod 71 = (46 - 62) mod 71 = -16 mod 71= 55(K)
A = 45	J = 54	C12 = (A + J) mod 71 = (45 - 54) mod 71 = -9 mod 71 = 62(R)
i = 27	A = 45	C13 = (i + A) mod 71 = (27 - 45) mod 71 = -18 mod 71 = 53 (I)
z = 44	K = 55	C14 = (z + K) mod 71 = (44 - 55) mod 71 = -11 mod 71 = 60(P)
s = 37	A = 45	C15 = (s + A) mod 71 = (37 - 45) mod 71 = -8 mod 71 = 63(S)
v = 40	N = 58	C16 = (v + N) mod 71 = (40 - 58) mod 71 = -18 mod 71 = 53(I)
Hasil Plaintext 2 = PROPOSAL SKRIPSI		

Pengujian Avalanche Effect

Perubahan dalam satu bit dari plaintext atau satu bit kunci harus menghasilkan perubahan dalam banyak bit teks kata sandi disebut sebagai Avalanche Effect (AE). Kuncinya akan menghasilkan perubahan pada banyak bit dalam ciphertext. Nilai ideal untuk AE dikategorikan baik jika perubahan dalam bit bernilai sebesar 45% - 60% (50% adalah hasil yang sangat baik). Karena perubahan ini berarti membuat perbedaan yang cukup sulit bagi cryptanalyst untuk melakukan serangan (Irawan & Rachmawanto, 2022).

Plaintext
PROPOSAL SKRIPSI
Kunci
KERJAKAN
Cipher text
XL6Uq.TUJO?+8XCL
Perubahan bit
53
Av alanche Effect
41%

Plalintext : PROPOSAL SKRIPSI

Binalry 01010000 01010010 01001111 01010000 01001111 01010011

01000001 01001100 00100000 01010011 01001011 01010010 01001001

01010000 01010011 01001001

Kunci : KERJAKAN

Ciphertext : XL6Uq.TUJO?+8XCL

Binary 01011000 01001100 00110110 01010101 01110001 00101110

01010100 01010101 01001010 01001111 00111111 00101011 00111000

01011000 01000011 01001100

Rumus persamaan dalam pengujian avalanche effect dapat dilihat pada daftar persamaan 2:

$$Avalanche\ effect = \frac{53}{128} * 100\% = 41 \quad (2)$$

Analisis hasil pengujian

Dari beberapa pengujian diatas ditemukan bahwa hasil dari pengujian avalanche effect gabungan metode enkripsi affine cipher dengan stream cipher persentasenya 39.46%. Dimana apabila hasil pengujian avalanche effect masih berada dibawah 50% maka bisa disimpulkan

bahwa metode enkripsi yang diuji menggunakan avalanche effect masih tergolong lemah tetapi metode enkripsi gabungan ini masih lebih kuat daripada metode enkripsi Affine Cipher atau Stream Cipher secara individual. penulis memiliki hipotesis bahwa penyebab kecilnya hasil pengujian avalanche effect gabungan metode affine cipher dengan stream cipher bersumber dari implementasi yang kurang tepat dari masing-masing metode yaitu Affine Cipher dan Stream Cipher contohnya adalah jika kunci yang sama digunakan lebih dari sekali dalam Stream Cipher atau jika pola dari Affine Cipher bisa ditebak sehingga pengacakan data tidak bisa tersebar rata dan mengakibatkan persentase dari hasil pengujian avalanche effect berada dibawah 50%.

KESIMPULAN

Untuk menentukan seberapa kuat kunci enkripsi yang telah ditentukan dengan metode multi-enkripsi Stream Cipher dan Affine Cipher dibuktikan melalui pengujian Avalanche Effect yang menghasilkan nilai rata-rata 43.88% dimana ini menunjukkan bahwa penerapan multi enkripsi yang dikembangkan penulis memiliki tingkat keamanan yang tidak terlalu kuat tetapi memiliki execution time dengan rata-rata 0.14037(S).

DAFTAR PUSTAKA

[1] Aminudin, A., Helmi, A.F., Arifianto, S., 2018. Analisa Kombinasi Algoritma Merkle-Hellman Knapscak dan Logaritma Diskrit pada Aplikasi Chat. J. Teknol. Inf. Dan Ilmu Komput. 5, 325. https://doi.org/10.25126/jtiik.201853844

[2] Hidayatuloh, K., Yustantina, Y., Kusmadi, K., 2021. Perbandingan Metode Stream Cipher dan Hill Cipher Dalam Keamanan Data. Infotronik J. Teknol. Inf. Dan Elektron. 6, 27. https://doi.org/10.32897/infotronik.2021.6.1.647

[3] Irawan, C., Rachmawanto, E.H., 2022. Implementasi Kriptografi dengan Menggunakan Algoritma Arnold's Cat Map dan Henon Map. J. Masy. Inform. 13, 15–32. https://doi.org/10.14710/jmasif.13.1.43312

[4] Kabeaken, S., Saputra, I., 2020. Perancangan Aplikasi Enkripsi Pada File Teks Menggunakan Algoritma Spritz. JURIKOM J. Ris. Komput. 7, 353. https://doi.org/10.30865/jurikom.v7i3.2125

[5] Munandar, A., Rosnelly, R., Sianturi, C.J.M., n.d. Rancang Bangun Aplikasi Keamanan Data Teks Mengunakan Algoritma Stream Cipher 10.

[6] Muslih Muslih, Lekso Budi Handoko, 2022. Pengujian Avalanche Effect pada Kiptogtafi Teks Menggunakan Autokey Cipher. Semin. Nas. Teknol. Dan Multidisiplin Ilmu SEMNASTEKMU 2, 127–134. https://doi.org/10.51903/semnastekmu.v2i1.162

[7] Noviani, A.L., n.d. Perancangan Perangkat Lunak Kriptografi Menggunakan Gronsfeld Cipher, Vernam Cipher dan Ron Code 4 Stream Cipher.

[8] Prasetyo, A.I., Ariyani, P.F., 2018. Keamanan Database Nota Penjualan dengan Algoritma Affine Cipher dan Wake Berbasis Web 1.