Playfair Cipher Block Chaining dan *Elliptic Curve* Untuk Pengamanan Pertukaran Data Rest

Playfair Cipher Block Chaining and Elliptic Curve for Data Rest Exchange Safeguard

Usman Zamari[*1], Retantyo Wardoyo [2]
[1]Prodi S2 Ilmu Komputer; FMIPA UGM, Yogyakarta
[2]Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]usman.zamari@mail.ugm.ac.id**, **[2]rw@ugm.ac.id**

Abstrak

REpresentational State Transfer (REST) merupakan representasi dari pertukaran data online serta memiliki kecepatan dan respon yang baik. Kemudahan pertukaran komunikasi tersebut jika tidak dilakukan pengamanan bisa mengakibatkan informasi terbaca oleh pihak yang tidak berwenang.

Kriptografi merupakan seni dan ilmu yang digunakan untuk menjaga atau mengamankan data. Memodifikasi Algoritme kriptografi dalam bentuk lain dari metode dasar, akan mendapatkan berbagai variasi metode tersebut. Algoritme Playfair Cipher Block Chaining (Playfair CBC) digunakan untuk mengamankan data request dengan menerapkan digraph terenkripsi untuk membangun kunci kembali. Kelemahan algoritme ini terletak pada pengamanan kuncinya sehingga diperlukan algoritme lain untuk melakukannya. Algoritme Elliptic Curve digunakan untuk menyelesaikan masalah pada pengamanan kunci tersebut.

Pengujian dilakukan dengan menggunakan 10 data yang bervariasi. Pengujian menunjukkan bahwa kunci dapat digunakan untuk melakukan enkripsi dan dekripsi. Melakukan analisis brute force pada algoritme yang digunakan. Analisis menunjukkan algoritme yang digunakan memiliki tingkat keamanan yang baik.

Kata kunci: *REST, Kriptografi, Playfair CBC, Elliptic Curve*

Abstract

Representational State Transfer (REST) is a representation for online data exchange and have good speed and response. The ease for communication exchange if not safeguard can result in unauthorized information is read by unauthorized persons.

Cryptography is an art and science used to keep or secure data. Modifying the cryptography algorithm in another form from the basic method will get a variety of its method. Playfair Cipher Block Chaining (Playfair CBC) algorithm is used to secure request data by applying encrypted digraph to build lock again. The weakness of this algorithm lies in securing the key so that another algorithm is needed to do so. The Elliptic Curve algorithm is used to solve the security lock problem.

Testing had done by using 10 different data. Testing indicates that the key can be used to perform encryption and decryption. Perform brute force analysis on the algorithm used. The analysis shows that the algorithm used has a good level of security.

Keywords: *REST, Cryptography, Playfair CBC, Elliptic Curve*

1. Pendahuluan

Perkembangan teknologi yang terjadi sekarang ini menurut Kopack dan Potts (2003), aplikasi perangkat lunak di dunia memiliki potensi untuk saling berkomunikasi. Komunikasi ini tidak dibatasi oleh lokasi, platform, sistem operasi, bahasa dan protokol. Menurut Al-Zoubi dan Wainer (2009), REpresentational State Transfer (REST)

menyediakan arsitektur dan desain yang mudah dipahami dan bisa diimplementasikan untuk pertukaran data atau komunikasi. Kemudahan pertukaran komunikasi tersebut jika tidak dilakukan pengamanan bisa mengakibatkan informasi terbaca oleh pihak yang tidak berwenang.

Keamanan data (data security) merupakan aspek penting dalam aplikasi online. Menurut Stallings (2003) sebagian persyaratan utama layanan keamanan mencakup tentang confidentiality, authentication, non-repudation dan integrity. Menurut Rahardjo, (2002), masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Masalah keamanan data seringkali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting.

Kriptografi merupakan seni dan ilmu yang digunakan untuk menjaga atau mengamankan data. Menurut Schneier (1996), dalam mengimplementasikan suatu algoritme kriptografi menjadi sangat penting untuk meningkatkan pengamanan data atau sistem. Memodifikasi Algoritme kriptografi dalam bentuk lain dari metode dasar, akan mendapatkan berbagai variasi metode tersebut. Dalam mempelajari kriptografi, konsep ini sangat penting untuk dipahami, karena pemanfaatan dan implementasi di lapangan akan sangat bergantung pada pemahaman tersebut. Dalam cryptanalysis, modifikasi algoritme kriptografi akan dapat meningkatkan satu kekuatan dari metode tersebut.

Algoritme Playfair Cipher Block Chaining (Playfair CBC) menurut Goyal dkk. (2015) menggunakan digraph terenkripsi untuk membangun kunci kembali. Dengan perubahan tersebut membuat perubahan ciphertext pada proses selanjutnya. Kelemahan algoritme Playfair Cipher Block Chaining pada proses enkripsi dan dekripsi menggunakan kunci yang sama atau kunci simetri. Menurut Munir (2006) kekuatan kriptografi simetri ditentukan oleh kunci yang digunakan. Pengamanan kunci perlu dilakukan agar tidak diketahuai orang lain. Agoritme Elliptic Curve (ECC) menghasilkan keamanan yang tinggi karena variasi pada perhitungannya dan memiliki ukuran kunci yang lebih kecil dibanding dengan algoritme kunci publik lainnya (RSA, ElGamal).

2. Metode Penelitian

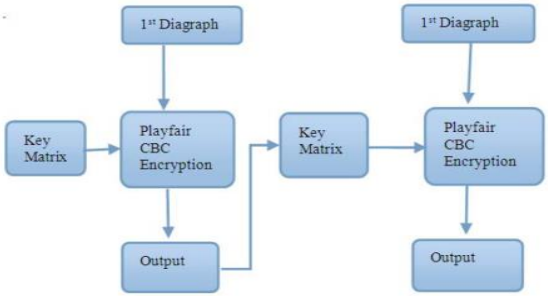
2.1 Analisis Sistem

Sistem REST *Server* yang telah dilengkapi dengan metode algoritme *Playfair* CBC dan *Elliptic Curve* ini merupakan perangkat lunak yang dapat digunakan untuk melakukan pertukaran data yang aman. Metode algoritme *Playfair* CBC dan *Elliptic Curve* digunakan untuk menangani masalah keamanan data ketika melakukan komunikasi/pertukaran data. *Elliptic Curve* digunakan untuk melindungi hak akses *Client Server* dan kunci *Playfair* CBC, sedangkan *Playfair* CBC digunakan untuk melindungi *plaintext* yang akan dikirimkan ke *Client Server*. Analisis kebutuhan sistem yang akan dibangun adalah sebagai berikut:

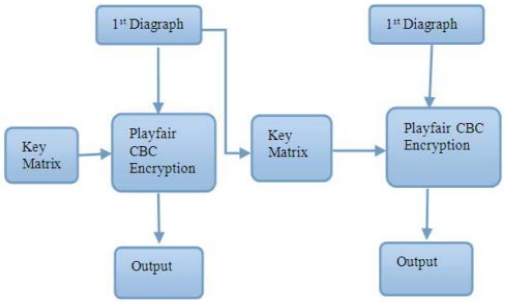
1 Sistem REST *Server* dapat digunakan oleh *Client Server* yang sudah mendaftarkan *token*.
2 Sistem REST *Server* dapat digunakan untuk melakukan pertukaran data dengan *Client Server*.
3 Untuk melakukan pertukaran data, *Client Server* harus melewati proses *authentication token* di REST *Server*.

Teori *Playfair* CBC

Playfair CBC menurut Goyal dkk. (2015) menggunakan *output* untuk membangun kembali kunci *matrix*nya. Untuk mengisi kunci *matrix*, jika terdapat huruf *alphabet* yang sama maka digunakan satu huruf saja. Di dalam *Playfair cipher*, *alphabet* kunci *matrix* disusun dalam *matrix* 5×5 dan juga dapat diimplementasikan dengan ukuran *matrix* berapapun. *Playfair* CBC menggunakan *output* untuk membangun kembali kunci *matrix*nya (Goyal dkk., 2015). Proses enkripsi dengan *Playfair* CBC sesuai Gambar 1 dan proses dekripsi sesuai dengan Gambar 2.



Gambar 1 Proses enkripsi *Playfair* CBC (Goyal dkk., 2015)



Gambar 2 Proses Dekripsi *Playfair* CBC (Goyal dkk., 2015)

Teori Elliptic Curve

Elliptic Curve (Kurva Eliptik) adalah sistem yang sama dengan RSA. *Elliptic Curve* digunakan untuk mengatasi masalah kunci pada RSA, karena RSA menyebabkan proses yang lebih lama jika melakukan transaksi dalam jumlah yang besar (Stallings, 2014). *Elliptic Curve* digunakan untuk menyelesaikan persamaan tertentu di dalam dua variabel. *Elliptic Curve* ada dua yaitu:

Singular jika $4a^3 + 27b^2 = 0$.

Non singular $4a^3 + 27b^2 \neq 0$.

Definisi 1, Misal $a, b \in \mathbb{R}$ menjadi konstanta $4a^3 + 27b^2 \neq 0$. Kurva eliptik non singular adalah himpunan E untuk menyelesaikan masalah $(x, y) \in \mathbb{R} \times \mathbb{R}$ dengan persamaan (Stinson, 2006)

$$y^2 = x^3 + ax + b \qquad (1)$$

Kurva Eliptik Modulo Prima

Misal p sebagai bilangan prima lebih dari 3 ($p > 3$). Kurva eliptik diatas $\mathbb{Z}_p$ dapat ditentukan sebagai bilangan real yaitu semua operasi di atas $\mathbb{R}$ diganti dengan operasi di $\mathbb{Z}_p$.

Definisi 2, Misal $p > 3$ sebagai bilangan prima. Kurva eliptik $y^2 = x^3 + ax + b$ diatas $\mathbb{Z}_p$ adalah himpunan pada solusi $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ yang sesuai (Stinson, 2006)

$$y^2 = x^3 + ax + b \ (mod\ p) \qquad (2)$$

dimana $a, b \in \mathbb{Z}_p$ harus memenuhi

$$4a^3 + 27b^2 \neq 0 \ (mod\ p) \qquad (3)$$

Titik E pada kurva eliptik diatas $\mathbb{Z}_p$ sesuai persamaan (3) adalah (Stalling, 2014):

$$y^2\ mod\ p = x^3 + ax + b \ (mod\ p) \qquad (4)$$

Penambahan operasi pada E ditentukan sebagai berikut (Stinson, 2006):

$$P = (x_1, y_1)\ dan\ Q = (x_2, y_2)$$

jika $x_2 = x_1\ dan\ y_2 = -y_1, maka\ P + Q = \theta$

jika tidak, maka $P + Q = (x_3, y_3)$, *dimana*

$$x_3 = \lambda^2 - x_1 - x_2\ mod\ p \quad (5)$$

$$y_3 = \lambda(x_1 - x_3) - y_1\ mod\ p \quad (6)$$

$$Dan \qquad \lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1}\ mod\ p,\ jika\ P \neq Q & (7) \\ (3{x_1}^2 + a)(2y_1)^{-1}\ mod\ P,\ jika\ P = Q & (8) \end{cases}$$

sehingga $P + O = O + P = P$ *untuk semua* $P \in E$.

Teori Representational State Transfer

REpresentational State Transfer (REST) merupakan standar arsitektur untuk mempermudah dalam berkomunikasi data berbasis web. Pada umumnya menggunakan *Hypertext Transfer Protocol* (HTTP) untuk komunikasi datanya. Metode umum yang digunakan dalam HTTP dengan REST adalah:

GET : Mengambil data dari *resource*
PUT : Mengubah data ke *resource*
POST : Mengirim data ke *resource*
DELETE : Menghaspus data dari *resource*

Pada umunya sebagian kode *response* dari sebuah *server* dengan REST adalah:

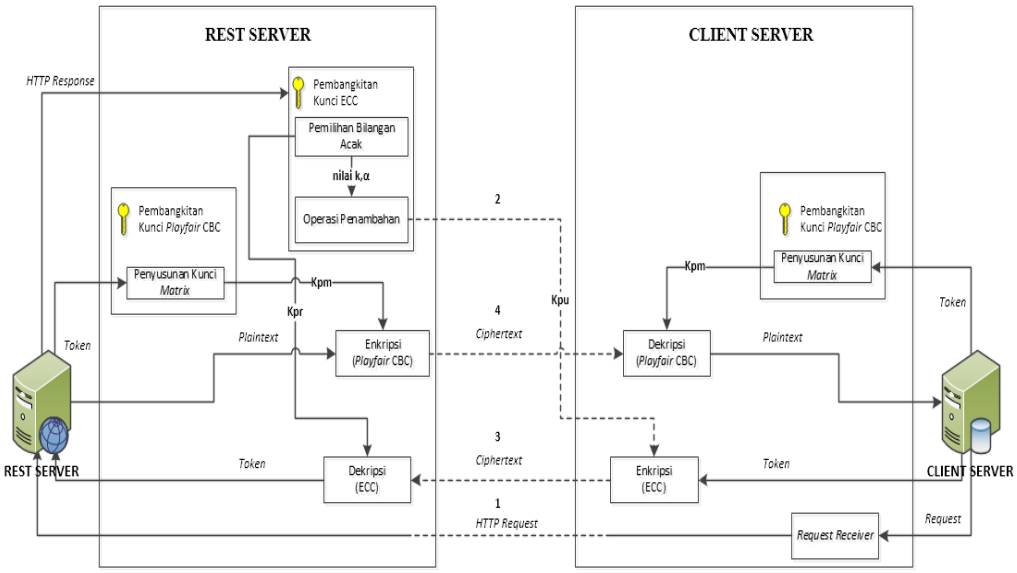
201 ok, ketika *request* dari *Client* sukses

400 *Bad Request*, ketika *Client* mengirimkan permintaan yang salah

500*Internal Server Error*, ketika *server* mengalami kegagalan

2.2 Perancangan Sistem

Secara garis besar terdapat tiga proses pada penelitian ini yaitu proses pembangkitan kunci, proses enkripsi dan proses dekripsi. Gambaran umum sistem diperlihatkan pada Gambar 3.



Gambar 3 Rancangan sistem

Dalam melakukan komunikasi, pihak *Client Server* terlebih dahulu harus mendaftarkan *token* di aplikasi REST *Server*. *Token* tersebut akan digunakan sebagai

pembangkit kunci *matrix Playfair* CBC, setelah itu *Client Server* mengirimkan permintaan dengan metode HTTP, kemudian REST *Server* meresponnya dengan membangkitkan sepasang kunci asimetri *Elliptic Curve* (kunci privat dan kunci publik). Kunci privat bersifat rahasia dan kunci publik tidak rahasia, setelah itu kunci publik tersebut dikirim ke *Client Server* untuk mengenkripsi *token*. *Client Server* kemudian memilih bilangan acak *integer* (k) disamping menggunakan kunci publik (Kpu) untuk mengenkripsi *token*. Hasil enkripsi menghasilkan *ciphertext* yang akan dikirimkan ke REST *Server*. Oleh REST *Server ciphertext* didekripsi menggunakan kunci privat (Kpr) yang telah dibangkitkan sebelumnya sehingga menghasilkan *plaintext*.

Pihak REST *Server* mengenkripsi pesan/data yang diminta *Client Server* dengan menggunakan *Playfair* CBC. Enkripsi tersebut dilakukan setelah melakukan proses *authentication token*. Token tersebut bersifat rahasia, karena digunakan untuk proses enkripi dan dekripsi. Hasil proses enkripsi oleh REST *Server* menghasilkan *ciphertext* yang kemudian dikirimkan ke *Client Server*. Selanjutnya *ciphertext* dekripsi oleh *Client Server* dengan menggunakan kunci *token* yang sama. Hasil proses dekripsi tersebut menghasilkan *plaintext*.

Membangkitkan kunci simetri (Playfair CBC)

Token yang sama akan direduksi dari *token* yang berulang, karena hanya 1 (satu) karakter saja yang digunakan yaitu karakter yang terletak di awal. *Token* yang direduksi akan digunakan untuk enkripsi dan dekripsi. Tabel 1 merupakan cara mereduksi *token* dimana kunci MIPAUGM sebagai masukannya.

Tabel 1 Mereduksi kunci MIPAUGM

M		I	P	A	U	G	1	2
3		$	!	“	#	%	&	‘
(		)	*	+	,	-	.	/
0		4	5	6	7	8	9	:
;		<	=	>	?	@	B	C
D		E	F	H	J	K	L	N
O		Q	R	S	T	V	W	X
Y		Z	[	{	]	^	_	`

Membangkitkan kunci asimetri (Elliptic Curve)

Pembangkitan kunci asimetri (*Elliptic Curve*) bertujuan untuk mendapatkan kunci publik dan kunci privat. Untuk membangkitkan kunci tersebut tergantung pada $(mod\ p)$, artinya semakin besar nilai $(mod\ p)$ akan membuat semakin banyak variasi pemilihan nilai a sebagai pembangkit grup kurva. Metode untuk membangkitkan kunci *Elliptic Curve* adalah sebagai berikut :

Persamaan kurva yang dipakai adalah

$x^3 + x + 6\ atas\ \mathbb{Z}_{131}$

Menguji persamaan kurva dimana $a, b \in \mathbb{Z}_p$ harus memenuhi

$4a^3 + 27b^2 \neq 0\ (mod\ p)$

$= 4(1)^3 + 27 \times (6^2) \neq 0\ (mod\ 131)$

$= 59 \neq 0$ *(persamaan kurva memenuhi)*

Menentukan elemen kurva eliptik berdasarkan persamaan atas $\mathbb{Z}_p$, sesuai dengan persamaan (4) dengan inputan nilai $x = \{0,1,2,3, \dots, p-1\}$. Jika persamaan tersebut

memenuhi nantinya akan digunakan sebagai variasi nilai a untuk pembangkit grup kurva dan nilai a tersebut dipilih secara random. Misalnya, inputan yang dimasukan setelah perhitungan nilai 0 dan 1 adalah $x = 2$ dan $y = 1,4$ dan 127 sehingga nilai $x(2)$ pada persamaan $x^3 + x + 6 \, mod \, p$ menghasilkan nilai 16 dan termasuk dalam kuadratik residu 131. Nilai $x(2)$ pada persamaan $y^2 \, mod \, 131$ dengan nilai yang sama pada $y_1(4)$ & $y_2(127)$.

Memilih bilangan acak *integer* dari selang $[1, p-1]$ yang akan dipakai sebagai kunci privat. Pembangkit grup kurva yang dipilih a adalah $(2,4)$ dan memilih kunci privat 4, sehingga pembangkitan K_{Publik} tersebut sebagai berikut :

$$1a = a = (2,4)$$
$$2a = a + a = (2,4) + (2,4)$$
$$2a = (58,36)$$
$$3a = 2a + a = (58,36) + (2,4)$$
$$3a = (74,11)$$
$$4a = 3a + a = (74,11) + (2,4)$$
$$4a = (5,23) \; menjadi \; K_{Publik}$$

Proses enkripsi dan dekripsi Playfair CBC

Plaintext disusun ke dalam *digraph* sebelum dilakukan enkripsi. Proses enkripsi *Playfair* CBC pada awalnya menggunakan kunci yang telah disusun kedalam *matrix* 8×8. Selanjutnya hasil *output* dari setiap proses enkripsi digunakan untuk membangkitkan kunci *matrix* kembali. Proses tersebut akan digunakan sampai enkripsi *digraph* terakhir. Adapun langkah-langkah proses enkripsi dan dekripsi *Playfair* CBC sebagai berikut:

1. Enkripsi

Melakukan enkripsi *plaintext* **MIPA** dengan kunci **MIPAUGM**, maka enkripsinya adalah sebagai berikut:Membuat MIPA dalam pasangan huruf *digraph* MI PA. Memasukkan kunci MIPAUGM ke *matrix* dan mengenkripsi MI

Tabel 2 Penambahan kunci MIPAUGM (Enkripsi)

M	I	P	A	U	G	1	2
3	$	!	“	#	%	&	‘
(	)	*	+	,	-	.	/
0	4	5	6	7	8	9	:
;	<	=	>	?	@	B	C
D	E	F	H	J	K	L	N
O	Q	R	S	T	V	W	X
Y	Z	[	\	]	^	_	`

Ciphertext yang dihasilkan adalah IP. Memasukkan kunci IP ke *matrix* dan mengenkripsi PA. *Ciphertext* yang dihasilkan adalah MU. Jadi *ciphertext* yang dihasilkan adalah **IPMU**

Tabel 3 Penambahan kunci IP (Enkripsi)

I	P	M	A	U	G	1	2
3	$	!	“	#	%	&	‘
(	)	*	+	,	-	.	/
0	4	5	6	7	8	9	:
;	<	=	>	?	@	B	C
D	E	F	H	J	K	L	N
O	Q	R	S	T	V	W	X
Y	Z	[	\	]	^	_	`

2. Dekripsi

Melakukan dekripsi IPMU dengan kunci **MIPAUGM**, maka dekripsinya adalah: (1) Membuat IPMU dalam pasangan huruf *digraph* IP MU (2) Memasukkan kunci MIPAUGM ke *matrix* dan mendekripsi IP.

Tabel 4 Penambahan kunci MIPAUGM (Dekripsi)

M	I	P	A	U	G	1	2
3	$	!	“	#	%	&	‘
(	)	*	+	,	-	.	/
0	4	5	6	7	8	9	:
;	<	=	>	?	@	B	C
D	E	F	H	J	K	L	N
O	Q	R	S	T	V	W	X
Y	Z	[	\	]	^	_	`

Ciphertext *yang dihasilkan adalah MI*. Memasukkan kunci IP ke *matrix* dan mendekripsi MU. *Ciphertext* yang dihasilkan adalah PA. Jadi hasil dekripsinya adalah MIPA.

Tabel 5 Penambahan kunci IP (Dekripsi)

I	P	M	A	U	G	1	2
3	$	!	“	#	%	&	‘
(	)	*	+	,	-	.	/
0	4	5	6	7	8	9	:
;	<	=	>	?	@	B	C
D	E	F	H	J	K	L	N
O	Q	R	S	T	V	W	X
Y	Z	[	\	]	^	_	`

Proses enkripsi dan dekripsi Elliptic Curve

Proses enkripsi *Elliptic Curve* menggunakan kunci publik dan bilangan integer yang dipilih selang $[1, p-1]$, sedangkan proses dekripsi menggunakan kunci privat. Hasil enkripsi tersebut menghasilkan *ciphertext* dengan pasangan nilai (c_i, c_i') setiap satu karakter. Adapun langkah-langkah proses enkripsi dan dekripsi *Elliptic Curve* sebagai berikut:

1. Enkripsi

Melakukan enkripsi *token* **@123** dengan kunci publik $(5,23)$, maka enkripsinya adalah sebagai berikut:

(1) Mengkode *plaintext* M menjadi sebuah titik kurva m_i

Plaintext M dipecah menjadi blok *plaintext* dengan *index* yang telah ditetapkan sebagai berikut:

0	= **'**	10	= **/**	20	= **C**	30	= **T**	40	= **"**	50	= **7**	60	= **H**
1	= **&**	11	= **:**	21	= **K**	31	= **{**	41	= **+**	51	= **?**	61	= **P**
2	= **.**	12	= **B**	22	= **S**	32	= **!**	42	= **6**	52	= **G**	62	= **X**
3	= **9**	13	= **J**	23	= **[**	33	= *****	43	= **>**	53	= **O**	63	= **`**
4	= **A**	14	= **R**	24	= **$**	34	= **5**	44	= **F**	54	= **W**		
5	= **I**	15	= **Z**	25	= **)**	35	= **=**	45	= **N**	55	= **_**		
6	= **Q**	16	= **3**	26	= **4**	36	= **E**	46	= **V**	56	= **%**		
7	= **Y**	17	= **(**	27	= **<**	37	= **M**	47	= **^**	57	= **-**		
8	= **2**	18	= **0**	28	= **D**	38	= **U**	48	= **#**	58	= **8**		
9	= **1**	19	**;**	29	**L**	39	**]**	49	**,**	59	= **@**		

$m_i = \mathrm{i}(\alpha)$

$$@(\alpha) = 59(2,4) = (82,48) \qquad 1(\alpha) = 9(2,4) = (7,116)$$

$$2(\alpha) = 8(2,4) = (43,68) \qquad 3(\alpha) = 16(2,4) = (72,32)$$

Jadi titik kurva yang didapatkan adalah $\{(82,48), (7,116), (43,68)(72,32)\}$.

(2) Menghitung c_i dan c_i' dari setiap pasangan titik kurva m_i

$$c_i = k_i(\alpha)$$
$$c_i' = k_i(Kpu) + m_i$$

$$c_1 = 3(2,4) = (74,11)$$
$$c_1' = 3(5,23) + (82,48) = (103,54) + (82,48) = (45,0)$$

Ciphertext yang dihasilkan adalah $(74,11)\ dan\ (45,0)$

$$c_2 = 3(2,4) = (74,11)$$
$$c_2' = 3(5,23) + (7,116) = (103,54) + (7,116) = (59,125)$$

Ciphertext yang dihasilkan adalah $(74,11)\ dan\ (59,125)$

$$c_3 = 3(2,4) = (74,11)$$
$$c_3' = 3(5,23) + (43,68) = (103,54) + (43,68) = (90,122)$$

Ciphertext yang dihasilkan adalah $(74,11)\ dan\ (90,122)$

$$c_4 = 3(2,4) = (74,11)$$
$$c_4' = 3(5,23) + (72,32) = (103,54) + (72,32) = (98,89)$$

Ciphertext yang dihasilkan adalah $(74,11)\ dan\ (98,89)$

Jadi *ciphertext* keseluruhan yang dihasilkan dari proses enkripsi *Elliptic Curve* adalah $\{((74,11),(45,0)),((74,11),(59,125)),((74,11)(90,122)),((74,11),(98,89))\}$

2. Dekripsi

Melakukan dekripsi *ciphertext* $\{((74,11),(45,0)), ((74,11), (59,125)), ((74,11)(90,122)), ((74,11),(98,89))\}$ dengan kunci privat 4, maka dekripsinya adalah sebagai berikut:

(1) Melakukan dekripsi setiap pasangan titik kurva (c_i, c_i')

$$m_i = c_i' - Kpr(c_i)$$

$$m_1 = (45,0) - 4(74,11) = (45,0) - (103,54) = (45,0) - (103,-54) = (45,0) + (103,77) = (82,48)\ representasi\ @$$

$$m_3 = (90,122) - 4(74,11) = (90,122) - (103,54) = (45,0) - (103,-54) = (90,122) + (103,77) = (43,68)\ representasi\ 2$$

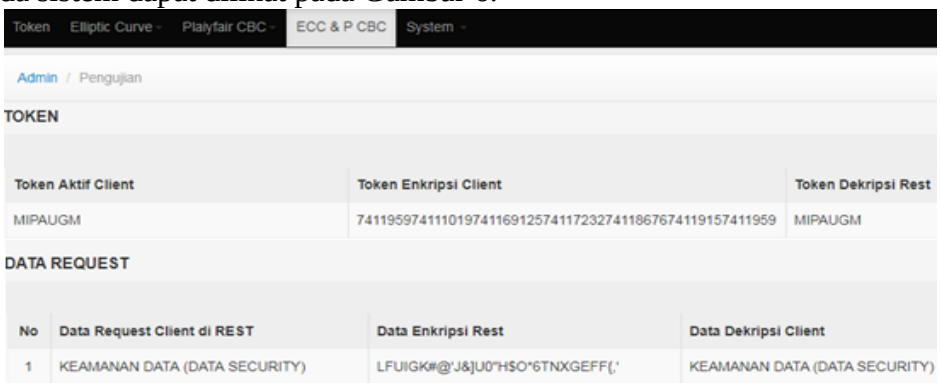
$$m_2 = (59,125) - 4(74,11)$$
$$= (59,125) - (103,54)$$
$$= (45,0) - (103,-54)$$
$$= (59,125) + (103,77)$$
$$= (7,116)\ representasi\ 1$$

$$m_4 = (98,89) - 4(74,11)$$
$$= (98,89) - (103,54)$$
$$= (45,0) - (103,-54)$$
$$= (98,89) + (103,77)$$
$$= (72,32)\ representasi\ 3$$

Jadi hasil dekripsinya adalah @123

3. Hasil dan Pembahasan

Analisis pengujian enkripsi dilakukan untuk mengetahui apakah *ciphertext* yang dilakukan oleh sistem hasilnya sama dengan yang dilakukan dengan manual. Sedangkan Analisis pengujian dekripsi dilakukan juga untuk mengetahuai apakah hasil *plaintext* yang dilakukan oleh sistem sama dengan yang dilakukan dengan manual. Analisis *Brute Force* dilakukan untuk mengetahui kompleksitas waktu yang dibutuhkan untuk melakukan penyerangan. Hasil pengujian enkripsi dan dekripsi terhadap satu data dari 10 data pada sistem dapat dilihat pada Gambar 6.



Gambar 4 Hasil enkripsi dan dekripsi terhadap token dan data *request*

3.1 Proses enkripsi dan dekripsi Playfair CBC

Pengujian enkripsi *Playfair* CBC menggunakan kunci yang berbeda dan data *text* yang berbeda. Data yang digunakan sebanyak 10.

1. Pengujian manual enkripsi KEAMANAN DATA (*DATA SECURITY*)

KE	=	**LF**	AM	=	**UI**	AN	=	**GK**	AN	=	**#@**
*D	=	**'J**	AT	=	**&]**	A*	=	**U0**	(D	=	**"H**
AT	=	**$O**	A*	=	***6**	SE	=	**TN**	CU	=	**XG**
RI	=	**EF**	TY	=	**F{**	)#	=	**,'**			

Tabel 6 merupakan hasil pengujian enkripsi *Playfair* CBC dan hanya ditampilkan 4 data dari 10 data yang digunakan.

Tabel 6 Enkripsi *Playfair* CBC

N	Kunci	*Plaintext*	*Ciphertext*
1	MIPAUGM	KEAMANAN DATA (*DATA SECURITY*)	LFUIGK#@'J&]U0"H$O*6TNXGEFF{,'
2	@123	KUNCI MEMEGANG PERANAN PENTING(KRIPTOGRAFI)	MSKEG,SMG@K;H,%V1QNY>1+O$OUJU !-VTLB!NHWCIP*&
3	FI%S^(P)O L 12UG8M	KUNCI HARUS BENAR BENAR AC AK DAN SULIT DITEBAK	RLTAR-K-- G(&<VD%S3CHWRC'T'LA6KCB #3,DPL!3)K?A)!
4	++546	KARAKTERISTIK TERSENDIRI	ICS@T{JNC@{CB7{LO5?I>5QO

2. Pengujian manual dekripsi LFUIGK#@'J&]U0"H$O*6TNXGEFF{,'

LF	=	**KE**	UI	=	**AM**	GK	=	**AN**	#@	=	**AN**
'J	=	***D**	&]	=	**AT**	U0	=	**A***	"H	=	**(D**
$O	=	**AT**	*6	=	**A***	TN	=	**SE**	XG	=	**CU**
EF	=	**RI**	F{	=	**TY**	,'	=	**)#**			

Tabel 7 merupakan hasil pengujian dekripsi *Playfair* CBC dan hanya ditampilkan 4 data dari 10 data yang digunakan. Berdasarkan hasil pengujian yang dilakukan antara sistem dan manual pada Tabel 6 dan 7 memperlihatkan hasil yang sama.

Tabel 7 Dekripsi *Playfair* CBC

N	Kunci	*Ciphertext*	*Plaintext*
1	MIPAUGM	LFUIGK#@'J&]U0"H$O*6TNXGEFF{,'	KEAMANAN DATA (*DATA SECURITY*)
2	@123	MSKEG,SMG@K;H,%V1QNY>1+O$OUJU !-VTLB!NHWCIP*&	KUNCI MEMEGANG PERANAN PENTING(KRIPTOGRAFI)
3	FI%S^(P)O L 12UG8M	RLTAR-K-- G(&<VD%S3CHWRC'T'LA6KCB #3,DPL!3)K?A)!	KUNCI HARUS BENAR BENAR AC AK DAN SULIT DITEBAK
4	++546	ICS@T{JNC@{CB7{LO5?I>5QO	KARAKTERISTIK TERSENDIRI

3.2 Proses enkripsi dan dekripsi Elliptic Curve

Pengujian enkripsi *Elliptic Curve* menggunakan kunci yang berbeda dan data *text* yang berbeda. Data yang digunakan sebanyak 10.

1. Pengujian manual enkripsi MIPAUGM dengan kunci publik (5,23)

M	$c_i = k_i\alpha$	$c'_i = k_i(Kpu) + m_i$
M	$3(2,4) = (74,11)$	$(103,54) + (38,17) = (95,9)$
I	$3(2,4) = (74,11)$	$(103,54) + (4,27) = (10,19)$
P	$3(2,4) = (74,11)$	$(103,54) + (24,112) = (69,125)$
A	$3(2,4) = (74,11)$	$(103,54) + (5,23) = (72,32)$
U	$3(2,4) = (74,11)$	$(103,54) + (107,100) = (86,76)$
G	$3(2,4) = (74,11)$	$(103,54) + (104,48) = (91,5)$
M	$3(2,4) = (74,11)$	$(103,54) + (38,17) = (95,9)$

Tabel 8 merupakan hasil pengujian enkripsi *Elliptic Curve* dan hanya ditampilkan 4 data dari 10 data yang digunakan. Pengujian dekripsi 741195974111019741169125741172327411867674119157411959 kunci privat (5,23) secara manual

Tabel 8 Enkripsi *Elliptic Curve*

No	Kunci Publik	K	*Plaintext*	*Ciphertext*
1	(5,23)	3	MIPAUGM	741195974111019741169125741172327411867674119157411959
2	(29,78)	5	@123	427364279719427411842717 58
3	(4,27)	3	FI%S^(P)OL 12UGM	741182487411901227411450741138177411231274113910574113110974111101267411119107411971974116112674117523741134317411111374111044 8
4	(75,23)	11	++546	678610573678610573678674116786410467866786

$Kprc_i$	$m_i = c_i' - Kpr(c_i)$	M
$4(74,11) = (103,54)$	$(95,9) + (103,77) = (38,17)$	M
$4(74,11) = (103,54)$	$(10,19) + (103,77) = (4,27)$	I
$4(74,11) = (103,54)$	$(69,125) + (103,77) = (24,112)$	P
$4(74,11) = (103,54)$	$(72,32) + (103,77) = (5,23)$	A
$4(74,11) = (103,54)$	$(86,76) + (103,77) = (107,100)$	U
$4(74,11) = (103,54)$	$(91,5) + (103,77) = (104,48)$	G
$4(74,11) = (103,54)$	$(95,9) + (103,77) = (38,17)$	M

Tabel 9 merupakan hasil pengujian dekripsi *Elliptic Curve* dan hanya ditampilkan 4 data dari 10 data yang digunakan. Berdasarkan hasil pengujian yang dilakukan antara sistem dan manual pada Tabel 8 dan 9 memperlihatkan hasil yang sama.

Tabel 9 Dekripsi *Elliptic Curve*

No	Kunci Privat	*Plaintext*	*Ciphertext*
1	4	741195974111019741169125741172327411867674119157411959	MIPAUGM
2	7	427364279719427411842717 58	@123
3	5	741182487411901227411450741138177411231274113910574113110974111101267411119107411971974116112674117523741134317411111374111044 8	FI%S^(P)OL 12UGM
4	23	678610573678610573678674116786410467866786	++546

3.3 Analisis brute force Playfair CBC

Banyaknya waktu yang digunakan untuk memecahkan *ciphertext* adalah.

$$Waktu = \frac{(64-n)!/(64-2\times N-n)!}{Estimasi}$$

$64 = Matrix\ 8 \times 8$

$n = Banyaknya\ karakter$

$N = Banyaknya\ digraph$

Dari rumus tersebut dapat dihitung berapa waktu estimasi untuk memecahkan *plaintext* tanpa mengetahui kuncinya yang dilakukan dengan menggunakan super komputer buatan cina yaitu *Sunway TaihuLight System* 93 *petaflop* $(93.000.000.000.000.000)/$ detik dan Komputer $i7$ 3,4 GHz $(3.000.000.000)/$ detik. Analisis perhitungan *Brute Force* diperlihatkan pada Tabel 10.

Tabel 10 Estimasi *brute force Playfair* CBC

No	Banyaknya *ciphertext*	*Estimasi Percobaan Sunway TaihuLight System*	*Estimasi Percobaan i7*
1	2	0 Detik	1 Detik
2	10	93 Detik	335 Hari
3	16	16 Tahun	498.665.335 Tahun
4	20	1.460.844 Tahun	45.286.193.252.826 Tahun

3.4 Analisis Brute Force Elliptic Curve

Banyaknya waktu yang digunakan untuk memecahkan *ciphertext* adalah.

$$Waktu = \frac{n + 2^{P/2} + \frac{64!}{n!\,(64-n)!}}{Estimasi}$$

$$n = Banyaknya\ pasangan\ ciphertext$$

$$p = Bilangan\ prima/\ modulo\ p$$

$$64 = Banyaknya\ karakter$$

Analisis perhitungan *Brute Force* diperlihatkan pada Tabel 11.

Tabel 11 Estimasi brute force Elliptic Curve

No	Banyaknya *ciphertext*	*P*	*Estimasi Percobaan Sunway TaihuLight System*	*Estimasi Percobaan i7*
1	2	131	9 Menit	551 Tahun
2	10	167	4 Tahun	144.569.532 Tahun
3	16	177	149 Tahun	4.626.225.036 Tahun
4	20	187	4.775 Tahun	148.039.201.163 Tahun

4. Kesimpulan

Pengamanan token dengan menggunakan Elliptic Curve dan pengamanan data request dengan Playfair CBC berhasil dilakukan dan perhitungan antara manual dan sistem menghasilkan data yang sama pada proses enkripsi atau dekripsi, sehingga akurasi data yang didapat adalah 100%. Perluasan ukuran matrix dari 5×5 menjadi 8×8 pada algoritme Playfair CBC berhasil dilakukan dan terlihat ciphertext lebih tidak terbaca, karena matrix 5×5 hanya berisi karakter $\mathrm{A} - \mathrm{Z}$ sedangkan matrix 8×8 ditambah karakter "123$!"#%|'()*~,-./0456789:;<=>?@[{]^_`" sedangkan penggunaan index karakter pada algoritme Elliptic Curve membuat penyerangan brute force semakin sulit, karena posisi index karakter dapat dirubah sesuai kesepakatan REST Server dan Client Server.

Algoritme yang digunakan dalam pengamanan data request menggunakan Playfair CBC, untuk pengembanganya dapat dimodifikasi dengan penggabungan kriptografi yang lain sehingga akan memperkuat algoritme sebelumnya dan sistem enkripsi dan dekripsi hanya untuk format data text, untuk pengembanganya dapat dilakukan dengan format yang lain (.pdf, .text dan docx/doc).

Daftar Pustaka

Al-Zoubi, K. dan Wainer, G., 2009, Using REST Web-Services Architecture for Distributed Simulation, IEEE, (114-121), https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5158326.

Goyal, P., Sharma , G. dan Kushwah, S. S., 2015, A New Modified Playfair Algorithm using CBC, IEEE, Dept.of Computer Science & Engineering ITM Group of Institutions, Gwalior (India), https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7546249.Kopack, M. dan Potts, S., 2003, Sams Teach Yourself Web Services in 24 Hours, Indianapolis, Indiana, USA.

Munir, R., 2006, Kriptografi, Informatika Bandung, Bandung.

Rahardjo, B., 2002, Kemanan Sistem Informasi Berbasis Internet, PT Insan Komunikasi , Bandung, Indonesia.

Schneier, B., 1996, Applied Kriptografi, Scond Edition : Protocols, Algoritms, and Source Code in C. John Wiley & Sons.

Stallings, W., 2003, Cryptography and Network Security Principles and Practice Third Edition, Pearson Education, New Jersey, USA.

Stallings, W., 2014, Cryptography and Network Security Principles and Practice Sixth Edition, Pearson Education, Edinburg Gate Harlow Essex CM20 2JE, England.

Stinson, D. R., 2006, Cryptography Theory And Practice Third Edition, Chapman & Hall/CRC Taylor & Franscis Group, Canada.

Webber, J., Parastatidis, S. dan Robinson, I., 2010, REST in Practice, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, United States of America.