# Implementation of the Autokey Algorithm ASCII Text Cryptography Using JavaFX GUI

Ricky Dirgantara<sup>1</sup>, Hans Valerian Lenice<sup>2</sup>, L. Budi Handoko<sup>3</sup>, Wise Herowati<sup>4</sup>, Folasade Olubusola Isinkaye<sup>5</sup>

<sup>1,2,3,4</sup>Faculty of Computer Science, University of Dian Nuswantoro, Semarang, Indonesia <sup>5</sup>Ekiti State University, Ado Ekiti, Nigeria

# **Article Info:**

#### Keywords:

Autokey Cipher Avalanche Effect Character Error Rate Cryptography Entropy

#### ABSTRACT

The advance of information and communication technology has transformed the world in ways we could never have imagined. From the analog era to the digital age, technology has paved the way for new innovations and discoveries, changing the way we interact, communicate, work, and even play. Data breaches are one of the threats that frequently arise in this era of digital information since the data has great value in enhancing the security of the data we transmit, one of which is using code. There are numerous ways to encode data in cryptography, but one popular technique is Autokey Cipher cryptography. Autokey cipher is a development of vigenere cipher which is a solution to overcome the repetition of certain characters in keys. Autokey operations are performed based on the key length and calculated using the tabula recta of the vigenere cipher, the key from autokey is a combination of plaintext to produce a new key along the plaintext. In this research we built the implementation using Java library JavaFX. The GUI also shows how the autokey cipher works, which will give us an understanding about the encryption and decryption process that is built with ASCII text. And the other additions are the GUI that shows the calculations such as Avalanche Effect (AE), Bit Error Rate (BER), Character Error Rate (CER), and Entropy.

Author Correspondence:

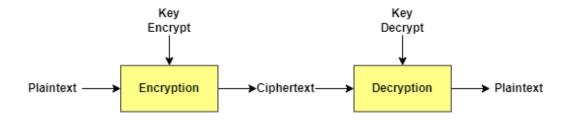
L. Budi Handoko, Faculty of Computer Science University of Dian Nuswantoro Email: handoko@dsn.dinus.ac.id

### 1. INTRODUCTION

Java is a programming language that is widely used for coding web applications. Java also supports a graphical user interface or GUI which is one of Java's main features. The main GUI we use is the Java library also known as JavaFX. GUI is a system of interactive visual components used in software, smartphones, and various other electronic devices [1], [2]. GUI are represented using several images which are referred to as GUI elements. With elements such as windows, icons and menus, computers can carry out commands such as opening, deleting, and moving files. One of the libraries provided by Java for building GUIs is JavaFX. JavaFX is a software platform for building Rich Internet Applications also known as RIA applications that can run on various devices. Such as desktop computers, web browsers on Windows, Linux and MacOS. Below is an overview of both processes in cryptography.

Cryptography is a process for disguising information so that it cannot be understood by other people, Cryptosystem is a method designed to carry out a cryptographic process while cryptanalysis is a process for re-disguising the results of a cryptosystem into understandable information [3], [4], [5], [6]. Plaintext can be said to be the original message/information that can still be understood or understood by humans, while ciphertext is a message that has been disguised so that it cannot be understood by humans [7],

[8]. What about Encryption and Decryption? Encryption is the process of changing from plaintext to ciphertext while decryption is the process of changing ciphertext back to plaintext as shown in Figure 1.



Mathematically, the encryption process or function (E) can be written as : E(M) = C

Where: M is plaintext (message) and C is ciphertext.

The decryption process or function (D) can be written as: D(C) = M

Figure 1. Common Cryptography Scheme [9]

The main problem from this research that we made is to implement a graphical user interface that integrated with Autokey Algorithm to make it easy to use and understand for users [10], [11], [12], [13]. The GUI should support many text types and formats, and that why we used ASCII text to make it easier, and it will assist users that use our program to understand how an algorithm keyboard should operate automatically throughout a process. In addition, efficiency, and speed of the encrypt and decrypt processes are quite important for long and complex texts that are encoded with ASCII. In addition, the primary concern in this study is the security and privacy of user data during the encrypting and decrypting processes. The validation and verification process to ensure that the encrypt and decrypt functions are appropriate and crucial in this course are highly important [14], [15], [16]. In addition, the GUI needs to be simple to learn or modify to support other encrypt and decrypt algorithms in the future.

Here, we want to introduce to people about how the autokey cipher algorithm is implemented or works in a GUI using the Java programming language. In this journal we also explain the calculations, namely Avalanche Effect, Bit Error Rate, Character Error Rate and Entropy, so that people can better understand the performance of the autokey algorithm and examples of its use that we describe in the GUI implementation.

#### 2. METHOD

For this Research Method we started by searching for information about the AutoKey Cipher Algorithm and several tutorials for implementing it in a GUI using JavaFX. From these two things, we knew of the basic concepts of how AutoKey cryptography can be implemented in GUI-based applications. In this way we can carry out encryption and decryption as well as other calculations. After doing this, we also don't forget to carry out testing or tests regarding our application so that it is suitable and works correctly as we want.

# 2.1. Cryptography

Cryptography is an important component for maintaining data confidentiality. Cryptography plays an important role in the security and protection of network data because it allows the storage and transmission of sensitive data over unsecured networks, such as the Internet, so that it cannot be read by anyone except the person intended to receive it [3], [17], [18], [19], [20]. Cryptography Symmetric is a type of cryptography, where there is only one key to encrypt and decrypt from the information. Meanwhile, cryptography asymmetric is a type of cryptography that uses two keys: public key that can be shared to public and a private key that is only known by the data owner [21], [22], [23]. Further information about the public key is that it can be shared with anyone, but for the private key it must be kept confidential. If someone has your public key and they can encrypt your messages but only you with the private key can decrypt them.

#### **2.2. ASCII**

ASCII is based on the simple idea of using numbers to represent text, for example numbers that represent letters "A", "q", or "!", collectively letters, numbers and symbols are called characters [24]. By signing a number to each of characters in a block of text, the block of text can be represented entirely by numbers and that easily stored in the computer memories, for example letter "A" is always assigned the number '065' conversely '065' always reverse for letter "A". ASCII is an english language subsets of Unicode meaning that all of ascii exists within the Unicode. ASCII is also commonly used in computer science particularly in programming language, text files and data conversions as displayed in Figure 2.

Ś as	cii										
\$ as 000	<b>ื</b>	^@	032	$0 \times 20$		<b>Ø64</b>	0×40	•	096	$0 \times 60$	•
001	$0 \times 01$	^^	032 033	$0 \times 21$	ŧ.	064 065	$0 \times 41$	Ã B	097	$0 \times 61$	a
002	$0 \times 02$	^R	034	Ø×22		066	0×42	${f B}$	098	0×62	a b
003	$0 \times 03$	^c	035	Ø×23	#	Ø67	0×43	<u>c</u>	099	Øx63	c d
004	0×04	ΛĎ	036	0x24	# \$ % &,	968	0×44	Đ	100	0x64	d
005 006	0×05	^É	037	0×25	×.	<b>069</b>	0×45	É F	101	0×65	e f
006 007	0×06	^F ^G	038 039	0×26	Č.	070	0×46	G	102	0×66	
007 008	0×07 0×08	^H	039 040	0×27 0×28	<	071 072	0×47 0×48	9	103 104	0x67 0x68	ទួ
009	0×09	^T	041	0x29	>	073	0x49	Ÿ	105	0x69	4
01 Ó	0x0a	^.1	042	Øx2a	*	074	0x4a	H I J	106	Øx6a	g h j k 1
011	0×0b	$\sim v$	043	Øx2b	+	075	0x4b	ĸ	107	Øx6b	ĸ
$\overline{012}$	0x0c	^T.	044	Øx2c		076	Øx4c	К L	108	0×6c	î
$\overline{013}$	$0 \times 0 d$	^M	045	0x2d	_	077	$0 \times 4d$	M	109	Øx6d	m
014	ØхØе	^M ^N	<b>Ø46</b>	Øx2e		078	Øx4e	MZO	110	0х6е	n
015	$0 \times 0 f$	^ሰ	047	$0 \times 2f$	· Ø	079	$0 \times 4f$	0	111	$0 \times 6f$	О
016	0×10	^ř	<b>Ø48</b>	0x30 0x31	Ø	080	0x50 0x51 0x52 0x53	$\mathbf{P}$	112	0×70	$\mathbf{p}$
017	0x11	^g	049	0×31	1	Ø81	Øx51	Q R	113	0x71 0x72	${f q}$
<b>Ø18</b>	0x12	^Ř	050	0×32	2	082	0x52	К	114	0x72	P.
019	0x13	^S ^T	<b>051</b>	0x33	3	083 084	0x53	S	115	0x73 0x74	s
020 021	0x14 0x15	^ti	<b>052</b>	0x34	4	084 085	0x54	Ú	116 117	0x74	t u
$\begin{array}{c} 021 \\ 022 \end{array}$	0x15	~Ŭ	053 054	0x35 0x36	2	Ø86	0x55 0x56	H	118	0x75 0x76	v
023	0x17	^u	055	0x37	9	087	0x57	ŭ	119	0x77	ŭ
И24	0×18	^Ÿ	<b>0</b> 56	Øx38	123456789	088	Øx58	Ÿ	120	0x78	×
025	0×19	^Ÿ	<b>0</b> 57	Øx39	ğ	089	Øx59	Ÿ	121	Øx79	Ŷ
025 026	Øx1a	^7	057 058	0x39 0x3a	-	090	0x59 0x5a	ÜWXXZE	122 123	0x79 0x7a	y 2 < > ≥
027	$0 \times 1 b$	^г	059	$0 \times 3 b$	;	091	0x5b 0x5c	Е	123	Øx7b	€
028	0x1c	^_	060	0x3c	,	092	0х5с	5	194	0x7c	H
029	0×1d	^i	<b>Ø61</b>	0x3d 0x3e	=	093	0x5d 0x5e	ž	125 126	0x7d 0x7e	3
030	0x1e	^^	062		?	094	Øx5e	^	126 127	0x7e	
031	$0 \times 1f$	^_	063	$0 \times 3f$	?	095	0x5f		199	$0 \times 7f$	Δ
							07.01	_	12.		
128	0×80	?				192		A			à
128 129	0×80 0×81	?	160 161	0xa0 0xa1	i	192	0xc0 0xc1	A A		0xe0 0xe1	à
129	0×81 0×82		160 161 162	0xa0 0xa1 0xa2	i	192 193 194	0xc0 0xc1 0xc2	A	224 225 226	0xe0 0xe1 0xe2	àáá
129 130	0x81 0x82 0x83	? ? <b>f</b>	160 161 162 163	0xa0 0xa1 0xa2 0xa3	÷¢€	192 193 194 195	0xc0 0xc1 0xc2 0xc3	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3	àáâa
129 130	0x81 0x82 0x83 0x84		160 161 162 163 164	0xa0 0xa1 0xa2 0xa3 0xa4	÷¢€	192 193 194 195 196	0xc0 0xc1 0xc2 0xc3 0xc4	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3 0xe4	`a`aca a:ac
129 130	0x81 0x82 0x83 0x84 0x85	f :	160 161 162 163 164	0xa0 0xa1 0xa2 0xa3 0xa4 0xa5	÷¢€	192 193 194 195 196	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3 0xe4 0xe5	a ä 8
129 130	0x81 0x82 0x83 0x84 0x85 0x86	f :	160 161 162 163 164 165 166	0xa0 0xa1 0xa2 0xa3 0xa4 0xa5 0xa6	÷¢€	192 193 194 195 196 197 198	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3 0xe4 0xe5 0xe6	a:a96 &
129 130 131 132 133 134 135	0x81 0x82 0x83 0x84 0x85 0x86 0x87	f :	160 161 162 163 164 165 166	0xa0 0xa1 0xa2 0xa3 0xa4 0xa5 0xa6 0xa7	÷¢€	192 193 194 195 196 197 198 199	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6 0xc7	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3 0xe4 0xe5 0xe6 0xe7	a:a96 &
129 130 131 132 133 134 135	0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88	<b>壬</b>	160 161 162 163 164 165 166 167	0xa0 0xa1 0xa2 0xa3 0xa4 0xa5 0xa6 0xa7	÷ ⇔⇔ ######	192 193 194 195 196 197 198	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6 0xc7	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3 0xe4 0xe5 0xe6 0xe7	a:a96 &
129 130 131 132 133 134 135 136	0×81 0×82 0×83 0×84 0×85 0×86 0×87 0×88	<b>壬</b>	160 161 162 163 164 165 166 167 168	0xa0 0xa1 0xa2 0xa3 0xa4 0xa5 0xa6 0xa7 0xa8	÷ ⇔⇔ ######	192 193 194 195 196 197 198	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6 0xc7	A	224 225 226 227	0xe0 0xe1 0xe2 0xe3 0xe4 0xe5 0xe6 0xe7 0xe8 0xe9	a:a96 &
129 130 131 132 133 134 135 136	0×81 0×82 0×83 0×84 0×85 0×86 0×87 0×88 0×89 0×8a	<b>壬</b>	160 161 162 163 164 165 166 167 168 169	OxaO Oxa1 Oxa2 Oxa3 Oxa4 Oxa5 Oxa6 Oxa7 Oxa8 Oxa8	.+ O⊄L \$\$#=#07 • od	192 193 194 195 196 197 198 199 200 201	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6 0xc7 0xc8	A	224 225 226 227 228 229 231 232 233 234	0xe0 0xe1 0xe2 0xe3 0xe4 0xe5 0xe6 0xe7 0xe9	a:a96 &
129 130 131 133 133 133 133 133 140	0×81 0×82 0×83 0×84 0×85 0×86 0×87 0×88	· · · · · · · · · · · · · · · · · · ·	160 161 162 163 164 165 166 167 168 169	0xa0 0xa1 0xa2 0xa3 0xa4 0xa5 0xa6 0xa7 0xa8	÷ ⇔⇔ ######	192 193 194 195 197 198 199 2001 2001 2023 204	0xc0 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6 0xc7	A	224 225 226 227 228 229 231 232 233 234	0xe0 0xe2 0xe2 0xe3 0xe4 0xe6 0xe7 0xe8 0xea 0xea	a:a96 &
129 1331 1332 1334 1336 1338 1338 1441	0x81 0x82 0x84 0x85 0x86 0x88 0x88 0x88 0x88 0x8b 0x8b 0x8b	· · · · · · · · · · · · · · · · · · ·	160 162 163 164 165 1667 168 169 170 172 173	Oxa0 Oxa1 Oxa2 Oxa3 Oxa4 Oxa6 Oxa6 Oxa8 Oxa8 Oxaa Oxaa	% po :დ] <b>— ჩ</b> \$# ებტ•-	192 193 194 195 196 199 201 202 203 203 205	9xc9 9xc2 9xc3 9xc4 9xc5 9xc6 9xc7 9xc9 9xc9 9xca	A	224 225 226 227 228 229 231 232 233 234	Oxed Oxed Oxed Oxed Oxed Oxed Oxed Oxed	a:a96 &
129 1331 1332 1334 1336 1338 1431 1442	0x81 0x82 0x834 0x85 0x86 0x86 0x88 0x88 0x86 0x86 0x86 0x86	· · · · · · · · · · · · · · · · · · ·	160 161 162 163 165 1667 1689 170 171 173 174	Oxad Oxad Oxad Oxad Oxad Oxad Oxad Oxad	% po :დ] <b>— ჩ</b> \$# ებტ•-	192 193 194 195 197 198 199 200 200 200 200 200 200 200 200 200 2	9xc9 9xc2 9xc3 9xc4 9xc6 9xc6 9xc9 9xca 9xcd 9xcd	A	224 225 226 227 228 229 231 232 233 234	Oxed Oxed Oxed Oxed Oxed Oxed Oxed Oxed	a:a96 &
129 113334 13334 13336 1339 1339 1441 1143	0x81 0x823 0x845 0x8867 0x889 0x889 0x880 0x886 0x886 0x886 0x86 0x86	· · · · · · · · · · · · · · · · · · ·	160 161 162 163 1645 1667 1689 170 171 172 1734 175	9xa9 9xa1 9xa2 9xa3 9xa4 9xa6 9xa6 9xa6 9xaa 9xaa 9xaa 9xaa 9xaa		192 193 194 195 196 197 200 201 202 203 204 206 207	9xc1 9xc2 9xc3 9xc4 9xc5 9xc6 9xc7 9xcca 9xcca 9xcca 9xcca	A	224 225 226 227 228 239 231 2334 235 236 237 238 239	0xe0 0xe2 0xe2 0xe3 0xe6 0xe6 0xe6 0xed 0xed 0xed 0xed	a:a96 &
129 113123 11335 11335 11339 11442 11442 11444 11444	0x81 0x823 0x8845 0x8867 0x888 0x888 0x88 0x88 0x88 0x88 0x88 0	· · · · · · · · · · · · · · · · · · ·	160 161 162 1634 165 166 168 170 172 173 1775 176	Oxa0 Oxa1 Oxa2 Oxa3 Oxa4 Oxa5 Oxa6 Oxa7 Oxa8 Oxaa0 Oxaa0 Oxab		193 193 194 195 197 198 199 201 202 206 206 208 208	0xc1 0xc2 0xc3 0xc5 0xc6 0xc6 0xc6 0xcd 0xcd 0xcd 0xcd	A	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	Oxe1 Oxe2 Oxe2 Oxe3 Oxe6 Oxe6 Oxe6 Oxecd Oxecd Oxecd Oxe6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe6	a:a96 &
1290 133123345 13335678 133441 11442 14444 145	0x81 0x823 0x845 0x886 0x886 0x88 0x88 0x88 0x88 0x88 0x	f++K xs<0?N??.	160 161 162 163 1665 1667 1689 170 177 1773 1774 1776 177	Oxad Oxad Oxad Oxad Oxad Oxad Oxad Oxad		192 193 194 195 196 199 201 202 203 205 207 209	9xc1 9xc2 9xc3 9xc5 9xc6 9xc6 9xcc9 9xccd 9xcd 9xcd 9xcd 9xdd 9xd1	А А А А А А А А А А А А А А А А А А А	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	9xe21 9xe23 9xe45 9xe65 9xee6 9xeed 9xeed 9xeed 9xeff 9xf1	a:a96 &
129 1133 1133 1133 1133 1133 1144 1144 114	0x82344567889 abcdef0120x88879890x88890x88800x88990x888990x89990x89990x8999	f · + + 大 × S < O ? N ? ?	160 161 163 1645 1667 1689 170 1773 1774 1776 1778	Oxad Oxad Oxad Oxad Oxad Oxad Oxad Oxad	.+ OH ☆ H = 100 +	192 193 194 195 199 199 200 203 203 206 207 209 209 209 209 209 209 209 209 209 209	9xc1 9xc2 9xc4 9xc5 9xc6 9xc6 9xc6 9xccd 9xccd 9xcd 9xdd 9xd2	44444444444444444444444444444444444444	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	Oxec23 Oxec23 Oxec45 Oxec67 Oxec60 Oxeccd Ox	a:a96 &
129 1131 1132 11334 11336 11338 1144 1144 1144 1144 1147	0x8234456 0x88456789 0x886789 0x8889 0x8880 0x8860 0x8910 0x910 0x910 0x910 0x910 0x910 0x910 0x910	f++K xs<0?N??.	160 161 163 1645 1667 1689 170 1773 1774 1776 1778	9xa1 9xa2 9xa3 9xa45 9xa6 9xa6 9xa6 9xaa 9xaa 9xaa 9xaa 9xab 9xb1 9xb1 9xb2		193 194 195 197 198 199 201 202 203 206 207 209 201 201 201 201 201 201 201 201 201 201	9xc1 9xc2 9xc3 9xc4 9xc6 9xc6 9xcd 9xcd 9xcd 9xcd 9xdd1 9xdd1 9xd3	ааа:aж#Сёмёённннойоо	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	Oxec23 Oxec23 Oxec45 Oxec6 Oxe6 Oxec6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe6 Oxe	a:a96 &
11113345678890112344567889011111111111111111111111111111111111	0x823 0x8845 0x885 0x8867 0x888 0x886 0x886 0x886 0x86 0x91 0x91 0x91	(f) 、	1601 1662 1663 1664 1668 1668 1771 1773 1774 1776 1778 1789	Oxaa1 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaa6 Oxaa0 Oxaac Oxaac Oxab0 Oxbb1 Oxbb2 Oxb4	୦୯1୫5# <b>=</b> /ଉ୦୯୦୩୪୮  ୫  ୦+୮୯୯୦	193 194 195 197 198 199 201 203 206 207 209 2112 2112	Oxec2 Oxec2 Oxec5 Oxec6 Oxec6 Oxec6 Oxecd Oxecd Oxecd Oxed1 Oxed1 Oxed2 Oxed4 Oxed4 Oxed4 Oxed4 Oxed4 Oxed4 Oxed4 Oxed4 Oxed4	ааа:«Аёсыеенннахооо	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	91000000000000000000000000000000000000	alaa e ceeeeleiliiiidiidiioooo
1230 11333 11333 11333 11333 1144 1144 114	0x82345 0xx845 0xx845 0xx887 0xx88 0xx88 0xx88 0xx88 0xx89 0xx97 0xx97 0xx95 0xx95 0xx95	(f) 、	1601 1662 1663 1664 1667 1669 1771 1775 1776 1776 1778 1780 181	Oxaa1 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaa7 Oxaad Oxaab Oxaad Oxaab Oxbb1 Oxbb4 Oxbb4 Oxb5	୦୯1୫5# <b>=</b> /ଉ୦୯୦୩୪୮  ୫  ୦+୮୯୯୦	1934 1945 1978 1978 1999 2003 2004 2007 2009 2011 2011 2011 2011 2011 2011 2011	9xc12 9xc23 9xc45 9xc66 9xc69 9xccd 9xcd1 9xdd1 9xdd3 9xdd3 9xdd3 9xdd3 9xd5	ааа:«Аёсыеенннахооо	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	012345 0xxee456789 0xxee66789 0xxee6678 0xxee6788 0xxee678 0xx	alaa e ceeeeleiliiiidiidiioooo
1230 11333 11333 11333 11333 1144 1144 114	0x82344567889 abcdef01234567889 ax88888889 ax88999999999999999999999	(計 *・+	1601 1662 1663 1665 1667 1669 1771 1775 1778 1779 1179 1181 1182	Oxaa1 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaa6 Oxaab Oxaab Oxaab Oxab0 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb6 Oxbb6	.+ OH ☆ H = 100 +	1934 1945 1978 1990 1990 1990 2003 2006 2007 2008 2008 2008 2011 2113 2114 2114 2114 2114 2114	9xc1 9xc23 9xc45 9xc67 9xc67 9xccd 9xccd 9xcd41 9xdd4 9xdd4 9xdd5 9xd6	аааажерешеннын өзөөөө	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	012345 0xxee456789 0xxee66789 0xxee6678 0xxee6788 0xxee678 0xx	a:a:a & cyeve:e:e:i:i:i:d:ii:d:ii:o:o:o:o:o:o
11111111111111111111111111111111111111	0x82344567889 abcdef01234567889 xx88888889 xx999923456799234567	(f) 、	16123 1663 1663 1668 1669 1773 1776 1778 1778 1778 1778 1882 1883	Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxaal Oxabl Oxbbl Oxbb Oxbb Oxbb Oxbb Oxbb		1934 1945 1978 1990 1990 1990 2003 2006 2007 2008 2008 2008 2011 2113 2114 2114 2114 2114 2114	0xc1 0xc2 0xc2 0xc5 0xc6 0xc6 0xc6 0xc6 0xc6 0xc6 0xc6 0xdd1 0xdd3 0xdd5 0xd4 0xd4 0xd7	аааажерешеннын өзөөөө	224 225 226 227 2289 231 2331 2334 2336 2337 2338 2349	012345667889 abcdef01123445667889 conserved by the conservation of	a:a& & cvete@:extxxxxdid@roocoo:o:o:o
11111111111111111111111111111111111111	0x82344567889 abcdef01234567889 ax88888889 ax88999999999999999999999	(計 *・+	1612 1663 1664 1666 1669 1771 1775 1778 1177 1178 1188 1188 1188	Oxaa1 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaa6 Oxaab Oxaab Oxaab Oxab0 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb6 Oxbb6		19345 1974 1975 1978 1978 1978 2001 2003 2005 2007 2011 2115 2117 2117	9xc1 9xc23 9xc45 9xc67 9xc67 9xccd 9xccd 9xcd41 9xdd4 9xdd4 9xdd5 9xd6	ааа;ая∉сечеенныехоооо хол	22567 22278 22278 223312 233345 23339 23339 2441 244444 244444 24444 2444 2444 244	01123456789 abcdeef01123456789000000000000000000000000000000000000	a:a& & cvete@:extxxxxdid@roocoo:o:o:o
11111334567890112344567890115534	00x884567889 abcdef0123445667889 abcdef0123445667889 abcdef0123445667899 acx8888888889999999999999999999999999999	、手・・+1人 NS <o?n??・・・・ 1~="" s<="" t="" td=""  =""><td>16123456 16623456678 1663456678 17712341776 17789 118833456 188456</td><td>Oxaa2 Oxaa2 Oxaa5 Oxaa6 Oxabb1 Oxabb2 Oxabb2 Oxabb3</td><td></td><td>1934 1934 1996 1998 1990 1990 1990 2002 2007 2007 2007 2007 2007 2007 2</td><td>9xc1234 9xc234 9xc25 9xc26 9xc29 9xc20 9xc20 9xc20 9xc20 9xdd12 9xdd6 9xdd9 9xdd9 9xdd9 9xdd9 9xdd9 9xdd9</td><td>ааажжеешеннныхоооо хору</td><td>22567 22278 22278 223312 233345 23339 23339 2441 244444 244444 24444 2444 2444 244</td><td>0123456789 abcdef0123456789 avxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</td><td>a:a&amp; &amp; cvete@:extxxxxdid@roocoo:o:o:o</td></o?n??・・・・>	16123456 16623456678 1663456678 17712341776 17789 118833456 188456	Oxaa2 Oxaa2 Oxaa5 Oxaa6 Oxabb1 Oxabb2 Oxabb2 Oxabb3		1934 1934 1996 1998 1990 1990 1990 2002 2007 2007 2007 2007 2007 2007 2	9xc1234 9xc234 9xc25 9xc26 9xc29 9xc20 9xc20 9xc20 9xc20 9xdd12 9xdd6 9xdd9 9xdd9 9xdd9 9xdd9 9xdd9 9xdd9	ааажжеешеннныхоооо хору	22567 22278 22278 223312 233345 23339 23339 2441 244444 244444 24444 2444 2444 244	0123456789 abcdef0123456789 avxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	a:a& & cvete@:extxxxxdid@roocoo:o:o:o
11111334567890112344567890115534	00000000000000000000000000000000000000	(f) ** ** ** ** ** ** ** ** ** ** ** ** **	16612345667 166345667 16689 11773457 11789 118834 118867 118867	Oxaa2 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxabbb		1934 194 1956 1978 1999 2012 2024 2011 2012 2011 2012 2011 2013 2011 2012 2011 2012 2013 2014 2014 2014 2014 2014 2014 2014 2014	Oxec23 Oxec23 Oxec40 Oxec6 Oxec6 Oxecd oxecd Oxec6 Oxec6 Oxec6 Oxec6 Oxed1 Oxed1 Oxed7 Oxed7 Oxed7 Oxed8 Oxe	аааажеременниндгоооохоры	22567 22278 22278 223312 233345 23339 23339 2441 244444 244444 24444 2444 2444 244	011234567889 abcdef011234456789 abcdef0100000000000000000000000000000000000	a:a& & cvete@:extxxxxdn/oooooooo
11111111111111111111111111111111111111	0x823456789 abcdef01234456789 abcdef000x888888888888888888998x899998x99998x9998x9998x9998x998x998x998x998x998x998x998x998x998x998x998x998x998x998x998x998x9	(f) ・・+	1661234 166234 166678 1177234 11778 11778 118834 11887 11887 11887 11887 11887 11887 11887	Oxaa2 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaa6 Oxaab Oxaab Oxaab Oxabb Oxbb3 Oxbb4 Oxbb4 Oxbb7 Oxbb7 Oxbb7 Oxbb7 Oxbb0		1934 194 1956 1978 1999 2012 2024 2011 2113 2115 2117 2117 2117 2117 2117 2117 2117	9xc123 9xc23 9xc25 9xc67 9xcc4 9xccd 9xccd 9xcd4 9xdd4 9xdd67 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdd 9xdd	ааамженинныхоооо хорго	22567 22278 22278 223312 233345 23339 23339 2441 244444 244444 24444 2444 2444 244	01123456789 abcdeef01123456789 abcdeef010000000000000000000000000000000000	a;a& & \$\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
11111111111111111111111111111111111111	1233456789 abcdef0123456789 abcdef0123456789 abcdef0123456789 abcdef0123456789 abcd	(方 · · + 1人 ×8く0?囚?? · · · · · · · · · · · · · · · · · ·	166123456678 16623456678 166451669 117711777 117811883456789 1188889	Oxaa2 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaacd Oxaacd Oxaacd Oxbb3 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb4 Oxbb6 Oxb		19345 19345 19967 1998 1999 1999 1999 1999 1999 1999 199	9xcc23 9xcc23 9xcc5 9xcc6 9xccb 9xccd 9xcdd1 9xdd3 9xdd6 9xdd9 9xdd9 9xdd6 9xd6 9x	аааажеременниндгоооохоры	22567 22278 22278 223312 233345 23339 23339 2441 244444 244444 24444 2444 2444 244	01234566789 ab cdeef01234566789 ab cdeef012344566789 ab cdeef012344566789 ab cdeef012344566789 ab cd	a:a& & cvete@:extxxxxdn/oooooooo
11111111111111111111111111111111111111	0x823456789 abcdef01234456789 abcdef000x888888888888888888998x899998x99998x9998x9998x9998x998x998x998x998x998x998x998x998x998x998x998x998x998x998x998x998x9	(f) ・・+	1661234 166234 166678 1177234 11778 11778 118834 11887 11887 11887 11887 11887 11887 11887	Oxaa2 Oxaa2 Oxaa3 Oxaa5 Oxaa6 Oxaa6 Oxaab Oxaab Oxaab Oxabb Oxbb3 Oxbb4 Oxbb4 Oxbb7 Oxbb7 Oxbb7 Oxbb7 Oxbb0		1934 194 1956 1978 1999 2012 2024 2011 2113 2115 2117 2117 2117 2117 2117 2117 2117	9xc123 9xc23 9xc25 9xc67 9xcc4 9xccd 9xccd 9xcd4 9xdd4 9xdd67 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdda 9xdd 9xdd	ааамженинныхоооо хорго	22567 22267 2229 22312 22334 22334 22334 22334 224434 2444 2444 24467 2447 2448	01123456789 abcdeef01123456789 abcdeef010000000000000000000000000000000000	a;a& & \$\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Figure 2. ASCII Table

#### 2.3. Autokey Cipher

Autokey in cryptography refers to a method of encryption where the key used in the encryption process is based on the original text. In the autokey method, the original text serves as the key, and this key is

☐ ISSN: xxxx-xxxx

automatically or cyclically applied to the text to produce the encrypted text. This method falls under the category of symmetric-key algorithms as shown in equation (1) and (2).

$$Ci = (Pi + Ki) \bmod 26 \tag{1}$$

Where:

C is Result of encrypted text (Ciphertext) from i

P is Plainteks from i

K is Key from i

The encryption process involves converting the text's original length (Pi) to the text's original length (Ki), and the result is expressed in modulo (M).

$$Pi = (Ci + Ki) \bmod 26 \tag{2}$$

Where:

P is Plaintext from i

C is Result of encrypted text (Ciphertext) from i

K is Key from i

The Decryption process involves reversing by subtracting the value of the encrypted text (C\_i) from the value of the key (Ki), and the result expressed in modulo (M).

#### 2.4. Avalanche Effect (AE)

The Avalanche Effect is very important because it ensures that even small changes in the autokey algorithm will produce the same output, this ensures that attackers cannot easily predict the original text through statistical analysis. This happens because in an auto-key algorithm, a small change in the original text or key must result in a significant change in the encrypted text.

$$AE = \frac{Total\ different\ bits\ from\ plaintext\ and\ ciphertext}{total\ characters} x\ 100\% \tag{3}$$

This equation above calculates the proportion of the encrypted text that changes when the original text or key is slightly changed.

#### 2.5. Bit Error Rate (BER)

Bit Error Rate (BER) is a measure of telecommunication signal integrity based on the quantity or percentage of transmitted bits that are received incorrectly. Essentially, the more incorrect bits, the greater the impact on signal quality. Bit error rate is an effective indicator of full end-to-end performance because it encompasses the receiver and transmitter as well as the media between them.

$$BER = \frac{Total\ different\ bits}{total\ bits} x\ 100\% \tag{4}$$

The equation above shows us how to calculate BER by the percentage difference in bits between the encrypted text and the plaintext.

# 2.6. Character Error Rate (CER)

Character Error Rate is to calculate percentage levels the accuracy of an encryption results by matching and comparing characters (letters, numbers, symbols). plaintext has compared to plaintext that is added or changed with a lower presentation.

$$CER = \frac{Total\ different\ bits\ from\ plaintext\ and\ ciphertext}{total\ characters} x\ 100\% \tag{5}$$

The equation above shows us how to calculate CER by the percentage of character differences between the generated encrypted text and the expected text.

#### 2.7. Entropy

Entropy is a measure of the amount of information in a message or text and is expressed in units of bits. Entropy is used to estimate the average number of bits to encode each element of the message. The higher the number of bits of entropy, the greater the complexity and the level of security in a cryptographic system. In general, entropy can be calculated using equation (6).

$$H(X) = -\sum_{i=1}^{n} a_i^2 log(p(S_i))$$
(6)

Where:

X is message

Si is the ith symbol in the message p(S) = the chance of occurrence of S i a is number of occurrences of S i

# 2.8. Proposed Method

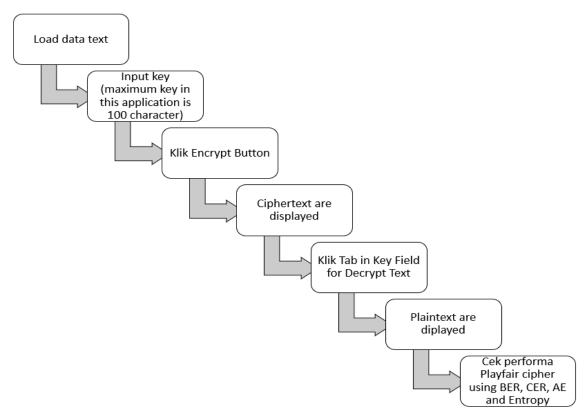


Figure 3. Proposed method based on Playfair Cipher

## 3. RESULTS AND DISCUSSION

This research is implemented using the Java programming language with JavaFX as its user interface, and it is executed using the IntelliJ IDEA software. The following is the GUI display when it is run, as shown in Figure 4.

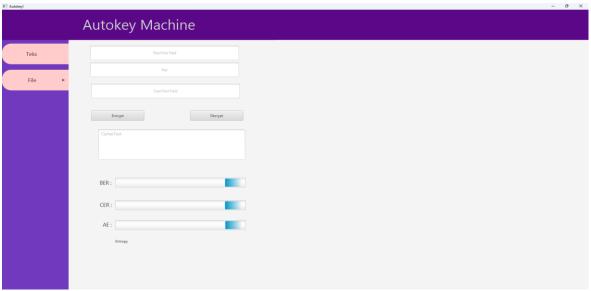


Figure 4. Display result

Figure 4 shows the initial view before inputting the plaintext and key; the Avalanche Effect, Character Error Rate, Bit Error Rate and Entropy item is not yet given results because the empty fields in the plaintext, key, and ciphertext and need to be filled in and entered first. After entering the plaintext and key/ciphertext in the provided columns, as shown in the display in Figure 4, and pressing the encrypt/decrypt button, the display will be complete. Once data security has been implemented by entering the plaintext and key/ciphertext, it is ensured that when executed by pressing the encryption/decryption button, the output results will be displayed as shown in Figure 5.

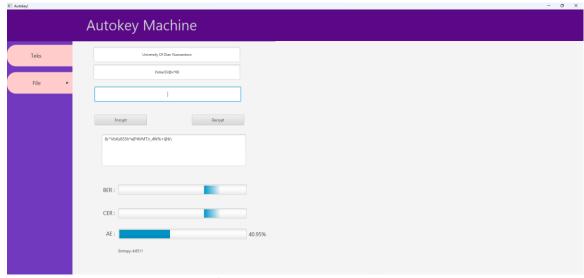


Figure 5. Encrypt Output Result

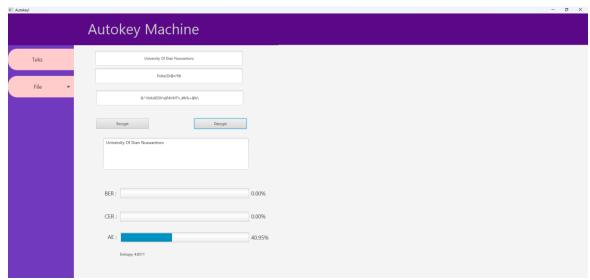


Figure 6. Decrypt Output Result

Figure 5 shows the encrypt output results when entering the plaintext and key. Meanwhile Figure 6, the decryption output displays the results when entering the same key and alo the results of the ciphertext from Figure 5, including the original text, encrypted text, decrypted text, Bit Error Rate (BER), Character Error Rate (CER), Avalanche Effect (AE), and Entropy.

Table 1. Test Results with Autokey Method

Plaintext	Key	Ciphertext		
Universitas Dian Nuswantoro	Polke!Di@n*66	&^VbKs8S5P}6Z:Q[k4v8a"]~&)@		
Testing number 45 with Autokey Method	Polke!Di@n*66	%U``Oo,i/dwx{Co!!ex.^)nK,+@[ReeN*^)^n		
RickyDirgantara@mhs	Polke!Di@n*66	#YPW_E.\(Px+wCQ-YNt		
HansVal3r1An_Lenice	Polke!Di@n*66	xQ[_ <b1 3 k%u u[uif<="" td=""></b1 3>		
Autokey_Ricky_hans@mhs.dinus!ac	Polke!Di@n*66	qea[Qf>IrXm"00XNZYA2R4 n %FcmMI		
R1ckY_W1th_H4nS??</td <td>Polke!Di@n*66</td> <td>1/? I1}Iw ~~ux\$[?%@</td>	Polke!Di@n*66	1/? I1}Iw ~~ux\$[?%@		

Based on Table 1, it had been seen that the autokey method can be applied to ASCII text with a combination of letters, numbers, and symbols, resulting in encrypted text or ciphertext in the form of symbols based on the ASCII table. Next, tests were conducted for Avalanche Effect, BER, CER, and Entropy with results as shown in Table 2, Table 3, and Figure 7.

Table 2. Avalanche Test Results

Plaintext	Key	Avalanche Effect %
Universitas Dian Nuswantoro	Polke!Di@n*66	43%
Testing number 45 with Autokey Method	Polke!Di@n*66	39%
RickyDirgantara@mhs	Polke!Di@n*66	42%
HansVal3r1An_Lenice	Polke!Di@n*66	37%
Autokey_Ricky_hans@mhs.dinus!ac	Polke!Di@n*66	38%
R1ckY_W1th_H4nS??</td <td>Polke!Di@n*66</td> <td>36%</td>	Polke!Di@n*66	36%

Based on Table 2 above, the average Avalanche Effect value from the 6 conducted tests is 39%. This means that the Avalanche Effect generated using the Autokey method is considered quite good. The use of ASCII text with a range of letters, numbers, and symbols enhances the level of information security derived from the use of the Autokey method. However, the Avalanche Effect value is considered quite good depending on the context and desired security requirements. In cryptography, the higher the avalanche effect is the better it gets. That means a small change in the original text or key should result in a significant change in the encrypted text.

Plaintext	Key	Bit Error Rate %	Character Error Rate %
Universitas Dian Nuswantoro	Polke!Di@n*66	0%	0%
Testing number 45 with Autokey Method	Polke!Di@n*66	0%	0%
RickyDirgantara@mhs	Polke!Di@n*66	0%	0%
HansVal3r1An_Lenice	Polke!Di@n*66	0%	0%
Autokey_Ricky_hans@mhs.dinus!ac	Polke!Di@n*66	0%	0%
R1ckY_W1th_H4nS??</td <td>Polke!Di@n*66</td> <td>0%</td> <td>0%</td>	Polke!Di@n*66	0%	0%

Table 3 Bit Error Rate (BER) and Character Error Rate (CER) Test Result

Table 3 shows us the Bit Error Rate and Character Error Rate values. The BER and CER values for each test are 0, which indicates that the encryption test has a high level of accuracy. Tests with high BER and CER values have lower CER values which indicates that the resulting ciphertext is better along with lower bit and character error rates. When evaluating algorithms for encryption or decryption, it is very important to pay attention and understand about the Character Error Rate and Bit Error Rate values. These values will show us how well the algorithm can produce precise and accurate encrypted text.

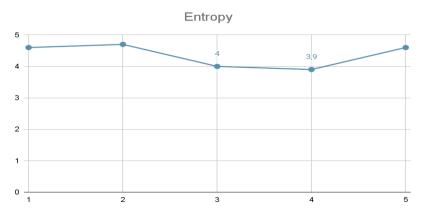


Figure 7. Entropy Test Result

Figure 7 shown that the average of the entropy test result obtained is 4 and if we look at the entropy value, it may be reasonable for randomness, but keep in mind that this value still depends on you or the final assessment set in the system. We now can understand if the high entropy results can be obtained if the text has a lot of long texts, numbers and symbols or complex randomness, and for low entropy results are text that has few short letters, numbers, and symbols.

### 4. CONCLUSION

This research was created using the Autokey Algorithm and the programming language used was Java and the library used from Java was JavaFX. The GUI that we created in this research has several features that have been created such as input plaintext fields, key fields, and ciphertext fields, other features are the results of input created such as ciphertext (results from plaintext encryption) and calculations that include things like Avalanche Effect, Character Error rate, Bit Error Rate, and Entropy Calculation. Apart from that, the IDE used to help run the GUI is Intellij IDE from Jetbrains. The purpose of this research is that the cryptography implementation created can be easily understood by users who use it. We hope that by making this research we can understand that cryptography is no longer used only for writing or completing code, but has developed into various functions such as encryption or decryption for data security, identity and message authentication, digital signatures and certificates, key exchange protocols, digital money, and so on.

### REFERENCES

- [1] Md. S. Hossain Biswas *et al.*, "A systematic study on classical cryptographic cypher in order to design a smallest cipher," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 12, p. p9662, Dec. 2019, doi: 10.29322/ijsrp.9.12.2019.p9662.
- [2] R. D. Ainul, S. Wibowo, and I. Zaini, "An Encrypted QR Code Using Layered Numeral Calculation for Low Powered Devices," in *Proceedings of the 4th International Conference on Informatics*,

*Technology andEngineering 2023 (InCITE 2023)*, 2023, pp. 607–617. doi: 10.2991/978-94-6463-288-0\_50.

- [3] S. M. Suhael, Z. A. Ahmed, and A. J. Hussain, "Proposed Hybrid Cryptosystems Based on Modifications of Playfair Cipher and RSA Cryptosystem," *Baghdad Science Journal*, May 2023, doi: 10.21123/bsj.2023.8361.
- [4] J. P. Sermeno, K. A. S. Secugal, and N. E. Mistio, "Modified Vigenere cryptosystem: An integrated data encryption module for learning management system," *International Journal of Applied Science and Engineering*, vol. 18, no. 4(Special Issue), pp. 1–10, 2021, doi: 10.6703/IJASE.202106\_18(4).003.
- [5] O. F. AbdelWahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, and A. A. M. Khalaf, "Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data," *Procedia Comput Sci*, vol. 182, pp. 5–12, 2021, doi: 10.1016/j.procs.2021.02.002.
- [6] K. R. Ilaga and C. A. Sari, "Analysis of Secure Image Crypto-Stegano Based on Electronic Code Book and Least Significant Bit," *Journal of Applied Intelligent System*, vol. 3, no. 1, pp. 28–38, 2018.
- [7] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2341–2352, Nov. 2022, doi: 10.1080/09720529.2020.1838744.
- [8] C. A. Sari, M. H. Dzaki, E. H. Rachmawanto, R. R. Ali, and M. Doheir, "High PSNR Using Fibonacci Sequences in Classical Cryptography and Steganography Using LSB," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 568–580, 2023, doi: 10.22266/ijies2023.0831.46.
- [9] I. Gede, A. Putra Dewangga, T. W. Purboyo, and R. A. Nugrahaeni, "A New Approach of Data Hiding in BMP Image Using LSB Steganography and Caesar Vigenere Cipher Cryptography," 2017. [Online]. Available: http://www.ripublication.com
- [10] L. B. Handoko and C. Umam, "Data Security Using Color Image Based on Beaufort Cipher, Column Transposition and Least Significant Bit (LSB)id 2 \*Corresponding author," 2023.
- [11] T. M. Aung, H. H. Naing, and N. N. Hla, "A complex transformation of monoalphabetic cipher to polyalphabetic cipher: (Vigenère-Affine Cipher)," *Int J Mach Learn Comput*, vol. 9, no. 3, pp. 296–303, Jun. 2019, doi: 10.18178/ijmlc.2019.9.3.801.
- [12] L. Budi Handoko and A. Rizqy, "File Cryptography Optimization Based on Vigenere Cipher and Advanced Encryption Standard (AES)," 2023.
- [13] S. K. Lakshmanan, L. Shakkeera, and V. Pandimurugan, "Efficient Auto key based Encryption and Decryption using GICK and GDCK methods," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1102–1106. doi: 10.1109/ICISS49785.2020.9316114.
- [14] E. Y. Purba, S. Efendi, P. Sirait, and P. Sihombing, "Collaboration of RSA Algorithm Using EM2B Key with Word Auto Key Encryption Cryptography Method in Encryption of SQL Plaintext Database," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Sep. 2019. doi: 10.1088/1742-6596/1230/1/012009.
- [15] B. Deepa, V. Maheswari, and V. Balaji, "An Efficient Cryptosystem Using Playfair Cipher Together with Graph Labeling Techniques," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1964/2/022016.
- [16] P. Pristiwanto, H. Sunandar, and B. Nadeak, "Analysis and Implementation of PlayFair Chipper Algorithm in Text Data Encoding Process," *Jurnal Info Sains: Informatikan dan Sains*, vol. 1, no. 1, pp. 19–23, 2020, [Online]. Available: http://ejournal.seaninstitute.or.id/index.php/InfoSains
- [17] M. S. Yousif, R. K. Salih, and N. M. G. Alsaidi, "A new modified playfair cipher," in *AIP Conference Proceedings*, American Institute of Physics Inc., Apr. 2019. doi: 10.1063/1.5095132.
- [18] N. A. Kako, H. T. Sadeeq, and A. R. Abrahim, "New symmetric key cipher capable of digraph to single letter conversion utilizing binary system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 1028–1034, May 2020, doi: 10.11591/IJEECS.V18.I2.PP1028-1034.
- [19] A. C. Licayan, B. D. Gerardo, and A. A. Hernandez, "Enhancing Playfair Cipher using Seed Based Color Substitution," in *Proceedings*, 2020 16th IEEE International Colloquium on Signal Processing & its Application (CSPA 2020), 2020, pp. 242–246.
- [20] P. Pristiwanto, H. Sunandar, and B. Nadeak, "Analysis and Implementation of PlayFair Chipper Algorithm in Text Data Encoding Process," *Jurnal Info Sains: Informatikan dan Sains*, vol. 1, no. 2, pp. 19–23, 2020, [Online]. Available: http://ejournal.seaninstitute.or.id/index.php/InfoSains

[21] S. M. Suhael, Z. A. Ahmed, and A. J. Hussain, "Proposed Hybrid Cryptosystems Based on Modifications of Playfair Cipher and RSA Cryptosystem," *Baghdad Science Journal*, May 2023, doi: 10.21123/bsj.2023.8361.

- [22] D. Babuc, "An Authentic Algorithm for Ciphering and Deciphering Called Latin Djokovic," *International Journal of Computer and Information Engineering*, pp. 17–10, 2023, [Online]. Available: https://www.researchgate.net/publication/374589454
- [23] E. Hari Rachmawanto *et al.*, "Testing Data Security Using a Vigenere Cipher Based on the QR Code," *Computer Network, Computing, Electronics, and Control Journal*, vol. 8, no. 4, pp. 701–708, 2023, doi: 10.22219/kinetik.v8i4.1734.
- [24] Permana langgeng wicaksono ellwid putra, C. A. Sari, and F. O. Isinkaye, "Secure Text Encryption for Iot Communication Using Affine Cipher And Diffie-Hellman Key Distribution On Arduino Atmega2560 IotT Devices," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 849–855, Aug. 2023, doi: 10.52436/1.jutif.2023.4.4.1129.