IMPLEMENTATION OF RC4 AND BASE64 COMBINATION ALGORITHM TO SECURE CLIENT DATABASE OF PT INFOKES

Nurhikmah Taliasih[1], Irawan Afrianto[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur No. 102 – 116 Bandung
E-mail : ntaliasih@gmail.com[1], irawan.afrianto@email.unikom.ac.id[2]

ABSTRACT

PT Infokes Indonesia is a company engaged in software development that focuses on health services. At present, PT Infokes database transactions with clients are carried out conventionally by sending databases that are stored in secondary storage via courier expeditions. This causes the database to be easily read and feared to be a gap for information leakage that is private to the public. In this study using the RC4 cryptographic algorithm that has advantages in speed and efficiency in storing data from the encryption results. RC4 cryptography used, combined with Base64 to add encoding after encryption. The results of the black box and white box testing in this study indicate that the implementation of the RC4 cryptographic algorithm and the Base64 algorithm are going well. Furthermore, test results using Wireshark indicate that the database transaction path has been secured with the HTTPS protocol and the database is encrypted. The results of testing by cryptanalysis using CrypTool on the combination of the RC4 and Base64 algorithms can not produce the original database, so the combination of the RC4 and Base64 algorithms has a better security level to secure the database compared to when using only the RC4 algorithm.

Keywords: Data Security, Cryptography, RC4, Base64, Database, Infokes

1. INTRODUCTION

PT Infokes Indonesia, which focuses on developing online integrated health information technology products and solutions in Indonesia, has produced several products including ePuskesmas, eHospital, eClinic, and others. In PT Infokes the client database archive is stored in secondary storage media. In addition, database transactions with clients are carried out conventionally, namely by sending databases that have been stored in secondary storage media through courier expeditions. This causes the database to be read easily by others and is feared to be a gap for information leakage that is privacy and confidential to the public. Although there has never been a case at PT Infokes, such information leakage has occurred , for example at PT Pindad [1] and Mangkubumi Sector Police Station [2] .

Cryptographic technique is an alternative solution that can be used in information security [1] [2] [3] [4] . The RC4 algorithm was chosen because it has advantages in data processing speeds [5] even reaching 10 times faster than DES [6] . In addition, RC4 has a high level of efficiency both in the storage of data in the database , because the result of encryption generated at jumlahn yes d ith original character [7] and on the RC 4, if the insert to be encrypted contains said repeatedly it will shortly ghasilkan ciphertext random [8] . Even so, the RC4 algorithm has a sensitivity to Bit Flipping Attack or BFA [9] and cryptanalysis which can find out the original message from an analysis of keys that might be used [10] . So that in this effort an effort was made to improve the performance of the RC4 algorithm by adding combinations using the Base64 algorithm to secure database transactions with clients.

1.1 Database

Generally, a database means a collection of interrelated data. Practically, the database can be considered as a structured data compiler that is stored in a reminder media (hard disk) whose purpose is so that the data can be accessed easily and quickly [11] .

1.2 Kriptografi

Cryptographic technique is the science that relies on mathematical techniques to deal with information security such as confidentiality, data integrity and entity authentication [12] .

The original message or data before encrypting is called the plaintext . While messages that have been randomized are called ciphertext . The process of converting plaintext to ciphertext is called encryption, while the process of converting ciphertext back to plaintext is called decryption [13] .

Figure 1. Encrypt – Decrypt Process [13]

1.3 RC4

The RC4 password system developed by Ronald Rivest in 1987 is the most widely used stream cipher algorithm . For example in the SSL / TLS protocol. RC4 is a byte -oriented stream cipher . Enter the RC4 encryption algorithm as a byte , then perform an XOR operation with a key byte , and generate a password byte [6] [12] .

Encryption process equation:

$ci = pi \oplus ki$ (1)

Decryption process equation:

$pi = ci \oplus ki$ (2)

1.4 Base64

Base64 algorithm is an algorithm for encoding and decoding . The purpose of the e ncoding is to change the form or format of the data. Base64 algorithms transform the data into a format ASCII which is based on the basic number 64 or can be regarded as one of the methods used to perform encoding (encryption) of the binary data. The characters produced in the Base64 transformation consist of A ... Z, a ... z and 0..9, and are added with the last two symbols which are + and / and one character equal to (=) used for adjustment and fulfill binary data or the term referred to as padding [4] [14].

Table 1. Base64 Index Tables [4]

Index (6 bit data)	Char *Encoding* Base64	Index (6 bit data)	Char *Encoding* Base64	Index (6 bit data)	Char *Encoding* Base64	Index (6 bit data)	Char *Encoding* Base64
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						pad	=

1.5 Internet

The internet is a group or collection of millions of computers. The use of the internet makes it possible to obtain information from computers in the group assuming that the computer owner grants access permission. To get information, a set of protocols must be used, which is a set of rules that establish how information can be sent and received [15] .

1.6 HTTPS *(HyperText Transfer Protocol)*

HTTPS is HTTP which uses SSL (Secure Socket Layer). SSL is an encryption protocol that is invoked via a web server using HTTPS. The use of SSL protocol on a network is very important to secure data in a network connection . SSL is a type of sockets communications that is between the transmission control protocol/internet protocol (TCP / IP) and the application layer [16] .

1.7 PHP

PHP is generally known as a script programming language that creates HTML documents on the fly that are executed on a web server , HTML documents generated from an application are not HTML documents created using a text editor or HTML editor. PHP is also known as a server side programming language [17] .

2. RESEARCH CONTENT

2.1 System Overview

The system to be built in this study is a security system with a combination of cryptographic algorithms and web-based encoding . Where the Data and Infrastructure division will create accounts for clients according to the order letter and the account data will be sent via email. Then the client can make a database request through the system. The Data and Infrastructure Division will upload the database on demand and encrypt it through a system that has been built. The encryption process is carried out with the RC4 cryptographic algorithm and Base64 encoding . Then the encrypted database or cipher encoded database will be stored in the database server . Encoded cipher database is then downloadable client to be decrypted files into database native to enter the secret key that is sent via email. The secret key is sent in the form of an encoded key because it has been changed before using the Base64 encoding algorithm.

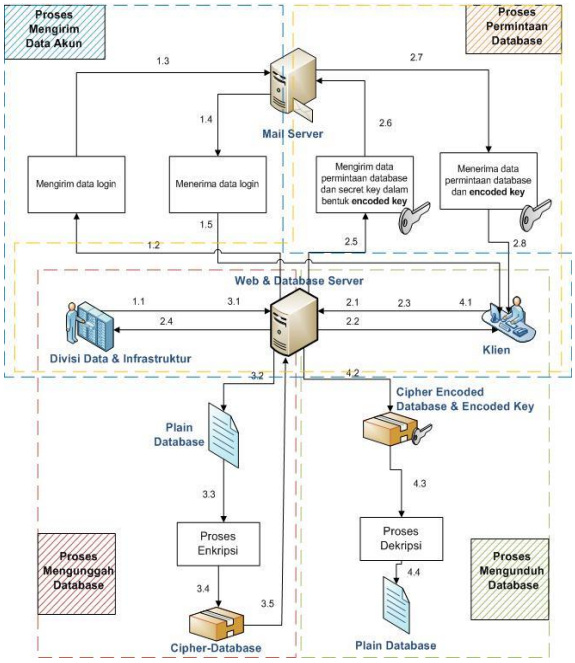


Figure 2. System Overview

2.2 Algorithm Analysis

In this study the algorithm used is the RC4 cryptographic algorithm and the Base64 algorithm. The algorithm analysis stage is used to find out the process sequence of the algorithm used so that it can be implemented into the system being built.

2.2.1 Analysis of Key Generation RC4

The key generation process is carried out before encrypting or decrypting the RC4 algorithm. RC4, which is part of the type of stream cipher, has the same method as the general concept of generating keystream on stream ciphers . Keystream is generated by the keystream generator and then XOR is done between the key and the plaintext . The RC4 algorithm uses an S-box (Substitution-box) with an array of 256 bytes measuring 16 x 16 as a keystream generator .

The key generation process in RC4 is as follows:

1) S-Box initialization

First, the initials i S-Box with a length of 256 bytes , with S [0] = 0, S [1] = 1, S [2] = 2, ..., S [255] = 255 so that the array S becomes:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

2) Padding the key K so that the key length K = 256

Initialization 256 byte key array K. Let secret key is: **ifk** the key to meet the entire array k and change the format into ASCII. K [0] = i = 105, K [1] = f = 102, K [2] = k = 107 , ..., [255] = i = 105.

So the array K becomes:

105	102	107	105	102	107	105	102	107	105	102	107	105	102	107	105
102	107	105	102	107	105	102	107	105	102	107	105	102	107	105	102
107	105	102	107	105	102	107	105	102	107	105	102	107	105	102	107
105	102	107	105	102	107	105	102	107	105	102	107	105	102	107	105
102	107	105	102	107	105	102	107	105	102	107	105	102	107	105	102
107	105	102	107	105	102	107	105	102	107	105	102	107	105	102	107
105	102	107	105	102	107	105	102	107	105	102	107	105	102	107	105
102	107	105	102	107	105	102	107	105	102	107	105	102	107	105	102
107	105	102	107	105	102	107	105	102	107	105	102	107	105	102	107
105	102	107	105	102	107	105	102	107	105	102	107	105	102	107	105
102	107	105	102	107	105	102	107	105	102	107	105	102	107	105	102
107	105	102	107	105	102	107	105	102	107	105	102	107	105	102	107
105	102	107	105	102	107	105	102	107	105	102	107	105	102	107	105
102	107	105	102	107	105	102	107	105	102	107	105	102	107	105	102
107	105	102	107	105	102	107	105	102	107	105	102	107	105	102	107
105	102	107	105	102	107	105	102	107	105	102	107	105	102	107	105

3) Permutation for the S-Box

The S-Box permutation operation will produce an array inside the S-Box that is different from the previous sequence. In this operation the variables i and j are used to index the arrays S [i] and K [i]. In the beginning, i and j initialize with 0. This operation is performed 256 times by repeating the formula (j + S [i] + K [i]) mod 256 followed by exchanging S [i] with S [j].

for i = 0 to 255
j = (j + S[i] + K[i]) mod 256
swap S[i] and S[j]

With the algorithm as above, with the initial value i = 0, j = 0 and repeated until i = 255, will produce an S array as below:

1st Iteration:
i = 0, then
j = (j + S[i] + K[i]) mod 256
= (j + S[0] + K[0]) mod 256
= (0 + 0 + 105) mod 256
= 105

Swap S[0] dan S[105] to produce an array :

105	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	0	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

The iteration continues until the 256th iteration. The end of the iteration and swap will produce the following array :

105	97	61	102	92	108	242	125	210	165	194	42	0	18	139	176
121	44	226	161	17	8	65	1	215	23	55	135	201	227	162	239
122	112	140	85	12	46	250	83	225	177	231	134	218	211	59	4
53	51	155	131	101	164	116	156	74	224	133	93	86	149	228	5
117	178	234	22	157	75	254	127	90	16	172	111	77	196	64	118
179	203	57	56	251	123	202	252	186	39	245	174	28	50	151	113
34	98	37	153	79	24	119	130	146	76	144	173	212	190	114	10
63	147	115	246	30	168	132	15	232	189	126	214	197	160	206	120
171	222	198	19	248	62	185	217	88	229	240	100	71	60	243	237
29	110	163	109	205	89	26	191	148	27	183	2	45	166	142	167
95	182	106	124	20	58	3	6	223	187	209	181	43	145	141	66
169	67	47	87	72	244	170	193	82	200	73	188	103	68	216	255
192	221	54	154	159	184	249	220	199	235	40	99	152	21	204	38
136	70	158	36	49	78	208	104	81	207	230	96	32	137	52	238
236	94	107	241	219	253	129	233	9	69	143	91	31	128	80	7
41	180	35	175	150	84	33	213	14	25	195	13	48	11	247	138

4) Generating a random key

Next is the process of generating a key that is obtained randomly by performing calculation operations by the number of bytes of plaintext . The calculation is as follows:

1st random key generation
First initialize the value i = 0; j = 0
i = (i + 1) mod 256
= (0 + 1) mod 256
= 1 j = (j + S[i]) mod 256
= (0 + S[1]) mod 256

= (0 + 97) mod 256=97

So the result:
S[i]=S[1] and S[j]=S[97]

Then do the swap on the array results until the last iteration:
S[i]=S[97]=98 and S[j]=S[1]=97

t = (S[i] + S[j]) mod 256
= (98 + 97) mod 256
= 195

K = S[t] = S[195] = 154 (the value to be XOR with the 1st plaintext byte)

The random key generation operation continues until as many bytes of the plaintext are encrypted and will produce the following random key :

154	99	54	33	86	200	242	36	0	29	165	238	174	108	144	91
241	6	248	77	197	179	69	196	243	137	109	16	173	8	179	230
238	150	4	202	58	201	209	141	48	221	244	112	157	35	178	39
195	203	237	172	15	231	51	85	37	20	190	11	195	188	9	215
118	46	87	13	46	253	133	170	67	181	209	85	...			

2.2.2 Analysis of the RC4 Encryption Algorithm

The RC4 encryption algorithm operates in bytes , meaning XOR is performed every one byte plaintext with one byte key from the keystream. The flowchart of the RC4 encryption algorithm is as follows:

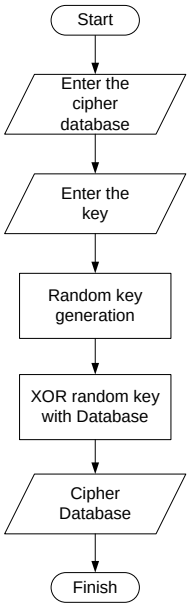


Figure 3. RC4 Encryption Algorithm Flowchart

From the previous key generation process a random key has been obtained , then an XOR operation between the bytes of the random key and the plaintext is shown in the following table:

Table 2. XOR Encryption Operation RC4

RANDOM KEY			PLAINTEXT			XOR		
Char	Des	Biner	Char	Des	Biner	Char	Des	Biner
š	154	10011010	-	45	00101101	·	183	10110111
c	99	01100011	-	45	00101101	N	78	01001110
6	54	00110110		32	00100000	[SYN]	22	00010110
!	33	00100001	M	77	01001101	l	108	01101100
V	86	01010110	y	121	01111001	ô	244	11110100
È	200	11001000	S	83	01010011	›	155	10011011
ò	242	11110010	Q	81	01010001	£	163	10100011
$	36	00100100	L	76	01001100	h	104	01101000
0	0	00000000		32	00100000		32	00100000
	29	00011101	d	100	01100100	y	121	01111001
¥	165	10100101	u	117	01110101	[DO2]	18	00010010
î	238	11101110	m	109	01101101	Š	138	10001010
®	174	10101110	p	112	01110000	Þ	222	11011110
1	108	01101100		32	00100000	!	33	00100001
□	144	10010000	1	49	00110001	¡	161	10100001
[	91	01011011	0	48	00110000	k	107	01101011
...								

From the XOR process between plaintex t and random key , the results of RC4 encryption are complete as shown:

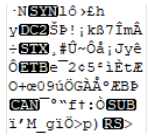


Figure 4. RC4 Encryption Results

2.2.3 Base64 Encoding Algorithm Analysis

Based on the results of RC4 encryption in **Figure 4**. RC4 Encryption Results , then encoding using Base64 algorithm with the following process steps:

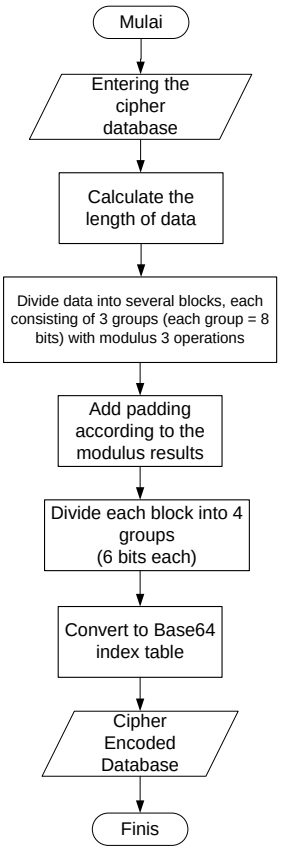


Figure 5. Base64 Encoding Algorithm Flowchart

In this study, the Base64 algorithm is operated after the RC4 algorithm so that the input is in the form of RC4 (cipher database) encryption results . Examples of implementation are as follows:

Base64 Encoding Algorithm Operations of the 1st byte group :

Cipher	·								N								[SYN]							
ASCII	183								78								22							
Bit	1	0	1	1	0	1	1	1	0	1	0	0	1	1	1	0	0	0	0	1	0	1	1	0
Index	45								52				56				22							
Cipher Encoded	t								0				4				W							

The operation is continued until the end to produce an encoded cipher database like **Figure 6.**

```
t04WbPSbo2ggeR
KK3iGha983zm3C
9wK4I9t+lOWhSn
nqlBdlrzKiNbLs
yHTGTyucMDn61k
fAxbrGQt4Yr7CT
ZoY60hrvkklfZ+
```

Figure 6. Base64 Encoding Results

2.2.4 Base64 Decoding Algorithm Analysis

If in the encoding process the data is grouped per 3 bytes = 24 bits (1 byte = 1 character), in the decoding process it is still grouped in blocks containing 24 data bits but, 1 character is represented by 6 data bits. The decoding process on Base64 is illustrated as shown below:

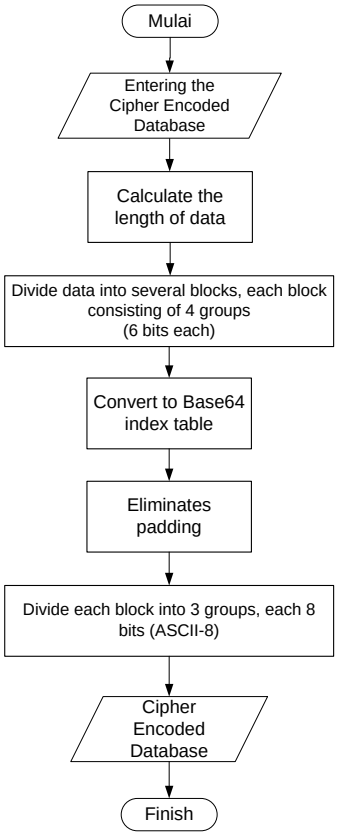


Figure 7. Base64 Decoding Algorithm Flowchart

Cipher encoded database that will be implemented according to the Base64 encoding results in **Figure 6**. So the process is as follows:

Base64 decoding algorithm operation of the 1st byte group :

Cipher Encoded	t								0				4				W							
Index	45								52				56				22							
Bit	1	0	1	1	0	1	1	1	0	1	0	0	1	1	1	0	0	0	0	1	0	1	1	1
ASCII	183								78								22							
Cipher	·								N								[SYN]							

The operation continues until the end to produce a cipher database as follows:

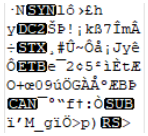


Figure 8. Base64 Decoding Results

2.2.5 Analysis of the RC4 Decryption Algorithm

RC4 uses the same decryption and encryption functions because the operation performed is just an XOR between random keys generated by plaintex t. So that in the decryption, the XOR process is carried out between random keys and the cipher database. The RC4 decryption process flowchart is as follows:

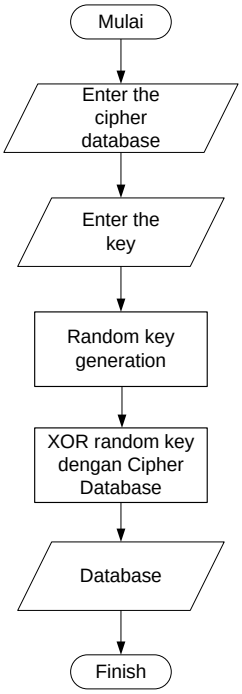


Figure 9. RC4 Decryption Algorithm Flowchart

In the decryption process generates the same key generation when encrypting. The XOR operations performed at the decryption stage are as follows:

Table 3. XOR Operation Decryption RC4

CIPHER DATABASE			*RANDOM KEY*			HASIL XOR		
Char	**Des**	**Bin**	**Char**	**Des**	**Bin**	**Char**	**Des**	**Bin**
-	183	10110111	š	154	10011010	-	45	00101101
N	78	01001110	c	99	01100011	-	45	00101101
	22	00010110	6	54	00110110		32	00100000
l	108	01101100	!	33	00100001	M	77	01001101
ô	244	11110100	V	86	01010110	y	121	01111001
›	155	10011011	È	200	11001000	S	83	01010011
£	163	10100011	ò	242	11110010	Q	81	01010001
h	104	01101000	$	36	00100100	L	76	01001100
	32	00100000	0	0	00000000		32	00100000
y	121	01111001		29	00011101	d	100	01100100
	18	00010010	¥	165	10100101	u	117	01110101
Š	138	10001010	î	238	11101110	m	109	01101101
Þ	222	11011110	®	174	10101110	p	112	01110000
!	33	00100001	l	108	01101100		32	00100000
¡	161	10100001	□	144	10010000	l	49	00110001
k	107	01101011	[	91	01011011	0	48	00110000
.....								

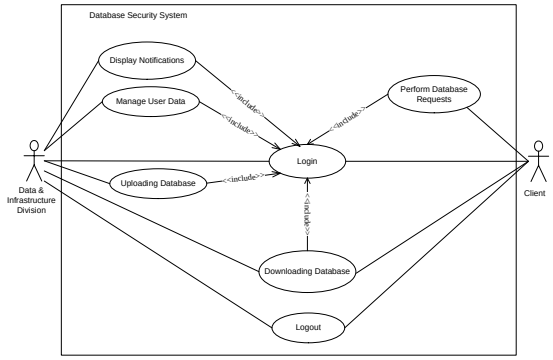
From the XOR process between the cipher database and the random key, it produces the plain database as follows:

```
-- MySQL
dump
10.16
Distrib
10.3.10-Ma
riaDB,
```

Figure 10. RC4 Decryption Results

2.3 Use Case Diagram

Use case diagrams are diagrams that illustrate interactions between systems built with users called actors as well as with other or external systems. The following **Figure 11** is a use case diagram consisting of two actors namely the Data & Infrastructure Division and the Client.



Gambar 1. *Use Case Diagram* Sistem Keamanan *Database*

2.4 Interface Desig

Interface design is the display design of a system that aims to describe the system to be built.

The design of the PT Infokes database security system interface for logging in is as follows:

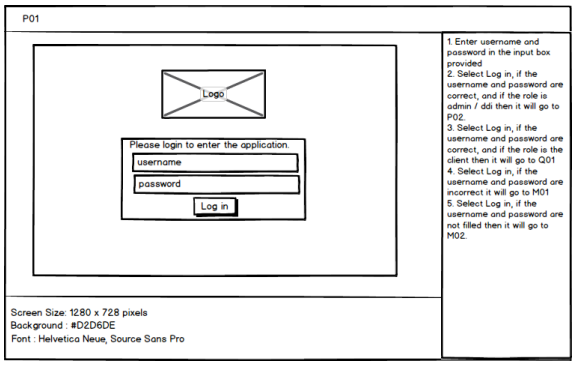


Figure 12. Login Interface Design

The interface design after logging in will be divided into two, according to the access rights. These are access rights for admin used by the Data & Infrastructure Division, and client access rights used by PT Infokes clients.

The design of the interface uploading the database on the Data & Infrastructure Division page is as follows:

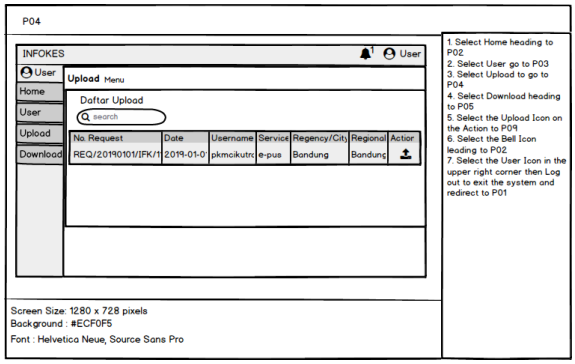


Figure 13. Uploading Database Interface Design

While on the client side, there is a database download interface design as shown below:

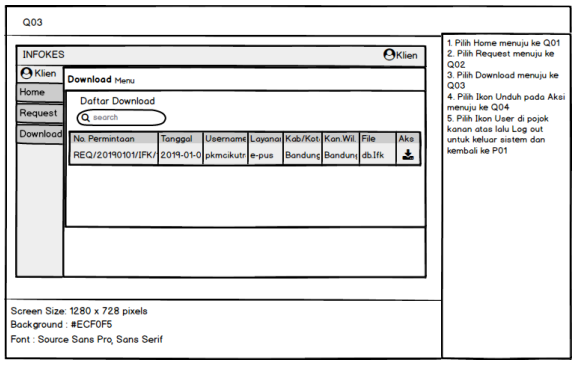


Figure 14. Downloading Database Interface Design

2.5 Testing

Tests were done there was some process that is black box testing, white box testing and database security testing by using CrypTool version 1.4.41 and Wireshark version 3.0.1.

2.5.1 Black Box Testing

Black box testing is a functional test phase carried out to determine the suitability of the functions in the system inputs and outputs along with the required specifications. The following is a black box test on the process of making a database request made by a client:

Table 2. Black Box Testing Performs Database Requests

Cases and Test Results (normal data)			
Data Input	**Which are expected**	**Observation**	**Conclusion**
Secret Key: if k	After clicking 'Process' the data is successfully saved and a *database* request notification appears on the Data & Infrastructure Division page and an *encoded key is* sent to the *email of the* client making the request.	After clicking 'Process' the data is successfully saved and a *database* request notification appears on the Data & Infrastructure Division page and an *encoded key is* sent to the *email of the* client making the request.	Be accepted
Cases and Test Results (blank data)			
Data Input	**Which are expected**	**Observation**	**Conclusion**
Secret Key : (not filled in)	Unable to make a request and the message "Data cannot be empty."	Unable to make a request and the message "Data cannot be empty."	Be accepted

2.5.2 White Box Testing

At this stage, the white box testing that will be used is a form of Base Path Testing. Tests carried out in the process of generating keys and encryption RC4.

The RC4 key generation and encryption test begins with checking the source code and then changing it to flowgraph . The white box testing phase is as follows:

1. Check the RC4 Key Generation and Encryption Source Code.

The following is the source code for the RC4 key generation and encryption process that the system runs..

Table 3. Source Code Key Generation and RC4 Encryption

Line	*Source Code*
1	$s = array();
2	for ($i = 0; $i < 256; $i++) {
3	$s[$i] = $i;
4	}
5	$j = 0;
6	for ($i = 0; $i < 256; $i++) {
7	$j = ($j + $s[$i] + ord($key[$i % strlen($key)])) % 256;
8	$x = $s[$i];
9	$s[$i] = $s[$j];
10	$s[$j] = $x;
11	}
12	$i = 0;
13	$j = 0;
14	$res = '';
15	for ($y = 0; $y < strlen($str); $y++) {
16	$i = ($i + 1) % 256;
17	$j = ($j + $s[$i]) % 256;
18	$x = $s[$i];
19	$s[$i] = $s[$j];
20	$s[$j] = $x;
21	$res .= $str[$y] ^ chr($s[($s[$i] + $s[$j]) % 256]);
22	}
23	return $res;

2. Make Key Generation and Encryption RC4 Flowgraph

The key generation and RC4 encryption flowgraph as shown in **Figure 15.**

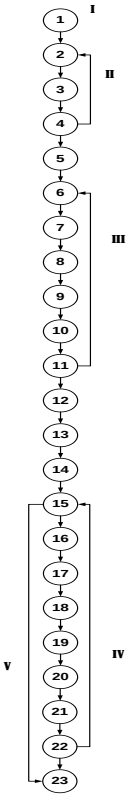


Figure 15. The key generation and RC4 encryption flowgraph

3. Calculating Cyclomatic complexity

a. Calculate cyclomatic complexity of the number of regions

V (G) = Region = 5

b. Calculate cyclomatic complexity of Edge and Node

V (G) = Edge - Node + 2

= 26 - 23 + 2 = 5

c. Calculate cyclomatic complexity from Predicate Code (P)

V (G) = P + 1

= 4 + 1 = 5

4. Determine the Independet Path

Path 1 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23

Path 2 = 1-2-3-4-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-15-16-17-18-19-20-21-22-23

Path 5 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-23

5. Creating a Graph Matrix of Key Generation and RC4 Encryption

In base path testing , a graph matrix is used which is a matrix of the number of nodes in a flowgraph . The graph matrix of key generation and RC4 encryption are as follows.

Table 4. The Graph Matrix of Key Generation and RC4 Encryption

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	n(E)-1
1		1																						0
2			1																					0
3				1																				0
4		1			1																			1
5						1																		0
6							1																	0
7								1																0
8									1															0
9										1														0
10											1													0
11						1						1												1
12													1											0
13														1										0
14															1									0
15																1							1	1
16																	1							0
17																		1						0
18																			1					0
19																				1				0
20																					1			0
21																						1		0
22															1								1	1
23																								0
Total																								4

6. Conclusion of Key Generation and RC4 Encryption Testing

Based on white box testing on key generation and RC4 encryption the same cyclomatic complexity value is 5. So that it can be concluded that the system is running well, because each test produces the same value.

2.5.3 Testing Using CrypTool

This test is carried out to open a database using CrypTool. For encryption security testing as a comparison, testing the results of RC4 encryption and testing the results of the combination of RC4 encryption and Base64 encoding.

Tests on the results of RC4 encryption using CrypTool, the results of cryptanalysis show that the data matches the original database or plain database as shown in **Figure 16.**

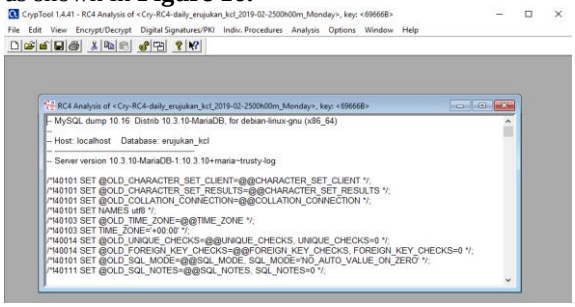


Figure 16. Cryptanalysis Results on RC4

While testing the results of RC4 encryption combined with Base64 encoding using CrypTool, the results of cryptanalysis showed that the data did not match the original database or plain database as shown in **Figure 17.**

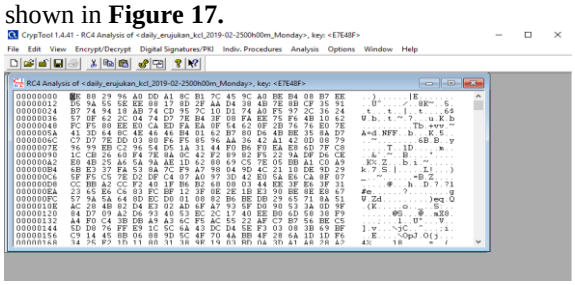


Figure 17. Cryptanalysis Results in a Combination of RC4 and Base64

2.5.4 Testing Using Wireshark

Database security testing using Wireshark is performed to find out whether the database being sent is safe or not. The test was carried out in two stages, namely testing when the web system that was built was still using the HTTP protocol and testing after using the HTTPS protocol. Each test is carried out by the process of uploading a database.

In testing with HTTP protocol , when the database upload process is carried out , the Wireshark application can easily monitor database delivery traffic . In this test it can be seen that the uploaded database can be read easily as shown in **Figure 18**.

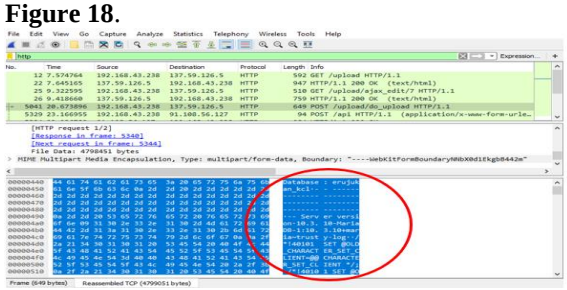


Figure 2. Database Security Test Results on the HTTP protocol

While the results of Wireshark monitoring on web systems that already use HTTPS protocols can be seen that the path from the user to the server has been secured with encryption. So the original

uploaded database cannot be read as shown in **Figure 19**.

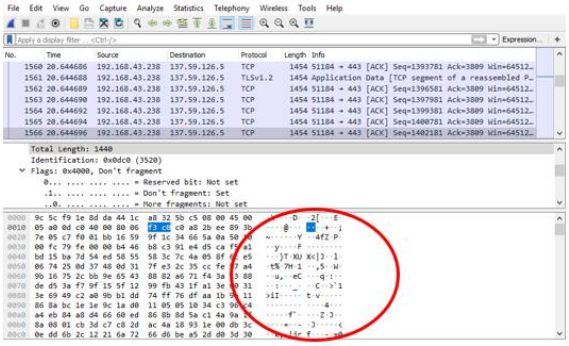


Figure 19. Database Security Test Results on the HTTPS Protocol

3. CLOSING

3.1 Conclusion

Based on the description of the analysis, design, implementation until testing, so that conclusions can be drawn as follows:

1. The system can secure the database with cryptography. Based on black box and white box testing, it shows that the upload process with RC4 encryption and Base64 encoding and download with Base64 decoding and RC4 decryption on the database are running well.
2. Based on testing using Wireshark, it shows that the results of monitoring data traffic cannot read the original database so the system can secure the database transaction process from the Data and Infrastructure Division to the client.
3. Based on testing using CrypTool, a combination of algorithms by adding Base64 to the RC4 cryptographic algorithm can improve the performance of RC4 from the cryptanalysis attacks carried out.

3.2 Suggestion

While suggestions for further development so that the system runs optimally are as follows:

1. Can do the upload process which includes the process of encrypting more than one database request made by the client.
2. It is expected that in the subsequent development of the algorithm used not only RC4 and Base64 but also developed using other algorithms.

REFERENCES

[1] A. D. Hidayat and I. Afrianto, "Sistem Kriptografi Citra Digital Pada Jaringan Intranet Menggunakan Metode Kombinasi Chaos Map dan Teknik Selektif," *Ultimatics*, vol. 9, pp. 59-66, 2017.

[2] M. Septian, "Penerapan Enkripsi Rabbit Stream Cipher Algorithm untuk Mengamankan File Bersifat Rahasia Di Polsek Mangkubmi Tasikmalaya," in *Skripsi*, Bandung, Universitas Komputer Indonesia, 2018.

[3] R. Aulia, A. Zakir and A. D. Purwanto, "Penerapan Kombinasi Algoritma Base64 dan ROT47 untuk Enkripsi Database Pasien Rumah Sakit Jiwa Prof. Dr. Muhammad Ildrem," *Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 2, pp. 146-151, 2018.

[4] F. W. Christanto, A. P. Rahangiar dan F. d. Fretes, "Penerapan Algoritma Gabungan RC4 dan Base64 pada Sistem Keamanan ECommerce," in *Seminar Nasional Aplikasi Teknologi Informasi 2012 (SNATI 2012)*, Yogyakarta, 2012.

[5] A. A. Okedola dan Y. N. Asafe, "RSA and RC4 Cryptosystem Performance Evaluation Using Image and Text File," *International Journal of Scientific & Engineering Research*, vol. 6, no. 5, pp. 289-294, 2015.

[6] W. K. Semarang, Memahami Model Enkripsi dan Security Data, Yogyakarta: Andi, 2013.

[7] S. A. Agusta and Triyani A. F. A, "Implementasi Algorithma Stream Cipher RC4 dalam Aplikasi Pendataan Alumni STMIK Amik Riau," *Inovtek Polbeng - Seri Informatika*, vol. 1, pp. 1-8, 2016.

[8] Y. Prayudi and I. Halik, "Studi dan Analisis ALgoritma Rivest Code 6 (RC6) dalam Enkripsi/Dekripsi Data," in *Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005)*, Yogyakarta, 2005.

[9] V. S. Arintamy, N. D. W. Cahyani and A. Mulyana, "Analisis Algorirtma RC4 Sebagai Metode Enkripsi WPA-PSK Pada Sistem Keamanan Jaringan Wireless LAN," *e-Proceeding of Engineering*, vol. 1, pp. 28-35, 2014.

[10] P. Xue , T. Li , H. Dong , C. Liu , W. Ma and S. Pei , "GB-RC4: Effective brute force attacks on RC4 algorithm using GPU," *Seventh International Green and Sustainable Computing Conference (IGSC)*, pp. 1-6, 2016.

[11] A. Kadir, Tuntutan Praktis Belajar Database Menggunakan MySQL, Yogyakarta: Andi, 2008.

[12] R. Sadikin, "Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java", Yogyakarta: Andi, 2012.

[13] D. Ariyus, "Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi", Yogyakarta: Andi, 2008.