Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID CRYPTOSYSTEM USING RC4+ AND MULTI-FACTOR RSA ALGORITHM FOR SECURING INSTANT MESSAGING

[1]DIAN RACHMAWATI, [2]SYAHRIL EFENDI, [3]ZIKRI AKMAL SANTOSO
[1,2,3]Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara,
Jl. Universitas No.9-A, Kampus USU, Medan 20155, Indonesia
E-mail: [1]dian.rachmawati@usu.ac.id

ABSTRACT

Communication and information exchange is the activity that apart from daily life. Instant messaging is one of the communication media by using internet protocol. In Many cases of tapping data and hacking information is a problem. Using a cryptography algorithm could secure the data information or message. Using a single algorithm of cryptography is still weak to send the data or information through the internet. Using a Hybrid cryptosystem scheme can solve this problem with the excellent implementation in instant messaging, that message will secure by RC4+ algorithm; this type is a symmetric algorithm that using one key, fast in the process but not secure enough, and the key of RC4+ algorithm will secure by Multi-factor RSA algorithm this type is Asymmetric algorithm has two key, but the process is slower than RC4+ but strong. The results showed that the hybrid combination of 2 algorithms cryptography successfully secures the message and is the key to ensuring that message, proving this scheme could secure faster than another combination of RC4+ and Multi-factor RSA.
Keywords: *RC4+, Multi-factor RSA, Cryptography, Hybrid, Instant Messaging*

1. INTRODUCTION

Communication and information exchange is the activity that apart from daily life. Instant messaging is one of the communication media. Instant messaging using internet protocol. Sending information with internet protocol is free and is not secure; many cases of tapping data and hacking information is a problem. Therefore the information and data need to secure by a cryptography algorithm. [1]

Cryptography is art or method for securing data and identify the original or plain data that have been secured. There are two processes that will be carried out using cryptography.[2]The first one is encryption, and the second is decryption. There are two types of a cryptography algorithm. The first is a symmetric algorithm, and the second is an Asymmetric algorithm. That two types have excellence in different cases. Using symmetric is not secure enough because only one key to encrypt and decrypt . Using Asymmetric is secure but slower than the symmetric algorithm.

RC4+ is a symmetric algorithm; this algorithm is a variant of the RC4 algorithm and claimed more secure and faster than RC4; Maitra announced this algorithm in 2008. And Multi-factor RSA is the one variant of RSA. More secure than RSA, this algorithm has slower encryption than RSA, and Multi-factor RSA must be used to encrypt and decrypt the small data [11].

Hybrid cryptography is one of the cryptography method, hybrid cryptography is a combination of two algorithms that the one algorithm using symmetric and another is using asymmetric, which is plain data or plain text will be encrypted by using a symmetric algorithm to generate ciphertext and the plain key of the symmetric algorithm will be encrypted by using asymmetric algorithm to generate cipher key , And the cipher key will be decrypted by asymmetric algorithm and the plain key that has been decrypted will be used to decrypt the cipher text.

This method is strong enough to secure the data than using single algorithm of symmetric and faster than using single algorithm of asymmetric end to end device.RC4+ will be used as symmetric algorithm to encrypt and decrypt the data or plain

text and Multi-factor RSA will be used to encrypt and decrypt the key of RC4+. By using this method RC4+ will be more secured as symmetric algorithm that using one key . and multi-factor RSA will be faster because this algorithm will be used to encrypt the key which is has small data or small string. This method is also implemented into a mobile-based application to make it easier for users to use it.

2. LITERATURE REVIEW

2.1. Instant Messaging

Instant Messaging (IM) technology is a type of online chat that offers real-time text delivery over the internet. Messenger LAN operates in a similar way via a local area network. Short messages are usually transmitted between two parties, when each user chooses to complete the thinking and selects "Send ". Some IM applications can use push technology to provide real-time text, which transmits message characters by character, as they are compiled. More advanced Instant Messaging can add file transfers, clickable hyperlinks, Voice over IP, or video chats.

Instant messaging is an application system that functions to exchange messages quickly and in real time through an intermediary internet network. Instant messaging itself can be used on several platforms, both desktop and mobile. On desktop platforms examples of applications that are commonly known are Yahoo IM, MIRc, etc. Whereas for example mobile is Line, Whatsapp, WeChat, KakaoTalk, etc.

2.2. Cryptography

Cryptography is a technique for securing data and ensuring the authenticity of data that has been secured. There are two processes that are carried out using cryptographic techniques namely the encryption and decryption process where the encryption process is making data unreadable by unauthorized parties such as passwords or encrypted text, whereas decryption is the opposite of encryption where this process is returned to be read by entitled party.[2]

Cryptographic algorithm based on key is divided into two, namely symmetric algorithm or also called conventional cryptographic algorithm is an algorithm that uses the same key for encryption and decryption process. The second is the asymmetric algorithm. Asymmetric cryptography is an algorithm that uses different keys for the encryption and decryption process. The encryption key can be distributed to the public and named as a public key while the decryption key is stored for own use and named as a private key. Therefore, this cryptography is also known as public key cryptography.

2.3. Hybrid Cryptography

Hybrid cryptography is a combination of two symmetric key encryption algorithms and asymmetric keys. Where the plaintext will be encrypted based on a symmetrical key to ciphertext. Then the key will be encrypted again with an asymmetric key which will eventually become cipher key. So that the cipher key and ciphertext will be sent together to the receiver. It will then be decrypted by the receiver using asymmetric private keys.

By using the hybrid cryptographic method the sender makes it possible to send messages more quickly with a small ciphertext. Because the plaintext is encrypted using a symmetric algorithm and the keys of the symmetric are secured by the asymmetric algorithm.[3]That can be seen in Figure 1 shows the hybrid cryptographic scheme.

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

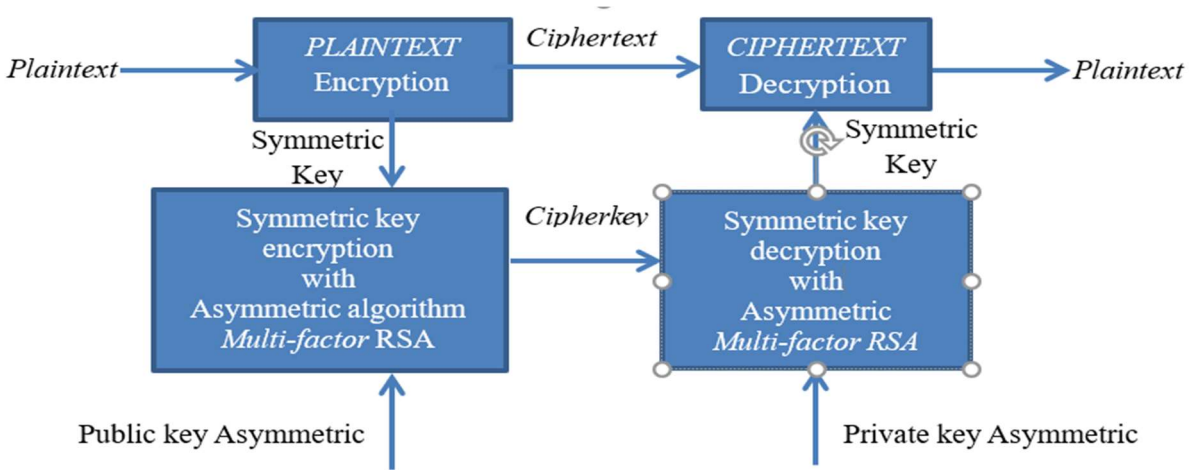

Figure 1. Hybrid Cryptographic Scheme

3. METHOD

3.1. RC4+

RC4 + algorithm is one of the symmetric key algorithm in the form of stream cipher. And is the development of the RC4 algorithm which is claimed that the RC4 + algorithm is faster and more efficient than the RC4 algorithm. [4] This algorithm was proposed by Paul and Maitra in 2008. The RC4 + algorithm has a structure similar to RC4 except that the RC4 + algorithm adds several operations to strengthen the cipher. In the mechanism of the RC4 + algorithm has a core part, namely Key Scheduling Algorithm (KSA) and Pseudo Random Generate Algorithm (PRGA).

KSA is the stage of the structure for encryption and decryption on the RC4 + algorithm. KSA on RC4 + is no different from KSA on RC4. KSA itself is a technique to modify the initialization array. Can be seen in figure 2.

```
for i from 0 to 255
   S[i] := i
End for
j := 0
for i from 0 to 255
   j := (j + S[i] + key[i mod keylength]) mod 256
   swap values of S[i] and S[j]
```

Figure 2. Ksa

PRGA is the next structural stage after KSA, PRGA on RC4 + is slightly different from RC4. It can be seen in Figure 3 which is the PRGA of the RC4 + algorithm.

```
while random:
i := i + 1
a := S[i]
j := j + a

Swap S[i] and S[j]   (b := S[j]; S[i] := b; S[j] := a)

c := S[i<<5 ⊕ j>>3] + S[j<<5 ⊕ i>>3]
output (S[a+b] + S[c⊕0xAA]) ⊕ S[j+b]
endwhile
```

Figure 3. PRGA

3.2. Multi-factor RSA

The Multi-factor Algorithm is the second variant of RSA which involves modification of the modulus structure in the previous Multi-Prime RSA variant. The RSA Multi-factor algorithm was proposed by Takagi and patented by Naito and Takagi (Boneh, 2002). Multi-Factor RSA has three components namely key generation, encryption and decryption.

The key generation pattern are

1. Determine b which is a lot of different prime numbers taken with the terms p1, p2, .. pb (b≥ 3)
2. Calculate $n \leftarrow \prod\_ (i = 1) \wedge b\ pi$
3. Calculate $\phi(n) \leftarrow \prod\_ (i = 1) \wedge b\ (pi - 1)$

4. Use the same e on a standard RSA public key, i.e. = 65537
5. Calculate $d = e - 1\ mod\ \phi\ (n)$, and make sure that e and $\phi\ (n)$ are relatively prime. That way we get public keys (n, e) and private keys (d, p1, p2, .. pb)

Encryption pattern is C= m^e mod n
Decryption patter is m= $\sum\ ai\ ai\ bi\ si=1\ (mod$ B)

3.3. Using RC4+ And Multi-factor RSA in hybrid scheme

In this scheme, RC4+ as symmetric algorithm will be used to encrypt the message and Multi-factor RSA as asymmetric algorithm will be used to encrypt the small key of RC4+ key. In Figure 4 , it appears that Bertold sent a message to Reiner. Bertold Plaintext will be encrypted with the RC4 + algorithm to produce ciphertext and RC4 + keys, then the key will be encrypted again with the Multi-factor RSA algorithm using Reiner's public key to produce a cipherkey. Ciphertext and cipherkey sent to Reiner. Cipherkey will be decrypted with Multi-factor RSA algorithm using privatekey Reiner to produce RC4 + keys, then Ciphertext will be decrypted using RC4 + keys to produce plaintext.

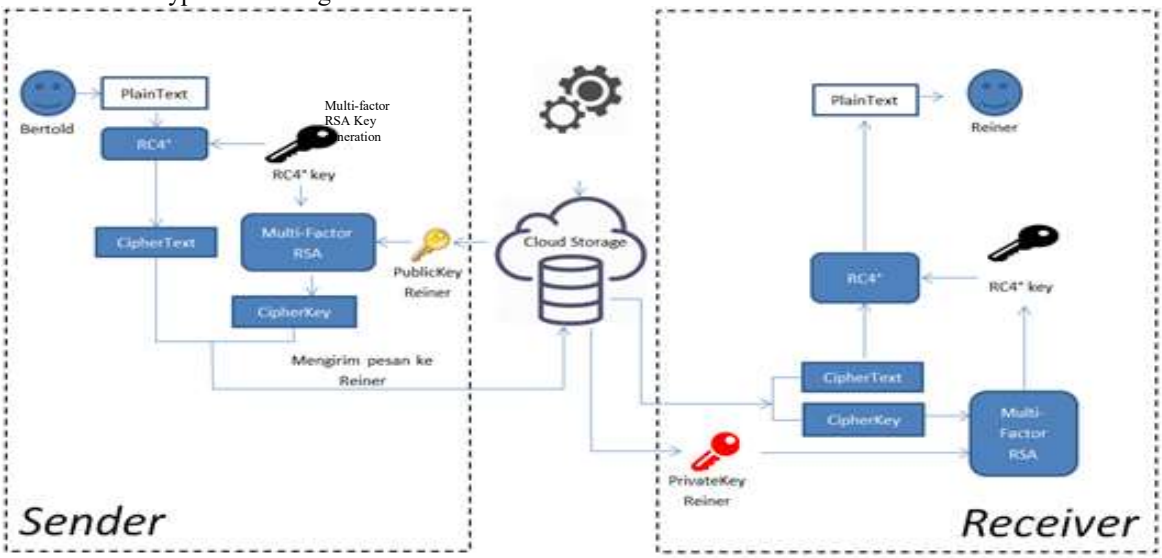

Figure 4. RC4+ And Multi-Factor RSA In Hybrid Scheme

4. RESULT AND DISCUSSION

4.1. Encrypt messages with RC4 +

The text / message sent is "hello" and the generated key is "ZTL", so:
plaintext = "haloo": {104,97,108,111,111} in ASCII
key = "ZTL": {90,84,76} in ASCII

4.1.1. Key Scheduling Algorithm (KSA) phase
Initialize key states from 0 to 255 so as to make 256 S [] array elements. Array can be seen in Table 1.

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1. S Array Elements

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										

After initial initialization, change the character on the plainkey string to ASCII value. key is "ZTL" so that {90,84,76} do initial initialization i = 0, j = 0 and key length (keylength) = 3.

- > i=0 and j=0
j = (j + S[i] + *key*[i mod *keylength*]) mod 256
j = (0+ S[0] + *key*[0 mod 3]) mod 256
j = (0 + 0 + *key*[0]) mod 256
j = (0 + 0 + Z) mod 256
j = (0 + 0 + 90) mod 256
j = 90 mod 256
j = 90
then swap S [i] and S [j] are the initial values of S [0] = 0 and S [90] = 90 to S [0] = 90 and S [0] = 90.

- > i=1 and j=90
j = (j + S[i] + *key*[i mod *keylength*]) mod 256
j = (90+ S[1] + *key*[1 mod 3]) mod 256
j = (90 + 1 + *key*[1]) mod 256
j = (90 + 1 + T) mod 256
j = (90 + 1 + 84) mod 256
j = 175 mod 256
j = 175
then swap S [i] and S [j] are the initial values of S [1] = 1 and S [175] = 175 to S [175] = 1 and S [1] = 175.

- > i=2 and j=175
j = (j + S[i] + *key*[i mod *keylength*]) mod 256
j = (175+ S[2] + *key*[2 mod 3]) mod 256
j = (175 + 2 + *key*[2]) mod 256
j = (175 + 2 + L) mod 256
j = (175 + 2 + 76) mod 256
j = 253 mod 256
j = 253
then swap S [i] and S [j] are the initial values of S [2] = 2 and S [253] = 253 to S [253] = 2 and S [2] = 253.

- > i=3 dan j=253
j = (j + S[i] + *key*[i mod *keylength*]) mod 256
j = (253+ S[3] + *key*[3 mod 3]) mod 256
j = (253 + 3 + *key*[0]) mod 256
j = (253 + 3 + Z) mod 256
j = (253 + 3 + 90) mod 256
j = 346 mod 256
j = 90
then swap S [i] and S [j] are the initial values of S [3] = 3 and S [90] = 0 to S [90] = 3 and S [3] = 0.

do the calculation up to i = 255 and produce a new array S [] like table in figure 2 at the end of the KSA process.

Table 2. New Array S

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

4.1.2. PRGA phase

The string "haloo" for each character is changed to ASCII so that {104,97,108,111,111}. First do the Initiate value i = 0, j = 0

After that, following the process are :

For “h” character
Given that the values i = 0 and j = 0, initialise a = S [i] and b = S [j] then proceed to find the values i, j, a, b, c, z as follows.
i = (i + 1) mod 256
i = (0 + 1) mod 256
i = 1
a = S[i]
a = S[1]
a = 175
j=(j+a) mod 256
j=(0+175) mod 256
j=175
then swap the values of S [i] and S [j] in the following way.
S[1] =175
S[175]= 99
S[1]=99 dan S[175]=175
S[i]= 99 = b
S[j]= 175 = a
Then proceed with the calculation in c in the following way.
c = (S [((i≪5)⊕(j≫3))mod 256] + S [((j≪5)⊕(i≫3))mod 256])mod 256
c = (S [((1≪5)⊕(175≫3))mod 256] + S [((175≪5)⊕(1≫3))mod 256])mod 256
c = (S [((32)⊕(21))mod 256] + S [((5600)⊕(0))mod 256])mod 256
c = (S [(53)mod 256] + S [(5600)mod 256])mod 256
c = (S [53] + S [(224])mod 256
c = (194 + 87)mod 256
c = 281 mod 256 = 25

z = ((S [(a+b)mod 256]+S [((c⊕170)mod 256)]) ⊕ S [(j+b)mod 256])mod 256
z = ((S [(175+99)mod 256]+S [((25⊕170)mod 256)]) ⊕ S [(175+99)mod 256])mod 256
z = ((S [(274)mod 256]+S [((179)mod 256)]) ⊕ S [(274)mod 256])mod 256
z = ((S [18]+S [179]) ⊕ S [18])mod 256
z = (215+224) ⊕ 215)mod 256
z = (439 ⊕ 215) mod 256 = 96
then xor the Ascii of “h” with value of z
cipher[0]= 104 ⊕ 96 = **8**

then calculate for the following character a,l,o,o and the value will be {h,a,l,o,o} ={8,21,206,181,239}, the cipher text is { Ĉ,к,Î,µ,ï} after converted char in ASCII

4.2. Key Encryption of RC4+ with Multi-factor RSA

It is known that the public key obtained is (n = 38399, e = 65537) and the key to be encrypted is "ZTL" so to encrypt the key use C = m^e mod n, where m is each character taken on the string "ZTL" following the calculation.

For “Z” char
C= m^e mod n
C= 90^{65537} mod 38399
C= 38205
For “T” char
C= m^e mod n
C= 84^{65537} mod 38399
C= 21406
For “L” char
C= m^e mod n
C= 76^{65537} mod 38399
C= 13794

So the cipherkey is obtained, which is {38205 21406 13794}

So the Encryption results obtained get the results of ciphertext and cipherkey, namely:
Message: "ĈкÎµï" = {8,21,206,181,239} and the key is {38205 21406 13794} .

4.3. Key Decryption of RC4+ with Multi-factor RSA

Known cipherkey = {38205 21406 13794} with privatekey (d = 20753, p1 = 47, p2 = 43, p3 = 19)

- **For c=38205**

1. calculate di= d mod (pi-1), where ∑ 1 ≤ i ≤ b
   d1=20753 mod(47-1)=7
   d2=20753 mod(43-1)=5
   d3=20753 mod(19-1)= 17
2. Calculate CRT from $m_i = c^{di} (\text{mod } p_i)$, ∑ 1 ≤ i ≤ b
   m_1 = 38205^7 (mod 47) = 43
   m_2 = 38205^5 (mod 43) = 4
   m_3 = 38205^{17} (mod 19)= 14
3. Calculate B= $\prod_{i=1}^{k} p_i$
   B = p_1 x p_2 x p_3 = 38399
4. Calculate Bi= B/p_i, ∑ 1 ≤ i ≤ k
   B_1 = B/p_1 = 38399/47 = 817

$B_2 = B/p_2 = 38399/43 = 893$
$B_3 = B/p_3 = 38399/19 = 2021$

5. Calculate $Si = B_i - 1 (\text{mod } p_i)$, $\sum 1 \leq I \leq k$
$s_1=817^{-1}(\text{mod } 5)$
$s_1=34.\ 817 \text{ mod } 5 =1$
$s_1=34$
$s_2=893^{-1}(\text{mod } 7)$
$s_2=10.\ 893 \text{ mod } 7 =1$
$s_2=10$
$s_3=2021^{-1}(\text{mod } 11)$
$s_3=11.\ 2021 \text{ mod } 11 =1$
$s_3=11$
6. Calculate the value of $m= \sum ai\ ai\ bi\ si=1\ (mod\ B)$
$M= a_1.B_1.s_1+a_2.B_2.s_2+a_3.B_3.s_3 \text{ (mod } 385)$
$M=43 \times 817 \times 34 +4 \times 893 \times 30 + 14 \times 2021 \times 11 \text{ (mod } 38399)$
$=1194454 + 107160 + 311234 \text{ (mod } 38399)$
$= 1612848 \text{ (mod } 38399)$
7. M= 90 (in ASCII) convert to char "Z"

- **For c = 21406**

1. Calculate di= d mod (pi-1), dimana $\sum 1 \leq i \leq b$
d1=20753 mod(47-1)=7
d2=20753 mod(43-1)=5
d3=20753 mod(19-1)= 17
2. Calculate CRT from $m_i = c^{di} (\text{mod } p_i)$, $\sum 1 \leq i \leq b$
$m_1 = 21406^7 \ (\text{mod } 47) = 37$
$m_2 = 21406^5 \ (\text{mod } 43) = 41$
$m_3 = 21406^{17} (\text{mod } 19)= 8$
3. Calculate $B= \prod_{i=1}^{k} p_i$
$B = p_1 \times p_2 \times p_3 = 38399$
4. Calculate $Bi= B/p_i$, $\sum 1 \leq i \leq k$
$B_1 = B/p_1 = 38399/47 = 817$
$B_2 = B/p_2 = 38399/43 = 893$
$B_3 = B/p_3 = 38399/19 = 2021$
5. Calculate $Si = B_i - 1 (\text{mod } p_i)$, $\sum 1 \leq I \leq k$
$s_1=817^{-1}(\text{mod } 5)$
$s_1=34.\ 817 \text{ mod } 5 =1$
$s_1=34$
$s_2=893^{-1}(\text{mod } 7)$
$s_2=10.\ 893 \text{ mod } 7 =1$
$s_2=10$
$s_3=2021^{-1}(\text{mod } 11)$
$s_3=11.\ 2021 \text{ mod } 11 =1$
$s_3=11$
6. Calculate value of $m= \sum ai\ ai\ bi\ si=1\ (mod\ B)$
$M= a_1.B_1.s_1+a_2.B_2.s_2+a_3.B_3.s_3 \text{ (mod } 385)$
$M=37 \times 817 \times 34 +41 \times 893 \times 30 + 8 \times 2021 \times 11 \text{ (mod } 38399)$
$=1027786 + 1098390 + 177848 \text{ (mod } 38399)$
$= 2304024 \text{ (mod } 38399)$
7. M= 84 (dalam ASCII) convert to char "T"

- **For c = 13794**

1. Calculate di= d mod (pi-1), dimana $\sum 1 \leq i \leq b$
d1=20753 mod(47-1)=7
d2=20753 mod(43-1)=5
d3=20753 mod(19-1)= 17
2. Calculate CRT from $m_i = c^{di} (\text{mod } p_i)$, $\sum 1 \leq i \leq b$
$m_1 = 13794^7 \ (\text{mod } 47) = 29$
$m_2 = 13794^5 \ (\text{mod } 43) = 33$
$m_3 = 13794^{17} (\text{mod } 19)= 0$
3. Calculate $B= \prod_{i=1}^{k} p_i$
$B = p_1 \times p_2 \times p_3 = 38399$
4. Calculate $Bi= B/p_i$, $\sum 1 \leq i \leq k$
$B_1 = B/p_1 = 38399/47 = 817$
$B_2 = B/p_2 = 38399/43 = 893$
$B_3 = B/p_3 = 38399/19 = 2021$
5. Calculate $Si = B_i - 1 (\text{mod } p_i)$, $\sum 1 \leq I \leq k$
$s_1=817^{-1}(\text{mod } 5)$
$s_1=34.\ 817 \text{ mod } 5 =1$
$s_1=34$
$s_2=893^{-1}(\text{mod } 7)$
$s_2=10.\ 893 \text{ mod } 7 =1$
$s_2=10$
$s_3=2021^{-1}(\text{mod } 11)$
$s_3=11.\ 2021 \text{ mod } 11 =1$
$s_3=11$
6. Calculate nilai $m= \sum ai\ ai\ bi\ si=1\ (mod\ B)$
$M= a_1.B_1.s_1+a_2.B_2.s_2+a_3.B_3.s_3 \text{ (mod } 385)$
$M=29 \times 817 \times 34 +33 \times 893 \times 30 + 0 \times 2021 \times 11 \text{ (mod } 38399)$
$= 805562 + 884070 + 0 \text{ (mod } 38399)$
$= 1689632 \text{ (mod } 38399)$
7. M= 76 (dalam ASCII) convert to char "L"

So we get a cipherkey that is decrypted into plainkey, namely "ZTL"

4.4. Decrypt messages with RC4 +

After the key decryption stage is carried out with Multi-factor RSA, the key "ZTL" is used to decrypt the ciphertext "ĈκÎμï" = {8,21,206,181,239}, as follows.

Perform the KSA stage again so as to produce the final KSA array in the following table 3.

Table 3. Final Ksa Table

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

Then do the PRGA stage like encryption as follows. 8,21,206,181,239
ĈκÎμï
• For "Ĉ" = 8
Obtained z = 96,
Then XOR is the ciphertext string character value with z:
Plaintext [0] = 8 ⊕ 96 so it yields = 104 in the character "h"
• For "κ" = 21
Obtained z = 116,
Then XOR is the ciphertext string character value with z:
Plaintext [1] = 21 ⊕ 116 so it produces = 97 in the character "a"
• For "Î" = 206
Obtained z = 162,
Then XOR is the ciphertext string character value with z:
Plaintext [2] = 206 ⊕ 162 so that it yields = 108 in the "l" character
• For "μ" = 181
Obtained z = 218,
Then XOR is the ciphertext string character value with z:
Plaintext [3] = 181⊕ 218 so that it produces = 111 in the character "o"
• For "ï" = 239
Obtained z = 128,
Then XOR is the ciphertext string character value with z:
Plaintext [4] = 239 ⊕ 128 so it produces = 111 in the character "o"

4.5. Processing time

4.5.1. Key generation processing time

Table 4. Key Generation Processing Time

Count of digit	Processing time(ms)	Average (ms)
2 digit	30,60837 1,45358 7,71025	13,2574
3 digit	3207,23028 3012,20921 89,56706	2103,002183
4 digit	1754,53752 2002,67049 4694,58109	2817,263033

The results show that the greater the prime number generated the greater the time required to generate the key shown in the following graph on Table 4.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.5.2. Encryption

Table 5. The Encryption Process Time

Count of Character	Processing time (ms)	Average (r
5 character	65,09104 64,34419 59,98454	63,13
10 character	63,9394 63,17214 65,70952	64,27
50 character	86,27556 85,86932 81,47976	84,54
500 character	91,4003 78,48362 93,99021	87,95
2000 character	187,24414 106,28658 106,43099	133,3

Thus the results obtained that the longer the character, the greater the time needed to perform the encryption process is seen in the encryption process time on Table 5.

4.5.3. Decryption

Table 6. The Decryption Process Time

Count of Character	Processing time (ms)	Average (ms)
5 character	169719,0208 337321,0208 128255,0208	211765,02
10 character	494605,0208 209942,0208 314340,0208	339629,02
50 character	411241,0208 343338,0208 340166,0208	364915,02
500 character	3995165,021 4996664,021 4668969,021	4553599,3
2000 character	15705294,02 12544754,02 13121921,02	13790656,

Based on the values generated in the table, the longer the character to be decrypted, the more time needed to perform the decryption process is seen in the decryption process time on Table 6.

The illustration of the system is seen in Figure 5.

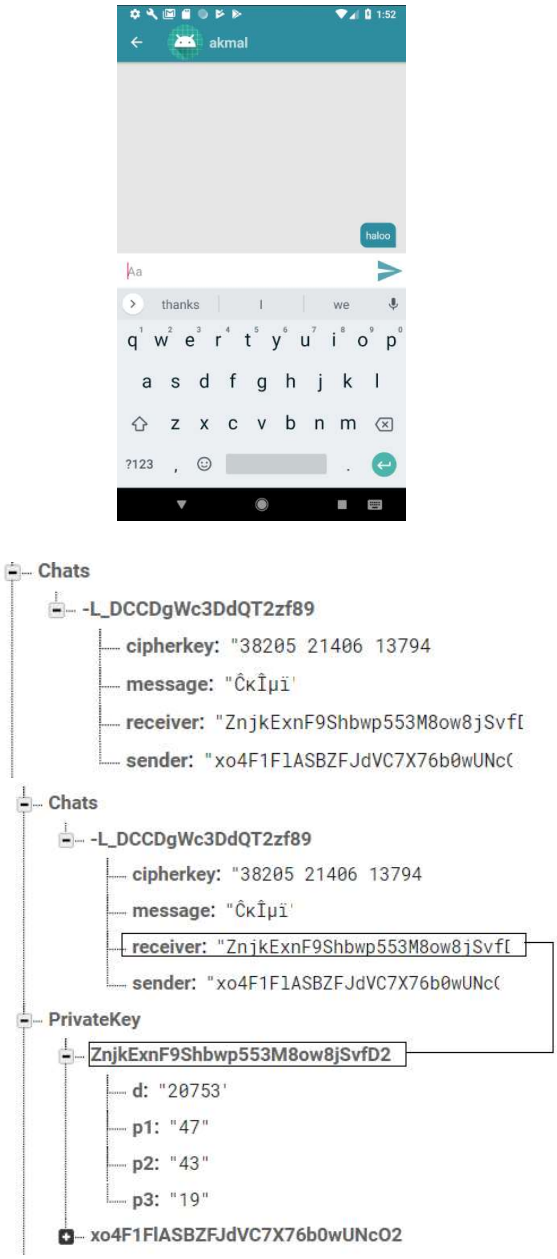

Figure 5. The Illustration Of The System

5. CONCLUSIONS

In this paper we got some result that concluded

The first is Using hybrid cryptography in the encryption and decryption process with the RC4 + algorithm as a symmetric algorithm and multi-factor algorithm as an asymmetric algorithm can return the message sent as before to the recipient. The Second Processing time on testing with both algorithms while getting speed comparison results on different characters, the number seen on the

test graph shows that processing time with character length is directly proportional. So it can be concluded that the longer the character, the longer the processing time generated by the algorithm.And The result of testing the comparison of processing time to the number of digits of prime numbers in the key generation results in that the more the number of digits of prime numbers, the greater the running time, so that the number of prime digits is proportional to the processing time. Future research hopes that a different prime generator method will be used or replace the concept of a hybrid cryptosystem with super encryption.

REFERENCES

[1] Muklison, Hadi, dan Setyawati. 2015. Security in *Instant Message* for android operating system. Jurnal EECCIS. Universitas Brawijaya Malang.Vol. 9 ,No.2 ,pp173-178.

[2] Rachmawati, D dan Lydia, M.S dan Siregar, W.A. 2018. *Hybrid Cryptosystem Implementation Using IDEA and Knapsack Algorithm for Message Security*. Journal of Physics: Conf. Vol.1090 , No.1 ,pp. 1-7

[3] Rachmawati, D dan Sharif, A dan Jaysilen dan Budiman M.A. 2018. *Hybrid Cryptosystem Using Tiny Encryption Algorithm and LUC Algorithm*. IOP Conference Series: Materials Science and Engineering. Vol.300 ,No.1, pp.1-7.

[4] Bau'dan, Rizki Akbar, F, Mawengkang, H dan Efendi S.2018. *Comparative analysis of* RC4+ *algorithm*, RC4 NGG *algorithm and* RC4 GGHN *algorithm on image file security*. IOP Conference Series: Materials Science and Engineering. Vol.420 , No.1. 1-9.

[5] Boneh, D. & Shacham, H. 2007. *Fast Variants of* RSA. Vol. 5, No. 1, pp. 1-9.

[6] Kapoor, Vivek & Yadav, Rahul. 2016. A Hybrid Cryptography Technique for Improving Network Security. International Journal of Computer Applications. Vol.141 ,No.11 , pp. 25-30.

[7] Cramer, Ronald dan Shoup, Victor. 2004. *Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack*. Aarhus University. Vol. 33, No. 2, pp.167 – 226.

[8] Ramadhani. 2018. *Three-Pass Protocol* Implementation with combination of RC4+ dan RC4A to securing document for Android.Skripsi.Universitas Sumatera Utara.

[9] Mollin R. 2007. *An Introduction to Cryptography* 2. Florida: Chapman & Hall/CRC.London,New York.

[10] Herzy , Fildza Mastura. 2018. Implementation of *Hybrid Cryptosystem* with Algorithm RC4+ *Cipher* and *Multiprime*-RSA for securing pdf file.Skripsi.Universitas Sumatera Utara

[11] M. I. Mihailescu and S. Loredana Nita, "Three-Factor Authentication Scheme Based on Searchable Encryption and Biometric Fingerprint," 2020 13th International Conference on Communications (COMM), 2020, pp. 139-144, doi: 10.1109/COMM48946.2020.9141956.