Journal of Al-Qadisiyah for Computer Science and Mathematics Vol. 15(1) 2023 , pp Comp. 189–196

University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

A Cryptosystem for Database Security Based on RC4 Algorithm

Saad A. Abdulameer *

College of Education for Women, Baghdad, University of Baghdad, Iraq.Email:saad@coeduw.uobaghdad.edu.iq

ARTICLE INFO

Received: 11 /01/2023
Rrevised form: 22 /02/2023
Accepted : 26 /02/2023
Available online: 31 /03/2023

Keywords:

Cryptosystem

Database

Security

RC4 algorithm

Encryption

Decryption

ABSTRACT

Because of vulnerable threats and attacks against database during transmission from sender to receiver, which is one of the most global security concerns of network users, a lightweight cryptosystem using Rivest Cipher 4 (RC4) algorithm is proposed. This cryptosystem maintains data privacy by performing encryption of data in cipher form and transfers it over the network and again performing decryption to original data. Hens, ciphers represent encapsulating system for database tables.

MSC..

https://doi.org/10.29304/jqcm.2023.15.1.1195

1. Main text

Here introduce the paper, and put a nomenclature if necessary, in a box with the same font size as the rest of the paper. The paragraphs continue from here and are only separated by headings, subheadings, images and formulae. The section headings are arranged by numbers, bold and 11 pt. Here follows further instructions for authors.

NOMENCLATURE

Introduction

Certainly, we have dealt with databases almost daily, either in a market, or when exploring its catalogue on the web, to check for a product in stock. YouTube, Facebook, iTunes and Amazon all depend on databases to offer services and products to the end user [1]. The sensitivity and importance of information and data in the

∗Corresponding author

Email addresses:

Communicated by ‘sub etitor’

database system should be secured from corruption and unauthorized access, and provide privileges that give the permission on the objects included within the database in a well-defined manner [2]. Cryptography is defined as the science of keeping information protected by changing it into a form that can only be read and processed by those who have the privilege [3]. In this research, using Rivest Cipher 4 (RC4) cryptography algorithm to encrypt and decrypt in database reveals that the RC4 algorithm can perform well and secure the authenticity of the data and make it invulnerable for irresponsible people [4]. It is considerably explicit and fast compared to other encryption algorithms. RC4 algorithm mostly consists of two phases: Key Scheduling Algorithm (KSA) to create, by the key, an elementary permutation of the S array and the Pseudo Random Generation Algorithm (PRGA) to initiate the key stream [5].

RC4 algorithm is utilized as the fastest algorithm of encryption for its lightweight, robust cipher according to memory usage, power consuming, CPU that is applied in protocols of popular emails like WEP and TLS/SSL [6].

The rest of this paper includes related works that discussed RC4 algorithm, then the methodology of the proposed system followed by the experimental results and finally the conclusions.

Related Works

2. Sriadhi *et al*. [6] showed a comprehensive understanding of the implementation of RC4 cryptographic algorithms, the visualization exposed the algorithm steps of key generation, padding, S-Box creation to the encryption/decryption process. The paper revealed that cryptography is simple to be imagined and learned.

Rifki *et al*. stated that the RC4 algorithm is well known stream cipher in cryptography of symmetric key for its use in many security protocols. Furthermore, compared with other stream ciphers, RC4 is the highest speed and the lowest complexity (Fig. 1). Statistics of communication protocols on the web show that the RC4 algorithm secures 50% of TLS traffic. The RC4 algorithm is composed of the Key Scheduling Algorithm KSA that is used to initialize the S-box using key of variable length and Pseudo Random Generation Algorithm PRGA to generate bytes of keystream [7].

Tripathy A. *et al* proposed a model consists of RC4 symmetric algorithm and Elliptic Curve Cryptography ECC asymmetric algorithm. The ECC algorithm generates the key to protect the session key required by the RC4 for encryption/decryption the data to maintain the authenticity of data. The encrypted data is saved on the cloud, the user who requests the data is authenticated by the cloud service provider who decrypts that data and delivers it to the authorized user [8].

Rachmawati *et al*. proposed a hybrid cryptosystem (RC4 algorithm for protecting messages and LUC algorithm for protecting the key of RC4) by which the message is highly secured than applying a single algorithm. In the decryption process, the original message retrieved. The encrypted cipher text and cipher key will be converted to the form of decimal numbers. Both algorithms need a time that is linearly proportional for encryption and decryption, which means, it takes longer time when the message and the key used are long. The average of the encryption time with RC4 and LUC algorithms was 287.06ms and 74.86ms, and the average of decryption time for both algorithms was 53.43ms and 94.26ms [9].

Methodology

The proposed cryptosystem works on database files with (.db) extension which are created by applications like SQLite database application. The cryptosystem encrypts the database tables for one table at a time selected by the sender. A cipher text file is generated of that table and transmitted to the receiver who has the public key file for the decryption process. A new database is created which includes the table decrypted from the cipher text (Fig. 2).

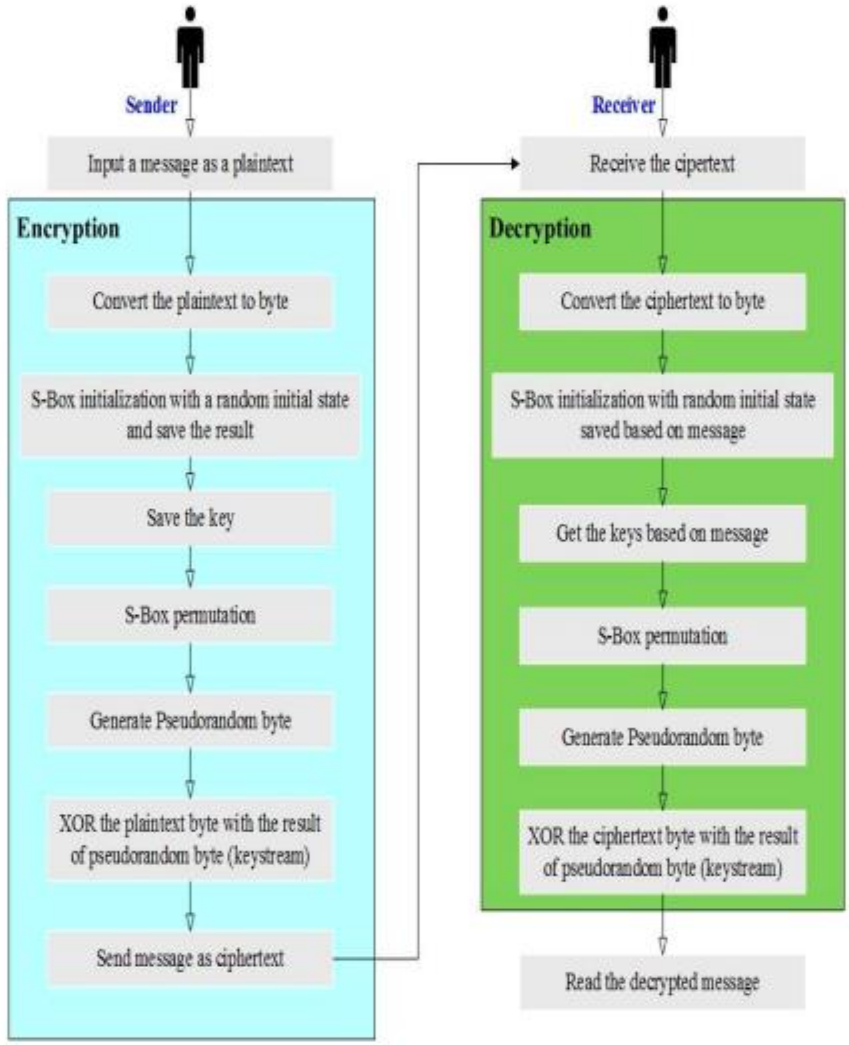

Fig. 1. RC4 Algorithm Diagram [7]

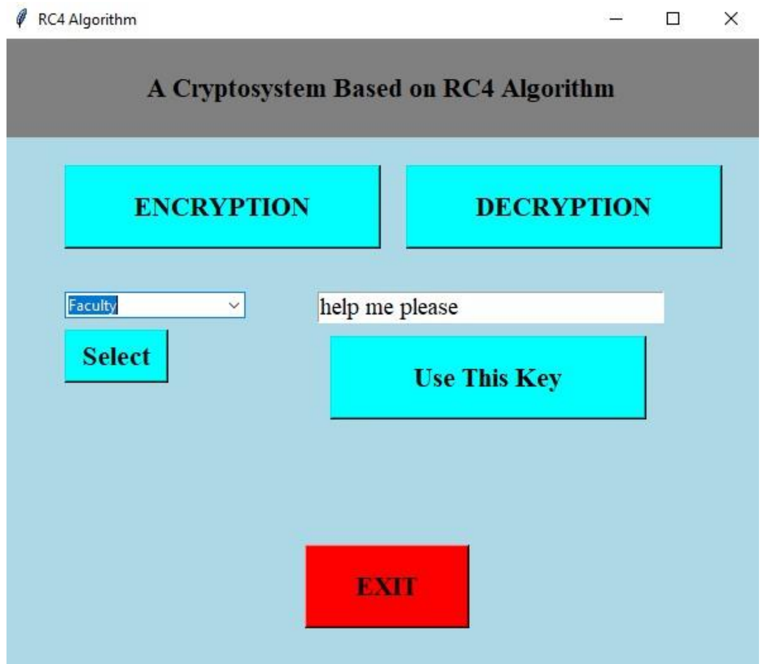

Fig. 2. The Proposed Cryptosystem

The first of the two algorithms composing the RC4 is the Key Scheduling Algorithm (KSA):

for *i = 0 -> N* **do**

S[i] = i

end for

j = 0

for *i = 0 -> N* **do**

j = (j + S[i] + key[i mod keylength]) mod N-1

swap*(S[i], S[j])*

end for

return *S*

The second is Pseudo Random Generating Algorithm (PRGA):

i = 0

j = 0

while *GeneratingOutput* **do**

i = (i + 1) mod N

j = (j + S[i]) mod N

swap_(S[i], S[j])_

K = S[(S[i] + S[j]) mod N]

return _K_

end while

where:

_N = 256_

i,j as integer

S as array of integer (S-Box array)

key as array of integer (Ascii of the key)

keylength as integer

Then, _K_ is XOR'ed with the value needs to be encrypted. For testing the proposed cryptosystem, a database called _College_ is prepared which includes a single table named _Faculty_ as shown in (Fig. 3) which becomes a plaintext for the algorithm (Fig. 4)

Name ▲1	Place of Birth	Birth Date	Degree	Scientific Name
Filter	Filter	Filter	Filter	Filter
SAMI ADIL	Babil	1973	Doctorate	Lecturer
BUSHRA AZIZ	Baghdad	1952	Doctorate	Professor
ALI TARIQ	Saladdin	1960	Master	Lecturer
AHMED KHALID	Duhok	1969	Doctorate	Assist. Prof.

Fig. 3 - The "Faculty" table

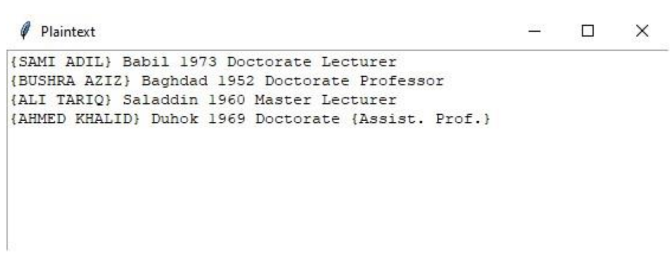
Plaintext
{SAMI ADIL} Babil 1973 Doctorate Lecturer
{BUSHRA AZIZ} Baghdad 1952 Doctorate Professor
{ALI TARIQ} Saladdin 1960 Master Lecturer
{AHMED KHALID} Duhok 1969 Doctorate {Assist. Prof.}

Fig. 4. The plaintext of the _Faculty_ table

As shown in (Fig. 2), when clicking on the ENCRYPTION button, a dialogue box appears to choose the database, and its tables is listed in the driven Combo Box to Select one of them (_Faculty_ table in this case). An Entry Box is asking to insert the public key, for example _"help me please"_, which is to be saved in a _".key"_ file and sent to the receiver in someway later.

The encryption process starts by reading the table records and encrypting them by the RC4 algorithm one record by one. The key and the first record (the plaintext) is converted into their ASCii code:

The key =

72	101	108	112	32	109	101	32	112	108	101	97	115	101

Plaintext =

'('	''''	'S'	'A'	'M'	'I'	' '	'A'	'D'	'I'	'L'	''''	'	'	' '	''''	'B'	'a'	'b'	'i'
'l'	''''	'	'	' '	'1'	'9'	'7'	'3'	'	'	' '	''''	'D'	'o'	'c'	't'	'o'	'r'	'a'
't'	'e'	''''	'	'	' '	''''	'L'	'e'	'c'	't'	'u'	'r'	'e'	'r'	''''	')'			

Plaintext code =

40	39	83	65	77	73	32	65	68	73	76	39	44	32	39	66	97	98	105	108
39	44	32	49	57	55	51	44	32	39	68	111	99	116	111	114	97	116	101	39
44	32	39	76	101	99	116	117	114	101	114	39	41							

The *S-Box* permutation table is initialized (Fig. 5):

S =

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

Fig. 5 - The *S-Box* table

After implementing the RC4 Algorithm by its two parts (KSA and PRGA) the *S-Box* table is permutated as shown in (Fig. 6):

S =

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

Fig. 6 - The permutated *S-Box* table

Then, the XOR operation is performed between the plaintext and this permutation *S-Box* to get the cipher text of the first record of the database table:

Encrypted record =

D	6	6	B	7	C	6	A	7	B	4	5	E	C	6	C	2	0	9	5	4	A	8	7	9	D	2	0	B	B	E	4	D	2	A	D
F	B	C	8	C	9	5	D	E	3	F	1	9	A	9	B	0	E	9	9	4	C	2	A	D	C	4	F	9	8	8	2	8	1	5	8
4	1	0	B	B	6	E	D	6	7	E	3	7	4	D	B	7	B	2	2	D	3	5	C	F	1	5	B	B	D	C	6	E	E		

And so on to the rest of the table records to generate a cipher text file of the *Faculty* table named "College_Faculty.txt" as shown in (Fig. 7) which is ready to be sent on the network to the receiver.

Fig. 7 **-** The ciphertext of the *Faculty* table

By clicking DECRYPTION button of the system by the receiver, as shown in Fig. 2, a dialogue box appears to choose the ciphertext file ("College_Faculty.txt" in this example). The public key is read from the *".key"* file, and the same steps of the RC4 algorithm that is used on the plaintext is performed on the ciphertext to decrypt it back to the original plaintext and generate a new database named "College_Enc.db" that contains the same *Faculty* table.

The approximate time for the encryption process of the (college.db) database that was used as an example was 0.067005s, while the decryption time was 0.141442s.

Conclusion

The proposed cryptosystem is advanced with the following features:

1. Databases can secured with this powerful cryptosystem.
2. As the RC4 algorithm is a stream algorithm, large amount of data can be processed without the need to be divided in blocks.
3. Well randomized public key depends on user input.
4. All printable characters ranged from 32-126 of ASCII code can be used in the encryption/decryption process.
5. The original database is not changed because of the outside encryption.
6. The complexity of ciphertext makes it invulnerable against different attacks.
7. Considerably low latency of encryption/decryption process.

References

[1] D. Lewis, Database systems: Volume 1, University of London International Programmes, 2016.

[2] D. S. B. Gupta and A. Mittal, INTRODUCTION TO DATABASE MANAGEMENT SYSTEM, NEW DELH: UNIVERSITY SCIENCE PRESS, 2017.

[3] W. Zakariyah, "Cryptography," 2021.

[4] A. F. Doni, O. A. H. Maria and S. Hanif, "Implementation of RC4 Cryptography Algorithm for Data," Journal of Physics: Conference Series, 2020.

[5] M. M. Abd Zaid and S. Hassan, "Lightweight RC4 Algorithm," Journal of AL-Qadisiyah for computer science and mathematics, vol. 11, no. 1, pp. 27-32, 2019.

[6] H. H. Al-Badrei and I. S. Al-Shawi, "Improvement of RC4 Security Algorithm", Advances in Mechanics, vol. 9, issuee3, pp. 1467-1476, 2021.

S. Sriadhi, R. Rahim and A. S. Ahmar, "RC4 Algorithm Visualization for Cryptography Education," Journal of Physics: Conf. Series, 2018.

[7] R. Rifki, A. Septiarini and H. R. Hatta, "Cryptography using Random Rc4 Stream Cipher on SMS for Android-Based Smartphones," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 9, no. 12, pp. 89-03, 2018.

[8] A. Tripathy, S. Rath, S. Swagatika and O. P. Jena, "Rivest Cipher 4 Cryptography and Elliptical Curve Cryptography Techniques to Secure Data in Cloud," Springer Nature Singapore Pte Ltd., pp. 661-668, 2021.

[9] D. Rachmawati, M. A. Budiman and D. F. Perangin-angin, "A hybrid cryptosystem approach for information security by using RC4 algorithm and LUC algorithm," Journal of Physics: Conference Series, vol. 1321, no. 032013, pp. 1-8, 2019.