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

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. Hence, ciphers represent encapsulating system for database tables.

Keywords:
- Cryptosystem
- Database
- Security
- RC4 algorithm
- Encryption
- Decryption

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

$$\text{for } i = 0 \rightarrow N \text{ do}$$

$$S[i] = i$$

$$\text{end for}$$

$$j = 0$$

$$\text{for } i = 0 \rightarrow N \text{ do}$$

$$j = (j + S[i] + \text{key}[i \mod \text{keylength}]) \mod N - 1$$

$$\text{swap}(S[i], S[j])$$

$$\text{end for}$$

$$\text{return } S$$

The second is Pseudo Random Generating Algorithm (PRGA):

$$i = 0$$

$$j = 0$$

$$\text{while } \text{GeneratingOutput} \text{ do}$$

$$i = (i + 1) \mod N$$

$$j = (j + S[i]) \mod N$$
**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)

![Fig. 3 - The “Faculty” table](image)

![Fig. 4. The plaintext of the Faculty table](image)

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 = \begin{pmatrix}
\end{pmatrix}

The \textit{S-Box} permutation table is initialized (Fig. 5):

\[
S = \begin{pmatrix}
0 & 2 & 5 & 4 & 1 & 8 & 7 & 6 & 3 & 10 & 11 & 14 & 15
\end{pmatrix}
\]

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

\[
S = \begin{pmatrix}
72 & 174 & 128 & 204 & 74 & 97 & 53 & 126 & 47 & 208 & 82 & 127 & 43 & 112 & 8 & 58
\end{pmatrix}
\]

The \textit{S-Box} permutation table is initialized (Fig. 5):

\[
\text{Plaintext} = \begin{pmatrix}
\end{pmatrix}
\]

\[
\text{Plaintext code} = \begin{pmatrix}
\end{pmatrix}
\]

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

\[
\text{Encrypted record} = \begin{pmatrix}
\end{pmatrix}
\]

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 be 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