AN ENCRYPTION ALGORITHM FOR IMPROVING DATABASE SECURITY USING ROT & REA

M. Sujitha[1]
Research Scholar, PG & Research Department of Computer Science,
Quaid-E-Millath Government College for Women (Autonomous),
Madras University, Chennai, India.
sujithamano26@gmail.com

M. Pushpa[2]
Assistant Professor, PG & Research Department of Computer Science,
Quaid-E-Millath Government College for Women (Autonomous),
Madras University, Chennai, India.
push_surya@yahoo.co.in

Abstract

Database is an organized collection of data, many user wants to store their personal and confidential data's in such database. Unauthorized persons may try to get the data's from database and misuse them without the owner's knowledge. To overcome such problem the advanced control mechanism, known as Database security was introduced. Encryption algorithm is one of the way to give protection to the database from various threat or hackers who target to get confidential information. This paper discuss about the proposed encryption algorithm to give security to such database.

Keywords: Database security, symmetric, encryption, REA and ROT.

1. Introduction

Database is an integral part of our day-to-day life and many database users store their sensitive data in their databases. When compared to other areas of computer security, Security to databases is sometimes not given much importance. Hackers were able to target large databases in recent years to get sensitive and confidential information from others database and surely it will raise over the next few years. So it is important to protect Database from unauthorized user and other risks [10]. This leads to the emergence of database security. There are various Database security techniques available some of them are Watermarking, Steganography and Encryptions. Out of the various database security techniques Encryption technique was proved to be the more secured[4] and easy technique. This paper "An encryption algorithm for improving database security using ROT & REA"is proposed to protect database from unauthorized users using the concepts of most popular encryption Algorithm ROT13 and the encryption algorithm REA.

2. Related works

2.1. *Data Encryption Standard*

DES is the block cipher which takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. It is a symmetric encryption technique which means both encryption and decryption use a same key . The only problem with this technique is that if the key is known to others the entire information is compromised. In this, the block size is 64 bits it also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key basically consists of 64 bits however, only 56-bits of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56-bits, and it is always quoted as such. Every 8th bit of the selected key is discarded i.e., positions 8, 16, 24, 32,40, 48, 56, 64 are removed from the 64-bit key leaving behind only the 56-bit key[3].

2.2. *Reverse Encryption Algoirthm*

Reverse Encryption algorithm or REA is a symmetric encryption algorithm that is used to protect sensitive data in database. REA is simple, secure and efficient, and takes a variable-length key, making it flawless for data security[6].

2.3. *ROT13*

The ROT13 encryption algorithm is a special case of the Caesar cipher, with a fixed key of 13.ROT13 has been under wide use for over 30 years for email and usenet(network). ROT13 was mostly useful for encrypting

messages.ROT13 is based on the principle that every letter in a reference alphabet corresponds to another letter in a rotated alphabet[9].

3. Database security

Database security is the mechanism that protect database against international or accidental threats. The security provided to database is to minimize losses caused by predicted events in a cost-effective manner without constraining users[1].Fig 1. shows the architecture of the database security:

Fig 1, *Database Security Architecture*

4. Database Security techniques

4.1. *Digital Water Marking*

Digital watermarking is an approach to cope with the miss use of the information technology in tampering, peculating and illegality situation [7]. These makers are usually invisible to human eyes and can be detected by specially designed detectors. Embedding of digital watermarks into the data is the best protection technique for digital assets from piracy

4.2. *Steganography*

Steganography is data hidden within data. It is the art and science of embedding the hidden content in unremarkable cover so that the existence of the secret gets hidden. Steganography protects from pirating copyrighted materials, as well as from unauthorized viewing.People used hidden tattoos or invisible ink in the past to convey hidden (stenographic) content [2].

Text Steganography is one of the steganography techniques which hide the text behind some other text file. It is the difficult form of steganography as the redundant amount of text to hide the secret data is scarce in text files [3].

4.3. *Encryption*

Database encryption refers to the use of encryption techniques to transform a plain text database into a (partially) encrypted database, thus making it unreadable to anyone except those who possess the knowledge of the encryption key(s)[8].

5. Database Encryption Algorithms

5.1. *REA- Reverse Encryption Algorithm*

The keys are concatenated to the text in the encryption process and removed from the text in the decryption process. Mathematical Divide operation is performed on the text data by 4 in the encryption process and multiple operations on the text by 4 has been done in the decryption process. Divide by 4operation is performed on the text to narrow the range domain of the ASCII code [6].

The cost time of the encryption and decryption operations can be reduced andthe performance is also improved by REA [5].Fig. 2 shows the working principles of REA encryption algorithm and Fig. 3 shows the simple example of the existing algorithm REA.

5.1.1. REA Encryption Algorithm:

1) Add the key before the data to be encrypted.
2) Replace the data to ASCII code and change that ASCII to binary data.
3) Reverse the binary data and convert 8 bits binary data in the form of ASCII code
4) Divide the converted ASCII code by 4 from Divide operation put the Quotient as the 1st character and Remainder as the 2nd character
5) Return encrypted data.

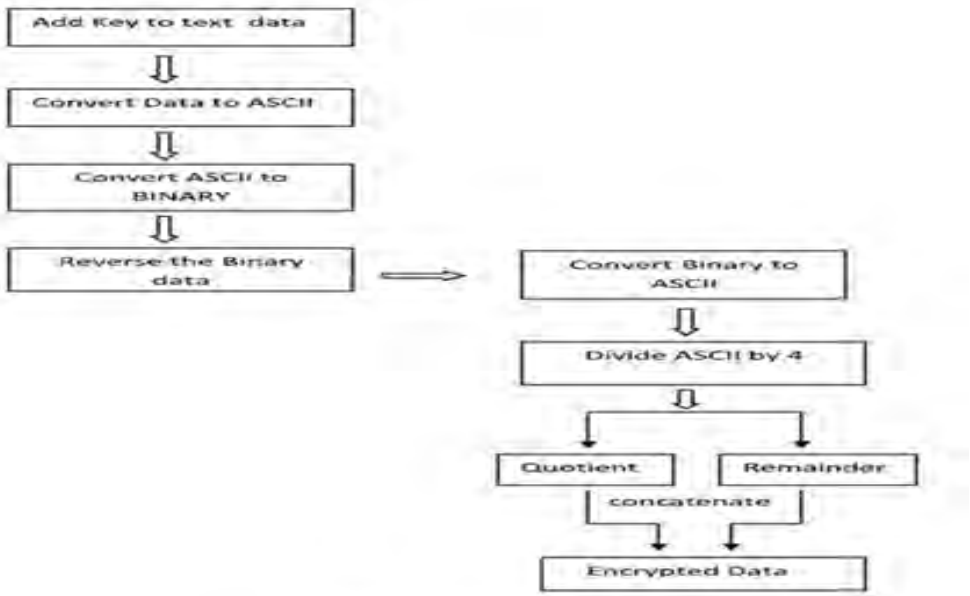

Fig 2. Working of REA Algorithm

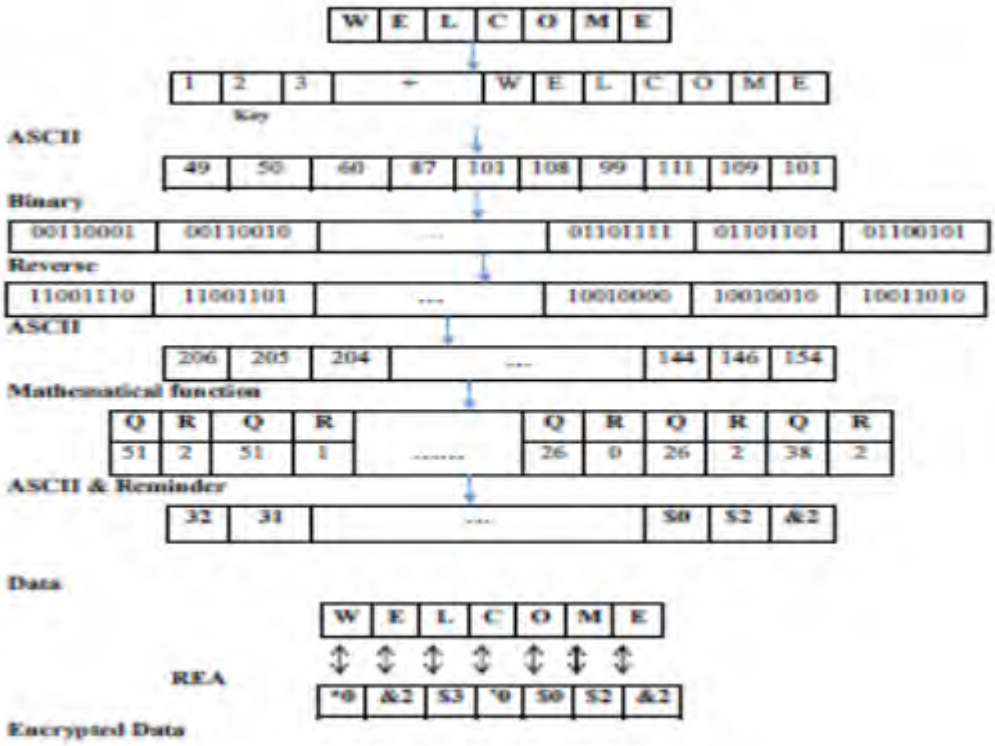

Fig. 3 *REA Encryption Steps*

5.1.2. Correlation Coefficient

Statistical analysis such as correlation coefficient factor Eq. (1) is used to measure the relationship between two variable the plaintext and its encryption for the REA Encryption Algorithm[6].

$$CorrCoef(x,y) = \frac{\sum_{i=1}^{n}(x_i - \mu(x))(y_i - \mu(y))}{\sigma(x)\sigma(y)} \qquad (1)$$

where μ(x) and μ(y) are the respective means of x and y in Eq. (2):

$$\mu(x) = \frac{1}{n}\sum_{i=1}^{n} x_i, \text{ and } \mu(y) = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (2)$$

x and y are variables of the plaintext and cipher text and the terms in the denominators (It is called the Standard deviation of x and y are Eq. (3) :

$$\sigma(x) = \sqrt{\sum_{i=1}^{N}(x_i - \mu(x))^2} \text{ and } \sigma(y) = \sqrt{\sum_{i=1}^{N}(y_i - \mu(y))^2} \qquad (3)$$

5.2. *ROT13*

ROT13 is a very simple encryption method. ROT13 replaces each one letter by 13th character further along the alphabet. The rotation of 13 was chosen because the Latin alphabet that is common in the western consists of 26 letters. The letter A of the reference alphabet corresponds to the letter N of the rotated alphabet and the letter N of the reference alphabet corresponds to the letter A in the rotated alphabet [9]. Fig. 4 narrates the working of ROT13 encryption algorithm. The secret key of ROT13 encryption algorithm is 13 and the example of the same is depicted in Fig.5.

5.2.1 The steps of Rot13 algorithm are presented as follows:

1) Read the char and the Rot value 13
2) Get the ASCII value of the char
3) If the ASCII value of the char Read is in between (‘65’ and ‘90’) or (‘97’ and ‘122 ‘) do step 4
4) Add previous ASCII value with ROT
5) If the summed value is (>90)
6) Compute Subtraction of ROT with the ASCII value
7) Obtain the char for the Previous Result
8) Repeat step – 1 to 7 for all the characters to be encrypted
9) Return the Encrypted text

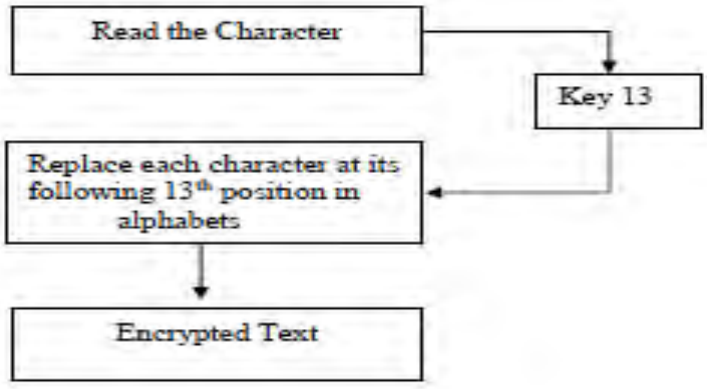

Fig 4. *ROT-13 Implementation*

5.2.2. Example for ROT13

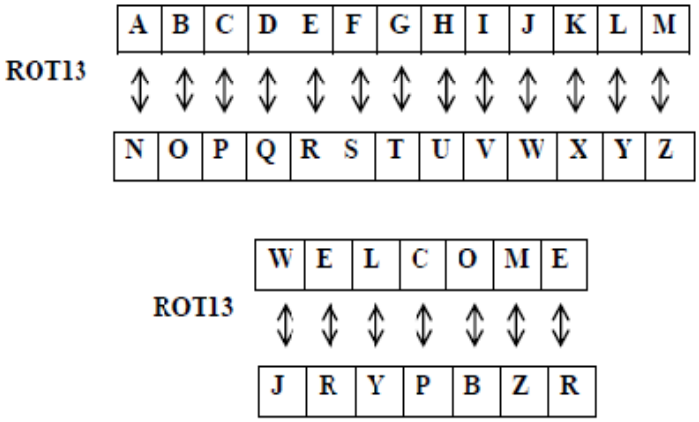

Fig. 5 *ROT13 Encryption Steps*

6. Proposed system

The proposed system "An encryption algorithm for improving database security using ROT13 & REA" includes the main concepts from the REA with the ROT13 encryption algorithms. The detailed steps of the existing and the proposed algorithms were discussed in the remaining of the paper .

6.1. *Proposed Algorithm*

The proposed algorithm is the combination of REA and ROT13 encryption algorithm. As previously said[9] more rounds of encryption bring more security. Instead of executing the same algorithm the proposed approach combines two existing algorithms which already proved to be to be best in database security. The steps incorporated in the proposed encryption algorithm are given in Fig. 6 and Fig. 7.

6.1.1. Steps of the proposed Algorithm

1. Input the data and the key value.
2. Concatenate the key to the data.
3. Convert the previous data to binary code.
4. With each 8 bits binary data divided into two portions and perform Rotate operation by 3 place for both the portions each.
5. Again Gather each 8 bits binary data from step 4 and convert to the decimal data.
6. Divide the previous decimal code by 4.
7. Obtain the ASCII code of the previous result of divide operation put the Quotient as the 1st character and Remainder as the 2nd character.
8. Return encrypted text.

Fig.6 Proposed Encryption Algorithems Step

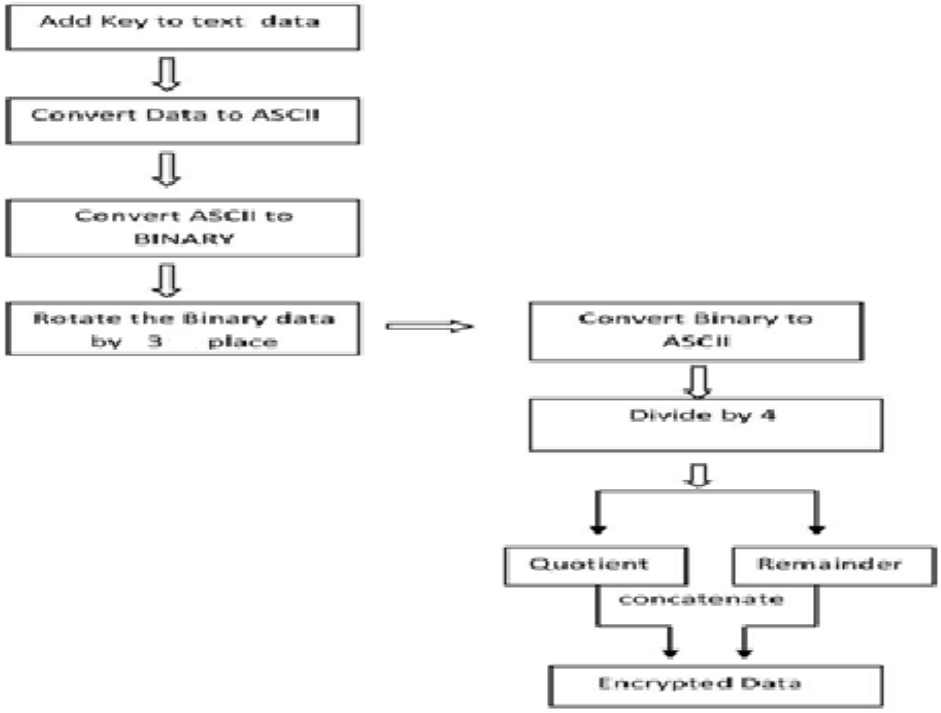

Fig 7. Implementation of Proposed Algorithm

6.1.2. Example for proposed algorithm (Rotate by 3)

Fig. 8 gives the encrypted content using the proposed algorithm and the processing steps were discussed followed by that

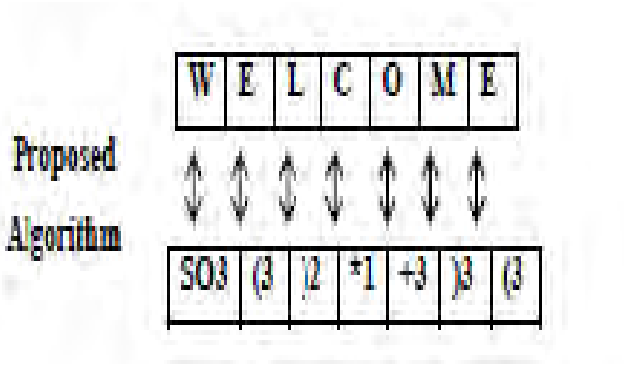

Fig.8 Encryped data using the proposed algorithm Algorithems Steps

Example

1) Add the key to the data:

123Welcome

2) Convert the data to ascii code.

1 -> 49, 2 -> 50, 3 -> 51, W -> 87, e ->101, l ->108, c ->99, o ->111, m ->109, e->101

3) Convert the ascii code to binary data:

00110001 00110010 00110011 01010111 01100101 01101100 01100011 01101111 01101101 01100101

4) Separate the each 8 binary data divide into two halves rotate it by 3 place and leave the 4th place as it is

Eg: 0011 0001 → 1001 0001

10010001 10011000 10011001 00111011 10100011 10100110 10101001 10101111 10100111 10100011

5) Gather each 8 bits from the binary data and obtain the asciicode :

145 152 153 59 163 166 169 175 167 163

6) Divide the previous ascii code by 4 and obtain the ascii of the result(put it as one ascii character) and obtain the remainder (put it as second character).

145/4 = 36 -> $(ascii of 36) and the remainder = 1(join ascii of quotient and remainder $1).

152/4 = 38->&(ascii of 38) and the remainder = 0 (join ascii of quotient and remainder &0).

153/4 = 38 ->&(ascii of 38) and the remainder = 1 (join ascii of quotient and remainder &1).

59/4 = 14 ->SO(ascii of 14) and the remainder = 3 (join ascii of quotient and remainder SO3).

163/4 = 40 -> ((ascii of 40) and the remainder = 3 (join ascii of quotient and remainder (3).

166/4 = 41 ->)(ascii of 41) and the remainder = 2 (join ascii of quotient and remainder)2).

169/4 = 42 -> * (ascii of 42) and the remainder = 1 (join ascii of quotient and remainder *1).

175/4 = 43 ->+(ascii of 43) and the remainder = 3 (join ascii of quotient and remainder +3).

167/4 = 41 ->)(ascii of 41) and the remainder = 3 (join ascii of quotient and remainder)3).

163/4 = 40 -> ((ascii of 40) and the remainder = 3 (join ascii of quotient and remainder (3).

7) And the encrypted text is

$1&0&1SO3(3)2*1+3)3(3

Conclusion

A Number of encryption algorithms are designed in a way that the process of encrypting the data is several times, so called “Rounds” to protect the database from unauthorized users and other risks security. Security provided by the proposed algorithm will be high enough, because it combines the features of the Existing algorithms ROT13 and REA, as they are individually proved to be best. The proposed algorithm might take more execution as it combines the two algorithms. The proposed algorithm is under process, yet it’s not been implemented.

Reference

[1] Thomas M.Connolly, Carolyn E.Begg(2008).” Database Systems: A Practical Approach to Design, Implementation, and Management”,4th Edition, Pearson Education .pp.[541-542].

[2] Asole, S.S., S.M. Mundada(april 2013), “A Survery on Security Databases From Unauthorized users”, International Journal of Scientific & Technology Research volume 2, Issue 4, ISSN 2277-8616

[3] Neha Rani,Jyoti Chaudhary(july 2013),"Text Steganography Techniques: A Review", International Journal of Engineering Trends and Technology (IJETT), Volume 4, Issue 7, pp-3013-3015.

[4] AbhaTamrakar, VintiNanda(july 2012),”An Encryption Algorithm for providing security to the relational database”, International Journal of Advanced and Innovation Research, Volume 1,Issue 2,ISSN: 2278 -7844.

[5] RobiniA.Chirde, KulkarniS.S.(January 2014),”Assessing Performance of Encrypted Databases under query processing with the REA Algorithm”, International Journal of Advance Research in Computer Science and Management Studies, Volume 2,Issue 1,ISSN:2321-7782.

[6] Ayman Mousa, Osama S.Faragallah, EL-Rabaie, S., NigmE.M. (March 2013),"Security Analysis of Reverse Encryption Alogrithm for Databases", International Journal of Computer Applications(0975-8887),volume 66-No.14,pp.19-26.

[7] Mohit H Bhesaniya, KunalThanki(Dec 2013),” Watermarking of Relational Databases”, International Journal for Research in Technological Studies, ISSN: Online Vol-1,Issue-110

[8] http://priyaprakharmrigank.blogspot.in/2012/10/introduction-des-is-block-cipher-which.html

[9] http://www.pruefziffernberechnung.de/Originaldokumente/2rot13.pdf

[10] http://www.brighthub.com/computing/smb-security/articles/61400.aspx