Embedded Quality Technical Education

A
SEMINAR REPORT
ON

Database Security Using Encryption

Third Year Computer Engineering

BY

Sakshi A Fokane
Exam Seat No : T150694244
Roll No : 42

Under The Guidance of

Prof.K.C.Kulkarni

DEPARTMENT OF COMPUTER ENGINEERING

University of Pune

Gokhale Education Society's

R. H. Sapat College of Engineering,
Management Studies and Research,
Nashik - 422 005, (M.S.), INDIA

Gokhale Education Society's

R. H. Sapat College of Engineering,
Management Studies and Research,
Nashik - 422 005, (M.S.), INDIA

CERTIFICATE

This is to certify that the seminar report entitled ***"Database Security Using Encryption"*** is being submitted herewith by "Sakshi Anil Fokane, 72007226D" has successfully completed his seminar work in partial fulfillment of requirements for the degree of Third Year Computer Engineering of Savitribai Phule Pune University.

Prof. Prof. K.C. Kulkarni	**Dr. D.V. Patil**
Seminar Guide	**Head of the Department**

Gokhale Education Society's

R. H. Sapat College of Engineering, Management Studies and Research,

Nashik - 422 005, (M.S.), INDIA

Seminar Approval Sheet

This Seminar entitled

"Database Security Using Encryption"

prepared and submitted by "Sakshi Anil Fokane" has been approved and accepted in partial fulfillment of the requirements for the degree Third Year Computer Engineering.

Prof. K.C.Kulkarni	Ms. R. D. Narwade
Seminar Guide	Seminar Coordinator

Acknowledgement

I would like to express my sincere gratitude to everyone for providing their invaluable guidance, comments and suggestion throughout the course of seminar project. I would specially thank Prof K.C. Kulkarni for timely checking my progress constantly motivating me to work harder.

In this report, I hope to highlight the enormous opportunities presented by technology for maintaining the security of the data stored in the database by using encryption.in my desire to work this report I have in no way any claim to come out with a perfect piece of work.

These few details lead me to realize that like all human endeavors this project is not perfect and may contain errors and shortcomings. Thus I remain open to all criticisms and suggestions which could present me with new sources of inspiration as I develop my ability to research and learn.

Abstract

**Security of data is the most important task in today's world. Over the years various encryption schemes have been developed in order to protect the database from various attacks by the intruders.This paper discuss the importance of database encryption and makes an in depth review of various database encryption techniques and compare them on basis of their merits and demerits.
https://www.overleaf.com/project/60754b61e6cb314d706a8bdd**

Contents

List of Figures

Chapter 1

Introduction

In this age of technology, all our work is being done by the computers. From chatting with friends on social networking websites, to making online payments through Net Banking, everything is being done online through computers. Since these facilities are efficient and make our work easy we use them in one way or the other. This means to use these online services we are storing all our personal and sensitive data in the databases of these websites and applications, which indeed make this data prone to various security threats. So protection of this important user data is one of the major priority, in order to avoid any misuse of data .

1.1 SEMINAR IDEA

A Encryption is typically defined as a Security for the valuable data stored in the database . Authorization and Authentication are two major processes that are used to protect the data from the front end(i.e. User Side) that is being accessed by the user, where authorization means whether a person has the rights to access the data or not, while authentication means identifying the user which is generally done by the use of username/password [1]. Another important way of protecting this data is by encrypting the data being saved in the databases of these websites. In this paper we will discuss the various database encryption schemes proposed by different authors, and also study their merits and demerits of these schemes .

1.2 MOTIVATION

The need of encrypting the data before saving it in a database is that by restricting the access through authorization and authentication of data can help to a certain limit, but what if the intruder somehow gets to the database. He has all the data of database and can misuse it as he likes, here encryption of data before saving it in database comes into play. If the datais encrypted before saving it in the database, even with the accessto the database the intruder cannot misuse this data. Fig 1,show how the intruder can access the contents of database .

Chapter 2

LITRATURE REVIEW

2.1 Paper Surveys

1. Sesay, Samba, Zongkai Yang, Jingwen Chen, and Du Xu. ”A secure database encryption scheme.” In Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE, pp. 49-53. IEEE, 2005.[4]:-

   Samba Sesay, et al [4], discuses the importance of database security in activities like E-Commerce and Enterprise Resource Planning (ERP). They also discuss the various loopholes that attackers use to access the database. Finally they proposed a system for database encryption which minimizes the time cost of encryption and decryption process, while covering all aspect of security like confidentiality, access control, integrity, authentication and non-repudiation. Although their system posses various advantages but also suffer a major disadvantage that queries like sum, average and counts cannot be performed directly.

2. Min-Shiang Hwang, Wei-Pang Yang, “A two-phase encryption scheme for enhancing database security”, Journal of Systems and Software, Volume 31, Issue 3, December 1995, Pages 257-265, ISSN: 0164-1212. (http://www.sciencedirect .com/science/article/pii/0164121294001022).[5]:-

   Min-Shiang Hwang and Wei-Pang Yang [5], proposed a 2 way encryption scheme based on the concept of one-way function and subkeys that ensures full security; moreover 2 additional algorithms were given that efficiently handles a of key management problem. They also compared their proposed scheme with the schemes of GI Davida.

3. Chang, Chin-Chen, and Chao-Wen Chan. ”A database record encryption scheme using the RSA public key cryptosystem and its master keys.” In Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on, pp. 345-348. IEEE, 2003.[13]:-

   Chin-Chen Chang and Chao-Wen Chan [13], gave two schemes of database encryption. Scheme 1 for field oriented encryption system, and Scheme 2 for record oriented encryption system. Both schemes based on RSA Master Keys, which also solves the key management problem and also helps in providing access rights to different users. The proposed schemes overcome the weakness of the schemes proposed by GI Davida.

4. Erez Shmueli, Ronen Vaisenberg, Ehud Gudes, Yuval Elovici, “Implementing a

database encryption solution, design and implementation issues", Computers Security, Volume 44, July 2014, Pages 33-50, ISSN 0167-4048[?]:-

Erez Shmueli, et al [15], evaluated all the traditional 5 architectures for database encryption and gave a new encryption architecture which removed the weakness of all earlier architectures. This scheme places the encryption module just above the database cache, inside the database management system (DBMS) and uses a dedicated technique to encrypt each database value together with its coordinates.

5. Kaur, Kamaljit, K. S. Dhindsa, and Ghanaya Singh. "Numeric to Numeric Encryption of Databases: Using 3Kdec Algorithm." In Advance Computing Conference, 2009. IACC 2009. IEEE International, pp. 1501-1505. IEEE, 2009.[?]:-

   Kamaljit Kaur, et al [16], discussed the importance of security in today's world.They proposed encryption of numeric data in the database using 3KDEC algorithm. The algorithm is easy to use and takes very less computations. Further they simulated various attacks like Brute Force Attack; Statistical Attack etc to show the encrypted numerical values cannot be cracked by the attackers. Moreover they stated that the algorithm is not limited to databases but can even be used in other areas where security is required.

Chapter 3

Existing System

3.1 SECURITY ANALYSIS OF MD5 ALGORITHM

A hash function is a one-way encryption function that takes a variable-size input plaintext m and generates a fixed-size hash output. It is computationally hard to decipher the hash and any attempt to crack it is practically infeasible. A “secure” hash function should be able to resist pre-image attacks and collision attacks. Due to the pigeonhole principle and birthday paradox, there will be some inputs that will produce the same hash result. The output length is of fixed size 128 bits, making a total of 2128 possible output hash values. This value, as big as it may seem, is not infinite, hence resulting in collisions. A. MD5 algorithm
MD5 (Message Digest Algorithm 5) was designed by Ron Rivest in 1991. MD5 processes a variable-length message into a fixed-length output of 128 bits. MD5 is a popular hash function. It works on blocks of 512-bits, and processes each block through 4 rounds, where each round in turn processes 16 sub-blocks (each 32-bits). The 512-bit message is divided into 16 sub-blocks before processing. If a message block is not up to 512-bits, it is padded as shown in Fig. 1. A detailed explanation of the algorithm can be found at [1].

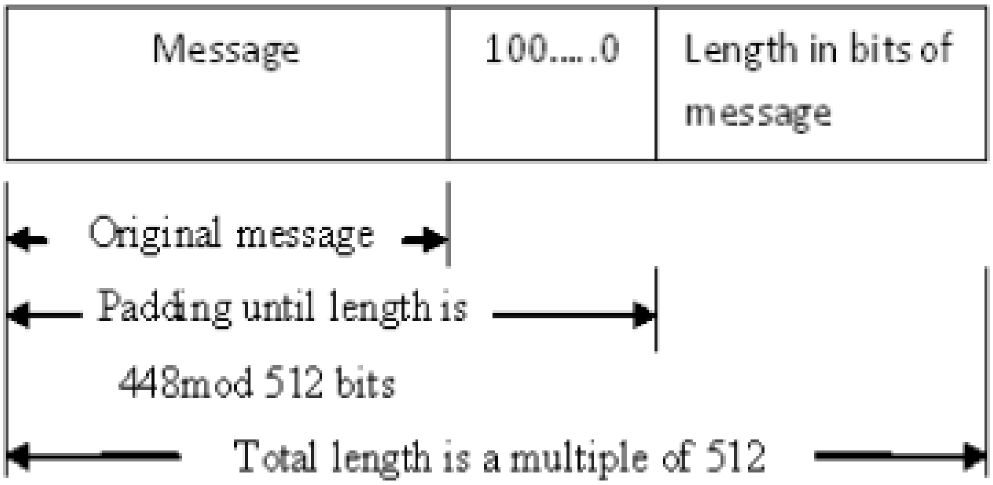


Figure 3.1: Length of message after padding (in bits)

3.2 APPLICATION OF MD5 ALGORITHM IN PASSWORD STORAGE SECURITY

It is highly insecure to store passwords in plaintext in the database. In order to increase the security of passwords, MD5 algorithms can be used to hash the original passwords and the hash values, instead of the plaintext are stored in the database. During authentication, the input password is also hashed by MD5 in a similar way, and the result hash value is compared with the hash value in the database for that particular user.

3.3 SECURITY ANALYSIS OF MD5

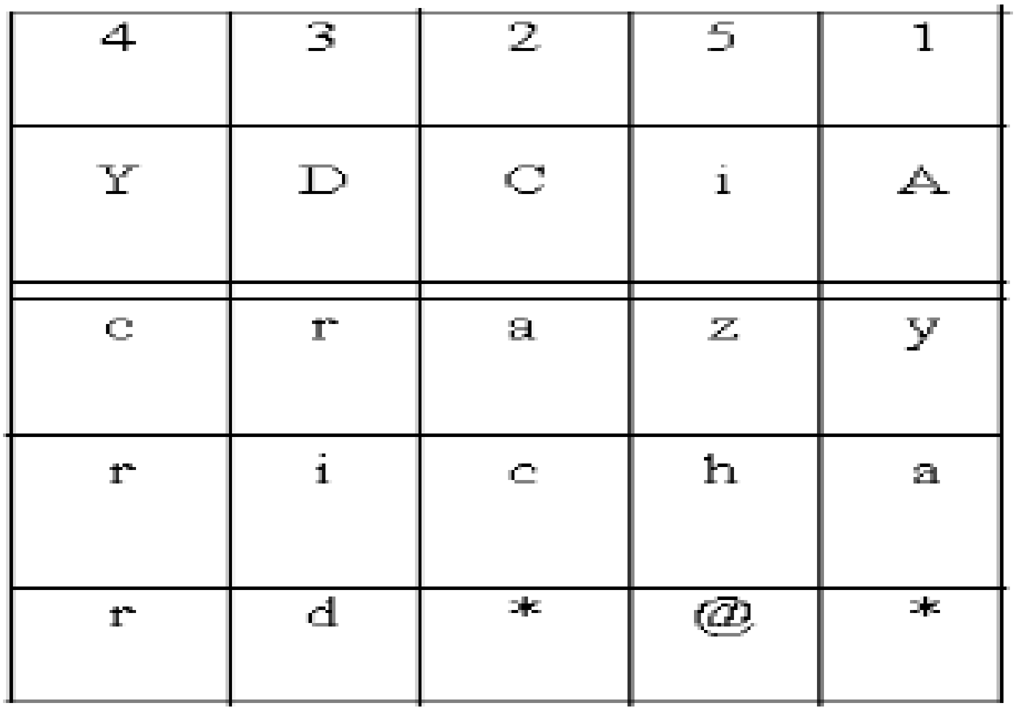
MD5 algorithm is prone to two main types of attack: dictionary attacks and rainbow tables.

1. Dictionary Attacks: In dictionary attacks, an attacker tries all the possible passwords in an exhaustive list called a dictionary. The attacker hashes each password from the dictionary and performs a binary search on the compromised hashed passwords. This method can be made much quicker by precomputing the hash values of these possible passwords and storing them in a hash table.

2. Rainbow Tables: Rainbow tables are made up of hash chains and are more efficient than hash tables as they optimize the storage requirements, although the lookup is made slightly slower.Rainbow tables differ from hash tables in that they are created using both reduction and hash functions. Reduction functions convert a hash value to a plaintext. The plaintext is not the original plaintext from which the hash value was generated, but another one. By alternating the hash function with the reduction function, chains of alternating passwords and hash values are formed. Only the first (chain's start point) and last plaintext (chain's end point) generated are stored in the table. To decipher a hashed password, we first process the hashed password through reduction functions until we find a match to a chain's end point. We then take that chain's corresponding start point and regenerate the hash chain and find the original plaintext to the hashed password. Rainbow tables are very easily available online now. There are many password cracking systems and websites that use rainbow tables also, for example, OphCrack. Of course, using rainbow tables do not guarantee a 100

3.4 RELATED WORK

We will now demonstrate how we can hash passwords in databases using an improved version of MD5. There are five main steps involved.First, a random key string of random length is first generated. Its character set is 0-9, a-z, A-Z.This random key string is used to generate the complex password and is also stored in the database for later use during password authentication. Secondly, the password is transformed into a complex password through columnar transposition cipher. Assuming that the random key is "YDCiA" and the password is "crazyrichard", the password is first converted into a matrix of 5 columns (same as length of random key) and the blank cells area alternately filled with “*” and “@”, as shown in Fig.2. Using columnar transposition cipher, the complex password generated is “ya*ac*ridcrrzh@”. Thirdly, the salt is calculated by finding the XOR value of the random key string with the complex password, row by row. In our example, the salt is ”



4	3	2	5	1
Y	D	C	i	A
c	r	a	z	y
r	i	c	h	a
r	d	*	@	*

Figure 3.2: Generate complex password through columnar transposition

Password storage security is one important aspect of data security as most systems nowadays require an authentication method using passwords. Hashing algorithms such as MD5 are commonly used for encrypting plaintext passwords into strings that theoretically cannot be deciphered by hackers due to their one-way encryption feature. However, with time, attacks became possible through the use of dictionary tables and rainbow tables. In this paper, we discussed different methods to thwart these attacks: (1) the use of a strong password to reduce the probability of it existing in a dictionary, (2) using salts, (3) key stretching and iteration hashing to make the MD5 computation slower, (4) chaining method, where the output of one iteration is used in the input of the next iteration and the use of a different initialization vector for each password, (5) encrypting the password before hashing and (6) XOR cipher to make the final hash value impossible to find in any rainbow table. An implementation of the proposed approach is carried out using C as programming language and Microsoft SQL Server as database. With our proposed approach, the attacker will now have to hack into the database, the server containing the application code as well as the external file.

Chapter 4

PROPOSED SYSTEM

4.1 Authentication and Authorization

Authorization and Authentication are two major processes that are used to protect the data from the front end(i.e. User Side) that is being accessed by the user, where authorization means whether a person has the rights to access the data or not, while authentication means identifying the user which is generally done by the use of username/password [1]. Another important way of protecting this data is by encrypting the data being saved in the databases of these websites. In this paper we will discuss the various database encryption schemes proposed by different authors, and also study their merits and demerits of these schemes.

4.2 DATABASE ENCRYPTION

Database Encryption is a process of encrypting the data in the database [2]. It is a key strategy to protect the contents of data within the database. The main idea behind this is that incase the intruder somehow is able to get to the database of the system; due to encryption he should not be able to misuse the data in the database.Figure 2, shows basic working of the database encryption and decryption process. The plain text/data to be saved in the database is first converted into cipher text using an appropriate algorithm and a specific key. Then this cipher text is saved into database. When the user wants to extract the data from the from the database, the cipher text is converted back to plain text using the decryption algorithm and the same key used in encryption. This will return the plain text to the user, when requested.

Database Encryption can be done in two possible ways[2]:

1. Encryption: It is a process in which plain text is converted to cipher text with help of key, and then using the same key we can decrypt the cipher text back to plain text[3]. Encryption is performed using various algorithms, with each algorithm having his own disadvantages.Most commonly advantages and used encryption algorithms are DES, RC2, AES$_1$28, $AES_256etc.referfigure.3$

2. Hashing: It is a one way process, in which plain text is converted into hashed value(encrypted form). Once the data is hashed using a Hash Function it cannot be changed back to plain Text[3]. Generally this approach is usedfor password encryption, whenever we need to login the password entered is encrypted using hash function and then matched with the password stored in the database which

is already in encrypted form, if both matches the user get access else it gets the message of invalid username/password. Most commonly used Hash Functions are MD4, MD5, SHA, SHA-1 etc. Figure 4, shows working of hashing.

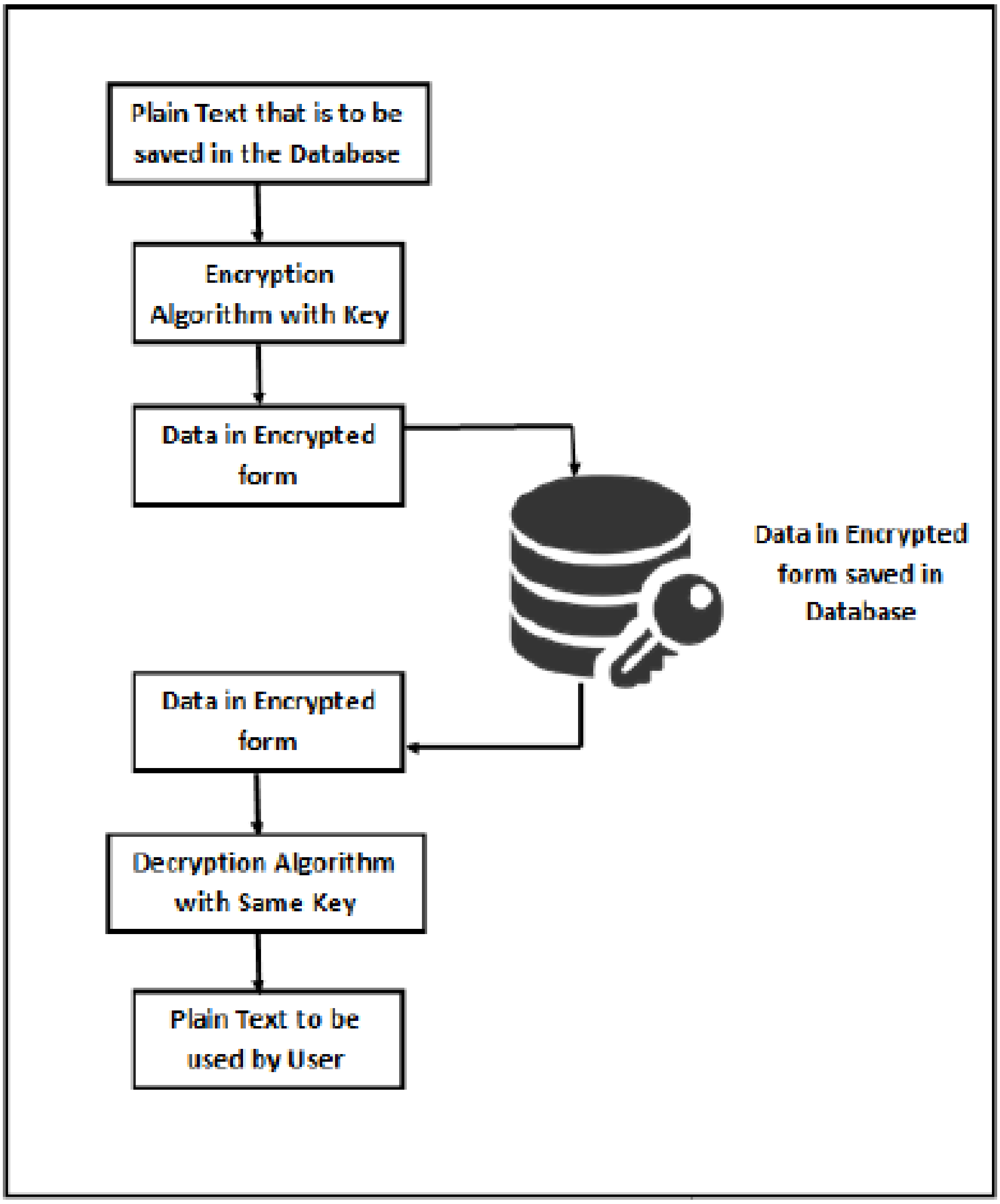

Plain Text that is to be saved in the Database
Encryption Algorithm with Key
Data in Encrypted form
Data in Encrypted form saved in Database
Data in Encrypted form
Decryption Algorithm with Same Key
Plain Text to be used by User

Figure 4.1: Database Encryption and Decryption process

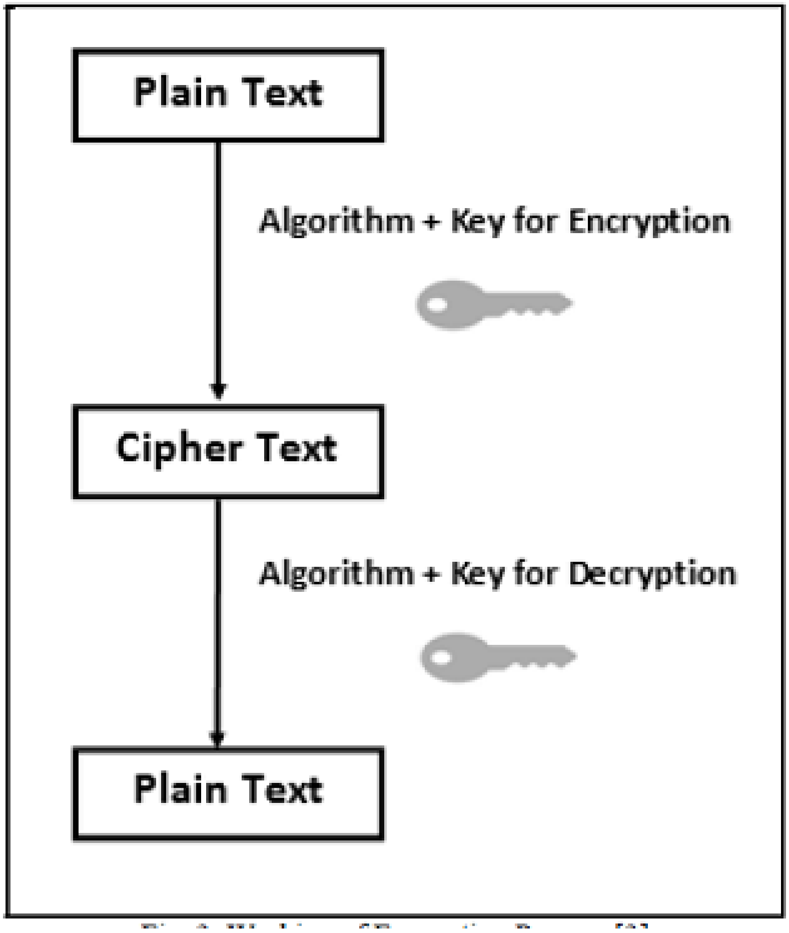

Plain Text
Algorithm + Key for Encryption
Cipher Text
Algorithm + Key for Decryption
Plain Text

Figure 4.2: Working Of Encryption process

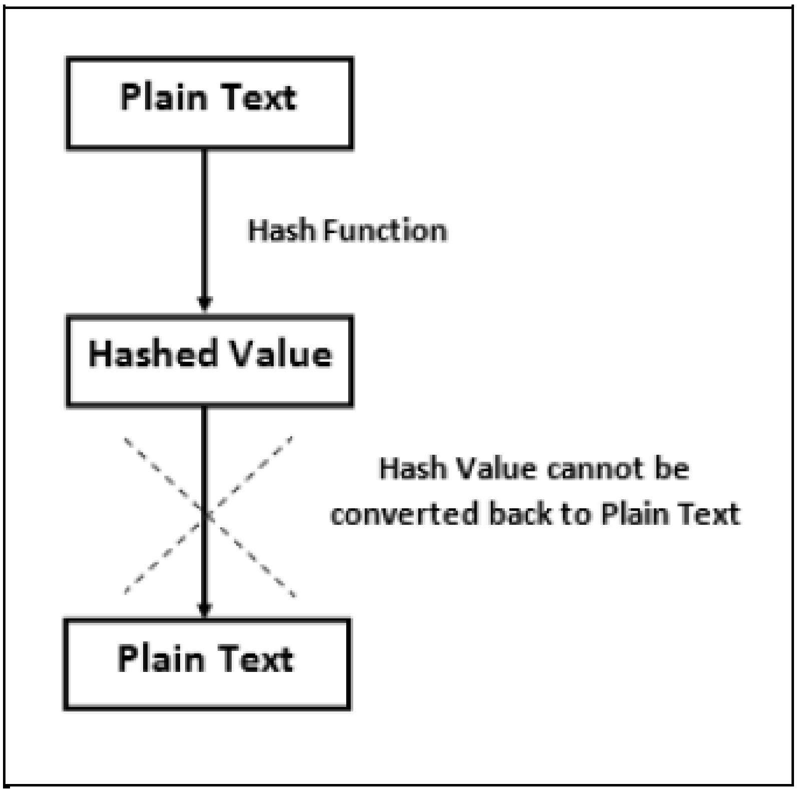
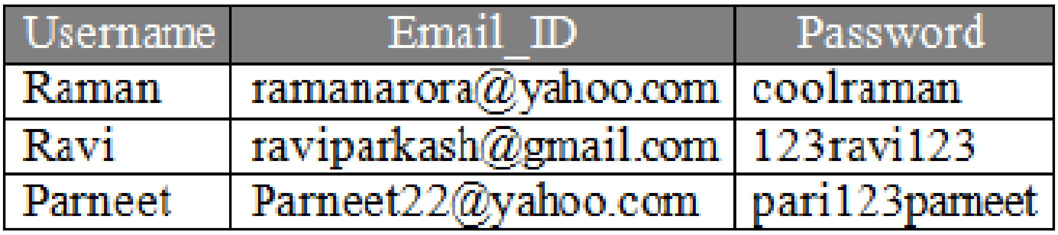


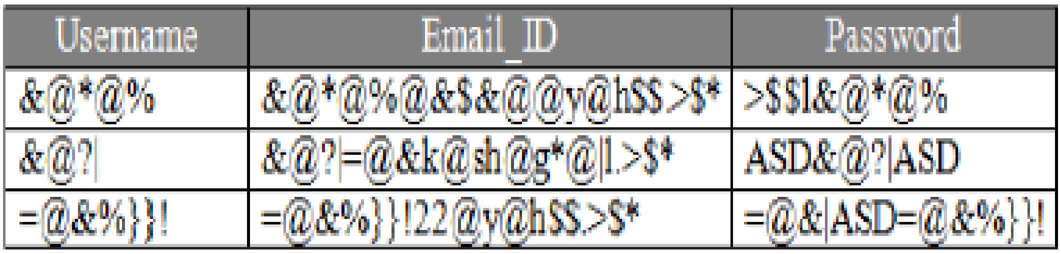
Figure 4.3: Working Of hashing process

Figure 5 and 6 shows database in normal and encrypted form. In figure 6, the contents in the database cannot be understood, so it becomes almost impossible for the intruder/hacker to misuse this data.



Username	Email_ID	Password
Raman	ramanarora@yahoo.com	coolraman
Ravi	raviparkash@gmail.com	123ravi123
Parneet	Parneet22@yahoo.com	pari123parneet

Figure 4.4: Database In Normal Form



Username	Email_ID	Password
&@*@%	&@*@%@&$&@@y@hSS>$*	>$$l&@*@%
&@?|	&@?|=@&k@sh@g*@|1.>$*	ASD&@?|ASD
=@&%}}!	=@&%}}!22@y@hSS>$*	=@&|ASD=@&%}}!

Figure 4.5: Database In Encryption Form

4.3 Merits

1. Time Cost of Encryption and Decryption operation is minimum.
2. Encryption Provides Security for Data at All Times Generally, data is most vulnerable when it is being moved from one location to another. Encryption works during data transport or at rest, making it an ideal solution no matter where data is stored or how it is used. Encryption should be standard for all data stored at all times, regardless of whether or not it is deemed "important".
3. Encrypted Data Maintains Integrity Hackers don't just steal information, they also can benefit from altering data to commit fraud. While it is possible for skilled individuals to alter encrypted data, recipients of the data will be able to detect the corruption, which allows for a quick response to the cyber-attack.
4. Encryption Protects Privacy Encryption is used to protect sensitive data, including personal information for individuals. This helps to ensure anonymity and privacy, reducing opportunities for surveillance by both criminals and government agencies. Encryption technology is so powerful that some governments are attempting to put limits on the effectiveness of encryption—which does not ensure privacy for companies or individuals.
5. Encryption is Part of Compliance Many industries have strict compliance requirements to help protect those whose personal information is stored by organizations. HIPAA, FIPS, and other regulations rely on security methods such as encryption to protect data, and businesses can use encryption to achieve comprehensive security.

6. **Encryption Protects Data across Devices Multiple (and mobile) devices are a big part of our lives, and transferring data from device to device is a risky proposition. Encryption technology can help protect store data across all devices, even during transfer. Additional security measures like advanced authentication help deter unauthorized users.**

4.4 Demerits

(a) **Cannot directly perform queries of statistical functions such as sum, average, counts etc.**

Chapter 5

RESULTS

i. **A lot of research work has been done in field of database encryption, with a sole motive to improve security of data in the database.A secure database encryption scheme has various merits like Security is guaranteed by use of two phase encryption. Problem of key management is also efficiently handled, also have some demerits like Lot of computation and Overheads required.**

Chapter 6

CONCLUSION

6.1 CONCLUSION

With advancement in Technology, nowadays everything is being done with computers, so security of these data in the database becomes an important issue. Many researchers have worked on this thing and proposed various algorithms and architectures. Each scheme has its own advantages and disadvantages. But none of them is fully secure, and contain certain loopholes or demerits with can be used by the attackers and the intruders to get access of the database. So there is a scope of improvement in this area and researchers are already working on it to find the perfect solution of this problem and find a scheme that is fully secure from all the possible security threats

6.2 Future Work:-

Data to any organization is a most valuable property. Security of sensitive data is always a big challenge for an organization at any level. In today's technological world, database is vulnerable to hosts of attacks. In this study major security issues faced databases are identified and some encryption methods are discussed that can help to reduce the attacks risks and protect the sensitive data. It has been concluded that encryption provides confidentiality but give no assurance of integrity unless we use some digital signature or Hash function. Using strong encryption algorithms reduces the performance. The future work could be carried out make encryption more effective and efficient.

Chapter 7

References

[1] *Baraani-Dastjerdi, Ahmad, Josef Pieprzyk, and Reihaneh Safavi-Naini. ”Security in databases: A survey study.” Department of Computer Science, The University of Wollongong (1996).*

[2] *Denny Cherry and Thomas Larock, “2 - Database Encryption, In Securing SQL Server”, edited by Denny Cherry, Thomas Larock, Syngress, Boston, 2011, Pages 27-71, ISBN : 9781597496254.*

[3] *Kessler, Gary C. ”An overview of Cryptography.” (2003). (http://www.sciencedirect.com/science/article/pii/B978159749625410000 22).*

[4] *Sesay, Samba, Zongkai Yang, Jingwen Chen, and Du Xu. ”A secure database encryption scheme.” In Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE, pp. 49-53. IEEE, 2005.*

[5] *Min-Shiang Hwang, Wei-Pang Yang, “A two-phase encryption scheme for enhancing database security”, Journal of Systems and Software, Volume 31, Issue 3, December 1995, Pages 257-265, ISSN: 0164-1212. (http://www.sciencedirect.com/science/article/pii/0164121294001022).*

[6] *Davida, George I., David L. Wells, and John B. Kam. ”A database encryption system with subkeys.” ACM Transactions on Database Systems (TODS) 6, no. 2, Page: 312-328, (1981).*

[7] *Lin, C. S., “An Application of an Encryption Algorithm to Database Security, Chap. 3, Ph.D. Thesis”, National Tsing Hua University, 1991.*

[8] *Lin, C. H., Chang, C. C., and Lee, C. T., “A record-oriented cryptosystem for database sharing”, in International Computer Symposium, pp. 328-329, 1990.*

[9] *T. Ge and S. Zdonik, “Fast, secure encryption for indexing in a column oriented DBMS,” in International Conference on Data Engineering - ICDE 2007. pp. 676–685. IEEE, 2007.*

[10] *Jacob, Stéphane. ”Cryptanalysis of a fast encryption scheme for databases.” In Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on, pp. 2468-2472. IEEE, 2010.*

[11] *Ecrypt - European Network Of Excellence In Cryptology, "The eSTREAM Stream Cipher Project," 2005 http://www.ecrypt.eu.org/stream/.*

[12] *Arshad, Noor Habibah, Saharbudin Naim Tahir Shah, Azlinah Mohamed, and Abdul Manaf Mamat. "Database encryption using enhanced affine block cipher algorithm." In Proceedings of the 10th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering, pp. 71-76. World Scientific and Engineering Academy and Society (WSEAS), 2008.*

[13] *Chang, Chin-Chen, and Chao-Wen Chan. "A database record encryption scheme using the RSA public key cryptosystem and its master keys." In Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on, pp. 345-348. IEEE, 2003*

[14] *Buehrer, D., and C. Chang. "A cryptographic mechanism for sharing databases." In The International Conference on Information Systems. Hangzhou, China, pp. 1039-1045. 1991.*