International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-4, April 2013

A Database Encryption Technique to Enhance Security Using Hill Cipher Algorithm

Jasdeep Singh Bhalla

Abstract*— *Data security has become one of the major challenges of the digital world. Security, privacy and integrity of data are required in every operation that is performed on the Internet. Therefore, in this paper, our focus will be on Database Security, as databases are considered as the storehouses of data. Generally, data of an organization or a company is stored in databases and is very crucial to the organization. Today, most of the organizations allow their clients to use their services (online banking, online shopping etc) by accessing their databases. This leads to a requirement of high level security to deal with information attackers. An information attacker tries to illegally acquire or modify the highly confidential data of the organization. In this paper, a new technique is being proposed for securing the database data items using Hill Encryption Algorithm which is being implemented in fortifying and strengthening the database. Database Security has become one of the most important challenges in database research. A lot of research is going on in building of new techniques for protecting database records.

Index Terms*— *Database Security, Database Encryption, Hill encryption, Database Protection

I. INTRODUCTION

Data security is one of the most important challenges in the world of digital media. Security and privacy of data along with integrity of data for every operation performed on the Internet is required. Whenever security of data is discussed, it is mostly in the context of secure data transfer over the unreliable communication networks. But the security of the data in databases is also significantly important. A database stores important data that needs to be protected from various data attackers. Although, various access control techniques have been successfully employed since the origin of database systems to protect database system from illegal accesses, but for a long time security of a database was considered an extra problem which was required to be addressed. Databases have an additional role apart from just storing data- that is to provide a more flexible access to the data items to the user at the same time it is this open access that makes databases vulnerable to various kinds of malicious activities.

Some common security **issues** that are **most commonly faced by database administrators are as follows**:

Unauthorized activities or any misuse of data by authorized database users, database administrators, or by unauthorized users or data hackers.

Computer viruses leading to incidents such as unauthorized access, leakage or disclosure of confidential data, deletion of or damage to the data item sets.

Overload of data, performance constraints, Storage constraints and capacity issues resulting in the inability of authorized users to use databases as they are intended to be used. Physical damage to database servers caused by any natural disturbance such as fires floods, Earthquakes etc. Other Physical damage causing problems are Computer overheating, lightning, accidental liquid spills, static discharge, electronic equipment failures and obsolescence.

Designing errors and programming bugs in databases can lead to inconsistent data.

Loss of data caused by the entry of invalid data commands or information, mistakes in database or system administration processes, illegal criminal damage etc.

The databases in today's time allow user to access their data using forms of web interface, allowing internal or external users to access the database. This user – database interaction leads to loss of data, damaged data and other problems. Thus, a database administrator have to spent a significant amount of time setting appropriate security permissions on the databases and web servers.

II. EXISTING DATABASE THREATS

In this various existing database threats are discussed that are unhealthy for databases.

A. SQL Injection Attacks

SQL injection attack is a very basic attack used either to gain an unauthorized or an illegal access to a database for retrieving important information directly from the database. The basic principles underlying SQL injection attacks are simple and these types of attacks are easier to execute and master. Many web applications these days, take user input from a form filled by the user. Generally, this input from the user is used in the construction of a SQL query submitted to a database.

SQL injection attack involves placing of SQL statements in the user's input as a way to fool the database and retrieve information from the database illegally.

Let us take a look at how it works by analyzing a very simple web application example that processes a client's request. Suppose a simple page for obtaining items from the database using item's number (itemNo). The page itself might be a basic HTML form that contains a textbox called ItemNo and a submit button. When this form is submitted, the following SQL query is executed:

✓ SELECT * FROM database WHERE itemNo = <item's Number>;

The result of the above query is then displayed on the result page as per the normal procedure.

Manuscript published on 30 April 2013.
***** Correspondence Author (s)
Jasdeep Singh Bhalla, Department of Computer Science, Bharati Vidyapeeth's College Of Engineering, G.G.S.I.P.U, New Delhi, India.

During a normal client enquiry, this form works quite well. Suppose 'abc' person visits the page and enters the *itemNo* as **5**. The following query would retrieve the appropriate results:

✓ SELECT * FROM database WHERE itemNo = 5;

However, the same code can be a dangerous when accessed by a data atacker. Imagine that *Malicious user (data attacker)* comes along and enters the following data in the *itemNo* field: ***"5; DROP TABLE database"***. This would cause the following query to execute:

➢ SELECT * FROM database WHERE itemNo = **5; DROP TABLE database**

By doing this, the *malicious user* successfully deletes the database table completely without even opening the database. Therefore, these types of attacks have fatal outcomes to an organizational database.

Protection from these attacks:

There are several steps that can protect such attacks:

- Implement **parameter checking** concept on all applications [4]. For example, if you're asking someone to enter an item number (or Roll number, Customer Number etc), make sure that input is numeric before executing the query by applying various data type checking techniques.
- **Limit the permissions** of the account that executes SQL queries by distributing certain permissions to different accounts. The rule of least privilege applies. If the account used to execute the query doesn't have permission to drop tables then table dropping will not executed. Use stored procedures to prevent users from directly interacting with SQL code.

B. Access Control in databases

Security is of utmost importance to a database administrator who seeks to protect the gigabytes of vital business data from the eyes of unauthorized outsiders and insiders attempting to exceed their authority by accessing the database illegally. All relational database management systems provide some sort of intrinsic security mechanisms designed to minimize these kinds of threats. Thus, focusing on the security mechanisms that are common to all databases that implement the Structured Query Language, we will discuss through the process of strengthening data access controls and ensuring the safety of the data of the database. All Server based databases support an individual user account concept similar to that used in computer operating systems that we use in our daily lives.

It is highly recommended to create individual database user accounts for each person who will be accessing the database in order to distribute certain permissions to certain accounts individually. It's technically possible to share accounts between users or simply use one user account for each type of user that needs to access your database. If a specific user leaves an organization and authority wishes to remove his or her access from the database, they will be forced to change the password that all users rely upon in case of one account. Instead, if different user accounts are made then only the password for the user that has left the organization is to be changed [4].

The methods for creating user accounts vary from platform to platform and organizational authority have to consult the DBMS-specific documentation for the exact procedure. Examples: Microsoft SQL Server users should investigate the use of the *sp_adduser* stored procedures whereas the Oracle database administrators will find the *CREATE USER* command useful.

For example, Microsoft SQL Server supports the use of Windows NT Integrated Security mechanism. Under this scheme, users are identified to the database by their Windows NT user accounts and they are not required to enter an additional user ID or password to access the database system. This approach is extremely popular among the database administrators because it shifts the burden of account management to the network administration staff and it provides the ease of a single sign-in to the end user. For a small number of users, creating user accounts and assigning permissions directly to them is sufficient. However, for a large number of users, maintaining accounts and proper permissions leads to extra burden. To overcome this burden, relational databases support the notion of roles. Database roles function in a similar way as Windows NT groups. User accounts are assigned to role and permissions are then assigned to the role as a whole rather than the individual user accounts. Suppose, an organization creates a DBA role and then add the user accounts of our administrative staff to this role. Once this is done, authority can assign a specific permission to all present administrators by simply assigning the permission to the role. Once again, the procedure for creating roles varies from platform to platform. MS SQL Server administrators should investigate the sp_addrole stored procedure while Oracle DBAs should use the CREATE ROLE syntax.

C. Cryptography in databases

There are many aspects based on security of applications, ranging from secure commerce and payments to private communications and protecting passwords & user details. One essential aspect for secure communications is cryptography. It is important to note that while cryptography is *necessary* for secure communications, it is not by itself sufficient. Within the context of any application-to-application communication, there are some specific security requirements, including:

- **Authentication of the user:** The process of validating user's identity with proper evidence.
- **Privacy/confidentiality of user's data:** Ensuring that no one can read the message except the intended receiver.
- **Integrity of data:** The received message by the receiver should not be altered in any way from the original message.
- **Non-repudiation:** A mechanism to prove that the sender really sent this message.

Cryptanalysis is known as a study of methods for obtaining the meaning of encrypted information, without accessing the secret information, which is normally required to do so.

III. EXISTING WORK

In this section we list some common security techniques that may prove useful in fortifying the database:

Securing a Database using Cryptography methodology

A database encryption scheme was proposed by Sesay et al. In this scheme, the users are divided into two levels: Level 1 (L1) and Level 2 (L2) [1], [2].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Level 1 users have the access to their own private encrypted data and the unclassified public data, whereas Level 2 users have the access to their own private data and also classified data which is stored in an encrypted form.

A novel database encryption mechanism was proposed by Liu et al [3]. The proposed mechanism performs column-wise encryption that allows the users to classify their data into sensitive and public data. This classification helps in selection of only that data (for encryption) which is critical and leaves the public data untouched, thereby reducing the burden of encryption and decryption of the whole database, as result of which the performance is not compromised.

Mixed Cryptography Database [5] scheme proposed by Kadhem et al involves the designing of a framework of encrypting the databases over the unsecured network in a diversified form that comprises owning of many keys by various parties. In the proposed framework, the data is grouped depending upon the ownership and on other conditions as well.

Securing a Database using Steganography methodology

Das et al. [6] presented various techniques in steganography that could be implemented to hide critical data and prevent them from unauthorized and illegal access. The various techniques includes- still image steganography, audio steganography, video steganography.

A method that uses steganography to hide data was presented by Naseem et al [7]. In his proposed scheme the data is embedded in the LSB's of the pixel values. The pixels values are categorized into different ranges and depending on the range certain number of bits is allocated to hide the sensitive data.

Kuo et al. presented a different approach in concealing of data items. In this scheme the image is divided into fixed number of blocks [8]. Histogram of each block is calculated along with the maximum and minimum points to mask the data. This mechanism increases the hiding capacity of the data.

Dey et al. employs a diversified approach in efficient hiding of sensitive data and escalate the data hiding capacity in still images [9], [10]. The technique involves use of prime numbers and natural numbers to enhance the number of bit planes to cloak the data in the images.

Securing Database using Access Control

Bertino et al. [11] provides an authorization technique for video databases. In this proposed scheme, the access to the database and to a particular stream of the video is granted only after verifying the credentials of that user. The credentials may not just be the user-id but it may be the characteristics that define the user and only after successful verification of the credentials the user is granted the permission to access the database.

Kodali et al. proposed a generalized authorization model for multimedia digital libraries [12], [13], [14]. This scheme involved integration of the three most common and widely used access control mechanisms such as- mandatory, discretionary and role-based models into a single framework to allow a unified access to the protected data. The technique also addresses the need of continuous media data while supporting the Quality of Service (QoS) constraints alongside preserving the operational semantics.

An authorization model was proposed by Rizvi et al [16], [17].This technique was based on authorization views which enables authorization transparent querying in which the user queries are formed and represented in terms of database relations and are acceptable only when the queries can be verified using the information contained in the authorization rules. The work presents the new techniques of validity and conditional validity which is an extension of the earlier work done in the same area.

Hill Cipher Algorithm

In the field of **cryptography**, **Hill cipher** is a polygraphic substitution, linear algebra based cipher technique. It was invented by Lester S. Hill in 1929 [19], it was the first polygraphic cipher in which it was practical to operate on more than three symbols at once.

The basic concept behind this algorithm is that each letter (Alphabet) is allotted a number generally starting from 0 in a continuous sequence one after the other as shown in figure 1.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
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

Figure 1: Alphabet Numbering

As shown in figure 1, Alphabets are numbered as A = 0, B =1, ... , Z=25, but this is not a fixed requirement of the cipher. The encryption of plain text takes **n** successive plain text letters and substitutes them for **n** cipher text letters.

In case n = 3, the encryption can be expressed in terms of the matrix multiplication as follows:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \bmod 26$$

Mathematical Model:

Abbreviations used are:

C = Cipher Text

P = Plain Text

K = Key used for Encryption of plain text to cipher text.

K^{-1} = Inverse of key K used for Decryption of cipher text to plain text.

$C = E_K(P) = KP \bmod 26$

$P = D_K(C) = K^{-1}C \bmod 26 = K^{-1}KP = P$

K is the key matrix and K^{-1} is the matrix inverse.

The inverse matrix can be calculated as $K.K^{-1} = I$ where I is the identity matrix.

IV. PROPOSED METHOD

In this paper, an innovative approach for database security using *Hill Cipher* encryption & decryption method is proposed. This cryptography based approach provides an additional level of security to the database systems. Important data or information crucial to the organization should be secured from illegal theft and attacks. This encryption scheme completely is based on key matrix that a organization decides before encryption and decryption. In this proposed scheme a common key is used to encrypt the data items.

Encryption Algorithm:

- Fetch the source text present in the database that is going to be encrypted.
- Fetch their position values (Table ID, Table Name, Row number, Column Number).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

- Choose an Encryption key matrix (say K) which would be used in Hill Cipher Algorithm for encryption of text.
- Obtain the cipher text using Hill Algorithm by applying the same key matrix which was derived in the previous step.
- Place this encrypted data in the corresponding field.
- Execute the query obtain cipher text values.

Decryption Algorithm:

- Obtain the cipher text for the purpose of decryption.
- Obtain Inverse of the key matrix (K^{-1}) used for encryption of plain text.
- Apply this inverse key matrix in the decryption of cipher text using Hill Cipher Algorithm.
- The above step will give plain text as a result.

This is a simple algorithm which provides an effective solution for data security problem because encryption of text encrypts the plain text to cipher text using key matrix and decryption of text decrypts the cipher text to plain text using inverse key matrix using hill algorithm. Therefore, this increases the level of database security which helps in eliminating database data item attacks to a large extent.

Hill cipher algorithm is a linear algebra based substitution algorithm which is fast and easy to apply. It is a good encryption scheme which is computationally secure. This encryption scheme will provide database security without affecting the system performance. Encrypted data is always slightly high in volume as compared to the original data. Decryption process is the inverse of the encryption process in which the inverse key matrix is used to decrypt the cipher text.

V. EXPERIMENTAL RESULTS

A. *Fetch Source text from the database.*

In figure 2, the database data items that were considered in the experiment are shown.

	A	B	C
1	ID	Name	Item Details
2	1	hellofriends	sdsd
3	2	abcdef	sssd
4	3	ghi	ccsd

Figure 2: Database Data Items

B. *Encryption of fetched data values from the database*

For Encryption of the data item that is fetched from the database, in our case -"hellofriends" *Hill Cipher* encryption algorithm is applied along with the key matrix K. Plaintext: "**HELLOFRIENDS**"

Encryption Key (k):

$$k = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Cipher Text for first three letters of the plain text:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 7\,'H' \\ 4\,'E' \\ 11\,'L' \end{pmatrix} \bmod 26 = \begin{pmatrix} 242 \\ 250 \\ 231 \end{pmatrix} \bmod 26 = \begin{pmatrix} 8\,'I' \\ 16\,'Q' \\ 23\,'X' \end{pmatrix}$$

Similarly, cipher text for remaining letters (L,O,F,R,I,E,N,D,S) is calculated.

Cipher Text: **"IQXIQPDNWYDK".**

C. *Decryption of encrypted database values*

For decryption use the *inverse* key matrix P = C mod 26. Inverse of K which is K^{-1} is used for decrypting the cipher text into plain text.

$$k = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \Rightarrow k^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

Plain Text for first three letters of the Cipher text:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \begin{pmatrix} 7\,'I' \\ 4\,'Q' \\ 11\,'X' \end{pmatrix} \bmod 26 = \begin{pmatrix} 215 \\ 238 \\ 349 \end{pmatrix} \bmod 26 = \begin{pmatrix} 7\,'H' \\ 4\,'E' \\ 11\,'L' \end{pmatrix}$$

Similarly, plain text for remaining letters (I,Q,P,D,N,W,Y,D,K) is calculated.

Plain Text: **"HELLOFRIENDS".**

VI. CONCLUSION

Database security is one of the important issues that demands for complete attention from researchers. To get a more secured database system, a lot of research is going on. In this paper, various threats that a database system suffers from and the need for security to eliminate these threats were discussed. Some existing security techniques that are being employed presently to augment and enhance the security of the database systems were explained. A database encryption technique was proposed using Hill Cipher Algorithm on the database data items. The basic concept behind Hill Cipher Algorithm was used and explained. The proposed method was applied on database items in experimental results (Section V).

VII. FUTURE SCOPE

A lot of research is being going on for improving the existing database security techniques. In this field, the highest priority is given to the work that is initiated towards securing the integrity of data items present in the database. The database security technique proposed in this paper can be improvised in future in terms of enhancing security. Hill Encryption Algorithm is used in this paper for encryption and decryption of data items present in a database. So in future, some other algorithm can be used for securing the database items.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

REFERENCES

[1] Samba Sesay, Zongkai Yang, Jingwen Chen, Du Xu, "A Secure Database Encryption Scheme", Second IEEE Consumer Communications and Networking Conference (CCNC), 3-6 Jan. 2005, pp. 49- 53.

[2] Srikanth Chava, "A Security Protocol for Multi-User Authentication", arXiv: 0804.1970v1 [cs.CR].

[3] E. Anupriya, Sachin Soni, Amit Agnihotri, Sourabh Babelay, "Encryption using XOR based Extended Key for Information Security – A Novel Approach", International Journal on Computer Science and Engineering (IJCSE), vol. 3, issue 1, Jan. 2011, pp. 146-154.

[4] Aarthi.G, Dr. E. Ramaraj, "A Novel Encryption approach in Database Security", International Journal of Computer& Organization Trends –Volume 2 Issue 1- 2012.

[5] Hasan Kadhem, Toshiyuki Amagasa, Hiroyuki Kitagawa, "A Novel Framework for Database Security based on Mixed Cryptography", Fourth International Conference on Internet and Web Applications and Services, 24-28 May 2009, pp.163-170.

[6] Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay, Sugata Sanyal, "Steganography and Steganalysis: Different Approaches", International Journal of Computers, Information Technology and Engineering (IJCITAE), Vol. 2, No 1, June, 2008, Serial Publications, pp. 1-11.

[7] M. Naseem, Ibrahim M. Hussain, M. Kamran Khan, Aisha Ajmal, "An Optimum Modified Bit Plane Splicing LSB Algorithm for Secret Data Hiding", International Journal of Computer Applications, Vol. 29, No. 12, 2011. Foundation of Computer Science, New York, USA, pp. 36-43.

[8] Wen-Chung Kuo, Dong-Jin Jiang, Yu-Chih Huang, "A Reversible Data Hiding Scheme Based on Block Division", Congress on Image and Signal Processing, Vol. 1, 27-30 May 2008, pp. 365-369.

[9] Sandipan Dey, Ajith Abraham, Sugata Sanyal, "An LSB Data Hiding Technique Using Prime Numbers", IEEE Third International Symposium on Information Assurance and Security, Manchester, United Kingdom, IEEE Computer Society press, USA, 29-31 Aug. 2007, pp.101-106.

[10] Sandipan Dey, Ajith Abraham, Bijoy Bandyopadhyay and Sugata Sanyal, "Data Hiding Techniques Using Prime and Natural Numbers" Journal of Digital Information Management, vol. 6, no. 3, 2008, pp. 463-485.

[11] Elisa Bertino, Moustafa A. Hammad, Walid G. Aref , Ahmed K. Elmagarmid, "An Access Control Model for Video Database Systems", Proceedings of the ninth international conference on Information and knowledge management, 2000, pp. 336 – 343.

[12] Naren Kodali, Csilla Farkas, Duminda Wijesekera, "An authorization model for multimedia digital libraries", International Journal on Digital Libraries, vol. 4, no. 3, 2004, pp. 139-155.

[13] Béchara Al Bouna, Richard Chbeir, "Multimedia-based authorization and access control policy specification".

[14] Shermann S.M. Chan, Qing Li, José A. Pino, "Access Control Mechanism for Collaborative Video Database Production Applications", Proceedings of IEEE Sixth International Symposium on Multimedia Software Engineering, 13-15 Dec. 2004, pp. 396- 402.

[15] Harshavardhan Kayarkar, "Classification of Various Security Techniques in Databases and their Comparative Analysis"

[16] S Rizvi, A Mendelzon, S Sudarshan, Prasan Roy, "Extending query rewriting techniques for fine-grained access control", Proceedings of the ACM SIGMOD international conference on Management of data, 2004, pp. 551 – 562.

[17] Q Wang, T Yu, N Li, J Lobo, E Bertino, "On the Correctness Criteria of Fine Grained Access Control in Relational Databases", Proceedings of the 33rd international conference on Very large data bases, 2007, pp. 555-566.

[18] Erez Shmueli, Ronen Vaisenberg, Yuval Elovici, Chanan Glezer, "Database Encryption – An Overview of Contemporary Challenges and Design Considerations", SIGMOD Record, September 2009 (Vol. 38, No. 3).

[19] Lester S. Hill, Cryptography in an Algebraic Alphabet, *The American Mathematical Monthly* Vol.36, June–July 1929, pp. 306–312.

[20] Amira Rezk, H. A. Ali, S. I. Barakat, "Database Security Protection based on a New Mechanism", International Journal of Computer Applications (0975 – 8887) Volume 49– No.19, July 2012