2009 IEEE International Advance Computing Conference (IACC 2009)
Patiala, India, 6–7 March 2009

Numeric To Numeric Encryption of Databases: Using 3Kdec Algorithm

Kamaljit Kaur [1], K.S Dhindsa [2], Ghanaya Singh [3]
1 Pursuing M.Tech (BBSBEC, Fatehgarh Sahib), 2 Asst. Professor, BBSBEC, Fatehgarh Sahib, 3 Release Manager, Miri InfoTech Chandigarh

***Abstract*—the volume of data storage capacity has changed a lot as compared with earlier times. As most computers were standalone and only the users had access to data, security was not a big concern. All this changed when computers became linked in networks, in form of small dedicated networks to large LANs, WANs and the World Wide Web. With the growth of networking the security of data became a big issue. Data passes through various networks, communication protocols, and devices to ultimately reach to the user which has made data security increasingly important. Security is becoming one of the most urgent challenges in database research and industry. Instead of building walls around servers, a protective layer of encryption should be provided around specific sensitive data-items. This also allows us to define which data stored in databases are sensitive and thereby focusing the protection only on the sensitive data, which in turn minimizes the delays or burdens on the system that may occur from other bulk encryption methods.**

This paper describes a highly original and new approach of securing numeric data of databases. It presents a practical solution to the problem where numeric data was converted to alphanumeric type and hence encrypted data was not possible to be stored in the existing numeric field. The proposed algorithm allows transparent record level encryption that does not change the data field type or fixed length.

Keywords-Encryption; Decryption; Symmetric Encryption; Block Cipher; Key Expansion; Substitution box; Row Shifting; AddKey.

I. INTRODUCTION

The best way of securing the data is to restrict access to the data which can be achieved by the process of authentication and authorization. A user should be asked for authenticating information before accessing the data and should only be allowed to perform the operations for which access rights are available. If the data to be accessed is on a local machine, applying access control is easy, but if data is accessed from a remote client, user credentials and data needs to be secured on the network. In such situations security protocols are used. Incase that a malicious user somehow breaches the above security provisions and gets access to data; the only solution is scrambling the data. So encrypting the data whether it is in-motion or at-rest – is the next level of security that will make the data worthless for the hacker [10].

Encryption is the process of disguising data in such a way to hide its substance, is a very effective way to achieve security for data at rest. Implementation of a database encryption strategy raises several important factors that must be taken into consideration like should the encryption be performed inside the database or in the application where the data is generated or in a hardware device? Should encryption keys be kept inside the database or somewhere else where it is more secure? Should the granularity of encrypting data be applied to a database, a table or a column level? [2]

In this paper we will focus on a security solution for protection of data at rest, specifically protection of numeric data that resides in the databases.

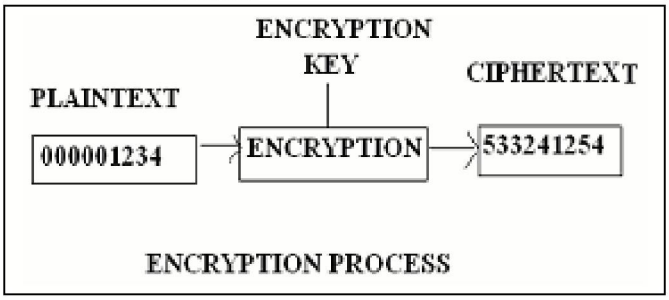

Figure 1. Illustrating encryption of Plaintext (as numeric data) to Ciphertext (again numeric data)

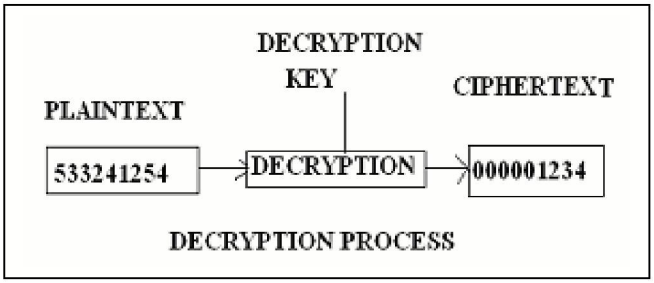

Figure 2. Illustrating decryption of numeric data as Plaintext into numeric data as Ciphertext

II. ABOUT 3KDEC ALGORITHM

The algorithm is named 3Kdec from its working as it encrypts numeric data which is in form of decimal using three keys which can be changed anytime.

3Kdec is a Symmetric key Block encipherment algorithm.

Symmetric key algorithm is one which uses a single secret key for both encryption and decryption. Encryption/Decryption as visualized in Figure 1 and 2 above can be considered as an electronic locking where sender puts the message in a box and locks the box using the shared secret key; receiver unlocks the

978-1-4244-2928-8/09/$25.00 © 2009 IEEE

box with the same key and takes out the message. The original message is called plaintext and the message sent through the channel after encryption is called ciphertext. To create ciphertext from the plaintext sender uses an encryption algorithm and a shared secret key. To create plaintext from the ciphertext receiver uses a decryption algorithm and the same secret key.

The key can be visualized as a set of values/numbers that the cipher as an algorithm operates on [1].

In symmetric key encipherment the encryption and decryption algorithms are inverses of each other. If P is the plaintext, C is the ciphertext, and K is the key, the encryption algorithm $E_K(x)$ creates the ciphertext from; the decryption algorithm $D_K(x)$ creates the plaintext from the ciphertext. Encryption algorithm $E_K(x)$ and the decryption algorithm DK(x) are inverses of each other and they cancel the effect of each other when applied one after the other on the same input.

Encryption: C= EK (P)

Decryption: P= DK (C)

That is,

P= DK (C) = DK (EK (x)) = EK (DK (x)) = x

Block Ciphers means a group of plaintext symbols of size m (where m>1) are encrypted together creating a group of ciphertext of the same size [5].

III. KEY COMPONENTS OF 3KDEC ALGORITHM

3Kdec algorithm uses:

- Numeric data to be encrypted.
- Three keys (stored as three 3 X 3 matrix)
- Substitution Box (commonly referred as S-Box) and an inverse S-box
- Variable number of Rounds (which can be 3, 6 or 9)

Algorithm operates on following steps in sequence for set number of rounds as illustrated in Figure 3:

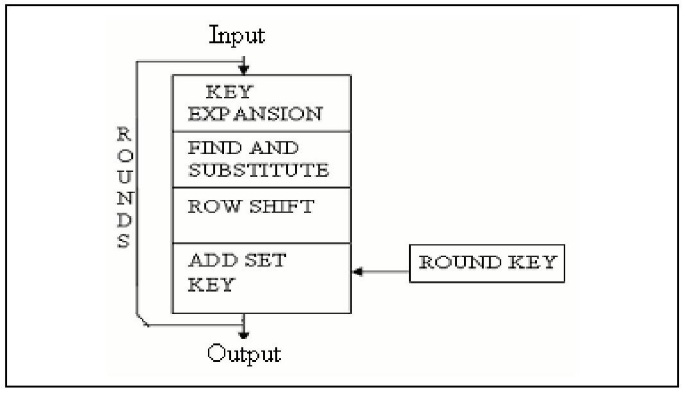

Figure 3. Illustration of 3Kdec Algorithm Working

1. Key Expansion
2. Find and Substitute
3. Row Shift
4. Add set Key

A. KEY EXPANSION

In this step, the single key of the three keys are expanded to as many as three keys summing up the total of nine keys to be used in the maximum nine rounds. Key1 is expanded as Key $_{10}$, Key $_{11}$ and Key $_{12}$ and similarly Key $_{20}$, Key $_{21}$, Key $_{22}$, Key $_{30}$, Key $_{31}$ and Key $_{32}$.

The complexity of key expansion is directly depending on number of rounds. Incase of 3 rounds there will be algorithm complexity of one; while incase of 6 rounds there will be a complexity of two and similarly three incase of 9 rounds.

The process of expansion of Key 1 into its constituent Key $_{10}$, Key $_{11}$ and Key $_{12}$ is as follows in Figure 4:

Key $_{10}$ is same as Key 1

Key $_{11}$ is shifting the row 1 zero times, row 2 elements one time and row 3 elements two times with respect to original key Key1.

Key $_{12}$ is shifting the row 1 one times, row 2 elements two times and row 3 elements zero times with respect to original key Key1.

With variable number of rounds and varying key expansions the complexity of algorithm increases.

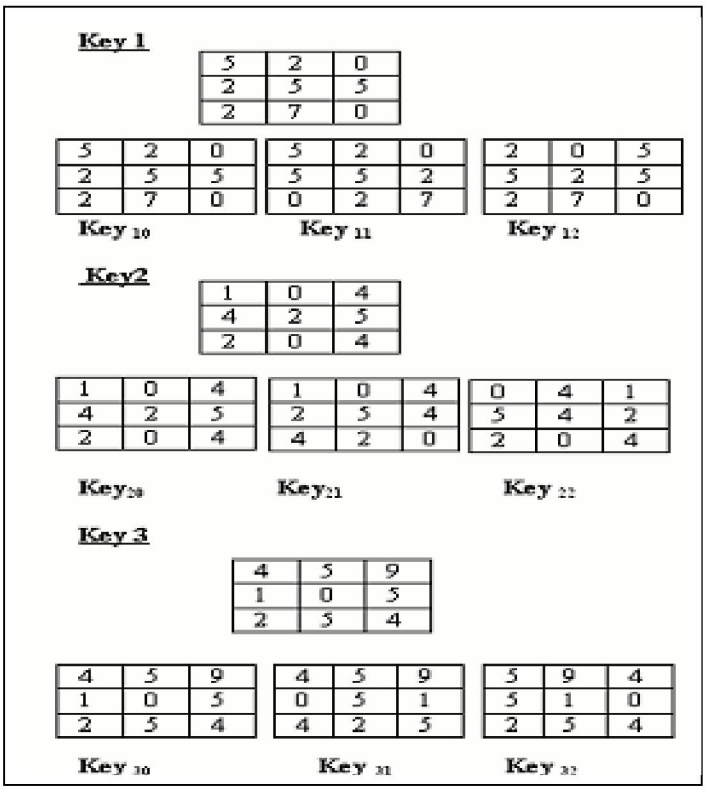

Figure 4. Key Expansion Process

B. FIND AND SUBSTITUTE 1.

In this step, the byte to be encrypted is found and substituted independently to provide the confusion effect.

There is no fixed mechanism or any mathematical correlation in the formation of S-box. The entries of S-box can be different in different encryption processes.

So the simple structure and variable entries of Substitution Box makes the algorithm eligible to be used as a Personal Encryption Algorithm where different S-Box variants can be used in encryption process depending on the party we are dealing with and the varying complexity levels can be set according to our needs.

Example:

0	1	2	3	4	5	6	7	8	9
7	3	6	1	0	9	2	5	4	8

Substitution Box

0	1	2	3	4	5	6	7	8	9
4	3	6	1	8	7	2	0	9	5

Inverse Substitution Box

Figure 5. S-box and Inverse S-box

C. ROW SHIFT

This transformation step shifts towards the left. The number of shifts depends on the row number of the matrix. This means that the first row of matrix is shifted zero times, second row of matrix is shifted one time and the third row is shifted two times towards the left.

During the decryption process, the Inverse Row Shift process is carried out and the shifting is done towards the right. The number of shifts is same as the row number.

Where Row 0 has: no shift

Row 1: Shift 1

Row 2: Shift 2

D. ADD SETKEY

With each round the matrix is added using XOR operation with the above expanded keys. This means during first round of encryption Key 10 is used. In the next round Key 11 and then Key 12 and so on…

IV. DESCRIPTION OF 3KDEC ALGORITHM

Let us have a look on the input data will be processed when our algorithm is applied to it:

Initially the user will be prompted for the plaintext to be encrypted and the number of rounds of the encryption process that the user want to carry out.

INPUT: 1234

The initial step is to pad the input data with 0s i.e. the input to be encrypted becomes: 000001234

NUMBER OF ROUNDS: 9

THREE SET OF KEYS where each key is 3X3 matrix

Then the three are expanded using the Key expansion process as described above. Number of key expansions depends upon the user input of number of rounds.

Now the three transformations: Find and Substitute; Row Shift and Add Set Key are carried out on input for the set number of rounds.

Figure 6 below illustrates the above stated process:

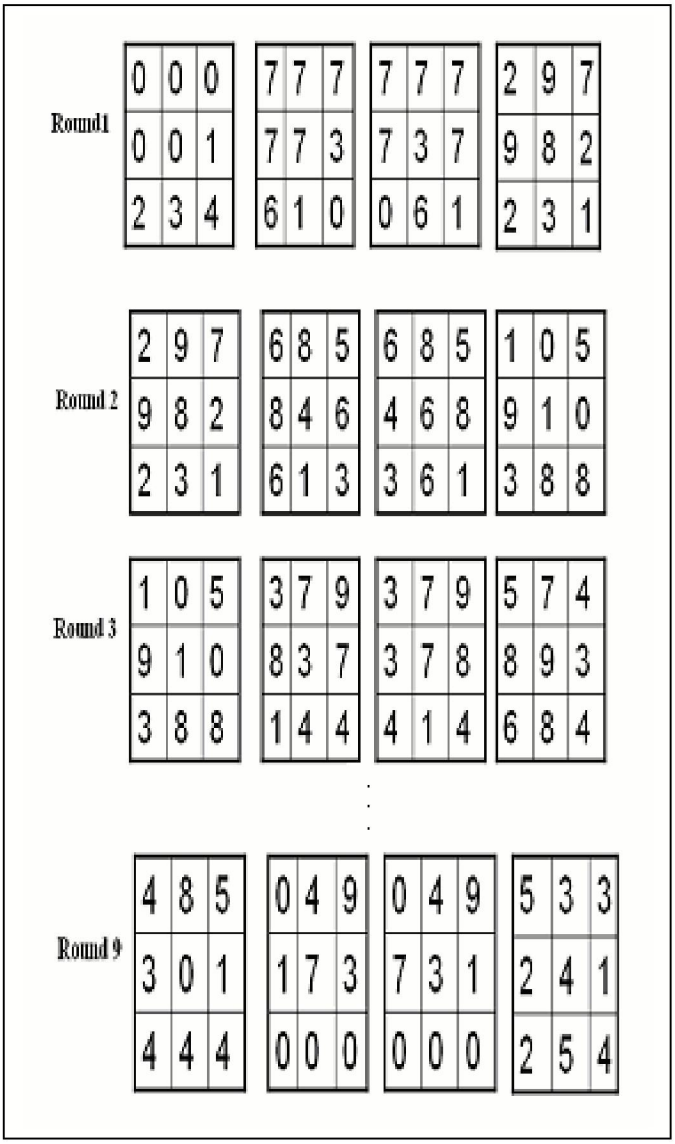

Figure 6. Encryption Process

Similarly the process of Decryption can be carried out as follows:

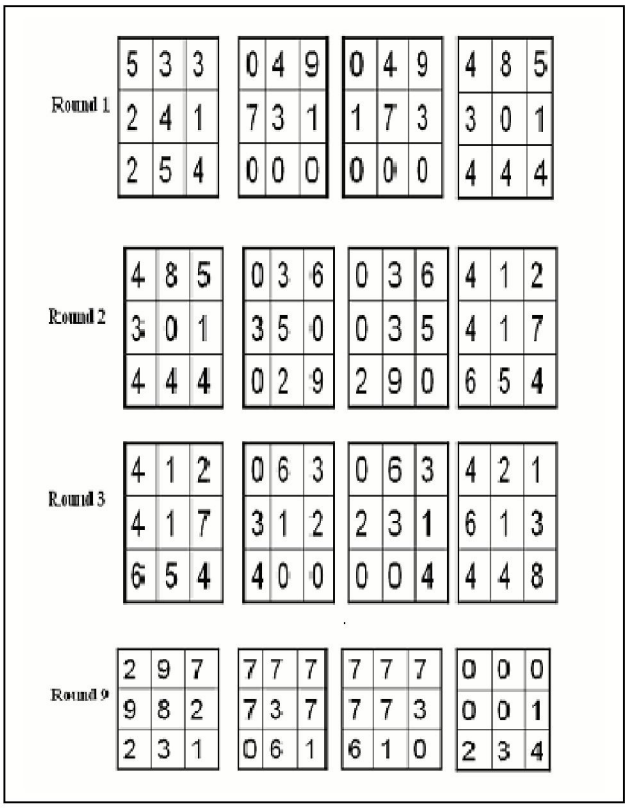

Figure 7. Decryption Process

STRENGTHS OF 3KDEC

- As the encryption is from numeric to numeric; one cannot know that the information is encrypted. It appears as if the encrypted data is itself the original content.
- As the number of keys are more and hence the key combination increases to 10^{27} which makes guessing of keys harder.
- As S-Box has simple structure and variable entries so the 3Kdec algorithm can be used as Personal Encryption Algorithm where different encryption processes can be carried out with varying degree of complexity depending on the user requirements.
- Since 3Kdec Algorithm encrypts numeric to numeric data, encrypted data is possible to be stored in the existing numeric field thereby algorithm does not change the data field type and set fixed data length.

V. 3KDEC ALGORITHM ANALYSIS

A. 3Kdec Algorithm Complexity

To analyze the computational complexity of a cryptographic algorithm we need an encryption algorithm to have low level of complexity and we need an algorithm used by the cryptanalyst who is trying to break the code to have high level of complexity. This means we need to perform encryption and decryption in short span of time, but at the same time we want the intruder to run her computer infinitely if trying to break the code.

Complexity of program is based on two types of resources: space complexity of algorithm which refers to amount of memory needed to store the algorithm and the data and the time complexity which refers to amount of time needed to run the algorithm and get the results.

The time complexity of 3Kdec algorithm can be measured (independent of the computer on which it is running) by defining the bit-operation complexity which counts the number of bit operations the computer needs to perform to create the output from the input.

Bit operation can be defined as the time required for the computer to add, subtract, multiply or divide two single bits or to perform one single bit shift.

For example: bit operation complexity of function that adds two integers each having, d decimal digits can be calculated as: $d \log_2 10$.

B. 3Kdec Algorithm Security

The main strength of 3Kdec algorithm is that it converts the numeric data input to numeric data output which makes it harder to see and guess whether the input is even encrypted or not.

1) Brute- Force attack

Also known as Exhaustive key search method tries to use all possible keys. As in our algorithm the three keys are expanded based on the number of rounds; there will be 10^{27} different key combinations. Lack of weak keys is another advantage of 3Kdec algorithm.

2) Statistical Attacks

The strong confusion and diffusion provided by the Substitution Box lookup and Row Shift transformations removes any frequency pattern in the plaintext.

3) Differential and Linear Attack

The algorithm was designed to be resilient to linear and differential attacks.

Differential cryptanalysis/Chosen-plaintext attack needs the analysis of encryption algorithm to collect information about plaintext-ciphertext relationships; with the goal of finding the cipher key [11]. The probabilistic relationship can be created from the information about the S-box input/output table. Incase of 3Kdec algorithm the S-box values are not fixed and does not depend on any algebraic structure the sender may use different S-box values each time attempting to encrypt the input. So the

attacker has to even guess what were the values used for the S-Box in encrypting that data.

Linear Cryptanalysis/Known-Plaintext Attack where the attacker needs to have information about plaintext/ciphertext pairs in addition to the intercepted ciphertext that needs to be broken along with the assumption that key has not changed. Relationship between previous pair is used to analyze the current ciphertext.But the strong point of our algorithm is that every time the three keys as well as S-box entries can be changed which makes it difficult to find the appropriate relationships among the plaintext and the ciphertext pair.

C. *Simplicity and Cost*

The 3Kdec algorithm has a simple structure that it can be easily implemented using cheap processors and minimum amount of memory.

VI. CONCLUSION

Understanding the need to secure your data is the first step towards securing it. In today's age every detail – personal to corporate secrets – is present in form of data. For computers and networks which store and transfer this data, it is just numbers. It is for us to realize the damage this data can do if it falls into the hands of an unscrupulous person. Whether the data is on your laptop, desktop, or on an organizations storage network, it must be secured and should not come in the hand of an unauthorized entity. Proper access control mechanism should be enforced for securing the data. While in motion, data should be well protected. It is advisable to encrypt the data before putting it on a network even if it passes through a secure channel. The algorithm can be implemented for securing any corporate related accounting information to data of personal use. This algorithm does not take into consideration decimal point numeric data which opens up another area of research and improvements.

VII. REFERENCES

[1] B. Schneier. Applied Cryptography. John Wiley & Sons, Inc., 1996.

[2] Behrouz A.Forouzan.Cryptography and Network Security, Tata McGraw Hill.

[3] G. Davida, D. Wells, and J. Kam. A database encryption system with Sub keys. ACM Transactions on Database Systems, 6(2), 1981.

[4] J. He and M. Wang. Encryption in relational database management systems. In Proc. Fourteenth Annual IFIP WG 11.3 Working Conference on Database Security (DBSec'00), School, the Netherlands, 2000.

[5] Koblitz, N., A Course in Number Theory and Cryptography. New York: Springer-Verlag, 1988.

[6] Maurer, U., "The Role of Cryptography in Database Security," ACM SIGMOD, 2004.

[7] Noor Habibah Arshad, Saharbudin Naim Tahir Shah, Azlinah Mohamed, Abdul Manaf Mamat," The Design and Implementation of Database Encryption", International Journal Of Applied Mathematics And Informatics, Issue 3, Volume 1, 2007,page115- page122

[8] Rakesh Agrawal Jerry Kiernan Ramakrishnan Srikant Yirong Xu," Order Preserving Encryption for Numeric Data", IBM Almaden Research Center

[9] RSA Security, Inc., "Securing Data at Rest; Developing a Database Encryption Strategy," White Paper, 2002.

[10] SK Bhatnagar,"Securing Data-At-Rest", Literature by Tata Consultancy Services.

[11] Srdjan Holovac "Securing Data at Rest: Database Encryption Solution using Empress RDBMS", Security Whitepaper.