www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN:2319-7242*
*Volume 2 Issue 4 April, 2013 Page No. 1072 -1076*

# Secure Migration of Various Database over A Cross Platform Environment

***R.Vinodha, Mr.R.Suresh***
M.Tech Networking,
, Associated Prof. Dept of IT
Sri Manakula Vinayagar Engineering College, Pondicherry.
rsvinodhaannai@gmail.com, sureshramanujam78@gmail.com

***ABSTRACT--****In this paper we aims to bring up the idea of different platform environment to compare a heterogeneous database system, where it may use different ways of storing like file formats, protocols, query language. The biggest challenge for any organization occurs during migration of large databases, which easily consume more terabytes. In different RDBMS, i.e. oracle database 10g, DB2, SQL Server, MySql and MS access etc where those database are not available previously for high speed migration and also performance is increased. This migration is independent of different platform and secured by strong cryptographic algorithm, in which the entire data conversion becomes reliable, fast and efficient. In this paper, we utilizing modified RSA algorithm with Armstrong numbers as input to make it more secured.*

## I. INTRODUCTION

Data migration is the process of transferring data between storage types, formats, or computer systems. Data migration is usually performed programmatically to achieve an automated migration, freeing up human resources from tedious tasks. It is required when organizations or individuals change computer systems or upgrade to new systems, or when systems merge. This can involve entering the data manually, moving disk files from one folder (or computer) to another, database insert queries, developing custom software, or other methods.

The specific method used for any particular system depends entirely on the systems involved and the nature and state of the data being migrated. A blob is a data type that can store binary data. This is different than most other data types used in databases, such as integers, floating point numbers, characters, and strings, which store letters and numbers.

Since blobs can store binary data, they can be used to store images or other multimedia files. For example, a photo album could be stored in a database using a blob data type for the images, and a string data type for the captions. Because blobs are used to store objects such as images, audio files, and video clips, they often require significantly more space than other data types. The amount of data a blob can store varies depending on the database type, but some databases allow blob sizes of several gigabytes.

The relational database should be converted into a format previously to transform that is understandable to each host in a Cross-platform environment. The database format that is used is XML, as it is platform independent. It is needed to preserve schema relational database while transferring relational database. So we are using XML format transferring data, which can support any platform. Before transfer, xml file is encrypted. At client-side, received file is decrypted and parsed to extract the original data. This data is integrated into one of the client-side RDBMSs specified by the user.

## II. SYSTEM ARCHITECTURE

The basic system architecture which consists of server-side and client-side. Server-side is the source of information from where data is migrated. Client side is the receiver of information which requests connection with the server for migration of data. At the server-side, files are transferred in encrypted form, represented by EC.

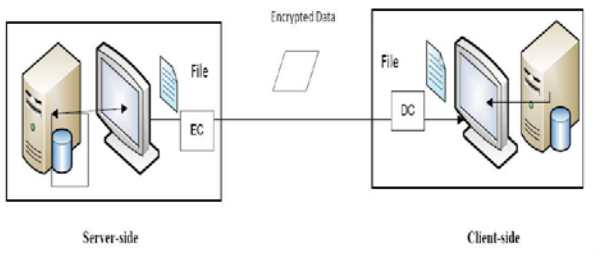


Fig 1. System architecture.

The transferred file is decrypted at the client-side represented by DC. Large amount of work has already been done on distributed database systems. Many algorithms have been implemented to achieve dynamic fragmentation and object allocation in distributed databases. Complexities arouse while considering heterogeneous database systems in

which sites are unaware of each other, database software and schemas used by the different systems may be different.

A. *SERVER-SIDE*

Following steps are involved at the server side during the transfer of data: 1) Extraction of a table for migration, 2) Converting the table into a format that is useful in efficient transfer and specifying the client side RDBMS details, 3) Establishing the connection with the server which is in waiting state (Pull Mechanism), 4) Transferring the table and record information along with client side RDBMS details.

B. *CLIENT-SIDE*

Following steps are involved at the client-side while receiving and integrating the data: 1) receiving the files sent by the server in the same format, 2) Parsing the XML file, 3) manipulation of the attribute information and Integration into the client-side RDBMS

## III. OVERVIEW OF TRANSFORMATION

***1 .Pre –migration transformation***

In pre-migration transformation some transformational activities are done previously before migration to aid the migration. This activities include server virtualization, data separation or server platform upgrades. The main purpose of this is to make transformation easier by changing data into required format. From a DCM point of view, it is worth restricting remigration transformational activities to only those that make the migration easier, faster or less risky.

***2. Migration transformation***

In a few instances, performing transformational activities during the migration events themselves might be required. To reduce complexity and risk, these activities should be kept to an absolute minimum. Where activities are performed, it is imperative that the ability to back out from the migration is retained should significant problems occur during the migration weekend.

***3. Post- Migration Transformation***

In this performing transformational activity after the migration has completed is a common requirement. Once the migration services have been successfully transitioned to Business as Usual, the Data Centre Migration programmed should wind-down. Any post-migration activities should be managed not as a programmed extension but as a separate project.

## IV. CRYPTOGRAPHY

Cryptography, to most people, is concerned with keeping communications private. Encryption is the transformation of data into some unreadable form. Its purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended. Decryption is the reverse of encryption; it is the transformation of encrypted data back into some intelligible form. Encryption and decryption require the use of some secret information, usually referred to as a key. The data to be encrypted is called as plain text. The encrypted data obtained as a result of encryption process is called as cipher text. Depending on the encryption mechanism used, the same key might be used for both encryption and decryption, while for other mechanisms, the keys used for encryption and decryption might be different. same key might be used for both encryption and decryption, while for other mechanisms, the keys used for encryption and decryption might be different.

**A. Types of Cryptographic Algorithms**

There are several ways of classifying cryptographic algorithms. In general they are categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use as in .The three types of algorithms are depicted as follows

***1) Secret Key Cryptography (SKC):*** Uses a single key for both encryption and decryption. The most common algorithms in use include Data Encryption Standard (DES), Advanced Encryption Standard (AES).

***2) Public Key Cryptography (PKC):*** Uses onekey for encryption and another for decryption.RSA (Rivest, Shamir, Adleman) algorithm isan example.

***3) Hash Functions****:* Uses a mathematical transformation to irreversibly "encrypt" information. MD (Message Digest) algorithm is an example.

## V. SERVER ARCHITECTURE

A Server is a computer or device on a network that manages network resources. For example, a file server is a computer and storage device dedicated to storing files. Any user on the network can store files on the server. Servers are often dedicated, meaning that they perform no other tasks besides their server tasks. On multiprocessing operating systems however, a single computer can execute several programs at once. A server in this case could refer to the program that is managing resources rather than the entire computer.

***A. What is Server Platform?***

A term often used synonymously with operating system. A platform is the underlying hardware or software for a system and is thus the engine that drives the server.

***B. Types of server***

***1]FTP-Servers***

One of the oldest of the Internet services, File Transfer Protocol makes it possible to move one or more files securely between computers while providing file security and organization as well as transfer control.

***2]Mail-Servers***

Almost as ubiquitous and crucial as Web servers, mail servers move and store mail over corporate networks via LANs and WANs and across the Internet.

***3] Print-server***

It is a computer that manages one or more printers and a network server is a computer that manages network traffic. There are so many servers according to requirement like Audio/video, Chat, Fax, News, Proxy,Web servers etc.

## VI. PROPOSED SYSTEM

***A. Introduction***

In proposed approach we maintain server database with following fields-Unique name and id of sender and receiver, and encrypted key(Armstrong Number).

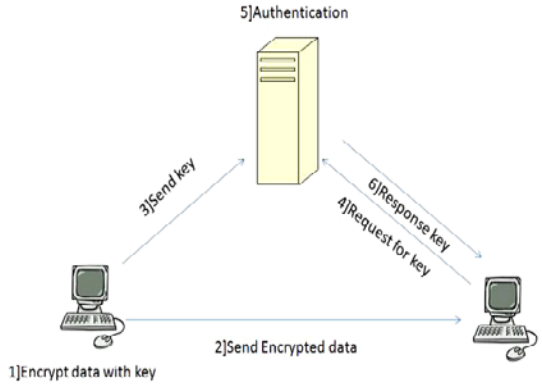


**Fig 3. Server Architecture**

Now, if sender „A" wants to send data to receiver „B", then he encrypts that data using randomly generated Armstrong number. That encrypted data is identified by unique timestamp given to it and sent to receiver. At the same time key (Armstrong Number) of encrypted data is sent to server with receiver "B" id and file name. Whenever receiver get that encrypted data he simply request for key to server. Now actual authentication is done by server, Server takes request from receiver with file name and receivers self id, and compare it with senders key name and receiver id. If both match then only that key is send to the receiver. Whenever receiver
gets key now he can decrypt that data easily.

***B. Illustration***
***1] Encryption:***

**Step 1**: Unimodular matrix is used to create encoding matrix given below. Take random Armstrong Number and add its total digits like. (n=1+5+3=9) and substitute it in Unimodular matrix as below

$$\begin{pmatrix} 8n^2+8n & 2n+1 & 4n \\ 4n^2+4n & n+1 & 2n+1 \\ 4n^2+4n+1 & n & 2n-1 \end{pmatrix}$$

After calculation Encoding matrix is

| | | |
|---|---|---|
| 720 | 19 | 36 |
| 360 | 10 | 19 |
| 361 | 9 | 17 |

**Step 2**: (Encryption of the actual data begins here)

Let the message to be transmitted be "ENCRYPT". First find the ASCII equivalent of the above characters.

| E | N | C | R | Y | P | T | Extra | Extra |
|---|---|---|---|---|---|---|---|---|
| 69 | 78 | 67 | 82 | 89 | 80 | 84 | -25 | -25 |

**Step 3:** Now add these numbers with the digits of the Armstrong number Encrypted matrix as follows:

| E | N | C | R | Y | P | T | Extra | Extra |
|---|---|---|---|---|---|---|---|---|
| 69 | 78 | 67 | 82 | 89 | 80 | 84 | -25 | -25 |
| +720 | 19 | 36 | 360 | 10 | 19 | 361 | 9 | 17 |
| 789 | 97 | 103 | 442 | 99 | 99 | 445 | -16 | -8 |

**Step 4:** Convert the above data into a matrix as follows:

$$A = \begin{pmatrix} 789 & 97 & 103 \\ 442 & 99 & 99 \\ 445 & -16 & -8 \end{pmatrix}$$

**Step 5:** Consider an encoding matrix...

$$B = \begin{pmatrix} 720 & 19 & 36 \\ 360 & 10 & 19 \\ 361 & 9 & 17 \end{pmatrix}$$

**Step 6:** After multiplying the two matrices (B * A) we get

$$C = \begin{pmatrix} 54262 & 56951 & 48860 \\ 27256 & 28495 & 24445 \\ 27075 & 28534 & 24482 \end{pmatrix}$$

The encrypted data is...

54262, 56951, 48860, 27256, 28495, 24445, 27075, 28534, 24482

The above values represent the encrypted form of the given message.

After storing this data into file it will be converted into byte array format as below:

-10, 119, -36, 120, 79, 125, -61, 118, -94.

***2] Decryption:***

Decryption involves the process of getting back the original data using decryption key.

**Step 1:(**Decryption of the original data begins here)
The inverse of the encoding matrix is:

$$D = \begin{pmatrix} -1 & 1 & 1 \\ 43363 & -43508 & -43216 \\ -21682 & 21755 & 21608 \end{pmatrix}$$

**Step 2**: Multiply the decoding matrix with the encrypted data (C*D)

**Step 3**: Now transform the above result as given below:

789, 97, 103, -53830, -56733, -53405, 27581, 28400, 26872.

**Step 4**: Subtract with the digits of the Armstrong numbers as follows:

```
  789  97  103  -53830  -56733  -53405  27581  28400  26872
+720 19  36     360      10       19      361    9        17
-------------------------------------------------------------------------
   69  78  67  -54190 -  56743  -53424  27220  28391  26855
```

**Step 5**: After converting the above data into byte array format and removing the extra parity bits we will get the original data.

69 78 67 82 89 80 84

**Step 6**: Obtain the characters from the above ASCII equivalent:

```
E   N   C   R   Y   P   T
69  78  67  82  89  80  84
```

## VII. ADVANTAGES

In above algorithm, Unimodular matrix is used to reduce the loss of data during encryption and decryption process. This encryption technique ensures that the data transfer can be performed with protection since it involves two main steps.

First step is to convert the characters into another form that means in ASCII values, Second step by adding with the digits of the Encoding matrix to form the required encrypted data.

Tracing process becomes difficult with this technique. This is because data is encrypted by key using Armstrong number and again this Armstrong number is encrypted by using receiver‟s key. So it is more secure. In this proposed technique encryption algorithm is too difficult to trace or hack externally. Server plays vital role in authentication process.

## VIII. FUTURE WORK AND ADVANTAGES

The above technique involves keys with a minimum length of 8 bits for Armstrong numbers. This minimum key length reduces the efforts taken to encrypt the data. The key length can be increased if needed, with increase in character length.

This increases the complexity thereby providing increased security. This technique ensures that the data transfer can be performed with protection since it involves two main steps. First step is to convert the characters into another form, by adding with the digits of the Armstrong numbers. Second step is to encode using a matrix to form the required encrypted data.

Tracing process becomes difficult with this technique. This is because the Armstrong number is used differently in each step. The key can be hacked only if the entire steps involved in the encoding process are known earlier.

This technique could be considered as a kind of triple DES algorithm since we use three different keys namely the colors, key values added with the colors and Armstrong numbers.

Unless all the three key values along with the entire encryption and decryption technique is known the data cannot be obtained. So hacking becomes difficult mainly because of the usage of colors. Simple encryption and decryption techniques may just involve encoding and decoding the actual data. But in this proposed technique the password itself is encoded for providing more security to the access of original data.

## IX.CONCLUSION

Secure transmission of data efficiently and without any kind of delay due to buffering. The XML format, since it can support any platform any kind of database can be transmitted using it which will save time of conversion of data before transferring to client. The above discussed work can be used for secure transmission of data efficiently and without any kind of delay due to buffering. The XML format, since it can support any platform any kind of database can be transmitted using it which will save time of conversion of data before transferring to client. The data which we are migrating from source to destination i.e., between heterogeneous platforms is sensitive (carrying information about database table and its corresponding schema), so it has to be secured. As multiple users can access the server simultaneously, synchronization has to be done.

## X. REFERENCE


[1] J. Zhiquan, L. Chengfei, S. Zhongxiu, Z. Xiaofang, C. Peipei, and G.Jianming, “Design and Implementation of a heterogeneous distributed database system,” in Journal of Computer Science and Technology, published by Springer Boston, vol. 5, no. 4, pp. 363-373.

[2]. P. Kokkinos, K. Christodoulopoulos, A. Kretsis, and E. Varvarigos, “Data Consolidation: A Task Scheduling and Data Migration Technique for Grid Networks,” in Eighth IEEE International Symposium on Cluster Computing and the Grid, 2008.

[3]. Lixian xing,Yanhong li,“Design And Application Of Data Migration In Heterogeneous Database,” IEEE International Forum On Information Technology And Applications,2010**.**

[4]. S.Belose, M.Malekar, G.Dharmawat, “Data Security Using Armstrong Numbers” International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, Volume 2, Issue 4, April 2012.

[5]. [1]. Chadi Kari, Yoo-Ah Kim, Alexander Russell, “Data Migration In Heterogeneous Storage System”, IEEE International conference on Distributed Computing Systems , DOI 10.1109/ICDCS, 2011.

[6]. Qingni Shen, Lizhe Zhang, Xin Yang, Yahui Yang, Zhonghai Wu, Ying Zhang, "SecDM: Securing Data Migration Between Cloud Storage Systems", IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, DOI 10.1109/DASC, 2011.

[7]. Lixian Xing, Yanhong Li, "Design And Application Of Data Migration System In Hetertogeneous Database", IEEE International forum on Information Technology and Applications, DOI 10.1109/IFITA, 2010.

[8]. P. Kokkinos, K. Christodoulopoulos, A. Kretsis, and E. Varvarigos, "Data Consolidation: A Task Scheduling and Data Migration Technique for Grid Networks," in Eighth IEEE International Symposium on Cluster Computing and the Grid, 2008.

[9]. Rand Bradley, "Push to Pull: How Lean Concepts Improve a Data Migration", IEEE computer society ,0-7695-2872-4/07, 2007.

[10]. Vijay Sundaram, Timothy Wood, Prashant Shenoy," Efficient Data Migration in Self-managing Storage Systems",IEEE Dept. of Computer Science,1-4244-0175-5/06, 2006.

[11]. Jiahong Wang, Norihisa Segawa, Masatoshi Miyazaki," On-Line Data Migration Approaches and Their Performance Comparisons",IEEE Software and Information Science, 0-7803-7080-5/01, 2001.