Journal of Engineering and Applied Science *Volume 4, September 2012*

© 2012 Cenresin Publications
www.cenresinpub.org

SECURITY OF DATABASE CONTENTS USING TRANSPARENT DATA ENCRYPTION IN MICROSOFT SQL SERVER ENTERPRISE EDITION

Rashid Husain
Department of Computer Science
Kebbi State University of Science and Technology, Aleiro, Kebbi State
Email: rashid65_its@yahoo.com

ABSTRACT

We can take several precautions to help secure the database such as designing a secure system, encrypting confidential assets, and building a firewall around the database servers. However, in a scenario where the physical media (such as drives or backup tapes) are stolen, a malicious party can just restore or attach the data base and browse the data. To protect against data thefts and frauds we require security solutions that are transparent by design. Transparent Data Encryption (TDE) provides transparent, standards based security that protects data on the network, on disk and on backup media. It is easy and effective protection of stored data by transparently encrypting data. TDE performs real-time I/O encryption and decryption of the data and log files. The encryption uses a data base encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module. TDE protects data 'at rest', meaning the data and log files. It provides the ability to comply with many laws, regulations, and guidelines established in various industries. The study deals with ways to create Master key, creation of certificate protected by the master key, creation of database master key and protection by the certificate and ways to set the database to use encryption in Microsoft SQL Server.

Keywords: *Transparent Data Encryption, TDE, Encryption, Decryption, Microsoft SQL Server, Key, Symmetric, Asymmetric.*

INTRODUCTION

Now-a-days life is vastly driven by Information Technology. Extensive use of IT is playing an important role in decision making in all commercial and non- commercial organizations. All activities are centered on data, its safe storage and manipulation. Data is vulnerable to a wide range of threats like, Weak Authentication, Backup Data Exposure, Denial of Service, etc. (Husain Rashid, 2012) This study is aimed at to deal with the most critical of those threats to which database is vulnerable. TDE shields database up to considerable extent against such treats. TDE is used to prevent unauthorized access to confidential database, reduce the cost of managing users and facilitate privacy managements. These latest technology arms users i.e. database administrators to solve the possible threats to security of data. This technology allows encrypting database on hard disk and on backup media. TDE is one way to encrypt database content. TDE offers encryption at a column, table, and table space level. This makes TDE one of the more highly configurable ways to encrypt database

content, though some of these configuration options come with a performance price. (Deshmukh and Qureshi, 2011)
TDE, nowadays, is the best possible choice for bulk encryption to meet regulatory compliance or corporate data security standards.
TDE is available as an "enterprise level" feature of Oracle Enterprise Edition and Microsoft SQL Server Enterprise Edition. (MSDN Library)

MATERIAL AND METHODS

Encryption and Decryption

Encryption is said to occur when data is passed through a series of mathematical operations that generate an alternate form of that data; the sequence of these operations is called an algorithm. To help distinguish between the two forms of data, the unencrypted data is referred to as the plaintext and the encrypted data as cipher text. Encryption is used to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting cipher text to its original plaintext is called decryption. (Walters and Kirish, 2010) This process is illustrated in the Figure 1 below:

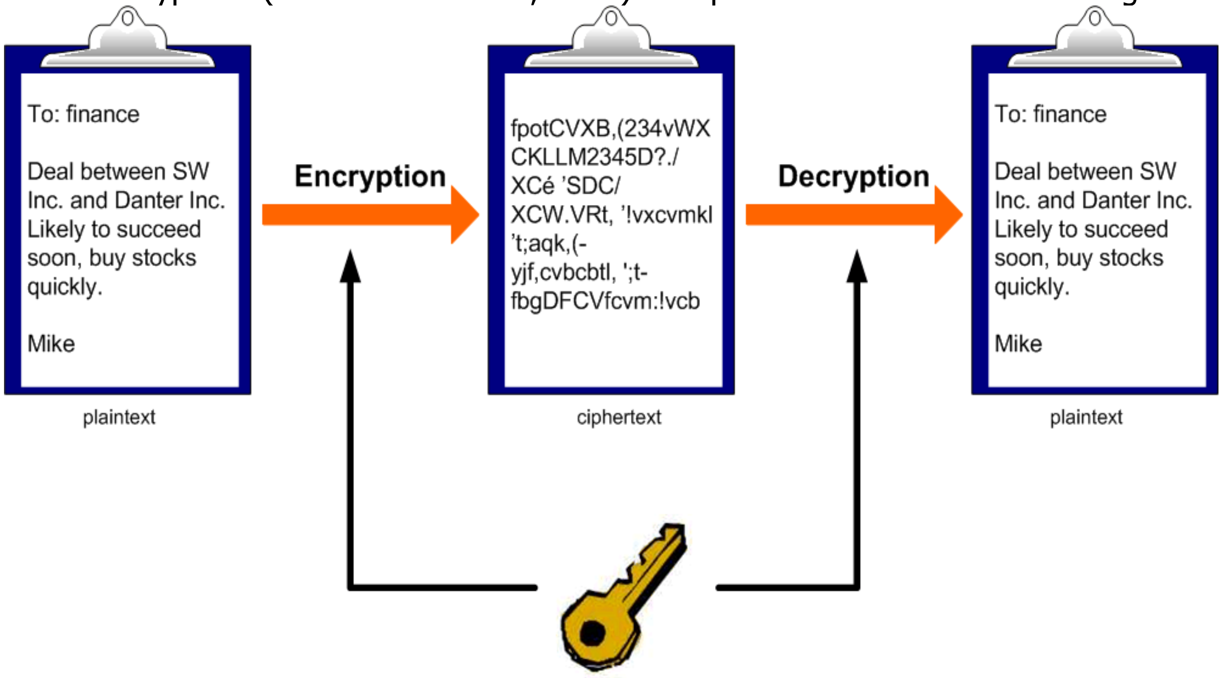
To: finance
Deal between SW Inc. and Danter Inc. Likely to succeed soon, buy stocks quickly.
Mike
plaintext
Encryption
fpotCVXB,(234vWX CKLLM2345D?./ XCé 'SDC/ XCW.VRt, '!vxcvmkl 't;aqk,(- yjf,cvbcbtl, ';t- fbgDFCVfcvm:!vcb
ciphertext
Decryption
To: finance
Deal between SW Inc. and Danter Inc. Likely to succeed soon, buy stocks quickly.
Mike
plaintext

Fig. 1: Process of Encryption and Decryption

Examples:

- Encryption of the word "abcd" could result in "wxyz" Reversing the order of letters in the plain text generates the cipher text.
- It is simple encryption and quit easy for attacker to retrieve the original data. A better solution encrypting this message is to create an alternative alphabet by shifting each letter by arbitrary number.

- Example: the word “abcd” is encrypted by shifting the alphabet by 4 letters to right then the result will be “efgh”. The “Ceasar Cipher” means exchange of letters or words with another.
- Normal Alphabet: abcdefghijklmnopqrstuvwxyz.
- Alphabet Shifted by 4: efghijklmnopqrstuvwxyzabcd.

The sender uses an encryption algorithm and the receiver uses a decryption algorithm.

2. Types of Encryptions

There are two types of encryption: symmetric key encryption and public (asymmetric) key encryption. Symmetric key and public key encryption are used, often in conjunction, to provide a variety of security functions for network and information security. (Deshmukh and Qureshi, 2011)

a) Symmetric Key Encryption

Encryption algorithms that use the same key for encrypting and for decrypting information are called symmetric-key algorithms. The symmetric key is also called a secret key because it is kept as a shared secret between the sender and receiver of information. Otherwise, the confidentiality of the encrypted information is compromised. Figure 2 shows basic symmetric key encryption and decryption. (Andrew S, 2010)

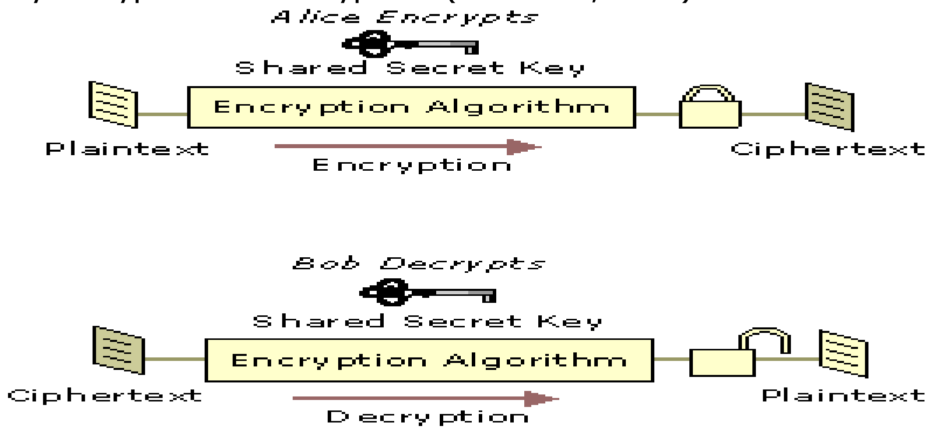

Fig.2: Encryption and Decryption with a Symmetric Key

b) Public (Asymmetric) Key Encryption

Encryption algorithms that use different keys for encrypting and decrypting information are most often called public-key algorithms but are sometimes also called asymmetric key algorithms . Public key encryption requires the use of both a private key (a key that is known only to its owner) and a public key (a key that is available to and known to other entities on the network). A user's public key, for example, can be published in the directory so that it is accessible to other people in the organization. The two keys are different but complementary in function. Information that is encrypted with the public key can be decrypted only with the corresponding private key of the set. Figure 3 shows basic encryption and decryption with asymmetric keys. (Andrew. S, 2010)

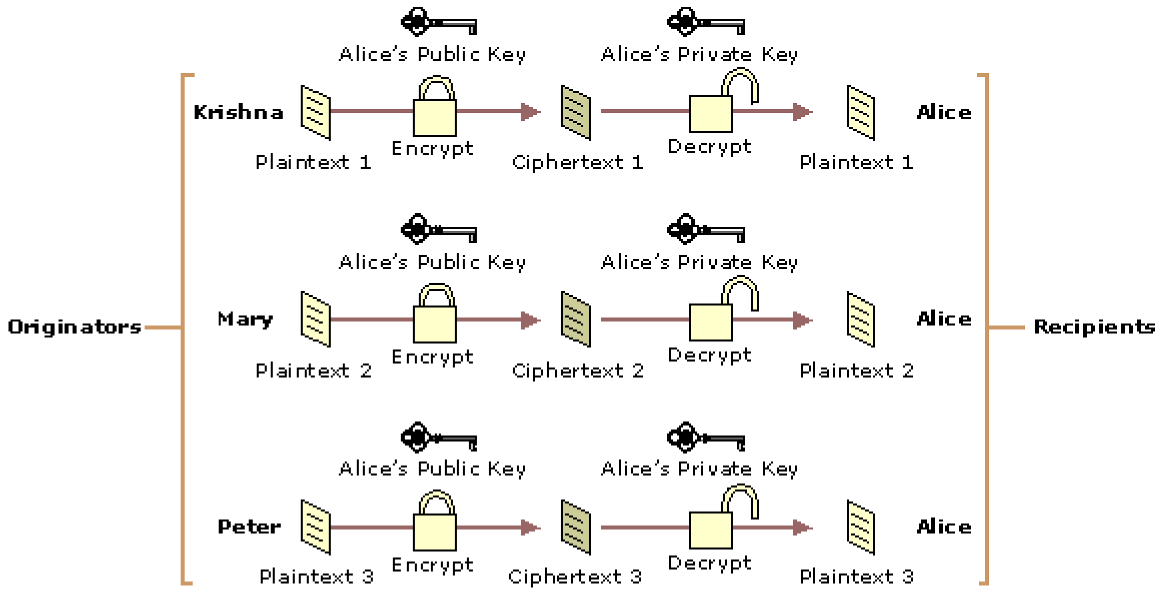

Fig.3: Encryption and Decryption with Asymmetric Keys

3. Transparent Data Encryption (TDE)

TDE is used for encryption and decryption of the data and log files. The encryption uses a Database Encryption key (DEK), which is stored in the database boot record for availability during recovery. It is Asymmetric key secured by using a certificate stored in the master database. TDE protects data and log files. It is a technology used to solve the problems of security of data means encrypting database on hard disk and on any backup media. TDE can be used to provide high levels of security to columns, table and tablespace that is database files stored on hard disk or CD's, and other information that requires protection. It is the technology used by Microsoft SQL Server and Oracle to encrypt database contents. (Cherry, 2011; MSDN Library) TDE encrypts data before it's written to disk and decrypts data before it is returned to the application. The encryption and decryption process is performed at the SQL layer, completely transparent to applications and users. Subsequently backups of the database files to disk or tape will have the sensitive application data encrypted. (Michel Otey, 2008)

4. Architecture of Transparent Data Encryption in Microsoft SQL Server 2008

In Microsoft SQL Server 2008 Transparent Data Encryption of the database file is performed at the page level. The pages in an encrypted database are encrypted before they are written to disk and decrypted when read into memory. Transparent data protection does not increase the size of the encrypted database. (Deshmukh and Qureshi, 2011)

a) To use TDE, follow these steps: (MSDN Library)

- Create a master key
- Create or obtain a certificate protected by the master key.
- Create a database encryption key and protect it by the certificate

- Set the database to use the encryption

The following illustration shows the architecture of TDE encryption:

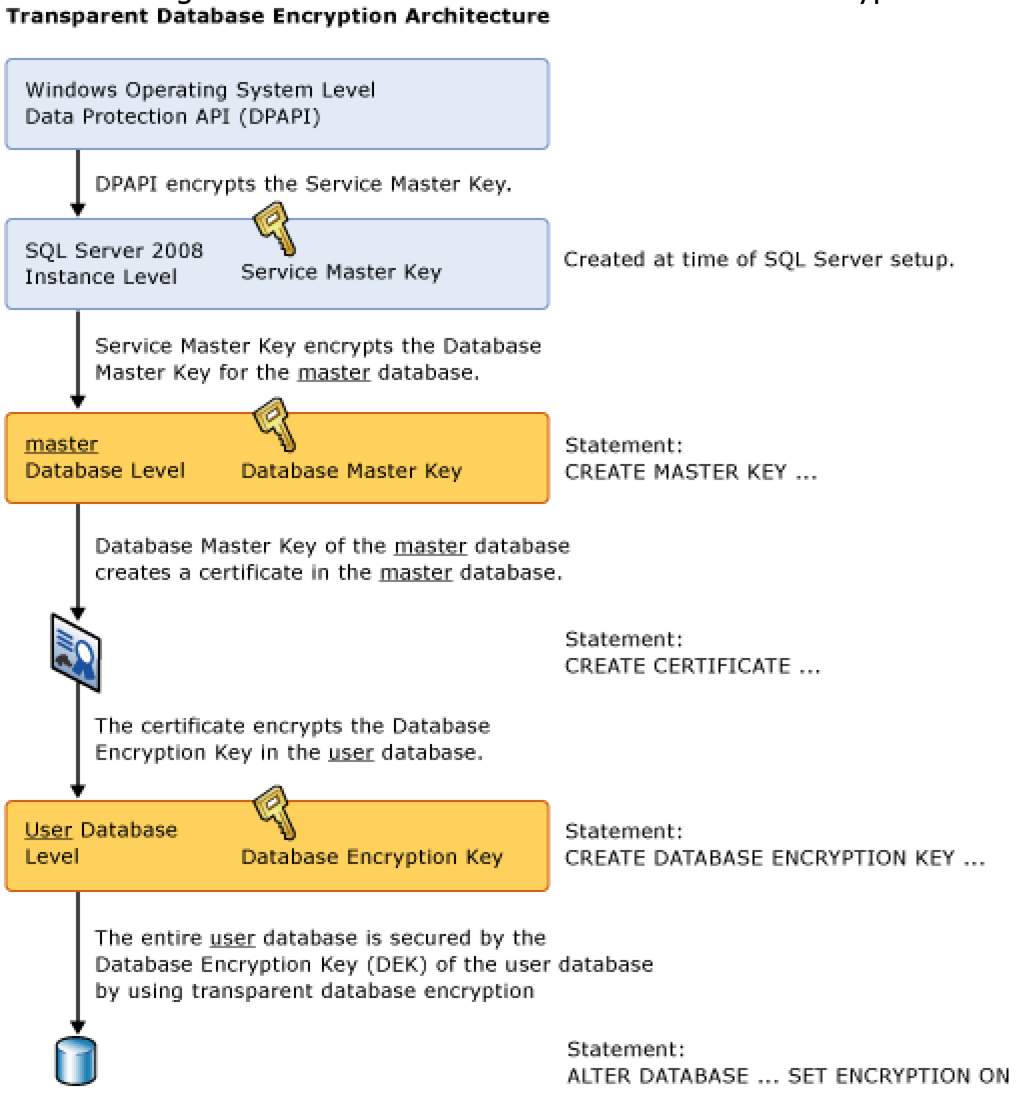
Transparent Database Encryption Architecture
Windows Operating System Level
Data Protection API (DPAPI)
DPAPI encrypts the Service Master Key.
SQL Server 2008
Instance Level
Service Master Key
Created at time of SQL Server setup.
Service Master Key encrypts the Database Master Key for the master database.
master
Database Level
Database Master Key
Statement:
CREATE MASTER KEY ...
Database Master Key of the master database creates a certificate in the master database.
Statement:
CREATE CERTIFICATE ...
The certificate encrypts the Database Encryption Key in the user database.
User Database
Level
Database Encryption Key
Statement:
CREATE DATABASE ENCRYPTION KEY ...
The entire user database is secured by the Database Encryption Key (DEK) of the user database by using transparent database encryption
Statement:
ALTER DATABASE ... SET ENCRYPTION ON

Fig.4: Transparent Database Encryption Architecture

In Microsoft SQL Server 2008 Transparent Data Encryption of the database file is performed at the page level. The pages in an encrypted database are encrypted before they are written to disk and decrypted when read into memory. Service Master Key is created at a time of SQL server setup, DPAPI encrypts the service Master key. (Walters and Kirisch, 2010)Service Master Key encrypts database Master key for the Master Database. The Database Master Key of the Master Database creates the certificate then the certificate encrypts the database encryption key in the user database. The entire database is secured by the database Master key of the user database by using Transparent Data Encryption. (Micheal Otey, 2008)

b) The following example illustrates encrypting and decrypting the AdventureWorks2012 database using a certificate installed on the server named MyServerCert. (MSDN Library)

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<UseStrongPasswordHere>';
go
CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'My DEK Certificate';
go
USE AdventureWorks2012;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO
ALTER DATABASE AdventureWorks2012
SET ENCRYPTION ON;
GO
```

The encryption and decryption operations are scheduled on background threads by SQL Server. You can view the status of these operations using the catalog views and dynamic management views.

c) The following table shows TDE catalog views and dynamic management views. (MSDN; Andrew. S, 2010)

Catalog view or dynamic management view	**Purpose**
sys.databases (Transact-SQL)	Catalog view that displays database information.
sys.certificates (Transact-SQL)	Catalog view that shows the certificates in a database.
sys.dm_database_encryption_keys (Transact-SQL)	Dynamic management view that provides information about the encryption keys used in a database, and the state of encryption of a databa

d. The following table provides links and explanations of TDE commands and functions.

Command or function	**Purpose**
CREATE DATABASE ENCRYPTION KEY(Transact-SQL)	Creates a key that is used to encrypt a database.
ALTER DATABASE ENCRYPTION KEY (Transact-SQL)	Changes the key that is used to encrypt a database.
DROP DATABASE ENCRYPTION KEY (Transact-SQL)	Removes the key that was used to encrypt a database.
ALTER DATABASE SET Options (Transact-SQL)	Explains the ALTER DATABASE option that is used to enable TDE.

5. Scope of Transparent Data Encryption

Extensive use of IT is playing vital role in decision-making in commercial as well as non-commercial organizations. In present scenario operations of any organization are badly affected if data is misused. Data is vulnerable to a wide range of threats like, Excessive privilege Abuse, Legitimate privilege Abuse, Weak Authentication, Backup Data Exposure, etc. (Deshmukh and Qureshi, 2011; Oracle Advanced security, 2007) The crucial role data plays in any organization itself explains its importance. So it is exposed to grave threats of theft, misuse or loss. This research is aimed to deal with the most critical of those threats to which database is vulnerable by focusing on TDE. TDE is used to prevent unauthorized access to confidential database, reduce the cost of managing users, and facilitate privacy managements. (Deshmukh and Qureshi, 2011) This latest technology enables users i.e. database administrators to counter the possible threats to security of data. TDE facilitates encrypting database on hard disk and on any backup media. TDE these days is the best possible choice for bulk encryption to meet regulatory compliance or corporate data security standards.

6. Use of Transparent Data Encryption

There are three important uses of Transparent Data Encryption as below (Michel Otey, 2008; Andrew. S, 2010)

1. Authentication
2. Validation
3. Data Protection

a) Authentication:

Authentication deals with the problem of verifying the identity of a user before permitting access to the requested resources. That is, an authentication mechanism prohibits the use of the system by unauthorized users by verification the identity of a user making a request. Microsoft SQL Server 2008 Advanced Security provides the ability for businesses to leverage their existing security infrastructures such as encrypt Master key, Database Master Key and Certificate.

b) Validation:

Validation describes the ability to provide assurance that a senders identity is true and that a column, Tablespace or file has not been modified. Encryption can be used to provide validation by making a digital certificate of the information contained within a database. Upon validation, the user can be reasonably sure that the data came from a trusted person and that the contents of the data have not been modified.

c) Data Protection:

Probably the most widely-used application of transparent encryption is in the area of data protection. The information that a business owns is invaluable to its productive operation, consequently, the protection of this information is very important. For people working in small offices and home offices, the most practical uses of transparent encryption for data protection are column, Tablespace and files encryption. This information protection is vital in the event of theft of the computer itself or if an attacker successfully breaks into the system. The encryption and decryption process is performed at the SQL layer, completely transparent to applications and users.

A. Problem and Discussion

1. Limitation of Transparent data encryption (Deshmukh and qureshi, 2011; Carl, Zvonko and zaky, 2001)

i. Transparent data Protection does not provide encryption across communication channels.

ii. When enabling transparent data Protection, you should immediately back up the certificate and the private key associated with the certificate. If the certificate ever becomes unavailable or if you must restore or attach the database on another server, you must have backups of both the certificate and the private key or you will not be able to open the database.

iii. The encrypting certificate or Asymmetric should be retained even if transparent data Protection is no longer enabled on the database. Even though the database is not encrypted, the database encryption key may be retained in the database and may need to be accessed for some operations.

iv. Altering the certificates to be password-protected after they are used by transparent data Protection will cause the database to become inaccessible after a restart.

v. The following operations are not allowed during initial database encryption, key change, or database decryption: (MSDN Library)

- Dropping a file from a file group in the database
- Dropping the database
- Talking the database offline
- Detaching a database
- Transitioning a database or file
- group into a READ ONLY state

vi. The following operations are not allowed during the CREATE DATABASE ENCRYPTION KEY, ALTER DATABASE ENCRYPTION KEY, DROP DATABASE ENCRYPTION KEY, or ALTER DATABASE, SET ENCRYPTION statements. (MSDN Library; Coles and Landrum, 2009)

- Dropping a file from a file group in the database
- Dropping the database
- Talking the database offline
- Detaching a database

- Transitioning a database or file group into a READ ONLY state
- Using an ALTER DATABASE command
- Starting a database or database file backup
- Starting a database or database file restore.
- Creating a snapshot

vii. The following operations or conditions will prevent the CREATE DATABASE ENCRYPTION KEY, ALTER DATABASE ENCRYPTION KEY, DROP DATABASE ENCRYPTION KEY, or ALTER DATABASE, SET ENCRYPTION statements. (MSDN Library; Coles and Landrum,2009)

- The database is read-only or has any read-only file groups.
- An ALTER DATABASE command is executing
- Any data backup is running.
- The database is in an offline or restores condition.
- A snapshot is in progress.
- Database maintenance tasks.

2. How Transparent Data Encryption Works

When enabling TDE, you should immediately back up the certificate and the private key associated with the certificate. If the certificate ever becomes unavailable or if you must restore or attach the database on another server, you must have backups of both the certificate and the private key or you will not be able to open the database. (MSDN Library)The encrypting certificate or asymmetric should be retained even if TDE is no longer enabled on the database. Even though the database is not encrypted, the database encryption key may be retained in the database and may need to be accessed for some operations. A certificate that has exceeded its expiration date can still be used to encrypt and decrypt data with TDE. Encryption of the database file is performed at the page level. (Coles and Landrum, 2009)The pages in an encrypted database are encrypted before they are written to disk and decrypted when read into memory. TDE does not increase the size of the encrypted database.

3. CREATE DATABASE USING MICROSOFT SQL SERVER 2008 (Deshmukh and Qureshi, 2011)

Creating a database that specifies the data and transaction log files, the following code 1 to create database name "Deals"

A. Code 1

```
USE master;
GO
CREATE DATABASE Deals
ON
(NAME = Deals_dat, FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL20-60,.MSSQLSERVER\MSSQL\DATA\dealdat.mdf', SIZE = 20, MAXSIZE = 60, FILEGROWTH = 6
)
```

```
LOG ON
( NAME = Deals_log,
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL20_60.MSSQLSERVER\
MSSQL\DATA\deallog.ldf', SIZE = 10MB, MAXSIZE = 30MB, FILEGROWTH = 10MB ) ;
GO
```

Description of above Code 1

CREAT DATABASE command is used to create database in SQL Server 2008, Deals is a name of Database. In this example we have created one data file name "Deals_dat' and one log file named "Deals_log" with specified size.

4. CREATION OF ENCRYPTION AND DECRYPTION OF DATABASE (Deshmukh and Qureshi, 2011)

The following code 2 illustrates encrypting and decrypting the Deals database using a certificate installed on the server named MyDealsCert.

A. Code 2

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<writeanypasswordhere>';
go
CREATE CERTIFICATE MyDealsCert WITH SUBJECT = 'It is my Certificate';
go
USE Deals;
GO
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE MyDealsCert;
GO
ALTER DATABASE Deals
SET ENCRYPTION ON;
GO
```

Description of Code 2

The Command CREATE MASTER KEY ENCRYPTION BY PASSWORD is used to create Master key encryption with password here user can assign any password. CREATE CERTIFICATE is used to create certificate as MyDealsCert, WITH SUBJECT is used to assign any subject as "It is my Certificate" for the database created in code 1 "Deals". The Command CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_128 it is an encryption key. By using ENCRYPTION BY SERVER CERTIFICATE command assigning the certificate "MyDealsCert" to the Server. By using the command SET ENCRYPTION keeping it
ON.

CONCLUSIONS

Transparent Data Encryption plays an especially important role in safeguarding data in transit. Microsoft SQL Server Enterprise Edition (SQL Server 2008)Transparent Data Encryption protects sensitive data on disk drives and backup media from unauthorized access, helping reduce the impact of lost or stolen media. The Transparent Data Encryption are developed under the present work has been successfully, created, implemented and thoroughly tested. We used this Transparent Data Encryption on few computer we found it to be very effective. The Transparent Data Encryption based on the study of various security measures were planned, designed and developed using techniques and tools described earlier. Transparent Data Encryption provides a highly configurable environment for application development. User can use Transparent Data Encryption to create both single-machine and networked environments in which developers can safely try out. Such a setup do not requires additional infrastructure and easily caters to the needs of beginners as well as advanced learners on one premise

REFERENCES

Husain Rashid., (2012), "Types of Attacks and Defense Metrics of Routing Mechanism for Mobile Network," *International Journal of Innovation in Computer Science and Technology*, pp. No. 23-33.

Deshmukh Anwar Pasha., and Oureshi Riyazuddin., (2011), "Transparent Data Encryption- solution for security of Database Contents," *International Journal of Advanced Computer Science and Applications*, PP. No. 25-28.

Deshmukh Anwar Pasha., and Oureshi Riyazuddin., (2011), "Transparent Data Encryption- rver 2008 and Oracle ," *International Journal of Advanced Computer Science and* MSDN Library, "Transparent Data Encryption SQL Server 2012", [Online] Available: http://msdn.microsoft.com/en-us/library/bb934049.aspx.

Rob Walters, Christian Kirisch, (2010). "*Database Encryption and key management for Microsoft SQL Server 2008*", by Create Space.

Micheal Otey, (2008). "*Microsoft SQL Server 2008 New Features*", McGraw Hill Osborn Media, 2 Edition

Denny Cherry,(2011). "*Securing SQL Server: Protecting your Database from Attackers*", Syngress 1 Edition.

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, (2001). " *Computer Organization*," McGraw hill Edition.

Andrew S. Tennanbaum, (2010). "*Computer Network*," 5th Edition, Prentice Hall.

Micheal Coles, Rodney Landrum, (2009). "*Expert SQL Server 2008 Encryption*", Apress Edition -1.

Oracle Advanced Security, (2007). "*An Oracle White Paper*", [Online] Available: http://www.oracle.com/technetwork/database/security/advancedsecurity.

MSDN Library, "*Encryption Connection to SQL Server*", http://msdn.microsoft.com/en-us/library/ms189067.aspx.