Chapter 64
Securing Client-Server Application Design for Information System Inventory

Ibnu Gunawan, Djoni Haryadi Setiabudi, Agustinus Noertjahyana and Yongky Hermawan

Abstract This paper will discuss how to design a client server application for the building supply store located at 3 different locations along with the networking system design as well as how to secure it. Santoso Building Stores is a company engaged in the sale of construction materials. The current system for the inventory in the store is still done manually, making frequent errors in the data and the resulting delay in presenting information. Based on these problems, the company needed a system to record the inventory through purchase and sale transactions, so that the report can be presented quickly. In order to meet these needs, it developed an application to support the activities of the company so that all inventory record scan be performed quickly and report scan also be presented in real time. Applications are developed using Microsoft Visual Basic 2010 and SQL Server 2008 R2 includes SQL Encryption feature. The experiments results showed that the application was able to meet the needs of the company, inventory system that is accurate, fast and safe.

Keywords Client-server · Encryption · Inventory · SQL server · Visual basic

64.1 Introduction

The company carries out business activities by recording the sales process, purchasing, etc. manually. The company carries out recording by using paper. Activity recording the purchase and sale transaction has experienced an error. Staff often makes mistakes in the calculation process, causing the company losses. In the sales process, the staff simply records the transaction in a note. When it goes on sale in

I. Gunawan (✉) · D.H. Setiabudi · A. Noertjahyana · Y. Hermawan
Faculty of Industrial Technology, Department of Informatics Engineering,
Petra Christian University, Surabaya, Indonesia 60236
e-mail: ibnu@petra.ac.id

F. Pasila et al. (eds.), *Proceedings of Second International Conference on Electrical Systems, Technology and Information 2015 (ICESTI 2015)*,
Lecture Notes in Electrical Engineering 365, DOI 10.1007/978-981-287-988-2_64

small quantities and the customer does not ask for a receipt, the staff did not report these sales transactions.

In the process of purchase, owners often order goods that are still in stock. This happens because the inventory can't be calculated precisely. When the ordered items arrive, the owner did not record the details of goods coming and when the due payment of goods. This causes frequent delays in the payment process.

In the process of inventory of goods also take a long time, because it does not have data on the number of stock. The company is still conducting an inventory manually so it takes a long time to complete. Owner of the company had plans to open branches in different locations. Certainly each branch must have accurate information about inventory, either from a warehouse or center. Given these problems, it is suitable for the company to use Client-server platform as similar as a system built by Kumar [1].

The server will be placed at the head office and branches can access directly via the connected network. The process is the server receives a client request and will respond with the right result. In the client-server system design has not forgotten aspects of data security. While sending data from the server to the client, the data must first be encrypted so it can't be read by unauthorized parties [2]. Applications developed will help the staff to keep records of all sales and purchases, as well as perform automatic inventory adjustment, so that errors in the calculation of stock can be minimized. Reports can also be directly presented to owner and accurately.

64.2 Client Server System

Client/server system usually runs at least two different computer systems. One computer acts as a client and the other as the server. But the client and server can reside on a single computer system. A server usually serves multiple client computer. Although there may be also that only serve one client only. This server functions normally performed by a file server, except when maximum performance is required it is used a special server that the client is usually in the form of a desktop computer that is connected to the network.

If the user wants to retrieve or store information, part of the client application will issue a demand that would send (usually over a network) to the server. The server then run the query and sends the information back to the client. A database cannot replace the client/server system, although the system is a client/server often use the database server to perform the activity. Applications designed to use Microsoft Access, Microsoft FoxPro, Paradox, or other database program is not a client/server system (even if the database is in network server). All are examples of network database application because all processing is done by the client [3].

64.3 Encryption in SQL Server 2008

Encryption demands have increased over the past few years. For instance, there has been a demand for the ability to store encryption keys "off-the-box" physically separate from the database and the data it contains. Also there is a recognized requirement for legacy databases and applications to take advantage of encryption without changing the existing code base. To address these needs SQL Server 2008 adds the following features to its encryption arsenal:

- *Transparent Data Encryption (TDE)*: Allows you to encrypt an entire database, including log files and tempdb database, in such way that it it transparent to client application.
- *Extensible Key Management (EKM):* Allows you to store and manage your encryption keys on an external device known as a hardware security module (HSM).
- Cryptographic random number generation functionality.
- Additional cryptography— related catalog views and dynamic management views.
- SQL Language extensions to support the new encryption functionality.

SQL Server 2008 represents the most advanced SQL Server encryption capability to date, and you can leverage even more encryption functionality using other tools. For example, you can encrypt an entire hard drive with SQL Server database on it via Windows BitLocker technology. You can also use SSL to encrypt your SQL Server communication, protecting your data in transit [4]. We can see the architecture of SQL server 2008 encryption in Fig. 64.1

64.4 System Analysis

This section will discuss one business process to be analyzed as purchase returns. Purchase returns process happened when the goods are received in damaged condition or disability. If the goods are defective or damaged, it can be returns back. If the goods are damaged, then also the number of items in the original sales notes and it's duplicate copies will be reduced, according to the quantity of goods with damaged or defective condition. The store owner could only exchange goods with the same type and size. Document flowchart purchase returns process can be seen in Fig. 64.2.

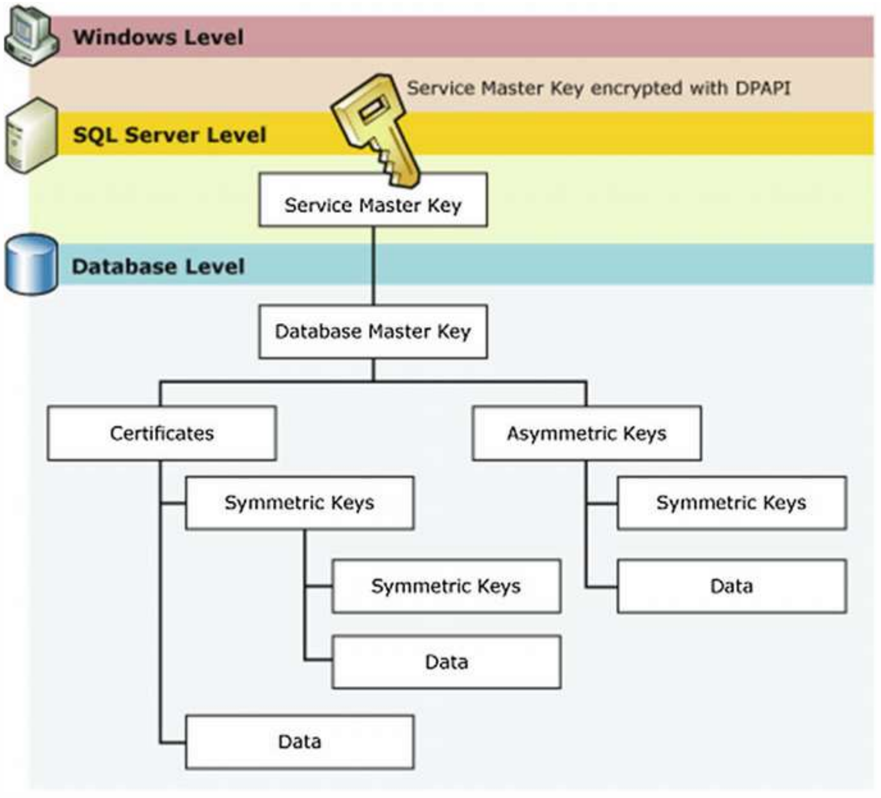


Fig. 64.1 SQL server 2008 encryption architecture

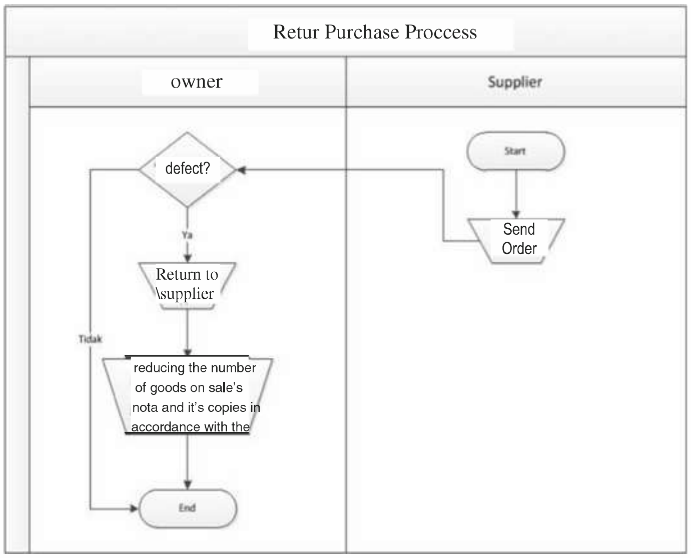


Fig. 64.2 Purchase return process

64.5 System Design

As we stated earlier in this paper, we will make a client server application for the building supply store located at 3 different locations along with the networking system design as well as how to secure it. Firstly, we designed a context diagram to show the whole process of the system. Context diagram on the buyer's side describes the process of purchasing data and payment, as well as the purchase invoice and payment. The supplier side shows purchasing data and payment process, order, and data charges for purchases and payments. The last side is owner who can look at reports such as purchasing reports, sales reports, report card stock and income statement. The context diagram is presented in Fig. 64.3.

Secondly, we attempt to encrypt the feature for SQL Server 2008 encryption by using the symmetric keys. For example, we create the sales.customer table using this syntax as in Fig. 64.4. Finally, we encrypt the data as in Fig. 64.5 and while it is on the client, we can see the syntax for decrypting the data. It can be seen in Fig. 64.6.

64.6 System Testing

In this paper, we present one of many screens since the program was built in Indonesian language. One of them is a program user interface of sale screen that we can see in Fig. 64.7.

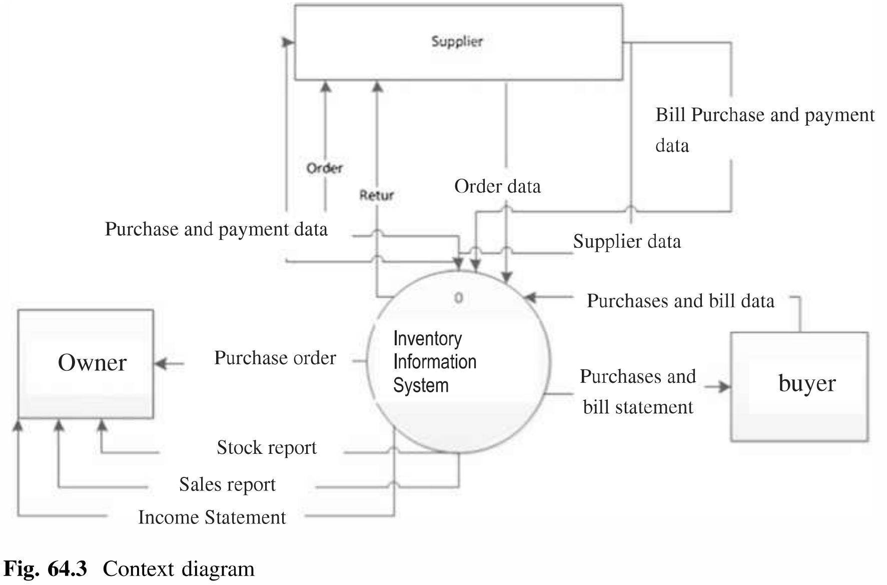
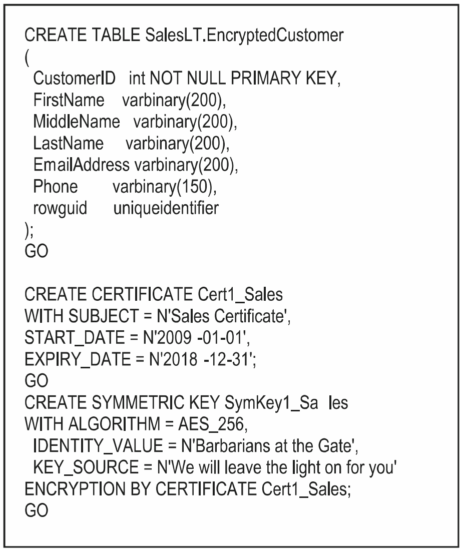


Fig. 64.3 Context diagram

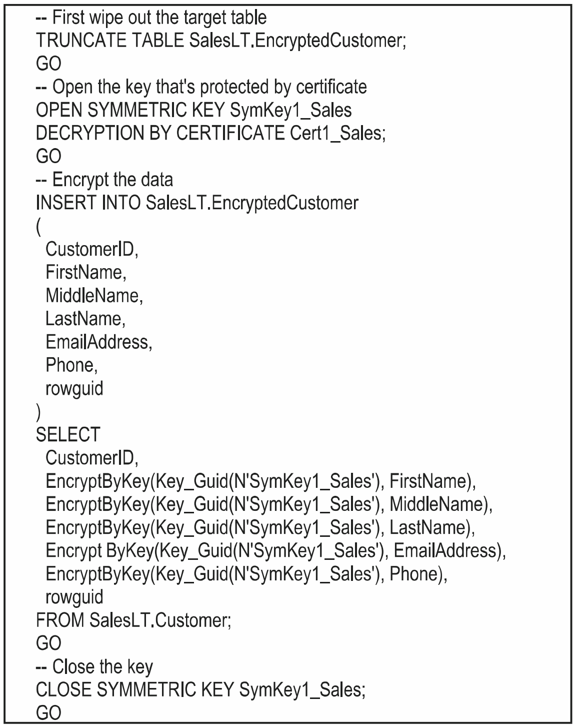
Fig. 64.4 Creating sales.customer table



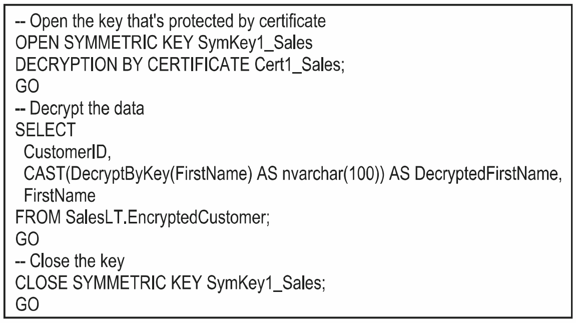
```
CREATE TABLE SalesLT.EncryptedCustomer
(
  CustomerID   int NOT NULL PRIMARY KEY,
  FirstName    varbinary(200),
  MiddleName   varbinary(200),
  LastName     varbinary(200),
  EmailAddress varbinary(200),
  Phone        varbinary(150),
  rowguid      uniqueidentifier
);
GO

CREATE CERTIFICATE Cert1_Sales
WITH SUBJECT = N'Sales Certificate',
START_DATE = N'2009 -01-01',
EXPIRY_DATE = N'2018 -12-31';
GO
CREATE SYMMETRIC KEY SymKey1_Sa  les
WITH ALGORITHM = AES_256,
  IDENTITY_VALUE = N'Barbarians at the Gate',
  KEY_SOURCE = N'We will leave the light on for you'
ENCRYPTION BY CERTIFICATE Cert1_Sales;
GO
```

Fig. 64.5 Encrypting the data

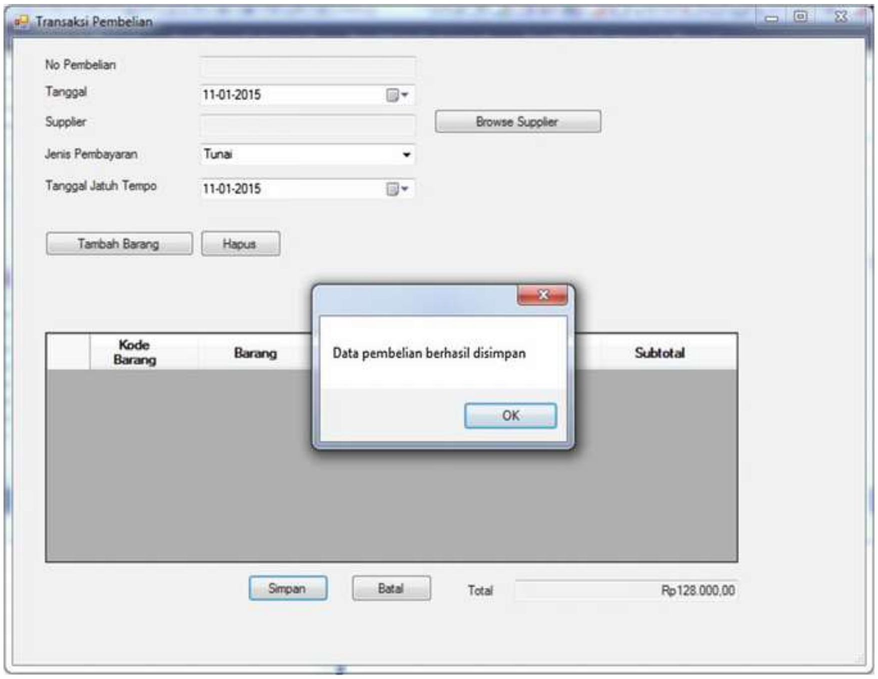


```
-- First wipe out the target table
TRUNCATE TABLE SalesLT.EncryptedCustomer;
GO
-- Open the key that's protected by certificate
OPEN SYMMETRIC KEY SymKey1_Sales
DECRYPTION BY CERTIFICATE Cert1_Sales;
GO
-- Encrypt the data
INSERT INTO SalesLT.EncryptedCustomer
(
  CustomerID,
  FirstName,
  MiddleName,
  LastName,
  EmailAddress,
  Phone,
  rowguid
)
SELECT
  CustomerID,
  EncryptByKey(Key_Guid(N'SymKey1_Sales'), FirstName),
  EncryptByKey(Key_Guid(N'SymKey1_Sales'), MiddleName),
  EncryptByKey(Key_Guid(N'SymKey1_Sales'), LastName),
  Encrypt ByKey(Key_Guid(N'SymKey1_Sales'), EmailAddress),
  EncryptByKey(Key_Guid(N'SymKey1_Sales'), Phone),
  rowguid
FROM SalesLT.Customer;
GO
-- Close the key
CLOSE SYMMETRIC KEY SymKey1_Sales;
GO
```



```
-- Open the key that's protected by certificate
OPEN SYMMETRIC KEY SymKey1_Sales
DECRYPTION BY CERTIFICATE Cert1_Sales;
GO
-- Decrypt the data
SELECT
  CustomerID,
  CAST(DecryptByKey(FirstName) AS nvarchar(100)) AS DecryptedFirstName,
  FirstName
FROM SalesLT.EncryptedCustomer;
GO
-- Close the key
CLOSE SYMMETRIC KEY SymKey1_Sales;
GO
```

Fig. 64.6 Decrypting the data



Transaksi Pembelian

No Pembelian

Tanggal 11-01-2015

Supplier

Jenis Pembayaran Tunai

Tanggal Jatuh Tempo 11-01-2015

Kode Barang	Barang	Subtotal

Data pembelian berhasil disimpan

Total Rp128.000,00

Fig. 64.7 Sales program screen

64.7 Conclusion

Based on the results and system test, we conclude as follows:

- Store 2 as the branch can connect to the database server is like being in one central store network.
- The application can generate reports so as to facilitate the user for getting the needed information, such as reports purchasing, sales, inventory and others.

References

1. Kumar, A.: Thin client web-based campus information systems for Fiji National University. Int. J. Softw. Eng. Appl. (IJSEA) **2**, 1 (2011)
2. Qureshi, D., Desmukh, D.: Transparent data encryption—solution for security of database contents. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **2**, 3 (2011)
3. Corporation, I.S.: Information Systems Essentials, 2nd edn. InterBase Software Corporation, California (1998)
4. Landrum, R., Coles, M.: Expert SQL Server 2008 Encryption. Apress, New York (2009)