



ISSN: 2350-0328

**International Journal of Advanced Research in Science,
Engineering and Technology**

Vol. 8, Issue 9 , September 2021

Warehousing Sales and Stock Management System with Audit Trail, SMS Alert and Two-Way Encryption

Abraham E. Evwiekpaefe, Oluwaremilekun M. Yusuf

Senior Lecturer, Department of Computer Science, Nigerian Defence Academy, Kaduna, Kaduna State, Nigeria

P. G. Student, Department of Computer Science, Nigerian Defence Academy, Kaduna, Kaduna State, Nigeria

ABSTRACT:As the business world keeps evolving as compared to the ratio of computer size which keeps reducing in capacity, it has become a necessity in our daily activities and in business as well. Also the knowledge of computer usage have become cheap and readily accessible. Therefore hiding of information/data in the computer database is no longer an efficient mode of keeping data save and secured because a “novice” in computer usage now can even do the basic. Hence the need for a further security measure, encryption of database. This research focused on developing a better system for stock, sales and warehousing management so as to show the possibility of encrypting vital information in the database and implementing a secured means of sending the decryption key. .NET framework was extensively used for the development of the front end while Microsoft SQL was used for the database development.

I. INTRODUCTION

Most consumers of warehousing sales and stock management system, bespoke and off the shelf software are ignorant of the technicality that is employed to solve their daily business needs. At the same time, they are fully aware of the functionalities/features that they want the software to exhibit and the minimum problem the software has to solve to pass as an accepted software for their daily use. With this in mind, it 100% squares on the shoulder of the developer to be able to deploy a software that meet the client’s features and requirement and at same time measure up to performance, security and data integrity standard.

Sales and stock management system is a fully functional and problem solving standalone module, while audit trail is a multi-discipline research area (cutting across various fields like accounting, forensic auditing and network security). Short Message Service (SMS) alert system is another standalone communicating module that strives on high level encapsulation and function inheritance while encryption is a never ending research area for computer scientist, data engineer, network experts and database administrators. Generally these features are hardly built into a single system, most end users always have to implement two to three software to get these features or consume a third party web services [1]. This will eventually add to the running cost and the hardware consumption there by making the whole system inefficient but may be effective.

Also one major source of concern in the field of encryption has always been the medium in which the key (to be used to decrypt) will be sent [2]. This has been an area of interest to most researchers because if there was a “safe medium” to use to send the key, it could as well have been better to send the whole information through the “safe medium”. As of present very few bespoke stock management software have all of these features in one package, the user usually have to get a third party software to achieve the audit trail or rely on the database administrator to keep the dataset request from each user and in some other cases the client will rely on the network administrator to keep track of each IP address request to the server. As for the encryption side this software sends the encryption key through SMS to the authenticated user’s phone number. Therefore, the key will be sent via SMS to the appropriate user/login profile and then the SMS consumes a web service which later consumes another web service (similar to inheritance in OOP) in the



ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 9 , September 2021

hierarchy that are randomly selected from a pool of web services. The problem exists such that a client or an organization does not get a comprehensive solution from one software. This paper gives that solution and add an additional encryption key passing through SMS.

II. REVIEW OF RELATED WORKS

[3]detailed audit trail in a behavioral and organizational context. The authors focused on determining the behavior pattern of users/staff by observing the audit trail and mining out large amount of data from the trail. The study concluded by introducing what could be achieved with audit trail for further research: Executive Information System (EIS), Auditing software (Computer Assisted Auditing Tools and Techniques – CATTs) and Control self-assessment tools, Decision aid tools

[4] proposed a methodology for continuous fraud detection that exploited security audit logs, changes in master records and accounting audit trails in enterprise systems. The study demonstrated how, an enterprise system, can be used for audit trail analysis in detecting financial frauds. The paper further used a case study of a suspected fraud to illustrate how to implement the methodology.

[5] examined a new cryptosystem that is suitable for database encryption. The system had the important property of having sub-keys that allowed the encryption and decryption of fields within a record. The system was based on the Chinese Remainder Theorem.

[6] Grubbs et al. (2016) developed a systematic approach for analyzing client-server applications that aim to hide sensitive user data from untrusted servers. The paper applied it to Mylar, a framework that uses multi-key searchable encryption (MKSE) to build Web applications on top of encrypted data. The study concluded that the problem of securing client-server applications against actively malicious servers is challenging and still unsolved.

[7] showed how to encrypt a relational database in such a way that it can efficiently support a large class of SQL queries. Their construction was based solely on structured encryption (STE) and does not make use of any property-preserving encryption (PPE). As such, their construction was only efficient under some conditions on the database. They also showed how to extend their solution to be dynamic while maintaining the scheme's optimal query complexity.

Furthermore, [8]Mattsson(2016) documented that encryption can provide strong security for data at rest, but developing a database encryption strategy that must take many factors into consideration. The paper further noted that organizations must balance between the requirement for security and the desire for excellent performance. The paper basically reviewed the performance aspects of database encryption, but never made recommendation for a particular one.

[9]proposed a model-driven application-level encryption solution to protect the privacy and confidentiality of health data in response to the growing public concern about the privacy of health data. Their approach combined the flexibility, security and independence from database vendors of application-level encryption and the transparency of database-level encryption. However, their approach was not cost effective and it has an additional overhead to the whole application.

Research Gap

Most of the related works did separate research on database encryption and audit trail each one standing alone (that is it not been done collectively as a single project). In this paper, provisions are made to combine both the database encryption and audit trail into a single system, and an additional step is been taken by introducing a SMS alert system that sends the encryption key to the valid authenticated user before certain high value information could be decrypted.

III. METHODOLOGY

Methodology adopted

The model used in developing the application is the waterfall model. The application was broken down into a set of linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. This approach was chosen because it gives an engineering design pattern to software development.

A class was declared that strictly handled encryption and decryption of the data. It's a public class that inherit from the Crypto class and implements the ICryptoTransform interface provided by the .NET framework (an interface is a description of all functions that an object must have in order to be of a particular class), and contain publicly declared variable that will contain the data to be encrypted. The data to be encrypted is read into a stream variable and the specific encryption algorithm is applied on the stream variable Also a method within the class generate the decryption key which is eight character long. It can contain alphanumeric character and at least one special character. Then a SMS class is created that handles all the message activity which include sending of the key, updating customers on the situation of their accounts. Lastly there is an audit class, which contains method to insert and read. The insert method is called upon on every activity performed by the user and the adequate parameters of page, date, action performed and user details are sent on each call of the insert method. The read method gives an output of a multidimensional array of the audit trail that is been queried from the database. In addition to all the classes mentioned above, other classes were defined to handle other basic functions like invoicing, stocking, supply and direct selling to mention just a few.

Programming Language used

This paper implemented a web based application and the following programming languages were used to accomplish the task: C#, Java script and ASPX.

C# is a programming language developed by Microsoft, and it contains the .NET framework. It is a general-purpose, multi-paradigm programming language encompassing static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. This language was chosen because it has a rich (.NET) library for good interface design and encryption and it was used for the server end coding. Java script a client side scripting language. It is used to perform functions on the client side there by, reducing the number of round trip to the server. It was used for this paper because it easy to implement and has a lot of efficient algorithm for simple task in its library. Active Server Page Extended (ASPX) was used to design web page presentation in .NET environment. It can run basically on any browser. The database administrator used is Microsoft SQL server 17 (MSSQL). Input and output design For the Encryption part the input can be any combination of string, numbers or alphanumeric. For this research the input for the encryption were sensitive data like customers name, balance, payments, etc. Figure 1 depicts the flowchart is as follows:

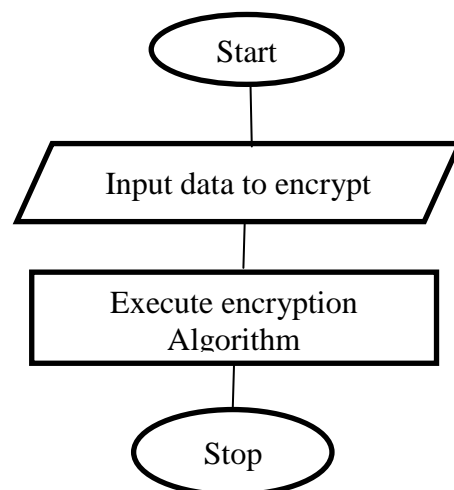


Figure 1: Flow chart of the encryption method

The class diagram of the encryption class of the encryption algorithm is shown in table1.

Table 1: Class diagram of the encryption class

| Encryption class |
|------------------|
| Text: string |
| Encryption() |
| EncryptString() |
| DecryptString() |

IV. RESULTS AND DISCUSSION

Results of randomly generated decryption key

This used a combination of a random number generating code and selected few alphabets special character. The figure 2 shows what the key generator looks like.

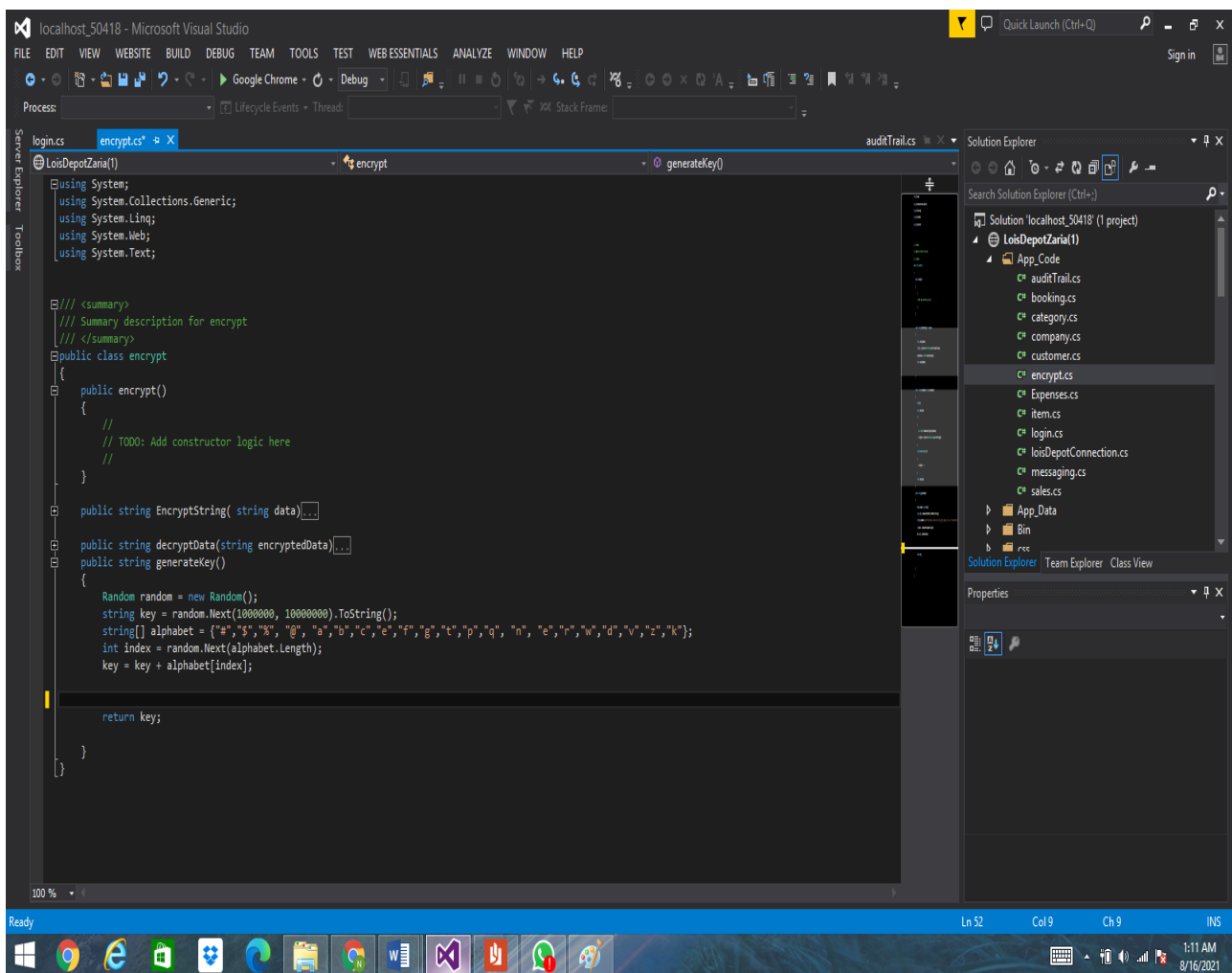
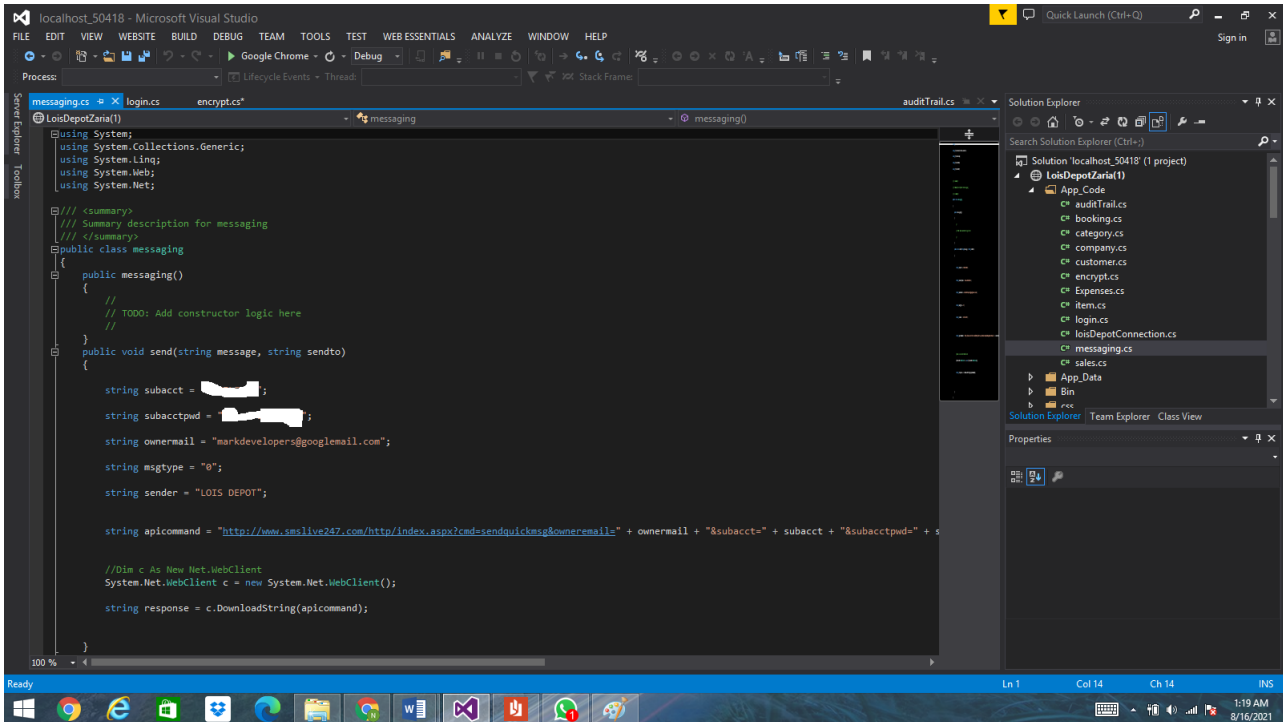


Figure 2: An interface showing a screenshot of randomly generated decryption key.

Result of the class handling sending of SMS alert

The result of the SMS class that handles all the message activities including the sending of the key, updating customers on the situation of their accounts, etc is shown in figure 3.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Net;

/// <summary>
/// Summary description for messaging
/// </summary>
public class messaging
{
    public messaging()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public void send(string message, string sendto)
    {
        string subacct = " ";
        string subacctpwd = " ";
        string ownermail = "markdevelopers@googlemail.com";
        string msgtype = "0";
        string sender = "LOIS DEPOT";

        string apicommand = "http://www.smslive247.com/http/index.aspx?cmd=sendquickmsg&owneremail=" + ownermail + "&subacct=" + subacct + "&subacctpwd=" + s

        //Dim c As New Net.WebClient
        System.Net.WebClient c = new System.Net.WebClient();
        string response = c.DownloadString(apicommand);
    }
}
```

Figure 3: An interface of the class handling sending of SMS alert

Result of the Audit Trail class

The result of the audit class, which contain the insert and read methods is shown in figure 4. Figure 4 depicts the activities performed by the user and the adequate parameters of page, date, action performed and user details. The read method gives an output of a multidimensional array of the audit trail that is been queried from the database.

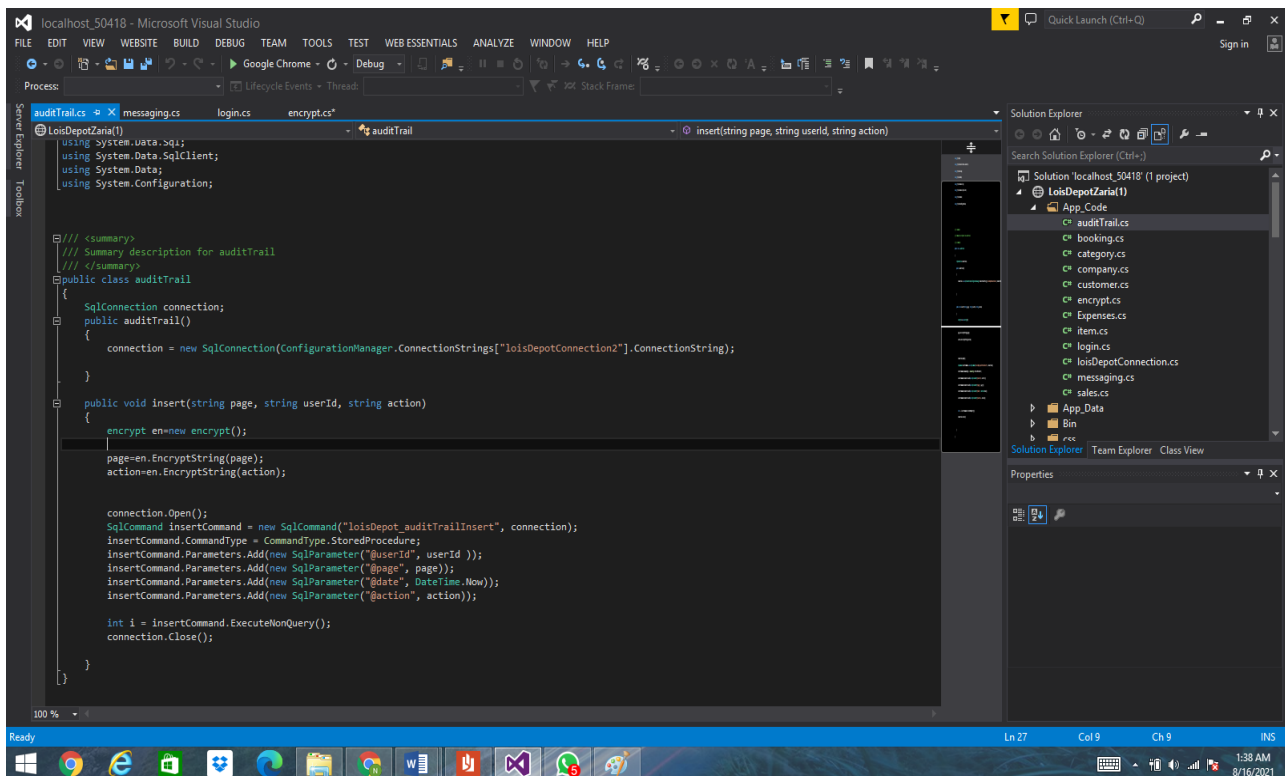


Figure 4: An interface showing the Audit Trail class

Result of the output of an encryption


The output text of an encryption says the value of 65,000, which is the total purchase of a customer. The encrypted value that will be sent to the database after running the encryption algorithm will be:

CFqf/8RVpeWZrH3lhtLbH0vXNvX2H9cu1N9sa19Yvd0=

A similar sequence will be done for the decryption process.

Results of the encryption and decryption process

In figure 5, the phone number column displays the encrypted texts. And once the page is loaded a key is sent to the phone number of the user viewing the page. Figure 6 shows the content of the text message. Once the user enters the key sent to his phone number and click decrypt the phone number column will be decrypted as shown in figure 6 and 7.



The screenshot shows a web interface for user management. At the top, there is a navigation bar with a 'New User' button and a search input field. Below this is a 'Decrypt' button. The main content is a table with columns for Action, ID, User Name, User Password, Valid, Role, Date Created, Last Date Modified, and Phone Number. Each row represents a user, and the phone number column contains an encrypted string. Each row also has 'Edit' and 'Status' buttons.





| Action | ID | User Name | User Password | Valid | Role | Date Created | Last Date Modified | Phone Number |
|---|--------------------------------------|-----------|---------------|-------|-----------|---------------------|---------------------|------------------|
|  Edit  Status | 9c89ea54-2a19-418c-9957-2b2a8bde3234 | mary | mary | True | Sales Rep | 27/08/2018 09:44:58 | 27/08/2018 09:44:58 | MDcwMzkoNzg1NzE= |
|  Edit  Status | 17b87f30-1a1f-4bfe-96f9-ed2b2df4f315 | nufi | nufi | True | Manager | 07/08/2021 12:25:41 | 07/08/2021 12:25:41 | MDcwMzkoNzg1NzE= |
|  Edit  Status | e7344d4a-5553-4c6f-be95-f19dc14f0c6e | omeza | omeza | False | Sales Rep | 08/10/2018 00:49:30 | 08/10/2018 00:49:30 | MDcwMzkoNzg1NzE= |
|  Edit  Status | dbafdfc6-d8f5-432d-b895-a91fd88614c2 | remi2 | remi2 | True | Manager | 06/08/2021 14:45:53 | 06/08/2021 14:45:53 | MDcwMzkoNzg1NzE= |
|  Edit  Status | 43d596cd-6894-47f2-82dc-2db9ba5e2ba5 | yemi | yemi | False | Sales Rep | 10/04/2019 05:39:54 | 10/04/2019 05:39:54 | MDcwMzkoNzg1NzE= |

Figure 5: list of user with their phone number encrypted

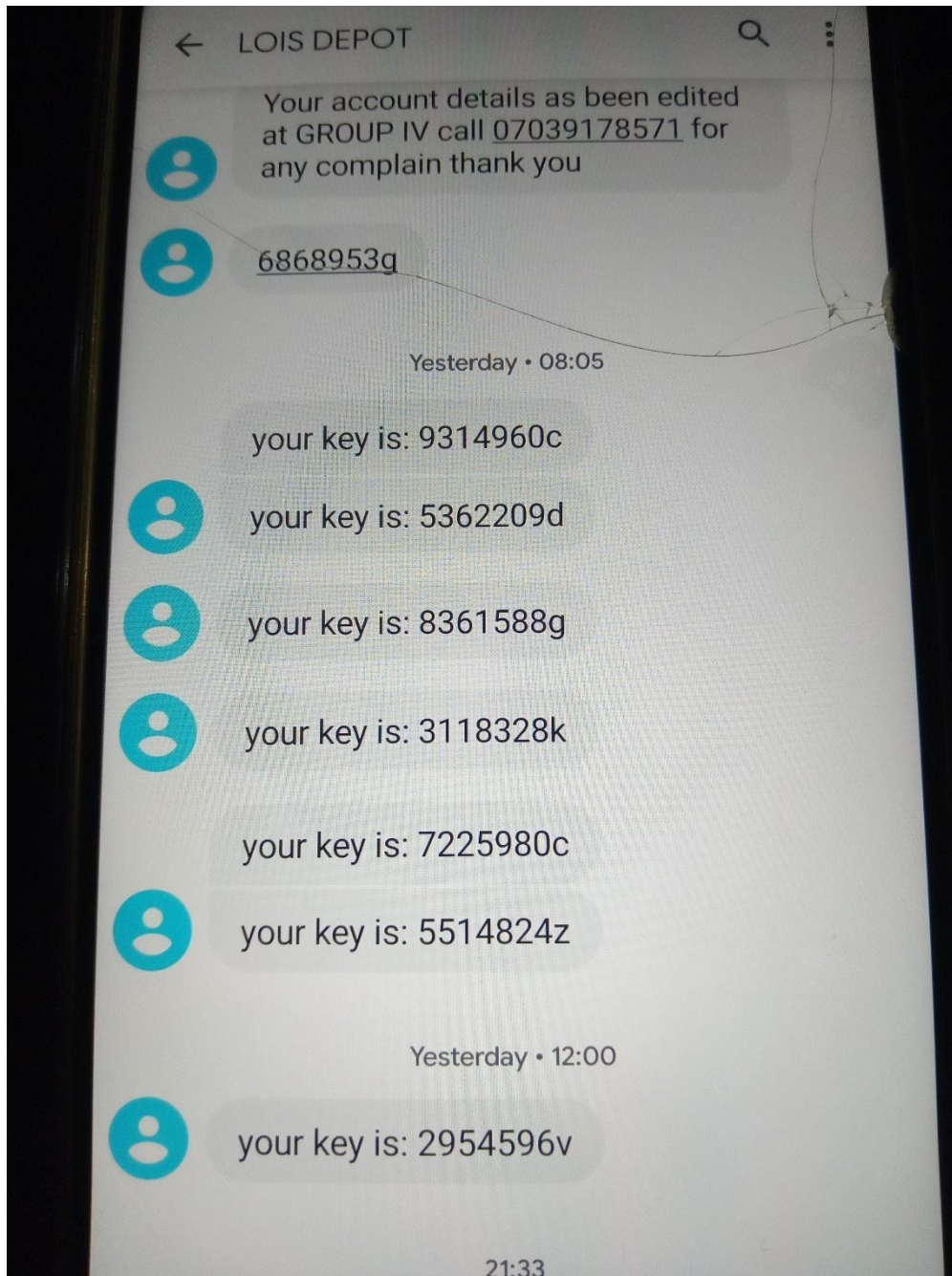
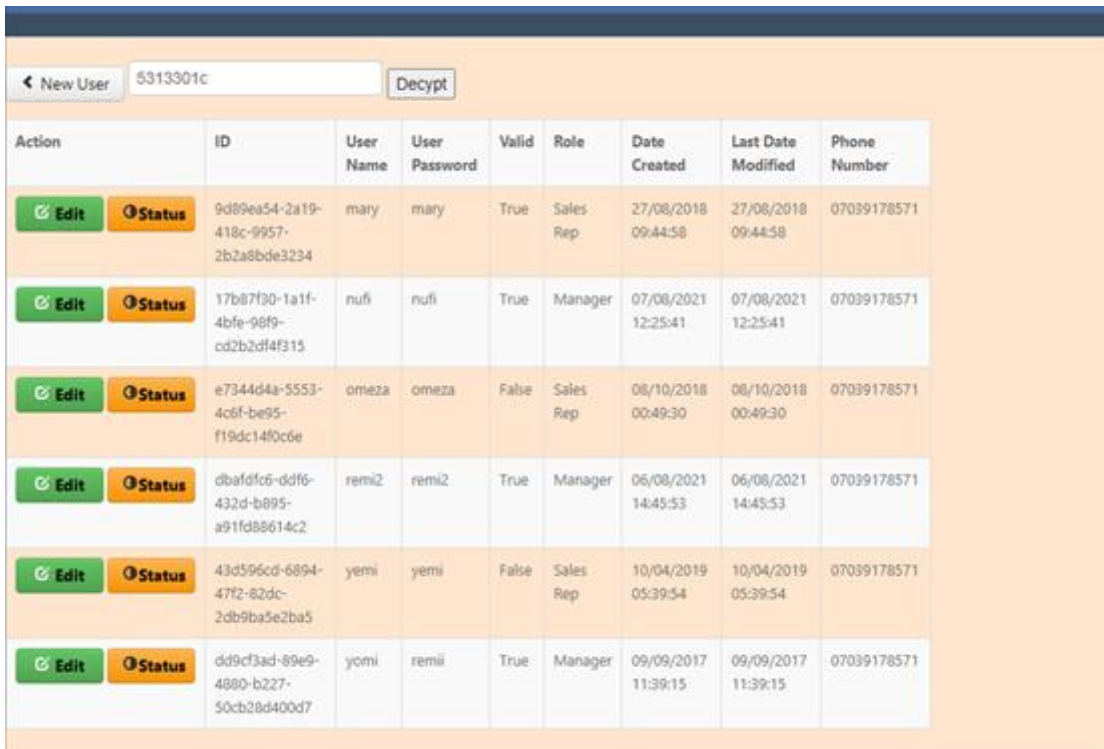


Figure 6: the decryption key that will be sent to the mobile phone numbers.















| Action | ID | User Name | User Password | Valid | Role | Date Created | Last Date Modified | Phone Number |
|---|--------------------------------------|-----------|---------------|-------|-----------|---------------------|---------------------|--------------|
|   | 9d89ea54-2a19-418c-9957-2b2a8bde3234 | mary | mary | True | Sales Rep | 27/08/2018 09:44:58 | 27/08/2018 09:44:58 | 07039178571 |
|   | 17b87f30-1a1f-4bfe-98f9-cd2b2df4f315 | nufi | nufi | True | Manager | 07/08/2021 12:25:41 | 07/08/2021 12:25:41 | 07039178571 |
|   | e7344d4a-5553-4c6f-be95-f19dc14f0c6e | omeza | omeza | False | Sales Rep | 08/10/2018 00:49:30 | 08/10/2018 00:49:30 | 07039178571 |
|   | dbafdfc6-ddf6-432d-b895-a91fd88614c2 | remi2 | remi2 | True | Manager | 06/08/2021 14:45:53 | 06/08/2021 14:45:53 | 07039178571 |
|   | 43d596cd-6894-47f2-82dc-2db9ba5e2ba5 | yemi | yemi | False | Sales Rep | 10/04/2019 05:39:54 | 10/04/2019 05:39:54 | 07039178571 |
|   | dd9cf3ad-89e9-4880-b227-50cb28d400d7 | yomi | remii | True | Manager | 09/09/2017 11:39:15 | 09/09/2017 11:39:15 | 07039178571 |

Figure 7: List of registered users with after decryption has been done.

Result of the saved encrypted data

Figure 8 shows the encrypted data was saved in the database demonstrating that the actual data that was transported through the network to the database are the encrypted data that cannot be decrypted without the actual key.

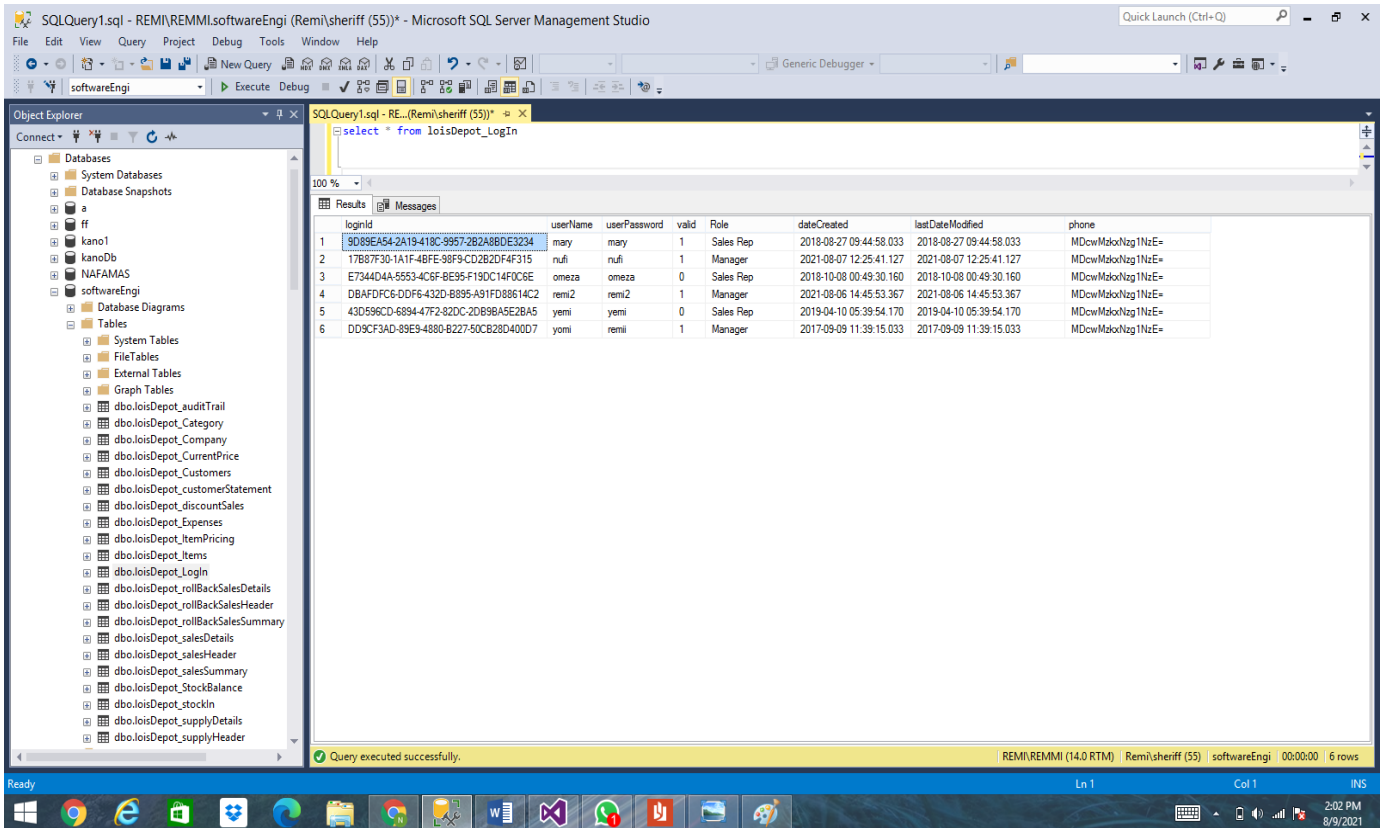


Figure 8: the database end showing that the data (phone number) is actually saved in an encrypted format.

Result of the saved audit trail

Figure 9 shows a direct view of the audit trail as saved in the database and just like previous figures the data becomes decrypted once the key is entered.



Figure 9: A sample of the saved audit trail



ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 9 , September 2021

V. DISCUSSION

From the results obtained, it is clear how the encryption of data works and how they are saved to the database. Most importantly, it showed how the encryption key(s) can be sent to the right users and the save medium to be used. This thereby improves data security in the sense that even if the system is under attack and the database hijacked, what the hijacker will see are encrypted data in the most important columns and these data can only be decrypted by keys that are generated at run time. The integrity and consistency of the data are also improved, since the data can only be properly viewed through only one channel. The channel through which the key is sent is a series of web services that can be best imagined as a hierarchy where each service is consuming the next service above it in the hierarchy till it consumes the final web service that actually sends the key. This hierarchy is arranged randomly and the arrangement keeps changing at unspecified time intervals.

VI. CONCLUSION

This paper demonstrated database encryption and decryption at application layer. It further introduced the obvious possibility of sending decryption key through SMS which has not been a popular concept before now. It displayed a simple but random way of generating decryption key. Finally, it introduced a form of encrypted audit trail.

REFERENCES

- [1] H. Marium, "Audit Trail: A modern way for staff management". IEEE 5th International Conference for Convergence in Technology (I2CT), 2019, pp. 1-4.
- [2] D. Fraser "What are encryption keys and how do they work?", 2018. Available online @ [www.https://medium.com/codeclan/what-are-encryption-keys-and-how-do-they-work-cc48c3053bd6](https://medium.com/codeclan/what-are-encryption-keys-and-how-do-they-work-cc48c3053bd6). Accessed 24th June, 2021.
- [3] S. Fragos, L. Stergioulas, and R. Gandecha, "Audit-trail-based modelling of the decision making process in Management and Accounting using sensitivity analysis", Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005, pp. 1-8.
- [4] P. J. Best , P. Rikhardsson and M. Toleman(2009). Continuous Fraud Detection in Enterprise Systems through Audit Trail Analysis Trail Analysis, Journal of Digital Forensics, Security and Law, Vol. 4, No. 1, 2009, pp.39-60
- [5] G. I. Davida, D. L. Wells and J. B. Kam, "A database encryption system with subkeys", ACM Transactions on Database, Vol. 6, No. 2, 1981, pp. 312-328
- [6] P. Grubbs, R. McPhersonUT, M. Naveed, T. Ristenpart and V. Shmatikov, "Breaking Web Applications Built On Top of Encrypted Data", 2016, CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1353-1364.
- [7] S. Kamara and T. Moataz, "SQL on Structurally-Encrypted Databases", 2018, Available online @ <https://www.iacr.org/archive/asiacrypt2018/11272302/11272302.pdf>. Accessed 21st May, 2021.
- [8] U. T. Mattsson, "Database Encryption - How to Balance Security with Performance", 2016. Available online @ <https://ssrn.com/abstract=670561> or <http://dx.doi.org/10.2139/ssrn.670561>. Accessed 20th May, 2021
- [9] D. Yun and K. Klein, "Model-Driven Application-Level Encryption for the Privacy of E-health Data", International Conference on Availability, Reliability and Security, 2010, pp. 341-346.