A Review on Securing Database using Steganography

[1]Sandra Sunny, [2]Alby Sunny, [3]Amitha Joseph

[1]Student, [2]Student, [3]AssistProfessor
[1]Department of Computer Sciences,
[1]Santhigiri College of Computer sciences, Vazhithala, India

Abstract: Steganography is the practice of hiding a secret message within another, seemingly innocuous message. The purpose of this technique is to conceal the existence of the hidden message, making it difficult for an outsider to detect. This method has been used throughout history for various purposes, including sending confidential information during wartime, delivering sensitive messages between spies, and communicating in secret during politically turbulent times. In the digital age, steganography has taken on new forms, such as hiding data within image, audio, and video files. Database steganography refers to the practice of hiding secret data within a database. It involves embedding confidential information within the structure or content of a database, making it difficult for unauthorized users to detect the hidden information. The paper presents a review on mainly two steganography technique for securing database. One uses steganography technique to hide a database record inside another database records and the other technique allows a user to create basic tables and records which are hidden from others inside an image.

Index Terms: Data Protection, Database, StegoDb, Steganography, cryptography, LSB, Json

I. INTRODUCTION

Confidentiality of data is a crucial issue in today's diverse environments involving computers, tablets and smart phones. In mobile environment for example there will be need to store financial data, personal information and passwords which should not reach hackers or other miscreants. Steganography differs from cryptography, which is the practice of encoding a message to make it unreadable to anyone except those who have the decryption key. While cryptography provides security through obscurity, steganography provides security through concealment, hiding the very existence of the message. This method has been used throughout history for various purposes, including sending confidential information during wartime, delivering sensitive messages between spies, and communicating in secret during politically turbulent times. In the digital age, steganography has taken on new forms, such as hiding data within image, audio, and video files. The use of advanced algorithms and encryption methods has made it possible to embed large amounts of data in seemingly innocuous digital files. With the advent of digital technology, it has become easier to embed hidden data within digital files, making it a popular tool for both malicious and benign purposes.

Most of the current Steganography techniques tend to affect the quality of the image as well if there is more data inserted into the image. In this paper we can see that this issue is avoided by using the best possible steganography algorithms at the same time finding the maximum amount of data that can be inserted into an image before starting the insert process. The problem with traditional steganography technique is that the application developer always faces the overhead of creating specialized procedures and programming logic for handling the encryption or data storing logic. Hence to avoid this and to make the life of an application developer much easier we are proposing a new steganography database architecture which is very easy to setup and provides the application developer easy to use methods which can be called easily while developing applications. The aim of this paper is to reviewing the two methods, a database inside another database and the other securing it under an image to secure a database.

II. CONCEPTS AND TECHNOLOGIES

A.Steganography

Steganography is the practice of concealing information within seemingly harmless cover media. The word "Steganography" comes from the Greek words "steganos" meaning "covered or hidden," and "graphia" meaning "writing." The goal of steganography is to hide the existence of the message itself. There are several types of steganography, some of them are Image Steganography, Audio Steganography, Video Steganography, Text Steganography and Network Steganography. There are various algorithms used in steganography, including LSB (Least Significant Bit) Algorithm, Discrete Cosine Transform (DCT) Algorithm, Spread Spectrum Algorithm etc. Since we are using Digital image steganography, we make use of LSB based steganography.

The Least Significant Bit (LSB) based steganography is a popular method of data hiding within digital media, particularly images. In this method, the least significant bits of the cover image are replaced with bits from the secret message. The LSB technique works by changing the value of the least significant bit of the cover image's pixels, which has a minimal impact on the visual quality of the image. The change is typically so small that it is not noticeable to the human eye. The LSB method can be used to hide any type of binary data, including text, images, and audio files. The advantage of the LSB method is that it is simple and straightforward to implement, and it does not require any special compression or encryption algorithms.

B. JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format that is used for transmitting data between a server and a client. It's a human-readable and easy to understand format, allowing values to be written as key-value pairs. JSON can be used in database steganography, by encoding secret messages within JSON data structures. This can be done by modifying the values of the properties in a JSON object in a way that doesn't alter its original meaning or functionality, but instead embeds additional information. These are the properties make JSON an ideal as a data-interchange language. Mainly JSON is built on two structures:

- A collection of name or value pairs. In other languages, it is realized as record, object, structure, dictionary, keyed list, hash tableor associative array.

- Values as ordered list. In many languages, this is realized as vector, list, array, or sequence. One of the biggest advantages of JSON format is that it maps to most of the existing data structures in programming languages and it has a layout that is quite simple enough to keep coding and db design simple. It is also simple and flexible enough to express most data in a fairly natural way.

C. Database Environment

The database is a shared resource therefore each user may need a different view of the data contained in the database. The database schema is the general description of the database. [1] Three types of schema in the database, external, a conceptual, and an internal level are depicted in Figure 1

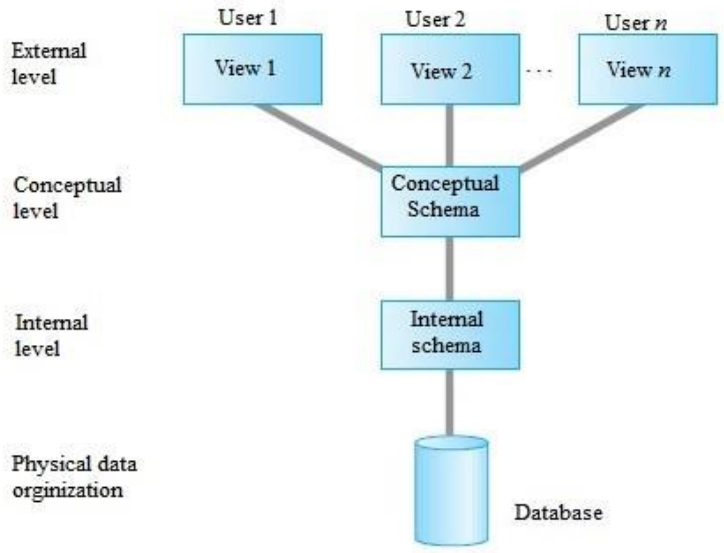

Fig 1: Three Schema architecture.

III. PROPOSED DATABASE STEGANOGRAPHY

The proposed DB system uses mainly two steganography technique for securing database. One uses steganography technique to hide a database record inside another database records and the other technique allows a user to create basic tables and records which are hidden from others inside an image.

1. Hiding Complete Database In Another Database.

The purpose of this concept is to conceal structured data in the unused space of a database record's memo field. Two methods of data hiding are proposed:

A.A hidden database table with the same structure as the cover table but with different data.

When the hidden-database table has the same structure as the cover-table, each stego-record consists of a cover-record and a hidden-record. Both records have the same field structure, except for the memo field in the cover-record, which holds the hiddenrecord. The hidden-record is embedded in the memo field of the cover-record and has the same structure. The cover-record, which is visible to unauthorized users, holds different values from the hidden-record. The role of the hidden-DBMS is limited to accessing and retrieving/storing the hidden data in a low-level manner, while all other data processing is handled by the cover-DBMS. B. A hidden database table with a different structure than the cover table and also different data.

When the hidden-database table has a different structure from the cover-table, the first hidden record must hold the file header of the hidden-table, which contains metadata about the record field properties.

Unauthorized users can access only the data in the cover-database table through the cover-DBMS, but cannot access the hidden data. Authorized users, on the other hand, can access the hidden data by using the hidden database management system (hiddenDBMS). [7] The combination of these two databases creates the compound database management system, known as the stegoDBMS, as shown in Figure 2.

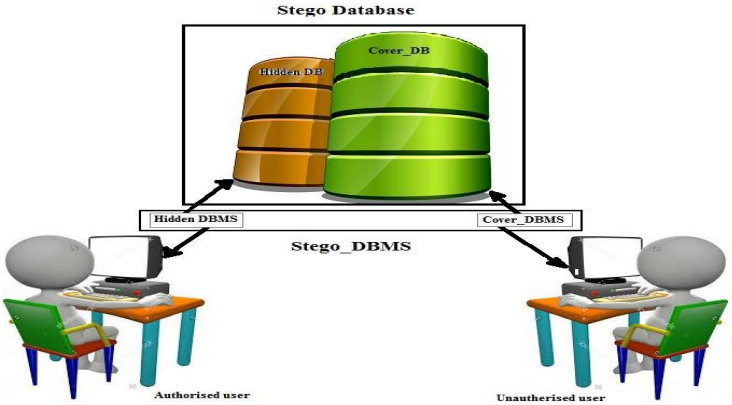

Fig 2: Simplified stego-DBMS.

Stego-Database Structure

A single database has a single conceptual schema and a single internal schema. However, a stego-DBMS is comprised of two databases, the cover-database and the hidden-database. [7]To fulfill the needs of a stego-DBMS under the ANSI-SPARC model, it has a three-level architecture including the stego-external, stego-conceptual, and stego-internal levels, as demonstrated in Fig 3.

The objective of the stego-architecture is to separate the unauthorized user's view of the cover-database from the physical representation of the cover-database, and at the same time, separate the authorized user's view of the hidden-database from its physical representation as well. This separation serves the same purposes as the original ANSI-SPARC model, and in this proposed system, it specifically aims to provide the database administrator (DBA) with the ability to alter the database storage structures without impacting the user's view and to change the conceptual structure of the database without affecting all users.

Components of a Stego-DBMS are split into several software components that perform specific functions, similar to a traditional DBMS. [7] Figure 4 illustrates the main components of a Stego-DBMS environment and how they interact with other software components. There are two types of user queries in a Stego-DBMS, queries made by authorized users and those made by unauthorized users.

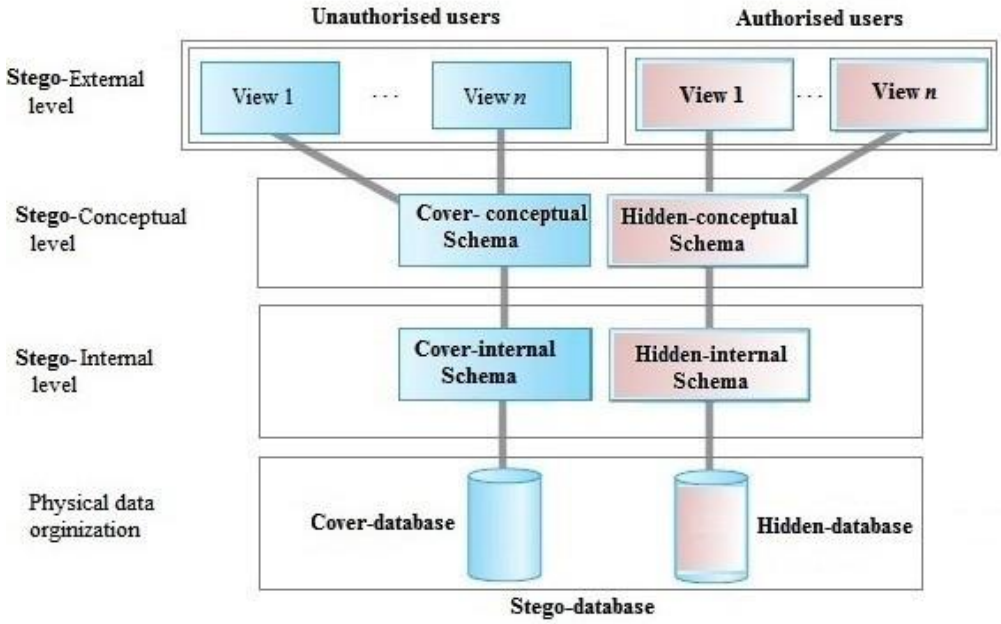

Fig 3: Stego-DBMS with ANSI-SPARC architecture.

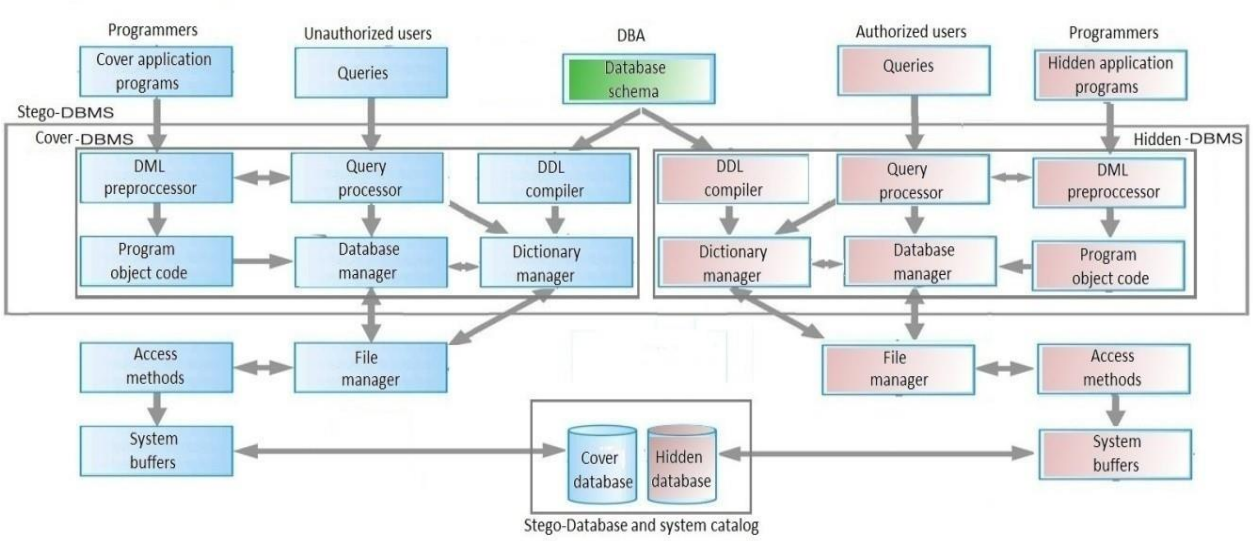

Fig 4: Components of Stego-DBMS.

Stego-Database Management System (Stego-DBMS) is a system that utilizes low-level commands and functions provided by programming languages. This allows the system to treat the table file as a stream of bytes or even bits, giving it the ability to display and access any part of the data contained in the table, even if it is beyond the boundaries defined in the table header. The Stego-DBMS can be activated through the use of an indistinguishable hot spot, the location of which is known only to authorized users.

The user interface for the Stego-DBMS is constructed from a complete set, and may differ from the user interface of the cover-DBMS.

2. Database inside an image using Steganography

The proposed Stego-DBMS will utilize digital images to store related data, while allowing data to be retrieved using standard queries instead of complex algorithms. The system will also provide application libraries that enable developers to use simple queries to fetch data and serve as a layer between the application and the stegano images during create, read, and delete operations. This paper proposes the use of a parallel data structure based on JSON to store information within JPEG images, using LSB-based steganography. JSON allows for easy storage and retrieval of related data, overcoming some of the limitations faced by traditional steganography from a developer's perspective. It is important to ensure that the image's integrity is not affected while storing the maximum amount of data possible. To achieve this, the optimum amount of data that can be stored in an image is calculated beforehand. The STEGDB package will provide the following functions for input images:

Stegdb.insert
Stegdb.upsert
Stegdb.delete
Stegdb.remove Stegdb.find.

[6]Figure 1 shows the architecture of the proposed Stego-DBMS, including the various operations handled by the STEGDB package.

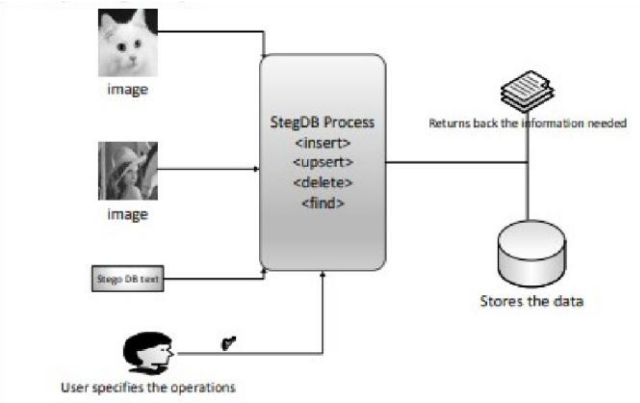

Fig 5: SteganoDB architecture.

For every request the package will check the function requested by the user.

- Inserts – The stegoDB package checks for the available space in the image and if the size request exceeds, the package willreturn an error.
- Upsert – The stegoDB package checks if there is an existing record, if the record exists it will be updated, else the record will beinserted as a new record.
- Delete – The stegoDB package will check and delete the record specified.
- Remove – The stegoDB package will remove all the records from the image.
- Find – The stegoDB package will list the records that match the search criteria. In order to encode the data in JSON structure,we will be using the LSB based Steganography algorithm. The entire encoding of data will be taken care by the package itself and the application developer needs not worry about the complexity of algorithms used internally within the system

Experimental analysis

The StegoDB comes as a package which can be included in .net based windows development tools. The initial package will also come with tools to create the database on a new image. Afterwards the DB functions can be called from the program created using the various .net technologies (C#, VB, C++) to perform the DB operations like insert, delete, upsert etc. The processing of the system will involve the below steps: First of all it gives an option to the user to select the cover image. The image file used in the Stego-DBMS can be in any format. [6] However, due to its advantage of reduced file size, the system utilizes the JPEG format for the output image as shown below.

Fig 6: Asking user for the image to be used for storing data.

In this system, the user can input information or data which is referred to as "stegan". The stegan data is then encoded and stored within a cover image, providing increased security for the data. The stegan data is similar to data found in a relational database and is hidden within the image until the user requests to retrieve it. [6] At that point, the stegan data becomes visible to the human visual system as shown below

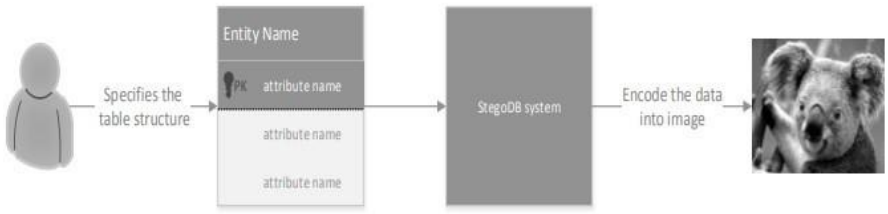

Fig 7: Data insertion step.

[6] At the retrieval side the data is first decoded from the cover image and then pass this data to the user as depicted below

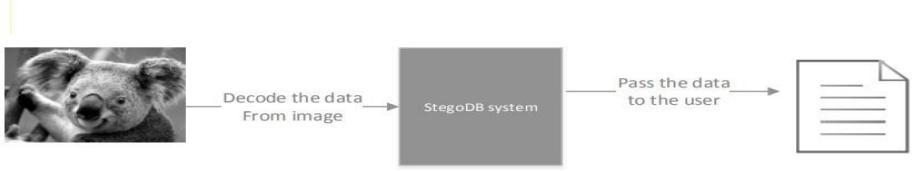

Fig 8: Data retrieval step.

The experimentation results were done on a PC by developing a Graphical User Interface in Windows to store the data into Stego DB and then retrieve and display it. [7] The below figure shows the main interface which allows to load an image and load data into it in JSON format.

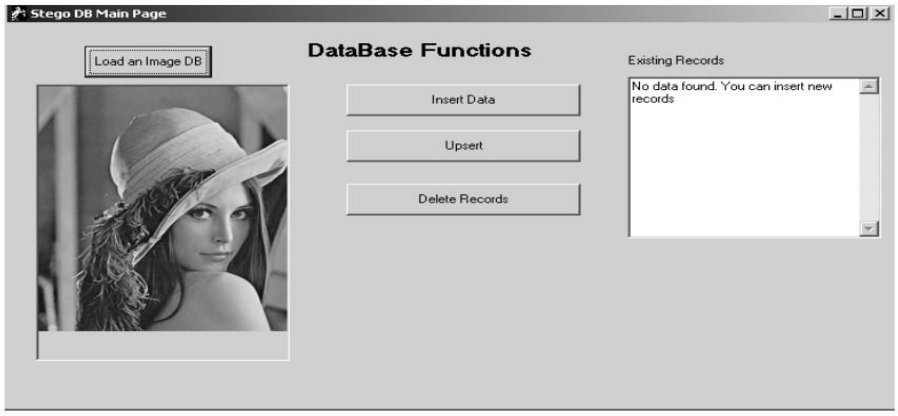

Fig 9: Main screen.

[7] There are separate options for inserting, upserting and deleting data. If there are no records existing it will display an errorsaying that no records could be found in below images

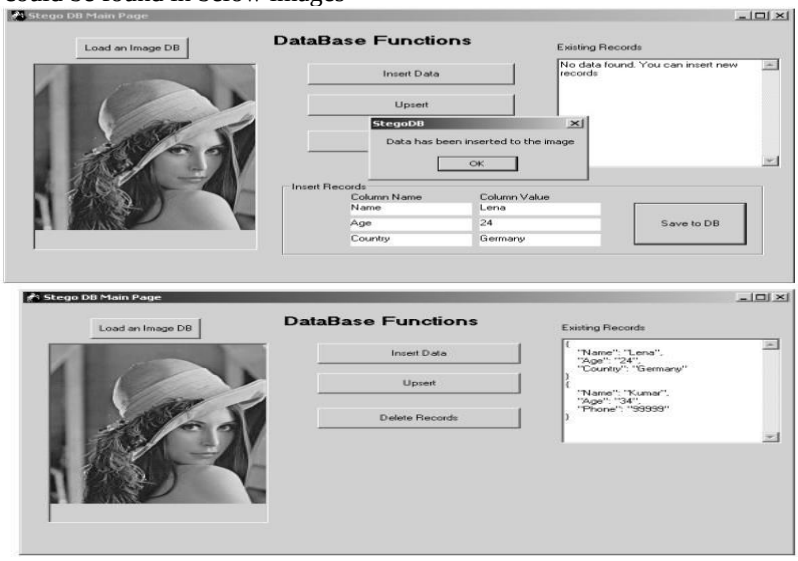

Fig 10: Data insertion step.

The records as per the input will be inserted into the image in JSON format using Steganography.

IV. COMPARATIVE STUDY OF SECURING DATABASE WITH TWO STEGANOGRAPHIC METHOD

Both methods of database steganography, hiding a database inside another database or inside an image, offer distinct advantages and disadvantages. The choice of which approach to use depends on the specific requirements and constraints of the situation.For smaller or limited amounts of data, hiding the database within a cover database may be the most suitable option. However, it is important to note that this approach does not provide additional security compared to simply storing the database as the outer database may be easily altered or damaged, potentially causing loss of the original data. Additionally, this method can result in

performance issues as the extraction of the original database from the outer database can be computationally intensive.On the other hand, hiding the database inside an image using steganography can be an effective method for protecting the confidentiality of the database and concealing its presence. This approach involves hiding the database in the least significant bits of the pixels in an image, making it more difficult to detect or alter the original data. However, it should be noted that this method is still susceptible to steganalysis. To enhance security, it is recommended to use a combination of steganography and encryption. The use of the StegoDB package is necessary for inserting and editing data on the images and there may be limitations in storing data before the image quality starts to degrade.

V. CONCLUSION

In our previous discussion, we explored the concept of steganography and the various methods for securing a database using this technique. Two of the most commonly utilized methods were discussed. Firstly, the technique of concealing a database within another database, which was found to be an ineffective method due to the numerous security risks involved. Secondly, the method of hiding the database within an image. To ensure the security of a database using steganography by hiding it inside an image, it is advisable to implement a combination of steganography and encryption. This approach will significantly increase the difficulty for any potential attacker to detect and access the hidden database, while providing an additional layer of protection for the sensitive data.

REFERENCES

1. Abraham Silberschatz, Henry F. Korth, S.Sudarshan,"Database Systems Concepts", 5th Edition, 2006.
2. Thomas M. Connolly and Carolyn E. Begg "Database systems A Practical Approach to Design, Implementation, and Management", 6th Edition, Pearson Education Limited 2005.
3. Paul J. Wagner "Database System Security ", University of Minnesota Summer School for Information Assurance, 2008.
4. Steganography Premium, 2004. www.clickok.co.uk
5. Ge Huayong, Huang Mingsheng and Wang Qian, “Steganography and Steganalysis based on Digital Image”, Proceedings of 4th International Congress on Image and Signal Processing, pp. 252-255, 2011.
6. R. Rejani, D. Murugan and Deepu V. Krishnan "Steganodb - A Secure Database Using Steganography", ICTACT Journal on Communication Technology, Septemper 2013, Vol: 04, Issue: 03.
7. Dr abdul Latif Ali Hussain **“**Database Steganography: Hiding Complete Database In Another Database”, AL-yarmouk
8. Journal, 2018, Volume 10, Issue :10