A Web Application to Perform Secured Searches on Encrypted Data

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By
MADHAV VISHVA PRATTIPATI

Spring 2015

Committee Members

Dr. Mario A Garcia                  ____________________________
Committee Chairperson

Dr. David Thomas                   ____________________________
Committee Member
ABSTRACT

There is a lot of importance in today’s computer technology for data storage. For every small or big organization, a firm, a company or an agency will need a storage space to store their respective data or information being it files, messages, images, etc,. When there is a need for organizations to store their data, they will also want their data to be safe and secure. This gives rise to data confidentiality and encryption of data. The basic reason for encryption of data is to reduce the security and privacy risks on untrusted servers. For an instance, if a user wanted to retrieve certain words of a particular document or file, the answer to such a query was not known previously without the loss of data confidentiality.

In this research, a solution to the problems of secured searching on encrypted data is implemented using the AES 128-bit. The system is encrypted using AES-128 bit encryption algorithm which has been declared the strongest among the other algorithms so far [20]. Encryption is done on text and numbers. A secured search scheme is implemented along with AES 128-bit encryption technique. The purpose of secured search scheme is to securely search for encrypted data meaning retrieving only requested data. Therefore, a system is designed in such a way that the ciphertext or the encrypted data stored on the untrusted server cannot learn anything about the plain text.
# TABLE OF CONTENTS

Abstract .......................................................................................................................... 2

Table of contents .......................................................................................................... 3

List of figures ............................................................................................................... 7

1. Background and rationale .................................................................................. 10
   1.1 Encryption and decryption ........................................................................ 11
      1.1.1 Symmetric and asymmetric encryption method ...................... 13
      1.1.2 AES algorithm .......................................................................... 13
      1.1.3 Security of AES ........................................................................ 16
         1.1.3.1 Brute force attack .......................................................... 17
         1.1.3.2 Mathematical attack ...................................................... 17
         1.1.3.3 Timing attack ................................................................. 17
   1.2 Existing methods for encryption ................................................................ 18
      1.2.1 DES algorithm ......................................................................... 18
      1.2.2 3DES algorithm ...................................................................... 18
      1.2.3 RSA algorithm ........................................................................ 19
      1.2.4 Security treatment with CBC encryption .................................... 19
   1.3 Methods for searching over encrypted data ............................................. 19
   1.4 Search considerations ................................................................................. 20
      1.4.1 Exact match .............................................................................. 20
      1.4.2 Word sub-match ....................................................................... 21
      1.4.3 Case sensitivity ......................................................................... 21
4. Implementation and results

4.1 Home page

4.2 Staff registration

4.3 Staff login

4.4 Opening new customer account

4.5 Deposit form

4.6 Withdrawal form

4.7 Searching for an account

4.8 Closing an account

4.9 Customer login

4.10 Account details summary

4.11 Encrypted database showing customer details

4.12 Code snippet for encryption on database

4.13 Encrypted database summarizing amount

4.14 Transferring funds

4.15 Customer password reset

5. Evaluation and testing

5.1 Test case 1

5.2 Test case 2

5.3 Test case 3

5.4 Test case 4

5.5 Test case 5

6. Conclusion and future work
LIST OF FIGURES

Figure 1: Overview of cryptography.................................................................10
Figure 2: Encrypting plaintext ........................................................................12
Figure 3: Decrypting ciphertext.................................................................12
Figure 4: AES algorithm based on 128-bit key ........................................15
Figure 5: Architecture of secured search technique.................................26
Figure 6: Flow chart of proposed technique .............................................28
Figure 7: Showing actual_table .............................................................30
Figure 8: Showing search_table.............................................................30
Figure 9: Architecture of proposed web application...............................32
Figure 10: Customer usecase diagram....................................................35
Figure 11: Staff usecase diagram..........................................................36
Figure 12: Global banking home page.......................................................38
Figure 13: Staff registration page ..........................................................39
Figure 14: Staff login page ......................................................................40
Figure 15: Opening a new customer account page.................................41
Figure 16: Deposit form page .................................................................42
Figure 17: Page displaying withdrawal form ............................................43
Figure 18: Displaying search form ..........................................................44
Figure 19: Account info page with input “tom” ........................................45
Figure 20: Search results for “tom” ..........................................................46
Figure 21: Showing user info page .........................................................47
Figure 45: Search results for ‘Bank’ on the search results page.........................72
Figure 46: Search form with input ‘p’ .................................................................73
Figure 47: Search_table showing username starting with ’p’...............................74
Figure 48: Search results for ‘p’ on search results page..................................75
1. BACKGROUND AND RATIONALE

In today’s technological world, every individual uses internet to store data, share data, upload data and many such things. Such kind of activities needs protection from the outside world in order to keep the data private and safe. This scenario gave rise to some techniques which protect the data from malicious or unauthorized users called encryption. The main purpose of encryption is to provide security to the data from unauthorized persons [5]. Cryptography is the study of secure and secret communication and encryption is a part of the cryptography’s study. Therefore, encryption involves many different techniques to secure data. Figure 1 [5] explains the basic classification of cryptography. It is divided primarily into two types: public key cryptography and private key cryptography. Private Key cryptography is further divided into block cipher and streams cipher.

![Figure 1: Overview of cryptography](image-url)
Block cipher is a method in which encryption takes place on groups of fixed length bits that are called blocks. They are used widely to encrypt data in bulk. On the other hand, in stream cipher, each binary digit is encrypted on at a time with the help of an encryption key. This is not used much as it adds large overheads.

1.1 Encryption and decryption

Encryption is a technique in which plaintext is converted into an unreadable form with the help of a generated key. The unreadable form is called ciphertext. It cannot be read by anybody except for privileged users. Decryption, on the other hand, is a technique in which the ciphertext is converted back into plaintext with the help of the generated key [1]. There are two possibilities here. The key used for encryption and decryption, both, can be same or different. There are various algorithms for encrypting and decrypting the data with the same or different key. The main target of any encryption algorithm is to increase the complexity or difficulty to break the ciphertext to plaintext without using the key. By this, it can be inferred that it is more difficult to decrypt a longer key [14].
In Figure 2, the plain text ‘365097432377’ which is in binary format is converted to an unreadable form ‘#we@8$h^8k&?j!g-‘ with the help of an encryption key. The encryption key is also in unreadable form, and this key length is generated depending on the encryption algorithm used.

![Encryption Diagram](image)

**Figure 2: Encrypting plaintext**

Figure 3 explains the decryption process of the ciphertext with the help of the decryption key. In this case, the encryption key and the decryption key are same. This process is called the symmetric key encryption method.

![Decryption Diagram](image)

**Figure 3: Decrypting ciphertext**
### 1.1.1 Symmetric and asymmetric encryption method

In symmetric encryption, a sender sends an encrypted message to the receiver using a private key. The same private key is used to decode the encrypted message. This is one of the oldest and best techniques which are commonly used [2]. The other name for symmetric encryption is private key cryptography. Some of the common algorithms which come under this method are DES (Data Encryption Standard), AES (Advanced Encryption Standard), Blowfish, IDEA (International Data Encryption Algorithm). In the current situation, AES is acquiring popularity because of its efficiency and security when compared to the other algorithms.

Asymmetric encryption is also called as public key encryption technique [21] which doesn’t use the same key for encrypting and decrypting. It uses two keys: one of which is the public key available to whosoever wants to encrypt data, and private key is available only to the user of the corresponding public key. In other words, the private key is maintained secretly. For asymmetric cryptography, there are certain set of properties which have to be satisfied [2]. Some of the common examples of asymmetric encryption are RSA algorithm, Digital Signature algorithm, Diffie-Hellman algorithm.

### 1.1.2 AES algorithm

The AES (Advanced Encryption Algorithm) is the latest standard encryption algorithm which is widely used after it was launched. The other name for AES algorithm is Rijndael. Two Belgian cryptographers named Joan Daemen and Vincent Rijmen developed AES, which is a block cipher technique for encrypting electronic data. It was approved by the U.S. National Institutes of Standards and Technology (NIST) in the year
2001. After the announcement of the AES technique, in competition to this, fifteen different techniques were taken to compare and evaluate the performance of AES. U.S government chose AES as a preferred technique out of all. AES algorithm replaced DES, which was most widely used before the establishment of AES [3]. There were many advantages of AES over DES, which could help in building a stronger encryption method.

Advanced encryption algorithm supports variable key sizes, unlike the other algorithms. It supports key sizes of 128 bits, 192 bits, and 256 bits. Each key size has different number of rounds of encryption. But it allows a fixed length of input data which is 128 bits, and these 128 bits are further divided into four fundamental operational blocks. These operational blocks are considered as an array which is organized in matrix form of the order 4x4. This matrix representation of the operational blocks is called state. The number of rounds performed on the key depends on the key size which is used. For a 128 bit key, 10 rounds are performed. For a 192 bit key, 12 rounds are performed. For a 256 bit key, 14 rounds are performed.

Figure 4 [2] explains the encryption and decryption process of the AES algorithm using a 128-bit key length. As 128-bit key length requires 10 rounds to be performed, each round performs few specific functions. Therefore, each round performs all the four byte-oriented transformations which except for the last round. The last round, i.e., the 10th round contains only SubBytes, ShiftRows, AddRoundKey leaving out MixColumns. As the figure 4 shows the initial round that is round0 is performed with AddRoundKey followed by the other rounds repeating nine times using other transformations. The sequence of transformations performed while encrypting are the same for the decryption
The encryption and decryption process of the data involves modification of the state array. The modification process consists of four different transformations of byte orientation [2] which are shown in figure 4 are explained below.

1. **SubBytes transformation:**

   This step involves a non-linear substitution wherein each byte is replaced with another byte within the matrix based on a table called S-box or substitution table. This substitution table two major operations and can also be invertible.
• GF \((2^8)\) field which is inverted, a polynomial which is irreducible with a modulo 
m(x) = x^8 + x^4 + x^3 + x + 1 [2].

• An affine transformation which is a 2D geometric transformation defined as \(Y = AX^{-1} + B\). In this definition, A is a fixed matrix of the order 8x8, B is also a fixed matrix of the order 8x1 and \(X^{-1}\)is the bytes of input from the state array.

2. **ShiftRow transformation:**

This step involves transposition which means moving bytes to a few blocks from its original position. Here, cyclic shifting is done on each row of the state array matrix to certain number of steps. The definition of the ShiftRows transformation in mathematical terms is 
\[ S_{r,c} = S_{r,((c+\text{shift}(r,4)) \mod 4)} \]
where \(\text{shift}(r,4)\) which is the shift value depends on the row number.

3. **MixColumns:**

This step consists of a mixing operation that is operated on the state array matrix’s columns which combines the four bytes present in each column. Or it can be said that a simple matrix multiplication of the columns takes place.

4. **AddRoundKey:**

This step involves a simple XOR operation based on the AES key schedule algorithm. The XOR operation is performed between the round key and each byte of the state array matrix. The round is obtained from the cipher key [4].

1.1.3 **Security of AES**

So far, AES was not prone to any kind of attacks practically. The strength of AES
is determined by some of the attacks which are prone to strike AES [2].

1.1.3.1 Brute force attack

It is a type of attack which is used against encrypted data such that it can break through it by checking all the possible passwords or keys systematically. It’s just like the trial and error method where it performs searches on the entire search space until it finds the right key. In AES algorithm, the minimum key length is 128 bit, and the number of possibilities of the key is $2^{128}$ which is a very huge key space. It is not feasible to search for key and thus considered impractical [2].

1.1.3.2 Mathematical attack

In recent days, Warren D. Smith proposed an analytical method based on mathematical derivation which has the possibility of cracking the AES algorithm. The analytical method is designed for working on the 256-bit key length of the AES algorithm to extract the pairs of plaintext-ciphertext. But, this is just an assumption as it was not practically implemented [2]. Therefore, there was no known mathematical attack working practically against the AES algorithm till now.

1.1.3.3 Timing attack

Timing attack is categorized under side channel attacks, and AES is quite insecure with timing attacks. Timing attack is also called as ciphertext-only attack which occurs in the implementation level that depends on the input rather than depending on the fixed run time. AES has high performance and has high speed software implementation procedure which is more susceptible to such kind of attacks [2]. The only solution to this can be adding delay such that the timing differences can be hidden. But by doing this, the
performance of the system is at stake.

1.2 Existing methods for encryption

There are some encryption techniques which are commonly used in applications to convert plaintext to ciphertext. They are explained below.

1.2.1 DES algorithm

One of the most commonly used encryption algorithm is data encryption standard algorithm which was established in the year 1972 by IBM. Later, it was approved by the U.S government as one of the standard encryption algorithm. The encryption process is begun with a 64 bit key length [6]. DES has good flexibility as it can be operated in different modes such as CBC, ECB, OFB, etc [4]. In DES algorithm, on every 64-bit block of data, a 56-bit length key is used. This is a private key block cipher encryption technique which uses the same key for encryption and decryption. It performs 16 rounds on the key.

1.2.2 Triple DES algorithm

Triple DES is a technique which applies the algorithm thrice. There are two kinds of triple DES, one in which three keys are used and in the other; two keys are used for encrypting plaintext and decrypting ciphertext. While using three keys, to encrypt the data, first key K1 is used for encrypting plaintext followed by K2 and K3. Here all the keys K1, K2, K3 are different from each other. To decrypt the ciphertext, encryption operation should be performed on the plaintext with key K3 initially and then with key K2 and key K1 to obtain the final result of the ciphertext. Now, when using two keys, the plaintext needs to be encrypted using key K1 followed by K2 to obtain the ciphertext. K1
is used for decrypting the ciphertext to get the final plaintext [6].

1.2.3 RSA algorithm

RSA algorithm is an asymmetric encryption technique or the public key encryption method in which the key used for encryption and decryption are different. The key used for encryption is available publically whereas the key used for decrypting the ciphertext is kept secret. RSA algorithm primarily involves three steps. The first is key generation; second is encryption and the third is decryption. RSA algorithm works well with large key unlike the other algorithms as it works with 2048 bits [7].

1.2.4 Security treatment with CBC encryption

The concept of security treatments of the symmetric encryption [9] is discussed to enable the users to have a better protection strategy towards their encrypted data. Four different notions of security were introduced against the attack of chosen plaintext. Some methods have been implemented to encrypt data using CBC, which is a block cipher encryption technique which is the most widespread encryption method in early 90’s. This methodology is also used in the searching techniques of encrypted data. But the major drawback of this system is that it cannot provide security if there is any accidental modification of data or any malicious tampering of data. Here, the initialization vector (IV) of the message is not kept safe. If attacker knows the IV of the message, it’s easier for him to guess the next plaintext. This kind of attack is known as the TLS CBC IV attack.

1.3 Methods for searching over encrypted data

There is a necessity to retrieve data from the database when it is stored in them. When
the data is stored in an encrypted format, retrieving data is a challenging task. The challenge faced here is to retrieve only the required word which is searched for and nothing else should be known to the server. The database has to search for the plaintext over the ciphertext which is available there. There are two major ways of searching. They are:

1. Decrypting all the columns

In this method, when a user requests for a particular word, the database tries to match it with the existing data which is present in it. But as the data in the database is encrypted, it decrypts all the data for it to find a proper match. This method takes a significant amount of time and also reduces the performance of the SELECT query used for retrieving the data from the database [22].

2. Decrypting only required fields

Here, only the requested data is searched for and decrypted [22]. This is a complex task as the database has to have a technique to recognize the field which has to be decrypted without touching other fields.

1.4 Search considerations

While performing searches, there will be different search patterns or scenarios. These scenarios help in improving the search functionality to obtain better results. Some of the scenarios [19] in which the query is matched for the desired word are mentioned below:

1.4.1 Exact match

The basic step in searching is to search for an exact match where the full word is entered which has to be retrieved. For example, if “hello” should be retrieved, then all the
rows and corresponding columns containing “hello” will be retrieved.

1.4.2 Word sub-match

Word sub-match matches partially entered value to all the values present in the record. For example, if a table contains the word ‘management’, it should be able to return that value even if the user searches for ‘manage’.

1.4.3 Case sensitivity

The search query can be made case sensitive or case insensitive. Case sensitivity is more complex than case insensitivity because sometimes the host environment cannot support it, and it largely depends on upper case and lower case that should be considered each time [19].

1.4.4 Proximity based query

Proximity based query retrieves data which is closely related to the word to be retrieved. There are certain bounds which are considered for proximity based query. These bounds are defined very close to each other. For example, B appears before A [19].

1.5 Private information retrieval

There were certain problems when other researchers tried to investigate the usage of an untrusted server when searching on encrypted data. There were some secure multi-parity computation functions [11] which were studied to find a solution. But if there was a solution for it, then it would require high overhead as it needs multiple servers which is a disadvantage. Another approach called the Private Information Retrieval (PIR) [12] where it enables the user to access the data from the database and retrieve the information privately and thus, there’s no identity disclosure of the retrieved data. Therefore, the
clients can have access to the data entries of a particular table without revealing the interest of the client. But this scheme is very hard to work. It has many practical issues wherein it requires several non-colluding servers. Also a large amount of bandwidth is consumed and does not guarantee data confidentiality. Private keyword searching is also very hard to achieve in this scheme and thus does not support it. Whereas, in the scheme mentioned in this paper has low computational complexity as well as works well with private keyword searching.
2. NARRATIVE

2.1 Problem statement

Many applications run on web servers which store data on untrusted servers. Such kinds of servers are prone to malicious attacks which can steal the data and misuse it. Therefore, there is a need to secure the servers from unauthorized personal. AES is one of the strongest algorithms to encrypt data with several rounds performed on the key which enables the attacker to easily decode the key. Searches on encrypted database is a challenging task as it requires the database to recognize the desired data from the available ciphertext.

2.2 Motivation

There were many real-time attacks on database servers of many companies. The hackers break into the database and steal all the records of it. The hackers try to exploit the security vulnerabilities of the database system in order to intrude into the system. These attacks created huge losses to the companies, and they were threatened by the hackers thus, creating damage to their properties. Some of the recent attacks are mentioned below:

1. A very well-known company ‘Sony Pictures’ was recently threatened by hijackers who successfully captured most of the sony’s database. This attack happened in November 2014. The hijackers compromised more than 170,000 emails and 30,000 documents from their databases which contained important files and conversations regarding legal opinions. The attackers threatened the company and posted many of these private issues on the web [16].
2. In August 2014, more than 200 health organizations which have interconnected databases were hacked and more than 4.5 million patient records were stolen. These hospitals network operates in more than 25 states, states including Texas, Alabama, Florida, etc. The hijackers took control over several patients’ records and gained access to social security numbers, credit/debit card details, addresses and phone numbers [17].

3. In December 2010, The Gawker Media which is an online forum was hacked. More than 1.2 million accounts of the commenter’s of the blogs were hacked [18].

On seeing all these security breaches on the databases, there is a need to secure the database servers to improvise the privacy and security of the untrusted servers.

### 2.3 Project Objective

The project’s main objective is to develop a dynamic web application which stores encrypted data files and solves the problem of searching on the encrypted data by using one of the best cryptographic technique called the AES 128-bit encryption standard. A secure online banking web application is developed which provides security to the bank’s customer’s data which is stored on external bank server. The AES 128-bit algorithm is approved by the U.S National Institute of Standards and Technology (NIST). It is a symmetric encryption technique using the same key for encryption and decryption. This algorithm is applied on the database or the server. Therefore, it has specific functionalities which have to be satisfied in order to protect the data from malicious attacks.
3. PROPOSED SYSTEM DESIGN

The proposed system is an online secure banking web application which helps the users or the customers of the banking system to securely store their data and transactions on the server. The primary implementation of the proposed system design is the application of AES 128-bit encryption technique on the database servers to attain confidentiality and security to the data from external hackers. The AES 128-bit algorithm is applied to MySql database such that the input plaintext is converted to ciphertext which is not readable. The strength of the ciphertext depends on the complexity of the produced ciphertext and the length of the key used for encryption. The AES encryption technique is applied on MySql server which works on back-end of the application.

To achieve secured searches on the encrypted data, secured search technique is implemented on MySql database which searches for plaintext in the database and retrieves all the data which matches the desired data to be retrieved [22]. It is a challenging task to retrieve plaintext from MySql tables while performing searches over the ciphertext.

The plaintext is encrypted using AES_ENCRYPT() function and to decrypt the data, the server searches for cipher text in the database and decrypts it using AES_DECRYPT(). These functions are applied on MySql database which acts as a web server. The main contribution to this research is performing secured search technique on the data which is requested.
3.1 Architecture of secured search technique

Figure 5 explains the architecture diagram of the proposed secured search algorithm which includes three conditions mentioned below the architecture.

![Architecture Diagram]

Figure 5: Architecture of secured search technique

C1: Checks for the condition if the requested query is on the encrypted data column or not

C2: Validation or authentication of the user

C3: If there exists any records

3.2 Proposed secured search technique

The proposed security search technique [22] contains two main tables called the actual_table and search_table as shown in figure 5. The purpose of maintain these two tables is for security. The actual_table consists both encrypted and unencrypted data along with a key column. The columns which are sensitive will be encrypted using the
AES 128-bit technique and stored in it. And the columns which contain general data are not encrypted. The search_table consists of a copy of sensitive data from the actual_table and also the key column. Here, the key column is kept encrypted, and the sensitive data from the actual_table is kept unencrypted. Now, the records present in the search_table are shuffled so that it’s not in the order of the actual_table. Therefore, the hacker will not know the exact key for the data which is required. Therefore, the relationship between the actual_table and the search_table are hidden. The search_table is stored in a secure schema which can be accessed only by the authorized persons for searching purposes. In addition to this, search_table columns are given different names when compared to the actual_table in order to deceive the hackers [22]. Here, the algorithm is applied not on just one encrypted column of the search_table but many other columns which are necessary to keep them safe.

The encryption of the columns depends upon the application which is being developed. The data which has to be encrypted can be more than one column as represented in the online secured banking application. In this case, the account number, username and amount are encrypted. Even if the user gives one input, it should be able to retrieve all the columns of the encrypted data in plaintext form.
3.3 Flowchart of the proposed technique

In figure 6, the control flow of the system is depicted which gives a clear picture of how the application tries to search over the encrypted data. The flow chart primarily explains the secured search algorithm which is implemented for the secured online banking application.

![Flowchart of the proposed technique](image)

Figure 6: Flow chart of the proposed technique

3.4 Search process

When an authorized user tries to retrieve a record from the database, the first thing it does is checking whether the search has to be performed on the encrypted column or the
unencrypted column as shown in figure 6. If the search has to be conducted on the unencrypted column, it directly passes the results which are available in the column to the user without touching the search_table. If the search has to be conducted on the encrypted column, then it performs search on the search_table which contains the key for decrypting the value for the required record from the actual_table. After running the search query, it returns the key to the actual_table depending on the search condition which is executed. The records are returned to the user from the actual_table using those keys. Therefore, it can be said that only those fields are returned which satisfy the written query. In this way, data confidentiality and performance is improved [22].

The decryption of records are done at two places: one is when the key has to be decrypted in the search_table and the other is when decrypting the actual record from the actual_table. Thus, the proposed technique has a good performance when compared to the traditional technique which decrypts all the records to search for data [22]. In proposed technique, only those records are decrypted which satisfies the query given by the user. Below is an example which explains the search process in detail.
Figure 7 represents the actual_table which contains encrypted as well as unencrypted columns along with ‘key’ column. There can be more than one encrypted column in the actual_table.

<table>
<thead>
<tr>
<th>Key</th>
<th>First name</th>
<th>Last name</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bob</td>
<td>Thomson</td>
<td>Encrypted</td>
</tr>
<tr>
<td>2</td>
<td>Peter</td>
<td>Drew</td>
<td>Encrypted</td>
</tr>
<tr>
<td>3</td>
<td>Paul</td>
<td>Walker</td>
<td>Encrypted</td>
</tr>
<tr>
<td></td>
<td>....</td>
<td>.....</td>
<td>.....</td>
</tr>
</tbody>
</table>

Actual_table

Figure 7: Showing actual_table

Figure 8 shows the search_table which contains actual_tables’ encrypted columns in plaintext form. It also shows that the column names are written differently to deceive the attacker.

<table>
<thead>
<tr>
<th>Abe (Key)</th>
<th>Xyz(Username)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted</td>
<td>Pwd90</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Bob320</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Pauline</td>
</tr>
<tr>
<td></td>
<td>.....</td>
</tr>
</tbody>
</table>

Search_table

Figure 8: Showing search_table
If the user searches with first name ‘bob’, the algorithm first checks if the search has to happen on the encrypted column or the unencrypted column. In this case, the column is unencrypted. So, the results are retrieved directly from the actual table, without traversing through the search_table. If user searches with username ‘pauline’, it checks if the username column is encrypted or not. As the username column is encrypted, the database searches over the search_table for ‘pauline’. After it finds the match, it takes the corresponding key and decrypts it. This decrypted result is again taken back to the actual_table to find the match for the same key in ‘key’ column. The corresponding row in which the match is found will be decrypted and those decrypted results will be sent to the user displaying them in the presentation layer.

3.5 Architecture of proposed web application

In figure 9, there are 5 layers which connect with each other for primary communication. The proposed system works exactly as shown in figure 7. The first layer is the client layer which contains the web browser. The user or the client accesses the web browser to connect to the pages. Second is the presentation layer which contains all the java server pages which help in connecting to the business logic. The presentation layer acts like a bridge between the client layer and presentation layer. According to the proposed system, this layer contains the jsp files like login.jsp, closeac.jsp.
Next, the business layer contains the business logic which plays the main role in executing the program and producing the result. They contain the java classes as this system is using java programming language. The business layer acts as a bridge between presentation layer and the data access layer. The data access layer contains the drivers to connect to the database server which is MySql in this case. The last layer is the database layer which stores all the data in MySql and to retrieve the data from it. Here, the encryption and decryption operations take place with the help of AES_ENCRYPT and AES_DECRYPT functions.

3.6 Modules and functionalities of the system

There are four main modules of the system whose functionalities have to be known. They are: staff module, customer module, database module and search module. There are
several functionalities of the four modules which are mentioned below:

3.6.1 **Staff module**

If the staff is not yet registered, they can register themselves who have unique id’s given by the bank organization. This prevents the unauthorized persons from misusing the account. The staff can create a new customer account with the specified details asked, except for the password which should be created by the individual customers themselves when they login. The staff can also search for a particular account and also delete an account. Here, the searching is done on the encrypted data. All the data entered by the staff is stored in the server in an encrypted form.

3.6.2 **Customer module**

The customer can view all the details of his account and the current balance. Changing the password is also enabled which will be securely stored in the database in an encrypted format. The customer can transfer amount from one account to another.

3.6.3 **Database module**

This is a very important module which stores all the data in an unreadable format. The database used is MySql, which performs encryption and decryption of the input plaintext using AES 128-bit encryption technique. The main functionality of this module to achieve provable secrecy.

3.6.4 **Search module**

In this module, secured search technique is applied that helps in finding the appropriate input which is asked to retrieve. The secured search is performed on the encrypted data (ciphertext) present in MySql database. The first step is giving the input
word which had to be retrieved. The input plaintext is passed to the database and runs the algorithm. If there is more than one match, then they are stored in an array list, even otherwise, the array list contains the match. This list is decrypted using the AES_DECRYPT() function using the same key and finally sent to the user. This way it retrieves only the matching data which achieves provable secrecy.

3.7 Advantages of the proposed system

One of the crucial advantages of using the proposed cryptographic scheme using AES 128 bit along with secured search technique for the problem of searching on encrypted data is provable secrecy [1]. Provable secrecy is where the untrusted server when having the ciphertext (encrypted data) cannot guess or learn about the plaintext. This is a very important functionality in which the servers retrieve the related document which was searched for without having any knowledge about the content of the document which is encrypted. Some standard definitions of security were adopted from the literature of provable security [10], and the hardness of the cryptographic primitives is measured in terms of the resources required to break them.
3.8 Use case diagrams

Figure 10 depicts the use case diagram for the customer module. There are five major tasks performed by the customer.

Figure 10: Customer use case diagram
Figure 11 depicts the usecase diagram for the staff module. There are eight major tasks performed by the staff member.

![Staff Use Case Diagram](image)

**Figure 11: Staff usecase diagram**

### 3.9 Environment

#### 3.9.1 Eclipse Luna

Eclipse Luna is a tool which is used for executing java programs. It is used to develop different kinds of applications including android and web applications.
3.9.2 XAMPP server

Xampp acts as a web server and provides functionalities that a web server possesses. It is an open source software.

3.9.3 Tomcat 7

Tomcat 7 is also open source software used to run JSP (java server pages) and Java EE.
4. IMPLEMENTATION AND RESULTS

An online bank application is developed which is implemented to secure the data stored by the customers of the bank.

The system is implemented by using eclipse IDE with the help of a Xampp server which acts as a web server in this case. The eclipse IDE contains all the classes and JSP’s which work for presenting the application through the application layer and also communicate with the database. The Xampp server stores all the customer records in an encrypted format.

4.1 Home page

In figure 12, the home page of the web application is displayed. The home page contains options for the staff to register and login. Other users cannot register as the staff because there are unique id’s given to them. Customer login is also provided here.

Figure 12: Global banking home page
4.2 Staff registration

Figure 13 shows the staff registration page where the staff member is allowed to register for online banking to manage the customers’ accounts. After the validation is done, the staff member can login and create a new customer account as well as make changes to several customer accounts. The staff member will not know the passwords created by the customers at any point of time.

![Staff registration page](image-url)

Figure 13: Staff registration page
4.3 Staff login

Figure 14 shows staff login page where the staff member should enter the username and password to proceed further. These details are validated from the database. The database should search among the encrypted ciphertext to match the input details of the staff member.

Figure 14: Staff login page
4.4 Opening new customer account

In figure 15, the staff member can open a new customer account filling all the details asked for. These details, when stored in the database are encrypted twice according to the proposed system design. Even if any one of the field is left empty, none of the records are stored into the database. It shows an error saying to fill all the fields. The bank account number is created automatically and displayed on the same page if the form is filled correctly without any error.

Figure 15: Opening a new customer account page
4.5 Deposit form

Figure 16 shows the deposit form filled by the staff member when the customer deposits money to the account through the bank. The primary parameters required to deposit to the customer’s account are the username and account number of the customer. Once it’s submitted, then a success message is displayed.

![Deposit form page](image.png)

Figure 16: Deposit form page
4.6 Withdrawal form

The customer can withdraw money from the bank when huge amount is required. The staff member updates the customer records by using the withdrawal form. The parameters to be known are account number and username of the customer’s account in figure 17.

![Withdrawal form screenshot](image)

Figure 17: Page displaying withdrawal form
4.7 Searching for an account

The staff member can search for an existing customer account from the database. As the database contains encrypted data, a secured search technique is implemented which contains two main tables such that it can search over the encrypted data when plaintext is given as input. Figure 18 shows the search page which has the “User info” which is used to check general information of the user and “Accounts info” used to search account related information of customer.

![Displaying search form](image)

Figure 18: Displaying search form
When “Accounts info” button is submitted, it navigates to next page displaying form to search for account related information like bank account number, username and amount. “tom” is given in the input field of username in figure 19.

Figure 19: Accounts info page with input “tom”
Figure 20 shows the search results with username “tom”. After the value “tom” is passed to database, it checks if the column is encrypted or not. If it's encrypted, it performs search over the search_table first and then retrieves data from the actual_table using the corresponding key.
Figure 21 shows the page displaying “User info” which contains general information regarding the customer. The staff can enter one value or more than one value. The User info page retrieves data from the actual_table as it doesn’t contain any sensitive data which is encrypted.
4.8 Closing an account

The staff member can close an existing account as shown in figure 22. Only the staff member has the authority to delete or close an account. When an account is closed, the customer’s record is deleted from all the tables present in the database. When the account number and the username of the closing account are given, it displays the profile details as well as the balance left in the account.

Figure 22: Page for closing account
Figure 23 shows the confirmation page for closing an account. When the account number and the username of the closing account are given, it displays the profile details as well as the balance left in the account. When the close A/C option is selected, it asks for another confirmation. Once the account is deleted, the operation cannot be rolled back.

Figure 23: Confirmation to close account
4.9 Customer login

The customer login page display the parameters required to validate the account. For any customer, it is a must to know their account number, username, and password. Just the username and password are not sufficient to login. In figure 24, when the user login’s for the first time, they are asked to enter the password twice.

![Customer login page]

Figure 24: Customer login page
4.10 Account details summary

The customer can view the summary of the account including balance. The email, phone number and address associated with the account are also displayed in figure 25.

![Figure 25: Summary of the customer’s account details](image)

Figure 25: Summary of the customer’s account details
4.11 Encrypted database showing customer details

Figure 26 shows the way in which the plaintext is stored in the database after the application of the AES 128-bit algorithm.

Figure 26: Showing details in encrypted form
4.12 Code snippet for encryption on database

Figure 27 shows the encryption technique and the methods used for encrypting the plaintext on MySQL database. AES_ENCRYPT() is used to encrypt the input which is coming from the presentation layer. The input which is reached here is already encrypted on the java side. Therefore, double encryption happens here.

```java
String sql = "INSERT into userdetails (username, firstname, lastname, phone, email, address, gender) VALUES "
    + "(AES_ENCRYPT('john', '' + key + ''),"
    + "AES_ENCRYPT('john', '' + key + ''),"
    + "AES_ENCRYPT('665554433', '' + key + ''),"
    + "AES_ENCRYPT('john@gmail.com', '' + key + ''),"
    + "AES_ENCRYPT('US', '' + key + '),"
    + "AES_ENCRYPT('male', '' + key + '))";
```

Figure 27: Code snippet for encryption on database
4.13 Encrypted database summarizing amount

Figure 28 shows the database of the customer along with the account number, username and amount. This is the account table in MySql which is used. The data looks like the below for all the customer data which is entered.

![Figure 28: Encrypted account table](image-url)
4.14 Transferring funds

In figure 29, the customer can transfer funds from one account to another account. The sender has to know the account number of the receiver in order to perform this transaction.

Figure 29: Transferring funds page
4.15 Customer password reset

The customer can reset the password with a different one after logging in for the first time in figure 30. When the user login’s for the first time, by default the username and password are given same.

Figure 30: Password reset page
5. EVALUATION AND TESTING

Testing should be done to check every possibility of detecting a fault or any malfunctioning of the developed system or application. Testing helps in improving the quality of the product and makes sure that the application performs what it’s supposed to do. For any product or application developed, the final stage of the software refinement is the testing stage. Basically testing is done by various professionals like the programmers, analysts, etc.

The system is tested with various inputs as shown below in the test cases section. The system is tested for encryption of data in the database, which is the first functionality of the proposed system. The second functionality is also tested if the search results are appropriately retrieved from MySql database. When the user adds a new customer account, the data pertaining to his account will be encrypted and stored in the database with the AES 128-bit algorithm. Next, when the user tries to retrieve a particular customer account either with his first name, last name, phone number, email, address, the data will be retrieved from the actual_table itself. When the user searches with the input username or account number, as these columns are encrypted, the data will be retrieved from the search_table.
The following are some test case scenarios:

5.1 **Test case 1**: Verifying if the customer information provided at registration is successfully encrypted or not.

Step 1: Filling all the fields

While opening a new customer into the database, all the required fields are to be filled. Otherwise, the database will not enter any of the record entered in it. The input given is text and numbers which will automatically encrypt as a new user in the database. Figure 31 shows the image filling all the customer details.

![Figure 31: Filling all fields with customer details](image-url)
Step 2: Viewing ciphertext of all the data entered

In figure 32, shows the ciphertext of all the data entered when the ‘ciphertext’ button is submitted. This ciphertext is retrieved from the database to verify if it is the same in the XAMPP server and user interface. A piece of code is written which brings the ciphertext of the entered values from the database to the presentation layer.

Figure 32: Showing ciphertext for the entered fields
Step 3: Submitting new account form

Figure 33 shows the result after submitting the new account form. It automatically generates the bank account number as it can be seen in figure 33. The account number is incremented for every new account created. Bank1004, which is highlighted in the below figure, is the account number assigned to the customer with username ‘peter’.

![Figure 33: Submission of new account form](image.png)
Step 4: Verifying if data is encrypted in database

In figure 34, the XAMPP database containing the ciphertext is highlighted for the new customer. On comparing figure 32 and figure 34, it can be assured that the data is encrypted in the database.

Figure 34: Verifying if data is encrypted
5.2 Test case 2: Verifying the process of successful search with first name “tom” in User info page.

Step 1: Entering first name “tom”

In figure 35, the search form is filled with the value “tom” in first name field.

Figure 35: Giving input in the search form
Step 2: Checking database if the ‘first name’ column is encrypted or not.

In figure 36, highlighted area shows the ‘first name’ column which is not encrypted. Therefore, the information is directly fetched from the actual_table.

Figure 36: Checking if the “first name” column is encrypted or not
Step 3: Page displaying results for input “tom”

Figure 37 shows the results page when “tom” is searched. It retrieves all the records of the columns which are requested in the User info page as they are not encrypted. Even if one column is encrypted, it goes to the search_table. Along with secure retrieval, it also satisfies the pattern match.

![Image of search results](image.png)

Figure 37: Results of search in application page for “tom”
5.3 Test case 3: Verifying the process of search with input as “pratti” in the username field.

Step 1: Entering username “pratti” in the input field.

In figure 38, the search form is filled with value “pratti” in username field. Once the search button is submitted, the username column is checked if it’s encrypted or not in the actual_table. If it’s encrypted, it’s redirected to search table to retrieve the key for the corresponding search.

![Figure 38: Displaying account info page with input “pratti”](image)
Step 2: Checking if the username column is encrypted or not

First, the input is evaluated if it is on the encrypted column or not. In figure 39, highlighted area shows that the username column is encrypted in the Xampp server which acts as a web server. So, the results cannot be displayed until they are decrypted using a key. Next, search will be performed on the search_table.

![Figure 39: Showing “username” column is encrypted]
Step 3: Control is passed to search_table to retrieve key

In figure 40, the highlighted region shows the match for the requested record “pratti”. The corresponding key is taken from there and passed on to actual_table for decryption. The key here is first decrypted and then passed on.

Figure 40: Matching username “pratti” to retrieve key
Step 4: Decrypting results from actual_table using the matched key from search_table

In figure 41, the highlighted region shows the values to be decrypted using the key which matched after decrypting the search_table key. Along with the results of the decrypted columns, other columns will also be retrieved as they are also requested in the user's query.

Figure 41: Matching key obtained from search_table to actual_table
Step 5: Results displaying records of “pratti” after successful secured search

Figure 42 displays all the records requested for the input “pratti” in the Account info page. The search_table is used only when search has to be performed on encrypted data. Therefore, here search_table is used to search as the requested column was encrypted.

![Secure Banking Complete Encryption](image_url)

Figure 42: Displaying results of “pratti” by using search_table
5.4 Test case 4: Verifying the search condition ‘sub-match’ for the input ‘Bank’ in account number field.

Step 1: Entering input ‘Bank’ in account number field.

Figure 43 depicts the form to search for the account details of the account holders holding bank account number starting with ‘Bank’.

![Search form with input ‘Bank’](image_url)
Step 2: Database displaying account number starting with ‘Bank’

As the search is done on the encrypted column, the search_table is given control to search for the account number starting with ‘Bank’ as highlighted in figure 44. After it matches the data in the search_table, the corresponding key is decrypted and again matched in the actual_table to decrypt the original results.

![Figure 44: Search_table showing account number starting with ‘Bank’](image)

Figure 44: Search_table showing account number starting with ‘Bank’
Step 3: Showing results for account number starting with ‘Bank’ in the search results page.

Figure 45 shows the search results for input ‘Bank’ in the search results page which are retrieved from the actual_table via the search_table. As all the account numbers start with ‘Bank’, all those are retrieved. Thus, achieving the sub-match query search as well.

Figure 45: Search results for ‘Bank’ on the search results page
5.5 Test case 5: Verifying the search condition ‘sub-match’ for the input ‘p’ in username field.

Step 1: Entering input ‘p’ in username field.

Figure 46 shows the search form with input ‘p’ in the username field. This shows all the results starting with username ‘p’.

![Figure 46: Search form with input ‘p’](image-url)
Step 2: Database displaying username starting with ‘p’

As the search is done on the encrypted column, the search_table is given control to search for username starting with ‘p’ as highlighted in figure 47. After it matches the data in the search_table, the corresponding key is decrypted and again matched in the actual_table to decrypt the original results.

Figure 47: Search_table showing username starting with ‘p’
Step 3: Showing results for username starting with ‘p’ in the search results page.

Figure 48 shows the search results for input ‘p’ in the search results page which are retrieved from the actual_table via the search_table. All the usernames starting with ‘p’ are retrieved. Thus, achieving the sub-match query search as well.

Figure 48: Search results for ‘p’ on search results page
6. CONCLUSION AND FUTURE WORK

A dynamic online banking web application is successfully developed using AES 128-bit algorithm and secured search technique. Security and privacy of data is achieved along with secured searches over the encrypted data. As the servers are loaded with encrypted data in the untrusted infrastructure, the implemented techniques achieve provable secrecy which means the untrusted server cannot learn anything about the plaintext while retrieving the users requested file. From the results provided in the testing and evaluation section, it is clear that each functionality is tested.

There is a lot of scope for future enhancements in this project. This project encrypts character strings and numbers. The research and implementation can be extended to image files also with encryption technique. It is complex to encrypt and search for encrypted data on extra-large files. So, a faster scheme can be developed to cut down the search time required to search on extremely large dataset which include limitations. And also, the idea of image format encryption can be developed in future along with text and numbers. At present, the search pattern searches for a word considering its starting letters. For example, if the user wants to search for ‘tom’, the database searches for all the words starting with ‘tom’ and also displays if other names which has ‘tom’ in the beginning. But, another search can be implemented in the future which can search for ‘in-between’ words as well. For example, if a search runs for the word ‘temporary’, the search results should retrieve all words even if the input is ‘rar’.
BIBLIOGRAPHY AND REFERENCES


APPENDIX
//Step 1: Check the searching column

String sql = "";
int flag = 0;
boolean isEncrSearch = false;
boolean isAnd = false;
String whereClause = " where";
String condStr = ""

if (searchModel.getAccountNo() != null
   && searchModel.getAccountNo() != "") || (searchModel.getUsername() != null
   && searchModel.getUsername() != "")
{
    //Retrieval of data from search table
    sql = "select AES_DECRYPT(accountno,'"
        + key
        + ")", AES_DECRYPT(username,'"
        + key
        + ")", AES_DECRYPT(amount,'"
        + key
        + ")", firstname, lastname, phone, email, address from actual_table";

    isEncrSearch = true;

    condStr += " secretkey in (select AES_DECRYPT(abc,'"+key")') from search_table";
    if (searchModel.getAccountNo() != null
        && searchModel.getAccountNo() != "") |
        flag = 1;
    isAnd = true;
    condStr += whereClause;
    condStr += " account like " + searchModel.getAccountNo() + ";"
    if (searchModel.getUsername() != null
        && searchModel.getUsername() != "") |
        flag = 1;

    if (isAnd) {
        condStr += " and username = " + searchModel.getUsername()
            + ";"
    } else {
        condStr += " username = " + searchModel.getUsername()
            + ";"
    }
    isAnd = true;
}

else/
flag = 0;
isAnd = false;
if (searchModel.getUsername() != null
   && searchModel.getUsername() != ")
{
    condStr += whereClause;
    flag = 1;

    if (isAnd) {
        condStr += " and username = " + searchModel.getUsername()
            + ";"
    } else {
        condStr += " username = " + searchModel.getUsername()
            + ";"
    }
    isAnd = true;
}
}
```java
// Retrieval of data from actual table
sql = "select firstname, lastname, phone, email, address from actual_table";
if (isSearchModel != null) {
    if (isSearchModel.getFirstName() != null) {
        condStr += " firstname like " + searchModel.getFirstName() + "\"; 
    
    if (isSearchModel.getLastName() != null) {
        condStr += " lastname like " + searchModel.getLastName() + "\"; 
    
    if (isSearchModel.getEmail() != null) {
        condStr += " email like " + searchModel.getEmail() + "\"; 
    
    if (isSearchModel.getAddress() != null) {
        condStr += " address like " + searchModel.getAddress() + "\"; 
    }
}
 głównie = true;
}

public static List<AccountDetails> getAccountList() {
    Connection conn = null;
    Statement stmt = null;
    String key = GetAESKey.getKey();
    int i = 0;
    try {
        conn = new GetDatabaseAccess().openConnection();
        stmt = conn.createStatement();
        String sql = "select AES_DECRYPT(username,\" + key + ") from account";
        List<AccountDetails> accountDetailsList = new ArrayList<AccountDetails>();
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            AccountDetails accountDetails = new AccountDetails();
            accountDetails.setAccountNo(rs.getString(2));
            accountDetails.setUsername(rs.getString(1));
            accountDetailsList.add(i, accountDetails);
            i++;
        }
        return accountDetailsList;
    } catch (SQLException se) {
        // Handle errors for JDBC
        se.printStackTrace();
        if there is an error then rollback the changes.
        System.out.println("Rolling back data here...\n");
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Error: " + e.getMessage());
    }
```
public static List<UserDetails> getUserDetailsList() {
    Connection conn = null;
    Statement stmt = null;
    String key = GetAESKey.getAESKey();
    try {
        conn = new GetDatabaseAccess().openConnection();
        stmt = conn.createStatement();

        String sql = "select AES_DECRYPT(username," + key + "," + "
                     _AES_DECRYPT(firstname," + key + "," + "
                     _AES_DECRYPT(lastname," + key + "," + "
                     _AES_DECRYPT(phone," + key + "," + "
                     _AES_DECRYPT(email," + key + "," + "
                     _AES_DECRYPT(address," + key + "," + "
                     _AES_DECRYPT(gender," + key + ") from userdetails";

        System.out.println(sql);
        ResultSet rs = stmt.executeQuery(sql);

        List<UserDetails> userDetailsList = new ArrayList<UserDetails>();
        while (rs.next()) {
            UserDetails userDetails = new UserDetails();
            userDetails.setUsername(rs.getString(1));
            userDetails.setFirstName(rs.getString(2));
            userDetails.setLastName(rs.getString(3));
            userDetails.setPhone(rs.getString(4));
            userDetails.setEmail(rs.getString(5));
            userDetails.setAddress(rs.getString(6));
            userDetails.setGender(rs.getString(7));
            userDetailsList.add(userDetails);
        }

        return userDetailsList;
    }
    catch (SQLException e) {
        // TODO: handle exception
        e.printStackTrace();
        System.out.println("SQL connection error: " + e.getMessage());
        return null;
    }
    catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
        System.out.println("Error: " + e.getMessage());
        return null;
    }
    finally {
        try {
            // stmt.close();
            conn.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}