Security Scheme Implementation in Mobile Information-Centric Network

GRADUATE PROJECT REPORT

Submitted to the Faculty of the Computer Science Department
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

Pratyusha Katepally
Spring 2015

Committee Members

Dr. Mario Garcia
Committee Chair person

Dr. Dulal Kar
Committee Member
ABSTRACT

In the modern era, one can never imagine the world without Mobiles and especially without internet service in them. Mobile internet usage is increasing rapidly. A user has many applications to download in the App stores. Using these mobile applications, users can easily get access and gather data or information from the web servers in Mobile Information Centric Networks (ICN). In ICN’s, the complete focus is on the content which is of the user’s interest and the users need not know from where the content is received or who carries the information. Security plays a major role in such scenarios. The producer of the content should be authenticated and also the content data should also be encrypted using some highly secure algorithms.

A security scheme in the Mobile Information-Centric networks for secure data transfer is proposed which uses the concept of self-signed certificates created using open SSL toolkit. In client-server architecture, when the clients send a request to the server, the server sends a self-signed certificate which has a public key associated with it. The client obtains and verifies the public key from the self-signed certificate of the server and establishes a secure connection. Then the client tries to obtain the content in the server’s database using AES (Advanced encryption standard technique), thus achieving a secure data transfer between the client and the server.
TABLE OF CONTENTS

ABSTRACT ................................................................................................................................... ii

TABLE OF CONTENTS ........................................................................................................... iii

LIST OF FIGURES ................................................................................................................. v

1 BACKGROUND AND RATIONALE ..................................................................................... 1

1.1 INTRODUCTION .............................................................................................................. 1

1.2 SECURITY CONSIDERATIONS ....................................................................................... 2

1.3 SYMMETRIC KEY ENCRYPTION .................................................................................. 2

1.3.1 Advantages of symmetric key encryption ............................................................... 3

1.3.2 Disadvantages of symmetric key encryption ......................................................... 3

1.4 SECURE SOCKETS LAYER (SSL) ............................................................................... 4

1.5 EXISTING SYSTEM ....................................................................................................... 4

1.5.1 Related work .......................................................................................................... 5

1.6 PROPOSED SOLUTION ............................................................................................... 6

1.7 MOBILE APPLICATION ............................................................................................... 7

2 NARRATIVE ......................................................................................................................... 9

2.1 PROBLEM STATEMENT ............................................................................................... 9

2.2 MOTIVATION .............................................................................................................. 9

2.3 PROJECT SCOPE ....................................................................................................... 10

3 SYSTEM DESIGN ............................................................................................................ 11

3.1 SYSTEM ARCHITECTURE ......................................................................................... 11

3.2 DATA FLOW DIAGRAM ............................................................................................. 12
3.3 USE CASE DIAGRAM .......................................................... 13
3.4 SEQUENCE DIAGRAM ...................................................... 14
3.5 CLASS DIAGRAM ............................................................. 15
4 SYSTEM IMPLEMENTATION ................................................ 17
5 SOFTWARE REQUIREMENTS .................................................. 24
  5.1 ECLIPSE LUNA ................................................................. 24
  5.2 H2 DATABASE ................................................................. 24
  5.3 TOMCAT SERVER ......................................................... 24
  5.4 ANDROID STUDIO ......................................................... 25
  5.5 OPEN SSL TOOL ............................................................. 25
6 TESTING AND EVALUATION .................................................. 26
  6.1 Test Cases ................................................................. 27
     6.1.1 Test case 1: When the searched keyword is sensitive data .......... 27
     6.1.2 Test case 2: When the searched keyword is numeric data .......... 28
     6.1.3 Test case 3: When the searched keyword is alphanumeric data ....... 30
7 CONCLUSION ................................................................. 33
8 FUTURE WORK ............................................................... 35
9 BIBLIOGRAPHY .............................................................. 36
LIST OF FIGURES

Figure 3-1 System design..................................................................................................... 11
Figure 3-2 Data flow diagram.............................................................................................. 13
Figure 3-3 Use case diagram.............................................................................................. 14
Figure 3-4 Sequence diagram ............................................................................................. 15
Figure 3-5 Class diagram of proposed scheme ................................................................. 16
Figure 4-1 Login and registration. ....................................................................................... 17
Figure 4-2 Validations in login page ................................................................................... 18
Figure 4-3 Email and mobile number validations in registration page ............................... 19
Figure 4-4 Password validations in registration page .......................................................... 20
Figure 4-5 Searching a keyword .......................................................................................... 20
Figure 4-6 Information in encrypted format. ....................................................................... 21
Figure 4-7 Information in decrypted format ......................................................................... 22
Figure 6-1 When user tries to retrieve plain text ................................................................. 27
Figure 6-2 When user tries to retrieve actual data ............................................................... 28
Figure 6-3 When user tries to retrieve numeric data ............................................................ 29
Figure 6-4 When user tries to retrieve actual numeric data ............................................... 30
Figure 6-5 When user tries to retrieve alphanumeric data ............................................... 31
Figure 6-6 When user tries to retrieve actual alphanumeric data ...................................... 32
1 BACKGROUND AND RATIONALE

1.1 INTRODUCTION

Mobile network usage is increasing rapidly. Using the mobile network, users can easily retrieve the required information from the webservers. Now-a-days, there is an increasing attention towards Information-Centric network (ICN) in private and public sectors. The main objective of ICN design is to search and retrieve the information from the internet. Mobile internet users focus only on the information required. They need not know where the content is actually stored and how the transmission process is done.

Users can search for a particular keyword through the network. When the matching content is found in the server database, the information is sent back to the client in the reverse path of request. In order to secure the data, the transmitted information will be encrypted using a suitable encryption technique before being sent to the client. This entire process follows a request reply mode communication. Public key encryption is known to comparatively secure. However, Public key encryption has some pros and cons when compared with other encryption techniques available today.
1.2 SECURITY CONSIDERATIONS

When the data has to be transmitted over a network, security plays a vital role. For this, the data being sent has to be encrypted in order to make sure that the data is secure. The basic types of encryptions are as follows:

- Symmetric encryption
- Asymmetric encryption

Previously, symmetric encryption was the only method available. Now people have asymmetric encryption also. It is not that asymmetric encryption has replaced symmetric encryption in all the cases. There are some cases where symmetric encryption is required and vice versa. Symmetric key encryption is a method of disordering the data by using the same key both to encrypt and to decrypt. This key is called as private key or a secret key. The usage of symmetric encryption is when a resting data needs to be secured.

In contrast, asymmetric key encryption uses two different keys which are mathematically linked together. The receiver in the communication creates two keys, a private and a public key. Public key is given to the public either through the website or through other media and is made accessible to everyone including the hacker. Symmetric key encryption is relatively faster than asymmetric key encryption whereas the latter is extremely secure.

1.3 SYMMETRIC KEY ENCRYPTION

Symmetric key encryption is a method of reordering the data using same key both to encrypt and to decrypt. This type of encryption is also known as secret key encryption. This is entirely different from asymmetric key encryption where two different keys which are mathematically linked together are used. There is a public key which is made available for
everyone on the network and a private key which is kept confidential. Public key is used as encryption key and private key is used as decryption key [7].

1.3.1 Advantages of symmetric key encryption

- Data is extremely secure

When it comes to security considerations, secure key algorithm is proven to be the best. One of the most popular and widely used symmetric key encryption standards is AES (Advanced encryption standard). It is designated by the US Government Security. When a 256 bit key length AES is used, it is proven to be the most secure encryption standard and it would take a billion years for an attacker to decode the secret key [10].

- Relatively fast

Symmetric key encryption or secret key encryption is relatively fast when compared with Asymmetric key encryption.

1.3.2 Disadvantages of symmetric key encryption

- Key sharing

One of the biggest problems in symmetric key encryption in two way communication scenario is finding out the safest way to share the key with the party with whom the data is shared. The secret keys are not simple texts. They are randomly generated. In contrast, in asymmetric key encryption, the private key is privately owned key. And hence, the sharing problem is eliminated.
1.4 SECURE SOCKETS LAYER (SSL)

Secure sockets layer is a security protocol used to secure data between a webserver and a browser using encryption. Millions of websites uses SSL to secure connections and keep their customers data safe. The websites which are secured by SSL display a padlock and also a green address bar possibly. Digital certificates which are issued by authorized Certificate authority (CA) like digicert, comodo SSL, godaddy etc. are to be installed on the web page to establish secured connection. These certificates include information such as company name, address, city, state, country, domain name of the organization which requests the digital certificates. The digital certificates also contain details like issued date, expiry date and also the details of the certificate authority who issues the certificate. When the browser requests a https connection to a website, the website’s certificate will be retrieved and checked whether it has not expired and also whether the certificate is being used by the website for which it has been issued. If any of these checks failed, then a warning is displayed to the user.

1.5 EXISTING SYSTEM

In the case of web browsers, the security is preserved to some extent using SSL (secure socket layer) concept, issuing digital certificates by a valid certificate authority. But, when it comes to mobile applications there are lots of vulnerabilities in SSL [1]. Now-a-days, mobile internet usage is increasing rapidly. There are many online banking transactions that occur on a daily basis. Security plays a vital role in such scenarios. But there are a lot of
SSL vulnerabilities in android and iOS ecosystems, leaving behind a challenge task to secure our personal information.

1.5.1 Related work

The main drawback of existing systems is vulnerabilities in security. At present there are existing systems which provide data security to some extent, but they cannot give solutions to common problems in security systems, like speed, cost, flexibility etc.

"The implementation of FPGA-based RSA public-key algorithm and its application in mobile-phone SMS encryption system" by Qun Ding is the recent work done on mobile security [11]. The author made a contribution to improve the present RSA algorithm. He proposed a security scheme consisting of the following aspects:

- **Using FPGA to implement RSA algorithm**

- **Transmitting short messages in mobile phone securely**

The major limitation in this proposed model is that it consumes more time to convert the plain text into cipher text using RSA algorithm because of the mathematical calculations and there is also a limitation for the message length in this proposed model. In addition, FPGA needs complex hardware making it difficult to deploy in mobile environment.

Another model “Pseudo-identity based encryption and its application in mobile ad hoc networks” proposed by Prakash Veeraraghavan is for mobile ad hoc networks [12].

According to the author, in the absence of any centralized infrastructure, usage of Public key infrastructure (PKI) is impossible. The author came up with a concept of threshold and identity based security schemes in which the symmetric key or secret key is shared among
n shareholders. The main drawback of the proposed scheme is usage of single symmetric key encryption which makes the process of encryption and decryption fast, but to gather at least $k$ divided secret keys from $k$ nodes take more time.

“An application of mobile phone encryption based on Fibonacci structure of chaos” is proposed by Fenmei Wang and Jing Ding [13]. In this proposed system, a new technique called chaotic encryption is used to authenticate the user in mobile applications. This consumes more CPU time and resources and there are some issues to be solved based on the real-time encryption systems.

An application-based encryption system, “A New Tool for Lightweight Encryption on Android” is proposed by Sushma Verma, Saibal Kumar pal, S.K Mutto [3]. This approach follows Hummingbird2 technique and secures the data on an android device leaving the user to choose which file to be encrypted on his/her phone. This approach is not proven to be suitable for mobile phones in client-server architecture. Where the security scheme proposed here works well in client and server architecture.

1.6 PROPOSED SOLUTION

To overcome these drawbacks, a social network based security scheme in mobile information-centric network is implemented based on self-signed certificates. The proposed scheme allows user to design a mobile application avoiding the security vulnerabilities [4]. In this application, a self-signed certificate is created by the webserver. When the user logs into the application, a request is made to the server to send its certificate which has lot of information including public key. The client receives the certificate verifies and gets the public key from the certificate.
The client generates a symmetric key using AES (advanced encryption standard) encrypts it using public key and sent over the channel to the server. The server decrypts the encrypted symmetric key with its associated private key. Then the server sends an acknowledgement back to the client confirming a secure connection. The client then sends a keyword to the server encrypting with symmetric key.

The server decrypts it using the same symmetric key. The matching keyword is searched by the server in its database. Once found, the content is sent to the client encrypting with the symmetric key. The client receives the encrypted content and decrypts the content using symmetric key and displays.

1.7 MOBILE APPLICATION

Mobile application software is a computer program which is designed to run on smartphones. They have emerged in the past and are a hot trend in the market now. The most common mobile operating systems are: Android from Google, iOS from Apple, Blackberry and Windows Phone from Microsoft. Android is free and open source mobile platform based on Linux kernel for making apps which is currently developed by Google.

The user-interface is based on human-computer interaction [8]. Before running the mobile applications on a real-time environment, they are first tested in the integrated development environments using emulators. Emulators provide an inexpensive way of testing the mobile applications.

In the proposed security scheme, the data is fetched from the database via a communication channel in an encrypted format [9]. User obtains the data in the encrypted
format and he/she has an option to decrypt the data. Thus, ensuring the data which is traversed through the encrypted communication channel is secure.
2 NARRATIVE

2.1 PROBLEM STATEMENT

There are many applications that provide security for shared content in different IP based computer networks. But in mobile networks, only limited security is available for content sharing. Accessing content data in mobile interface is quite interesting because there are many variations in content accessing in computer and mobile devices. This needs to make many changes in the interface of an application for using in the mobile device. Accordingly, content accessing in the application from the internet needs to be secured.

For example, when users log into their mobile banking app, usually they do not think about security or data privacy, being in an illusion that the company has placed a highly secure app in the app store [9]. Some apps include payment gateways through which user can flexibly buy movie tickets etc. But in reality, there are lots of vulnerabilities in the mobile apps even though it is issued by one of the G2000 companies. The common attacks in mobile applications are man in the middle attacks (MITM) and brute force attacks.

2.2 MOTIVATION

Mobile internet usage is increasing rapidly with time. Security standards in mobile applications play a vital role in this scenario. Security vulnerabilities in android and iOS systems tend to brute force attacks and man-in-the-middle attacks. This leads to exposure of user’s banking information and personal information such as usernames, passwords, social security numbers etc.

According to Cloud security alliance blog, “Fire Eye also recently published data that reported security flaws in the most commonly downloaded Android apps and found that a
significant number of the apps are susceptible to MITM attacks. Fire Eye reported that as of July 2014, out of the 1,000 most downloaded apps in the Google Play store, 73% of the apps that use encryption standards to communicate with a remote server do not verify security consideration completely. Of the 10,000 random apps in the Google Play store, 40% do not check server certificates, exposing data they exchange with their servers to potential theft [9].”

2.3 PROJECT SCOPE

This software product is designed for phone and tablet users, which is developed for Android mobile operating system. The users of the product can use the services of the secured data content when they are connected to the internet as the data exchange is from the H2 database (the java SQL light weight database) through tomcat server. The server side application is developed in java programming language. The client side application is developed in java and xml is used for styling. The app is designed to retrieve the content in the database via encrypted communication channel. The data which is searched can be plain text, numeric or alphanumeric data.

Examples:

Plain text: 'A student is a learner or someone who attends an educational institution'

Numeric: '1939833173635'

Alphanumeric: ‘Masters@2014’
3 SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

Figure 3-1 illustrates the system design of the proposed security scheme.

![Diagram](image)

**Figure 3-1 System design**

A social network based security scheme is implemented in a mobile information centric network. An android emulator is used to demonstrate the proposed scheme.

This project includes a Restful (Representational State Transfer) web service. The proposed system uses the concept of self-signed certificate and AES encryption algorithm. When the
user logs into the application, it sends a request to server for sending it self-signed certificate with a public key associated with it.

The client fetches the public key from the self-signed certificate. The client generates a symmetric key using AES algorithm and encrypts it using the public key before sending to the server. The server decrypts the symmetric key using the associated private key. Then both the client and the server have the same AES symmetric key. The server acknowledges the client an acknowledgment confirming the secured connection between both the client and the server. The client sends the server, the keyword entered by the user encrypted by the symmetric key.

The server receives the request, decrypt the keyword using the same symmetric key. The server searches for the matching keyword in its database (H2). Once the content is retrieved, it is encrypted using symmetric key before sending to the client. The client receives the encrypted data and decrypts the matching content using the same symmetric key. Then the decrypted content is displayed in the client.

Thus the proposed scheme achieves adequate security preventing the unauthorized users to read or tamper the data. Thereby enhancing the security levels of the personal, health and banking information the user enters in any of his/her mobile applications.

3.2 DATA FLOW DIAGRAM

Figure 3-2 illustrates the data flow diagram of the proposed scheme
The user first starts the application in android emulator. He/she then logs in (if registered). The credentials are verified in the local database SQL lite. There is an option to search the keyword of user’s choice. The matching keyword and its corresponding data are retrieved from the server database. Then the security of the content is verified by seeing the encrypted data. Apply the symmetric key and decrypt the data. View secured content. And once he/she is done with searching they log out of the application.

3.3 USE CASE DIAGRAM

Figure 3-3 illustrates the use case diagram of the proposed scheme
In the use case diagram shown in Figure 3-3, all the use cases represent the functionalities of the user and the database. User has to register and login the android application. If a user is a new user, the login details will be updated in the SQL lite database. If the user is a registered user, he/she will be authenticated by checking the login details in the database. When a user searches for a keyword, the server interacts with the database and searches for a specific keyword.

3.4 SEQUENCE DIAGRAM

Figure 3-4 illustrates the sequence diagram of the proposed scheme.
The sequence diagram shown in Figure 3-4 shows the interaction between all the processes in implementing the social network based security scheme in Mobile Information Centric Networks. When a user logs into the application, a request is sent in order to verify the authentication. Once login is successful, the user can search for the required content from the database via services provided by the application.

### 3.5 CLASS DIAGRAM

Figure 3-5 illustrates the class diagram of the proposed scheme
Figure 3-5 Class diagram of proposed scheme
4 SYSTEM IMPLEMENTATION

The mobile app for security scheme implementation in mobile information centric network is implemented in Android using Android Studio, the web service was implemented in Eclipse Luna and the user data is stored in a H2 database in the backend. Here are some of the implementation details. This software product is designed for phone and tablet users, which is developed for Android mobile operating system. The users of the product can use the services of the secured data content when they are connected to the internet as the data exchange is from the H2 database (the java SQL light weight database) through a tomcat server. Figure 4-1 illustrates the login and registration pages of the mobile app.

![Figure 4-1 Login and registration](image_url)
The login and registration modules are implemented with validations. When the user tries to hit the login button without entering username and password, there is a toast message which says “Please enter username or password”. When the user tries to log into the application with wrong username and password, there is a toast messages which says “The user name and password you entered is incorrect”.

Figure 4-2 illustrates the validations in login page.

![Figure 4-2 Validations in login page](image)

When the user wants to register before he logs into the application, he/she has to enter certain information. Client side there are some validations made to register into the application.

Figure 4-3 illustrates email and mobile number validations in registration page.
Figure 4-3  Email and mobile number validations in registration page

Figure 4-3 illustrates email and mobile number validations in registration page.

The application is developed in java programming language. The app is designed to retrieve the content in the database via encrypted communication channel. The data which is searched can be plain text, numeric or alphanumeric data. The Android platform abstracts many of the lower level implementation details that are involved in mobile communication that occurs internally or externally. The application is designed with a simple user interface with the purpose of providing great usability to the users.
Figure 4-4 Password validations in registration page

Figure 4-5 illustrates the page to search for a keyword in the mobile application.

Figure 4-5 Searching a keyword
For the first time when a user downloads the application, the user has to register with the application using his identification information (username, password, email and mobile number). Once the user is registered then he can login to the application and start fetching the secured content from the database in an encrypted format. If the keyword matches with the database content, then the information is retrieved back.

In the proposed security scheme implementation, the plain text in the content is encrypted using secret key encryption technique i.e., Advanced encryption standard.

Figure 4-6 illustrates the page where the user gets the data in encrypted format.
The obtained information needs to be decrypted in order to view the sensitive information.

The plain text can be decrypted using symmetric key generated at the time of encryption. In real time, user will be entering his/her personal information in the mobile applications. The information may include username, password (alphanumeric), plain text (sensitive information), and credit card (numeric data) information.

Figure 4-7 illustrates the page where the user decrypts the encrypted data.

When the user types the keyword of required information, he/she gets back the data from the database in an encrypted format. This confirms that the channel of communication
between server and client is encrypted. Once the user gets the encrypted information, 
he/she has an option in the mobile application to decrypt the data by hitting the ‘decrypt’ 
button.
5 SOFTWARE REQUIREMENTS

The following software components are required to implement the project.

5.1 ECLIPSE LUNA

Eclipse is an integrated development environment (IDE) developed by Eclipse foundation and is mostly written in Java programming language. It works on operating systems like Linux, Mac OS Windows and Solaris. It is platform independent. It is available in the market in multiple languages. It is a programming tool which enables us to develop programming projects.

5.2 H2 DATABASE

H2 Database is the java sql light weight database. It is very fast and open source application. It has JDBC (Java database connection) API (application program interface). It has browser base console application.

5.3 TOMCAT SERVER

It is an open source web server. It is developed by Apache software foundation (ASF). It is platform independent Web applications are the Web sites which usually consist of several modules such as JSP, servlets, Web pages. The objective of the tomcat is to organize all these multiple modules into a single directory for every Web application. For creating the servlet tomcat need two different directories the first one is called as the deployment directory in which live code is kept and the second one is the development.
5.4 ANDROID STUDIO

Android Studio is an Integrated Development Environment (IDE) used for developing apps on the android platform. The new SDK packages and add-on API has made the IDE more interactive and easy to use. The material design helps to build the UI attractive. Using android emulator, the development of android application is made easy.

5.5 OPEN SSL TOOL

Open SSL is an open source tool kit for implementing SSL/TLS protocol. Using Open SSL self-signed certificates can be generated. Open SSL supports a number of cryptographic algorithms like RSA, DSA etc. It is used to create self-signed certificates, private key etc.
6 TESTING AND EVALUATION

Testing is a key to success. In software development cycle testing plays a major role testing helps to find the errors in the products which are not known in the initial stage of development. In testing phase a set of inputs is passed to the system and the output is evaluated to check whether the system is meeting its requirement’s as expected.

Testing helps the developer to find the loopholes in the application before it is fully completed. The goal of testing is to find errors in the applications or the programs and report them to the user, such that the user can improve the quality of the product by readdressing all the issues.

The main purpose of the proposed schema is to secure the content of the user. There are lot of online banking applications and online shopping applications available in the mobile app stores. When users log in to their mobile banking app, usually they don’t think about security or data privacy, being in an illusion that the company has placed a highly secure app in the app store. Some apps include payment gateways through which user can flexibly buy movie tickets etc. But in reality, there are lots of vulnerabilities in the mobile apps which are the main reasons for the personal information to be hacked.

The types of information user usually enters may be plain sensitive data, credit card information, username and password while logging in. These sensitive data needs to be secured with high security levels.
6.1 Test Cases

6.1.1 Test case 1: When the searched keyword is sensitive data

When the user tries to fetch sensitive data in the mobile application, the text should be sent via an encrypted channel. This ensured that the data is secured while traversing from database to the client via a webserver.

Figure 6-1 illustrates the testing case when plain text is retrieved in encrypted format.

![Figure 6-1 When user tries to retrieve plain text](image-url)
When the user tries to retrieve the plain text, he/she searches for a particular keyword by entering the keyword and hitting the search button. The user will then be able to see the encrypted format of the content which matches the keyword. If the user wants to see the actual data he/she needs to hit the decrypt button. Then the user will be able to view the information he/she actually needed.

Figure 6-2 illustrates the testing case when actual data is retrieved.

![Figure 6-2](image)

**Figure 6-2 When user tries to retrieve actual data**

### 6.1.2 Test case 2: When the searched keyword is numeric data

When the user tries to fetch numeric data in the mobile application, the data should be sent via an encrypted channel. This ensures that the data is secured while traversing from
database to the client via a webserver. The numeric data can be any credit card information or balance money in the bank account.

Figure 6-3 illustrates the testing case when numeric data is retrieved in encrypted format.

![Figure 6-3 When user tries to retrieve numeric data](image)

When the user tries to retrieve the numeric data, he/she searches for a particular keyword. By entering the keyword and hitting search button. The user will then be able to see the encrypted format of the content which matches the keyword. If the user wants to see the actual data he/she needs to hit the decrypt button. Then the user will be able to view the information he/she actually needed.
Figure 6-4 illustrates the testing case when actual numeric data is retrieved.

![Image of a mobile app interface with a search bar containing 'number' and an encrypted text field with an encrypted string. Below the encrypted text field, there's a decrypt button with the decrypted text '1039833173635'.](image)

**Figure 6-4 When user tries to retrieve actual numeric data**

### 6.1.3 Test case 3: When the searched keyword is alphanumeric data

When the user tries to fetch alphanumeric data in the mobile application, the data should be sent via an encrypted channel. This ensured that the data is secured while traversing from database to the client via a webserver. The alphanumeric data can be any user name or password while logging into the application.

Figure 6-5 illustrates the testing case when alphanumeric data is retrieved in encrypted format.
When the user tries to retrieve the alphanumeric data, he/she searches for a particular keyword. By entering the keyword and hitting search button. The user will then be able to see the encrypted format of the content which matches the keyword. If the user wants to see the actual data he/she needs to hit the decrypt button. Then the user will be able to view the information he/she actually needed.

Figure 6-6 illustrates the testing case when actual alphanumeric data is retrieved
Figure 6-6 When user tries to retrieve actual alphanumeric data
Mobile network uses to retrieve information from the service providers. Often, the data is not securely transmitted over the network in mobile applications. The proposed security scheme provides a way to secure the data that is being transferred by using self-signed certificates combining advanced encryption standard (AES) and RSA algorithms. Using advanced encryption standard for encrypting the plain text reduces the time to encrypt the data and using RSA algorithm enhances the security of the system by encrypting the symmetric key which is generated while using the AES algorithm. The proposed system is tested for different types of plain texts which include numeric data, alphanumeric data and character data. The algorithm is able to encrypt and decrypt all the types of data which are mentioned. The following conclusions can be made after testing the application for various inputs.

1) When a character data is fetched through the application, user gets the encrypted text which can be decrypted by the private key he/she have.

2) In most of the banking applications, users enter their personal information which includes SSN, credit card information etc. When numeric data is fetched through the application, the user can see the data in an encrypted format, which confirms the security.
3) In any mobile application, when the user enters the authentication information like username or password, he/she enters the alphanumeric data which include numbers, alphabets and special characters. When the alphanumeric data is fetched through the application, the user is able to view only the encrypted alphanumeric data.
8 FUTURE WORK

- The proposed system can be extended to iOS platform

- A client server architecture model is proposed here, where the data is fetched from the server database through the client. It can be extended where data can be uploaded from the client side and the server side gets encrypted data.

- The proposed system can be extended to encrypt images and text files etc.
9 BIBLIOGRAPHY


