

PROFESSIONAL TRAINING REPORT

AT

**Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in
Information Technology

By

**JAYA RASIKA.J
(Reg.No.37120034)**



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119, TAMILNADU

AUGUST 2019



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **JAYA RASIKA.J (Reg.No.37120034)** who underwent the professional training in **"ANDROID APPLICATION DEVELOPMENT"** under our supervision from July 2019 to August 2019.

INTERNAL GUIDE

Dr. R.SUBHASHINI M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Dr. R.SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examine

DECLARATION

I **JAYA RASIKA.J(Reg. No. 37120034)** hereby declare that the Professional Training Report on “**ANDROID APPLICATION DEVELOPMENT**” done by me under the guidance of **Dr.R.SUBHASHINI M.E., Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. SUBHASHINI M.E., Ph.D.,for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

10

LIST OF ABBREVIATION

TABLE OF CONTENTS

1.	INTRODUCTION	11
	1.1 OUTLINE OF PROJECT	11
	1.2 EXISTING SYSTEM	12
	1.3 LIMITATIONS	12
	1.4 PROPOSED SYSTEM	12
	1.5 FEATURES	12
	1.6 HARDWARE AND SOFTWARE SPECIFICATION	13
	1.7 LITERATURE SURVEY	13
2.	TECHNOLOGIC LEARNED	16
	2.1 ENCRYPTION BASED CHANNEL CODING ALGORITHM FOR SECURE SMS	16
	2.1.1 IMPLEMENTATION OF SECURITY MESSAGE PROTOCOL FOR PDA PUSH SERVICE	16
	2.1.2 HIGH SECURITY COMMUNICATION PROTOCOL	17
	2.1.3 EVALUATION ON END TO END SECURITY ARCHITECTURE FOR MOBILE BANKING SYSTEM	17
	2.1.4 SECURE INFORMATION TRANSMISSION SCHEME WITH A SECRET KEY BASED ON POLAR CODING	17

2.2 DATA FLOW DIAGRAM	18
2.3 UML DESIGN	19
2.4 USE CASE DIAGRAM	20
2.5 SEQUENCE DIAGRAM	21
2.5.1 CLASSDIAGRAM	22
2.5.2 COLLABRATION DIAGRAM	23
2.5.3 ACTIVITY DIAGRAM	24
 3. ANALYSIS	 26
3.1 USER PROFILE MODULE	26
3.1.1 SMS COMMUNICATION	26
3.1.2 AUTHENTICATION SERVER	27
3.1.3 SYMMETRIC KEY	27
3.2 HIGH-LEVEL DESCRIPTION OF THE ALGORITHM	28
3.3 MD5 ALGORITHM	28
 4. DESIGN	 30
4.1ANDROID	30
4.2 FEATURES	30
4.2.1 LINUX KERNEL	31
4.2.2 ANDROID RUNTIME	31
4.2.3LIBRARIES	31
4.3 APPLICATION FRAMEWORK	32
4.4 ANATOMY OF AN ANDROID APPLICATION	32

4.5 ARCHITECTURE	33
4.5.1 ANDROID BUILDINGBLOCKS	33
4.6 DEVELOPMENTTOOLS	34
4.7 LIFECYCLE OF ACTIVITY	35
4.7.1 CREATE ANDROID APPLICATION	35
4.8 TESTING	36
4.8.1 WHITE BOXTESTING	36
4.8.2 BLACK BOX TESTING	36
4.8.3 UNIT TESTING	36
4.8.4 FUNCTIONAL TESTING	37
4.8.5 PERFORMANCE TESTING	37
4.8.6 INTEGRATION TESTING	37
4.8.7 OBJECTIVE	37
5. SUMMARY AND CONCLUSION	38
5.1 CONCLUSION	38
5.2 FUTURE WORK	38
APPENDIX	38
SOURCE CODE	38
SCREEN SHORT	42
REFERENCES	44

LIST OF FIGURES

FIG NO	FIGURE NAME	PAGE NO
1.1	Sender side	15
1.2	Receiver side	15
2.1	Level-1	18
2.2	Level-2	19
2.4	Flow chat	21
2.5	Sequence diagram	22
2.5.1	Class diagram	23
2.5.2	Collaboration diagram	24
2.5.3	Activity diagram	25
3.1	User profile model	26
3.1.1	SMS communication	26
3.1.2	Authentication server	27
3.1.3	Symmetric key	27
3.3	MD5 diagram	29
4.5	Architecture	33
4.7	Lifecycle	35
4.7.1	Creating android app	36
6.1	Key for Decryption	42
6.2	Enc SMS	42
6.3	Dec SMS	43
6.4	Chipper SMS	43

ABSTRACT

Now a days this advanced technology has allowed people to communicate in various ways, one of them is using mobile phone device. However, security issues such as authenticity and confidentiality of data or information are still cannot be guaranteed. Messages delivered through SMS can be easily stolen by unauthorized parties because SMS sent via BTS will be accepted by Message Service Center (SMSC), where the operator can view the message contents. There are 2 main features that specify and differentiate between RSA and RC6, they are the Time Consumed for Encryption & Decryption message and the Ability to protect the data from unauthorized parties. This study begins with a comparative analysis of performance and security of the most useful algorithms: RC6 (Rivets Cipher) and RSA (Rivets Shamir and Adelman), then the complexity of encryption and decryption algorithms to obtain better algorithms are discussed. As proof of concept, a prototype for encryption and decryption of SMS was developed based on Android platform. The result of this study shows that all messages can be encrypted using a 16-bit key to generated before being sent to the receiver and also after delivered to the receiver which has an better security and time performance in their mobile phones.

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF PROJECT:

Short Message Service (SMS) has become one of the fastest and strong communication channels to transmit the information across the worldwide. Sometimes, we send the confidential information like password, pass code, banking details and private identity to our friends, family members and service providers through an SMS. SMS messages are transmitted as plaintext between mobile user (MS) and the SMSC (SMSC), using wireless network. SMS contents are stored in the systems of network.

So, the traditional SMS operators and can be read by their personnel. Since, the SMS is sent as plaintext, thus network operators can easily access the content of SMS during the transmission at SMSC service offered by various mobile operators surprisingly does not provide information security of the message being sent over the network. In order to protect such confidential information, it is strongly required to provide end-to-end secure communication between end users. The above requirements can be accomplished by proposing a protocol called Cipher-SMS which provides end-to-end security during the transmission of SMS over the network. The Cipher-SMS protocol achieved by using cryptographic algorithms of AES and MD5, The Cipher-SMS protocol prevents the SMS information from various attacks including SMS disclosure, over the air (OTA) modification, replay attack, man-in-the-middle attack, and impersonation attack.

1.2 EXISTING SYSTEM:

- EasySMS which provides end-to-end secure communication through SMS between end users. EasySMS is executed which makes available the symmetric shared key between both MS and then ciphering of message takes place using a symmetric key algorithm. The working of the protocol is presented by considering two different scenarios are SMS-Sec and PK-SIM protocols.

- SMS-Sec protocol can be used to secure an SMS communication sent by Java's Wireless Messaging API while the PK-SIM protocol

- proposes a standard SIM card with additional PKI functionality.

Both protocols are based on client-server paradigm.

- In EasySMS protocol, a cryptographic encryption algorithm AES/MAES is maintained to provide end-to-end confidentiality to the transmitted SMS in the network.

1.3 LIMITATIONS:

- EasySMS Provide SMS security with symmetric key cryptography, the existing protocol is completely based on symmetric key cryptography.
- The transmission of symmetric key to the mobile users is efficiently managed by the protocol.
- Security loses when hacking key transmission between Mobile.

1.4 PROPOSED SYSTEM:

- Without network the Cipher-SMS provides end-to-end security during the transmission of SMS. The Cipher-SMS protocol achieved by using cryptographic algorithms of AES and MD5.
- The Cipher-SMS protocol prevents the SMS information from various attacks including SMS disclosure, over the air (OTA) modification, replay attack, man-in-the-middle attack, and impersonation attack.
- Proposed SMS based framework provides a low-bandwidth, reliable, efficient and cost-effective solution for SMS Transmission. Cipher-SMS is the first protocol completely based on the symmetric key cryptography of AES and hash cryptography of MD5 for cellular network.
- This Cipher-SMS sends lesser number of transmitted bits, generates less computation overhead, and reduces bandwidth consumption and message exchanged as compare to existing protocols.

1.5 FEATURES:

- This protocol produces group communication and computation overheads, utilizes bandwidth efficiently, and reduces message exchanged during authentication than EasySMS (existing) protocols.

- Here preferred a symmetric key algorithm of AES with MD5 because these algorithms are 1000 times faster than the asymmetric algorithms and improve the efficiency of the system.

- Achieved more security than EasySMS by using AES with MD5.

1.6 HARDWARE AND SOFTWARE SPECIFICATION:

a. Hardware:

- i.** 1 GB RAM
- ii.** 80 GB Hard Disk
- iii.** Intel Processor

b. Software:

- i.** Windows OS
- ii.** JDK 1.7
- iii.** Android Eclipse(ADT Bundle)
- iv.** SQLite Database

c. System Requirement:

Operating System: Windows 7 ultimate 32 bit.

1.7 LITERATURE SURVEY:

[1] TITLE : Design and implementation of encrypted SMS on Android smartphone combining ECDSA - ECDH and AES

AUTHOR : **Muhammad HaikaiAzam ; Dodi Isakson; EdwidMashona Judie**

A Short Message Service (SMS) is a popular mechanism to communicate. However, for an important purpose, a mechanism to protect SMS messages is needed so the SMS messages cannot be easily read by entities who are not permitted to see. Keeping an authentication of messages, and ensuring messages are received by a right entity are also an important aspect. Maintaining confidentiality of messages, by combining symmetric and asymmetric scheme are used. The combination can be implemented by using the Advanced Encryption Standard (AES) and Elliptic Curve Diffie-Hellman (ECDH). Even so, there is a weakness while using ECDH protocol. It does not provide an authentication mechanism between entities. This means an

authenticity and an integrity of messages are not fully protected. This research proposes a solution to maintain it by adding the Elliptic Curve Digital Signature algorithm (ECDSA). ECDSA is used to handle the authentication of key exchange.

[2] TITLE : Encrypted SMS application on Android with combination of Caesar cipher and Vigenary algorithm

AUTHOR : FreiFarrant ; Siti Umami Masrour ; Nolan

The use of SMS application is not a new topic this days, But it is un-denied that the SMS method is the classic way to do the messenger activity but still use by the user in communication activity until now. According to research before with Abdur Rahman author ever read, therein explained that the encryption method used is the Caesar cipher with a number of 95 characters. And there are limitations to input only 50 word text message, authors also wish to test security and performance when combining the two methods of encryption. The author tries to combine the Caesar cipher encryption method and Viginere cipher encryption method. The number of characters 94 and there is no limitation on the input text message. So, in this research the author intents to develop the recent research to improve the previous research to make it better. In the end of the chapter of this thesis, the author will give the conclusion of the study that this application was made to improve the security of the encryption of text messages, and prove that the merger of two classical methods of cryptography can be implemented on Android smartphone.

[3] TITLE : Text Compression and Encryption through Smart Devices for Mobile Communication

AUTHOR : Raffaele Pizzolatto ; Bruno Charpentier ; Aniela; Arcangelo Castiglione ; Francesco Palmieri

Today's life of e-citizens relies more and more on the "always-on" paradigm. This transformed our lives in a way that it is difficult, and sometimes impossible, to deal with day-by-day activities without being 'connected". The technology that enable an individual to take advantage, in the best way, of the Ubiquitous Computing paradigm is the Mobile Computing Communication that is fostered by the wide adoption of smart devices such as mobile phone, smartphones, tablets and so on. In this work we investigate the possibility of reliably sending a small file via Short Message Service (SMS) by using data compression for a more effective mobile data exchange in which basic GSM is the only available data communication option. We present an application for portable devices, called Smart Text Compression and Encryption for Short Message Service (STCESMS)

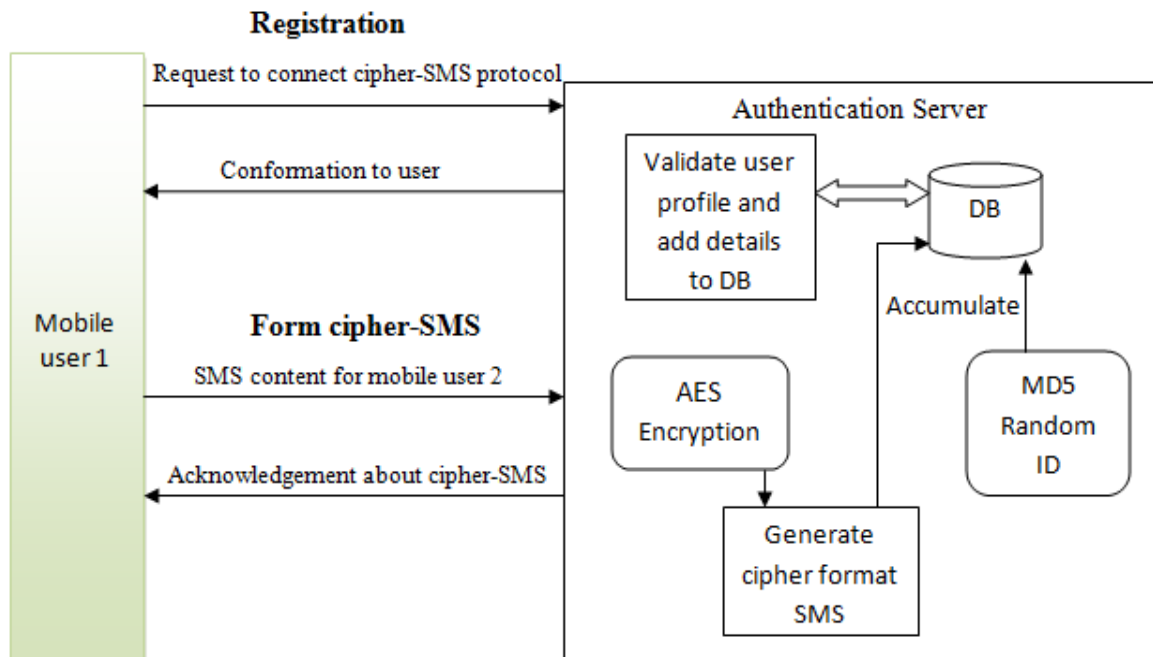


Fig 1.1 Sender side

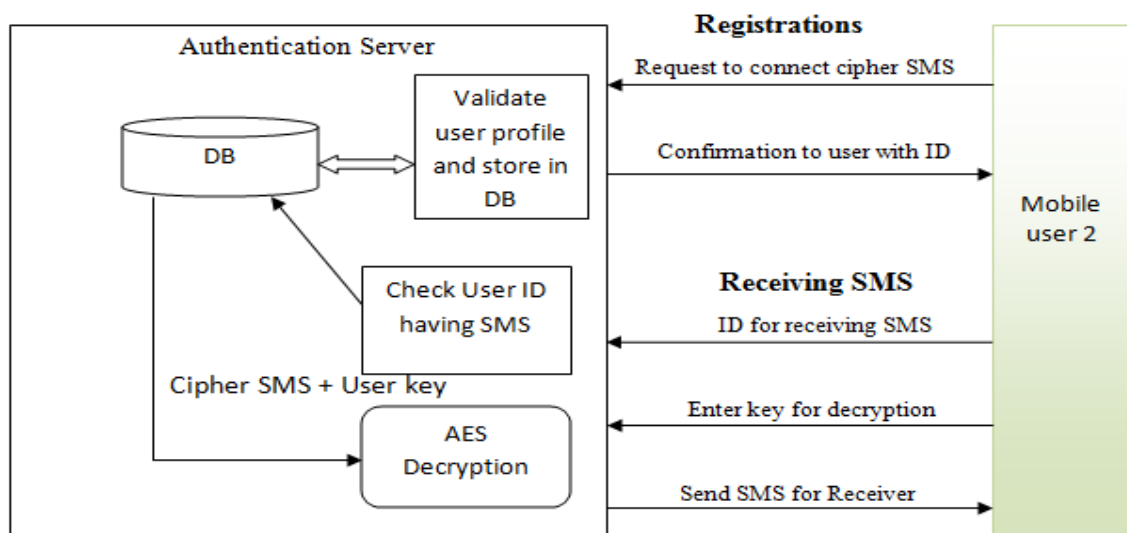


Fig 1.2. Receiver side

CHAPTER 2

TECHNOLOGIES LEARNED

2.1 ENCRYPTION BASED CHANNEL CODING ALGORITHM FOR SECURE SMS:

SMS has a variety of advantages and disadvantages for M-Commerce purpose. The advantages are it is easy to use, a common messaging tool among consumers, works across all wireless operators, affordable for mobile users, no specific software required for installation, allows banks and financial institutions to provide real-time information to consumers and employees and stored messages can be accessed without a network connection. Most important disadvantage of SMS is that it does not offer a secure environment for confidential data during transmission and there is no standard procedure to certify the SMS sender. There is a need for an end to end SMS Encryption with errorless message transmission in order to provide a secure with error free data transmission for communication. These two factors are important for SMS. In this paper, we have analyzed about mainly JCCC and Soft Input Decryption (SID). We proposed a novel theoretically scheme NTRU Sign algorithm in this paper. We expect that it will improve the current security level speed and provide reliable message at receiver end.

2.1.1 The Implementation of Security Message Protocol for PDA PUSH Service:

In this paper, we propose and implement a service model to transfer messages safely for PDA on CDMA wireless networks and a secure message transfer protocol which considers characteristics of PDA. The proposed PUSH service uses SMS (short message service) to connect an offline client device with the wired network for data communication. After receiving SMS message, client device process the SMS message and creates a data channel through RAS (remote access service), and then the data of the server can be pushed to client. The implemented securing protocol can provide safe data transmission on each communication channel through two ways channels of SMS and data. This protocol can reduce a number of transmissions for exchanging a safe session.

2.1.2 High Security Communication Protocol for SMS:

Nowadays, short message service (SMS) is faced with various security threats. Thus, the fields of high confidentiality(e.g., mobile E-commerce) require a higher level of security protection on SMS. Secure communication in incredible mobile network has veryimportant significance. This paper presents a high security communication protocol for SMS. Through authentication, decryption and integrity protection, it establishes an end-to-end secure channel between server-side and mobile terminal. Through analyzed it by so logic, this protocol is proved to ensure confidentiality, integrity and non-repudiation of SMS messages.

2.1.3 Performance evaluation on end-to-end security architecture for mobile banking system:

The advantage of mobile penetration enables mobile operators to provide value added service such as secured mobile banking, mobile commerce and provide enhanced security for internet banking. Mobile banking is attractive because it is a convenient approach to perform banking from anywhere any time, but there are security concerns in the implementation, which include problems with GSM, network, SMS, GPRS protocols. In this paper an end-to-end security framework using PKI for mobile banking is proposed. Performance of the proposed model is presented in this paper.

2.1.4 A Secure Information Transmission Scheme with a Secret Key Based on Polar Coding:

In this letter, a new secure information transmission scheme based on polar codes with a pre-shared secret key is proposed. In polar codes, after the channel polarization is induced, good split channels are used to transmit the user message and bad channels are utilized to support the reconstruction of the message by sharing fixed information. If the fixed information in bad channels is secret, an adversary gets difficulty in reconstructing the user message in good channels without knowledge of the fixed information. From this observation, we construct a secure information transmission scheme. By appending pre-/post-processing that imposes a dependency between the transmitted message sub-blocks.

2.2 DATAFLOW DIAGRAM:

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out to precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results

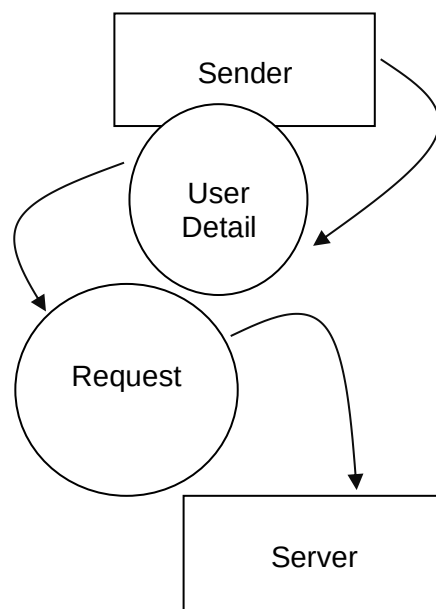


Fig 2.1 Level -1

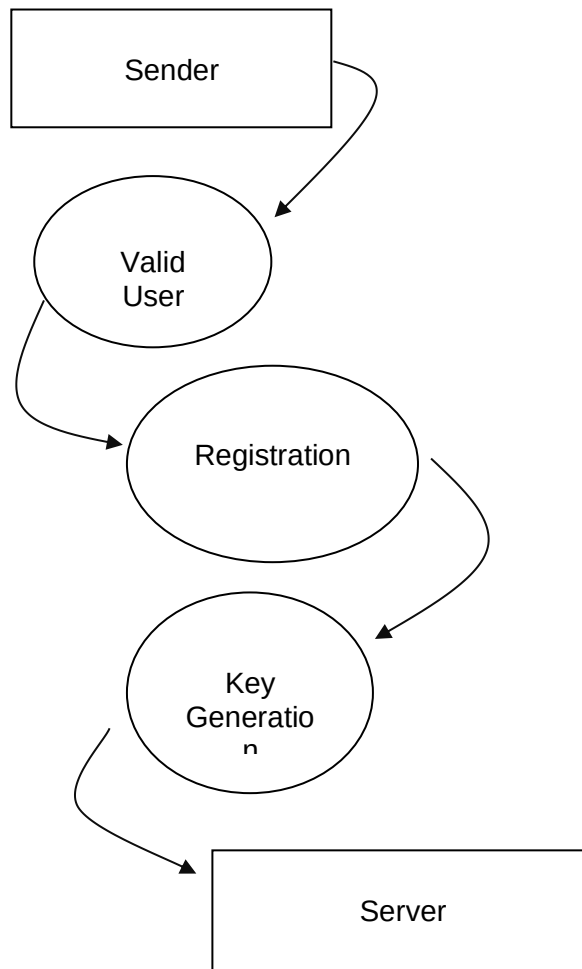


Fig 2.2 Level-2

2.3 UML DESIGN:

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the Object Management Group in January 1997.

Object Management Group is continuously putting effort to make a truly industry standard.

- UML stands for Unified Modeling Language
- UML is different from the other common programming languages like C++, Java, COBOL, etc.
- UML is a pictorial language used to make software blue prints.

UML can be described as a general purpose, visual modeling language to visualize, specify, constructs and document software system. Although UML is generally used to model software systems but it is not limited within this boundary. It is also used to model non software systems as well like process flow in a manufacturing unit etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with objectoriented, analysis and design. After some standardization UML is become an Object Management Group standard.

The most important is to define some general purpose,modeling language which all modelers can use and also it needs to be made simple to understand and use.UML diagrams are not only made for developers but also for business users, common people and anybody interested to understand the system. The system can be a softwareornon-softwareSoitmust be clear that UML is not a development method rather it accompanies with processes to make a successful system.

2.4 USE CASE DIAGRAM:

Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. Use case diagrams are done in an early phase of a software development project. They express how it should be possible to use the final system. Use cases are a good way to express the functional requirements of a software system, they are intuitive and easy to understand so they can be used in negotiations with non-programmers. The participants in a UML use case diagram are use cases, one or several actors and relations, associations and generalizations between them.

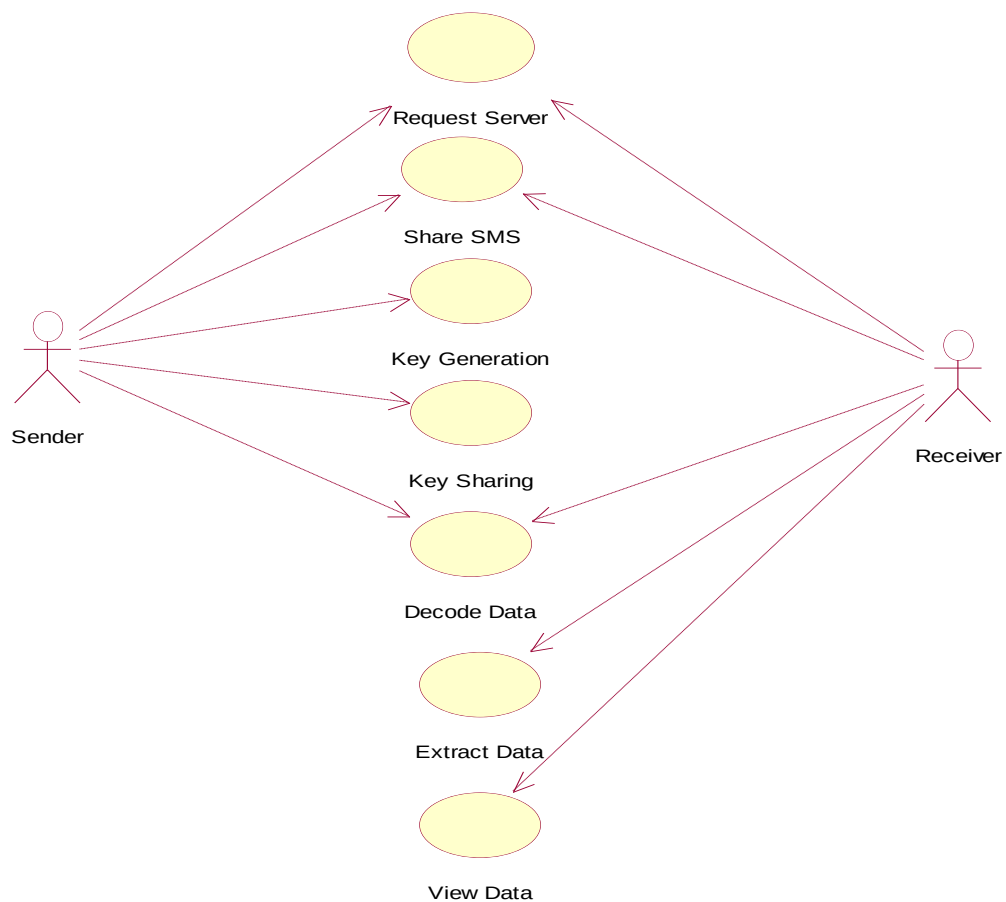


Fig 2.4 Flowchart

2.5 SEQUENCE DIAGRAM :

IT shows the interaction between a set of objects, through the messages that may be dispatched between them. The diagrams consists of interacting objects and actors, with messages in-between them it is common to focus the model on scenarios specified by use-cases. It is also often useful input to the detailed class diagram to try to model the specified

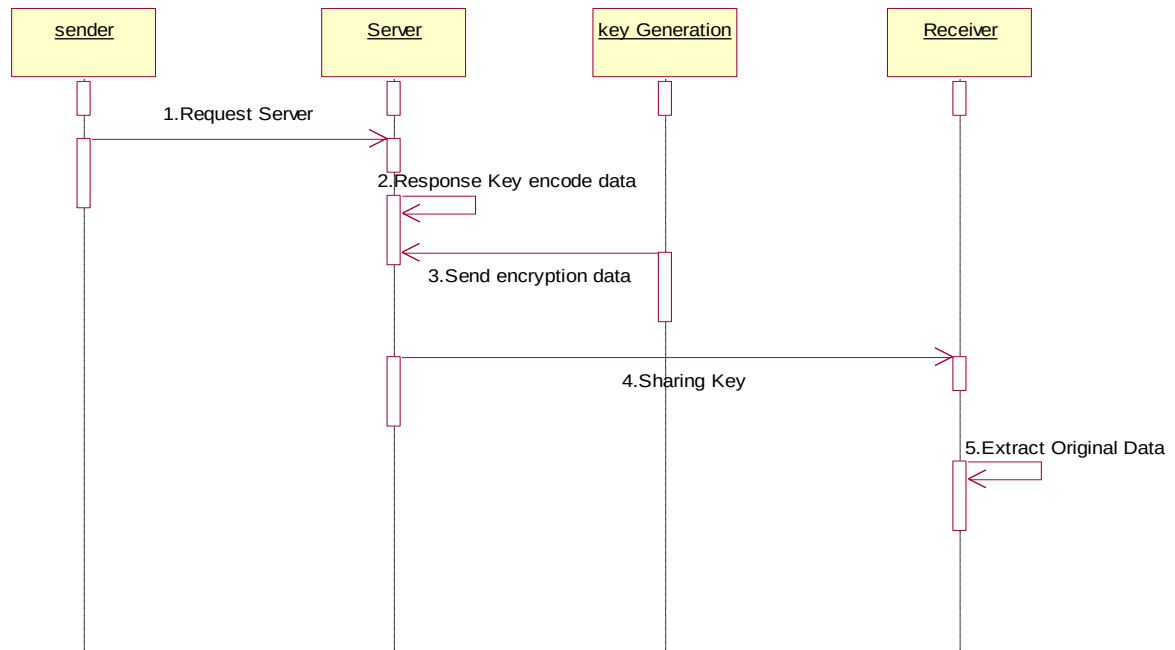


Fig 2.5 Sequence diagram

2.5.1 CLASS DIAGRAM:

It express the static structures of a system divided into different parts called classes and also which relations the classes have to each others.

Class diagrams can profitably be used both in the early phases of a project and during detailed design activities. A class diagram consists of the parts classes, associations and generalizations, and can exist in several different levels.

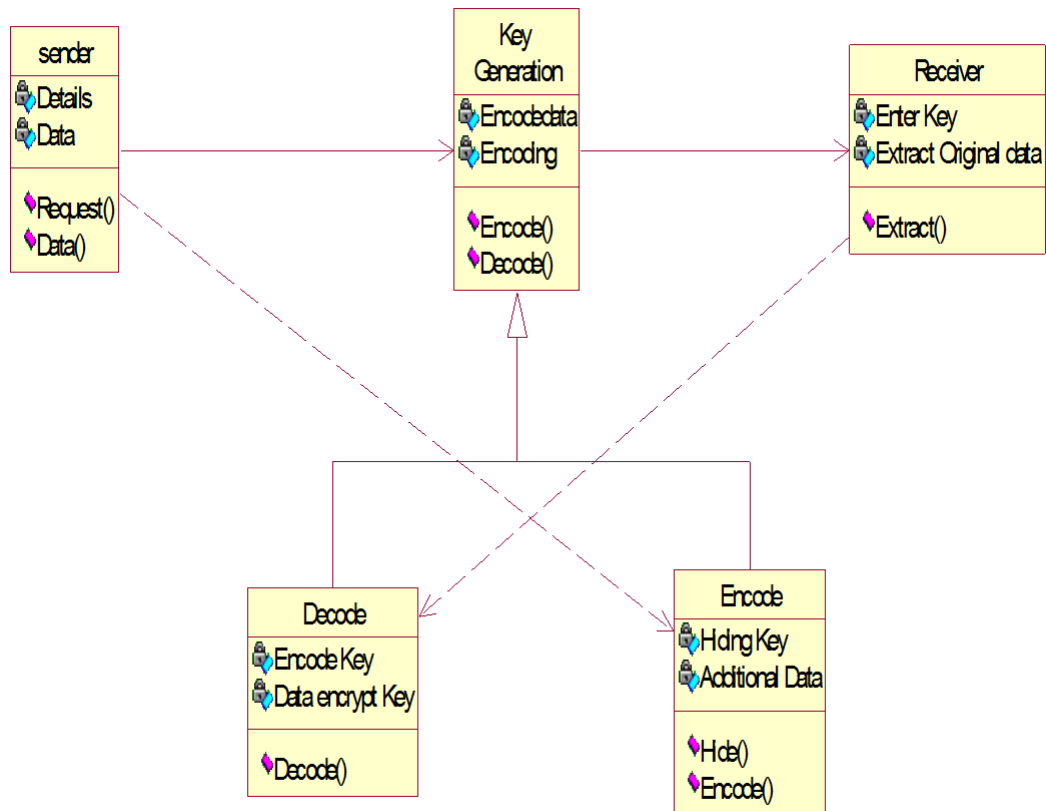


Fig 2.5.1 class diagram

2.5.2 COLLABRATION DIAGRAM :

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.

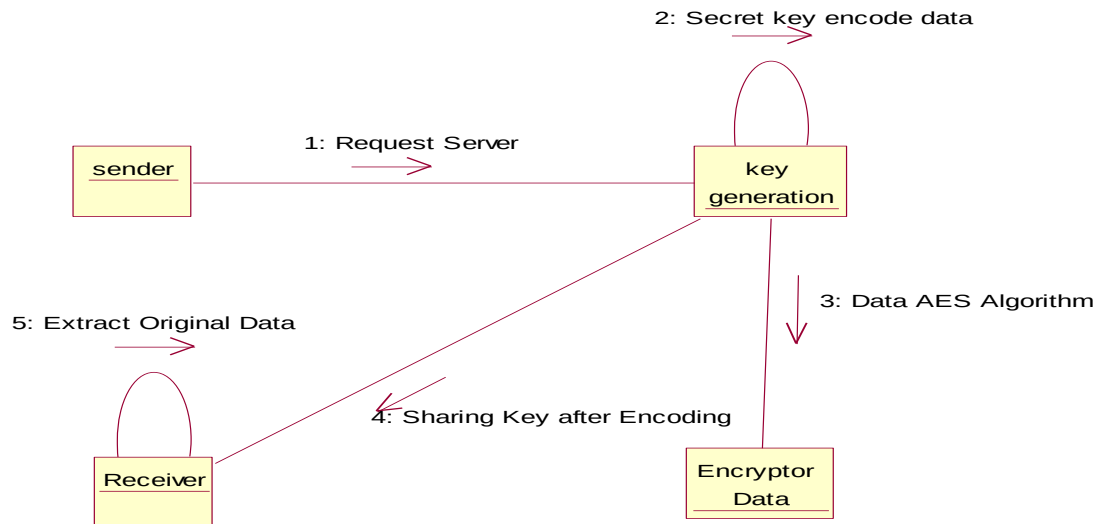


Fig 2.5.2 collaboration diagram

2.5.3 ACTIVITY DIAGRAM:

It shows the flow through a program from an defined start point to an end point. Activity diagrams describe the workflow behaviour of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel. Basic elements in activity diagrams are activities, branches (conditions or selections), transitions, forks and joins.

Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for analyzing a use case by describing what actions need to take place and when they should occur; describing a complicated sequential algorithm; and modelling applications with parallel processes. Activity diagrams do not give detail about how objects behave or how objects collaborate.

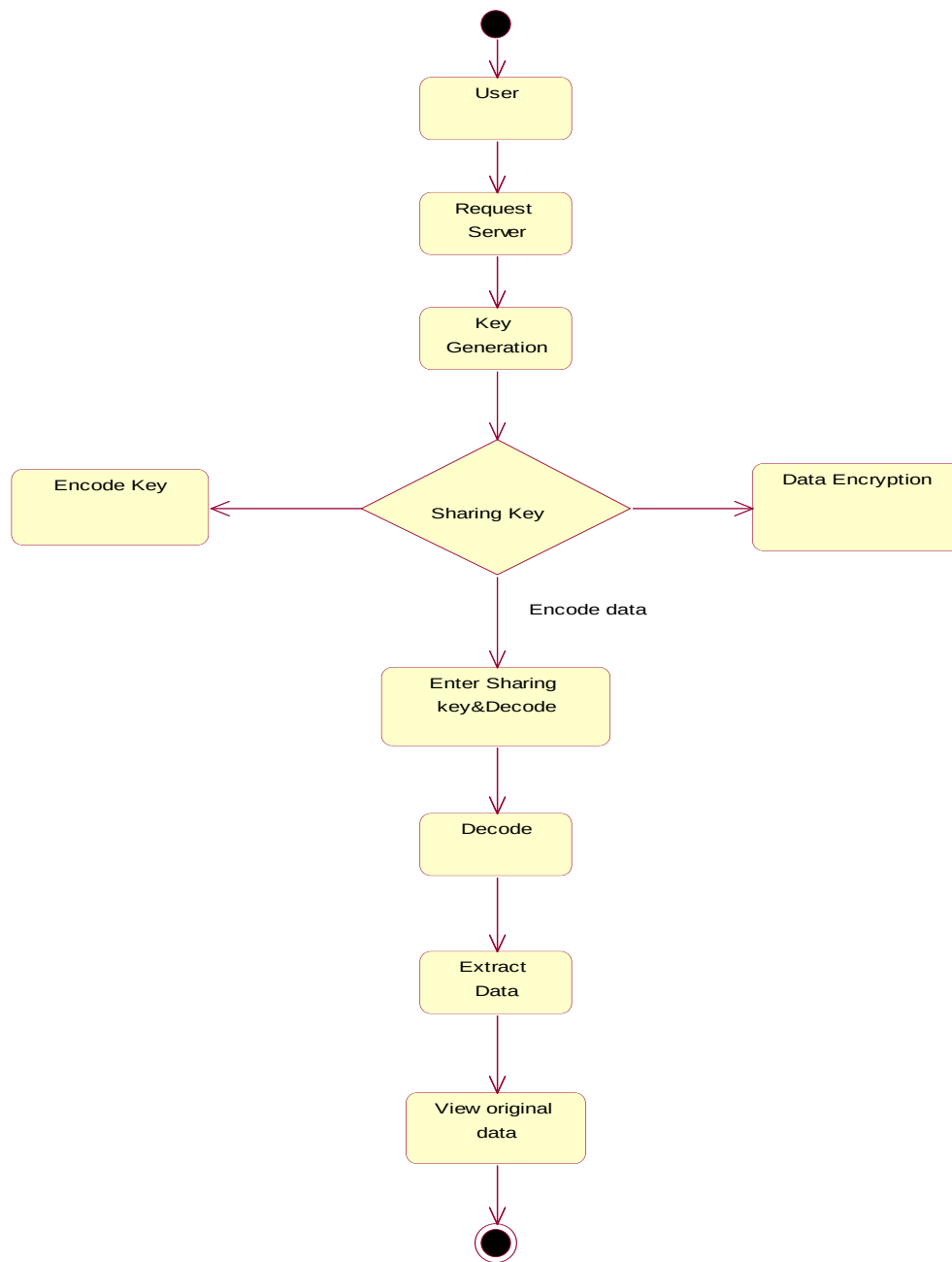


Fig 2.5.3 activity diagram

CHAPTER 3

ANALYSIS

3.1 User Profile Module:

The mobile device that receive the user details with some parameters, that recognize the authenticate user. this restricts the non-owner users to see information about the SMS we send. However, any mobile device using this service can get some additional profile examination has to be handled with some unique parameter. Through this function, the mobile device can allow authenticated profile owner to access the data and send secure SMS to others.

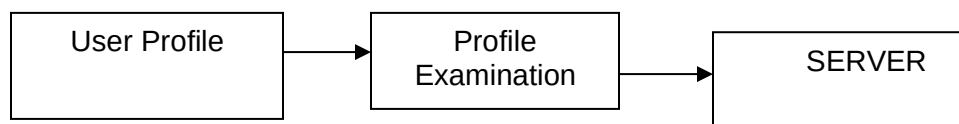


Fig 3.1 user profile module

3.1.1 SMS Communication:

The Authenticated mobile user can send the SMS with some key to the server. The mobile who wants to send SMS must be registered with server. The mobile sends the SMS with certain key to server. the server can encrypt the original message using AES algorithm and the send SMS to receiver through base station and mobile station

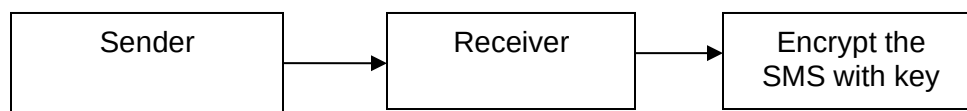


Fig 3.1.1 SMS communication

3.1.2 Authentication Server:

The Encrypted message can travel through base station. Receiver receives the message in secure inbox. Now the receiver wants to decrypts the message. So the receiver requests to the key using random number generator from server. then server generates the random number and send it to the receiver.

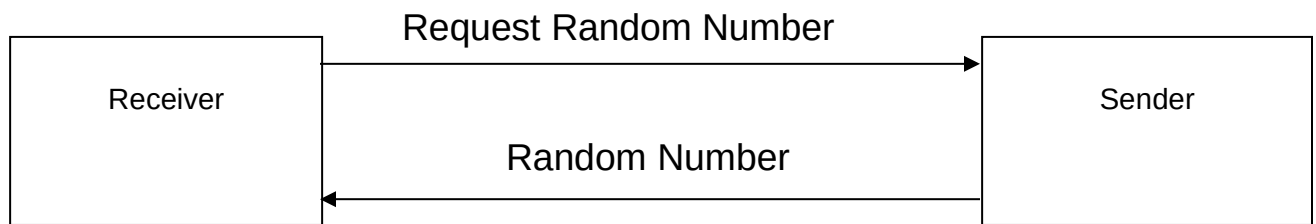


Fig 3.1.2 Authentication server

3.1.3 Symmetric Key:

Server recognizes the random number from receiver; from this server authenticate the authorized receiver. Then server sends the symmetric key to receiver. After getting symmetric key, receiver decrypts the encrypted message and extracts the original message in secure inbox.

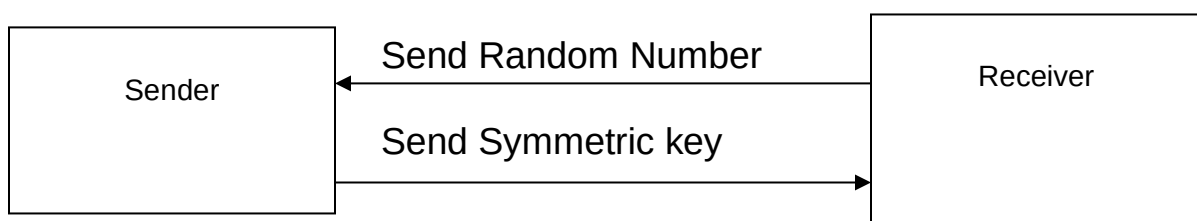


Fig 3.1.3 symmetric key

3.2 High-level description of the algorithm:

1. KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule
2. Initial Round
3. Rounds
 1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
 2. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
 3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
 4. AddRoundKey
4. Final Round (no MixColumns)
 1. SubBytes
 2. ShiftRows
 3. AddRoundKey

3.3 MD5 Algorithm:

MD5 is an algorithm that is used to verify data integrity through the creation of a 128-bit message digest from data input (which may be a message of any length) that is claimed to be as unique to that specific data as a fingerprint is to the specific individual. MD5, which was developed by Professor Ronald L. Rivest of MIT, is intended for use with digital signature applications, which require that large files must be compressed by a secure method before being encrypted with a secret key, under a public key cryptosystem. MD5 is currently a standard, Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321.

According to the standard, it is "computationally infeasible" that any two messages that have been input to the MD5 algorithm could have as the output the same message digest, or that a false message could be created through apprehension of the message digest. MD5 is the third message digest algorithm created by Rivest. All three (the others are MD2 and MD4) have similar structures, but MD2 was

optimized for 8-bit machines, in comparison with the two later formulas, which are optimized for 32-bit machines. The MD5 algorithm is an extension of MD4, which the critical review found to be fast, but possibly not absolutely secure. In comparison, MD5 is not quite as fast as the MD4 algorithm, but offers much more assurance of data security.

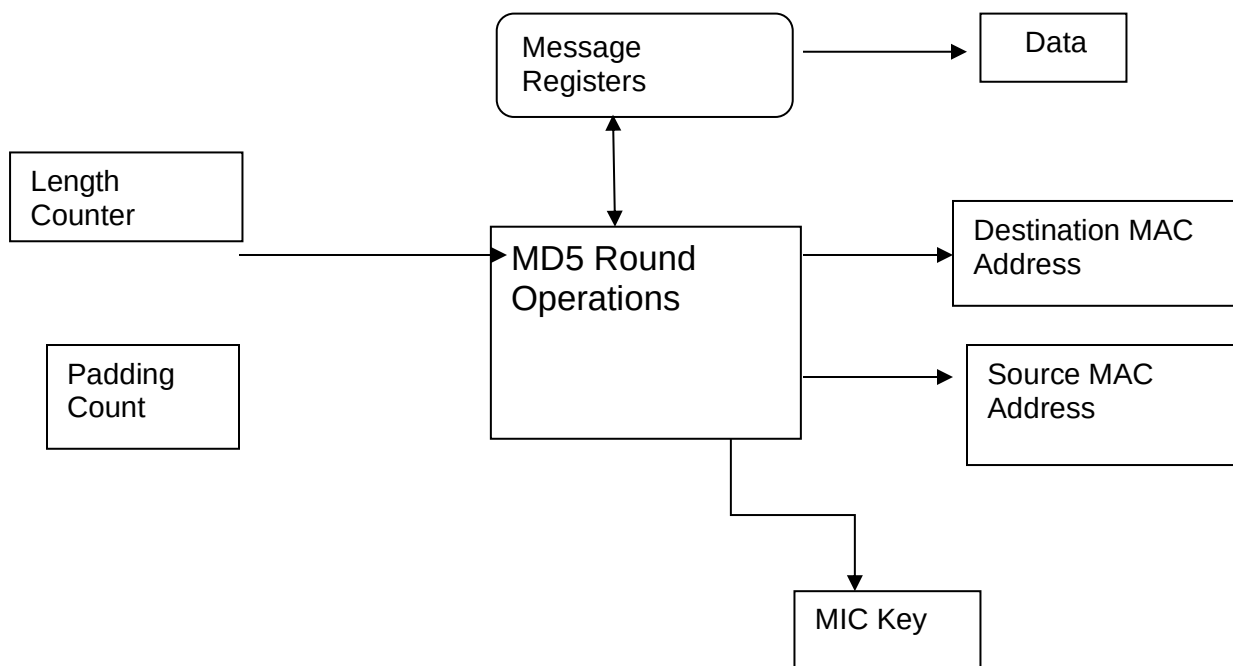


Fig 3.3 MD5 diagram

CHAPTER 4

DESIGN

4.1 Android:

A free, open source mobile platform. A Linux-based, multiprocessing, Multithreaded OS. Android is not a device or a product it's not even limited to phones you could build a DVR, a handheld GPS, an MP3 player, etc.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

- Makes mobile development easy.
- Full phone software stack including applications
- Designed as a platform for software development
- Android is open
- Android is free
- Community support

4.2 Features:

- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source Web Kit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3Dgraphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth,EDGE, 3G, and Wi-Fi** (hardware dependent)

- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plug in for the Eclipse IDE

4.2.1 Linux Kernel:

Android relies on Linux version 2.6 for core system services such as

- security
- memory management
- process management
- network stack
- driver model
- The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

4.2.2 Android Runtime:

- Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.
- Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.
- The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint.
- The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

4.2.3 Libraries:

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.

- System C Library
- Media Library
- Surface Manager
- Lib Web Core
- SGL
- 3D libraries
- Free Type
- SQLite

4.3 Application Framework:

- **Views** – used to build applications (lists, grid, buttons, text boxes and even embeddable web browser)
- **Content providers** – enable applications to access data from other applications or share their own data.
- **Resource manager** – provides access to non-code resources such as localized strings, graphic and layout files.
- **Notification manager** – enables applications to display custom alerts in status bar
- **Activity manager** – manages lifecycle of applications and provides navigation back stack

4.4 Anatomy of an Android Application:

There are four building blocks for an Android application:

- **Activity** - a single screen
- **Broadcast Receiver**- to execute in reaction to an external event(Phone Ring)

- **Service** - code that is long-lived and runs without a UI(Media Player)
- **Content Provider** - an application's data to be shared with other applications

4.5 Architecture:

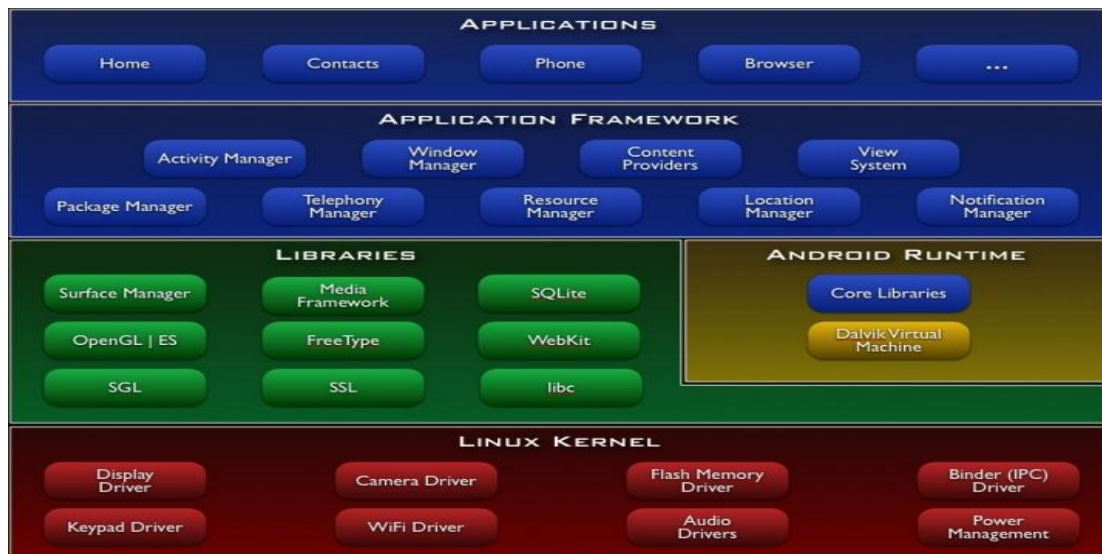


Fig 4.5 architecture

4.5.1 Android Building Blocks:

These are the most important parts of the Android APIs:

- **AndroidManifest.xml**
the control file-tells the system what to do with the top-level components
- **Activities**
an object that has a life cycle-is a chunk of code that does some work
- **Views**
an object that knows how to draw itself to the screen

- **Intents**
a simple message object that represents an "intention" to do something
- **Notifications AND Services**
is a small icon that appears in the status bar(SMS)
for alerting the user
Is a body of code that runs in the background

4.6 Development Tools:

The Android SDK includes a variety of custom tools that help you develop mobile applications on the Android platform. Three of the most significant tools are.

- **Android Emulator** -A virtual mobile device that runs on our computer -use to design, debug, and test our applications in an actual Android run-time environment
- **Android Development Tools Plug-in** -for the Eclipse IDE - adds powerful extensions to the Eclipse integrated environment
- **Dalvik Debug Monitor Service (DDMS)** -Integrated with Dalvik - this tool let us manage processes on an emulator and assists in debugging
- **Android Asset Packaging Tool (AAPT)** – Constructs the distributable Android package files (.apk)
- **Android Debug Bridge (ADB)** – provides link to a running emulator. Can copy files to emulator, install .apk files and run commands.

4.7 Lifecycle of activity:

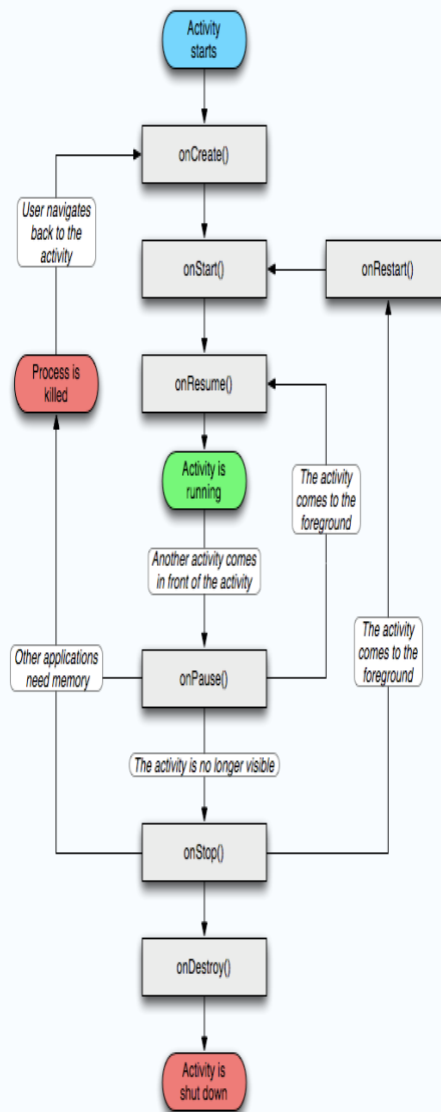


Fig 4.7 lifecycle

4.7.1 Create Android Application:

The first step is to create a simple Android Application using Eclipse IDE. Follow the option **File -> New -> Project** and finally select **Android New Application** wizard from the wizard list. Now name your application as **Hello World** using the wizard window as follows:

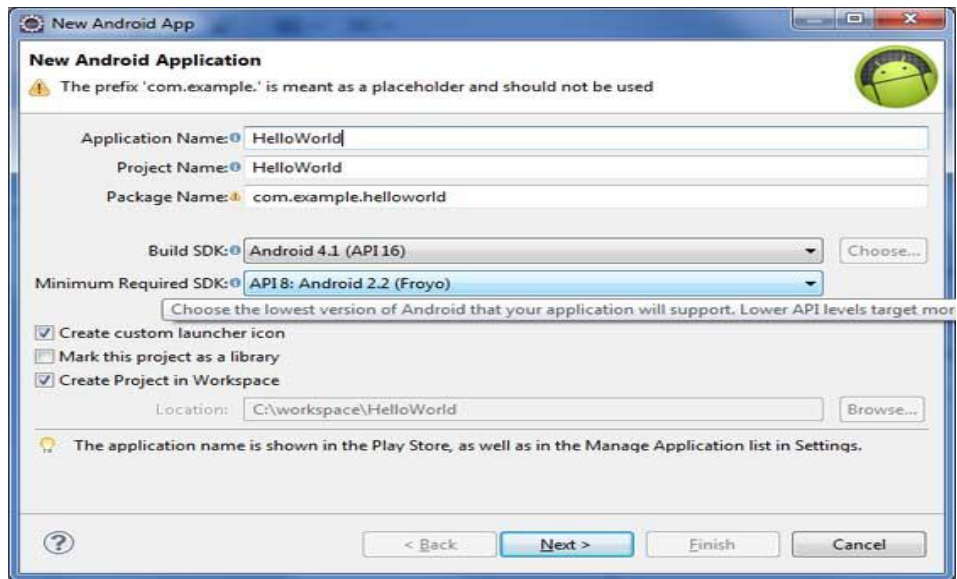


Fig 4.7.1 creating android app.

4.8 Testing:

The various levels of testing are

1. White Box Testing
2. Black Box Testing
3. Unit Testing
4. Functional Testing
5. Performance Testing
6. Integration Testing
7. Objective

4.8.1 White Box Testing:

- Execution of every path in the program.

4.8.2 Black Box Testing:

- Exhaustive input testing is required to find all errors.

4.8.3 Unit Testing:

- Unit testing, also known as Module Testing, focuses verification efforts on the module. The module is tested separately and this is carried out at the programming stage itself.

- Unit Test comprises of the set of tests performed by an individual programmer before integration of the unit into the system.
- Unit test focuses on the smallest unit of software design- the software component or module.
- Using component level design, important control paths are tested to uncover errors within the boundary of the module.
- Unit test is white box oriented and the step can be conducted in parallel for multiple components.

4.8.4 Functional Testing:

- Functional test cases involve exercising the code with normal input values for which the expected results are known, as well as the boundary values

4.8.5 Performance Testing:

- Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization of the program unit. It occurs throughout all steps in the testing process

4.8.6 Integration Testing:

- It is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with in the interface.
- It takes the unit tested modules and builds a program structure.
- All the modules are combined and tested as a whole.
- Integration of all the components to form the entire system and a overall testing is executed.

4.8.7 Objective:

- The objective is to take unit-tested modules and build a program structure that has been dictated by design.

CHAPTER 5

5.1 CONCLUSION:

Easy SMS protocol is successfully designed in order to provide end-to-end secure communication through SMS between mobile users. The analysis of the proposed protocol shows that the protocol is able to prevent various attacks. The transmission of symmetric key to the mobile users is efficiently managed by the protocol. This protocol produces lesser communication and computation overheads, utilizes bandwidth efficiently, and reduces message exchanged during authentication than SMSec and PK-SIM protocols

5.2 Future work:

In Future work We will use Multimedia Message Service (image, Audio, Video) has become one of the fastest and strong communication channels to transmit the information.

APPENDIX

A.SOURCE CODE:

```
package com.example.encdecsms;

import java.security.Key;
import java.util.ArrayList;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import android.app.Activity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.Toast;

public class
EncDecSMSActivity extends
Activity {

    /** Called when the activity is first created. */

    EditTextrecNum;

    EditTextsecretKey;

    EditTextmsgContent;

    Button send;

    Button cancel;

    @Override

    public void onCreate(Bundle savedInstanceState) {
```

```

setContentView(R.layout.main);
    recNum = (EditText) findViewById(R.id.recNum);
    secretKey = (EditText) findViewById(R.id.secretKey);
    msgContent = (EditText) findViewById(R.id.msgContent);
    send = (Button) findViewById(R.id.Send);
    cancel = (Button) findViewById(R.id.cancel);
    cancel.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
    finish();
    }
    });
    send.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
    String recNumString = recNum.getText().toString();
    String secretKeyString = secretKey.getText().toString();
    String msgContentString = msgContent.getText().toString();
    if (recNumString.length() > 0 &&secretKeyString.length() > 0
    &&msgContentString.length() > 0
    &&secretKeyString.length() == 16) {
    byte[] encryptedMsg = encryptSMS(secretKeyString,
    msgContentString);
    String msgString =byte2hex(encryptedMsg);
    sendSMS(recNumString, msgString); finish(

```

```

} else

Toast.makeText(
getBaseContext(),
"Please enter phone number, secret key and the message. Secret key must be 16
characters!",
Toast.LENGTH_SHORT).show();
}
});
}

public static void sendSMS(String recNumString, String encryptedMsg) {
try {
SmsManagersmsManager = SmsManager.getDefault();
ArrayList<String> parts = smsManager.divideMessage(encryptedMsg);
smsManager.sendMultipartTextMessage(recNumString, null, parts,
null, null);
} catch (Exception e) { e.printStackTrace();
}
}
public static String byte2hex(byte[] b) {

String hs = "";

String stmp = "";

for (int n = 0; n <b.length; n++)

```

B. SCREEN SHOTS:

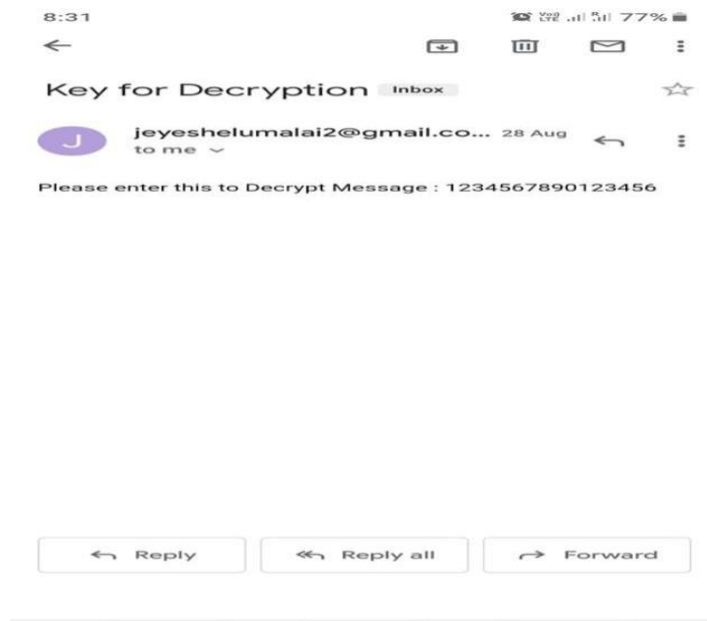


Fig 6.1 Key for decryption

This fig 6.1 shows the encrypted code received to the mobile through mail used to decrypt the text message



Fig 6.2 Enc33 SMS

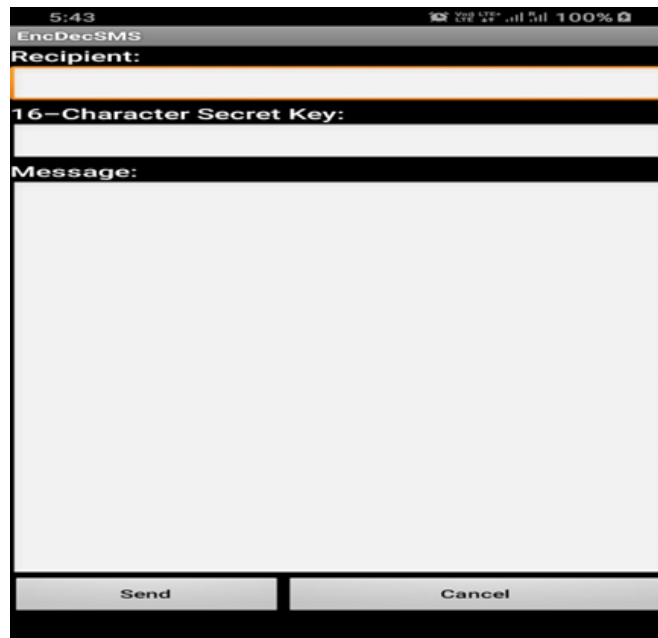


Fig6.3 Dec SMS

The above screen shot shows the panel to enter the recipient details to send the encrypted message

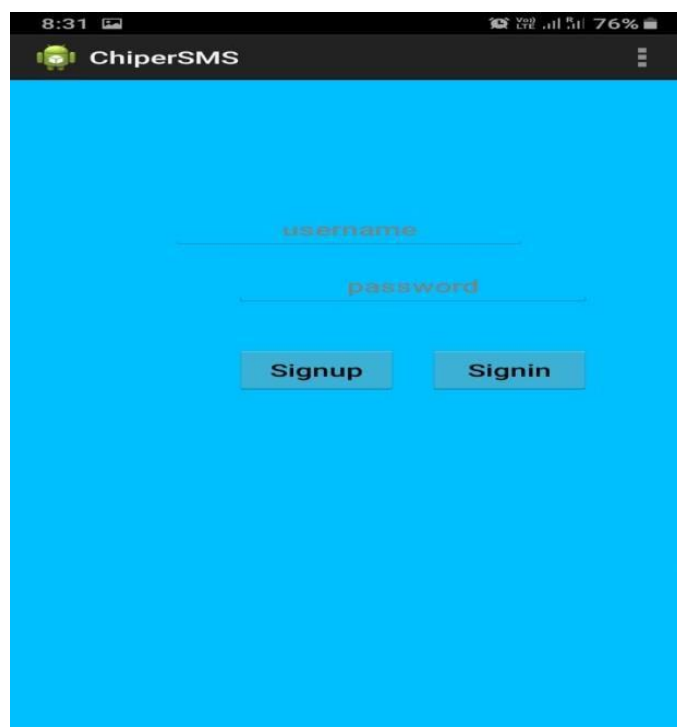


Fig 6.4 Chiper login

The above figure shows about the application used to send the encrypted messages to the recipient

References

[1]<https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=ENCRYPT%20SMS%20ANDROID>

[2]<https://www.scopus.com/home.uri>

[3] Muhammad Haikai Azam ; Dodi Isaksson Suharto ; Erwin; “Design and implementation of encrypted SMS on Android smartphone combining ECDSA - ECDH and AES”, proceeding on Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast), (2016).

[4] Frei Farrant ; Siti Umami Masrour ; Nolan; “Encrypted SMS application on Android with combination of caesar cipher and vigenary algorithm”, proceeding on International Conference on Cyber and IT Service Management (CITSM), (2014).

[5] Raffaele Pizzolatto ; Bruno Carpentaria ; Aniela; Arcangelo Castiglione ; Francesco Palmieri, “Text Compression and Encryption through Smart Devices for Mobile Communication”, proceeding on Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, (2013).

[6] Opium Salim Stimuli ; Handrail; Nurul; Erna Bushati Nibbana, “Hybrid RC4 and Affine Ciphers to Secure Short Message Service on Android”, proceeding on Third International Conference on Informatics and Computing (ICIC), (2018).

[7] Manoj Patel ; Vinay Sadhu ; Anurag Jain; ”SMS text Compression and Encryption on Android O.S”, proceeding on International Conference on Computer Communication and Informatics, (2014).