VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELGAUM – 590014

A Project Report on

“Integrating Lamport's OTP Authentication Scheme With Elliptic Curve Cryptography”

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering
in
Information Science & Engineering

Submitted by:

ANAND KUMAR	**1PI12IS013**
SUMIT KUMAR	**1PI12IS114**

Under the guidance of

Internal Guide
Mr. Raj Alandkar
Assistant Professor,
Department of ISE,
PESIT

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

PES INSTITUTE OF TECHNOLOGY

100 Feet Ring Road, BSK 3rd Stage, Bengaluru – 560085

January 2016 – May 2016

PES INSTITUTE OF TECHNOLOGY

100 Feet Ring Road, B S K 3rd Stage,
Bengaluru-560085

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **“Integrating Lamport's OTP Authentication Scheme With Elliptic Curve Cryptography”** carried out by **Anand Kumar**, bearing USN **1PI12IS013**, **Sumit Kumar**, bearing USN **1PI12IS114**, are bonafide students of **PES INSTITUTE OF TECHNOLOGY**, Bangalore, an autonomous institute, under VTU, in partial fulfillment for the award of degree of **BACHELOR OF ENGINEERING IN INFORMATION SCIENCE & ENGINEERING** of **Visvesvaraya Technological University, Belgaum** during the year **2016**. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the above said degree.

Mr. Raj Alandkar
Assistant Professor,
Department OF ISE
PESIT

Dr. Shylaja S S
Professor and Head,
Department of ISE
PESIT

Dr. K. S. Sridhar
Principal
PESIT

External Viva

Name of the Examiners **Signature with Date**

1.__________________________ __________________

2.__________________________ __________________

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

P.E.S. Institute of Technology

Department of Information Science and Engineering

Bengaluru – 560085

DECLARATION

We, **Anand Kumar and Sumit Kumar**, students of Eighth Semester B.E., in the Department of Information Science and Engineering, **P.E.S. Institute of Technology, Bangalore** declare that the project entitled **"Integrating Lamport's OTP Authentication Scheme With Elliptic Curve Cryptography"** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Engineering** in **Information Science and Engineering of Visvesvaraya Technological University, Belgaum** during the academic year **2015-16**. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

Name and USN		Signature
Anand Kumar	1PI12IS013	
Sumit Kumar	1PI12IS114	

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, and whose guidance and encouragement helped us in completing the project successfully.

We consider it a privilege to express gratitude and respect to all those who guided us throughout the course of the completion of the project.

We would like to express our heartfelt thanks to **Mr. Raj Alandkar**, Assistant Professor, Department of Information Science and Engineering, PESIT, our project guide, for his constant guidance, encouragement, support and invaluable advice without which this project would not have become a reality.

We would like to thank **Dr. Mamatha H R,** Professor, Dept. of ISE, PESIT without whom the project coordination with the department would not have been feasible.

We express our gratitude to **Dr. Shylaja S S**, Head of the Department of Information Science and Engineering whose guidance and support has been invaluable.

We extend our sincere thanks to **Dr. K S Sridhar,** Principal, PESIT for providing us with a congenial environment for carrying out the project.

Last, but not the least, we would like to thank our friends whose invaluable feedback helped us to improve the software by leaps and bounds, and our parents for their unending encouragement and support.

ABSTRACT

Establishing end-to-end authentication between devices and applications in Internet of Things (IoT) is a challenging task. Due to heterogeneity in terms of devices, topology, communication and different security protocols used in IoT, existing authentication mechanisms are vulnerable to security threats and can disrupt the progress of IoT in realizing Smart City, Smart Home and Smart Infrastructure, etc. To achieve end-to-end authentication between IoT devices/applications, the existing authentication schemes and security protocols require a two-factor authentication mechanism.

Therefore, as part of this project we review the suitability of an authentication scheme based on One Time Password (OTP) for IoT and proposed a scalable, efficient and robust OTP scheme.

Our proposed scheme uses the principles of lightweight Identity Based Elliptic Curve Cryptography scheme and Lamport's OTP algorithm. We evaluate analytically and experimentally the performance of our scheme and observe that our scheme with a smaller key size and lesser infrastructure performs on par with the existing OTP schemes without compromising the security level.

Our proposed scheme can be implemented in real-time IoT networks and is the right candidate for two-factor authentication among devices, applications and their communications in IoT.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1:

INTRODUCTION

1.1 Overview

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography.

In cryptography, a Lamport signature or Lamport one-time scheme is a method for constructing a digital signature. Lamport signatures can be built from any cryptographically secure one-way function; usually a cryptographic hash function is used.

A Lamport OTP does not require a shared secret. Lamport OTP's are used for validating a series of successive logins

1.2 Existing System

Revolution in the field of Internet of Things (IoT) is driving numerous applications in the area of Smart City, Smart Home, Smart Health and etc., to enhance the living standards of the people globally. To realize this, a plethora of digital devices are deployed which communicate with each other directly or through gateway or applications. From IoT application perspective, we envisage IoT as interconnected Applications, Devices, Gateways and Cloud platforms. Devices are grouped into different clusters wherein cluster head is denoted as gateway. Gateway manages the devices which belong to its cluster. Devices within the cluster communicate with each other directly or through the gateway. Further, these gateways are managed by IoT cloud platforms. These platforms are distributed geographically and communicate with each other.

Thus to enable secured and integrated communications across IoT, the application and devices need to authenticate each other through a cloud platform. Moreover, communication protocols differ from one application to other and are vulnerable to different security threats. Apart from this, some of the widely used IoT communication protocols such as MQTT (Message Queue Telemetry Transport), Constrained Application Protocol (CoAP) have no inbuilt security mechanisms. Though CoAP and other protocols for IoT have augmented security solutions such as Datagram Transport Layer

Security (DTLS), Secure Sockets Layer (SSL) and Transport Layer Security (TLS), they are vulnerable to known threats.

The strengths and weaknesses of several authentication schemes and their mitigations are proposed in some paper. OTP based on OneWay Function (RSA) is proposed by Bicakci et al. RSA based OTP and SecureID token algorithms were broken and attacks are reported worldwide. It is shown that using number theoretic concepts such as Chinese Remainder Theorem (CRT), Multiple Polynomial Quadratic Sieve (MPQS), attacks such as Integer Factorization, Discrete Logarithmic, Quantum Factoring, Forward, Fixed Point, Partial Key Exposure, Square Root, etc., are possible against RSA scheme. Yeh et al. proposed OTP based authentication scheme based on challenge response model. This scheme suffers from pre-play and impersonates attacks. Further, Linear Secret Sharing (LSS) based OTP generation is described in some paper. In some paper, the author introduced the concept of OTP to envisage password authentication over insecure channel. S/Key OTP system is designed based on Lamport’s OTP algorithm. Goyal et al. proposed an efficient OTP algorithm to authenticate device for (t + 1)th time, it needs to submit the (t)th received OTP. But this scheme requires re initialization.

Based on Lamport’s OTP algorithm, other OTP mechanisms based on HMAC (HOTP, TOTP) are discovered. Further, TOTP uses HMAC based OTP schemes with MD5, SHA1, SHA256 AND SHA512 are depicted. Though HMAC based authentication schemes are standardized by NIST, ANSI, IETF and are used in Secure Socket Layer (SSL), TLS, IPSec, etc., protocols, are prone to attacks such as birthday, forgery, full key recovery and collision.

To overcome this, OTP based on bilinear paring is discussed in some paper. However, bilinear pairing based scheme is computationally complex and practically deploying for authenticating transactions may be infeasible. Hence we argue that our proposed OTP generation technique which is based on the principles of IBE-ECC which does not require storage of the private information of the users/devices for generating OTP is a feasible solution for authenticating IoT devices/applications and communications between them.

Major Drawbacks of the Existing System

- OTP based on OneWay Function suffers from pre-play and impersonates attacks

- OTPs based on HMAC (HOTP, TOTP) are prone to attacks such as birthday, forgery, full key recovery and collision
- bilinear pairing based scheme is computationally complex and practically deploying for authenticating transactions may be infeasible

1.3 Proposed System

Identity based cryptography is a public key cryptosystem introduced by Shamir. Further, elliptic curve based pairing was developed by Boneh and Franklin. We adapt lightweight IBE-ECC to design a novel OTP scheme, which is a suitable candidate for authenticating IoT devices and applications. To envisage this, PKG performs the job of the OTP generator and validator at IoT cloud platform.

To overcome the above limitations, we propose an IBE curve OTP scheme and replace the hash function in Lamport's OTP algorithm with our proposed function based on IBE scheme. Here initial input to the algorithm is device/application identity, time, counter and public parameters. Initially algorithm computes the secret key of the device/application which is a torsion group point and a new torsion point is computed by using the x component of the current torsion point and time. Thus algorithm repeatedly computes the new torsion point from the previous torsion point and time till the desired number of times the operation needs to be performed. The output of the algorithm is an OTP which can be a truncated value of x component of the resultant torsion point.

Supporting Diagram for reference

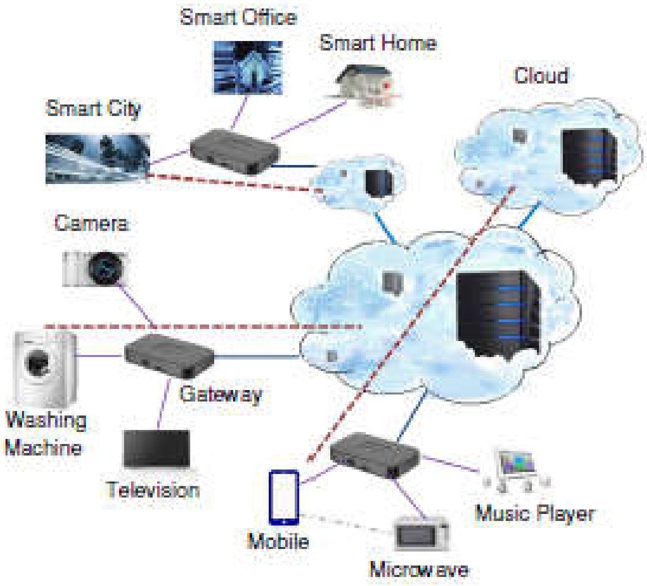


Fig 1.1 IoT Architecture

Major Advantages of the Proposed Sytem

- Hardness of our proposed OTP generation algorithm based on Lamport’s OTP technique is equivalent to solving Computational Diffie Hellman (CDH)
- The proposed method doesn’t suffers from pre-play and impersonates attacks
- The proposed system are not prone to attacks such as birthday, forgery, full key recovery and collision

CHAPTER 2

LITERATURE SURVEY

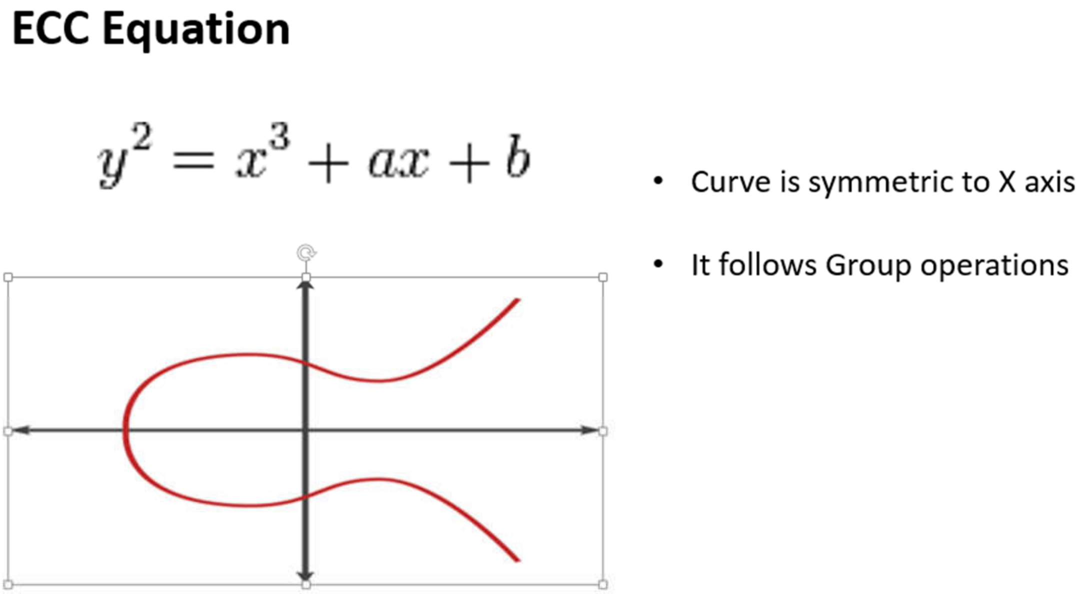
The theory of elliptic curves is a classical topic in many branches of algebra and number theory, but recently it is receiving more attention in cryptography. An elliptic curve is a two-dimensional (planar) curve defined by an equation involving a cubic power of coordinate x and a square power of coordinate y. One class of these curves is elliptic curves over finite fields, also called Galois fields. These elliptic curves are finite groups with special structures, which can play naturally, and even more flexibly, the roles of the modulus groups in the discrete logarithm problems.

Elliptic curves have been used actively in designing many mathematical, computational and cryptographic algorithms, such as integer factoring, primality proving, public key cryptosystems and pseudo-random number generators, etc. Essentially, elliptic curve cryptosystems promise a better future for cryptography: more security against powerful attacks in the era of computing capability.

Many research papers in Elliptic Curve Cryptography (ECC) have been published by researchers all over the world. However, the idea of using elliptic curves in cryptography is still considered a difficult concept and is neither widely accepted nor understood by typical technical people. The problem may stem from the fact that there is a large gap between the theoretical mathematics of elliptic curves and the applications of elliptic curves in cryptography.

2.1 Elliptic Curve

- Discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington)
- Shorter keys requires less CPU consumption and memory utilization
- It is as stronger as other cryptographic systems with larger keys
- Can be used for Encryption, Digital Signature, pseudo random generators, and other tasks



ECC Equation

$$y^2 = x^3 + ax + b$$

- Curve is symmetric to X axis
- It follows Group operations

Fig 2.1 ECC Curve

2.2 Group operations on Elliptic Curve

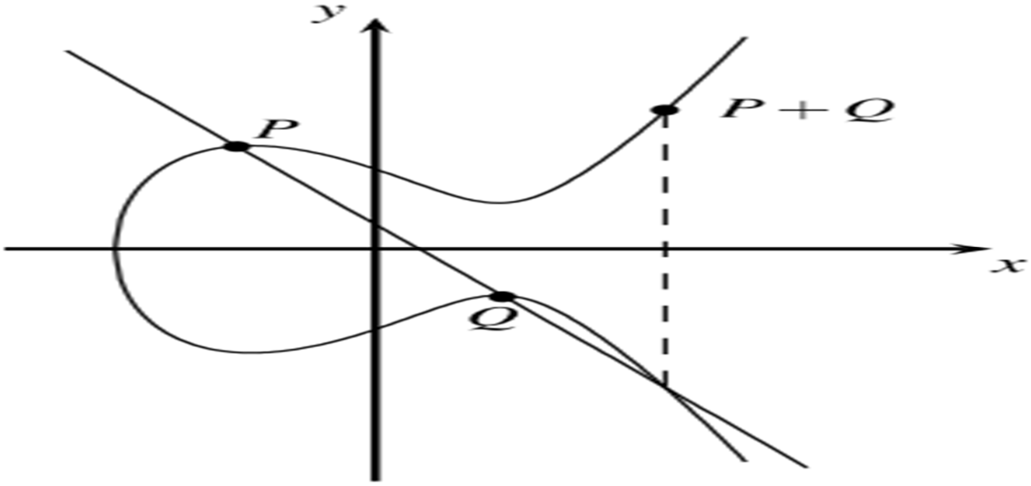
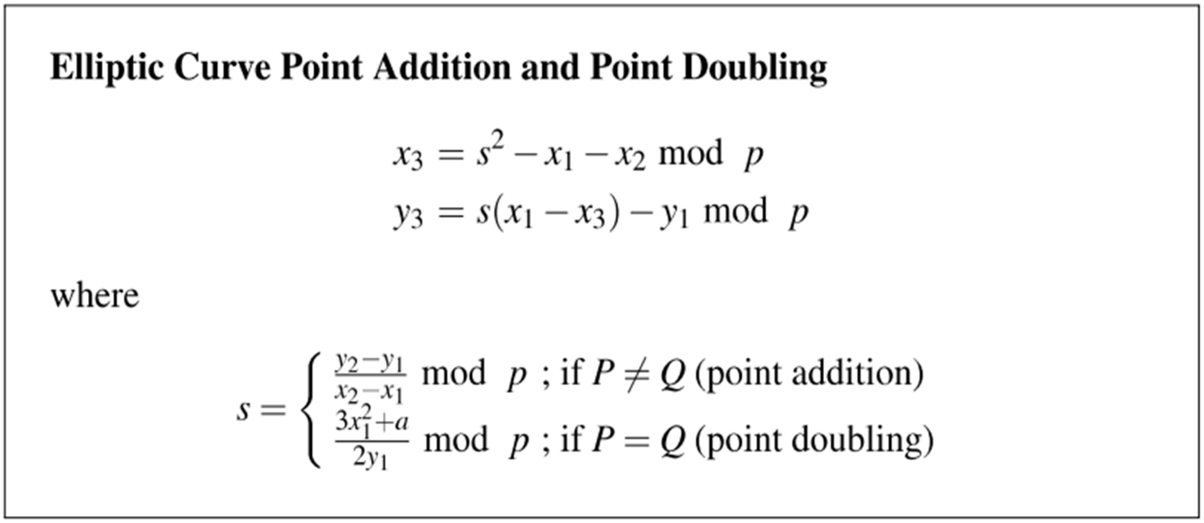


Fig 2.2 Addition of points on Curve



Elliptic Curve Point Addition and Point Doubling

$$x_3 = s^2 - x_1 - x_2 \bmod p$$
$$y_3 = s(x_1 - x_3) - y_1 \bmod p$$

where

$$s = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \bmod p \text{ ; if } P \neq Q \text{ (point addition)} \\ \frac{3x_1^2 + a}{2y_1} \bmod p \text{ ; if } P = Q \text{ (point doubling)} \end{cases}$$

Fig 2.3 Formulae

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

A **software requirements specification** (SRS) – a requirements specification for a software – is a complete description of the behavior of a system to be developed. In addition to a description of the software functions, the SRS also contains non-functional requirements. Software requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

3.1.1 Purpose

The purpose of this document is to provide Software Requirement Specification for Integrating Lamport's One Time Password Authentication Scheme with Elliptic Curve Cryptography. The software requirement specification document enlists all necessary requirements for project development.

3.1.2 Scope

The software product produced is an application by name "Integrating Lamport's One Time Password Authentication Scheme with Elliptic Curve Cryptography".

Establishing end-to-end authentication between devices and applications in Internet of Things (IoT) is a challenging task. Due to heterogeneity in terms of devices, topology, communication and different security protocols used in IoT, existing authentication mechanisms are vulnerable to security threats and can disrupt the progress of IoT in realizing Smart City, Smart Home and Smart Infrastructure, etc. To achieve end-to-end authentication between IoT devices/applications, the existing authentication schemes and security protocols require a two-factor authentication mechanism. Therefore, as part of this project we review the suitability of an authentication scheme based on One Time Password (OTP) for IoT and proposed a scalable, efficient and robust OTP scheme. Our proposed scheme uses the principles of lightweight Identity Based Elliptic Curve Cryptography scheme and Lamport's OTP algorithm. We evaluate analytically and experimentally the performance of our scheme and observe that our scheme with a smaller key size and lesser infrastructure performs on par with the existing OTP schemes without compromising the security level. Our proposed scheme can be implemented in real-time IoT networks and is the right candidate for two-factor authentication among devices, applications and their communications in IoT.

3.1.3 Definitions, Acronyms, and Abbreviations

Various divisions in the project

- **Elliptic Curve Cryptography Core Algorithm**

  Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography (based on plain Galois fields) to provide equivalent security. Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

- **Lamports One Time Password Authentication Scheme**

  The Lamport algorithm for generating and applying one-time passwords (OTPs) is a simple solution that provides great value in the right context. Not only can the Lamport OTP scheme provide effective security for distributed client/service interactions, but it's also simple to comprehend and implement. Louis Iacona introduces the Lamport algorithm, then describes an OTP reference implementation for an extensible, Java-based library.

  There's a subtle beauty in simple things that present great value. To paraphrase Albert Einstein, a solution to a problem should be as simple as it can be, but no simpler. Applying a one-time password (OTP) scheme between distributed systems makes it more difficult for a would-be intruder to access and gain unauthorized control of restricted resources such as data, physical devices, or service end points. An OTP scheme is obviously a step up from completely open access, or access limited only by physical network barriers. But a solution based on an OTP challenge also has some advantages over static, infrequently changing passwords, because the window of opportunity to gain access to credentials is much smaller. There's a practical place for either type of authentication, or even both used in concert.

The Lamport OTP approach is based on a mathematical algorithm for generating a sequence of "passkey" values, each successor value based on the value of its predecessor.

- **Client and Server applications**

  These applications will be implemented as a standalone java application that will integrate the Elliptic Curve Cryptography and Lamports OTP scheme. The client and server applications will be executing in different hosts and we will show the communication between them will be encrypted using Elliptic curve cryptography. And also the clients' authentication will be established using Lamports One time password authentication scheme.

3.2 General Description

3.2.1 Product Perspective

- Generating the private key and public key on request
- Providing various NIST standards like NIST_P_192, NIST_P_256, NIST_P_384, NIST_P_521
- Implementing Encryption and Decryption option
- Saving the private key and public key to the file system
- Persisting the public key and private key for later use.
- Implementing Lamports' forward function
- Implementing Lamports inverse function
- Implementing socket layer for secured communication between a client and server.

3.2.2 Product functions

- The user of this project will be executing the Elliptic curve cryptography tool for specifying the NIST standard to be used and performing various operations like encryption, decryption, saving the keys etc.
- The end user of the client will be initiating the communication with the server by providing the server host and port combination
- The user of the server will be accepting the client request and will be responding to the client request
- The server will be verifying the OTP of the client in each and every interaction
- The messages from server to client and from client to server will be encrypted using elliptic curve cryptography algorithm

3.2.3 User Characteristics

- In this project we will need a small amount of configuration work to be done in the Eclipse box to set up the application
- User also needs to install MySQL software for persisting the login credentials of both the server and the client.
- The input provided by the users should be any sample messages that are used for communication between client and server.

3.2.4 Assumptions and Dependencies

- JDK has to be installed in all the machines where the Client is deployed and also where the Server program has been executed.
- The Relational Database software like MySQL or ORACLE should be installed for providing the data storage.
- There shall not be any firewall or other engines that prevents the remote requests from the interface portal.
- There shouldn't be any permission related issues on any cluster. The host operating system should take of permitting all the requests to the cluster from the interface layer.

3.3 Requirements

3.3.1 Functional Requirements

- Account access management feature for both client and the server
- Elliptic curve cryptography's encryption feature
- Elliptic curve cryptography's decryption feature
- Elliptic curve cryptography's key generation feature
- Lamports forward OTP function
- Lamports inverse OTP function
- Each message to and fro client and server should be encrypted using elliptic curve cryptography.

3.3.2 Non Functional Requirements

- Should be easier to access it from the various browsers available.
- Response time of the applications should reflect the real time observations.
- The algorithm should never fail in any of the test cases.
- Each user's activity should be separated from the other user's activities

3.3.3 Software Requirements

- Windows Builder plugin
- Operating System: Windows XP or higher
- JDK 1.6
- RDBMS like MySQL or ORACLE
- Eclipse

3.3.4 Hardware Requirements

- Processor: Intel Pentium 4 or higher

- RAM: Min 512MB
- Hard Disk: 40GB

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.[citation needed] The UML has become the standard language in object-oriented analysis and design.[citation needed] It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.[citation needed]

System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. In other words the first step in the solution to the problem is the design of the project.

The design of the system is perhaps the most critical factor affecting the quality of the software. The objective of the design phase is to produce overall design of the software. It aims to figure out the modules that should be in the system to fulfill all the system requirements in an efficient manner.

The design will contain the specification of all these modules, their interaction with other modules and the desired output from each module. The output of the design process is a description of the software architecture.

The design phase is followed by two sub phases

- High Level Design
- Detailed (Low) Level Design

4.2 System Architecture Diagram

The below figure shows a general block diagram describing the activities performed by this project.

The entire architecture has been implemented in nine modules which we will see in high level design and low level design in later chapters.

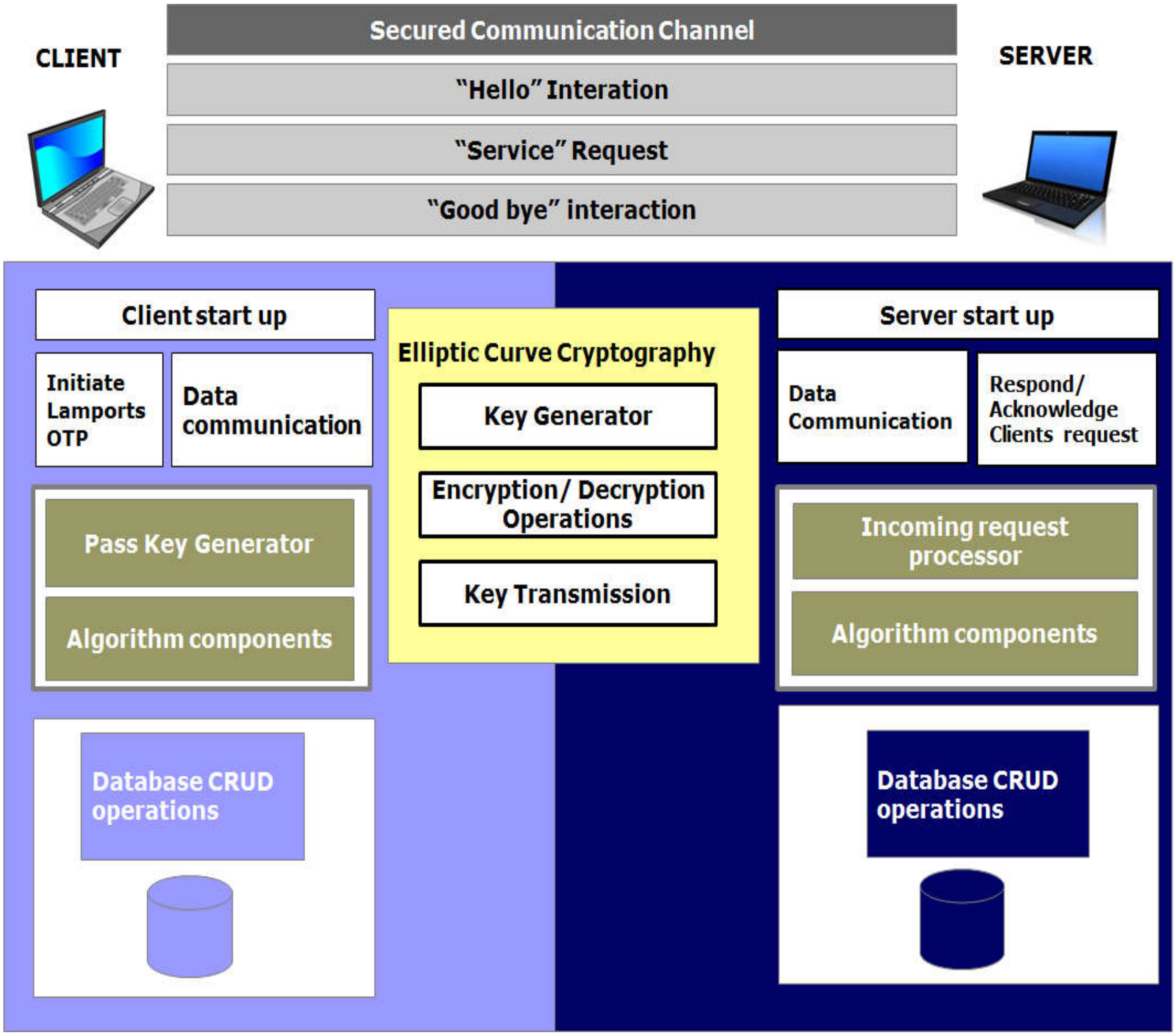


Fig 4.1 System Architecture Diagram

Various divisions in the project

- **Elliptic Curve Cryptography Core Algorithm**

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography (based on plain Galois fields) to provide equivalent security. Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

- **Lamports One Time Password Authentication Scheme**

The Lamport algorithm for generating and applying one-time passwords (OTPs) is a simple solution that provides great value in the right context. Not only can the Lamport OTP scheme provide effective security for distributed client/service interactions, but it's also simple to comprehend and implement. Louis Iacona introduces the Lamport algorithm, then describes an OTP reference implementation for an extensible, Java-based library.

There's a subtle beauty in simple things that present great value. To paraphrase Albert Einstein, a solution to a problem should be as simple as it can be, but no simpler. Applying a one-time password (OTP) scheme between distributed systems makes it more difficult for a would-be intruder to access and gain unauthorized control of restricted resources such as data, physical devices, or service end points. An OTP scheme is obviously a step up from completely open access, or access limited only by physical network barriers. But a solution based on an OTP challenge also has some advantages over static, infrequently changing passwords, because the window of opportunity to gain access to credentials is much smaller. There's a practical place for either type of authentication, or even both used in concert.

The Lamport OTP approach is based on a mathematical algorithm for generating a sequence of "passkey" values, each successor value based on the value of its predecessor.

- **Client and Server applications**

These applications will be implemented as a standalone java application that will integrate the Elliptic Curve Cryptography and Lamports OTP scheme. The client and server applications will be executing in different hosts and we will show the communication between them will be encrypted using Elliptic curve cryptography. And also the clients' authentication will be established using Lamports One time password authentication scheme.

4.3 High Level Design

In the high level design, the proposed functional and non-functional requirements of the software are depicted. Overall solution to the architecture is developed which can handle those needs

4.3.1 Data flow diagram

A data flow diagram is the graphical representation of the flow of data through an information system. DFD is very useful in understanding a system and can be efficiently used during analysis.

A DFD shows the flow of data through a system. It view a system as a function that transforms the inputs into desired outputs. Any complex systems will not perform this transformation in a single step and a data will typically undergo a series of transformations before it becomes the output.

With a data flow diagram, users are able to visualize how the system will operate that the system will accomplish and how the system will be implemented, old system data flow diagrams can be drawn up and compared with a new systems data flow diagram to draw comparisons to implement a more efficient system.

Data flow diagrams can be used to provide the end user with a physical idea of where the data they input, ultimately as an effect upon the structure of the whole system.

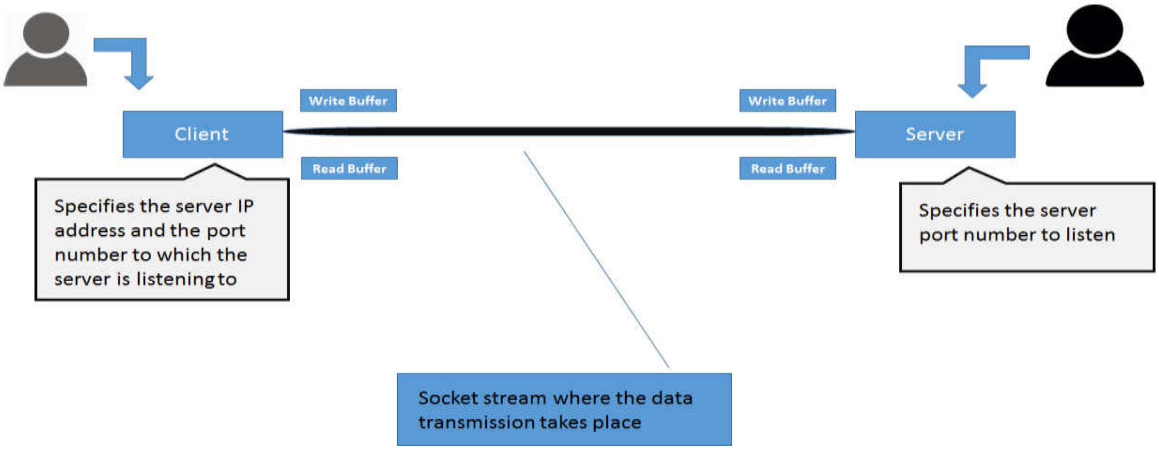

Write Buffer
Write Buffer
Client
Server
Read Buffer
Read Buffer
Specifies the server IP address and the port number to which the server is listening to
Specifies the server port number to listen
Socket stream where the data transmission takes place

Fig 4.2 Module 1: Basic Client Server socket communication set up

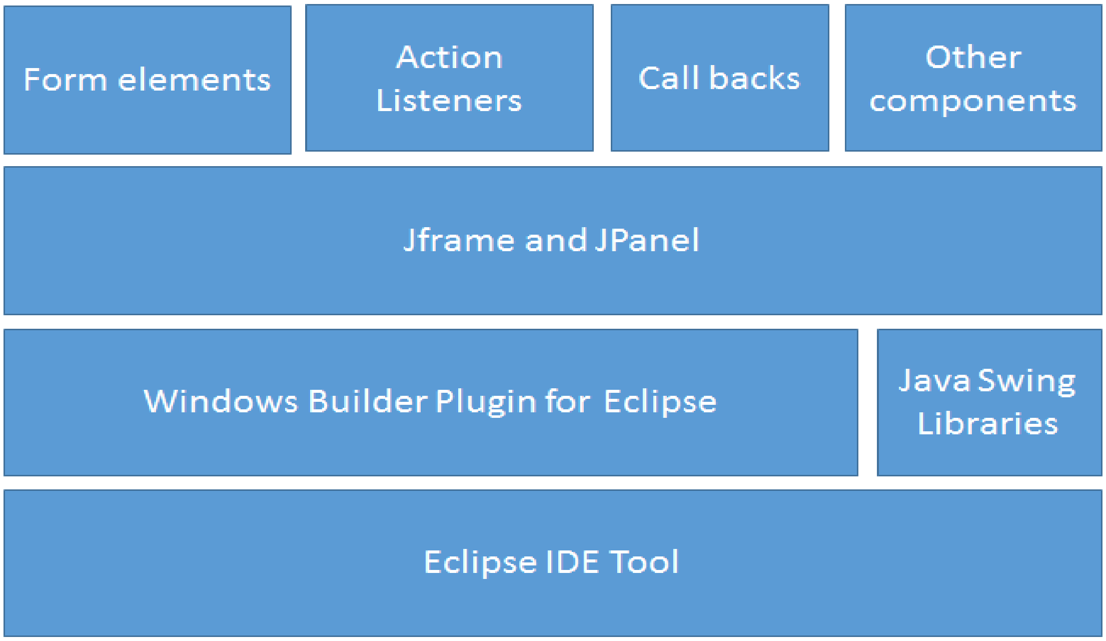


Fig 4.3 Module 2: Development of User Interface for client and server

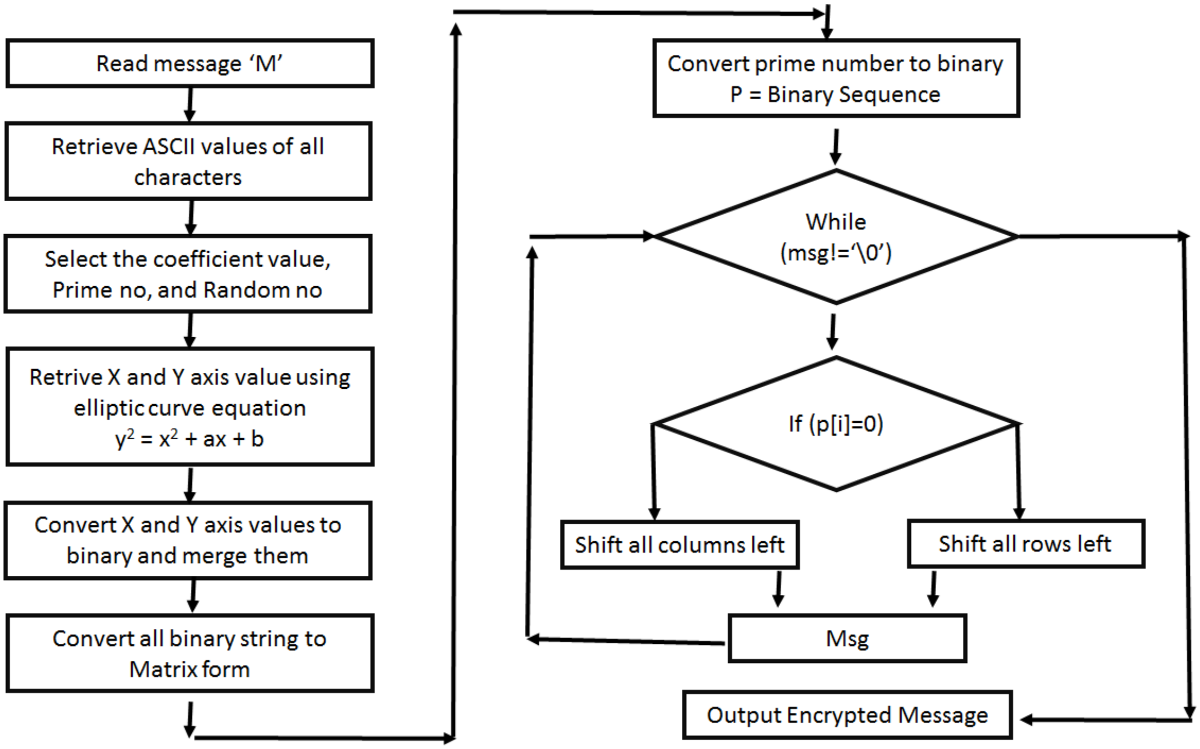
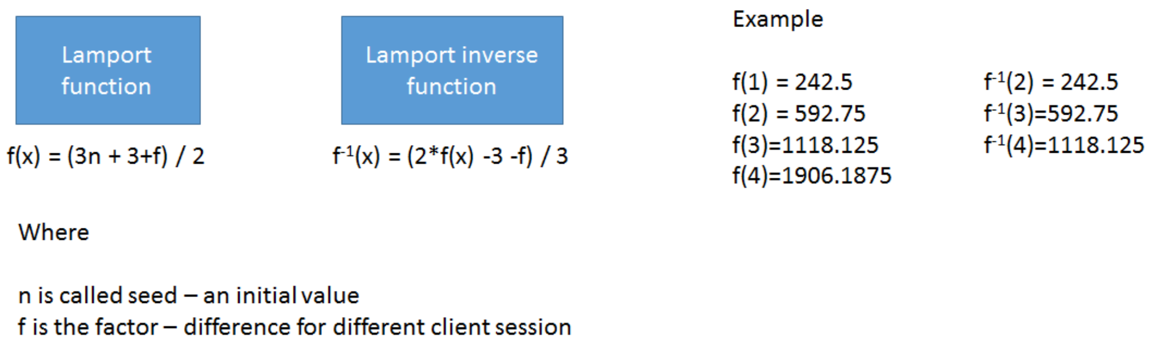

Read message 'M'
Retrieve ASCII values of all characters
Select the coefficient value, Prime no, and Random no
Retrive X and Y axis value using elliptic curve equation $y^2 = x^2 + ax + b$
Convert X and Y axis values to binary and merge them
Convert all binary string to Matrix form
Convert prime number to binary P = Binary Sequence
While (msg!='\0')
If (p[i]=0)
Shift all columns left
Shift all rows left
Msg
Output Encrypted Message

Fig 4.4 Module 3: Implementation of Elliptic Curve Cryptography Algorithm



Lamport function

$f(x) = (3n + 3+f) / 2$

Lamport inverse function

$f^{-1}(x) = (2*f(x) -3 -f) / 3$

Example

$f(1) = 242.5$
$f(2) = 592.75$
$f(3)=1118.125$
$f(4)=1906.1875$

$f^{-1}(2) = 242.5$
$f^{-1}(3)=592.75$
$f^{-1}(4)=1118.125$

Where

n is called seed – an initial value
f is the factor – difference for different client session

Fig 4.5 Module 4: Implementation of Lamport function and Inverse function

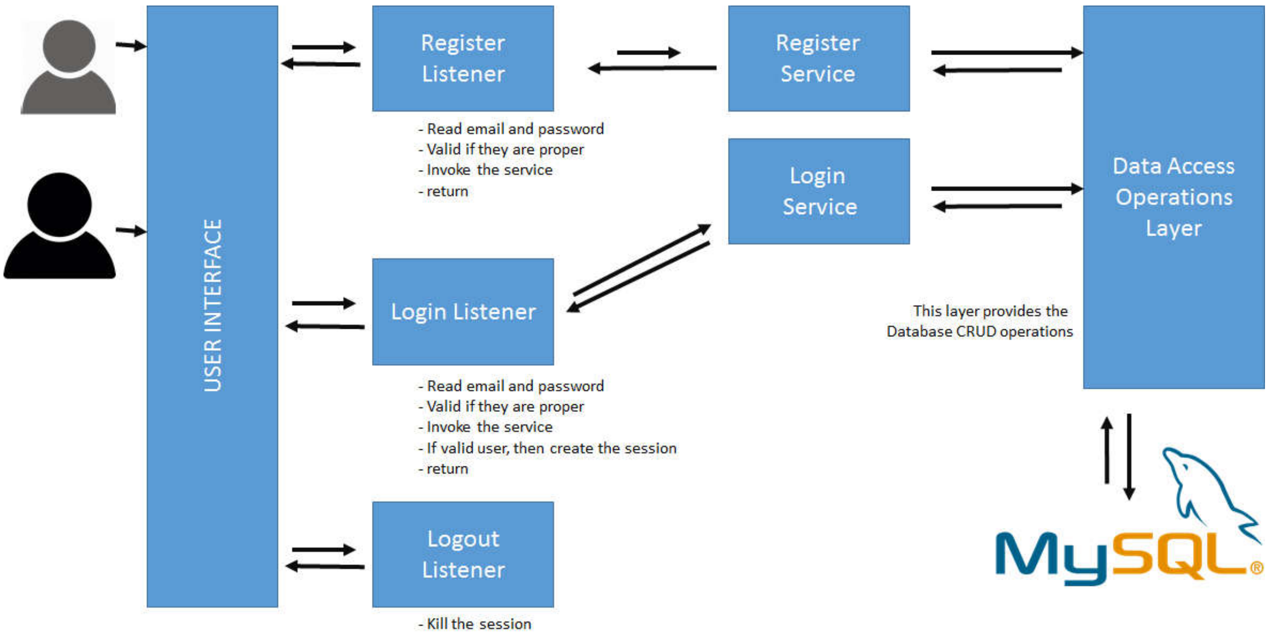

USER INTERFACE
Register Listener
- Read email and password
- Valid if they are proper
- Invoke the service
- return
Login Listener
- Read email and password
- Valid if they are proper
- Invoke the service
- If valid user, then create the session
- return
Logout Listener
- Kill the session
Register Service
Login Service
Data Access Operations Layer
This layer provides the Database CRUD operations
MySQL

Fig 4. 6 Module 5: Implementation of Account access control operation

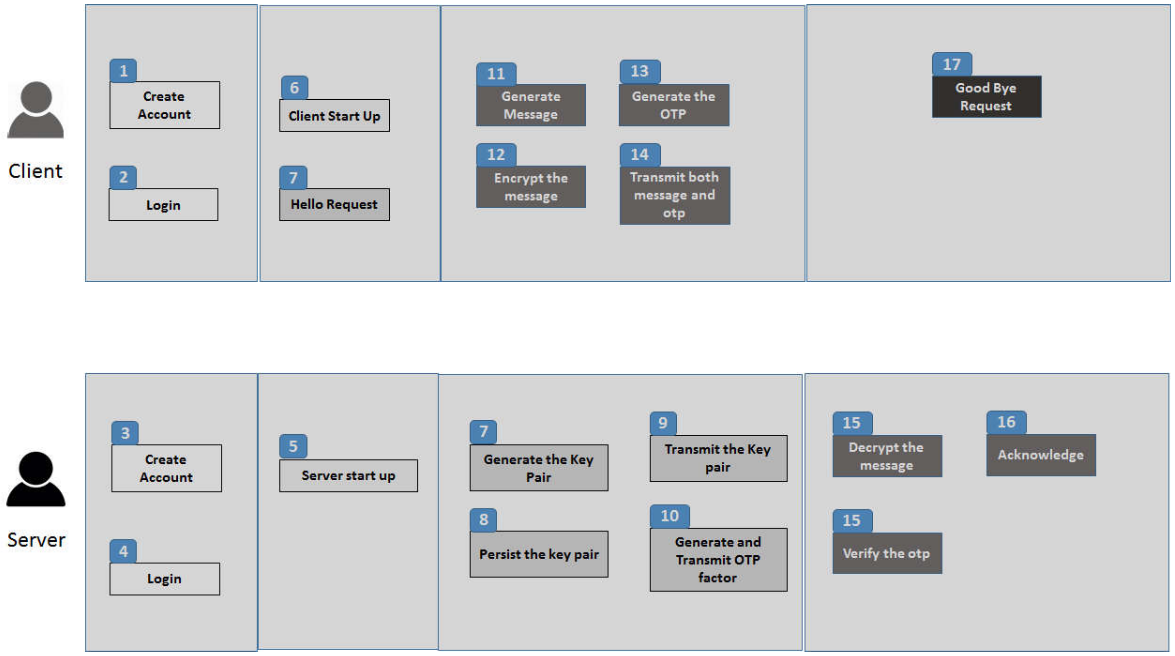


Fig 4.7 Module 6: Integration Module

4.4 LOW LEVEL DESIGN

During the detailed phase, the view of the application developed during the high level design is broken down into modules and programs. Logic design is done for every program and then documented as program specifications. For every program, a unit test plan is created.

The entry criteria for this will be the HLD document. And the exit criteria will the program specification and unit test plan (LLD).

4.4.1 Use case diagram

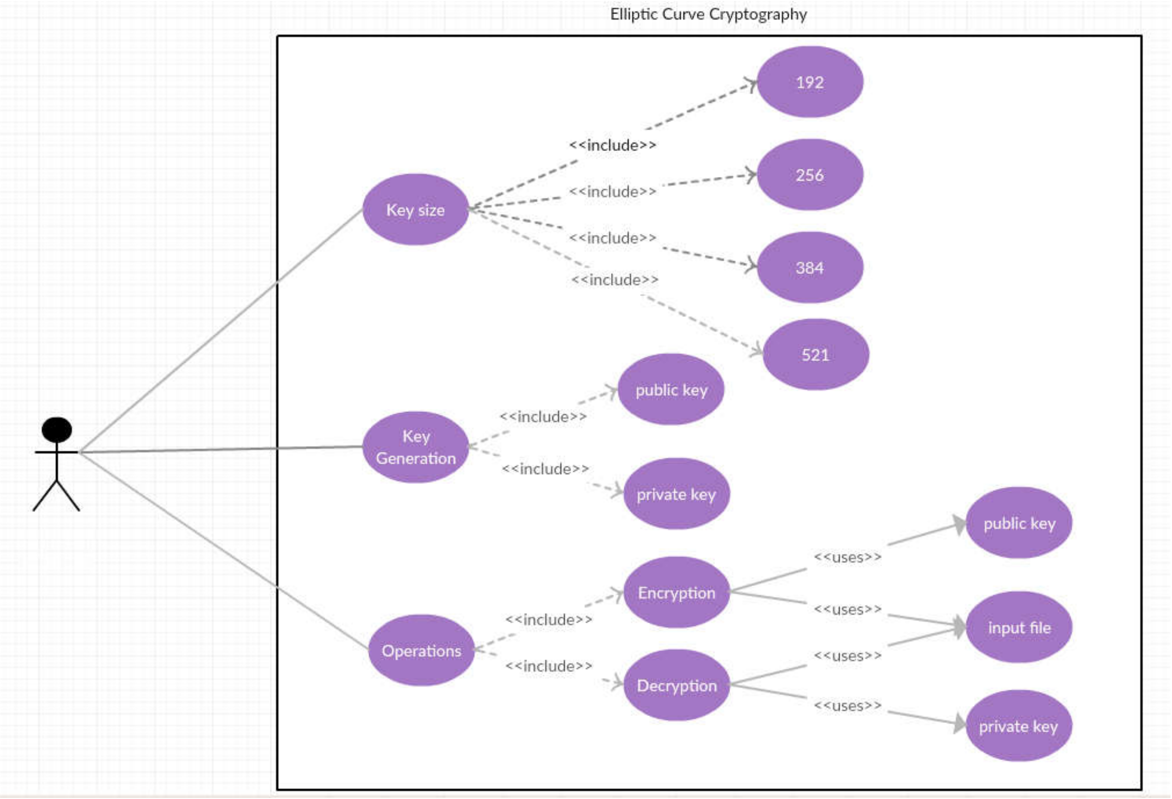


Fig 4.8 Use Case 1

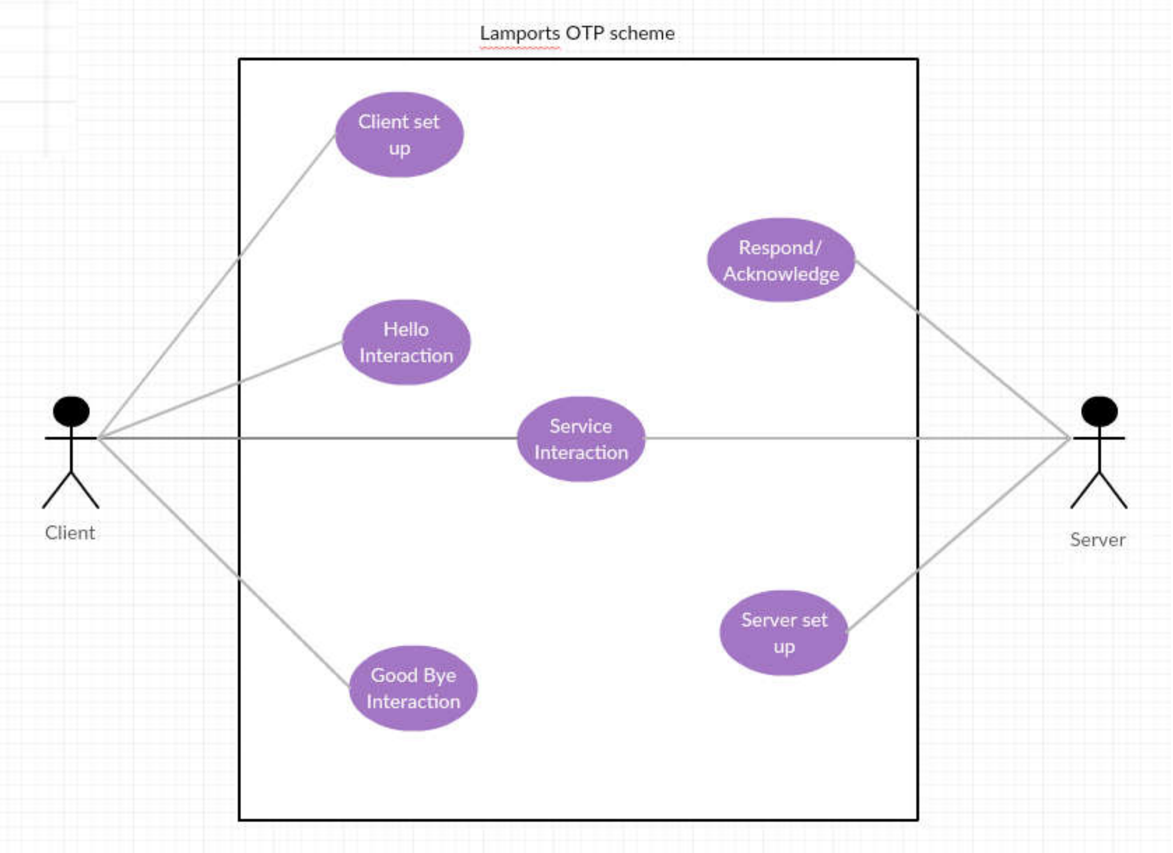


Fig 4.9 Use Case 2

The external objects that interact directly with the system are called ***actors.*** Actors include humans, external devices and other software systems. The important thing about actors is that they are not under control of the application. In this project, user of the system is the actor.

To find use cases, for each actor, list the fundamentally different ways in which the actor uses the system. Each of these ways is a *use case.*

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In other words, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

Implementation is one of the most important phases of the Software Development Life Cycle (SDLC). It encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. Specifically, it involves coding the system using a particular programming language and transferring the design into an actual working system.

This phase of the system is conducted with the idea that whatever is designed should be implemented; keeping in mind that it fulfills user requirements, objective and scope of the system. The implementation phase produces the solution to the user problem.

5.2 Implementation Steps

The implementation steps of the project is as follows:

1. Installation of Eclipse
2. Installation of Apache Tomcat Server
3. Installation of MySQL Database
4. Create database ECCDB.
    a. Create table Client and Server having attributes username and password
5. Run both the Client and Server projects.
6. A socket connection is formed between client and server
7. Keys are generated and message is sent across connection after getting encrypted.

5.3 Pseudo Code

5.3.1 Private Key Generation :

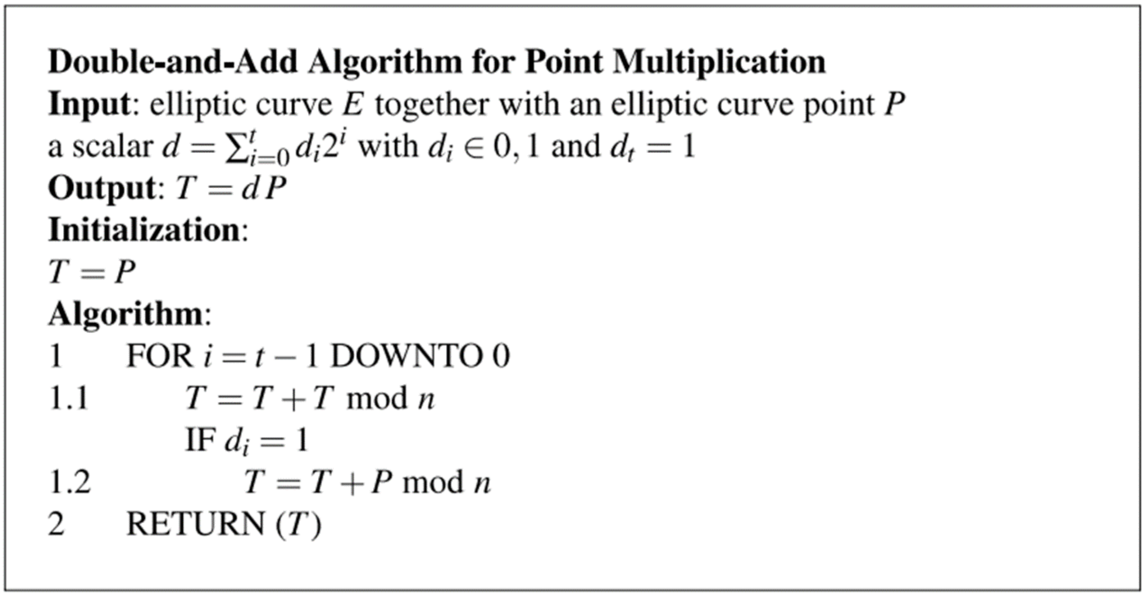
```
Randomly select the "private key", such that it is relatively prime to "p":
Private Key-→Big Integer
DO
    Private Key = new Big Integer (p.bitLength(), rnd)

WHILE (Private Key is relatively prime to p)
```

5.3.2 Public Key Generation :

```
Calculate the public key = Private Key * g.
ECPoint g →curve.getBasePoint()
ECPoint Public Key = curve.multiply(g, private Key)
//scalar multiplication using double and add algorithm
```

5.3.3 Scalar Multiplication:



Double-and-Add Algorithm for Point Multiplication

Input: elliptic curve E together with an elliptic curve point P

a scalar $d = \sum_{i=0}^{t} d_i 2^i$ with $d_i \in 0,1$ and $d_t = 1$

Output: $T = dP$

Initialization:

$T = P$

Algorithm:

```
1      FOR $i = t - 1$ DOWNTO 0
1.1        $T = T + T$ mod $n$
           IF $d_i = 1$
1.2            $T = T + P$ mod $n$
2      RETURN ($T$)
```

Fig 5.1 Double and Add Algorithm

5.3.4 Encryption :

- Encrypt each encoded point into a pair of points [C_1, C_2]

[C_1, C_2] = [kG, P_m + kP_G],

k is a randomly generated integer such that 1 <= k < p-1,

G is the base point,

P_m is the encoded point from the plain text,

P_G is the point provided in the public key.

5.3.5 Decryption :

- We have to get back the message ‘M’ that was sent to us

  M=C2-k*C1

- Calculate the encoded point(point on the graph)

  P_m = C_2 - kC_1, where:

  [C_1, C_2] is the cipher text (we received)
  k is the private key.

- Decode the encoded point, and the message is decrypted back to original message.

5.3.6 Lamport's OTP :

Lamport’s OTP generation algorithm is based on successive computation of hash function over some random n number of times with initial random seed. Formally, let H(x) be the hash Function which takes x as its random seed. Then the Lamport’s technique computes OTP which is Hn(x)

Hn(x) = Hn(Hn 1(Hn 2(...........H2(H(x))).....)).

The sequence of passwords generated by the above equation is Hn(x), Hn 1(x) ...H2(x),H(x), x.

CHAPTER 6:

TESTING

6.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. The system has been verified and validated by running the test data and live data.

6.2 Levels of Testing

6.2.1 Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In object-oriented programming a unit is often an entire interface, such as a class, but could be an individual method.

For unit testing first we adopted the code testing strategy, which examined the logic of program. During the development process itself all the syntax errors etc. got rooted out. For this developed test case that result in executing every instruction in the program or module i.e. every path through program was tested. Test cases are data chosen at random to check every possible branch after all the loops.

Test case 1: User Interface

Table 6.1: Test cases for user interface

Steps	**Test Action**	**Results**
Step 1	Execute ECC tool	ECC tool successfully launched
Step 2	Client on Generate Key Pair	Key pair successfully generated
Step 3	Click on save keys	Keys saved successfully
Step 4	Click on Encryption	Encryption window opened
Step 5	Click on upload file and select public key	Public key uploaded successfully
Step 6	Click on save cipher text	Cipher text saved successfully
Step 7	Click on decryption	Decryption window loaded successfully
Step 8	Click on upload file and select private key	Private key loaded successfully
Step 9	Execute client tool	Client tool loaded successfully
Step 10	Test all the features of client tool	All the client features are working properly
Step 11	Execute server tool	Server tool loaded successfully
Step 12	Test all the features of server tool	All the server features are working properly

6.2.1.1 User Input

In User Interface the data entry is done by uploading sample text files and accessing various pages in the browser

6.2.1.2 Error Handling

In this system we have tried to handle all the errors that occurred while running the application. The common errors we saw were reading a tuple with an attribute set to null and database connection getting lost.

For Testing we used Top-Down design a decomposition process which focuses as the flow of control, at latter strategies concern itself with code production. The first step is to study the overall aspects of the tasks at hand and break it into a number of independent modules. The second step is to break one of these modules further into independent sub modules. One of the important features is that each level the details at lower levels are hidden. So unit testing was performed first and then system testing.

6.2.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors with an interface. This testing is done with simple data and developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance.

Steps to perform integration testing:

Step 1: Create a Test Plan
Step 2: Create Test Cases and Test Data
Step 3: Once the components have been integrated execute the test cases
Step 4: Fix the bugs if any and re test the code
Step 5: Repeat the test cycle until the components have been successfully integrated

Table 6.2: Test cases for integration testing

Name of the Test	Integration testing
Test plan	To check whether the system works properly when all the modules are integrated.
Test Data	Any sample messages that are used for communication between client and server.

6.2.3 System testing

Ultimately, software is included with other system components and the set of system validation and integration tests are performed. System testing is a series of different tests whose main aim is to fully exercise the computer-based system. Although each test has a different role all work should verify that all system elements are properly integrated and formed allocated functions.

Table 6.3: Test cases for Input-Output

Name of the Test	**System Testing**
Item being tested	Over all functioning of GUI with all functions properly linked.
Sample Input	Any sample messages that are used for communication between client and server.
Expected Output	All the modules like login, execution, etc
Actual Output	Application reacts to user inputs in expected manner.
Remarks	Successful

6.2.4 Validation Testing

At the culmination of black box testing, software is completely assembled is as a package. Interfacing errors have been uncovered and the correct and final series of tests, i.e., validation tests begins. Validation test is defined with a simple definition that validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

6.2.5 Output Testing

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system is required tests the output displayed or generated by the system under consideration. The output format is considered in two ways, one is on screen format and the other is printed format. The output format on the screen is found to be corrected as the format was designated in the system has according to the user needs. As for the hard copy the output comes according to the specification requested by the user. The output testing does not result in any correction in the system.

6.2.6 Test data and Output:

Taking various kind soft data plays a vital role in system testing. After preparing the test data system under study is tested using the test data. While testing, errors are again uncovered and corrected by using the above steps and corrections are also noted for future use.

6.2.7 User acceptance Testing:

User acceptance testing of the system is the key factor for the success of the system. A system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

6.2.8 GUI Testing:

GUI testing is use to ensure the visual clarity of the system, flexibility of the system, user friendliness of the system. The various components which are to be tested are:

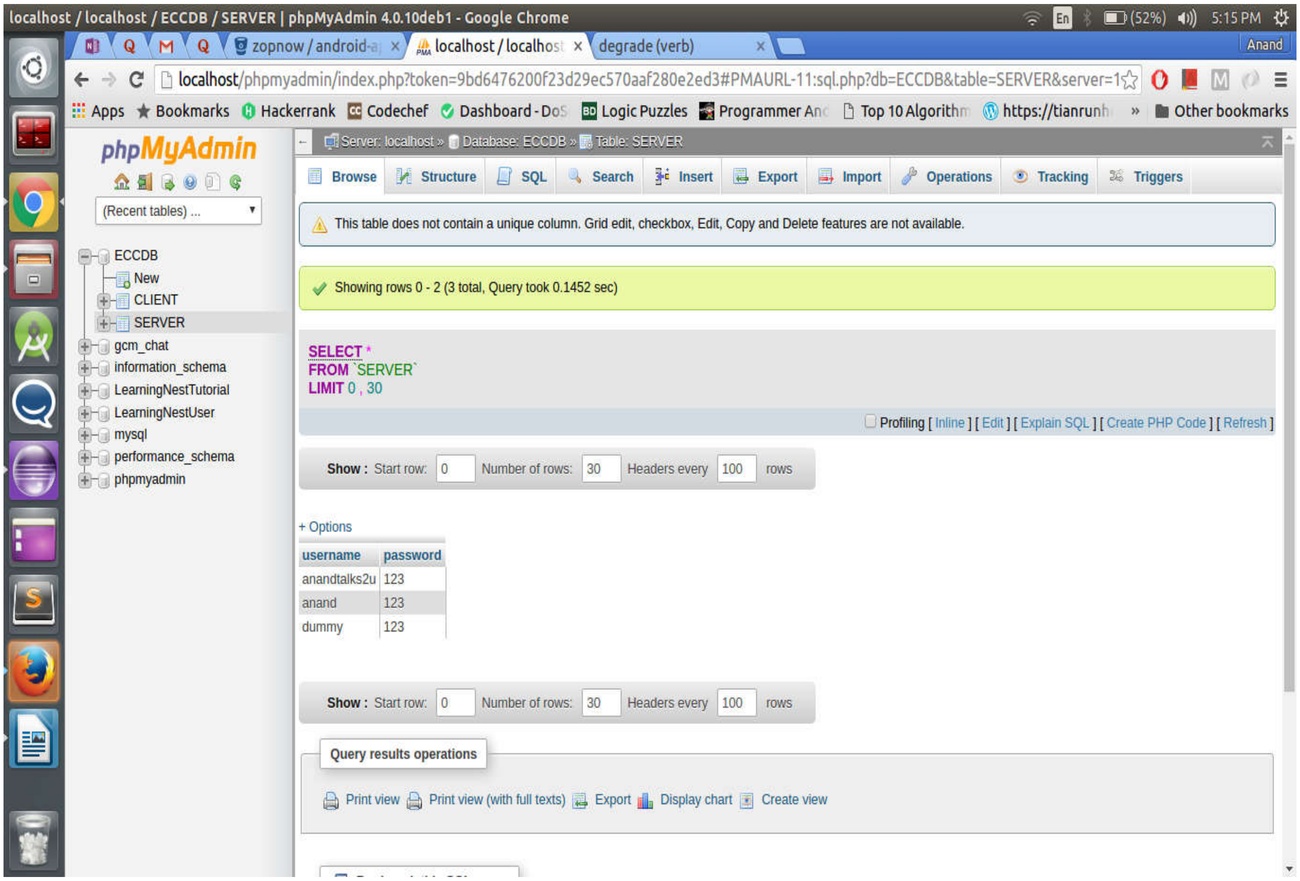
- Relative layout

- Various Links and Buttons

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Output Screen


localhost / localhost / ECCDB / SERVER | phpMyAdmin 4.0.10deb1 - Google Chrome
5:15 PM
localhost/phpmyadmin/index.php?token=9bd6476200f23d29ec570aaf280e2ed3#PMAURL-11:sql.php?db=ECCDB&table=SERVER&server=1
Apps
Bookmarks
Hackerrank
Codechef
Other bookmarks
phpMyAdmin
(Recent tables) ...
ECCDB
New
CLIENT
SERVER
gcm_chat
information_schema
LearningNestTutorial
LearningNestUser
mysql
performance_schema
phpmyadmin
Server: localhost » Database: ECCDB » Table: SERVER
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Tracking
Triggers
Profiling [Inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Show : Start row: 0 Number of rows: 30 Headers every 100 rows
+ Options
Query results operations
Print view
Print view (with full texts)
Export
Display chart
Create view

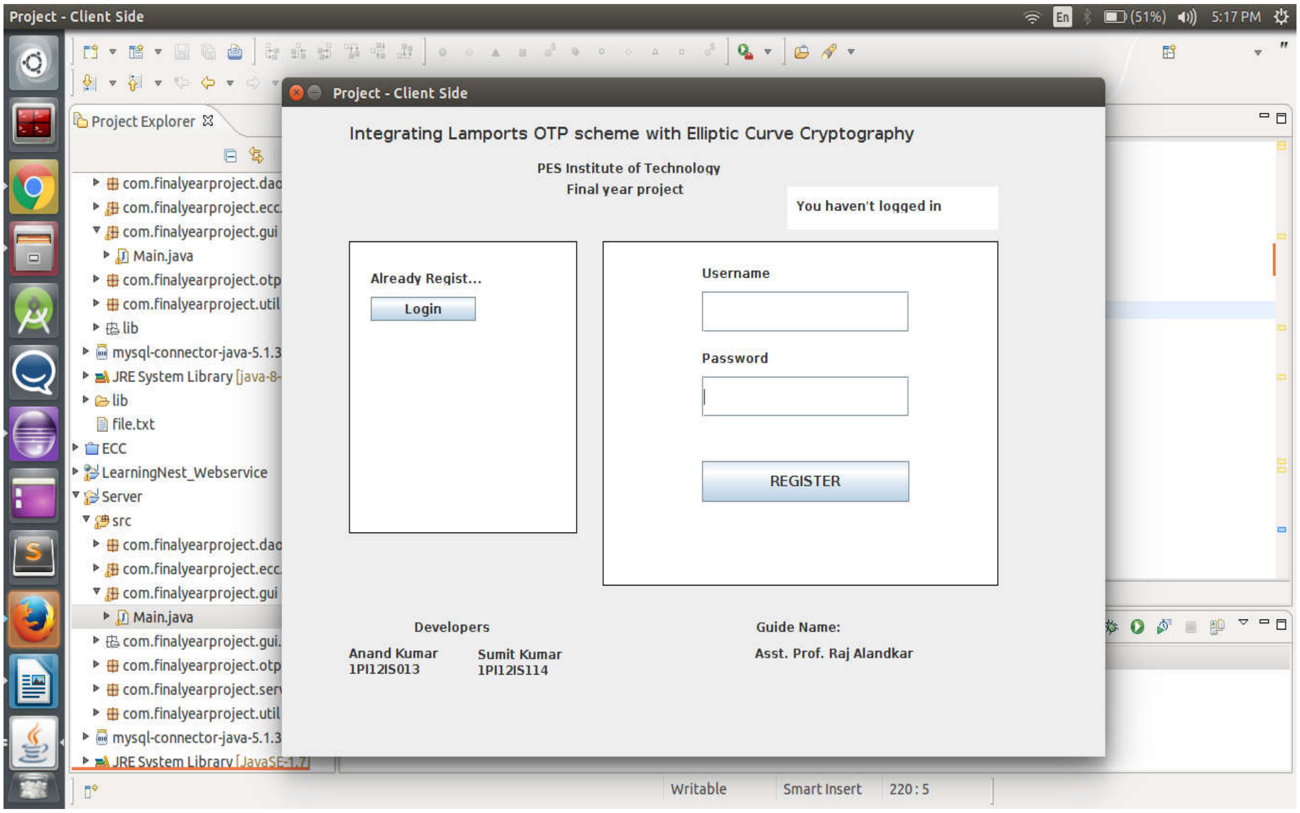
This table does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 2 (3 total, Query took 0.1452 sec)

```
SELECT *
FROM `SERVER`
LIMIT 0 , 30
```

username	password
anandtalks2u	123
anand	123
dummy	123

Fig 7.1 Database for Client and Server



Project - Client Side

Integrating Lamports OTP scheme with Elliptic Curve Cryptography

PES Institute of Technology
Final year project

You haven't logged in

Already Regist...

Username

Password

Developers

Anand Kumar
1PI12IS013

Sumit Kumar
1PI12IS114

Guide Name:
Asst. Prof. Raj Alandkar

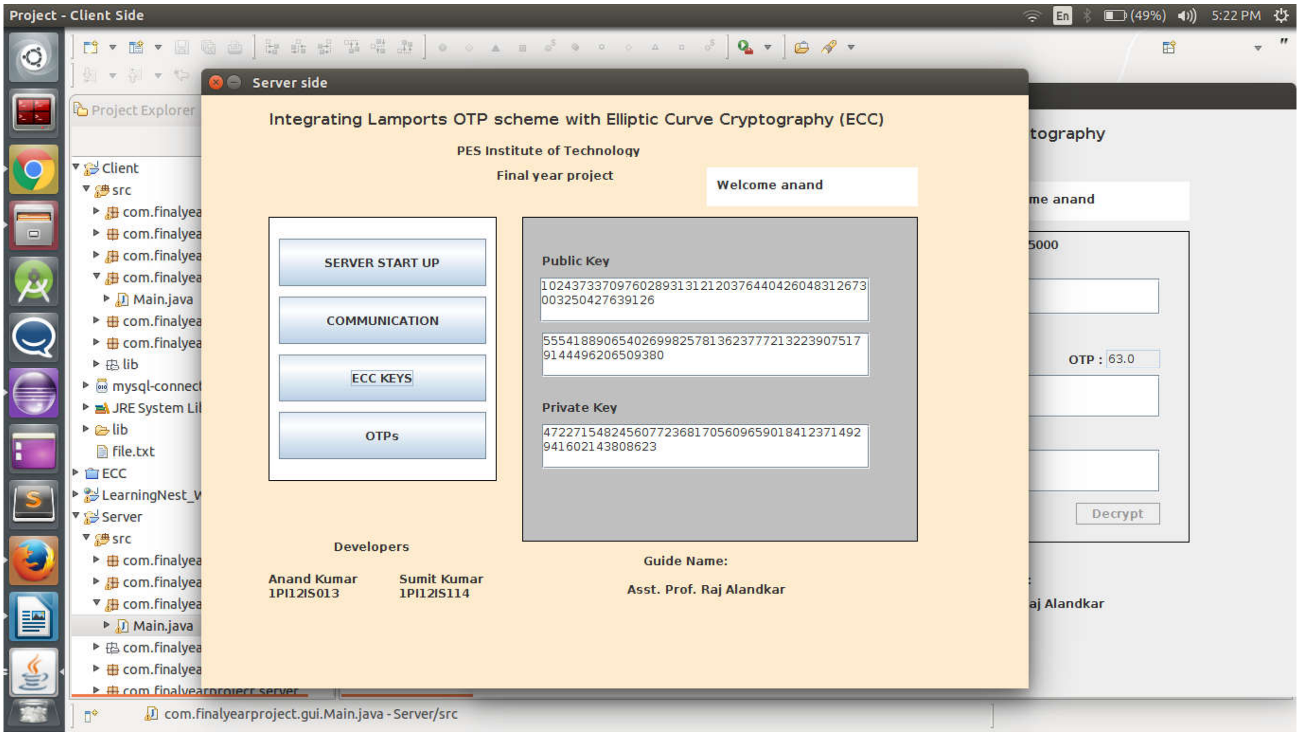
Fig 7.2 Client Side UI

Client/Server Steps to login or Register:

1) New user(Client/Server) register with username and password
2) Once registered successfully, will be allowed to login through login panel
3) After entering valid login details user is given access to the communication panel

Initiating Socket connection:

1) Server starts the connection on a particular port no.
2) Client needs to provide IP Address and port no on which server is started.
3) Client gets connected to Server, they are ready to communicate with each other



Integrating Lamports OTP scheme with Elliptic Curve Cryptography (ECC)

PES Institute of Technology

Final year project

Welcome anand

Public Key

1024373370976028931312120376440426048312673
003250427639126

5554188906540269982578136237772132239075 17
9144496206509380

Private Key

4722715482456077236817056096590184123714 92
941602143808623

Developers

Anand Kumar 1PI12IS013

Sumit Kumar 1PI12IS114

Guide Name:

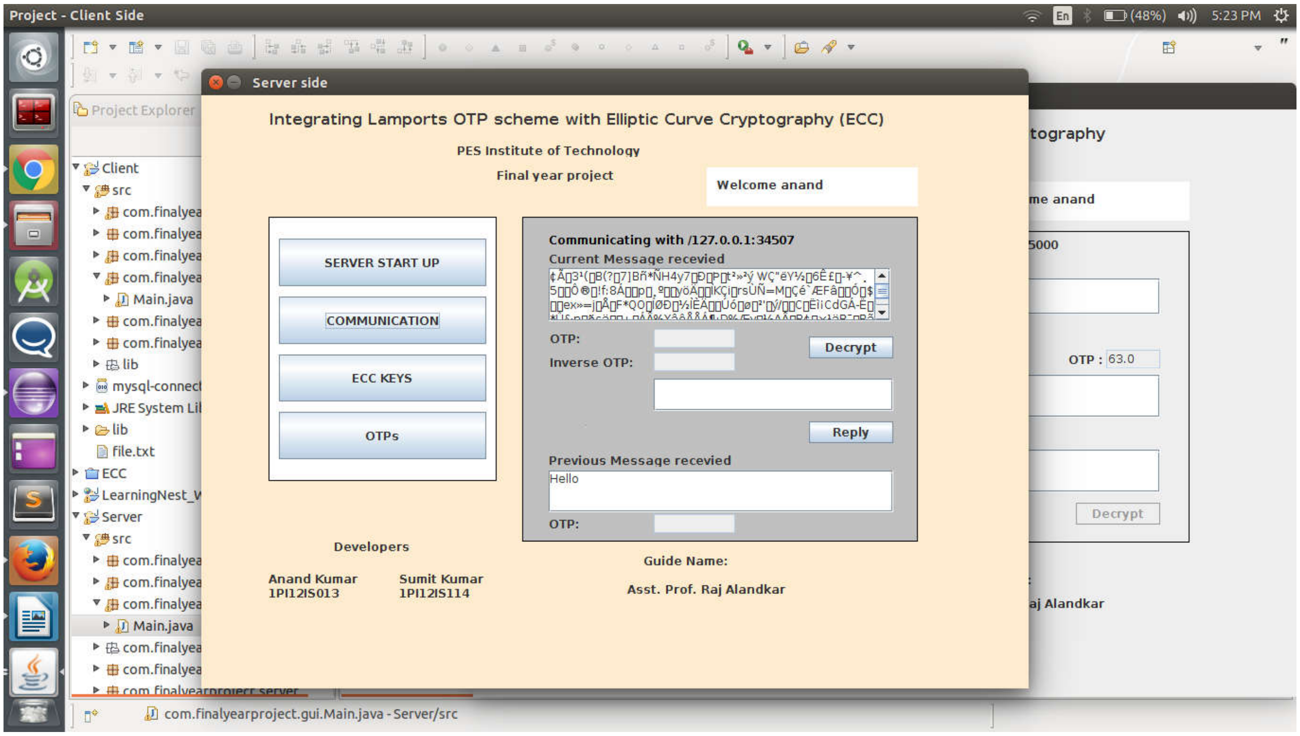
Asst. Prof. Raj Alandkar

OTP : 63.0

Fig 7.3 Key Generation

Encryption and Decryption:

1) Private and Public Keys are used to encrypt the message.
2) This message is received as a cipher text on the other side.
3) Cipher text is decrypted and we get back the original message.



Integrating Lamports OTP scheme with Elliptic Curve Cryptography (ECC)

PES Institute of Technology

Final year project

Welcome anand

Communicating with /127.0.0.1:34507

Current Message recevied

```
¢Ã□3¹(□B(?□7]Bñ*ÑH4y7□Ð□P□t²»²ý WÇ"ëY½□6Ê£□-¥^.
5□□Ô®□!f:8Á□□p□,º□□yöÄ□□IKÇi□rsÙÑ=M□Çé`ÆFâ□□Ó□$
□□ex»=j□Å□F*QO□ÍØÐ□½ÌÈÄ□□Uó□ø□²'□ý/□□C□ËìiCdGÁ-Ë□
```

OTP:

Inverse OTP:

Previous Message recevied

Hello

OTP:

Developers

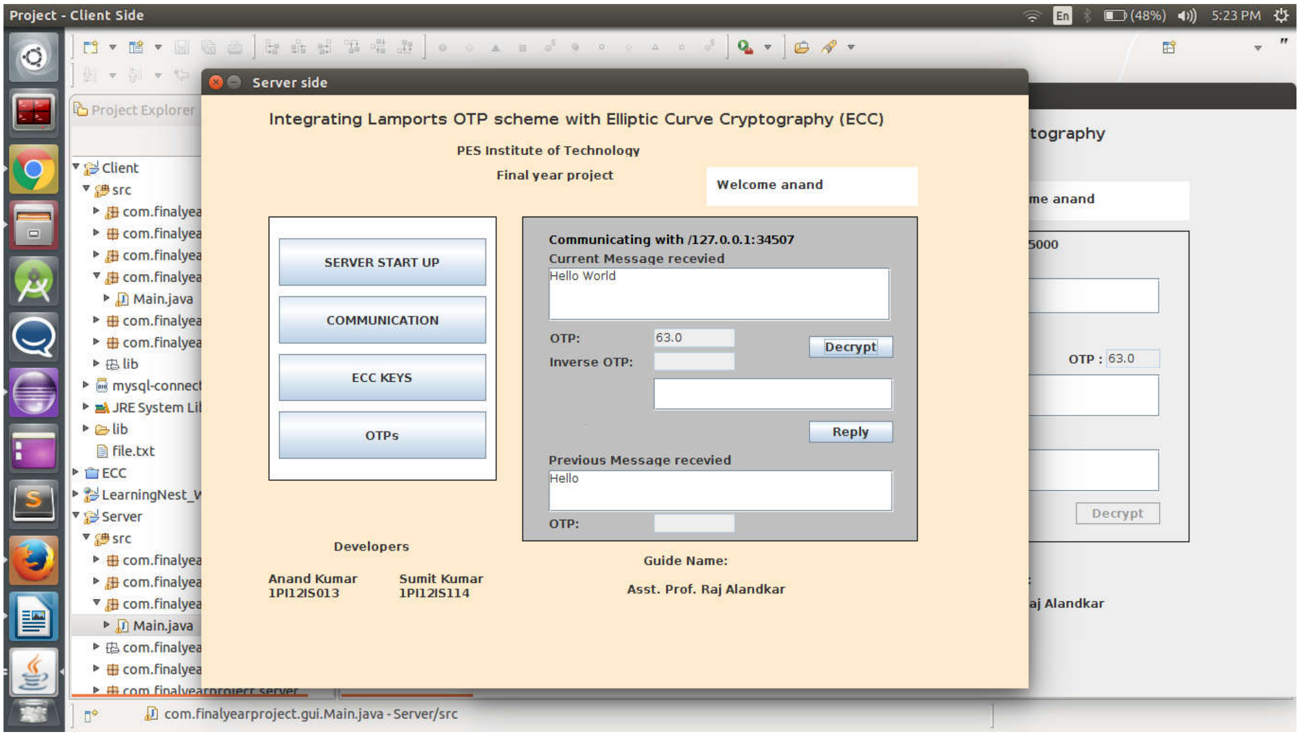
Anand Kumar
1PI12IS013

Sumit Kumar
1PI12IS114

Guide Name:

Asst. Prof. Raj Alandkar

Fig 7.4 Encrypted Message



Integrating Lamports OTP scheme with Elliptic Curve Cryptography (ECC)

PES Institute of Technology

Final year project

Welcome anand

Communicating with /127.0.0.1:34507

Current Message recevied

Hello World

OTP: 63.0

Inverse OTP:

Previous Message recevied

Hello

OTP:

Developers

Anand Kumar
1PI12IS013

Sumit Kumar
1PI12IS114

Guide Name:

Asst. Prof. Raj Alandkar

Fig 7.5 Decrypted Message and OTP

CHAPTER 8

CONCLUSION

In this project we have reviewed the existing OTP schemes used for end-to-end authentication in IoT and have proposed a lightweight, robust and scalable OTP scheme by using the principles of IBE-ECC. Since we do not store the keys, key size is small and do not depend on the previous keys (memory less), our scheme requires lesser resources for operation as compared to the existing schemes such as HOTP, TOTP, Bicakci et al., Yeh et al., Lamport's hash based algorithm and Chefranov and Goyal et al, etc. We have demonstrated that our proposed scheme with a smaller key size and lesser infrastructure performs on par with the existing OTP schemes, without compromising the security level. Since our scheme requires less resources and the key size is smaller as compared to the existing schemes, it can be viewed as a prominent candidate for large and diverse IoT systems such as Smart City, Smart Home and Smart Infrastructure deployments. As part of our future work, we are in the process of deploying our proposed scheme on a real IoT platform such that real-time performance evaluation can be obtained.

CHAPTER 9

FUTURE ENHANCEMENT

- Currently this algorithm is generating keys for NIST_P_192, It can be extended to run on more complex curves such as NIST_P_224, NIST_P_256, and so on**.**

- Since this is a prototype for a better and safer communication hence we will aim for the client to be embedded devices and in such cases battery consumption needs to be taken care.

- As part of our future work, we are in the process of deploying our proposed scheme on a real IoT platform such that real-time performance evaluation can be obtained

CHAPTER 10

BIBLIOGRAPHY

[1] J. Antonio J, L. Latif, and S. Antonio, “The internet of everything through ipv6: An analysis of challenges, solutions and opportunities,” in Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, ser. JoWUA ’13. Innovative Information Science & Technology Research Group, 2013, pp. 97–118.

[2] G.-M. Oscar, K. Sandeep S, H. Sye, Loong Keoh Rene, and S. Rene, “Security considerations in the ip-based internet of things,” in IETF Draft-garcia-core-security-06, ser. Internet Draft ’14. IETF, 2014, pp. 1–45.

[3] A. Hiltgen, T. Kramp, and T. Weigold, “Secure internet banking authentication,” in IEEE Security and Privacy, 2006, pp. 21–29.

[4] M. Parikshit N, A. Bayu, P. Neeli R, and P. Ramjee, “Identity authentication and capability based access control (iacac) for the internet of things,” in Journal of River Publications. River Publishers, 2013, pp. 1–40.

[5] L. Chen-Xu, L. Yun, Z. Zhen-Jiang, and C. Zi-Yao, “The novel authentication scheme based on theory of quadratic residues for wireless sensor networks,” in International Journal of Distributed Sensor Networks. Hindawi, 2013.

[6] N. Huansheng and L. Hong, “Directed path based authentication scheme for the internet of things,” in Journal of Universal Computer Science, 2012, pp. 1112–11 131.

[7] C. Schmitt and B. Stiller, “Two-way authentication for iot,” in IETF, ser. ACE Working Group ’14. IETF, 2014, pp. 1–19.

[8] L. Leslie, “Password authentication with insecure communication,” in Communications of the ACM, ser. J.UCS ’12. New York, NY, USA: ACM, 2012, pp. 770–772.

[9] G. Zhao, X. Si, J. Wang, X. Long, and T. Hu, “A novel mutual authentication scheme for internet of things,” in Modelling, Identification and Control (ICMIC), Proceedings of 2011 International Conference on, June 2011, pp. 563–566.

[10] V. Cakulev, G. Sundaram, and I. Broustis, “Ibake: Identity-based authenticated key exchange,” in RFC 6539, ser. Informational ’12. IETF, 2012, pp. 1–13.

[11] M. Parikshit N, A. Bayu, P. Neeli R, and P. Ramjee, “Novel threshold cryptography-based group authentication (tcga) scheme for the internet f things (iot),” in 7th IEEE ANTS. IEEE, 2013, pp. 1–6.

[12] D. M’Raihi, S. Machani, and J. Rydell, “Hotp: An hmac-based onetime password algorithm,” in IETF RFC 4226, ser. Network Working Group ’05. IETF, 2005, pp. 1–37.

[13] D. M’Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, “Totp:time-based one-time password algorithm,” in IETF RFC 6238, ser. Informational ’11. IETF, 2011, pp. 1–16.

[14] K. Mijin, L. Byunghee, K. Seungjoo, and W. Dongho, “Weaknesses and improvements of a one-time password,” in International Journal of Future Generation Communication and Networking, 2009, pp. 29–38.

[15] Y. Huang, Z. Huang, H. Zhao, and X. Lai, “A new one-time password method,” vol. 4. Elsevier, 2013, pp. 32–37.

[16] E. Mohamed Hamdy, K. Muhammad Khurram, and A. Khaled, “Onetime password system with infinite nested hash chains,” in Communications in Computer and Information Science, ser. Security Technology, Disaster Recovery and Business Continuity Book Chapter’05. Springer,2005, pp. 161–170.

[17] D. Boneh, “Twenty years of attacks on the rsa cryptosystem,” in Notices of the AMS, ser. AMS’99. AMS, 1999, pp. 1–16.

[18] M. Christopher, “One-time password scheme via secret sharing techniques,” in Master of Science Thesis. University of New Orleans, 2011, pp. 1–50.

[19] V. Goyal, A. Abraham, S. Sanyal, and S. Han, “The n/r one time password system,” in Proceedings of International Conference on Information Technology: Coding and Computing, ser. ITCC’05. IEEE, 2005, pp. 733–738.

[20] X. Wang, H. Yu, W. Wang, H. Zhang, and T. Zhan, "Cryptanalysis on hmac/nmac-md5 and md5-mac," in Advances in cryptology- EUROCRYPT 2009. Springer, 2009, pp. 121–133.

[21] K. Jongsung, B. Alex, P. Bart, and H. Seokhie, "On the security of hmac and nmac based on haval, md4, md5, sha-0 and sha-1," in SCN 2006. LNCS, Springer-Verlag, 2006, pp. 1–18.

[22] F. P.A., L. G, and N. P.Q, "Full key-recovery attacks on hmac/nmacmd4 and nmac-md5," in CRYPTO '07. Springer, Dec 2007, pp. 1–18.

[23] L. Yunjin and K. Howon, "Insider attack-resistant otp (one-time password) based on bilinear maps," in International Journal of Computer and Communication Engineering, 2013, pp. 304–308.

[24] A. Shamir, "Identity-based cryptosystems and signature schemes," in Proceedings of CRYPTO 84 on Advances in Cryptology. NewYork, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 47–53.