ONE TIME PASSWORD BASED LOCK SYSTEM

By

NC Bukhtawar Umbreen

NC Ayesha Bashir

NC Faiza Murtaza

PC Sehrish Ajmal Malik

Project Supervisor

Brig. Fahim Arif

Submitted to the Faculty of Electrical Engineering, Military College of Signals National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E. Degree in Telecom Engineering

JUNE 2014

CERTIFICATE OF CORRECTNESS AND APPROVAL

Certified that work contained in this thesis titled “ONE TIME PASSWORD BASED LOCK SYSTEM”, carried out by Bukhtawar Umbreen, Ayesha Bashir, Faiza Murtaza and Sehrish Malik under the supervision of Brig. Fahim Arif for partial fulfillment of Degree of Bachelors of Telecommunication Engineering, is correct and approved.

Approved By

Brig. Fahim Arif

EE Department

Military College of Signals, NUST

Dated: June 2014

ABSTRACT

ONE TIME PASSWORD BASED LOCK SYSTEM

The One Time Password Lock System project is an implementation to ensure safe authentication by verifying end-users' identities. This document gives the design and implementation of a system that authenticates the user on the basis of two factors i.e. a static PIN (something that you know) and token generating time-based one-time passwords (something that you possess). A time-based one-time password is valid for only one login session and changes after some time interval that is fixed.

The user will carry the portable token which is a hardware device based on Arduino platform and implements RFC-6238 standard defined by OATH to generate time-based one-time passwords (TOTP) valid or 60 seconds. When the user comes near the door lock, these passwords will be communicated with the server using wireless Bluetooth communication protocol. In this way, at the authentication end on the door, server will be synchronized with the token. The user will first enter the static PIN and then use the password displayed on the token to authenticate to the system. This combination of information on Token and PIN will form the basis of two-factor authentication to ensure safe authentication.

DECLARATION

No portion of work presented in this thesis has been submitted in support of another award or qualification either in this institution or anywhere else.

In The Name Of Allah, the Most Benevolent, the Most Merciful.

DEDICATION

This dissertation is dedicated to my mother, father and siblings for their support throughout the project, to my supervisor for putting his faith in us and my friends specially Ayesha Bashir without whose collaboration this project wouldn't have been possible.

-Bukhtawar

Dedicated to My Family, My Supervisor, Bukhtawar and all my friends who stood by me, when times were really hard. It couldn't have been possible without all their love and support.

-Ayesha

This is dedicated to My Parents, supervisor and friends for their endless support.

-Faiza

Dedicated to my beloved parents, siblings, nephews and nieces for their perseverance and endurance throughout the project's tenure..and especially to my father and my friends for their unfailing faith in me that pushed me forward at every step..

- Sehrish

ACKNOWLEDGEMENTS

All praises for Allah Almighty who gave us the strength to accomplish this mighty task despite numerous difficulties and hardships on our way.

We offer our utmost gratitude to our Supervisor, Brig. Fahim Arif for his constant guidance and encouragement. The project wouldn’t have been possible without his expert advice, unfailing patience and invaluable efforts.

We are indebted to Mr. Saeed Anwar from University of Engineering and Technology for taking out time through their busy schedules and helping us throughout the development of our project.

A special thanks to our friends for lively discussions, exchange of ideas and source of encouragement during the entire project. We are also grateful to our parents, family and well-wishers for their admirable support. Lastly, we acknowledge the Lab attendants of EE Lab who rendered their help during the period of our project work.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Bterm	Bluetooth Terminal
HMAC-SHA1	Hash-based Message Authentication Code-(Secure Hashing Algorithm 1)
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
NIST	National Institute of Standards and Technology
COM (port)	Communication Port
OATH	Open Authentication
PIN	Personal Identification Number
RAM	Random Access Memory
RFC-6238	Request for Comments: 6238
RS-232	Recommend Standard number 232
OTP	One Time Password
TOTP	Time-based One Time Password
FIPS	Federal Information Processing Standard
MAC	Message Authentication Code
Ipad	Inner Pad
Opad	Outer Pad
LCD	Liquid Crystal Display
PCB	Printed Circuit Board

Chapter 1: Introduction

1.1 Project Overview

1.2 Problem Statement

1.3 Approach

1.4 Objectives

1.5 Limitations

1.6 Organization of thesis

Chapter 1: Introduction

This chapter gives an explanatory overview of the project i.e. where exactly does the problem lie in safely authenticating a user and how these problems are addressed by the proposed project. It then explains the project objectives, limitations and finally organization of the thesis at the end of this chapter.

1.1. Project Overview

A static password/PIN is definitely not an unbreachable form of authentication to a system because it does not address the issues related to identity theft and acts such as impersonating an authorized user (spoofing). The project aims at designing and developing a hardware token based on Arduino prototyping platform for generating and displaying time-based one-time passwords and authenticating a user using it.

1.2. Problem Statement

One Time Password based Lock System is the solution to the corporate computer and hardware security problems faced in safe user authentication. Access to any system can be controlled on the basis of many authorization and authentication techniques. The requirement of user is safe authentication by verifying end-users' identities. For user identity authentication two steps are followed: identification of the user and validation of its identity. Every other security system, nowadays, relies only on static passwords. A password is such a key to a system that is something you know and could be compromised easily by hacking tools. According to security experts in industry and academia, static passwords pose very serious threats and are the most compromisable factor. More sophisticated systems today use an additional factor in addition to passwords to make the password attack difficult.

Unsafe authentication can pose the following threats:

a) Spoofing: which is an act to impersonate an authorized user and enter/login to a network which is unauthorized for them. Trying to compromise a user's identity may have a motive as serious as trying to perform an act of terrorism, stealing sensitive information, performing robbery etc

b) Unauthorized access can also pave a way to capture all the information in a network at a particular

1.3. Approach

NIST (National Institute of Standards and Technology) specifies that an un-breachable authentication would satisfy two factors, where one of these factors (atleast) should be immune to replay attack. Since the only technology showing immunity towards replay attacks is one-time passwords (OTPs), thus the "One-Time Password Lock System" employs smart hardware token generating OTPs to authenticate a user. Their specialty is that they expire after a short time interval adding a new layer of protection and making it extremely difficult for unauthorized users to illegally access any network/system.

The project entitled "One Time Password Lock System" is a representation of a security system, offering authorization on the basis of two factors; a PIN that you know and a token that you possess that is capable synchronizing itself to the authentication server by the wireless transmission of the codes. Figure 1-4 illustrates the approach adopted by

“One Time Password Lock System” to address the problems faced by networks/systems today

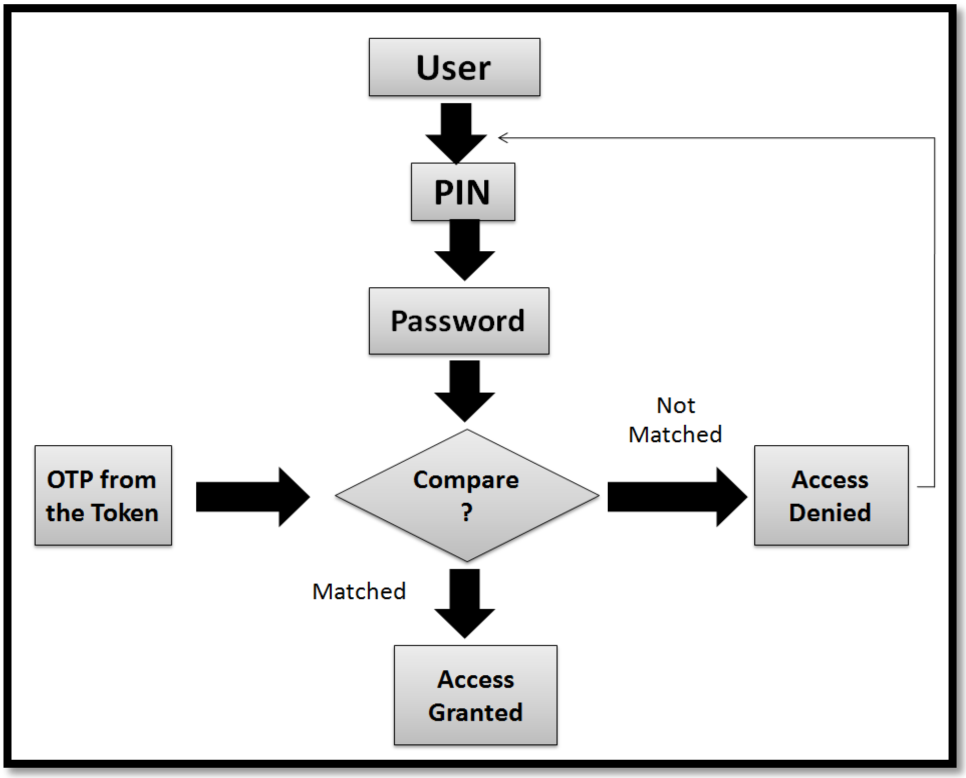


Figure 1-1: Approach of One Time Password Lock System

The server end will prompt the user to enter the static PIN of 6 digits. At the correct entry, the server will start receiving the codes and user will be asked to enter the one-time password displayed on the token. With both the factors correctly entered the server will drive the door open thus granting access to the user.

1.4. Objectives

“One-Time Password Based Lock System” safely authenticates the user on the basis of two factors: a static password of 9 digits followed by a one-time password of 6 digits. This password will be valid for 60 seconds.

The project meets the following user requirements of:

- Strong and multi-factor authentication
- Live generation of passwords valid for one session
- Protection against replay and brute force attacks
- Safe mode enabled in Bluetooth communication

1.5. Limitations

Any kind of communication can be susceptible to attacks. Many ways of compromising Bluetooth security have been developed, and the problem only promises to become worse as Bluetooth devices pervade everyday life.

Bluetooth communication may be jammed if a strong jammer is used so any other form of wireless communication that is prone to jamming and interception can be used to enhance the security to a greater level.

1.6. Organization of thesis

Chapter1 It has already given a brief overview of “One Time Password Lock System”, the problems faced in user authentication to a network and the solution that project has come up with.

Chapter2 Presents the review of literature carried throughout the tenure of the project and lists some of the current projects regarding One Time password tokens.

Chapter 3 Technological requirements i.e. hardware and software specifications are discussed in this chapter.

Chapter4 Gives overview of One Time Password Lock System’s design and explains in detail the development carried out

Chapter5 Analyses and evaluates the performance of One Time Password Lock System on this basis of results of various tests carried out. Basing on the performance benefits of having such a system is also discussed.

Chapter 6 Discusses some future enhancements and finally concludes the chapter

At the end of the document are the appendixes giving application codes in Arduino development language. The list of reference material used while preparing this document follows the Appendixes.

Chapter 2: Literature Review

Chapter 2: Literature Review

Technological advancements are taking place all over the world. Over the recent years, the idea of having one-time password based login has gained immense popularity in developed countries. This chapter gives a gist of the study done during the period of project i.e. websites/books/conference papers/patents etc referred. The chapter also gives details about similar projects done elsewhere in world, their limitations and finally how these issues are addressed by "One Time Password Lock System".

2.1. Strong Authentication Solutions

User authentication ensures that a user does not get an illegal/unauthorized access to the system or networks' possessions that are not meant for it. In the most general form of authentication, passwords are used to authenticate a client/server. These password-protected systems are widely deployed in areas of banking, academia and many governmental organization accesses **[1]**. The basic dilemma of the password-protected system is that a memorized password can be guessed/ searched/ hacked by an attacker. Nucleus Research and KnowledgeStorm highlighted the susceptibility of passwords in a research conducted in October (2006). The results showed that 1 out of every 3 people pen down their computer passwords, dejecting their own security **[2]**.

2.2. One Time Passwords

Technology is reducing all the inefficiencies of work performed by humans with its feet paving more rapidly and taking less significant time. To prevent or reduce the damage caused by spyware & phishing attacks; many security-sensitive industries like banks, Colleges/Universities, governmental departments are deploying *one-time password* systems, where each user makes use of every password only once. If at any time a password is breached, it will cause a very limited damage as one password can only be used to impersonate the user once **[3]**.

OTP generation makes use of randomness/ pseudo-randomness. This is necessary because we can't afford the prediction of later OTPs by observation of former ones. Various kinds of approaches are used for OTP generation; which are listed below:

- Generated on the basis of **time-synchronization** between the user and authentication server and providing with limited time passwords **[4].**

  The merit of time synchronized protocols is that it does not require complex calculations or a certification authority but only a counter to maintain the synchronization. On the other hand to achieve tolerance, the actual time has to be separated into time slots and pair the valid passwords for each time slot. However, the algorithm becomes ineffective in cases where multiple simultaneous authentications are attempted on a single time slot **[5]**.

- Generation of OTPs by a mathematical **algorithm** in which every new password is **based on the previous password** (chaining of OTPs).

- Generation of OTPs by a mathematical **algorithm** in which every new password is **challenge-based** and/or a counter.

There are many different applications of OTPs e.g. systems/networks employing use hardware security tokens/cards that is possessed by the user for the generation of OTPs.

Some other networks consist of authentication softwares running on user's cell phone. Some networks/systems produce OTPs on the authentication server side and launch them to the user using SMS (out-of-band channel) messaging. Finally, in some other networks/systems, paper tokens are used in which the password is printed on the paper in the form of a grid, which the user must carry with him/her **[4]**.

2.2.1. HMAC SHA1

Message authentication codes (MACs) come into use when two parties, sharing some secret key information, communicate. Cryptographic hash based MAC functions are called HMAC.

HMAC is specified in a Federal Information Processing Standard (FIPS). It calculates and verifies the MAC on the basis of secret key information. Definition of parameters of HMAC is shown below **[6]**:

B Input block size (bytes) to SHA-1 i.e. B = 64

H SHA-1

ipad Byte x’36’ repeated B no. of times

K Secret key, between the two communicating parties

K0 In order to form a B byte key, K is appended with zeros

L Output block size (bytes) of SHA-1 i.e. L = 20

opad Byte x’5c’ repeated B no. of times

t The number of bytes forming the MAC

text message on which HMAC is computed; the maximum value of length of data (in bits) depends on the hash algorithm used

x'N' Hexadecimal notation; 'N'= 4 bits

|| Concatenation

⊕ Exclusive-or

Cryptographic Keys

Key size (K) should be equal to or greater than L/2, where L is the size of output from SHA1. For keys longer than B-bytes, key is first hashed using SHA-1 and then the resulting L-byte string is used as the HMAC key, K. Keys are selected at random using a FIPS-approved key generation method. The keys are protected in the same manner in which consistency of data to be authenticated is protected**[6]**.

Truncated Output

The output of HMAC function L is truncated. To use a truncated HMAC output, the t leftmost bytes of the result is used as the MAC code. The output length, L, should be atleast 4 bytes (i.e., $4 \leq t \leq L$). and 't' shall be at least L/2 bytes (i.e., $L/2 \leq t \leq L$ **[6].**

HMAC Specification

To compute a MAC over any data using the HMAC function, the following operation is performed **[6]**:

$$\mathbf{MAC(text)_t = HMAC(K, text)_t = H((K0 \oplus opad\ )||\ H((K0 \oplus ipad)\ ||\ text))_t}$$

Table 2-1: Step by Step Process in HMAC algorithm

STEPS	DESCRIPTION
Step #1	If size of K = B, then K0 = K. Skip the next 2 steps
Step #2	If size of K > B, K=SHA-1(K): result is L byte K
Step #3	If size of K < B, create a B-byte string K0 by appending zeros to K
Step #4	K0 ⊕ ipad
Step #5	(K0 ⊕ ipad) || text
Step #6	SHA-1((K0 ⊕ ipad) || text)
Step #7	K0 ⊕ opad
Step #8	(K0 ⊕ opad) || SHA-1((K0 ⊕ ipad) || text)
Step #9	SHA-1((K0 ⊕ opad)|| SHA-1((K0 ⊕ ipad) || text))
Step #10	Select the leftmost t bytes of the result as the MAC

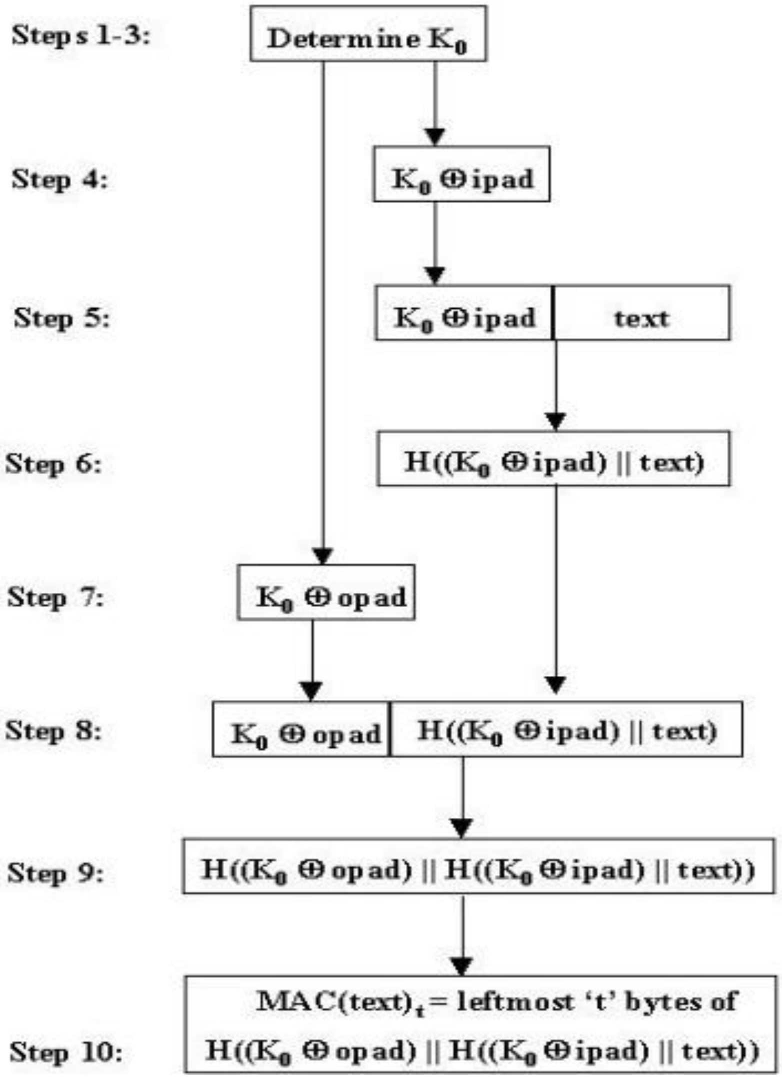

Steps 1-3:
Determine K_0
Step 4:
$K_0 \oplus$ ipad
Step 5:
$K_0 \oplus$ ipad
text
Step 6:
$H((K_0 \oplus ipad) || text)$
Step 7:
$K_0 \oplus$ opad
Step 8:
$K_0 \oplus$ opad
$H((K_0 \oplus ipad) || text)$
Step 9:
$H((K_0 \oplus opad) || H((K_0 \oplus ipad) || text))$
Step 10:
$MAC(text)_t$ = leftmost 't' bytes of
$H((K_0 \oplus opad) || H((K_0 \oplus ipad) || text))$

Figure 2- 1: Diagramatic explanation of HMAC

2.2.2. Security

The motivation to use one-time passwords for authentication is that the breach of one password won't affect the whole session. The one-time password mutually authenticates the client and the server. For TOTPs we don't have any other long-term values like public keys or certificates. Knowledge of the shared password is the basis to provide authentication. Banks, governments, and corporate virtual private networks (VPNs) have deployed One-time password systems to reduce the effects of password compromise.

Bank customers nowadays keep the sheets of paper to list down the One time passwords.. Hardware one-time password generators are used by the online shoppers and gamers. If these passwords are not being used safely and securely, the money being spent on deploying them is a total waste **[3]**.

2.3. Smart Token as a factor of authentication

One-time passwords are deployed in various forms like electronic tokens, a chip-and-pin card in combination with a reader device, or on sheets of paper as some European banks do **[3]**. Basically one-time password schemes can be classified into the following four categories:

Based on the mathematical algorithm

By using the one-way hash chain in 1981, Lamport proposed the one-time password authentication strategy. If an indefinite series of passwords is required, then we need a new seed value after the exhaustion of the old hash. If a password file is maintained to verify the user's authentication request this increases the risk of tampering as well as maintenance cost. Therefore, many researchers [7] [8] [9] [10] [11] have proposed various user authentication schemes using smart card to overcome these short comings.

Based on the smart card

Smart cards have been widely adopted in many remote authentication schemes as they are tamper-resistance and convenient in managing a password file. However, carrying the cards and the reader is a responsibility to the user. The card and the reader are far from ubiquitous, thus this barrier has restricted their application in authentication schemes.

Based on the time-synchronized token

The time-synchronized one-time passwords are usually implemented as physical hardware tokens. Inside the token is an accurate clock that has been synchronized with the clock on the authentication server.

Based on the Short Message Service (SMS)

SMS is a ubiquitous that is it can reach any where, and is available in all handsets. SMS is a best effort delivery, the phone company tries to deliver it, but do not guarantee it's delivery, or if it does the time it will take to reach destination is not known. Therefore it is important that one-time passwords have a time to live as a security feature. Moreover, the SMS based scheme, incurs some extra charges. Thus, it is impractical and not a low total cost solution [12].

An, *security token*, *identity token*, *access token*, or simply *token*, is a physical device that aids authentication. Bankcard, remote garage door opener, or smart cards can be a secure storage device containing passwords. Token can be an active device that yields *one-time pass codes*, either *time-synchronous* (changing in synchrony with a master at the host) or *challenge–response* (responding to a one-time challenge). It is a portable, secure storage device accessed at the client end. Password is used to obtain a pass code that is transmitted to the host for authentication. A *pass code* is a secret number that is machine-generated or machine-stored, so it can be lengthy, more random, and thus changing [13].

Types of token:

There are a number of options available in the market that could act as a token. They can be broadly categorized in 3 classes.

a) **Paper token:** it can be a disseminated list of OTPs, or a crisscross of codes the user uses to enter in response to any challenge.

b) **Soft tokens:** they rely on a software algorithm/application running on the client's computer.

c) **Hardware–based tokens:** they include physical and rational mechanisms to guard their data and prevent photocopying.

Paper-based tokens are easy to compromise. Any soft token is also insecure unless it is associated to some hardware (computer hardware token), but it will limit portability. For choice of a secure two-factor authentication mechanism it is therefore suggested that a hardware token should be used.

Types of hardware tokens

Hardware tokens can be further sub-divided into three categories.

a) **Disconnected tokens:** they generate authentication information which the user uses manually in the authentication process. They have no logical connection to any user's computer.

b) **Connected tokens:** physical connection exists with the user's computer.

c) **Contactless tokens:** they are logically (but not physically) connected to the user's computer

d) Most disconnected tokens rely on OTP technique. For each use, a new password/ PIN are generated, which is valid only for one login session and is derived cryptographically. Connected tokens reduce the user interaction, thus making the process simpler and reducing the chances of human error. However, the issue is of connectivity because not all computers support smart card readers. In our present community, a connected hardware token is a suitable choice, but for online purposes a disconnected is the best choice.

e) Contactless tokens have all the advantages of connected tokens, but there is still the need to have the appropriate contactless card/token reader [14].

Strengths and Weaknesses of One-time Password Tokens

An OTP token offers ***strong*** defense as it can store or create a pass codes valid for a very short interval and thus incur less risk of being guessed randomly that is it is safe against brute force attacks. The user must remember to carry the physical object, as it is *weak* in defending against theft.

There are three possible locations where Authentication token can be attacked: at the client, in the transmission channel, and at the host.

Client Attack: High strength pass codes from lower strength passwords are generated by OTP token. Authenticator to the server (password or PIN) prior to the token, should be strong so that it can’t be easily guessed.

Eavesdropping, Theft, and Copying Attacks: As one session’s pass code is useless in another session, token that generates a one-time pass code will not succumb to this type of attack [13].

The tolerable cost of an authentication system is application dependent. The last future works suggested by Kemal Bicakci and Nazife Baykal is designing an OTP protocol with a stronger authentication goal [15].

2.4. Bluetooth Protocol

Bluetooth is a protocol for wireless communication of data over small distances. The devices using Bluetooth are physically close.

Every Bluetooth transceiver IC is embossed with a unique 48-bit (12-digit hexadecimal value) MAC address (device identification address). For two devices to communicate using it, the inquiring device must have some way of determining the other device’s MAC address [16]. The Bluetooth address is used at all layers Bluetooth protocol [17]. The Bluetooth address is used at all layers of the Bluetooth communication process, from the low-level radio protocols to the higher-level application protocols [16].

The device supports wireless communication up to 10 m (or up to 100 m if the transmitter’s power is increased). Devices need no line of sight communication as such but the devices have a limited connection distance. Bluetooth data rates are comparable to

1–2 Mb/s which is absolutely enough for file sharing applications. Its key strength lies in its ability to concurrently handle both data and voice communication [18].

Architecture overview:

Bluetooth chip implements the RF, baseband and link management portions of the Bluetooth specification. It performs radio transmission, reception and digital signal processing functions. Some other important functions comprise connection establishment, data (asynchronous) & voice (synchronous) connection support, error rectification, and confirmation. [18].

Most Bluetooth devices operate at 2.4GHz which is globally available and a license-free band. This bandwidth is reserved for industrial, scientifical and medicinal use all around the World. As the radio band is available to all kinds of users within the specified regulations, the unwanted interference becomes unpredictable. Interference is, thus, a very important issue for Bluetooth users. [19].

Connection Process

Based on three progressive states a Bluetooth connection between two devices is developed [17]:

1. Inquiry – When two Bluetooth devices know nothing about each other, an inquiry must be run to try to discover the other. One device sends out the inquiry request, and any device that is listening for such request responds with its address, name and other information.

2. Paging (Connecting) – The process of forming a connection between two Bluetooth devices is known as paging. Before this connection can be initiated, the devices must know each other’s address (which is found in the inquiry process).

3. Connection – After completion of paging process the device enters connection process. When device is connected, it can either be actively participating or it can be put into a low power sleep mode.

    a) Active Mode – This is the regular connected mode, in this mode device is actively transmitting or receiving data.

    b) Sniff Mode – Here the device is less active and is in the power saving mode. It’ll sleep and only listen for transmissions at a set interval (e.g. every 100ms).

    c) Hold Mode – Hold mode is temporary and power-saving. A device sleeps for a defined period and then returns back to active mode after an interval. The master commands a slave device to hold [17].

    d) Park Mode – The deepest of sleep modes is park mode. A master commands a slave to “park”, and that slave will become inactive until the master tells it to wake back up.

Master and Slave Configuration

Bluetooth networks (**piconets**) use a master/slave model to control data sending between devices [17].

Devices have to hop to new frequencies after each new packet transmission; however devices have to agree with each other. The master is the Bluetooth device that sets the FH (frequency hopping) sequence. The slave has to synchronize to the master (in time and frequency). The baseband part of the Bluetooth calculates the FH sequence using the clock and MAC address of the master chip. Slaves transmit data using Time Division Multiplexing (TDM) **[20]**.

In case of one Master and one Slave, the system is a P2P (point-to-point connection). When more than one slave is connected to one master, the system is a P2M (point-to-multipoint). Slaves are not interconnected to each other in any way and one slave can connect to one master at a time.

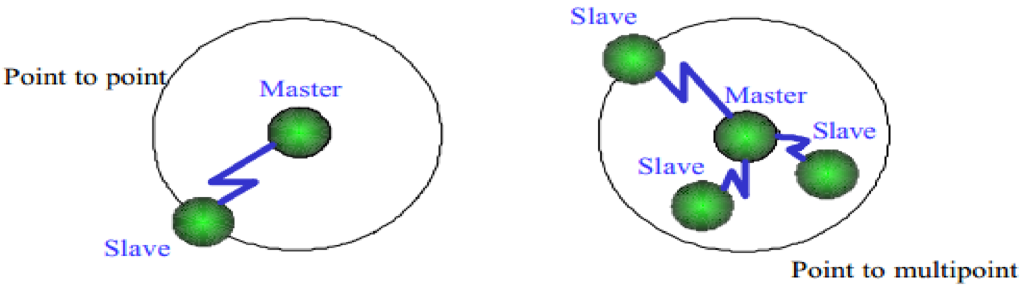


Figure 2-2: Master Slave Piconets

Slaves are bound by their masters and they transmit and receive only from their master. [17]

Security of Bluetooth

To prevent unauthorized usage and eavesdropping in Bluetooth system basic security elements need to be considered though it is mainly intended for short-range connectivity between personal devices. Security features are included at the link level. They are based on a secret link key that is shared by a pair of devices. A pairing procedure is used for the generation of key, when the two devices communication for the first time.

When connection is being established an authentication process is carried out. This verifies the identities of the units involved. Conventional challenge-response routine is used by the authentication process. The claimant produces signed response (SRES) that the verifier compares with its own SRES. Then it decides if the challenger may continue with connection establishment. Eavesdropping on the link is a danger inherent to radio communications to prevent it the payload of each packet is encrypted. Encryption is done by stream ciphers; where the payload bits are modulo-2 added to a binary key stream. Bluetooth is capable of providing limited number of security elements at the lowest level. At higher layers more advanced security procedures can be implemented [19].

Bluetooth Power management

Bluetooth devices operated using batteries; therefore special notice is taken to decrease the power expenditure in the design. Many tests have proved that Bluetooth devices are low power devices and have no negative impact on health [19].

The range and transmission power of a Bluetooth module is defined by its power class:

Table 1-2: Transmit Power Ranges of Bluetooth

Class #	**Max Output Power (dBm)**	**Max Output Power (mW)**	**Max Range**
Class #1	20 dBm	100 mW	100 m
Class #2	4 dBm	2.5 mW	10 m
Class #3	0 dBm	1 mW	10 cm

Some modules operate in only 1 power class, while others are potent of varying their power classes [17].

2.5. Work Done around the Globe

Work Done in Public Server implementations throughout the world

a) Google has implemented TOTP in its "Google Authenticator" application. The project includes implementations of OTPs for several platforms like Android, iOS, Blackberry etc.

b) Facebook uses TOTP for its "Login approval"

c) Dropbox has enabled the OTP technology for account access

d) Gemalto develops smart cards; integrating OTP strong authentication in either Java™ or Microsoft .NET systems.

e) OCBC Bank of Singapore makes the online banking safer through the introduction of OCBC Token

“One-Time Password Door Lock” project is being implemented on Arduino board, which is open source prototyping platform. It integrates many small independent projects like Bluetooth serial communication, mechanical lock opening and cryptographic algorithm implementations into one big project to form a complete system representation of safe authentication. Unlike present OTP authentication systems which only counteract fraudulent internet attacks, this system can be implemented to achieve not just that but also used to provide two-level protection to remote systems that require extreme care like bank lockers, entrance to a high security organization etc. It also differs in the aspect that the user will be always asked for entry of PIN (which he secretly knows) and then prompted to enter the password displayed by the token (which only he possesses).

Much of the work can be found on linking the hardware tokens to online banking, social websites and other networking applications all over the world, whereas application of such a system is still lagging in Pakistan. With this project we hope to see safer authenticating systems/networks/banking solutions soon in Pakistan.

List of similar Projects in MCS and elsewhere:

A similar project was carried out in 2013 in Italy. The difference between that project and our project is that

a) It linked the Google Authenticator application to the one time password token.

b) The one time password was displayed on serial terminal through PC.

c) There was wired communication between the token and the server.

Moreover, no such work regarding one time passwords has been carried out in Pakistan before that too using Arduino platform.

Chapter 3: Technological Requirements

3.1 Introduction to Technological Requirements

3.2 Hardware Requirements

3.3 Software Requirements

Chapter 3: Technological Requirements

This chapter provides a brief overview of the technological requirements for the project "One Time Password based Lock system". For ease of understanding, technological requirements are subdivided into hardware, software and operating systems requirements.

3.1. Introduction to Technological Requirements

"One Time Password based Lock System" is divided into three integral parts:

1. Smart Token
2. Authenticating Server
3. Bluetooth Communication between Token and Server

An Arduino UNO R3 board, LCD and an HC-05 Bluetooth module are integrated together to produce a token whereas an authenticating server incorporated the use of an Arduino Mega 2560 R3 board, LCD, 4x3 keypad, HC-05 Bluetooth module and a servo motor sg-90. Communication between the two boards has been wirelessly done via HC-05 Bluetooth module. All programming has been done on Arduino IDE 1.0.5 for both the server and the token.

The technological requirements are sub-divided into hardware, software and operating systems requirements and are discussed in the following sub-sections. Figure 3-1 shows valid breakdown of Technological Requirements.

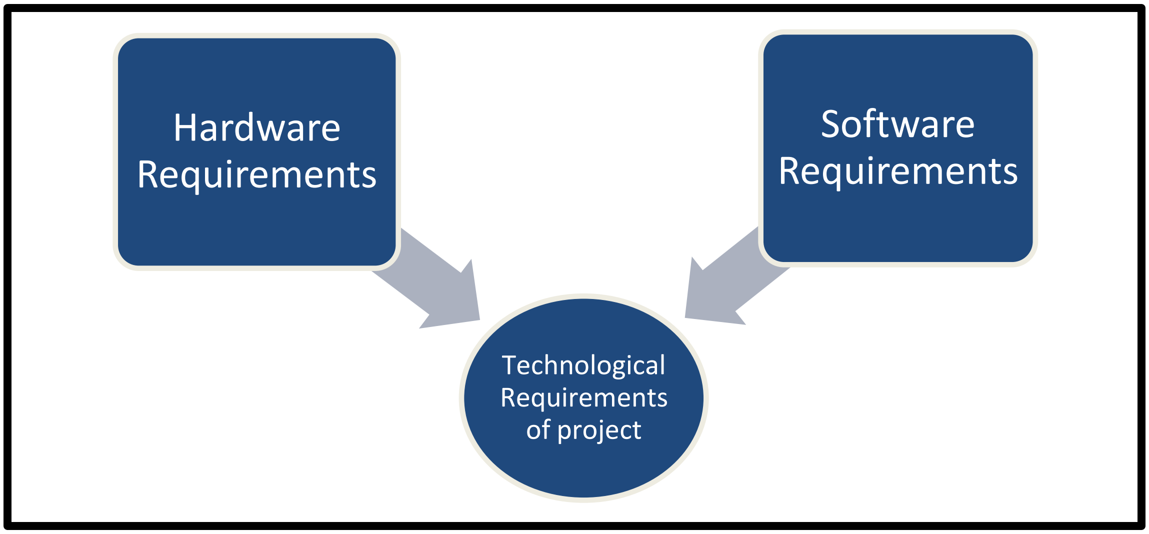


Figure 3-1: Breakdown of Technological Requirements

3.2. Hardware Requirements

The Hardware required for the implementation of the project includes the following.

3.2.1. Arduino UNO R3 Board

The Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

The ATmega328 has 32 KB (with 0.5 KB used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

It has the following specifications.

Table 2-1: Specifications of Arduino UNO

Microcontroller	**ATmega328**
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (out of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

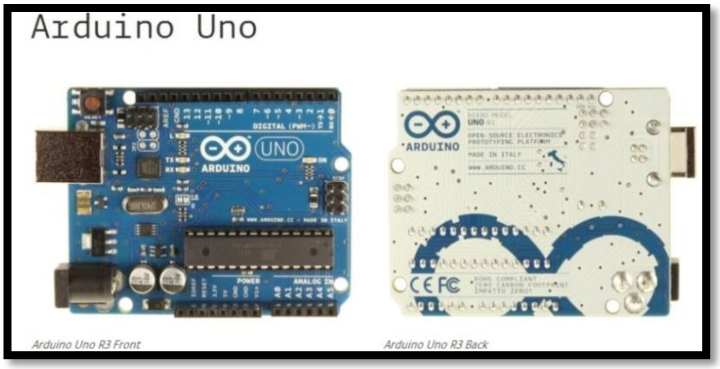


Figure 3-2: Arduino UNO R3

3.2.2. Wireless Bluetooth Transceiver HC-05 RS232 / TTL Base Board

Each HC-05 Bluetooth transceiver module can be transferred to either master or slave mode at any time. Communication between the two Arduino boards is performed via this module. At the token end, it acts as a master while at the server end, it is configured to slave mode. Their synchronous working is shown by the simultaneous blinking of the red LED on both the HC-05 modules. It has the following specifications.

Table 3-3: Specifications of HC-05

Antenna	build-in 2.4GHz antenna
Bluetooth Spec v2.0+EDR Compliant	v2.0+EDR Compliant
Memory	external 8Mbit FLASH
Voltage	3.1V~4.2V
Current	• current in pairing is in the range of 30 ～40mA

	- current in communication is 8mA
USB protocol	Full Speed USB1.1 and compliant with 2.0
Dimensions	(27mm×13mm×2mm)
Emission power	3dBm (≤4dBm, Class 2)
Output Power	-4 to +6dBm
Sensitivity (Bit error rate)	Can reach -80dBm(≤-84dBm at 0.1% BER)
Supports baud rates	2400 -1382400
Modulation	GFSK(Gaussian Frequency Shift Keying)
Security	Authentication and encryption
Working temperature	-20 ~ +75 Centigrade
Speed	Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps

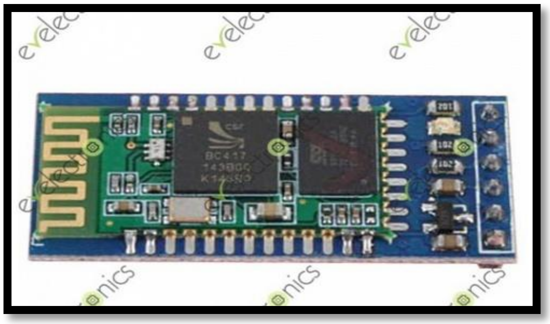


Figure 3-3: HC-05 Bluetooth Transceiver

3.2.3. Servo Motor SG-90

A servo motor is an actuator that accurately performs precise control of position, rotation angle, velocity and acceleration and is highly responsive. It consists of a motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller. By rotating a shaft connected to the engine throttle at an angle of 180 degrees, the servo motor drives a lock open. The wire colors denote:

a) Red = Battery(+)

b) Brown = Battery(-)

c) Orange = Signal

Following are its technical details:

Table 3-4: Technical Details of SG90

Modulation:	Analog
Torque:	**4.8V:** 25.0 oz-in (1.80 kg-cm)
Speed:	**4.8V:** 0.10 sec/60°
Weight:	0.32 oz (9.0 g)
Dimensions:	Length: 0.91 in (23.1 mm)

	Width: 0.48 in (12.2 mm) Height: 1.14 in (29.0 mm)
Motor Type:	3-pole
Gear Type:	Plastic
Rotation/Support:	Bushing
Rotational Range:	180°
Pulse Width:	500-2400 µs

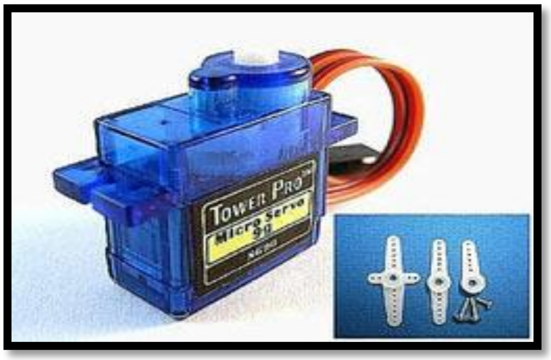


Figure 3-4: SG90 Servo motor

3.2.4 4X3 Keypad

This keypad has 12 buttons, arranged in a telephone-line 3x4 grid. It's made of a thin, flexible membrane material with an adhesive backing so you can attach it to nearly anything. The keys are connected into a matrix, so you only need 7 microcontroller pins (3-columns and 4-rows) to scan through the pad. It includes a 7-pin extra-long header strip so you can plug this into a breadboard with ease.

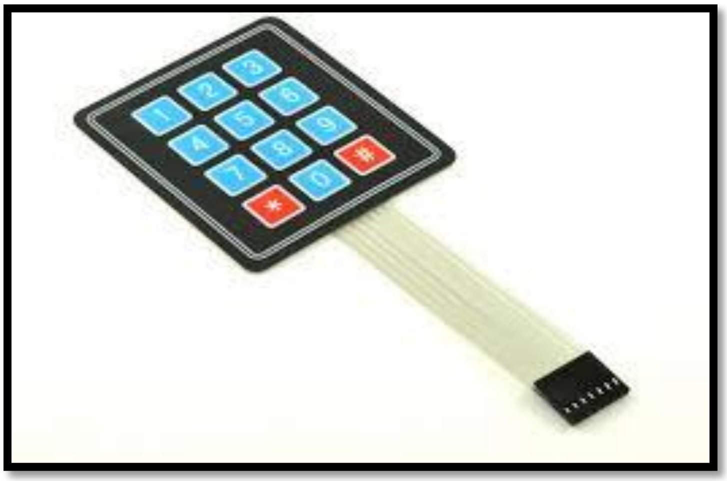


Figure 3-5: 4X3 Membrane Keypad

It has the following technical specifications:

Table 3-5: Specifications of 4x3 Keypad

Weight	7.5 grams
Keypad dimensions	70mm x 77mm x 1mm (2.75" x 3" x 0.035")
Length of cable + connector	85mm
Connector	Dupont 7 pins, 0.1" (2.54mm) Pitch

Mount Style	Self-Adherence
Max. Circuit Rating	35VDC, 100mA
Insulation Spec	100M Ohm, 100V
Dielectric Withstand	250VRms (60Hz, 1min)
Contact Bounce	<=5ms
Life Expectancy	Life Expectancy
Operation Temperature	-20 to +40 °C

3.2.5 16X2 LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments) and animations.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Figure 3-6: 16X2 LCD

Pin Description:

Table 3-6: Pin Description of 16x2 LCD

Pin No	**Function**	**Name**
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment (through a variable resistor)	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low when write to the register; High when read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable

7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Features:

a) 5X8 dots with cursor

b) 16 char *2 lines display

c) 4 bit or 8-bit MPU interfaces

d) Built in controller

e) Display mode and backlight variations

f) ROHS compliant

Table 3-7: Features of 16x2 LCD

Module size	80.0(L)mm *36.0 (W)mm *max 13.5(H)mm
Supply voltage for LCD	3.0V
Input Voltage	3.1-3.5 (typically 3.3V)
Supply Current	1.5mA (max 2.5)
Backlight supply voltage	3.0 V

3.2.6 Arduino MEGA 2560 R3 Board

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (15 of which can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. It features the ATmega16U2 programmed as a USB-to-serial converter, instead.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM.

Table 3-8: Specifications of Arduino MEGA 2560

Microcontroller	**ATmega2560**
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

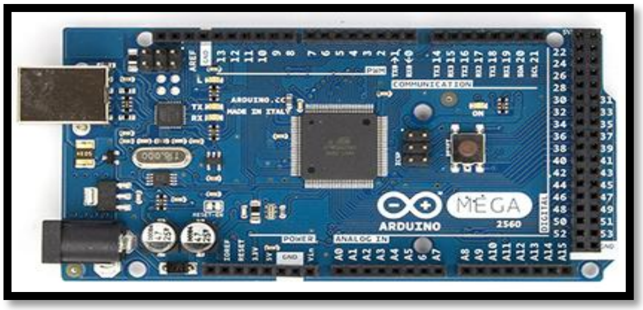


Figure 3-7: Arduino MEGA 2560 R3 Board

3.2. Software Requirements

3.2.1. Arduino IDE 1.0.5

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. The Arduino software consists of an integrated development environment (IDE) and the core libraries.

The IDE is a programming environment that has been written in Java and is based on the Processing development environment. The core libraries are written in C and C++ and compiled using AVR-GCC and AVR Libc.

Chapter 4: Hardware Design and Development

4.1 Architectural Design of One Time Password Lock System

4.1.1 System Block Diagram

4.1.2 Logical Distribution of Project

4.1.2.1 One Time Password generating Token

4.1.2.2 Wireless Communication

4.1.2.3 Authenticating Server

Chapter 4: Design and Development

In this chapter we will discuss the methodology and approach leading to the development of this project. Chapter 3 has already explained in detail the technological requirements.

4.1. Architectural Design of One Time Password Lock System

The project "One Time Password Based Lock System" is the Hardware implementation of Time Based One Time Passwords generating token along with whole lock system. Architectural design of the project can be best understood by the system block diagram and flow chart.

4.1.1. System Block Diagram

Figure 4-1 shows the system block diagram of TOTP Based Lock System. For ease of understanding, the system has been logically subdivided into 3 parts. The first part is programming and hardware development of time based one time passwords generating device (token), Second basic part is establishing wireless communication between token and server and finally programming of server to verify the passwords after their wireless reception.

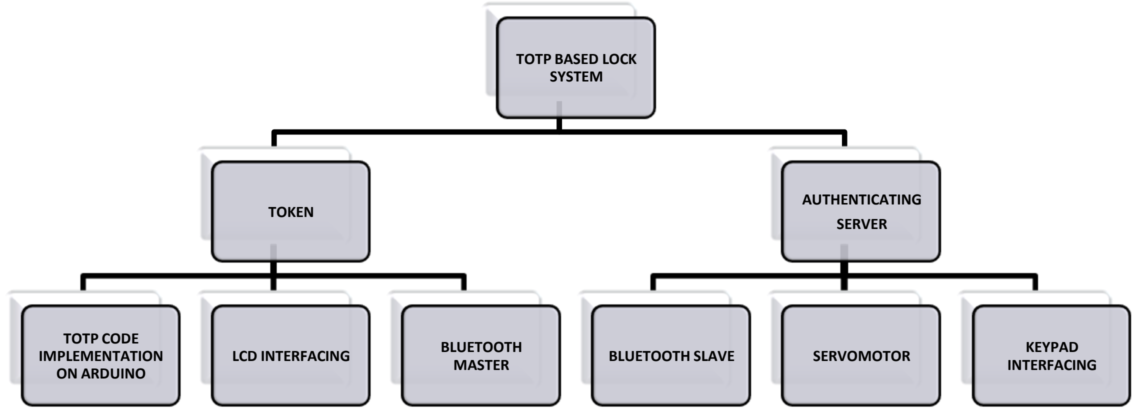


Figure 4-1: System Representation of OTP based Lock System

4.1.2 Logical distribution of project

The project has been divided into three major parts:

4.1.2.1*One-Time Password generating Token*

<u>Software Implementation</u>

One Time Passwords are generated through the HMAC (Hashed Message Authentication Code) SHA-1 Algorithm. SHA 1 is the hashing Algorithm for message authentication codes. The inputs to HMAC algorithm are the UNIX timestamp along with the shared key of the token .The other input to HMAC function will be the number of time steps between the initial counter time T0 and the current UNIX time. This will decide after how much time new password will be generated. The 160 bit output will be truncated through a function. The truncated output will be the one-time password. The token will display this 6 digit code on a LCD interfaced to it. This code will keep changing after every 60 seconds.

Created and imported libraries of interfacing SHA-1("Sha1"), TOTP ("totp") and formulated sketch by combining all these libraries to implement HMAC SHA-1 .This programming is done in Arduino.ide software. The code has been attached in Appendix.

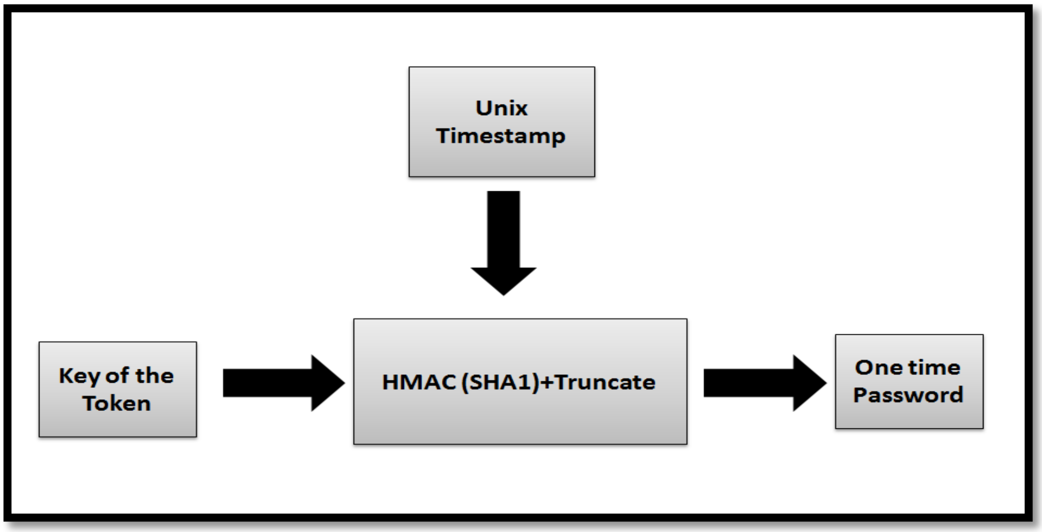


Figure 4-2: Logical explanation of generation of TOTPs

Hardware Implementation

This code has been uploaded on Arduino UNO board and LCD is interfaced with it in order to display the 6 digit password. And also Bluetooth interfaced for sending TOTP.

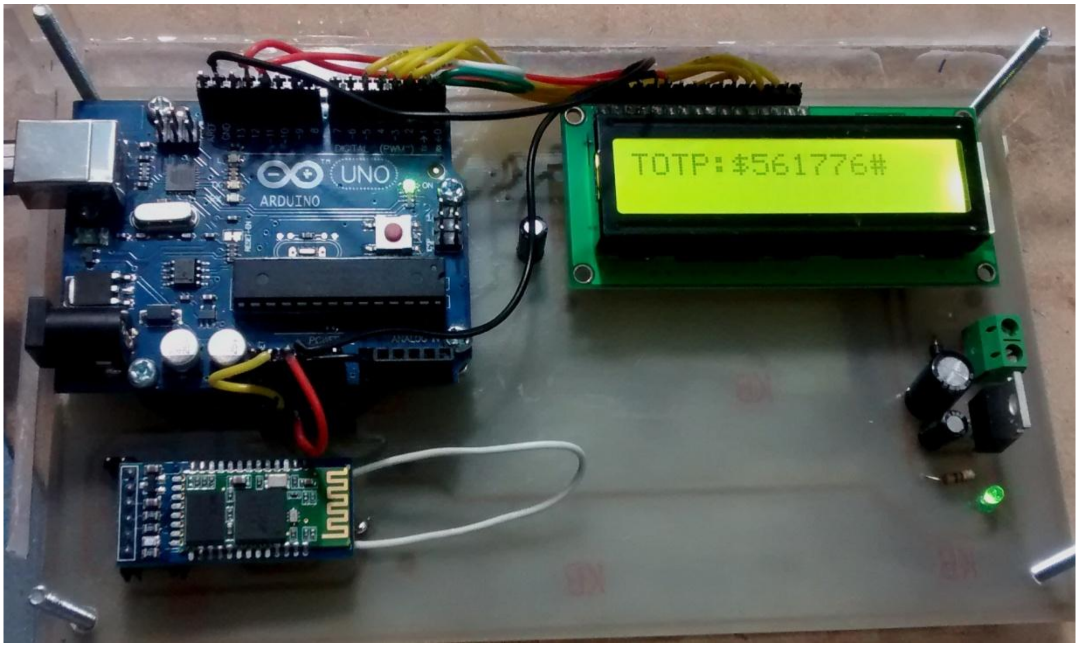


Figure 4-3: Final Display of Token

LCD Interfacing:

16x2 LCD has been interfaced with Arduino boards. Pin configuration for LCD interfacing is as follow:

Table 4-1: Hardware interfacing of LCD

LCD	Arduino UNO
Pin 1	GND
Pin 2	5V
Pin 3	Wiper of potentiometer
Pin 4	Pin 12

Pin 5	GND
Pin 6	Pin 11
Pin 11	Pin 5
Pin 12	Pin 4
Pin 13	Pin 3
Pin 14	Pin 2
Pin 15	GND
Pin 16	5V

Bluetooth module interfacing:

For wireless communication from the Token to the server Bluetooth module HC-05 has been interfaced with the Arduino Board. The pin configuration for this connection is as follow:

Table 4-2: Hardware interfacing of HC-05

HC-05	**Arduino UNO**
Tx	Rx (pin 8)
Rx	Tx (pin 9)

3.3V	3.3V
GND	GND

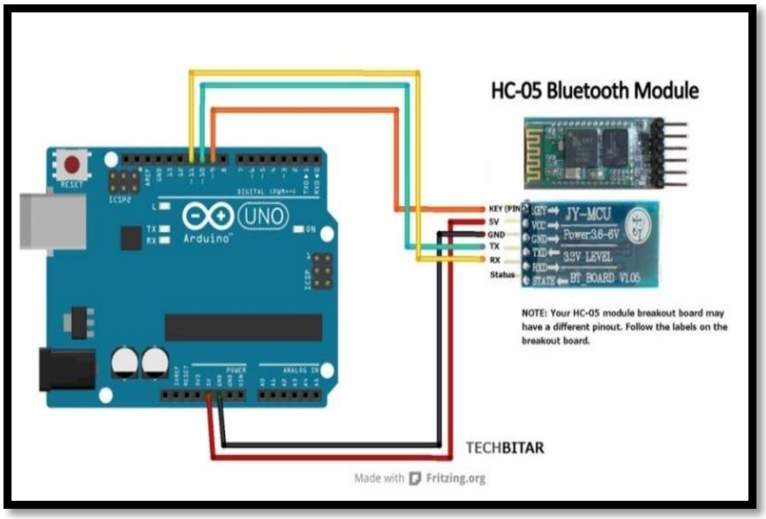


Figure 4-4: Wiring of HC-05 with Arduino UNO

4.1.2.2 Wireless Communication

Wireless link has been established between the token (that generates One Time Passwords) and the authenticating server. Token sends the TOTP (Time Based One Time Passwords) to the server .Wireless link is established through Bluetooth Communications using Bluetooth module HC-05.

Roles in which HC-05 works:

HC-05 is capable of acting as MASTER and SLAVE. When Bluetooth is in MASTER mode it sends data to the SLAVE module it is paired with. Through the AT command modification for the Bluetooth modules we configure them as MASTER or SLAVE.

Work modes of HC-05:

Bluetooth modules work in two basic modes **auto connection work mode** and **command mode.** First we need to bring the Bluetooth module to command mode in order to set it as MASTER for the token as the token has to send the 6 digit passwords. For bringing HC-05 to command mode it is connected to hyper terminal via USB to serial converter then by modifying the AT commands one of the module is set as MASTER and it is paired with the other.

After configuring the Bluetooth the MASTER module is interfaced with the Token and SLAVE module is configured with the Authenticating server.

HC-05 communicates using the RS-232 protocol with a baud rate of 38400.

Figure 4-5: HC-05 BT Transceiver

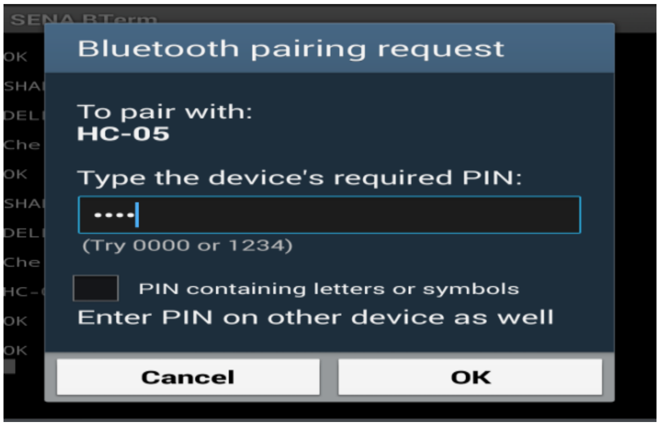
Now the two modules will be known with the following names in the document:

a) MASTER for HC-05 interfaced with token

b) SLAVE for the HC-05 attached with the authenticating server

Bluetooth address of token (MASTER) 20:13:02:20:01:74 and of server (SLAVE) 20:13:05:14:27:59 were found by detecting the HC-05 through SENA BTerm android application.

SENA BTerm is emulator that creates an environment for wireless communication. We installed the application on android phone to detect the HC-05 modules. Later on this application was used to see whether the MASTER module is sending data on the wireless link. When HC-05 is acting as MASTER the application will become a SLAVE and start receiving data from the MASTER.



Bluetooth pairing request

To pair with:
HC-05

Type the device's required PIN:

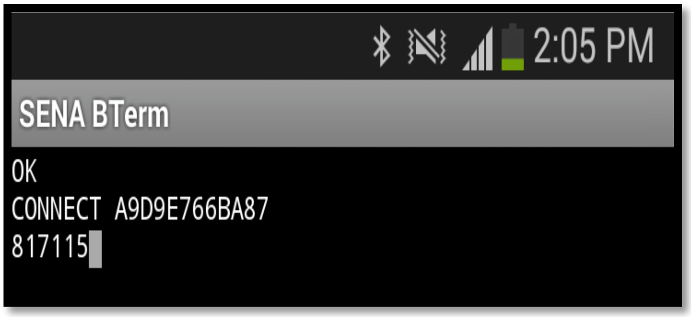
••••

(Try 0000 or 1234)

□ PIN containing letters or symbols

Enter PIN on other device as well

Figure 4-6: Pairing of HC-05 with Android



```
OK
CONNECT A9D9E766BA87
817115
```

Figure 4-7: Displaying result on SENA B TERM

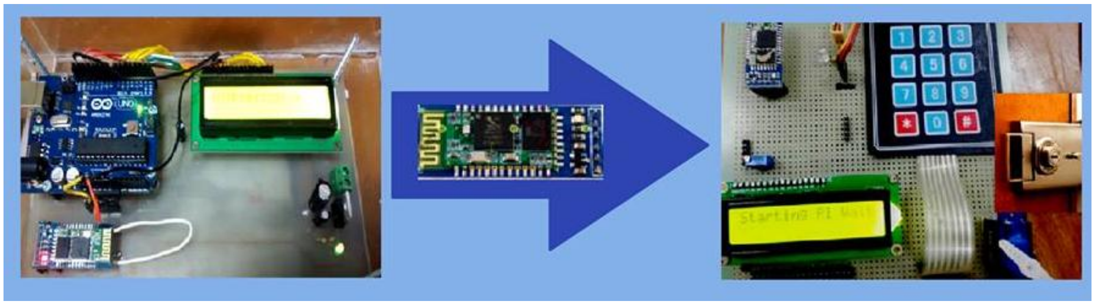


Figure 4-8: Full System Representation

4.1.2.3 Authenticating Server

The third basic part of the project is the development of the authenticating server. To explain this we first describe the software implementation of the server followed by the Hardware.

Software Implementation

Arduino.ide software is used for the programming of the server. Server is programmed to receive the TOTP through the SLAVE interfaced with the SERVER. After the reception of TOTP the server asks the user to enter the PIN that is a 9 digit password unique with respect to a user. Upon verification of this PIN server will ask the user to enter the TOTP that is visible on the LCD of the token which is in possession of the user. The code is then uploaded on the Arduino Mega board.

Hardware Implementation

Hardware to be interfaced with the server:

a) Arduino Mega Board

b) Keypad 3x4

c) LCD 16x2

d) HC-05 Bluetooth module

e) SG-90 servo motor

We have to interface many modules with the server so we use Arduino Mega board as it provides more number of digital pins.

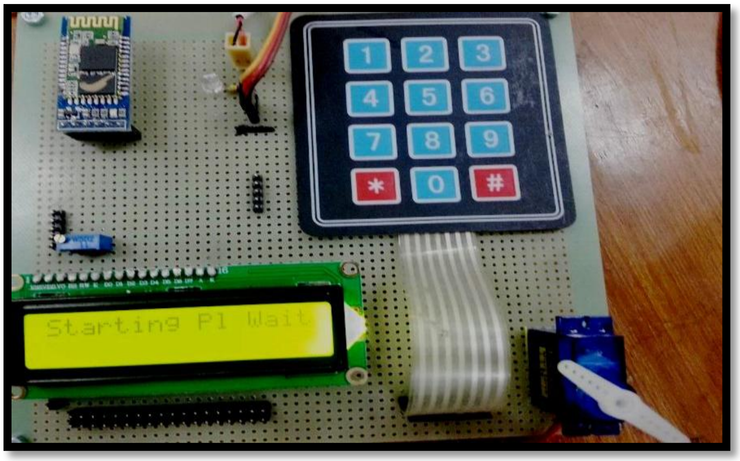


Figure 4-9: Final Server Implementation

Bluetooth interfacing:

HC-05 Bluetooth module is interfaced with the server to receive the TOTP. Address of the SLAVE is already paired with the MASTER while their configurations now they will work in auto connection work mode according to the AT commands set for communication .We use the UART 2 pins of Arduino Mega board i.e digital pin 17 will act as receiver and pin 16 as transmitter other connections are as follow:

Table 4-3: Pin Configurations of HC-05 with Arduino MEGA

HC-05	Arduino MEGA
Tx	Rx (pin 17)
Rx	Tx (pin 16)
3.3V	3.3V
GND	GND

When Bluetooth is turn on it will take a while to get paired and synchronized with the MASTER. Once the connection is established now the link can't be intercepted with the Android applications. The two Bluetooth modules will talk to each other only.

Keypad interfacing:

Keypad is interfaced with the server through which the user will enter the PIN and the TOTP into the server for authentication; the Pin connections for this interface are as follow:

Table 4-4: Pin Configurations of Keypad with Arduino MEGA

Keypad	**Arduino MEGA**
Pin 1	Pin 22
Pin 2	Pin 24
Pin 3	Pin 26
Pin 4	Pin 28
Pin 5	Pin 30
Pin 6	Pin 32
Pin 7	Pin 34

LCD interfacing:

16X2 LCD interfaced with the server so that the user can know when to enter the PIN , is PIN verified or not ,enter the TOTP and then is access granted or denied.LCD is interfaced with the following digital pins of Arduino mega board.

Table 4-5: Pin Configurations of LCD with Arduino MEGA

LCD	**Arduino MEGA**
Pin 1	GND
Pin 2	5V

Pin 3	Wiper of potentiometer
Pin 4	Pin 38
Pin 5	GND
Pin 6	Pin 40
Pin 11	Pin 41
Pin 12	Pin 45
Pin 13	Pin 47
Pin 14	Pin 49
Pin 15	GND
Pin 16	5V

Servo motor interfacing:

Server asks the user to enter the PIN which only he knows, upon its verification the server will prompt the user to enter the TOTP from the token which is in his possession. When the TOTP is verified the server will send pulse to the servo motor interfaced to drive the lock open. The servo will rotate 180 degrees and open the lock and will rotate -180 degree to lock the door again. Connections of SG90 with the Arduino MEGA Board are as follow:

Table 4-6: Pin Configurations of Servo SG90 with Arduino MEGA

SERVO	Arduino MEGA
GND	GND
Pulse	Pin 9
VCC	From external source

The servo draws a lot of current for its operation therefore it is not given power from the board as it may damage the Arduino board. Therefore we provide the Vcc through an external adapter.

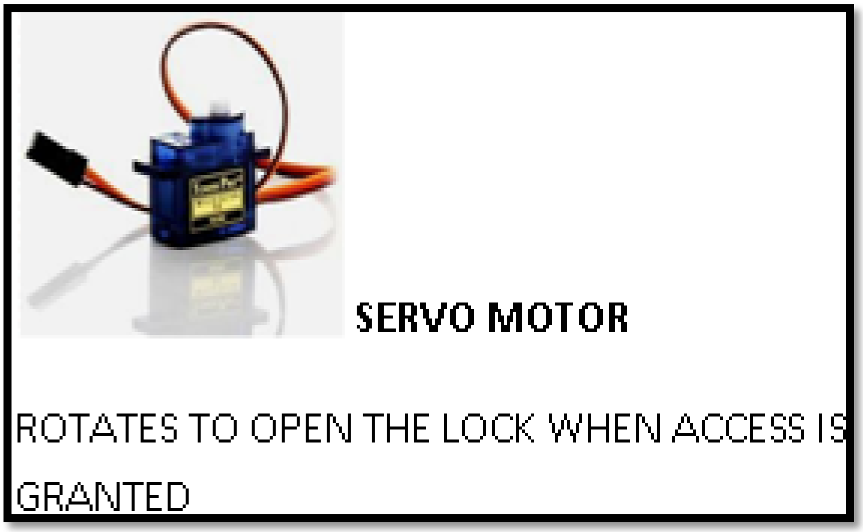


Figure 4-10: Servo Motor

The server is attached outside the door that is locked and the token is in possession of the user. Servo motor is placed with the lock inside the room and opens the lock upon correct entry of TOTP and thus allows success to the user.

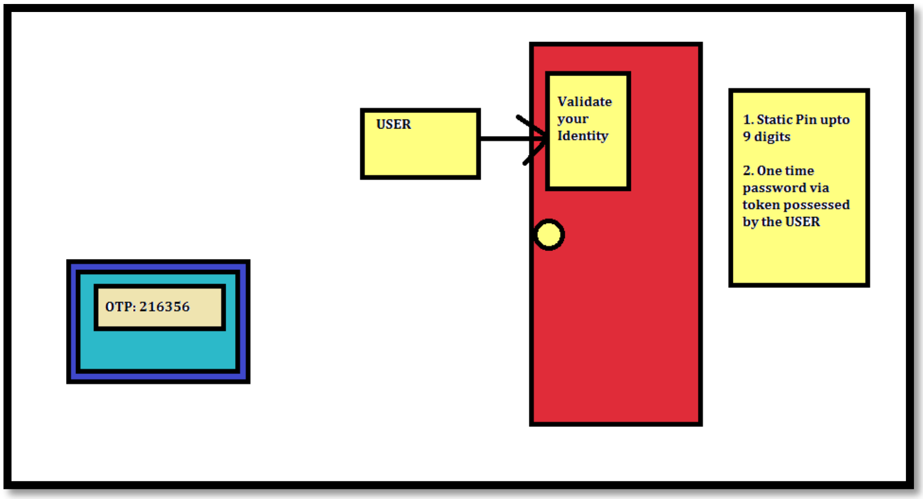

USER
Validate your Identity
1. Static Pin upto 9 digits
2. One time password via token possessed by the USER
OTP: 216356

Figure 4-11: Final Form of the project

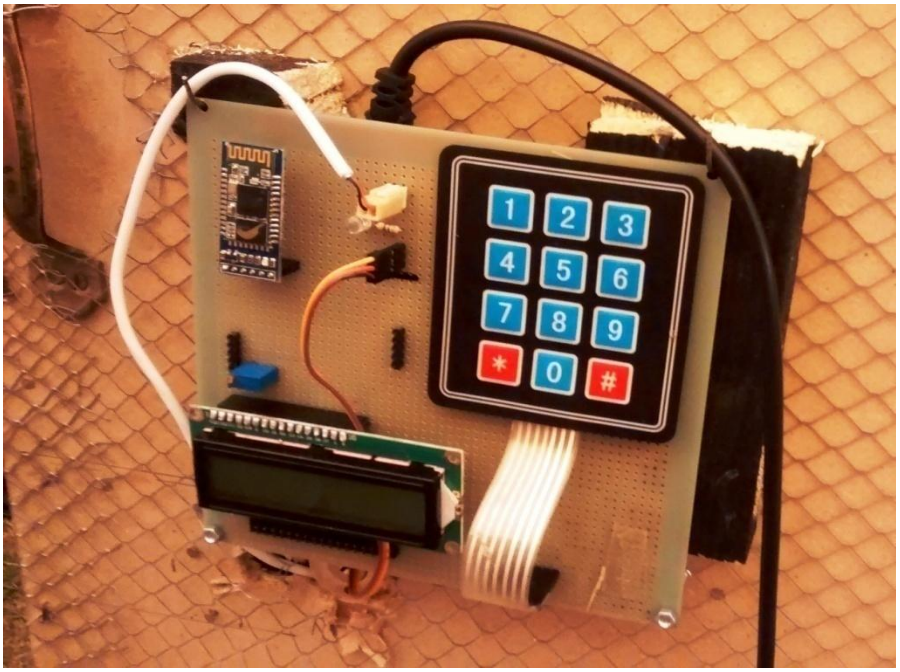


Figure 4-12: Final Display of Server

For the token and the server PCB designing was done in order to give them a compact form.

Chapter 5: Analysis and Evaluation

Chapter 5: Analysis and Evaluation

In this chapter, some test cases are discussed that were used to evaluate the performance of the both, hardware and software pertaining to the project.

5.1. Test Cases

Testing any project/product/program is essential to check and ensure the provision of intended functionality and quality of the product. A number of tests were therefore conducted by the syndicate to prove all the claims and evaluate the performance of project on the basis of the results.

5.1.1 Testing the Arduino Board

Voltage at the Arduino board is 5v we tested it by connecting the board through the serial port with the laptop and measuring voltage with the Digital Multi Meter. To check whether the Arduino board is working fine and to practice how to upload sketch on the board, a small sketch of LED blinking in which the LEDs on Arduino board blinked after an interval was uploaded using Arduino.ide software. This code is available as an example in the Arduino.ide examples.

5.1.2 Testing serial communication

Serial communication is tested through the echo test. In this the transmitter of the Arduino board is connected to the receiver through a wire. Then the code is uploaded that serially writes the data on the Rx pin of the board and then if data is received the Rx pin will write the received data on serial monitor of the Arduino.ide software. This test is done for both the Arduino boards one acting as the token and the other acting as the server. Results prove that the boards are capable of sending and receiving data serially.

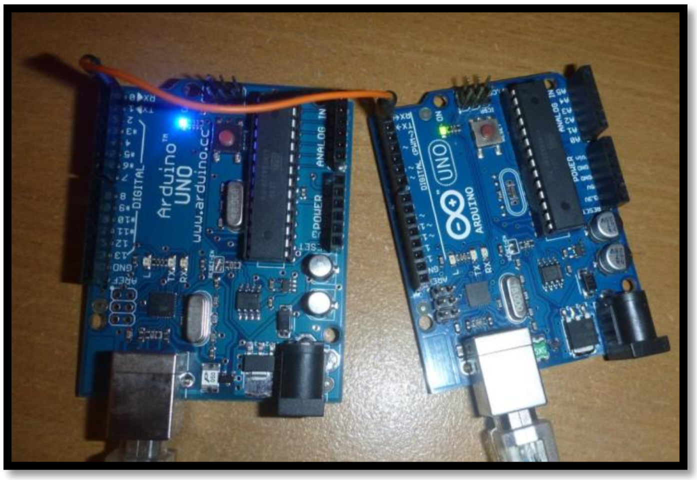
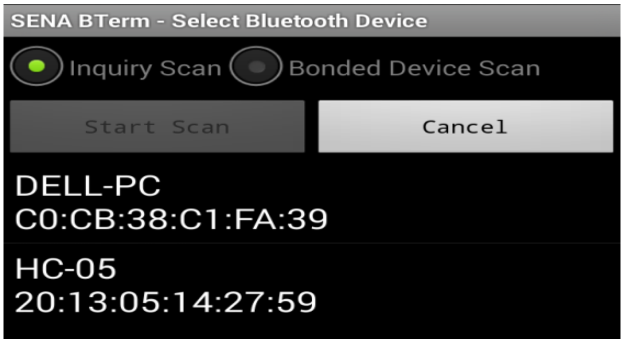


Figure 5-1: Wire Test between two boards

5.1.3 Testing the Bluetooth module

Bluetooth module was interfaced to the board by the pin connections mentioned in the previous chapter. SENA BTerm android application was installed on the phone. When the Arduino board was given power, the led of Bluetooth started blinking and on scanning the devices from the application the HC-05 was detected and its address was note down. Address for both the MASTER and SLAVE module was noted through this procedure.

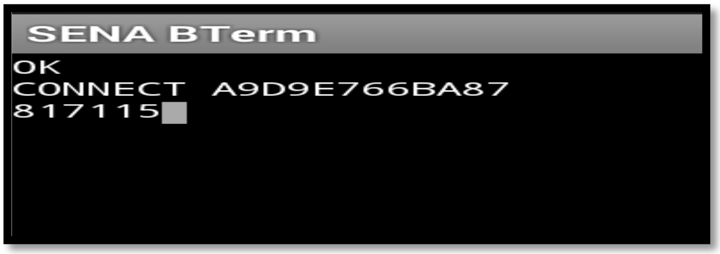


DELL-PC
C0:CB:38:C1:FA:39

HC-05
20:13:05:14:27:59

Figure 5-2: Detection of HC-05 MAC address

Now a sketch was uploaded on the Arduino board that sent the codes to the android application after every 60 seconds. As the application is capable of acting as master as well as slave the Bluetooth module acts as a MASTER as it is sending the codes and device is kept neither in master nor in slave mode. When the device is connected to HC-05 then it itself gets into slave mode and starts receiving the codes and displays them on the mobile screen.



```
OK
CONNECT  A9D9E766BA87
817115
```

Figure 5-3: Reception of data on SENA Bterm

In this way it is tested that the Bluetooth is sending data through the wire between transmitter and receiver and it also sends it wirelessly to remote devices.

5.1.4 Testing LCD 16X2

16x2 LCD was interfaced with the UNO and MEGA board with their respective pin connections mentioned in the previous chapter. A sketch already available in the Arduino.ide examples was uploaded on both the boards only the pin connections were changed in the code according to the hardware pins connected. The LCDs displayed the 'HELLO WORLD' whcich was prove to correct interference of LCD with the boards.

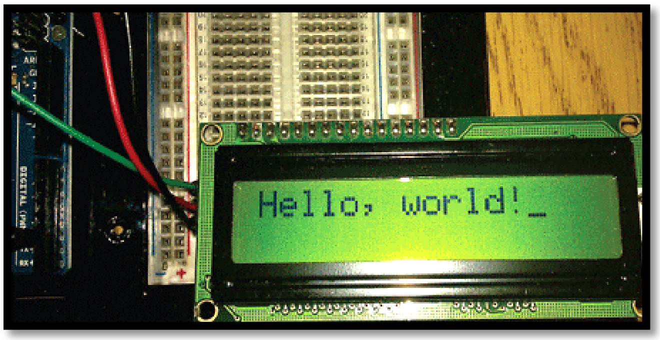
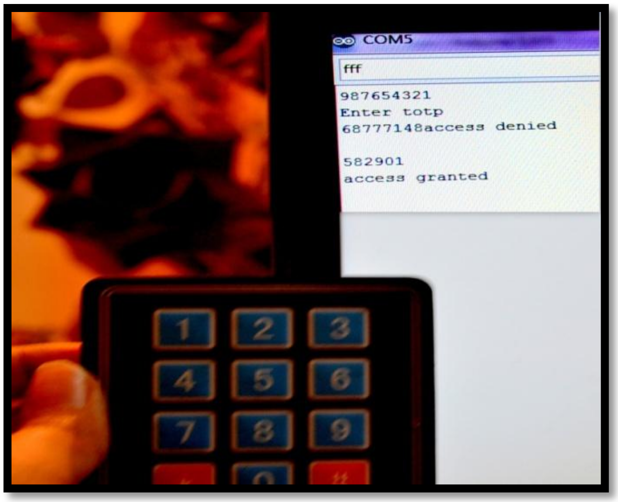


Figure 5-4: Testing the LCD Display with Arduino Mega

5.1.5 Testing KEYPAD 4X3

The keypad was interfaced to the Arduino Mega board (the authenticating server). A sketch was developed on the Arduino.ide software, keypad.h library was added from the Arduino libraries. This sketch took two inputs from the user through the keypad and saves them to two distinct variables. One input is the PIN and the other is the Time Based One Time Password. This test showed that how the keypad will work and that the keypad is capable of storing the input from the user into two different variables.



```
fff
```

```
987654321
Enter totp
68777148access denied

582901
access granted
```

Figure 5-5: Serial Monitor Results of Authentication Server

5.1.6 Testing the Servo motor

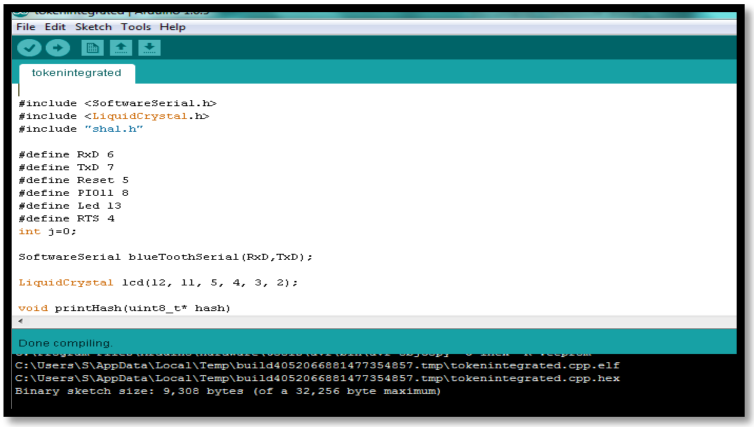
Servo motor has three pins VCC, ground and pulse .The pulse is connected to the PIN 9 on Arduino mega board. An external adapter was used to give supply to the servo motor the supply was connected to VCC of the motor and ground pin of the adapter and the motor were together ground with the Arduino board. Servo motor test code already available in the Arduino.ide software was uploaded on the Arduino Mega board with pin 9 as the pulse pin mentioned in the code. When the pulse pin of the motor was plugged in the Arduino Mega board the motor start came to its mean position after which it started rotating degrees according to the sketch.

5.2. Analysis of the three basic parts of the project

5.2.1 Smart Token

Keyed-hash message authentication code (**HMAC**) is a specific construction for calculating a message authentication code (MAC) it involves a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the *data integrity* and the *authentication* of a message. Any cryptographic hash function, such as SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed as HMAC-SHA1.HMAC-SHA1 was implemented as a C Library and imported in the IDE environment. Time based one-time passwords were derived from it in the algorithm as such a new password is displayed on LCD screen after every 60 seconds. After correctly verifying the output on PROTEUS (ISI7 Professional), the code was uploaded on Arduino.

IDE & Simulation Results:



```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include "sha1.h"

#define RxD 6
#define TxD 7
#define Reset 5
#define PIO11 8
#define Led 13
#define RTS 4
int j=0;

SoftwareSerial blueToothSerial(RxD,TxD);

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void printHash(uint8_t* hash)
```

Done compiling.

```
C:\Users\S\AppData\Local\Temp\build4052066881477354857.tmp\tokenintegrated.cpp.elf
C:\Users\S\AppData\Local\Temp\build4052066881477354857.tmp\tokenintegrated.cpp.hex
Binary sketch size: 9,308 bytes (of a 32,256 byte maximum)
```

Figure 5-6: Token Sketch

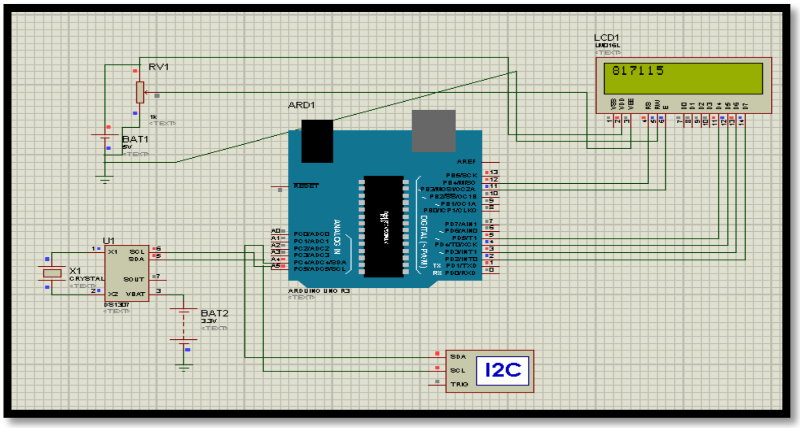


Figure 5-7: Token Simulation on Proteus

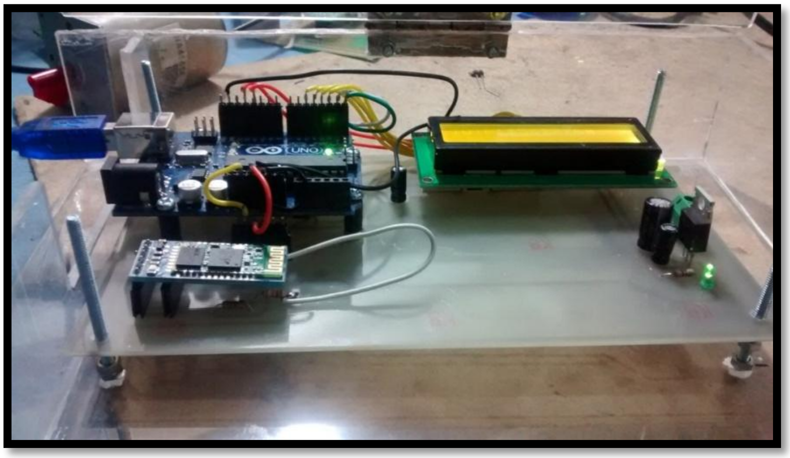


Figure 5-8: Real time working of TOTP token

Analysis:

The analysis of the token results is:

a) Time or a counter is used to synchronize the token and the authentication server which share a secret key.

b) The one-time password has a limited lifetime i.e. 60 second, after 60 seconds a new password will appear on screen and previous password will be no longer valid. When this is the case, the attacker who has learnt the one-time password can use it only within this time range.

c) The user doesn’t have to manage/remember one-time passwords. The one-time password is generated by the token and presented to the user if he needs to authenticate. Then the passwords are not taken down or hidden under the keyboard. In addition to this the user doesn’t use the same password for many services. Consequently, the attacker cannot automatically impersonate the user in many places.

d) The TOTPs generated are not linear, so one TOTP cannot be derived from the other.

e) Smart tokens have many advantages when using strong authentication including

    i. **storage capacity**

    ii. **processing power**

    iii. **portability**

    iv. **easy to use**

f) They are inherently more secure than other OTP tokens because they generate a unique, non-reusable password for each authentication.

5.2.2 Bluetooth Communication

The TOTP generated are transmitted through Bluetooth Transceiver HC-05 to the server

Analysis of Bluetooth communication:

a) Baud rate: 9600 that is it transmits 9600 samples per second.

b) Safe mode of HC-05 enabled, which encrypts the code when communicating with server

c) 6 digit codes sent by RS232 standard, Communication using RS-232 is packetized meaning that data is broken into pieces and then each piece is transmitted separately. In almost all cases, RS-232 data is sent one byte at a time.

d) Bluetooth communication is safer as it is has small range so this will only activates the server when the user who knows the PIN as well as possesses the token will approach near the server.

e) Bluetooth requires less power so power management is easy.

f) Bluetooth does not have any hazardous effects on the surrounding people.

g) Bluetooth are in CMODE 0 they will only get paired to each other not any other Bluetooth device.

h) Once the Bluetooth connection is established it can't be intercepted through android.

i) If the Bluetooth communication is jammed it will delay the access for validate user but any fraud user can't gain access until he has the PIN and Token.

5.2.3 Authenticating Server

Sketch and simulation results of the server are as follow:

<u>IDE and Simulation Results:</u>

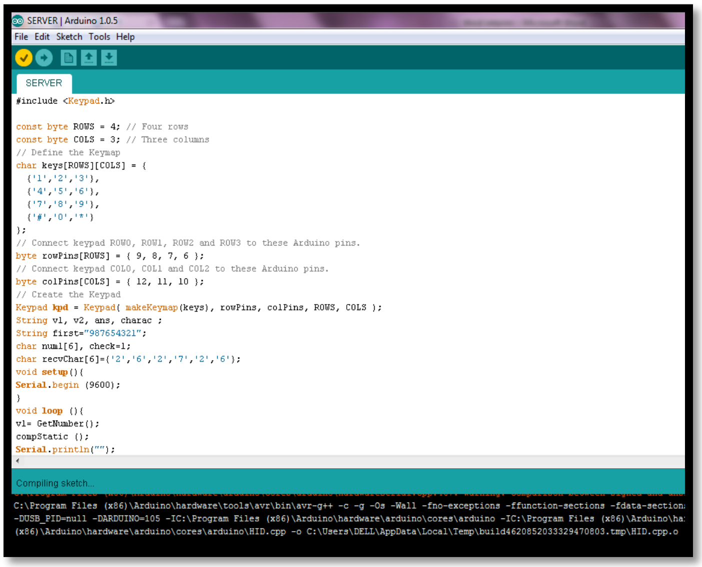
The figure below shows the verification of server's sketch in IDE. After this the hex file generated is simulated in ISIS to show the working of server.

a) It gets PIN verifies it.

b) Gets TOTP from user and verifies it

c) Displays results on LCD

d) Drives the motor to open the lock



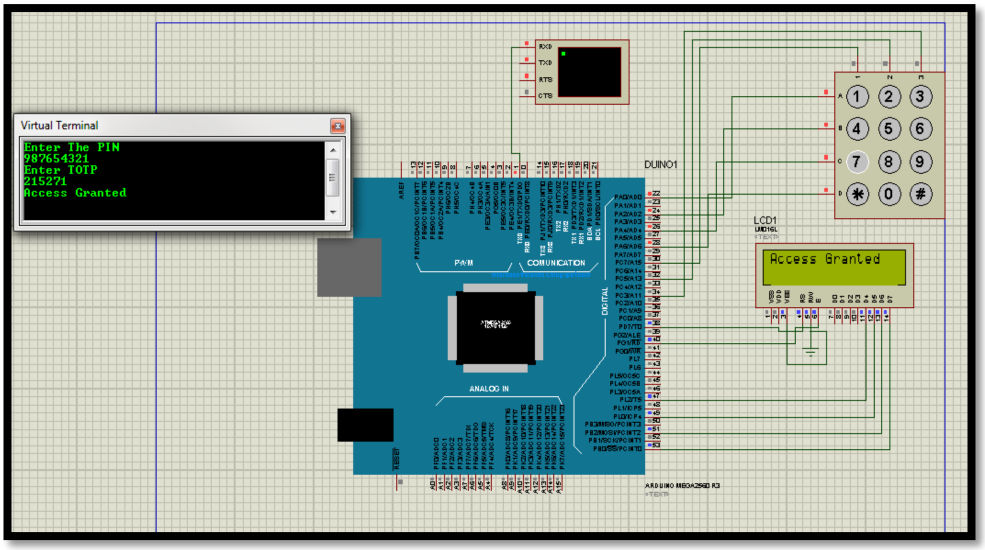
```
#include <Keypad.h>

const byte ROWS = 4; // Four rows
const byte COLS = 3; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'#','0','*'}
};
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = { 9, 8, 7, 6 };
// Connect keypad COL0, COL1 and COL2 to these Arduino pins.
byte colPins[COLS] = { 12, 11, 10 };
// Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
String v1, v2, ans, charac ;
String first="987654321";
char num1[6], check=1;
char recvChar[6]={'2','6','2','7','2','6'};
void setup(){
Serial.begin (9600);
}
void loop (){
v1= GetNumber();
compStatic ();
Serial.println("");
```

Compiling sketch...

```
C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-g++ -c -g -Os -Wall -fno-exceptions -ffunction-sections -fdata-section
-DUSB_PID=null -DARDUINO=105 -IC:\Program Files (x86)\Arduino\hardware\arduino\cores\arduino -IC:\Program Files (x86)\Arduino\ha
(x86)\Arduino\hardware\arduino\cores\arduino\HID.cpp -o C:\Users\DELL\AppData\Local\Temp\build4620852033329470803.tmp\HID.cpp.o
```

Figure 5-9: Server Sketch



```
Enter The PIN
987654321
Enter TOTP
215271
Access Granted
```

Figure 5-10: Simulation of Server

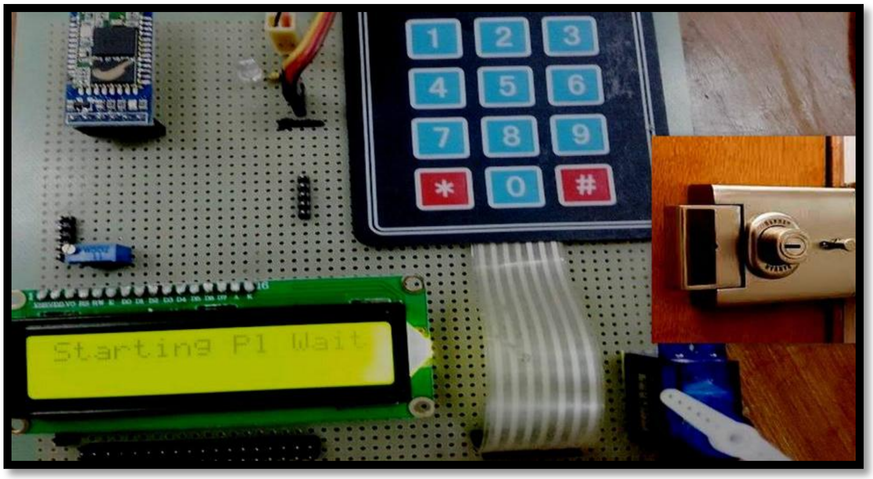


Figure 5-11: Real time working of Server

Analysis:

a) TOTP code is wirelessly received and changes every 60 seconds. In this way we don't need a data base of One Time Passwords at the server and same server can verify codes from different tokens as the TOTP received through Bluetooth is compared with the one entered.

b) No extra memory is required to store all generated TOTPs rather not saving the TOTPs makes it more secure. SERVER saves nothing but the user PIN.

c) 9 digits static PIN will be the first factor of authentication and TOTP is the second.

d) PIN is 9 digits long; such a lengthy PIN makes it more difficult to hack.

e) Even if Token is lost or stolen still only that person will be able to get access who knows the PIN. In this way multifactor authentication is ensured.

f) Server will only be activated when token comes in near vicinity, otherwise it won't accept anything typed from the keypad.

Chapter 6: Future Work and Conclusion

6.1 Future Work

6.1.1 Integration with software applications

6.1.2 Encryption of passwords

6.1.3 Implementation on variable hardware platform

6.1.4 Work on Confidentiality and Integration

6.1.5 Incorporation of Element of Inherence

6.2 Conclusion

Chapter 6: Future Work and Conclusion

In this chapter we will discuss the future aspects for further development in the proposed project. Moreover a summarized view of the entire project will also be given which will highlight One-Time Password Lock System's potential application areas and conclude the discussion.

6.1. Future Work

A project like ours can be customized easily according to the demands and the requisites of the emerging complexity in access systems. Some of the enhancements are discussed in this chapter that can be made to the project in future to enhance its features and make it more efficient.

6.1.1. Integration with software applications

Integrating the token with a server based on software applications could increase its usage for securing social networking applications. The availability of an online server would increase the range of applicability of the technology.

6.1.2. Encryption of passwords

The TOTP communicated by the token to the server could be made more secure. Encrypting the password using algorithms like AES could further increase the level of authenticity promised by the project.

6.1.3. Implementation on variable hardware platform

The project could be implemented on other hardware platforms or nano boards to make the technology more compact, portable and easier to use.

6.1.4. Work on Confidentiality and Integration

Since our project circles around the authenticity of a security system, further work could be done on the project regarding:

a) Confidentiality

b) Integrity

6.1.5. Incorporation of element of Inherence

We incorporated two factors to achieve authentication in our project. Multi factor authentication involves the use of a third factor as well. By integrating the element of inherence (something the user is) such as a biometric characteristic or a fingerprint, the level of authenticity would be much enhanced.

6.2. Conclusion

Strong authentication solutions address the limitations of static passwords by incorporating an additional security measure. The project authenticates a user using two factors as a basis of its authentication:

a) A static password known to the user

b) The token providing a one-time access code/password having a lifetime of 60 seconds

The use of PIN and Password together are able to reinforce the security levels practically available to systems. The objective of safe user authentication is met, in order to protect the user from identity theft. Since a **one-time password** (OTP) is a password that is valid for only one login session or transaction. Our project is successfully able to avoid a number of shortcomings that are associated with traditional (static) passwords. It is practically not vulnerable to replay attacks. This means that a potential intruder who manages to record an OTP that was already used to log into a service or to conduct a transaction will not be able to abuse it, since it will no longer be valid

The token generated OTP is transmitted to the server board using wireless Bluetooth (v2) protocol. The HC-05 Bluetooth transceiver can be set to safe and encryption mode of transmission as well. In other words, the communication of password to server is safe and secure. Another attribute of our project is the use of HMAC- SHA1 in the token generation. This, unlike most other authentication protocols, makes our TOTP system efficient as brute force is practically not possible in such a short time.

Our project aims to contribute to the beginning of an era where the vulnerabilities due to static passwords will no longer be known and a highly secure, multi factor authentication system will be reinstated in all network based security systems as well as in security and accessibility related issues of our everyday lives .It could become a revolutionary improvement to the current banking system to ensure the safety of lockers and client accounts in addition to multiple access systems.

Appendix

1. ***Appendix A – Code to generate TOTP and their Bluetooth communication***
2. ***Appendix B – Code to receive the TOTPs and authenticate***

APPENDIX-A

TOTP generation and Communication.ino

```
#include <LiquidCrystal.h>

#include "sha1.h"


int j=0;

int wait = 0;



LiquidCrystal lcd(12, 11, 5, 4, 3, 2);


void printHash(uint8_t* hash)

{

  int i;

  for (i=0; i<20; i++) Serial.println(hash[i]);

  Serial.println();

}


uint8_t hmacKey1[]={ 0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x21, 0xde, 0xad, 0xbe, 0xef, 0x6f,
0x21, 0xde, 0xad, 0xbe, 0xef };


long birthTime = 1339345870;

long intern = 0;

long oldOtp = 0;
long truncatedHash = 0;



void setup()
{
 Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("TOTP: ");



 }

void loop()
{
   if(intern == 0) intern = birthTime;
  else{

   uint8_t byteArray[8];
   long time = intern / 60;

   byteArray[0] = 0x00;
   byteArray[1] = 0x00;
   byteArray[2] = 0x00;
   byteArray[3] = 0x00;
   byteArray[4] = (int)((time >> 24) & 0xFF) ;
   byteArray[5] = (int)((time >> 16) & 0xFF) ;
```

```
byteArray[6] = (int)((time >> 8) & 0XFF);
byteArray[7] = (int)((time & 0XFF));

uint8_t* hash;
uint32_t a;
Sha1.initHmac(hmacKey1,16);
Sha1.writebytes(byteArray, 8);
hash = Sha1.resultHmac();

int  offset = hash[20 - 1] & 0xF;

int j;
for (j = 0; j < 4; ++j) {
 truncatedHash <<= 8;
 truncatedHash  |= hash[offset + j];
}

truncatedHash &= 0x7FFFFFFF;
truncatedHash %= 1000000;


if(truncatedHash != oldOtp){
 oldOtp = truncatedHash;
 wait = 0;
 lcd.setCursor(5, 0);
 lcd.print (truncatedHash);
 lcd.print('$');
```

```
   Serial.print('$');
   Serial.print(truncatedHash);
   Serial.print('#');
   lcd.setCursor(0, 1);
   lcd.print("                ");



  }
  else wait++;

  if(wait % 2 == 0){
    lcd.setCursor(wait/2, 1);
  }
  delay(1000);
  intern++;
  }
}
```

APPENDIX-B

Reception and authentication.ino

```
#include <LiquidCrystal.h>

#include <Wire.h>

#include <Keypad.h>

#include<Servo.h>


LiquidCrystal lcd(38, 40, 47, 49, 51, 53);


const byte ROWS = 4; // Four rows

const byte COLS = 3; // Three columns

// Define the Keymap

char keys[ROWS][COLS] = {

  {'1','2','3'},

  {'4','5','6'},

  {'7','8','9'},

  {'*','0','#'}

};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.

byte rowPins[ROWS] = { 22, 24, 26, 28};

// Connect keypad COL0, COL1 and COL2 to these Arduino pins.

byte colPins[COLS] = { 30, 32, 34 };

// Create the Keypad
```

```
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
String v1,v2, ans, charac ,Data;
String first="987654321";
char num1[6], array[6], check=1;
Servo myservo;
char m;
int pos=0;
unsigned char DataGet=0,i=0,DataComplete=0;
char RxData[10];
char DataAcquired=0;
long int iData,rData;
void setup()
{
   Serial.begin (9600);
   Serial2.begin(9600);
   lcd.begin(16, 2);
   lcd.setCursor(0,0);
   Serial.print("Start");
   lcd.print("Starting Pl Wait");
   Serial.println("");
   lcd.setCursor(0,14);
   myservo.attach(9);
}

void loop ()
```

```
{
 char x=Serial2.available();
  if(x>0)
  {
    x=0;
    char Rx=Serial2.read();
    Serial.write(Rx);
    if(Rx=='$')
      {
      DataGet=1;
      }
    if(DataGet==1 && Rx!='$'&& Rx!='#')
     {
       RxData[i++]=Rx;
     }
     if(Rx=='#')
     {
        RxData[i++]='\0';
        DataComplete=1;
        DataGet=0;
        i=0;
     }
  }
  if(DataComplete)
  {
```

```
 DataComplete=0;
 DataAcquired=1;
 Serial.print(RxData);
}
if(DataAcquired)
{
 lcd.clear();
 lcd.print("Enter The PIN");
 Serial.print("Start\r\n");
 lcd.setCursor(0,0);
 myservo.attach(9);
 v1= GetNumber();
 compStatic ();
 Serial.println("");
 lcd.setCursor(0,0);
 v2=GetNumber();
 Serial.println("");
 Serial.print("Current str is \r\n");
 Serial.print(RxData);
 Data=String(RxData);
 //iData=atoi(RxData);
 //Data=String(RxData);

  if (Data==v2)
  {
        Serial.print("Access Granted");

        Serial.println("");

        lcd.clear();

        lcd.setCursor(0,0);

        lcd.print("Acess Granted");

        Serial.println("");

        v2.replace(v2,"");

  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees

 {                               // in steps of 1 degree

   myservo.write(pos);                         // tell servo to go to position in variable 'pos'
// waits 15ms for the servo to reach the position

 }

        delay(5000);


   delay(5000);

 for(pos = 180; pos>=1; pos-=1)     // goes from 180 degrees to 0 degrees

 {

   myservo.write(pos);                         // tell servo to go to position in variable 'pos'
// waits 15ms for the servo to reach the position

 }


        DataAcquired=0;

     }

   else

         {

        lcd.clear();
```

```
      lcd.setCursor(0,0);
      Serial.print("Access Denied");
      lcd.print("Access Denied");
      Serial.println("");
       }
  }
}

String GetNumber()
{
  char key = kpd.getKey();
  while(key != '#')
  {
    key = kpd.getKey();
    switch (key)
    {
      case NO_KEY:
        break;

      case '0': case '1': case '2': case '3': case '4':
      case '5': case '6': case '7': case '8': case '9':
        //lcd.print(key);

        Serial.print(key);
        lcd.print('*');
```

```
          //num = num * 10 + (key - '0');
          charac=String (key);
          ans=ans+charac;
         // ans.toCharArray(array,4);
          //Serial.print (array);
          break;

         case '*':
         ans.replace(ans,"");
         break;
     }



   }

   return ans;
}
 void compStatic()
 {
    if (v1==first)
    {Serial.println("");
    Serial.print("Enter TOTP");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Enter TOTP");
    }
```

Bibliography

References

[1] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, pp. 723-728, 2009.

[2] N. Dragoljub, "“Stronger security”," *Card Technology Today*, vol. 19, no. 1, pp. 9-10, 2007.

[3] D. S. Kenneth G. Paterson, "One-time-password-authenticated key exchange," University of London, UK; Queensland University of Technology, Australia, September 4, 2009.

[4] WIKIPEDIA. [Online]. http://en.wikipedia.org/wiki/One-time_password

[5] S. Aljareh and A. Kavoukis, "EFFICIENT TIME SYNCHRONIZED ONE-TIME PASSWORD SCHEME TO PROVIDE SECURE WAKE-UP AUTHENTICATION ON WIRELESS SENSOR NETWORKS," *International Journal Of Advanced Smart Sensor Network Systems (IJASSN)*, vol. Vol 3, no. No.1, Jan. 2013.

[6] R. G. Kammer and C. L. Shavers. (2001) The Keyed-Hash Message Authentication Code.

[7] J. K. J. a. Y. M. T. H. Y. Chien, "An efficient and practical solution to remote

authentication: smart card," *Computers & Security*, vol. 21, no. 4, pp. 372-375, 2002.

[8] W. K. S. H. C. Hsiang, "Weaknesses and improvements of the Yoon–Ryu–Yoo remote user authentication scheme using smart cards," *Computer Communications*, vol. 32, no. 4, pp. 649-652, 2009.

[9] M. S. H. a. L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, 2000.

[10] H. M. Sun, "An efficient remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958-961, 2000.

[11] W. T. Z. a. D. G. F. J. Xu, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723-728, Jun. 2009.

[12] K.-C. Liao, W.-H. Lee, M.-H. Sung, and T.-C. Lin, "A One-Time Password Scheme with QR-Code Based on Mobile Phone," in *Fifth International Joint Conference on INC, IMS and IDC*, 2009, pp. 2069-2071.

[13] O. L., ""Comparing passwords, tokens, and biometrics for user authentication"," in *Proceedings of the IEEE,* vol. 91, 2003.

[14] D. d. Borde, "“Selecting a two-factor authentication system”," *Network Security*, vol. 2007, no. 7, pp. 17-20.

[15] N. B. K. Bicakci, ""One-Time Passwords: Security Analysis using BAN Logic and Integrating with Smartcard Authentication"," in *Lecture Notes in Computer Science*. Springer, 2003.

[16] A. Huang and L. Rudolph, *Bluetooth for Programmers*. 2005.

[17] Jimbo. learn.sparkfun.com. [Online]. https://learn.sparkfun.com/tutorials/bluetooth-basics/how-bluetooth-works

[18] N. Gunasekaran, R. S. Reddy, and K. V. S. S. S. S. Sairam, "Bluetooth in Wireless Communication," *IEEE Communications Magazine*, pp. 91-96, Jun. 2002.

[19] D. Djonin and J. Zhu, "THE BLUETOOTH SYSTEM," University of Victoria, Project Report, 2001.

[20] Miscellaneous: TO DO.

[21] W.-T. Z. D.-G. F. Jing Xu, "" An improved smart card based password authentication scheme with provable security"," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723-728, 2009.

[22] "google-authenticator - Project Hosting on Google Code". [Online]. https://code.google.com/p/google-authenticator/

[23] www.facebook.com. [Online]. https://www.facebook.com/help/413023562082171/

[24] www.dropbox.com. [Online]. https://blog.dropbox.com/2012/08/another-layer-of-

security-for-your-dropbox-account/

[25] (2006) www.gemalto.com. [Online]. http://www.gemalto.com/brochures/download/ent_otp_secure_access.pdf

[26] (1932) www.ocbc.com. [Online]. http://www.ocbc.com/personal-banking/security/token.html