**MULTI-FACTOR AUTHENTICATION WITH ASYNCHRONOUS TOKEN AND SHORT MESSAGE SERVICE**

by

Angel Anna Mathew

Submitted in partial fulfilment of the requirements for the degree of

MASTER OF ENGINEERING

Major Subject: Internetworking

at

DALHOUSIE UNIVERSITY

Halifax, Nova Scotia   
July 2019

© Copyright by Angel Anna Mathew, 2019

**Dalhousie University**

**Faculty of Engineering**

**Internetworking**

The undersigned hereby certify that they have read and award a pass in INWK 6800 for the course/seminar entitled "Multi-factor authentication with asynchronous token and short message service" by *Angel Anna Mathew* in partial fulfilment of the requirements for the degree of Master of Engineering.

Dr. Bill Robertson

**DALHOUSIE UNIVERSITY**

**INTERNETWORKING PROGRAM**

**AUTHORITY TO DISTRIBUTE REPORT**

Title:

Multi-factor authentication with asynchronous token and short message service

The Internetworking Program may make available or authorise others to make available individual photo/microfilm or soft copies of this report without restrictions after July 17, 2019.

The author attests that permission has been obtained for the use of any copyrighted material appearing in this report (other than brief excerpts requiring only proper acknowledgement in scholarly writing) and that all such use is clearly acknowledged.

Full Name of Author: Angel Anna Mathew

Signature of Author: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: July 17 2019

Table of Contents

LIST OF FIGURES vi

LIST OF SYMBOLS AND ABBREVIATIONS vii

ACKNOWLEDGEMENTS viii

EXECUTIVE SUMMARY ix

1 INTRODUCTION 1

1.1 Authentication Infrastructure 2

1.2 Authentication solution 3

1.2.1 Identification 3

1.2.2 Challenge key 4

1.2.3 Smart Card 4

1.2.4 Response Key 4

1.3 Authentication Implementation 4

1.4 Outline 5

2 LITERATURE REVIEW 6

2.1 Steps involved in MFA 6

2.2 Technologies Used in MFA 7

2.3 Flow of Control 7

2.4 Summary 8

3 METHODOLOGY 9

3.1 Authentication Solution 9

3.1.1 Symbols used 10

3.1.2 Initialization 10

3.1.3 Time factor-based challenge generation 11

3.1.4 Encrypting the generated challenge 11

3.1.5 Hashing the encrypted challenge 11

3.1.6 HC sent in SMS 11

3.1.7 HC entered in smart card 11

3.1.8 HR is entered to get access 12

3.2 Summary 12

4 MFA AUTHENTICATION TECHNOLOGIES 13

4.1 Authentication Factors 13

4.2 Multi-factor Authentication Technologies 14

4.2.1 Client-side template 15

4.2.2 Server-side business logic 15

4.2.3 Asynchronous software-based token 15

4.2.4 Twilio helper library to generate SMS 15

4.3 Summary 16

5 MFA IMPLEMENTATION 17

5.1 Specifications of server and client 17

5.1.1 Server and client hardware specifications 17

5.1.2 Server software specifications 17

5.1.3 Client Software Specifications 18

5.2 Specifications of asynchronous token 18

5.2.1 Specifications of mobile device 18

5.3 Generating Time- factor based Challenge 18

5.4 Summary 19

6 OUTCOMES OF MFA 21

6.1 Login Page 21

6.2 SMS send to users’ phone 22

6.3 Track Delivery Status of Message Sent 23

6.4 Asynchronous Token 24

6.5 Login Page with Response Form 26

6.6 Summary 28

7 CONCLUSION AND RECOMMENDATIONS 29

7.1 Conclusion 29

7.2 Recommendations 30

REFERENCES 31

LIST OF FIGURES

Figure 1.1 A Cloud Based Authentication Infrastructure 2

Figure 1.2 The Framework for challenge transfer using SMS 3

Figure 2.1 Flow Chart 7

Figure 4.1 Multifactor Authentication Technologies 14

Figure 5.1 Time factor based challenge generation 19

Figure 6.1 Login Page 21

Figure 6.2 Invalid Credentials 22

Figure 6.3 SMS sent from Twilio account 23

Figure 6.4 Status callback of SMS sent 24

Figure 6.5 Asynchronous Token Challenge Entry Screen 25

Figure 6.6 Response Screen of Asynchronous Token 26

Figure 6.7 Login Page with Resposne Form 27

Figure 6.8 Response Form with Error Message 27

Figure 6.9 Home Page 28

LIST OF SYMBOLS AND ABBREVIATIONS

API Application Programming Interface

CSS Cascading Style Sheet

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

MFA Multi-factor Authentication

OOB Out - of Band

OTP One Time Password

PIN Personal Identification Number

REST Representational State Transfer

SFA Single Factor Authentication

SMS Short Messaging Service

TFA Two Factor Authentication

URL Universal Resource Locator

ACKNOWLEDGEMENTS

Firstly, I would like to thank god for showering his blessings on me to make this report a success.

I am highly indebted to Dr. Bill Robertson and Dr. Muhammad Raza, Dalhousie University Halifax for their valuable guidance, encouragement, and suggestions.

I take this opportunity to thank each member of the faculty who has helped me make this report a success.

I wish to express my sincere thanks to all my classmates for their support they have given me throughout the course. With pleasure I would like to express my deep sense of indebtedness to my family members for their motivations in making this work a success.

EXECUTIVE SUMMARY

The role of credentials alone to secure user account is nominal. They can be stolen by attackers through various techniques. A supplementary factor of security should be included to control security vulnerabilities. Here multi-factor verification is employed along with an asynchronous token and out of band authentication method. Out of band authentication method provides a separate channel for communication, which makes it difficult the attacker to gain access. A three-step verification is served to increase steps of security. When the user tries to log in to his account along with valid credentials, he is taken to a screen to enter the response key. A time factor-based challenge is sent in a Short Message Service (SMS) to the user’s mobile phone, which is an out of band method. The user has a smart card which is his mobile app in which he enters the challenge. The smart card computes the challenge and gives back a response. The user then enters it in the directed page and with successful verification he is taken to his account. Here authentication by knowledge and ownership are combined along with time factor-based challenge and out of band authentication. This way of multi-factor verification can notably weaken security breach.

# INTRODUCTION

Static password is the most widely accepted form of authentication. Anyhow since it’s a string of alphanumeric characters there are methods to crack it easily. Combination of strings can be used to improve its complexity, but it will be difficult for the user to remember the password. There comes the use of passphrase which is a sequence of combined forms of string. Even then hackers use methods to crack it such as brute force attack, dictionary attack, shoulder surfing and sniffing [Chh2012]. Since only a single factor which is the password is used for verification, this method is called Single Factor Authentication (SFA). As reported by SafeNet Global Authentication Survey, companies endorse cloud-based authentication and mobile MFA to hold labour pool [Ann2014]. Thus, to improve toughness of cracking security several factors are introduced.

There are several ways to implement Two Factor Authentication (2FA), some of the significant methods are by using:

* Synchronous and asynchronous tokens
* One Time Password (OTP) authentication [Naf2012]
* Out of Band (OOB) authentication [Sar2012]

Synchronous tokens are synchronized with the server and uses an algorithm that calculates a number at the server and client device. When the user enters this number, he obtained from the token, he gets verified. Synchronization can be done based on time and event. In the working of asynchronous tokens, the challenge obtained from the authentication server is entered the token to generate a response which the user uses for getting access granted. In verification based on OTP [Naf2012], an OTP is sent to the mobile which the user uses for getting the access granted. However, OTP faces several challenges like static password such as sniffing and social engineering. Using OOB authentication user is asked to enter the code he obtained in his phone for the purpose of verification.

## Authentication Infrastructure

In this paper asynchronous challenge response technique combined with authentication with mobile phones, thus providing three tiers of security. When the user tries to login to the cloud server a dynamically generated unique challenge is send to his phone as an SMS. He then enters this challenge along with a user pin (pin is generated by the user during initial registration) in his mobile application which as a smart card [Naf2012]. The application computes a response for this challenge and gives him back. He then enters the response, for passing the final tier of authentication. The server checks the response based on the authentication algorithm and if it matching the user is provided access, else access is denied. As shown in Figure 1.1 when user tries to login to the cloud server, one tier of authentication service is provided by the asynchronous token at the client side.

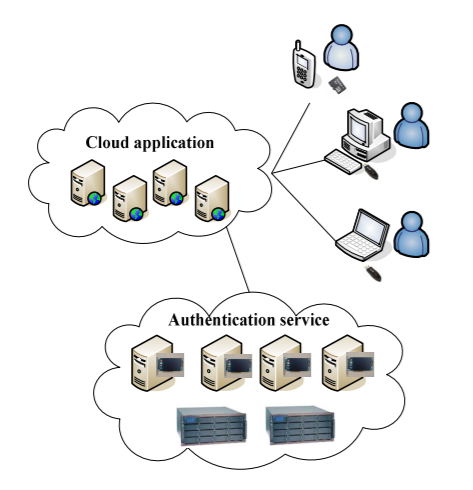


Figure 1.1 A Cloud Based Authentication Infrastructure [ZHA2015]

In the Figure 1.2, it shows how a challenge is send to the user. Here the server is deployed in cloud and user’s phone is used a smart card.

1. Initially users the credentials, which is verified by server.
2. The server then generates a unique challenge, which is sent to the user’s phone.
3. The smart card, which is the phone computes this challenge and generates a response, which the user enters for verification.
4. The server verifies this challenge and if verifies, grants access to user.

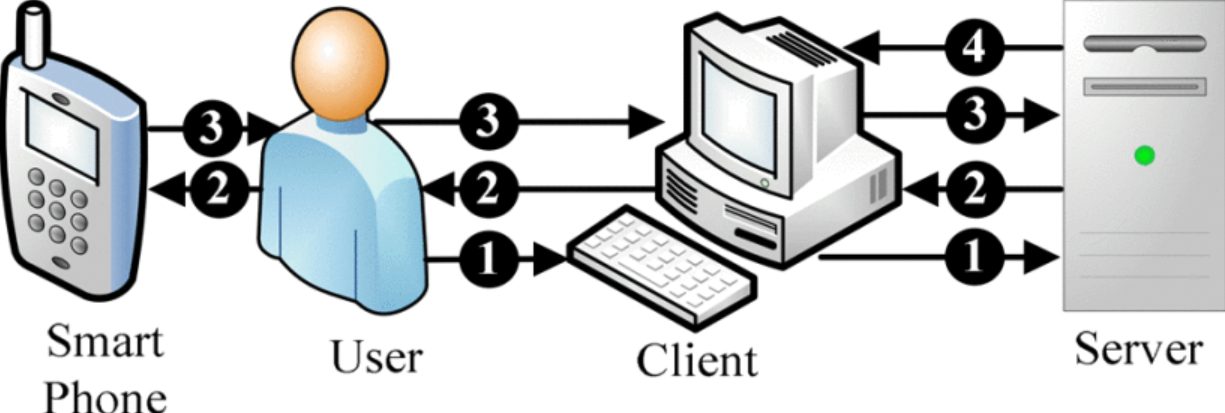


Figure 1.2 The framework for challenge transfer using SMS [Moh2011]

## Authentication solution

### Identification

User initially logins with username and password. The username can only be the combination of alphabets and letters. Password should be the combination of alphanumeric characters and be at least 6 digits long. A 4-digit pin should also be set by the user at the time of registration. However, pin is not required for identification.

### Challenge key

When a registered user tries to log-in with username and password, the server checks for their authentication. If they are authenticated, combination of time factor and a random number are encrypted with the key. The key is a base64 encoded [Chh2012] value of the pin. This encrypted value is hashed and send as challenge to the user’s phone.

### Smart Card

The challenge received on users’ phone is entered to smart card, which is a mobile application. Once the challenge is entered, the smart card computes the hash value of challenge. Absolute value of this combination is given as response.

### Response Key

The response key obtained from smart card is entered on the screen. The server checks if this key matches to the combination of hashed encrypted value of challenge and hashed value of pin. If they match, access is granted to user.

## Authentication Implementation

This multi-factor authentication project is implemented using a framework called Flask [Pyt2016], with Python 3.7 as the server-side language. The client side is developed using Hyper Text Markup Language (HTML5) and styled using Cascading Style Sheets (CSS3). JetBrains PyCharm [Jet2019] is the Integrated Development Environment (IDE) used for coding python. The server sends text messages to user with Twilio programmable SMS [Twi2019]. Twilio takes care of the mobile carrier and global regulations. Firstly, an account in Twilio must be created, then using Twilio number SMS can be send mobile numbers around the globe. Twilio uses Representational State Transfer Application Programming Interface (REST API) to send outbound messages.

Once the challenge is received through the SMS, it is entered into a mobile application that is developed in android. Java and extensible Markup Language (XML) are mainly used for developing this android application. Android Studio [And2019] is used as the IDE. This application computes the response and is given to the user. The user then enters the response to the page directed by server. If the response is valid, the user is given the access, thus authentication is successful.

## Outline

The following are the chapters that explain MFA in detail. The first chapter was primarily focusing on introducing the rudimentary concepts of MFA.

Chapter 2 is Literature Review, which mainly focus on giving a flow of control of MFA starting from the login page and ending on the home page.

Chapter 3 is Methodology, which explains the authentication solution used in implementing MFA.

Chapter 4 is MFA implementation, which describes the methods for implementing MFA. Both the hardware and software specifications used in demonstrating the implementation of MFA.

Chapter 5 is Outcome of MFA; this chapter deals with demonstrating the outputs obtained after running MFA on an application. Furthermore, the chapter also has screenshots of the web pages and mobile application screens.

Chapter 6 is Conclusion and Recommendations; this discusses the results of all experiments and draws conclusions. The chapter also provides some recommendations and suggestions for any future work

# literature review

In traditional authentication methods single factor was used. Typically, a static password was used to verify if he is the claimed subject. This introduced several vulnerabilities in security. Anyone with the password can access the subject’s resources. Hackers were also able to access the password using brute force attack and dictionary attack. Even though passwords were encrypted, with the help of vulnerabilities it can be decrypted. Once the password is decrypted, the hacker can easily access the resources. There is no barrier for him. As a matter of fact, generally 63% of data breach is caused by lost, stolen or default passwords in 2016[And2019]. With the introduction of two-factor authentication(2FA) this is controlled to a greater extend.2FA can be performed with synchronous token and asynchronous token. In synchronous token method, the token should be synchronized [Sar2016] with the server, while in asynchronous token, the token works independently. However, if the intruder cracks the password and gets access to the token, then he can easily get authenticated. So even 2FA also can’t be considered as one of the secured methods for authentication. Thus, there is a need for application of one more layer of verification. So, in this report an SMS sent to users mobile and the asynchronous token are taken together to provide extra layer of authentication.

## Steps involved in MFA

The user tries to login with username and password, if his credentials are valid, he is credit to another page, where he has the provision to enter response key. Simultaneously an SMS is sent to users’ mobile phone with a challenge. The user enters a pin code (which is already known to the user) and a challenge to the smart card which is his mobile application. The smart card computes a response key for the challenge given. The user then enters the response key on the page he is asked to, if it’s a valid response key he is authenticated successfully to the account.

## Technologies Used in MFA

The technologies that are involved in MFA are client-side template, server-side business logic, Twilio helper library to generate SMS and asynchronous software-based token [Zha2019]. When the user tries to log in to his account along with user credentials, he is taken to a form on the login page to enter the response key. The login page is developed in flask framework using Python and HTML5. Simultaneously, a challenge based on current date and time is generated by the server and is sent in an SMS to the user’s mobile phone using out of band method. Here out of band method provides an additional alternate way for communication if one path is hijacked by attacker. The SMS is send using a Twilio helper library [Twi2019]. The soft token computes the challenge and gives back a response.

## Flow of Control

The initial step is to open the webpage of the login page. Figure 2.1 describes the flow of control of MFA. The process starts with the login page. Here a user is assumed to be registered user by default. The user’s username and password are entered on to the login page, they are already stored in the database. The database used in the implementation of project is SQLite 3 [Sql2019] with SQLAlchemy [The2019].Once the credentials are authenticated the user is directed to login page with a form to submit the response key. The user also receives an SMS in his mobile with the challenge question .The user then enters the challenge question in the asynchronous token (which is a mobile application ). Here an android mobile application is used as the asynchronous token. The token computes a response key for the challenge. User then enters the response key on the input field in login page. If its valid the user in taken to home page.

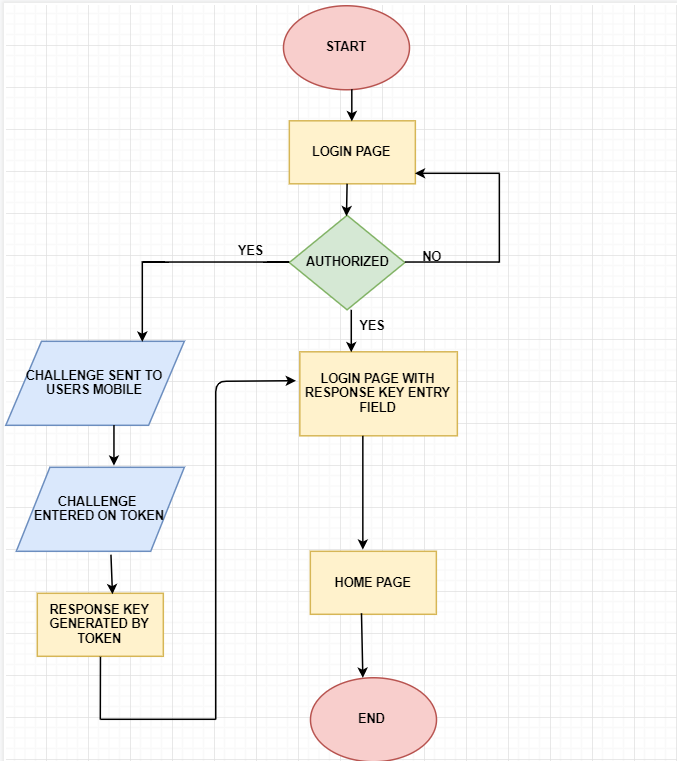


Figure 2.1 Flow chart

## Summary

The initial step for the implementation of MFA begins at login page, when user gives valid credentials, he is taken to the login page with a form to enter response key. Furthermore, a message is also sent to his mobile phone. Twilio API [Twi2019] is used to implement the message service. Asynchronous token plays a role to compute the response key. Once the response key is generated it is given to the form of response key in the login page. With a valid response key, the user is authenticated to the home page. This chapter also discuss about the technologies used in the MFA: template, business logic, software-based token and Twilio helper library are them.

# METHODOLOGY

The MFA demonstrated here, uses SMS sent to users mobile and the asynchronous token to provide extra layer of authentication. When the user tries to login with username and password, if his credentials are valid, he is credit to another page, where he has the provision to enter response key. Simultaneously an SMS is sent to users’ mobile phone with a challenge. The user enters a pin code (which is already known to the user) and a challenge to the smart card which is his mobile application. The smart card computes a response key for the challenge given. The user then enters the response key on the page he is asked to, if it’s a valid response key he is authenticated successfully to the account. The whole project is implemented in Python 3.7, serving the server side and android, HTML5 [Hoo2019] serving the client side. The login authentication part is developed in Python and HTML5, with CSS styling. The smart card computations are done in android. Moreover, Twilio [Twi2019] is used as the cloud communication platform to send SMS to user.

## Authentication Solution

The authentication protocol used for MFA [Moh2011] is divided into several phases. In each phase certain set of actions are performed. In the phases given below, first one deals with the symbols that are used in the consecutive phases. Initialization happens in the second phase, here the user is assumed to be registered in the database. Thus, when user enters his credentials it is checked with the database. In the third phase, time factor-based challenge is generated by the server. In the next phases the challenge is encrypted and hashed which is explained in detail below. In the sixth phase the hashed challenge is send as a message to the user in his registered number. In seventh phase the user enters the challenge in the asynchronous token to generate the response. And finally, the user enters the response in his page to access his account. The following sections describe the terms and operations used and performed in this protocol.

### Symbols used

Firstly, the following symbols are used to further explain the steps involved in MFA. The symbols are mainly for username, password challenge generated using time-factor, hashed function, encrypted algorithm and decrypted algorithm:

* Uname : Username
* Upwd : Password
* Upin : PIN-code
* T : Time factor
* R : Random number
* C : Challenge
* H() : Hash Function
* HC : Hashed Challenge
* HR : Hashed Response
* Ek() : Encryption algorithm, k is the encryption key
* Dk() : Decryption algorithm, k is the decryption key

### Initialization

The username and password are added to the database prior. The database used is SQLite3 [Sql2019]. When the user enters the credentials, they are checked against the values in the database.

### Time factor-based challenge generation

The MFA [Moh2011] generates a challenge each time when user tries to login. The challenge generated based on T and R. T is the time factor which depends on the current date, time in hours minutes and seconds. R is a randomly generated number between 10000 and 100000000. The integer values obtained from both are added together to generate C. This step gives C.

### Encrypting the generated challenge

For encrypting the challenge key k, is required. For deriving a cryptographic key a Password Based Key Derivation Function (PBKDF2) is used. It uses instance of hash algorithm (SHA-256) [Naf2012], randomly generated salt value, the number if iterations to perform on hash function and instance of backend. The derived key is used for encrypting the challenge. This step gives Ek(C)

### Hashing the encrypted challenge

The encrypted challenge Ek(C), is hashed using hash function. Moreover, its absolute value is also taken to reduce the complexity and length. This value is then sent as SMS to the user. This value can be called HC.

### HC sent in SMS

The encrypted hashed challenge is sent in SMS to the user. Twilio takes care of this step. A Twilio account and Twilio number is required to perform this step.

### HC entered in smart card

User enters HC in an android app in his phone, which acts as the smart card to generate response. User pin (already know to the user) is also required in this step. When user enters the HC, the smart card computes the hash code of HC . This added value is hashed again to generate hash of the response, which is HR

### HR is entered to get access

The HR generated by smart card is entered by the user in the page directed after login page. The sever again takes the hash value of the addition to generate H’R. If HR equals H’R, the user is granted access successfully. Else access is denied to the user.

## Summary

In this chapter each step that is done in the authentication is clearly explained in detail. After initial login, user get the hashed encrypted challenge. The cryptographic key used is derived from password based key derivation function. It’s a key of 32 bits length. The hash function used here is SHA-256.Then this hashed challenge is computed in the smart card to generate the response. The response is also a hashed value. This is used in the next phase of verification. If the response entered by user matches with the response computed by the server, user is successfully authenticated.

# MFA AUTHENTICATION Technologies

This project deals with multi-factor authentication employment along with user credentials, SMS send to user mobile using out-of-band method, time-factor based challenge generation and asynchronous soft token. The multi-step verification is served to increase steps of security. When the user tries to log in to his account along with valid credentials, he is taken to a form on the login page to enter the response key. Simultaneously, a challenge based on current date and time is generated [Naf2012] the server and is sent in an SMS to the user’s mobile phone using out of band method [. The user has a smart card which is his mobile app in which he enters the challenge. The smart card plays the role of the soft asynchronous token. The smart card [Moh2011] computes the challenge and gives back a response. The user then enters it in form in login page and with successful verification he is taken to his account.

## Authentication Factors

The authentication factors that are primarily used in MFA are:

* Authentication by knowledge
* Authentication by ownership
* Authentication by time-factor based challenge

Figure 4.1 shows the various factors that are involved in multi factor authentication. Initially the user logins with his credentials, thus authentication by knowledge is met. Since user is the only one who knows the credentials, thus authentication by knowledge is performed here. Knowledge based authentication involves mainly password, username and user-pin. When user credentials are validated a time factor-based challenge, which depends on the current date and time, is generated [Zha2016]. The generated challenge is sent to users registered mobile number using out of band method. When user receives the SMS, he enters the challenge to an asynchronous soft token which he owns. The token generates a response for the entered key which the user uses for the final authentication method to access his account.



Figure 4.1 Multi-factor Authentication Technologies [Tec2019]

## Multi-factor Authentication Technologies

The technologies that are involved in the multi-factor authentication primarily are software based asynchronous tokens which is a mobile application and out of band method to send SMS to user mobile. When the user tries to log in to his account along with user credentials, he is taken to a form on the login page to enter the response key. Simultaneously, a challenge based on current date and time is generated by the server and is sent in an SMS to the user’s mobile phone using out of band method [Moh2012]. The soft token computes the challenge and gives back a response.

### Client-side template

Python [Pyt2016] with flask framework using HTML5 is used to develop the client-side template. Flask is a framework used to develop web application using Python. Initially Flask module is imported, and an instance of Flask class is created. A default page is then routed and when the user goes to this default page a defined function is activated.

### Server-side business logic

Python with flask framework using HTML5 is used to develop the server-side business logic. The following are the packages that are used to develop the business logic:

* Twilio
* Cryptography
* Sqlalchemy
* Tabledef
* OS

### Asynchronous software-based token

The asynchronous security token application generates the response key. The token are frequently used in MFA [Moh2011], where devices such us smartphones, tablets or laptops the plays the role of authentication by owner ship. In this project the asynchronous token application is developed in android studio. An Android Package (APK) file is used for the installation of the application on to the mobile device.

### Twilio helper library to generate SMS

If the credentials are authenticated, an SMS with the challenge is sent to the user. The challenge is generated based on time factor and randomly generated number. The SMS which carries the challenge is send to user using Twilio. The REST API of Twilio is used to send outbound messages. The content of the SMS has the challenge generated by the server.

## Summary

This chapter deals with the technologies that are involved in the multi-factor authentication. It primarily with software based asynchronous tokens and out of band method to send SMS to user mobile. When the user tries to log in to his account along with user credentials, he is taken to a form on the login page to enter the response key. Simultaneously, a challenge based on current date and time is sent in an SMS to the user’s mobile phone using out of band method. Using out of band method a peripheral communication method is applied using a different communication channel. The SMS which carries the challenge is send to user using Twilio. The REST API of Twilio is used to send outbound messages. The soft token computes the challenge and gives back a response.

# mfa Implementation

In order to practically implement MFA, a personal computer (PC), an asynchronous token and a mobile phone is required. Here a mobile application act as the

# MFA implementation

In order to practically implement MFA, a personal computer (PC), an asynchronous token and a mobile phone is required. Here a mobile application acts as the PyCharm is used as the IDE for developing Python code in flask framework. Twilio helper library for Python is used to send message to outside mobile number. When message must be sent, HTTP POST (Hypertext Transfer Protocol) request is made from the Twilio number to the API. Android Studio [And2019] is used as the IDE for developing the asynchronous token which is a mobile device. The following sub-sections explains in detail the hardware and software specifications required.

## Specifications of server and client

A computer plays the role of server and client. Python with flask framework using HTML5 is used to develop the server and client-side business logic and template. The following are the hardware and software specifications of the client and server used for implementing and running MFA.

### Server and client hardware specifications

Processor : Intel Core-i7

Speed : 2.97 GHz

RAM : 8.00 GB

System type : 64-bit Operating System

### Server software specifications

IDE : JetBrains PyCharm 2019.1.1 x64

Languages : Python 3.4

Framework : Flask

Database : SQLite3

Libraries : Twilio, Cryptography

OS : Windows (>XP)

### Client Software Specifications

Browser : Google Chrome (>4.0)

## Specifications of asynchronous token

Here an application in the mobile device is used as the asynchronous token. The application is developed in android studio. It runs in all android device with android version 9. An Android Package (APK) file is used for the installation of the application on to the mobile device.

### Specifications of mobile device

API level : 28

Version : Android 9.0

CPU : x86

Disk Space : 9.0 GB

## Generating Time- factor based Challenge

The Figure 5.1 shows how a time-factor based challenge is generated, initially integer value of current date and time is chosen. It is then added to a random value between 10000 and 100000000. This value is then encoded into bytes and an instance of PBKDF2 [Naf2012] is generated. It then uses instance of hash algorithm (SHA-256). The value of the key is then made URL (Uniform Resource Locator) safe base64 encoded key. The final step is encrypting the challenge.

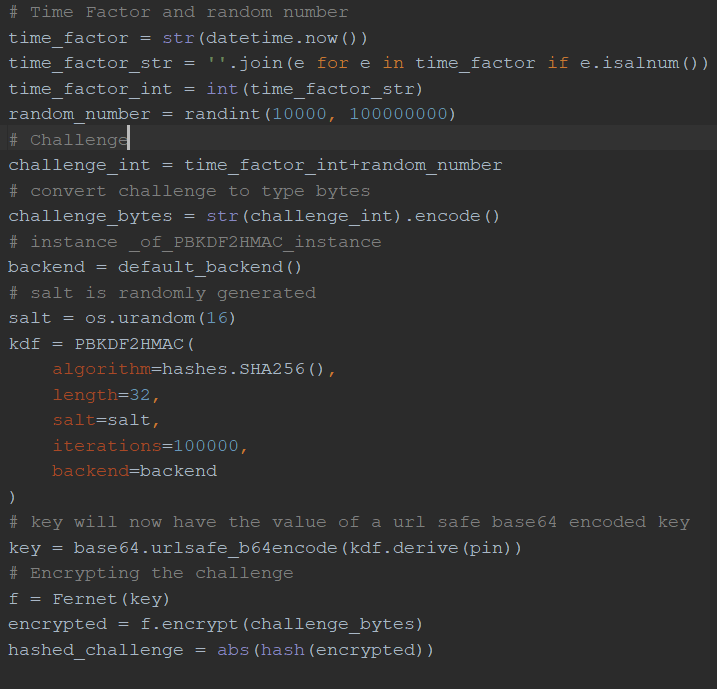


Figure 5.1 Time-factor based challenge generation

## Summary

In this, the software and hardware specifications that are required for demonstrating and implementing MFA is primarily focused. If these specifications are met, MFA can be easily demonstrated. An android phone is mainly required to work asynchronous token. Android studio is used to develop the mobile application which serves as the asynchronous token. Here a mobile application acts as the asynchronous token and user’s PC acts as the interface for the user and his mobile phone acts as the source to receive the message. PyCharm [Pyt2019] is used as the IDE for developing Python code in flask framework. Twilio helper library for Python is used to send message to outside mobile number. The role of PyCharm as the IDE is significant to develop the python code in flak framework. Twilio helper library is used to send message service to the users’ phone. And the final verification is done at the response key checking form on the initial login page in the web browser.

# outcomes of MFA

## Login Page

If the user tries to login to the account, the login page is displayed. Figure 6.1 is the screenshot of the login page. Here user enters the user name and password. Password should be the combination of alphanumeric characters and be at least 6 digits long.

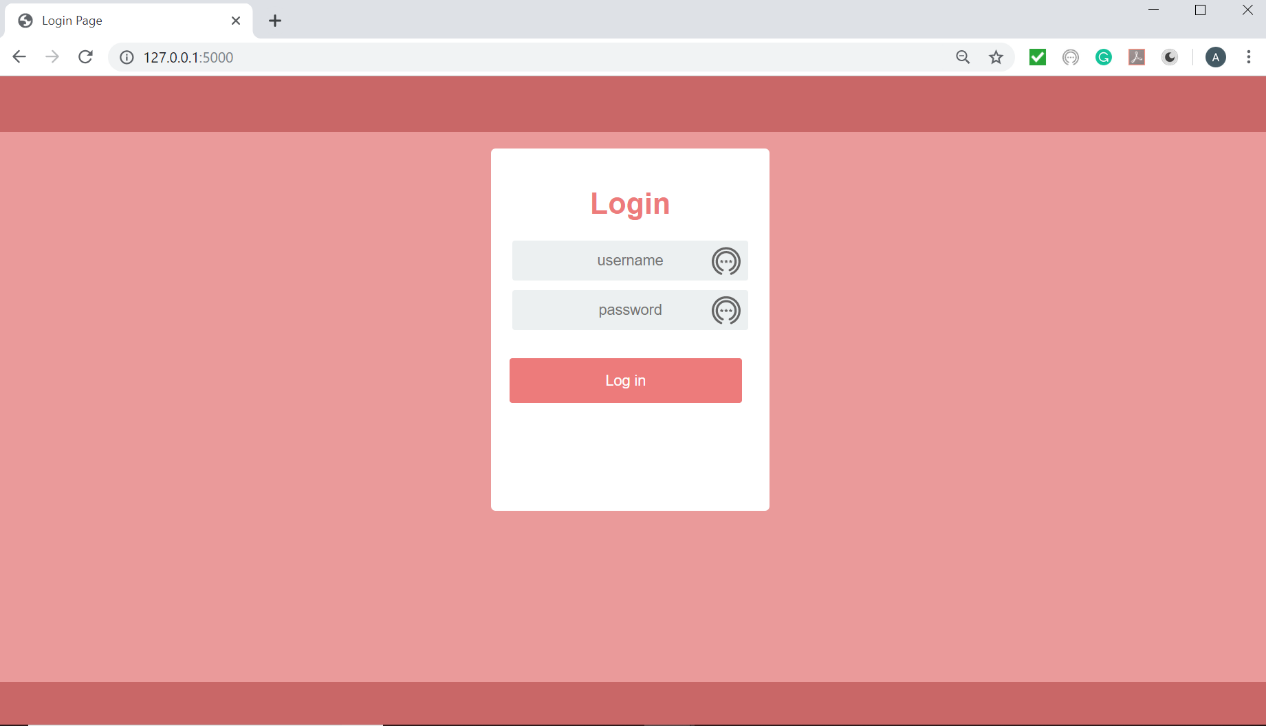


Figure 6.1 Login page

If invalid credentials are entered by user, “an invalid credentials” message is displayed on the screen. Figure 6.2 is the screenshot of invalid attempt by user. This warns the user about the invalid attempt he had made.

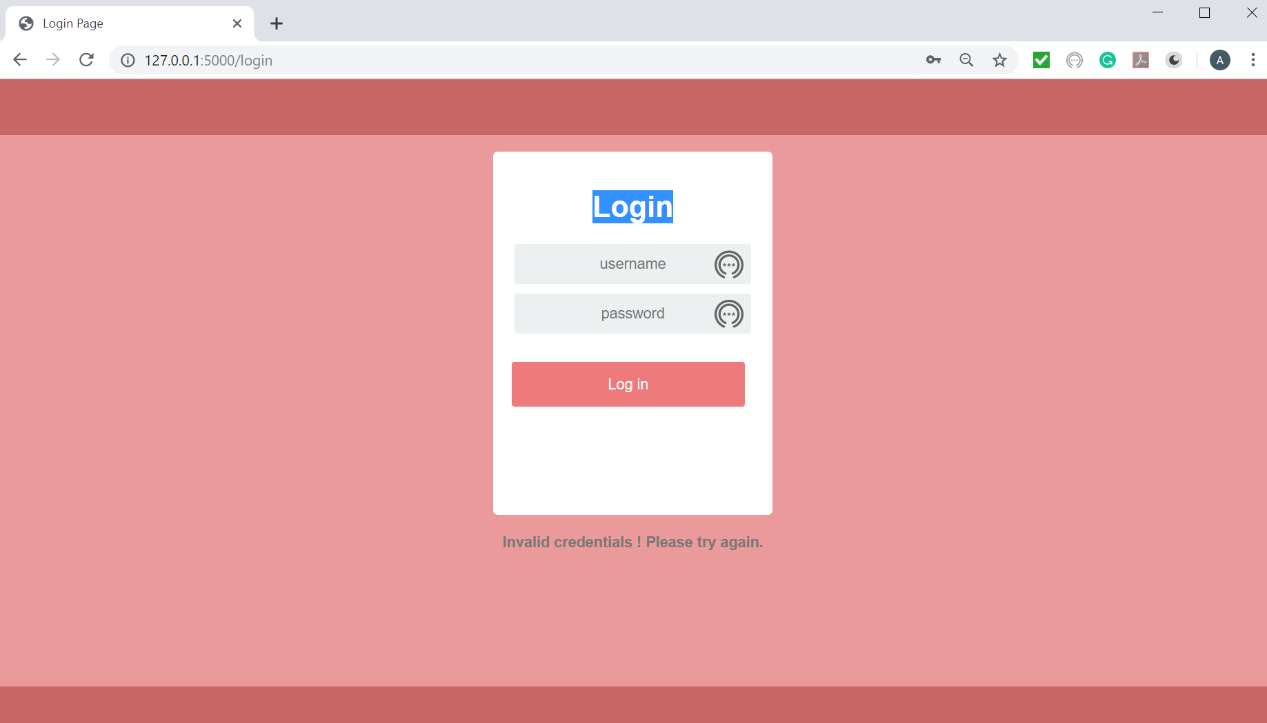


Figure 6.2 Invalid credentials

## SMS send to users’ phone

If the credentials are authenticated, an SMS with the challenge is sent to the user. The challenge is generated based on time factor and randomly generated number. The combined time factor and random number is then encrypted. The encrypted number is then hashed to generate the challenge.

The SMS which carries the challenge is send to user using Twilio. The REST API of Twilio is used to send outbound messages. Figure 6.3 is the screenshot of the SMS received on the phone from the Twilio number created [Twi2019]. Twilio can be used to send messages around the globe. The content of the SMS has the challenge generated by the server. This challenge is then entered to the smart card to compute the response. While user types the challenge, he has to be careful about typo error.



Figure 6.3 SMS sent from Twilio account

## Track Delivery Status of Message Sent

If you track the status of the message sent, Figure 6.4 is obtained. It shows the status of the message as delivered. The number to which the message is sent, the number from which the message is sent. Message status id and account id is also given while observing the status. Hookbin [Hoo2019] service is used to view this HTTP request.

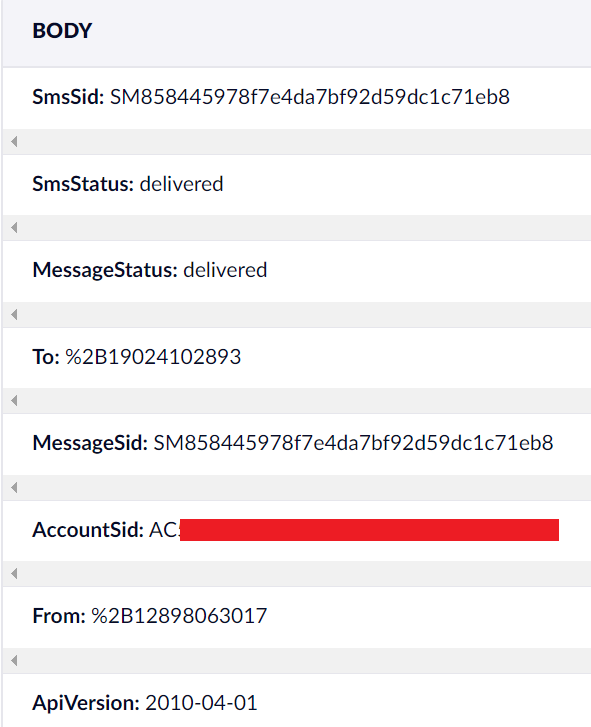


Figure 6.4 Status call back of SMS sent

## Asynchronous Token

Once the user gets the challenge in the message. He enters this in the smart card. The smart card computes the response key with the hash value of the challenge given. Figure 6.5 is the screenshot of the mobile screen with the smart card.

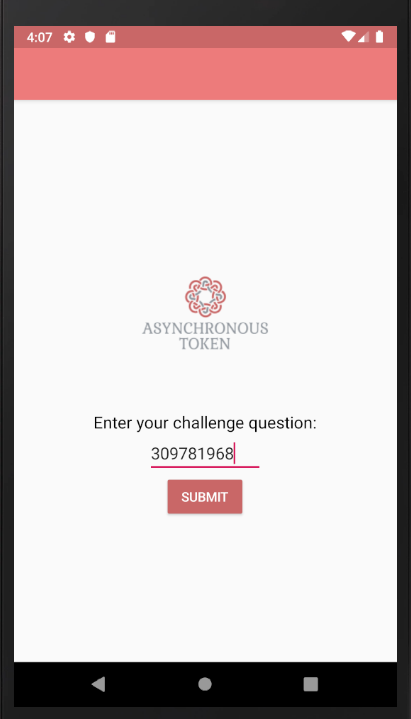


Figure 6.5 Asynchronous token challenge entry screen

The smart card is developed in android. It has two screens , first one is the screen to enter the challenge and second screen gives back the computed response. Figure 6.6 is the second screen with the computed response key of the challenge.

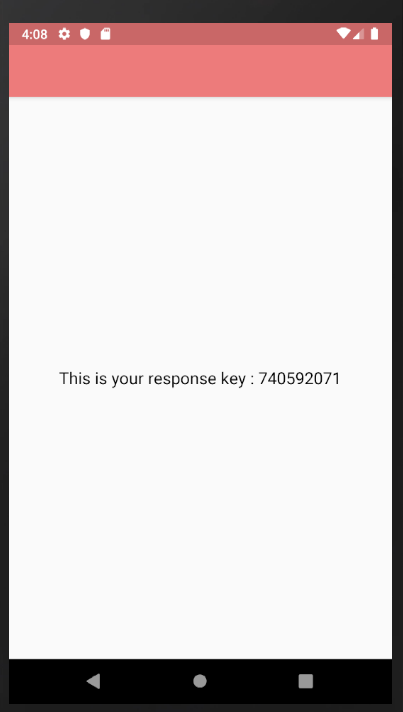


Figure 6.6 Response screen of asynchronous token

## Login Page with Response Form

Once the user gets this, he enters the response key in the response form of login page. Figure 6.7 shows a response form of the login page, which has a provision for input.

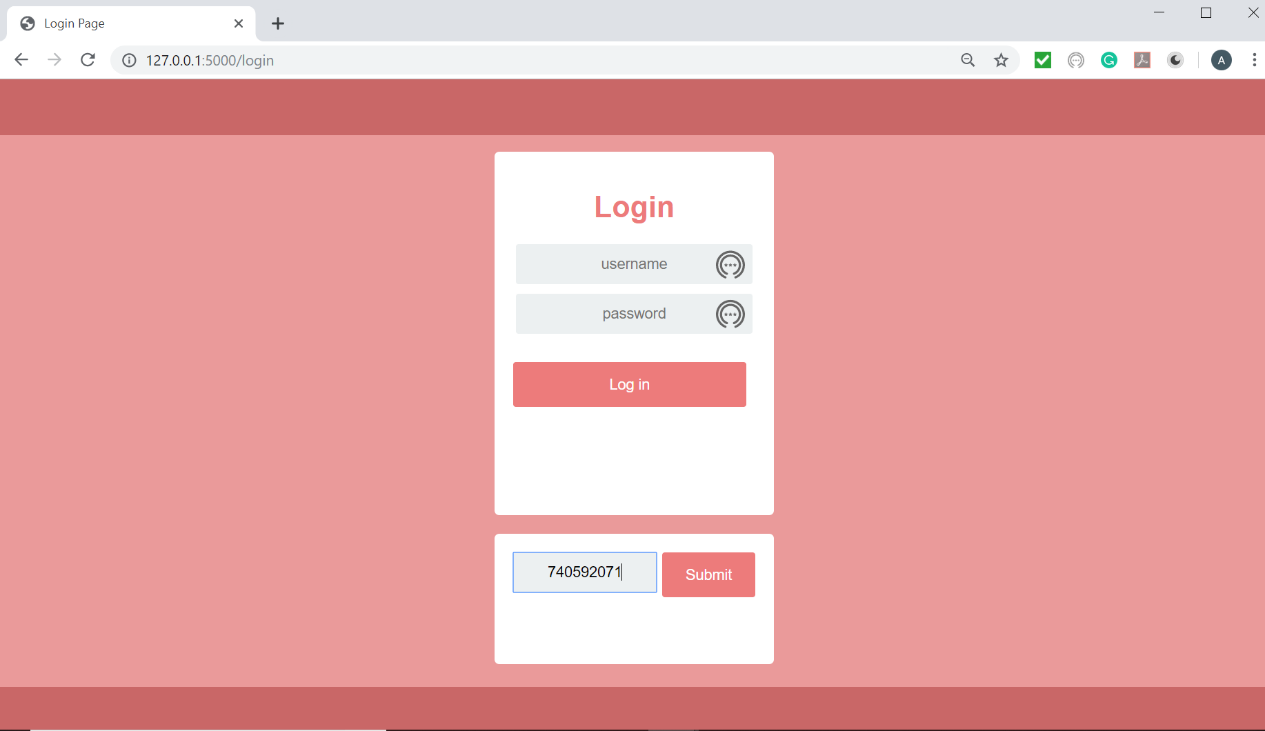


Figure 6.7 Login page with response form

User enters the response key obtained from smart card in this column in response page. If the response key entered by user doesn’t match with the response key computed by server, then an error message is displayed. Figure 6.8 is the response page with error massage due to mismatch in the response key.

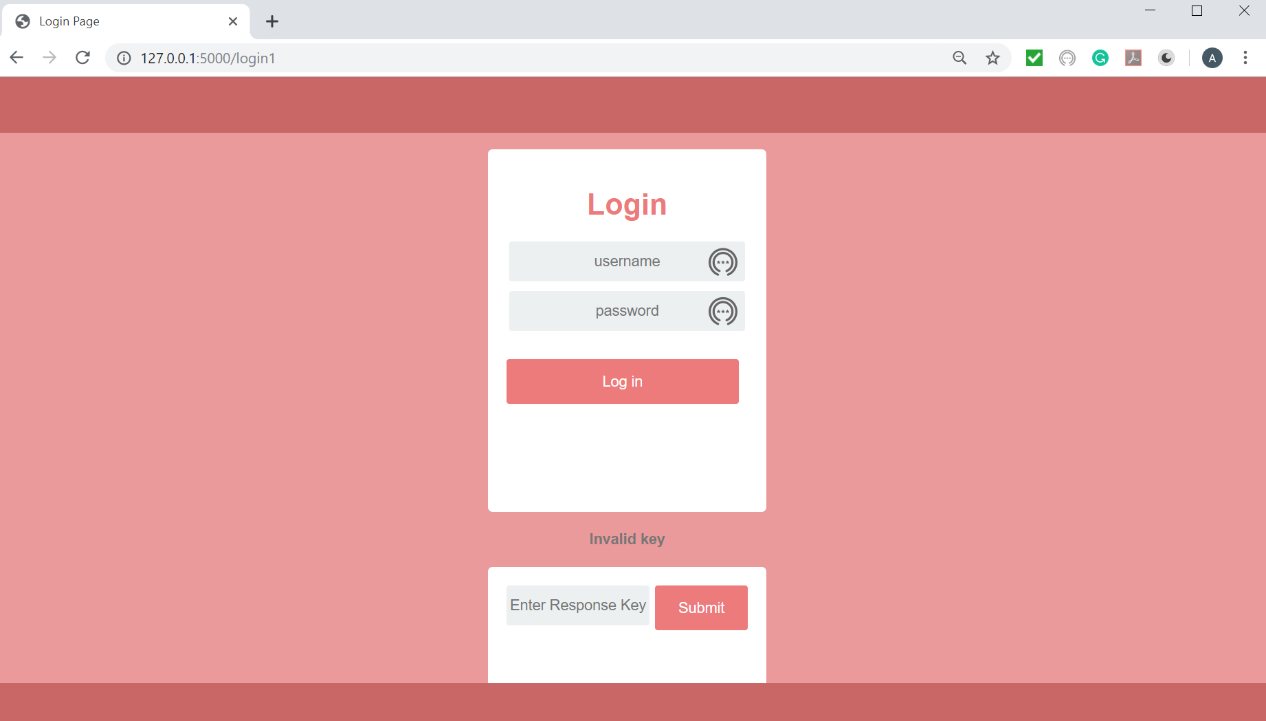


Figure 6.8 Response form with error message

If the response key entered by user matches with the response key computed by the server, then the user is authenticated, and he is taken to the home screen. Figure 6.9 is the screenshot of the home screen where an authenticated user is taken to.

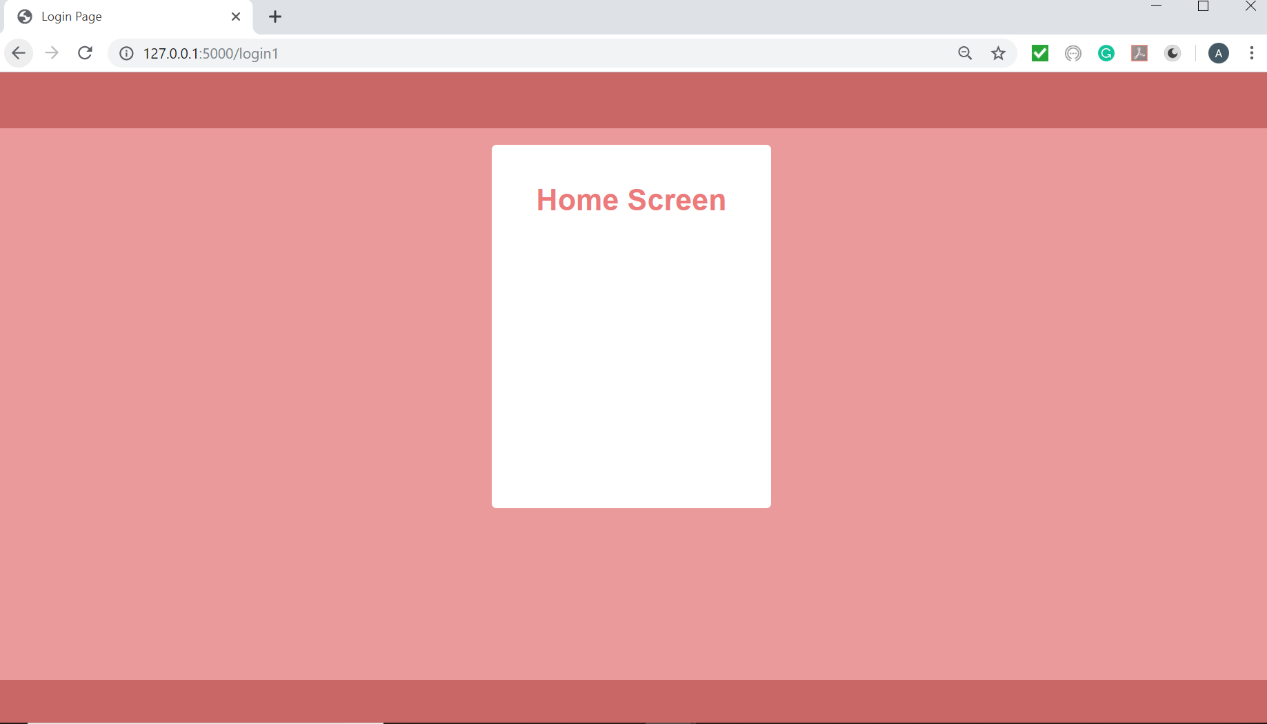


Figure 6.9 Home Page

## Summary

In this chapter screenshots of the implemented sections in MFA is given. After initial the initial login, which is explained in the first figure. User gets the hashed encrypted challenge. The cryptographic key which is 32 bits length used is derived from password based key derivation function. This hashed challenge is send your users phone, which is shown in the second figure. The user then enters the challenge in his smart card to generate the response.

# conclusion and recommendations

This project deals with multi-factor verification employment along with an asynchronous token. A three-step verification is served to increase steps of security.

## Conclusion

When the user tries to log in to his account along with valid credentials, he is taken to a form on the login page to enter the response key. Simultaneously, a challenge is sent in a SMS to the user’s mobile phone. The user has a smart card which is his mobile app in which he enters the challenge. The smart card computes the challenge and gives back a response. The user then enters it in form in login page and with successful verification he is taken to his account.

The following attributes are achieved with MFA:

* Authentication by ownership
* Authentication by knowledge
* Reduce security breach
* Increase the levels of security

However, it also infuses limitations for the user, which are:

* User wants to carry synchronous token always.
* Attacker can easily crack the code used to generate the response key since authentication is not required in smart card this process.
* User needs to memorize the credentials
* Delay in receiving the SMS on users’ phone due to network connectivity can also affect authentication.
* Asynchronous token as an android application, faces issues due to native in nature.

Sometimes user also finds MFA time consuming since, multiple number of steps are involved in the authentication process. However, if you compromise in the number of steps for authentication and time consumed, then you should also compromise in the level of security.

## Recommendations

After analyzing all the merits and demerits of using MFA, certain suggestions are recommended that can increase ease of user to access the desired resources. An authentication step must be added on the asynchronous token, thus only after the entry of the valid credentials the user should be able to enter the challenge question. Response generated should also be related to the hashed value of the challenge as well as the credential entered by the user [Zha2015]. The delay in SMS received due to network connectivity can be solved by providing an alternate method to receive the challenge. There should be a provision to send the challenge in email if users prefer that way. Thus, network connectivity issues can be met to an extent. The native nature of asynchronous token developed in android can be solved by developing asynchronous token as hybrid application. Thus, user with Android or iOS can run the mobile application to compute the response key.

Furthermore, other measure that can improve the security in MFA are encrypting and hashing the entries in the database. Thus, attacker find it difficult to alter the database entries. Limiting the number of attempts to login can also reduce security breach to a greater extend. The account of user should be locked after certain number of failed login attempts, in addition an alert message should also be send to the users registered number.

# REFERENCES

[And2019] Android Studio, “Download android studio and sdk tools”, <https://developer.android.com/studio>, June 12 2019.

[Ann2014] Annual Authentication Survey, “SafeNet News Authentication Survey”, <http://cn.safenetinc.com/news/2014/authentication-survey-2014-reveals-moreenterprises-adopting-multi-factorauthentication/?LangType=2052#sthash.dSnZBDMn.dpuf>, May. 21 2019.

[Chh2012] Randeep Kaur Chhabra, Ashok Verma, “Strong authentication system along with virtual private network: A secure cloud solution for cloud computing”, International Journal of Electronics and Computer Science Engineering. Volume-1Number-3, April 2012.

[Hoo2019] Hookbin Service, “Capture and Inspect HTTP request”, <https://hookbin.com>, May 23 2019.

[Jet2019] JetBrains PyCharm, “The python IDE for professional development”, <https://www.jetbrains.com/pycharm>, April 16 2019.

[Moh2011] Mohamed Hamdy Eldefrawy, “OTP-Based Two-Factor Authentication Using Mobile Phones”, Eighth International Conference on Information Technology, April 2011.

[Naf2012] Kawser Wazed Naïf, Tony Shekhar Kar, Sayed Anisul Hoque, Dr. M. M. A Hashem. “A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, June 2012.

[Pyt2016] Python Spot Online, “Flask getting started”, <https://pythonspot.com/flask-hello-world/>, March 12 2019.

[Sar2012] Maninder Singh, Sarbjeet Singh, “Design and Implementation of Multitier Authentication Scheme in Cloud”, International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012.

[Sql2019] SQLite3, “Using SQLite 3 for Flask”, <http://flask.pocoo.org/docs/1.0/patterns/sqlite3>, April 4 2019.

[Tec2019] Tech Target, “Multi Factor Authentication”, <https://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA>, July 10 2019.

[The2019] The SQLAlchemy, “Python SQL Toolkit”, <https://www.sqlalchemy.org,> May 17 2019.

[Twi2019] Twilio online , “Twilio SMS python QuickStart”, [https://www.twilio.com/docs/sms/quickstart/python,](https://www.twilio.com/docs/sms/quickstart/python,January) June 2 2019.

[Zha2015] Guifen Zhao, Ying Li, “Asynchronous Challenge-Response Authentication Solution based on Smart Card in Cloud Environment”, International Conference on Information Science and Control Engineering, April 2015.