National College of Ireland

Protecting speech-based One – Time passwords from man in the middle attacks

MSc Research Project
Cyber Security

Pratik Prakash Raut
Student ID: 20185847

School of Computing
National College of Ireland

Supervisor: Prof. Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Pratik Prakash Raut
Student ID:	20185847
Programme:	MSc in Cyber Security **Year:** 2022
Module:	MSc Research Project
Supervisor:	Prof. Niall Heffernan
Submission Due Date:	18/09/2022
Project Title:	Protecting speech based One – time password from man in the middle attack
Word Count:	6402 **Page Count:** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Pratik Prakash Raut

Date: 18/09/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Protecting speech – based One – time passwords from Man in Middle Attacks

Pratik Raut
20185847

Abstract

Today, as the world is witnessing the evolution of cyber security in all aspects of our digital world, more sophisticated authentication mechanisms are coupled with the conventional password-based systems. These include one-time passwords, passphrases, token-based cards and other biometric schemes that are used to enhance or add an additional layer of security. But these authentication mechanisms are still subject to attacks such as keylogging, brute-forcing, shoulder surfing, and the man in the middle technique. One of the most popular and widely used two-factor authentication system is the One-time password (OTP) authentication mechanism. To prevent keylogging and shoulder surfing attacks, it is proposed that the traditional One-Time password (OTP), which is normally entered using a keyboard on the user machine, be spoken out by the user on its microphone to prevent key logging attacks. Furthermore, speech recognition module can be applied to convert the decrypted user voice sample into the One – Time Password (OTP) text, thereby reducing the possibility of shoulder surfing or keylogging. The speech sample will also be encrypted and transmitted to the server side to prevent any man-in-the-middle attacks from taking place. The main focus would be on the encryption of the voice samples and recognition of the digits spoken out by the user using an accurate speech recognition technique to prevent any delay in the authentication process and avoid the loss of data. The paper proposes to introduce a scheme of encryption with speech recognition over the One-Time Password (OTP) based systems, which improves the security of the application without compromising on the user experience. Although many encryption and speech recognition approaches are accepted as industry standard, the proposed system comprises of the python speech recognition library and the implementation of monoalphabetic cipher algorithm, which is in line with the requirements of the proposed authentication solution.

Keywords: OTP, Multi Factor Authentication, Speech Recognition, Key logging, Shoulder surfing, Encryption, Monoalphabetic cipher, Man in the middle attack.

Contents

1 Introduction

The process of identifying or validating the identity of a user who claims to be authorized to access any application is called as authentication. Authentication systems are utilized to limit the access of the users who has been granted permission to use the services offered by the system. Users are normally granted access to the system if the user credentials like username and password which are stored in a database are cross verified to check the identity of the user. If this successfully matches, the user is granted access. The Authentication mechanism are used to provide security to applications such as banking apps, social media sites, and various online services which are dependent on online accounts so that only authorize user can access the profile and use the application. In the recent past, traditional authentication methods of password or passphrase have been improved with addition of new authentication schemes such as one-time passwords, biometrics, push notifications, token-based authentication schemes, captcha-based schemes etc. The fundamental of any authentication method are based on something that a user might be in possession of, in addition to anything

that a user might know or be able to recall. In the case of passwords, for example, Authentication systems like passwords require users to remember the phrase assigned as a password. The term multifactor authentication refers to authentication methods that follow a two or three-step approach to verifying the identity of a user. Using a combination of elements like Inherence, knowledge and Possession

For example, Knowledge is the password which is set by the user and possession can be a One-Time Password which is sent to an email of the user or Text message on user's number. Here the authentication scheme takes advantage of the fact that apart from the password known to the user, there can be an email ID or phone number associated with the user, which is only accessible to anyone else except the authorised user. Multi-factor authentication offers an extra layer of security to the authentication mechanism. Attackers have used various attacks such as brute force, SQL injection, keylogging, cross-site scripting, etc. To gain access to users account and data over the years. With this there are some attacks which hackers use to steal the data like Shoulder surfing and man in the middle attack. A shoulder surfing assault occurs when a hacker is able to have a physical access like a view to the screen or keypad to gain personal information. But to perform the attack the attacker should be in close proximity of the user. A man in the middle (MITM) attack is said to be executed when an attacker intercepts the communication channel between a user and an application, either to listen the conversation or to act like one of the parties, making it appear as if a normal data exchange is taking place ('What is MITM (Man in the Middle) Attack | Imperva', no date) An attack's aim is to steal information such as login credentials, account details, and credit card numbers. The most common targets are users of e-commerce sites, financial apps, and other sites that require users to sign in. This study will propose the use of two factor authentication, which is a One-time password in which random characters will be generated to enhance security and voice recognition in which the Voice sample will be converted into One Time Password (OTP) that will possibly prevent attacks like brute forcing, shoulder surfing, and keylogging. The voice sample of One-time password (OTP) will be encrypted using the monoalphabetic substitution cipher before transmitting it over the network while keeping in mind the time and data integrity constraints to prevent the man in the middle attack

1.1 Research Question

How can we protect the speech based One-Time password (OTP) from man in the middle attack using encryption technique?

The structure of the research paper has been discussed below.

Section 1 consist of introduction to the domain and topic is provided with the motivation for choosing the topic. Additionally, the introduction contains a brief explanation of how the system works. In the second section, which is dedicated to the literature review, the various research works that were read in order to obtain a better understanding of the subject matter are reviewed critically. Also, you will find a list of all the papers that were used as references when doing the research. Section 3 comprises of all of the steps that were taken while conducting the research as well as the decision which were made when working toward the conclusion of the research. Additionally, the tools and programming language that were utilized during the development of the artifact can be found in this section. In Section 4,

you'll see a graphical representation of the architecture of the suggested model that breaks down each component individually. Diagrams in the format of the Unified Modelling Language (UML), Sequence Diagrams, and Use Case Diagrams have been displayed in this area. The implementation step is discussed in Section 5, and during this stage, the final artifact and the functionalities of the component are summarized. In Section 6 which is Evaluation, number of different scenarios and evaluation criteria have been applied in order to evaluate how effective the suggested model is. The conclusion of the article can be found in Section 7, where it summarizes the most essential issues discussed throughout the rest of the document, as well as the areas in which more enhancements may be done in the near or far future to make the suggested model even more effective. The list of references that have been utilized for the paper may be found in the very final section.

2 Related Work

There have been different types of Authentication mechanisms developed for Authorizing a user. This Authentication mechanism was based on Token, Passphrases, Use of biometrics like the retina, fingerprint etc, Voice and Passwords. Every authentication mechanism has its own benefits and drawbacks. Therefore, this authentication mechanism is vulnerable to attacks like Brute Force, SQL injection, Cross-site Scripting and Man in the middle attack. In the upcoming section we will look into the previous work of Authentication mechanism, Speech recognition and Encryption techniques.

2.1 Authentication mechanism

The paper (Lamport, 1981) highlights the drawbacks of the Authentication mechanism in which the very first vulnerability discussed is the password which is been stored in plaintext as the password was not encrypted the attacker was able to read the password. The next discussed is the attacker was able to intercept the communication channel and with the help of Man in the middle attack his was able to access the password and the last one is where the password used is very weak and easy to guess so by using a brute force attack the attacker was able to guess the password. The proposed solution for such attack in (Lamport, 1981) was to use an encryption scheme to encrypt the password which could provide a security level to different types of attacks. But still the passwords based authentication can still be vulnerable to attacks. As stated in (Pandya, Jhajj and Pawar, 2017) as there is progress in cyber security domain the attacks are also more advanced.

2.1.1 Passphrase Authentication

As the Passwords are highly vulnerable to brute force type of attacks. Passphrase where an ideal replacement for passwords. In (Yan et al., 2004) they found out that users are creating a passphrase which have keyword present they are likely good in converting short term memories into long term. As mentioned in (Yan et al., 2004) it is much more easier to remember a passphrase than a password. So, if we consider a sentence that can be used as a password is said to be a passphrase. As people can use normal sentences as passphrase it can be considers much easier to remember as compared to a password. Likewise, it has its on disadvantages where the user has to remember a lot of words. (Shay et al., 2012) study states

that the entropy of a three to four word used as a passphrase is same as that of a password which has four words. The paper (Shay et al., 2012) also highlights the policies used which are very strict for authentication scheme. Because of which the passphrase authentication was difficult to use as there were more error while login, users were having difficulties in memorising the passphrases. There were more spelling mistakes. Therefore, the passphrase authentication system become more difficult. To address this issue there was a additional help to remember the passphares with the help of a hint. (Addas, Thorpe and Salehi-Abari, 2019) suggested geographical hints so that the user can recollect the right passphrase easily.

2.1.2 Geographical Authentication

Using a image for authentication instead of password or passphrase has been possible. In this authentication system users doesn't have to write a password or passphrase they just use a image or a graphical object for authentication which make its more easier and save time while authentication. (Biddle, Chiasson and Van Oorschot, 2012) states that the use of Geographical authentication is a better system as the users are able to remember the image much more easily as compare to passphrase or a password. And as images are used for authentication the attacks like brute forcing or keylogging are been eliminated. (Biddle, Chiasson and Van Oorschot, 2012) highlights three different types of graphical authentication mechanisms which are Recall , cued point and recognition. They key advantages of this authentication systems are the image which is used can be replayed and new image can be choose as and when required which eliminates the threat of shoulder surfing. As the image can be changes after certain login attempts. But there are some drawbacks as well. Images can sometime be hard to read. This system requires more computing power because image processing is involved. Also when the image is transferred over the network it should be transferred in a secure way which requires more computing power with high speed data transfer rate which is dependent on the size of the image. Therefore, Geographical Authentication system may require many resources for a single login.

2.1.3 Token based authentication

Token based authentication system is normally used to strengthen the authentication system as an extra layer of protection. Token based authentication has changed over years. This scheme gives users the liberty to use the password once and then forget it as the password that has been generated won't be used again. As specified by (Aloul, Zahidi and El-Hajj, 2009) this authentication scheme add up more randomness and increases the entropy making it more difficult for an attacker to predict the password (Tanvi, Sonal and Kumar, 2011). This helps in generating random numbers with time stamps to expire the token after a certain duration to prevent any unauthorised use of the token. But as highlighted in (Tanvi, Sonal and Kumar, 2011) This token generation requires a hardware which can be costly.

2.1.4 One Time Password

Lamport proposed a authentication mechanism which was one time password. (Lamport, 1981) proposed a hash-based authentication scheme which used dynamic passwords. Later (Groza and Petrica, 2005) proposed numbers to replace the hashes. The proposed

methodology would generate random numbers this was possible because of function which would repeat cyclically. As the number were generated randomly it was possible to generate a greater number of passwords which made it difficult to bypass the mechanism. But as there where a greater number of passwords the computing power and time required to authenticate the user was considerably increased. For the drawbacks in (Lamport, 1981) proposed model (Eldefrawy, Alghathbar and Khan, 2011) proposed a solution in which the entropy or randomness would increase by verifying the seed values which was been fed into the function. But this was not convenient because it required users protocol initiation. To address this flaw (Groza and Petrica, 2005) proposed functions which were performed on groups of composite integers. But this required more computing power. (Yassin *et al.*, 2013) proposed a a One-time password (OTP) system which was using cloud architecture. Eliminating the requirement of a external device. While encrypting the OTP with RSA algorithm and using a asymmetric encryption to protect the OTP. (Cheng, 2011) put forth the use of mobile and cloud together which also removes the dependency of a single device reliance. In the proposed solution a proxy server is used to store public key whereas a different server is used for generation of OTP. But as there is advance in the cyber security domain an extra layer of security is very important.

2.2 Voice Recognition

There have been various Speech recognition modules which are used to authenticate an OTP. In this kind of authentication, a user request for an OTP through his phone and gets an audio OTP. But first the device, username and password have to be authenticated to request an OTP. (Cha, Kim and Kim, 2011) proposed a system which combines OTP and Voice recognition model. (Cha, Kim and Kim, 2011) paper have addressed the challenge of lowering the dispersion, voice and integrated the chaotic signals to normalise the voice graph. But as the noise levels can differ the normalization would lead to more time and computing power. Also there is a challenge with the speech recognition module to understand the pronunciation or insufficient feedback. (Ping, 2021) addressed this issue by use of MATLAB which collects the speech data and performs a pre processing to reduce the time dependency. (Ping, 2021) used a HMM model for speech decoding and calculated the recognition of speech with the help of posterior probability which signifies reduction of the algorithms runtime. (Bava, 2020) used speech recognition for giving the input as voice instead of typing out the message which tackled the security concern of shoulder suffering and keylogging. But the prosed idea didn't include the encryption of the voice sample and it was possible for attacks to use man in the middle attack and sniff the communication channel to access the data which is been transmitted.

2.3 Encryption

Since the voice sample of the user trying to authenticate itself is transmitted from the client machine to the backend server is prone to Man in the Middle attack, it is paramount to use a strong encryption algorithm to prevent any form of interception by an unauthorised malicious hacker.

Hence, the monoalphabetic substitution algorithm is applied on the voice sample to secure the channel prior to encryption, thus making the voice sample incomprehensible.

There are mainly 2 forms of speech encryption, namely digital and analog. Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are two of the most widely used speech encryption algorithms. These algorithms are relatively more sophisticated in implementation and also required a much larger bandwidth for transmitting the media over a secured channel(Ambika and Radha, 2012). Several speech encryption algorithms including blind source separation-based approaches are used in the current world. (Goldburg, Sridharan and Dawson, 1993) Chaotic maps encryption is also very common for encrypting voice and image data. Being erratic and being sensitive to the initial conditions make the chaotic systems a popular choice to create cryptosystems. Extensive research has also been carried out in this area for this reason.

In cryptography, a substitution cipher is a technique to convert raw data to cipher data, with the help of a key. (Vatsa, Mohan and Vatsa, 2012)The same key is used to decipher the message by performing the inverse substitution method to extract the original message. The raw audio file recorded by the user in our case is read byte-by-byte and converted into a ciphertext using this algorithm. The monoalphabetic substitution system is an encryption technique wherein every occurrence of the plaintext message is converted into a cipher-text. This method is also known as the cyptogram and is based on a key that is used for the rearrangement of letters. The same key is also used to decrypt the the encrypted message on the receiving end. The same letter in the original message will always be substituted with the same corresponding cipher-text letter.(Vatsa, Mohan and Vatsa, 2012) Although the messages encrypted using the monoalphabetic substitution cipher mechanism can be broken using frequency analysis, this method provides an advantage over other encryption algorithms being fast and robust, thus providing a better user experience and significantly faster turnaround times. On the other hand, polyalphabetic cipher techniques substitute different occurrences of the same plaintext letter by different ciphertext letters. The cryptoalphabet to be used for each occurrence of the original message is guided by a key. The number of letters encrypted before a polyalphabetic substitution cipher returns to its first cipher alphabet is called its period. A larger period indicates a stronger cipher. (Vatsa, Mohan and Vatsa, 2012)Even though polyalphabetic substitution cipher algorithms are comparatively more difficult to decipher without the presence of the key, it takes a longer time for encryption and decryption. Due to this drawback, it is difficult to use this mechanism in real-time system which needs synchronous encryption and decryption in an application.

3 Research Methodology

The approach proposed in this paper primarily focuses on speech recognition and encryption of the OTP as an additional layer of security. This solution will mainly help to prevent some of the most popular attacks associated with passwords and pass phrases, namely key logging, shoulder surfing and man in the middle. The implementation comprises mainly of 3 components. The first component of this application will run on the client machine (user computer) and will be used to record the voice sample of the user input speech and save the encrypted voice sample file on the user's machine. This encrypted audio sample cannot be played using any media player unlike the original voice file, which will be recorded in the Waveform Audio file format(.wav). The .wav file will then be encrypted using the

monoalphabetic substitution cipher algorithm, which uses a fixed key embedded within the application to encrypt the contents of the file.
The second component of the application will run on the server side, which will be responsible for decrypting the audio file using the same embedded key. Following this, the third component will be called that will convert the decrypted audio sample (in .wav file format) to the actual OTP in the text format. The converted text will then be pre-processed, removing any additional spaces if introduced during speech to text conversion, and matched with the OTP that was generated by the application corresponding to the authorised user. If the OTP spoken by the user exactly matches the OTP generated for the authorised user, the authentication process will be successful and the user will be granted permission to access the application. To implement this solution, we need to take into consideration 2 main factors discussed below. The primary factor to be considered is the security of the application. This means that the OTP generated cannot be weak, since it can be easily cracked using brute force mechanism. But while considering the security aspect, we also need to take into account the second factor, which is the hassle that the user needs to face in the authentication process. In case there is any loss of data in the encryption/decryption process, or there is a lack of adequate accuracy in the speech recognition module, the user will be denied access to the application resources, even though he has spoken out the correct OTP associated with his account. Hence, the trade-off needs to be handled well in order to create a balance between the security and the user experience. The metric used to consider this trade-off has been discussed in the further sections in detail.

3.1 Development Kit Used

The proposed solution was developed on windows 10 operating system. Flask was used as a python front end framework, sqlite3 was used as a database and the backend scripts were written in python. Open-source python libraries such as yagmail was used to send the OTP on the user email ID. The Speech Recognition python library was used to convert the audio file to text format. The python Random library has been used to generate the random numbers.
In addition to this, the sympy library has been used to implement the monoalphabetic cipher encryption algorithm in python.

3.2 OTP Based Authentication

OTP-based authentication is a type of two-factor authentication that uses a one-time password (OTP) to verify a user's identity. The OTP is generated by an authentication server and is typically sent to the user's mobile phone via SMS or to their email ID through an SMTP server. This OTP received needs to be then entered by the user for successful authentication, in addition to the username and password. OTP-based authentication can be more secure than traditional username and password-based authentication, as it adds an extra layer of security. However, it is important to note that OTP-based authentication is only as secure as the underlying OTP generation and delivery mechanism. If an attacker is able to intercept or predict the OTP, they may be able to gain access to the user's account. The OTP can be any combination of a predefined length of random numbers generated by the system. The OTP is typically six digits long, but longer or shorter OTPs can be used. OTP-based

authentication is used by a number of online services, including social media applications, e-commerce websites, payment portals and many major banking applications.

3.3 Random String Generation

A mathematical function that produces a string of characters chosen at random is referred to as a random string generator. This is accomplished by randomly selecting the characters from a hat.

A mathematical function that produces a string of characters chosen at random is referred to as a random string generator. This is accomplished by randomly selecting the characters from a hat. Typically, the function is utilized when generating passwords or other string-based data that must be challenging to figure out. This data must be secure and cannot be easily guessed. Employing a random number generator to select each character in a string as one's method of choice when generating a random string is by far the most frequent method. You can choose the characters from a larger group, such as all the letters in the alphabet, or from a smaller group, such as all the letters except for the vowels or only numbers. Utilizing a cryptographic hash function is yet another method that can be used to generate a random string. A mathematical function known as a cryptographic hash function is one that accepts data of any length as input and generates data of a predetermined length as output. Hash functions in cryptography always provide a random output, regardless of whether or not the input data is random. These are the two most common methods for generating a random string, but there are many alternative methods available.

3.4 Encryption and Decryption

The encryption and decryption of audio files refers to a process that is used to protect the information contained in the files from unauthorized access. The process typically involves the use of a key that is used to encrypt the file, and then a second key that is used to decrypt the file. The keys can be generated using a variety of methods, including public-key cryptography. The algorithm used in the implementation of the application is the Monoalphabetic Substitution Cipher. In cryptography, a substitution cipher is a technique to convert raw data to cipher data, with the help of a key. The same key is used to decipher the message by performing the inverse substitution method to extract the original message. The raw audio file recorded by the user in our case is read byte-by-byte and converted into a ciphertext using this algorithm. The monoalphabetic substitution system is an encryption technique wherein every occurrence of the plaintext message is converted into a cipher-text. This method is also known as the cyptogram and is based on a key that is used for the rearrangement of letters. The same key is also used to decrypt the the encrypted message on the receiving end. The same letter in the original message will always be substituted with the same corresponding cipher-text letter. Although the messages encrypted using the monoalphabetic substitution cipher mechanism can be broken using frequency analysis, this method provides an advantage over other encryption algorithms being fast and robust, thus providing a better user experience and significantly faster turn around times.

3.5 Speech Recognition

The accuracy of modern speech recognition systems has improved dramatically, and some systems are now able to recognize a wide range of accents and dialects. The python Speech Recognition module is an open-source python library that can be used to transcribe audio files, such as MP3s or WAV file formats. It is compatible with python versions 2.6,2.7 and 3.3+. It also requires several other libraries including PyAudio, and can be integrated seamlessly with the Google Cloud Speech API. It supports a variety of languages and can be used to transcribe audio from multiple speakers. It is powered by machine learning and can be used to create custom speech recognition models. This enables the library to be customized to the specific needs of an application. The python Speech Recognition module also has a number of features that make it suitable for a variety of applications. These include the ability to transcribe audio from multiple speakers, support for a variety of languages, and the ability to create custom speech recognition models.

4 Design Specification

We will discuss the system design used for the application in the proposed solution. The below mentioned figures gives a description of the components and modules used to create the system. The application primarily consists of seven key modules, namely, user registration module, user login module, OTP generator module, SMTP server to send email to the registered user, monoalphabetic encryption/decryption module, speech recognition module and the application resources/services.

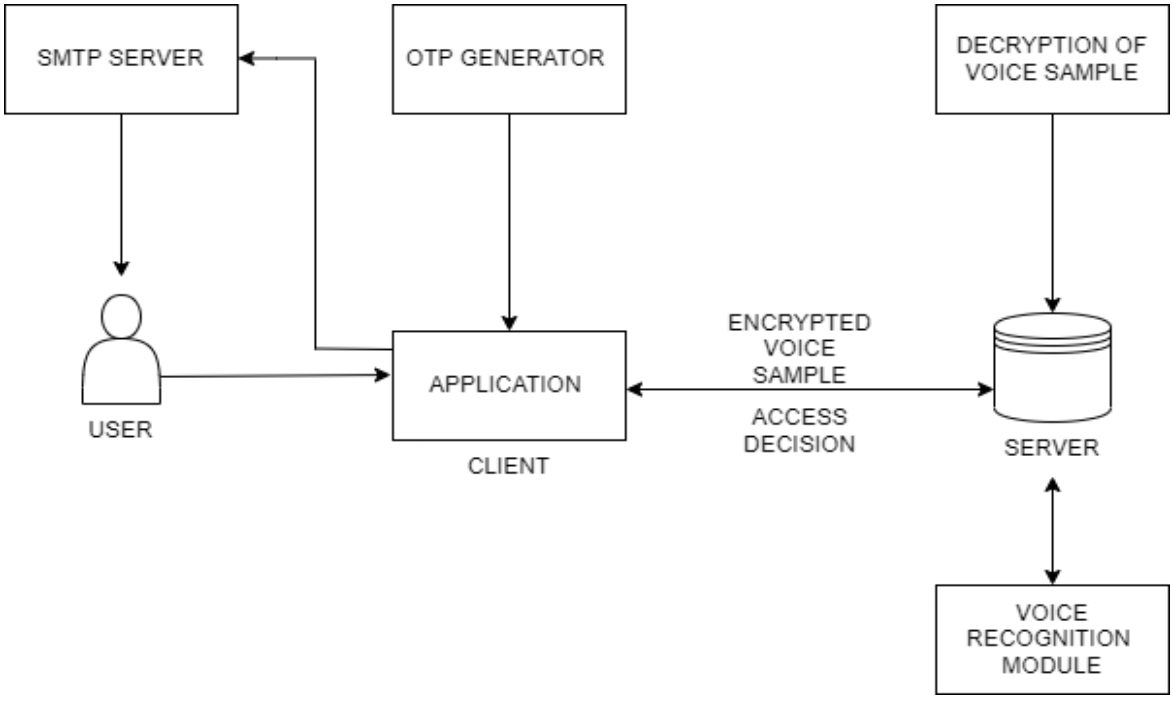

Figure 1 : Proposed Architecture

The description of each module has been elaborated in the subsequent sections. The parameter for OTP generations has been kept configurable and after analysing several combinations, it has been decided to use a numeric OTP of length 6 (ranging from 000000 to 999999), which gives the optimal balance between security and data integrity. The factors taken into consideration has been described in detail in the further sections.

4.1 UML Diagram

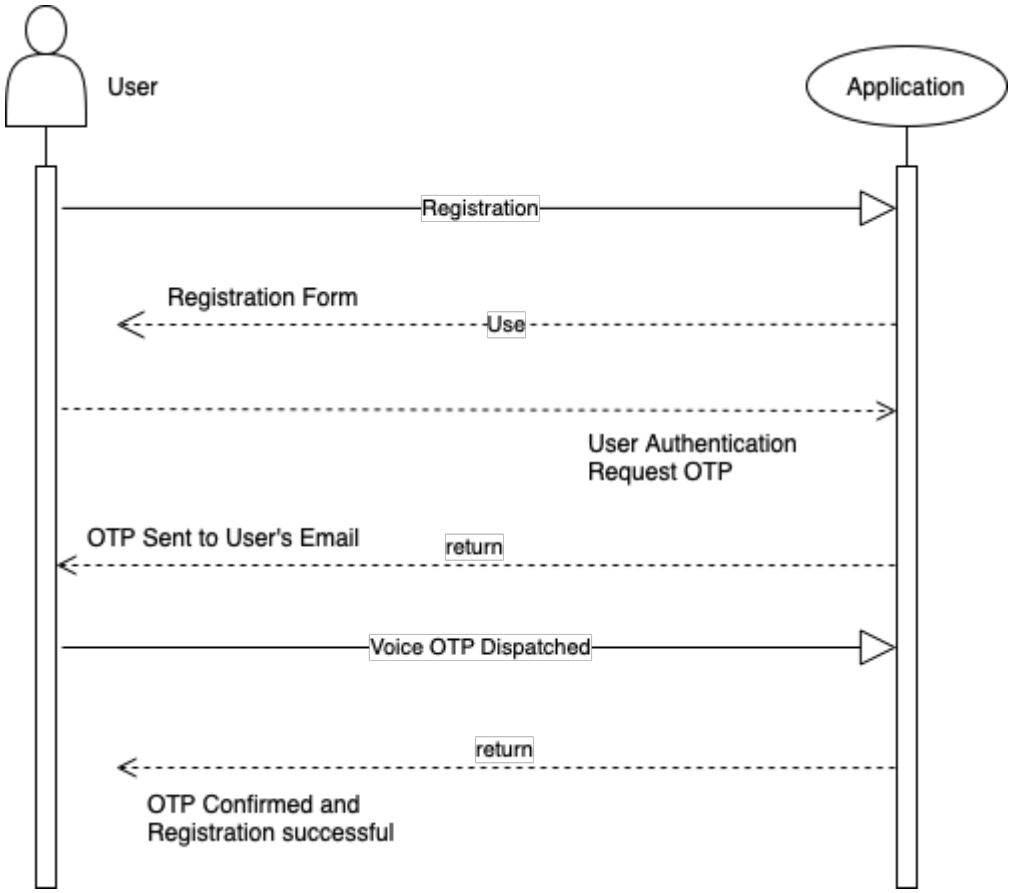

Figure 2 : UML Diagram

4.2 Sequence Diagram

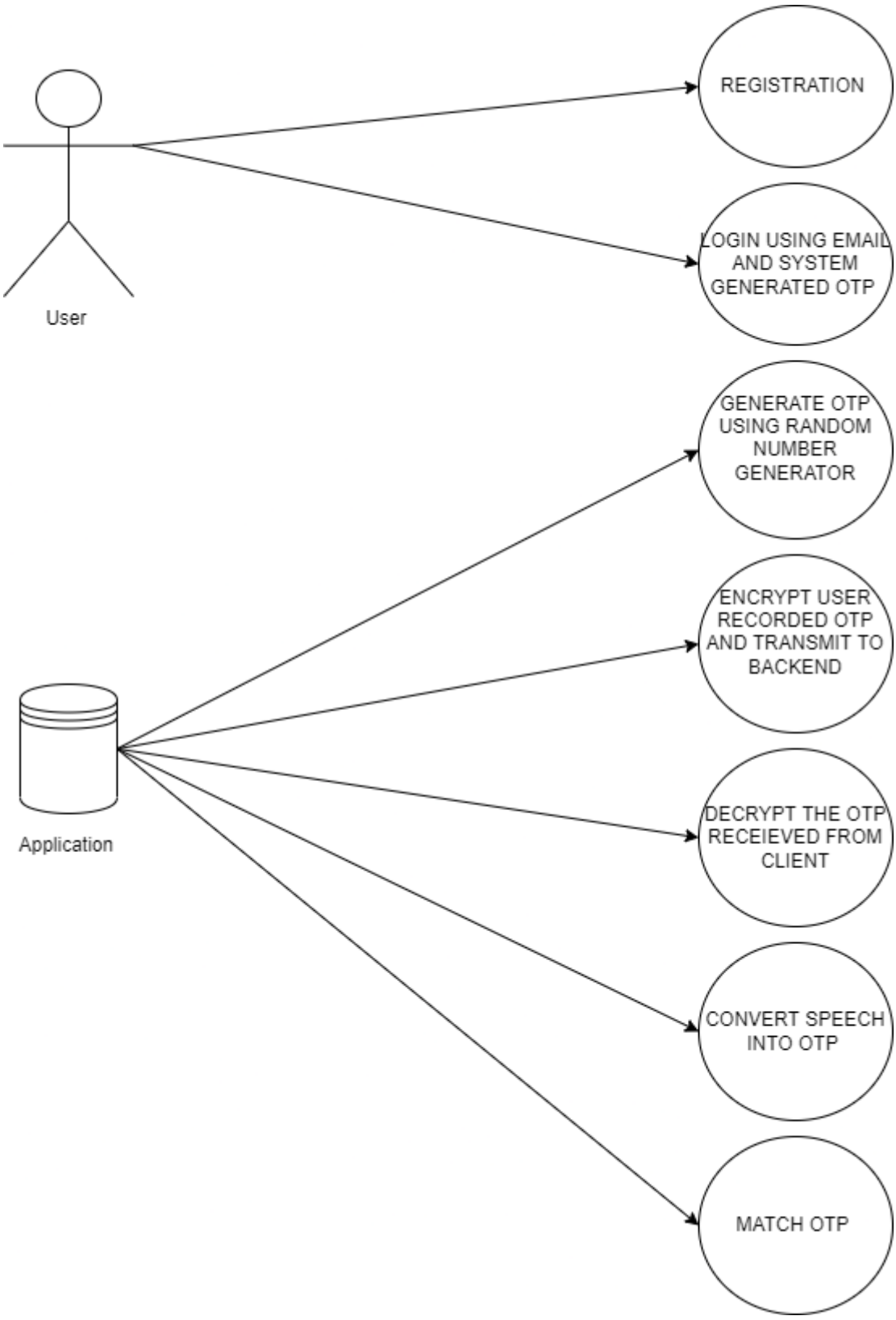

Figure 3 : Sequence Diagram

5 Implementation

This section will give a brief overview of the various components used in the application. Each of the module work sequentially as a pipeline, which implies that the output of one module is taken as an input for the other. The application design can be divided into 3 major modules, as follows:

1. Registration Page
2. Login Page
3. User Verification Page
4. Application Home Page

5.1 Application overview

The first part of the application is the user registration portal. This will be used by the any new user to register for the application. A form has been created that will take inputs from the user such as the user ID to be used, the email ID of the user and the password chosen by the user. Upon clicking the register button, the next step will be the verification of the user. This phase will be used to verify if the email address provided by the user belongs to the mentioned user. For this purpose, a One Time Password will be sent to the user provided email address when he tries to login into the application for the first time. This OTP needs to be spoken out by the user using the microphone of the user device. A user can register twice using the same email address, but the user ID chosen needs to be unique.

Step 1: User Registration

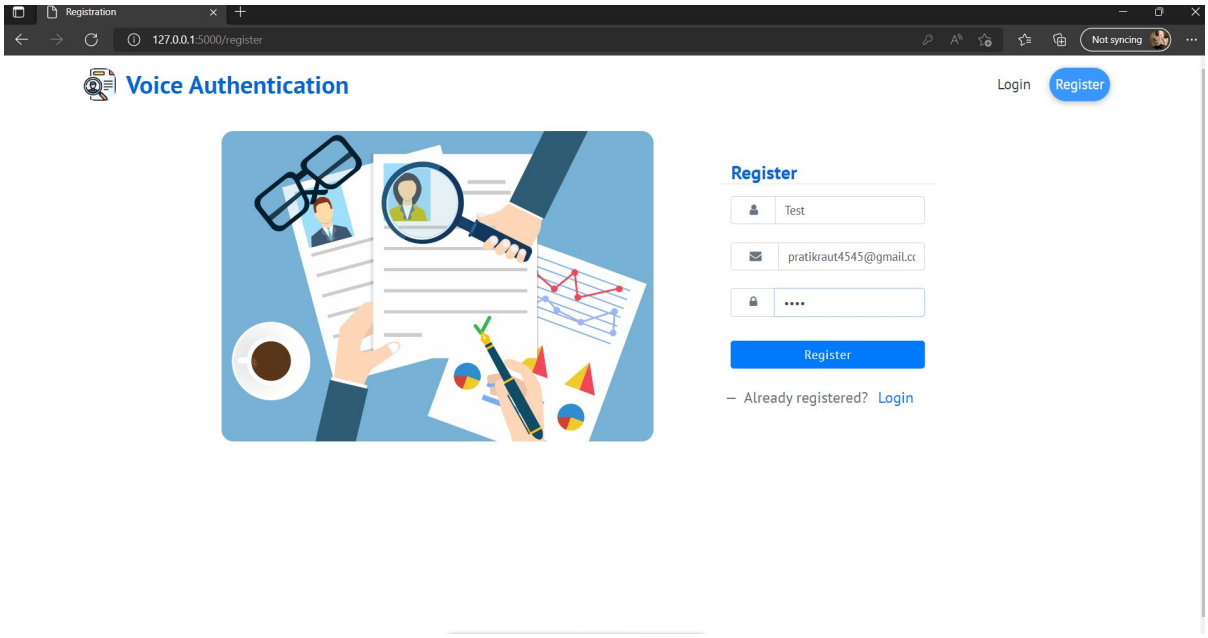

Figure 4 : Registration Page

Step 2: User Login

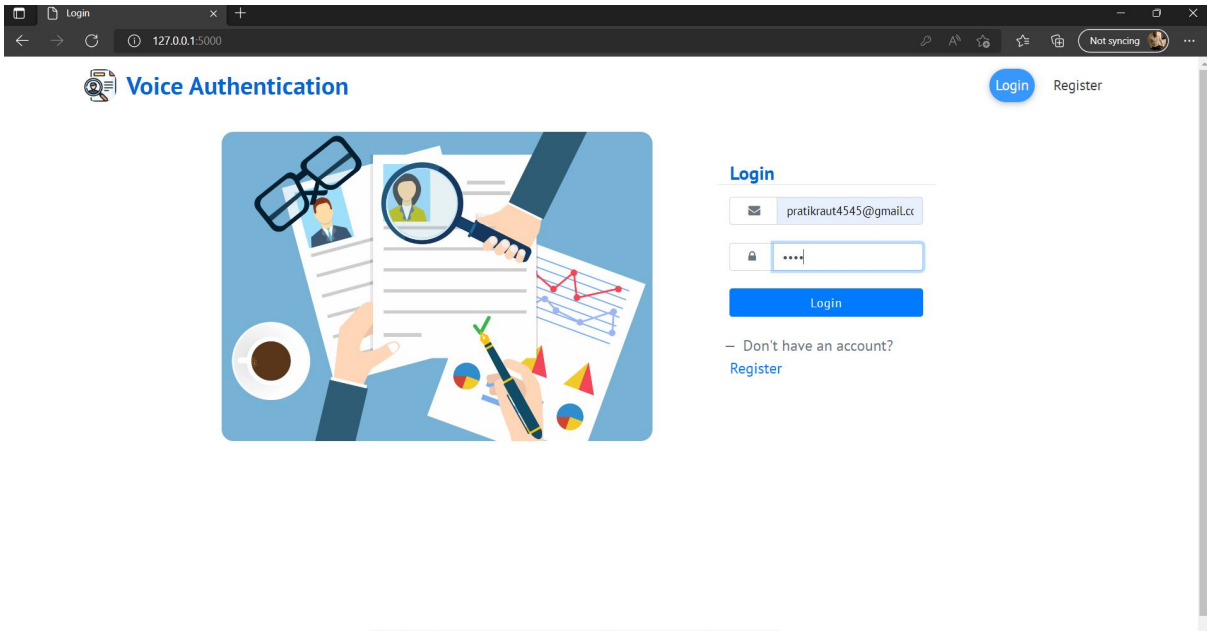

Figure 5: Login Page

Step 3: User receives an OTP on his registered email address

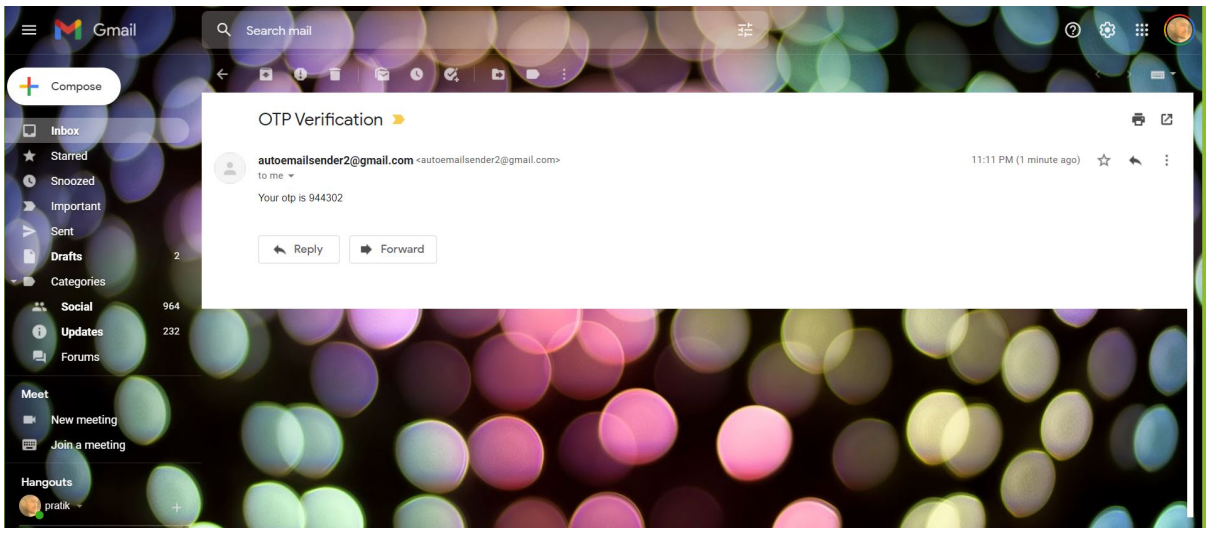

Figure 6: OTP Received on email

Step 4: User is prompted to record the OTP using device microphone.

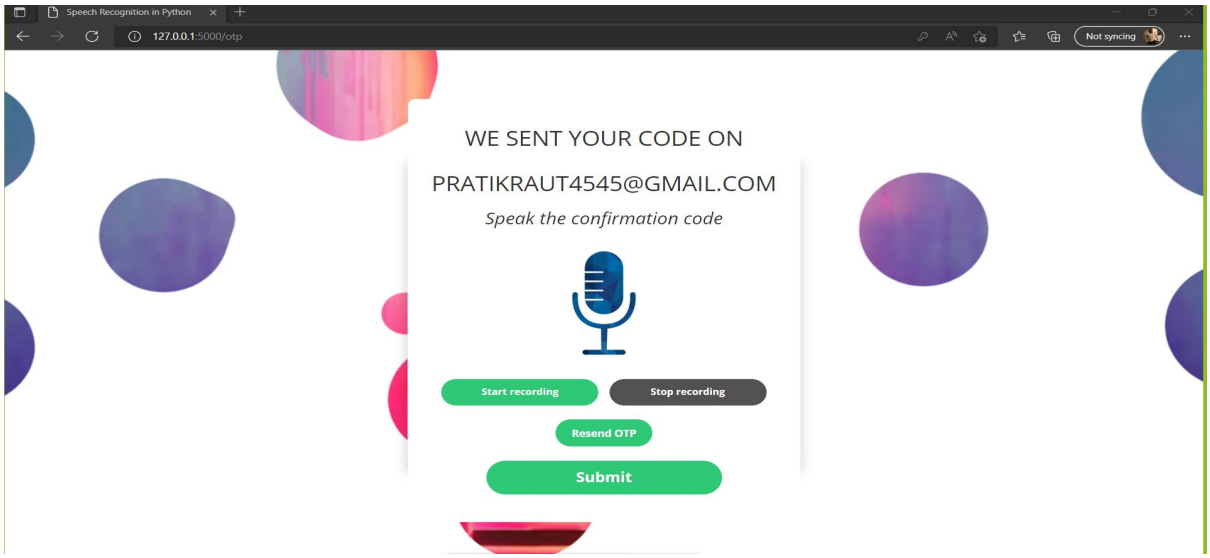

Figure 7: User records OTP

Step 5: OTP is Encrypted on users machine (Client side)

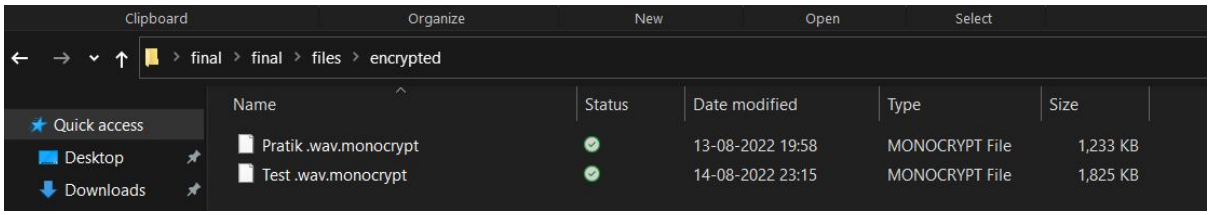

Figure 8: Encrypted Audio File

Step 6: OTP is Decrypted on the server

Figure 9: Decrypted Audio File

Step 7: Server Decrypts the Encrypted audio file and converts speech to text

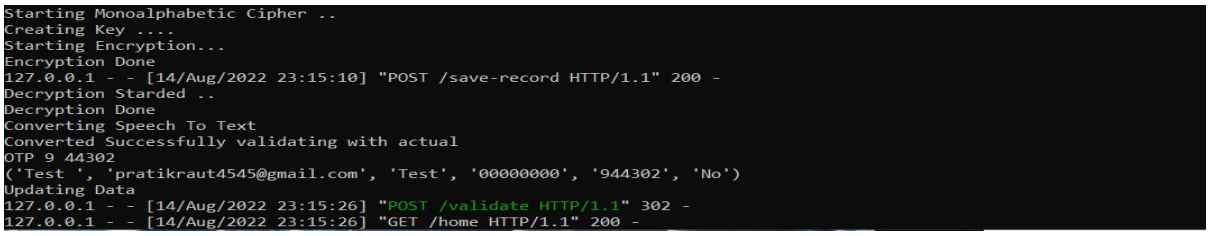

Figure 10: Backend processing

Step 8: User successfully logged in

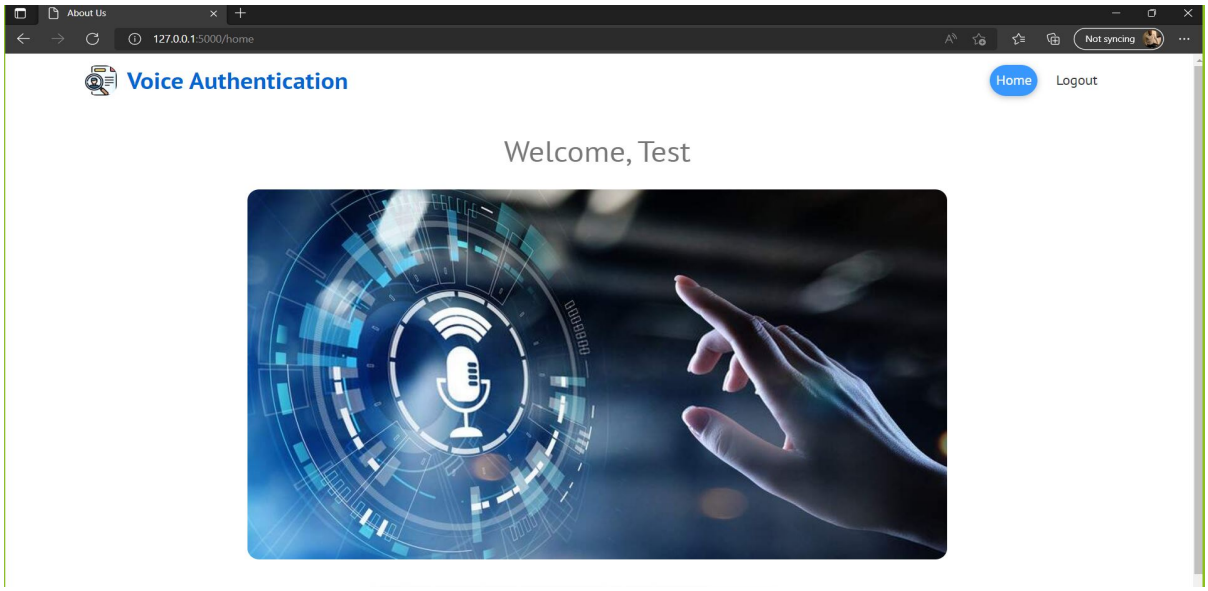

Figure 11: Application Home Page

6 Evaluation

We need to first evaluate the different possible combinations of length and numeric/alpha numeric OTPs to be sent to the user based on the efficiency and accuracy of the application and more specifically the speech recognition module.
Based on the results of this evaluation, we can select the optimal length of characters of the OTP and whether an alpha numeric or numeric OTP should be used.

	No of Characters	OTP	Authentication Status
Numeric	2	OTP 1	Passed
	3	OTP 2	Passed
	4	OTP 3	Passed
	5	OTP 4	Passed
	6	OTP 5	Passed
Alphanumeric	2	OTP 6	Passed
	3	OTP 7	Passed
	4	OTP 8	Failed
	5	OTP 9	Failed
	6	OTP 10	Failed

Table 1: OTP Length Comparison

Based on the above table, we can see that for an OTP of length ranging from 2 to 6, the authentication based on encryption decryption and speech recognition is always performed successfully. In case of alpha numeric OTPs, the authentication is successful only when the length of the OTP is 2 or 3. For OTPs with length more than 4, the speech recognition

module is not able to recognize the characters accurately and hence the authentication fails even though the user has spoken the correct OTP.
Apart from the aspects considered above, we also need to consider the time taken for the entire login process to be completed. If the time taken is more, then it will lead to a longer time for the user to login which in turn compromises the user experience.
Here is a graph that shows the number of digits in the OTP (ranging from 2 to 8) against the time taken for the login process to complete.

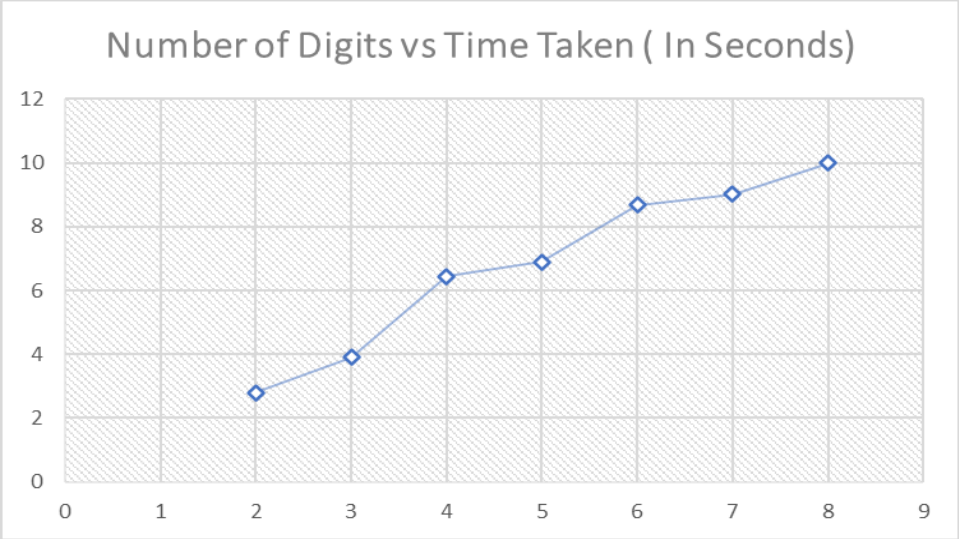

Figure 12: OTP Digit vs Time Taken Chart

6.1 Discussion

Hence, considering the trade-off between the security, accuracy, and the user experience of the system, we can conclude that it is best to use either a numeric keyword of length 6 or an alpha numeric OTP of length 3 will be the optimal choice. I have decided to use a numeric OTP of length 6 in my application, since it is both convenient for the application user and provides a good level of security being accurate at the same time. We also need to consider that even though increasing the number of digits in the numeric OTP further will make the system more secure, it will compromise on the user experience, since the user will have to speak for a longer time.

7 Conclusion and Future Work

The application and user data security depends on the authentication mechanism used. Hence it is of upmost importance that the algorithm and implementation is designed in a manner that will provide the best user experience, without compromising on the data integrity and security. Implementing a speech-based OTP mechanism helps in prevent keylogging attacks. In addition to this, the voice sample is also encrypted before transmitting it to the application backend server over a secured channel. Analysis has also shown that it is vital to use an accurate speech to text conversion API/library, to avoid any loss or corruption of data. Thus, the performance of the system completely depends on the encryption algorithm implemented as well as the speech to text conversion mechanism used. The monoalphabetic substitution

cipher performs well in encryption and decryption and the 6-digit OTP takes a turnaround time of around 8 seconds, which is acceptable considering the additional security it provides. As a scope of further improvement, we can enhance and improve the speech to text conversion algorithms and test out various open-source libraries for the same. This will enable us to use alpha numeric OTPs, making the system more secure. Apart from this, we can also work towards creating a system, that will record the user voice sample based on various test phrases. When speaking out the OTP received, the input voice sample can be matched with the voice sample of the user recorded during the login process and this will provide an additional layer of security. This will help prevent any unauthorised user to access the application even if it maliciously accesses the OTP intended for the authorised user.

Acknowledgment

I am thankful to my research supervisor, Professor Niall Heffernan. He has been a great mentor, providing me with his guidance, feedback and encouragement given at each step of the research work.

References

Addas, A., Thorpe, J. and Salehi-Abari, A. (2019) 'Geographic Hints for Passphrase Authentication', in *2019 17th International Conference on Privacy, Security and Trust (PST)*, pp. 1–9. Available at: https://doi.org/10.1109/PST47121.2019.8949033.

Aloul, F., Zahidi, S. and El-Hajj, W. (2009) 'Two factor authentication using mobile phones', in *2009 IEEE/ACS International Conference on Computer Systems and Applications*, pp. 641–644. Available at: https://doi.org/10.1109/AICCSA.2009.5069395.

Ambika, D. and Radha, V. (2012) *Secure Speech Communication – A Review*. Available at: https://www.semanticscholar.org/paper/Secure-Speech-Communication-%E2%80%93-A-Review-Ambika-Radha/0d8f32a4d1a72dd9f3c1a78642f804fcd443e7a9 (Accessed: 9 April 2022).

Bava, A.G. (2020) *Speech based OTP system to prevent shoulder surfing.* masters. Dublin, National College of Ireland. Available at: http://norma.ncirl.ie/4487/ (Accessed: 9 April 2022).

Biddle, R., Chiasson, S. and Van Oorschot, P.C. (2012) 'Graphical passwords: Learning from the first twelve years', *ACM Computing Surveys*, 44(4), p. 19:1-19:41. Available at: https://doi.org/10.1145/2333112.2333114.

Cha, B., Kim, N. and Kim, J. (2011) 'Prototype Analysis of OTP Key-Generation Based on Mobile Device Using Voice Characteristics', in *2011 International Conference on Information Science and Applications*, pp. 1–5. Available at: https://doi.org/10.1109/ICISA.2011.5772393.

Cheng, F. (2011) 'Security Attack Safe Mobile and Cloud-based One-time Password Tokens Using Rubbing Encryption Algorithm', *Mobile Networks and Applications*, 16(3), pp. 304–336. Available at: https://doi.org/10.1007/s11036-011-0303-9.

Eldefrawy, M.H., Alghathbar, K. and Khan, M.K. (2011) 'OTP-Based Two-Factor Authentication Using Mobile Phones', in *2011 Eighth International Conference on Information Technology: New Generations*, pp. 327–331. Available at: https://doi.org/10.1109/ITNG.2011.64.

Goldburg, B., Sridharan, S. and Dawson, E. (1993) 'Design and cryptanalysis of transform-based analog speech scramblers', *IEEE Journal on Selected Areas in Communications*, 11(5), pp. 735–744. Available at: https://doi.org/10.1109/49.223875.

Groza, B. and Petrica, D. (2005) *ONE TIME PASSWORDS FOR UNCERTAIN NUMBER OF AUTHENTICATIONS*. Available at: https://www.semanticscholar.org/paper/ONE-TIME-PASSWORDS-FOR-UNCERTAIN-NUMBER-OF-Groza-Petrica/c5ea711b450d8358c8edb45a5c17a501484110c7 (Accessed: 8 April 2022).

Lamport, L. (1981) 'Password authentication with insecure communication', *Communications of the ACM*, 24(11), pp. 770–772. Available at: https://doi.org/10.1145/358790.358797.

Pandya, I., Jhajj, S. and Pawar, R. (2017) 'A steganographic approach to mitigate password attacks', in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 248–253. Available at: https://doi.org/10.1109/ICACCI.2017.8125848.

Ping, L. (2021) 'English Speech Recognition Method Based on HMM Technology', in *2021 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pp. 646–649. Available at: https://doi.org/10.1109/ICITBS53129.2021.00164.

Shay, R. *et al.* (2012) 'Correct horse battery staple: exploring the usability of system-assigned passphrases', in *Proceedings of the Eighth Symposium on Usable Privacy and Security*. New York, NY, USA: Association for Computing Machinery (SOUPS '12), pp. 1–20. Available at: https://doi.org/10.1145/2335356.2335366.

Tanvi, P., Sonal, G. and Kumar, S.M. (2011) 'Token Based Authentication Using Mobile Phone', in *2011 International Conference on Communication Systems and Network Technologies*, pp. 85–88. Available at: https://doi.org/10.1109/CSNT.2011.24.

Vatsa, S., Mohan, T. and Vatsa, A. (2012) 'Novel Cipher Technique Using Substitution Method', *International Journal of Information and Network Security (IJINS)*, 1. Available at: https://doi.org/10.11591/ijins.v1i4.598.

'What is MITM (Man in the Middle) Attack | Imperva' (no date) *Learning Center*. Available at: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/ (Accessed: 10 April 2022).

Yan, J. *et al.* (2004) 'Password memorability and security: empirical results', *IEEE Security Privacy*, 2(5), pp. 25–31. Available at: https://doi.org/10.1109/MSP.2004.81.

Yassin, A. adil *et al.* (2013) 'Cloud Authentication Based on Anonymous One-Time Password', *Lecture Notes in Electrical Engineering*, 214, pp. 423–431. Available at: https://doi.org/10.1007/978-94-007-5857-5_46.