OTP BASED SECURE AUTHENTICATION

1MUKEH MEHTA, 2ADITYA A. BELHE, 3PRINCE RAJ SINGH, 4MAQSOOD CHOUDHARY, 5SMITA CHOPADE

1,2,3,4,5Electronics and Telecommunication Engineering Department, FCRIT Vashi, E-mail:1mehtamukesh004@gmail.com, 2abpokemon7@gmail.com, 3prince203310@gmail.com, 4maqsood_choudhary@hotmail.com

Abstract— One of the members in the domain of Automatic Identification and Data Capture (AIDC) technologies is One Time Password (OTP). It is a fast and reliable means of authenticating or verifying an authorized person. Security is considered as one of the most important areas of concern in most of the organizations. In this paper, we are proposing an authentication service that makes use of one time passwords to provide security at the organizational level. One-time password (OTP) is always used as one of stronger authentication scheme among all password-based solutions. To reduce the damage of phishing and spyware attacks, banks, governments, and other security-sensitive industries are deploying one-time password systems which is one of the emerging technologies in the field of security. The innovation in one time password contributes to develop a more secured login procedure to access the web-based security system.

Keywords— OTP Generating Module, Python, Authentication, Security.

I. INTRODUCTION

Identification of persons is essential at places like government and corporate organizations, airports, railway stations and banks. Auto-identification means automatic identification of entities. We have various methods for auto-identification. Some of them are barcode systems, optical character recognition, biometrics, smart cards and RFIDs. Nowadays OTP technology is widely used for online banking, online shopping, verifying a person while making mail accounts, etc.

A one-time password (OTP) is a password that is valid for only one login session or transaction, on a computer system or other digital device. OTPs avoid a number of shortcomings that are associated with traditional (static) password based authentication; a number of implementations also incorporate two factor authentication by ensuring that the one-time password requires access to something a person has (such as a small key ring fob device with the OTP calculator built into it, or a smartcard or specific cell phone) as well as something a person knows (such as a PIN).

A major advantage is that a user who uses the same (or similar) password for multiple systems, is not made vulnerable on all of them, if the password for one of these is gained by attacker. A number of OTP systems also aim to ensure that a session cannot easily be intercepted or impersonated without knowledge of unpredictable data created during the previous session, thus reducing the attack surface further.

OTPs have been discussed as a possible replacement for, as well as enhancer to, traditional passwords. On the downside, OTPs are difficult for human beings to memorize. Therefore, they require additional technology to work.[1]

Identity theft is one of the fastest growing types of crime nowadays, since it is more profitable than other types of crime. The aim is to authenticate the person(Authorized) who is trying to access a particular service which includes Internet Banking, Online Shopping,and Creating Mail ID’s etc. The most important advantage that is addressed by OTPs is that, in contrast to static passwords, they are 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. [1]

II. PROBLEM STATEMENT

III. PROPOSED BLOCK DIAGRAM

4.1. Python Scripting Language

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple paradigms, including oriented, imperative and functional programming or procedural styles. It features a dynamic system and automatic memory management.
and has a large and comprehensive standard library.[2]

4.2. TOTP (Timer based OTP) Using Python

Basically we define TOTP as $\text{TOTP} = \text{HOTP}(K, T)$ where $T$ is an integer and represents the number of time steps between the initial counter time $T_0$ and the current Unix time. More specifically, $T = (\text{Current Unix time} - T_0)/X$, where the default floor function is used in the computation. The implementation of this algorithm must support a time value $T$ larger than a 32-bit integer when it is beyond the year 2038. The value of the system parameters $X$ and $T_0$ are pre-established during the provisioning process and communicated between a prover and verifier as part of the provisioning step. TOTP is a variant of HOTP, which specifies the calculation of a one-time password value, based on a representation of the counter as a time factor. [3]

![Figure 2: TOTP using Python](image)

V. WORKING OF THE SYSTEM

As shown in figure 1, level 2 user has to enter his/her valid email Id (which is stored in central database) and then press submit button. After this the user will get receive an email from CES containing one OTP (One Time Password) generated by the pythonas described in section 4.2. The user then has to enter the received OTP within the predefined time (60 seconds in our case). If the user enters the valid OTP after this predefined time then the server will display a web message as shown in figure 7. If the user enters the valid OTP during the predefined time then it is checked whether the entered OTP is correct or not. If it is correct, user will get access to level 2 documents. Also if any user try to access the document by entering invalid email id (which is stored in central server database) then CES will send an email to CAS informing about the unauthorized access. TheCAS authority will then take the necessary actions accordingly.

VI. RESULTS AND DISCUSSION

Figures show the screenshot of final implementation of our project.

![Figure 3: Link to Level 1 documents and gateway for accessing Level 2 documents](image)

Figure 3 shows the gateway for level 2 documents. User has to enter valid ID as shown in figure and submit it.

When the user clicks on the submit button the central email server sends the email containing OTP to the authorized user as shown in Figure 4.

![Figure 4: E-mail to authorized user containing OTP](image)

As shown in Figure 5, user has to enter the received OTP in the password block within stipulated time (60 seconds in our case) as we have used time based OTP.

![Figure 5: UI for entering OTP](image)

Figure 6 shows the web page which is displayed when the user enters valid OTP. This page also contains the Level 2 documents.

![Figure 6: Link to Level 2 documents if valid OTP is entered](image)

Figure 7: Message for invalid OTP

Web message as shown in Figure 7 is displayed when the entered OTP is invalid or when the OTP is not entered within specified time. In any case user will have to repeat the procedure again.

![Figure 7: Message for invalid OTP](image)

Figure 8: E-mail to Central Authentication Server
When the user with user ID (that is not stored in our database) tries to access level 2 documents, the Central E-Mail Server sends an email as shown in Figure 8 to the Central Authentication Server indicating unauthorized access.

We have also secured our database created in excel by a password as it is seen in Figure 9, which provides additional security to the system. Figure 10 shows the small database containing the email ID of our group members.

### CONCLUSION

OTP based secure authentication has been implemented and the results have been incorporated in this paper. It is concluded that OTP as a stand-alone security system provides a basic level of security. To enhance the level of security provided, OTP can be integrated with other stand-alone security systems like RFID and biometric authentication. The backend databases can also be implemented using software languages like SQL, JavaScript for further enhancement of security.

### REFERENCES

2. [https://pythonhosted.org/otpauth/](https://pythonhosted.org/otpauth/)