Abstract
This paper presents a secure presence verification system for service providers at medical institutions. The goal of the system is to securely verify that the patients were visited by the service providers at the agreed date and time by utilizing TOTP based OTP passwords shown in a format of QR codes and a special mobile application allowing to scan the QR codes and store the history of recognized OTP codes.

Reference

Available at:
http://archive-ouverte.unige.ch/unige:120318

Disclaimer: layout of this document may differ from the published version.
Physical presence verification using TOTP and QR codes

Emin Huseynov[1(0000-0003-4198-5400)] and Jean-Marc Seigneur[2(0000-0002-3271-2905)]

1 University of Geneva, Geneva, Switzerland
emin@huseynov.com
2 University of Geneva, Geneva, Switzerland
Jean-Marc.Seigneur@unige.ch

Abstract. This paper presents a secure presence verification system for service providers at medical institutions. The goal of the system is to securely verify that the patients were visited by the service providers at the agreed date and time by utilizing TOTP based OTP passwords shown in a format of QR codes and a special mobile application allowing to scan the QR codes and store the history of recognized OTP codes.

Keywords: one-time passwords, presence verification, QR codes, access systems, totp

1. Introduction

Presence verification features are highly demanded in a number of industries, and one of the most demanded features for medical institutions for cases where different parties are contracted to periodically visit patients to provide services. This applies not only to external service providers but also to providers within the same institutions. Often, hospital systems provide services within an enterprise, demonstrating the level of interoperability abstraction within the same organization [1].

Usual methods of presence verification are consisting of a simple paper-based schedule table where the service providers just put the current date and time and their signature [2]. Being simple enough and easy to implement, this method, however, does not guarantee the accuracy, as the signatures on the paper are easy to back-date and/or forward date. The method introduced in this work will allow to securely implement presence verification using a special mobile application and a static hardware token displaying two-dimensional barcodes.
2. Concept

The system proposed in this paper is based on three main components: a stationary hardware token, a mobile app and a verification server. The stationary hardware token is an isolated device that is to be installed next to the patient’s bed and can be powered using a regular electrical network. The mobile app is to be installed on the smartphones carried by the service providers’ employees and will be used to record their presence using the hardware tokens showing the current OTP in a QR code format. The verification process is based on Time-based One-time Password Algorithm (TOTP) algorithm [3].

3. Theory and Implementation

The proof of concept implementation is quite simple: we are planning to use an Android tablet as the stationary device and an Android phone for the verification client to run. For production use, the stationary device needs to be hardened both physically and at the operating system level to prevent the modification, especially system time adjustment, which may allow replay attacks.

3.1. Theory

The idea of using one-time passwords was formulated in November 1981 by Lamport [4]. In principle, OTP can be generated in different ways, but the majority of existing classic as well as modern MFA systems rely on modern RFC-based TOTP standard, in this section we will give a brief overview of the theoretical background of OTP generation methods this standard uses.

Bellare, Canetti and Krawczyk presented a construction called HMAC (for Hash-based Message Authentication Code) [5], which keyed message authentication codes from public unkeyed cryptographic hash functions, by simulating the keying of the hash function’s initialization vector. HMAC function is defined as shown in equation 1, where $\oplus$ denotes the bitwise exclusive-or operation, $k$ denotes the concatenation operation, and opad and ipad are constants for outer padding and inner padding respectively.

\[
\text{HMAC}(K, m) = H((K \oplus \text{opad})kH((K \oplus \text{ipad})km))
\] (1)

The Time-based One Time Password Algorithm (TOTP) was proposed by M’Raihi et. al. in [3], and presents a type of key-derivation function that takes a single secret (such as a password or passphrase) as an input and produces a single key as an output. TOTP is a variation of HOTP algorithm, however, with TOTP, the key is dependent on the secret as well as on the current time, instead of arbitrary counter value defined in HOTP; so current epoch time is considered a counter instead and is calculated as shown in equation 2, where $T0$ is a predetermined epoch relative to which all times are
counted and $T_s$ is a time step. Time steps used in the majority of TOTP implementations are 30 or 60 seconds.

$$ T_c = \frac{\text{Current Time} - T_0}{T_s} $$

(2)

Once the value of the time counter is calculated, the TOTP output can be calculated by signing the master secret ($K$) and the time counter ($T_c$) using HMAC function as shown in Equation 3.

$$ TOTP(K) = HMAC(K, T_c) $$

(3)

The master secret ($K$) used to calculate the HMAC hash is also called a shared secret. The reason for that is because this key is shared between the authentication server and the client generating OTP codes. The secret is shared during the registration. So, the values used to generate TOTP are exactly the same at both parties (client and the server): the secret is the same and the time counter used is based on system clocks, ideally the same or within the 30/60 seconds offset. So, the validation process of user-submitted OTP codes is as simple as comparing with server generated OTP codes.

The OTP code will be shown as QR codes. This is not the primary area of this research and will not be reviewed in detail; presenting OTP as QR images is only to simplify the process of entering the value of the OTP to the mobile app. For the rare cases of mobile devices not having a camera, the OTP can be entered manually.

3.2. Implementation

In our implementation, in addition to the components having the shared secrets stored (the validation server and the QR generator) there will be a third component used to read the current OTP as a QR and storing its value together with the current timestamp. The combination of the OTP and the current timestamp will be the basic principle of the presence verification. The action flow is illustrated on Fig 1.

---

**Fig. 1.** Action flow of physical presence verification
The visits will be logged on the employees’ mobile device and can be uploaded to the verification server to be verified using the TOTP algorithm. The process will look like shown on Fig 2.

![Image](image.jpg)

**Fig. 2.** The process of scanning the QR code to log the presence

### 4. Methodological framework

In this section we will describe a framework and technology for creating the components for creating a proof of concept for the proposed presence verification method.

The stationary part, that can also be called TOTP token will consist of a computational component, such as Raspberry PI or Arduino [6] (using common programming frameworks based on Python, PHP or C++), with internet access (for time synchronization) or real-time clock as TOTP algorithm relies on accurate time to calculate one-time passwords. The device will also have a display device, possibly e-Ink based – to minimize power consumption, which will display a QR code to be scanned by the client app. The QR code will be encoding a string, $T_s + OTP$, composed of the current timestamp (seconds since 1 January 1970) and the corresponding OTP value. An example of such string is “1555618251-677690”, where 1555618251 is the timestamp as of 18-04-2019, 10:12PM CET, and 677690 is the OTP value calculated based on the shared seed stored on this device and the current timestamp. The process flow is shown on Fig 3.

![Diagram](diagram.png)

**Fig 3.** TOTP based QR generation process flow diagram
The client app, in its turn, will be as simple as a QR decoder, which will only scan the QR code shown by the stationary TOTP token, decode the T\textsubscript{s}+OTP value and store it locally. An additional subsystem allowing to upload scanned T\textsubscript{s}+OTP value history to a central location can also be implemented (an additional user identification and authorization mechanism needs to be put in place). Such an app can easily be created under different platforms using native technologies, or even hybrid cross-platform frameworks, such as Cordova or Ionic [7] etc.

And the third, final component of the system is a server or software component which will have access to the same shared seed as the TOTP token and will be able to verify whether the T\textsubscript{s}+OTP pairs are correct. As per RFC 6238 [3] recommendations, the verification process may include a certain number ofOTP values calculated with timestamp steps before and after the submitted timestamp to accommodate time drift of the devices.

5. **Conclusion and future work**

The idea presented in this paper is indeed a work in progress and is yet to be finalized at least as a proof of concept stage. However, it is already possible to state that this would be a significant improvement in the area of keeping records of a physical presence of a person in a particular location compared to methods currently in use, mainly paper based. The concept can be used to be implemented in areas like hospital services or other similar domains. For example, this can be used to quickly and securely log visits of a doctor or cleaning services employees in a patient’s room.

The future work should include the development of the stationary TOTP-QR display device, the presence reader app for popular platforms as well as OTP verification mechanisms.

6. **References**