Improving user experience with TOTP hardware tokens by implementing QR codes and hid keyboard emulation

Huseynov, Emin

How to cite

HUSEYNOV, Emin. Improving user experience with TOTP hardware tokens by implementing QR codes and hid keyboard emulation. In: 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT) Conference Proceedings. Tashkent, Uzbekistan. Tashkent, Uzbekistan : AICT, 2020. doi: 10.1109/AICT50176.2020.9368574

This publication URL: https://archive-ouverte.unige.ch/unige:145766
Publication DOI: 10.1109/AICT50176.2020.9368574

Improving user experience with TOTP hardware tokens by implementing QR codes and HID keyboard emulation

Emin Huseynov
Faculté des Sciences de la Société, Université de Genève
Geneva, Switzerland
emin@huseynov.com

***Abstract*— The paper presents a concept of a device that will introduce an effort to minimize user interaction by transferring the one-time password codes from hardware devices to the main systems via alternative methods to avoid manual entry by implementing keyboard emulation and quick response code generation**

Keywords— digital identification, identity management, authentication, authorization, privacy, context-based authentication, strong security, one-time password, TOTP, user experience, QR codes, HID interface

I. Introduction

User experience is one of the most important aspects of multi-factor authentication (MFA) systems. The main and obvious reason for this is the fact that most of the users prefer to avoid adding additional complexity to their login activity even by sacrificing security [1]. One of the aspects of negative user experience is reported when using TOTP (Time-based one-time password protocol) hardware tokens where users must enter OTP (one-time password) generated by the tokens into the login form, which is perceived as complex and error prone as the digits must be typed in manually.

In this paper, I will propose a concept of a device that will introduce an effort to minimize user interaction by transferring the OTP code from hardware devices to the main systems via alternative methods to avoid manual entry. The device will use two methods to achieve this: HID (human interface device) keyboard emulation via USB (universal serial bus) which will allow sending the digits directly to the main system; and QR code (Quick Response code) encoding that can be scanned using the native camera application of Apple's devices running iOS operating system, which will allow scanning the QR code, and copying and pasting the decoded data instead of entering the code manually.

II. Evaluation of existing solutions

In this chapter we will focus on existing popular hardware solutions.

This paper is sponsored by *TOKEN2 Multifactor authentication products and services*, a Swiss company focusing on strong authentication solutions. TOKEN2's team used to be a part of a multifactor authentication research project at the University of Geneva, which has led to a spin-off startup company back in 2013.

A. Classic hardware tokens

A classic hardware token is a standalone physical device with no external interface nor connectivity that is used to generate one-time password codes used when a user is authenticating themselves during a logon process.

1) User experience analysis

Classic TOTP hardware tokens propose no alternative methods of transferring the OTP – by design, the OTP must be typed in manually. This is one of the main reasons of user resistance during MFA. Token2 C202 [2] is an example of such hardware tokens – it has no interface for transferring the information other than an LCD display showing the generated code containing 6 digits as illustrated on Figure 1.

Figure 1. Token2 c202 TOTP hardware token

2) Security analysis

If focusing on the device security itself without taking into account other components (such as man-in-the-middle attacks or MFA bypass), the security level of standalone hardware tokens is quite high. The only risk that can be thought of is the risk of the generated OTP to be looked up by an attacker. This risk is quite low as the attack window is narrow (only 30 seconds for the example of C202 token), also compromization of the first factor (username and password) and physical access to the token are required for an attack to be successful.

B. FIDO USB keys with companion apps

FIDO (Fast ID Online) [3] is a set of platform-independent security specifications for strong authentication. Several manufacturers have integrated TOTP functionality into their FIDO keys. Since these keys do not have a built-in battery, the

TOTP mechanism they offer is based on time counter value sent over from the host machine. This causes a few disadvantages both from user experience and, more importantly, security aspects. Although there are multiple hardware devices using the same approach, we will review one of the most popular devices as an example, Yubikeys from Yubico [4].

Figure 2. Yubico YubiKey 5 NFC security key

1) User experience analysis

Yubico Authenticator is an companion app available under multiple platforms that allows generating OTP based on the credentials stored on the YubiKeys (seeds). The principle behind it is illustrated on Figure 3. The application is sending the current timestamp over USB or NFC to the YubiKey connected to the host machine. The TOTP algorithm implemented inside the YubiKey generates the OTP and send it back via the same channel. The app shows the OTP on the app interface allowing to copy it to the clipboard if needed. Using the app in combination with the hardware keys also guarantees that the sensitive data (seeds) never leave the hardware keys and only the final authentication data (OTP) is transferred to the host machine.

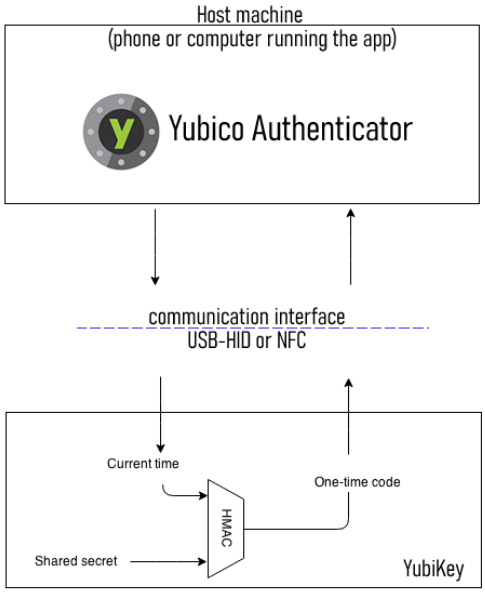

Figure 3. Yubico Authenticator application

While using TOTP with YubiKey has its advantages, such as storing the seeds in a more secure manner compared to regular mobile applications and the hardware lasting longer as it has no battery nor any other active components, the user experience suffers from having to have a companion app installed and the relying on USB or NFC communication capabilities of the host device.

2) Security analysis

With the current mechanism, there is a potential security risk of the TOTP code replay attack. If the attacker has access to the USB key, even for a few minutes only. The attackers can set the time of the host system in the future and write down the OTP code the authenticator app generates. This process can be repeated a significant number of times, so the attacker would have, let's say, 100 OTP codes that the victim's key will display at certain times in the near (or far) future. This already means that the second factor is compromised and is equal to attackers having access to the seed. Once this has been achieved, the attackers only need to get access to the first factor, e.g. the password, which is much easier to implement, and later, when the time of validity of the OTP codes arrives, both factors will become compromised.

C. Gap analysis

Our analysis shows that both reviewed solutions have shortcomings both in user experience and security aspects. The solution presented in the next chapter is meant to close these gaps and minimize the risks and negative user experience as much as possible.

III. Proposed solution – EVVIS-QR device

A hardware token showing the generated OTP value both as digits and as a QR code was already proposed by one of the previous academic papers [5] and was intended to provide an error-free method for TOTP based electronic visit verification information systems (hence the name – EVVIS). In the current paper the solution will be enhanced further to implement sending the OTP via USB-HID interface. Also, as a part of user experience improvement a particular use case of the modern Apple iOS based devices will be reviewed.

A. Operating principle of the EVVIS hardware token

To produce a proof of concept and subsequently a commercial product, I will be using various software development tools for embedded systems.

Due to the nature of the product, to protect the commercial interests of the sponsoring organization, as well as the format and the volume of the source code, the principle will be presented in a flowchart format.

The algorithm generating the one-time password is described in RFC 6238 [6]. In this paper, only the modifications and improvements of the standard OTP scheme will be presented. As shown on Figure 4, the shared secret stored on the device together with the current timestamp produced by the real-time clock component are used as arguments for calculating the TOTP as per RFC 6238. The OTP is shown on the display

similar to standard hardware tokens. As an addition, the device generates a QR code image based on the OTP value and shows together with digits-only OTP.

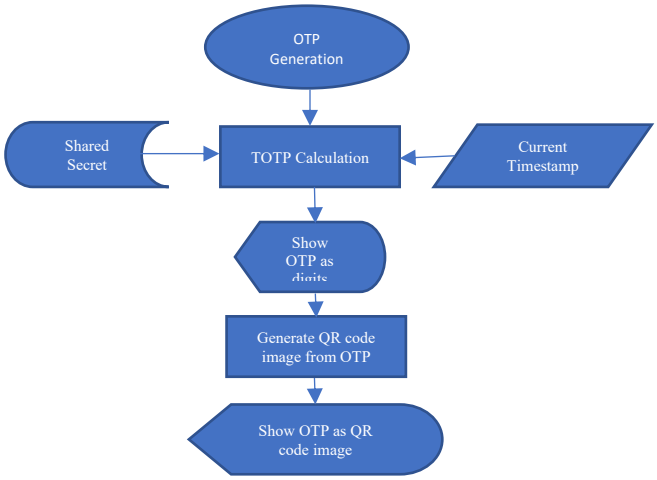

Figure 4. QR code based on TOTP algorithm

The OTP is shown on the display like standard hardware tokens. As an addition, the device generates a QR code image based on the OTP value and shows together with digits-only OTP as illustrated on Figure 5.

Figure 5. EVVIS device displaying OTP and QR code

B. Improving user experience on iOS devices with EVVIS

The devices showing the OTP as QR code have the most potential of greatly improving user experience when a special app is used. However, more frequently, users need to log in using standard applications, such as the native web browser, where we should rely on standard features of the operating system. In this section, we will leverage the native features of Apple's mobile operating system and present this as an example of how user experience can be improved with the proposed solution compared to classic hardware tokens.

Starting from version 12 [7], iOS has the ability of decoding QR images built-in, without the need of an additional software. The QR code scanner tool is accessible directly from the control center and can be launched from within any app by swiping up from the bottom of the screen as illustrated on Figure 6.

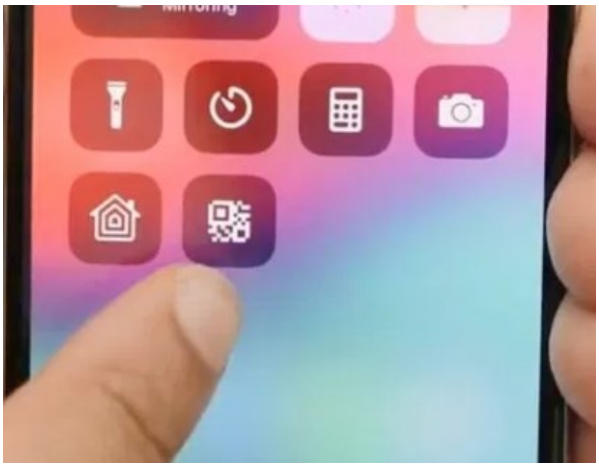

Figure 6. Control center access on iOS12

This feature can be used to simplify the process of logging in to systems with TOTP-based two-factor authentication with the EVVIS device used as its second factor by avoiding typing in the OTP manually. For this illustration, we will be using an iPhone 8 running iOS 13.2 and an example of using native browser app (Safari) to login to a web page; the user action workflow is as follows:

1. On the login screen, where OTP is needed to be entered, user swipes from the bottom of the screen upwards and activates the Control Center
2. User scans the QR code shown on the EVVIS device. As the data encoded in the QR image contains only digits, the Control Center will immediately prompt to Copy the OTP to the clipboard
3. Switch back to the browser window (using app switching) and invoke the context menu on the OTP input field and paste

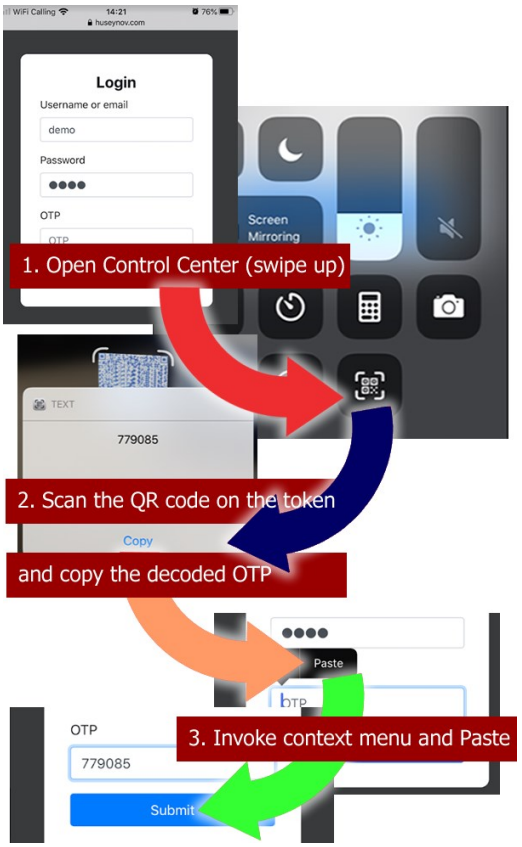

Figure 7. Using QR code to enter OTP value

The process, illustrated on Figure 7, is not only improving the speed of the process (even if slightly), but, more importantly, helps avoiding input errors when the OTP is typed in manually. The speed of the process (yet to be evaluated and compared to manual input speed) can be further improved by leveraging the Shortcuts app functionality of modern iOS (v 12.0 and higher) [8]. A shortcut within this app is a quick way to get one or more tasks done with your apps. By creating a set of tasks using Shortcuts app we can minimize the number of actions required to be done by user to copy the OTP from the EVVIS device.
The Figure 8 shows an example of tasks created with Shortcuts that merges 3 different user actions into one, namely:

1. Launching the QR reader
2. Getting the text encoded in the QR code
3. Copying the recognized text to clipboard

will be replaced by one action: launching the Shortcut task only

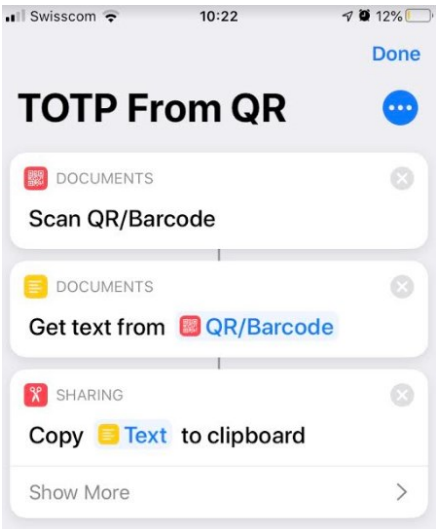

Figure 8. Minimizing user actions using Shortcuts

C. Sending the OTP via USB HID interface

The idea of devices utilizing the USB interface and emulating Human Interface Devices (HID), namely USB keyboard has already been reviewed in a number of papers [9] [10]. With such solutions the OTP generated can be sent as keystrokes via HID interface emulating pressing the digit keys.
The user actions in this case, assuming the token is permanently plugged into the USB port of the workstation, during the authentication process, are consisting of the following three steps:

1. Clicking on the text field of the secondary login step (i.e. the OTP field)
2. Pressing the button to trigger the HID emulation process
3. Completing the login process (by hitting Enter or clicking on Submit)

In many login forms, when prompting to enter the OTP, the cursor is automatically set to the required text field, so there will be no need for the first step. Additionally, if the HID keystroke sequence will include an additional key code of the Enter key, the only action required by users will be pressing the button on the token device.

D. Production phase and commercialization

A market-ready product based on the described functionality was already launched by the sponsor of this paper. A “burner” application allowing to set the seed and TOTP algorithm parameters was also developed as shown on Figure 9.

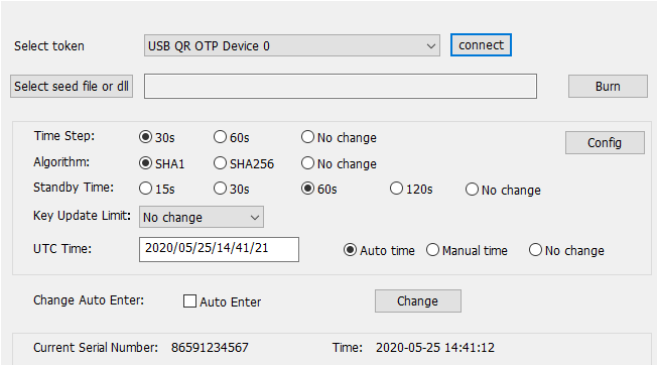

Figure 9. USB Burner app for Windows

IV. CONCLUSION AND FUTURE WORK

A. Conclusion

The paper described a device combining a classic TOTP hardware device with two additional features that could potentially and significantly improve the user experience by replacing the process of manually typing the digits generated by pressing a button or scanning a QR code. While the main target area of the product is still the electronic visit verification, the device can still be used as a more user-friendly TOTP hardware token without accessory applications needed. The user experience measurement was done by comparing the required number of steps/actions.

B. Future work

As a continuation of the same research domain a more tangible user experience measurement activity is required, possibly involving researchers specializing in the relevant areas.

V. REFERENCES

[1] Thuy O., "Over 90 percent of Gmail users still don't use two-factor authentication," The Verge, 23 1 2018. [Online]. Available: https://www.theverge.com/2018/1/23/16922500/gmail-users-two-factor-authentication-google. [Accessed 14 4 2020].

[2] Token2 Switzerland, "Token2 C202 TOTP hardware token," Token2.com, 01 01 2020. [Online]. Available: https://www.token2.com/shop/product/token2-c200-hardware-token. [Accessed 14 4 2020].

[3] FIDO Alliance, "What is FIDO?," FIDO, 01 01 2020. [Online]. Available: https://fidoalliance.org/what-is-fido/. [Accessed 14 4 2020].

[4] YubiCo, "The YubiKey," YubiCo, 1 1 2020. [Online]. Available: https://www.yubico.com/products/. [Accessed 16 4 2020].

[5] Huseynov E., Seigneur J-M., "Physical presence verification using TOTP and QR codes," in *34th International Conference on ICT Systems Security and Privacy Protection - IFIP SEC 2019*, Lisbon, Portugal, 2019.

[6] M'Raihi D. et al., *TOTP: Time-Based One-Time Password Algorithm,* Internet Engineering Task Force (IETF), 2011.

[7] Apple Inc., "About iOS 12 Updates," Apple, 05 05 2018. [Online]. Available: https://support.apple.com/en-us/HT209084. [Accessed 25 03 2020].

[8] Apple Inc., "Shortcuts app release notes," Apple, 19 04 2019. [Online]. Available: https://support.apple.com/en-us/HT209087. [Accessed 25 05 2020].

[9] Engelhardtsen F. B., "Leostick_otp - DIY Google Authenticator OTP USB token," 01 01 2013. [Online]. Available: https://github.com/fritjof/leostick_otp. [Accessed 25 05 2020].

[10] Huseynov E., Context-aware multifactor authentication for the augmented human, Doctoral Thesis, Geneva: University of Geneva, 2020.