Chapter 50

Context-Aware Multifactor Authentication Survey

Emin Huseynov and Jean-Marc Seigneur
University of Geneva, Carouge, Switzerland

1. INTRODUCTION

Multifactor authentication was described in 1988 in US Patent 4,720,860, where the second factor was presented as "nonpredictable code" [1]. The first implementations of such systems were based on isolated hardware devices generating such codes. In this type of system, that device can be regarded as a context for additional authentication factors. Thus, the idea of using context to replace or deliver additional authentication factors is not new and was used in a large number of projects. We will review a few of them to illustrate the principle of context-aware multifactor authentication.

It is important first to define what is meant by context. The definitions of context may vary depending on the research areas, but broadly, context refers to different ambient conditions of objects that are parts of the authentication mechanism. Authors also define context as information that characterizes situations of objects that are considered relevant to the user−application interaction [2]. Context is typically the location, identity, and state of people, groups, and computational and physical objects. In our work the following examples can be shown as context factors:

- A carried device (classic multifactor authentication)
- Environmental (sound, light, images, videos, and ambient temperature)
- Internal [biometrics, Internet Protocol (IP) address, global positioning system (GPS)]

Some of these factors, such as physical location (proximity to a certain device) will be used as the primary factor, but others will be used as additional verification or protection mechanisms in combination with the primary factor. An example of such a combination is transmitting a physical location together with the exact ambient temperature of a beacon device to the authenticating server that can then be verified with the same data submitted by the client trying to authenticate, to improve the security of the process and minimize data forgery and replay attack possibilities.

Section 2 reviews classic method of two-factor authentication, Section 3 presents modern approaches, and Section 4 provides a comparative summary of some of the reviewed methods in the context of user experience and security.

2. CLASSIC APPROACH TO MULTIFACTOR AUTHENTICATION

Multifactor authentication using carried devices (a hardware token or an application on a mobile device) as a context was among the first implementations of strong security. We consider this context to be classic and review some of the most commonly used variations of such systems in this section.

Hardware Tokens

One-time passwords (OTP), as generated by a standalone hardware token, can be considered a classic method of multifactor authentication. In this example, this hardware device is serving as a proximity context proving the user has access to a physical device. For our survey, the type of the algorithms used to generate an OTP is not critical; however, we can review a number of modern hardware types to compare with each other. Most token producers are moving or have already moved to hash message authentication code (HMAC)-based [HMAC-based OTP(HOTP)]

Computer and Information Security Handbook. http://dx.doi.org/10.1016/B978-0-12-803843-7.00050-8

standard [30], and in most of cases its time-based variant, time-based OTP (TOTP) and the principle of TOTP hardware or software tokens are exactly the same; therefore we review some of the tokens that do not use TOTP as their algorithm. Hardware tokens can be of two types: (1) disconnected tokens, separate devices that have no direct connection to client system (users have to type the OTPs manually using keyboards); and (2) connected tokens, which transmit the generated OTPs to the client via a physical connection, usually universal serial bus (USB).

SecurID

RSA SecurID is one of the oldest hardware tokens on the market and used to be a de facto standard for two-factor authentication (Fig. 50.1). It uses RSA proprietary algorithms to generate OTPs. It is still popular, but after a security breach in 2011 [3], it is no considered secure by end users.

Yubico

Yubico offers a number of products that appear to achieve the real minimum of a user's interaction that is required to submit a second factor [4] and can be shown as examples of connected tokens (Fig. 50.2). Yubico's Nano and Neo hardware tokens are designed to send generated OTPs via near field communication (NFC) or USB (emulating keyboard input). These devices are indeed making two-factor authentication much easier for end users; however there are still some disadvantages. It is impossible to use the USB-based token on most mobile devices without additional equipment, and the activity range of NFC tokens is limited to 15 cm [5]. Furthermore, the number of accounts each token can use is limited to a maximum of two keys per device.

FIGURE 50.1 Hardware token for two-factor authentication.

FIGURE 50.2 Yubico connected tokens.

Software Tokens

Software tokens are applications running on a computer device, usually mobile devices. Modern mobile operating systems allow complex and powerful mobile applications to be created, so software tokens provide additional features such as multiple profiles, Quick Response (QR) code-based enrollment, and cloud backup. However, the main functionality of software tokens (generation of OTPs) is supported even on models that are not defined as "smartphones." This section reviews a variety of such applications under different platforms.

MobileOTP

MobileOTP (MOTP) was one of the first software tokens designed for two-factor authentication. The first version of MOTP was published in 2003 and was intended to be run primarily on regular phones with Java support (not smartphones) [6]. OTP codes generated by MOTP are alphanumeric codes generated based on the MD5 hash of a secret seed, current timestamp, and a personal identification number (PIN) code entered by the end user each time OTP needs to be generated. MOTP has the same level of security [7]; however, it was originally designed to work on ordinary cell phones (now called "conventional"), and therefore it lacks some features, especially the enrollment process designed to be done in the "client-to-server" direction. This means than the unique key (secret) needs to be generated on the device and then entered to the user's profile on the authentication server (Radius, Database or plaintext configuration files; there are hundreds of server side realizations done for MOTP). Fig. 50.3 shows photos of MOTP Java applications (MIDlets) running on a Nokia 3510 phone.

Google Authenticator

Google Authenticator (Fig. 50.4) is a mobile application that uses TOTP or HOTP algorithms as described by Request for Comments (RFC) 6238 [8]. The algorithm of OTP generation is based on an HMAC-Secure Hash Algorithm 1 hash of a secret key and a counter value (timestamp in the case of TOTP). The enrollment process, which is different from MOTP, is server to client, and in most cases it is based on a QR code with manual entry in case the device does not have a camera.

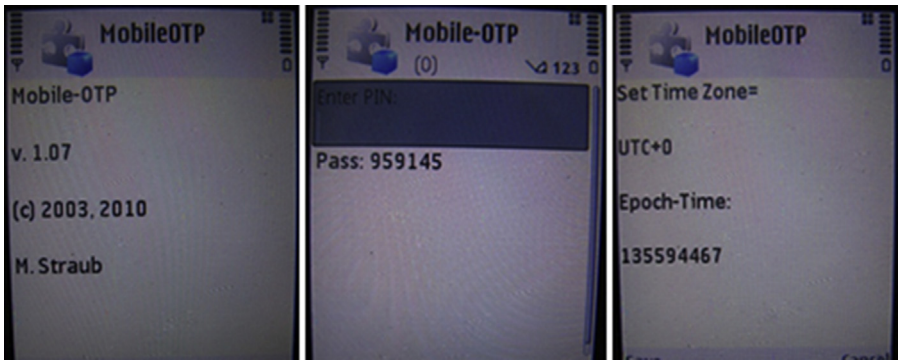

FIGURE 50.3 MobileOTP version 1.07 running on a Nokia 3510.

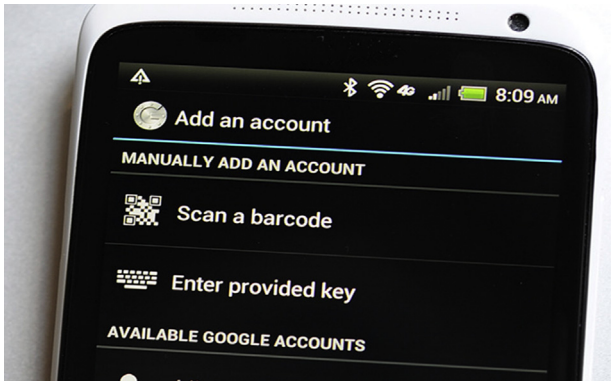

FIGURE 50.4 Google Authenticator mobile application for Android.

Google Authenticator is available on Android, Blackberry, and Apple iOS platforms. A number of applications with similar functionalities exist on alternative platforms [9].

Comparison of MobileOTP and Google Authenticator

Let us review the difference between Google Authenticator and Mobile OTP. Google Authenticator uses an HOTP-based TOTP algorithm, which is similar to MOTP but different from MOTP. With TOTP-based systems, the key is generated on the server and then shown to the client during the enrollment process. In particular with Google Authenticator, the key is shown as a QR code to be scanned by the app, which makes the enrollment process extremely easy. This, in our opinion, is the main advantage of Google Authenticator, and this is why such systems are becoming popular. However, there are some more key factors that are different, as shown in Table 50.1.

TABLE 50.1 Google Authenticator Compared With MobileOTP

	MobileOTP	Google Authenticator (TOTP)
OTP generation algorithm	MD5-based	HMAC-Secure Hash Algorithm 1—based
OTP validity time[a]	10 s	30 s
Additional PIN protection[b]	Yes	No
Key generation	Client side	Server side
Easy enrollment (with QR)	No	Yes
RFC based	No	Yes

HMAC, hash message authentication code; *OTP,* one-time password; *PIN,* personal identification number; *QR,* Quick Response; *RFC,* Request for Comments; *TOTP,* time-based one-time password.
[a]*OTP regeneration interval on the client, the server side algorithm can be set to accept previous OTPs by adjusting the timestamp used: a loop from [time ()-300] to [time ()] will accept OTPs generated during an interval of 300 s. For example, Google seems to be accepting the current and one previous OTP.*
[b]*With MobileOTP, a PIN is basically a portion of the key and is stored only on the server. Users only have to remember it and type in the client app whenever an OTP is needed. With Google Authenticator TOTP algorithm keys stored both on client and server sides need to be equal. As per the comparison table, the advantage of MobileOTP-based systems would be an additional layer of protection (PIN), although some may regard this as an inconvenience. At the same time, the lack of PIN code protection (or at least the possibility of having one) is the main shortcoming of Google Authenticator.*

Other Software Token Alternatives

Most of other software tokens use push-notifications to deliver OTP to the user's mobile device. In this case, the OTP is generated on a central server, which then sends this information over a cellular or Wi-Fi network to end users. Most such apps use TOTP as a fallback method in case the device is not online. There are alternative apps that offer additional PIN code protection to standard TOTP profiles.

Alternative Types of Strong Authentication

Although HOTP/TOTP is the standard algorithm described by the Internet Engineering Task Force for implementing two-factor authentication, other technologies provide the

same level of security using different approaches and algorithms. We will not review the proprietary systems providing such strong authentication, but will cover open-source strong authentication tokens.

Short Message Service–Based Strong Authentication

A common technology used to deliver OTPs is text messaging. Because text messaging is a ubiquitous communication channel directly available in nearly all mobile handsets and, through text-to-speech conversion, to any mobile or landline telephone, text messaging has the great potential to reach all consumers with a low total cost to implement. The implementation is simple: the authentication service creates a random value (usually digits-only PIN), and transmits it to the user's mobile phone (in some implementations this can be done via landline numbers). Then, the user enters the code received, which serves as the second factor for the authentication.

Paper Token

A number of implementations of strong authentication use a list of OTPs that are printed on a piece of paper. The logic behind it is simple: Both the server and client have a list of numbered passwords, and when logging in, the server chooses a password and prompts the user to enter it [10]. Fig. 50.5 illustrates an example of a real-life paper token–based production system [11]. The login page that is shown randomly selects a password to be entered. The user has a laminated piece of paper with these passwords, which is used to find the requested password.

Paper-Based Token Printed by Automated Teller Machine

A similar approach is used in the electronic banking system of AzeriCard. In its implementation, the lists of passwords are printed out from participant banks' automated teller machines (ATMs). As per user instructions published on the website [12], "To connect to an 'Internet Banking' system, it is necessary to obtain a list of one-time passwords in any of the information kiosks or ATMs of the Bank. To do this, in the ATM menu, select 'Payment,' then 'Services,' then 'A list of IB passwords,'" and the machine will print out a list similar to the one shown in Fig. 50.6. Although access to this list is secured with the additional factor of the banking card and its PIN code, this can be regarded as another example of paper-based strong authentication.

3. MODERN APPROACHES TO MULTIFACTOR AUTHENTICATION

Multifactor authentication is an area in which new inventions frequently appear. In this section, we will review methods, techniques, and products that were designed to replace, complement, simplify, or strengthen classic methods.

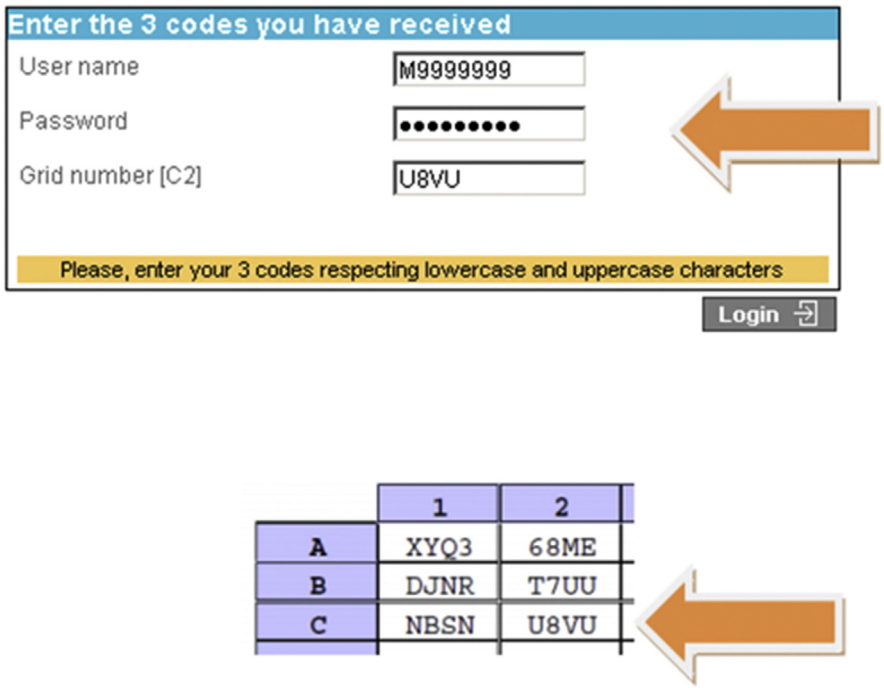

FIGURE 50.5 Example of a paper token–based system.

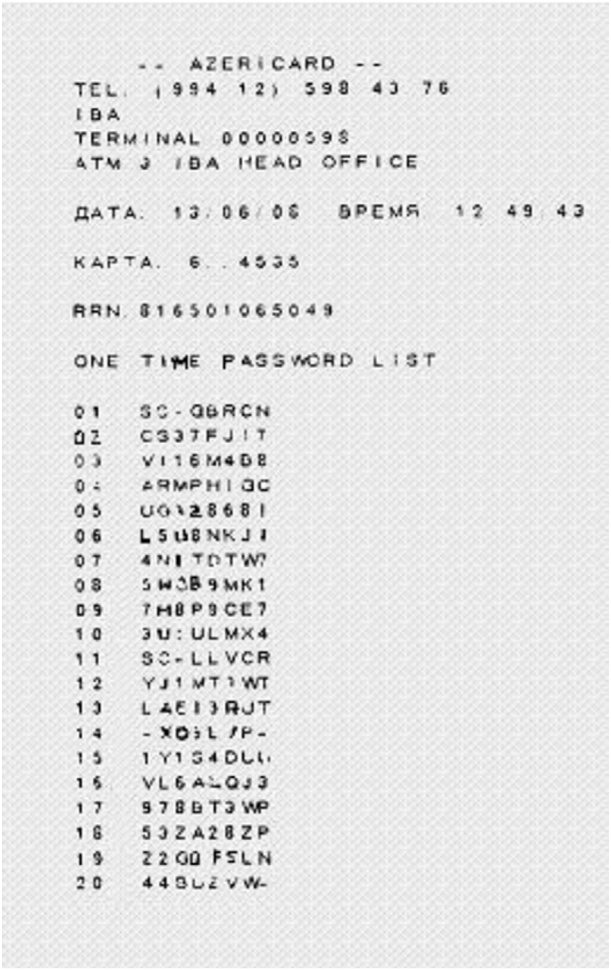

FIGURE 50.6 One-time password list printed by an automated teller machine.

Static or Pseudodynamic Context

The context factors reviewed next do not change frequently enough to qualify as strong security factors. We will review them to illustrate and compare them with other, more secure context factors.

Internet Protocol Address

It may sound controversial, but any type of authentication over a Transmission Control Protocol (TCP)/IP network already has the possibility of context-aware authentication: the IP address of the client device. Thus, restricting access to an IP address or a range of an IP addresses theoretically can be regarded as a context factor. Furthermore, because IP address databases also have associations with countries, cities, and in some cases particular buildings, they can also be considered proximity context factors.

However, it is relatively easy to forge or spoof IP addresses, so using IP address restriction itself cannot provide a satisfactory level of security. In some cases, this method can complement other methods to provide an acceptable security level. In the following example, it is used in conjunction with OTP to secure plain File Transfer Protocol (FTP) connections.

One-Time Password–File Transfer Protocol

As a protocol, FTP is known to be insecure by design and is not recommended. It should be replaced by Secured FTP or Secure Sockets LayerFTP; however, in some situations users prefer to keep using the standard plaintext FTP. OTP-FTP [13] is an attempt to make the plaintext FTP protocol as secure as possible by introducing a Web panel protected with two-factor authentication that generates a temporary session password for an FTP user and restricts access to a specific IP address. In this example, IP address is a context factor used as an additional protection mechanism.

Global Positioning System Coordinates as Security Context Factor

In some cases, access to a certain resource is restricted to a geographic location. This idea is presented in the "Secure Spatial Authentication Using Cell Phones" report [14]. The main principle is that the cell phones send GPS coordinates to both the authentication server and the cellular network, which are then compared and validated if these coordinates match. The report states that for this concept to be secure, a "using a tamper proof GPS module" should be used, which would be hard to ensure, and GPS coordinates would be easy to forge with special devices.

Ambient Temperature as a Context

A number of Bluetooth low energy (BLE) devices are equipped with temperature sensors, so it is technically possible to verify the ambient temperature when authorizing a user. Estimote devices are an example of such a device [15]. The logic behind it is simple (Fig. 50.7): the device is equipped with a temperature sensor that sends the current temperature both to the client via BLE and to the authentication server via an IP/Internet connection. Because the sensor has a high accuracy level and the current temperature may keep changing, the ambient temperature may be a constantly changing factor and can serve as a dynamic second factor.

Disadvantages of this method are obvious: In addition to hardware requirements and the accuracy level of the sensor, there is a narrow attack window inherent in the nature of the factor itself (in most areas ambient temperature stays constant and does not change significantly). However, in some cases this may be a convenient method (for outdoor areas with frequently changing temperatures).

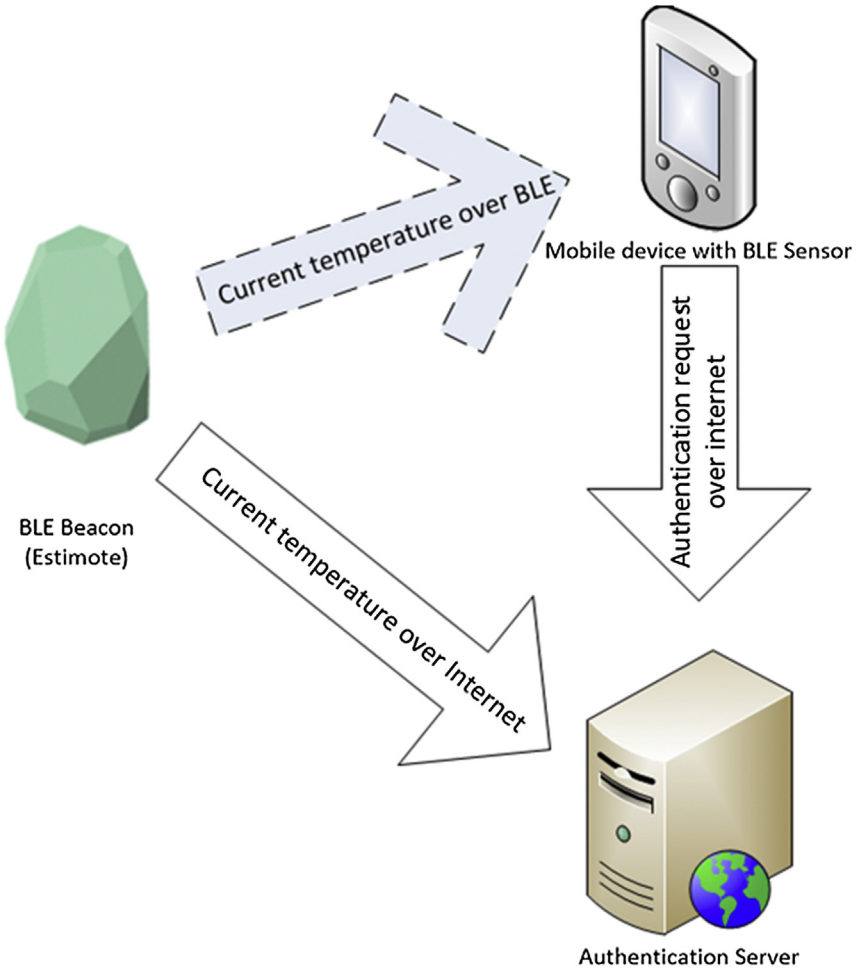

FIGURE 50.7 Temperature-based context-aware authentication with Bluetooth low energy (BLE) beacon.

Face and Voice Recognition as a Context

A number of services provide application program interfaces (APIs) to integrate users' faces and/or voice recognition as a second authentication factors. Face2.in [16] and KeyLemon [17] provide APIs as well as additional integration components (such as WordPress plugins) to enable this.

Biometrics is also relied upon in more critical areas such as online banking. Wells Fargo [18] implemented biometric logins to their online banking system using a mobile application. The application, shown in Fig. 50.8, uses both voice and face recognition to identify users.

Although using biometrics as a context in multifactor authentication significantly increases the overall security level, these factors (the user's face and voice) cannot be considered dynamic; in fact, the entropy level of these factors allows to them to be called more static factors. Some vendors try to increase entropy by asking users to move or blink their eyes, but this can easily be circumvented by using a good-quality photo of the user's face [19] and recorded voice samples to generate a voice verification response.

Dynamic Context

The context factors reviewed in this section are dynamic. In addition, they provide higher security because the attack vector is minimal, especially compared with static factors.

Bluetooth Low Energy Beacons

A user's physical location is one example of a context. Being close to a device can verify the location as well. A number of techniques use BLE (Bluetooth 4.0) to determine proximity to a device. BLE beacons periodically broadcast sets of data that includes the unique ID of a device as well as additional parameters (major ID, minor ID, and others) that can be used to deliver OTP data to end users' devices.

Authy Authy Bluetooth is a TOTP-based implementation of two-factor authentication. Because it is based on BLE protocol, the use of Authy Bluetooth [20] is limited to

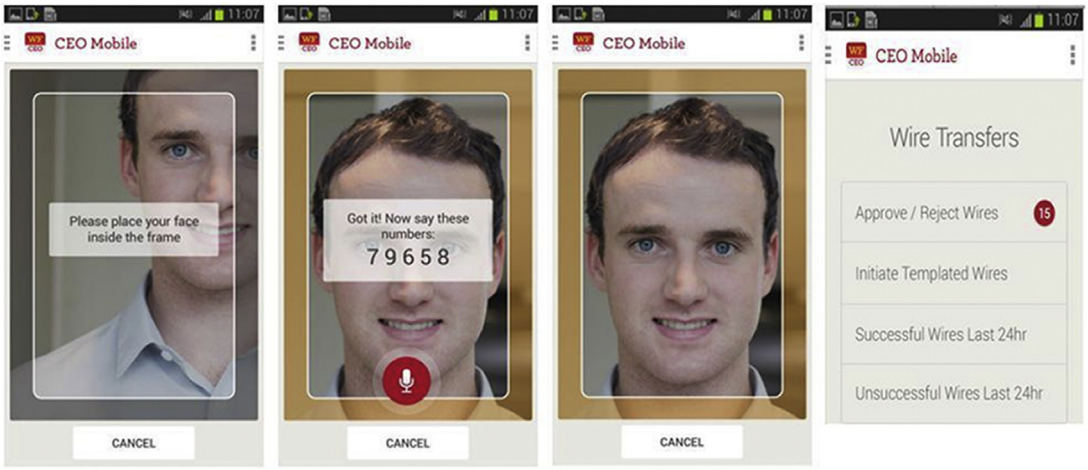

FIGURE 50.8 Biometric verification in Wells Fargo mobile application.

situations in which both a client access system (a laptop) and the system running the token (a mobile phone with the Authy app) support the BLE protocol. For this reason, Authy Bluetooth is supported only on recent Mac devices as clients, iPhone 4s and above, plus Android devices running version 4.4.4 and above with BLE support as a mobile device. In addition, the current implementation can hardly be considered a system with "minimal user interaction," because users need to: launch the Authy application and select an account, and after the current OTP is copied to clipboard, users are advised to paste it to the form requiring the OTP.

Bluetooth Low Energy One-Time Password Tokens Another system with a similar implementation was proposed by Rijswijk-Deij [21], in which TOTP broadcasts are emitted by a BLE-based token beacon device. This concept may be further developed using Eddystone, an open beacon format announced by Google [22].

SAASPASS Authentication using a virtual iBeacon as a second factor has been used in the product offered by SAASPASS [23]. Installed on a user's smartphone, the application automatically transmits the generated OTP to a special connector (currently available only for Macs). This connector searches for BLE packets transmitted in the near or immediate range, and if the packet parameters match, it passes the values gathered to the application (a browser) emulating keyboard keys. The drawbacks of these systems are that BLE is supported only on limited types of devices and Bluetooth is usually not activated on devices, because users often perceive it to be a battery hog.

Wi-Fi

A number of concepts use Wi-Fi as a proximity context. Wi-Fi can indeed be a good (or in some cases better) alternative to BLE given the range and wider spread, especially as far as hardware is concerned.

Wi-Fi Proximity A system based on WIFI Service Set Identifier (SSID) was proposed by Namiot [24], in which SSID is used as a context-aware application concept. This report describes using the information exchanged between access points and client devices to determine proximity data, and using this proximity data to send promotional information, similar to Apple's iBeacon technology but based on Wi-Fi rather than BLE. This report uses the similar concept of employing Wi-Fi SSID to relay information, but it does not provide security analysis because the system is not intended to be used as a security mechanism.

Wireless Fidelity One-Time Password This solution implements two-factor authentication without affecting the user's experience, by introducing minimum user interaction based on standard Wi-Fi [25]. The main idea is to create a dynamically changing SSID that has OTP encrypted in it. Client devices only need to "see" the name of the network without connecting to it and use a special shared encryption key to decrypt the OTP broadcasted via SSID.

Sound as a Context

A few projects employ sound as the context. A common prerequisite for using these projects is to have a device equipped with a microphone and a speaker.

Sound-Proof In Sound-Proof [26] the second authentication factor is based on verification of the user's phone's presence near the main device (Fig. 50.9). The system compares ambient noise recorded by the main system (a laptop) with sound recorded by the mobile phone. This comparison is triggered by a push notification (the ambient sound does not need to be recorded all of the time); it is done

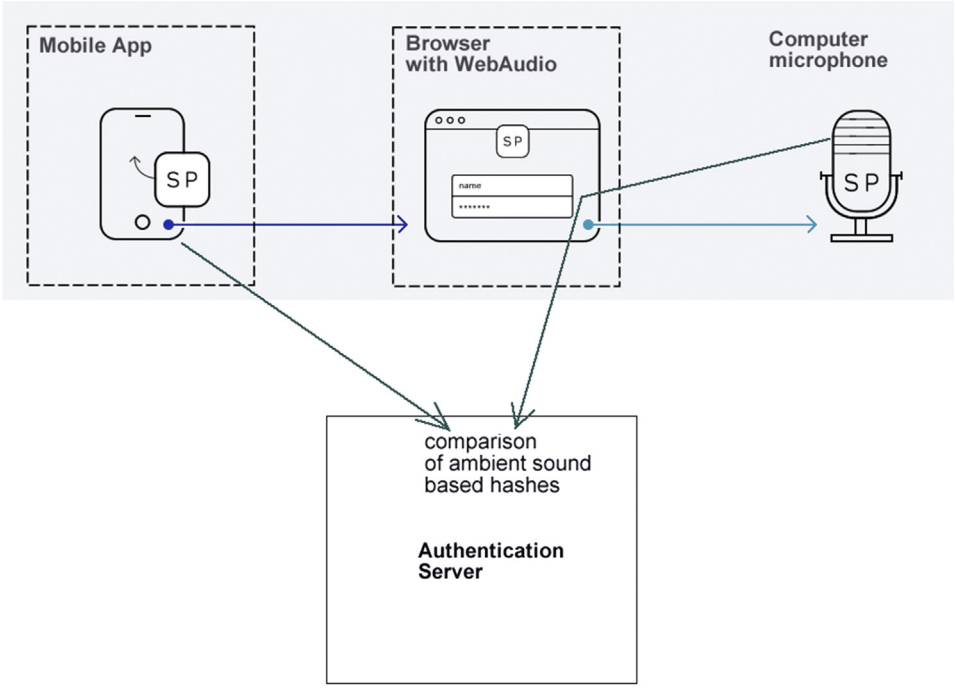

FIGURE 50.9 Sound-based authentication example. *SP*, Sound-Proof.

only for a short time when the second factor is requested by the system. In case there is no ambient sound at all (which is presented as a very rare case), the system suggests that the user "clears the throat" to generate some sound.

SlickLogin This concept is similar to Sound-Proof, but instead of relying on ambient sound, the mobile application generates inaudible sounds (ultrasound) that is captured by the main system's microphone and sent to the server for comparison [27].

4. COMPARATIVE SUMMARY

We have reviewed a variety of modern multifactor authentication systems. Each method has advantages and disadvantages. Although some methods are more secure, others are more convenient for end users. We compared MOTP and Google Authenticator in the previous section; in this section we will compare and evaluate some multifactor authentication methods, taking into account aspects of their security and user experience.

The principle to be used in the surveyed research is often based on beacons of different types. The main idea is that beacons transmit the same set of data to both the client and the authentication server. Based on the results of the comparison, the server accepts or denies the authentication request. Surveyed research uses existing beacon technologies to determine the proximity factor, such as BLE beacons (Eddystone or iBeacon) as well as innovative types, such as Wi-Fi SSID broadcast beacons, ultrasound, and Light Fidelity [28]. BLE beacons will be enhanced with additional context factors such as temperature and humidity. As surveyed earlier, biometrics-based authentication has also been used as a context (based on face recognition). This has been done for comparison purposes only, because it does not really fall under determined two-factor authentication, as the second factor in this case (the user's face or voice) is also static. A diagram of all varieties of context factors is shown in Fig. 50.10.

User Experience

As far as the user's experience is concerned, the ideal authentication mechanism should require no additional actions from users. This concept is described in the "Zero Interaction Authentication" report [29] and can be applied equally to multifactor authentication. In Table 50.2 we use this approach to evaluate the techniques reviewed in this report from the point of view of the user's experience. In addition to desktop systems, the user's experiences with mobile devices are also compared.

Security

To evaluate the security level of different token types (Table 50.3), we will review the attack type that can be applied to them and probability of the attack succeeding (see checklist: "An Agenda for Action for Token Threat/Attack Mitigation Mechanisms"). It is assumed that the first factor (username and password) has already been

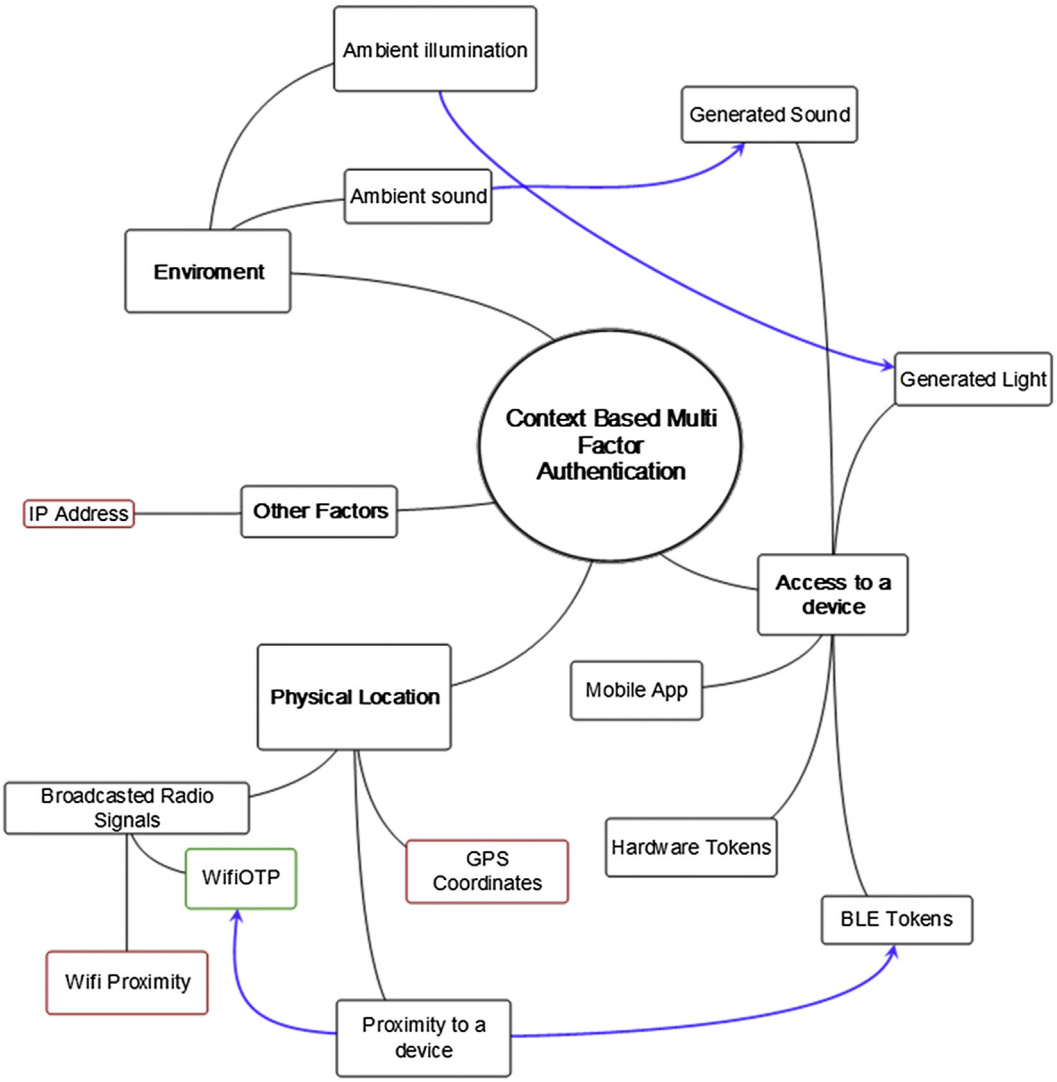

FIGURE 50.10 Diagram of context types as factors for multifactor authentication. *IP*, Internet Protocol; *WifiOTP*, Wi-Fi one-time password.

compromised. The security level is evaluated based on both the attack vector and probability.

Security Comparison Summary

Based on the comparison of security and user experience, we can provide a comparison summary (Table 50.4) of different token types when used on a desktop system. To simplify the comparison, we will compare a few classic and a few modern tokens and use the lowest ratings given in previous comparisons. The score will be calculated as an average of security and user experience ratings.

5. SUMMARY

Implementing multifactor authentication is a balance of security and user experience. Ideal multifactor authentication should simplify the user's interaction by minimizing or completely eliminating actions required to add a factor or factors for authentication. We have reviewed a number of existing multifactor authentication systems and critically evaluated them from the points of view of security and the user's experience. As an outcome of this review, we created a comparative summary of different methods that clearly show pros and cons of each system. This brings us to the conclusion that although classic methods are still secure and usable, modern methods provide the same or better security while significantly improving the user's experience.

Finally, let us move on to the real interactive part of this chapter: review questions/exercises, hands-on projects, case projects, and the optional team case project. The answers and/or solutions by chapter can be found in Appendix K.

TABLE 50.2 Token Types Comparison Based on User Experience

Token Type	User Experience: Scale of 1–10 (10 = high)	
	Desktop	Mobile
Classic hardware tokens	1	1
Connected token (universal serial bus keyboard emulation)	9	1
Software tokens	1	2
Push notification based tokens with Internet access	8	8
Tokens with Bluetooth low energy[a] broadcasts	9	2
Wi-Fi one-time password tokens[b]	9	8–10[c]

[a]Provided both client system and token are equipped with BLE modules.
[b]Only requires a Wi-Fi module to be present and enabled on the device
[c]On Android platforms only. 8: when using WifiOTP in standard apps with the Android keyboard method by users with multiple keyboards installed. 9: the same for users with a single keyboard. 10: with specially adapted apps.

CHAPTER REVIEW QUESTIONS/ EXERCISES

True/False

1. True or False? Multifactor authentication using carried devices (a hardware token or an application on a mobile device) as a context was among the first implementation of strong security.
2. True or False? Two-time passwords (OTPs) generated by a standalone hardware token can be considered a classic method of multifactor authentication.
3. True or False? Yubico offers a number of products that appear to be achieving the real minimum of a user's interaction required to submit a second factor, and can be shown as an examples of connected tokens.
4. True or False? Software tokens are applications running on a computer device, usually standalone devices.
5. True or False? MobileOTP (MOTP) is one of the first software tokens designed for three-factor authentication.

An Agenda for Action for Token Threat/Attack Mitigation Mechanisms

The token-related mechanisms that assist in mitigating the threats/attacks that are examined in this checklist include (check all tasks completed):

_____**1.** Theft: Use multifactor tokens that need to be activated through a PIN or biometric.
_____**2.** Duplication: Use tokens that are difficult to duplicate, such as hardware cryptographic tokens.
_____**3.** Discovery: Use methods in which responses to prompts cannot easily be discovered.
_____**4.** Eavesdropping: Use tokens with dynamic authenticators in which knowledge of one authenticator does not assist in deriving a subsequent authenticator.
_____**5.** Eavesdropping: Use tokens that generate authenticators based on a token input value.
_____**6.** Eavesdropping: Establish tokens through a separate channel.
_____**7.** Offline cracking: Use a token with a high-entropy token secret.
_____**8.** Offline cracking: Use a token that locks up after a number of repeated failed activation attempts.
_____**9.** Phishing or pharming: Use tokens with dynamic authenticators in which knowledge of one authenticator does not assist in deriving a subsequent authenticator.
_____**10.** Social engineering: Use tokens with dynamic authenticators in which knowledge of one authenticator does not assist in deriving a subsequent authenticator.
_____**11.** Online guessing: Use tokens that generate high-entropy authenticators.
_____**12.** Multiple factors: Make successful attacks more difficult to accomplish.
_____**13.** Employ physical security mechanisms to protect a stolen token from duplication.
_____**14.** Impose password complexity rules to reduce the likelihood of a successful guessing attack.
_____**15.** Employ system and network security controls to prevent an attacker from gaining access to a system or installing malicious software.
_____**16.** Perform periodic training to ensure the subscriber understands when and how to report compromise (or suspicion of compromise) or otherwise recognize patterns of behavior that may signify an attacker attempting to compromise the token.
_____**17.** Employ out-of-band techniques to verify proof of possession of registered devices (cell phones).

TABLE 50.3 Token Types Comparison Based on Security Strength

Token Type	Attack Vector and Method	Exploit Probability	Security Level: Scale (10 = high)
Disconnected hardware token	Trojan/keylogger on client systems	Only in real time via man-in-the-browser attacks	10
Universal serial bus connected hardware tokens	Trojan/keylogger on client systems	Only in real time via man-in-the-browser attacks	10
Push notification-based "online" software tokens	Compromised mobile devices	Can be exploited by intercepting notifications on mobile operating system level	8
"Offline"[a] software tokens	Compromised mobile devices	Can be exploited if secret hash from token app has been stolen	5
Wi-Fi OTP	Trojan/keylogger on client systems	Only in real time via man-in-the-browser attacks	10
BLE-broadcast OTP	Intercepting BLE broadcasts	Probability is minimal because physical proximity is required	9

BLE, Bluetooth low energy; *OTP,* one-time password.
[a]*Using OTP generation based on HOTP or MobileOTP.*

TABLE 50.4 Overall Token Types Comparison

Token Type	Security	User Experience	Score
Classic hardware tokens	10	1	5.5
Time-based OTP Mobile applications	5	2	3.5
Push notification-based mobile applications	8	8	8
Bluetooth low energy–based mobile applications	9	2	5.5
Wi-Fi OTP	10	9	9.5

OTP, one-time password.

Multiple Choice

1. What is a mobile application that uses TOTP or HOTP algorithms as described by RFC 6238?
   A. Google Authenticator
   B. MOTP
   C. Mobile OTP
   D. Apple iOS platform
   E. All of the above
2. Most other software tokens, are using _____ to deliver OTP to the user's mobile device.
   A. Push notifications
   B. Risk assessments
   C. Scales
   D. Access
   E. Active monitoring
3. A common technology used for the delivery of OTPs is:
   A. Organizations
   B. Text messaging
   C. Worms
   D. Logs
   E. All of the above
4. There are a number of implementations of strong authentication that use a list of one-time passwords that are printed on a piece of:
   A. Token
   B. Wood
   C. Paper
   D. Metal
   E. All of the above
5. It may sound controversial, but any type of authentication over a TCP/IP network already has the possibility of context-aware authentication: the IP address of the:
   A. Systems security plan
   B. TrustPlus
   C. Denial of service

D. Client device
E. All of the above

EXERCISE

Problem

How does an organization go about using tokens with regards to multistage authentication?

Hands-on Projects

Project

Under certain circumstances, it may be desirable to raise the assurance level of an electronic authentication session between a subscriber and a relaying party in the middle of the application session. How does an organization go about doing this?

Case Projects

Problem

Based on attacks, how would an organization go about categorizing the different types of authentication factors that comprise the token?

Optional Team Case Project

Problem

How would an organization go about categorizing credentials (objects that bind identity to a token)?

REFERENCES

[1] K.P. Weiss, SecurID. RSA Security Inc., 1988 U.S. Patent 4720860. http://www.google.com/patents/US4720860.

[2] A.K. Dey, G.D. Abowd, D. Salber, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Hum. Comput. Interact. 16 (2) (2001) 97–166.

[3] M.J. Schwartz, RSA SecurID Breach Cost $66 Million, Information Week, July 2011. http://www.informationweek.com/news/security/attacks/231002833.

[4] Yubico | Trust the Net with YubiKey Strong Two-Factor Authentication, 2011. https://www.yubico.com/.

[5] M. Abel, RFC 4729-IETF Tools, 2006. https://tools.ietf.org/html/rfc4729.

[6] Mobile-OTP: Strong Two-factor Authentication with Mobile Phones, 2005. http://motp.sourceforge.net/.

[7] This Elaboration – Mobile-OTP, 2011. http://motp.sourceforge.net/md5.html.

[8] J. Rydell, RFC 6238-IETF Tools, 2011. https://tools.ietf.org/html/rfc6238.

[9] Google Authenticator – AlternativeTo, 2013. http://alternativeto.net/software/google-authenticator/.

[10] Paper Token: Gutenberg's Version of One Time Passwords, 2010. http://www.quuxlabs.com/blog/2010/09/paper-token-gutenbergs-version-of-one-time-passwords/.

[11] Procedure to Follow to Use the Secured Access of the Website, 2011. http://www.lamutuelle.org/amfiweb/free/documents/Information_acces_securise_anglais.pdf.

[12] User Manual – Internet Banking, 2013. https://www.hbservice.com/instructions/Instruction1-engl.htm.

[13] Как сделать обычный сервер ETO Ноќнастоящему безоПасным и одновременно удобным, 2013. http://habrahabr.ru/post/205152/.

[14] A. Durresi, et al., Secure spatial authentication using cell phones. Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on 10 April 2007 543–549.

[15] Estimote SDK Updated with Accelerometer and Temperature Sensor Support, 2014. http://blog.estimote.com/post/81380655308/estimote-sdk-updated-with-accelerometer-and.

[16] FACE2.in, 2015. https://face2.in/.

[17] KeyLemon – Face Recognition Technology, 2009. https://www.keylemon.com/.

[18] Wells Fargo to Roll Out Biometric Logins by June | Bank Innovation, 2016. http://bankinnovation.net/2016/01/wells-fargo-to-roll-out-biometric-logins-by-june/.

[19] Q. Xiao, Security issues in biometric authentication, in: Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC 15 June 2005, 2005, pp. 8–13.

[20] Two-Factor Authentication • Authy, 2012. https://www.authy.com/.

[21] R. van Rijswijk-Deij, Simple Location-Based One-time Passwords.

[22] Beacons | Google Developers, 2015. https://developers.google.com/beacons/?hl=en.

[23] Bluetooth, BLE, Two-factor Authentication | SAASPASS, 2014. https://www.saaspass.com/about/bluetooth-ble-two-factor-authentication.html.

[24] D. Namiot, Network proximity on practice: context-aware applications and Wi-Fi proximity, Int. J. Open Inf. Technol. 1 (3) (2013).

[25] E. Huseynov, J.-M. Seigneur, WifiOTP: Pervasive Two-factor Authentication Using Wi-fi SSID Broadcasts, 2015.

[26] N. Karapanos, et al., Sound-Proof: Usable Two-factor Authentication Based on Ambient Sound, arXiv preprint arXiv:1503.03790, 2015.

[27] Google Acquires SlickLogin, the Sound-based Password Alternatives, 2014. http://techcrunch.com/2014/02/16/google-acquires-slicklogin-the-sound-based-password-alternative/.

[28] Z. Zhou, et al., Lifi: line-of-sight identification with wifi. INFOCOM, 2014 Proceedings IEEE 27 April 2014 2688–2696.

[29] H.T.T. Truong, et al., Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication. Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on 24 March 2014 163–171.

[30] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O. Ranen, HOTP: An HMAC Based One-time Password Algorithm, RFC 4226, IETF, 2005.