Multi-Factor Authentication for Shibboleth Identity Providers

DE MELLO, Emerson Ribeiro, WANGHAM, Michelle Silva, LOLI, Samuel Bristot, DA SILVA, Carlos <http://orcid.org/0000-0001-9608-439X>, DA SILVA, Gabriela Cavalcanti, DE CHAVES, Shirlei Aparecida and LOLI, Bruno Bristot

Published version

DE MELLO, Emerson Ribeiro, WANGHAM, Michelle Silva, LOLI, Samuel Bristot, DA SILVA, Carlos, DA SILVA, Gabriela Cavalcanti, DE CHAVES, Shirlei Aparecida and LOLI, Bruno Bristot (2020). Multi-Factor Authentication for Shibboleth Identity Providers. Journal of Internet Services and Applications, 11 (1), p. 8.

Journal of Internet Services and Applications

RESEARCH Open Access

Multi-factor authentication for shibboleth identity providers

Emerson Ribeiro de Mello[1*], Michelle Silva Wangham[2], Samuel Bristot Loli[1], Carlos Eduardo da Silva[3], Gabriela Cavalcanti da Silva[4], Shirlei Aparecida de Chaves[1] and Bruno Bristot Loli[1]

*Correspondence:
mello@ifsc.edu.br
[1]Federal Institute of Santa Catarina (IFSC), São José, SC, Brazil
Full list of author information is available at the end of the article

Abstract
The federated identity model provides a solution for user authentication across multiple administrative domains. The academic federations, such as the Brazilian federation, are examples of this model in practice. The majority of institutions that participate in academic federations employ password-based authentication for their users, with an attacker only needing to find out one password in order to personify the user in all federated service providers. Multi-factor authentication emerges as a solution to increase the robustness of the authentication process. This article aims to introduce a comprehensive and open source solution to offer multi-factor authentication for Shibboleth Identity Providers. Based on the Multi-factor Authentication Profile standard, our solution provides three extra second factors (One-Time Password, FIDO2 and Phone Prompt). The solution has been deployed in the Brazilian academic federation, where it was evaluated using functional and integration testing, as well as security and case study analysis.

Keywords: Multi-factor authentication, Federated identity management, Shibboleth identity provider

1 Introduction

Identity proofing establishes that a subject is who they claim to be. Digital authentication establishes that a subject attempting to access a digital service is in control of one or more valid credentials associated with that subject's digital identity [1].

The classic paradigm for authentication systems identifies three categories of authentication [1]: something you know (e.g., a password); something you have (e.g., a hardware token); something you are (e.g., a fingerprint); and in [2] someone you know (e.g., social network). There are advantages and drawbacks for each of the above categories that can deny legitimate user access. Passwords can be forgotten, smart cards misplaced and biometrics can become temporarily unavailable, such as, loss of voice or poor quality fingerprint. Furthermore, biometrics is not secure if used as a single authentication factor since it can be replicated (an attacker may obtain a copy of the subscriber's fingerprint and build a replica) [2].

Currently, password-based credentials are the most used by authentication mechanisms, despite of their weaknesses [2, 3]. Users tend to use easy-to-guess passwords, use the same password in multiple accounts, write the passwords or store them in a insecure way in their machines, etc. There are multiple opportunities for identity theft (the act of impersonating users with stolen credentials) [1], and this problem has been receiving increasing attention due to its high financial and social costs [4]. Moreover, attackers have the option of using many techniques to steal passwords such as shoulder surfing, snooping, sniffing, guessing, phishing, social engineering, brute force attacks, bought leaked passwords, etc [5].

The Federated Identity Management (FIM) model allows identity portability across distinct domains [6]. In this model, the user only needs to authenticate with a single Identity Provider (IdP) to access different service providers (SP). The FIM model also offers the convenience of Single Sign On (SSO), which allows the user to be authenticated in his/her home identity provider only once, regardless of how many service providers will be accessed. The Security Assertion Markup Language (SAML) specifications [7] set is prominent to develop identity management systems based on this model.

Identity theft is a serious concern in the FIM model since federated SSO simplifies the job of the attackers. After a successful identity theft within a federation, attackers are provided with means to compromise resources of all federated service providers, potentially leading to disclosure of sensitive data [8].

Multi-factor authentication (MFA), sometimes called Two Factor Authentication (2FA), emerges as a solution to increase the robustness of the authentication process. MFA combines more than one authentication factor chosen from independent credential categories, such as password and hardware token, or two or more factors from the same category such as the case of One-Time Password (OTP). In this case, it is assumed that even though an attacker compromises one of the factor, the level of protection is greater due to the fact that the attacker needs to compromise the remaining factors. The benefits of adopting MFA are to provide a resilient way of authenticating users and is considered as one of the most cost-effective mechanisms [3].

The Shibboleth framework [9] uses SAML specifications, and it is the most adopted solution in academic federations. In June of 2017, the REFEDS (Research and Education FEDerations group) announced the REFEDS *Multi-Factor Authentication Profile* (MFA Profile) [10] that defines a SAML authentication context to express MFA in SAML assertions and specifies requirements that an authentication event must meet to communicate the usage of MFA. Thus, this profile allows SPs to request user MFA from IdPs, and in turn, the IdPs can notify SPs that MFA was successful.

However, existing non-proprietary MFA solutions in the Shibboleth IdP do not follow this specification. Thus, resulting in different third party products with specific architecture and implementations, which are usually not interoperable with other MFA solutions and require customization effort in order to be adopted or upon new versions of the Shibboleth framework.

This article aims to describe a comprehensive solution for Shibboleth IdPs so that they are able to authenticate their users with multiple authentication factors. Our main contribution is a consistent implementation of a multi-factor authentication architecture for Shibboleth IdPs. The proposed solution is based on the MFA Profile standard [10], which guarantees the interoperability with other multi-factor authentication solutions. The solution was designed to have low coupling with the IdP, to be user-friendly, flexible to the end-user and IdP operators, and extensible, which allows new technologies to be used as extra authentication factors. The lifecycle management of the second factor is also supported in the proposed solution. A prototype was implemented and evaluated, considering four extra authentication factors: Time-base One-Time Password (TOTP) [11], FIDO2 [12], Phone Prompt and Backup codes. Android and iOS mobile applications were also developed to act as a 2FA for Phone Prompt technology. The evaluation consisted of functional and integration testing, security and case study analysis.

The remainder of the article is organized as follows. Section 2 analyzes some related work. The proposed solution and the developed prototype are described in Section 3 and the prototype evaluation is discussed in Section 5. Finally, Section 6 presents the concluding remarks as well as future work suggestions.

2 Related work

In 1991 Weiser [13] presented the vision that the computer of the 21st century would be mobile and ubiquitous. This vision became true mainly due to smartphones, which from the point of view of information security, made it easier for the user to deal with multi-factor authentication systems. In addition, they reduce the cost of manufacturing, distributing, and storing tokens. In fact, many commercial solutions use smartphones to receive short text messages or to execute OTP applications [5].

The use of telecom networks to send OTP has been considered insecure by NIST [1]. This led to the development of several solutions for MFA. Google Phone Prompt [14] is a 2FA standard for Google accounts that was created as a response to usability and security issues in existing OTP products. This solution is exclusive for Google service users and depends on smartphones and Internet connectivity. There are other companies that offer a similar product such as *Authy OneTouch* from Authy, *Push-to-Accept* from Secureauth [15] and *DUO push* [16] from DUO Security. All of which are proprietary solutions.

Although there are several products for MFA, to the best of our knowledge, there is not a standardized way of integrating them into a Shibboleth IdP. Among MFA solutions for the Shibboleth framework, it is possible to find: FAME [17]; *Multi-Context Broker* (MCB) [18]; *Authentication Flow Selection* [19]; and the practical experiment described in [20]. FAME framework [17] enables multi-factor and multilevel authentication in Shibboleth. FAME adds the authentication factors Level of Assurance (LoA) in the generated SAML assertion. This information will be used to apply the fine-grained access control later on. However, the integration of FAME with Shibboleth requires the change of the Shibboleth Handle Service, which is responsible to identify and authenticate the user. FAME was released prior to MFA Profile publication, thus FAME does not offer support to MFA Profile.

MCB [18] and Authentication Flow Selection [19] extended the authentication flow from the Shibboleth IdP in order to allow the second-factor authentication. However, when those were created, there was not a standard specification to indicate how to perform 2FA authentication and how to notify the SPs that a 2FA authentication was used. Thus, these solutions do not guarantee the interoperability among them or with any other MFA solution in Shibboleth IdPs. It is important to highlight that these solutions depend on third-party solutions, such as DUO Security, not only for 2FA authentication but also to offer the application that will be used as a second factor by the user.

In [20], the authors describe a practical experiment that integrates an external password-less authentication system based on FIDO2 in the Shibboleth framework. In the proposed solution, the Shibboleth IdP redirects the user client (browser) to the FIDO2 server. At this phase, the user gets authenticated locally on his or her device (e.g. Windows Hello biometric authentication) and the authentication result is translated to the FIDO2 server. If authentication is successful, the FIDO2 server then redirects the client back to the IdP, which then issue the SAML assertion with the user's attributes to the SP. This approach provides single factor authentication using FIDO2, however, it does not offer a multiple factor authentication solution.

Different from the approaches presented above, our proposal is a comprehensive open source solution that does not change the Shibboleth framework. Based on the REFEDS MFA Profile, our approach offers components to extend the Shibboleth IdP authentication flows to support multiple factors, leveraging the MFA Profile specification to express information about the multi factor authentication in SAML assertions. In this way, we guarantee interoperability between different products that follow the MFA Profile, while supporting different authentication factors to be plugged into a Shibboleth IdP. In fact, to the best of our knowledge, our approach is the first and comprehensive open source solution to offer multi-factor authentication for Shibboleth Identity Providers. The following section details the developed solution in this work.

3 Multi-Factor authentication in shibboleth identity providers

The Shibboleth framework [21], which implements the SAML specification, has been used by several research and education (R&E) federations around the world, and until 2017, there was not a standard to offer multi-factor authentication to Shibboleth IdPs. From version 3.3 the Shibboleth framework included support for multi-factor authentication, however a functional implementation that allows MFA is not provided. In our proposal, we have used the MFA Profile [10] specified by REFEDS, developing a concrete implementation for adding MFA into Shibboleth IdPs, making them capable of interoperating with existing solutions that implement the SAML specification.

According to the MFA Profile specification, SPs now have the option of requesting IdP to perform multi-factor authentication of its users, or to completely ignore how users are authenticated. Our proposal allows a seamless integration with SPs in both scenarios. The later scenario means that SPs are not aware if the user was authenticated on an IdP using one or two factors, or even which technology was used as the second factor. On the other hand, in the case multi-factor authentication is a mandatory requirement by the SP, the MFA Profile allows SPs and IdPs to express such information as part of the messages exchanged between them.

Figure 1 presents a high-level view of our proposed solution. The Multi-Factor Provider (MFaP) – the main component of this proposal – is a Java application that can be deployed

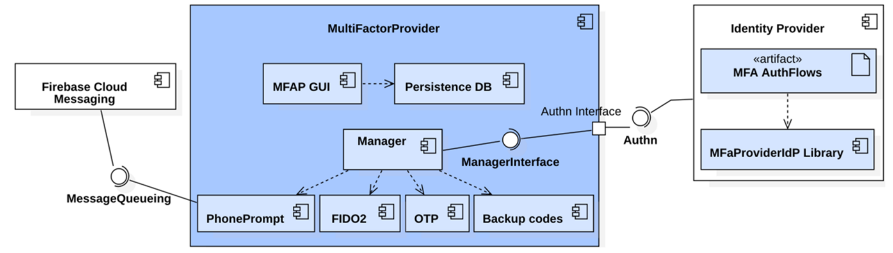


Fig. 1 Component diagram of Multi-Factor Provider (MFaP)

on any Java application server (e.g., the same server where Shibboleth IdP is deployed, or in a different computer). The interaction between the MFaP and the IdP is done through a REST interface provided by the former and the authentication flow scripts [19] by the latter. We developed a MFaProviderIdP library to be deployed in the Shibboleth IdP, which is used by IdP authentication flows to invoke the MFaP application. We are aware that it is difficult to setup, deploy and maintain an identity provider within a production environment. Thus, the current proposal has a loose coupling from the identity provider point of view.

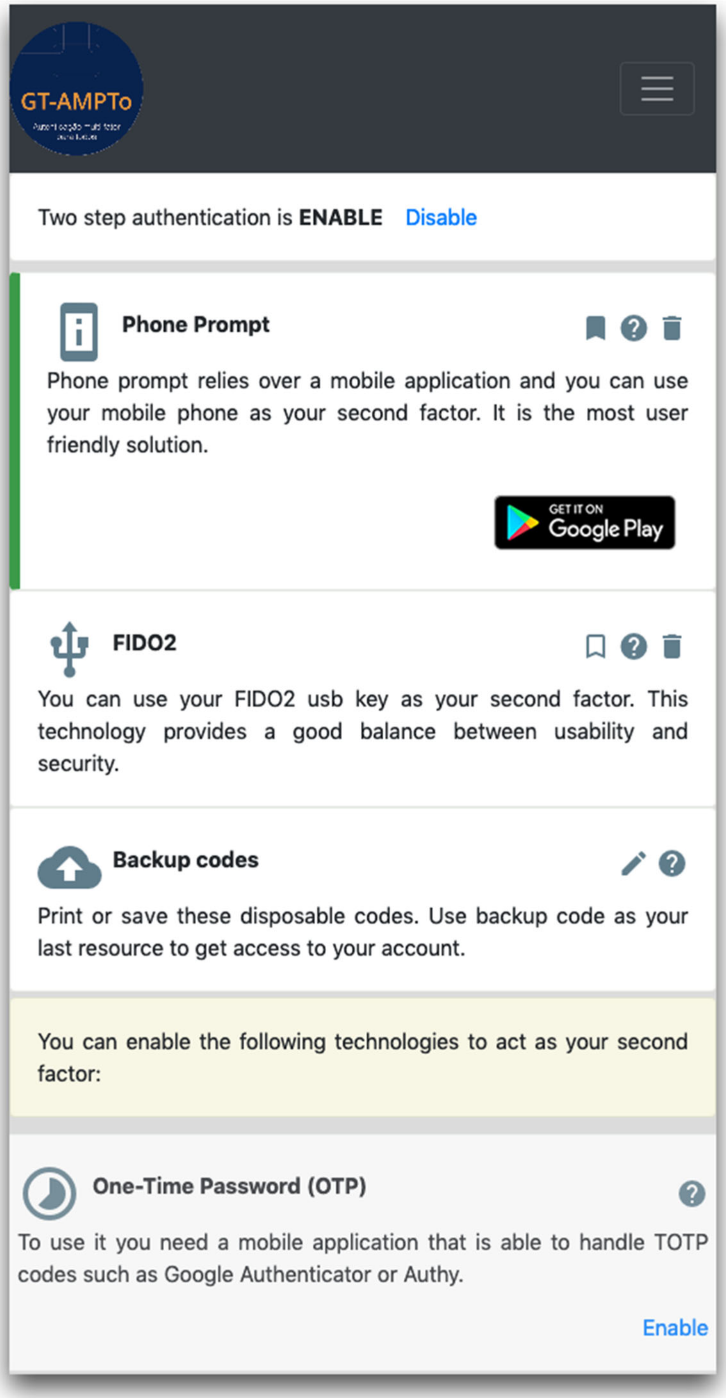
The Shibboleth framework uses Spring Web Flow to orchestrate IdP's business logic that includes the user authentication process. In the authentication flow, it is possible to perform steps such as: how to collect user credentials; how to validate that credentials; or even how to request extra credentials. A Shibboleth IdP can implement one or more authentication flows where each flow can have its own sub-flows or interact with other authentication flows at the same level. Thus, an authentication request is handled by one authentication flow that can invoke its sub-flows or another authentication flow and, at the end, the IdP produces an *Authentication Result* [22].

The IdP authentication flow (*MFA AuthFlows* in Fig. 1) is one of the XML configuration files of a Shibboleth IdP, where it is possible to write rules combining different authentication factors that result in a combined sequence of challenges. In this way, the modification of an IdP for inclusion of MFaP follows standard Shibboleth administration practices for its deployment, update and maintenance. In this proposal, the MFA AuthFlow XML is the place where the Shibboleth IdP consumes the REST interface provided by the MFaP (see Manager and MFA AuthFlows components on Fig. 1). We used REST for this communication in order to allow the loose coupling between the MFaP and the Shibboleth IdP.

The user and the IdP operator flexibility is the core of our solution. MFaP was designed to allow the use of different types of technologies as the second factor. Thus, the IdP operator is able to choose which technologies will be offered to provide support. Additionally, each user has the opportunity to enable the second-factor authentication on the user's account and to choose which IdP supported technologies will be his/her second factor.

Currently, the Manager component in Fig. 1 is able to work with four different technologies to act as extra authentication factors: Phone Prompt, TOTP, FIDO2 and Backup codes. Nevertheless, as previously mentioned, our solution can easily be extended to incorporate other authentication factors. This is achieved by following the strategy design pattern. This way, we have defined an abstract Java interface that is consumed by the Manager component. Thus, the addition of a new factor corresponds to the deployment, in the MFaP, of a Java class that implements this interface and acts as an entry point for the desired authentication factor and the respective XML configuration at the Shibboleth IdP. It is essential to mention that no changes are needed in the REST API provided by the MFaP or the MFaProviderIdP library, deployed in the IdP. Also, any information about selecting a particular authentication factor is done following the MFA Profile specification (through the XML configuration) and passed as parameters between the IdP and the MFaP.

MFaP GUI component in Fig. 1 provides a dashboard (see Fig. 2) where users can enable or disable their second factors. The MFaP GUI is a service provider that has an unique trust relationship with the Shibboleth IdP where the MFaP was deployed. Thus, the access



Two step authentication is **ENABLE**

Phone Prompt

Phone prompt relies over a mobile application and you can use your mobile phone as your second factor. It is the most user friendly solution.

FIDO2

You can use your FIDO2 usb key as your second factor. This technology provides a good balance between usability and security.

Backup codes

Print or save these disposable codes. Use backup code as your last resource to get access to your account.

You can enable the following technologies to act as your second factor:

One-Time Password (OTP)

To use it you need a mobile application that is able to handle TOTP codes such as Google Authenticator or Authy.

Fig. 2 Second factor life-cycle management provided by MFaP Dashboard

to the dashboard page will be granted only to users authenticated by this specific IdP. The MFaP GUI dashboard provides a modular graphical interface. Each extra factor technology has its own card (a visual component) on the dashboard. The IdP's operator can determine which MFaP supported technologies he or she wants to enable to his or her users. It can be done by switching some lines in the MFaP configuration file to true or false.

Even though the IdP performs user authentication, user data such as username, password hash, etc. are usually stored in a database that is shared by other institution's information system. For example, LDAP is the most common solution to store user credentials that are used in authentication processes. Some institutions also use LDAP to store a rich set of attributes for their users such as address, the institutions' affiliations, department, school, etc. Other may use relational database to store such kind of information. Our proposed solution provides an uniform layer for persistence (Persistence DB component in Fig. 1) that abstracts away from the underlying database technology used by the institution. The current implemented solution uses MongoDB – a NoSQL database – to store information related to extra authentication factors for each user. Thus, if the institution chooses to keep all the information about its users in an unique database solution (for example, in its current LDAP), it can easily do so by to extending the Persistence DB layer using this database solution.

We have developed a mobile app called AMPToAPP for both Android and iOS platforms to support Phone Prompt authentication technology. AMPToAPP has been designed to support more than one user account in a single or multiple IdPs. Its current implementation allows the use of Phone Prompt as authentication factors, being ready to use without any modification by any institution that deploys MFaP. Nevertheless, it can be easily modified for using institutional branding, for instance. The source code of our solution is under Apache License 2.0, and it is available in a Git repository[1].

In the sequence, we describe each of the supported technologies for multi-factor authentication currently supported by MFaP.

3.1 Time-based one-time password – (TOTP)

Currently, the most implemented 2FA method uses OTP acquired from SMS, phone calls or mobile applications. The TOTP [11] standard is used by several 2FA solutions and smartphones has helped its rise. The TOTP smartphone application is a cheap solution and it can be deployed easily because almost everyone has a smartphone and TOTP applications (Android or iOS) can be downloaded for free.

Smartphone TOTP applications have as advantage the possibility to use a unique application to manage all user TOTP tokens for different institutions; a lot of users already use TOTP applications on their smartphones; and it is also possible to use it even when there is no Internet connection.

TOTP has been implemented using the GoogleAuth [23] library, a Java server side library that implements functionalities to create TOTP shared secrets and to generate and verify TOTP passwords. In our implementation the TOTP module is composed by two main classes: `TOTPFactor` (model) and `TOTPService` (service).

The `TOTPFactor` contains following attributes:

- `sharedKey` – Shared secret between client and server. The secret is generated in the MFaP and shared with the client through a QR code using an app like Google Authenticator or Authy;

[1] https://git.rnp.br/GT-AMPTo/mfap-installation-guide

- `timeStep` – Regular time interval in which the generated one-time password is valid. The RFC 6238 [11] recommendation is 30 seconds, in order to balance security and usability. The GoogleAuth library uses this recommended interval as a fixed value, however the TOTPFactor entity implements this attribute aiming to enable configuration of future library updates or library exchange;
- `lastCodeUsed` – A one-time password can only be used once. Thus, this attribute stores the last valid used code to avoid subsequent requests reusing the same code;
- `usedAt` – A hash algorithm is used in the calculation performed to generate the one-time password and therefore a collision, even with very low probability, may eventually occur - a valid one-time password that is the same as the last one used, but at a different time interval. Therefore, in addition to saving the last one-time password used, the instant in which it was used is also saved.

The service layer class has been implemented in order to be exchangeable to any other library that implements TOTP algorithm, as long as it keeps the same signature for the methods used in the web layer.

The authentication process using TOTP technology is illustrated by Fig. 3. In step 1, the user enters his/her first factor credential. The IdP verifies that the user has 2FA enabled and sends an authentication request to the MFaP (step 2). The MFaP then presents the user with a Web page requesting the OTP code (step 3). At this point the user needs to search and open the TOTP application in his/her smartphone, check the temporary number on the screen and type it on the IdP Web page (step 4) before this code expires (which usually happens in 30 seconds). The MFaP receives the code (Step 5) and verify it, and in case of successful authentication, returns a response to the IdP (step 6) for finishing the authentication process. It is important to notice that, since it generates an authentication code based on a previously exchanged secret key and the current time, the TOTP application does not need to communicate with the MFaP in order to authenticate the user.

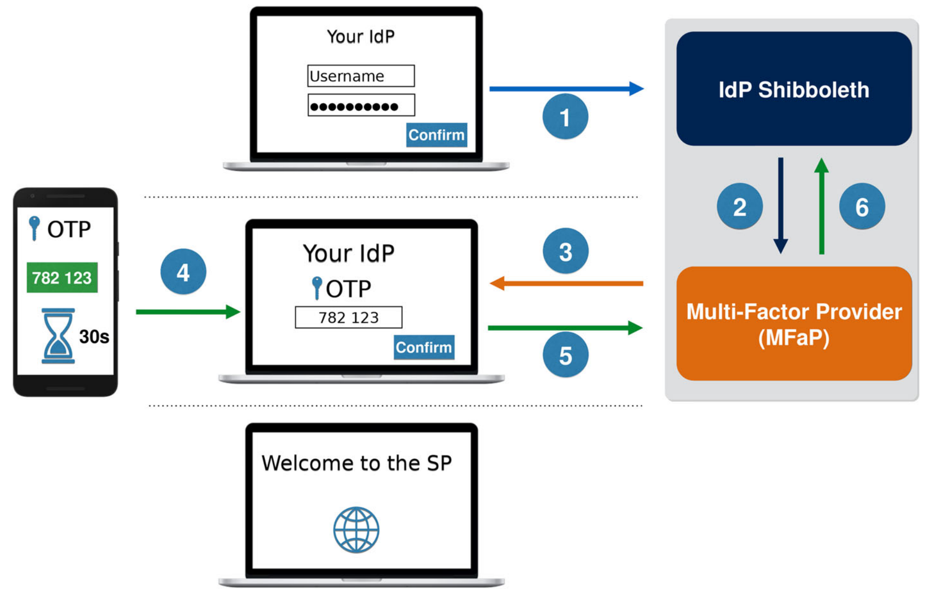


Fig. 3 Authentication process using TOTP as the second factor

3.2 Phone prompt

Second factor authentication usually provides a negative impact over usability, such as the need to open the TOTP application and manually type the generated number into the Web page. It is usually the case where the code expires whilst the user is still typing it.

Solutions that rely on Phone prompt, also called push dialog, aim to increase the robustness of the authentication process with a minor impact on the usability. This 2FA technology is based on the user receiving a notification on his/her smartphone, which opens a prompt with a simple question: *Are you trying to authenticate right now? (Yes or No)*. It is more user friendly compared to the TOTP solution.

The implementation of the Phone prompt solution requires an application on smartphone and a message queue service to intermediate the communication between the IdP and the mobile application on the user's device. The Google Firebase Cloud Messaging (FCM) [24] offers a reliable service to exchange messages with Android, iOS or even Web applications. In this work, we have developed a mobile application for Android and iOS and both of them use FCM to exchange messages with our MFaP.

The authentication process message exchange is illustrated by Fig. 4. In step 1, the user enters his/her first factor credential. The IdP verifies that the user has Phone Prompt as his/her 2FA and it then sends an authentication request to the MFaP (step 2). The MFaP generates a nonce and sends a message to the user device through FCM (steps 3 and 4). The Phone Prompt application goes to the foreground on user's device and the user presses the YES button. The FCM sends the user's response to the MFaP (step 5) and to the IdP (step 6) where the authentication process is finished.

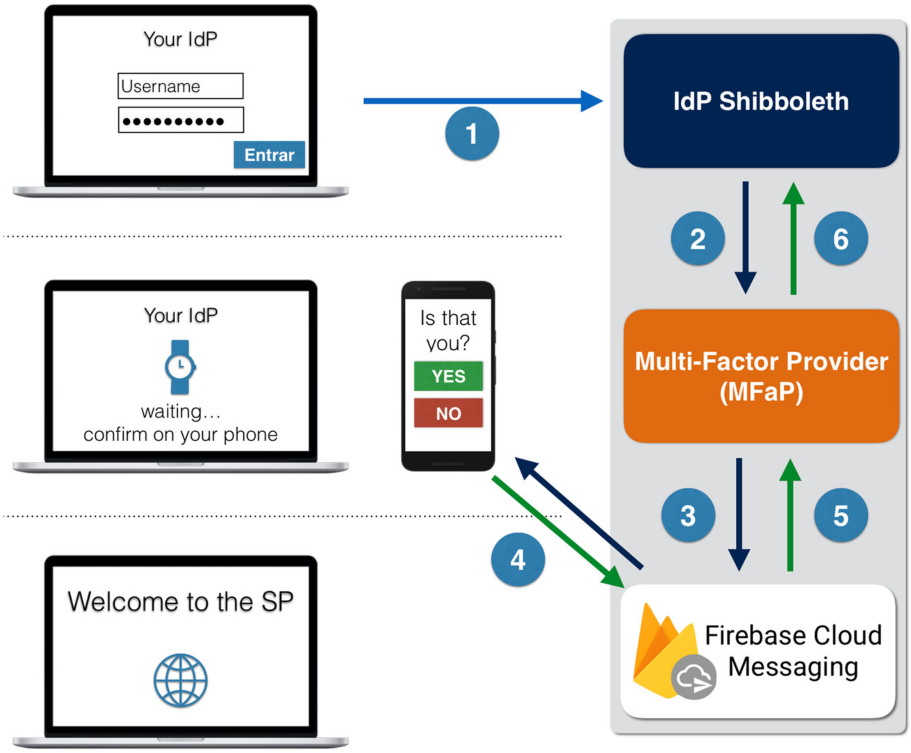


Fig. 4 Authentication process using Phone Prompt as the second factor

3.3 FIDO2

The Fast IDentity Online (FIDO) Alliance, together with the World Wide Web Consortium (W3C), developed open standards to create a strong user authentication solution based on public key cryptography, to be user friendly, and to preserve user privacy, which can also be used as a 2FA authentication method.

The FIDO Universal Second Factor (U2F) specification [25] allows the second factor authentication without the use of a sophisticated device that requires energy source such as a smartphone. FIDO Universal Authentication Framework (UAF) specification [26] allows a password-less experience. That is, the user does not need a password as a first factor or even as a second factor. The user is authenticated locally in the device using his/her biometrics, and this authentication is transposed to remote services (relying party – RP) through the protocols of the FIDO specification. FIDO Alliance proposes a solution strongly based on the trust of RP over the user device being used during the authentication. In this case, the user device (tablet, smartphone, smartwatch, or even a personal computer) must be certified by FIDO Alliance.

Based on the FIDO UAF and FIDO U2F standards, Web Authentication (WebAuthn) [12] and Client to Authenticator Protocol (CTAP) [27] emerged under the term FIDO2. They are today an industry standard for robust authentication. In this work, we choose it to offer a 2FA option that does not depend on, a smartphone or even a 2FA device that relies on an Internet connection.

The WebAuthN specification defines a standard API that has been built into the major Web browsers and platforms to allow online services to use FIDO authentication. Together with the FIDO CTAP [27], it allows external devices such as smartphones or USB tokens to use WebAuthN and become authenticators for desktop and web applications. In this way, the browser uses the USB key to authenticate the user.

The FIDO2 specifications support different authentication scenarios, with varying devices, such as keys, smartphones and smartwatches, communicating through USB, Bluetooth, or NFC. Currently, MFaP supports only one associate FIDO2 USB key per user account. Therefore, it is possible to use the same FIDO2 USB Key in different MFaP accounts.

Our implementation of FIDO2 WebAuthn is based on the Yubico webauthn server demo [28]. It provides a Java library that has been used as the core of MFaP FIDO2 module REST API. Our API is based on the `WebAuthnRestResource` class, and includes all relying party operations provided by Java WebAuthnServer library, such as, generating authentication challenges and verifying responses, registration, login and deletion. These operations, and the general business logic of FIDO2 module are implemented by the `WebAuthnServer` class. We have also implemented the `CredentialRepository` interface to allow the use of our Persistence DB abstraction layer for storing keys and other information related to FIDO2.

The authentication process using FIDO2 WebAuthn as the second factor is demonstrated in Fig. 5. In the first step, the user provides his/her first factor (username and password) on the Shibboleth IdP login page. The IdP interacts with MFaP and confirms that the user has FIDO2 enabled as his/her second factor. Then, IdP/MFaP provides to the user a Web page with an embedded WebAuthn JavaScript routine, where a FIDO2 challenge is generated (step 2). The user's browser WebAuthn API is triggered and it shows the WebAuthn authentication dialog, where the user is requested to connect his/her USB

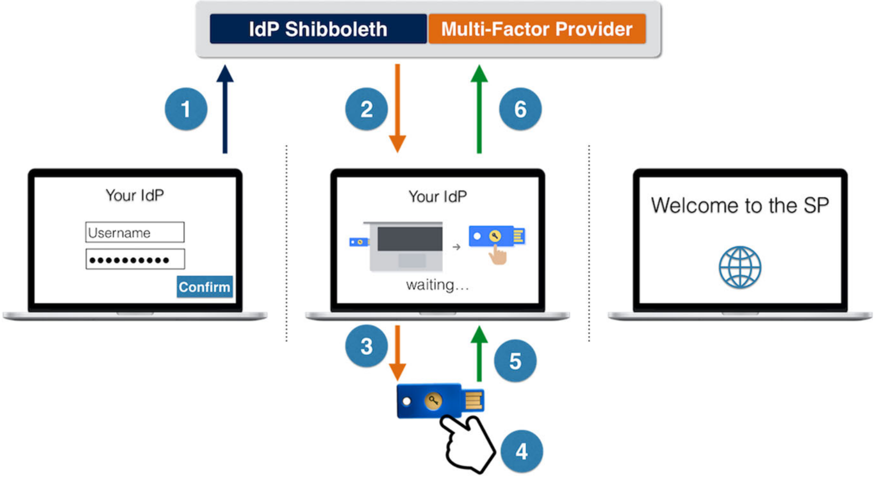


Fig. 5 Authentication process using a FIDO2 USB device as the second factor

key and touch on its physical button (steps 3 and 4). The USB key generates the response to the challenge and sends it to user's browser (step 5). The user's browser forwards the response to the IdP/MFaP (step 6). Finally, MFaP verifies that the response is correct and the authentication process is successfully concluded, and the user is redirected to the SP web page.

3.4 Backup codes

Printed backup codes are usually used by many commercial service providers as a last resort for users, who have 2FA enabled to recover access to their accounts. In our solution, when the user associates a second factor to his/her account, a set with ten disposable codes is generated automatically, and the user is invited to print or save them in a file.

Backup codes are generated by MFaP using the same library of the TOTP (GoogleAuth). Each disposable code can be used only once, and MFaP keeps track of those backup codes and when they were used, so the user can manage what codes have already been used and how many are still left. The user could generate a new set of codes when the current one runs out (a pencil icon close to the Backup codes in Fig. 2).

Although backup codes only act as a second factor and can not be used separately from the first factor, these should be kept in a safe place. Otherwise, an attacker, who obtains them and who already knows the victim's first factor, will be able to impersonate him/her. This condition is not exclusive of the current proposal and it is present in many commercial service providers, which relies on backup codes as a second factor.

It is important to notice that backup codes are not intended to be used as a regular second factor, but as a safety measure for the case of losing or not having access to the token device. Thus, it should be treated as a last resort before interacting with a human support operator.

4 Second factor life-cycle management

MFaP has been designed as a comprehensive solution for IdP operators, and thus has included aspects related to the life-cycle management of second factor. Such management is composed by events such as, registration, replacement and revocation. The procedure adopted by an organization to handle each one of these events will directly impact the robustness of its user authentication process. In this work, we assume that the first factor of each MFaP's user consists in the tuple (username, password). These are the same credentials that allow user to have access to his institutional email account, a typical scenario for academic federations.

In this proposal, the second factor's registration is done through a self-service provided by MFaP (See Fig. 2). From the user's perspective, the MFaP Dashboard is a service provider as any other service provider who delegates the authentication to IdP. Thus, an user has to use his first factors to access the MFaP Dashboard if this user does not yet have a second factor associated with his/her account. Otherwise, the user should use the first and second factor to have access to the dashboard.

The registration of the TOTP is based on a QR code – presented on the MFaP web page – scanned by the user containing the private key that will act as seed for generating authentication codes. After that, the user needs to type in the generated code to confirm the registration process with MFaP. For the Phone Prompt registration process the user selects Phone Prompt as the second factor and the MFaP dashboard provides a QR Code, which contains amongst other information the MFaP's application key on FCM service. The user then reads[2] the QR Code using the AMPToApp and it sends a message to the MFaP using FCM. After that, both MFaP and the AMPToApp show a successful registration message. For the FIDO2 WebAuthN registration, the user selects FIDO2 as the second factor, inserts the FIDO2 token in the USB port and click the registration button on the Web page. Once the token starts flashing, the user has to touch the physical button on the token. Employing the FIDO2 protocol and the standard Web API, the MFaP then exchange cryptographic keys with the USB device and registers it in its database.

If a malicious entity has access to the first factor of its victim and this victim had not yet registered a second factor to his/her account, it could explore this self-service to associate a second factor to the victim account and the latter will not be able to access his/her account anymore. In that case, the victim will have to contact IT support of his/her institution and ask them to remove the second factor associated with his/her account and then the user should redefine his/her password.

MFaP allows users to register more than one technology as his/her extra authentication factor and the user can choose which will be the default technology in the authentication process. The other technologies will act as backup or convenience options that can be selected by the user at anytime. In the example of Fig. 2, the user has three enabled technologies where the Phone Prompt is the default technology and it can be observed because of the filled bookmark icon in the Phone Prompt card and also the left green border. If the user wants to use the FIDO2 as the default, then, he/she has to click on the FIDO2 bookmark icon. Backup codes can not be the default technology.

The recovery process aims to give back the access to the user who lost his/her credentials. If the user has only a first factor enabled, then it is possible to send to him/her a link by email where the user will be able to reset and create a new password. However, if the user loses access to the second factor, this procedure is not suitable. MFaP follows NIST recommendation [29] for such situation, where the user should register more than one option to acts as the second factor, thus, the user will have a backup option for the

[2] The user has 1 min to read the code before it expires.

2FA process. In case the user does not have any factor, he/she will be locked out from the account and need to contact IdP's operator for manual recovery procedure.

The replacement of a second factor could be motivated by the acquisition of a new mobile phone. In the current proposal, in order to replace the current second factor, the user should use the same self service that he/she used to associate a second factor to the account. In that case, the user should use the first factor and the current second factor to gain access to the MFaP Dashboard and then, he/she is able to revoke the current second factor (see trash can icon in Fig. 2) and associate a new one to the account.

Finally, the current solution also offers a command line interface where the MFaP's operator is able to remove the second factor associated with any user account.

5 Evaluation

In order to evaluate MFaP we have performed a series of experiments, looking at different aspects of our solution. All experiments were performed using the GidLab testbed[3] [30], where a set of identity and service providers were deployed over the GidLab infrastructure. Our solution has been deployed and validated on the following environments: Ubuntu Linux 16.04 LTS, Shibboleth IdP version 3.3.1 and Shibboleth SP version 2.6.0.

We start by considering an IdP operator point of view in deploying MFaP into an existing IdP. In the sequence we discuss about a series of functional tests, followed by a security analysis on the different factors supported by MFaP.

5.1 Deploying MFaP into existing IdPs

This section presents a brief discussion on the deployment of MFaP into an existing IdP. This has been done to evaluate the impact of employing MFaP into the activities associated with the management of an IdP.

The Brazilian National Research and Education Network (RNP - *Rede Nacional de Ensino e Pesquisa*) is the entity who manages and supports the Brazilian Federated Academic Community (CAFe - *Comunidade Acadêmica Federada*). RNP offers a set of procedures, scripts and an IdP template – a virtual machine image – to help institutions to be part of CAFe.

In this work, we have used as basis the RNP IdP template to developed a set of Python scripts for helping with the MFaP installation process. Thus, to install MFaP into a computer where the IdP is running, the IdP administrator should execute our Python script, answer a few questions and the script will download MFaP code from RNP git server, generate a set of configurations files and deploy the MFaP application on local Apache Tomcat server. After that, the MFaP will be completely integrated with Shibboleth IdP.

The same Python script could be used to get and apply MFaP code updates. To update MFaP code, the IdP administrator should execute the script and at this time he/she does not need to answer more questions because the configurations files were already generated during the MFaP installation.

The initial assessment of the installation script has been performed through a series of experiments using the GidLab infrastructure. Those allowed us to identify and fix some issues with the IdP template provided by the RNP and with our installation script. Such

[3] GidLab is an experimental testbed for identity management solutions maintained by RNP.

issues were related to the way the IdP image was built by the RNP and the way the installation script automated the process. For instance, a self-signed certificate was available in that image, requiring that the curl command included the necessary flag to ignore that. Also, the communication between the IdP and the MFaP is over TLS, requiring an extra step of adding that certificate to the Java Virtual Machine trust store, to be reliable by the application. Another issue was related to name resolution. The IdP was not able to make requests to its own localhost address, to communicate with the MFaP that was running in the same machine. This was because the file was configured in the image to use 127.0.1.1 resolving the hostname. And according to Debian documentation for a permanent IP address system, a permanent IP address should be used instead of 127.0.1.1.

To certify that the installation script works properly, we had run a pilot phase involving three Brazilian Universities (UFBA - Federal University of Bahia, UFPE - Federal University of Pernambuco and UFRGS -Federal University of Rio Grande do Sul) and the RNP. Each IdP administrator employed our installation script to install and update MFaP. All four IdPs were successfully integrated with MFaP and no impact has been noticed on the other systems maintained by the institutions that rely on the IdP for authentication. As a result of this experiment we collected relevant feedback that helped us to improve and to fix a minor problem in our installation script.

5.2 Functional testing

The functional tests were performed in two phases: a set of tests conducted by the development team; and some preliminary trials involving deployed pilots at one Brazilian University and at RNP.

The tests conducted by the development team explored the GidLab testbed. We have created a test plan to evaluate all 2FA technologies implemented by MFaP and all aspects of the second factor life-cycle management for each technology. The test plan contains a total of 38 functional tests of the complete solution, and allowed us to identify and perform some improvements to the final solution before it is made available. Those ranged from visualization of the authentication factors under different configurations of registered factors, the different activities related to life-cycle management, and the actual authentication using the four factors currently implemented.

In our current implementation the mobile app (AMPToApp) can act as a second factor authentication for more than one account in a single or multiple IdPs. A set of tests were performed using a mobile device with a single app and it was possible to successfully validate the expected behavior considering multiple accounts in one or more IdPs. In another set of tests, three mobile devices were used simultaneously where it was possible to verify the solution behavior when there were concurrent second factor registration requests and concurrent 2FA processes. The results of the experiments in both sets were as expected.

The tests involving the three 2FA technologies (TOTP, Phone Prompt, and FIDO2) considered the successful registration of one or more devices in the same or different user accounts, and simultaneous registrations by different users and devices, for example, three users from the same IdP, using three different smartphones or FIDO2 keys, simultaneously register the second factor successfully. There were also tests involving timeouts in both registration and authentication, and attempts of reusing generated authentication credentials. All the results from the test cases turned out as expected.

We have also performed some experiments to evaluate the SP behavior when handling 2FA in four different scenarios: (1) SP requires second factor authentication and user has registered a second factor at his/her IdP; (2) SP requires second factor authentication and user has not registered a second factor at his/her IdP; (3) SP does not require second factor authentication and user has registered a second factor at his/her IdP; (4) SP does not require second factor authentication and user has not registered a second factor at his/her IdP. Notice that scenario #4 corresponds to the traditional federation authentication where both SP and IdP only deals with a single authentication factor. Nevertheless, we have included this scenario in order to test whether the inclusion of 2FA with MFaP would affect the normal operation of IdPs. All scenarios were performed varying the 2FA technology being employed when suitable to do so. For example, scenario #2 does not require a second factor by the user, which receives an error message explaining that the SP requires a second factor authentication. The results of the experiments were as expected.

In the second experimental phase, we have prepared a pilot plan to evaluate the integration of the MFaP in the IdPs (certification and testing environment) from the UFBA and the RNP. After conducting the tasks of the plan, a preliminary satisfaction evaluation has been performed with the participants. Due to the sensitive nature of the authentication service in the institution, these tasks were performed by the IdP operators and UFBA technical personnel, which restricted the study to only seven participants. Nevertheless, given their experience with day-to-day operations of the IdP and user support, we believe that they were able to provide valuable insight on how to improve our solution.

When asked how often they authenticated themselves using a second factor, 57% responded that they only did it to experiment with the technology while the remaining 43% used it on a daily basis. We have also asked their opinion on the ease of use for each second factor technology, with the results presented in Fig. 6.

We can notice a strong indication about the ease of the TOTP usage, with all responses being between normal and very easy. Phone prompt has also been well received by the participants, with only one of them answering that it was hard to use. On the other hand,

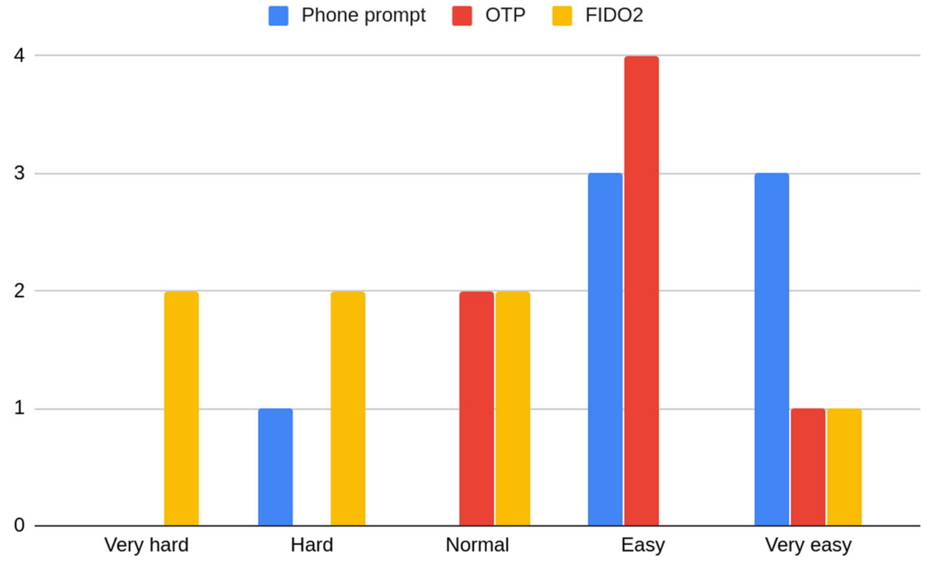


Fig. 6 Results of the easy of use question from the satisfaction survey conducted with technicians of UFBA

we can notice that the use of the FIDO2 USB stick is considered as "not easy" by almost all participants.

When asked about problems found during authentication, 85% of participants indicated that they faced some type of error. As for the reasons, five participants indicated problems with the FIDO2 USB stick, with one answer about errors in the TOTP application (Google Authenticator) in an iOS device, but a successful authentication using the freeOTP application, and one with an error in the generated QRCode image.

All received feedback have been taken on-board by the development team resulting in several improvements to our solution. In particular FIDO2, the feedback allowed us to identify several improvements regarding its documentation, mainly considering that this is a new technology that the majority of users are not familiar with. Another comment provided some improvements to the Web dashboard interface for managing the different factors. Concerning the problems with TOTP, we have identified that the Google Authenticator for iOS did not work properly if the optional parameter "algorithm" was added to the URI. Removing that parameter solved the issue. In general, the pilot results indicate that the established requirements (easy MFaP deployment and code updating) were met, while the satisfaction evaluation results were helpful to improve the usability of the MFaP dashboard and the mobile app.

5.3 Security considerations

This section presents a discussion about some security considerations on MFaP. Four different types of technologies were proposed to be used as an extra authentication factor for the users' IdP. Thus we detail possible attacks on each technology and to the whole solution, that includes the interactions between Shibboleth IdP and MFaP.

We start by considering the MFaP application. The REST API provided by MFaP requires a client authentication and it only relies on HTTPS protocol. Moreover, MFaP is a Java application that can be deployed in the same application server where Shibboleth IdP is already deployed (recommended option). As the MFaP REST API is consumed only by the IdP, it can be limited to the localhost interface to reduce the attack surface. In that case, a remote attacker will not be able to capture any packet exchanged between the MFaP and the IdP.

In the sequence we present some consideration for each of the implemented 2FA technologies. For all scenarios in the next subsections, we assumed an attacker who already compromised the victim's first factor, that is, the attacker already has the user's username and password.

5.3.1 Phone prompt

An attacker will only be able to personify a victim if the attacker has access to the victim's device, where the Phone Prompt application is enabled.

Considering that a victim is not trying to get an authentication at an IdP, an attacker, after a successful first authentication, is able to send authentication request notifications to the victim's device with the intention to make the victim confirm the second authentication step and consequently have access to the MFaP dashboard.

If the distracted victim clicks the "Yes" button, the attacker will then gain access to the victim account and could access the MFaP Dashboard to disable the victim's second factor authentication or even to register a new device to act as a second factor. If the attacker successfully remove the second factor, the victim will lose access to the account and will receive a notification informing about the second factor removal. Therefore, the victim can quickly become aware that the account was compromised and can take the proper measures to restore the authentication factors.

It is important to highlight that if the user did not initiate a 2FA authentication, he/she should always click the "No" button to avoid this type of attack.

Considering that the aware victim is under attack, he/she can use the Phone prompt application feature to block subsequent notifications for one hour. Meanwhile, the user or the administrator has time to change the compromised password for a new one.

MFaP uses FCM to send notifications to the Phone Prompt application (see Subsection 3.2). All communications between MFAP and FCM, or between FCM and the user's smartphone, rely on the Transport Layer Security (TLS) protocol [31]. Thus, confidentiality and integrity properties are granted by the TLS.

When the user enters his/her username and password, MFaP sends a notification through FCM to the user's device and the IdP login web page shows the message "*waiting... continue on your phone*" (see Fig. 4). This message will be there until the user's device sends a response message through the FCM to the MFaP or until the time expires (one minute). In this case, the user should try again with Phone prompt or with a different technology (i.e. TOTP, backup codes, etc).

Each message sent by the MFaP has a nonce, the category (e.g., register, unregister or authentication) and a code (e.g., 200 OK, 302 FAIL, etc). The message sent by the user's device should include the nonce received from the MFaP, otherwise the MFaP will discard it. All messages sent by the MFaP have a collapse key that identifies a group of messages that can be collapsed by the FCM queue service. That is, the FCM will send only the last message with a given collapse key (if the user's device is not connected) to user's device. The maximum message lifespan was set to one minute, after that, the message expires and the FCM will not delivery it to the user's device or to the MFaP.

If the attacker starts several authentication requests in parallel, the MFaP will send just one message for the first authentication request and the other authentication requests will not generate new notifications on the victim's device. That is, even under attack, the MFaP will send at most one message per minute to the victim's device. This behaviour protects the whole solution from a potential flood attack. However, the victim might not be able to use Phone Prompt if the attack takes a long period. If the victim uses the backup code, which is always available in the MFaP, he/she will not be subject to the denial of service attack.

5.3.2 Time-based one-time password

According to security considerations from RFC 6238 [11], all communications should take place over a secure channel (i.e. TLS), the shared key should be chosen using a strong pseudo-random generator and a time-step size of 30 s. In our implementation, we followed all of these recommendations. The IdP's login web page, the place where the user has to provide the one-time-password, relies on HTTPS.

To bypass the TOTP, a remote attacker has to capture the shared key used by the MFaP and the user. During the TOTP registration process, the MFaP shows a web page that contains a QR Code image representing a shared key for the TOTP. The user has to use a TOTP application on his/her smartphone to read this QR Code. The shared key could be leaked to a remote attacker only if he/she previously implanted a malware on the victim's phone. Thus, this malware should be able to read the QR Code image present on the MFaP registration web page. The possibility for this kind of attack is remote, but plausible.

5.3.3 FIDO2

According to FIDO2 specifications [12], the user's credentials are unique for each user account and across different IdPs. They never leave the user's device and are never stored on an IdP or MFaP. Thus, the same FIDO2 device could be used by a user in different accounts, on the same or different IdPs, and it will not leak any information that can be used by the IdP to track him/her. A remote attacker is not able to capture the user's credentials and the user presence test [12] makes malware attacks unpractical on the user's device.

5.3.4 Backup codes

Each backup code can be used just once and it is provided on the IdP authentication web page, that relies on HTTPS. On our current implementation, backup codes are 8 digit numbers, generated using the same algorithm that is used to generate TOTP codes, and the IdP web page limits the number of attempts and also introduces a delay between subsequent attempts to reduce the effectiveness of a force brute attack. Thus, the main risks associated with this technology is a malware installed on the user's computer that can have access to the backup codes file, or a careless user who may lose or delete this file.

6 Conclusion

This article presented a multi-factor authentication provider (MFaP) for Shibboleth Identity Providers. The MFaP is based on SAML MFA Profile to extend the IdP authentication flows and it can be easily integrated to existing IdPs. To the best of our knowledge, our solution is the first comprehensive open source solution that does not rely on third-party services. Our approach supports the use of different technologies as second authentication factors, allowing each user to set single or multi authentication factors, and which technologies can be used as extra factors.

MFaP comprises a set of components for managing authentication factors, a Web-based GUI allowing the management of 2FA technologies by IdP operators, and a self-service portal for end users. We have also developed a mobile app (AMPToAPP) that can be immediately used by any institution who wishes to deploy MFaP into its IdP.

MFaP has been evaluated by means of a comprehensive test plan and pilots using IdP operators from different Brazilian institutions. These included not only the normal operation of the solution, but also its installation, maintenance and update. Furthermore, we have performed a security analysis of the solution.

Although we have achieved significant results with MFaP, some limitations have been identified. In its current implementation, we restricted the FIDO2 USB token to one per user, per IdP. This still allows a user to reuse the same USB token in different IdPs, but does not allow scenarios where a user might have two or more tokens in the same IdP, for example, a backup token as means of recovery. The pilot experiments provided relevant feedback, but all participant users had technical knowledge, not representing the majority of IdP users.

We are currently planning a new deployment of our solution with a larger number of active users. In fact, we envision the realization of a more comprehensive set of usability experiments, exploring a population with diverse levels of IT knowledge, age range, etc. We also plan on implementing and experimenting with other FIDO2 specifications, such as the use of other FIDO2 devices as second authentication factor.

Finally, we also suggest an in-depth solution threat analysis considering the Shibboleth IdP integrated with the MFaP, the dashboard, and mobile apps. An academic cybersecurity competition or challenge could be organized to identify some vulnerabilities in the proposed solution including the open-source software analysis.

Abbreviations

2FA: Two factor authentication; CAFe: Federated academic community; CTAP: Client toAuthentication protocol; FCM: Firebase cloud message; FIDO: Fast IDentity online; FIM: Federated identity management; IdP: Identity provider; LDAP: Lightweight directory access protocol; MFA: Multi-factor authentication; MFaP: Multi-factor provider; NIST: National Institute of Standards and Technology; OASIS: Organization for the advancement of structured information standards; OTP: One-time password; REFEDS: Research and education FEDerations group; REST: Representational state transfer; RNP: Brazilian national research and education network; RP: Relying part; SAML: Security assertion Markup Language; SMS: Short message service; SP: Service provider; SSO: Single sign-on; TLS: Transport layer security; TOTP: Time-based one-time password; U2F: Universal 2nd factor; UAF: Universal authentication framework; UFBA: Federal University of Bahia; UFPE: Federal University of Pernambuco; UFRGS: Federal University of Rio Grande do Sul; W3C: World wide web consortium; WebAuthN: Web authentication

Acknowledgements

We would like to thank the R&D RNP team for their various contributions to develop the GT-Ampto project. We would like to thank the universities UFBA, UFPE and UFRGS involved during the pilot phase. We would like to thank the GidLab for support to create and deploy the testbed necessary to the development of this work.

Authors' contributions

Mainly, ERM, MSW and CES were responsible for drafting and writing the manuscript. ERM focused on the main conception and the writing of the Multi-Factor Authentication, Life-cyle management and Security Considerations sections. MSW focused on the Introduction, Related Work and Conclusion sections. CES focused on reviewing the whole article and writing FIDO2 and Functional Testing sections. SBL, GCS, SAC and BBL were responsible for the majority of the technical work, implementing the solution and conducting experiments. SBL focused on developing the core of MFaP and of the whole solution. GCS focused on developing FIDO2. SAC focused on developing TOTP, Backup Codes and MFaP Dashboard and the writing of the TOTP section. BBL focused on developing mobile applications for Android and iOS. All authors reviewed and approved the manuscript.

Authors' information

Emerson Ribeiro de Mello is a professor at the Federal Institute of Santa Catarina (IFSC), Brazil (since 2007). He received a PhD in Electrical Engineering in 2009 from the Federal University of Santa Catarina. He also received an MS degree in Electrical Engineering in 2003 from the Federal University of Santa Catarina and a BS degree in Computer Science from the Unoeste University in Brazil (2000). From December 2011 until December 2015 he was the IT Director of the Federal Institute of Santa Catarina. In 2016 he was a visiting researcher at the School of Computing Science, Newcastle University, UK. His research interests include Reputation Systems, Distributed Systems, Security, Trust Models and Identity Management.

Michelle Silva Wangham is a full professor at University of Vale do Itajaí (Brazil). She received her M.Sc. and Ph.D. on Electrical Engineering from the Federal University of Santa Catarina (UFSC) in 2004. She was a Visiting Researcher at University of Ottawa in 2015/2016. Her research interests are vehicular networks, security in embedded and distributed systems, identity management, and network security. She is a consultant for the Brazilian National Research and Education Network (RNP) acting as the coordinator of Identity Management Technical Committee (CT-GID) and a member of the Network Monitoring Technical Committee. Since 2013, she has been coordinating the GIdLab project, a testbed for R&D in Identity Management.

Samuel Bristot Loli is a software developer at Federal Institute of Santa Catarina (Brazil). Currently, he is a M.Sc. student in Computer Science at Federal University of Pernambuco (UFPE). His research interests are software architecture, refactoring, distributed systems, integration and object-relational mapping.

Carlos Eduardo da Silva is a Senior Lecturer at Sheffield Hallam University, UK. Previously a professor at Metropole Digital Institute (IMD) of the Federal University of Rio Grande do Norte (UFRN), Brazil (2014-2019). He received a PhD in Computer Science from the University of Kent at Canterbury, UK (2011), a MSc (2007) and a BSc in Computer Science (2005), from UFRN. He was on a sabatical visit to Department of Computer Science at the University of York, UK during 2016, working on the topics of Information Security, and Self-adaptive Systems. His current research interests include the interplay between Software Engineering and Information Security in the domains of Self-adaptive Systems, Identity Management and Access Control, Business Process Management, Cloud Computing and IoT.

Gabriela Cavalcante da Silva is graduated in Information Technology at the Federal University of Rio Grande do Norte (UFRN) with Sandwich Degree (2015-2016) fulled funded in France (Polytech Nice Sophia) by the exchange program BRAFITEC. Undergraduate in progress in Computer Science at UFRN. Areas of interest include Self-adaptive Systems, Information Security, IoT and Data Science.

Shirlei Aparecida de Chaves holds a BSc Degree in Information Systems (2005) and a M.Sc Degree on Computer Science(2010), both from the Federal University of Santa Catarina. She has more than 18 years of experience in the IT field, specially with deployment and support of distributed systems. Since 2014 she works as an IT Analyst at the Federal Institute of Santa Catarina. In addition to her work as an IT Analyst, she works as a research assistant on identity management projects and assists the community in implementing and using the Helios online voting system. Her research interests include distributed systems, privacy and security.

Bruno Bristot Loli is a software developer from Criciuma, Brazil, mainly interested in web and mobile development. Bruno has worked as a developer at Thomson Reuters, and currently works at CIASC, a state owned company located in Florianopolis.

Funding

This work was result of GT-AMPTo project that had been supported by the Brazilian National Research and Education Network (RNP), inside of its R&D working group program.

Availability of data and materials

Data sharing is not applicable for this article as no dataset was generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Federal Institute of Santa Catarina (IFSC), São José, SC, Brazil. [2]University of Vale do Itajaí (UNIVALI), São José, SC, Brazil. [3]Sheffield Hallam University (SHU), Sheffield, UK. [4]Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil.

Received: 8 November 2019 Accepted: 25 September 2020
Published online: 02 December 2020

References

1. NIST. Digital Authentication Guideline: NIST Special Publication 800-63-3; 2017. https://doi.org/10.6028/NIST.SP.800-63-3.
2. Brainard J, Juels A, Rivest RL, Szydlo M, Yung M. Fourth-Factor Authentication: Somebody You Know. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06. New York: Association for Computing Machinery; 2006. p. 168–78. https://doi.org/10.1145/1180405.1180427.
3. Dasgupta D, Roy A, Nag A. Multi-Factor Authentication. Cham: Springer; 2017, pp. 185–233. https://doi.org/10.1007/978-3-319-58808-7_5.
4. Bhargav-Spantzel A, Squicciarini AC, Xue R, Bertino E. Multifactor identity verification using aggregated proof of knowledge. IEEE Trans Syst Man Cybern Part C (Appl Rev). 2010;40(4):372–83. https://doi.org/10.1109/TSMCC.2010.2045755.
5. Aloul F, Zahidi S, El-Hajj W. Multi factor authentication using mobile phones. Int J Math Comput Sci. 2009;4(2):65–80.
6. Arias-Cabarcos P, Almenárez F, Trapero R, Díaz-Sánchez D, Marín A. Blended identity: Pervasive idm for continuous authentication. IEEE Secur Priv. 2015;13(3):32–39. https://doi.org/10.1109/MSP.2015.62.
7. Cantor S, Kemp J, Philpott R, Maler E. Assertions and Protocolos for the OASIS Security Assertion Markup Language (SAML) v2.0. OASIS; 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.
8. Jensen J. Federated identity management challenges. In: Availability, Reliability and Security (ARES), 2012 Seventh International Conference On. IEEE; 2012. p. 230–5. https://doi.org/10.1109/ares.2012.68.
9. Shibboleth Consortium. https://www.shibboleth.net. Accessed 28 Oct 2019.
10. Refeds. REFEDS MFA Profile. 2017. https://refeds.org/profile/mfa. Accessed 28 Oct 2019.
11. M'Raihi D, Machani S, Pei M, Rydell J. TOTP: Time-Based One-Time Password Algorithm: RFC Editor; 2011. http://www.rfc-editor.org/rfc/rfc6238.txt.
12. W3C. Web Authentication: An API for accessing Public Key Credentials Level 1. 2019. https://www.w3.org/TR/webauthn. Accessed 28 Oct 2019.
13. Weiser M. The computer for the 21st century. Sci Am. 1991;265(3):94–104.
14. Google. Making Google prompt the primary choice for 2-Step Verification. Google Off Blog. 2017. https://gsuiteupdates.googleblog.com/2017/10/makinggoogle-prompt-primary-choice-for-2sv.html. Accessed 28 Oct 2019.
15. SecureAuth Documentation. https://docs.secureauth.com. Accessed 28 Oct 2019.
16. Secure Two-Factor Authentication App. https://duo.com/product/trusted-users/two-factor-authentication/duo-mobile. Accessed 28 Oct 2019.
17. Zhang N, Yao L, Chin J, Shi Q, Nenadic A, McNab A, Rector A, Goble C. Plugging a scalable authentication framework into shibboleth. In: 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05); 2005. p. 271–6. https://doi.org/10.1109/WETICE.2005.48.
18. Langenberg D. Multi Context Broker. 2015. https://spaces.internet2.edu/display/InCAssurance/Multi-Context+Broker. Accessed 28 Oct 2019.
19. Cantor S. Authentication Flow Selection. 2015. https://wiki.shibboleth.net/confluence/display/IDP30/AuthenticationFlowSelection#AuthenticationFlowSelection-FlowSelection. Accessed 28 Oct 2019.
20. Morii M, Tanioka H, Ohira K, Sano M, Seki Y, Matsuura K, Ueta T. Research on integrated authentication using passwordless authentication method. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), vol. 1; 2017. p. 682–5. https://doi.org/10.1109/COMPSAC.2017.198.
21. Cantor S, Scavo T. Shibboleth architecture: Protocols and Profiles. Protoc Profiles. 2005;10:1–19.
22. Joie CL. Authentication. 2017. https://wiki.shibboleth.net/confluence/display/IDP30/Authentication. Accessed 28 Oct 2019.
23. Google Authenticator Server Side Code. https://github.com/wstrange/GoogleAuth. Accessed 28 Oct 2019.
24. Firebase Cloud Messaging. https://firebase.google.com/products/cloud-messaging. Accessed 28 Oct 2019.
25. Srinivas S, Balfanz D, Tiffany E, Czeski A. Universal 2nd Factor (U2F) Overview. 2017. https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.pdf.

26. Machani S, Philpott R, Srinivas S, Kemp J, Hodges J. FIDO UAF architectural Overview. 2017. https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/fido-uaf-overview-v1.1-ps-20170202.pdf.
27. Lindemann R, Brand C, Czeskis A, Jones MB, Hodges J, Kumar A, Powers A, Verrept J, Ehrensvärd J. Client to Authenticator Protocol "(CTAP)". 2019. https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fidoclient-to-authenticator-protocol-v2.0-ps-20190130.pdf.
28. Yubico Webauthn Server Demo. https://developers.yubico.com/java-webauthn-server/webauthn-server-demo. Accessed 28 Oct 2019.
29. Grassi PA, Fenton JL, Newton EM, Perlner RA, Regenscheid AR, Burr WE, Richer JP, Lefkovitz NB, Danker JM, Choong YY, et al. Nist special publication 800-63b. digital identity guidelines: Authentication and lifecycle management. Bericht NIST. 2017. https://doi.org/10.6028/NIST.SP.800-63b.
30. GidLab Testbed. https://gidlab.rnp.br. Accessed 28 Oct 2019.
31. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3: RFC Editor; 2018. https://tools.ietf.org/html/rfc8446.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.