Two-Factor Authentication @ IfI (2FA)

If you are here for the first time, you probably want to setup your first 2nd factor. To start, we suggest you go with the *PUSH token* or a *TOTP token*. You find detailed instruction how to set them up below.

If you feel completely overwhelmed or do not understand a word, we suggest to to dive in first with enrolling a PUSH or a TOTP token (scroll down for the step-by-step guides) and read then read the rest of this document afterwards. Chances are good, that things are much clearer then :)

The **main goal of 2-factor authentication** (2FA) is to **mitigate the impact of successful phishing and malware attacks**, whereby attackers might get hold of your if credentials (username + password). Without 2FA, such an attacker could use IfI account credentials to impersonate you, read/delete all your data, etc. With 2FA, there is a **second line of defense**, namely the 2nd factor which needs to be entered (besides username + password), which is much more difficult to obtain for a remote attacker.

2-factor authentication (2FA) and single sign-on (SSO)

When loggin in to services that are 2FA-enabled you will be **redirected to our keycloak server ⇒ kc.ifi.uzh.ch**.

Here you enter your IfI credentials and the 2nd factor. **Before logging in, please double-check the URL**. If you have any suspicions that you are not on the correct page for logging in, please do not enter your passwors / 2nd factor and let us know.

Check the screenshot (to the right) to see how the **SSO login** on the keycloak server looks like.

Best Practices

With the above goal in mind, we suggest the following best practices:

- Use a password manager / key chain to store your IfI credentials (username + password) but **avoid using the password manager to store the 2nd factor** too. Why? Because if an attacker gets hold on your password manager, **all is lost**.
- Use your **mobile phone** to provide the second factor (see PUSH and TOTP tokens below).
- To avoid, that your mobile phone becomes a **single point of faillure** (i.e. no way to login in

case the phone is lost or has no power), use a **hardware token such as a yubikey**. you can easily put it on your (physical) key chain, so you always have it with you. You can use TOTP tokens or passkey with such a yubikey.
- **Always protect your 2nd factor**:
  - **Keep your mobile phone locked**, and use a **fingerprint** or **PIN** to unlock it (pattern unlock is not of much use, i.e. easily spotted and broken).
  - **Keep your hardware token / yubikey in your pocket**. If you just keep it plugged to your laptop, anyone can use it who gets hold of your laptop. If you absolutely want to keep the yubikey plugged in, buy a (more expensive) **yubikey with fingerprint authentication** and/or use **PIN protection** for the stored passkeys.
  - If you absolutely must store your 2nd factor in your password manager, make sure that you have **protected your password manager** (auto lock, strong (secret) passphrase to unlock it). And also make sure that your laptop is always locked (screen lock) when you leave it unsupervised (i.e. if you leave the office shortly, etc.). Please note: **these two things are absolutely essential anyhow**.
  - **Enable locking for the authenticator app** (PrivacyIDEA / Google / MS authenticator) holding e.g. PUSH and TOTP tokens, such that even if someone gains access to your phone cannot use it without **fingerprint** or **PIN** (in addition to locking your mobile phone).

Overview of Token Types

Token type	Application / Hardware Token	Devices	Security notes (see also Best Practices)
PUSH	PrivacyIDEA authenticator	mobile phone	enable locking
OTP	PrivacyIDEA / MS / Google / etc. authenticator	mobile phone / laptop	enable locking
Passkey	stored on Yubikey	yubikey	enable PIN
Passkey	stored in your password manager	mobile phone / laptop	enable locking
yubico OTP	stored on Yubikey	yubikey	remove Yubikey / Yubikey with fingerprint

Notes:

- enable locking ⇒ make sure that the authenticator app / password manager is locked (automatically) and needs a fingerprint or a PIN before usage.
- remove Yubikey ⇒ Yubikeys without fingerprint or PIN protection need to be removed from the laptop when not used and stored in a secure location, i.e. together with your other keys in your pocket. Alternatively use Yubikey with fingerprint sensor.

Enrolling and Using 2nd Factors / Tokens

PrivacyIDEA Authenticator

The PrivacyIDEA Authenticator is an app for Android or IOS mobile phones. Besides the standard TOTP (6-digit one-time password) token it also provides the PUSH token, which is a very unobtrusive way to provide the 2nd factor. You only need to confirm the login on your phone and authenticate that with

your fiingerprint.

PUSH token

To setup the PUSH token, do the following:

1. Open pi.ifi.uzh.ch, click on *Enroll Token*. The *PUSH* token should be selected by default.
2. Install the *PrivacyIDEA Authenticator* on your phone. You can use the QR codes provided to do that.
3. Add a *Description* (to distinguish your tokens easily) and click *Enroll Token*.
4. Scan the new QR code with the *PrivacyIDEA Authenticator* app.

Your PUSH token is now ready to be used.

We strongly recommend to protect/lock your tokens in the PrivacyIDEA Authenticator app. To do so, swipe right on the token and choose lock.

TOTP (time-based one-time password) token

This is a very common token type which you most probably already use somewhere. You can use it with any Authenticator App, such as Microsoft Authenticator, Google Authenticator, FreeOTP, etc. If you use or plan to use the PUSH token (see above), you can use the PrivacyIDEA Authenticator App.

For all Authenticator apps the procedure is basically the same:

1. Open pi.ifi.uzh.ch, click on *Enroll Token* and choos the *TOTP* token.
2. If necessary, install the *PrivacyIDEA Authenticator* on your phone. You can use the QR codes provided to do that.
3. Add a *Description* (to distinguish your tokens easily) and click *Enroll Token*.
4. Scan the new QR code with the *PrivacyIDEA Authenticator* app (or the Authenticator app of your choosing).

Your TOTP token is now ready to be used.

We strongly recommend to protect/lock your tokens in the PrivacyIDEA Authenticator app. To do so, swipe right on the token and choose lock. If you use another Authenticator app, please also make sure that is used some kind of authentication (fingerprint) before usage.

Using yubico-5 hardware tokens

Please check also the official website and setup guide. You can store various kinds of tokens on the Yubikey. We decribe **Passkeys** and **Yubikey OTP** tokens here. To work with the yubikey, you will need the **Yubico Authenticator**, are available at the yubico website.

Setting up Passkeys on the Yubikey

You can work with passkeys in several ways. One possibility is to store them on the Yubikey, which is much better than to store them in your password manager (or any other software client) on your laptop.

1. Insert the Yubikey, start the *YubiKey Authenticator* and go to *Passkeys*. Unlock the passkey store with your PIN, or set a PIN if you have not done that yet.
2. Open pi.ifi.uzh.ch, click on *Enroll Token* and choose the token type *Passkey* add a *Description* (to distinguish your tokens easily) and click *Enroll Token*.
3. Depending on your environment, your password manager might pop-up and ask to store the passkey. Do not store the passkey there, instead close/cancel that dialog. A new dialog box should pop up, asking you for the PIN of your Yubikey passkey store. Enter the pin and follow the instructions, i.e. press the button on the Yubikey.
4. Check the *YubiKey Authenticator* for the just saved passkey.

Your passkey is now ready to be used a 2nd factor.

Setting up a Yubico OTP token

You can configure the YubiKey to generate an one-time password (OTP) when simply pressing the button of the yubikey, and you can use this as a 2nd factor.

1. Insert the YubiKey and use to *YubiKey Authenticator* to setup a Yubico OTP and assign it to the button (short or long touch). Choose *Slots* ⇒ *Short/Long touch* ⇒ *Yubico OTP*.
2. Assign a *public ID* or simply use the *serial* (click the diamonds). Generate the *Private ID* and the *Secret key* (using the circular arrows).
3. Copy the *Secret key* and click *Save*.
4. Open pi.ifi.uzh.ch, click on *Enroll Token* and choose the token type *Yubikey AES mode*.
5. Paste the *Secret key* (copied from the YubiKey Authenticator) to the *OTP Key* field (PrivacyIDEA), add a *Description* (to distinguish your tokens easily) and click *Enroll Token*.

Your token is now ready to be used a 2nd factor. When asked for it, place the cursor in the field where you need to enter the 2nd factor and press the button on the YubiKey.

If you plan to leave the Yubikey plugged into your laptop, this token is not safe, since it can be issued simply by touch, and does not need a pin, fingerprint or any other form of authentication. Thus if someone gets hold of your laptop, that someone also has access to your 2nd factor. To be safe, use a Yubkey with fingerprint sensor (see also Overview).

Special Case: SSH and sudo

We enforce 2FA also for SSH and sudo. For technical reasons, it is not possible authenticate over keycloak and therefore there is no single sign-on (SSO) for SSH and sudo. This means, that you will have to present your 2nd factor each time you login (SSH) or use sudo. When using SSH and sudo a lot, this can become painful, so there are two mechanisms in place to mitigate this situation a bit:

- if you use **key-based authentication** with SSH, you will neither need to enter your password, nor a 2nd factor. Check e.g. here on how to setup key-based authentication. **Important: check the notes below**.
- sudo has a **cache time of 2hrs**, which means a sudo session will ask for your password and 2nd factor when you use it the first time, but then only again if the session was idle for 2hrs.

Always protect your ssh (private) key with a passphrase. **Never** use a ssh keys with an empty passphrase, as everybody that gets hold on your keys could use them.

Of course you want to avoid typing the passphrase each time you login. you can do that using **ssh-agent**. Here, you can unlock your (private) key once you login to your laptop/desktop and use it without passphrase (until you logout or shutdown your laptop/desktop).

To setup key-based authentication and use the ssh-agent, please check the following tutorials (if necessary):

- SSH Essentials: Working with SSH Servers, Clients, and Keys (digital ocean)
- Create SSH Keys with OpenSSH on macOS, Linux, or Windows (digital ocean)

The second tutorial also explains how to use the ssh-agent.