AUTH

User’s Manual

AH52126-UM01E_02

2022-11-02

AXELL CORPORATION

Trademarks

- Adobe, Acrobat, and Reader are trademarks or registered trademarks of Adobe Inc. in the United States and/or other countries.
- Arm® and CMSIS are registered trademarks of Arm Limited.
- CentOS, Red Hat® Enterprise Linux®, and Fedora® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.
- Facebook® is a registered trademark of Meta Platforms, Inc.
- Firefox is a registered trademark or trademark of the Mozilla Foundation in the United States and other countries.
- FIDO® is a trademark or registered trademark of FIDO Alliance, Inc.
- GitHub® is a registered trademark of GitHub, Inc.
- Google Chrome™ and Gmail™ are trademarks of Google LLC.
- Linux® is a registered trademark of Linus Torvalds in the United States and other countries.
- Mac, macOS, Safari, and Xcode are trademarks of Apple Inc., registered in the United States and other countries and regions.
- Microsoft Edge, Visual Studio, and Windows are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.
- Opera is a trademark or registered trademark of Opera Software ASA.
- RSA® is a registered trademark of RSA Security LLC.
- SHALO is a trademark or registered trademark of Axell Corporation.
- Ubuntu is a trademark or registered trademark of Canonical Ltd.

Information on software used with this product

Software owned by a third-party and licensed to Axell Corporation

Software (SHALO AUTH dedicated software) and data for the firmware updater that can be downloaded from “auth.shalo.jp”, and the firmware of SHALO AUTH dongle include Open Source Software (OSS). Open Source Software contained in this product and the respective Open Source Software licenses can be confirmed by using the following:

Launch GUI Tool, click [**Help**] > [**About**] > [**Acknowledgements**].

Contact information

If you need support for this product or have any inquiries about it, please email us at shalo@axell.co.jp.

Revision History

Document Number	Date	Page	Description
AH52126-UM01E_1.00	2022-02-24		Initial Release
AH52126-UM01E_1.01	2022-05-10	p.36, 40, 44 p.41 p.97 p.98-99 p.167	Add uninstalling software (Chapter 3) Add installing a prerequisite in Linux (Section 3.5.2) Add description of Acrobat® 64-bit (Section 7.2.1) Add description to deregister the PKCS #11 module from Acrobat® (Section 7.2.2) Add a symptom into troubleshooting by symptom (Section 11.5.5)
AH52126-UM01E_02	2022-11-02	p.1	Update Trademarks Add "Information on software used with this product"

Conventions used in this manual

Styles and formats

In addition to regular styles, this manual uses the following styles for special purposes:

Bold Indicates important information in running text and character strings provided by the user at the terminal.

Italic Indicates information that varies depending on the user's environment.

`mono` Indicates command names and command options.

The format below indicates the name and content of a file. You can see a line number at the left end of each line.

file.txt

```
1  This is a message.
```

The following format shows a single command line:

```
command -p environment-specific-string
```

The following format shows input and output in the terminal command prompt:

```
$ command↵
Message
```

Special characters used in the above prompt have the following meanings:

`>` Indicates the PowerShell prompt.

`$` Indicates the Bash prompt. It is used in Cygwin, Git for Windows, Linux, and macOS 10.14 Mojave or earlier.

`%` Indicates the Zsh prompt. It is used in macOS 10.15 Catalina or later.

↵ Indicates pressing the Enter key.

Symbols used in this manual

ℹ Indicates reference or supplemental information.

⚠ Indicates an important note.

How this manual is organized and how to read it

This manual contains 12 chapters. The following is an overview of each chapter:

Chapter 1 Provides a brief introduction to SHALO AUTH. It also covers the operating environment and general specifications.

Chapter 2 Explains information you need to be familiar with before using SHALO AUTH.

Chapter 3 Explains how to get started with SHALO AUTH and install dedicated software in each of the operating systems of Windows, macOS, and Linux.

Chapter 4 Explains how to use the SHALO Keyring key tool, one of the SHALO AUTH dedicated tools.

Chapter 5 Explains how to use the SHALO Smith administration tool, one of the SHALO AUTH' dedicated tools.

Chapter 6 Explains how to use SHALO AUTH for two-step verification in Web services from Google, Facebook, and GitHub.

Chapter 7 Explains how to use SHALO AUTH to secure PDF files.

Chapter 8 Explains how to authenticate users with SHALO AUTH through SSH authentication.

Chapter 9 Explains how to use SHALO AUTH through SSH authentication in GitHub.

Chapter 10 Provides convenient ways to make use of SHALO AUTH, such as how to enable a remote PC to use SHALO AUTH connected to your local PC.

Chapter 11 Contains frequently asked questions and solutions to them when SHALO AUTH is used.

Chapter 12 Provides developers with various specifications of the PKCS #11 module for SHALO AUTH.

You may not need to read all the chapters in this manual, depending on your purpose of using SHALO AUTH. To help you effectively learn how to use SHALO AUTH, the following shows how you should read this manual for different purposes.

- **Users who use SHALO AUTH for FIDO U2F**

  Read from Chapter 1 up to Section 2.2, and only if you use Linux, read Section 3.5.1. Then read Chapter 6, which explains how to configure two-step verification in Web services, as necessary.

- **Users who use SHALO AUTH with PDF files**

  Read from Chapter 1 to Chapter 4, except for Section 2.2. Then, read Chapter 7.

- **Users who use SHALO AUTH for SSH authentication**

  Read from Chapter 1 to Chapter 4, except for Section 2.2. Then, read Chapter 8.

- **Users who use SHALO AUTH for SSH authentication in Git**

  Read from Chapter 1 to Chapter 4, except for Section 2.2. Then, read Chapters 8 and 9.

Contents

Chapter 1

Introduction to SHALO AUTH

This chapter provides a brief introduction to SHALO AUTH.

Topics in this chapter

1. What is SHALO AUTH?
2. Applications
3. Operating environment
4. General specifications
5. Usage notes

1.1 What is SHALO AUTH?

SHALO AUTH (Figure 1) is a security key that can be connected via USB. It supports Windows, macOS, and Linux, and is available with OS-standard device drivers.

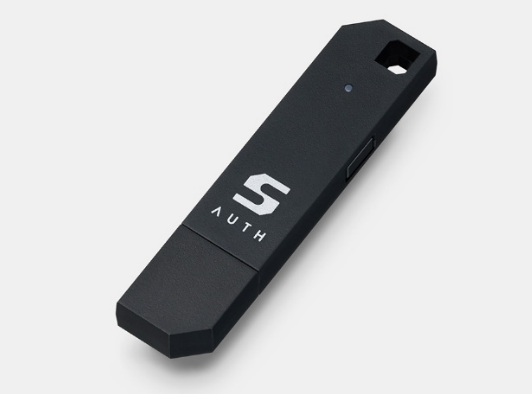


Figure 1 Appearance of SHALO AUTH

SHALO AUTH has the following two main features:

- FIDO U2F security key
- General security key

FIDO U2F security key

SHALO AUTH has been certified by FIDO as an authenticator of Authenticator Certification Level 2 (L2) for U2F certification. It can be used as a two-factor authentication security key in major Web browsers, such as Google Chrome, Safari, Microsoft Edge, and Firefox.

General security key

SHALO AUTH supports RSA and ECDSA public key cryptography as a general security key, and is available for managing keys securely and managing certificates; encrypting and decrypting; and issuing and verifying digital signatures.

RSA	ECDSA
Key length: 1,024 to 4,096 bits	Curves: P-192/P-224/P-256/P-384/ P-521/secp192k1/secp224k1/secp256k1

List 1: Supported public key cryptography

The features of the general security key are available through the PKCS #11 API, the Cryptographic Token Interface industrial standards. With the support of the PKCS #11 API, developers can use SHALO AUTH to build their own hardware authentication solutions.

An example includes PDF file security in Adobe® Acrobat® and Adobe® Acrobat® Reader®. These software systems support the PKCS #11 API, enabling you to use PDF files in ways such as:

- Encrypting PDF files and allowing users to browse them only with SHALO AUTH
- Signing PDF files electronically using SHALO AUTH

SHALO AUTH can also be applied for user authentication via SSH or for Git access, which many developers are familiar with. SSH is used to communicate securely with remote PCs and with virtual machines on cloud environments. By using SHALO AUTH for user authentication via SSH, you will have secure access without storing keys locally. As SSH is used as a secure communication infrastructure in the Git version control system and other systems, you can also take advantage of SHALO AUTH in these systems.

1.2 Applications

SHALO AUTH can be used mainly for:

- Two-factor authentication in Web services from Google, Facebook, and other vendors
- Viewing encrypted PDF files
- Two-factor authentication and SSH authentication in Git platforms, such as GitHub
- User authentication or a digital signature using PKCS #11-compliant software

Two-factor authentication in Web services

When using SHALO AUTH for two-factor authentication, the user is authenticated by entering the ID and password for a Web service and then pressing the button on SHALO AUTH.

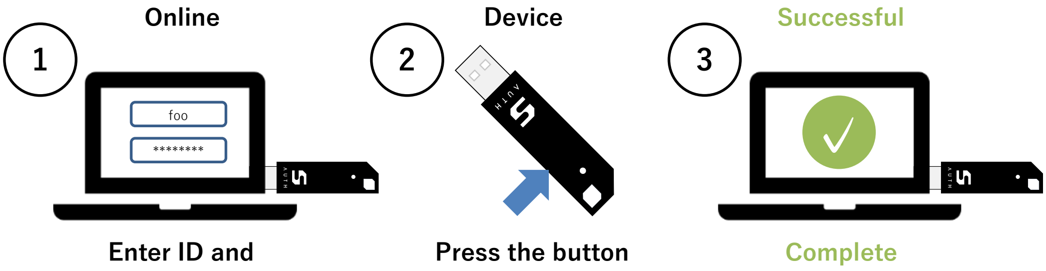


Figure 2 Two-factor authentication procedure

Viewing encrypted PDF files

With the security features of Adobe® Acrobat®, you can create a PDF file that can be viewed only in an environment with a particular SHALO AUTH device. This PDF file is encrypted for SHALO AUTH, which is responsible for decrypting the file when the user tries to view it in Adobe® Acrobat® or Adobe® Acrobat® Reader®.

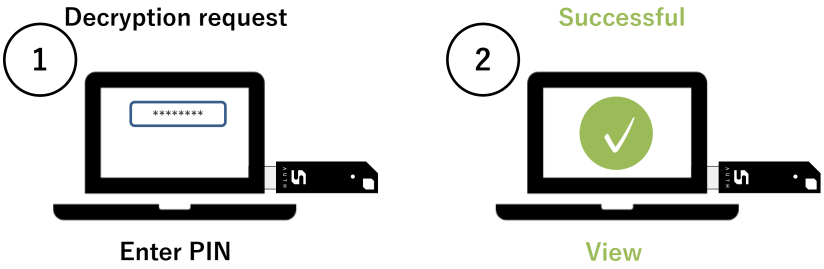


Figure 3 Procedure for viewing an encrypted PDF file

SSH authentication

When using SHALO AUTH for SSH authentication, the user is authenticated by entering a SHALO AUTH user PIN, not the remote PC password.

This user PIN is intended to cause SHALO AUTH to generate a digital signature for authentication. The digital signature will be generated only when the user who has SHALO AUTH enters the correct user PIN.

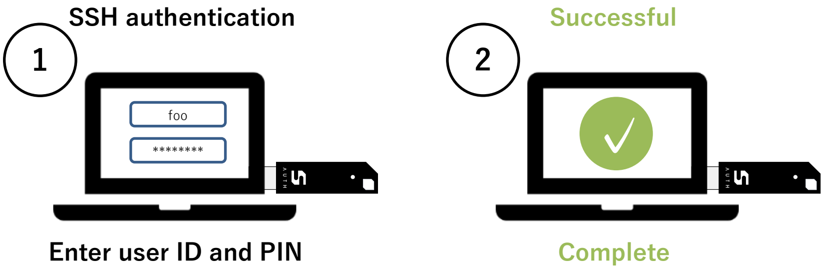


Figure 4 Procedure for PKCS #11 authentication via SSH

User authentication or digital signature using PKCS #11-compliant software

When using PKCS #11-compliant software, the user is authenticated or signs something digitally through SHALO AUTH by entering a SHALO AUTH user PIN in the software.

The operation will be processed only when a user who has SHALO AUTH enters the correct user PIN.

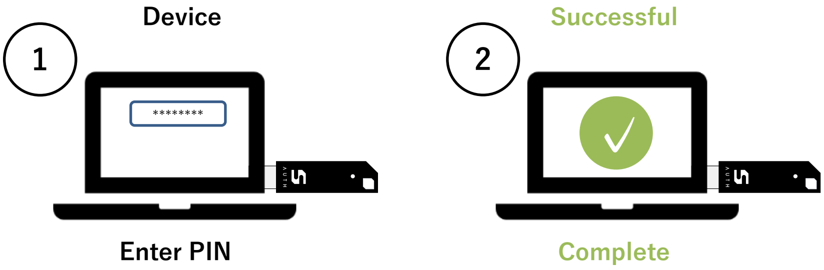


Figure 5 Procedure for user authentication or digital signature

1.3 Operating environment

Axell has checked that SHALO AUTH and SHALO AUTH dedicated software work on PCs with a USB port that run one of the operating systems listed in the following table.

Operating system	Version
Windows	Windows 10 for x86-based processors Windows 10 for x64-based processors
macOS	macOS High Sierra (10.13) or later Intended for Intel processors and Apple Silicon processors
Linux	Red Hat Enterprise Linux 7 or later CentOS 7 or later Ubuntu 18.04 LTS or later Fedora 33 or later * All of the above are intended for x64-based processors only.

SHALO AUTH can be used as a U2F security key in Web browsers listed in the following table.

Web browser	Version
Google Chrome	Version 41 or later
Firefox	Version 67 or later
Microsoft Edge	Version 79 or later (only for Chromium-based versions)
Safari	Version 13 or later

1.4 General specifications

Hardware specifications

Item	Description
Interface	USB 2.0
Compatible connector	USB Type-A
Power supply	USB bus powered +5 V ±5%
Dimensions	68.6 x 16 x 8 mm (including the cover)
Guaranteed operating environment	Temperature: -20 to 70°C, Humidity: 20 to 80% (non-condensing)
Weight	7 g
Certification	VCCI (Class B), FIDO U2F L2

FIDO U2F features

Feature	Description
Compliant with	U2F v1.2
Authentication algorithm	ECDSA P-256 with SHA-256
Upper limit of FIDO authentication keys to be generated	1,000,000 keys
Whether the user is present is verified by	Pressing the button on the device

PKCS #11 features

Feature	Description
Compliant with	PKCS #11 v2.40
SO PIN (Security Officer PIN)	UTF-8 string of 4 to 256 bytes in length With a protection feature that will lock the PIN in the event of 5 consecutive authentication failures
User PIN	UTF-8 string of 4 to 256 bytes in length With a protection feature that will lock the PIN in the event of 5 consecutive authentication failures
Cryptographic processing	RSA: Encryption, decryption, signature, verification; ECDSA: Signature, verification, random number generation, message digest generation
Data management	Maximum of 12 data sets of approximately 8 Kbytes can be stored per device. RSA private key, RSA public key, ECDSA private key, ECDSA public key, and X.509 certificate-based data are supported. Read-protected private keys are supported.
Message digest	SHA-1, SHA-256, SHA-384, and SHA-512
Random number generator	NIST SP 800-90A-compliant CTR-DRBG (AES-256 based)
RSA	RSA cryptography based on PKCS #1 Key lengths of 1,024 to 4,096 bits are supported.
ECDSA	The following FIPS 186-4-compliant elliptic curve signatures: secp192k1, secp192r1 (P-192), secp224k1, secp224r1 (P-224), secp256k1, secp256r1 (P-256), secp384r1 (P-384), and secp521r1 (P-521)

1.5 Usage notes

If your PC does not recognize SHALO AUTH, disconnect SHALO AUTH from the PC once and then reconnect the device to it.

When you connect SHALO AUTH to a self-powered USB hub, disconnect SHALO AUTH from the USB hub after your PC is shut down or enters standby, or after the hub is disconnected from the PC. Otherwise, the PC may not recognize SHALO AUTH after startup.

Chapter 2

Preparing SHALO AUTH for use

This chapter contains topics you should read before using SHALO AUTH.

Topics in this chapter

1. Appearance and features of SHALO AUTH
2. Understanding U2F
3. Understanding PKCS #11
4. Introduction to SHALO AUTH dedicated software

2.1 Appearance and features of SHALO AUTH

SHALO AUTH has one LED on the front and one button on the side. A cover protects its USB plug. To use SHALO AUTH, uncover it and connect it to a USB port.

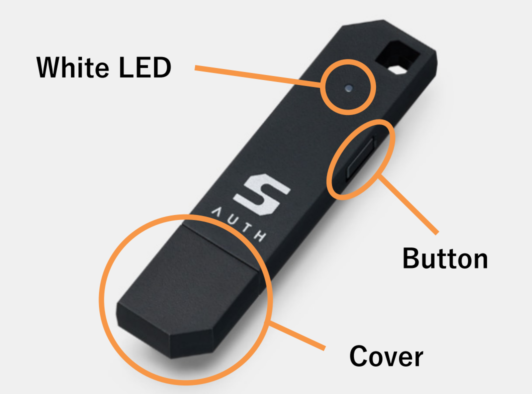


Figure 6 Appearance of SHALO AUTH

White LED

The white LED is usually off. It turns on to let the user know its state. The following table lists and describes the lighting patterns of the white LED and their meanings.

Timing	Lighting pattern	Description
After connection to a PC	On	The device is doing a self-test.
	1 to 3 flashes per second	An error was detected during the self-test.
Working with software	On	Data is being written to the device. Do not disconnect it from the PC.
	5 flashes per second	If SHALO Keyring or SHALO Smith is running, indicates that they are selected or working on the device. Otherwise, the software program is waiting for approval from the user based on U2F. Press the button for approval.
Pressing and holding button for about 30 seconds	10 flashes per second	The device is now ready to be restored to the factory settings without the SO PIN. The device keeps flashing for 10 seconds.

Button

The button is used mainly when the user approves a SHALO AUTH operation. This is explained in the next section.

2.2 Understanding U2F

SHALO AUTH can be used as a U2F security key. U2F, developed by the FIDO Alliance, is a mechanism for verifying identity in Web services.

The identity verification mechanism in U2F combines the following two factors:

Knowledge "Information only an individual user would know," such as an ID or password

Possession "Something only an individual user should possess," such as a USB token or smartphone

The way to verify identity by using two factors for authentication as described above is called **two-factor authentication (2FA)**.

The user makes a physical movement to authorize the U2F security key having that key conduct identity verification processing. SHALO AUTH flashes its LED to prompt the user to give authority, and the user then presses the button on the side of SHALO AUTH to do so.

A similar term to this that you may be familiar with is **two-step verification**. This is a way to verify identity through authentication with an ID and password (first step), followed by another form of authentication (second step). It does not matter if the second step employs a different factor than that of the first step.

Do not press the button if you will not allow the use of SHALO AUTH. If the LED flashes without any user interaction, malicious software may be attempting to secretly make use of SHALO AUTH.

Procedure for using U2F

If you want to use SHALO AUTH's U2F features for identity verification in U2F-compliant Web services, you need to have a Web browser that supports U2F and a PC with a USB port.

U2F is used in the following three actions:

1. Registering a security key
2. Verifying identity
3. Deregistering a security key

Deregistering the security key from the Web service enables you to prevent a new owner of the key from spoofing you later after the key is disposed of or transferred.

This section explains these actions in turn.

How to register a security key

Register the security key in a Web service's user settings. This registration process is quick and easy to do through a Web browser. Specifically, take the following three steps:

Step 1 During the operation of registering SHALO AUTH in the Web service, the LED on SHALO AUTH flashes, prompting you to authorize the use of SHALO AUTH.

Step 2 When you press the button on SHALO AUTH to authorize, SHALO AUTH then generates FIDO authentication keys (a pair of private and public keys) dedicated to this Web service.

Step 3 Register the SHALO AUTH information and the generated public key in the Web server.

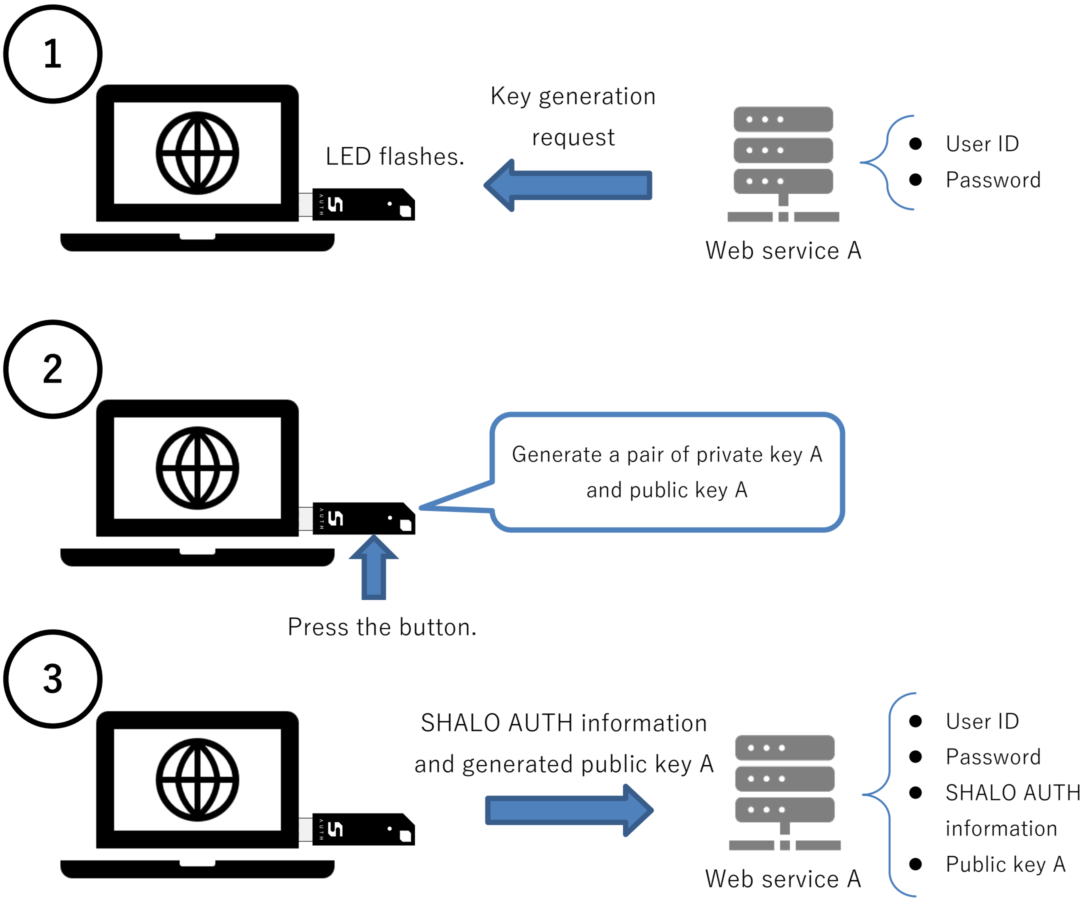


Figure 7 Registering a U2F security key

Many Web services **strongly recommend using two-factor authentication together with a different authentication method or a recovery method**. This is to avoid losing your ability to verify your identity due to damaging or losing the security key.

How to verify identity

First, enter your user ID and password in the identity verification process with U2F. Then, take the following three steps:

Step 1 When SHALO AUTH receives an authentication request from the Web service, the LED on SHALO AUTH flashes, prompting you to authorize the use of SHALO AUTH.

Step 2 When you press the button on SHALO AUTH to authorize, SHALO AUTH generates a digital signature with the private key dedicated to this Web service.

Step 3 The digital signature generated by SHALO AUTH is then sent to the Web service. The Web service verifies the identity by validating the digital signature generated by SHALO AUTH with the public key you registered.

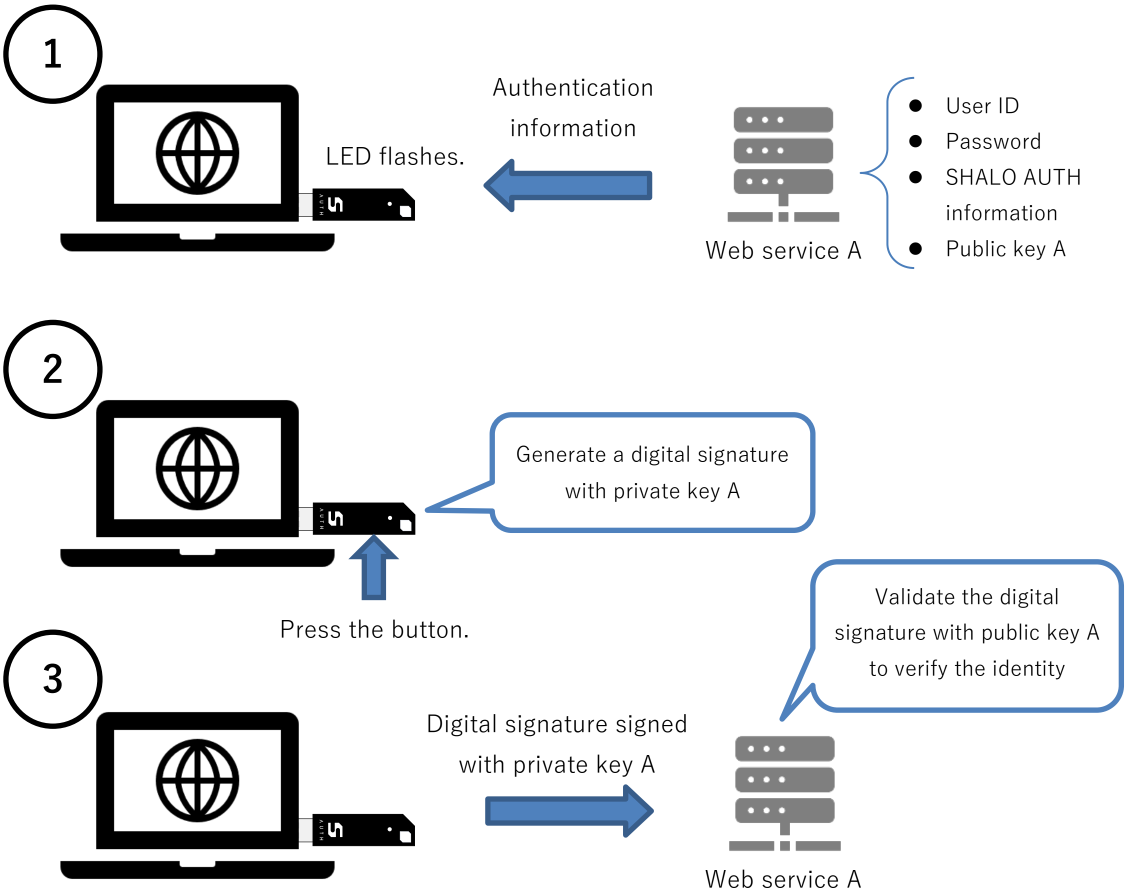

1
Authentication information
LED flashes.
Web service A
● User ID
● Password
● SHALO AUTH information
● Public key A
2
Generate a digital signature with private key A
Press the button.
3
Validate the digital signature with public key A to verify the identity
Digital signature signed with private key A
Web service A

Figure 8 Identity verification with U2F

How to deregister a security key

You can deregister a security key from a Web service by removing the registered U2F security key in the Web service's user settings.

This operation will remove the SHALO AUTH information and public key maintained by the Web service. Different Web services have different FIDO authentication keys registered, and therefore deregistration from one Web service does not affect any others.

SHALO AUTH can **disable all the FIDO authentication keys** that have been generated so far, in case it is disposed of or transferred. As such, it is possible to remove the information used for U2F from SHALO AUTH and have the device recognized as a new one. This can prevent the next SHALO AUTH owner from spoofing the previous owner even if the previous owner neglected to deregister through the Web service.

2.3 Understanding PKCS #11

2.3.1 What is PKCS #11?

PKCS #11 is the API to manipulate cryptographic tokens in software, and is widely used by applications that provide digital signatures or user authentication with those tokens.

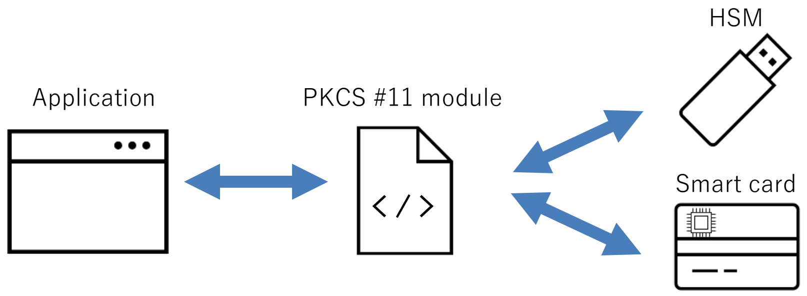


Figure 9 Where PKCS #11 is used

Cryptographic token

A cryptographic token refers to a cryptographic device, such as hardware security modules (HSMs) and smart cards. An HSM is a device that stores cryptographic keys securely and uses them to provide cryptographic processing functionality. A smart card refers to an IC card with a built-in IC chip, and its role is the same as that of an HSM. Common examples you may be familiar with are the following cards with **personal identification numbers (PINs)**:

- Credit card
- ATM card
- ID card

PKCS #11 features

PKCS #11 provides three main features with the cryptographic token. They are shown in the following table.

Feature	Description
PIN authentication	The following users can be distinguished from each other through authentication with a personal identification number (PIN): ● Security Officer (SO) ● General user (User) ● Public user (Public) A certain number of incorrect PIN entries can lock the PIN, thus protecting the cryptographic token.
Data management	Cryptographic keys and certificates can be securely stored and managed in the cryptographic token. This data is used for cryptographic processing. The feature can restrict the use of each data set to its owner and permanently prohibit it from being loaded from the cryptographic token.
Cryptographic processing	Stored keys can be used to encrypt or decrypt data and create and verify digital signatures. Message digests and high-security random numbers can be generated.

2.3.2 PIN authentication

In PKCS #11, entering PINs enables you to authenticate the following two types of roles:

Security Officer Is responsible for issuing cryptographic tokens and managing the PINs.

User Performs cryptographic processing by using secret information in the cryptographic token.

The User owns the cryptographic token. PIN authentication is not required for using public information in the cryptographic token. In SHALO AUTH, the PIN for the Security Officer is referred to as a **SO PIN** and that of the User is as a **user PIN**. Both PINs can accept 4 to 256 alphanumeric characters and symbols.

When someone purchases SHALO AUTH as an individual, the purchaser has the roles of both the Security Officer and User. Both roles can have the same pin.

Relationships between features and the PINs

The following table summarizes the relationships between a feature and which PIN is required to operate it:

Feature		Required PIN
Management	Initially configuring the cryptographic token	SO PIN (not required initially)
	Changing the SO PIN	SO PIN
	Configuring and unlocking the user PIN	SO PIN
Normal use	Changing the user PIN	User PIN
	Creating, reading, or removing data protected by the cryptographic token	User PIN
	Providing cryptographic processing with the key for data protected by the cryptographic token	User PIN
	Creating, reading, or removing public data by the cryptographic token	Not required
	Providing cryptographic processing with the public data key of the cryptographic token	Not required
	Providing cryptographic processing that does not use data of the cryptographic token	Not required

Locking the PINs

During PIN authentication, **five consecutive failures locks the PIN.** PIN authentication is then prohibited until the PIN is unlocked. The following table shows how to unlock a PIN.

Type of PIN	How to unlock
User PIN	Reset the user PIN as the Security Officer.
SO PIN	Restore the cryptographic token to the factory settings. **All the information and FIDO authentication keys are removed.**

⚠ FIDO authentication keys are not removed even if you initially configure SHALO AUTH for PKCS #11. **Restoring the device to the factory settings will remove all the FIDO authentication keys.**

2.3.3 Data management

Data capacity

SHALO AUTH can store up to 12 sets of the following three types of data defined by PKCS #11:

- Private key for public key cryptography (RSA or ECDSA)
- Public key for public key cryptography (RSA or ECDSA)
- X.509 certificate

The SHALO AUTH dedicated software stores these three types of keys as a set when storing a key in SHALO AUTH. This software enables you to store four sets of keys.

As an X.509 certificate contains public key information, SHALO AUTH will be able to handle up to six sets of key pairs if you use only one of either X.509 certificates or public keys.
In this case, use a different PKCS #11 application for data management.

Data set identification

In PKCS #11, information called an **CKA_ID attribute** is added to data to distinguish the relations among multiple data sets. Data items with the same CKA_ID attribute are considered to belong to the same set.

Both a private key and a public key that form a certain key pair have the same CKA_ID attribute. An X.509 certificate that is issued for the public key also has the same CKA_ID attribute.

Data protection

In SHALO AUTH, you can protect data by:

- Prohibiting any change to it
- Prohibiting any removal of it
- Requiring user PIN authentication prior to use of or access to the data
- Prohibiting export of the data (for the private key only)

When data is saved with the SHALO AUTH dedicated software, it is managed as shown in the table below. If you want to manage data under conditions that are not in the table, use a different PKCS #11 application.

Data type	Change	Removal	User PIN authentication protection	Export
Private key for public key cryptography	Possible	Possible	Required	Not possible
Public key for public key cryptography	Possible	Possible	Not required	Possible
X.509 certificate	Possible	Possible	Not required	Possible

2.4 Introduction to SHALO AUTH dedicated software

SHALO AUTH offers the following two software programs for the general security key:

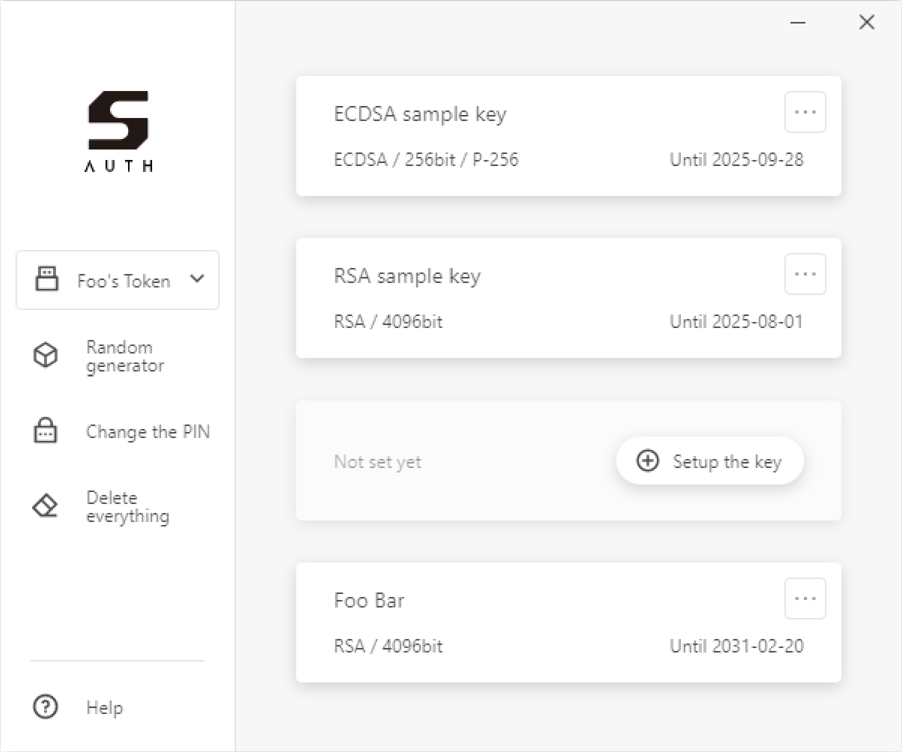
SHALO Keyring Software program, used to store key data in SHALO AUTH

SHALO Smith Software program, used to manage SHALO AUTH

SHALO Smith is also used to disable all the FIDO authentication keys when you dispose of or transfer SHALO AUTH.

SHALO Keyring

SHALO Keyring is a software program for storing cryptographic keys handled by the general security key functionality in SHALO AUTH.



ECDSA sample key
ECDSA / 256bit / P-256 Until 2025-09-28

RSA sample key
RSA / 4096bit Until 2025-08-01

Not set yet

Foo Bar
RSA / 4096bit Until 2031-02-20

Figure 10 SHALO Keyring window

SHALO Keyring provides the features listed below. The PKCS #11 SO PIN is not required for these purposes.

- Setting up SHALO AUTH
- Adding or removing cryptographic keys
- Changing the user PIN
- Generating passwords or random number sequences

Chapter 4 explains how to use SHALO Keyring.

SHALO Smith

SHALO Smith is a software program for managing SHALO AUTH.

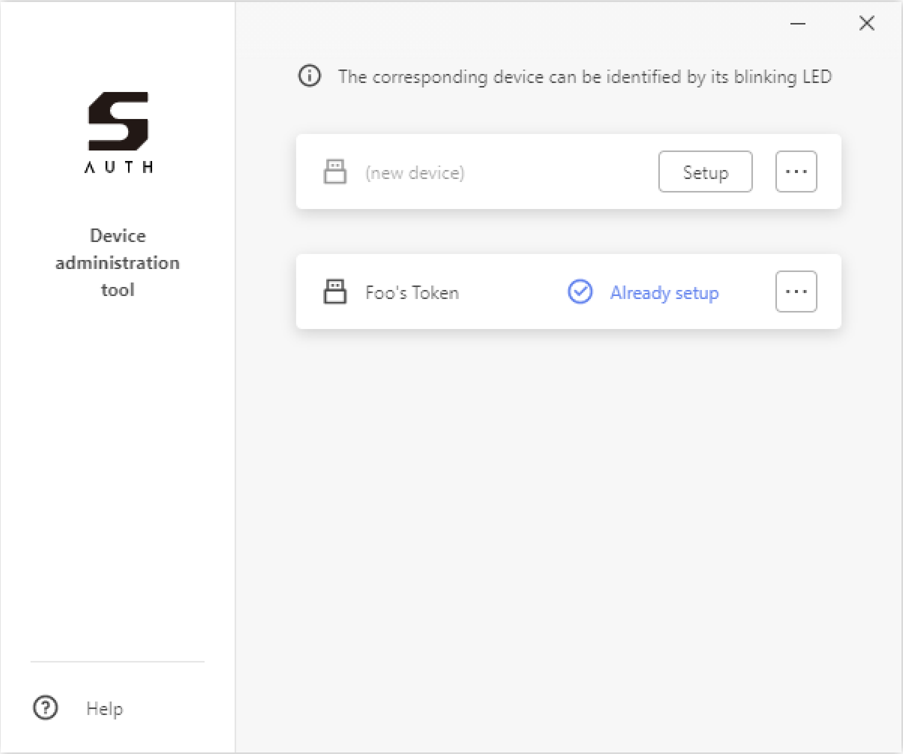

AUTH
Device administration tool
Help
The corresponding device can be identified by its blinking LED
(new device)
Setup
...
Foo's Token
Already setup
...

Figure 11 SHALO Smith window

SHALO Smith provides the features listed below. You need a SO PIN to operate them.

- Setting up SHALO AUTH
- Changing the SO PIN
- Resetting and unlocking the user PIN
- Restoring the device to the factory settings removing all FIDO authentication keys

Chapter 5 explains how to use SHALO Smith.

Chapter 3

Installation

This chapter explains how to install and uninstall the SHALO AUTH dedicated software.

The software runs on the following OSs:

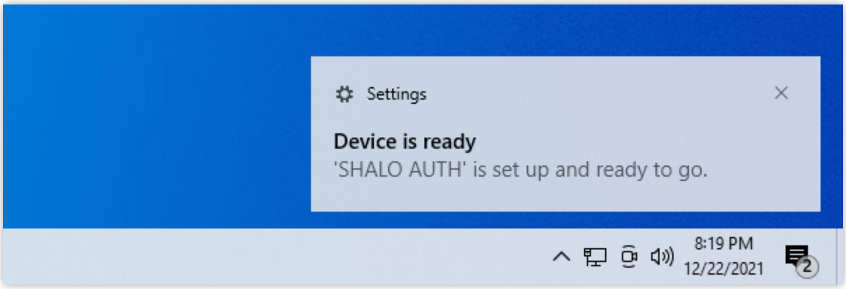
- Windows
- macOS
- Linux

Topics in this chapter

1. Installation in Windows
2. Uninstallation in Windows
3. Installation in macOS
4. Uninstallation in macOS
5. Installation in Linux
6. Uninstallation in Linux

3.1 Installing software in Windows

SHALO AUTH works with standard drivers that come with Windows. Connecting SHALO AUTH to a USB port of your PC for the first time will start the setup process automatically. When the setup process finishes, you will see a notification on your desktop as shown in the following figure.



Settings

Device is ready

'SHALO AUTH' is set up and ready to go.

Figure 12 Setup completion notification from SHALO AUTH

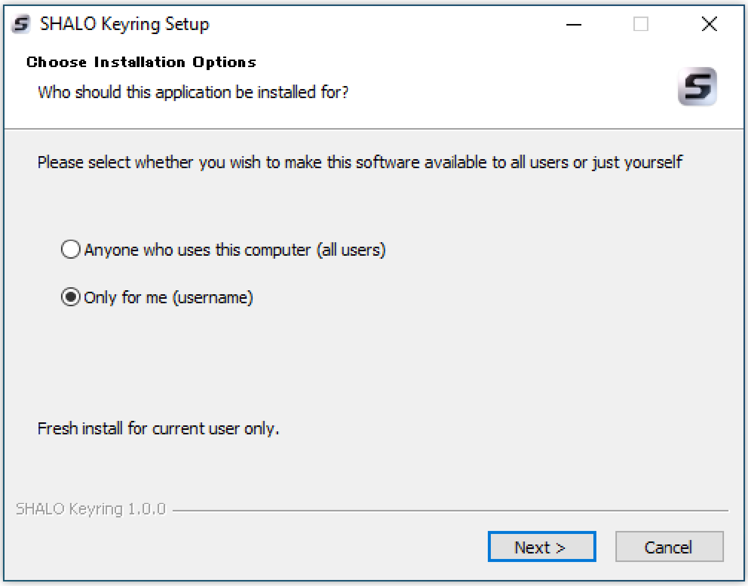
If you use SHALO AUTH as a U2F security key only, you do not have to read the rest of this section.

3.1.1 Installing SHALO Keyring

SHALO Keyring for Windows can be downloaded from https://auth.shalo.jp.

To install it, run the shalo_keyring_*x.y.z*_windows.exe file you downloaded (where x.y.z indicates the version number). The installation process contains three steps.

First, select an installation option in the first screen. If you do not have administrator privileges, select [**Only for me**].



SHALO Keyring Setup

Choose Installation Options

Who should this application be installed for?

Please select whether you wish to make this software available to all users or just yourself

○ Anyone who uses this computer (all users)

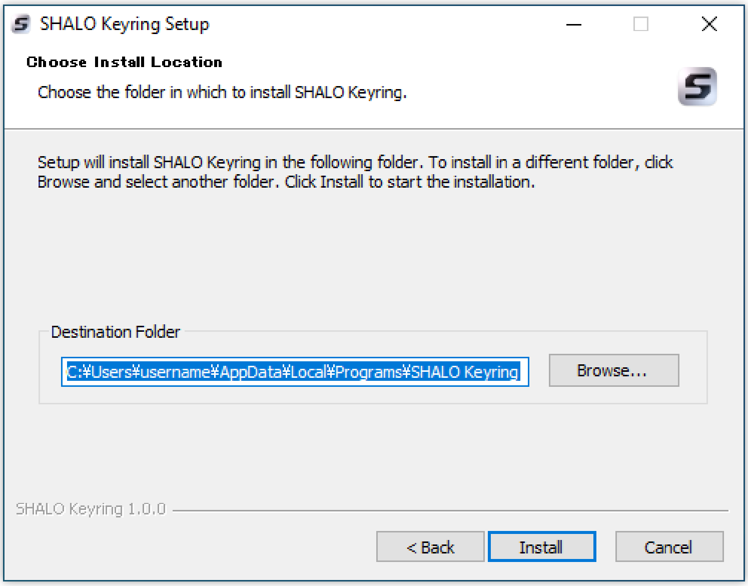
● Only for me (username)

Fresh install for current user only.

SHALO Keyring 1.0.0

Figure 13 SHALO Keyring installation options

Next, specify where to install the program. If the correct location is specified, click [**Install**].



SHALO Keyring Setup

Choose Install Location

Choose the folder in which to install SHALO Keyring.

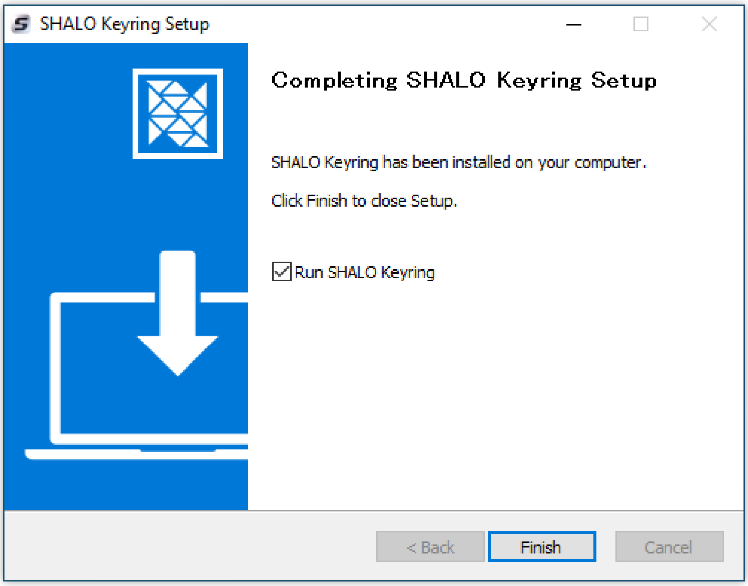
Setup will install SHALO Keyring in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:¥Users¥username¥AppData¥Local¥Programs¥SHALO Keyring

Figure 14 Specifying where to install SHALO Keyring

When the installation process is complete, you will see the screen below. Click [**Finish**] to exit.



SHALO Keyring Setup

Completing SHALO Keyring Setup

SHALO Keyring has been installed on your computer.

Click Finish to close Setup.

☑ Run SHALO Keyring

Figure 15 SHALO Keyring installation is complete

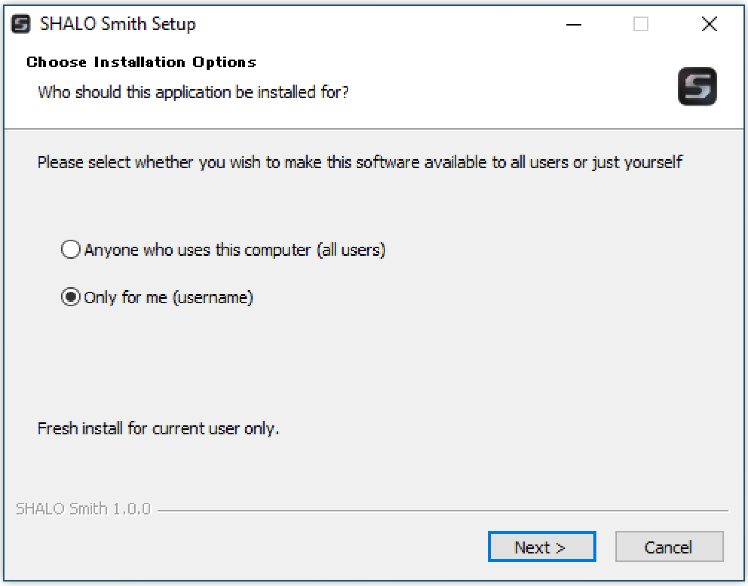
You can start SHALO Keyring with a shortcut created on your desktop or in the Start menu.

3.1.2 Installing SHALO Smith

SHALO Smith for Windows can be downloaded from https://auth.shalo.jp.

To install it, run the shalo_smith_*x.y.z*_windows.exe file you downloaded (where x.y.z indicates the version number). The installation process contains three steps.

First, select an installation option in the first screen. If you do not have administrator privileges, select [**Only for me**].



SHALO Smith Setup

Choose Installation Options

Who should this application be installed for?

Please select whether you wish to make this software available to all users or just yourself

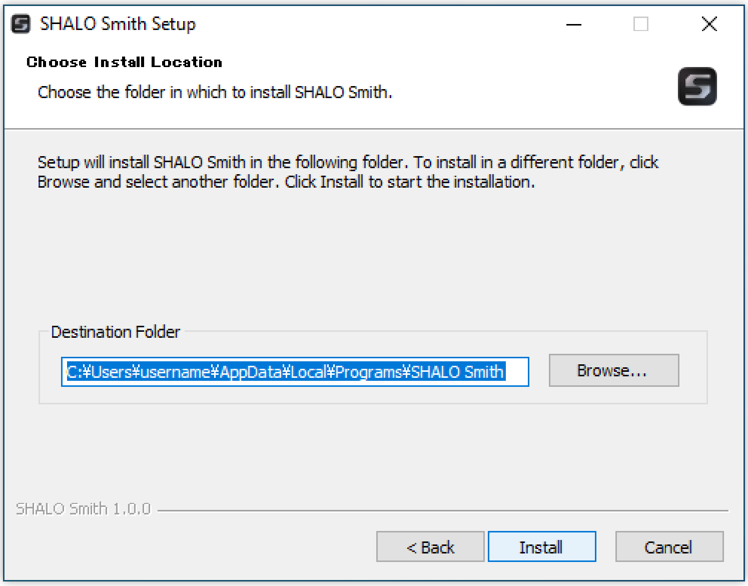
○ Anyone who uses this computer (all users)

● Only for me (username)

Fresh install for current user only.

Figure 16 SHALO Smith installation options

Next, specify where to install the program. If the correct location is specified, click [**Install**].



SHALO Smith Setup

Choose Install Location

Choose the folder in which to install SHALO Smith.

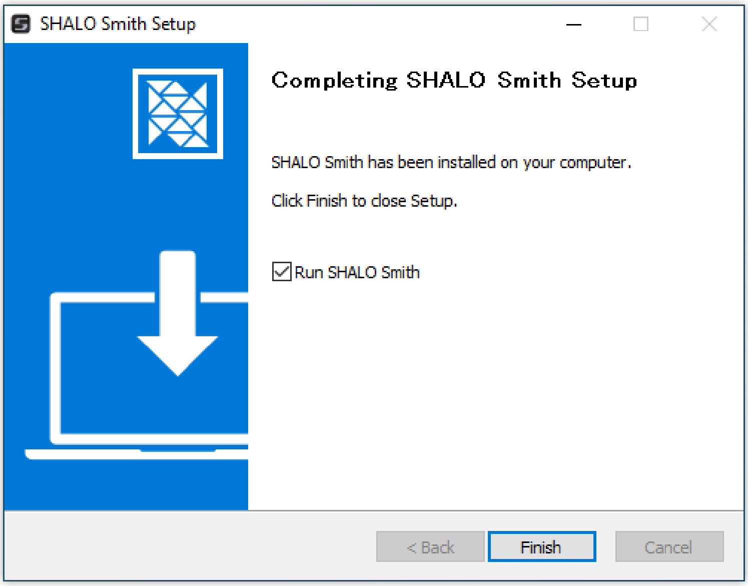
Setup will install SHALO Smith in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:¥Users¥username¥AppData¥Local¥Programs¥SHALO Smith

Figure 17 Specifying where to install SHALO Smith

When the installation process is complete, you will see the screen below. Click [**Finish**] to exit.



SHALO Smith Setup

Completing SHALO Smith Setup

SHALO Smith has been installed on your computer.

Click Finish to close Setup.

☑ Run SHALO Smith

Figure 18 SHALO Smith installation is complete

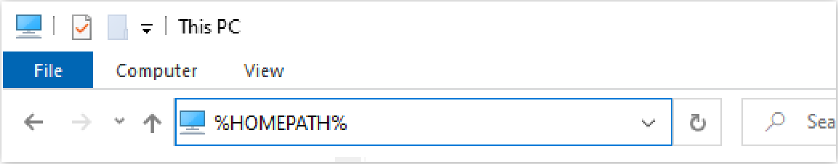
You can start SHALO Smith with a shortcut created on your desktop or in the Start menu.

3.1.3 Installing the PKCS #11 module

The PKCS #11 module for Windows can be downloaded from https://auth.shalo.jp.

To install the PKCS #11 module, extract the downloaded ZIP file into a folder named shalo_pkcs11 in the home directory. Use the following procedure:

1. In File Explorer, go to the home directory.
   You can go there by typing "%HOMEPATH%" in the address bar of File Explorer and then pressing the Enter key, as shown below.



2. Create a folder with the name of "**shalo_pkcs11**".
3. Right-click the shalo_pkcs11_*x.y.z*_windows.zip file you downloaded (where x.y.z indicates the version number), and in the menu, select [**Extract All...**].
4. Specify the folder created in step 2 as the folder into which the file is extracted.

The shalo_pkcs11 folder after the installation will appear as shown below.

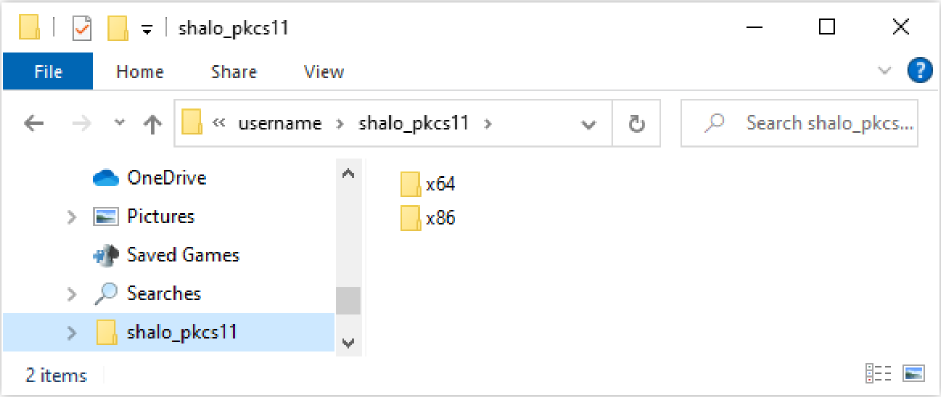


Figure 19 shalo_pkcs11 folder in the home directory

If your system drive is C:, the absolute path to the shalo_pkcs11 folder you created is as shown below. *username* must be read as your own Windows username.

```
C:\Users\username\shalo_pkcs11
```

The following table lists the paths to the installed PKCS #11 module by application.

Module usage	Mode	Relative path from the home directory
Windows application	32 bit	`shalo_pkcs11\x86\slpkcs11-vc.dll`
	64 bit	`shalo_pkcs11\x64\slpkcs11-vc.dll`
Application ported to Windows (MinGW, Cygwin, Git for Windows)	32 bit	`shalo_pkcs11\x86\slpkcs11-mingw32.dll`
	64 bit	`shalo_pkcs11\x64\slpkcs11-mingw64.dll`

3.2 Uninstalling software in Windows

3.2.1 Uninstalling SHALO Keyring

You can uninstall SHALO Keyring by using the following procedure:

1. Right-click on **Start**, then select [**Apps and Features**].
2. Select “SHALO Keyring” from the apps list, and then click [**Uninstall**]. SHALO Keyring Uninstall window will appear.
3. Click [**Next**] on the uninstall window.
4. Click [**Finish**] to close the uninstall window.

3.2.2 Uninstalling SHALO Smith

You can uninstall SHALO Smith by using the following procedure:

1. Right-click on **Start**, then select [**Apps and Features**].
2. Select “SHALO Smith” in the apps list, and then click [**Uninstall**]. SHALO Smith Uninstall window will appear.
3. Click [**Next**] on the uninstall window.
4. Click [**Finish**] to close the uninstall window.

3.2.3 Uninstalling the PKCS #11 module

To uninstall the PKCS #11 module, delete the folder containing the module using File Explorer. The procedure is shown below:

1. Exit software that uses the module. Unregister the module from the software, if needed.
2. In File Explorer, go to the home directory.
3. Delete the shalo_pkcs11 folder.

If the PKCS #11 module is registered in Acrobat®, deregister the module from Acrobat® according to Section 7.2.2.
If the authentication agent is configured to start automatically, remove SHALO AUTH from the authentication agent.

3.3 Installing software in macOS

SHALO AUTH works with standard drivers that come with macOS. If you use SHALO AUTH as a U2F security key only, you do not have to read this section.

3.3.1 Installing SHALO Keyring

SHALO Keyring for macOS can be downloaded from https://auth.shalo.jp.

To install it, double-click the shalo_keyring_*x.y.z*_macos.dmg file you downloaded (where x.y.z indicates the version number) to open it.

In the window that appears as shown in the following figure, drag and drop the SHALO Keyring icon on the left into the Applications folder on the right to complete the installation.

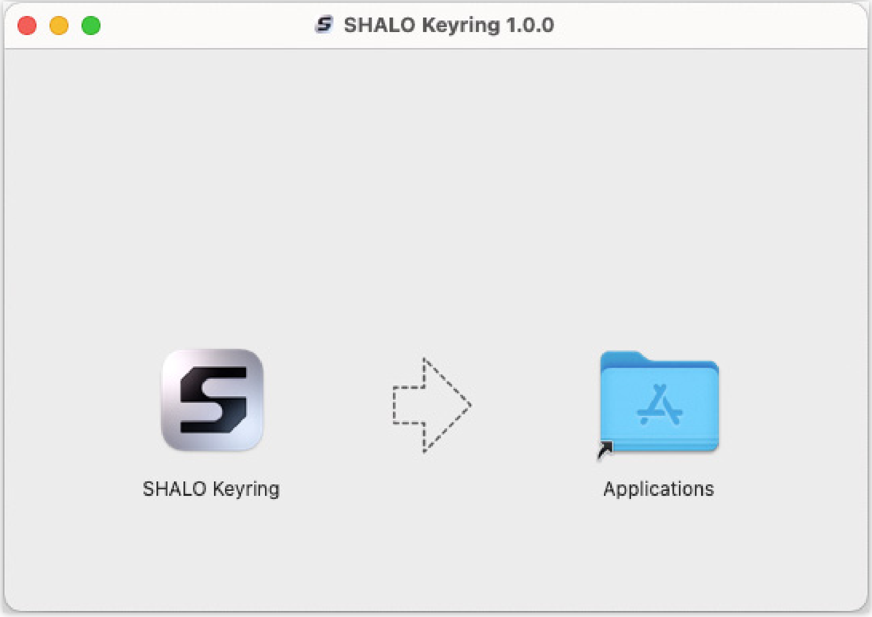


Figure 20 SHALO Keyring installation

You can start SHALO Keyring from Launchpad or the Applications folder.

3.3.2 Installing SHALO Smith

SHALO Smith for macOS can be downloaded from https://auth.shalo.jp.

To install it, double-click the shalo_smith_*x.y.z*_macos.dmg file you downloaded (where x.y.z indicates the version number) to open it.

In the window that appears as shown in the following figure, drag and drop the SHALO Smith icon on the left into the Applications folder on the right to complete the installation.

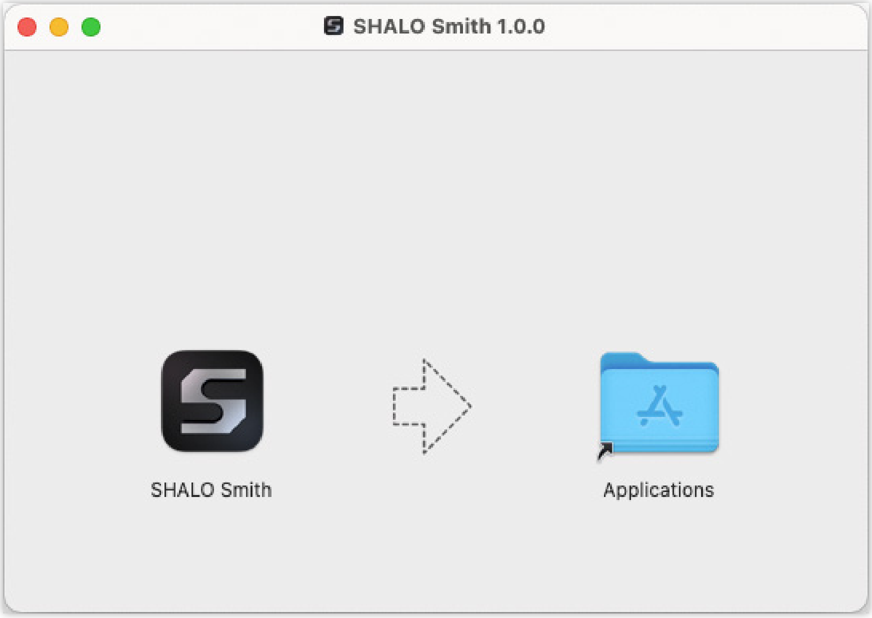


Figure 21 SHALO Smith installation

You can start SHALO Smith from Launchpad or the Applications folder.

3.3.3 Installing the PKCS #11 module

The PKCS #11 module for macOS can be downloaded from https://auth.shalo.jp.

When you double-click the shalo_pkcs11_*x.y.z*_macos.zip file you downloaded (where x.y.z indicates the version number) in Finder, the ZIP file is extracted to create the libslpkcs11.dylib file of the PKCS #11 module.

To install the module, **as the root user, copy** this libslpkcs11.dylib file to `/usr/local/lib`.

`/usr/local/lib` is the default whitelist for ssh-agent.

If you extracted the ZIP file in the Downloads folder, open the Terminal and run the following command:

```
% sudo cp ~/Downloads/libslpkcs11.dylib /usr/local/lib/↵
```

If you extracted the ZIP file in a different folder, change `~/Downloads` to the path to that folder.

The following message may appear, depending on your macOS environment:

```
cp: directory /usr/local/lib does not exist
```

If this happens, run the following commands to create a directory as the root user, and then copy the file:

```
% sudo mkdir -p /usr/local/lib↵
% sudo cp ~/Downloads/libslpkcs11.dylib /usr/local/lib/↵
```

3.4 Uninstalling software in macOS

3.4.1 Uninstalling SHALO Keyring

You can uninstall SHALO Keyring by using the following procedure:

1. Go to Application folder in Finder.
2. Drag "SHALO Keyring" to the **Trash** in the Dock or select it then press Command-Delete.
3. Open the Terminal and run the following command:

```
% rm -r ~/Library/Application\ Support/shalo-keyring↵
```

3.4.2 Uninstalling SHALO Smith

You can uninstall SHALO Smith by using the following procedure:

1. Go to Application folder in Finder.
2. Drag "SHALO Smith" to the **Trash** in the Dock or select it then press Command-Delete.
3. Open the Terminal and run the following command:

```
% rm -r ~/Library/Application\ Support/shalo-smith↵
```

3.4.3 Uninstalling the PKCS #11 module

To uninstall the PKCS #11 module, **as the root user, delete** the libslpkcs11.dylib file from `/usr/local/lib`. Open the Terminal and run the following command:

```
% sudo rm /usr/local/lib/libslpkcs11.dylib↵
```

If the PKCS #11 module is registered in Acrobat®, deregister the module from Acrobat® according to Section 7.2.2.
If the authentication agent is configured to start automatically, remove SHALO AUTH from the authentication agent.

3.5 Installing software in Linux

SHALO AUTH works with standard drivers that come with Linux. However, you need the operations in Section 3.5.1 if you want to use the device without root privileges. This is also the case if you use SHALO AUTH as a U2F security key.

3.5.1 Installing an udev rules file

An udev rules file for SHALO AUTH must be installed before a non-root user can use SHALO AUTH.

The following download software for Linux contains the udev rules file for SHALO AUTH with the file name 60-usb-shalo-auth.rules:

- SHALO Keyring
- SHALO Smith
- PKCS #11 module

To install it, **as the root user, copy** 60-usb-shalo-auth.rules in the downloaded file to `/etc/udev/rules.d`, and run the `udevadm` command to apply the new rules immediately.

To do this, open the terminal and run the following commands:

```
$ tar xvzf file-you-downloaded↵
$ sudo cp 60-usb-shalo-auth.rules /etc/udev/rules.d/↵
$ sudo udevadm control --reload-rules↵
```

The udev rules are not applied to SHALO AUTH that has already been connected at the time of rules installation. If you want to reapply them, reconnect SHALO AUTH.

3.5.2 Installing a prerequisite

libfuse2 must be installed before you can successfully run SHALO Keyring or SHALO Smith. If you are using Ubuntu 22.04 LTS or later, you can install it by running the following commands on the terminal:

```
$ sudo apt update↵
$ sudo apt -y install libfuse2↵
```

Ubuntu prior to 22.04 LTS or other Linux distribution generally include libfuse2, so the above is not necessary.

3.5.3 Installing SHALO Keyring

SHALO Keyring for Linux can be downloaded from https://auth.shalo.jp.

To extract the shalo_keyring_*x.y.z*_linux.tar.gz file you downloaded (where x.y.z indicates the version number), run the following command on the terminal:

```
$ tar xvzf shalo_keyring_x.y.z_linux.tar.gz↵
shaloKeyring.appimage
60-usb-shalo-auth.rules
```

This command causes the following files to be created:

shaloKeyring.appimage	SHALO Keyring for Linux
60-usb-shalo-auth.rules	udev rules file for SHALO AUTH

If the udev rules file (60-usb-shalo-auth.rules) has not been installed yet, see Section 3.5.1 to install it.

shaloKeyring.appimage has no predetermined installation location. You can install it anywhere convenient for your management. To start SHALO Keyring, run shaloKeyring.appimage.

3.5.4 Installing SHALO Smith

SHALO Smith for Linux can be downloaded from https://auth.shalo.jp.

To extract the shalo_smith_*x.y.z*_linux.tar.gz file you downloaded (where x.y.z indicates the version number), run the following command on the terminal:

```
$ tar xvzf shalo_smith_x.y.z_linux.tar.gz↵
shaloSmith.appimage
60-usb-shalo-auth.rules
```

This command causes the following files to be created:

shaloSmith.appimage	SHALO Smith for Linux
60-usb-shalo-auth.rules	udev rules file for SHALO AUTH

If the udev rules file (60-usb-shalo-auth.rules) has not been installed yet, see Section 3.5.1 to install it.

shaloSmith.appimage has no predetermined installation location. You can install it anywhere convenient for your management. To start SHALO Smith, run shaloSmith.appimage.

3.5.5 Installing the PKCS #11 module

The PKCS #11 module for Linux can be downloaded from https://auth.shalo.jp.

To extract the shalo_pkcs11_*x.y.z*_linux.tar.gz file you downloaded (where x.y.z indicates the version number), run the following command on the terminal:

```
$ tar xvzf shalo_pkcs11_x.y.z_linux.tar.gz↵
libslpkcs11.so
60-usb-shalo-auth.rules
```

This command causes the following files to be created:

libslpkcs11.so	PKCS #11 module for Linux
60-usb-shalo-auth.rules	udev rules file for SHALO AUTH

If the udev rules file (60-usb-shalo-auth.rules) has not been installed yet, see Section 3.5.1 to install it.

To install the module, **as the root user, copy** the libslpkcs11.so PKCS #11 module to `/usr/lib`. Open the terminal and run the following command:

```
$ sudo cp libslpkcs11.so /usr/lib/↵
```

`/usr/lib` and `/usr/local/lib` are the directories that are on the whitelist for ssh-agent.
Many Linux distributions for desktop automatically run ssh-agent when the user logs in to the GUI. ssh-agent is started at this time, but it is not easy to add a whitelist to the agent.
In this manual, you avoid modifying the startup option by adding the PKCS #11 module to `/usr/lib`, one of the default whitelists.

3.6 Uninstalling software in Linux

3.6.1 Uninstalling the udev rules file

To uninstall the udev rules file, **as the root user, delete** 60-usb-shalo-auth.rules from `/etc/udev/rules.d`. Open the Terminal and run the following command:

```
$ sudo rm /etc/udev/rules.d/60-usb-shalo-auth.rules↵
```

3.6.2 Uninstalling SHALO Keyring

To uninstall SHALO Keyring, delete the extracted shaloKeyring.appimage file. Then, run the following command in the Terminal to erase the configuration files.

```
$ rm -r ~/.config/shalo-keyring↵
```

3.6.3 Uninstalling SHALO Smith

To uninstall SHALO Smith, delete the extracted shaloSmith.appimage file. Then, run the following command in the Terminal to erase the configuration files.

```
$ rm -r ~/.config/shalo-smith↵
```

3.6.4 Uninstalling the PKCS #11 module

To uninstall the PKCS #11 module, **as the root user, delete** the libslpkcs11.so file from `/usr/lib`. Open the Terminal and run the following command:

```
$ sudo rm /usr/lib/libslpkcs11.so↵
```

If the authentication agent is configured to start automatically, remove SHALO AUTH from the authentication agent.

Chapter 4

Using the SHALO Keyring key tool

This chapter explains the key tool, SHALO Keyring. SHALO Keyring is a software program for configuring cryptographic keys in SHALO AUTH for the general security key functionality.

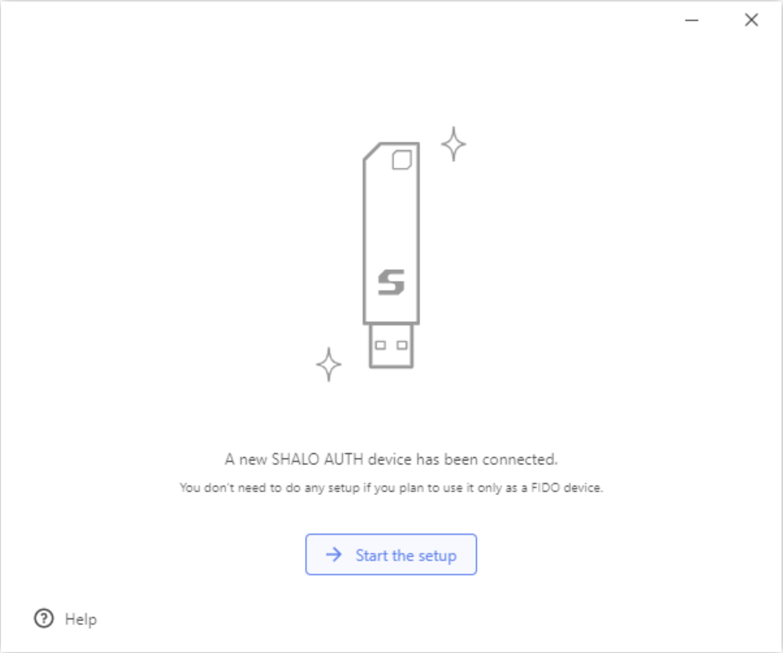
If you use SHALO AUTH as a U2F security key only, you do not have to read this chapter.

Topics in this chapter

1. Setting up SHALO AUTH
2. Viewing the state of SHALO AUTH
3. Generating a new key
4. Importing an existing key
5. Removing a key
6. Obtaining a public key
7. Changing the user PIN
8. Generating a password or random number sequence
9. CKA_ID attribute of key data

4.1 Setting up SHALO AUTH

SHALO Keyring displays the window shown in Figure 22 when detecting a new SHALO AUTH device. You can start the setup process by clicking [**Start the setup**] in this window.



A new SHALO AUTH device has been connected.

You don't need to do any setup if you plan to use it only as a FIDO device.

Figure 22 SHALO AUTH setup by SHALO Keyring

During the setup process, the tool initializes the data area for general security key functionality and configures the following management information:

Device label	An individual name used to identify multiple SHALO AUTH.
User PIN	The password for when the user uses the device. It allows the use of protected cryptographic keys.
SO PIN	The password for management. It is used to reset the user PIN or to restore SHALO AUTH to the factory settings.

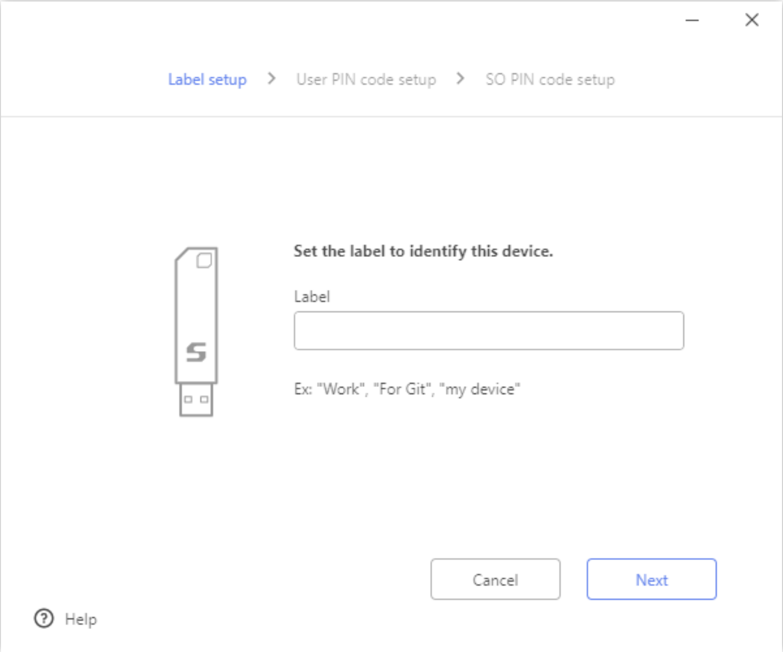
This setup process does not affect any U2F security key functionality.
If you have registered SHALO AUTH as a U2F security key in a Web service before the above setup process, you can still continue to use the device in that service.

If you have previously set up a SHALO AUTH device and want to set it up again, you must use SHALO Smith to restore the device to its factory settings. When the device is restored to its factory settings, the U2F security key information in it is also removed.

During the SHALO AUTH setup process, configure the device label, user PIN, and SO PIN in this order.

Specifying the device label

The device label can include alphanumeric characters and symbols as well as character strings in Japanese and other languages. The maximum number of characters in the label depends on the types of characters. If the label is too long, you will see a warning.



Set the label to identify this device.

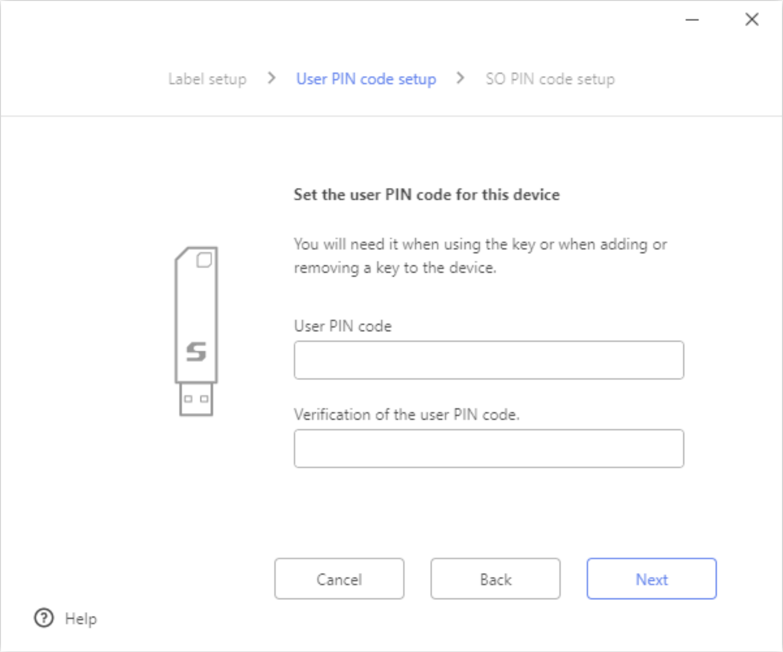
Label

Ex: "Work", "For Git", "my device"

Figure 23 Specifying the device label

Specifying the user PIN

The user PIN can include alphanumeric characters and symbols. Specify a user PIN between 4 and 256 characters long. Enter the user PIN twice for confirmation.



Set the user PIN code for this device

You will need it when using the key or when adding or removing a key to the device.

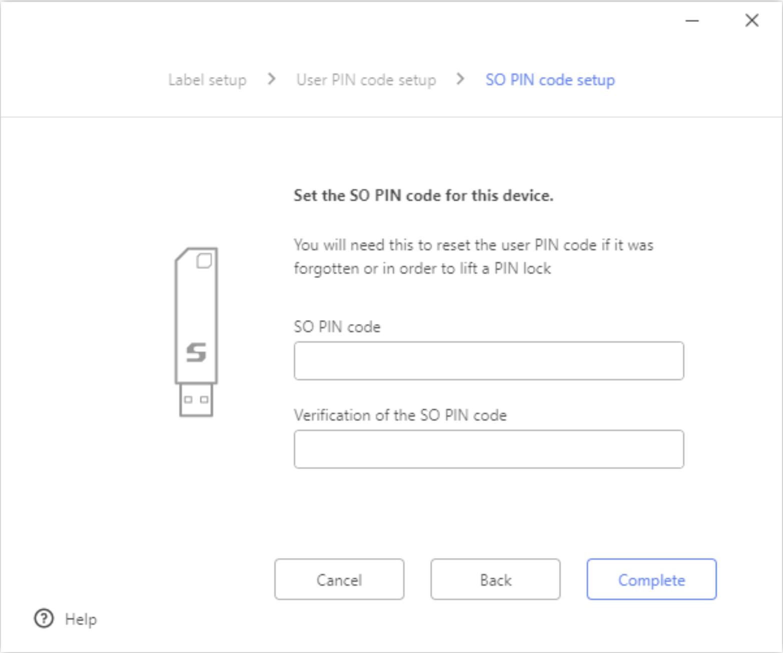
User PIN code

Verification of the user PIN code.

Figure 24 Specifying the user PIN

Specifying the SO PIN

The SO PIN can include alphanumeric characters and symbols. Specify a SO PIN between 4 and 256 characters long. Enter the SO PIN twice for confirmation.



Set the SO PIN code for this device.

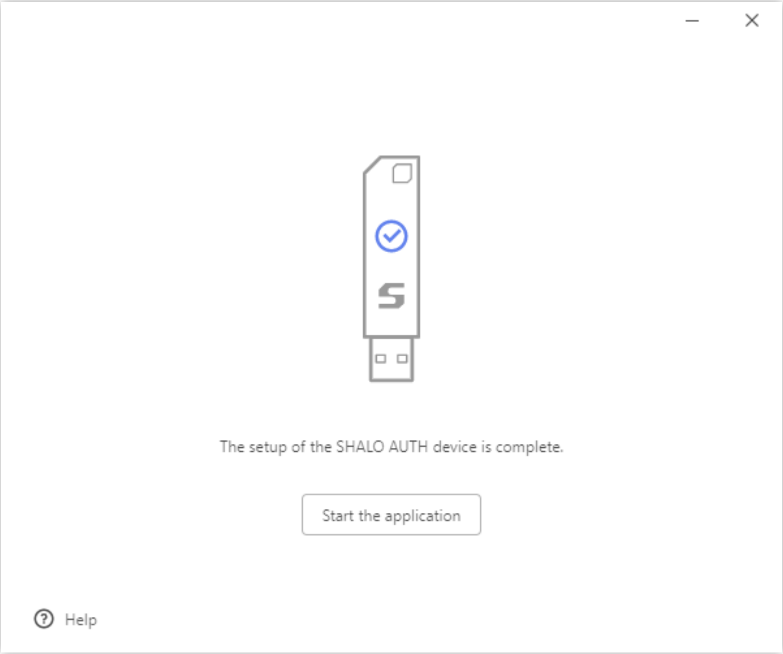
You will need this to reset the user PIN code if it was forgotten or in order to lift a PIN lock

SO PIN code

Verification of the SO PIN code

Figure 25 Specifying the SO PIN

When the setup process is complete, you will see the window below. Finally, click [**Start the application**].

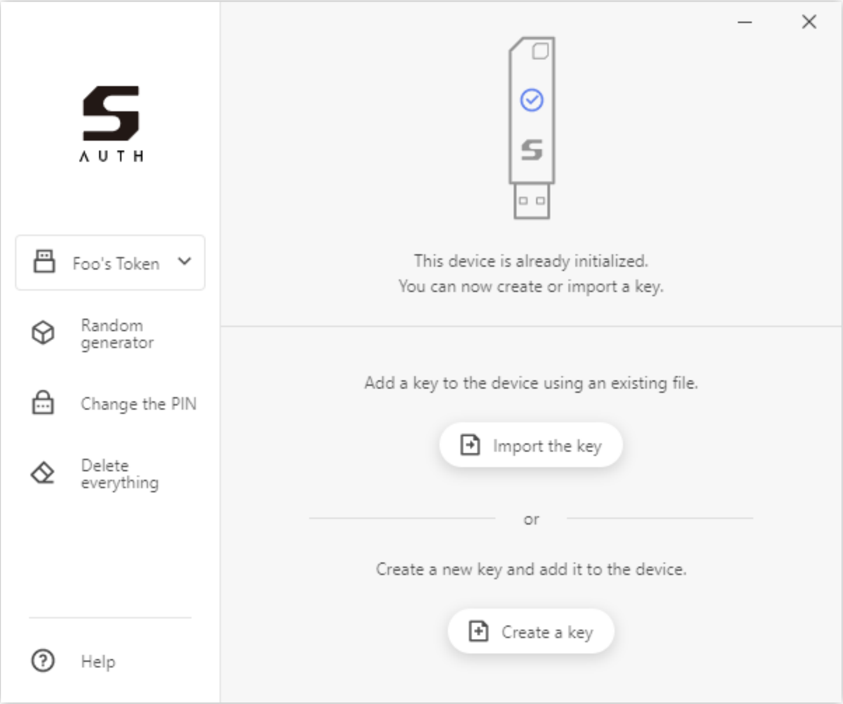


The setup of the SHALO AUTH device is complete.

Figure 26 SHALO Keyring window after the completed setup process

4.2 Viewing the state of SHALO AUTH

SHALO Keyring has different window layouts depending on the state of SHALO AUTH. When SHALO AUTH has just been set up, one window shown in Figure 27 appears, but when SHALO AUTH has one or more keys configured, another window shown in Figure 28 will appear.



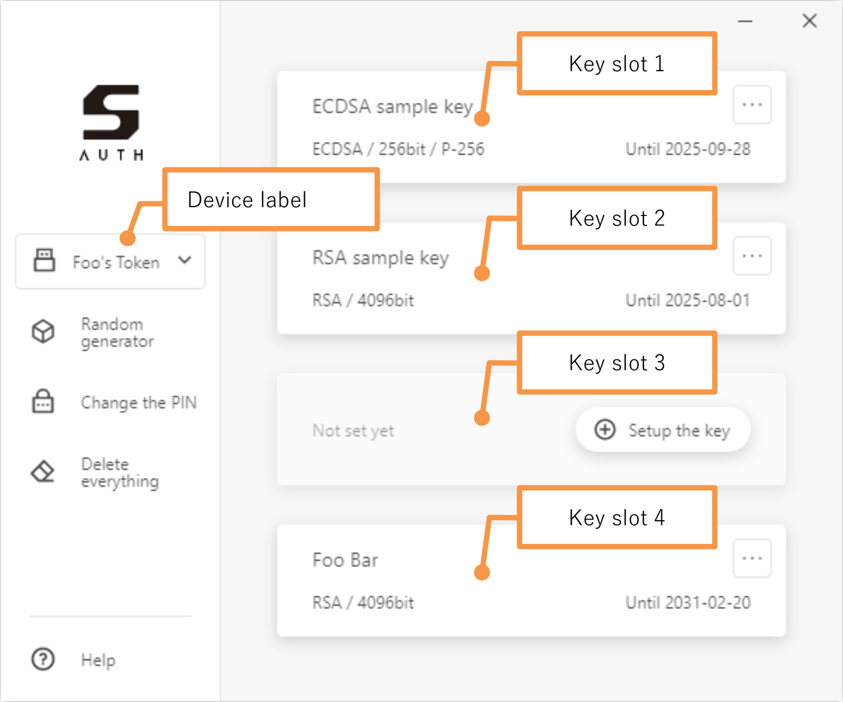
This device is already initialized.
You can now create or import a key.

Add a key to the device using an existing file.

or

Create a new key and add it to the device.

Figure 27 When you connect SHALO AUTH to the PC immediately after the setup



ECDSA sample key
ECDSA / 256bit / P-256 Until 2025-09-28

RSA sample key
RSA / 4096bit Until 2025-08-01

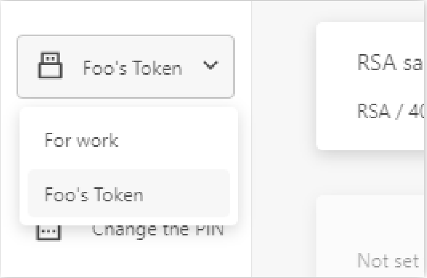
Not set yet

Foo Bar
RSA / 4096bit Until 2031-02-20

Figure 28 When you connect SHALO AUTH with added keys to a PC

Device label

This area displays the device label of SHALO AUTH you are viewing the information in and working with.



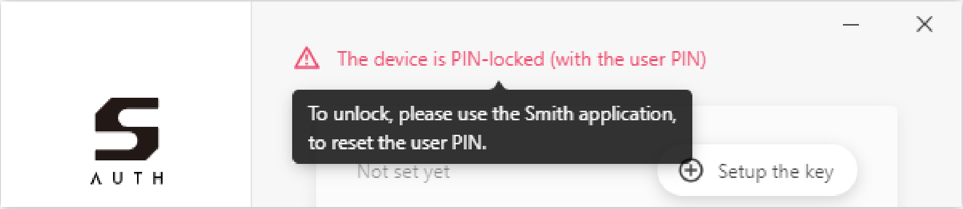
Clicking the label displays the list of SHALO AUTH devices connected to the PC. When you select a device label from the list, the information of the selected SHALO AUTH device now appears in the window. In addition, actions such as [**Change the PIN**] and [**Delete everything**] are applied to the selected SHALO AUTH device.

SHALO Keyring can work with up to eight SHALO AUTH devices at once.

The LED of the selected SHALO AUTH device flashes. If multiple SHALO AUTH devices are connected to the PC, you can distinguish the device you are manipulating from others by looking at whether the LEDs are flashing.

State of the device

If the device is in an abnormal state, a warning is displayed in red at the top of the window. When you hover the mouse cursor over the warning, the solution appears in the tool tip.

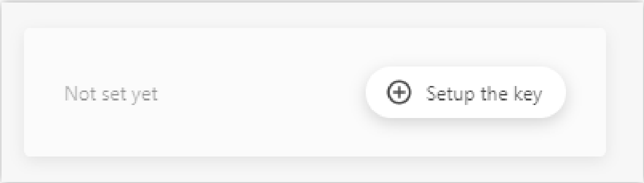

The device is PIN-locked (with the user PIN)

To unlock, please use the Smith application, to reset the user PIN.

Key slot

SHALO Keyring can store four sets of key data in SHALO AUTH. A storage area for the data is called a **key slot**, and the information of key slots 1 to 4 is arranged vertically for display.

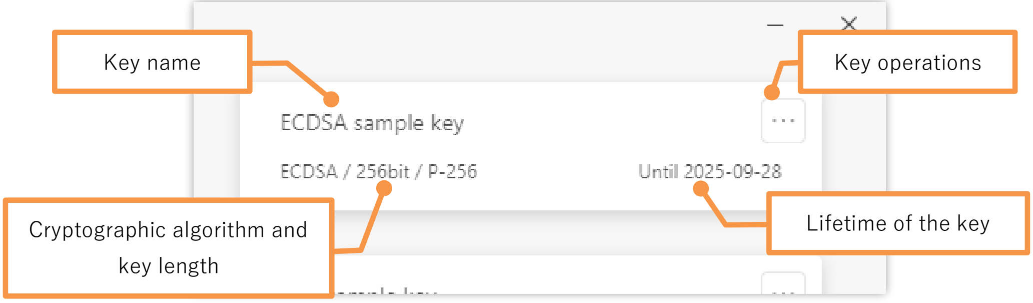
When a key slot has no key, the slot is displayed as shown in the following figure.



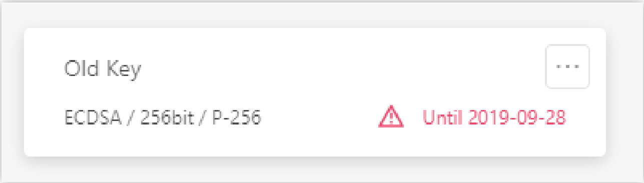
When a key slot does have a key, the following three sets of information are displayed.

1. Key name
2. Cryptographic algorithm and key length
3. Lifetime of the key

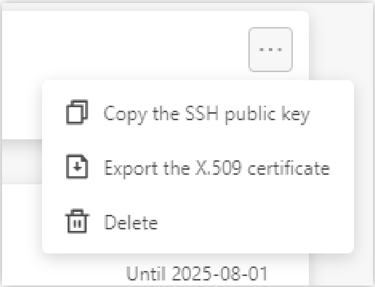
This information is displayed as shown in the following figure.



When the key has expired, the lifetime is displayed in red as follows:

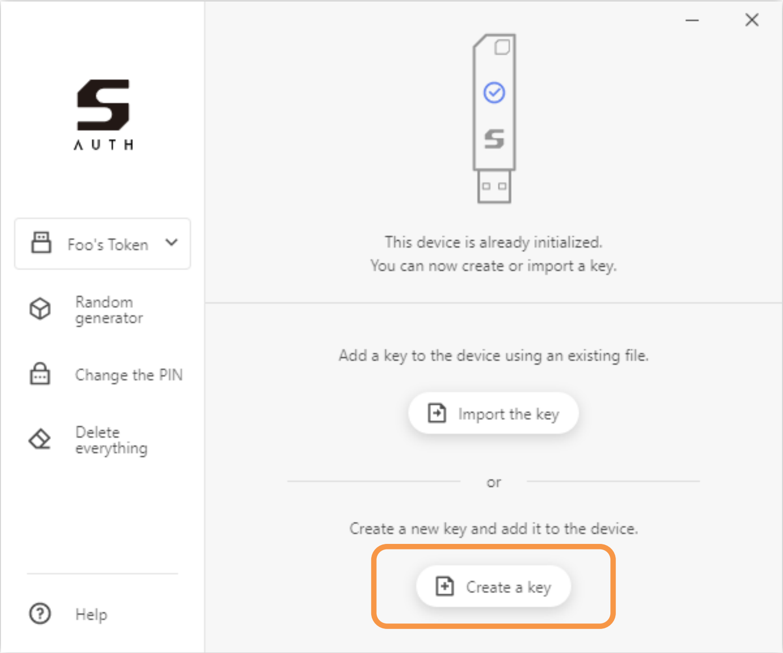


You can perform actions on the key from the menu shown in the following figure, which is displayed by clicking [···].



4.3 Generating a new key

SHALO Keyring has the capability to create keys, which can be stored in SHALO AUTH in combination with the X.509 certificate. To do this, in SHALO Keyring, click [**Create a key**].



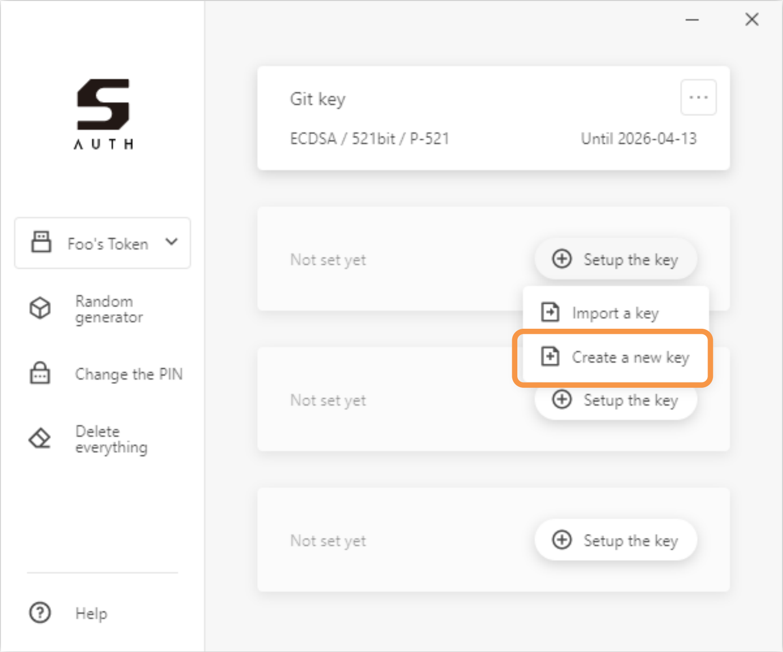
This device is already initialized.
You can now create or import a key.

Add a key to the device using an existing file.

Create a new key and add it to the device.

Figure 29 Creating a key in SHALO AUTH immediately after the setup

If SHALO Keyring looks as shown in the following figure, click [**Setup the key**] and then [**Create a new key**] in an undefined slot field.



Git key
ECDSA / 521bit / P-521 Until 2026-04-13

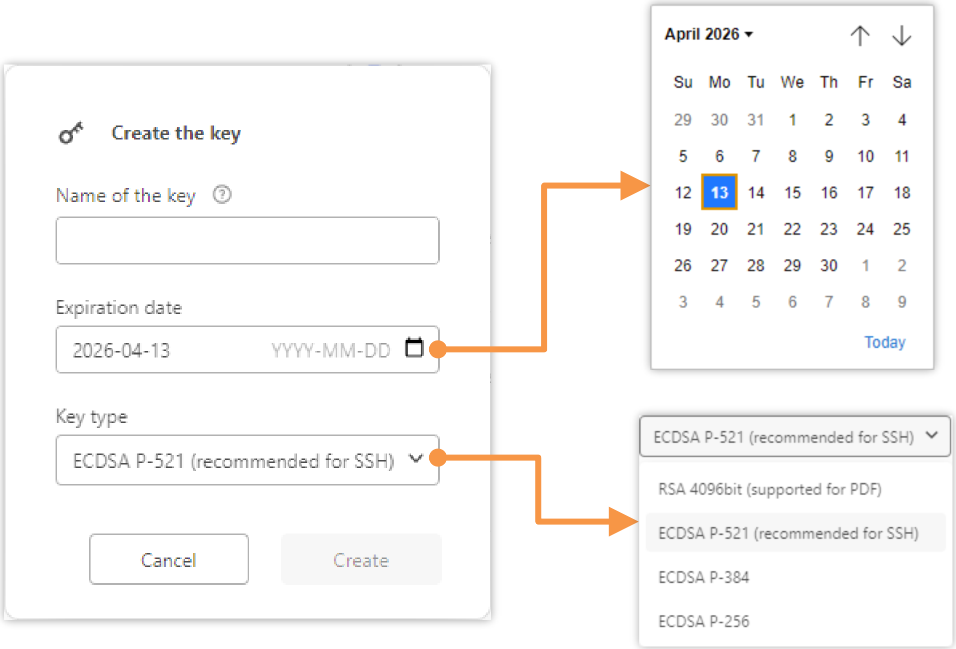
Not set yet

Not set yet

Not set yet

Figure 30 Creating a key with a storage location specified for it

Specify information for the key you create in the key generation window shown in the figure below. Then, click [**Create**], and enter your user PIN when prompted. The key is created if the user PIN is successfully authenticated.



Create the key

Name of the key

Expiration date: 2026-04-13

Key type: ECDSA P-521 (recommended for SSH)

Figure 31 The key generation window

Key name

The name for identifying the key you create. It is displayed in SHALO Keyring and is also used in the X.509 certificate as the subject field, which is the name to which the certificate belongs.

Key lifetime

Specify the lifetime of the key. You can either enter a year, month, and day in YYYY-MM-DD format, or click the icon on the right and select the date in the calendar that appears.

This lifetime is used as an expiry date for the X.509 certificate in the public key. The lifetime of the key takes effect only in applications that support X.509 certificates.

Key type

Specify a cryptographic algorithm of the key to be create. In general, select ECDSA P-521. In the ECDSA, the security strength decreases in the order of P-521, P-384, and P-256.

Select RSA for keys that encrypt or sign PDF files or that are used in an environment where the ECDSA is unavailable.

4.4 Importing an existing key

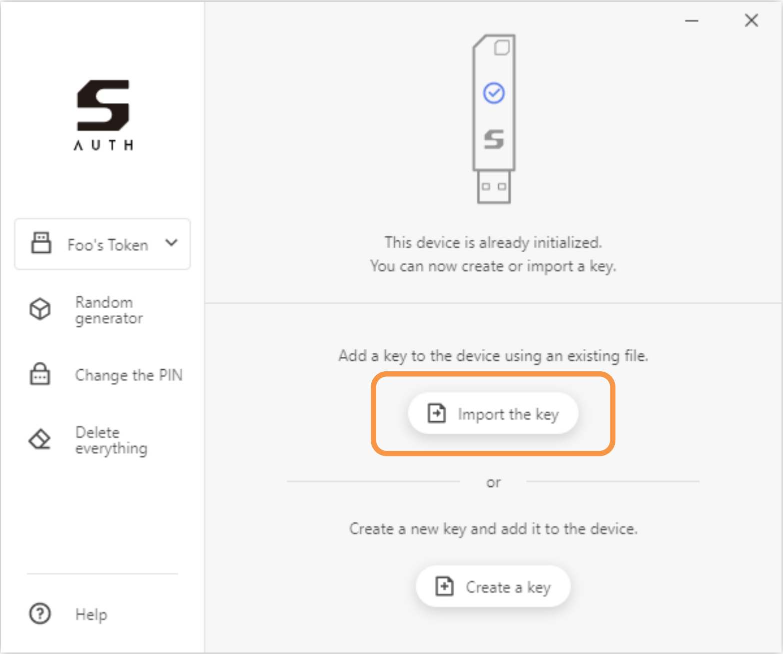
SHALO Keyring can load a key from a file and import it to SHALO AUTH. When doing so, the tool creates the X.509 certificate for the key and stores the certificate in SHALO AUTH together with the key.

SHALO Keyring supports the three types of key data formats listed in the following table.

Data format	Extension	Description
PEM	.pem	**For an RSA key:** It is text that begins with "`-----BEGIN RSA PRIVATE KEY-----`" and ends with "`-----END RSA PRIVATE KEY-----`". **For an ECDSA key:** It is text that begins with "`-----BEGIN EC PRIVATE KEY-----`" and ends with "`-----END EC PRIVATE KEY-----`".
OpenSSH	.pem	It is text that begins with "`-----BEGIN OPENSSH PRIVATE KEY-----`" and ends with "`-----END OPENSSH PRIVATE KEY-----`".
PuTTY	.ppk	It is a key file generated with puttygen, which comes with PuTTY.

If you want to import a key file other than what is listed in the above table, see Section 11.3 and convert it into the PEM format.

To import a key from a file into SHALO AUTH, click [**Import the key**] in SHALO Keyring.



This device is already initialized.
You can now create or import a key.

Add a key to the device using an existing file.

Create a new key and add it to the device.

Figure 32 Importing a key into SHALO AUTH immediately after the setup

If SHALO Keyring appears as shown in the following figure, in an undefined slot field, click [**Setup the key**] and then [**Import a key**].

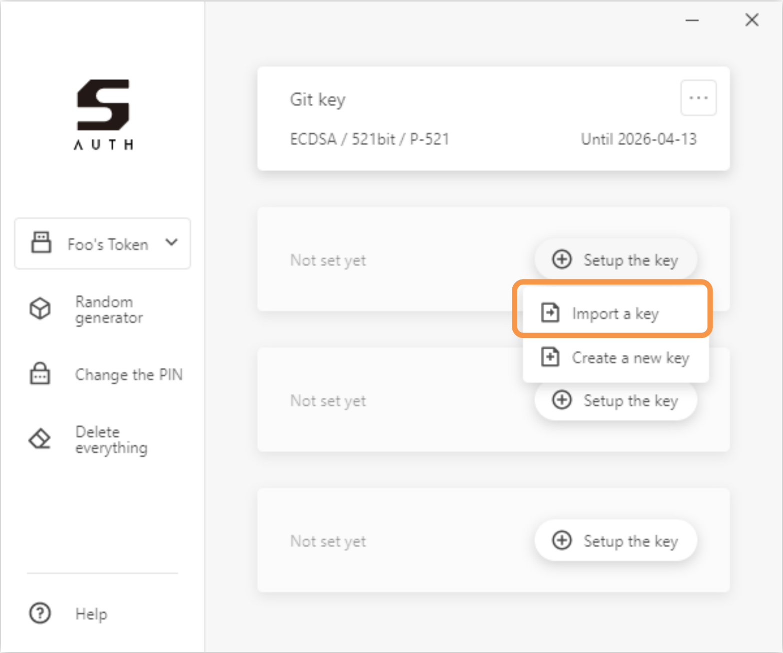


Figure 33 Importing a key with a storage location specified for it

Either way, the window shown in the figure below will appear. Drop a key file into the box, or click [**Open a file**] to select the key file.

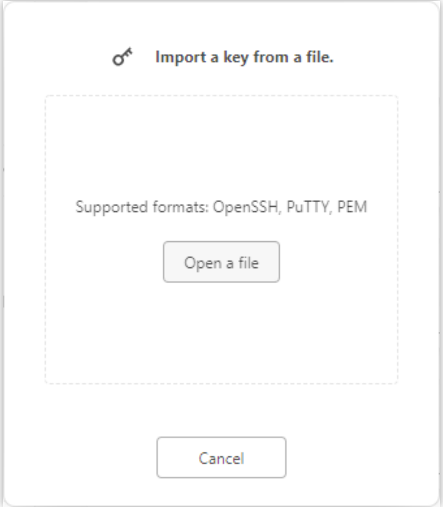
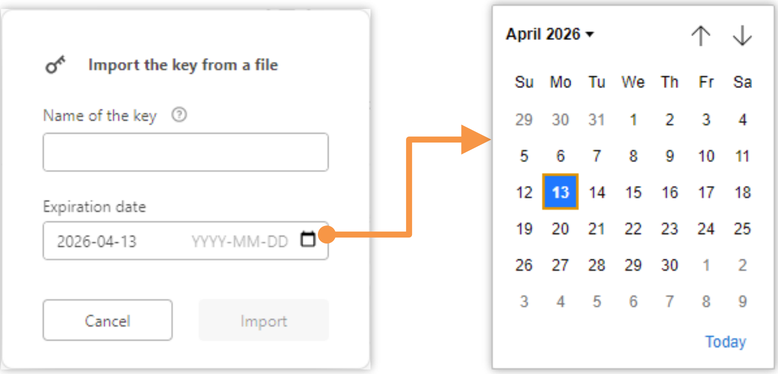

Import a key from a file.
Supported formats: OpenSSH, PuTTY, PEM
Open a file
Cancel

Figure 34 Importing a key file

When the specified key file is encrypted with a passphrase, you will be prompted to enter the passphrase.

In the window shown in the figure below, specify the information to add when the key is imported into SHALO AUTH. After specifying it, click [**Import**] and enter your user PIN to import the key into SHALO AUTH.



Import the key from a file

Name of the key

Expiration date: 2026-04-13 YYYY-MM-DD

April 2026

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Figure 35 Additional information of the key to be imported

Key name

The name to identify the key you will import. It is displayed in SHALO Keyring and is also used in the X.509 certificate as the subject field, which is the name to which the certificate belongs.

Key lifetime

Specify the lifetime of the key. You can either enter a year, month, and day in YYYY-MM-DD format, or click the icon on the right and select the date in the calendar that appears.

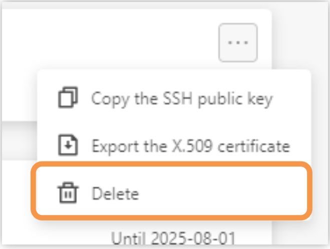
This lifetime is used as an expiry date for the X.509 certificate in the public key. The lifetime of the key takes effect only in applications that support X.509 certificates.

4.5 Removing a key

In SHALO Keyring, there are two ways to remove a key, including:

- Specifying a key slot and removing the key
- Erasing all data

Specifying a key slot and removing the key



In the key slot you want to empty, click [···], and in the menu, select [**Delete**]. Enter your user PIN when prompted. The key is removed if the user PIN is successfully authenticated.

Erasing all data

On the left side of the window, click [**Delete everything**]. This action will also remove PKCS #11 objects saved by applications other than SHALO Keyring.

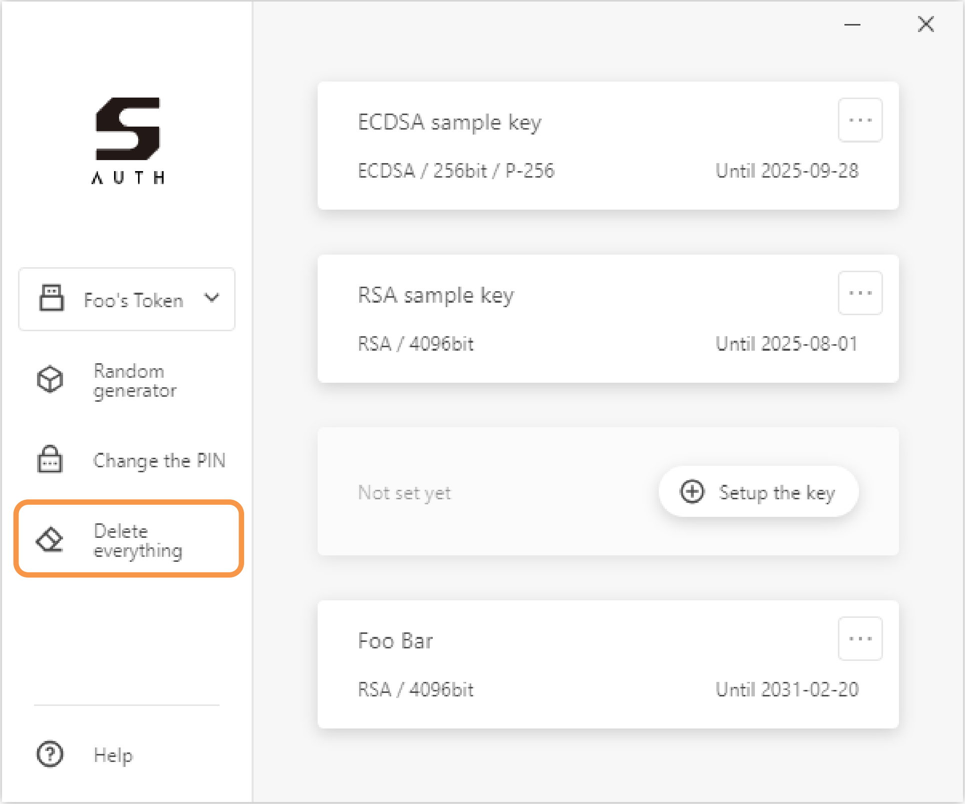
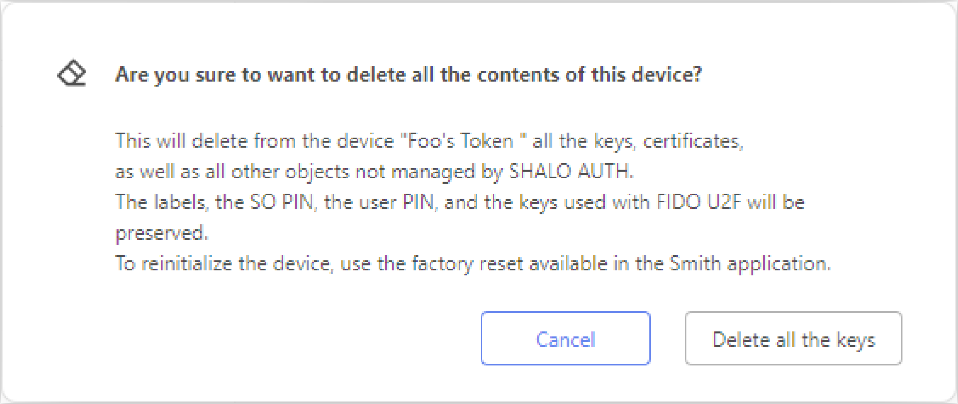


Figure 36 Performing [**Delete everything**]

The actions above do not affect the labels, SO PIN, user PIN, or keys used in FIDO U2F.

Read the warning below, and if you are sure you want to erase the keys, click [**Delete all the keys**]. Enter your user PIN when prompted. All the keys are erased if the user PIN is successfully authenticated.



Are you sure to want to delete all the contents of this device?

This will delete from the device "Foo's Token " all the keys, certificates,
as well as all other objects not managed by SHALO AUTH.
The labels, the SO PIN, the user PIN, and the keys used with FIDO U2F will be preserved.
To reinitialize the device, use the factory reset available in the Smith application.

Figure 37 Warning before [**Delete everything**]

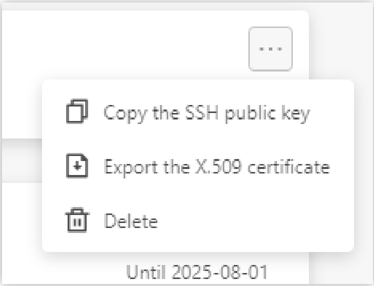
4.6 Obtaining a public key

SHALO Keyring provides the capability to obtain public keys in the following two ways:

- Public key in a data format used by SSH
- X.509 certificate

An SSH public key can be obtained only when the cryptographic algorithm for the key is RSA or one of P-521, P-384, and P-256.

In any case, click [⋯] for the key slot to open the menu shown in the following figure.



SSH public key

An SSH public key contains text data that starts with one of the following:

- `ssh-rsa`
- `rsa-sha2-256`
- `rsa-sha2-512`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

From the menu, select [**Copy the SSH public key**] to copy the SSH public key to the clipboard. You can then use it by pasting it to other software.

X.509 certificate

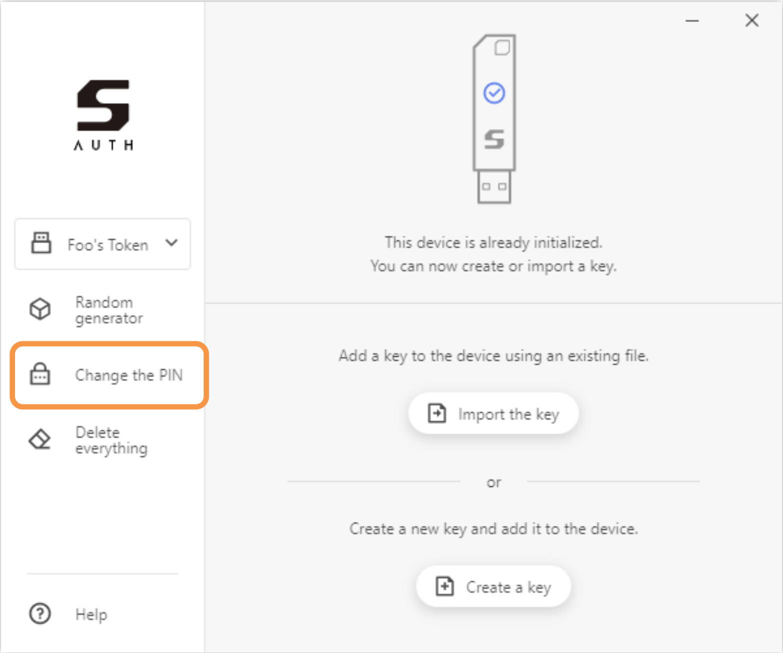
An X.509 certificate can be saved to a file in PEM format.

In the menu, click [**Export the X.509 certificate**], specify the name of the target file, and save the certificate to the file.

The certificate is a text-formatted file that begins with "`-----BEGIN CERTIFICATE-----`" and ends with "`-----END CERTIFICATE-----`".

4.7 Changing the user PIN

To change the user PIN, on the left side of the window, click [**Change the PIN**].



This device is already initialized.
You can now create or import a key.

Add a key to the device using an existing file.

Create a new key and add it to the device.

Figure 38 Performing [**Change the PIN**]

In the window shown in the following figure, enter the current user PIN and a new user PIN, and then click [**Change**].

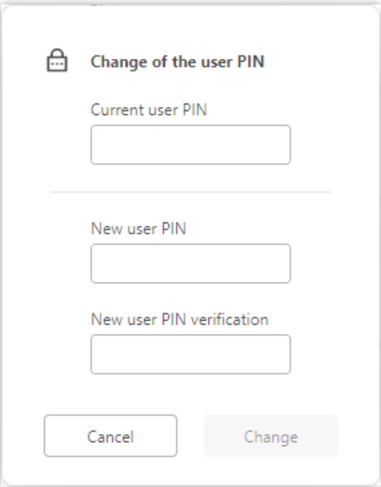


Figure 39 Change of the user PIN window

When locked, the user PIN cannot be unlocked in the Change of the user PIN window. Use SHALO Smith to reset the user PIN (Section 5.4).

4.8 Generating a password or random number sequence

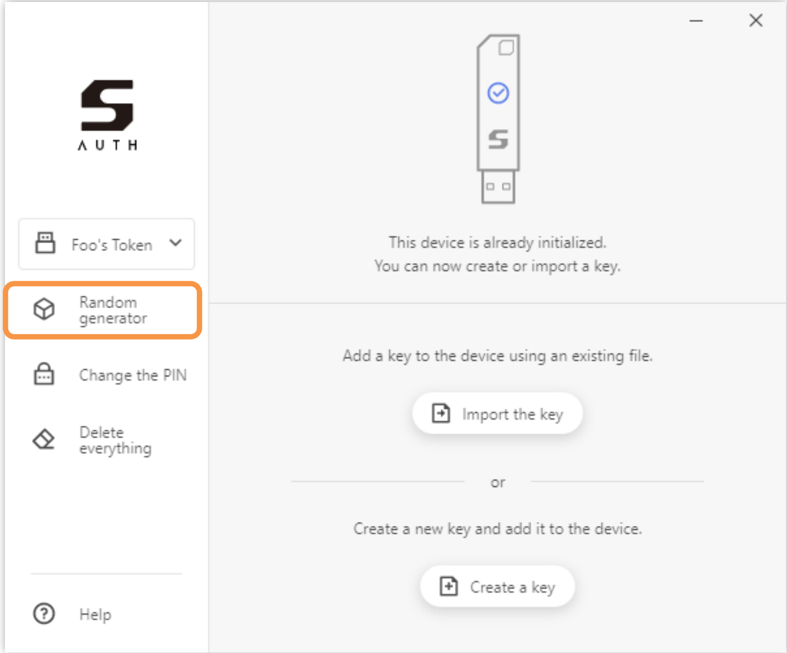
SHALO Keyring can generate a password or random number sequence by using SHALO AUTH's hardware random number generator. The following table shows random number generation and output conditions by use.

Used for	Specifiable conditions	Generated quantities	Delimiter
Password	1 to 64 characters long Whether to allow uppercase characters, lowercase characters, numbers, and symbols, respectively.	Up to 8 passwords	Newline
Integer value	Minimum value: -32768 to +32767 Maximum value: -32768 to +32767	Up to 32 rows and 32 columns in tabular format	No delimiter Comma Space Tab character
Hexadecimal string	Bit length: 1 to 64 bits Whether the string is prefixed with "0x"	Up to 16 rows and 16 columns in tabular format	No delimiter Comma Space Tab character

When symbols are enabled for the password, the following characters become available:
~ ! @ # $ % ^ & * ( ) _ + - = { } [ ] \ | : ; " ' < > , . ? /

Method

To generate a random number, on the left of the window, click [**Random generator**].



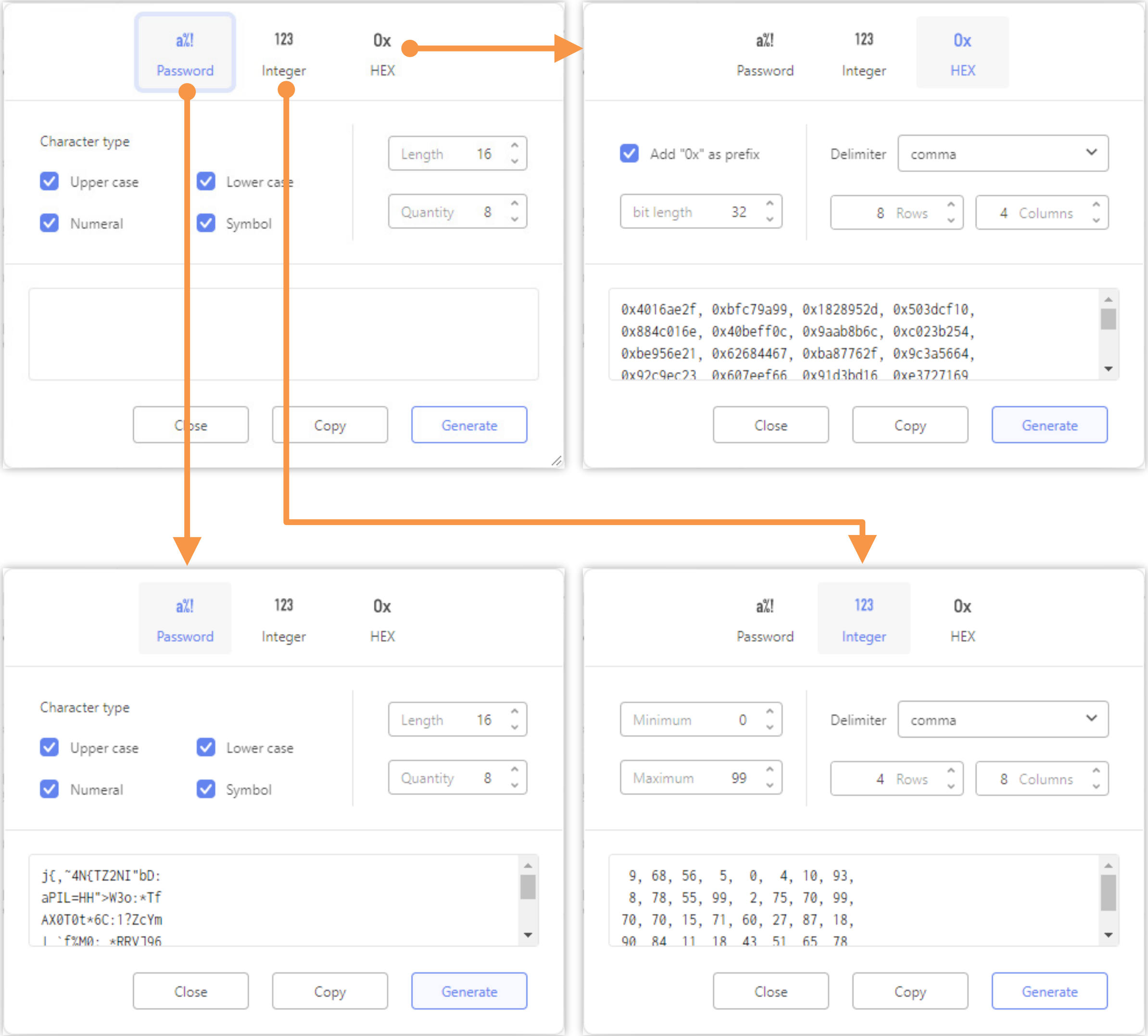
This device is already initialized.
You can now create or import a key.

Add a key to the device using an existing file.

Create a new key and add it to the device.

Figure 40 Performing [**Random generator**]

In the window shown in Figure 41, specify the purpose of the random number and the generation conditions. Clicking [**Generate**] generates and shows a random number in the window. Clicking [**Copy**] copies the whole random number being displayed to the clipboard.



Character type
☑ Upper case ☑ Lower case
☑ Numeral ☑ Symbol
Length 16
Quantity 8

☑ Add "0x" as prefix
Delimiter comma
bit length 32
8 Rows
4 Columns

```
0x4016ae2f, 0xbfc79a99, 0x1828952d, 0x503dcf10,
0x884c016e, 0x40beff0c, 0x9aab8b6c, 0xc023b254,
0xbe956e21, 0x62684467, 0xba87762f, 0x9c3a5664,
0x92c9ec23  0x607eef66  0x91d3bd16  0xe3727169
```

Character type
☑ Upper case ☑ Lower case
☑ Numeral ☑ Symbol
Length 16
Quantity 8

```
j{,~4N{TZ2NI"bD:
aPIL=HH">W3o:*Tf
AX0T0t*6C:1?ZcYm
L `f%M0: *RRV]96
```

Minimum 0
Maximum 99
Delimiter comma
4 Rows
8 Columns

```
 9, 68, 56,  5,  0,  4, 10, 93,
 8, 78, 55, 99,  2, 75, 70, 99,
70, 70, 15, 71, 60, 27, 87, 18,
90  84  11  18  43  51  65  78
```

Figure 41 Random number generation window and examples of generating a random number

4.9 CKA_ID attribute of key data

SHALO Keyring reserves four CKA_ID attributes shown below for the key slots. When storing a private key, public key, or X.509 certificate in SHALO AUTH, SHALO Keyring assigns a CKA_ID attribute that corresponds to the destination key slot.

Key slot	CKA_ID attribute (hexadecimal number)	CKA_ID attribute (string)
Key slot 1	41 58 54 4F 4F 4C 4B 45 59 23 31	AXTOOLKEY#1
Key slot 2	41 58 54 4F 4F 4C 4B 45 59 23 32	AXTOOLKEY#2
Key slot 3	41 58 54 4F 4F 4C 4B 45 59 23 33	AXTOOLKEY#3
Key slot 4	41 58 54 4F 4F 4C 4B 45 59 23 34	AXTOOLKEY#4

If you separately manage data with other PKCS #11-compliant software, do not use these reserved CKA_ID attributes.
If you use the reserved CKA_ID attribute, the data may be manipulated by SHALO Keyring, or the data may not be manipulated correctly by SHALO Keyring.

Chapter 5

Using the SHALO Smith administration tool

This chapter explains the administration tool, SHALO Smith. SHALO Smith is a software program dedicated to issuance and management tasks for SHALO AUTH.

Before transferring or disposing of SHALO AUTH, you must use SHALO Smith to restore the device to the factory settings.

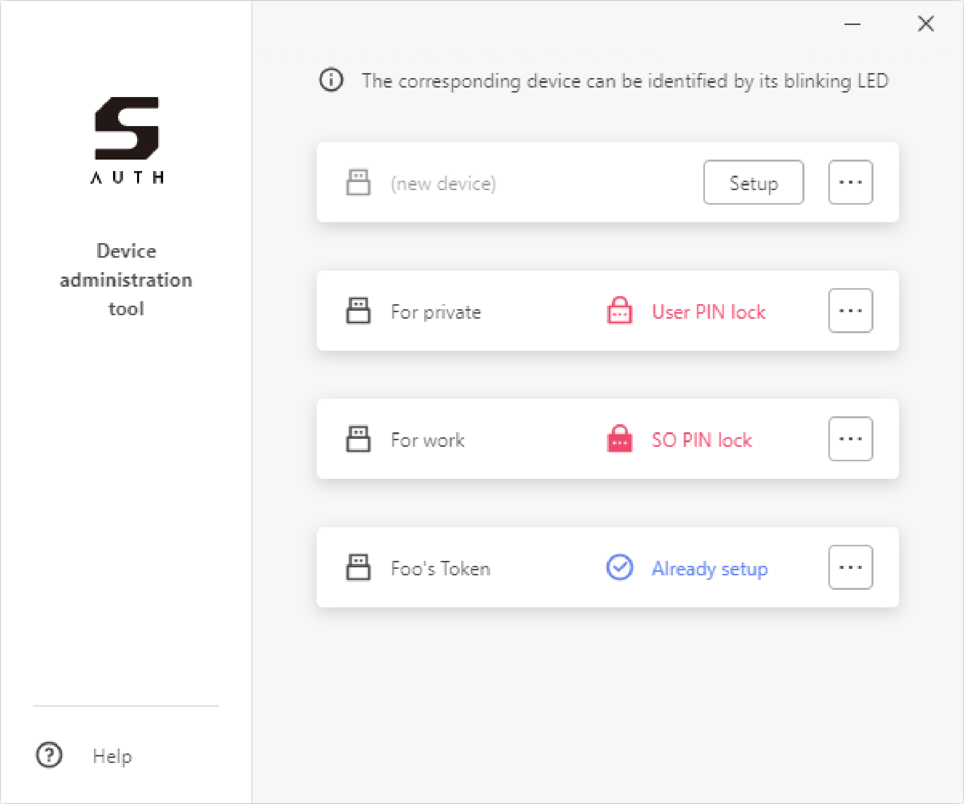
Topics in this chapter

1. Viewing the state of SHALO AUTH
2. Setting up SHALO AUTH
3. Restoring SHALO AUTH to the factory settings
4. Resetting the user PIN
5. Changing the SO PIN

5.1 Viewing the state of SHALO AUTH

SHALO Smith can show and manage up to eight SHALO AUTH devices connected to a PC.

Figure 42 shows a SHALO Smith window. In this example, one new SHALO AUTH device and the three others you have previously set up are connected to the PC.

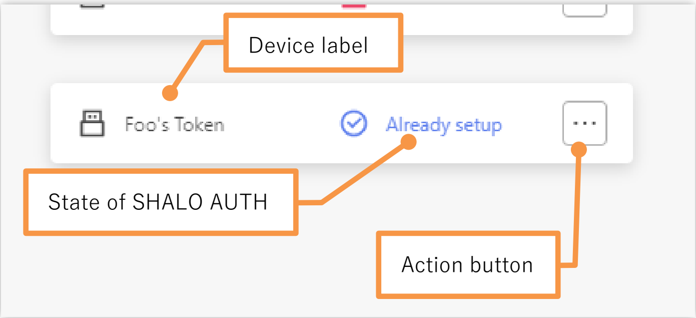


The corresponding device can be identified by its blinking LED

(new device)		Setup	···
For private	User PIN lock		···
For work	SO PIN lock		···
Foo's Token	Already setup		···

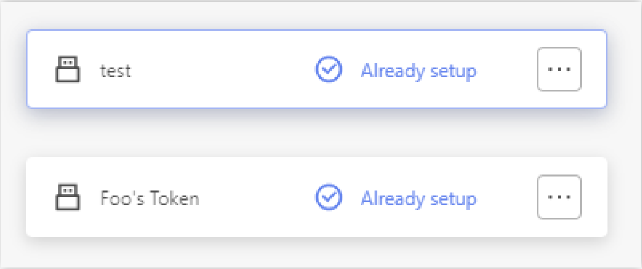
Figure 42 Four SHALO AUTH devices displayed by SHALO Smith

Each of the vertically aligned white boxes indicates a single SHALO AUTH device. You can see the device label on the left side in the box and the state of the device on the right, as shown in the following figure.



Identifying SHALO AUTH

When you click a box with the mouse, the box now appears in a light color as shown in the following figure, and the LED for the corresponding SHALO AUTH device flashes.



State of SHALO AUTH

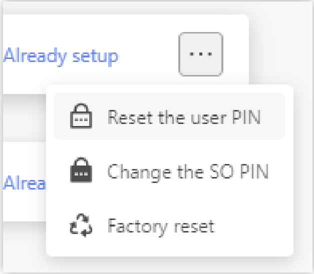
The following table shows the states that SHALO AUTH can take.

State indicated	Description
[Setup] button	The device is not set up. You can set it up by clicking the button (Section 5.2).
[Already setup]	The device is set up and in the normal state.
[User PIN lock]	The device is set up, but the user PIN is locked. To unlock it, you need to reset the user PIN (Section 5.4).
[SO PIN lock]	The device is set up, but the SO PIN is locked. Restoring it to the factory settings (Section 5.3) is the only way to unlock it.

Actions for SHALO AUTH

You can set up a new SHALO AUTH device by clicking [**Setup**].

When you click [···] on the far right, the menu appears as shown in the figure to the right, enabling you to perform actions related to administration of SHALO AUTH.



The only action you can perform on a new SHALO AUTH device is restoring it to the factory settings.
Doing so will remove all FIDO authentication keys used for U2F.

5.2 Setting up SHALO AUTH

During the setup process, the tool initializes the data area for general security key functionality and configures the following management information. SHALO Keyring can also be used for setup.

Device label An individual name used to identify multiple SHALO AUTH.

User PIN The password for when the user uses the device. It allows the use of protected private keys.

SO PIN The password for management. It is used to reset the user PIN or to restore SHALO AUTH to the factory settings.

This setup process does not affect any U2F security key functionality.
If you have registered SHALO AUTH as a U2F security key in a Web service before the above setup process, you can still continue to use the device in that service.

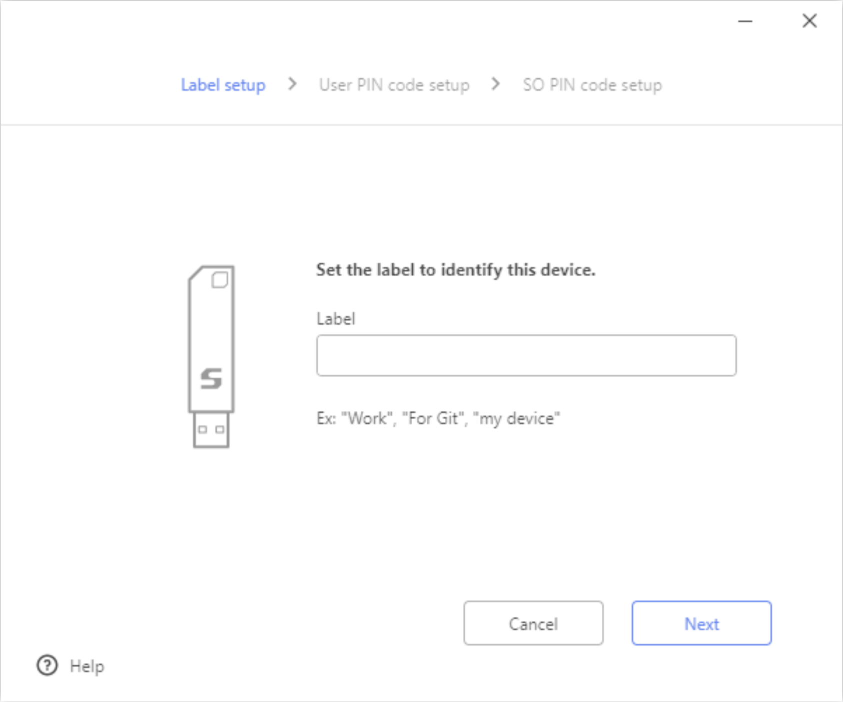
If you have previously used a SHALO AUTH device and want to set it up again, you must restore the device to the factory settings. When the device is restored to its factory settings, the U2F security key information in it is also removed.

Procedure

Click [**Setup**] to start the SHALO AUTH setup process. During the setup process, configure the device label, user PIN, and SO PIN in this order.

Specifying the device label

The device label can include alphanumeric characters and symbols as well as character strings in Japanese and other languages. The maximum number of characters in the label depends on the types of characters. If the label is too long, you will see a warning.



Set the label to identify this device.

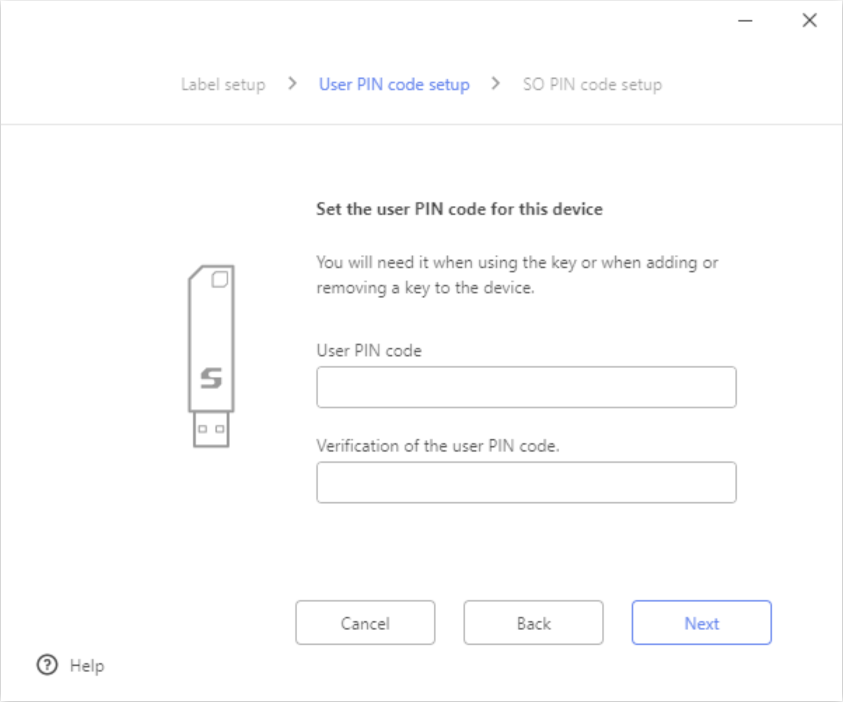
Label

Ex: "Work", "For Git", "my device"

Figure 43 Specifying the device label

Specifying the user PIN

The user PIN can include alphanumeric characters and symbols. Specify a user PIN between 4 and 256 characters long. Enter the user PIN twice for confirmation.



Set the user PIN code for this device

You will need it when using the key or when adding or removing a key to the device.

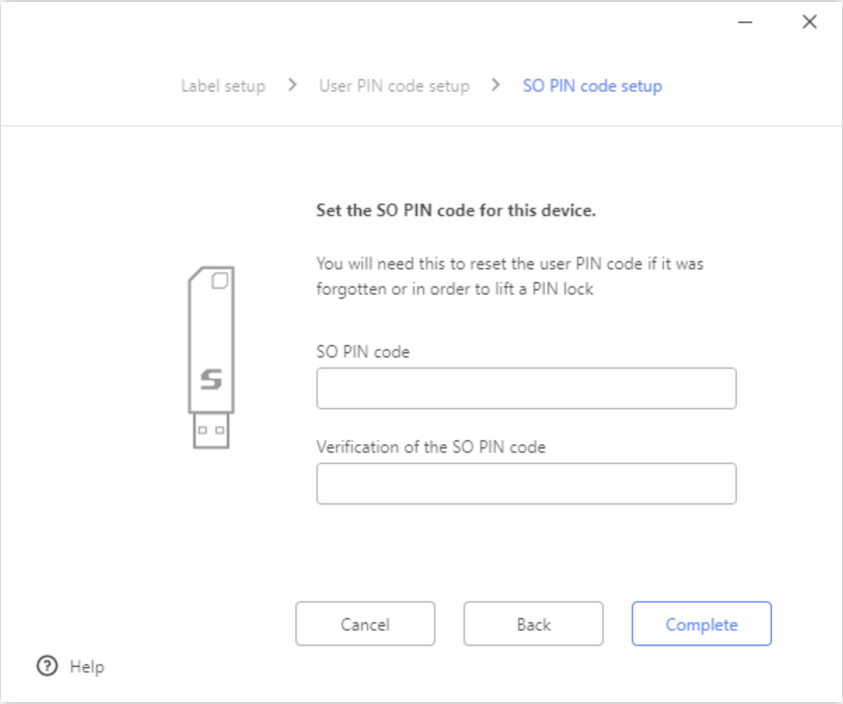
User PIN code

Verification of the user PIN code.

Figure 44 Specifying the user PIN

Specifying the SO PIN

The SO PIN can include alphanumeric characters and symbols. Specify a SO PIN between 4 and 256 characters long. Enter the SO PIN twice for confirmation.



Set the SO PIN code for this device.

You will need this to reset the user PIN code if it was forgotten or in order to lift a PIN lock

SO PIN code

Verification of the SO PIN code

Figure 45 Specifying the SO PIN

5.3 Restoring SHALO AUTH to the factory settings

When you restore SHALO AUTH to the factory settings, **all the following information is removed:**

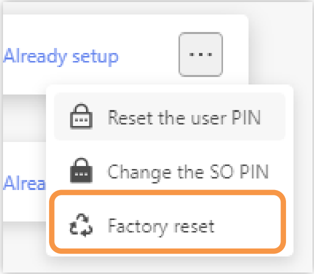
- SO PIN
- User PIN
- Device label
- All PKCS #11 data
- All FIDO authentication keys

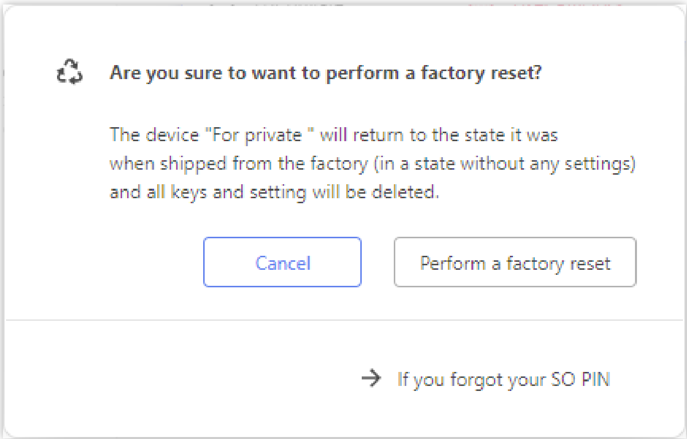
Restoring a device to the factory settings also removes all the FIDO authentication keys used for U2F in it. Therefore, the device is recognized as an unregistered one even by Web services for which the device was previously on their list.

We **strongly recommend** that you restore SHALO AUTH to the factory settings before transferring or disposing of the device. Doing so can prevent a malicious user from being authenticated through the transferred SHALO AUTH device in any Web services where the device is registered as the two-factor authentication device.

Procedure

1. Click [⋯] on the right side of the target device.
2. From the menu, select [**Factory reset**].
3. In the window shown in Figure 46, click [**Perform a factory reset**].
4. Input the SO PIN and click [**Authenticate**].




Are you sure to want to perform a factory reset?

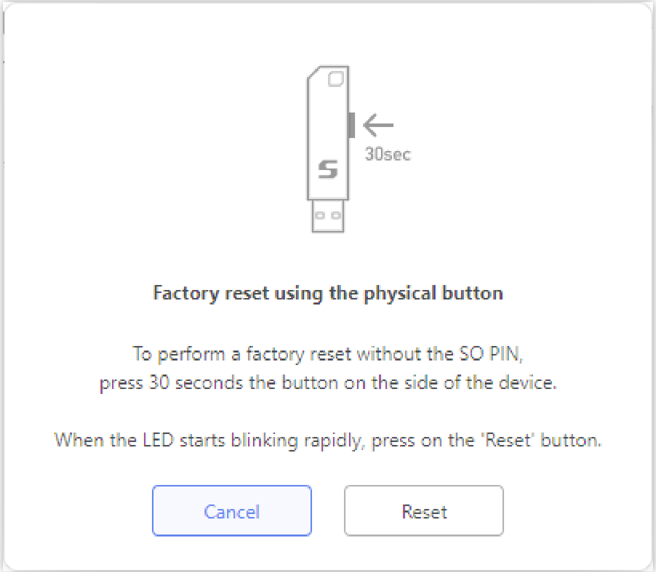
The device "For private " will return to the state it was when shipped from the factory (in a state without any settings) and all keys and setting will be deleted.

Figure 46 Factory reset window

Troubleshooting when you forgot the SO PIN

You can restore the device's factory settings without the SO PIN via the following procedure:

1. In the window shown in Figure 46, click [**If you forgot your SO PIN**].
2. After the window shown in Figure 47 appears, press and hold the button on the side of the SHALO AUTH device until its LED starts flashing rapidly. This takes about 30 seconds.
3. While the LED is flashing, click [**Reset**].



Factory reset using the physical button

To perform a factory reset without the SO PIN,
press 30 seconds the button on the side of the device.

When the LED starts blinking rapidly, press on the 'Reset' button.

Figure 47 Restoring the factory settings without the SO PIN

This is how you can restore the SHALO AUTH's factory settings even if the SO PIN is locked.

If you click [**Reset**] when the LED is not flashing, the number of possible PIN authentication attempts decreases by one because the software considers it an authentication failure with the SO PIN. Repeating this action will lock the SO PIN.

5.4 Resetting the user PIN

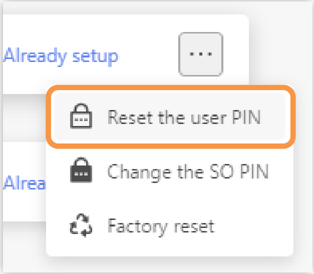
When you reset the user PIN:

- The user PIN is changed to a new one.
- The user PIN is unlocked.
- The number of possible authentication attempts before the PIN is locked is reset to five.

When resetting the user PIN, you have to enter the current SO PIN, not the current user PIN.

Procedure

1. Click [···] on the right side of the target device.
2. From the menu, select [**Reset the user PIN**].
3. In the windows shown in Figure 48, input both PINs and then click [**Reset**].



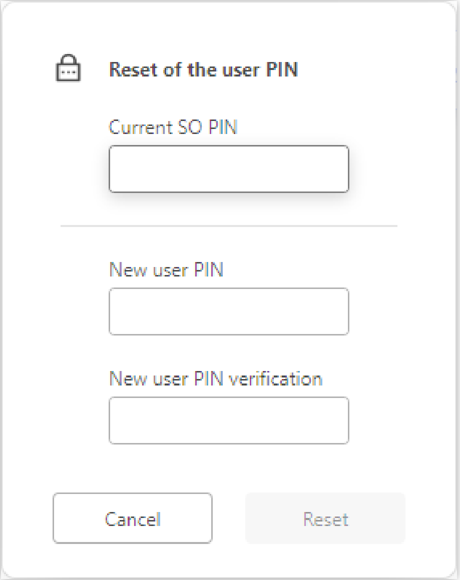
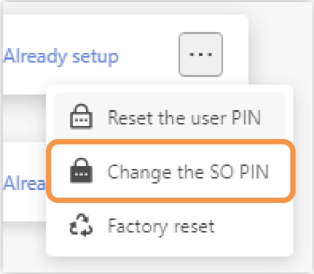


Figure 48 Resetting the user PIN

5.5 Changing the SO PIN



To change the SO PIN, use the following procedure:

1. Click [···] on the right side of the target device.
2. From the menu, select [**Change the SO PIN**].
3. Enter the current SO PIN and a new SO PIN, and then click [**Change**].

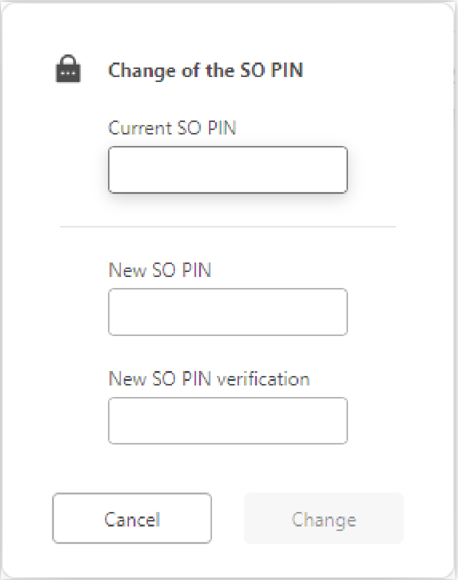


Figure 49 Changing the SO PIN

Chapter 6

Using U2F in Web services

This chapter explains how to use the U2F functionality of SHALO AUTH for two-step verification in Web services.

If you use the U2F functionality of SHALO AUTH for two-step verification, we **strongly recommend** that you use additional login methods in case loss or damage occurs. This chapter therefore assumes that the two-step verification process has been enabled in advance for each Web service's account.

For an overview of U2F, see Section 2.2.

Topics in this chapter

1. U2F settings for Google
2. U2F settings for Facebook
3. U2F settings for GitHub

6.1 U2F settings for Google

6.1.1 Registering SHALO AUTH

If the two-step verification process has already been turned on, you can add SHALO AUTH to your Google account by following the procedure below. Make sure that SHALO AUTH is disconnected from the PC.

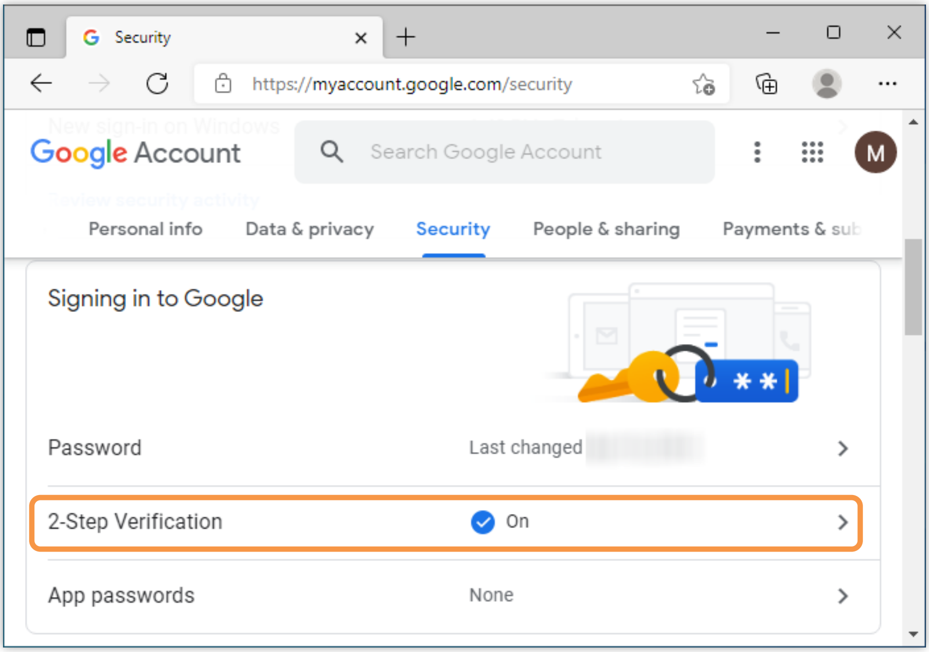
1. Open https://myaccount.google.com/ in a Web browser and log in.
2. Select [**Security**].
3. In [**Signing in to Google**], click [**2-Step Verification**].
4. On the page for the two-step verification process, click [**ADD SECURITY KEY**].
5. Click [**NEXT**].
6. Connect SHALO AUTH to the PC, wait until SHALO AUTH's LED flashes, and then press the button.
7. Input the name of the security key and click [**DONE**].
8. Log out, and check that you can log in with SHALO AUTH.

The following explains the procedure together with screenshots.

The explanations in this subsection are based on the information correct at the time of writing this manual.
Note that the website screenshots may differ from those in the manual.

Steps 1 to 3

As shown in the following figure, click [**2-Step Verification**].



Signing in to Google

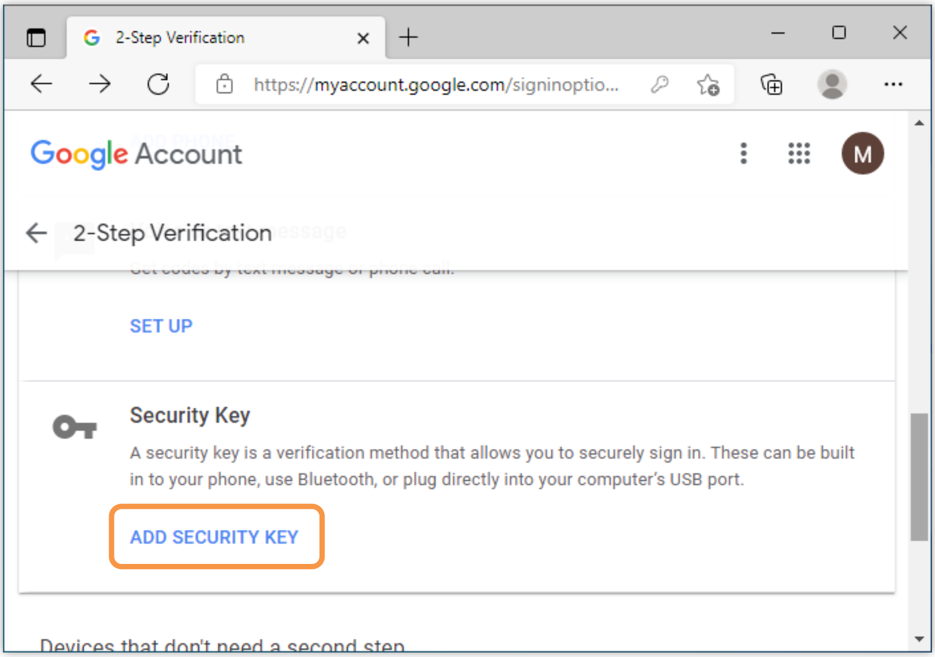
Password Last changed [redacted]

2-Step Verification On

App passwords None

Step 4

As shown in the following figure, scroll down the page, and under [**Security Key**], click [**ADD SECURITY KEY**].



Google Account

2-Step Verification

SET UP

Security Key

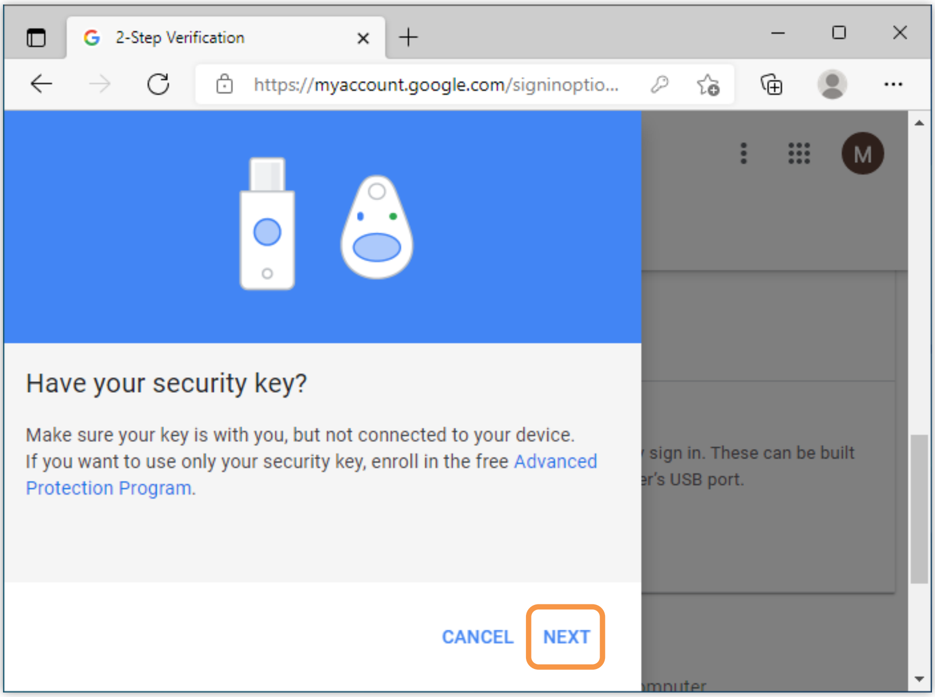
A security key is a verification method that allows you to securely sign in. These can be built in to your phone, use Bluetooth, or plug directly into your computer's USB port.

ADD SECURITY KEY

Devices that don't need a second step

Step 5

The page will appear as shown in the figure below. Click [**NEXT**].



Have your security key?

Make sure your key is with you, but not connected to your device.
If you want to use only your security key, enroll in the free Advanced Protection Program.

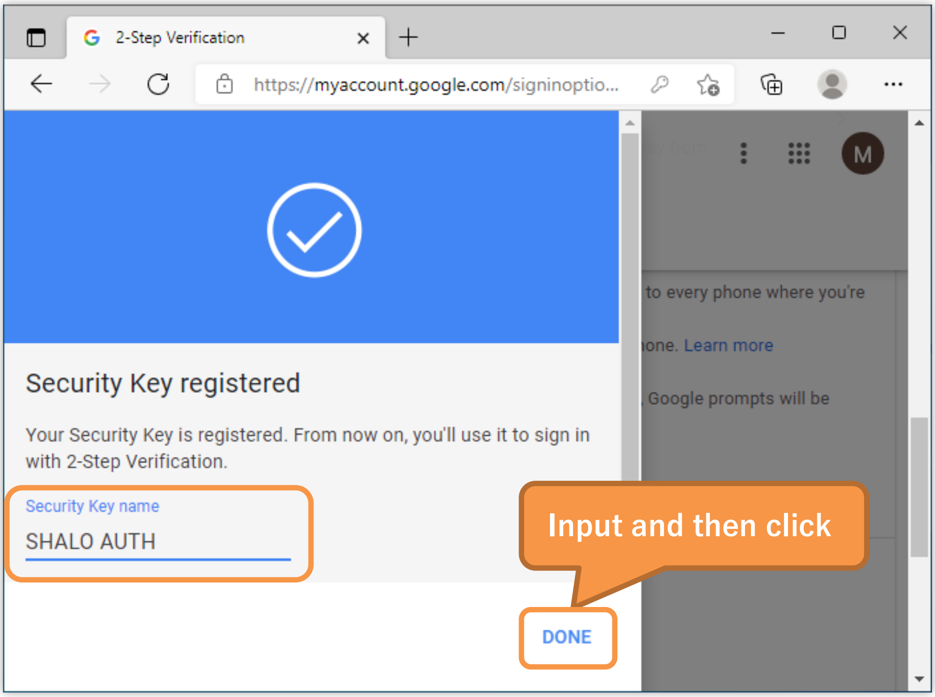
CANCEL NEXT

Step 6

Connect SHALO AUTH to the PC as instructed on the screen. Wait until SHALO AUTH's LED flashes, and press its button.

Step 7

Specify the name of the security key. This name is for identifying the key registered with your Google account. It does not affect the SHALO AUTH device itself. Input the name, and then click [**DONE**].



Security Key registered

Your Security Key is registered. From now on, you'll use it to sign in with 2-Step Verification.

Security Key name: SHALO AUTH

Step 8

Log out and then log in again to check that you can log in successfully. Input the login credentials, wait until SHALO AUTH's LED flashes, and then press its button.

6.1.2 Deregistering SHALO AUTH

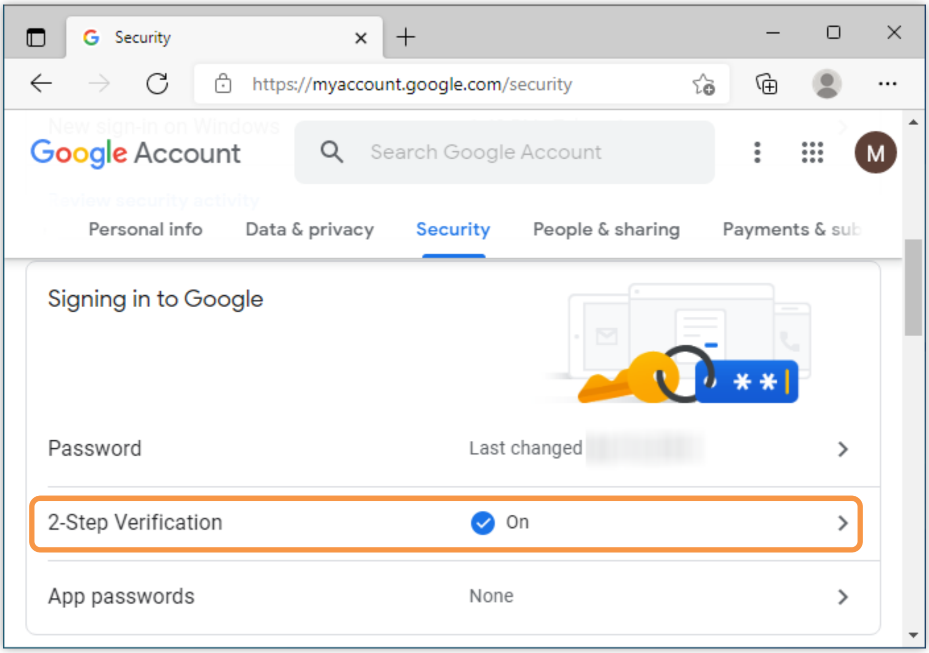
You can deregister SHALO AUTH from your Google account by using the following procedure:

1. Open https://myaccount.google.com/ in a Web browser and log in.
2. Select [**Security**].
3. In [**Signing in to Google**], click [**2-Step Verification**].
4. Click the edit icon next to the security key you want to deregister.
5. Click [**REMOVE THIS KEY**].

The following explains the procedure together with screenshots.

Steps 1 to 3

As shown in the following figure, click [**2-Step Verification**].



Signing in to Google

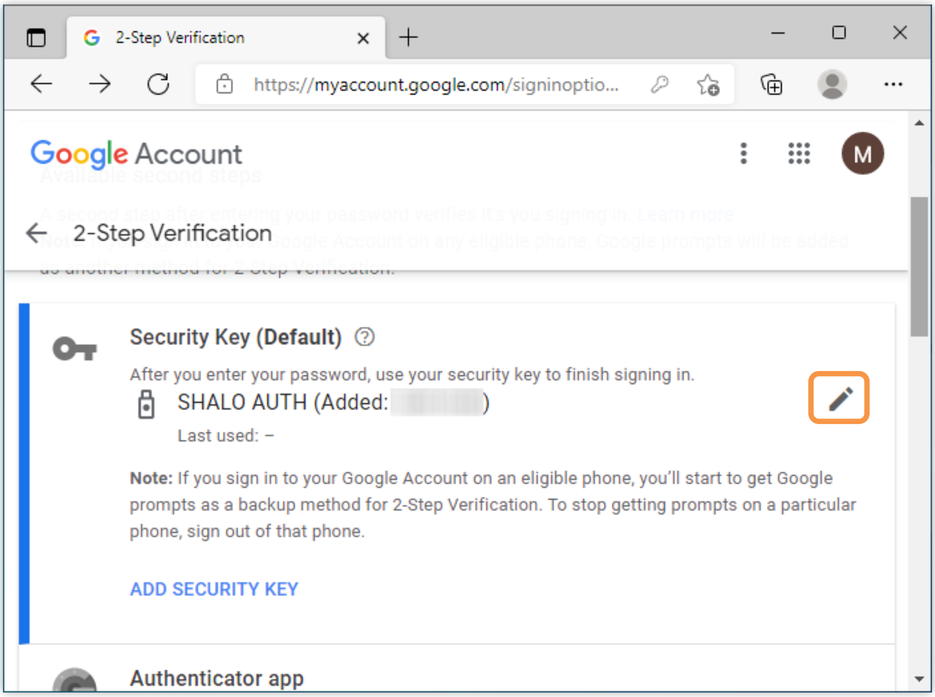
Password Last changed [redacted]

2-Step Verification ✓ On

App passwords None

Step 4

As shown in the following figure, click the edit icon next to the security key you want to remove.



2-Step Verification

Security Key (Default)

After you enter your password, use your security key to finish signing in.

SHALO AUTH (Added: [redacted])

Last used: –

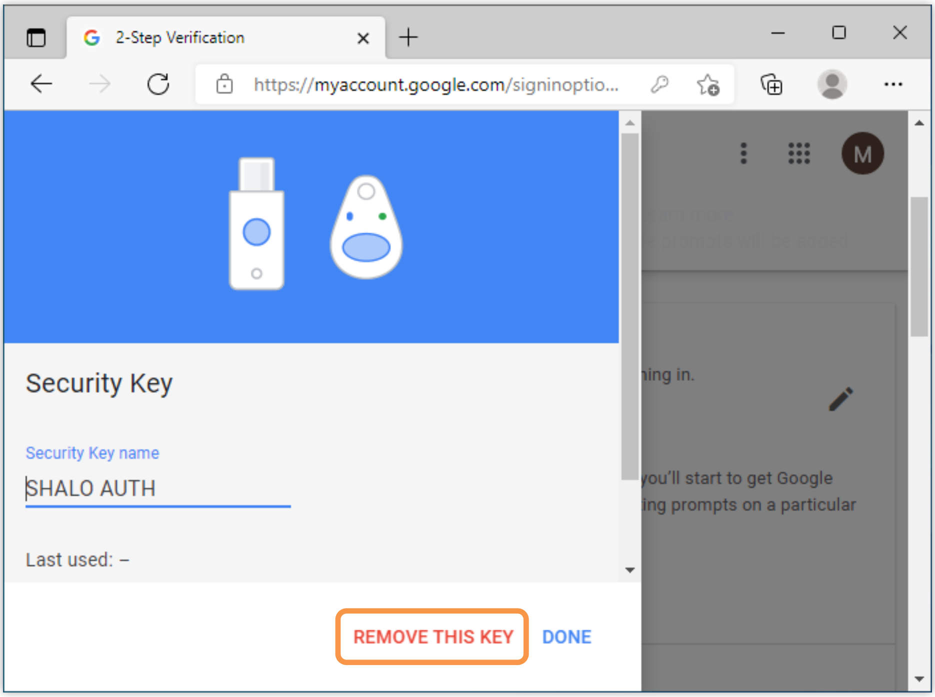
Note: If you sign in to your Google Account on an eligible phone, you'll start to get Google prompts as a backup method for 2-Step Verification. To stop getting prompts on a particular phone, sign out of that phone.

ADD SECURITY KEY

Authenticator app

Step 5

As shown in the following figure, click [**REMOVE THIS KEY**].



Security Key

Security Key name

SHALO AUTH

Last used: –

REMOVE THIS KEY DONE

6.2 U2F settings for Facebook

You can configure the two-step verification settings in the Facebook account settings page by selecting [**Security and Login**] > [**Two-factor authentication**].

6.2.1 Registering SHALO AUTH

You can register SHALO AUTH in two-step verification for your Facebook account by using the procedure below. Make sure that SHALO AUTH is disconnected from the PC.

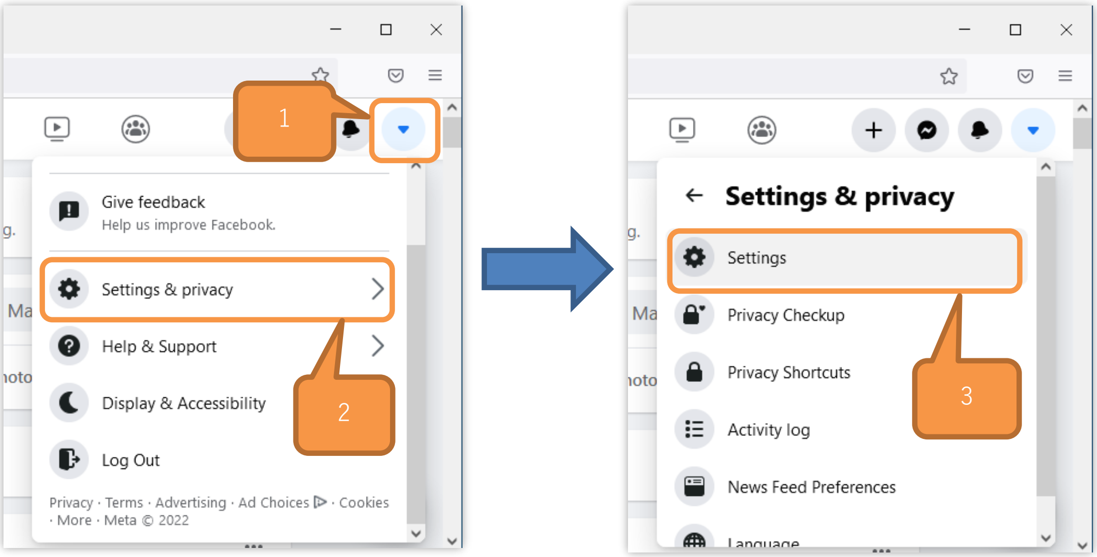
1. Log in to the Web version of your Facebook account.
2. Click the [**Account**] icon, and click [**Settings & privacy**]-[**Settings**].
3. In the [**Settings**] page, click [**Security and Login**].
4. Click [**Edit**] next to [**Use two-factor authentication**].
5. In [**Add a backup method**], click [**Setup**] for [**Security key**].
6. Connect SHALO AUTH to the PC, wait until SHALO AUTH's LED flashes, and then press its button.
7. Input the name of the security key and click [**Save**].
8. Click [**OK**].
9. Log out, and check that you can log in with SHALO AUTH.

The following explains the procedure together with screenshots.

The explanations in this subsection are based on the information correct at the time of writing this manual.
Note that the website screenshots may differ from those in the manual.

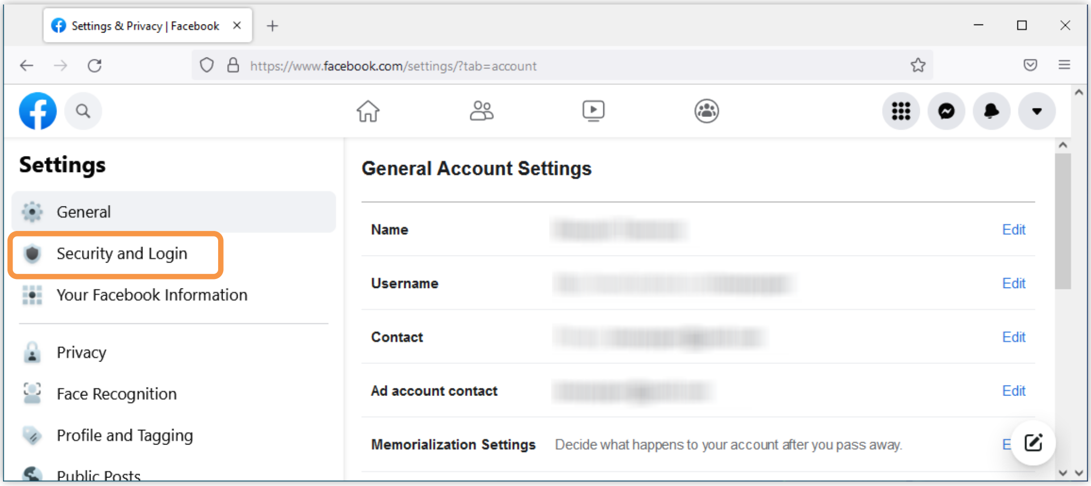
Steps 1 to 2

As shown in the following figures, click the icon and menu items in Facebook in order.



Step 3

The page will appear as shown in the figure below. Click [**Security and Login**].



Settings & Privacy | Facebook

https://www.facebook.com/settings/?tab=account

Settings

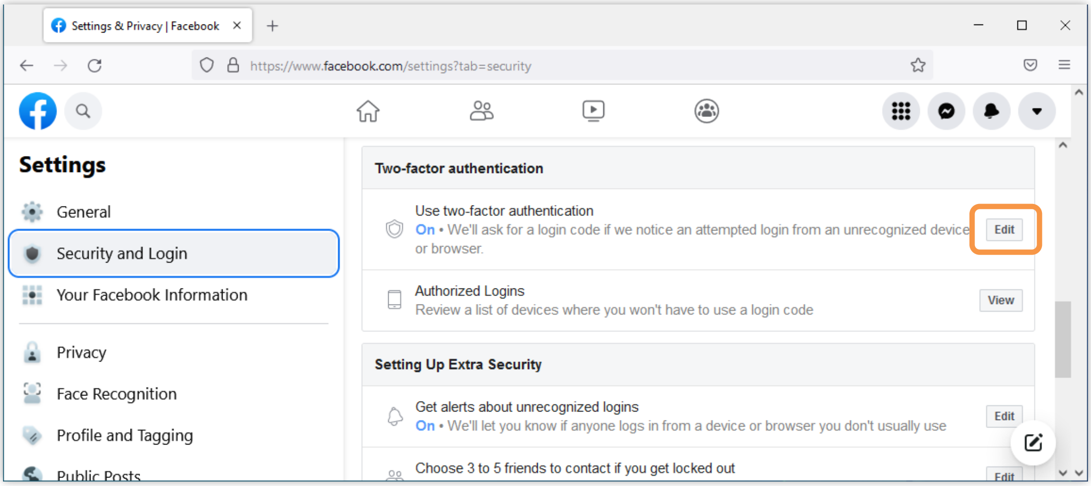
- General
- Security and Login
- Your Facebook Information
- Privacy
- Face Recognition
- Profile and Tagging
- Public Posts

General Account Settings

Name	[redacted]	Edit
Username	[redacted]	Edit
Contact	[redacted]	Edit
Ad account contact	[redacted]	Edit
Memorialization Settings	Decide what happens to your account after you pass away.	E

Step 4

Scroll down the page, and click [**Edit**] next to [**Use two-factor authentication**].



Settings & Privacy | Facebook

https://www.facebook.com/settings?tab=security

Settings

- General
- Security and Login
- Your Facebook Information
- Privacy
- Face Recognition
- Profile and Tagging
- Public Posts

Two-factor authentication

Use two-factor authentication
On • We'll ask for a login code if we notice an attempted login from an unrecognized device or browser. Edit

Authorized Logins
Review a list of devices where you won't have to use a login code View

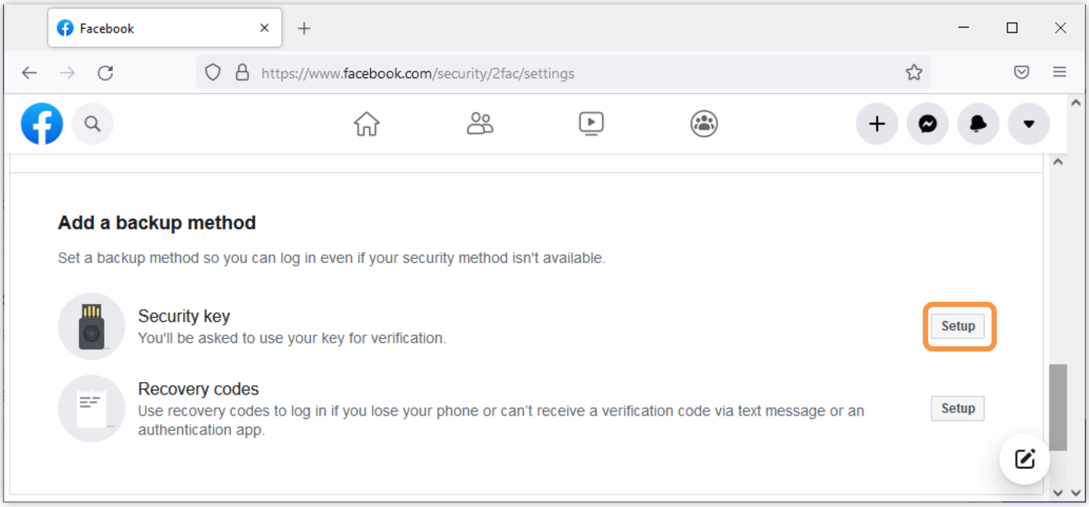
Setting Up Extra Security

Get alerts about unrecognized logins
On • We'll let you know if anyone logs in from a device or browser you don't usually use Edit

Choose 3 to 5 friends to contact if you get locked out Edit

Step 5

Scroll down the page, and in [**Add a backup method**], click [**Setup**] for [**Security key**] as shown in the following figure.



Facebook

https://www.facebook.com/security/2fac/settings

Add a backup method

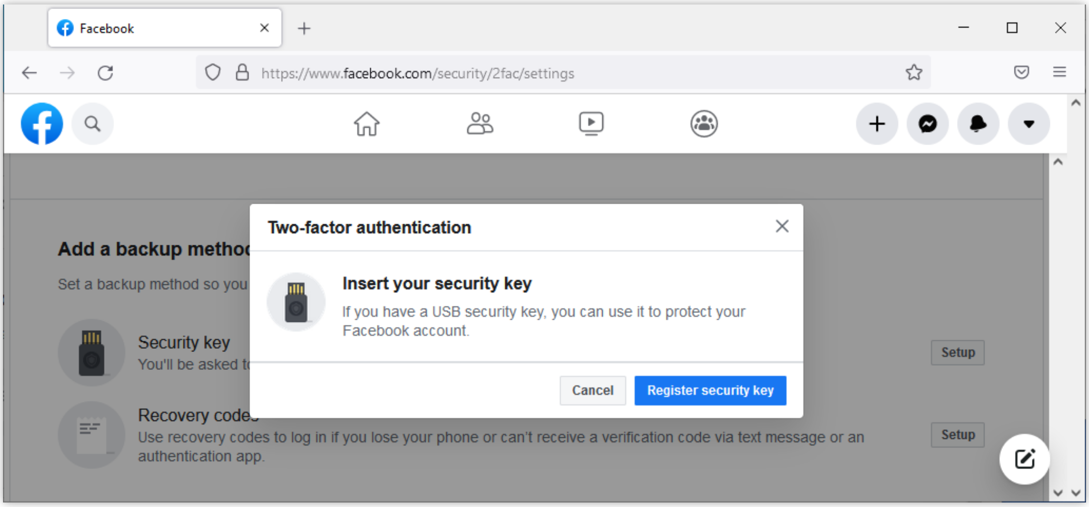
Set a backup method so you can log in even if your security method isn't available.

Security key
You'll be asked to use your key for verification.
Setup

Recovery codes
Use recovery codes to log in if you lose your phone or can't receive a verification code via text message or an authentication app.
Setup

Step 6

Connect SHALO AUTH to the PC as shown in the figure below. Wait until SHALO AUTH's LED flashes, and then press its button.



Facebook

https://www.facebook.com/security/2fac/settings

Two-factor authentication

Insert your security key

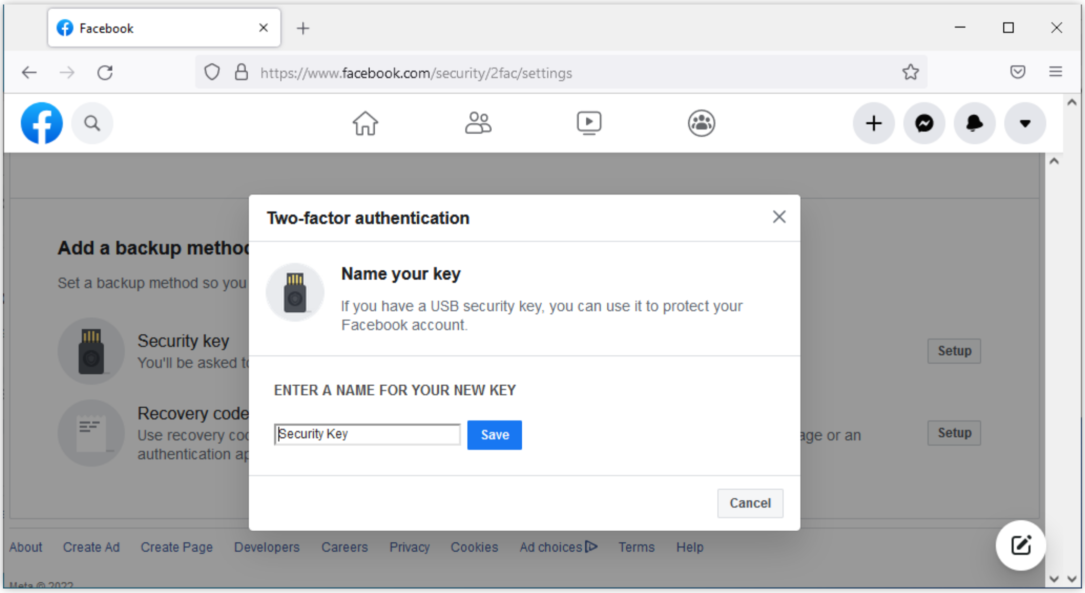
If you have a USB security key, you can use it to protect your Facebook account.

Cancel | Register security key

Step 7

When SHALO AUTH is registered successfully, the window will appear as shown in the figure below. Specify the name of the security key here. This name is for identifying the key registered with your Facebook account. It does not affect the SHALO AUTH device itself.

Input the name of the security key and click [**Save**].



https://www.facebook.com/security/2fac/settings

Two-factor authentication

Name your key

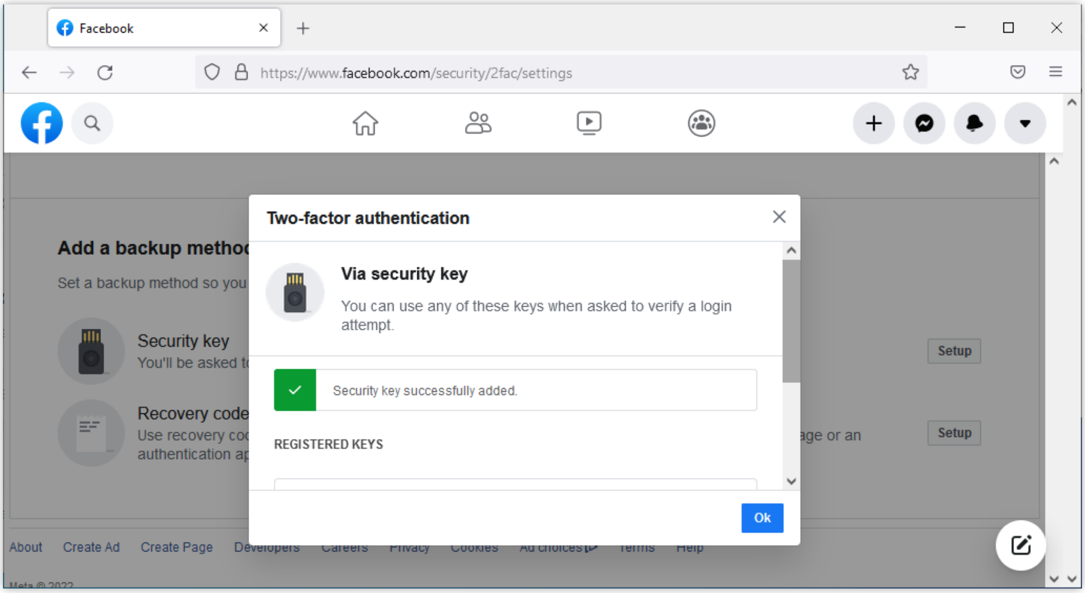
If you have a USB security key, you can use it to protect your Facebook account.

ENTER A NAME FOR YOUR NEW KEY

Security Key

Step 8

When everything is complete, the window shown in the figure below will appear. Click the [**OK**] button to close the window.



https://www.facebook.com/security/2fac/settings

Two-factor authentication

Via security key

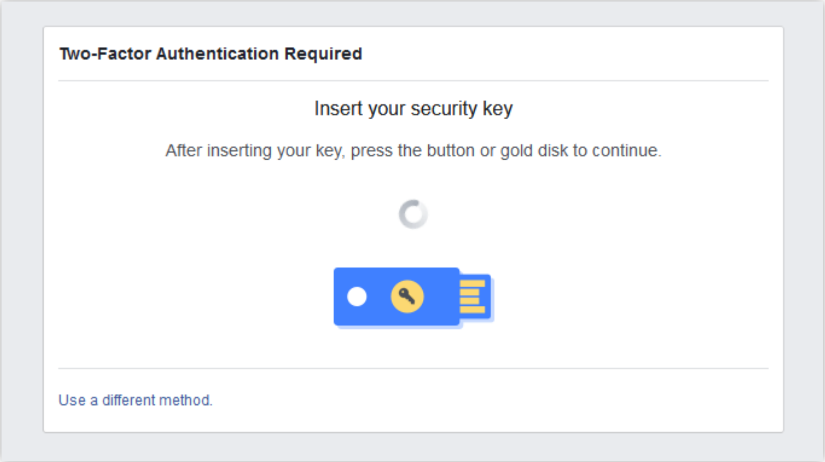
You can use any of these keys when asked to verify a login attempt.

✓ Security key successfully added.

REGISTERED KEYS

Step 9

Log out and then log in again to check that you can log in successfully. In the following window that appears, make sure that SHALO AUTH's LED is flashing and then press its button.



Two-Factor Authentication Required

Insert your security key

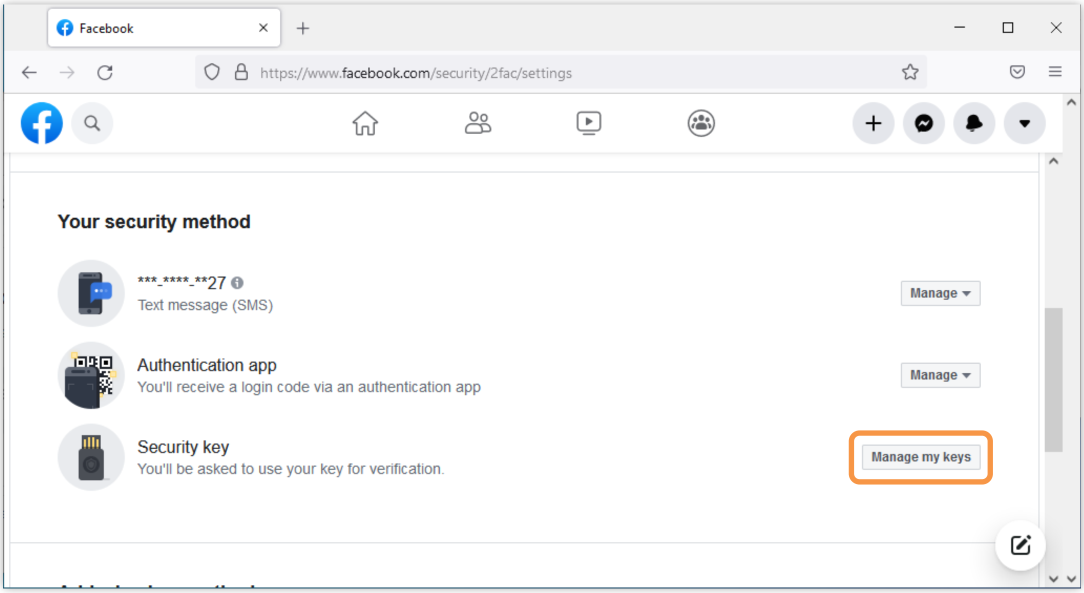
After inserting your key, press the button or gold disk to continue.

Use a different method.

6.2.2 Deregistering SHALO AUTH

You can deregister SHALO AUTH in the page you opened by selecting [**Security and Login**] > [**Two-factor authentication**] in steps 1 to 4 of the previous subsection.

Click [**Manage my keys**] next to [**Security key**] first.



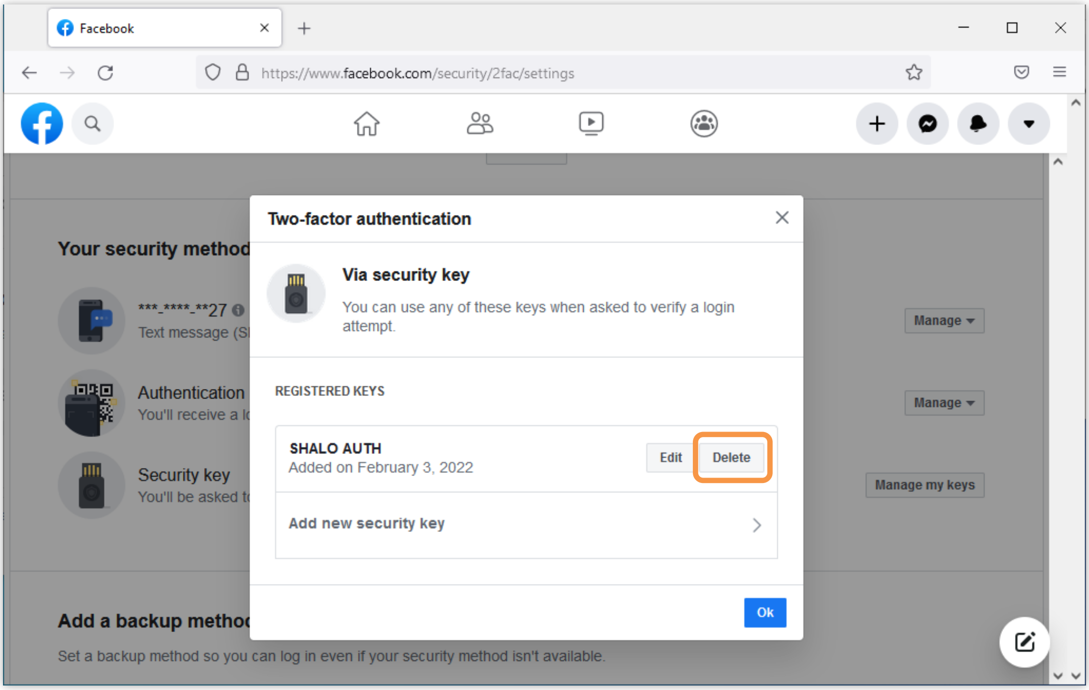
Your security method

-*-**27
Text message (SMS)

Authentication app
You'll receive a login code via an authentication app

Security key
You'll be asked to use your key for verification.

The security key you have previously registered is displayed. To complete the deregistration process, click [**Delete**] next to the security key you want to remove.



Two-factor authentication

Via security key
You can use any of these keys when asked to verify a login attempt.

REGISTERED KEYS

SHALO AUTH
Added on February 3, 2022

Add new security key

Add a backup method
Set a backup method so you can log in even if your security method isn't available.

6.3 U2F settings for GitHub

6.3.1 Registering SHALO AUTH

Configure the TOTP mobile app or SMS in advance to enable two-factor authentication. Make sure that you store the recovery code you will receive at this time in a safe location.

You can register SHALO AUTH in two-factor authentication for your GitHub account by using the procedure below. Make sure that SHALO AUTH is disconnected from the PC.

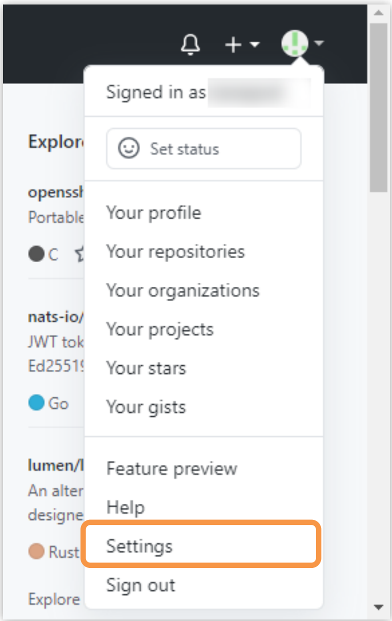
1. Open https://www.github.com in a Web browser and log in.
2. Click the profile image in the upper-right corner, and then click [**Settings**].
3. In the side bar on the left, click [**Account Security**].
4. Click [**Add**] next to [**Security Keys**].
5. In [**Security Keys**], click [**Register new security key**].
6. Input the nickname for the security key and click [**Add**].
7. Connect SHALO AUTH to the PC, wait until SHALO AUTH's LED flashes, and then press its button.
8. Sign out, and check that you can sign in with SHALO AUTH.

The following explains the procedure together with screenshots.

The explanations in this subsection are based on the information available at the time of writing this manual.
Note that the website screenshots may differ from those in the manual.

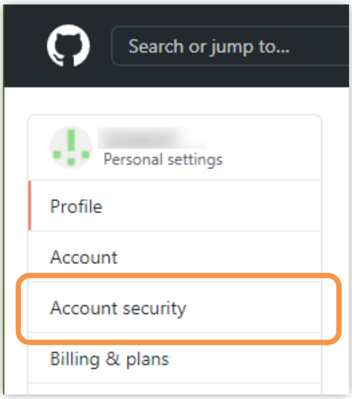
Steps 1 to 2

Log in to GitHub, click the profile image in the upper-right corner, and click [**Settings**].



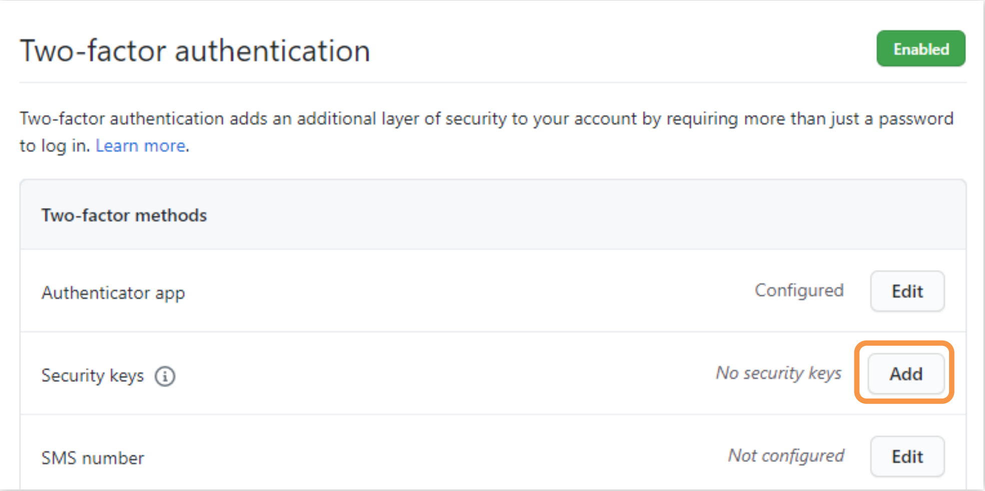
Step 3

In the side bar on the left, click [**Account Security**].



Step 4

Click [**Add**] next to [**Security Keys**].



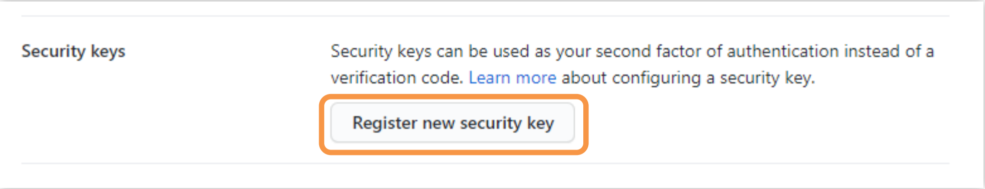
Two-factor authentication

Two-factor authentication adds an additional layer of security to your account by requiring more than just a password to log in. Learn more.

Two-factor methods		
Authenticator app	Configured	Edit
Security keys ⓘ	*No security keys*	Add
SMS number	*Not configured*	Edit

Step 5

In [**Security Keys**], click [**Register new security key**].

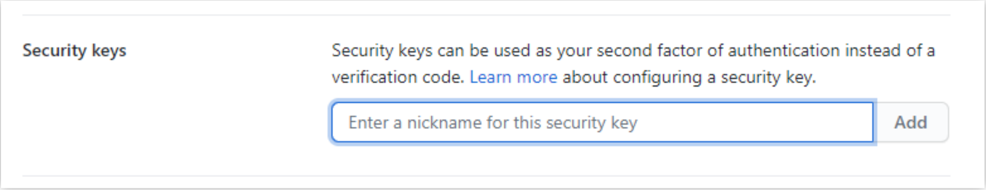


Security keys

Security keys can be used as your second factor of authentication instead of a verification code. Learn more about configuring a security key.

Step 6

Input the nickname for the security key and click [**Add**].

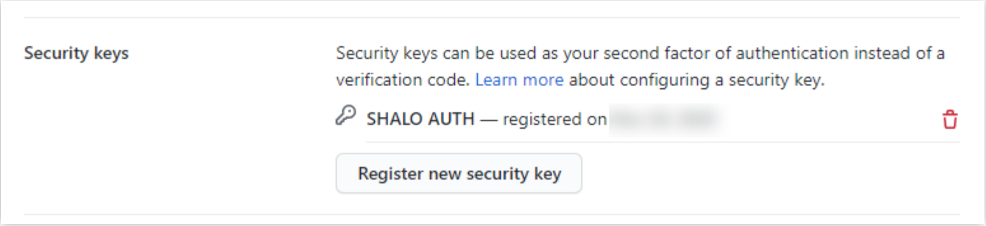


Security keys

Security keys can be used as your second factor of authentication instead of a verification code. Learn more about configuring a security key.

Step 7

Connect SHALO AUTH to the PC, wait until SHALO AUTH's LED flashes, and then press its button. When the registration process is successful, the nickname you specified will appear as follows.



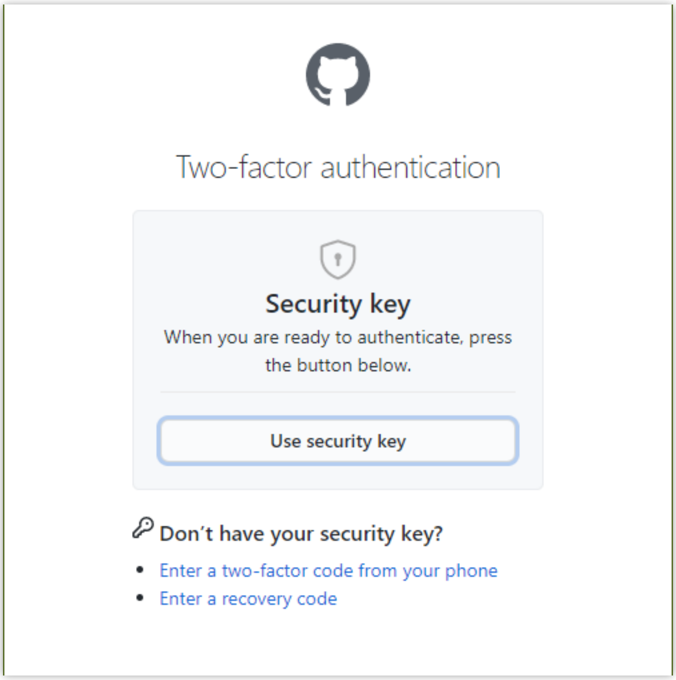
Security keys

Security keys can be used as your second factor of authentication instead of a verification code. Learn more about configuring a security key.

SHALO AUTH — registered on [redacted]

Step 8

Sign out and then sign in again to check that you can sign in successfully. Click [**Use security key**], wait until SHALO AUTH's LED flashes, and then press its button.



Two-factor authentication

Security key

When you are ready to authenticate, press the button below.

Don't have your security key?

- Enter a two-factor code from your phone
- Enter a recovery code

6.3.2 Deregistering SHALO AUTH

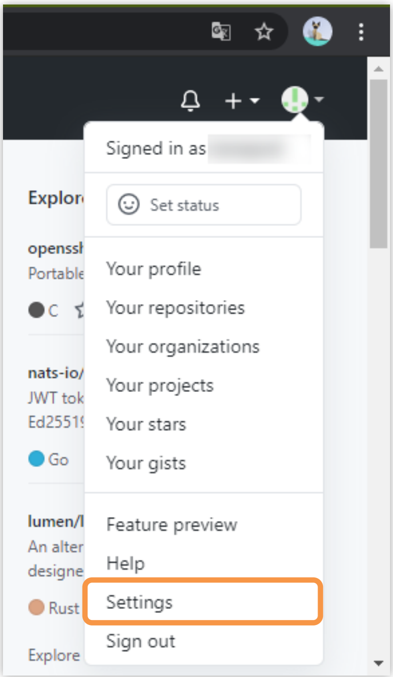
You can deregister SHALO AUTH from your GitHub account by using the following procedure:

1. Open https://www.github.com in a Web browser and log in.
2. Click the profile image in the upper-right corner, and then click [**Settings**].
3. In the side bar on the left, click [**Account Security**].
4. Click [**Edit**] next to [**Security Keys**].
5. In [**Security Keys**], click the icon next to the nickname for the security key you want to remove.

The following explains the procedure together with screenshots.

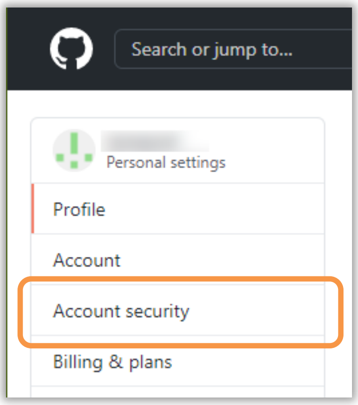
Steps 1 to 2

Log in to GitHub, click the profile image in the upper-right corner, and click [**Settings**].



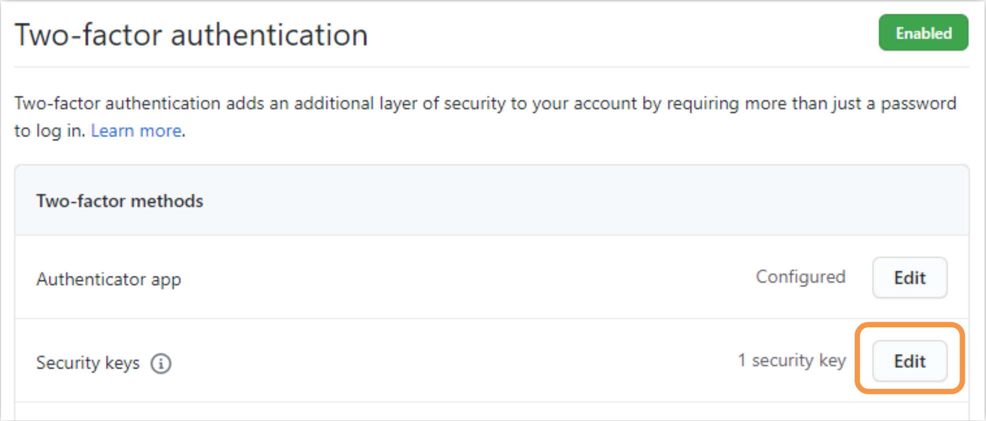
Step 3

In the side bar on the left, click [**Account Security**].



Step 4

Click [**Edit**] next to [**Security Keys**].

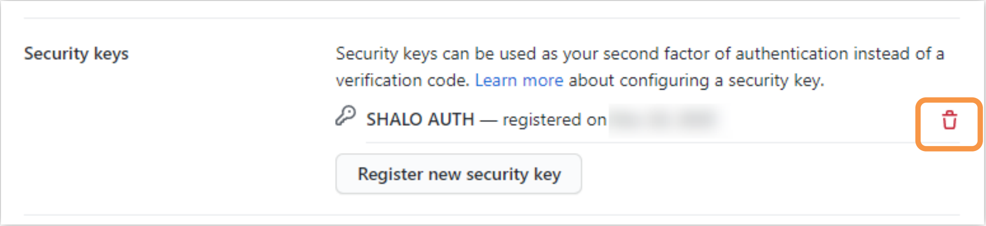

Two-factor authentication

Two-factor authentication adds an additional layer of security to your account by requiring more than just a password to log in. Learn more.

Two-factor methods		
Authenticator app	Configured	Edit
Security keys ⓘ	1 security key	Edit

Step 5

In [**Security Keys**], click the icon next to the nickname for the security key you want to remove.


Security keys

Security keys can be used as your second factor of authentication instead of a verification code. Learn more about configuring a security key.

SHALO AUTH — registered on [redacted]

Chapter 7

Using SHALO AUTH in PDF files

Adobe® Acrobat® and Adobe® Acrobat® Reader® for Windows or macOS can use SHALO AUTH through the PKCS #11 module. They can also use the keys stored in SHALO AUTH as digital IDs for Acrobat®.

This chapter explains how to use SHALO AUTH to secure PDF files.

Topics in this chapter

1. Understanding PDF file security
2. Registering the PKCS #11 module with Acrobat®
3. Importing a digital ID from SHALO AUTH
4. Giving the certificate of the digital ID to other people
5. Encrypting a PDF file with a digital ID
6. Viewing an encrypted PDF file
7. Signing a PDF file electronically with a digital ID

7.1 Understanding PDF file security

Adobe® Acrobat® and Adobe® Acrobat® Reader® for Windows or macOS (hereafter called Acrobat®) can use SHALO AUTH through the PKCS #11 module.

Combining the security for PDF files with SHALO AUTH enables:

- Certain SHALO AUTH owners to view the PDF files by encrypting them.
- The PDF files to be signed electronically using SHALO AUTH.

These operations employ the personal identification information called a **digital ID** in Acrobat®. This section explains the digital ID, followed by PDF encryption and electronic signing. The subsequent sections explain how to use SHALO AUTH in Acrobat®.

Digital ID

A digital ID is information used to identify an individual, and consists of the following two components:

- Private key for public key cryptography
- Certificate (public key for public key cryptography and personal information)

These are the same as the SHALO AUTH-managed data described in Section 2.3.3. Acrobat® supports the PKCS #11 API and thus can use keys stored in SHALO AUTH as the digital IDs.

Use RSA keys as the digital IDs.
This is because Acrobat® cannot use ECDSA keys through PKCS #11.

Encrypting PDF files

By encrypting PDF files, you can prevent them from being viewed by the general public. You can encrypt PDF files by:

- Protecting them with a password
- Protecting them through a certificate

Password protection is a method of encrypting files so that **only users who know the password can view them**. The user who creates the files and those who view them use the same password.

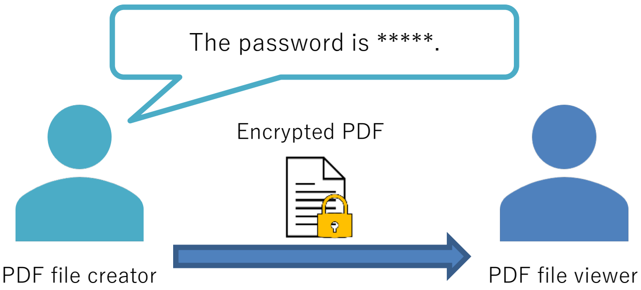


Figure 50 Providing a password-protected PDF file

In contrast, certificate protection is a method of encrypting files **so that only the users who have been certified with a certificate can view them**. This method is used to encrypt the PDF files, and employs a public key contained in the certificate of the digital ID provided by the viewer. **A digital ID is required to view the files.**

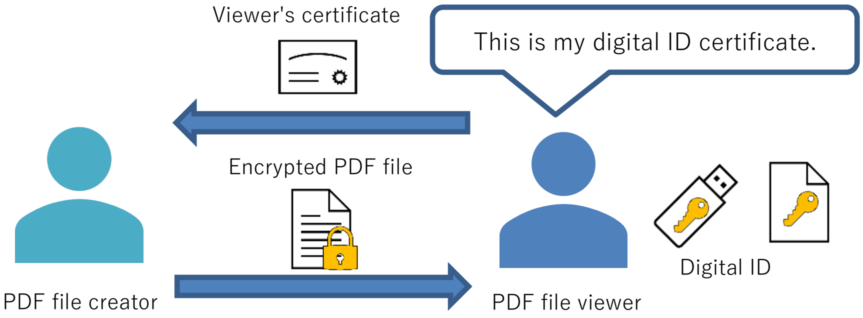


Figure 51 Providing an encrypted PDF file intended for the viewer's digital ID

The creator must prepare the digital ID for viewing to prevent the ID from being duplicated. The viewer is then given the SHALO AUTH device with the digital ID stored in it.

In this way, the creator does not have to create a digital ID for each PDF file. By managing the digital ID certificate, the creator can use the certificate and encrypt other PDF files for the same viewer.

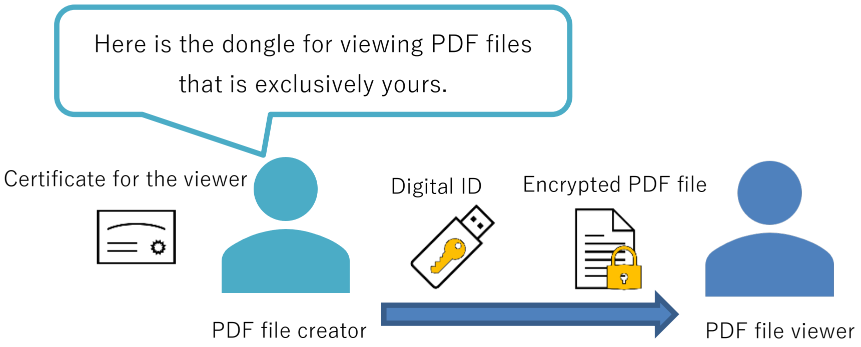

Here is the dongle for viewing PDF files that is exclusively yours.
Certificate for the viewer
Digital ID
Encrypted PDF file
PDF file creator
PDF file viewer

Figure 52 Providing an encrypted PDF file together with the dongle for viewing prepared by the creator

Electronic signature

A viewer can know the facts below based on the electronic signature assigned to a PDF file with a digital ID. The creator's digital ID certificate is required for this purpose.

- Whether the file was actually created by the creator.
- Confirmation that the file has not been tampered with.

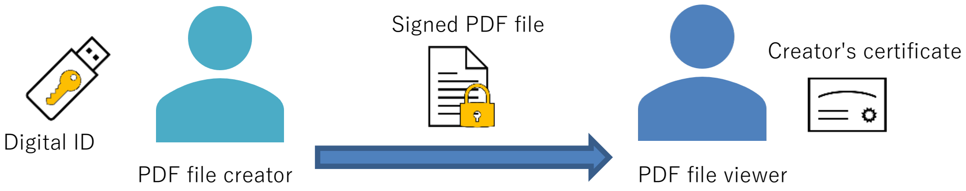


Figure 53 Validating the PDF file with the electronic signature

7.2 Configuring Acrobat®

7.2.1 Registering the PKCS #11 module with Acrobat®

Before using SHALO AUTH in Acrobat®, register SHALO AUTH's PKCS #11 module with Acrobat®.

Notes for when using SHALO AUTH in Acrobat®

ℹ You can use SHALO AUTH even if you connect SHALO AUTH to the PC while Acrobat® is running.

ℹ When you enter the user PIN for SHALO AUTH in Acrobat®, you do not have to enter the PIN again until you log out explicitly or exit Acrobat®.

⚠ If you want to use SHALO AUTH in Acrobat® for Windows, you must disable the protected mode.

⚠ Acrobat® may start using the general security key functionality of a SHALO AUTH device connected to a PC arbitrarily, and from that point, other software cannot use the functionality. If this happens, exit Acrobat®.

⚠ Do not disconnect SHALO AUTH while Acrobat® is running. This prevents you from performing operations through SHALO AUTH even when you reconnect it.
If you see the "PKCS #11 error" in Acrobat® after disconnecting SHALO AUTH, then exit Acrobat®.

Registration procedure

To register the PKCS #11 module with Acrobat®, use the following procedure:

1. Windows: In the menu, click [**Edit**] > [**Preferences...**].
   macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
2. Windows only: Click [**Security (Enhanced)**], and in the [**Sandbox Protections**] section, clear the [**Enable Protected Mode at startup**] check box and restart Acrobat®.
3. Click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**].
4. Select [**PKCS#11 Modules and Tokens**] and click [**Attach Module**].
5. Select SHALO AUTH's PKCS #11 module file.

The following explains the procedure together with screenshots.

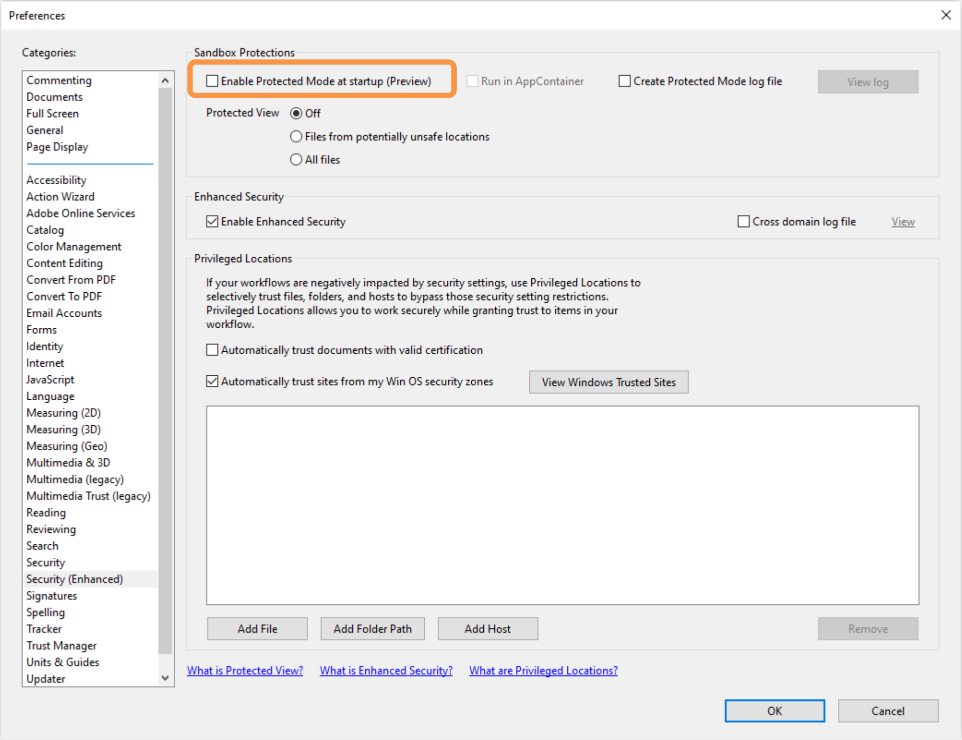
Step 1

In Windows: In the menu, click [**Edit**] > [**Preferences...**].

In macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
The details of the first menu items depend on the application types.

Step 2 (Windows only)

Under Categories, click [**Security (Enhanced)**], and in the [**Sandbox Protections**] section, check that the [**Enable Protected Mode at startup**] check box is cleared.


Preferences

Sandbox Protections

□ Enable Protected Mode at startup (Preview) □ Run in AppContainer □ Create Protected Mode log file

Protected View ● Off ○ Files from potentially unsafe locations ○ All files

Enhanced Security

☑ Enable Enhanced Security □ Cross domain log file

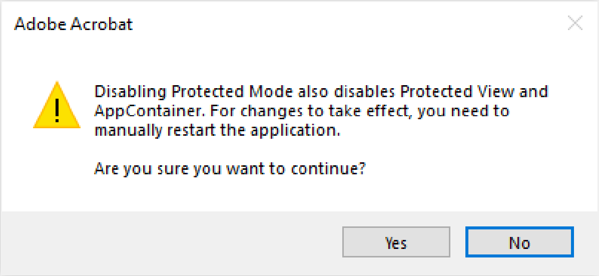
Privileged Locations

If your workflows are negatively impacted by security settings, use Privileged Locations to selectively trust files, folders, and hosts to bypass those security setting restrictions. Privileged Locations allows you to work securely while granting trust to items in your workflow.

□ Automatically trust documents with valid certification

☑ Automatically trust sites from my Win OS security zones

If the check box is selected, clear it. At this time, the window below will appear. Click [**Yes**], exit and start Acrobat® again, and then perform step 1.

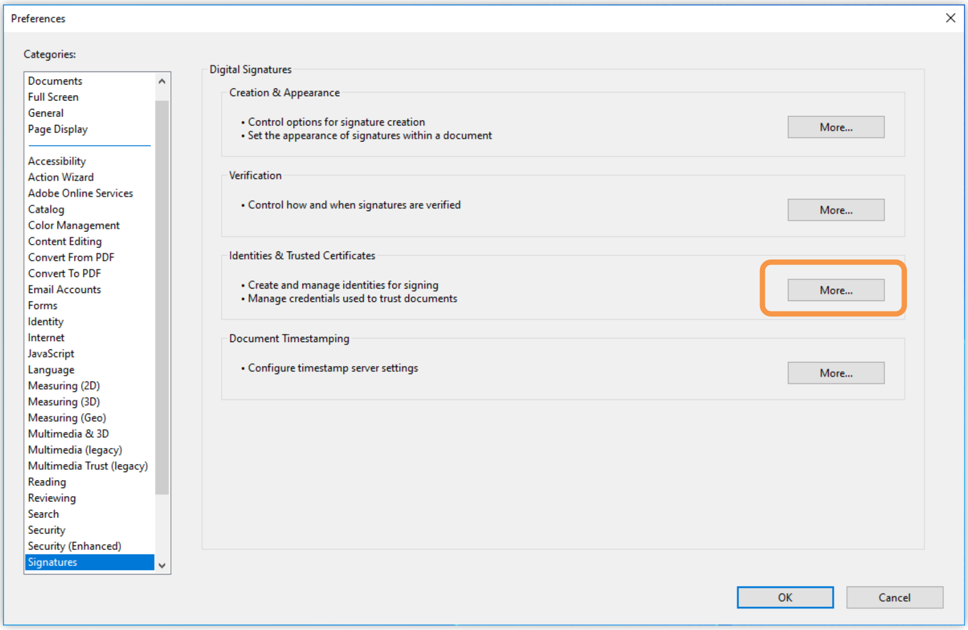

Adobe Acrobat

Disabling Protected Mode also disables Protected View and AppContainer. For changes to take effect, you need to manually restart the application.

Are you sure you want to continue?

Step 3

Under Categories, click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**].



Digital Signatures

Creation & Appearance

- Control options for signature creation
- Set the appearance of signatures within a document

More...

Verification

- Control how and when signatures are verified

More...

Identities & Trusted Certificates

- Create and manage identities for signing
- Manage credentials used to trust documents

More...

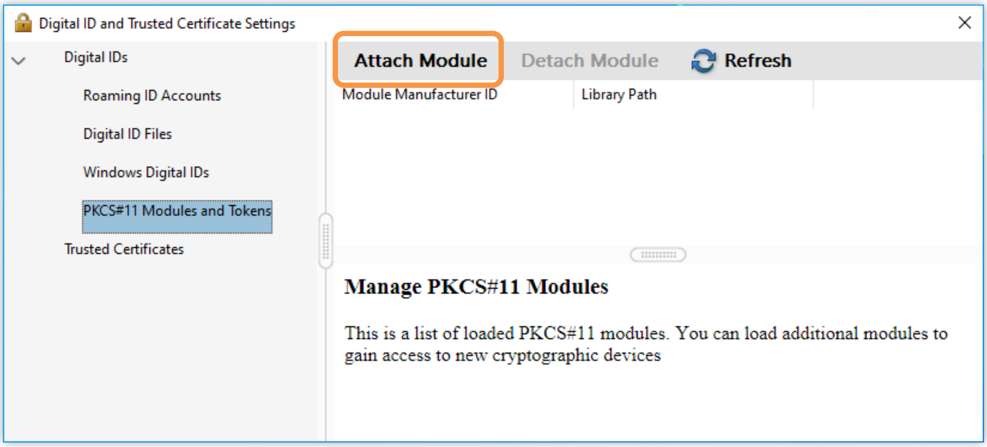
Document Timestamping

- Configure timestamp server settings

More...

Step 4

In the window shown in the following figure, select [**PKCS#11 Modules and Tokens**] and click [**Attach Module**].



Manage PKCS#11 Modules

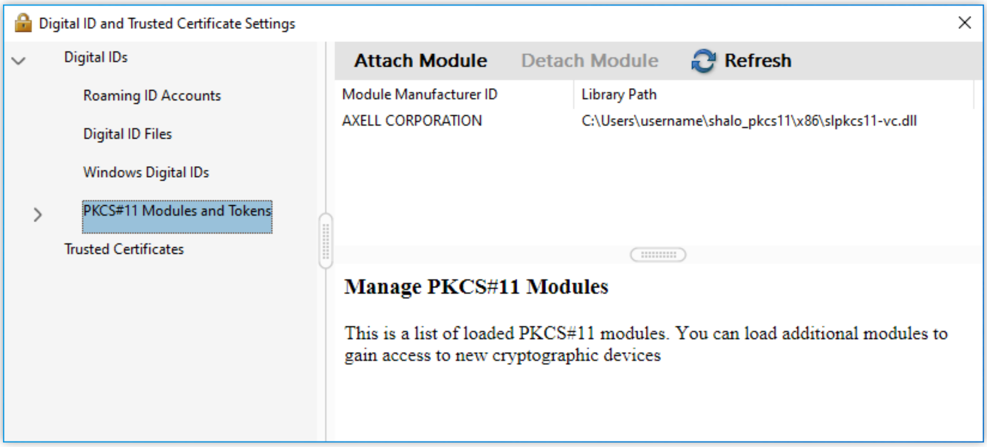
This is a list of loaded PKCS#11 modules. You can load additional modules to gain access to new cryptographic devices

Step 5

Select SHALO AUTH's PKCS #11 module. Depending on your environment, select the following file:

Windows (Acrobat 32-bit)	`C:\Users\`*`user-name`*`\shalo_pkcs11\x86\slpkcs11-vc.dll`
Windows (Acrobat 64-bit)	`C:\Users\`*`user-name`*`\shalo_pkcs11\x64\slpkcs11-vc.dll`
macOS	`/usr/local/lib/libslpkcs11.dylib`

When the file is loaded successfully, the module is registered with the list of modules as shown in the following figure.

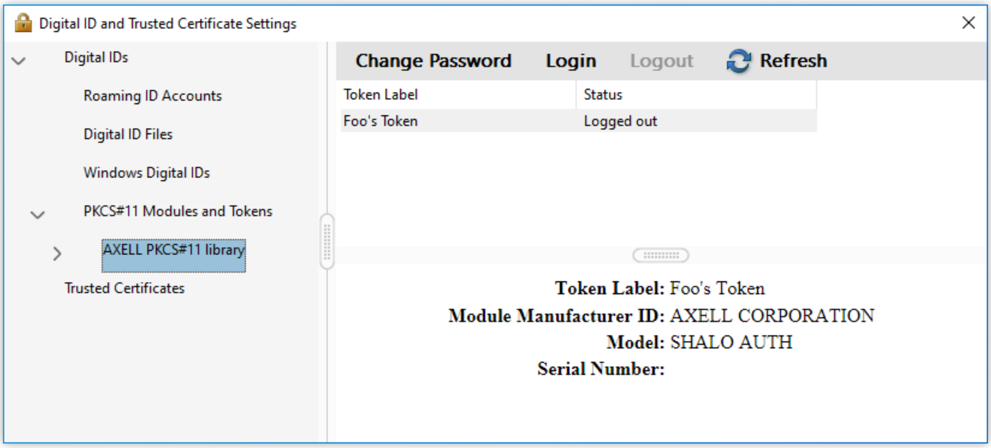


Module Manufacturer ID	Library Path
AXELL CORPORATION	C:\Users\username\shalo_pkcs11\x86\slpkcs11-vc.dll

Manage PKCS#11 Modules

This is a list of loaded PKCS#11 modules. You can load additional modules to gain access to new cryptographic devices

[**AXELL PKCS#11 library**] is added to the level under [**PKCS#11 Modules and Tokens**].



Token Label	Status
Foo's Token	Logged out

Token Label: Foo's Token
Module Manufacturer ID: AXELL CORPORATION
Model: SHALO AUTH
Serial Number:

The level under [**AXELL PKCS#11 library**] displays device labels of SHALO AUTH devices connected to the PC.

7.2.2 Deregistering the PKCS #11 module from Acrobat®

To deregister the PKCS #11 module from Acrobat®, use the following procedure:

1. Windows: In the menu, click [**Edit**] > [**Preferences...**].
   macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
2. Click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**].
3. Select [**PKCS#11 Modules and Tokens**] and select the module of SHALO AUTH from the list.
4. Click [**Detach Module**].

The following explains the procedure together with screenshots.

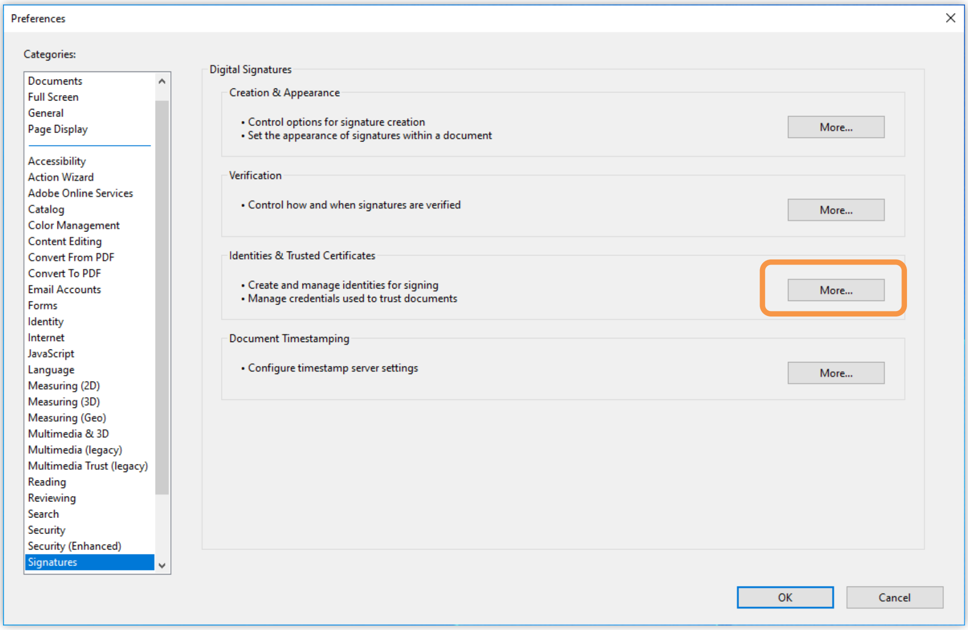
Step 1

In Windows: In the menu, click [**Edit**] > [**Preferences...**].

In macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
The details of the first menu items depend on the application types.

Step 2

Under Categories, click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**].



Digital Signatures

Creation & Appearance
- Control options for signature creation
- Set the appearance of signatures within a document

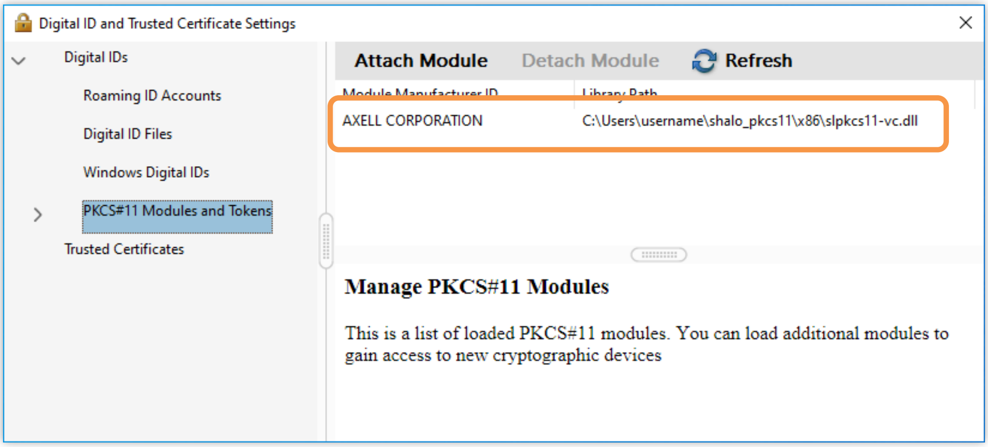
Verification
- Control how and when signatures are verified

Identities & Trusted Certificates
- Create and manage identities for signing
- Manage credentials used to trust documents

Document Timestamping
- Configure timestamp server settings

Step 3

In the window shown in the following figure, select [**PKCS#11 Modules and Tokens**], and select the module of SHALO AUTH.



Module Manufacturer ID	Library Path
AXELL CORPORATION	C:\Users\username\shalo_pkcs11\x86\slpkcs11-vc.dll

Manage PKCS#11 Modules

This is a list of loaded PKCS#11 modules. You can load additional modules to gain access to new cryptographic devices

Step 4

Click [**Detach Module**].

7.3 Importing a digital ID from SHALO AUTH

Import a certificate in SHALO AUTH as a digital ID before using SHALO AUTH in Acrobat® for the first time, or when changing a key in SHALO AUTH.

To do this, connect SHALO AUTH to the PC and use the following procedure in Acrobat®:

1. Windows: In the menu, click [**Edit**] > [**Preferences...**].
   macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
2. Click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**].
3. Click [**Digital IDs**] and check that the key information in SHALO AUTH has been loaded.
4. (If the key information has not been loaded) Click [**PKCS#11 Modules and Tokens**] > [**AXELL PKCS#11 library**], and from the list of token labels, select the device label of SHALO AUTH you use, and click [**Login**]. Then, enter the user PIN as a password.

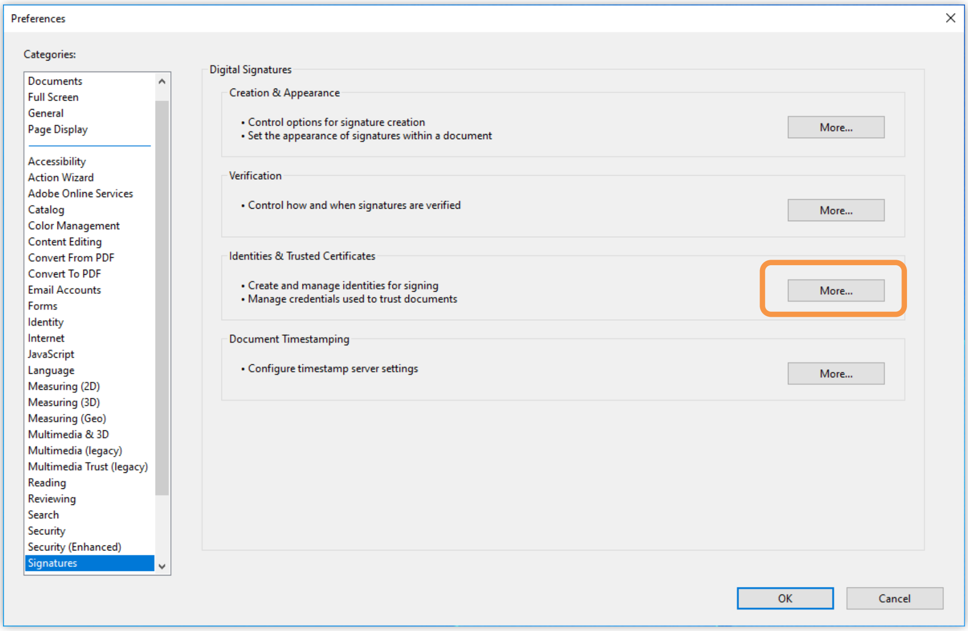
The following explains the procedure together with screenshots.

Steps 1 to 2

In Windows: In the menu, click [**Edit**] > [**Preferences...**].

In macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
The details of the first menu items depend on the application types.

In the Preferences window that appears, under Categories, click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**] as shown in the following figure.



Creation & Appearance
• Control options for signature creation
• Set the appearance of signatures within a document

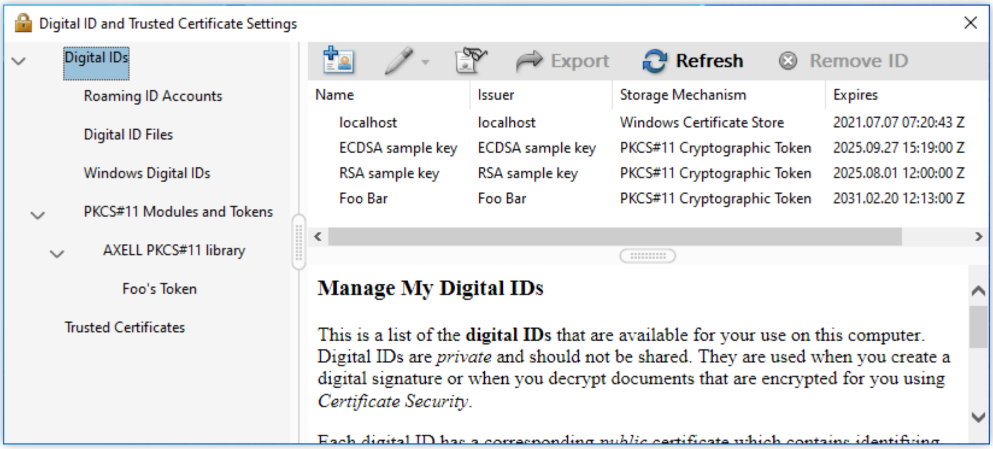
Verification
• Control how and when signatures are verified

Identities & Trusted Certificates
• Create and manage identities for signing
• Manage credentials used to trust documents

Document Timestamping
• Configure timestamp server settings

Step 3

Click [**Digital IDs**] and check that the key information in SHALO AUTH has been loaded. The certificates from SHALO AUTH are displayed as “PKCS#11 Cryptographic Token” in the Storage Mechanism column.



Name	Issuer	Storage Mechanism	Expires
localhost	localhost	Windows Certificate Store	2021.07.07 07:20:43 Z
ECDSA sample key	ECDSA sample key	PKCS#11 Cryptographic Token	2025.09.27 15:19:00 Z
RSA sample key	RSA sample key	PKCS#11 Cryptographic Token	2025.08.01 12:00:00 Z
Foo Bar	Foo Bar	PKCS#11 Cryptographic Token	2031.02.20 12:13:00 Z

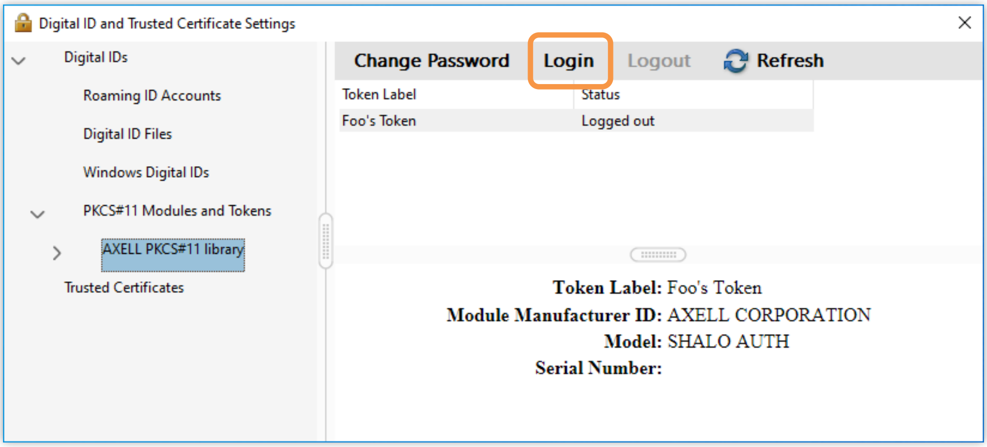
Manage My Digital IDs

This is a list of the **digital IDs** that are available for your use on this computer. Digital IDs are *private* and should not be shared. They are used when you create a digital signature or when you decrypt documents that are encrypted for you using *Certificate Security*.

Each digital ID has a corresponding *public certificate* which contains identifying

Step 4 (if the certificate has not been loaded)

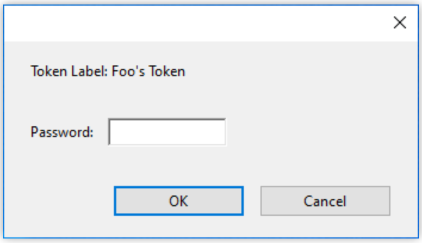
Click [**PKCS#11 Modules and Tokens**] > [**AXELL PKCS#11 library**], and from the list of token labels, select the device label of SHALO AUTH you use, and click [**Login**].



Token Label	Status
Foo's Token	Logged out

Token Label: Foo's Token
Module Manufacturer ID: AXELL CORPORATION
Model: SHALO AUTH
Serial Number:

In the following window, enter the user PIN.



Token Label: Foo's Token

Password:

Click the child element of [**AXELL PKCS#11 library**] and check if the key information has been loaded properly.

7.4 Giving the certificate of the digital ID to other people

If you want to provide the digital ID's certificate to other people, you must output the certificate in Acrobat®-specific file format, not as an X.509 certificate.

To do this, connect SHALO AUTH to the PC and use the following procedure in Acrobat®:

1. Windows: In the menu, click [**Edit**] > [**Preferences...**].
   macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
2. Click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**].
3. Expand [**PKCS#11 Modules and Tokens**] and select the relevant SHALO AUTH device.
4. Select the certificate and click [**Export**].
5. Specify an export option and export the certificate.

The following explains the procedure together with screenshots.

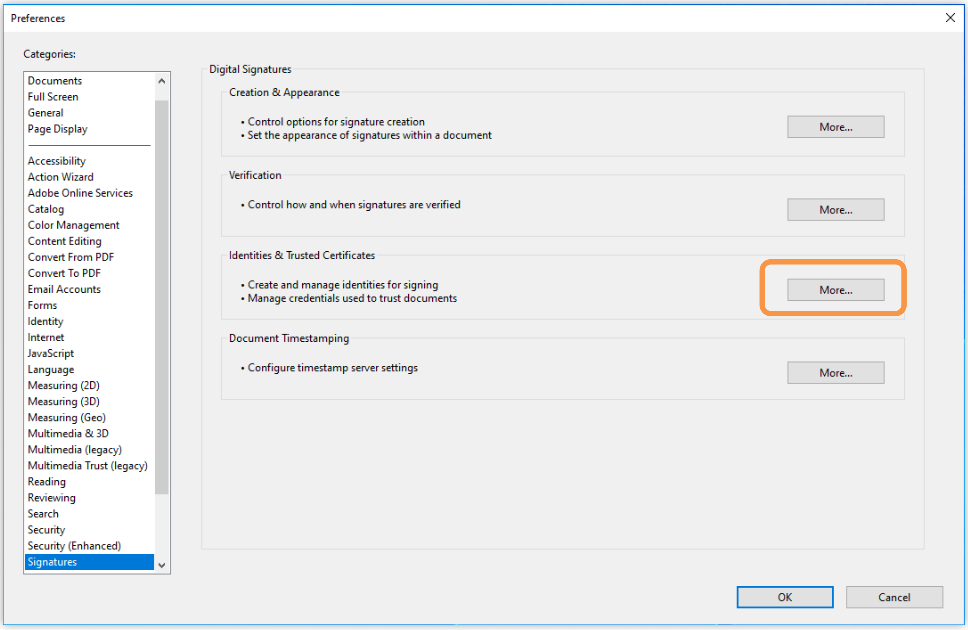
⚠ Take care not to provide an ECDSA key certificate.
Users cannot view any PDF files encrypted by an ECDSA key through SHALO AUTH.

Steps 1 to 2

In Windows: In the menu, click [**Edit**] > [**Preferences...**].

In macOS: In the menu, click [**Acrobat Reader**] > [**Preferences...**] or [**Acrobat Pro DC**] > [**Preferences...**].
The details of the first menu items depend on the application types.

In the Preferences window that appears, under Categories, click [**Signatures**], and in the [**Identities & Trusted Certificates**] section, click [**More...**] as shown in the following figure.



Digital Signatures

Creation & Appearance

- Control options for signature creation
- Set the appearance of signatures within a document

Verification

- Control how and when signatures are verified

Identities & Trusted Certificates

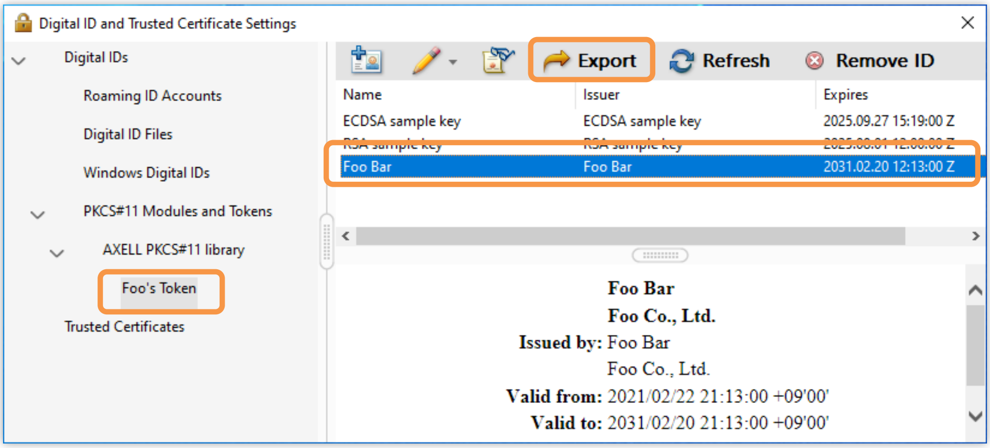
- Create and manage identities for signing
- Manage credentials used to trust documents

Document Timestamping

- Configure timestamp server settings

Steps 3 and 4

In the window shown in the figure below, expand [**PKCS#11 Modules and Tokens**] and select the device label of the relevant SHALO AUTH device. Select the certificate and click [**Export**].



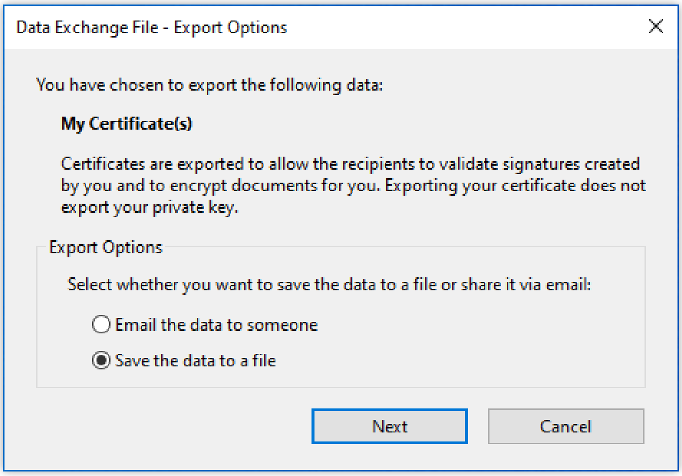
Name	Issuer	Expires
ECDSA sample key	ECDSA sample key	2025.09.27 15:19:00 Z
RSA sample key	RSA sample key	2025.08.01 12:00:00 Z
Foo Bar	Foo Bar	2031.02.20 12:13:00 Z

Foo Bar
Foo Co., Ltd.
Issued by: Foo Bar
Foo Co., Ltd.
Valid from: 2021/02/22 21:13:00 +09'00'
Valid to: 2031/02/20 21:13:00 +09'00'

You can also click [**Digital IDs**] and select a certificate from there.
When multiple SHALO AUTH devices are connected, you can find a certificate easily by selecting it from the device labels.

Step 5

In the window shown in the figure below, select an export option and click [**Next**]. Then, proceed as instructed in each window.



Data Exchange File - Export Options

You have chosen to export the following data:

My Certificate(s)

Certificates are exported to allow the recipients to validate signatures created by you and to encrypt documents for you. Exporting your certificate does not export your private key.

Export Options

Select whether you want to save the data to a file or share it via email:

○ Email the data to someone

● Save the data to a file

7.5 Encrypting a PDF file with a digital ID

This section explains how to encrypt a PDF file through certificate protection. The digital ID's certificate is needed for encrypting PDF files. You can specify more than one viewer for encrypted PDF files, and can do this in the following two ways:

- Specify the viewers with the digital ID (SHALO AUTH digital ID) they own.
- Specify the viewers with the digital ID's certificate file.

The user PIN does not have to be entered even when the SHALO AUTH's digital ID is specified.

Take care not to encrypt PDF files with an ECDSA key certificate.
Users cannot use SHALO AUTH to view any PDF files encrypted by an ECDSA key.

Adobe® Acrobat® is needed in order to encrypt PDF files.
Adobe® Acrobat® Reader® does not have the ability to encrypt PDF files.

To encrypt a PDF file, use the following procedure in Adobe® Acrobat®:

1. Open a PDF file.
2. In the menu, click [**File**] > [**Properties...**].
3. In the Document Properties window, open the [**Security**] tab, and from Security Method, select [**Certificate Security**].
4. Select document components to encrypt and an encryption algorithm, and click [**Next**].
5. To enable the PDF file to be viewed through SHALO AUTH connected to the PC, select a digital ID and click [**OK**]. Otherwise, click [**Cancel**] and click [**Continue anyway**].
6. When specifying viewers who have a digital ID certificate, click [**Browse...**] and select the certificate file for the digital ID.
7. Click the viewer's digital ID and then click [**Permissions...**].
8. Configure the settings appropriately according to your operating policy, and click [**OK**].
9. Click [**Next**] and click [**Finish**].
10. Close Document Properties and save the PDF file.

You must grant appropriate permissions to viewers to encrypt PDF files.
If inappropriate permissions are granted, they can generate PDF files that can evade viewing restrictions.

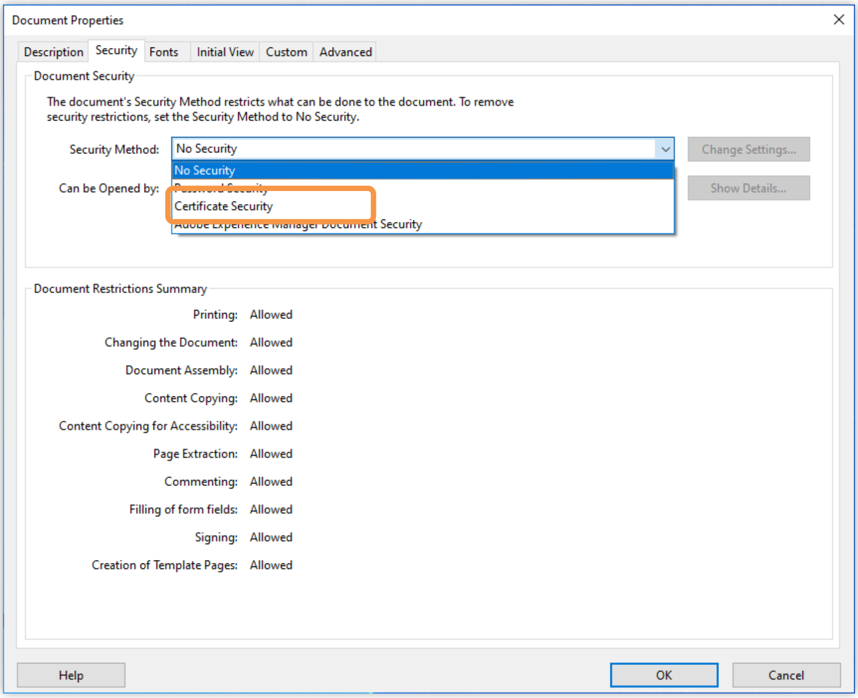
The following explains the procedure together with screenshots.

Steps 1 to 2

Open the PDF file you want to encrypt, and in the menu, click [**File**] > [**Properties...**].

Step 3

In the Document Properties window, open the [**Security**] tab, and from Security Method, select [**Certificate Security**] as shown in the following figure.



Document Properties

Document Security

The document's Security Method restricts what can be done to the document. To remove security restrictions, set the Security Method to No Security.

Security Method: No Security

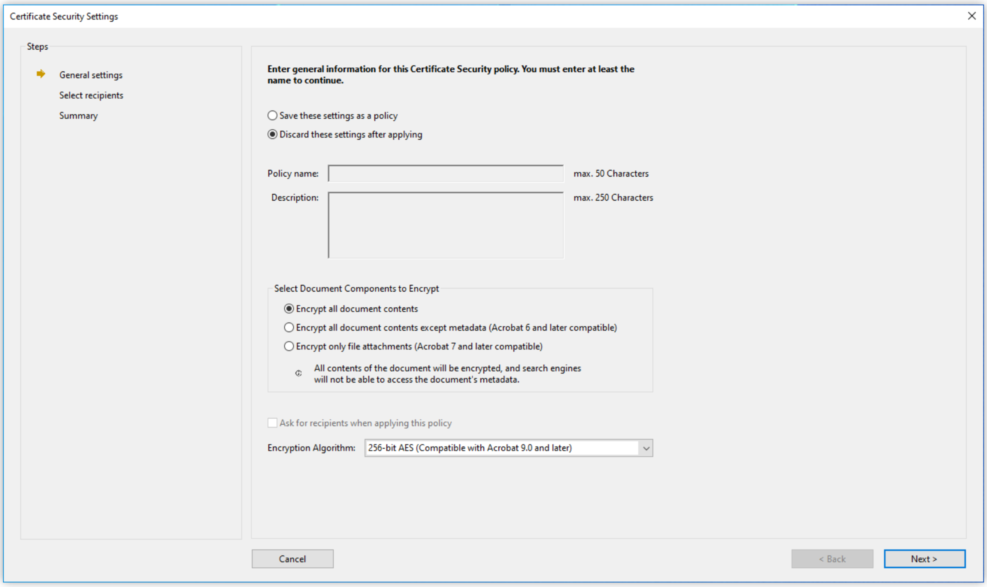
Can be Opened by:

Document Restrictions Summary

Printing: Allowed
Changing the Document: Allowed
Document Assembly: Allowed
Content Copying: Allowed
Content Copying for Accessibility: Allowed
Page Extraction: Allowed
Commenting: Allowed
Filling of form fields: Allowed
Signing: Allowed
Creation of Template Pages: Allowed

Step 4

Select document components to encrypt and an encryption algorithm, and click [**Next**].



Certificate Security Settings

Enter general information for this Certificate Security policy. You must enter at least the name to continue.

○ Save these settings as a policy
● Discard these settings after applying

Policy name: max. 50 Characters

Description: max. 250 Characters

Select Document Components to Encrypt

● Encrypt all document contents
○ Encrypt all document contents except metadata (Acrobat 6 and later compatible)
○ Encrypt only file attachments (Acrobat 7 and later compatible)

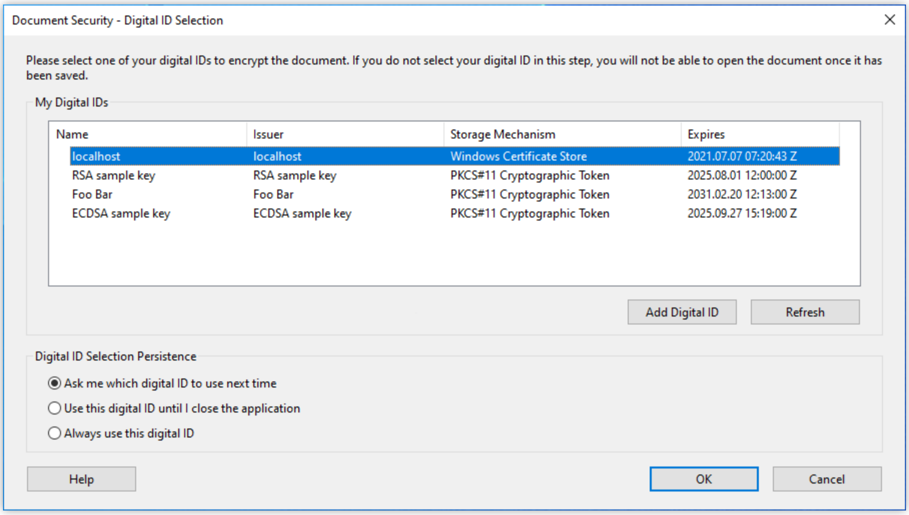
All contents of the document will be encrypted, and search engines will not be able to access the document's metadata.

□ Ask for recipients when applying this policy

Encryption Algorithm: 256-bit AES (Compatible with Acrobat 9.0 and later)

Step 5

Available digital IDs are displayed in the [**My Digital IDs**] section. To enable viewing of the PDF file through SHALO AUTH connected to the PC, select a digital ID and click [**OK**].



Document Security - Digital ID Selection

Please select one of your digital IDs to encrypt the document. If you do not select your digital ID in this step, you will not be able to open the document once it has been saved.

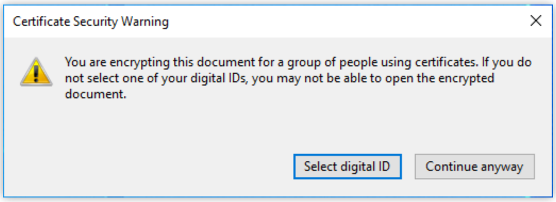
My Digital IDs

Name	Issuer	Storage Mechanism	Expires
localhost	localhost	Windows Certificate Store	2021.07.07 07:20:43 Z
RSA sample key	RSA sample key	PKCS#11 Cryptographic Token	2025.08.01 12:00:00 Z
Foo Bar	Foo Bar	PKCS#11 Cryptographic Token	2031.02.20 12:13:00 Z
ECDSA sample key	ECDSA sample key	PKCS#11 Cryptographic Token	2025.09.27 15:19:00 Z

Digital ID Selection Persistence

● Ask me which digital ID to use next time
○ Use this digital ID until I close the application
○ Always use this digital ID

Otherwise, click [**Cancel**] and in the following window, click [**Continue anyway**].

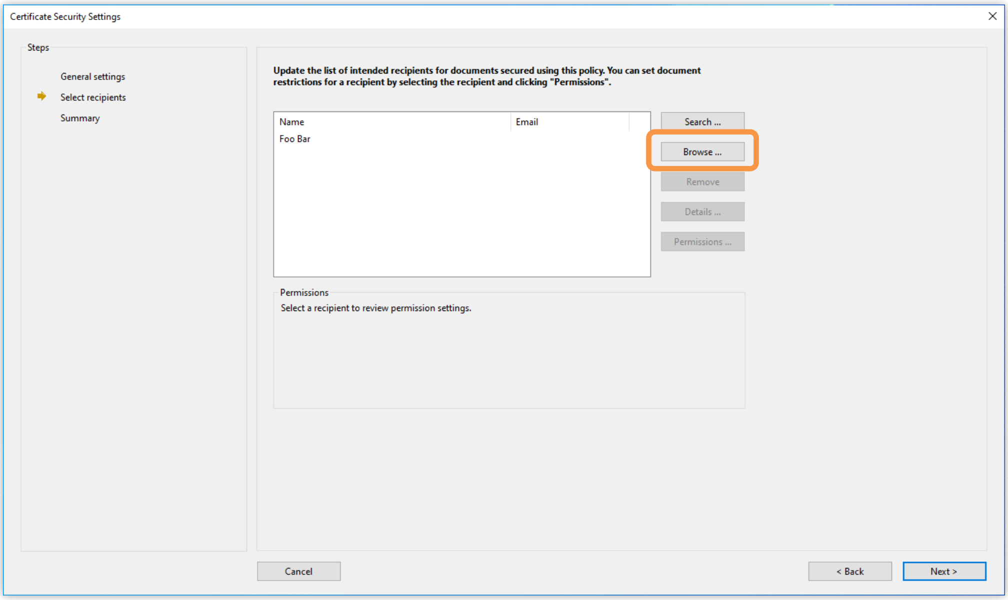


Certificate Security Warning

You are encrypting this document for a group of people using certificates. If you do not select one of your digital IDs, you may not be able to open the encrypted document.

Step 6

When specifying viewers who have a digital ID certificate, click [**Browse...**] and select the certificate file for the digital ID.



Certificate Security Settings

Update the list of intended recipients for documents secured using this policy. You can set document restrictions for a recipient by selecting the recipient and clicking "Permissions".

Name	Email
Foo Bar	

Permissions

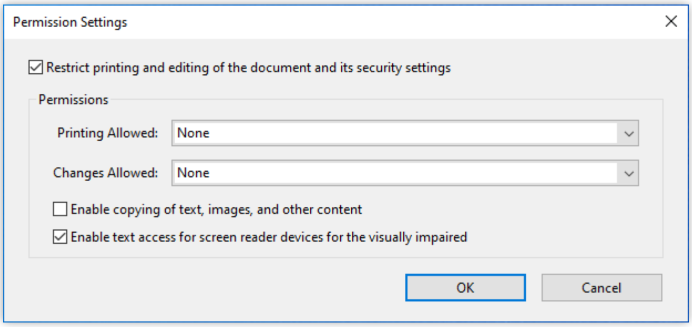
Select a recipient to review permission settings.

Step 7

Click the viewer's digital ID and then click [**Permissions...**].

Step 8

The window below will appear. Configure the settings appropriately according to your operating policy, and click [**OK**].



Permission Settings

☑ Restrict printing and editing of the document and its security settings

Permissions

Printing Allowed: None

Changes Allowed: None

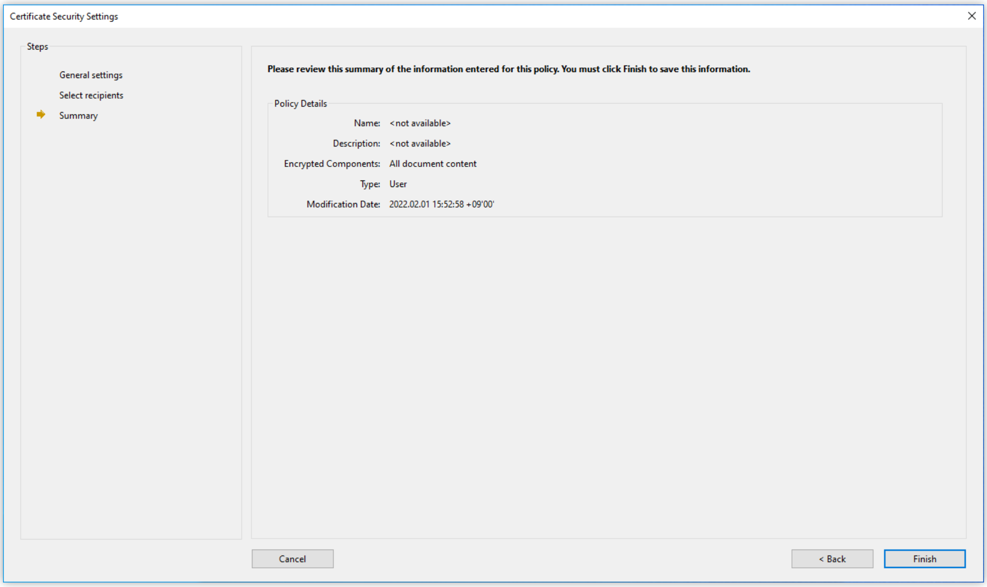
☐ Enable copying of text, images, and other content

☑ Enable text access for screen reader devices for the visually impaired

Inappropriate permissions can lead to generation of data that evades viewing restrictions.
For example, if printing is permitted, a user can create data that is not encrypted by printing data to a virtual printer.

Step 9

Click [**Next**] to display the window below. Click [**Finish**].



Certificate Security Settings

Steps
- General settings
- Select recipients
- Summary

Please review this summary of the information entered for this policy. You must click Finish to save this information.

Policy Details

Name:	<not available>
Description:	<not available>
Encrypted Components:	All document content
Type:	User
Modification Date:	2022.02.01 15:52:58 +09'00'

Step 10

Close Document Properties and save the PDF file.

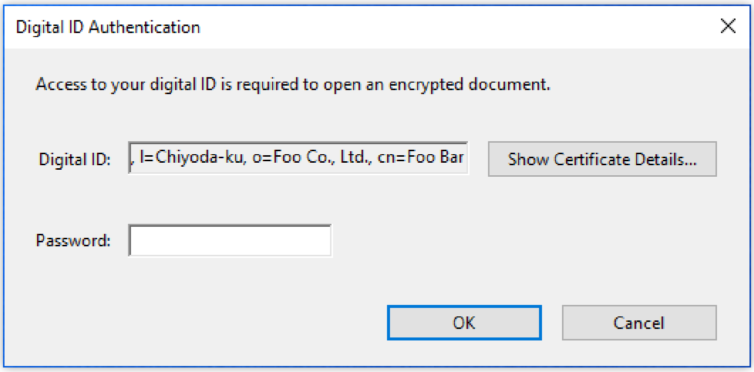
7.6 Viewing an encrypted PDF file

You can view an encrypted PDF file by connecting SHALO AUTH to the PC and opening the file in Acrobat®.

If other software is using SHALO AUTH's general security key functionality, you must let it stop using SHALO AUTH.
It can employ the FIDO U2F security functionality even when using SHALO AUTH's general security key functionality.

When Acrobat® is using SHALO AUTH, other software cannot use the device until you exit Acrobat®. Before other software can use SHALO AUTH's general security key functionality, you need to exit Acrobat®.

In general, either of the widows below will appear when you open an encrypted file. You will see the PDF file's content once you have entered the user PIN in Acrobat®.



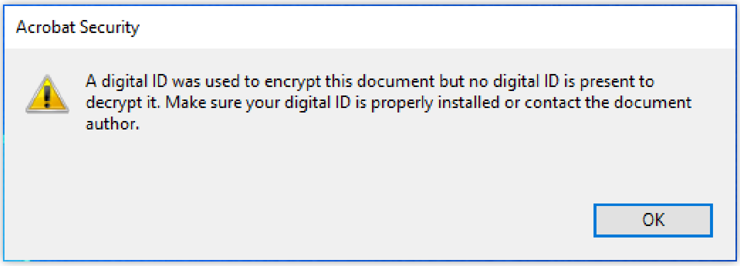
Digital ID Authentication

Access to your digital ID is required to open an encrypted document.

Digital ID: , l=Chiyoda-ku, o=Foo Co., Ltd., cn=Foo Bar

Password:

Figure 54 When a digital ID necessary for viewing is registered with Acrobat®



Acrobat Security

A digital ID was used to encrypt this document but no digital ID is present to decrypt it. Make sure your digital ID is properly installed or contact the document author.

Figure 55 When a digital ID necessary for viewing is not registered with Acrobat®

In Figure 54, input the user PIN for SHALO AUTH in the password field, and click [**OK**]. If the PIN authentication is successful and the file is decrypted properly, you can view the PDF file.

In Figure 55, Acrobat® does not have the digital ID needed for viewing the PDF file. See Sections 7.2 and 7.3 to import the digital ID from SHALO AUTH into Acrobat®. Then open the PDF file again.

7.7 Signing a PDF file electronically with a digital ID

To sign a PDF file electronically through SHALO AUTH, use the procedure below.

1. Open a PDF file in Acrobat®.
2. Select [**Tools**] and click [**Certificates**].
3. Click [**Digitally Sign**].
4. Specify the area for displaying the electronic signature in the PDF file by dragging it with the mouse.
5. Select a digital ID to use for signing the file.
6. Input the user PIN for SHALO AUTH (if you are prompted to do so). Then click [**Sign**].
7. Specify a file where the PDF file will be stored.

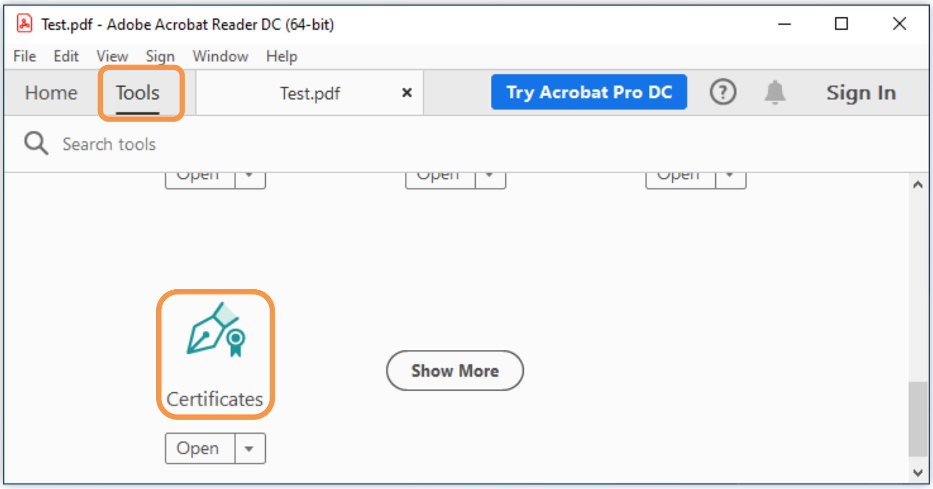
If other software is using SHALO AUTH's general security key functionality, you must let it stop using SHALO AUTH.
It can employ the FIDO U2F security functionality even when using SHALO AUTH's general security key functionality.

When Acrobat® is using SHALO AUTH, other software cannot use the device until you exit Acrobat®. Before other software can use SHALO AUTH's general security key functionality, you need to exit Acrobat®.

The following explains the procedure together with screenshots.

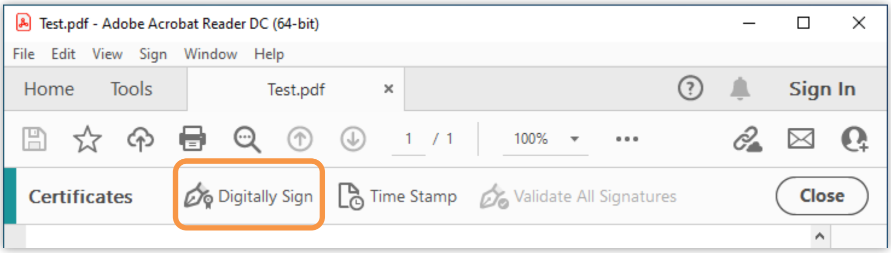
Step 2

As shown in the following figure, select [**Tools**] and click [**Certificates**].



Step 3

When Certificates appears instead of Tools as shown in the following figure, click [**Digitally Sign**].

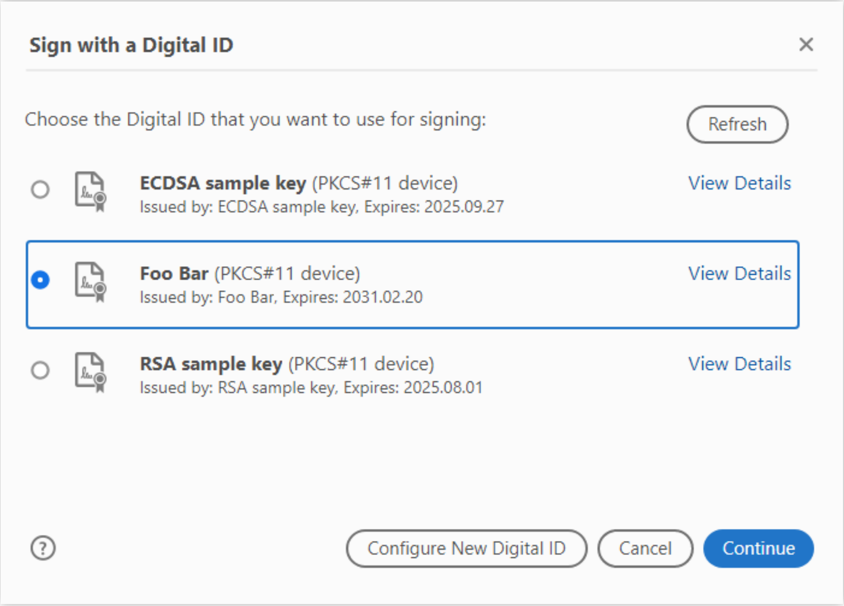


Step 4

Specify the area for displaying the electronic signature in the PDF file by dragging it with the mouse.

Step 5

As shown in the figure below, the certificates stored in SHALO AUTH are listed. From the list, select the digital ID to use for the signature and click [**Continue**].



Sign with a Digital ID

Choose the Digital ID that you want to use for signing:

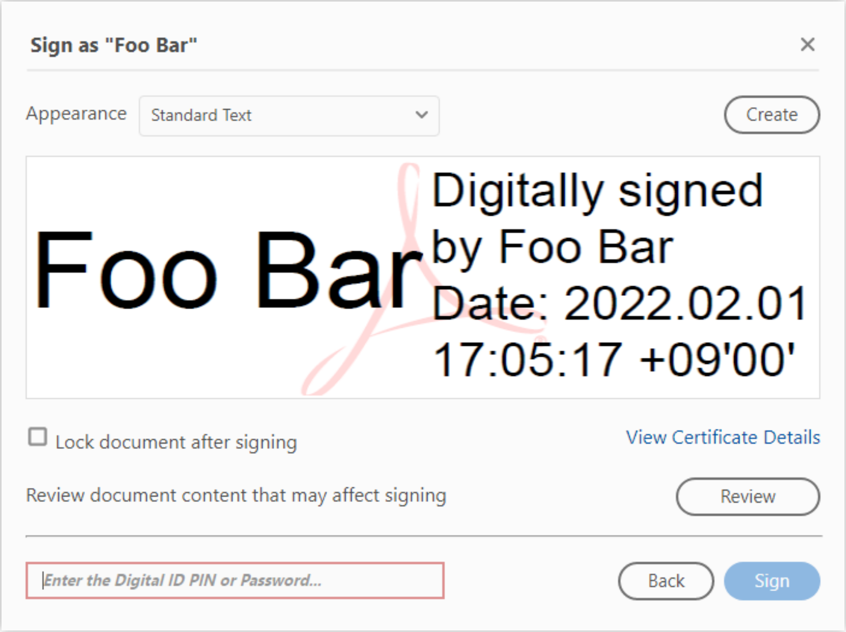
○ **ECDSA sample key** (PKCS#11 device)
Issued by: ECDSA sample key, Expires: 2025.09.27

● **Foo Bar** (PKCS#11 device)
Issued by: Foo Bar, Expires: 2031.02.20

○ **RSA sample key** (PKCS#11 device)
Issued by: RSA sample key, Expires: 2025.08.01

Step 6

If you are prompted to enter the PIN as shown in the figure below, enter the user PIN for SHALO AUTH. Then click [**Sign**].



Sign as "Foo Bar"

Appearance: Standard Text

Foo Bar Digitally signed by Foo Bar Date: 2022.02.01 17:05:17 +09'00'

□ Lock document after signing

Review document content that may affect signing

Enter the Digital ID PIN or Password...

Step 7

In the dialog box for saving the file that appears, specify a file where the electronically signed PDF file is stored.

Chapter 8

Using SHALO AUTH for SSH authentication

An SSH client that supports PKCS #11 can authenticate users through SHALO AUTH. In addition, even if software does not support PKCS #11, it can use SHALO AUTH to work with an authentication agent that does support PKCS #11.

This chapter explains how to authenticate users through SHALO AUTH in OpenSSH and PuTTY, which are typical examples of SSH software. It also explains how to use authentication agents provided by such software.

Topics in this chapter

1. What is SSH?
2. Preparing SSH keys for use
3. Preparing the authentication agent for use (Windows – OpenSSH)
4. Preparing the authentication agent for use (Windows – PuTTY-CAC)
5. Preparing the authentication agent for use (macOS)
6. Preparing the authentication agent for use (Linux)
7. Using SSH clients

8.1 What is SSH?

Secure Shell (SSH) is a communication protocol for communicating securely with remote hosts. Its two main uses are as follows:

- To log in to remote hosts
- To transfer files

You can use SSH when an **SSH server** is running on the remote host. Run an **SSH client** on the local PC, which connects to the SSH server on the remote host.

8.1.1 SSH clients

The following SSH applications are widely used:

- OpenSSH An SSH server and client typically used in major OSs
- PuTTY-CAC An SSH client for Windows that supports cryptographic tokens
- Tera Term A terminal software program for Windows that supports various control terminals
- WinSCP A file transfer software program for Windows that uses SSH

These software programs provide password-less user authentication through SHALO AUTH. OpenSSH and PuTTY-CAC use SHALO AUTH via the PKCS #11 module. Tera Term and WinSCP use SHALO AUTH indirectly with the help of the mechanism called an **authentication agent** provided by other software.

Of the above four programs, OpenSSH and PuTTY-CAC provide an authentication agent. Tera Term and WinSCP use PuTTY-CAC's authentication agent. This chapter explains these four software programs.

Notes on OpenSSH

⚠ To use the ECDSA, use OpenSSH 8.0p1 or later.
To see the version, run `ssh -V` on the terminal.

⚠ For restrictions on OpenSSH in each environment, see Section 11.4.

Note on PuTTY-CAC

⚠ Use PuTTY-CAC Release 0.70 Update 7 or later.

8.1.2 Authentication agent

An authentication agent is a resident program with secret keys and cryptographic tokens that is fully responsible for the authentication process. The following figure shows the relation between the authentication agent and applications.

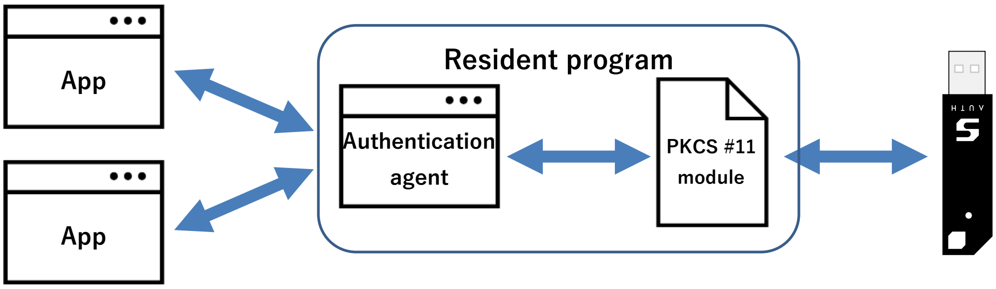


Figure 56 Use of SHALO AUTH via the authentication agent

The applications delegate the process to the authentication agent based on the secret keys or cryptographic tokens. When the authentication agent is used, individual applications do not have to have the secret keys or cryptographic tokens configured for them.

The authentication agent is also useful for cryptographic tokens. Because the authentication agent keeps running, users do not have to reenter their user PIN after entering it once.

Procedure for using the authentication agent

To enable the authentication agent to use SHALO AUTH, use the following procedure:

1. If the authentication agent is not running, start it.
2. Connect SHALO AUTH to the PC.
3. Load the PKCS #11 module into the authentication agent.

Any SSH clients that are run after the procedure above is done will authenticate users via the resident authentication agent.

Before you disconnect SHALO AUTH from the PC, or if you want to use SHALO AUTH directly in other applications, unload the PKCS #11 module from the authentication agent.

If you want the authentication agent to use SHALO AUTH again, reload the PKCS #11 module into the agent.

8.2 Preparing SSH keys for use

Users can be authenticated through SSH by having SSH keys at hand and:

- Registering the SSH private key with SHALO AUTH
- Registering the SSH public key with the remote host

8.2.1 Registering an SSH key with SHALO AUTH

Using SHALO Keyring, register the SSH key with SHALO AUTH. The following cryptography is available for the SSH key:

- RSA: Key length 2,048 to 4,096 bits
- ECDSA: P-256, P-384, and P-521

For how to generate the key with SHALO Keyring, see Section 4.3, and for how to register an existing key with SHALO AUTH, see Section 4.4.

8.2.2 Registering the SSH public key with the remote host

Add the SSH public key in the ~/.ssh/authorized_keys file of the remote host. If the SSH public key is stored in the *key.pub* file under the home directory, run the following command:

```
$ cat ~/key.pub >> ~/.ssh/authorized_keys↵
```

You can get the SSH public key from SHALO AUTH.
For how to do this with SHALO Keyring, see Section 4.6. For how to do this without SHALO Keyring, see Section 11.1.

8.3 Preparing the authentication agent for use (Windows – OpenSSH)

In Windows, OpenSSH's authentication agent, ssh-agent, is available in Git for Windows and Cygwin. However, the Windows 10-standard OpenSSH authentication agent cannot use SHALO AUTH.

8.3.1 Making the agent start automatically

Add the following statements to ~/.bashrc in individual environments so that ssh-agent will start automatically when Git Bash or Cygwin is run.

Data added to ~/.bashrc

```
export SLPKCS11FILE=pkcs11file
ssh-add -l > /dev/null 2>&1
if [ "$?" == 2 ] ; then
  SSH_AGENT_FILE=~/.ssh-agent
  test -f $SSH_AGENT_FILE && source $SSH_AGENT_FILE > /dev/null
  ssh-add -l > /dev/null 2>&1
  if [ "$?" == 2 ] ; then
    (umask 066; ssh-agent -P "/usr/lib/*,/usr/local/lib/*,$SLPKCS11FILE" > $SSH_AGENT_FILE)
    source $SSH_AGENT_FILE > /dev/null
    setx SSH_AUTH_SOCK "$SSH_AUTH_SOCK" > /dev/null
    setx SSH_AGENT_PID "$SSH_AGENT_PID" > /dev/null
  fi
fi

alias shalo-add='ssh-add -s $SLPKCS11FILE'
alias shalo-remove='ssh-add -e $SLPKCS11FILE'
```

pkcs11file in the first line must have one of the paths to the PKCS #11 module listed in the following table.

Environment	File path to the PKCS #11 module
Git for Windows 64 bit	/c/Users/*user-name*/shalo_pkcs11/x64/slpkcs11-mingw64.dll
Git for Windows 32 bit	/c/Users/*user-name*/shalo_pkcs11/x86/slpkcs11-mingw32.dll
Cygwin 64 bit	/cygdrive/c/Users/*user-name*/shalo_pkcs11/x64/slpkcs11-mingw64.dll
Cygwin 32 bit	/cygdrive/c/Users/*user-name*/shalo_pkcs11/x86/slpkcs11-mingw32.dll

The following table lists the paths to ~/.bashrc in Windows:

Environment	File path in Windows
Git for Windows	C:\Users\user-name\\.bashrc
Cygwin	*Cygwin-installation-directory*\home\user-name\\.bashrc

8.3.2 Registering or deregistering SHALO AUTH

The configuration in the previous subsection makes the following aliases available in Git Bash or Cygwin:

`shalo-add`	Loads the PKCS #11 module into ssh-agent.
`shalo-remove`	Unloads the PKCS #11 module from ssh-agent.

Registering SHALO AUTH with the authentication agent

Connect SHALO AUTH to the PC and then run `shalo-add`:

```
$ shalo-add↵
Enter passphrase for PKCS#11:  Input the user PIN.↵
Card added: /c/Users/user-name/shalo_pkcs11/x64/slpkcs11-mingw64.dll
```

Stopping the authentication agent from using SHALO AUTH

Run `shalo-remove`. This also applies to when you disconnect SHALO AUTH.

```
$ shalo-remove↵
Card removed: /c/Users/user-name/shalo_pkcs11/x64/slpkcs11-mingw64.dll
```

8.4 Preparing the authentication agent for use (Windows – PuTTY-CAC)

PuTTY-CAC's authentication agent, Pageant, is available in Windows.

8.4.1 How to start and stop

To start Pageant, run pageant.exe, one of the PuTTY-CAC files. You will not see any window even when running pageant.exe. Instead, a tray icon is added to the notification area, as shown in Figure 57.

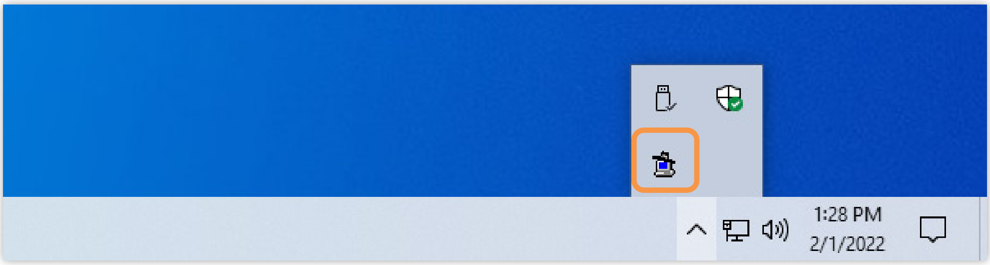


Figure 57 Tray icon for Pageant

Clicking the tray icon shows the context menu for Pageant.

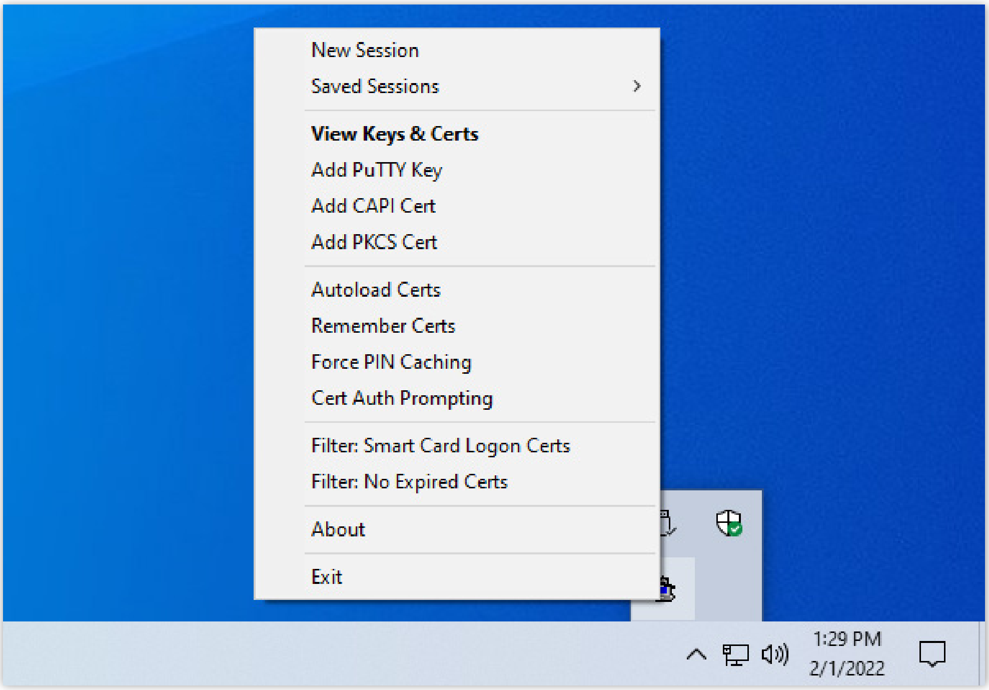


Figure 58 Context menu for Pageant

To exit Pageant, click [**Exit**] in this context menu.

8.4.2 Registering keys

To register SHALO AUTH with Pageant, you need to register SSH keys held by SHALO AUTH one by one. There is no way to register all the keys in SHALO AUTH at once.

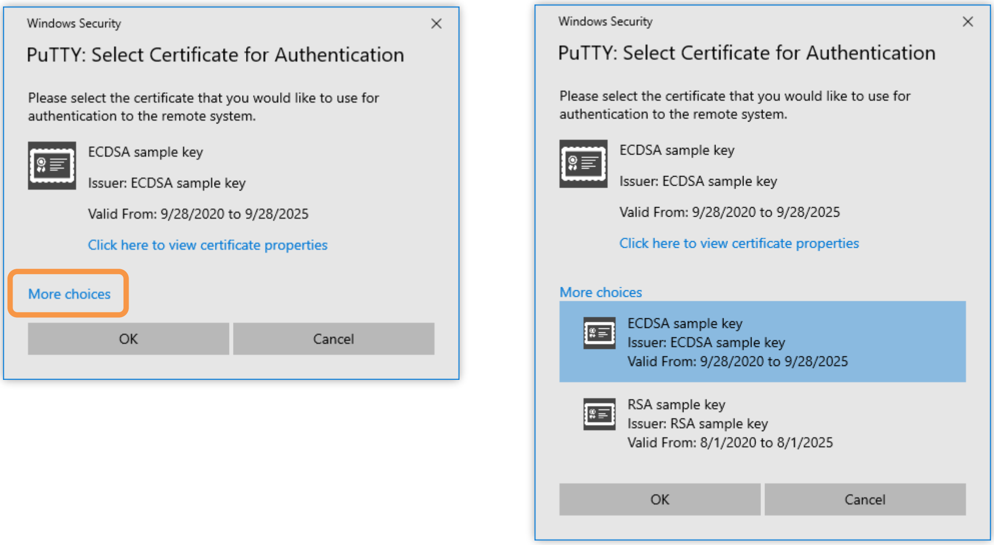
To register a single SSH key, use the following procedure:

1. In the Pageant context menu, select [**Add PKCS Cert**].
2. In the file selection dialog box, select the PKCS #11 module of SHALO AUTH.
3. From the list of all the certificates held by SHALO AUTH, select one certificate you want to register with Pageant.

In step 2, select a different PKCS #11 module depending on the 32-bit or 64-bit version of PuTTY-CAC, as shown in the table below. Select a file to suit your environment.

Software	File path to the PKCS #11 module
PuTTY-CAC 32-bit version	C:\Users\user-name\shalo_pkcs11\x86\slpkcs11-vc.dll
PuTTY-CAC 64-bit version	C:\Users\user-name\shalo_pkcs11\x64\slpkcs11-vc.dll

In step 3, you will see the certificate selection dialog box shown in the figure below (left). Clicking [**More choices**] shows all the available key certificates, as shown below (right). Select one key certificate from the list that you want to use for SSH authentication, and click [**OK**].



Windows Security

PuTTY: Select Certificate for Authentication

Please select the certificate that you would like to use for authentication to the remote system.

ECDSA sample key
Issuer: ECDSA sample key
Valid From: 9/28/2020 to 9/28/2025

Click here to view certificate properties

Windows Security

PuTTY: Select Certificate for Authentication

Please select the certificate that you would like to use for authentication to the remote system.

ECDSA sample key
Issuer: ECDSA sample key
Valid From: 9/28/2020 to 9/28/2025

Click here to view certificate properties

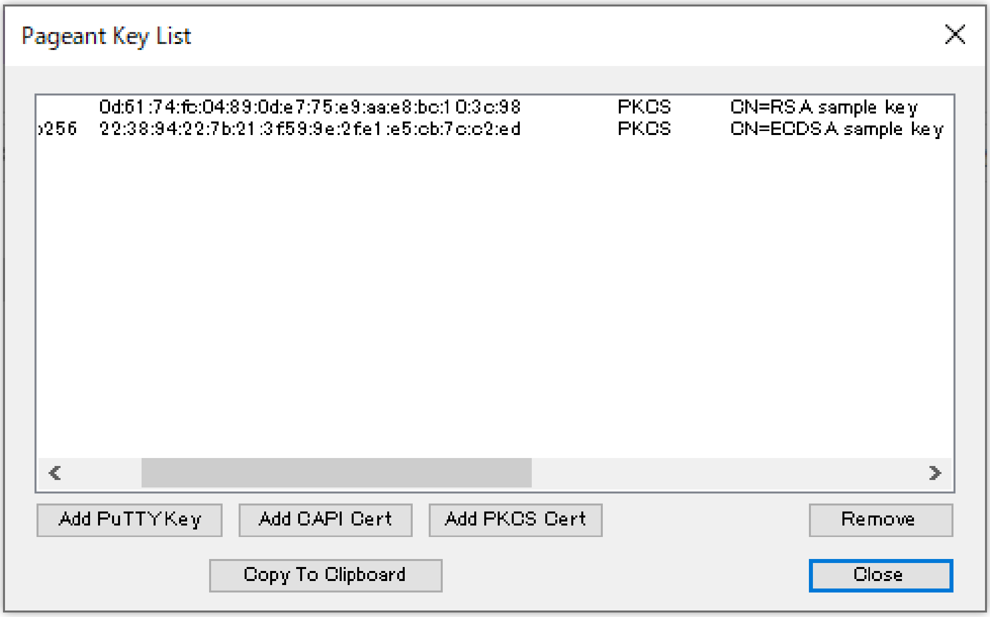
ECDSA sample key
Issuer: ECDSA sample key
Valid From: 9/28/2020 to 9/28/2025

RSA sample key
Issuer: RSA sample key
Valid From: 8/1/2020 to 8/1/2025

Figure 59 Selecting a key in the certificate selection dialog box

8.4.3 Viewing or removing registered keys

You can view the list of registered keys in the Pageant Key List window. To open the window, click [**View Keys & Certs**] in the Pageant context menu.



```
      0d:61:74:fc:04:89:0d:e7:75:e9:aa:e8:bc:10:3c:98   PKCS   CN=RSA sample key
›256  22:38:94:22:7b:21:3f:59:9e:2f:e1:e5:cb:7c:c2:ed   PKCS   CN=ECDSA sample key
```

Figure 60 List of keys registered with Pageant

Removing a registered key

To remove a registered key, in the window above, select the key and click [**Remove**].

8.4.4 Enabling the agent to load keys automatically

You can allow Pageant to automatically load the PKCS #11 module and SSH keys registered with Pageant at agent startup.

To enable this feature, in the Pageant context menu, select [**Remember Certs**] to add a check mark to this item.

When using SHALO AUTH, it is convenient to enable this feature and use it as follows:

- Start Pageant after connecting SHALO AUTH to the PC.
- Exit Pageant before disconnecting SHALO AUTH.

⚠ If SHALO AUTH is not connected when you start Pageant, the keys that are registered will be deregistered. In this case, reregister them.

8.5 Preparing the authentication agent for use (macOS)

The OpenSSH authentication agent, ssh-agent, is available for macOS.

The macOS-standard OpenSSH agent is configured to start ssh-agent automatically when the ssh-add command is run.
A socket for ssh-agent is set up in the SSH_AUTH_SOCK environment variable by launchd. We do not recommend that you stop the macOS-standard ssh-agent service.

Adding aliases

Add definitions to the shell configuration file. The target configuration files are listed in the following table.

Shell type	Configuration file name
Bash (default shell in macOS 10.14 Mojave or earlier)	~/.bashrc
Zsh (default shell in macOS 10.15 Catalina or later)	~/.zshrc

What is added to the configuration file

```
export SLPKCS11FILE=/usr/local/lib/libslpkcs11.dylib

alias shalo-add='ssh-add -s $SLPKCS11FILE'
alias shalo-remove='ssh-add -e $SLPKCS11FILE'
```

This addition will make the following aliases available in the terminal:

`shalo-add`	Loads the PKCS #11 module into ssh-agent.
`shalo-remove`	Unloads the PKCS #11 module from ssh-agent.

Registering SHALO AUTH with the authentication agent

Connect SHALO AUTH to the Mac and then run `shalo-add` once:

```
$ shalo-add↵
Enter passphrase for PKCS#11:  Input the user PIN.↵
Card added: /usr/local/lib/libslpkcs11.dylib
```

Stopping the authentication agent from using SHALO AUTH

Run `shalo-remove`. This also applies to when you disconnect SHALO AUTH.

```
$ shalo-remove↵
Card removed: /usr/local/lib/libslpkcs11.dylib
```

8.6 Preparing the authentication agent for use (Linux)

The OpenSSH authentication agent, ssh-agent, is available for Linux.

8.6.1 Making the agent start automatically

Add the following statements to `~/.bashrc` so that **`ssh-agent`** will start automatically and properly when you log in to Linux.

Data added to ~/.bashrc

```
export SLPKCS11FILE=/usr/lib/libslpkcs11.so

ssh-add -l > /dev/null 2>&1
if [ "$?" == 2 ] ; then
  SSH_AGENT_FILE=~/.ssh-agent
  test -f $SSH_AGENT_FILE && source $SSH_AGENT_FILE > /dev/null

  ssh-add -l > /dev/null 2>&1
  if [ "$?" == 2 ] ; then
    (umask 066; ssh-agent > $SSH_AGENT_FILE)
    source $SSH_AGENT_FILE > /dev/null
  fi
fi

alias shalo-add='ssh-add -s $SLPKCS11FILE'
alias shalo-remove='ssh-add -e $SLPKCS11FILE'
```

8.6.2 Registering or deregistering SHALO AUTH

The configuration in the previous subsection makes the following aliases available:

`shalo-add` Loads the PKCS #11 module into ssh-agent.

`shalo-remove` Unloads the PKCS #11 module from ssh-agent.

Registering SHALO AUTH with the authentication agent

Connect SHALO AUTH to the PC and then run **`shalo-add`**:

```
$ shalo-add↵
Enter passphrase for PKCS#11:  Input the user PIN.↵
Card added: /usr/lib/libslpkcs11.so
```

Stopping the authentication agent from using SHALO AUTH

Run **`shalo-remove`**. This also applies to when you disconnect SHALO AUTH.

```
$ shalo-remove↵
Card removed: /usr/lib/libslpkcs11.so
```

8.7 Using SSH clients

This section assumes that you have completed the following tasks described in Section 8.2:

- Register the SSH private key with SHALO AUTH.
- Register the SSH public key with the remote host.

8.7.1 Using ssh

`ssh` is a client program of OpenSSH. When ssh-agent is running, `ssh` automatically uses ssh-agent to authenticate users. For how to use SHALO AUTH without ssh-agent, see Section 10.1.

To use `ssh`, run the following:

```
ssh user-name@host-name
```

The following shows an example of connecting to a remote host with the host name “hostname” as a user which has the user name “username”:

```
$ ssh username@hostname↵
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-58-generic x86_64)

- omitted -
username@ubuntu:~$
```

Neither the user PIN nor password is necessary.
A prompt to enter the password implies that SHALO AUTH has not been registered properly with ssh-agent, or that the SSH public key has not been registered correctly with the remote host.

Warning upon the first connection

When you connect from the local PC to a remote host through `ssh` for the first time, `ssh` displays the message below. The purpose of this is to warn about connections to unknown remote hosts or spoofed host names, based on the public keys recorded by `ssh`, of remote hosts the tool previously connected to.

```
The authenticity of host 'hostname (IP-address)' can't be established.
ECDSA key fingerprint is SHA256:remote-host-fingerprint.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

If the fingerprint of the remote host is known, check if it is the same as the displayed one. If you can be sure that the destination is authentic, type **yes** and press the Enter key. Then, `ssh` will store this remote host and its fingerprint in the ~/.ssh/known_hosts file and start the user authentication process.

8.7.2 Using plink

`plink` is a command-line-based connectivity tool for PuTTY. When Pageant is running, `plink` automatically uses Pageant to authenticate users.

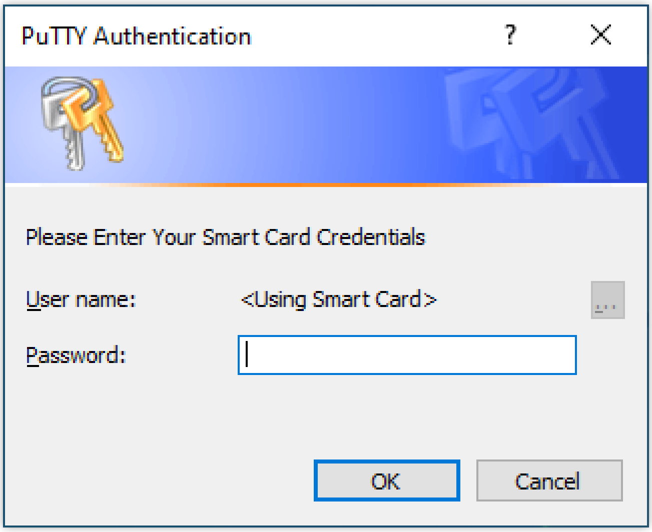
To run `plink`, add the PuTTY-CAC directory to the PATH environment variable, or run the following command in the PuTTY-CAC directory:

```
plink user-name@host-name
```

The following shows an example of connecting to a remote host with the host name “hostname” as a user which has the user name “username” by employing `plink` in PowerShell:

```
PS C:\PuTTY-CAC>plink username@hostname↵
Using username "username".
```

If you have not entered the user PIN in Pageant, an authentication window for PuTTY will appear as shown in the figure below. In [**Password**], enter the user PIN for SHALO AUTH and click [**OK**].



PuTTY Authentication

Please Enter Your Smart Card Credentials

User name: <Using Smart Card>

Password:

Figure 61 PuTTY authentication window

When the authentication is successful, the message below will appear. Pressing the Enter key establishes a connection with the remote host.

```
Access granted. Press Return to begin session. ↵
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-58-generic x86_64)

- omitted -
username@ubuntu:~$
```

Warning upon the first connection

When you connect from the local PC to a remote host through PuTTY for the first time, `plink` displays the message below. The purpose of this is to warn about connections to unknown remote hosts or spoofed host names, based on the public keys recorded by PuTTY, of remote hosts the tool previously connected to.

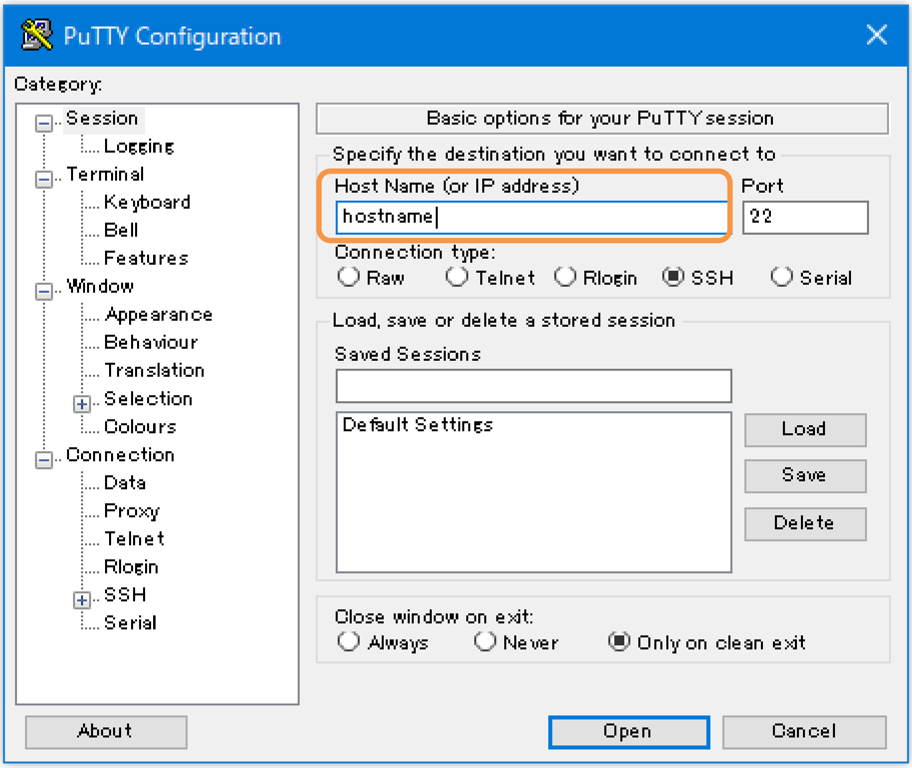
```
WARNING - POTENTIAL SECURITY BREACH!
The server's host key does not match the one PuTTY has
cached in the registry. This means that either the
server administrator has changed the host key, or you
have actually connected to another computer pretending
to be the server.
The new ssh-ed25519 key fingerprint is:
ssh-ed25519 255 remote-host-specific-data
If you were expecting this change and trust the new key,
enter "y" to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating
the cache, enter "n".
If you want to abandon the connection completely, press
Return to cancel. Pressing Return is the ONLY guaranteed
safe choice.
Update cached key? (y/n, Return cancels connection)
```

If the fingerprint of the remote host is known, check if it is the same as the displayed one. If you can be sure that the destination is authentic, type **y** and press the Enter key. Then, `plink` will store this remote host and its fingerprint in the registry and start the user authentication process.

8.7.3 Using putty

`putty` is a GUI-based connectivity tool for PuTTY. When Pageant is running, `putty` automatically uses Pageant to authenticate users. This subsection explains how to use `putty` to make an SSH connection in three steps. The description is based on PuTTY Release 0.74.

First, in the window below that appears when `putty` starts, type the name of the host to connect to in [**Host Name**], and click [**Open**]. As an example here, a connection is established to a remote host with the host name "hostname" as a user which has the user name "username."



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): hostname

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

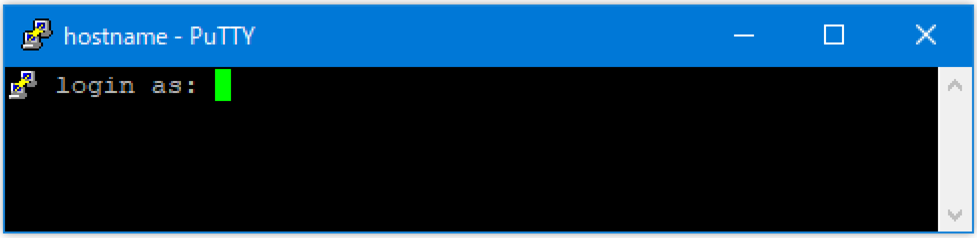
Saved Sessions:

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 62 Entering the destination host in the PuTTY configuration window

A PuTTY terminal window will then appear, prompting you to enter the name of the login user as shown in the figure below. At this time, input the user name and press the Enter key.

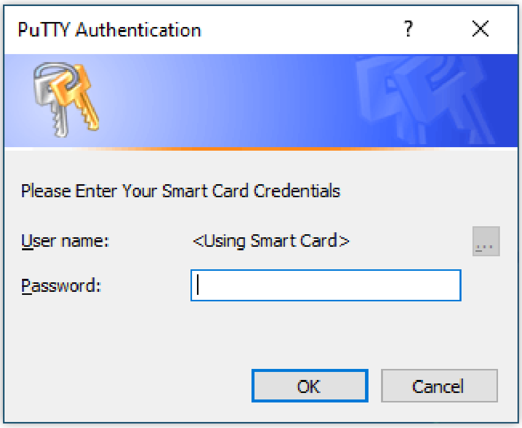


hostname - PuTTY

```
login as: 
```

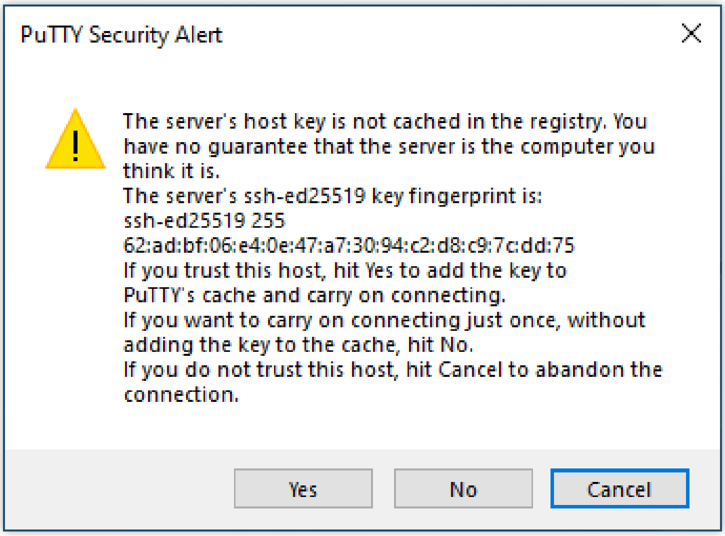
A prompt to enter the password in the terminal window implies that the key in SHALO AUTH has not been registered properly with Pageant, or that the SSH public key has not been registered correctly with the remote host.

Finally, if you have not entered the user PIN in Pageant, an authentication window for PuTTY will appear as shown in the figure below. In [**Password**], enter the user PIN for SHALO AUTH and click [**OK**]. When successfully authenticated by the remote host, you will see a message from the host in the PuTTY terminal window.



Warning upon the first connection

When you connect from the local PC to a remote host through PuTTY for the first time, PuTTY displays the message below. The purpose of this is to warn about connections to unknown remote hosts or spoofed host names, based on the public keys recorded by PuTTY, of remote hosts the tool previously connected to.



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255
62:ad:bf:06:e4:0e:47:a7:30:94:c2:d8:c9:7c:dd:75
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

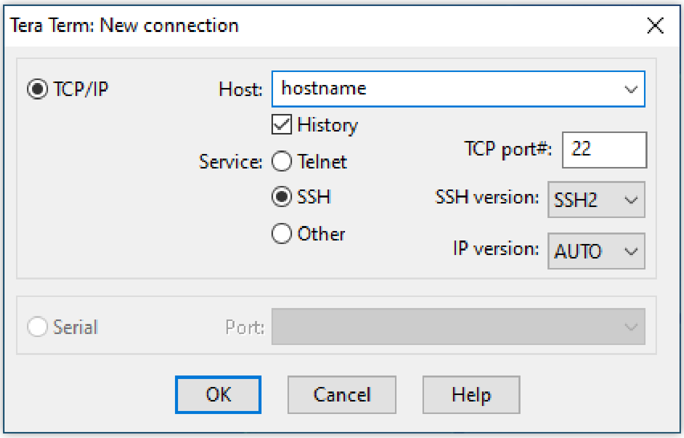
Figure 63 Warning about the server you connected to through PuTTY for the first time

If the fingerprint of the remote host is known, check if it is the same as the displayed one. If you can be sure that the destination is authentic, click [**Yes**]. Then, `putty` will store this remote host and its fingerprint in the registry and start the user authentication process.

8.7.4 Using Tera Term

With Pageant, Tera Term can authenticate users through SHALO AUTH. This subsection explains how to use Tera Term to make an SSH connection in three steps. The description is based on Tera Term version 4.105.

First, in the window shown in the figure below that appears when Tera Term starts, type the name of the host to connect to in [**Host**], and click [**OK**]. As an example here, a connection is established to a host with the host name “hostname” as a user which has the user name “username.”



Tera Term: New connection

● TCP/IP Host: hostname

☑ History

Service: ○ Telnet

● SSH

○ Other

TCP port#: 22

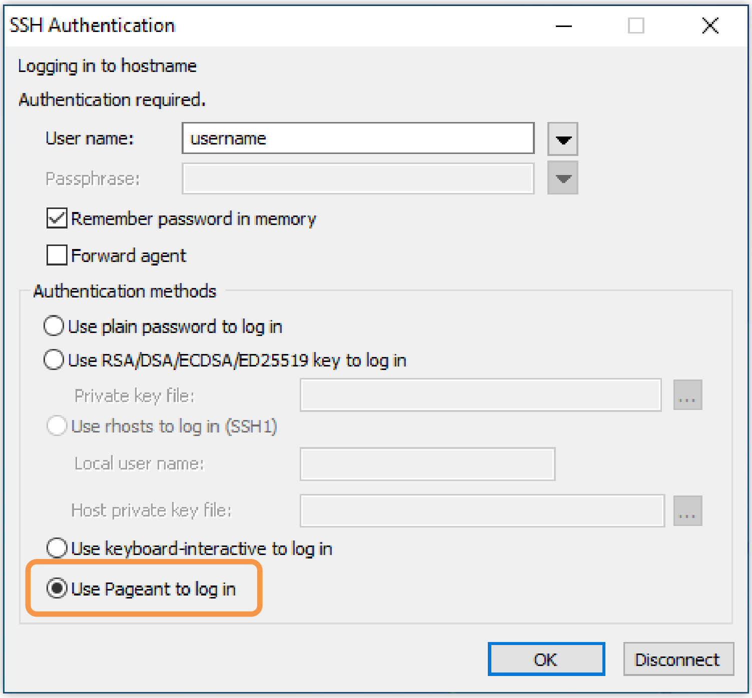
SSH version: SSH2

IP version: AUTO

○ Serial Port:

Figure 64 Entering the destination in Tera Term

Next, in the SSH Authentication window in the figure below, select [**Use Pageant to log in**], type the name of the login user in [**User name**], and click [**OK**].



SSH Authentication

Logging in to hostname

Authentication required.

User name: username

Passphrase:

☑ Remember password in memory

□ Forward agent

Authentication methods

○ Use plain password to log in

○ Use RSA/DSA/ECDSA/ED25519 key to log in

Private key file:

○ Use rhosts to log in (SSH1)

Local user name:

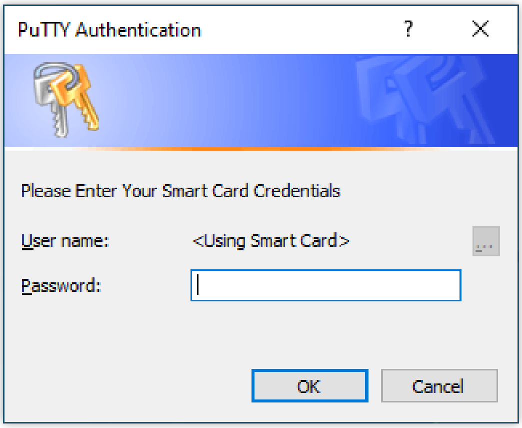
Host private key file:

○ Use keyboard-interactive to log in

● Use Pageant to log in

Figure 65 SSH settings in Tera Term

Finally, if you have not entered the user PIN in Pageant, an authentication window for PuTTY will appear as shown in the figure below. In [**Password**], enter the user PIN for SHALO AUTH and click [**OK**]. When successfully authenticated by the remote host, you will see a message from the host in the Tera Term terminal window.



PuTTY Authentication

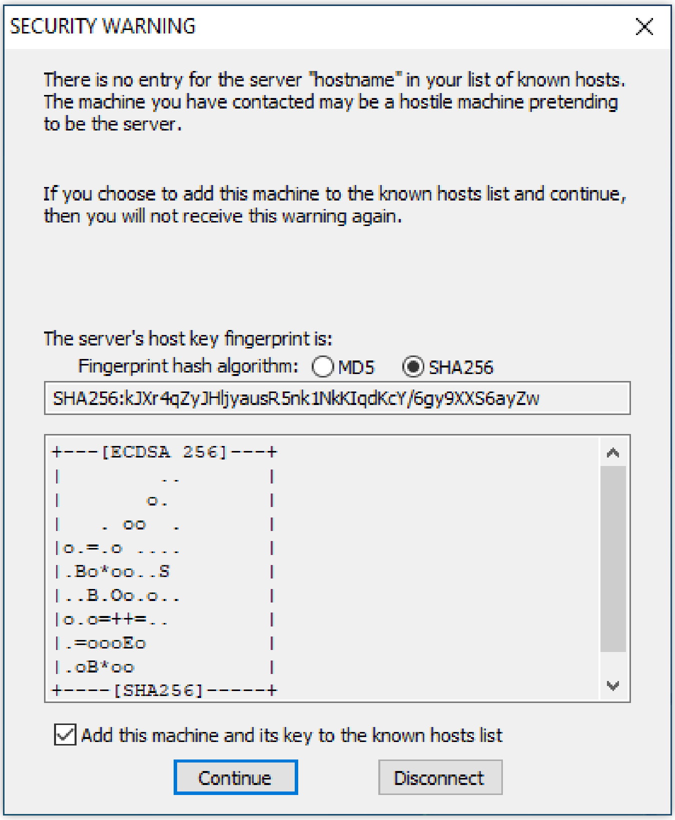
Please Enter Your Smart Card Credentials

User name: <Using Smart Card>

Password:

Warning upon the first connection

When you connect from the local PC to a remote host through Tera Term for the first time, Tera Term will display the message below. The purpose of this is to warn about connections to unknown remote hosts or spoofed host names, based on the public keys recorded by Tera Term, of remote hosts the tool previously connected to.



SECURITY WARNING

There is no entry for the server "hostname" in your list of known hosts. The machine you have contacted may be a hostile machine pretending to be the server.

If you choose to add this machine to the known hosts list and continue, then you will not receive this warning again.

The server's host key fingerprint is:

Fingerprint hash algorithm: ○ MD5 ● SHA256

SHA256:kJXr4qZyJHljyausR5nk1NkKIqdKcY/6gy9XXS6ayZw

```
+---[ECDSA 256]---+
|        ..       |
|       o.        |
|   . oo  .       |
|o.=.o ....       |
|.Bo*oo..S        |
|..B.Oo.o..       |
|o.o=++=..        |
|.=oooEo          |
|.oB*oo           |
+----[SHA256]-----+
```

☑ Add this machine and its key to the known hosts list

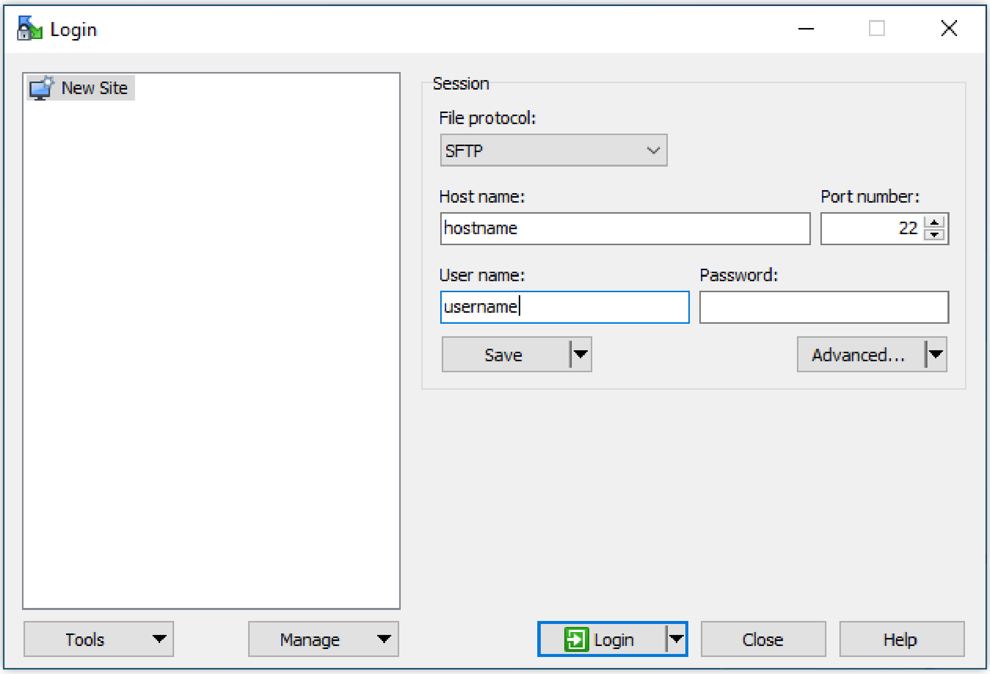
Figure 66 Warning about the server you connected to through Tera Term for the first time

If the fingerprint of the remote host is known, check if it is the same as the displayed one. If you can be sure that the destination is authentic, click [**Continue**]. Then, Tera Term will store this remote host and its fingerprint in a file and start the user authentication process.

8.7.5 Using WinSCP

WinSCP is a file transfer software program over SCP/SFTP and uses SSH. With Pageant, WinSCP can authenticate users through SHALO AUTH. The description is based on WinSCP version 5.15.10.

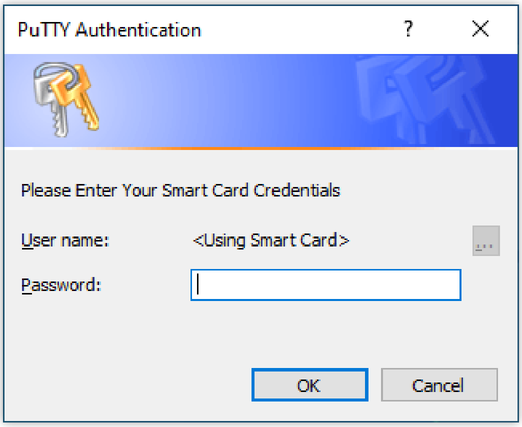
WinSCP is configured to use Pageant by default. In the window below, type names only in [**Host name**] and [**User name**], and click [**Login**]. As an example here, a connection is established to a host with the host name “hostname” as a user which has the user name “username.”



File protocol: SFTP
Host name: hostname
Port number: 22
User name: username
Password:

Figure 67 Entering the destination in WinSCP

If the PuTTY Authentication window appears, input the user PIN for SHALO AUTH in [**Password**] and click [**OK**]. When successfully authenticated by the remote host, you will see the home directory of the remote host in the WinSCP window.



PuTTY Authentication

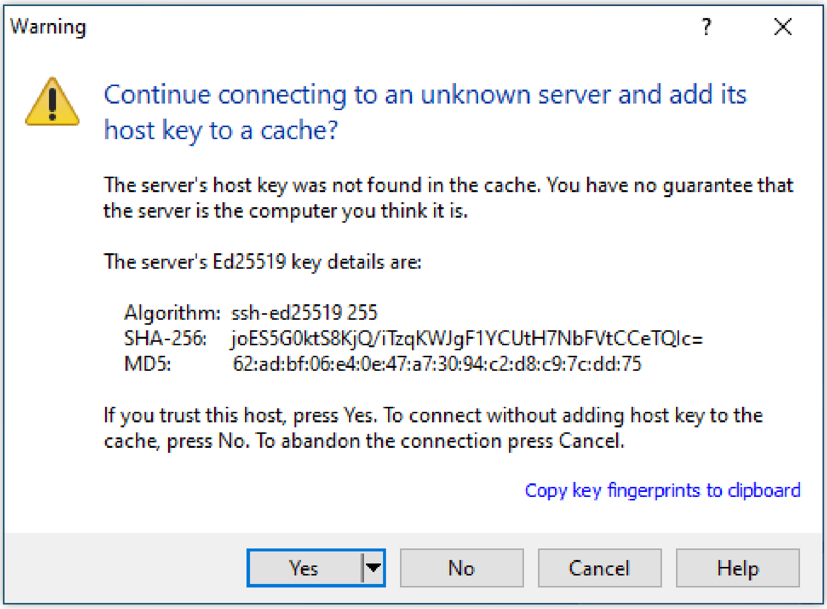
Please Enter Your Smart Card Credentials

User name: <Using Smart Card>

Password:

Warning upon the first connection

When you connect from the local PC to a remote host through WinSCP for the first time, WinSCP displays the message below. The purpose of this is to warn about connections to unknown remote hosts or spoofed host names, based on the public keys recorded by WinSCP, of remote hosts the tool previously connected to.



Warning

Continue connecting to an unknown server and add its host key to a cache?

The server's host key was not found in the cache. You have no guarantee that the server is the computer you think it is.

The server's Ed25519 key details are:

```
Algorithm:  ssh-ed25519 255
SHA-256:    joES5G0ktS8KjQ/iTzqKWJgF1YCUtH7NbFVtCCeTQIc=
MD5:        62:ad:bf:06:e4:0e:47:a7:30:94:c2:d8:c9:7c:dd:75
```

If you trust this host, press Yes. To connect without adding host key to the cache, press No. To abandon the connection press Cancel.

Figure 68 Warning about the server you connected to through WinSCP for the first time

Chapter 9

Using SHALO AUTH for SSH authentication in Git

Git is a distributed version control system that keeps track of, and manages, records of changes to source code of programs. It uses the SSH protocol to communicate securely.

This chapter addresses GitHub as a Git platform and explains how to use SHALO AUTH for SSH authentication to access GitHub from a Git client.

Topics in this chapter

1. Git and SSH authentication
2. Registering the SSH public key with GitHub
3. Testing SSH connections
4. Compatibility information of Git clients
5. Configuring Git clients

9.1 Git and SSH authentication

Git mainly uses the following two types of protocols for data transfer:

HTTP protocol	Uses a user name and password for authentication.
SSH protocol	Performs authentication with SSH keys.

In the HTTP (HTTPS) protocol, a repository is specified with `https://` as follows:

```
https://server/user/project.git
```

In the SSH protocol, a repository is specified with `ssh://` as follows:

```
ssh://user@server/project.git
```

Additionally in the SSH protocol, a repository can be specified in an abbreviated form like an SCP command:

```
user@server:project.git
```

To use the SSH protocol

Specify a repository in the form of the SSH protocol when cloning a remote repository. SSH authentication will always be used to transfer data between the cloned repository and the remote repository.

Changing the HTTP protocol used for a repository to the SSH protocol

Using the feature of changing the remote URL of a repository, you can change the HTTP protocol used for a repository to the SSH protocol.

To view the current remote URL, go to the repository in the terminal and run `git remote -v`. In GitHub, you will see the following:

```
$ git remote -v↵
origin  https://github.com/user-name/repository.git (fetch)
origin  https://github.com/user-name/repository.git (push)
```

To change the remote URL, use `git remote set-url` to specify a URL in the SSH protocol format for origin:

```
git remote set-url origin git@github.com:user-name/repository.git
```

Note that in hosting servers other than GitHub, the path structure in the URL differs from the one above.

9.2 Registering the SSH public key with GitHub

SHALO AUTH can use RSA, or P-256, P-384, or P-521 of ECDSA as SSH keys. Unlike the regular way to register SSH public keys with SSH remote hosts, register the keys with GitHub in a Web browser.

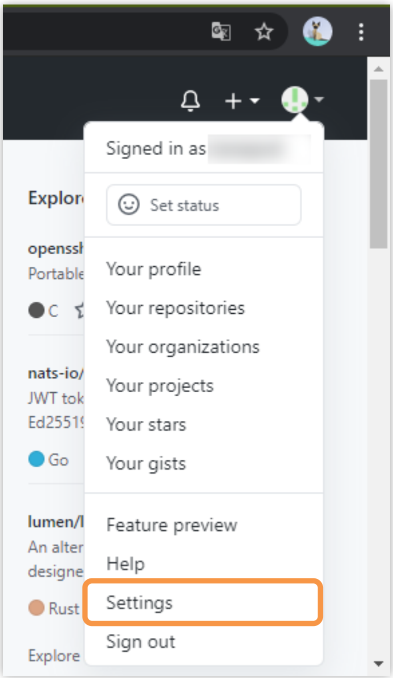
To do so, use the following procedure:

1. Open https://www.github.com in a Web browser and log in.
2. Click the profile image in the upper-right corner, and then click [**Settings**].
3. In the side bar on the left, click [**SSH and GPG keys**].
4. Click [**New SSH Key**].
5. Type the name of the key in [**Title**], enter the SSH key in [**Key**], and then click [**Add SSH key**].

The following explains the procedure together with screenshots.

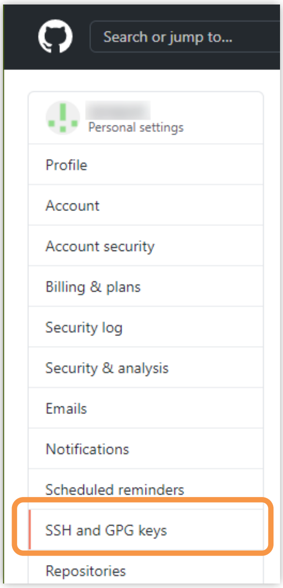
Steps 1 to 2

Log in to GitHub. Then, click the profile image in the upper-right corner and select [**Settings**].



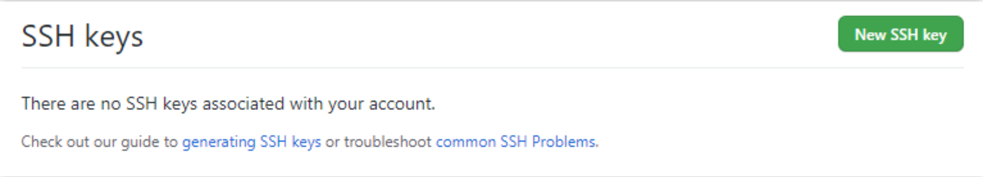
Step 3

In the side bar on the left, click [**SSH and GPG keys**].



Step 4

Click [**New SSH Key**].

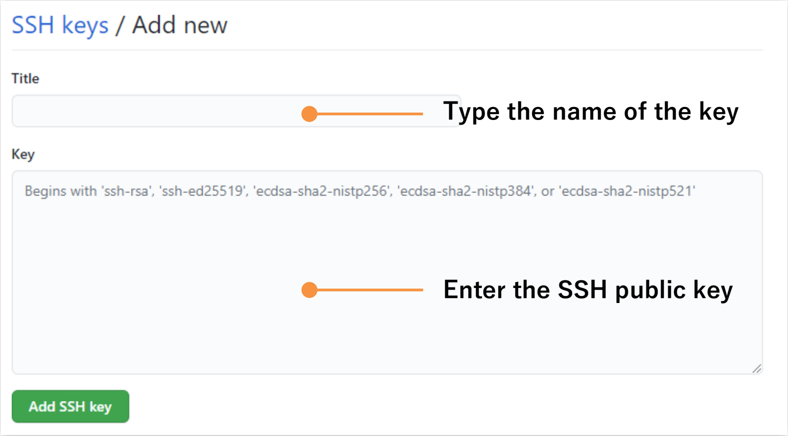


There are no SSH keys associated with your account.

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

Step 5

Type the name of the key in [**Title**], and enter the SSH public key in [**Key**]. Finally, click [**Add SSH key**]. For details about SSH public keys, see Section 4.6.


SSH keys / Add new
Title
Type the name of the key
Key
Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'
Enter the SSH public key
Add SSH key

9.3 Testing SSH connections

You can test an SSH connection with GitHub by connecting to it with the following settings:

Host name	github.com
User name	git

Make sure that you perform an SSH connection test with GitHub described in this section. Otherwise, your Git client will receive the warning when it connects to the SSH server for the first time, and will not work correctly.

9.3.1 When ssh-agent is used as the authentication agent

Register SHALO AUTH with ssh-agent and then run the following command:

```
ssh -T git@github.com
```

When the client connects to GitHub for the first time with the ssh command, the following messages are displayed:

```
The authenticity of host 'github.com (IP ADDRESS)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

This message asks you to verify the public key of the destination server in order to prevent server spoofing. Check that the key matches the public key of GitHub, type **yes**, and then press the Enter key.

GitHub's public keys are exposed at the following URL:
https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/githubs-ssh-key-fingerprints

Once you have been successfully authenticated, *user-name* in the following message is replaced with your GitHub account name, and the SSH connection is now established.

```
Hi user-name! You've successfully authenticated, but GitHub does not
provide shell access.
```

If SSH authentication fails, you will see the following message:

```
git@github.com: Permission denied (publickey).
```

In this case, see Section 11.5.9 as a reference and check the SSH public key registered with GitHub and your SHALO AUTH environment.

9.3.2 When Pageant is used as the authentication agent

Register SHALO AUTH with Pageant and then run the following command:

```
plink -T git@github.com
```

You will not be able to see the messages below in putty, the GUI-based connectivity tool, which closes the window quickly.

When the client connects to GitHub for the first time through plink, the following messages are displayed:

```
The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n)
```

This message asks you to verify the public key of the destination server in order to prevent server spoofing. In response, type **y** and press the Enter key.

plink does not present the public key of the server in SHA 256, so you will not be able to check that the key matches one of the public keys exposed in the following GitHub URL:
https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/githubs-ssh-key-fingerprints

Once you have been successfully authenticated, *user-name* in the following message is replaced with your GitHub account name, and the SSH connection is now established.

```
Hi user-name! You've successfully authenticated, but GitHub does not
provide shell access.
```

If SSH authentication fails, you will see the following message:

```
FATAL ERROR: No supported authentication methods available (server sent:
publickey)
```

In this case, check the SSH public key registered with GitHub and your SHALO AUTH environment.

9.4 Compatibility information of Git clients

This section provides a matrix to show whether SHALO AUTH will work with major software programs that have Git client functionality, in environments comprising a combination of certain OSs and authentication agents.

The explanations in this subsection are based on the information correct at the time of writing this manual.
It does not necessarily guarantee that these software programs will operate with SHALO AUTH.

In these operating environments, the authentication agents described in Chapter 8 use SHALO AUTH. In the following matrix, OpenSSH of Git for Windows is used in the Windows (ssh-agent) column.

Software	Windows (ssh-agent)	Windows (Pageant)	macOS (ssh-agent)	Linux (ssh-agent)
git command 2.30.1	✓	✓*1	✓	✓
GitHub Desktop 2.6.0	✓	✓*1	✓	—
GitKraken 7.4.1	N/A	✓*2	✓*2	✓*2
Sourcetree 3.3.9	N/A	✓*3	✓	—
TortoiseGit 2.11.0.0	✓	✓	—	—
Visual Studio 2017	N/A	N/A	—	—
Visual Studio 2019	✓	✓*1	—	—
Visual Studio 2019 for Mac	—	—	✓	—
Visual Studio Code	✓	✓*1	✓	✓
Xcode 12	—	—	N/A	—

✓ Available

N/A Not available

— Software does not support the OS

*1 Add the absolute path to plink.exe of PuTTY to the GIT_SSH environment variable (Subsection 9.5.1).

*2 In GitKraken, select [**Preferences**] > [**SSH**] and select the [**Use local SSH agent**] check box (Subsection 9.5.2).

*3 In Sourcetree, select [**Options**] > [**General**] and clear the [**Automatically start SSH agent when Sourcetree opens**] check box (Subsection 9.5.3).

When you use Pageant as the authentication agent, test the SSH connection with GitHub using plink. When you use ssh-agent, test the SSH connection using ssh.

9.5 Configuring Git clients

This section explains how to configure the software programs marked with *1 to *3 in Section 9.4. The Git clients with no asterisks do not require the configuration here.

9.5.1 GIT_SSH environment variable (only when Pageant is used in Windows)

The **git** command starts **ssh** internally. If the GIT_SSH environment variable is specified, the command starts and uses the program designated there, instead of **ssh**. Therefore, if you specify plink.exe in the GIT_SSH environment variable, you can use Pageant of PuTTY-CAC for authentication in Git.

This subsection explains how to specify the GIT_SSH environment variable in three steps.

First, type “edit environment variables” in the Windows search box as shown in the following figure, and click [**Edit environment variables for your account**].

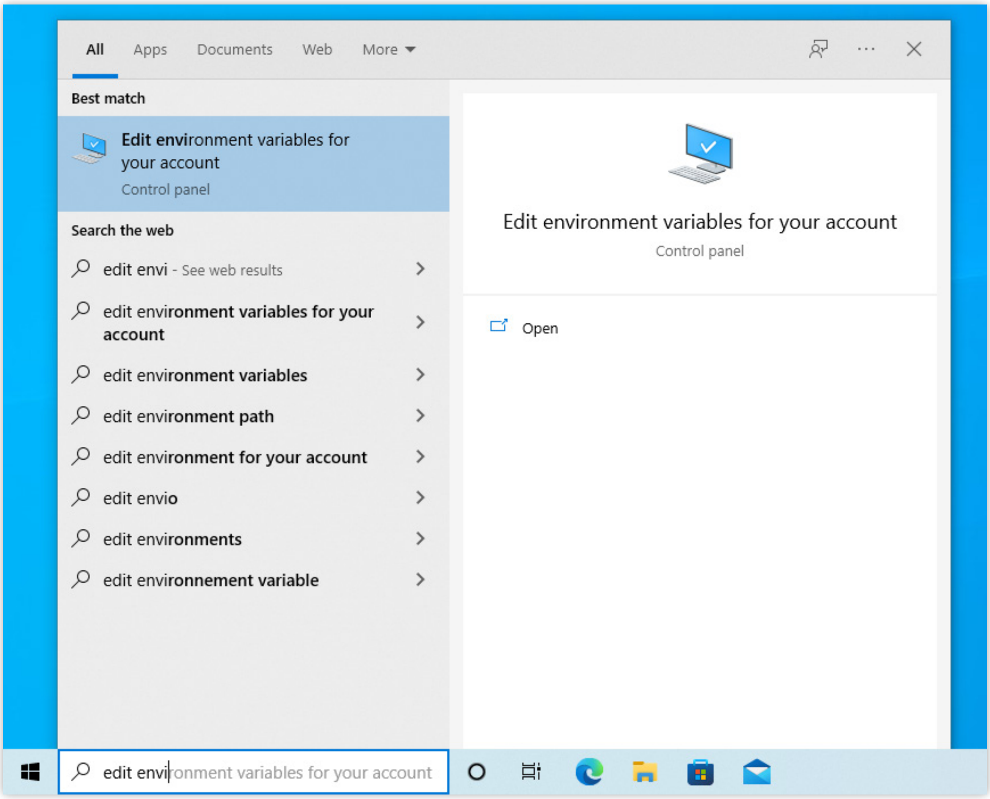
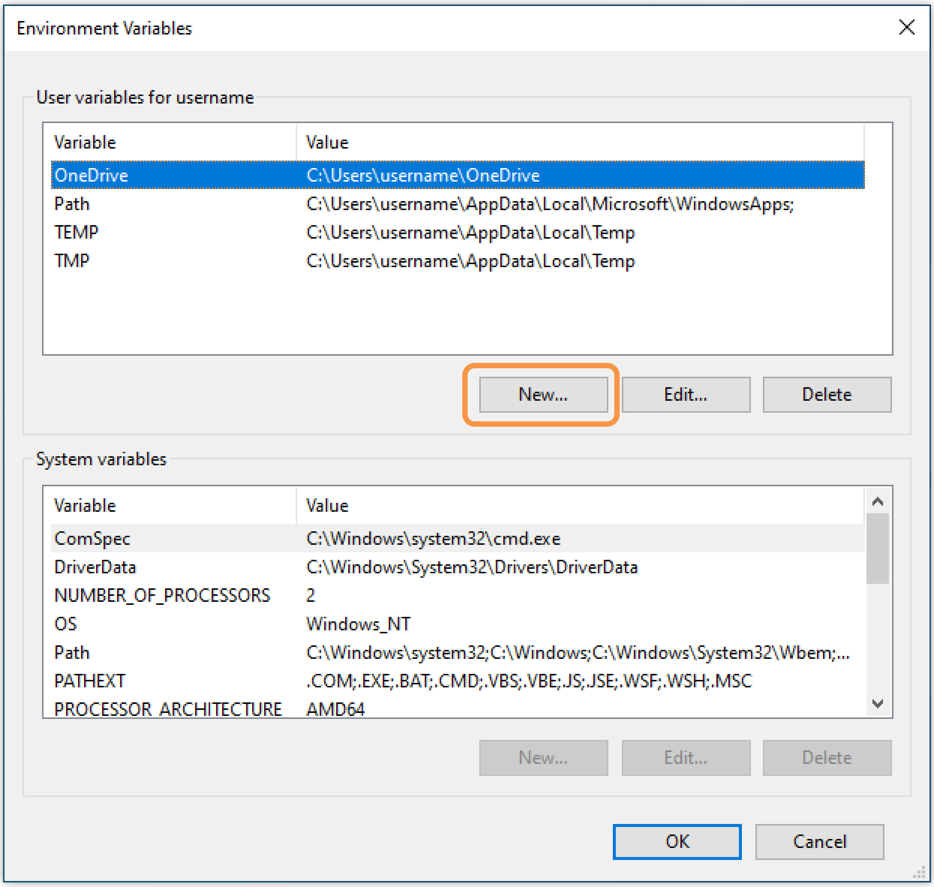

All
Apps
Documents
Web
More
Best match
Edit environment variables for your account
Control panel
Search the web
edit envi - See web results
edit environment variables for your account
edit environment variables
edit environment path
edit environment for your account
edit envio
edit environments
edit environnement variable
Edit environment variables for your account
Control panel
Open
edit environment variables for your account

Figure 69 Searching for "edit environment variables" via the search box

Next, in the Environment Variables window, click [**New...**] under “User variables for *your account name.*”

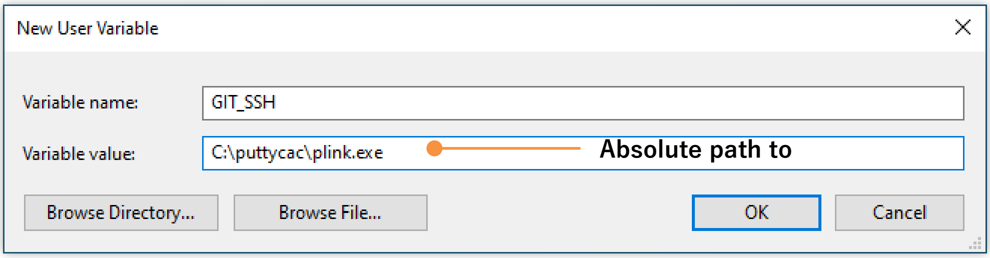


Variable	Value
OneDrive	C:\Users\username\OneDrive
Path	C:\Users\username\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\username\AppData\Local\Temp
TMP	C:\Users\username\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	2
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Figure 70 Adding an environment variable

Finally, type **GIT_SSH** in [**Variable name**], click [**Browse File...**], and select PuTTY-CAC's plink.exe. Then click [**OK**].



Variable name: GIT_SSH

Variable value: C:\puttycac\plink.exe

Absolute path to

Figure 71 Creating the GIT_SSH environment variable

9.5.2 GitKraken

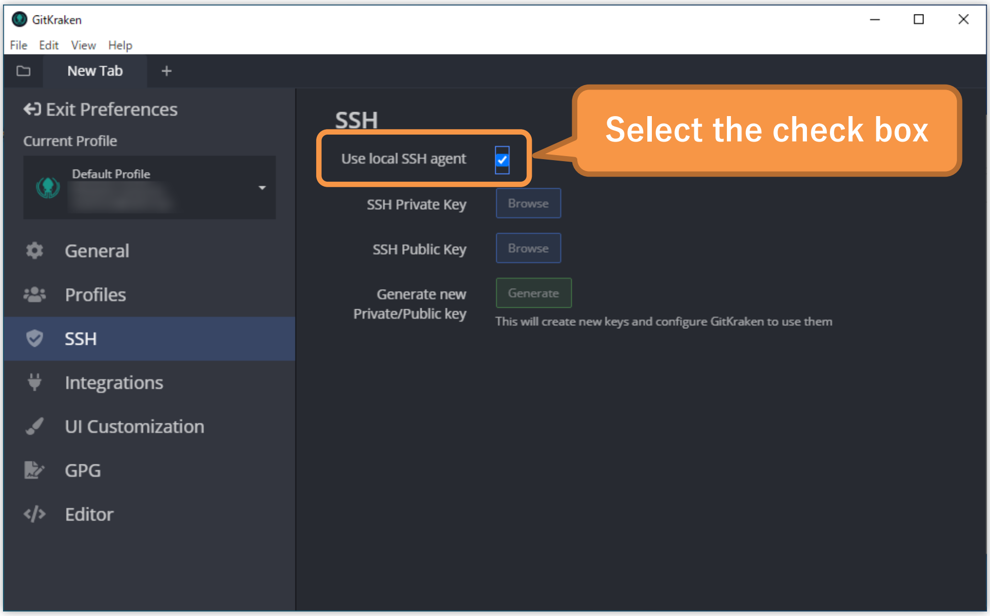
This subsection is based on GitKraken 7.4.1. Screen layouts and behavior may vary in other versions of the program.

GitKraken can use Pageant in the Windows version, and can use ssh-agent in the macOS and Linux versions. However, these authentication agents are disabled by default.

To allow GitKraken to use them, carry out the following procedure:

1. In the GitKraken menu, click [**File**] > [**Preferences...**].
2. In the window shown in the figure below, select [**Preferences**] > [**SSH**].
3. Select the [**Use local SSH agent]** check box.

The following figure shows where you configure the setting in the GitKraken window.



Use local SSH agent ☑

SSH Private Key Browse

SSH Public Key Browse

Generate new Private/Public key Generate

This will create new keys and configure GitKraken to use them

Select the check box

Figure 72 SSH setting in GitKraken

9.5.3 Sourcetree (Windows only)

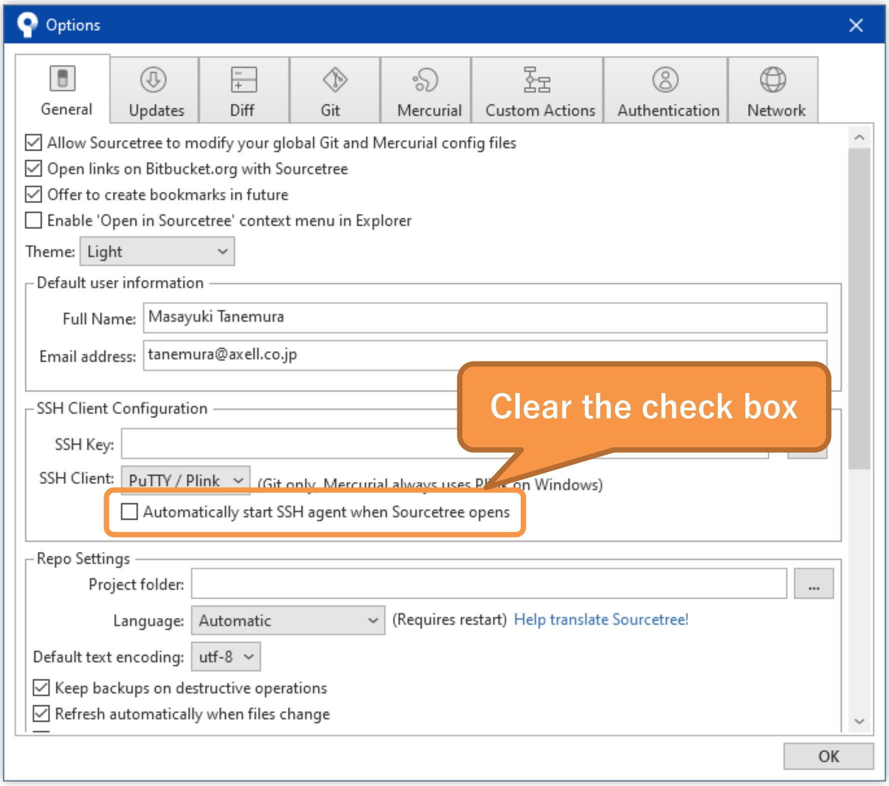
This subsection is based on Sourcetree 3.9.1. Screen layouts and behavior may vary in other versions of the program.

Sourcetree for Windows provides PuTTY, which does not support PKCS #11, and starts Pageant for that PuTTY version at startup. You need to prevent this built-in Pageant from starting in Sourcetree for Windows.

To prevent Pageant from starting at startup of Sourcetree, use the following procedure:

1. In the Sourcetree menu, click [**Tools**] > [**Options**].
2. In the Options window in the figure below, select the [**General**] tab.
3. Clear the [**Automatically start SSH agent when Sourcetree opens**] check box.

The following figure shows where you configure the setting in Sourcetree.



☑ Allow Sourcetree to modify your global Git and Mercurial config files
☑ Open links on Bitbucket.org with Sourcetree
☑ Offer to create bookmarks in future
☐ Enable 'Open in Sourcetree' context menu in Explorer
Theme: Light

Default user information
Full Name: Masayuki Tanemura
Email address: tanemura@axell.co.jp

SSH Client Configuration
SSH Key:
SSH Client: PuTTY / Plink (Git only. Mercurial always uses Pl on Windows)
☐ Automatically start SSH agent when Sourcetree opens

Repo Settings
Project folder:
Language: Automatic (Requires restart) Help translate Sourcetree!
Default text encoding: utf-8
☑ Keep backups on destructive operations
☑ Refresh automatically when files change

Figure 73 SSH setting in Sourcetree

Chapter 10

Tips for better use

This chapter gives information for making better use of SHALO AUTH.

Topics in this chapter

1. Using SHALO AUTH from OpenSSH without an authentication agent
2. Using SHALO AUTH in remote hosts accessed via SSH
3. Using SHALO AUTH in remote hosts accessed through Remote Desktop

10.1 Using SHALO AUTH from OpenSSH without an authentication agent

This section explains how to use SHALO AUTH in OpenSSH without any authentication agent. Although this approach has a limitation that multiple `ssh` instances cannot use SHALO AUTH at one time, it is useful in restricted environments where no authentication agent is allowed.

There are two ways to do this:

- Use the `-I` option of ssh.
- Register SHALO AUTH in the ssh configuration file (~/.ssh/config).

`-I` option of ssh

You can specify the PKCS #11 module with the `-I` option in the ssh command. The format for this is as follows:

```
ssh -I pkcs11file user-name@host-name
```

When you run the command, you will see the "Enter PIN for '*label-of-SHALO-AUTH*'" message. Then, input the SHALO AUTH user PIN and press the Enter key.

The example below shows the result of command execution. In this example, the SHALO AUTH label is "Foo's Token," and a connection is established with the remote host with the host name "hostname" as a user which has the user name "username."

```
$ ssh -I pkcs11file username@hostname↵
Enter PIN for 'Foo's Token': Input the user PIN↵
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)
- omitted -
```

From the following table, select and specify the path to the file to suit your environment for *pkcs11file*.

Environment		File path to the PKCS #11 module
Windows	Gir for Windows 32 bit	`/c/Users/`*`user-name`*`/shalo_pkcs11/x86/slpkcs11-mingw32.dll`
	Gir for Windows 64 bit	`/c/Users/`*`user-name`*`/shalo_pkcs11/x64/slpkcs11-mingw64.dll`
	Cygwin 32 bit	`/cygdrive/c/Users/`*`user-name`*`/shalo_pkcs11/x86/slpkcs11-mingw32.dll`
	Cygwin 64 bit	`/cygdrive/c/Users/`*`user-name`*`/shalo_pkcs11/x64/slpkcs11-mingw64.dll`
macOS		`/usr/local/lib/libslpkcs11.dylib`
Linux		`/usr/lib/libslpkcs11.so`

When SLPKCS11FILE is added to the shell configuration file as described in Sections 8.3, 8.5, or 8.6, you can specify `$SLPKCS11FILE` for *pkcs11file*.

~/.ssh/config

~/.ssh/config is the configuration file for `ssh`. By specifying the `-I` option equivalent of `ssh` in the configuration file, you can omit this option in the `ssh` command. However, you still have to provide the user PIN when ssh is run.

> This approach allows you to use SHALO AUTH from the git command even when no authentication agent is used. However, a repository that uses Git LFS is not available. This also applies to GUI-based Git clients for which you cannot provide the user PIN.

In ~/.ssh/config, configure the settings for each remote host the client connects to. The minimum configuration for enabling the tool to use the PKCS #11 module is shown below. Change the italicized strings according to your environment.

```
Host name
     Hostname IP-address-or-remote-host-address
     PKCS11Provider absolute-path-to-PKCS#11-module
```

From the following table, select and specify the absolute path to the PKCS#11 module to suit your environment.

Environment		File path to the PKCS #11 module
Windows	Gir for Windows 32 bit	`/c/Users/`*`user-name`*`/shalo_pkcs11/x86/slpkcs11-mingw32.dll`
	Gir for Windows 64 bit	`/c/Users/`*`user-name`*`/shalo_pkcs11/x64/slpkcs11-mingw64.dll`
	Cygwin 32 bit	`/cygdrive/c/Users/`*`user-name`*`/shalo_pkcs11/x86/slpkcs11-mingw32.dll`
	Cygwin 64 bit	`/cygdrive/c/Users/`*`user-name`*`/shalo_pkcs11/x64/slpkcs11-mingw64.dll`
macOS		`/usr/local/lib/libslpkcs11.dylib`
Linux		`/usr/lib/libslpkcs11.so`

10.2 Using SHALO AUTH in remote hosts accessed via SSH

When an SSH connection with a remote host is established through an authentication agent, you can allow the remote host to use the authentication agent connected to the local PC. This is called **ssh agent forwarding**.

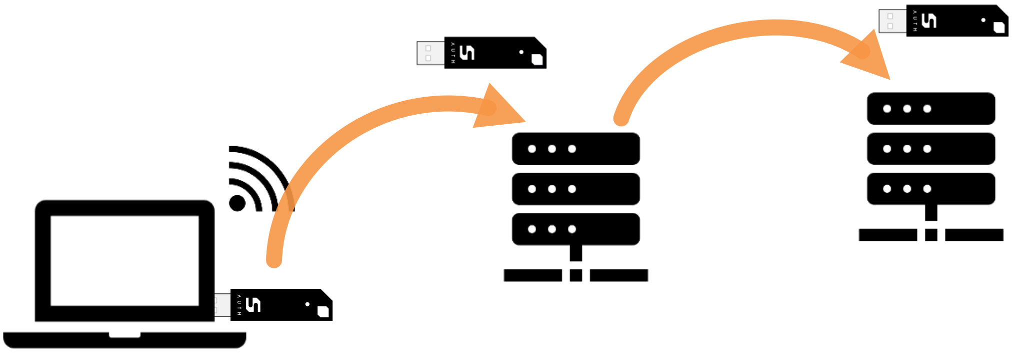


Figure 74 Bringing the authentication agent on the local PC into the remote host

For this purpose, the remote host must be configured to allow ssh agent forwarding. SSH clients `ssh`, `plink`, and `putty` all support this feature.

The local PC can still use SHALO AUTH while the remote host is still connected.

The remote host accessed via SSH can only use the functionality provided by the authentication agent. It cannot use the U2F security key functionality.

Configuring the SSH server on the remote host

Modify the configuration file for the SSH server on the remote host. In the sshd_config configuration file, enable AllowAgentForwarding as follows:

sshd_config

```
AllowAgentForwarding yes
```

Establishing a connection with ssh

To enable ssh agent forwarding in the ssh command, add the `-A` option:

```
ssh -A user-name@host-name
```

The following command shows an example of testing an SSH connection with a GitHub account that uses a different SSH key to that of the remote host after the client connects to the host:

```
$ ssh -A username@hostname↵
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.8.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Feb 22 19:06:07 2021 from 192.168.1.1
username@hostname:~$ ssh -T git@github.com↵
Hi username! You've successfully authenticated, but GitHub does not provide
shell access.
```

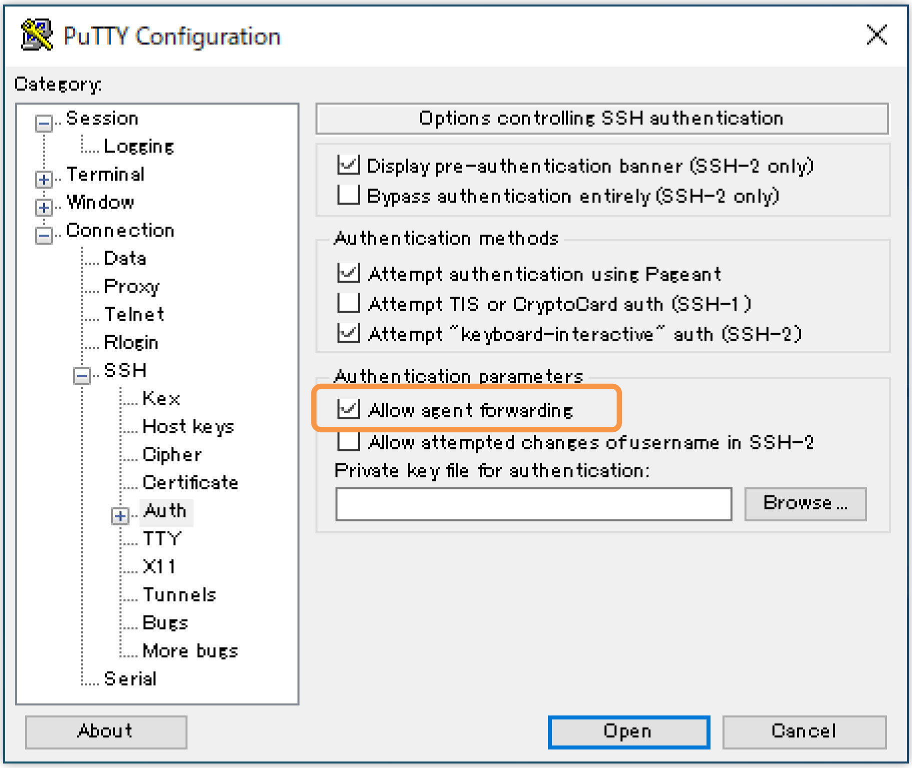
Establishing a connection with plink

Similar to the ssh command, plink's `-A` option enables ssh agent forwarding.

```
plink -A user-name@host-name
```

Establishing a connection with putty

In putty, click [**Connection**] > [**SSH**] > [**Auth**] and select the [**Allow agent forwarding**] check box, and then make a connection with the remote host.



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
□ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
□ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☑ Allow agent forwarding
□ Allow attempted changes of username in SSH-2

Private key file for authentication:

Figure 75 Enabling ssh agent forwarding in PuTTY

10.3 Using SHALO AUTH in remote hosts accessed through Remote Desktop

With Windows Remote Desktop, a remote PC accessed through Remote Desktop can use a SHALO AUTH device that is connected to the local accessing PC. This can be achieved using a feature called **RemoteFX USB redirection**.

When SHALO AUTH is redirected to the remote PC accessed through Remote Desktop, the PC can work with SHALO AUTH in the same way as a device disconnected from the local accessing PC and connected directly to the remote PC. The SHALO AUTH dedicated software and the PKCS #11 module must be installed in the remotely accessed PC.

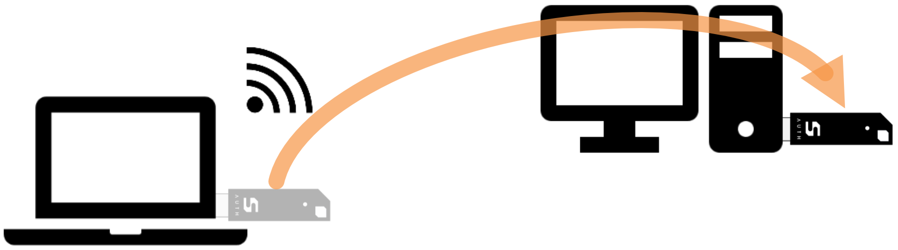


Figure 76 Disconnecting SHALO AUTH from the local PC and connecting it to the remote PC

⚠ The local accessing PC cannot use a SHALO AUTH device that has been redirected to the remote PC accessed through Remote Desktop.

⚠ The remote PC accessed through Remote Desktop can use the general security key functionality (PKCS #11) only. It cannot use the U2F security key functionality.

The requirements for using RemoteFX USB redirection are as follows:

Remote PC	Windows 10 Pro or Windows 10 Enterprise
Local PC	Windows 10 Pro or Windows 10 Enterprise

⚠ The feature is not available in Windows 10 Home edition or macOS.

The rest of this section explains the environmental settings for local accessing and remotely accessed PCs, followed by how to redirect SHALO AUTH.

10.3.1 Configuring the remotely accessed PC

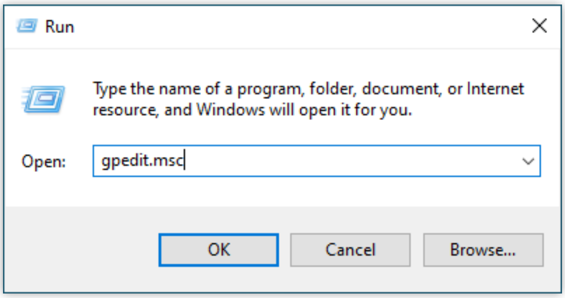
On the remotely accessed PC, use the following procedure:

1. Start the Local Group Policy Editor.
2. In the left pane, click and expand the following items: [**Computer Configuration**] > [**Administrative Templates**] > [**Windows Components**] > [**Remote Desktop Services**] > [**Remote Desktop Session Host**] > [**Device and Resource Redirection**]
3. Double-click [**Do not allow supported Plug and Play device redirection**].
4. Select the [**Disabled**] radio button and click [**OK**].

The following explains the procedure together with screenshots.

Step 1

Right-click the Start button (or press the Windows key+ X) and select [**Run**]. In the following window, type gpedit.msc and click [**OK**].

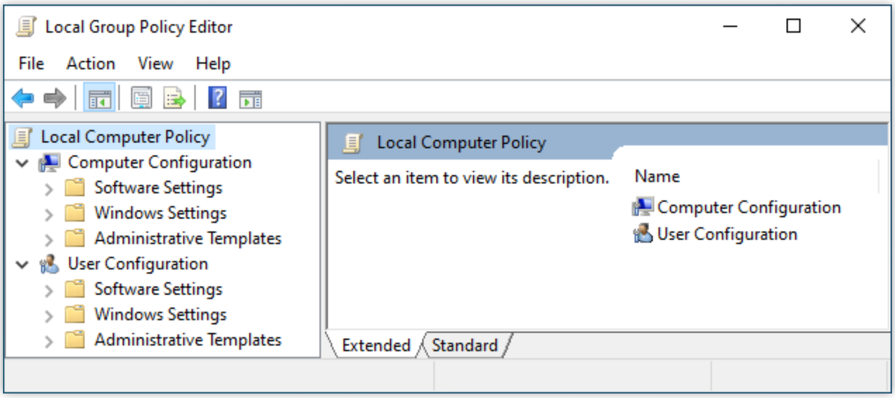


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: gpedit.msc

Step 2

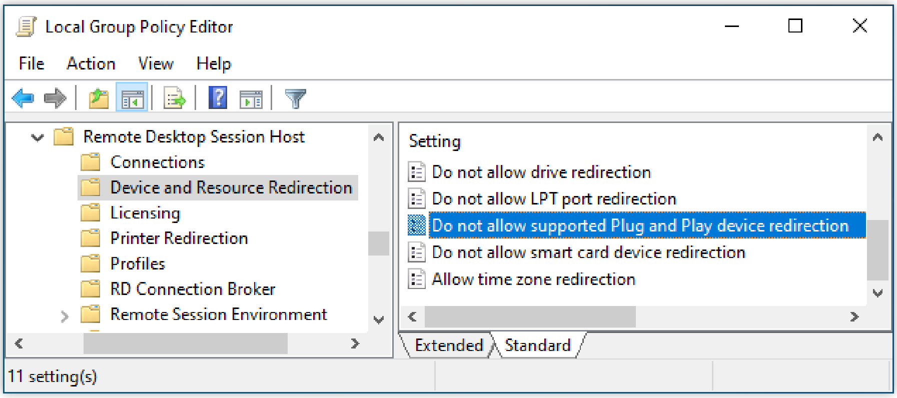
The Local Group Policy Editor appears as shown in the figure below. In the left pane of the window, click and expand the following items: [**Computer Configuration**] > [**Administrative Templates**] > [**Windows Components**] > [**Remote Desktop Services**] > [**Remote Desktop Session Host**] > [**Device and Resource Redirection**]



Select an item to view its description.

Step 3

In the window shown in the following figure, double-click [**Do not allow supported Plug and Play device redirection**].

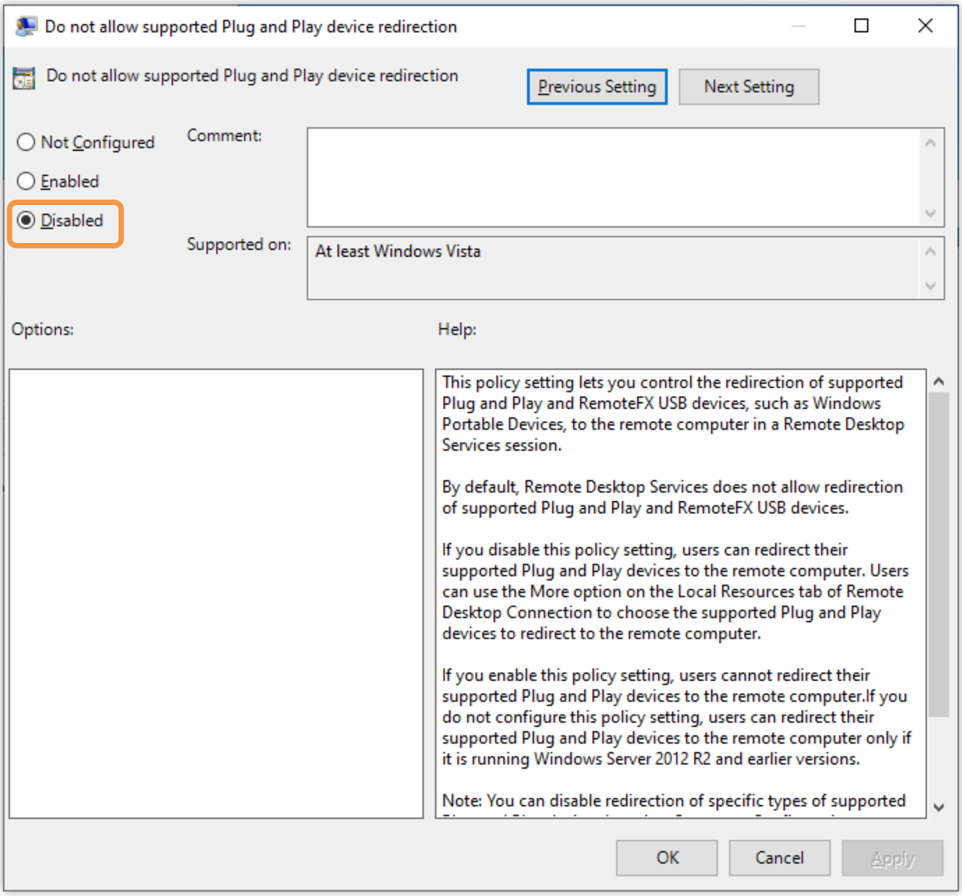

Local Group Policy Editor
File
Action
View
Help
Remote Desktop Session Host
Connections
Device and Resource Redirection
Licensing
Printer Redirection
Profiles
RD Connection Broker
Remote Session Environment
Extended
Standard

Setting
Do not allow drive redirection
Do not allow LPT port redirection
Do not allow supported Plug and Play device redirection
Do not allow smart card device redirection
Allow time zone redirection

11 setting(s)

Step 4

The window below will appear. Select the [**Disabled**] radio button and click [**OK**] here.



Do not allow supported Plug and Play device redirection

○ Not Configured
○ Enabled
● Disabled

Supported on: At least Windows Vista

This policy setting lets you control the redirection of supported Plug and Play and RemoteFX USB devices, such as Windows Portable Devices, to the remote computer in a Remote Desktop Services session.

By default, Remote Desktop Services does not allow redirection of supported Plug and Play and RemoteFX USB devices.

If you disable this policy setting, users can redirect their supported Plug and Play devices to the remote computer. Users can use the More option on the Local Resources tab of Remote Desktop Connection to choose the supported Plug and Play devices to redirect to the remote computer.

If you enable this policy setting, users cannot redirect their supported Plug and Play devices to the remote computer.If you do not configure this policy setting, users can redirect their supported Plug and Play devices to the remote computer only if it is running Windows Server 2012 R2 and earlier versions.

Note: You can disable redirection of specific types of supported

10.3.2 Configuring the local accessing PC

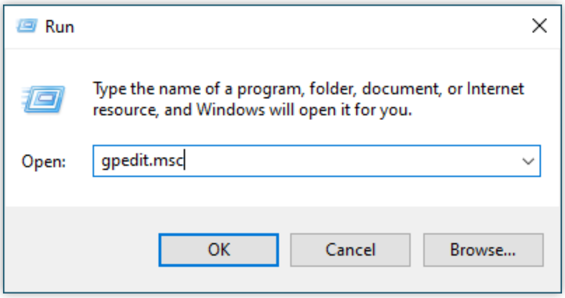
In the local accessing PC, use the following procedure:

1. Start the Local Group Policy Editor.
2. In the left pane, click and expand the following items: [**Computer Configuration**] > [**Administrative Templates**] > [**Windows Components**] > [**Remote Desktop Services**] > [**Remote Desktop Connection Client**] > [**RemoteFX USB Device Redirection**]
3. Double-click [**Allow RDP redirection of other supported RemoteFX USB devices from this computer**].
4. Select the [**Enabled**] radio button, select [**Administrators and Users**] under RemoteFX USB Redirection Access Rights, and click [**OK**].
5. Restart Windows.

The following explains the procedure together with screenshots.

Step 1

Right-click the Start button (or press the Windows key + X) and select [**Run**]. In the following window, type gpedit.msc and click [**OK**].



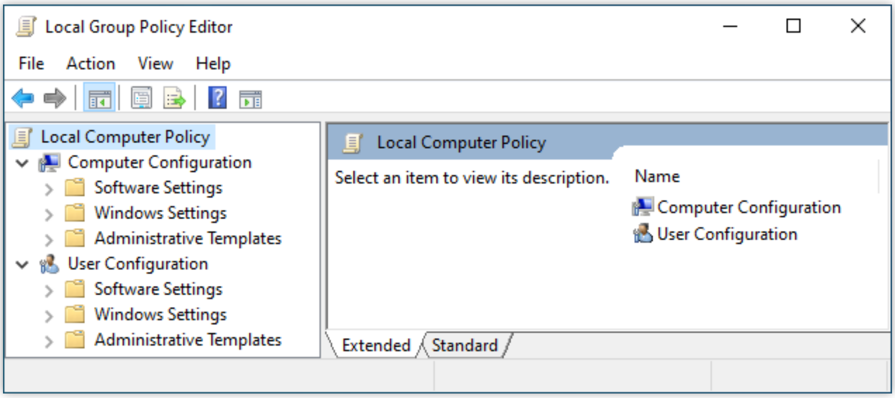
Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: gpedit.msc

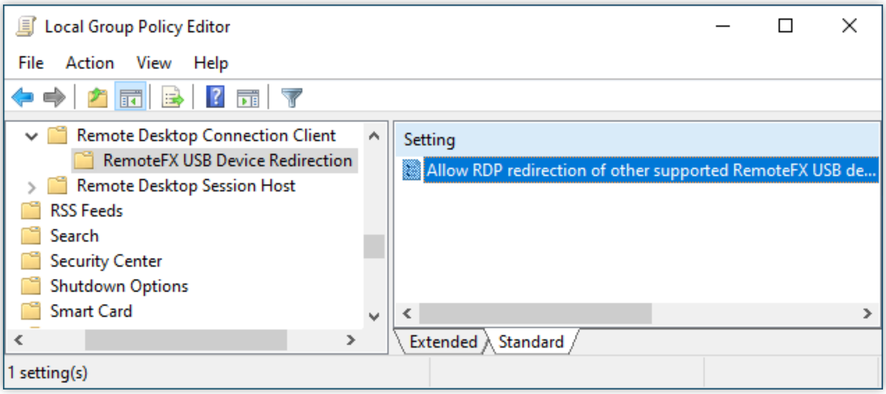
Step 2

In the left pane of the window, click and expand the following items: [**Computer Configuration**] > [**Administrative Templates**] > [**Windows Components**] > [**Remote Desktop Services**] > [**Remote Desktop Connection Client**] > [**RemoteFX USB Device Redirection**]



Step 3

The window will appear as shown in the figure below. Double-click [**Allow RDP redirection of other supported RemoteFX USB devices from this computer**] here.



- Remote Desktop Connection Client
  - RemoteFX USB Device Redirection
- Remote Desktop Session Host
- RSS Feeds
- Search
- Security Center
- Shutdown Options
- Smart Card

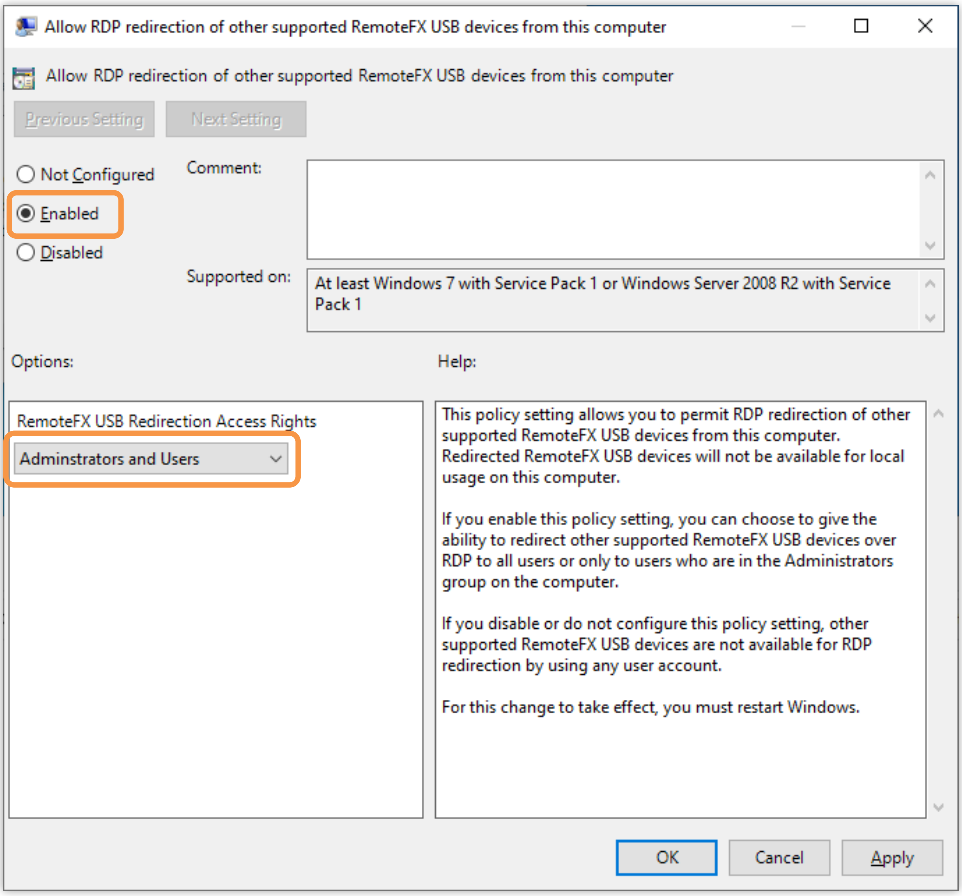
Setting

Allow RDP redirection of other supported RemoteFX USB de...

1 setting(s)

Step 4

The window below will appear. In this window, select the [**Enabled**] radio button, select [**Administrators and Users**] under RemoteFX USB Redirection Access Rights, and click [**OK**].



Allow RDP redirection of other supported RemoteFX USB devices from this computer
Previous Setting
Next Setting
OK
Cancel
Apply

Allow RDP redirection of other supported RemoteFX USB devices from this computer

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows 7 with Service Pack 1 or Windows Server 2008 R2 with Service Pack 1

Options:

RemoteFX USB Redirection Access Rights: Adminstrators and Users

Help:

This policy setting allows you to permit RDP redirection of other supported RemoteFX USB devices from this computer. Redirected RemoteFX USB devices will not be available for local usage on this computer.

If you enable this policy setting, you can choose to give the ability to redirect other supported RemoteFX USB devices over RDP to all users or only to users who are in the Administrators group on the computer.

If you disable or do not configure this policy setting, other supported RemoteFX USB devices are not available for RDP redirection by using any user account.

For this change to take effect, you must restart Windows.

Step 5

Restart Windows.

10.3.3 Redirecting and disconnecting SHALO AUTH

Connect to the remote PC through Remote Desktop, and then enter the full-screen mode. Click the icon you can see circled in the connection bar that appears at the top of the screen, as shown below.

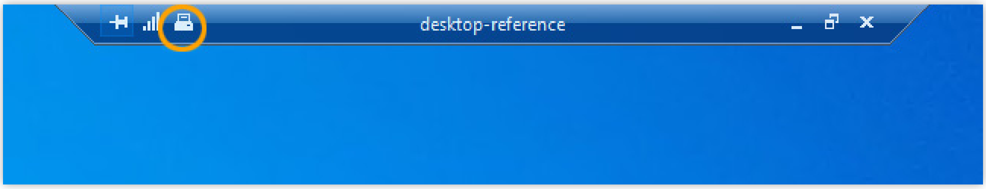
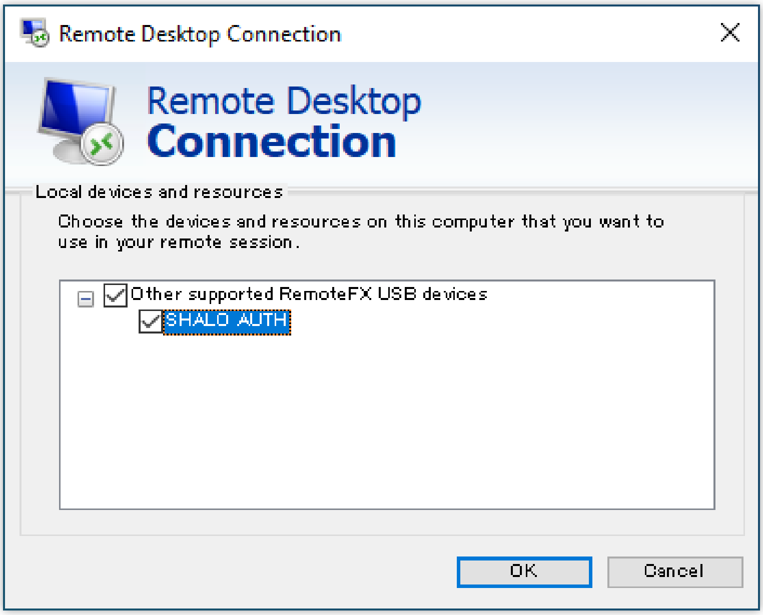


Figure 77 Connection bar for Remote Desktop

If the connection bar is not visible, you can show it by moving the mouse cursor to the top-middle area. The connection bar will always be visible when you click the pin on it.

The window below will appear. To use SHALO AUTH on the remote PC accessed through Remote Desktop, select the [**SHALO AUTH**] check box and click [**OK**]. To disconnect SHALO AUTH from the remote PC accessed through Remote Desktop, clear the check box.


Remote Desktop Connection

Remote Desktop **Connection**

Local devices and resources

Choose the devices and resources on this computer that you want to use in your remote session.

☑ Other supported RemoteFX USB devices

☑ SHALO AUTH

Figure 78 Device redirected to the remote PC

Chapter 11

Frequently asked questions

This chapter contains frequently asked questions and solutions to them related to the use of SHALO AUTH.

Topics in this chapter

1. How can I load an SSH public key without using SHALO Keyring?
2. How can I create a key without using SHALO Keyring?
3. How can I import a key in .pfx, .p12, or DER format?
4. Which versions of OpenSSH have restrictions in terms of using SHALO AUTH?
5. Troubleshooting by symptom

11.1 How can I load an SSH public key without SHALO Keyring?

You can use OpenSSH to load an SSH public key from SHALO AUTH. This can be done in two ways:

- Use the PKCS #11 module.
- Use an authentication agent.

When using the PKCS #11 module

Use the `-D` option in `ssh-keygen` to specify the file path to the PKCS #11 module:

```
ssh-keygen -D pkcs11file
```

The name of the PKCS #11 module depends on the environment. For the file name of the module, see Chapter 3.

If the PKCS #11 module is used by a different application or registered with an authentication agent, the ssh-keygen command fails.

In the following example, all the public keys in SHALO AUTH are printed, and then only the public key for testkey2 is stored in the key.pub file.

```
$ ssh-keygen -D $SLPKCS11FILE↵
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBA
3/YCyF+KOni2K0nLT625u5teJ8hAubFhr+2LYkBGbADxcNQm4fgpHi+U4nqIddJ1OVl+asi5u
I0BZAK6Nq+qI= testkey1
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBF
8QnCuzFzd2lyn3AEmfLbLjnJZLxdlNdw9F3GZyEK9XROEUL/m6FAY1W4WPnDbWVnOtoBj3DEE
zb1774UHuBEg= testkey2

$ ssh-keygen -D $SLPKCS11FILE | grep testkey2 > key.pub↵
```

When using an authentication agent

When SHALO AUTH is registered with an OpenSSH authentication agent using `shalo-add`, the SSH public key managed by the authentication agent can be loaded through the `ssh-add` command with the `-L` option:

```
$ ssh-add -L↵
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBA
3/YCyF+KOni2K0nLT625u5teJ8hAubFhr+2LYkBGbADxcNQm4fgpHi+U4nqIddJ1OVl+asi5u
I0BZAK6Nq+qI= testkey1
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBF
8QnCuzFzd2lyn3AEmfLbLjnJZLxdlNdw9F3GZyEK9XROEUL/m6FAY1W4WPnDbWVnOtoBj3DEE
zb1774UHuBEg= testkey2
```

11.2 How can I create a key without using SHALO Keyring?

If you want to store a private key as a file, you need to create the key without using SHALO Keyring. This section explains how to do it by using the following three software programs:

- OpenSSH
- PuTTY
- OpenSSL

11.2.1 Using OpenSSH

OpenSSH's `ssh-keygen` command enables you to create a pair of SSH private and public keys that comply with RSA or ECDSA public key cryptography.

The following table lists and describes the main options of the `ssh-keygen` command.

Option	Description
`-t` *key-type*	Specify the type of key to create. The possible values are `rsa` or `ecdsa`.
`-b` *bit-length*	In RSA, the bit length represents the key length. In ECDSA, it is the number (256, 384, or 521) that follows letters P-, the elliptical curve name.
`-C` *comment*	Is a comment string for the SSH public key.

The following table shows the commands to create keys in each cryptography.

Key to create	Command	Note
RSA key	`ssh-keygen -t rsa -b` *`Length`*	Specify 2048 to 4096 for *`Length`*.
ECDSA key on P-256	`ssh-keygen -t ecdsa -b 256`	
ECDSA key on P-384	`ssh-keygen -t ecdsa -b 384`	
ECDSA key on P-521	`ssh-keygen -t ecdsa -b 521`	

To create a key, use the following procedure:

1. Open a terminal program (In Windows, CMD, Git Bash, Cygwin, or other programs).
2. Specify options appropriate for the key you create, and run the `ssh-keygen` command.
3. When you see the “Enter file in which to save the key” message, specify the name of the file for the key, and press the Enter key.
4. When you see the “Enter passphrase” message, type the passphrase for encrypting and protecting the key file, and press the Enter key.
5. When you see the “Enter same passphrase again” message, type the passphrase again and press the Enter key.

The passphrase is used to encrypt the private key file. Make sure to remember the passphrase because you need to enter it again when importing the private key into SHALO AUTH.

The next example shows how to create a 4,096-bit key pair in RSA. It uses “test_comment” as a comment.

```
$ ssh-keygen -t rsa -b 4096 -C test_comment↵
Generating public/private rsa key pair.
Enter file in which to save the key (/home/foo/.ssh/id_rsa): shalo↵
Enter passphrase (empty for no passphrase): ↵
Enter same passphrase again: ↵
Your identification has been saved in shalo.
Your public key has been saved in shalo.pub.
The key fingerprint is:
SHA256:ss3DI0VU54cWl4Hp8eOw41r0zeOsyQrCT3vrOdWvhz4 test_comment
The key's randomart image is:
+---[RSA 4096]----+
|        ... oooo |
|       .   o++.  |
|        .  .+o.  |
|       .   .o.o  |
|      . S   .+.. |
|      .B   .oo.+ |
|      oo*o .o..o+|
|       .+oo++ E +|
|         o+*+=+* |
+----[SHA256]-----+
```

The private key is stored in the file with the name you entered (“shalo” in the above example). The public key is stored in the file with the name that has the string you typed, followed by “.pub” (shalo.pub in the above example).

11.2.2 Using PuTTY

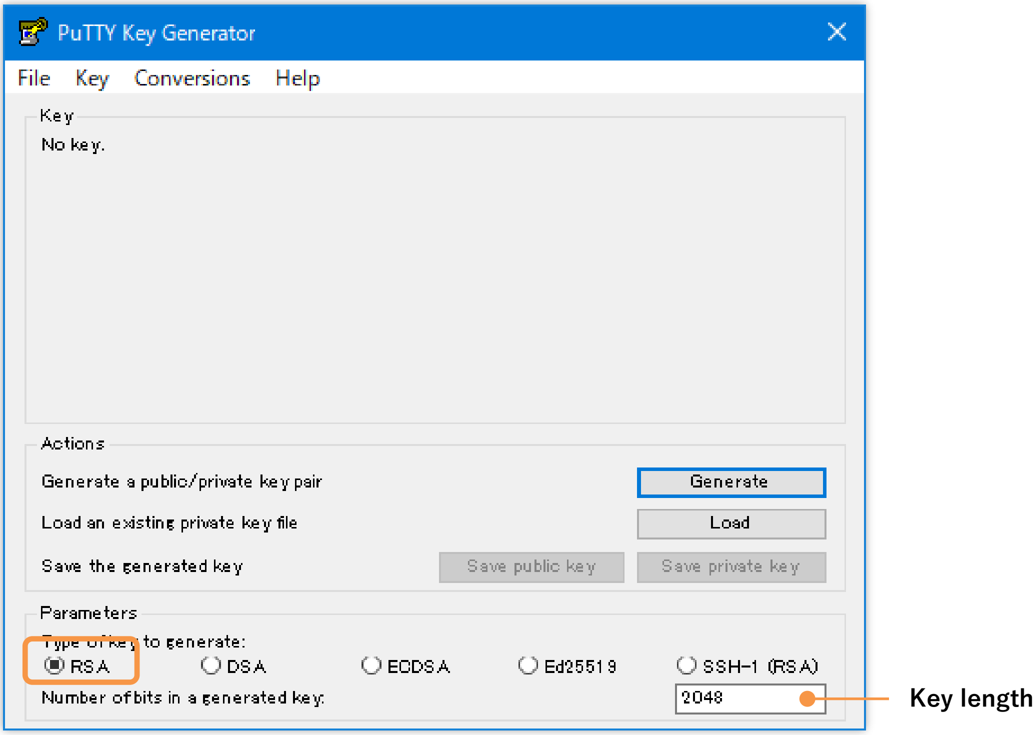
`puttygen`, which comes with PuTTY, enables you to create SSH keys using GUI operations. To create a key, use the following procedure:

1. Start `puttygen`.
2. Specify the key you want to generate.
3. Click [**Generate**].
4. Move the mouse cursor within the [PuTTY Key Generator] window until the progress bar reaches the right-hand end.
5. In [**Key comment**], type a comment, and in each of [**Key passphrase**] and [**Confirm passphrase**], input the passphrase.
6. Click [**Save private key**] to save the private key as a file.
7. Click [**Save public key**] to save the public key as a file.

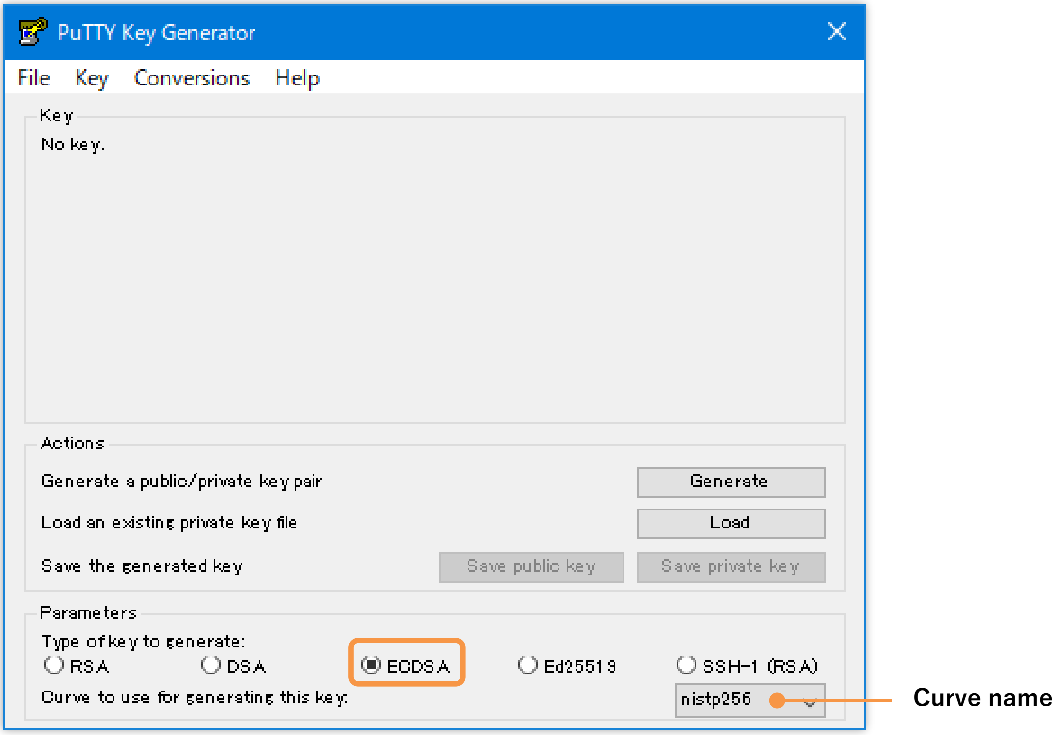
The following explains the procedure together with screenshots.

Steps 1 to 2

When you start `puttygen`, the window below will appear. If you want to generate an RSA key, select the [**RSA**] radio button and enter the key length in the entry field at the bottom of the window. If you want to generate an ECDSA key, select the [**ECDSA**] radio button.


PuTTY Key Generator
File Key Conversions Help
Key
No key.
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)
Number of bits in a generated key:
2048
Key length

When you select the [**ECDSA**] radio button, you can select a curve name as shown in the figure below. Of the curve names, [**nistp256**] means P-256, [**nistp384**] means P-384, and [**nistp521**] means P-521.



No key.

Type of key to generate:
○ RSA ○ DSA ● ECDSA ○ Ed25519 ○ SSH-1 (RSA)

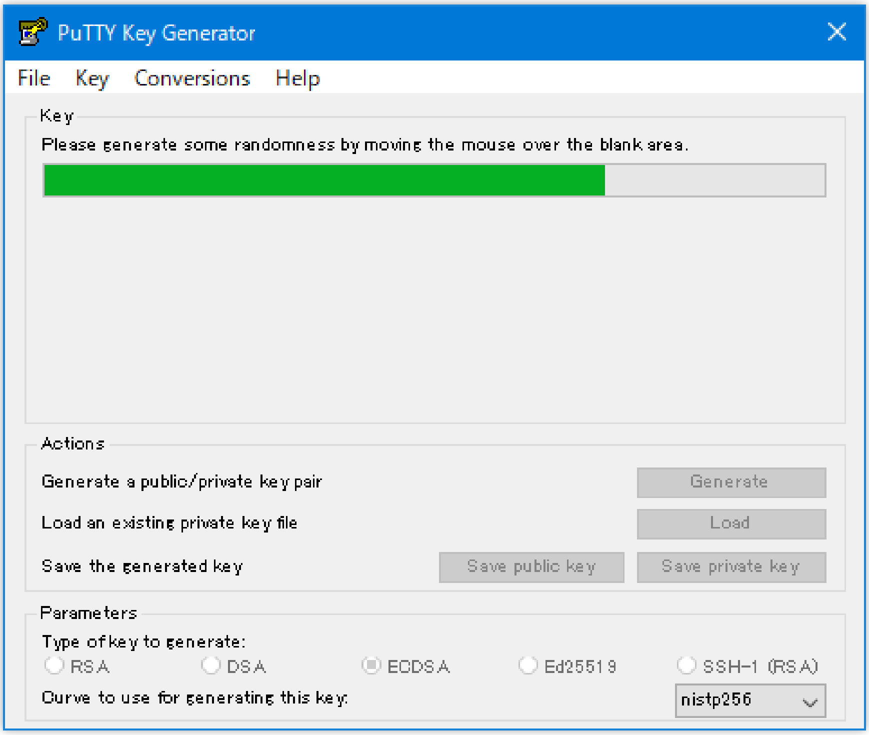
Curve to use for generating this key: nistp256

Step 3

Click [**Generate**].

Step 4

Move the mouse cursor within the window until the progress bar reaches the right-hand end, as shown below.



Please generate some randomness by moving the mouse over the blank area.

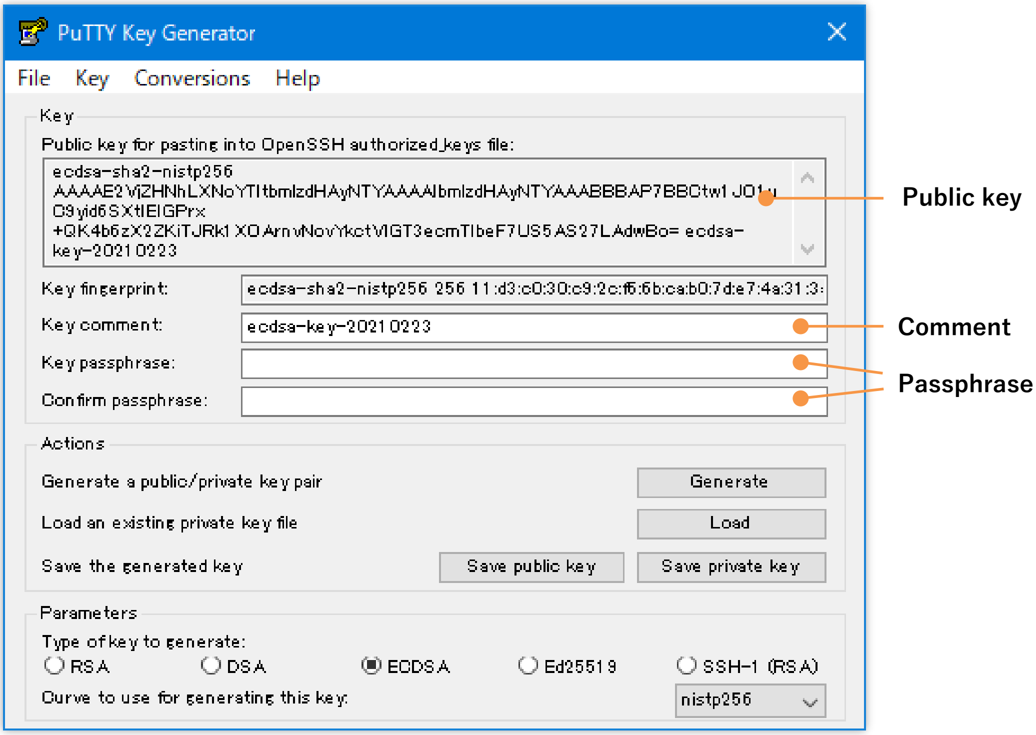
Type of key to generate:
○ RSA ○ DSA ● ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Curve to use for generating this key: nistp256

Step 5

When the keys are generated, they are displayed in the window, as shown below. In [**Key comment**], type a comment, and in each of [**Key passphrase**] and [**Confirm passphrase**], input the passphrase.

The passphrase is used to encrypt the private key file. Make sure to remember the passphrase because you need to enter it again when importing the private key into SHALO AUTH.



Public key for pasting into OpenSSH authorized_keys file:

ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAP7BBCtw1JO1u
C9yid6SXtIEIGPrx
+QK4b6zX2ZKiTJRk1XOArnvNovYkctVIGT3ecmTIbeF7US5AS27LAdwBo= ecdsa-
key-20210223

Key fingerprint: ecdsa-sha2-nistp256 256 11:d3:c0:30:c9:2c:f6:6b:ca:b0:7d:e7:4a:31:3:

Key comment: ecdsa-key-20210223

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

Type of key to generate:
○ RSA ○ DSA ● ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Curve to use for generating this key: nistp256

Step 6

Click [**Save private key**] to save the private key as a file.

Step 7

Click [**Save public key**] to save the public key as a file.

The public key is also displayed in the edit field at the top of the window. You can copy it for use.

11.2.3 Using OpenSSL

OpenSSL enables you to create a variety of key pairs of private and public keys in addition to the SSH key pairs. The keys are output in PEM format.

In RSA

Create an RSA key by using the `openssl genrsa` command. The format is as follows:

```
openssl genrsa -out output-file key-bit-length
```

The following shows an example of creating a key with a key length of 4,096 bits and storing it in the rsakey.pem file:

```
$ openssl genrsa -out rsakey.pem 4096↵
Generating RSA private key, 4096 bit long modulus (2 primes)
.........................................................................
...................................++++
................................................++++
e is 65537 (0x010001)
```

In ECDSA

Create an ECDSA key by using the `openssl ecparam` command. The format is as follows:

```
openssl ecparam -genkey -name curve-name -out output-file
```

You can see the names of curves supported by OpenSSL by using the command below. Some environments support only some of the curves.

```
openssl ecparam -list_curves
```

The following curves supported by SHALO AUTH have different names in OpenSSL:

● secp192r1 (P-192)	`prime192v1`
● secp256r1 (P-256)	`prime256v1`

The following shows an example of creating a key on secp256r1 (P-256) and storing it in the eckey.pem file:

```
$ openssl ecparam -genkey -name prime256v1 -out eckey.pem↵
```

11.3 How can I import a key in .pfx, .p12, or DER format?

SHALO Keyring does not support keys in PKCS #12 format (with the extension of pfx or p12) or in DER format. By converting them into PEM format with OpenSSL, you can import them through SHALO Keyring.

This section explains how to convert keys into PEM format using OpenSSL.

pfx or p12 format

To store a private key in PEM format, use the command below. If you do not encrypt the PEM file, specify the additional `-nodes` option.

```
openssl pkcs12 -in input-file -nocerts -out output-file
```

To save the private key in the server.pfx file as key.pem, do the following:

```
$ openssl pkcs12 -in server.pfx -nocerts -out key.pem↵
Enter Import Password:  Type the password for the input file↵
Enter PEM pass phrase:  Type the password for the output file↵
Verifying - Enter PEM pass phrase:  Confirm the password for the output
file↵
```

RSA private key in DER format

To convert an RSA private key from DER format into PEM format, use the following command:

```
openssl rsa -inform DER -in input-file -outform PEM -out output-file
```

ECDSA private key in DER format

To convert an ECDSA private key from DER format into PEM format, use the following command:

```
openssl ecparam -inform DER -in input-file -outform PEM -out output-file
```

11.4 Which versions of OpenSSH have limitations on the use of SHALO AUTH?

You can check the configuration of OpenSSH by using the following command:

```
$ ssh -V↵
OpenSSH_8.5p1, OpenSSL 1.1.1k  25 Mar 2021
```

This example shows that the version of OpenSSH is 8.5p1 and the cryptography library being used is OpenSSL 1.1.1k.

The following table lists the OpenSSH configurations in which the RSA and ECDSA keys in SHALO AUTH are available.

OpenSSH configuration	RSA key	ECDSA key
OpenSSH 5.2p1 or earlier	N/A	N/A
OpenSSH 5.3p1–OpenSSH 7.9p1	✓	N/A
OpenSSH 8.0p1 or later + OpenSSH 1.0	✓	N/A
OpenSSH 8.0p1 or later + OpenSSH 1.1	✓	✓
OpenSSH 8.0p1 or later + LibreSSL 2.9 or earlier	✓	N/A
OpenSSH 8.0p1 or later + LibreSSL 3.0 or later	✓	✓

OpenSSH supports ECDSA keys only on P-256, P-384, and P-521.

OpenSSH version for each environment

The table below lists the standard versions of the OpenSSH package in each environment. The ECDSA keys of SHALO AUTH are unavailable in the environments whose background is orange.

Environment	Configuration (result output by ssh -V)
Git for Windows 2.21.0 or earlier	Combination of OpenSSH_7.9p1 and OpenSSL 1.1.1a or earlier
Git for Windows 2.22.0 or later	Combination of OpenSSH_8.0p1 and OpenSSL 1.1.1c or later
Git for Windows 2.31.1	OpenSSH_8.5p1, OpenSSL 1.1.1k 25 Mar 2021
Cygwin 3.2.0	OpenSSH_8.5p1, OpenSSL 1.1.1f 31 Mar 2020
macOS BigSur	OpenSSH_8.1p1, LibreSSL 2.7.3
Ubuntu 18.04.5 LTS	OpenSSH_7.6p1 Ubuntu-4ubuntu0.3, OpenSSL 1.0.2n 7 Dec 2017
Ubuntu 20.04.2 LTS	OpenSSH_8.2p1 Ubuntu-4ubuntu0.2, OpenSSL 1.1.1f 31 Mar 2020
CentOS 7.9-2009	OpenSSH_7.4p1, OpenSSL 1.0.2k-fips 26 Jan 2017
CentOS 8.3.2011	OpenSSH_8.0p1, OpenSSL 1.1.1g FIPS 21 Apr 2020
Fedora 33-1.2	OpenSSH_8.4p1, OpenSSL 1.1.1g FIPS 21 Apr 2020
Fedora 34-1.2	OpenSSH_8.5p1, OpenSSL 1.1.1k FIPS 25 Mar 2021

11.5 Troubleshooting by symptom

11.5.1 User PIN is locked

A user PIN is locked when five consecutive entry attempts fail. When this happens, use SHALO Smith to reset the user PIN (Section 5.4).

11.5.2 SO PIN is locked

A SO PIN is locked when five consecutive entry attempts fail. There is no way to restore the SO PIN only.

The keys in SHALO AUTH are not removed even if the SO PIN is locked.
You can still use the keys unless the user PIN is locked.

To restore the SO PIN, restore SHALO AUTH to the factory settings (Section 5.3) and set it up again.

When you restore SHALO AUTH to the factory settings, all the data held by SHALO AUTH is removed and the settings as the U2F security key previously registered are also disabled.

11.5.3 LED keeps flashing when SHALO AUTH is connected to PC

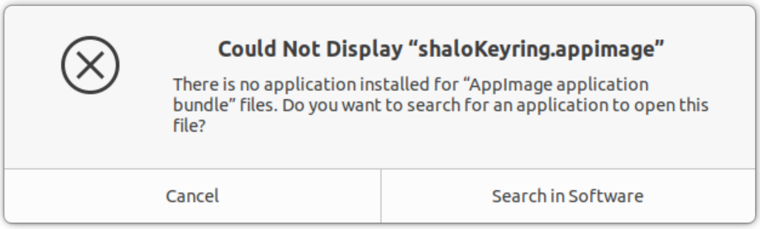
When SHALO AUTH detects an unrecoverable error, its LED keeps flashing one to three times per second. Such a SHALO AUTH device is no longer available. Use the following procedure to dispose of the authentication information:

- In Web services that use SHALO AUTH as the U2F security key, deregister SHALO AUTH.
- If the public key in SHALO AUTH is registered with the sever, remove it from the server.

11.5.4 shaloKeyring.appimage/shaloSmith.appimage does not start in Linux (1)

Symptom

When you tried to start shaloKeyring.appimage or shaloSmith.appimage from the Linux GUI, the following window appeared.



Could Not Display "shaloKeyring.appimage"

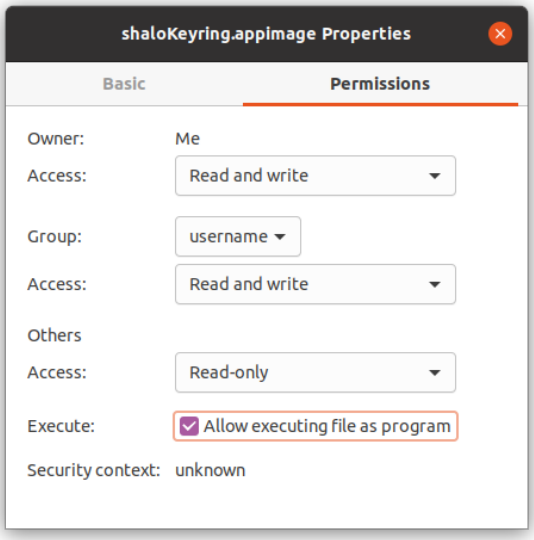
There is no application installed for "AppImage application bundle" files. Do you want to search for an application to open this file?

Cause

There is no permission to execute the .appimage file.

Solution

Grant the execute permission to the .appimage file. Right-click the .appimage file in question. From the context menu, select [**Properties**], and in the following window, select the [**Allow executing file as program**] check box.



shaloKeyring.appimage Properties

Owner: Me
Access: Read and write
Group: username
Access: Read and write
Others
Access: Read-only
Execute: ☑ Allow executing file as program
Security context: unknown

11.5.5 shaloKeyring.appimage/shaloSmith.appimage does not start in Linux (2)

Symptom

When you tried to start shaloKeyring.appimage or shaloSmith.appimage, the window did not appear.

Cause

The environment does not meet the prerequisites.

Solution

shaloKeyring.appimage and shaloSmith.appimage launched from the terminal output error messages to the terminal when errors occur. The error messages can be used to determine the cause of the problem.

An example would be as follows:

```
$ ./shaloKeyring.appimage↵
dlopen(): error loading libfuse.so.2
AppImages require FUSE to run.
You might still be able to extract the contents of this AppImage
if you run it with the --appimage-extract option.
See https://github.com/AppImage/AppImageKit/wiki/FUSE
for more information
```

In the above example, the application requires libfuse2, which is solved by installing libfuse2, see Section 3.5.2.

11.5.6 SHALO Keyring/Smith does not recognize SHALO AUTH

Symptom

Even though SHALO AUTH is connected to the PC, SHALO Keyring/Smith cannot find the device.

Cause

SHALO Keyring/Smith tried to access SHALO AUTH, but other software was using it.

Solution

Make sure that multiple software programs are not using one SHALO AUTH device at one time. The following software uses SHALO AUTH:

1. Authentication agent
2. SHALO Keyring/Smith
3. Adobe® Acrobat®/Adobe® Acrobat® Reader® (when it accesses SHALO AUTH)

Make sure that both SHALO Keyring and SHALO Smith are not running at the same time.

11.5.7 `ssh -I` command fails with “C_GetTokenInfo ～ failed: ??”

Symptom

You tried to connect to an SSH server with the PKCS#11 module specified in `ssh -I`, but the command failed as shown below:

```
$ ssh -I pkcs11file username@hostname↵
C_GetTokenInfo for provider pkcs11file slot 0 failed: 48
username@hostname: Permission denied (publickey).
```

Cause

The PKCS#11 module tried to access SHALO AUTH, but other software was using it.

Solution

Make sure that multiple software programs are not using one SHALO AUTH device at one time. The following software uses SHALO AUTH:

4. Authentication agent
5. SHALO Keyring/Smith
6. Adobe® Acrobat®/Adobe® Acrobat® Reader® (when it accesses SHALO AUTH)

11.5.8 `ssh -I` command results in “C_GetAttributeValue failed: 18” message

Symptom

You tried to connect to an SSH server with the PKCS#11 module specified in `ssh -I`, but the following message was output:

```
$ ssh -I pkcs11file username@hostname↵
C_GetAttributeValue failed: 18
username@hostname: Permission denied (publickey).
```

Cause

The “C_GetAttributeValue failed: 18” message is output when OpenSSH detects a key that is not supported. It may be output even if a connection to an SSH server is successful.

Solution

If a connection to an SSH server fails with this message output, use a key supported by OpenSSH in your environment. For the key types supported by OpenSSH, see Section 11.4.

11.5.9 Unable to log in to an SSH server through ssh-agent

Symptom

You registered SHALO AUTH with ssh-agent by using `shalo-add`, but attempts to connect to the SSH server failed as shown below:

```
$ ssh username@hostname↵
username@hostname: Permission denied (publickey).
```

Cause

The public key registered with the SSH server is not loaded into ssh-agent. You can view the public keys loaded into ssh-agent by using `ssh-add -L`.

```
$ ssh-add -L↵
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBA
3/YCyF+KOni2K0nLT625u5teJ8hAubFhr+2LYkBGbADxcNQm4fgpHi+U4nqIddJ1OVl+asi5u
I0BZAK6Nq+qI= testkey1
```

Solution

If the target key in SHALO AUTH is not loaded into ssh-agent, check the key type with SHALO Keyring to see if OpenSSH supports it. For the key types OpenSSH supports, see Section 11.4.

When the key is loaded correctly into ssh-agent, see if the public key is registered with the SSH server.

11.5.10 Unable to register SHALO AUTH with ssh-agent

Symptom

When running `shalo-add` and entering the user PIN, you received the message:

```
Could not add card "path-to-PKCS#11-library": agent refused operation
```

Cause

Possible causes are:

- The PKCS #11 module has already been registered.
- A different application is using the PKCS #11 module.
- The user PIN is not correct.
- The user PIN is locked.
- SHALO AUTH does not have the key, or the key stored in SHALO AUTH is not supported.
- The path to the PKCS #11 module is not permitted by ssh-agent.

Solution

To find the cause, use the following procedure:

1. Run `shalo-remove` and then run `shalo-add` again.
2. Exit the applications that are using the PKCS #11 module. If the PKCS #11 module is registered with Acrobat®, exit Acrobat®.
3. Using SHALO Keyring or SHALO Smith, check the state of SHALO AUTH (Section 4.2 and Section 5.1).
4. Using the `ssh -I` option, connect to the server and view the error output (Section 10.1).

If you still cannot find the cause, follow Chapter 3 to check if the PKCS #11 module has been installed in the correct directory.

In OpenSSH 7.9p1 or later, you can output the operating states of ssh-agent to the console to see the error information. Two terminals are required to do this.

At one terminal, start ssh-agent in debug mode as shown below. The operations are output to this terminal when ssh-add is executed.

```
$ ssh-agent -d > ~/agenttmp↵
```

At the other terminal, register SHALO AUTH as shown below:

```
$ source ~/agenttmp > /dev/null↵
$ ssh-add -v -s $SLPKCS11FILE↵
```

If the PKCS #11 module is not on the whitelist, you will receive the following:

```
refusing PKCS#11 add of "file-path-to-PKCS-#11-module": provider not white
listed
```

If SHALO AUTH cannot be found, “returned no slots” is displayed at the end. Most of the time, this occurs because other software is using SHALO AUTH.

```
debug1: provider /usr/lib/libslpkcs11.so: manufacturerID <AXELL CORPORATIO
N> cryptokiVersion 2.40 libraryDescription <AXELL PKCS#11 library> library
Version 1.3
debug1: pkcs11_register_provider: provider /usr/lib/libslpkcs11.so returne
d no slots
```

If no keys are found in SHALO AUTH, “returned no keys” is displayed at the end.

```
debug1: provider /usr/lib/libslpkcs11.so: manufacturerID <AXELL CORPORATIO
N> cryptokiVersion 2.40 libraryDescription <AXELL PKCS#11 library> library
Version 1.3
debug1: provider /usr/lib/libslpkcs11.so slot 0: label <device-label> manu
facturerID <AXELL CORPORATION> model <SHALO AUTH> serial <> flags 0x40d
debug1: pkcs11_provider_finalize: 0x55fba2e1ddc0 refcount 1 valid 1
debug1: pkcs11_provider_unref: 0x55fba2e1ddc0 refcount 1
debug1: pkcs11_add_provider: provider /usr/lib/libslpkcs11.so returned no
keys
```

If the error is related to the user PIN, “C_Login failed” is displayed along the way. This is probably due to an incorrect or locked user PIN.

```
debug1: provider /usr/lib/libslpkcs11.so: manufacturerID <AXELL CORPORATIO
N> cryptokiVersion 2.40 libraryDescription <AXELL PKCS#11 library> library
Version 1.3
debug1: provider /usr/lib/libslpkcs11.so slot 0: label <device-label> manu
facturerID <AXELL CORPORATION> model <SHALO AUTH> serial <> flags 0x5040d
 C_Login failed: 164
debug1: pkcs11_provider_finalize: 0x55fba2e0fc10 refcount 1 valid 1
debug1: pkcs11_provider_unref: 0x55fba2e0fc10 refcount 1
debug1: pkcs11_add_provider: provider /usr/lib/libslpkcs11.so returned no
keys
```

Chapter 12

PKCS #11 module information

This chapter provides various specifications of the PKCS #11 module for SHALO AUTH.

Topics in this chapter

1. Supported API functions
2. Supported key types
3. Supported mechanisms
4. Supported attributes

12.1 Supported API functions

Key generation, key wrapping, and object copy features are not supported. The following table lists the supported and unsupported API functions.

Supported API functions		Unsupported API functions
C_Initialize	C_FindObjectsFinal	C_GetOperationState
C_Finalize	C_EncryptInit	C_SetOperationState
C_GetInfo	C_Encrypt	C_CopyObject
C_GetFunctionList	C_EncryptUpdate	C_GetObjectSize
C_GetSlotList	C_EncryptFinal	C_DigestKey
C_GetSlotInfo	C_DecryptInit	C_SignRecoverInit
C_GetTokenInfo	C_Decrypt	C_SignRecover
C_GetMechanismList	C_DecryptUpdate	C_VerifyRecoverInit
C_GetMechanismInfo	C_DecryptFinal	C_VerifyRecover
C_InitToken	C_DigestInit	C_DigestEncryptUpdate
C_InitPIN	C_Digest	C_DecryptDigestUpdate
C_SetPIN	C_DigestUpdate	C_SignEncryptUpdate
C_OpenSession	C_DigestFinal	C_DecryptVerifyUpdate
C_CloseSession	C_SignInit	C_GenerateKey
C_CloseAllSessions	C_Sign	C_GenerateKeyPair
C_GetSessionInfo	C_SignUpdate	C_WrapKey
C_Login	C_SignFinal	C_UnwrapKey
C_Logout	C_VerifyInit	C_DeriveKey
C_CreateObject	C_Verify	C_GetFunctionStatus
C_DestroyObject	C_VerifyUpdate	C_CancelFunction
C_GetAttributeValue	C_VerifyFinal	C_WaitForSlotEvent
C_SetAttributeValue	C_SeedRandom	
C_FindObjectsInit	C_GenerateRandom	
C_FindObjects		

12.2 Supported key types

Key type	Algorithm	What is supported
CKK_RSA	RSA	RSA key of 1,024 to 4,096 bits
CKK_EC	ECDSA	Following elliptical curves: secp192r1 (P-192) secp192k1 secp224r1 (P-224) secp224k1 secp256r1 (P-256) secp256k1 secp384r1 (P-384) secp521r1 (P-521)

12.3 Supported mechanisms

Digesting mechanisms

Mechanism	Note
CKM_SHA_1	Supports both single- and multiple-part operations.
CKM_SHA256	Supports both single- and multiple-part operations.
CKM_SHA384	Supports both single- and multiple-part operations.
CKM_SHA512	Supports both single- and multiple-part operations.

RSA mechanisms

Mechanism	Op	MinKey	MaxKey	Encrypt	Decrypt	Sign	Verify
CKM_RSA_X_509	Single	1024	4096	✓	✓	✓	✓
CKM_RSA_PKCS	Single	1024	4096	✓	✓	✓	✓
CKM_SHA1_RSA_PKCS	Both	1024	4096			✓	✓
CKM_SHA256_RSA_PKCS	Both	1024	4096			✓	✓
CKM_SHA384_RSA_PKCS	Both	1024	4096			✓	✓
CKM_SHA512_RSA_PKCS	Both	1024	4096			✓	✓

Op: Single supports the single-part operations only.

Op: Both supports both single- and multiple-part operations.

EC mechanisms

Mechanism	Op	MinKey	MaxKey	Encrypt	Decrypt	Sign	Verify
CKM_ECDSA	Both	192	521			✓	✓
CKM_ECDSA_SHA1	Both	192	521			✓	✓
CKM_ECDSA_SHA256	Both	192	521			✓	✓
CKM_ECDSA_SHA384	Both	192	521			✓	✓
CKM_ECDSA_SHA12	Both	192	521			✓	✓

Op: Both supports both single- and multiple-part operations.

12.4 Supported attributes

Attributes held by all objects

Attribute	Default value	Note
CKA_TOKEN	False	Supported by hardware functionality.
CKA_PRIVATE	False	Supported by hardware functionality.
CKA_MODIFIABLE	True	Supported by hardware functionality.
CKA_COPYABLE	True	C_CopyObject() is not supported.
CKA_DESTROYABLE	True	Supported by hardware functionality.

Additional attributes supported by RSA private key objects

Attribute	Required	Note
CKA_CLASS	✓	CKO_PRIVATE_KEY at all times
CKA_KEY_TYPE	✓	CKK_RSA at all times
CKA_LABEL		
CKA_ID		
CKA_SUBJECT		
CKA_MODULUS	✓	
CKA_PUBLIC_EXPONENT	✓	
CKA_PRIVATE_EXPONENT	✓	Protected by CKA_SENSITIVE.
CKA_PRIME_1	✓	Protected by CKA_SENSITIVE.
CKA_PRIME_2	✓	Protected by CKA_SENSITIVE.
CKA_EXPONENT_1	✓	Protected by CKA_SENSITIVE.
CKA_EXPONENT_2	✓	Protected by CKA_SENSITIVE.
CKA_COEFFICIENT	✓	Protected by CKA_SENSITIVE.
CKA_SENSITIVE		
CKA_DECRYPT		
CKA_SIGN		

Additional attributes supported by RSA public key objects

Attribute	Required	Note
CKA_CLASS	✓	CKO_PUBLIC_KEY at all times
CKA_KEY_TYPE	✓	CKK_RSA at all times
CKA_LABEL		
CKA_ID		
CKA_SUBJECT		
CKA_MODULUS	✓	
CKA_PUBLIC_EXPONENT	✓	
CKA_ENCRYPT		
CKA_VERIFY		

Additional attributes supported by EC private key objects

Attribute	Required	Note
CKA_CLASS	✓	CKO_PRIVATE_KEY at all times
CKA_KEY_TYPE	✓	CKK_EC at all times
CKA_LABEL		
CKA_ID		
CKA_SUBJECT		
CKA_EC_PARAMS	✓	
CKA_VALUE	✓	Protected by CKA_SENSITIVE.
CKA_SENSITIVE		
CKA_SIGN		

Additional attributes supported by EC public key objects

Attribute	Required	Note
CKA_CLASS	✓	CKO_PRIVATE_KEY at all times
CKA_KEY_TYPE	✓	CKK_EC at all times
CKA_LABEL		
CKA_ID		
CKA_SUBJECT		
CKA_EC_PARAMS	✓	
CKA_EC_POINT	✓	
CKA_VERIFY		

Additional attributes supported by public key objects

Attribute	Required	Note
CKA_CLASS	✓	CKO_CERTIFICATE at all times
CKA_CERTIFICATE_TYPE	✓	CKC_X_509 at all times
CKA_LABEL		
CKA_ID		
CKA_SUBJECT		
CKA_VALUE	✓	
CKA_ISSUER		
CKA_SERIAL_NUMBER		

Maximum data length of attributes of variable-length data types

If an attribute is of a data type with a variable length, such as a Byte array or string, and its length cannot be determined, there is no limitation to the data length for the attribute that is within 8 Kbytes as a single object.

Cautions

- The contents of this document (i.e., this “Cautions”) and specification (i.e., this “Specification”) are current as of November 2022.
- Transfer or duplication of part or all of this Specification without permission from AXELL is prohibited.
- When using the product described in this Specification (i.e., “the Product”), please use it correctly according to the contents of this Specification.
- ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR THE PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, AXELL (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF THE PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.
- AXELL will not be held liable for any opportunity losses, indirect damages due to irregularities or defects of the Product, or semiconductor device damages when using the Product at a value exceeding the absolute maximum rating.
- The application examples and constants shown in this Specification are provided to explain the standard operation and usage of the Product. During mass-production design, take external conditions into consideration. The incorporation of the Product in the design of a customers’ equipment shall be done under the full responsibility of the customers.
- The product application examples, information, and data in this Specification are provided only as examples and do not imply any guarantee related to intellectual property rights such as patent rights or copyrights of third parties or any other rights. Therefore, AXELL assumes no responsibility with regard to (1) infringement of intellectual property rights of third parties, or (2) any consequences arising from the use of the Product.
- While AXELL always strives to improve the quality of the Product, the possibility of defects cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in the Product, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, over current prevention and anti-failure features.
- While AXELL endeavors to make the information contained in this Specification complete, this Specification does not cover every possible phenomenon. If there is any risk of malfunctions due to compliance with this Specification, customers must take appropriate measures, such as consulting AXELL in advance. Customers should note that AXELL will not necessarily be held liable for reparation or compensation even if the contents of this Cautions and/or Specification were complied with by the customer.
- The Product is intended for use in general electronic equipment such as audio visual systems, office automation equipment, communication equipment, home electric appliances and amusement equipment. The product is not intended for use in equipment and/or systems (a) which may directly affect human lives (including, without limitation, life support systems) and (b) which require extremely high reliability and whose failure or malfunction may cause enormous damage (including, without limitation, transportation equipment control unit, nuclear control systems, military equipment), and Customers shall not use the Product for those equipment and/or systems. Customers shall consult AXELL sales representative in advance if they should consider use of the Product in the said equipment and/or systems.
- The company names and the product names used in this document are generally trademarks or registered trademarks of respective companies.

AXELL CORPORATION

Axell Corporation
Akihabara UDX SouthWing 10F 4-14-1
Sotokanda, Chiyoda-ku, Tokyo
TEL +81-3-5298-1670 FAX +81-3-5298-1671
https://www.axell.co.jp/