STRONGKEY

FIDO Demo Client
User Guide

Version 3.0

Copyrights and Notices

Contents

1—Introduction

StrongKey CryptoCabinet (SKCC) is an open-source web application allowing end users to encrypt files within a corporate environment and share those files securely with others while storing all encryption keys securely within a secure vault on-premises.

SKCC was originally created to demonstrate how to write web applications using StrongKey's open-source *StrongKey CryptoEngine (SKCE)* software. SKCE is the underlying engine that encrypts files of any type and any size, optionally storing them in public clouds—such as *Amazon Web Services' Simple Secure Storage (AWS S3)*, Microsoft's *Azure*, and *Eucalyptus Walrus*.

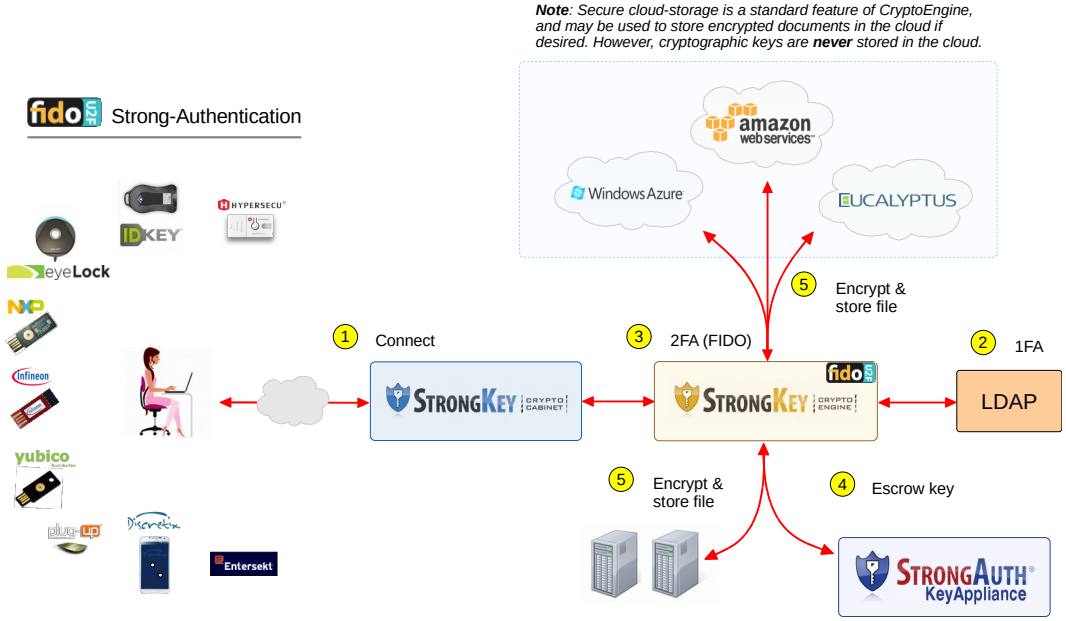
SKCE also allows documents to be digitally signed for establishing authenticity while simultaneously verifying their integrity.

More recently, StrongKey built into SKCE a *Fast Identity Online (FIDO) Universal 2nd Factor (U2F)* server to support the burgeoning protocol for strong authentication. SKCE is now an officially FIDO Certified™ U2F server.

The SKCC web application was FIDO-enabled to take advantage of the FIDO server built into SKCE, to demonstrate how to use the FIDO strong authentication capability in SKCE to protect end-user credentials within web applications.

This document walks through a demonstration of how to use your FIDO U2F Authenticator (a.k.a. Token) with SKCC on a demo site established by StrongKey.

The high-level architecture of the infrastructure is represented in following diagram:



1.1—Prerequisites

In order to successfully work with this demonstration, you must have the following:

- A FIDO Certified™ or FIDO Ready™ U2F Authenticator
  - HyperSecu

  - Neowave

  - Yubico

> **NOTE:** While this document mentions three types of Authenticators in the text—HyperSecu, Neowave and Yubico U2F Authenticators—SKCC has been tested successfully with nearly a dozen different Authenticators—Discretix, eGis Technologies, Entersekt, EyeLock, Infineon, NXP, Plug-Up, Sonavation, ST Microelectronics, etc. As such, SKCC will work with any FIDO Ready™ or FIDO Certified™ U2F Authenticator available on the market.

- A release of the Google Chrome browser that supports the U2F protocol; this document describes the use of SKCC with version 43 or above
- A computer with Microsoft Windows, Apple OS-X, or CentOS Linux installed

1.1.1—Linux Users

Linux PCs require the following task to be completed before beginning the demo:

1. As the *root* user, or using *sudo*, **modify** the */etc/udev/rules.d/70-u2f.rules* file. If it doesn't already exist, create it.
2. **Add the following text** to the file:

```
ACTION!="add|change", GOTO="u2f_end"
KERNEL=="hidraw*", SUBSYSTEM=="hidraw", ATTRS{idVendor}=="*",
ATTRS{idProduct}=="*", TAG+="uaccess"
LABEL="u2f_end"
```

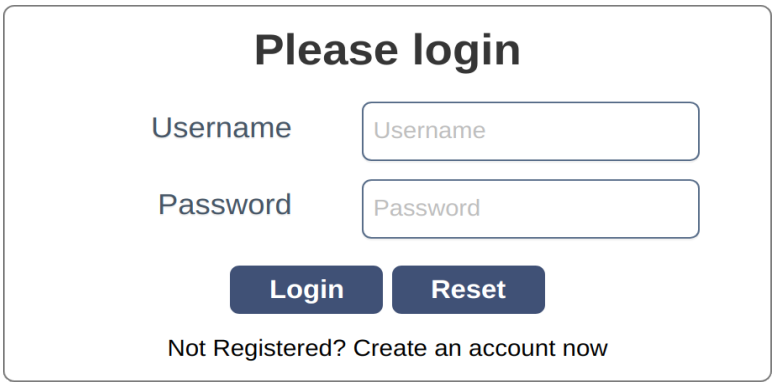
3. **Reboot** the Linux PC.

2—FIDO Registration

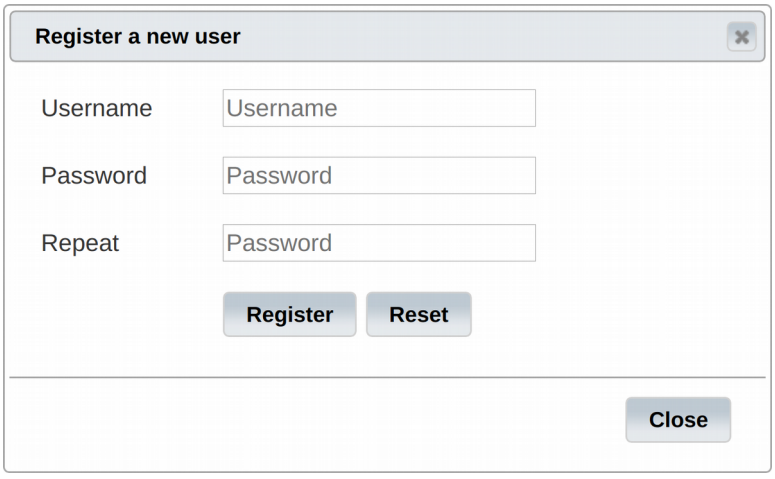
2.1—Register a New User Account

This section explains how to create a new account on the public site where SKCC is hosted. After creating the account, you will be asked to login with the newly created credential into SKCC *without* a FIDO Authenticator.

1. Using Chrome, connect to https://fidodemo.strongauth.com/skcc.
2. From the SKCC application home page, click Not Registered? Create an account now.



3. A registration panel opens.

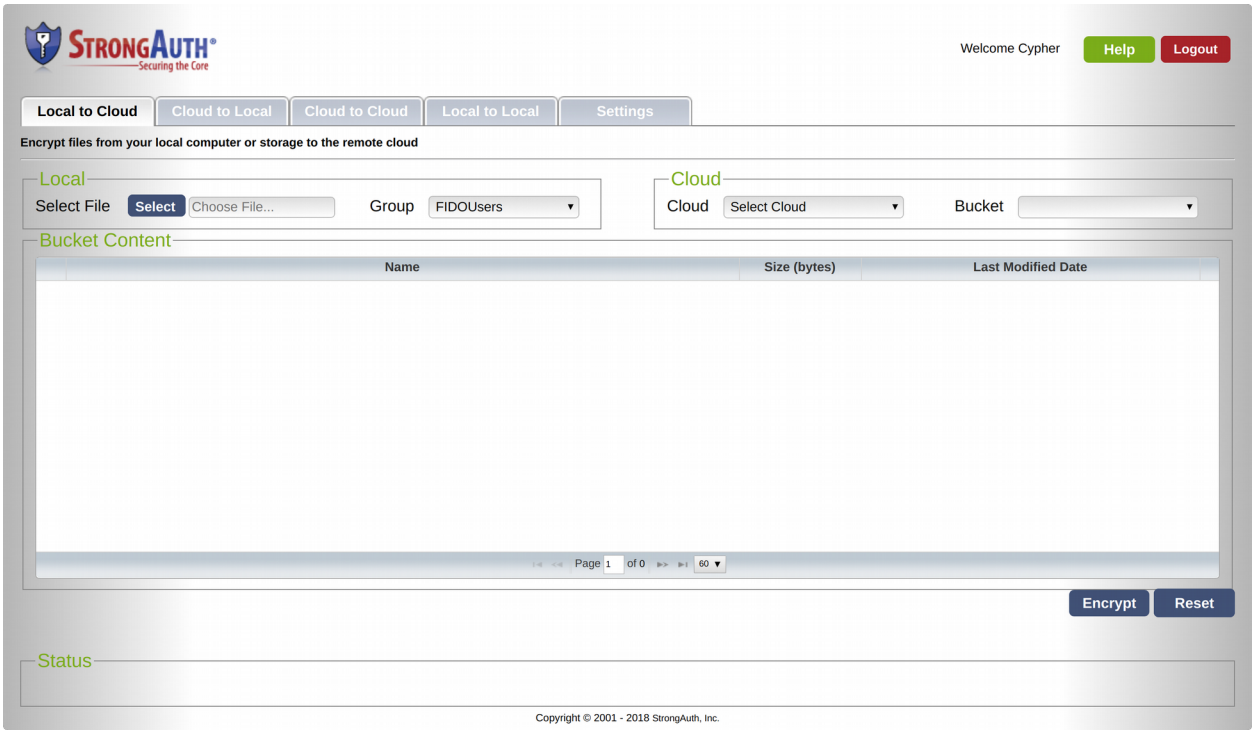


4. Supply a Username while keeping the following rules in mind:
   - *Username* must be between 3 and 30 characters in length
   - *Username* must consist of only lowercase letters (a–z), numbers, and periods
5. Supply a Password and repeat it while keeping the following in mind:
   - *Password* must be at least 6 characters long
6. Click Register to create the account.

7. If the following error message occurs, it implies the *Username* already exists on the system and was chosen by another user

```
[LDAP: error code 68 – The entry
cn=test1,ou=users,ou=v2,ou=SKCE,ou=StrongAuth,ou cannot be added
because an entry with that name already exists]
```

   Click Close and register with a different *Username*.
8. If the credentials are correct, the SKCC application opens:



9. If you see the message, "Successfully added <Username>," the account was created successfully. Click Close to be taken back to the *Login* page.
10. Supply the new credentials and click Login.

2.2—Enable 2-step Verification

In this section of the demonstration you will enable 2-step Verification—a process by which a random, one-time code is sent to a registered email address (supplied by you), so the system can verify your identity.

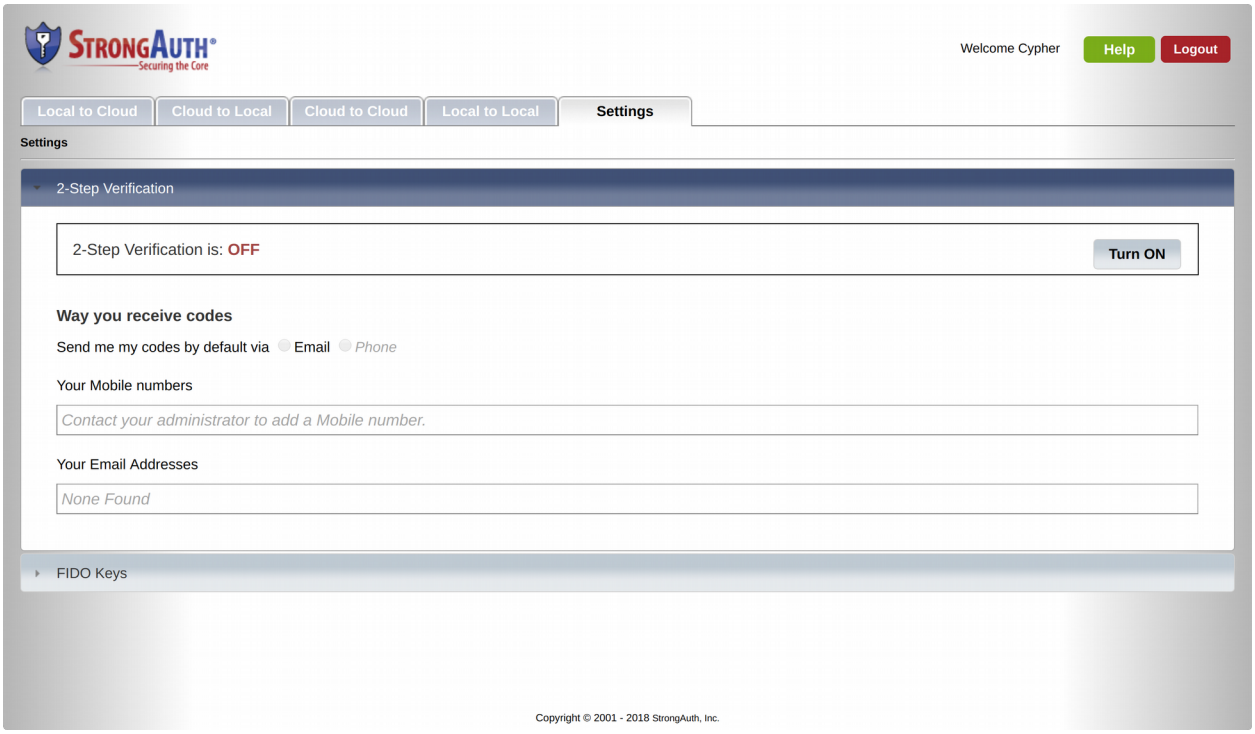
If FIDO authentication is so strong and easy as to supplant passwords, it begs the question: *Why is 2-step Verification necessary?*

Designers of web applications must take into account that a user may forget their FIDO Authenticator at home before coming into work, may lose their FIDO Authenticators, or Authenticators may become accidentally inoperable. In case of any of these events, a web application must allow legitimate users to get back into their accounts without having to spend inordinate amounts of time with Support staff.

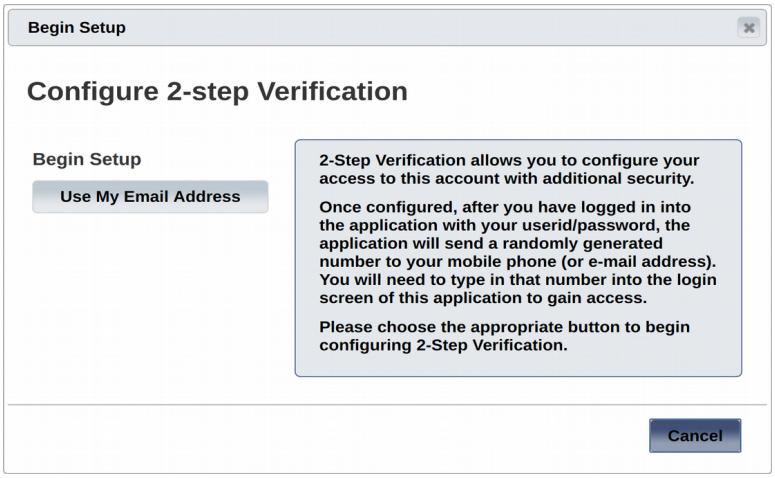
On the assumption that a user's mobile phone or email account is generally secure or in control of the legitimate user, 2-step Verification is a reliable mechanism to enable users to take control of their accounts using a one-time random code sent to email or a mobile phone.

If the user opts to use 2-step Verification to authenticate to the web application—after receiving the random code, the user must supply that code to the web application to gain access to the account. SKCC implements this mechanism as follows:

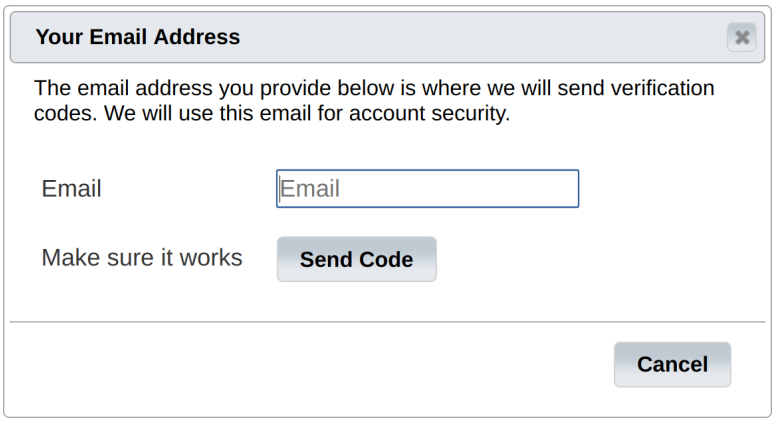
1. In the SKCC application, **click** the **Settings** tab:



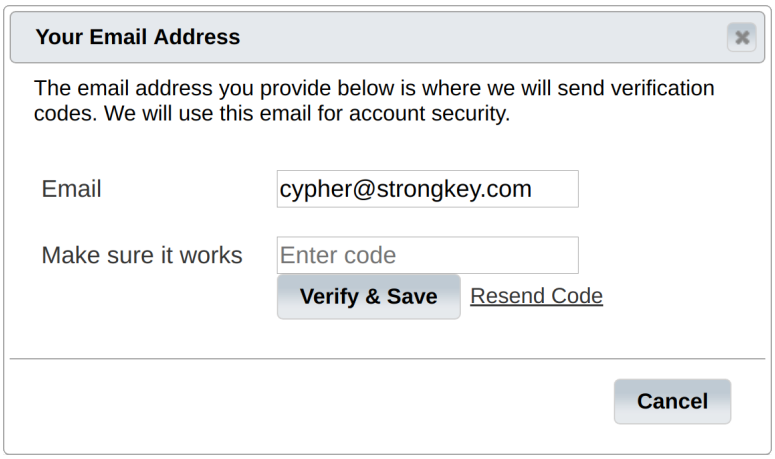
2. If it is not already expanded, click **2-step Verification** to expand the top panel.
3. Click **Turn ON** to activate 2-step Verification. A configuration panel opens.



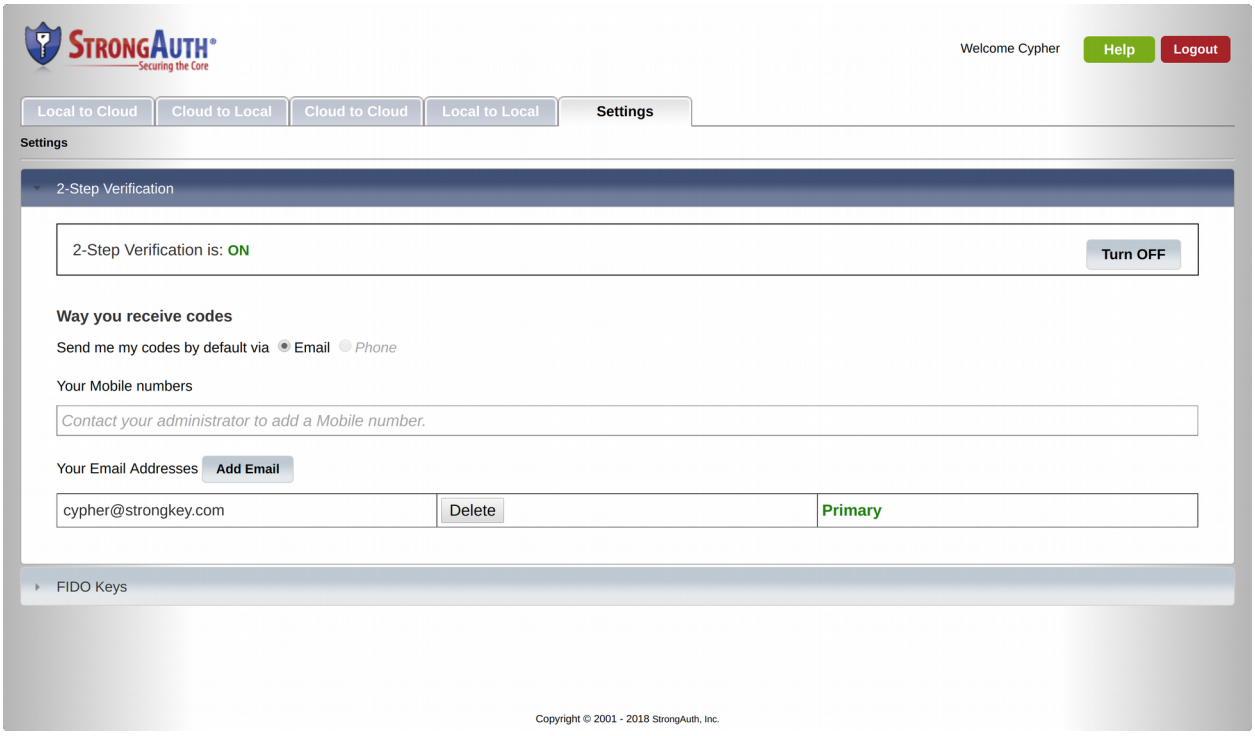
4. Click **Use My Email Address** to configure an email address:



5. **Supply an email address** to which you have access and click **Send Code**.
6. Check the newly registered email account for an email from *info@strongauth.com* and a subject of *Your verification code*. It will have a 6-digit code in the body of the email.
7. Type this 6-digit code in the **Make sure it works** field, and click **Verify & Save**. In case the code times out, or in case the email goes to the Spam folder or is accidentally deleted, click **Resend Code** to send a new verification code to your email address:



8. If your code is verified successfully, you will see 2-step Verification turned *ON*, You will also see the email address you provided stored in the *Your Email Addresses* section of the 2-step Verification accordion page.



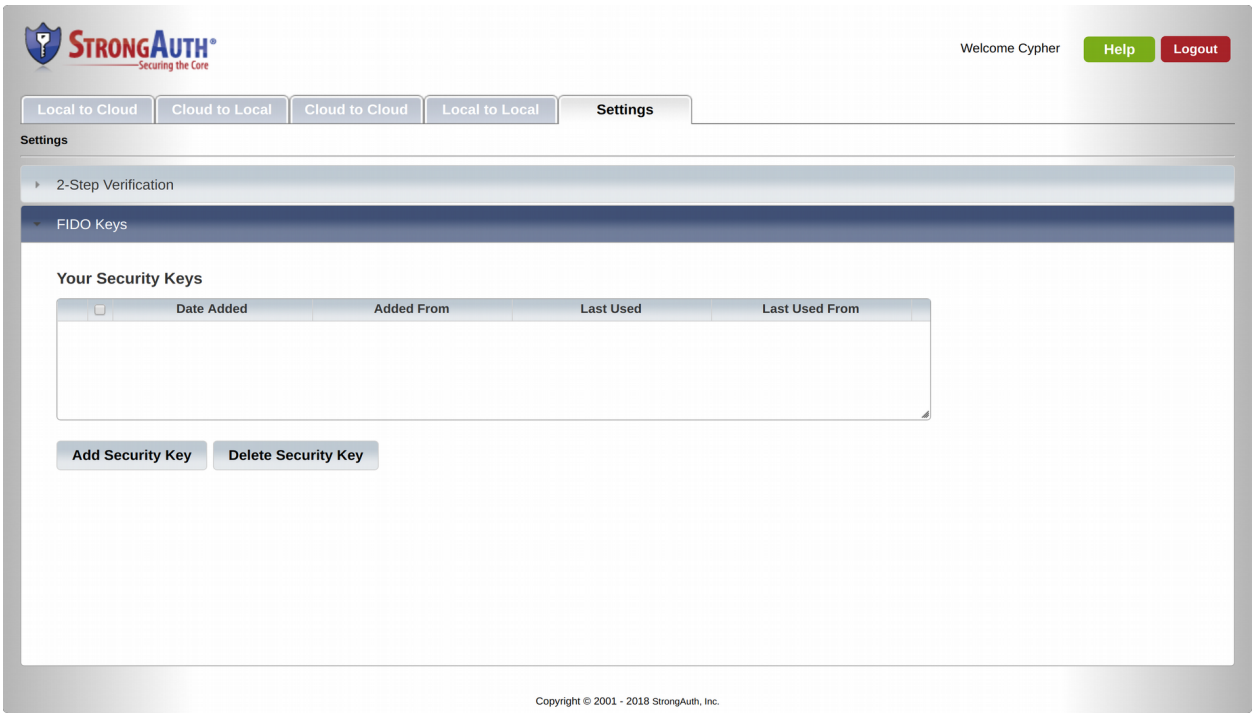
9. If the code you typed in is incorrect, an error message will indicate the code was incorrect. Try again or have a new code sent.

2.3—Register a FIDO Authenticator with an Account

This section explains how to register a unique FIDO cryptographic key (generated on your FIDO Authenticator) with your account.

The industry and text in this document might sometimes refer to this process as "registering a FIDO Authenticator" or "registering a FIDO Token" to simplify it. Please recognize that it really implies the generation of a new and unique cryptographic key pair and the public key of that pair being registered with the website.

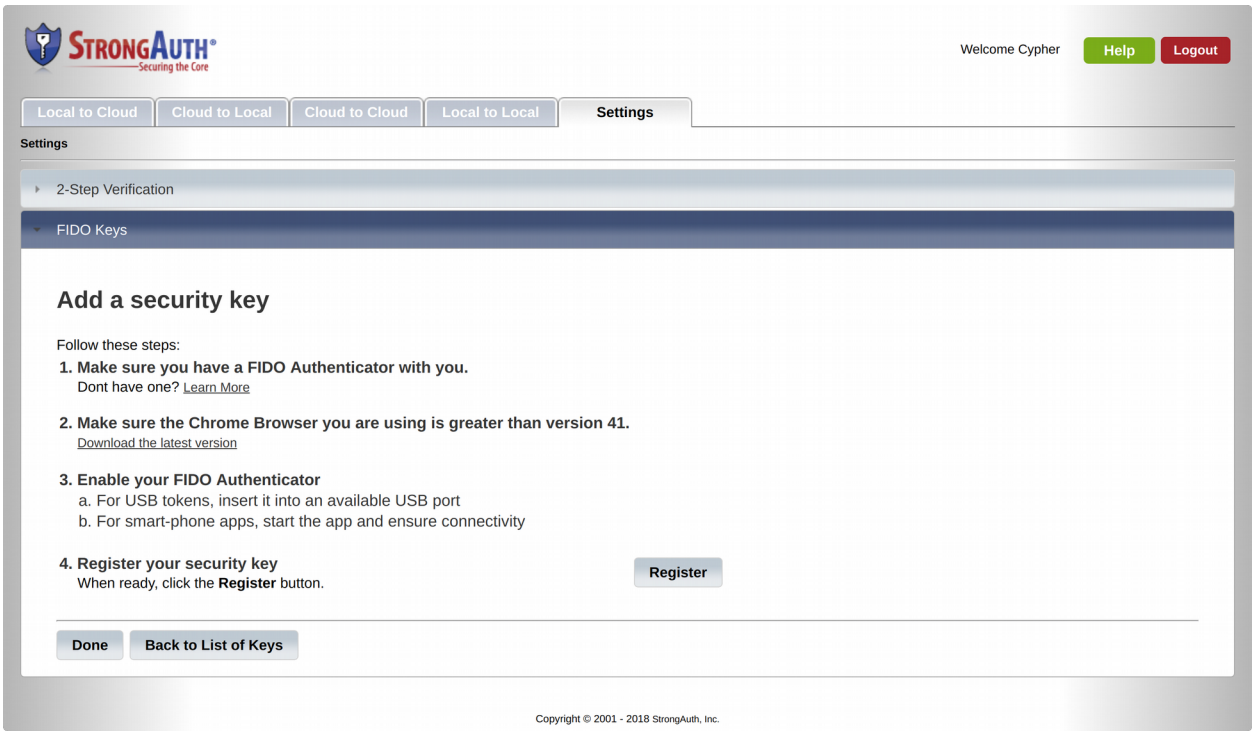
1. If logged out of SKCC, login again—you will now be prompted again to enter the one-time verification code before being granted access to SKCC (because of the previous 2-step Verification process).
2. Navigate to the *Settings* tab and click the **FIDO Keys** label.



3. The page displays a list of **Your Security Keys**. Unless you have registered FIDO keys with this instance of SKCC before (under the username you used to login into the application), the list will be empty.
4. Click **Add Security Key** to register a new cryptographic key on a FIDO Authenticator, for this site, with your account.
5. SKCC will open a panel with instructions on how to register your FIDO Authenticator. Review the instructions to ensure you can meet the requirements.
6. Since the FIDO U2F protocol currently supports using Authenticators only as a *Human Interface Device (HID)* over the *Universal Serial Bus (USB)*, plug the FIDO Authenticator into an available USB port on your computer.

7. Once plugged in, wait for a little (no more than a minute) to ensure any necessary HID device drivers are installed and registered with the operating system. The Microsoft Windows operating system will specifically notify you on the status bar when the Authenticator is ready to use.

> **NOTE:** The installation of device drivers is done only once by the operating system—subsequent operations with the FIDO Authenticator should be possible as soon as the device is plugged into the USB port.

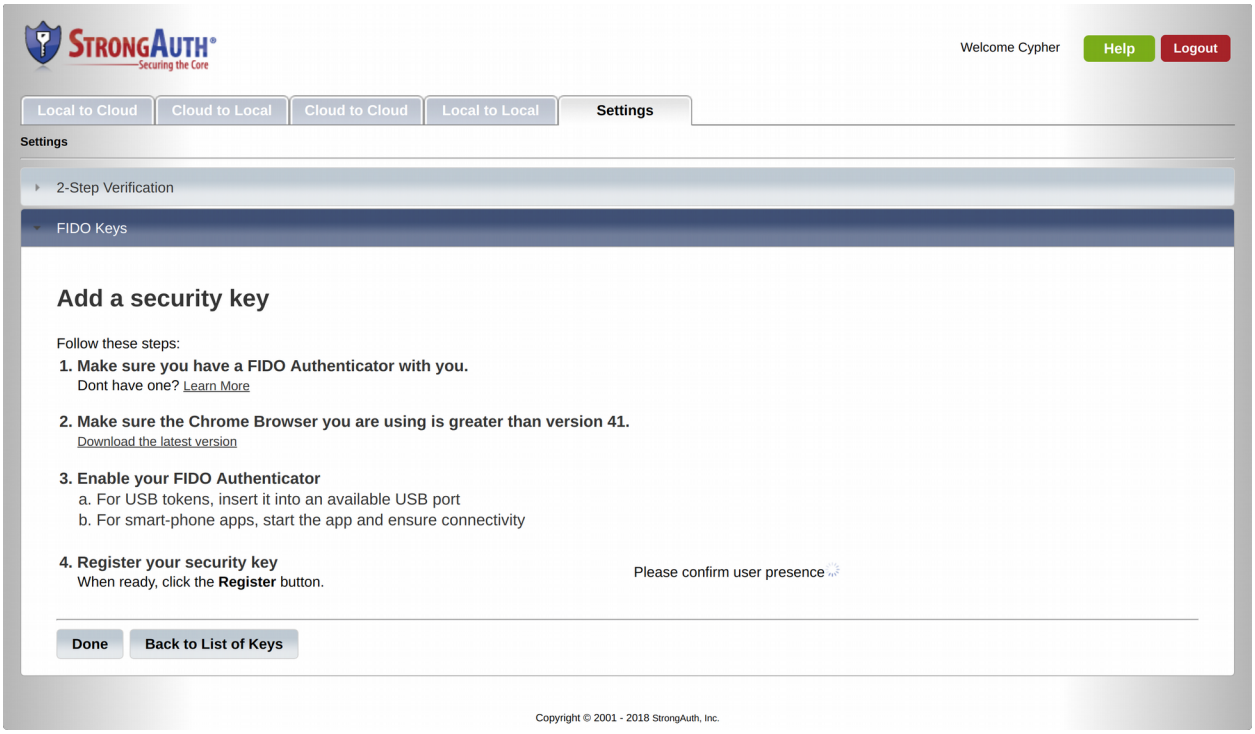


8. When the FIDO Authenticator's device driver is enabled, click **Register**.
9. Within seconds, SKCC should prompt with a message: “Please confirm user presence.”

    This is a requirement of the U2F protocol. U2F-based strong authentication mandates that the user prove to the FIDO Server that they possess a valid U2F Authenticator and are in physical proximity to the device where the browser is running.

    “User presence” is implemented differently from Authenticator to Authenticator, depending on how the manufacturer chose to design their Token. Some expose a metal plate with a blinking *Light Emitting Diode (LED)*, which must be touched by a human finger to verify user presence (Yubico); others have a raised button with a blinking LED that must be pressed or pinched to verify user presence (Hypersecu); yet others require that the Authenticator be removed from the USB port and reinserted to verify user presence (Neowave and Plug-up).

Depending on the type of Authenticator you have, perform the appropriate operation to verify user presence.



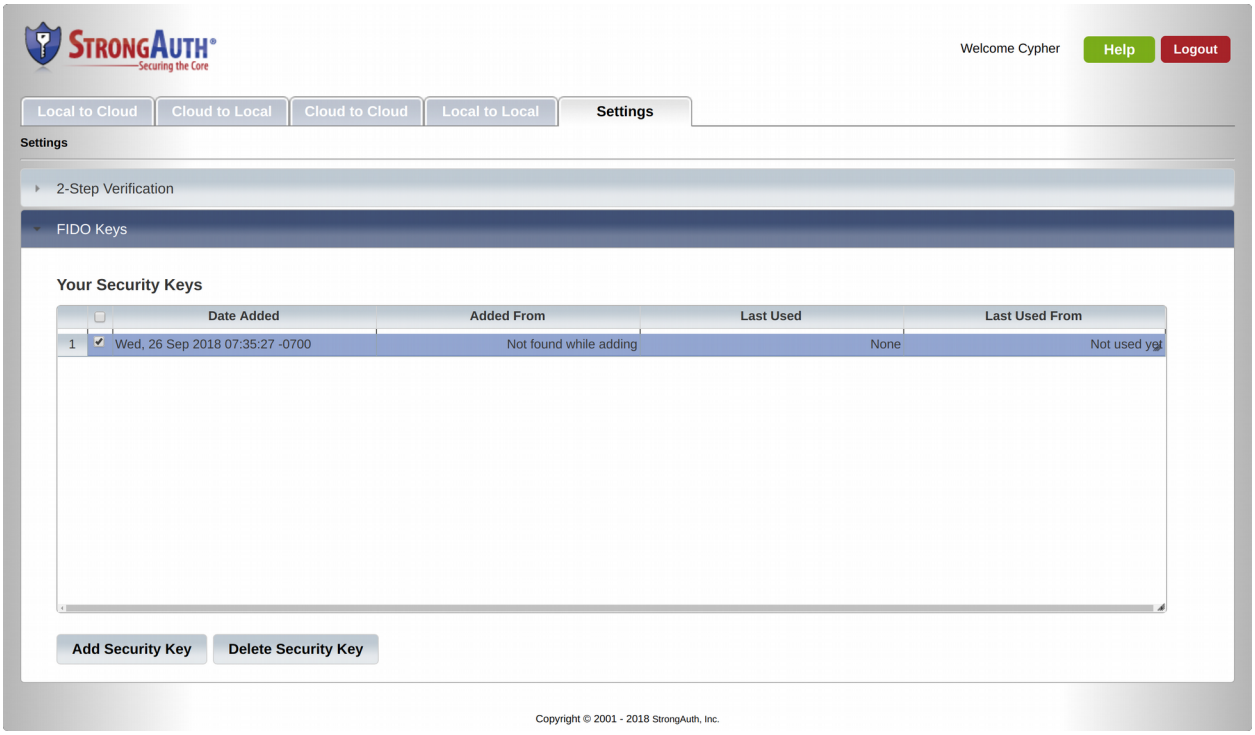
10. There are at least two reasons that a FIDO key registration operation might fail:
    - There is a challenge sent by the FIDO server (in the SKCE) that can time out within 30 seconds; if it does time out, a response from the FIDO authenticator/browser to the FIDO server will be invalid
    - The driver for the FIDO Authenticator may not yet have been installed when the user-presence action was performed; as a result the FIDO Authenticator will not be able to digitally sign the challenge sent by the FIDO Server

    In both cases, you are likely to see the following error:

    SKCC_GUI_MSG_5002: A timeout occurred while waiting for the security key to be tapped. Please try again.

11. To attempt the registration once again, either refresh the browser-page or click one of the buttons—**Done** or **Back to List of Keys**—and then click **Add Security Key** to start over.
12. If key registration succeeds, the message appears: "Successfully registered security key."
13. Click **Done** to go back to the *FIDO Keys* section of the *Settings* tab.
14. When FIDO keys are registered to the account, this page displays the date/time when they were generated, the date/time when they were last used, and—if geo-location retrieval is enabled—the geographical location (city) from which the key was last used.

15. Delete a key on this page, by selecting the key to delete in the list and clicking **Delete Security Key**. See below for why FIDO keys might be deleted from an account.



16. Log out by clicking on the **Logout** button in the upper right.

If FIDO provides strong authentication and protects against hackers, why would one or more keys need deleting?

Good question! As long as you control the FIDO Authenticator and have it in your possession, the premise is that the keys are "good" and can be trusted. However, there is always a possibility that a FIDO Authenticator might get lost; or a batch of Authenticators may be declared unsafe due to a manufacturing defect discovered after the Authenticator was sold on the market, or that there is a vulnerability discovered in an implementation of an Authenticator.

In the above cases, to protect the user account, registered FIDO keys must be deleted to prevent unauthorized people from accessing your account. This design allows users to "manage" keys on their own and protect them from such risks. Once deleted, anyone—including the legitimate user—will be unable to use that FIDO Authenticator to authenticate to that web application (if there are other keys on that Authenticator, registered at other websites, those may also need to be deleted).

In the event the user loses their Authenticator and deletes their registered keys from their account (after having authenticated with 2-step Verification), and then finds the lost Authenticator, they can use the same Authenticator to generate a new key pair and register the key for the same site and account. This is possible because, once a user has deleted their registered key with a site, the FIDO protocol does not recognize that key on the Authenticator even if the key is still present on the Authenticator.

The user can choose to use multiple FIDO Authenticators—a primary and a backup—to register multiple keys with an application site, and use either Authenticator to access the web application. The loss of one Authenticator does not force them to go through a 2-step Verification—they can use the alternate Authenticator to login. The user may also choose to carry one FIDO Authenticator on their key chain, and leave one on their desk at home, or one permanently plugged into their computer; all of these use cases are permissible.

3—Authentication

Now that you have a FIDO Authenticator registered with SKCC, you can strongly authenticate to the user account with the FIDO U2F Authenticator, encrypt and decrypt files, and pair strong authentication with CAPTCHA.

3.1—Authenticate with FIDO and UserID/Password

This example of strong authentication shows a use case authenticating with a userID/oassword and a FIDO Authenticator to access a web application.

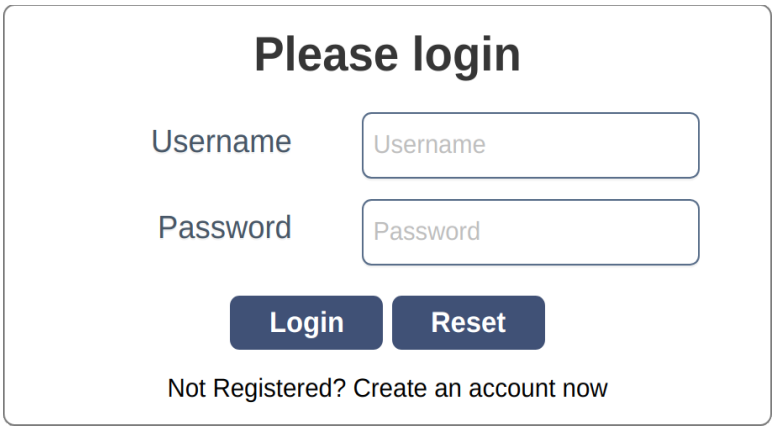
The benefit of this mode of authentication—FIDO+userID/password—is that the user can choose to use a FIDO Authenticator that doesn't mandate local authentication (on the FIDO Authenticator), along with a *Personal Identification Number (PIN)* or some biometric—because the userID/password still protects access to their account even if the FIDO Authenticator is compromised through loss, negligence, or other mishap.

Later sections of this document demonstrate using two other authentication modes:

- FIDO+CAPTCHA—when the web application chooses to dispense authenticating with the userID/password and only uses FIDO strong authentication. This is useful on an internet-facing website to prevent random, drive-by attempts to bog down your web application with authentication requests, while keeping it convenient for legitimate users with FIDO Authenticators because they don't have to remember a password to the site anymore.
- FIDO—when the web application dispenses with userID/password and CAPTCHA completely, and only requires a FIDO Authenticator for strong authentication. This is useful for web applications on an intranet; authentication requests are likely to come only from trusted entities with FIDO Authenticators, *and* when the FIDO Token includes a mechanism to authenticate using a PIN or biometric match on the Authenticator. One doesn't want a legitimate user to lose an Authenticator lacking local authentication, allowing someone else to masquerade as the legitimate user if they happen to find the Authenticator and connect to the web application.

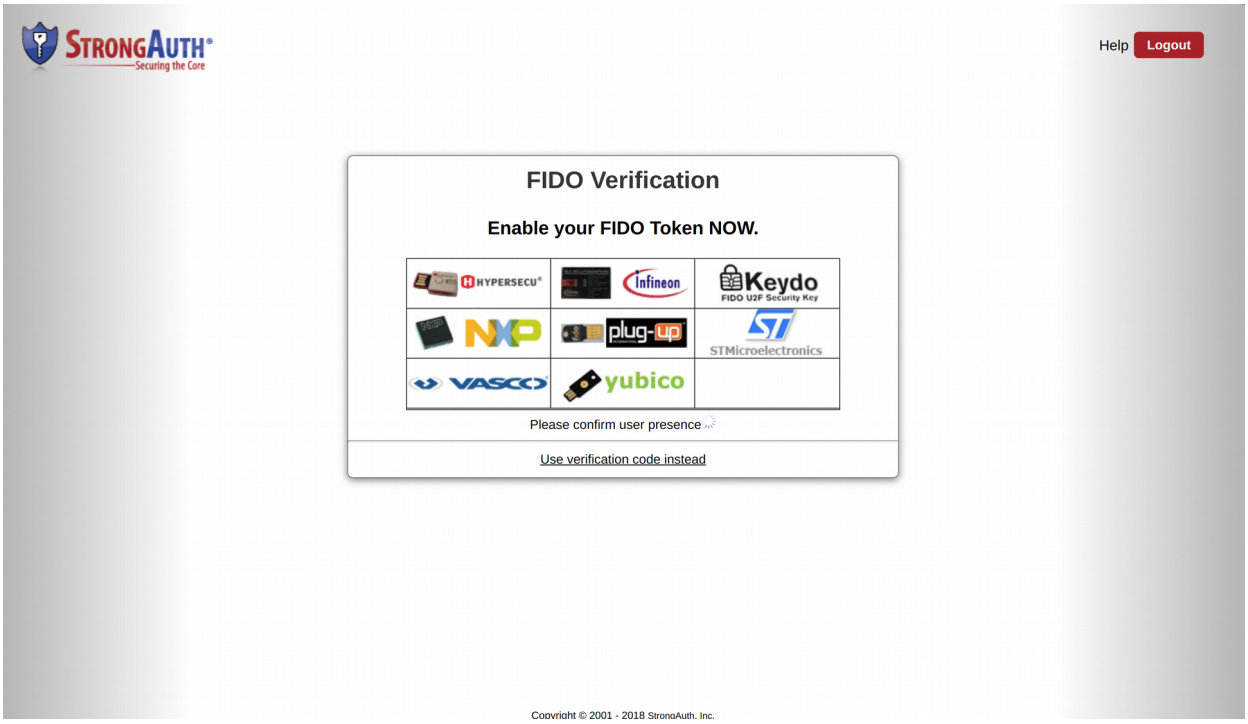
StrongKey is happy to discuss these details at any time; just let us know.

1. At the login page, **type the Username and Password** for the credential you created in this demonstration. When done, click **Login**.



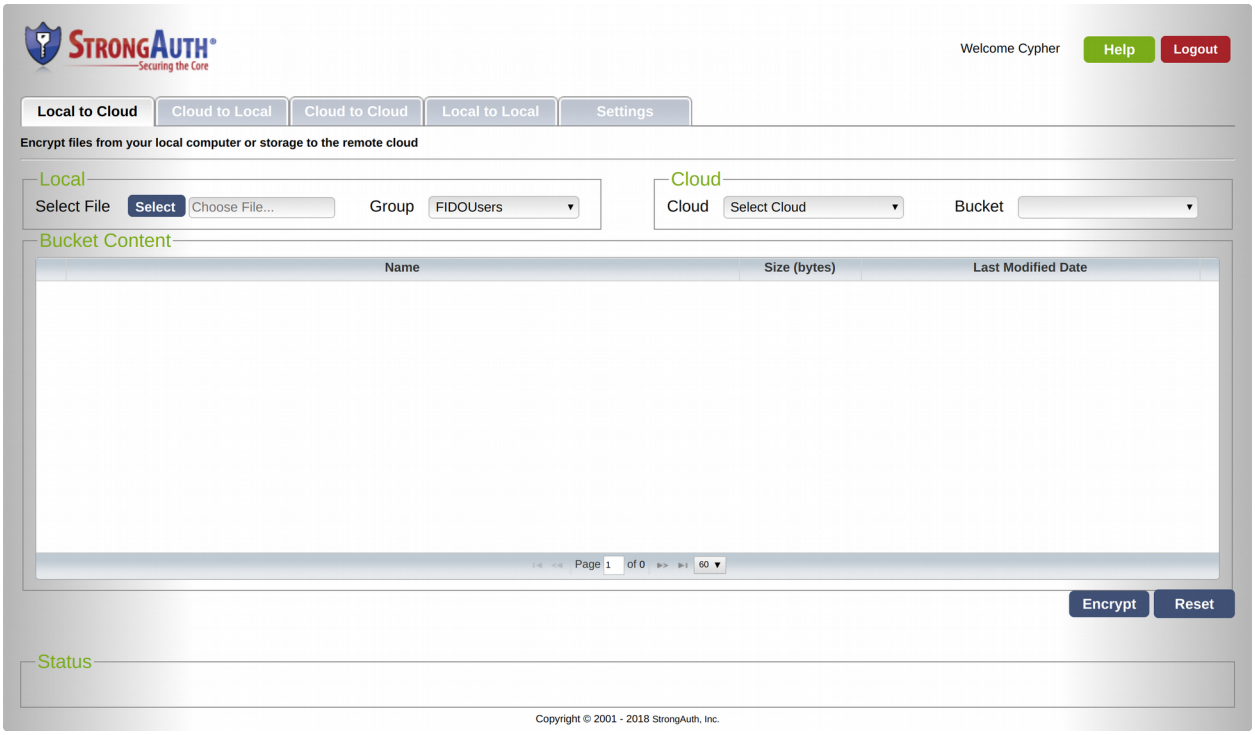
2. Since FIDO authentication is enabled (by the fact that a FIDO key is registered for the application) SKCC automatically prompts with a challenge on the FIDO Verification page asking, “Please confirm user presence.”

3. This page displays the logos of various FIDO U2F Authenticators that were previously tested by StrongKey. Any FIDO CertifiedTM U2F Authenticator will work with SKCC.



Insert the FIDO Authenticator into an available USB port and, depending on the type of Authenticator, perform the appropriate operation to verify user presence.

4. If authentication succeeds, the SKCC web application opens.



5. If you see an error message indicating that a timeout occurred while waiting, click **Retry** and perform the appropriate operation to verify user presence when prompted.
6. If you don't have your FIDO Authenticator, click **Use verification code instead** to use 2-step Verification to authenticate to SKCC.

3.2—Authenticate with FIDO and CAPTCHA

In this section of the demonstration, you will strongly authenticate to using both the FIDO U2F Authenticator and *Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA)* instead of a password.

The benefit of this mode of authentication is that the web application can dispense authenticating the user with a password, thus allowing them to forget the password to the account and never having to reset it. This is useful on an internet-facing website to prevent random, drive-by attempts to bog down the web application with spurious authentication requests while keeping it convenient for legitimate users with FIDO Authenticators.

This demonstration requires connecting to a slightly different URL with the browser: https://fidodemo.strongauth.com/pno. It leads to the same web application, but through a login page that does not prompt for the user's password.

1. At the login page for SKCC, type the **Username** and the **CAPTCHA code** visible on the page for the credential you created in this demonstration. When done, click **Login**:

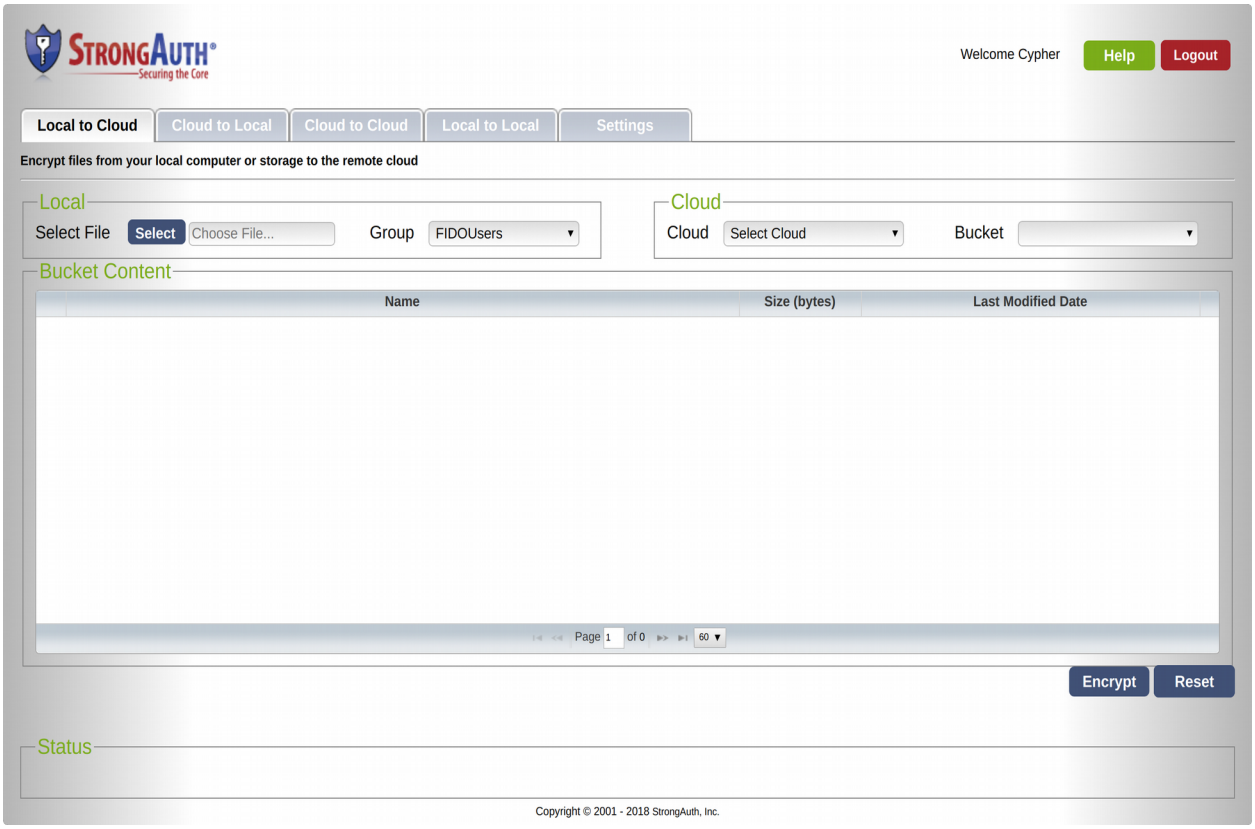
2. Since FIDO authentication is enabled (by the fact that a FIDO key is registered for the application) SKCC automatically prompts with a challenge on the *FIDO Verification* page asking, "Please confirm user presence."

   This page displays the logos of various FIDO U2F Authenticators that were previously tested by StrongKey. Any FIDO Certified™ U2F Authenticator can be used with SKCC.

3. Insert the FIDO Authenticator into an available USB port and, depending on the type

of Authenticator, perform the appropriate operation to verify user presence:

4. If the strong authentication succeeds, SKCC opens.



5. If you see an error message indicating a timeout occurred while waiting, click **Retry** and perform the appropriate operation to verify user presence when prompted.
6. If you don't have your FIDO Token, click **Use verification code instead** to use 2-step Verification to authenticate to SKCC.

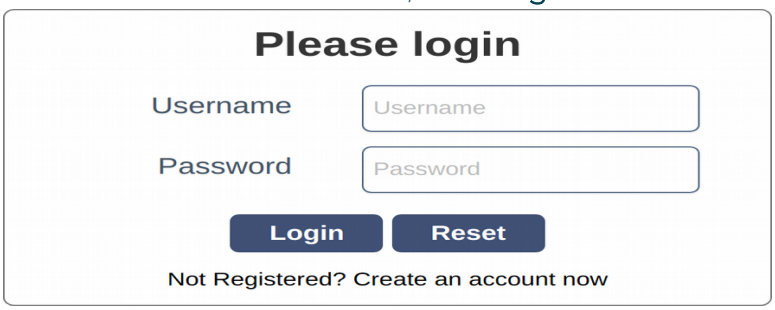
3.3—Authenticate with FIDO Only

In this section of the demonstration, you will strongly authenticate to the account with just the FIDO U2F Authenticator and nothing else—no password or CAPTCHA. Your username is always required in all forms of U2F strong authentication to identify you.

The benefit of this mode of authentication is that the web application can completely dispense with the password or CAPTCHA. If the web application is designed well (to remember the username from a cookie, then it will allow the user to move seamlessly from application to application by clicking on bookmarks, links or URLs without having to type anything – and yet be strongly authenticated with their FIDO Authenticator. This is most useful for intranet web applications where users are authorized to access the applications internally.

This demonstration requires connecting to a slightly different URL with the browser: https://fidodemo.strongauth.com/pnoc. It leads to the same web application, but through a login page that does not prompt for the user's password.

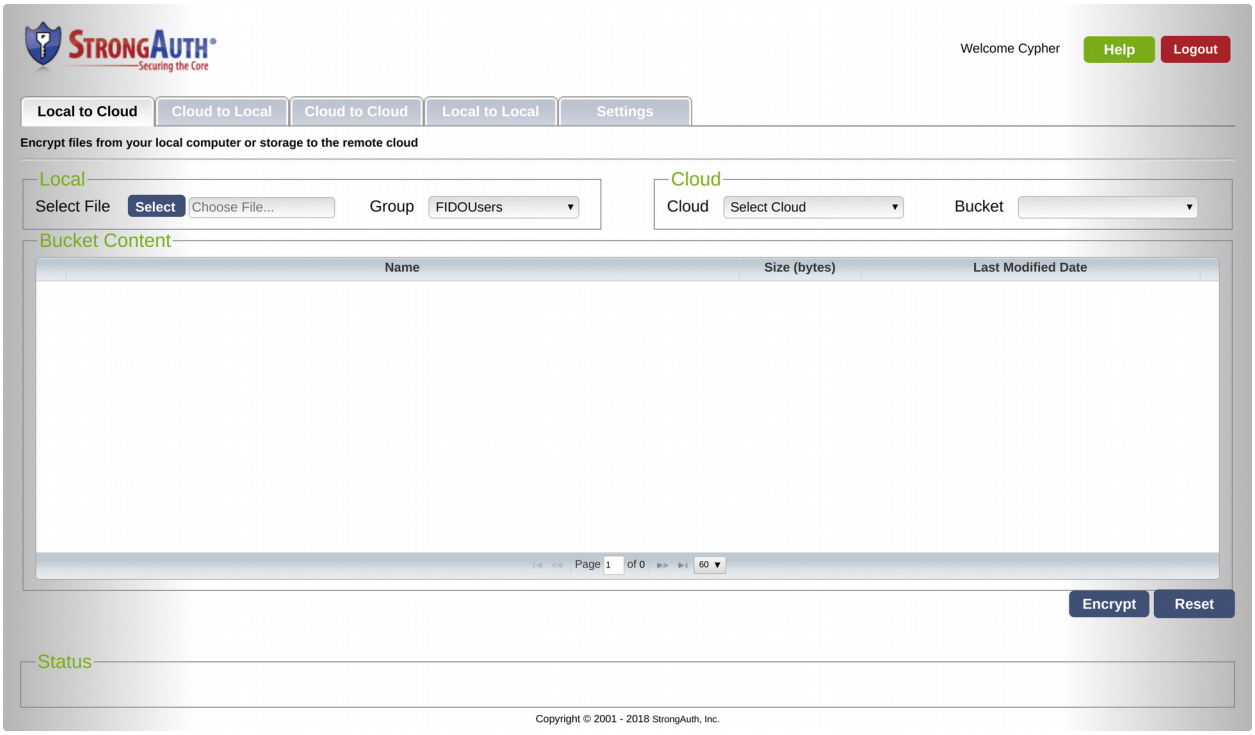
1. At the login page for SKCC, type the **Username** for the credential you created in this demonstration. When done, click **Login**.



2. Since FIDO authentication is enabled (by the fact that a FIDO key is registered for the application) SKCC automatically prompts with a challenge on the *FIDO Verification* page asking, "Please confirm user presence."

3. **Insert the FIDO Authenticator** into an available USB port and, depending on the type of Authenticator, perform the appropriate operation to verify user presence.

4. If the strong authentication succeeds, you are presented with the SKCC web application.



5. If you see an error message indicating a timeout occurred while waiting, click **Retry** and perform the appropriate operation to verify user presence when prompted.
6. If you don't have your FIDO Authenticator, click **Use verification code instead** to use 2-step Verification to authenticate to SKCC.

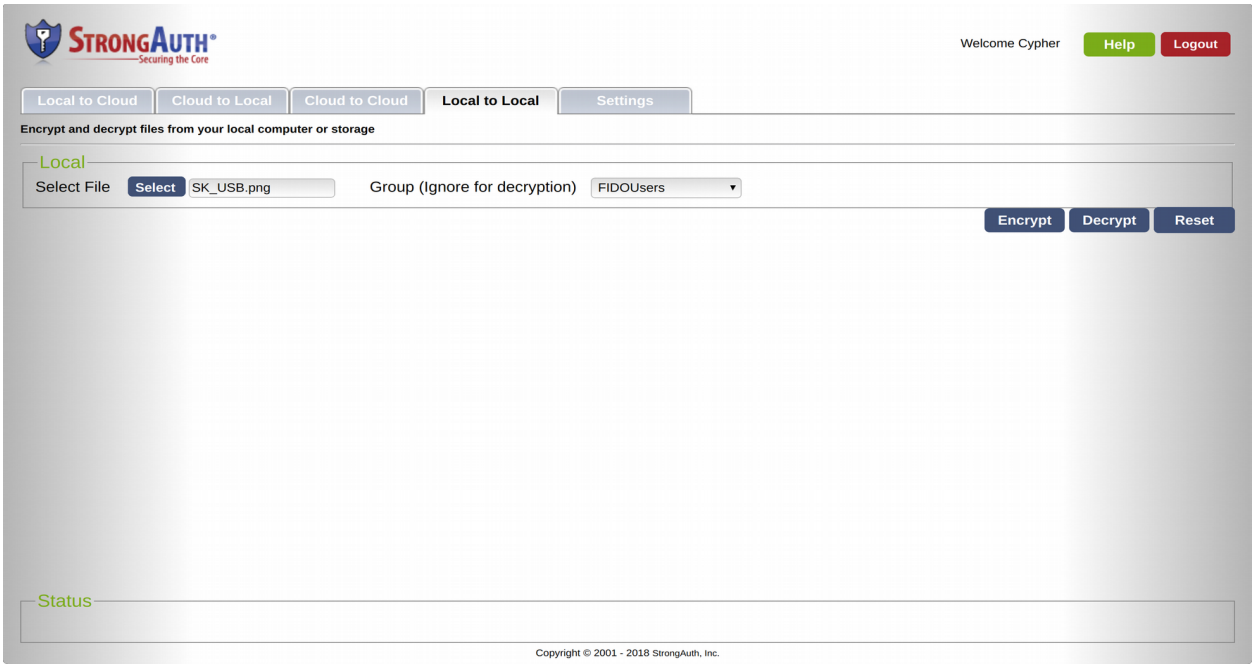
This completes the demonstration of the SKCC and the FIDO strong authentication with your Authenticator. SKCC and SKCE are capable of doing a lot more to protect your sensitive data. Feel free to download them from SourceForge and test them out internally within your company.

If you have any questions, please email us at info@strongkey.com.

4—Encrypt and Decrypt a File

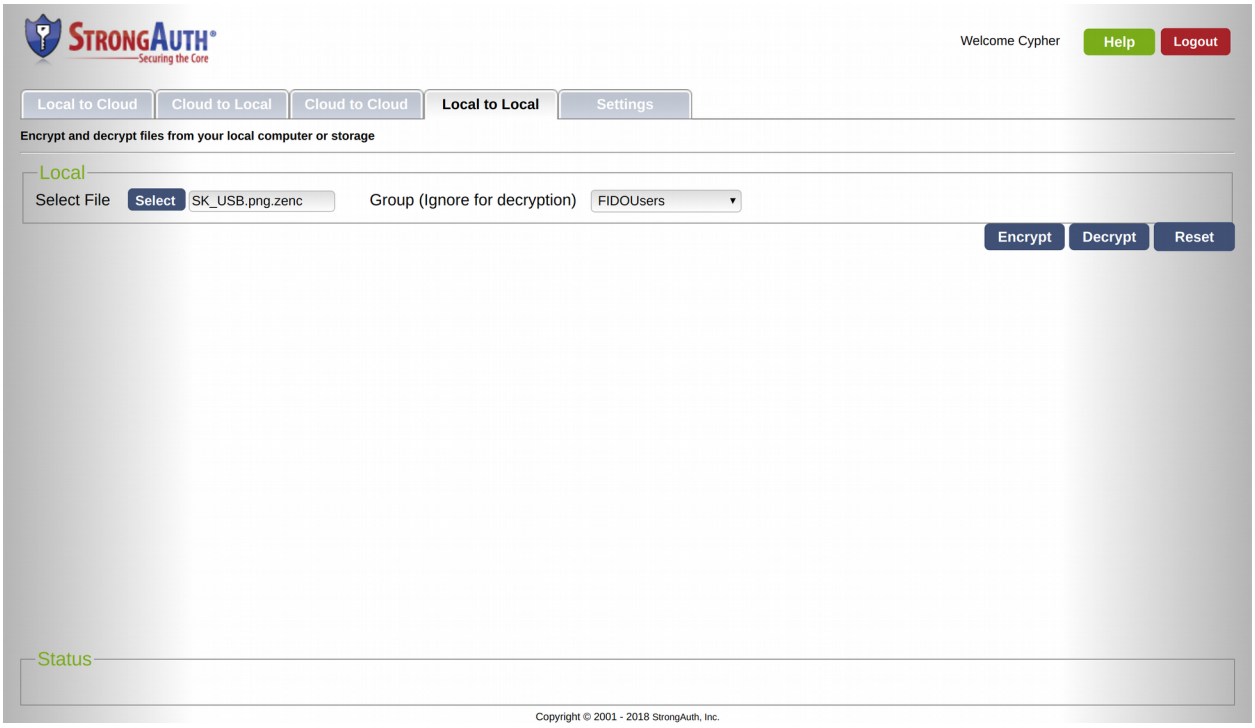
Since SKCC is a web application that encrypts/decrypts files using a centralized key management system and FIDO-based strong authentication, in this section of the demonstration you will learn how to encrypt and decrypt a file local to your computer.

1. After authenticating to SKCC, navigate to the **Local to Local** tab by clicking on it. The *Local to Local* label implies that the file to encrypted (or decrypted) is sourced from your local computer, and the destination for the encrypted (or decrypted) file is your local computer. The other tabs allow for using public or private cloud storage for the encryption/decryption operations, but they are disabled on the *fidodemo.strongauth.com* site.
2. Using the **Select** button, choose a file from your local computer for encryption.
3. By default, the demonstration site is configured with only a single *decryption group* from a *Lightweight Directory Access protocol (LDAP)* Directory Server. Users from this *decryption group*—called *FIDOUsers* on this demonstration site—are authorized to decrypt the file being encrypted. Users on Live/Production sites may select any number of decryption groups from their Directory Server, thereby authorizing one or more users within those groups to decrypt encrypted files.



4. Click **Encrypt**. This initiates an upload of the selected file to the SKCC servlet, where it is handed off to the SKCE Encryption Engine, which in turn generates a symmetric key, escrows the key on a DEMO *StrongAuth KeyAppliance (SAKA)* cluster, encrypts the plaintext file, and returns a *zipped encrypted (.ZENC)* file to be saved in the default download folder of your browser.
5. Click on the **Reset** button to reset the application for another cryptographic operation.

6. Using the **Select** button, select the downloaded encrypted file with the *.ZENC* extension:



7. Click **Decrypt**.
8. SKCC now initiates the upload of the .ZENC file, hands it off to SKCE, which parses the metadata of the ciphertext file to determine the authorized groups/users and the required decryption key (among other things).

   Once authorization is completed, SKCE retrieves the required cryptographic key from the DEMO SAKA cluster using the algorithm determined from the .ZENC file's metadata, decrypts the file, and returns the result to the local computer for storage (without the .ZENC extension).

STRONGKEY

20045 Stevens Creek Boulevard Suite 2A
Cupertino, CA 95014
USA