Changing Password

Changing email password (which means POP3/SMTP password) is certainly doable, but it's not very user-friendly at the moment. There is no UI, and you would have to use the command-line interface to perform it.

Prerequisites

We assume the following:

- You have your private key saved, either as a file or as a list of words. This key was given to you when your identity was created.
- You are comfortable running commands in the command line.

If you don't have your private key, you are out of luck - we don't store user private keys, it's the whole idea. You would have to create a new key and reserve a new name.

Outline

Before we start running commands, it is important to understand the process. There are no user names and passwords within Ubikom ecosystem. Instead, we have private keys (and public keys that are associated with them). However, email clients do not know how to work with keys, so we do a little trick - we pass private key as a name and password, and then the server re-creates the private key based on that.

Each user has two keys. There is a master key (which you could download from the identity page when your identity was created). This key is not given to anyone, and it's not used during daily operations. You keep it secret and only use it when you need to make some changes to your key, name, or endpoint where you receive messages.

Also, you have an email key, which is generated from your POP3/SMTP name and password. This key has limited permissions, it can only encrypt and decrypt messages, but it can't do anything else (for example, re-assign your name to a different key). To do that, you must have the master key.

With this in mind, changing your password would take the following steps:

- Get ubikom-cli command line utility.
- Create your email key from your current user name and password.
- Create a new email key from your current user name and a new password.
- Register your new email key as a child of the master key.
- Re-assign your name to the new email key.
- (Optional) Disable your current email key.

Get ubikom-cli

The best way to get ubikom-cli is to download the source code and run “make build” from the root directory. The generated binary will be located under build/(platform).

Create your email key

We assume your user name is “testuser1001”, your current password is “abcdefgh” and your private key is stored in the file named “testuser1001.private_key”.

Run the following command:

```
$ ubikom-cli create key --out=testuser1001.email_key --from-password=abcdefgh --salt=testuse
Passphrase (enter for none):
Confirm passphrase (enter for none):
15:40:26 WRN saving private key without passphrase
15:40:26 INF private key saved location=testuser1001.email_key
```

We pressed enter to skip the passphrase, and our key was saved in the file testuser1001.email_key.

To make sure your email key is correct, run the following command:

```
$ ubikom-cli lookup key --key=testuser1001.email_key
registration timestamp: 28 Sep 21 15:22 EDT
parent key: 03432be80b9bd5ee1ccfe0c5d6827001b4ba27c9d450864ebdabc8d3fa5dada7f2
```

If your key is incorrect, you will get an error message. Your output will be different (timestamp and the parent key).

Create a new email key

This is similar to the last step, but the password will be different:

```
$ ubikom-cli create key --out=testuser1001.email_key1 --from-password=12345678 --salt=testus
Passphrase (enter for none):
Confirm passphrase (enter for none):
15:40:26 WRN saving private key without passphrase
15:40:26 INF private key saved location=testuser1001.email_key1
```

Now we need to register the newly created key:

```
$ ubikom-cli register key --key=testuser1001.email_key1
15:40:52 DBG generating POW...
15:40:55 DBG POW found pow=01f3540487e78f6a
15:40:55 INF key registered successfully
```

Register the new email key as a child of the master key

Because the old email key was a child of the master key, the new key must be set up in the same way:

```
$ ubikom-cli register child-key --key=testuser1001.private_key --child=testuser1001.email_ke
15:41:46 DBG generating POW...
15:41:48 DBG POW found pow=10dbb7119033c54a
15:41:48 INF child key registered successfully
```

Now if you do key lookup for the new email key, you will see the same parent key.

Re-assign your name to the new email key

Now we are ready to re-assign your name to the new email key:

```
$ ubikom-cli register name testuser1001 --key=testuser1001.private_key --target=testuser1001
15:43:40 DBG generating POW...
15:43:42 DBG POW found pow=5b503dc82a3c4d41
15:43:42 INF name registered successfully
```

As you can see, we use the master key to sign the request, and the new email key to assign the name to.

(Optional) Disable the old key

This step is not strictly necessary, as from now on all the users will use the new key to encrypt messages addressed to you. But it makes things a bit more secure as you explicitly prevent anyone from using your old email key.

Disabling the key is one-way street: once it's disabled, it's dead forever.

To disable the key, do:

```
$ ubikom-cli disable key --key=testuser1001.email_key
```

You will be asked to confirm your decision. After a brief meditation, re-issue the command with the confirmation flag.

Aftermath

Because your old email key is gone, all email currently in flight or store on the proxy server will become undecryptable. All new messages sent to you will be fine, as they will be encrypted with the new key.

Practically it means you might lose a few messages you haven't downloaded already.