Address Verification when Changing Keys for Unchained Capital Vaults

Don’t Trust. Verify.

by @vicariousdrama
649093 - 649108

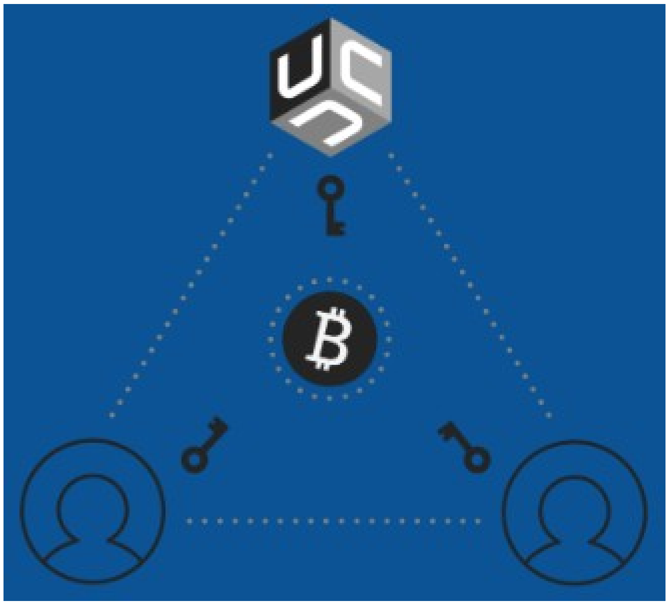


Table of Contents

Summary

Unchained Capital describes itself as a bitcoin native financial services company offering collaborative custody multisignature vaults and loans for bitcoin holders.

The vaults, which are free to setup with a KYC profile, allow the client to control 2 keys while they control a backup key. Periodically, a user may mark a key as lost or stolen or otherwise need or desire to replace it. Within the web application for managing Vaults, addresses are displayed that the user should verify independently to ensure that their signing devices will have the ability to spend. This steps for address verification are not covered in detail within the application so I've prepared this article to give some guidance in that regard.

If you follow along and have a vault to try this with, I hope this will improve your understanding of how to verify addresses in general.

Change History

2020-09-19 – Initial Document
2023-02-19 – Add footers for page numbers, adjust table of contents

Initial Vault & Key Configuration

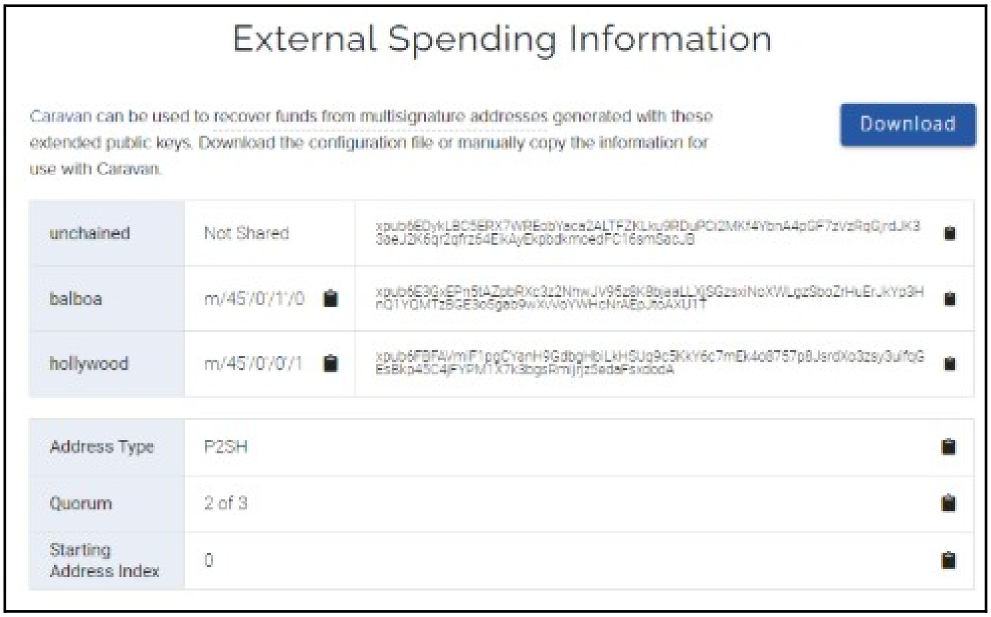
For the purposes of this article, I have a vault that uses 2 keys that are derived from a single hardware device. The names and BIP32 paths are as follows

- hollywood   - m/45'/0'/0'
- balboa   - m/45'/0'/1'

When the vault was created with these keys, the "account number" for each was 0, making the full base derived path as follows

- hollywood   - m/45'/0'/0'/0
- balboa   - m/45'/0'/1'/0

The xpubs for each, along with the unchained key can be seen on the **External Spend Information** dialog accessible from the menu of the vault's **Transact** section. Anytime a vault is created, or a key is changed, the information from this screen should be retained. I recommend printing it and keeping with your records. The **Download** button produces a JSON file that can be used directly with Caravan, converted for use with Electrum and other wallets and is also suitable for printing.



External Spending Information

Caravan can be used to recover funds from multisignature addresses generated with these extended public keys. Download the configuration file or manually copy the information for use with Caravan.

unchained	Not Shared	xpub6EDykLBC5ERX7WREobYaca2ALTFZKLku9RDuPCi2MKf4YbnA4pGF7zVzRqGjrdJK33aeJ2K6qr2qfrz64EikAyEkpbdkmoedFC16smSacJB
balboa	m/45'/0'/1'/0	xpub6E3GxEPn5tAZpbRXc3z2NhwJV95z8K8bjaaLLXjSGzsxiNoXWLgzSboZrHuErJkYp3HnQ1YQMTzBGE3o5gab9wXVVoYWHcNrAEpJtoAXU1T
hollywood	m/45'/0'/0'/1	xpub6FBFAVmiF1pgCYanH9GdbgHbiLkHSUq9c5KkY6c7mEk4o8757p8JsrdXo3zsy3ufqGEsBkp45C4jFYPM1X7k3bgsRmjjrjz5edaFsxdodA

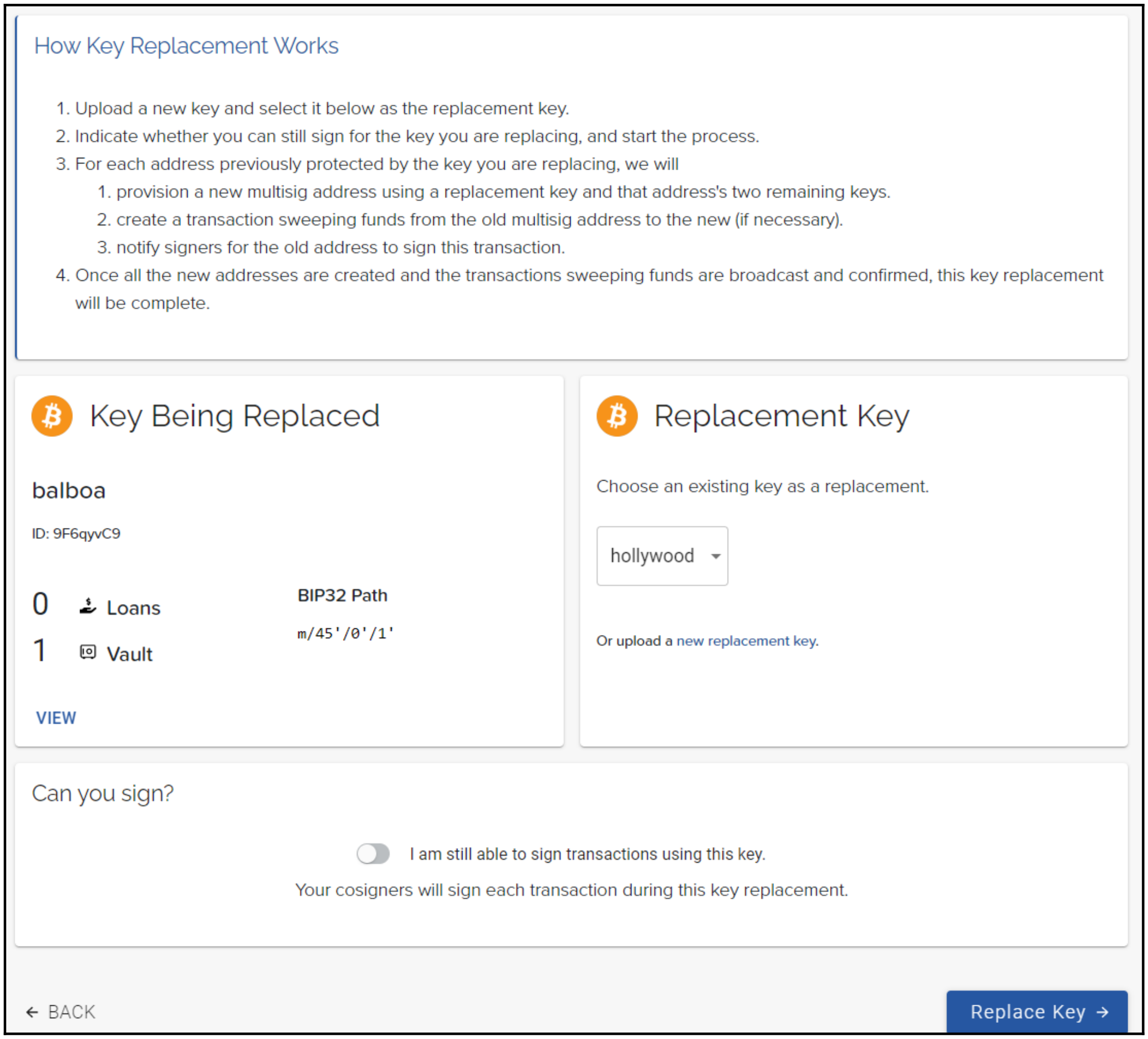
Address Type	P2SH
Quorum	2 of 3
Starting Address Index	0

In this case, the unchained key has the following xpub

> xpub6EDykLBC5ERX7WREobYaca2ALTFZKLku9RDuPCi2MKf4YbnA4pGF7zVzRqGjrdJK33aeJ2K6qr2qfrz64EikAyEkpbdkmoedFC16smSacJB

Key Replacement Process

When a key is replaced on a vault, a new “wallet” is formulated with new addresses. A transaction to sweep funds from the old “wallet” to the new one in the vault is established, signed with remaining keys, and broadcast. The web application does a good job of explaining this and walking through the process.



How Key Replacement Works

1. Upload a new key and select it below as the replacement key.
2. Indicate whether you can still sign for the key you are replacing, and start the process.
3. For each address previously protected by the key you are replacing, we will
   1. provision a new multisig address using a replacement key and that address's two remaining keys.
   2. create a transaction sweeping funds from the old multisig address to the new (if necessary).
   3. notify signers for the old address to sign this transaction.
4. Once all the new addresses are created and the transactions sweeping funds are broadcast and confirmed, this key replacement will be complete.

Key Being Replaced

balboa

ID: 9F6qyvC9

0 Loans
1 Vault

BIP32 Path
m/45'/0'/1'

Replacement Key

Choose an existing key as a replacement.

hollywood

Or upload a new replacement key.

Can you sign?

○ I am still able to sign transactions using this key.

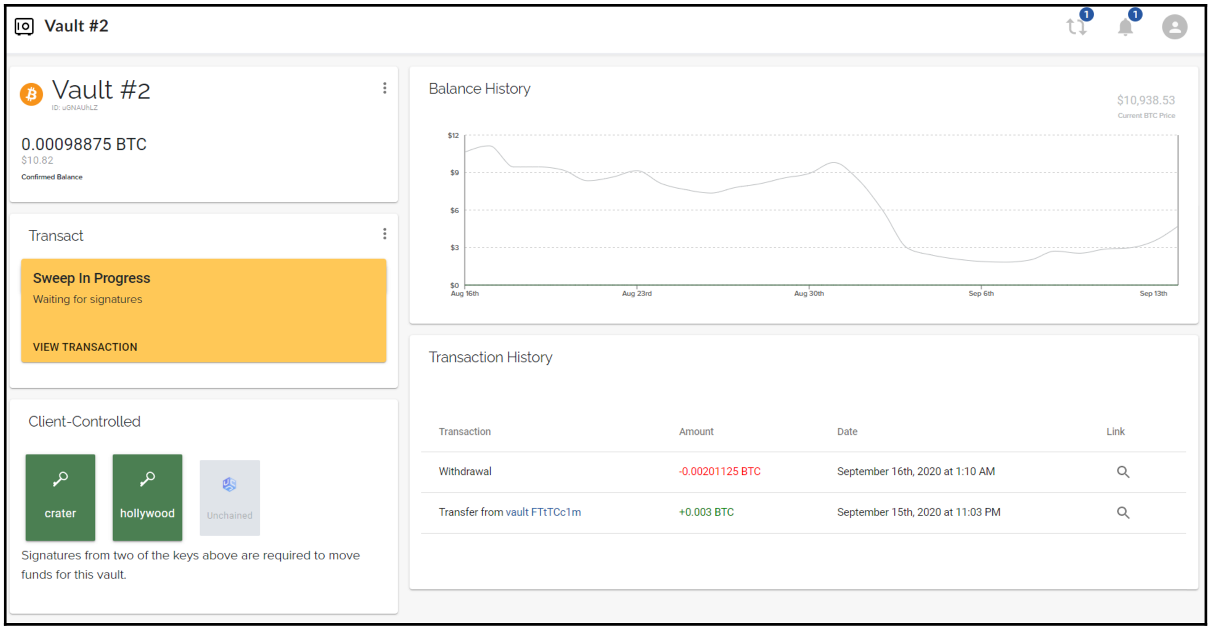
Your cosigners will sign each transaction during this key replacement.

For purposes of this example, I create a new key which has the following base BIP32 path

- crater - m/45'/0'/2'

I plan to replace balboa and specify crater as the Replacement Key. Since I still have access to balboa to be able to sign, I toggle that option on and continue clicking **Replace Key**.

The vault screen depicts the new key and that a Sweep transaction is in progress.



Vault #2

ID: uGNAUhLZ

0.00098875 BTC

$10.82

Confirmed Balance

Transact

Sweep In Progress

Waiting for signatures

Client-Controlled

Signatures from two of the keys above are required to move funds for this vault.

Balance History

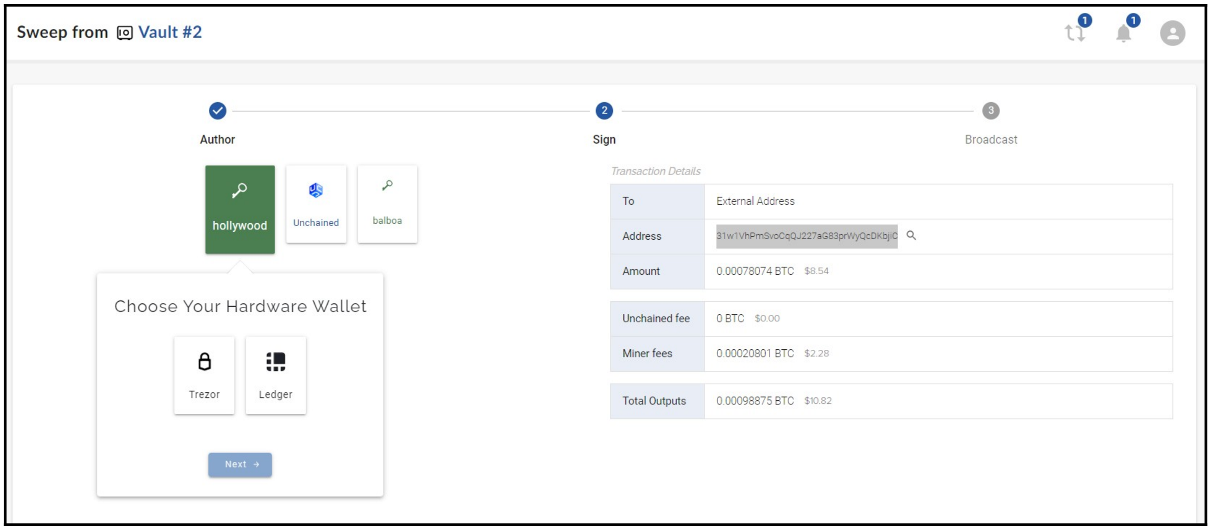
$10,938.53

Current BTC Price

Transaction History

Transaction	Amount	Date	Link
Withdrawal	-0.00201125 BTC	September 16th, 2020 at 1:10 AM	🔍
Transfer from vault FTtTCc1m	+0.003 BTC	September 15th, 2020 at 11:03 PM	🔍

Transaction details show the keys that can sign for this transaction. Note that balboa is present, but crater is not. The address funds are to be sent to is **31w1VhPmSvoCqQJ227aG83prWyQcDKbjiC** and is what I want to verify.



Sweep from Vault #2

Choose Your Hardware Wallet

Transaction Details

To	External Address
Address	31w1VhPmSvoCqQJ227aG83prWyQcDKbjiC
Amount	0.00078074 BTC $8.54
Unchained fee	0 BTC $0.00
Miner fees	0.00020801 BTC $2.28
Total Outputs	0.00098875 BTC $10.82

The External Spend Information for the vault still reflects the existing key setup with unchained, balboa, and hollywood, and cannot be used to verify the address in external tools.

Determining the New External Spend Information

To verify the address for the new wallet within the vault, its important to understand how the multi-signature wallet is setup.

The full derivation path for keys used by Unchained Capital vaults and addresses is segmented as follows

Depth 1: 45'	Hardened. Indicates it is for multisig
Depth 2: 0'	Hardened. Indicates it is for mainnet
Depth 3: 0'	Hardened. The account number
Depth 4: 0	Product key for an account, incremented as used in a vault or a loan
Depth 5: 0	0 for a deposit address, 1 for a change address. Unchained Capital doesn’t support change addresses yet so this is always 0.
Depth 6: 0	The address depth which increments as addresses are used.

Depth 4 is what we need to track to for the new key. Recall that when the replacement key crater was made, it had a BIP 32 path of m/45'/0'/2'.

Each time a key is used in a different vault, it’s product key number is incremented. The very first time it is used on a vault, it starts at 0.

The xpubs that comprise the new wallet are as follows

- xpub for hollywood key with derived path m/45'/0'/0'/1
- xpub for new crater key with derived path m/45'/0'/2'/0
- Xpub for unchained key xpub6EDykLBC5ERX7WREobYaca2ALTFZKLku9RDuPCi2MKf4YbnA4pGF7zVzRqGjrdJK33aeJ2K6qr2qfrz64EikAyEkpbdkmoedFC16smSacJB

If I had not created crater, and instead replaced with another key in my account, then the 4th depth would have incremented. For example, A key associated with 3 other vaults (active or closed) with a base path of m/45'/0'/99'/ would have derived paths of m/45'/0'/99'/0, m/45'/0'/99'/1, and m/45'/0'/99'/2. The next derived path would be incremented to m/45'/0'/99'/3.

Most of the remaining aspects of External Spend Information remain the same. The Address Type is P2SH, and the Quorum is 2 of 3. The starting address index will differ and likely directly associated to the total number of addresses already used in the vault.

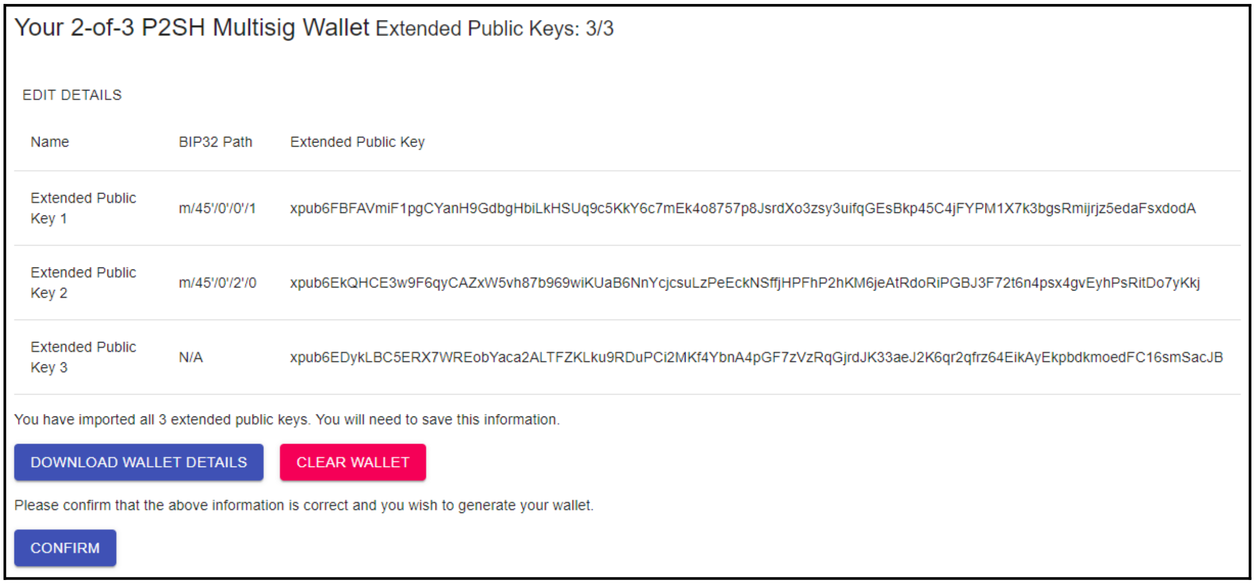
Verification with Caravan

With the newly derived External Spend Information, we can load this up in the Caravan which is accessible here:
https://unchained-capital.github.io/caravan/#/wallet

For **Extended Public Key 1**, connect the hardware device for the remaining key, choose the type and specify the BIP32 Path down to the product key taking care to set the apostrophes where required for hardening. In my example, I enter **m/45'/0'/0'/1** for my hollywood key. Click **Import Extended Public Key** and follow on screen instructions.

Similarly for Extended Public Key 2, repeat the process, but for the new key. In my example, I enter **m/45'/0'/2'/0** for my crater key and click **Import Extended Public Key.**

Finally for Extended Public Key 3, I choose **Enter as text**, and specify the xpub for the Unchained key. Upon doing so, a summary is displayed



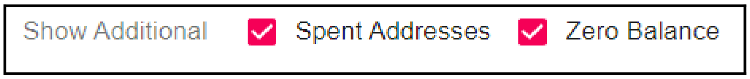
Your 2-of-3 P2SH Multisig Wallet Extended Public Keys: 3/3

Name	BIP32 Path	Extended Public Key
Extended Public Key 1	m/45'/0'/0'/1	xpub6FBFAVmiF1pgCYanH9GdbgHbiLkHSUq9c5KkY6c7mEk4o8757p8JsrdXo3zsy3uifqGEsBkp45C4jFYPM1X7k3bgsRmijrjz5edaFsxdodA
Extended Public Key 2	m/45'/0'/2'/0	xpub6EkQHCE3w9F6qyCAZxW5vh87b969wiKUaB6NnYcjcsuLzPeEckNSffjHPFhP2hKM6jeAtRdoRiPGBJ3F72t6n4psx4gvEyhPsRitDo7yKkj
Extended Public Key 3	N/A	xpub6EDykLBC5ERX7WREobYaca2ALTFZKLku9RDuPCi2MKf4YbnA4pGF7zVzRqGjrdJK33aeJ2K6qr2qfrz64EikAyEkpbdkmoedFC16smSacJB

You have imported all 3 extended public keys. You will need to save this information.

Please confirm that the above information is correct and you wish to generate your wallet.

When clicking confirm, you will likely see that there is 0 BTC in the multisig wallet, and no records to display for addresses. At the bottom of the screen, check the boxes for Spent Addresses and Zero Balances.



Show Additional ☑ Spent Addresses ☑ Zero Balance

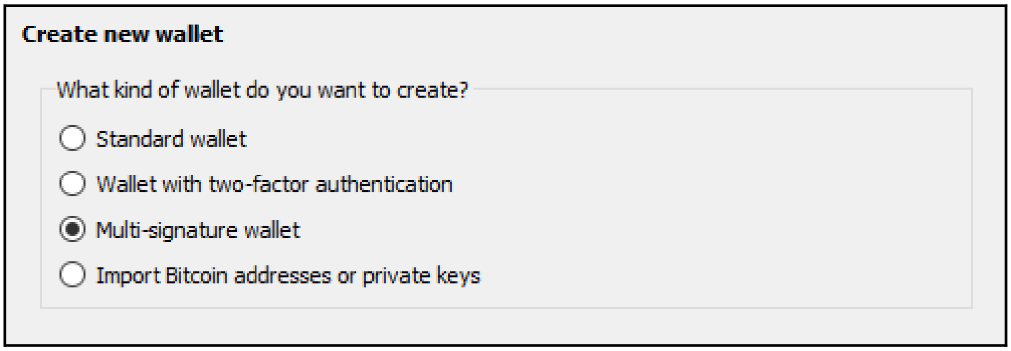
The addresses are displayed, starting with the first 10.

I can verify address **31w1VhPmSvoCqQJ227aG83prWyQcDKbjiC** is at index 2.

Verification with Electrum

This process can also be done with Electrum.

Create new multi-signature wallet



Create new wallet

What kind of wallet do you want to create?

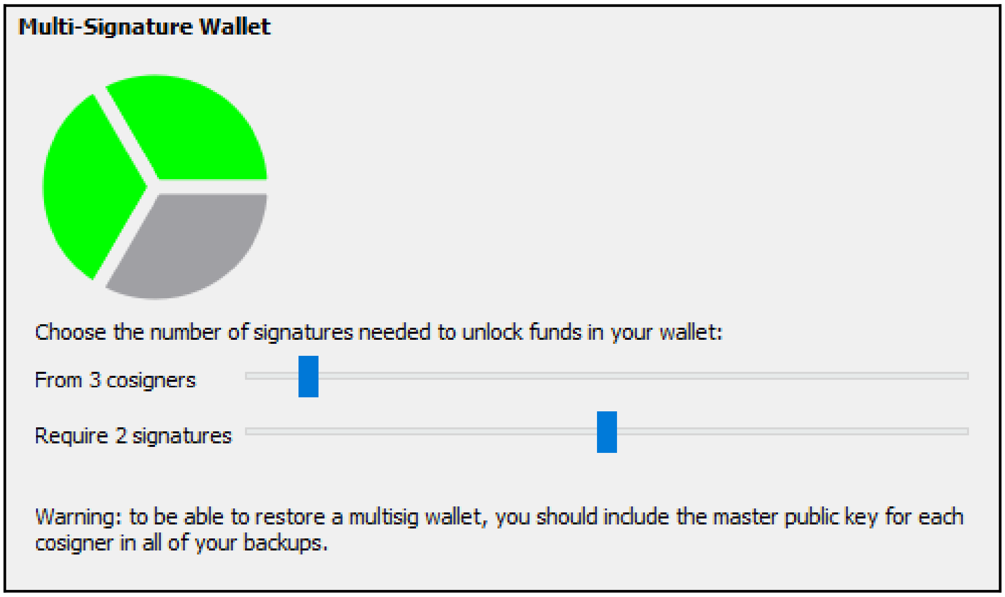
○ Standard wallet

○ Wallet with two-factor authentication

● Multi-signature wallet

○ Import Bitcoin addresses or private keys

Specify 2 signatures required, of 3 cosigners



Multi-Signature Wallet

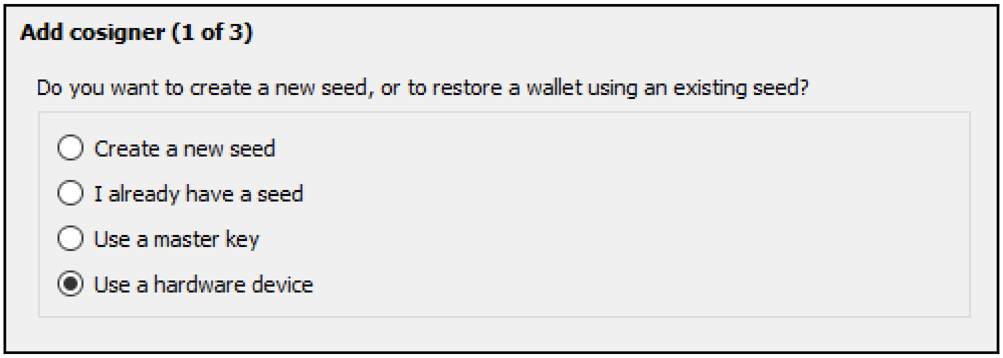
Choose the number of signatures needed to unlock funds in your wallet:

From 3 cosigners

Require 2 signatures

Warning: to be able to restore a multisig wallet, you should include the master public key for each cosigner in all of your backups.

Import the public key for cosigner 1

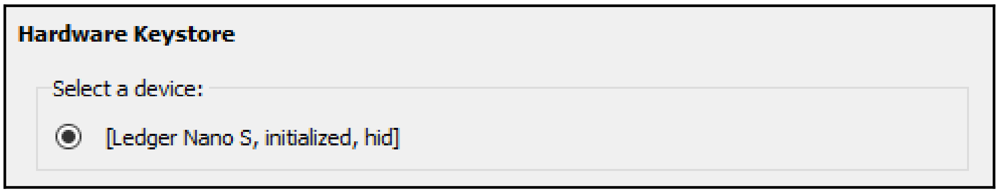


Add cosigner (1 of 3)

Do you want to create a new seed, or to restore a wallet using an existing seed?

○ Create a new seed
○ I already have a seed
○ Use a master key
● Use a hardware device

Scan devices

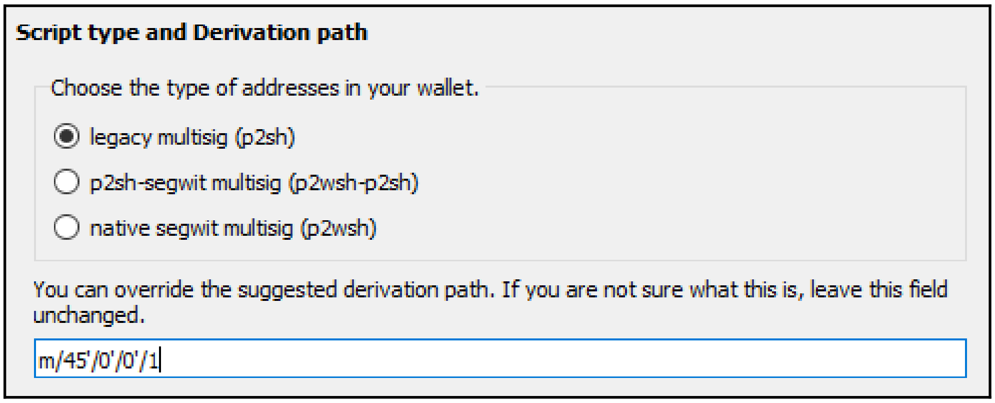


Hardware Keystore

Select a device:

● [Ledger Nano S, initialized, hid]

Specify script type and derivation path



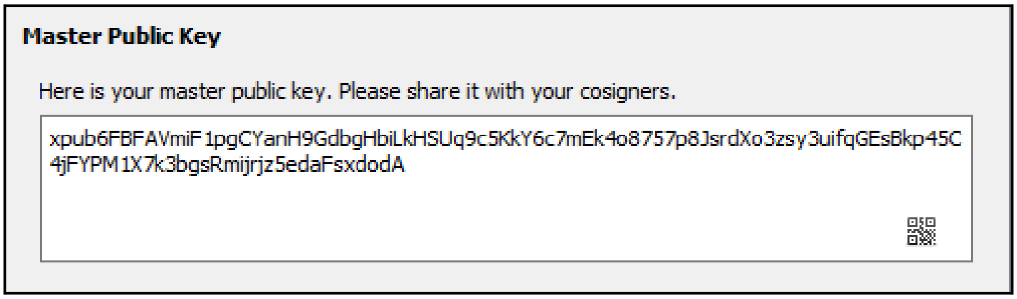
Script type and Derivation path

Choose the type of addresses in your wallet.

● legacy multisig (p2sh)
○ p2sh-segwit multisig (p2wsh-p2sh)
○ native segwit multisig (p2wsh)

You can override the suggested derivation path. If you are not sure what this is, leave this field unchanged.

```
m/45'/0'/0'/1
```

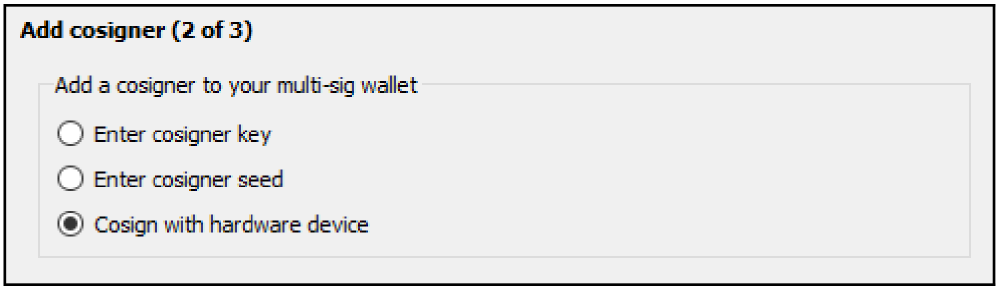


Master Public Key

Here is your master public key. Please share it with your cosigners.

```
xpub6FBFAVmiF1pgCYanH9GdbgHbiLkHSUq9c5KkY6c7mEk4o8757p8JsrdXo3zsy3uifqGEsBkp45C4jFYPM1X7k3bgsRmijrjz5edaFsxdodA
```

For device 2, I repeat the process

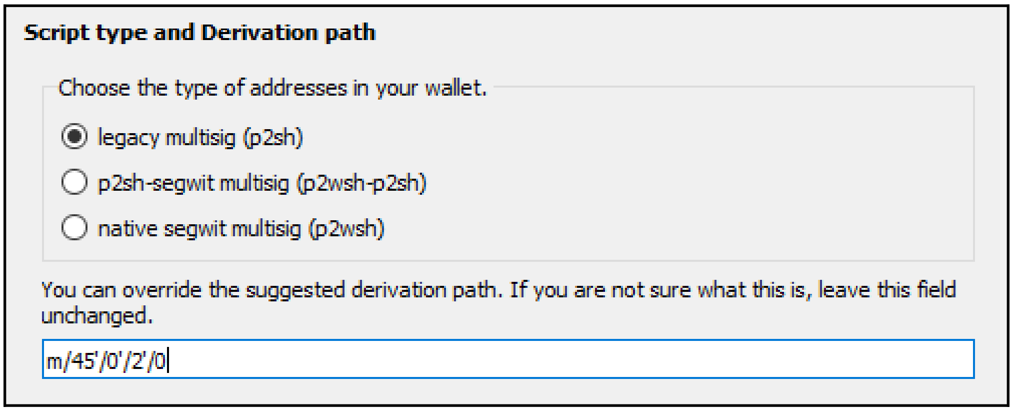


Add cosigner (2 of 3)

Add a cosigner to your multi-sig wallet

○ Enter cosigner key
○ Enter cosigner seed
● Cosign with hardware device

And use it’s device and derivation path



Script type and Derivation path

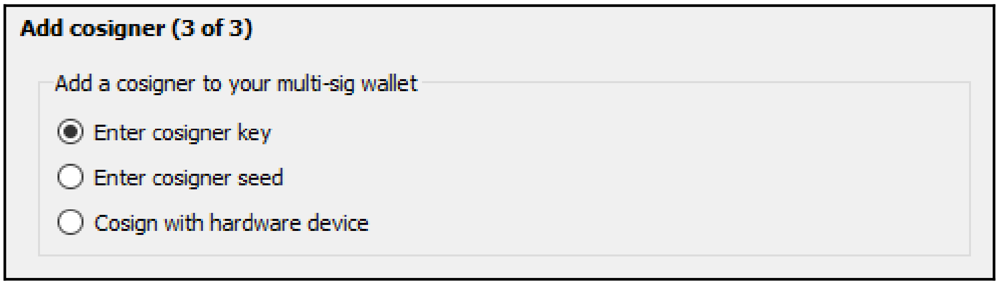
Choose the type of addresses in your wallet.

● legacy multisig (p2sh)
○ p2sh-segwit multisig (p2wsh-p2sh)
○ native segwit multisig (p2wsh)

You can override the suggested derivation path. If you are not sure what this is, leave this field unchanged.

m/45'/0'/2'/0

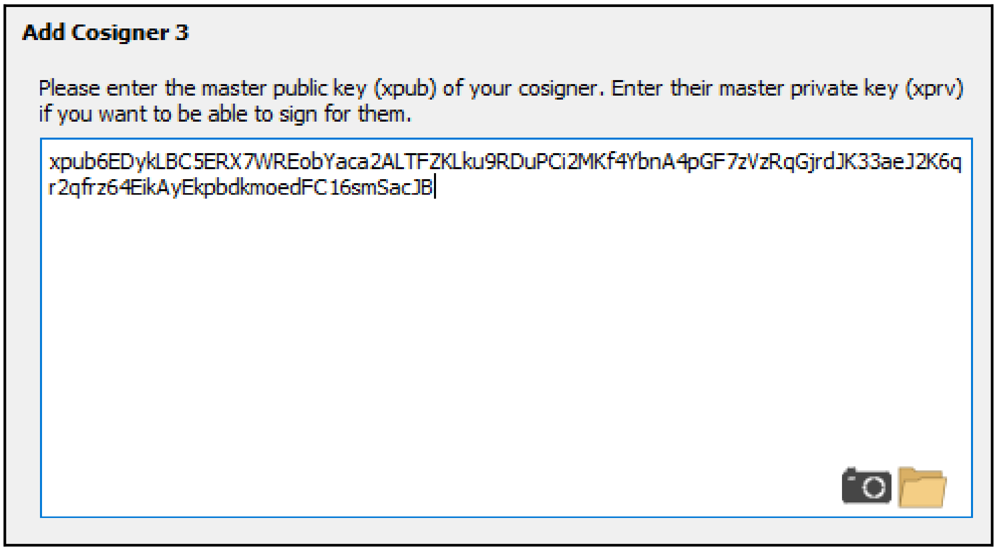
And lastly, cosigner 3 for the unchained key



Add cosigner (3 of 3)

Add a cosigner to your multi-sig wallet

● Enter cosigner key
○ Enter cosigner seed
○ Cosign with hardware device

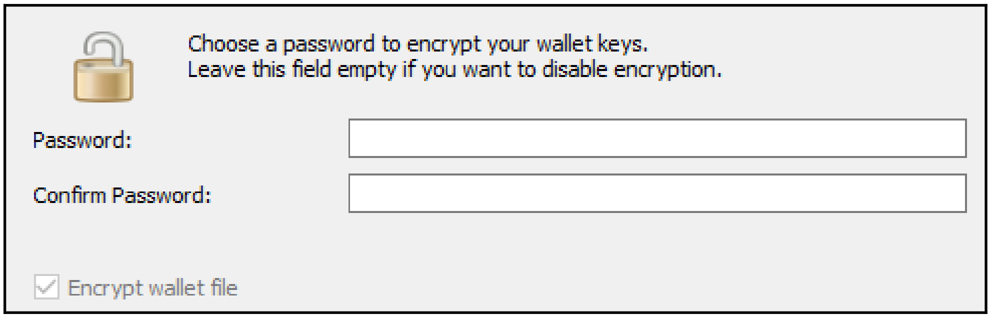


Add Cosigner 3

Please enter the master public key (xpub) of your cosigner. Enter their master private key (xprv) if you want to be able to sign for them.

xpub6EDykLBC5ERX7WREobYaca2ALTFZKLku9RDuPCi2MKf4YbnA4pGF7zVzRqGjrdJK33aeJ2K6qr2qfrz64EikAyEkpbdkmoedFC16smSacJB

Optionally specify a password

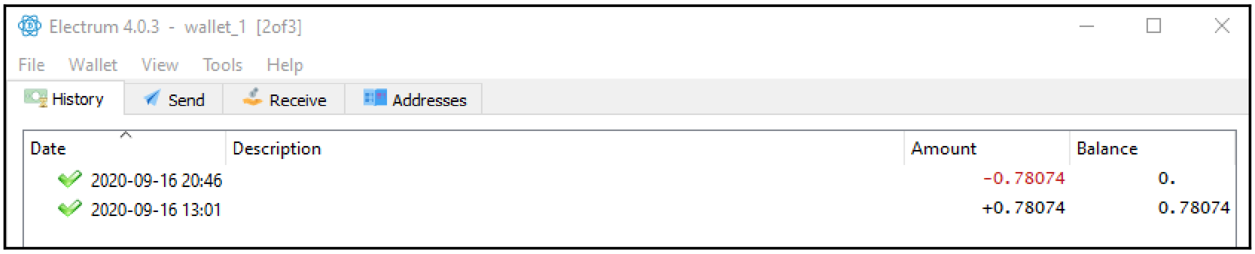


Choose a password to encrypt your wallet keys.
Leave this field empty if you want to disable encryption.

Password:	
Confirm Password:	

☑ Encrypt wallet file

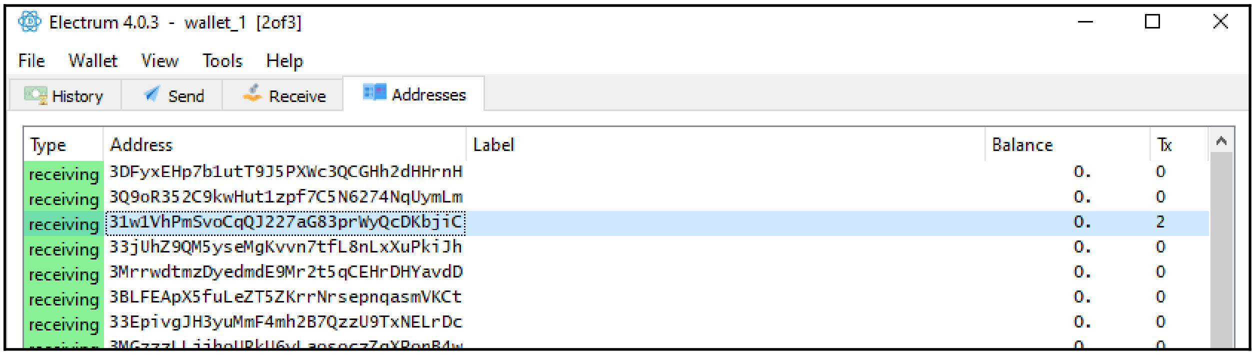
The History is displayed



Electrum 4.0.3 - wallet_1 [2of3]

Date	Description	Amount	Balance
✓ 2020-09-16 20:46		-0.78074	0.
✓ 2020-09-16 13:01		+0.78074	0.78074

And the addresses



Electrum 4.0.3 - wallet_1 [2of3]

Type	Address	Label	Balance	Tx
receiving	3DFyxEHp7b1utT9J5PXWc3QCGHh2dHHrnH		0.	0
receiving	3Q9oR352C9kwHut1zpf7C5N6274NqUymLm		0.	0
receiving	31w1VhPmSvoCqQJ227aG83prWyQcDKbjiC		0.	2
receiving	33jUhZ9QM5yseMgKvvn7tfL8nLxXuPkiJh		0.	0
receiving	3MrrwdtmzDyedmdE9Mr2t5qCEHrDHYavdD		0.	0
receiving	3BLFEApX5fuLeZT5ZKrrNrsepnqasmVKCt		0.	0
receiving	33EpivgJH3yuMmF4mh2B7QzzU9TxNELrDc		0.	0

From this I can verify that the address **31w1VhPmSvoCqQJ227aG83prWyQcDKbjiC** intended for Key Replacement is present in index 2. Funds swept to the address would be spendable by me. Two transactions are shown here as I've since swept funds out of this wallet when my testing was concluded.

Verification with Electrum Watch Wallet

A watch wallet is convenient to have for verifying addresses on a wallet, as well as reviewing transactions over time. For the aforementioned example, the following public keys are used for this wallet which you can use to test this on your own.

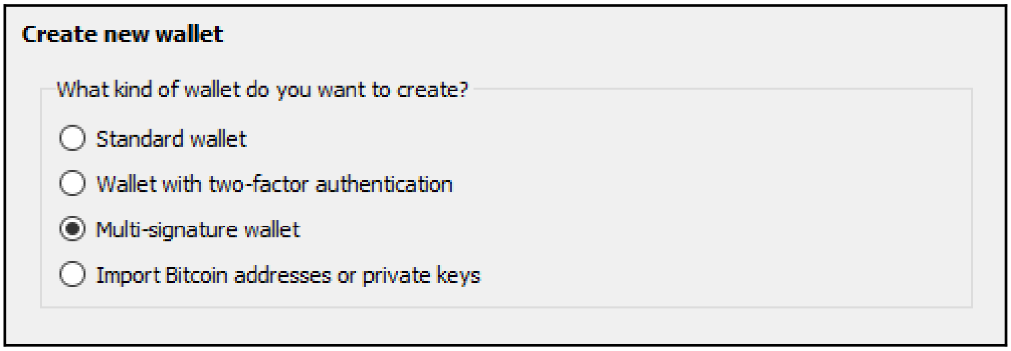
These public keys were derived by using both Caravan and Electrum in prior sections

xpub6FBFAVmiF1pgCYanH9GdbgHbiLkHSUq9c5KkY6c7mEk4o8757p8JsrdXo3zsy3uifqGEsBkp45C4jFYPM1X7k3bgsRmijrjz5edaFsxdodA

xpub6EkQHCE3w9F6qyCAZxW5vh87b969wiKUaB6NnYcjcsuLzPeEckNSffjHPFhP2hKM6jeAtRdoRiPGBJ3F72t6n4psx4gvEyhPsRitDo7yKkj

xpub6EDykLBC5ERX7WREobYaca2ALTFZKLku9RDuPCi2MKf4YbnA4pGF7zVzRqGjrdJK33aeJ2K6qr2qfrz64EikAyEkpbdkmoedFC16smSacJB

Create new multi-signature wallet

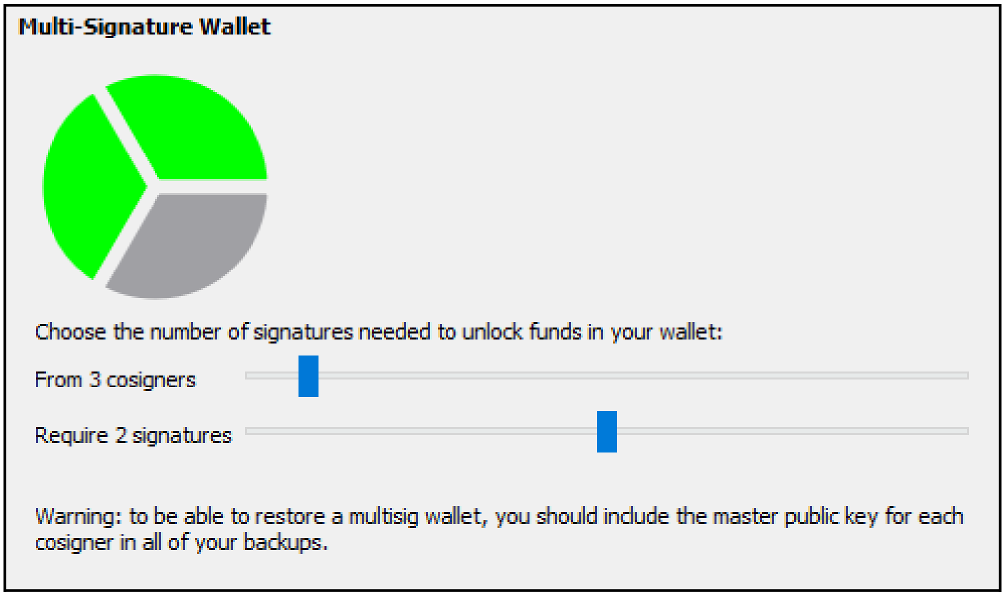


Create new wallet

What kind of wallet do you want to create?

○ Standard wallet
○ Wallet with two-factor authentication
● Multi-signature wallet
○ Import Bitcoin addresses or private keys

Specify 2 signatures required, of 3 cosigners



Multi-Signature Wallet

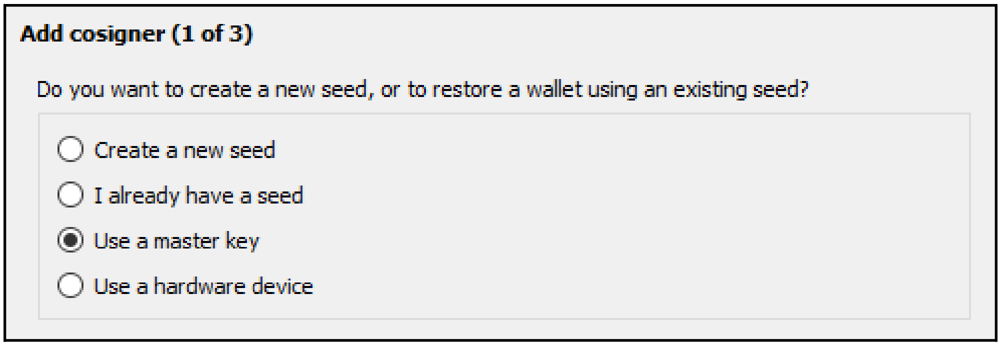
Choose the number of signatures needed to unlock funds in your wallet:

From 3 cosigners

Require 2 signatures

Warning: to be able to restore a multisig wallet, you should include the master public key for each cosigner in all of your backups.

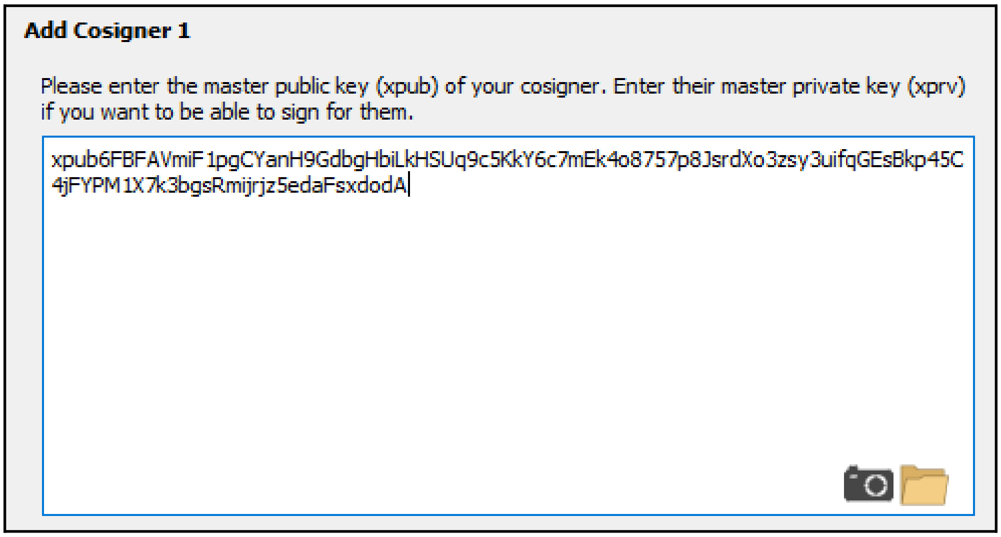
Specify the master key for cosigner 1



Add cosigner (1 of 3)

Do you want to create a new seed, or to restore a wallet using an existing seed?

○ Create a new seed
○ I already have a seed
● Use a master key
○ Use a hardware device

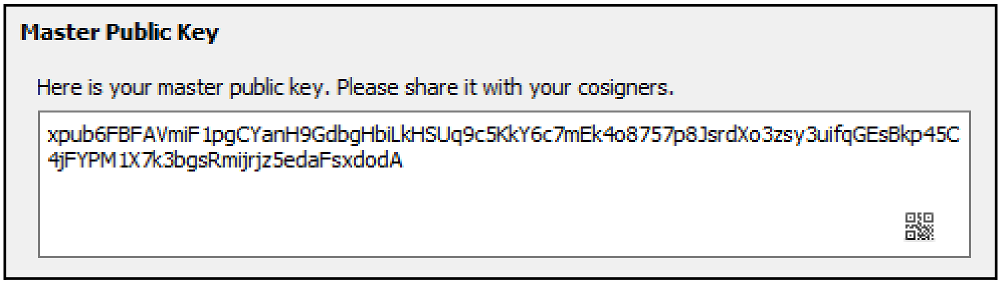


Add Cosigner 1

Please enter the master public key (xpub) of your cosigner. Enter their master private key (xprv) if you want to be able to sign for them.

```
xpub6FBFAVmiF1pgCYanH9GdbgHbiLkHSUq9c5KkY6c7mEk4o8757p8JsrdXo3zsy3uifqGEsBkp45C4jFYPM1X7k3bgsRmijrjz5edaFsxdodA
```

It will display it back to you

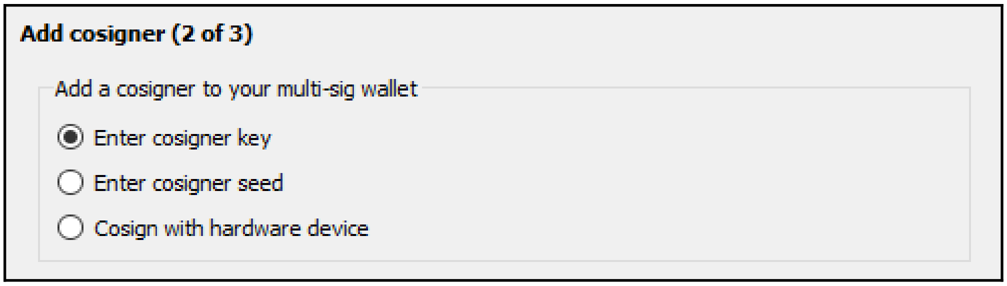


Master Public Key

Here is your master public key. Please share it with your cosigners.

```
xpub6FBFAVmiF1pgCYanH9GdbgHbiLkHSUq9c5KkY6c7mEk4o8757p8JsrdXo3zsy3uifqGEsBkp45C
4jFYPM1X7k3bgsRmijrjz5edaFsxdodA
```

For cosigner 2 and 3, repeat the process with those keys



Add cosigner (2 of 3)

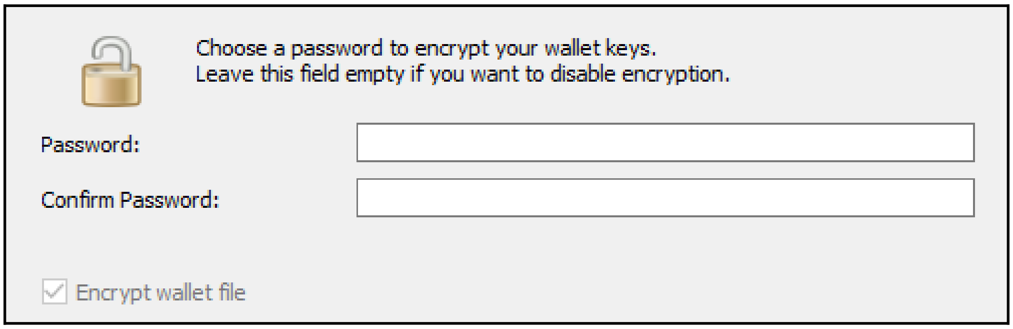
Add a cosigner to your multi-sig wallet

● Enter cosigner key

○ Enter cosigner seed

○ Cosign with hardware device

Optionally specify a password



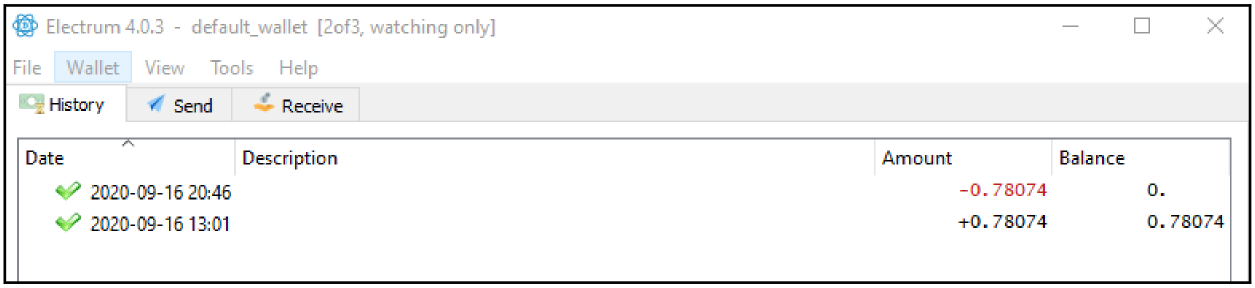
Choose a password to encrypt your wallet keys.
Leave this field empty if you want to disable encryption.

Password:

Confirm Password:

☑ Encrypt wallet file

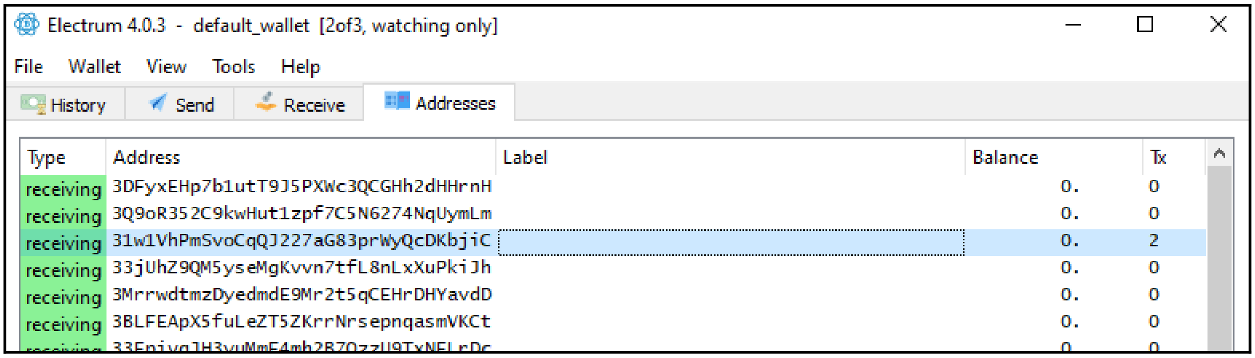
When this wallet loads, it will display its history as follows



Electrum 4.0.3 - default_wallet [2of3, watching only]

Date	Description	Amount	Balance
✓ 2020-09-16 20:46		-0.78074	0.
✓ 2020-09-16 13:01		+0.78074	0.78074

If the **Addresses** tab is not displayed, from the menubar, select **View**, and then **Show Addresses**. Switch to the Addresses tab. The address in index 2 is the one used for the Key Replacement process previously



Type	Address	Label	Balance	Tx
receiving	3DFyxEHp7b1utT9J5PXWc3QCGHh2dHHrnH		0.	0
receiving	3Q9oR352C9kwHut1zpf7C5N6274NqUymLm		0.	0
receiving	31w1VhPmSvoCqQJ227aG83prWyQcDKbjiC		0.	2
receiving	33jUhZ9QM5yseMgKvvn7tfL8nLxXuPkiJh		0.	0
receiving	3MrrwdtmzDyedmdE9Mr2t5qCEHrDHYavdD		0.	0
receiving	3BLFEApX5fuLeZT5ZKrrNrsepnqasmVKCt		0.	0
receiving	33EnivqJH3vuMmE4mh2B7QzzU9TxNELrDc		0.	0

Conclusion

Determining the intended key information for a wallet is essential to ascertaining whether you have access to an address when sweeping funds during a key replacement process in Unchained Capital.

Both Caravan and Electrum are useful wallet facilitators to be able to see addresses associated with a Multi-signature wallet.

When sending funds between wallets, Don’t Trust. Verify.