Twitter Thread by Ben Kaufman



Ben Kaufman
@_benkaufman

I've received many questions from people considering to set up a #Bitcoin multisig wallet but confused about the backup process, what should be backed up, and why.

■So here's a thread on backing up multisig wallets - what, why, and how.

The main caveat in a multisig wallet is that, while you need only a threshold of devices (ie. 2 of 3, 3 of 5, etc.) to sign a transaction, losing access to even a single device could potentially prevent you from being able to spend the funds - if you don't back up properly!

The reason is that (usually) in order to make a Bitcoin transaction, it is not enough to be able to sign it, but you also need to provide the "terms for spending", that is, the Script (code) that is used to lock the coins.

That's because the full script is usually not stored directly on the blockchain, but only as a hash (a unique identifier of it). Then on spending, you need to provide both the original script code which matches that hash, and satisfy any condition that the script may include.

In a single key wallet, this is trivial.

The script is composed of the public key of your wallet and a few standard commands - and satisfying it requires the signature for that key, so both the script and signature can be constructed by having access to that one private key.

With a multisig, this is not the case. The script for a multisig contains the list of public keys of all possible cosigners - which means it can be constructed only by knowing the public keys of ALL devices in the multisig, not just of the devices you are signing with.

To demonstrate, let's assume a 2 of 3 multisig. When you want to get an address of your multisig, your wallet software is taking the public keys of all 3 cosigners, lists them in a standard way inside a script, hashes it, then with this hash constructs the address.

When receiving bitcoin to that address, there is a "commitment" on the blockchain for all 3 public keys, out of which 2 signatures are needed. But the public keys themselves are not listed yet on the blockchain - you must provide them too when spending from the address.

Now assuming you have lost access to one private key (a device and its mnemonic backup), you won't be able to tell its public key anymore - so you won't be able to reconstruct the script of the address. So even though you have enough signatures, you won't be able to access funds.

The solution - which is the extra step of a multisig backup - is to also keep a backup of the list of public keys (xpubs) of all cosigners, from which you will be able to recreate the script.

Since this list of public keys does not contain signing (private) keys, there is no risk of losing funds even if an attacker manages to access it, but this would be a privacy concern since the attacker will then be able to see the balance on the multisig wallet.

That's why it is not recommended to leave this file on some cloud backup or some other insecure option.

It's recommended that you keep it with each one of the cosigners/ seed backups so that you can recover regardless of the combination of signers you eventually use.

Besides the list of public keys, it is also necessary to know the script type used (whatever native SegWit, nested SegWit, or legacy), and the threshold chosen in order to construct the multisig script. These are possible to know by guessing (brute force), but better keep it too.

The last piece which you should include is the derivation paths used for each device. Each derivation path tells your wallet how to turn each master public key, into a different individual "disposable" public key used to generate every new address.

This data of derivation paths is standardized so that all wallets should use the same derivation path by default, but it is best to keep that around too especially if you use a unique one for some reason.

This might sound like a lot of components, but to simplify usage, there’s a standard for storing and recovering such information, which is called Output Descriptors.

https://t.co/WAPJHlXtbK

Just as an example of what the final backup might look like, here are a few pics of a PDF backup generated from @SpecterWallet

This contains the individual devices keys list, as well as everything in an output descriptor format (and a bit of extra data like wallet name etc.)

Specter Backup File

(keep this information with each of your key backups)

My Multisig Wallet

2 of 3 multisig (sorted, native segwit)



Scan this QR code with Specter-Desktop
or any other compatible wallet.

Cosigner 1

Device type: Bitcoincore

Master fingerprint: 9e0f0f93

Derivation path: m/48h/0h/0h/2h

Master public key: xpub6E9dpcqt4igzh2oH4zwSXhXojJ61wXojKXs3
7UxyBfARNbf78iWgaEbzqvK899pTBbRwLuGC9xBx
USrE2PLazCURkG5zU8rHb9XhuqogEw6

Cosigner 2

Device type: Bitcoincore

Master fingerprint: 425cd4dd

Derivation path: m/48h/0h/0h/2h

Master public key: xpub6EsndNSyrW7mtZfGq7M5touZwqKv9NvNYJ5o
s5jf7Uo1BocRGo6t2oCtq2cpbomfvHbtJoCWnKMs
k2CvjK2hR9jv8gnxvtHLXPRiJ1ATtkQ

You could save this by simply printing the PDF, storing it on an SD card, or just keep it encrypted on a cloud backup. What you need to remember is:
1. You need this information to recover the wallet.
2. Anyone with access to this information can see your wallet history.

So to simply sum it up, the additional piece you should backup when using a multisig wallet is the output descriptor - a list of the public keys of the cosigners with some data on the wallet.

This information is possible to derive only with access to all devices in your multisig and is necessary to spend from it.
So to remove the single point of failure of losing access to one cosigner, you should keep this backup along with each of your devices. fin/