PV204 Security technologies

Bitcoin II. – Bitcoin hardware wallets, multisig, (CoinJoin, PayJoin)

Petr Švenda *svenda@fi.muni.cz* *@rngsec*
Centre for Research on Cryptography and Security, Masaryk University

CROCS
Centre for Research on Cryptography and Security

Three main goals of bitcoin end-user security

1. Safety (against you loosing access to your funds)
   - Backup of mnemonic phrase (paper, steel plate, Shamir)
   - Native bitcoin script multisignature (2-of-3)
2. Security (against attacker trying to steal your funds)
   - Hardware wallet to generate seed and manage secret keys
   - Secure confirmation of transaction details (address, value, fee) on display
3. Privacy (against third party observing your actions)
   - Use your own fullnode
   - Practice labeling and coin control
   - Use Coinjoin mixing (Wallet Wasabi 2, Trezor Suite)

Masterplan for this seminar

1. OP_RETURN use for TimeStamping (short example)
2. Hardware wallet use (Sparrow + ColdCard)
3. Multisignature wallet use (Sparrow wallet as coordinator)

4. (If interested)
   - Recovery of wallet into different client (Sparrow → Electrum)
   - CoinJoin mixing (Whirlpool)

STORING ARBITRARY DATA ON BLOCKCHAIN

Electrum Testnet 4.1.5 - default_wallet [standard]

File Wallet View Tools Help

History | Send | Receive | Channels | Coins | Console

Pay to: OP_RETURN 707632303420697320 6b69636b696e67

Description: test opreturn

Amount: 0 BTC

Max

Clear | Save | Pay...

Confirm Transaction

Amount to be sent: 0. BTC

Advanced

Mining fee: 0.0000014 BTC

Fee rate: ETA

Warning: The fee for this transaction seems unusually high. (100.00% of amount)

Cancel | Send

Inputs & Outputs

Details

tb1qn6nnzj99dccydlfvv6f0s9keyk… 6ate5ze3 0.00100000 tBTC

OP_RETURN pv204 is kicking 0.00000000 tBTC

tb1q015kvpeh04wvtunpw6v24eh5f0… x3le6g3t 0.00099860 tBTC

0.00099860 tBTC

Storing arbitrary data on blockchain

Getty | Iryna Yeroshko

- Your data has to be stored by all full-nodes forever!
  - Writing nonsense messages / spam etc. is like carving into tree bark
  - Always ask yourself what is the value of such storage and if it is fair to others!
- Bitcoin P2P protocol has some additional limitation rules about transaction data allowing to propagate
  - More stricter rules than Bitcoin consensus rules (what can be in block mined)
  - E.g., 4MB NFT picture in segwit-discounted data is valid transaction and is accepted once mined, yet nodes will not propagate such transaction into mempools
  - These P2P rules are indication what network considers “good” behavior

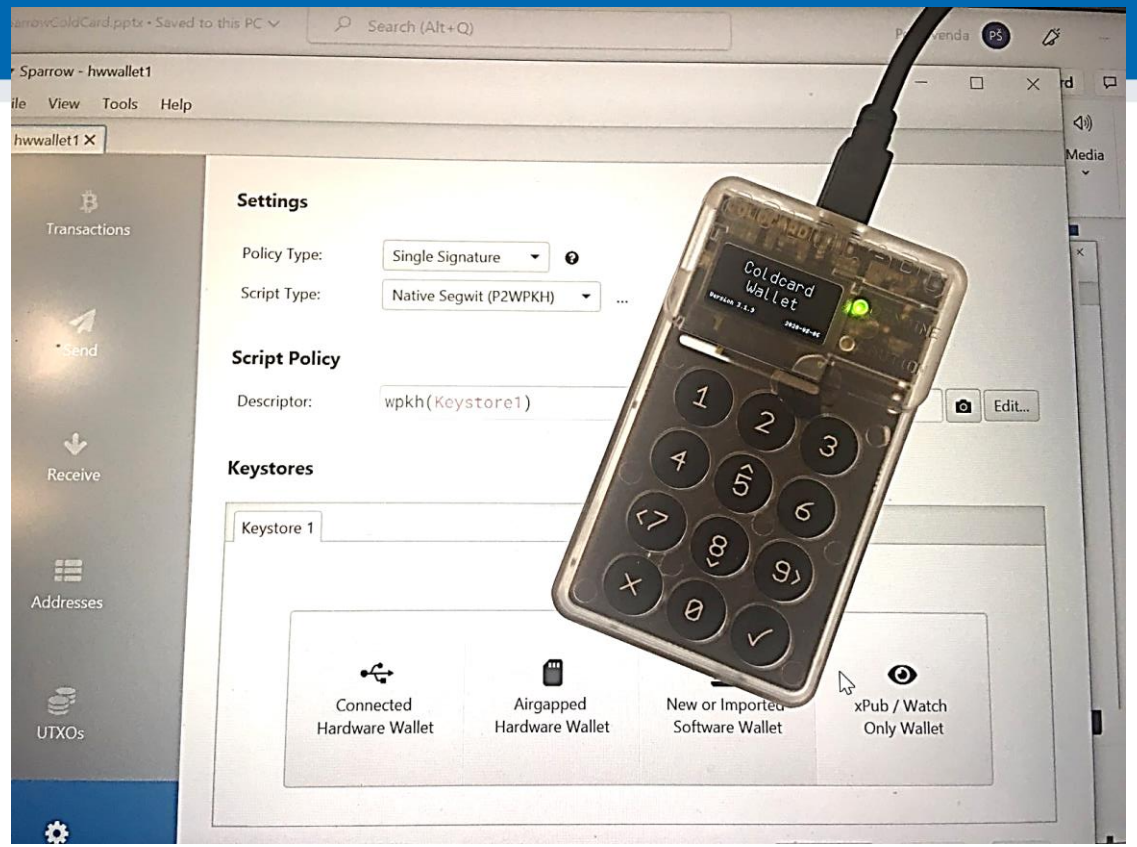
OpenTimestamps protocol (https://opentimestamps.org/)

- Prove that document existed at date X (at latest)
- Merkle tree of all submitted document hashes within given period committed to Bitcoin blockchain (OP_RETURN, every ~10 hours)
  - https://petertodd.org/2016/opentimestamps-announcement
- Currently free to use (only one OP_RETURN embed for all files)
  - Client needs to remember own Merkle tree path + file => *.ots file

```
$ pip3 install opentimestamps-client
$ ots stamp secret.txt

$ ots info secret.txt.ots
$ ots verify secret.txt.ots
Assuming target filename is 'secret.txt'
Calendar https://alice.btc.calendar.opentimestamps.org: Pending confirmation in Bitcoin blockchain
```

https://github.com/opentimestamps/opentimestamps-c
https://mempool.space/tx/5cfb6d1eee37cfd3dc51d01f
$ ots verify secret.txt.ots
Assuming target filename is 'secret.txt'
Calendar https://alice.btc.calendar.opentimestamps.or
Calendar https://finney.calendar.eternitywall.com: Tim



SINGLE-SIGNATURE HARDWARE WALLET

Steps today

1. Generate secret seed in hardware wallet
2. Backup it (mnemonics on paper)
3. Connect wallet with PC software wallet (receive and spend wallet)
4. Erase wallet, test recovery of mnemonics
5. Create receive-only mobile wallet (stack additional sats)

Before we start…

- You have one hardware wallet per person, erase it afterwards
- Do not walk around with phones, cover mnemonic words by hand
- Hide your mnemonic words against any exposure

- **VERY IMPORTANT!!!**
  - ColdCard is real hardware wallet (~$100)
  - “Bricked” if correct PIN is forgotten unknown (no “reset” button)
  - **For this tutorial**, **always set PIN to 12 34 !!!**
    - (for **real use with ColdCard you bought**, always set to PIN only you know)

Steps of hardware wallet usage

1. Prepare ColdCard hardware, generate and backup new wallet
   – No computer required, everything happens on ColdCard device
2. Prepare Sparrow on PC with private keys stored on ColdCard
   – Public information from ColdCard wallet is exported to Sparrow
3. Receive tBTC to ColdCard wallet (via Sparrow)
   – No ColdCard required, only public keys are required
4. Erase ColdCard, perform recovery
   – Real verification that your backup works
5. Send tBTC from ColdCard wallet (via Sparrow)
   – Private keys on ColdCard required, checks and signing happens on ColdCard

1. PREPARE COLDCARD HARDWARE, GENERATE AND BACKUP NEW WALLET

Login into your ColdCard device

1. Connect via USB cable
2. Enter PIN Prefix: **USE `12`!!!**, press OK
   - Write on paper shown words (what they are for?)
3. Enter rest of PIN: **USE `34`!!!**, press OK
4. Generate New Wallet, write on paper 24 words, verify 24 words
5. State: ‘Ready to Sign’ option shall be displayed

(later after wallet generation)

- Move to Advanced → Danger Zone → Testnet mode
- Change to ‘Testnet: BTC’

We will work with testnet BTC => need to tell wallet to use testnet addresses

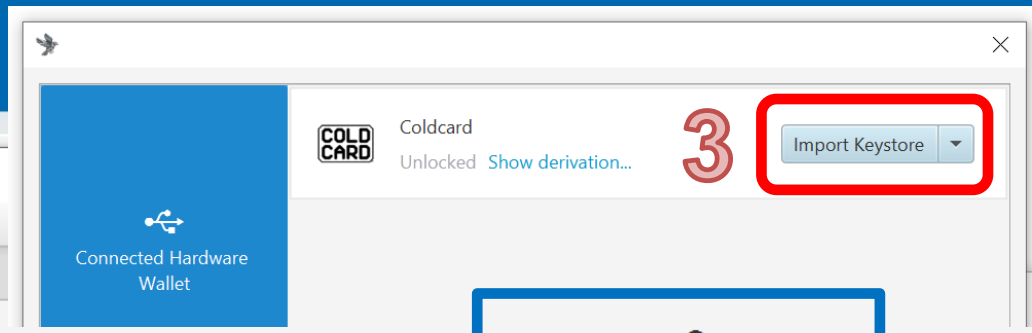

1
Coinkite
293761048487
DONT TRUST.
VERIFY.
DONT TRUST.
VERIFY.
2
3
accept terms and
continue.
5
Choose PIN Code
Advanced
Bag Number
Help
THERE IS
ABSOLUTELY NO
WAY TO RECOVER A
FORGOTTEN PIN!
Write it down.
There is
ABSOLUTELY NO
WAY to 'reset
the PIN' or
'factory reset'
6
Enter PIN Prefix
7
Enter rest of PIN
8
New Wallet
Import Existing
Help
Advanced
Generating...
9
Ready To Sign
Advanced

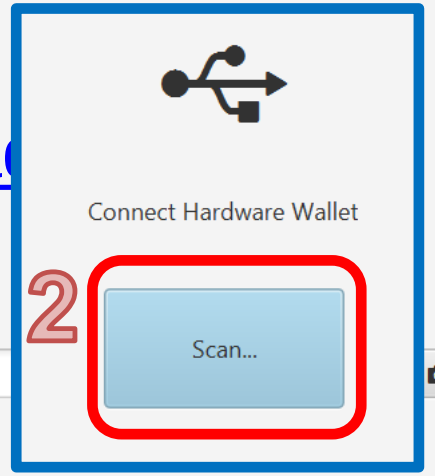
2. PREPARE SPARROW ON PC WITH PRIVATE KEYS STORED ON COLDCARD

Create wallet

- Sparrow Wallet https://www.sparrowwallet.com/downloa
  - Linux: Tools → Install udev rules
- `sparrow -n testnet`
- Prepare ColdCard wallet (see before)
- Connect ColdCard wallet via USB cable
- File → New wallet

1. Connected Hardware Wallet
2. Scan
3. Import Keystore

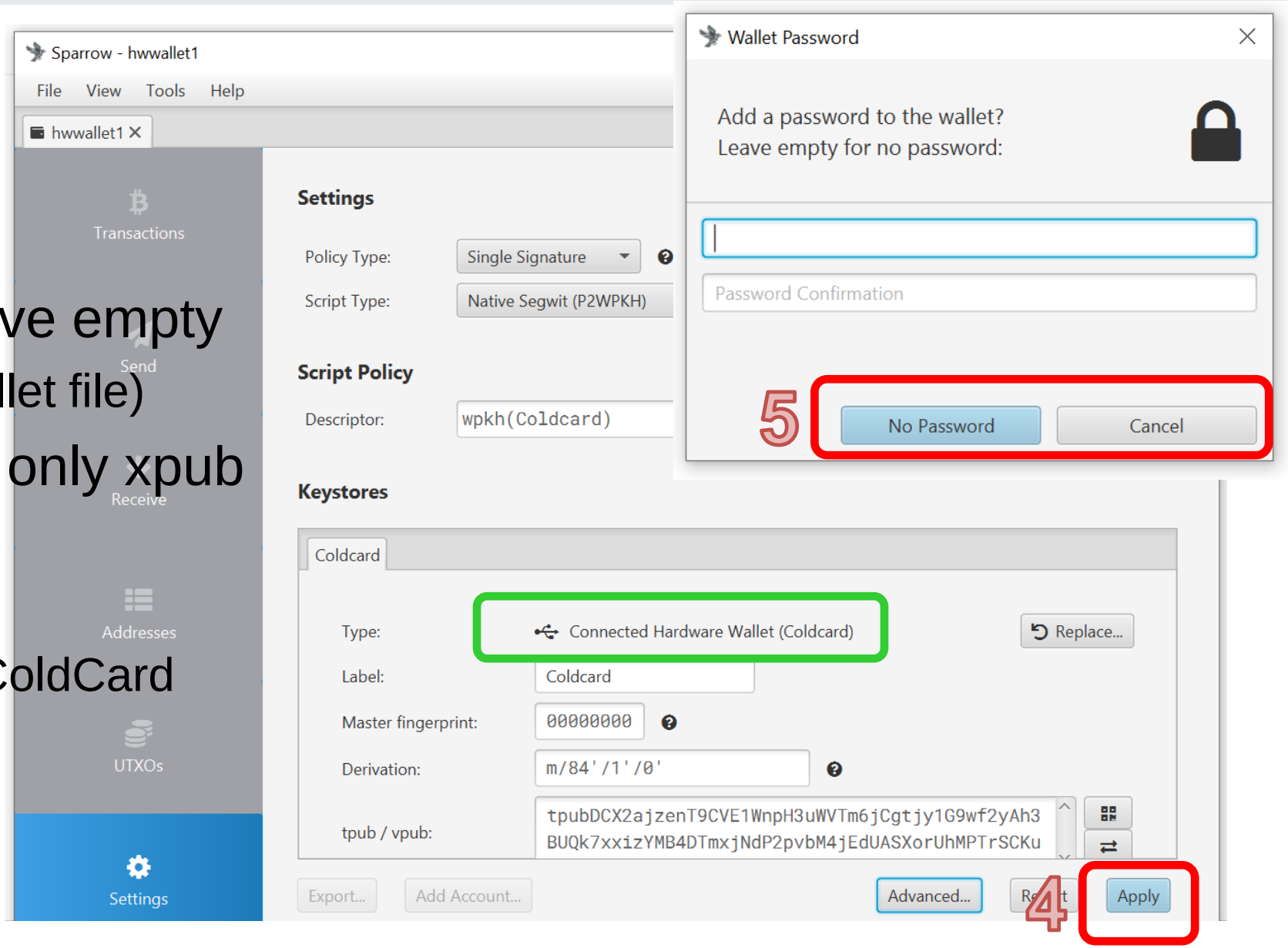




Create wallet

4. Apply
5. Set password or leave empty
   – (encryption of local wallet file)

- Local wallet contains only xpub
  – *.mv.db file
  – File→Open wallet
  – Private key(s) are on ColdCard



3. RECEIVE TBTC TO COLDCARD WALLET (VIA SPARROW)

Task: send some tBTC from software to hardware wallet

- Exactly same procedure as for sending between software-only wallets
  - Hardware wallet's private key is not required for receiving
- Person with ColdCard shall receive one transaction from every other person (PC1 and CC)
- Obtain his/her receive address
  - Via messenger: CC → Receive tab → Copy address → send via Signal → PC1
  - Via QR: CC → Receive tab ; PC1 → Send → camera icon → scan address QR
- Enter some sats into Amount box
  - Observe visualized transaction below (more inputs may be added)

PC1

Sparrow - wallet1

File View Tools Help

wallet1 | to coldcard wallet

Transactions
Send
Receive
Addresses
UTXOs
Settings

Send

Pay to: tb1q7a777xe6jppnk2az43qqq5r856gm0dhcwl9y4g

Label: to coldcard wallet

Amount: 354,290 sats $ 72.13 Max

+ Add

Fee

Target Blocks | Mempool Size

Range: 1 2 4 8 16 32 64 128 256 512 1024

Rate: 1.01 sats/vB High Priority

Fee: 141 sats $ 0.03

0 kvB 16:04 17:30

internal send (cha...

Add Mix Partner?

Transaction

to coldcar...

tb1quj0t...

Fee

Optimize: Efficiency | Privacy Analysis... Clear Create Transaction »

CC

Sparrow - hwwallet1

File View Tools Help

hwwallet1

Transactions
Send
Receive
Addresses
UTXOs
Settings

Receive

Address: tb1q7a777xe6jppnk

Label: from swwallet wallet1

Derivation: m/84'/1'/0'/0/0

Last Used: Never

Required ScriptPubKey

Script:

Output Descriptor

Descriptor: wpkh(034bff5cbec46af1833f5e222bc66006f fcdbb94e222e67f5160c35d0cddb4df6b)

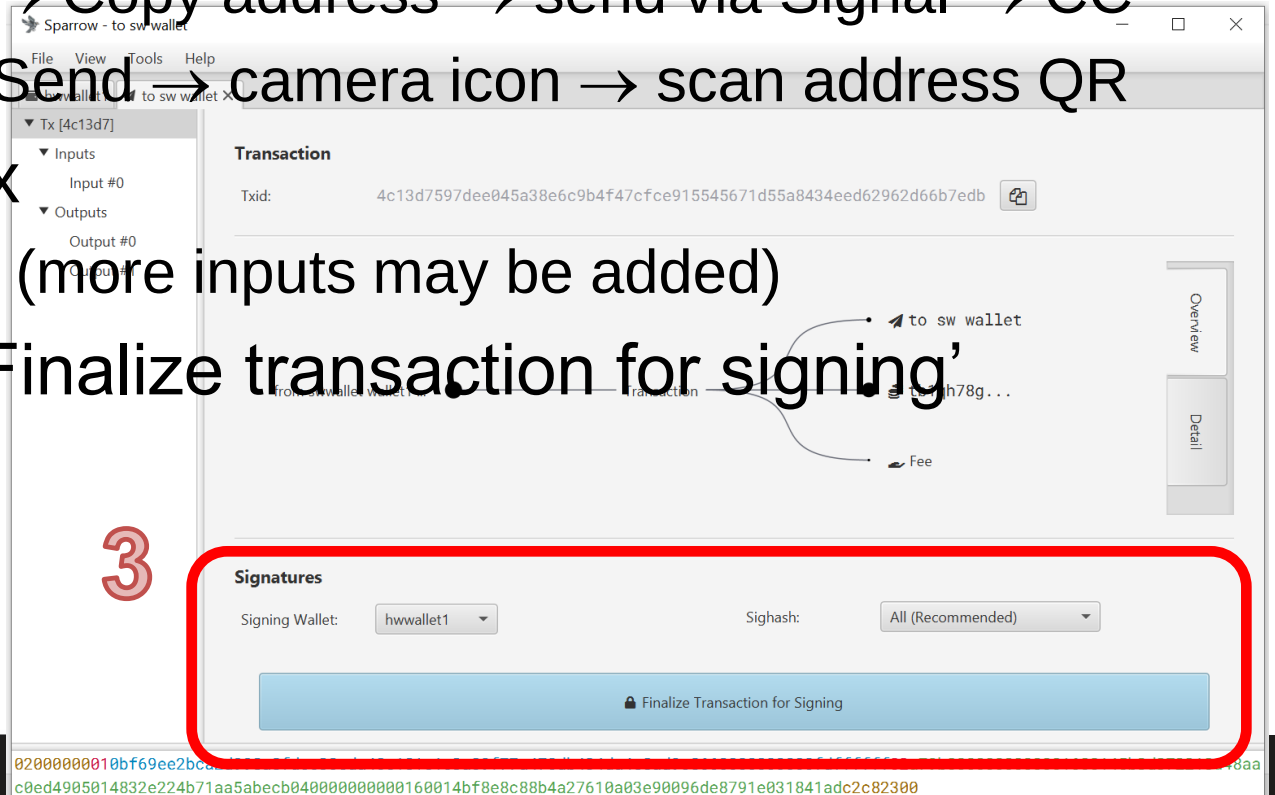
Display Address | Get Next Address

4. SEND TBTC FROM COLDCARD WALLET (VIA SPARROW)

Task: send some tBTC from hardware to software wallet

- Person with ColdCard sends to at least one other person (CC → PC1)

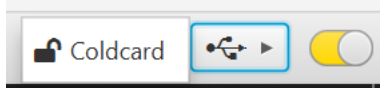
1. Obtain PC1's receive address
   - Via messenger: PC1 → Receive tab → Copy address → send via Signal → CC
   - Via QR: PC1 → Receive tab ; CC → Send → camera icon → scan address QR
2. Enter some sats into Amount box
   - Observe visualized transaction below (more inputs may be added)
3. Click 'Create transaction', click 'Finalize transaction for signing'



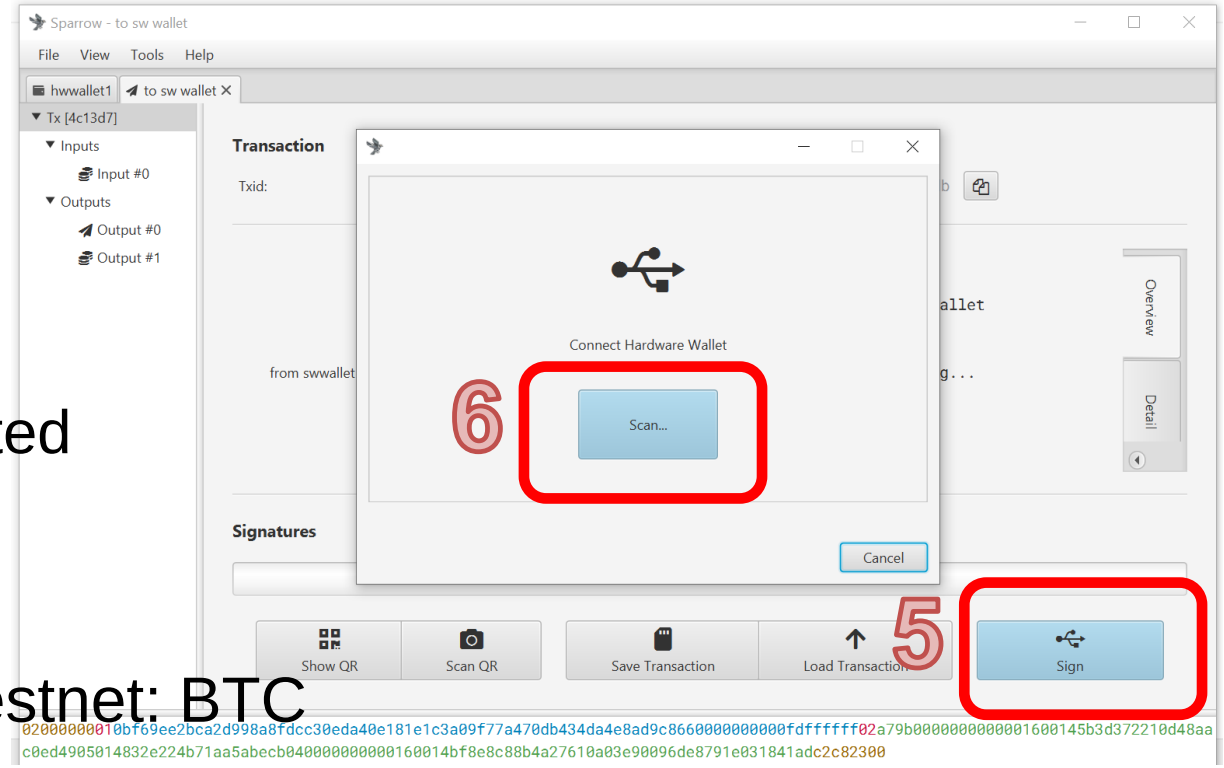
Send some tBTC from hardware to software wallet

4

4. Connect ColdCard via USB
   - Enter PIN Prefix, press OK
   - Enter rest of PIN => ‘Ready To Sign’
5. Click ‘Sign’ in Sparrow
6. Click ‘Scan’ in Sparrow

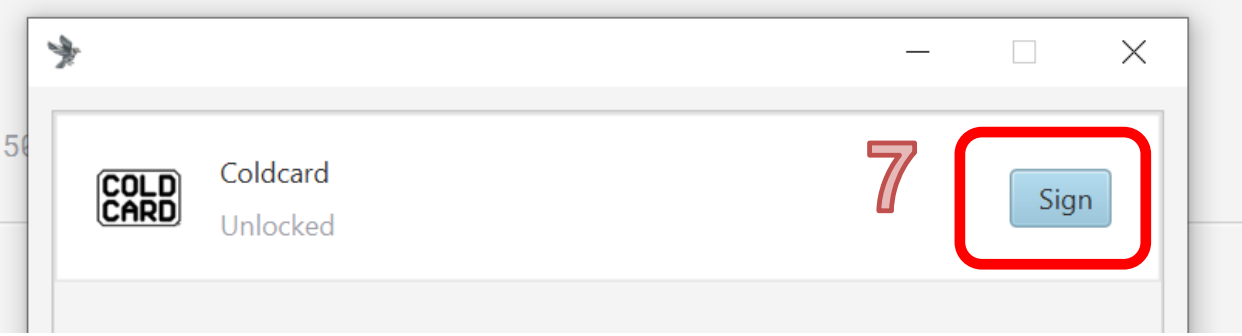
- Note: 

  - Look for icon after is ColdCard connected
  - If icon is not visible, try to reconnect
  - If icon is visible but Scan fails, check
    - ColdCard:Settings→Blockchain→Testnet: BTC



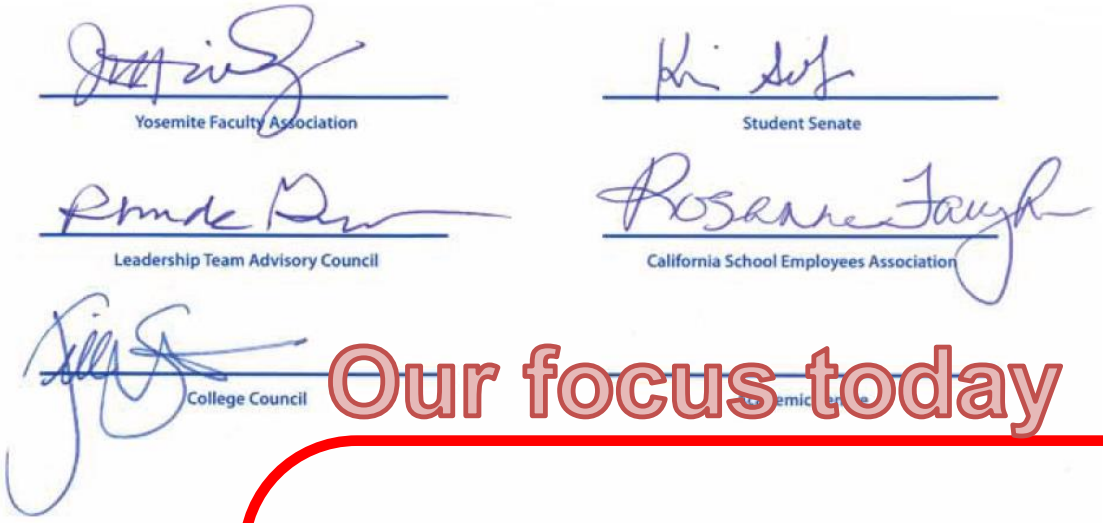
Send some tBTC from hardware to software wallet

7. Select ColdCard and click ‘Sign’



8. Verify on ColdCard’s screen (compare with your Sparrow)
   – Amount, address, fee, changeback, changeback address
   – Press OK if match
9. Click ‘Broadcast Transaction’
   – Transaction is now complete, broadcast to network

1. THRESHOLD SECRET SHARING
2. MULTISIGNATURES
3. MULTI-PARTY CRYPTO COMPUTATION



Analogically for decryption (single person decrypts, multiple people, k-of-n)

Our focus today

Single signature

Signature

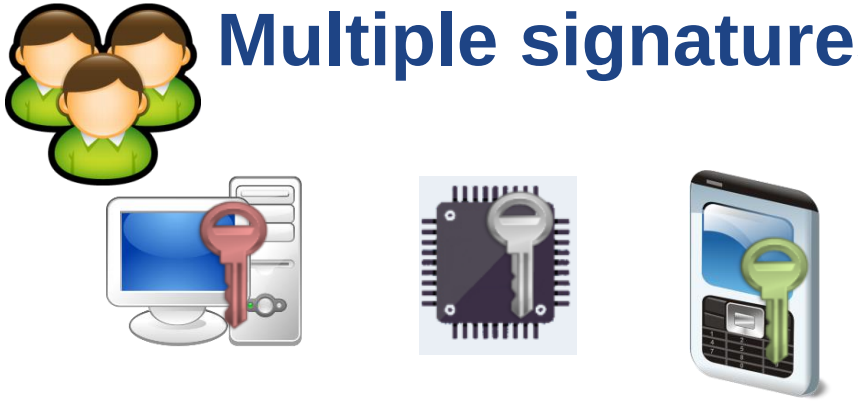
Shamir TSS

Share 1

Share 2

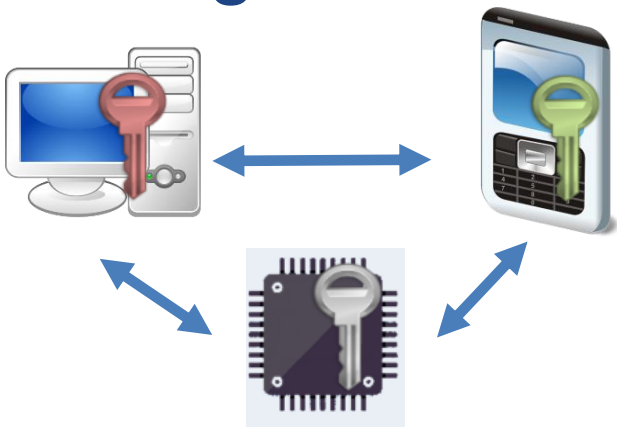
Share 3

Multiple signatures



Signature | Signature | Signature

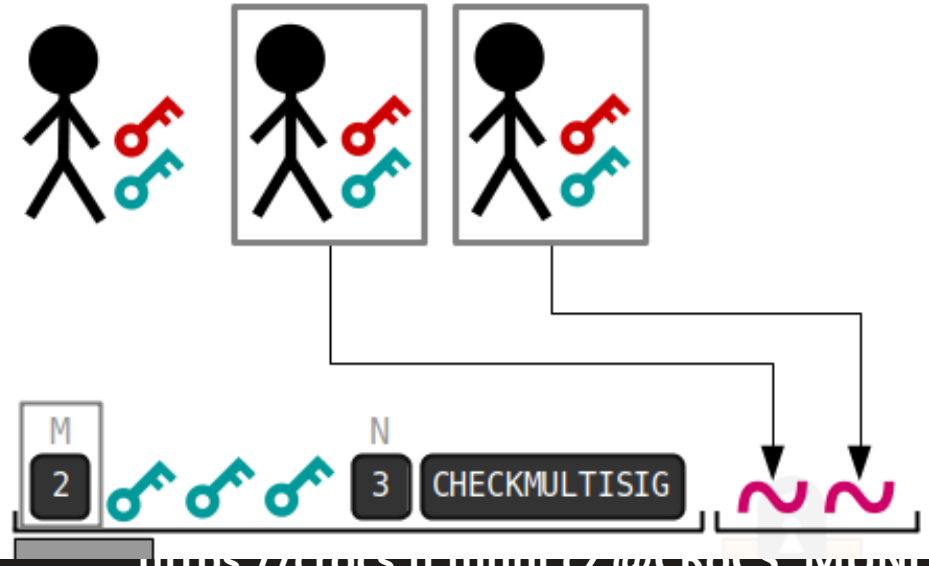
MPC signature



Signature

2. Multisignatures

- Lock script constructed to require multiple signatures (OP_CHECKMULTISIG)
  – transaction valid only if multiple signers provide signatures for unlock script
- n-out-of-n or m-out-of-n, https://en.bitcoin.it/wiki/Multisignature
- P2MS, P2MS wrapped in P2SH
  – https://learnmeabitcoin.com/technical/p2ms

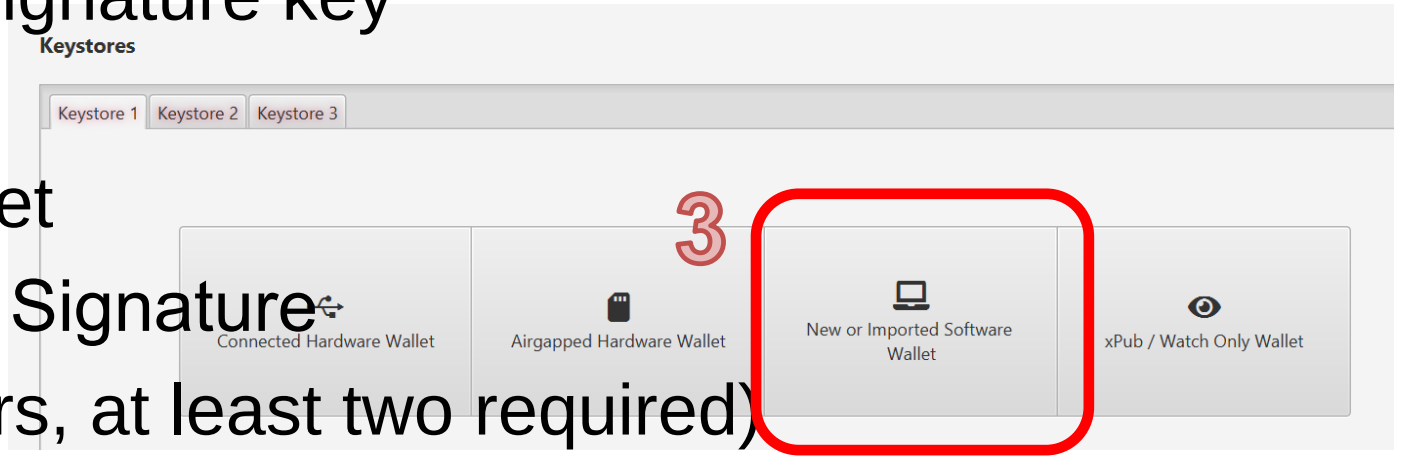


Task: Create multisignature wallet

- Form groups of three members
  - (can be also done with three Sparrow instances on the same if you test alone)
  - Make sure you can send short messages to each other (Signal/WhatsApp) or have camera read QR codes
- Run Sparrow wallet on testnet (-testnet)
- Quorum 2-out-of-3 will be used (3 members, 2 enough to authorize)
- Every participant will create one keystore with knowledge of private key(s) and then import remaining two xpubs (tpubs on testnet) for other two signers
- Some tBTC will be send to multisig wallet
- Cooperation of two members will be used to create new transaction

Create multisignature wallet I.

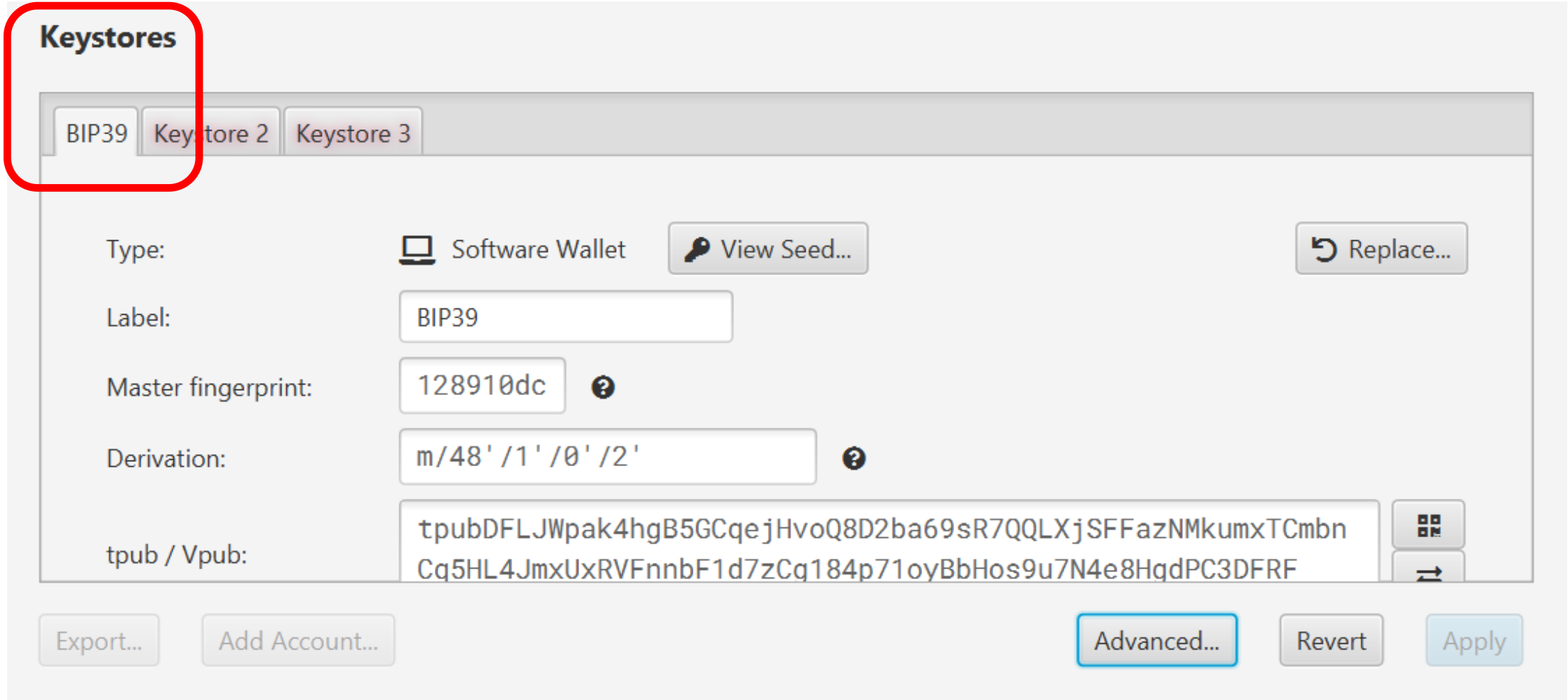
- Every participant creates one signature key
- File → New wallet
- New or Imported Software wallet



1. Change ‘Policy Type:’ to Multi Signature
2. Set M of N to 2/3 (three signers, at least two required)
3. Set Keystore 1 as ‘New or Imported Software wallet’
4. Setup Keystore 1 as before (singlesig wallet, 12 words, Import keystore)

Keystore 1 now created

4



Create multisignature wallet II.

- Insert xpubs/pubs for other two signers (your group members)

5. Transfer tpub from your Keystore 1 to other two members (Signal/QR code)
   - Paste received tpubs into Keystore 2 and 3 (select ‘xPub / Watch Only Wallet’)
6. Set Derivation same as for Keystore 1 (m/48'/1'/0'/2’)
   - For both Keystore 2 and Keystore 3
7. When all three keystores are filled, Apply button is enabled (click it)
8. Let one member to send some tBTC to multisig wallet
   - Receive, send from singlesig wallet (do not send all funds)
   - All members shall see new tBTC coming to multisig wallet

Keystores

BIP39 | Keystore 2 | Keystore 3

Type: Software Wallet — View Seed...

Label: BIP39

Master fingerprint: 128910dc

Derivation: m/48'/1'/0'/2'

tpub / Vpub: tpubDFLJWpak4hgB5GCqejHvoQ8D2b
Cq5HL4JmxUxRVFnnbF1d7zCg184p71

Export... | Add Account...

Keystores

BIP39 | Keystore 2 | Keystore 3

Type: Watch Only Wallet — Import...

Label: Keystore 2

Master fingerprint: 00000000

Derivation: m/48'/1'/0'/2'

5

tpub / Vpub: tpubDFLJWpak4hgB5GCqejHvoQ8D2ba69sR7QQLXjSFFazNMkumxTCmbnCq5
HL4JmxUxRVFnnbF1d7zCg184p71oyBbHos9u7N4e8HgdPC3DFRF

Keystores

BIP39 | Keystore 2 | Keystore 3

Type: Watch Only Wallet — Import...

Label: Keystore 3

Master fingerprint: 00000000

6

m/48'/1'/0'/2'

tpubDEbH1xDZn981WBe736Bc2Ps2Hp8941f69ckMptWL9F
1cJEgir7Hpknxnzr63iGd1Zg1Tm8K1Km7Fv3xDeEeZPduJ

rivation: m/48'/1'/0'/2'

ub / Vpub: tpubDED2dnYnViJHDyoTpexKab5iVxKS9uH2iE5HHZvfdFVPbHvzDxHXL2c2
iQJCwN6R4sqBrbhmKvdnDPMQk8mEvRGwMSvMqPNiVnNmMPc3u5w

7

Add Account... | Advanced... | Revert | Apply

Singlesig wallet

Multisig wallet

8b

Transactions

Send

Receive

Addresses

UTXOs

Settings

Send

Pay to: tb1qf8h27nlel63rjly5yzrs5xnq9l07t0plqayznnzzt08xweqzc8gqah6

Label: to multisig

Amount: 1,000,042 sats $ 294.28 Max

+ Add

Fee

Target Blocks Mempool

Range: 1 2 4 8 16 32 64 128 256 512 1024

Rate: 1.00 sats/vB CPFP High Priority

Fee: 153 sats $ 0.05

0 kvB

09:29

to europen base ... Transaction

to multi

tb1qkte

8c

Optimize: Efficiency Privacy Analysis... Clear Create Transactio

Sparrow - multisig1

File View Tool Help

multisig1

8a

Transactions

Send

Receive

Addresses

UTXOs

Settings

Receive

Address: tb1qf8h27nlel63

Label:

Derivation: m/48'/1'/0'/2'/0/0

Last Used: Never

Required ScriptPubKey

Script: OP_0 <wsh>

Output Descriptor

Descriptor: wsh(sortedmulti(
d6e954ba618f06b4
ac035c82b91f3c9d

STATE: MULTISIG WALLET IS CREATED, SOME FUNDS ARE AVAILABLE CAN SEND TRANSACTION 2 OF 3

Send transaction from multisig wallet (to singlesig wallet)

- Open any singlesig wallet (e.g., one of your group members)
  - Generate new receive address Receive→Address:

Person 1

1. PC1: One member goes to his/her multisig wallet and starts transaction
   - Send → Pay To: paste singlesig address, set label and amount
2. PC1: Create Transaction → Finalize Transaction for Signing → Sign
   - Partially Signed Bitcoin Transaction (PSBT) is now created
3. PC1→PC2: Transfer to one of group members (PC2)

Person 2,3…

- Option a): Show QR → variable QR displayed, scan from another machine
  4. PC2: File → Open Transaction → From QR…
- Option b): Save Transaction → file *.psbt, load file from second machine
  4. PC2: File → Open Transaction → File…

1

multisig1

Transactions
Send
Receive
Addresses
UTXOs
Settings

Send

Pay to: tb1ql29w297c63xwxcq9m8y2xrtetpcfx5vqq9t5ln

Label: to singlesig wallet

Amount: 383,924 sats $ 113.18 Max

+ Add

Fee

Target Blocks | Mempool Size

Range: 1 2 4 8 16 32 64 128 256 512 1024

Rate: 1.01 sats/vB High Priority

Fee: 190 sats $ 0.06

0 kvB
09:23

a6061120..:0 — Transaction → to singles... / tb1qlede... / Fee

Optimize: Efficiency | Privacy | Analysis...

Clear | Create Transaction »

4

Sparrow - multisg2

File View Tools Help

New Wallet Ctrl+N
Open Wallet... Ctrl+O
Open Transaction
Save Transaction... Ctrl+S
Import Wallet...
Export Wallet...
Close Tab Ctrl+W
Quit Ctrl+Q

4b

File... Ctrl+F
From QR... Ctrl+U

4a

Receive

Fee

Range:

multisig1 | to singlesig wallet

Tx [6c26b7]
- Inputs
  - Input #0
- Outputs
  - Output #0
  - Output #1

Transaction

Txid: 6c26b792071103fec60e278dff31355bf9134d4839bc8796bb82a720e98e610c

a6061120..:0 — Transaction → tb1qlede... / to singles... / Fee

Overview | Detail

Signatures

3a multisig1 3b

Show QR | Scan QR | Save Transaction | Load Transaction | Sign

2

Lister - [h:\to singlesig wallet.psbt]

File Edit Options Encoding Help

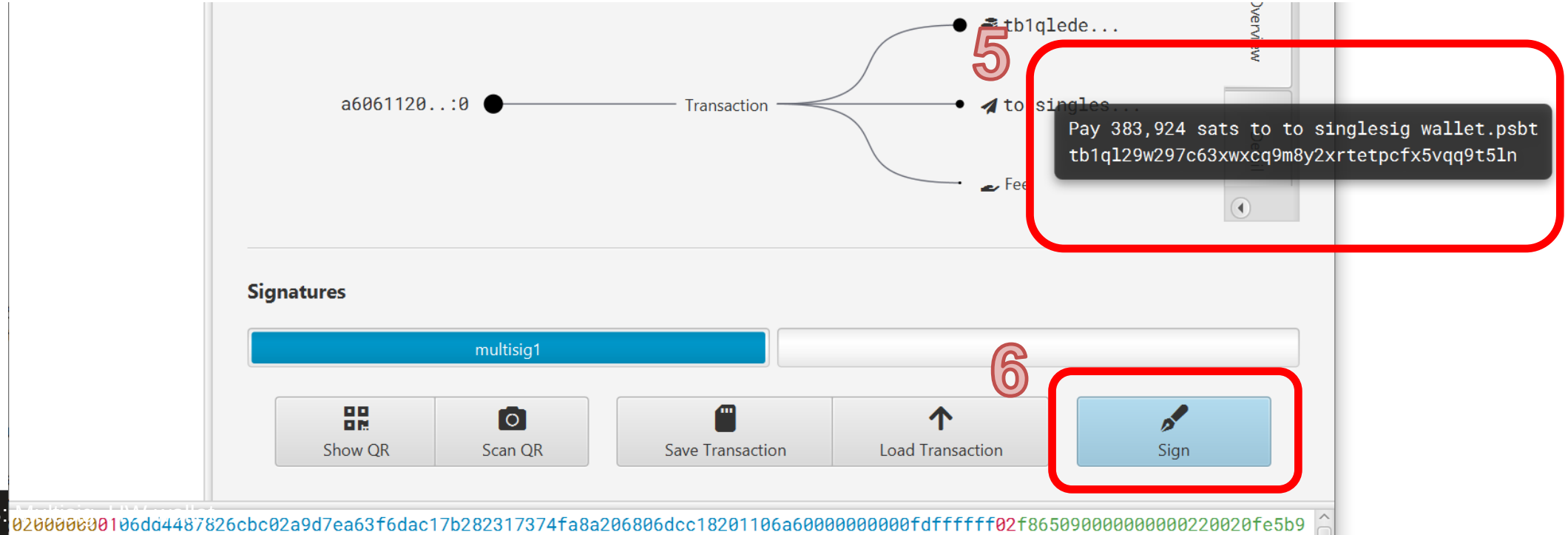
psbtÿ..}.....ÝD■■1¼.©×êcöÚÁ{(#.7O¨ç
■÷Ó'yF■».5.R¬..-«ÿtk.â×´Û.....ú■
¢%.tËx®å.'■êÈ..[.K■.|■D€˜E-çúm..`à.é
€...€...€O..5■Ï.).⅜ú€...¨■®²■..Ç■õó.
Ç■■î©l■>..ñ.■Ý.....0..€...€...€...€C
■■ä'■Zð.\;¨v■þ¨fàÉ{¥'à2..ú¶Ú■mðþKapí
Úù■Ø¨/òJSu■9'¶©TúuOZS9Çu¥±y.....ýÿÿÿ
gd.ÁÐ'C.....²òø:#ØÙ'/ÐA½Ê■U3@d!Ù
H)ÌB[Îgd.ÁÐ"..■-ñýûÂ.■ÌjG■■Rw(z..'■Ê
)■úÙC■²#. dJûé■#.=XH..òIÚ¤..º■Ãÿ.kÞz

02000000010 6dd4487826cbc02a9d7ea63f6dac17b282317374fa8a206806dcc18201106a60000000000fdffffff02f8650900000000000220020f

Send transaction from multisig wallet (to singlesig wallet)

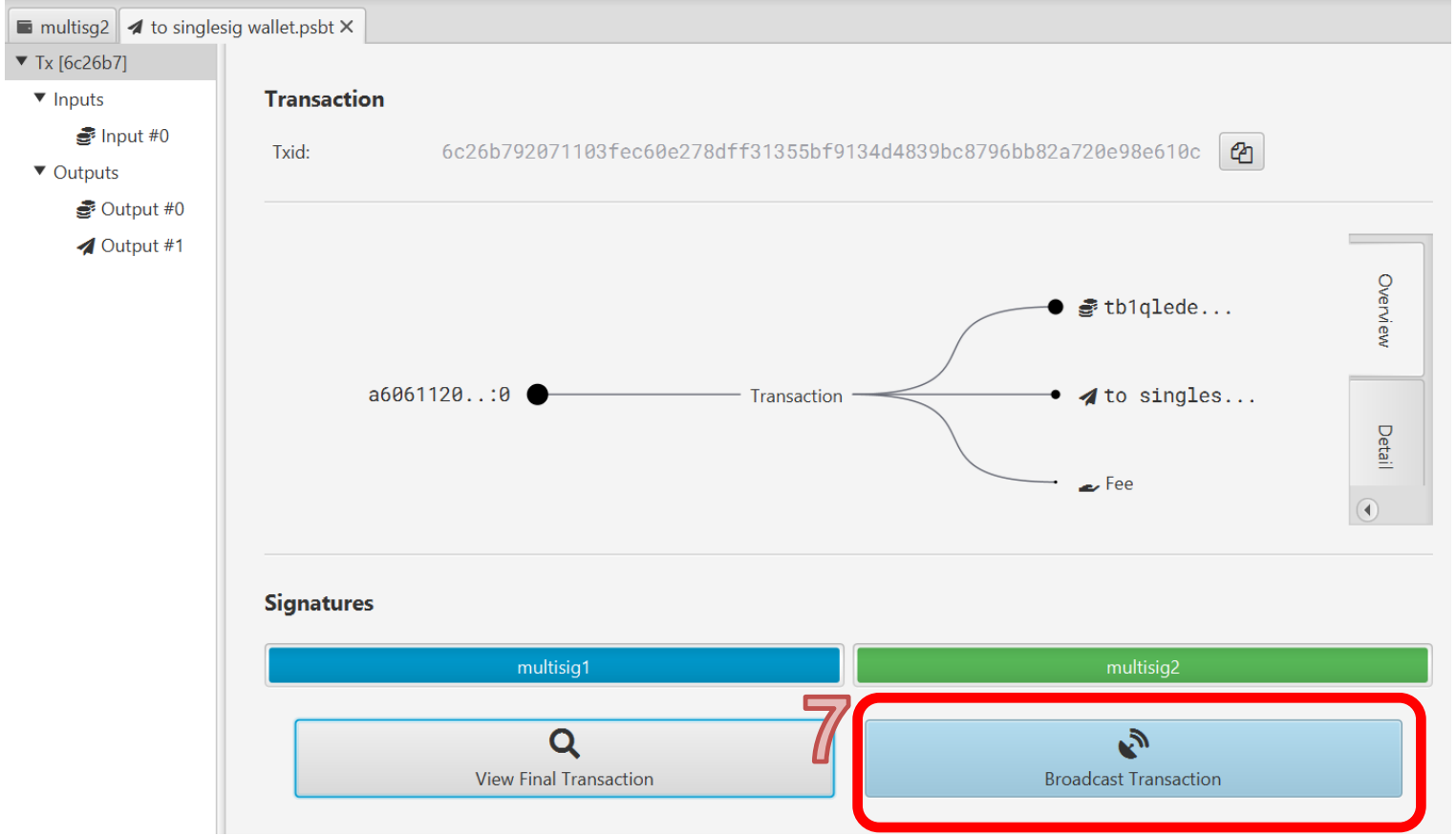
- (PSBT transaction is loaded in Sparrow wallet of second signer)

5. Check transaction parameters (address, amount, fee…)
6. If happy, click Sign button and 7. Broadcast



Send transaction from multisig wallet (to singlesig wallet)

- (Signatures from multisig1 and multisig2 signers are visible)



Analyze transaction via blockchain explorer

- https://blockstream.info/testnet/tx/
- Paste Txid after transaction is mined
- P2WSH WITNESS SCRIPT
  - OP_CHECKMULTISIG

multisig1 | to singlesig wallet

Tx [6c26b7]
- Inputs
  - Input #0
- Outputs
  - Output #0
  - Output #1

Transaction

Txid: 6c26b792071103fec60e278dff31355bf9134d4839bc8796bb82a720e98e610c

6c26b792071103fec60e278dff31355bf9134d4839bc8796bb82a720e98e610c

DETAILS

#0 a606112018cc6d8006a2a84f371723287bc1daf663ead7a902bc6c82874 0.01000042 tBTC

WITNESS

<empty> 304402202f519276e023c5d3cca1e27c9a3b363a3ace160d2049104f2987fad9439fb2230220644afbe98b231a3d58482e8df249daa41a0fba9ec3ff0d6bde7afd547aaf33bf01 304402207eb3c0d5d342e0f913871be8465e3556027f83ad1c7d97a2dedd66d60aac376102206b3aa76983f02ce97d4bee992d5637c2b6ae63e97722c09a03cafe1dd9f984a001 5221029e2df179fbc2029fcc6a4783805277287a2e109295ca60d6e954ba618f06b45d210313ccf34791e4f3f94f145ae34f849c3481463b6a5fe64ac035c82b91f3c9dc1f2103ec9ddac163c0296b5f2913633aeb4495

P2WSH WITNESS SCRIPT

OP_PUSHNUM_2 OP_PUSHBYTES_33 029e2df179fbc2029fcc6a4783805277287a2e109295ca60d6e954ba618f06b45d OP_PUSHBYTES_33 0313ccf34791e4f3f94f145ae34f849c3481463b6a5fe64ac035c82b91f3c9dc1f OP_PUSHBYTES_33 03ec9ddac163c0296b5f2913633aeb44952d4b805ec3e1212b0e5aef45b8c2c688 OP_PUSHNUM_3 OP_CHECKMULTISIG

#0 tb1qledelgj5s7hd3ryk7lfjw72xnwa36dg622kpxr9d40lhg6ewug4q4hmcsf 0.00615928 tBTC

TYPE	V0_P2WSH
SCRIPTPUBKEY (ASM)	OP_0 OP_PUSHBYTES_32 fe5b9fa25487aed88c96f7d32779469bbb1d351a52ac130cadabff746b2ee22a
SCRIPTPUBKEY (HEX)	0020fe5b9fa25487aed88c96f7d32779469bbb1d351a52ac130cadabff746b2ee22a
SPENDING TX	Unspent

#1 tb1ql29w297c63xwxcq9m8y2xrtetpcfx5vqq9t5ln 0.00383924 tBTC

TYPE	V0_P2WPKH
SCRIPTPUBKEY (ASM)	OP_0 OP_PUSHBYTES_20 fa8ae517d8d44ce36005d9c8a30d795870935180
SCRIPTPUBKEY (HEX)	0014fa8ae517d8d44ce36005d9c8a30d795870935180
SPENDING TX	Unspent

Questions

- Which option is better for backup (not loosing possibility to spend)? 1-of-3 or 3-of-3?
- Which option is better against and attacker (prevent her to spend your coins)? 1-of-3 or 3-of-3?
- What are advantages and disadvantages of 2-of-3 vs. 3-of-5?
- Can you authorize transaction of one signer not available? Two?
- Can multisig participants see all funds locked to multisig wallet?
- What shall you do if one signer loses control of funds?
- What you need to do if you would like to add another signer into quorum?
- Why is multisig transaction bigger than the singlesig one?
- Can you say if funds are locked (UTXO) to multisig wallet?
- Can you say parameters of multisig before funds are spent? After?
- Is Taproot (P2TR) changing anything?

ASSIGNMENT 4

Assignment 4 – analysis of Bitcoin transactions

- Analyze one block and related transactions from Bitcoin blockchain
  - Every student will have different block equal to the UČO (e.g., block ‘4085’ for P.S.)
- Preparation:
  - Download table from IS (hw04_task1_table.odt) and use it for **Task 1**
- Produce (2-3xA4) text solution for **Tasks 2, 3, 4 and 5**
  - Provide answers to questions asked, add annotated transactions graphs…
- Submit before **18.4.2024 23:59** into IS HW vault
  - Soft deadline: -1.5 points for every started 24 hours

Assignment 4 – analysis of Bitcoin transactions

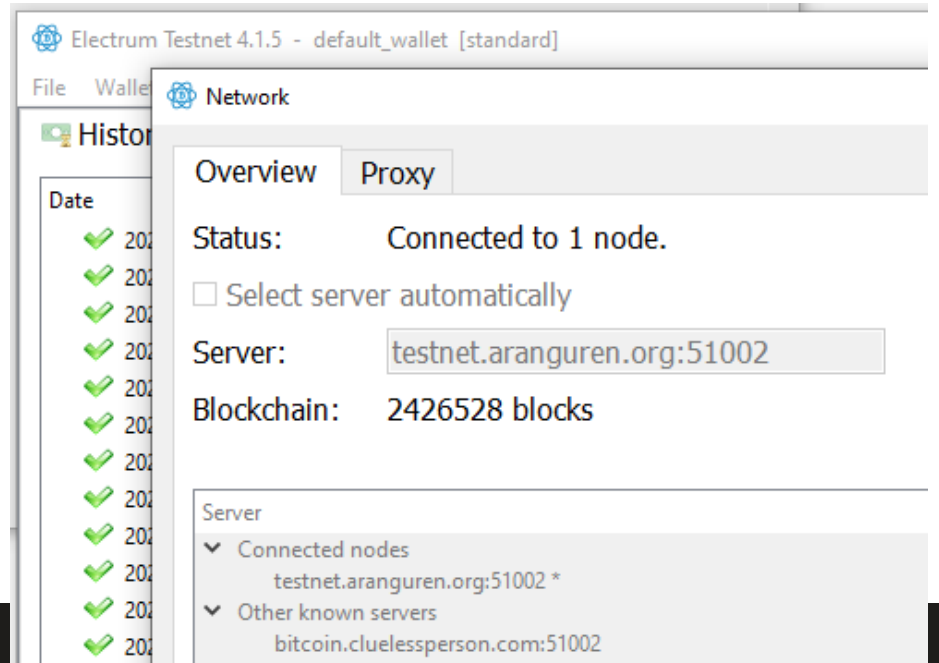
- **Task1: Basic info about “your” block**
  - Fill into hw04_task1_table.ods file from IS and submit (ideally included into the report file)
- **Task 2: Transaction with the largest WU size**
  - Find the biggest transaction by weight units, find its total size in bytes, discuss its purpose (inputs, outputs, any other info you can find)
- **Task 3: Multisignature transaction**
  - Find and discuss parameters of one multisignature transaction, annotate the lock and unlock script in details
- **Task 4: Chain analysis of coinbase transaction from “your” block**
  - Analyze the spending graph of coinbase transaction. Try to analyze the source of other bitcoins used as other input(s) with some of the coinbase tx output. Analyze at least 5 transaction hops (forward, backward for other inputs)
  - Draw graph, try to attribute entities, explain the likely meaning of transaction(s)…
- **Task 5: OP_RETURN**
  - Find at least three examples of OP_RETURN in your block. Try to figure out what is the use of it? (Readable string? OpenTimestamps?...)

IF YOU WOULD LIKE TO LEARN MORE ☺

RECOVERY OF WALLET (ELECTRUM)

Electrum wallet - preparation

- Download Electrum wallet: https://electrum.org/#download
  - Note: for real use, always verify PGP signature
- IMPORTANT: Run it on testnet, specify Electrum indexing server
  - electrum.exe --testnet -1 -s testnet.aranguren.org:51002:s



Task: Restore wallet created in Sparrow into Electrum

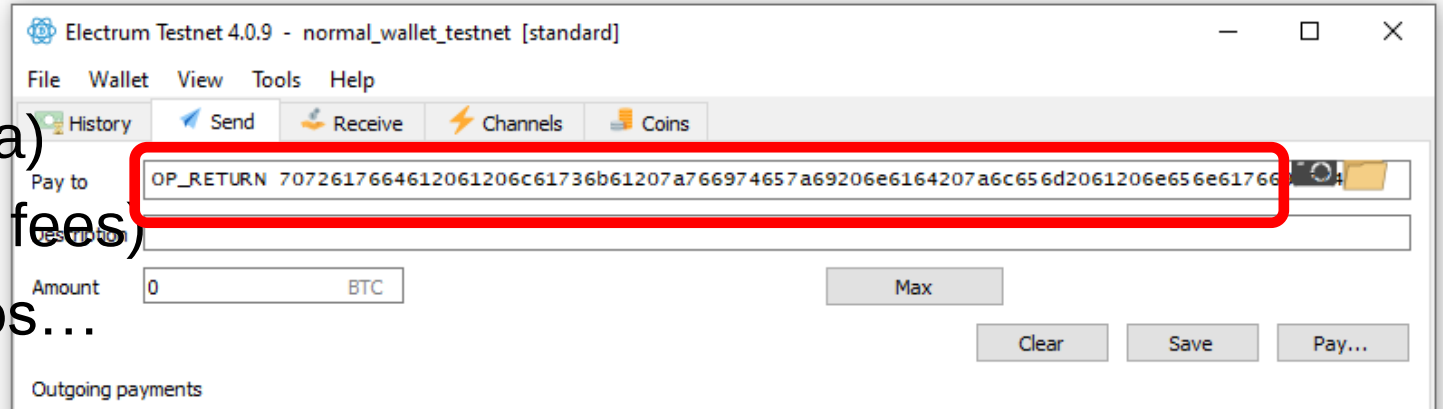
- Assumption: You have Sparrow wallet (testnet) created from last week
- Restore the master seed into different wallet software (Electrum)
  - Note: Only master seed + standardized derivation path is required
  - More detailed export including transaction labels possible
- Option 1: Using mnemonics words
  - : New/Restore → Standard wallet → ‘I already have seed’ → BIP39 words *insert_your_words_from_sparrow*
- Option 2: Export from Sparrow wallet (including transaction labels)
  - : File → Export wallet → Electrum → Export file => *.json file
  - : File → Open → *.json file

Note: Sparrow wallet does not have support for OP_RETURN yet

OP_RETURN (ELECTRUM WALLET)

Task: Store custom (limited) data into blockchain

- OP_RETURN instruction in lock script for provably non-spendable tx
  - Script execution never TRUE, full nodes can drop from list of UTXOs
- Send via Electrum (Pay to)
  - ‘OP_RETURN’ + ‘data’ (in hexa)
  - 0 amount (sender only pays tx fees)
  - 40 bytes, usable for timestamps…



- Locate tx on blockchain
  - https://mempool.space/testnet/
- With 1 peer: Find three ideas what to include and why
  - What information, how encoded, how retrieved, what are security benefits

Electrum Testnet 4.1.5 - default_wallet [standard]

File Wallet View Tools Help

History | Send | Receive | Channels | Coins | Console

Pay to: OP_RETURN 707632303420697320 6b69636b696e67

Description: test opreturn

Amount: 0 BTC

Max

Clear | Save | Pay...

Confirm Transaction

Amount to be sent: 0. BTC

Advanced

Mining fee: 0.0000014 BTC

Fee rate: ETA

Warning: The fee for this transaction seems unusually high. (100.00% of amount)

Cancel | Send

Inputs & Outputs

Details

tb1qn6nnzj99dccydlfvv6f0s9keyk… 6ate5ze3 0.00100000 tBTC

OP_RETURN pv204 is kicking 0.00000000 tBTC

tb1q015kvpeh04wvtunpw6v24eh5f0… x3le6g3t 0.00099860 tBTC

0.00099860 tBTC

OpenTimestamps protocol (https://opentimestamps.org/)

- Prove that document existed at date X (at latest)
- Merkle tree of all submitted document hashes within given period committed to Bitcoin blockchain (OP_RETURN)
  - https://petertodd.org/2016/opentimestamps-announcement
- Currently free to use (only one OP_RETURN embed)
  - Client needs to remember Merkle tree path + file => *.ots file

```
$ pip3 install opentimestamps-client
$ ots stamp secret.txt

$ ots info secret.txt
$ ots verify secret.txt.ots
Assuming target filename is 'secret.txt'
Calendar https://alice.btc.calendar.opentimestamps.org: Pending confirmation in Bitcoin blockchain
```

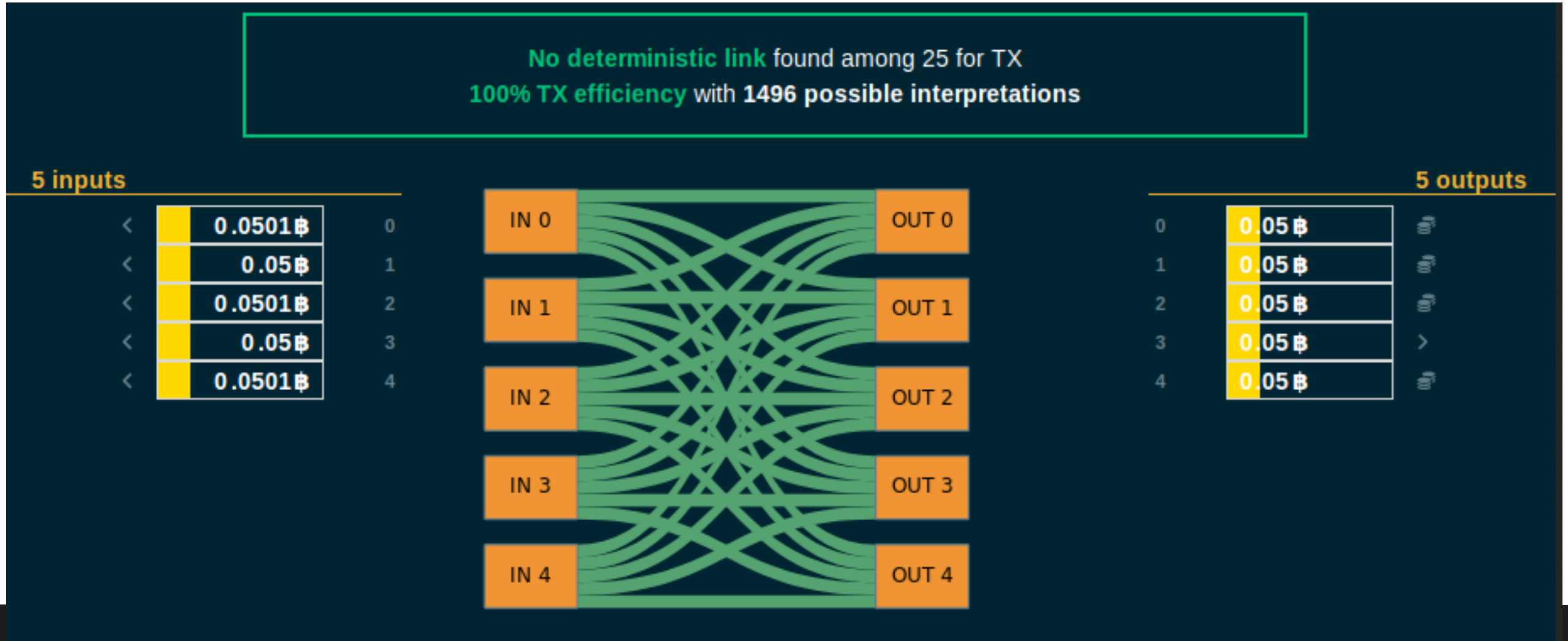
https://github.com/opentimestamps/opentimestamps-c
https://mempool.space/tx/5cfb6d1eee37cfd3dc51d01f
$ ots verify secret.txt.ots
Assuming target filename is 'secret.txt'
Calendar https://alice.btc.calendar.opentimestamps.or
Calendar https://finney.calendar.eternitywall.com: Tim

WHIRLPOOL COINJOIN

CoinJoin implementations

- Wasabi wallet https://github.com/zkSNACKs/WalletWasabi/
  - Centralized trustless coordinator, Tor, selected number of rounds executed within hours
    - https://docs.wasabiwallet.io/using-wasabi/CoinJoin.html
  - Wasabi 2.0 (beta) will offer non-equal output coinjoin https://blog.wasabiwallet.io/privacy-guarantees-of-wasabi-wallet-2-0/
  - Anonymity set decrease over the time as people send their outputs to KYC exchanges
- Samourai Whirpool https://docs.samourai.io/en/whirlpool
  - CoinJoin with variable number of rounds, centralized trustless coordinator
  - CoinJoin runs until output is send away from Whirpool (days/months)
  - If not fullnode then xpub must be provided => privacy risk, decreased anonymity set
    - e.g., Samurai RoninDojo https://ronindojo.io/
  - Clients: Samourai wallet / Whirpool cli, SparrowWallet (using Samourai code)
- JoinMarket
  - No central coordinator, market Maker(s) run own fullnode and provide liquidity
  - Coinjoin transaction creation is coordinated by Taker who is paying also fee (on-chain and to the Maker)
  - JoininBox - JoinMarket cmdline-focused distribution https://github.com/openoms/joininbox

Example Whirlpool CoinJoin mixing transaction (0.05 pool)

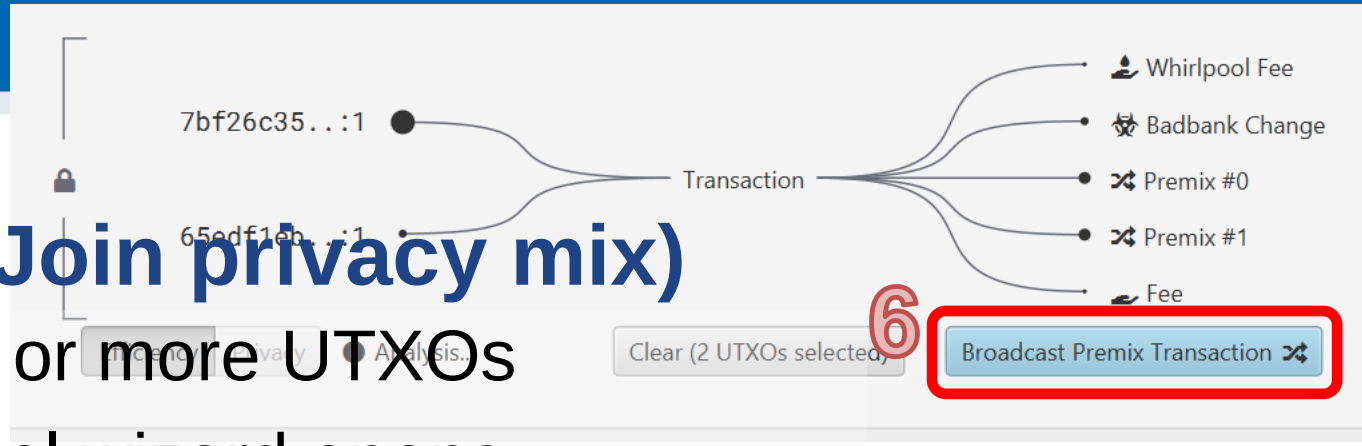


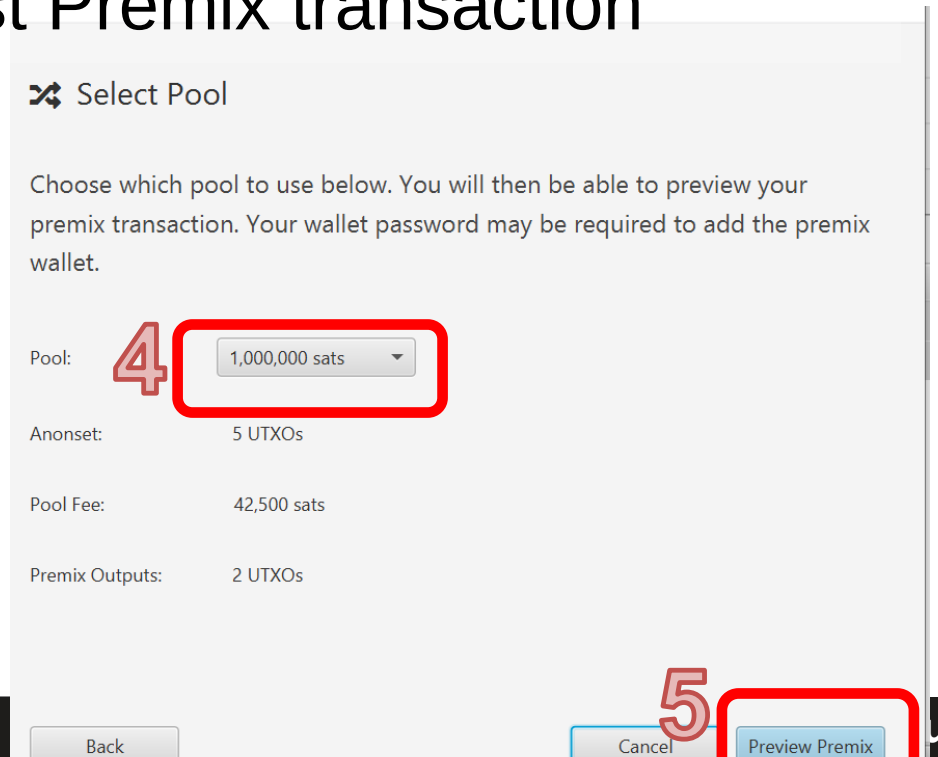
Whirpool CoinJoin privacy mix

- Open your standard Sparrow single signature wallet (created before)
- Work alone – mixing participants are found automatically
  - Connection to Whirlpool mixing coordinator is done via Tor
- Funds mixed are always available (you control private key)
  - can be spend them anytime

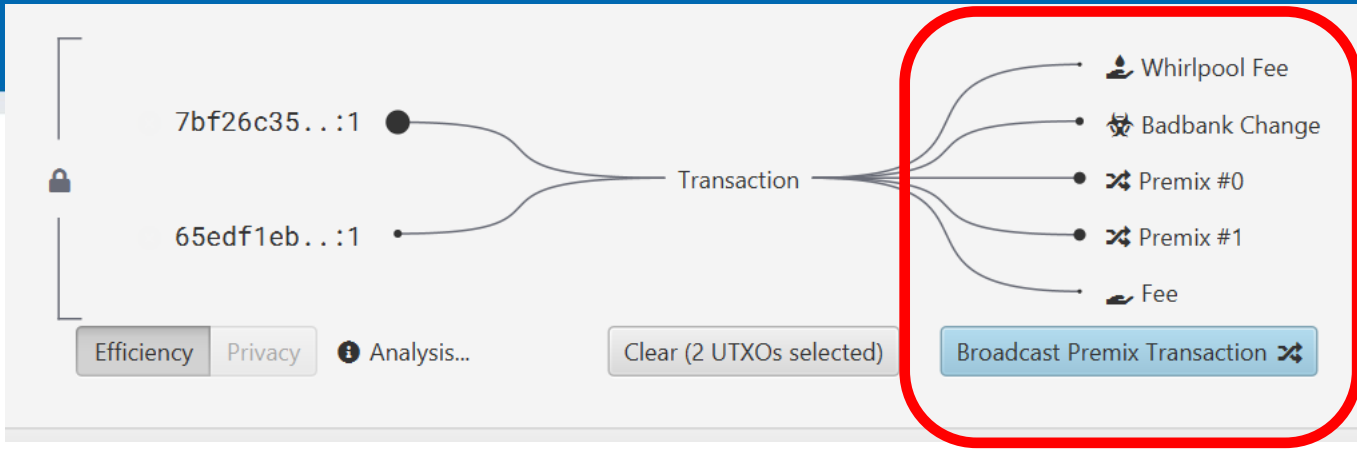
Samourai Whirlpool (CoinJoin privacy mix)

1. Click UTXO tab, 2. select one or more UTXOs
3. Click Mix Selected => Whirlpool wizard opens
4. Click Next until Select Pool, select 100k sats pool
5. Preview Premix, 6. Broadcast Premix transaction





Premix transaction TX0



- Whirlpool fee – one-time payment to Whirlpool coordinator (Samourai)
  - Based on pool size, NOT amount mixed (but smaller mixed UTXOs as result)
- Fee – mining fee to miners (based on actual blockspace demand)
- Premix #0, #1 … #N – initial premixed inputs of same size
  - These UTXOs will be input to mixing rounds
- Badbank change – remaining sats which cannot be put into another Premix #N+1 (as is smaller than mixing pool minimal size)
  - “toxic waste” – this UTXO is still tied to original input transaction (~your identity)
  - Do not merge with any mixed outputs (deanonimized)

Mixing procedure

- When TX0 is send to mempool, new UTXO(s) display in Premix tab
  - Wait till TX0 is confirmed, multiple UTXOs created based on the pool size and mixed amount
- Automatically, new Whirpool mixing transaction is created
  - New UTXO is displayed in Postmix tab
- As new blocks are mined, Postmix UTXOs are automatically included in subsequent mixing transaction(s) – Mixes column
  - Mixed unless wallet user send them elsewhere (continuous increase of anonymity set)
  - Mixed when someone creates new TX0 (new UTXO is paying for mining fees)
- Sparrow wallet must run for active mixing
  - Mixing is resumed automatically if Sparrow wallet is started again
- Funds can be spent anytime, options with improved privacy, send to another wallet after defined number of mixes…

europen | europen3 | europen2 | europen4 | europen5 | [3da528]

Deposit
Premix
Postmix
Badbank

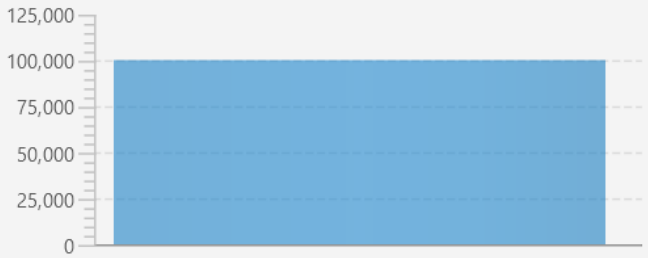
Transactions
Send
Addresses
UTXOs
Settings

Unspent Transaction Outputs

Balance: 100,000 sats $ 30.00

Mempool: 0 sats

UTXOs: 1



Date	Output	Mixes	Label	Value
2022-05-22 13:49	3da52874..:4	2		100,000

Registered input (1 of 7)

Stop Mixing | Mix to... | Clear | Send Selected (100,000 sats)

Analyze mixing transaction

1. Analyze using Sparrow wallet visualization
   – UTXO, symbol of magnifier , click topmost item Tx […]
2. Analyze using blockchain explorer
   – Copy txid, use https://blockstream.info/testnet/tx/

- For mainnet transactions, other privacy estimation tools exist
  – Always use Tor when accessing! (do not link your IP with transactions of interest)
  – https://KYCP.org (single transaction, examples)
  – https://oxt.me (graph of transactions, forensic analysis)

1

europen | europen3 | europen2 | europen4 | europen5 | [3da528]

- Tx [3da528]
  - Inputs
    - Input #0
    - Input #1
    - Input #2
    - Input #3
    - Input #4
  - Outputs
    - Output #0
    - Output #1
    - Output #2
    - Output #3
    - Output #4

Transaction

Txid: 3da52874471746d2fec8f6c246392a5274a21d4fe557f435c5bb76d39a0f3e77

10565c06..:1
1202b481..:3
377a2765..:2
c0d1f65f..:0
e8dc6e2e..:0

Transaction

tb1qr8al...
tb1qymxf...
tb1qfp30...
tb1qf5mw...
tb1q7ue3...
Fee

Overview
Detail

Blockchain

Status:	3 Confirmatio
Block Height:	2226930
Timestamp:	2022-05-22 1
Block Hash:	000000000

PRIVACY ANALYSIS

2 Possibly a CoinJoin transaction

3da52874471746d2fec8f6c246392a5274a21d4fe557f435c5bb76d39a0f3e77

DETAILS +

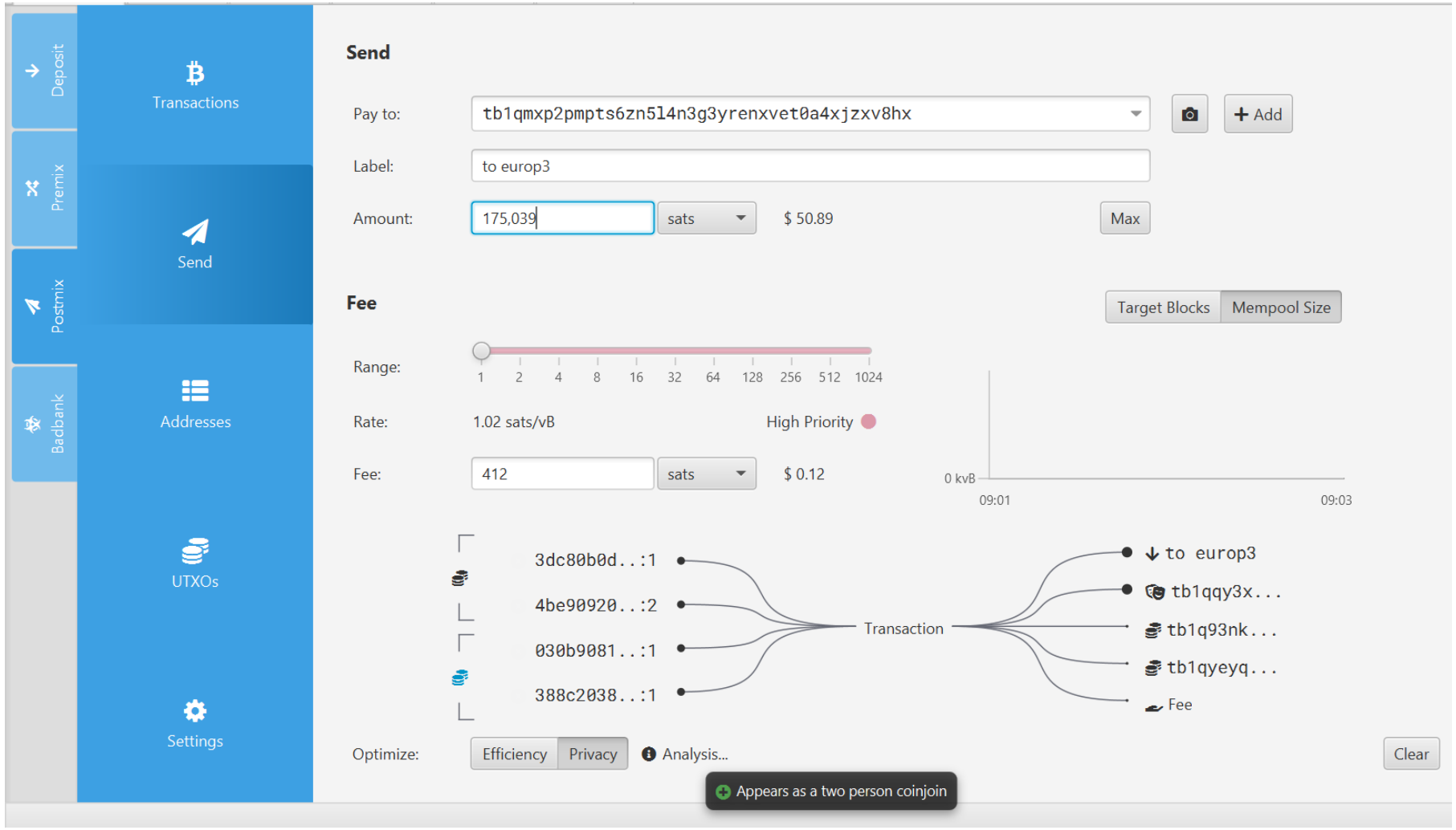
#0	10565c0673127c469c20c09ba794fbf037829bae501c55e6b92257e850b96c52:1	0.001 tBTC
#1	1202b4818529834229e49c3cca7df364536b0f7baa6f080795536ee70f8d15d8:3	0.00100302 tBTC
#2	377a2765e7ef45ffed10f0f3c5f573ebd753d1a2313abddb24e652de6b836ba1:2	0.00100302 tBTC
#3	c0d1f65ff9082972a33ba81f2ad93f82d7fcd37391185e7b1aa70242d89d5c01:0	0.001 tBTC

#0	tb1qr8al4cc0s5zu2pyqqrzdts243sr7k9hmc80ket	0.001 tBTC
#1	tb1qymxfyvf37gr3ztst8p8dj5cmwr48w7rda72glg	0.001 tBTC
#2	tb1qfp309nz0l0fd2zqaa2s3lr9prykmw08tpe9chc	0.001 tBTC
#3	tb1qf5mwxawpcj8hraz5nd6e2u9s388n0a5xpjwmrf	0.001 tBTC
#4	tb1q7ue3sy6rjmzdadz9t8wzaqc98ev5kqcsyvfmau	0.001 tBTC

Post-mix spending

- CoinJoin mixing breaks on-chain heuristics (input→output)
- Your UTXO is now private, but must be also used privately later
- Do not use mixed (Postmix) and unmixed (Badbank) UTXOs!
- Fake/real collaborative spent (PayJoin)
  - Two or more people spending together (inputs from both, outputs to both)
  - Simulated PayJoin (all inputs yours, but looks like collaborative spent)
- Coin control
  - Whole UTXO send to new address (no change)
- Atomic swap – trustless exchange of UTXOs (even on different chains)
  - Utilizes timelock – transaction must be finished by both parties till deadline, otherwise cancel

Postmix spent – simulated PayJoin



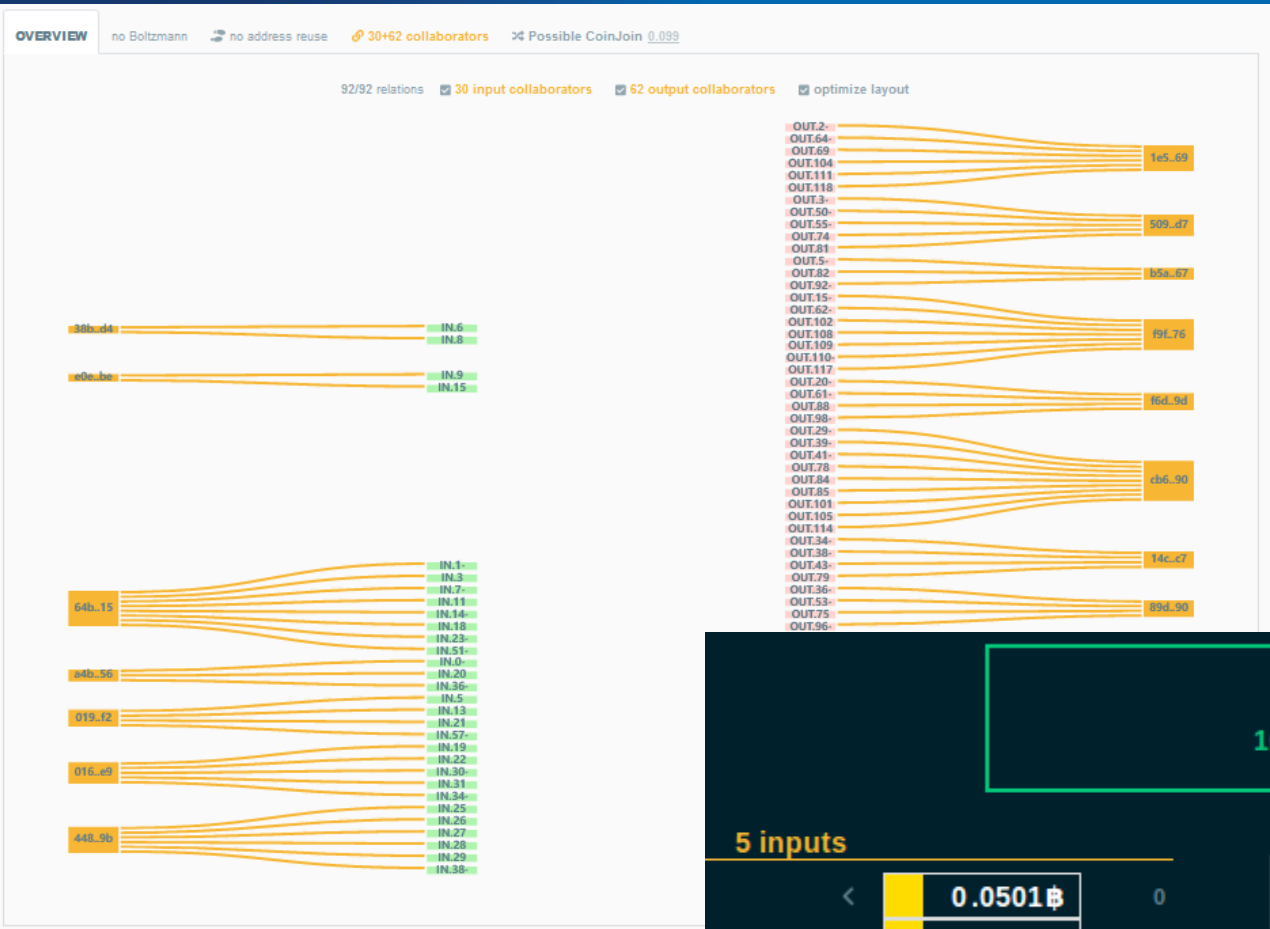
Questions

- Does Whirpool CoinJoin require online connectivity?
- How many other participants are required?
- How many mixing rounds are enough?
- What is the difference between mixing pools?
- Who is paying for the mixing transaction?
- What happens if you create transaction using both Postmix UTXO and Badbank UTXO?

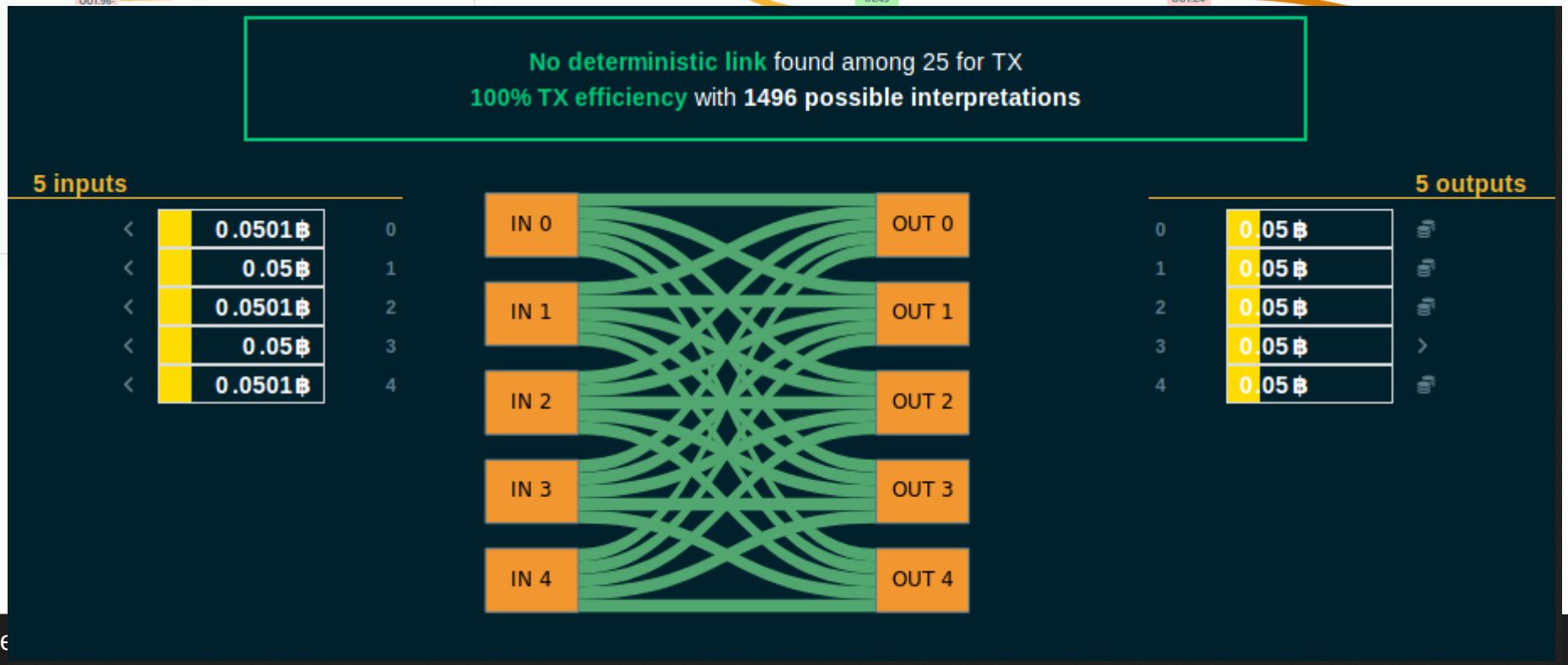
COINJOIN / PAYJOIN TRANSACTIONS

Analyze CoinJoin and PayJoin transactions

- Group of 3 students, share screen
- Example CoinJoin transactions
  - https://nioctib.tech/#/transaction/92a78def188053081187b847b267f0bfabf28368e9a7a642780ce46a78f551ba (example from https://en.bitcoin.it/wiki/CoinJoin)
  - https://blockstream.info/tx/c69aed505ca50473e2883130221915689c1474be3c66bcf7ac7dc0e26246afc8 (example from Wasabi wallet https://wasabiwallet.io/)
- Example PayJoin transaction
  - https://nioctib.tech/#/transaction/7104bae698587b3e75563b7ea7a9aada41d9c787788bc2bf26dd201fd7eca8a2
- Analyze with https://oxt.me and https://kycp.org
  - https://kycp.org/#/c69aed505ca50473e2883130221915689c1474be3c66bcf7ac7dc0e26246afc8
- Anything special in Lock and Unlock script?
- How can you find out if given TX is CoinJoin transaction?
- How can you find out if given TX is PayJoin transaction?


OVERVIEW
no Boltzmann
no address reuse
30+62 collaborators
Possible CoinJoin 0.099
92/92 relations
30 input collaborators
62 output collaborators
optimize layout


62/62 relations
6 addresses reuses
28 input collaborators
23 output collaborators
5 CoinJoin exit merges
optimize layout

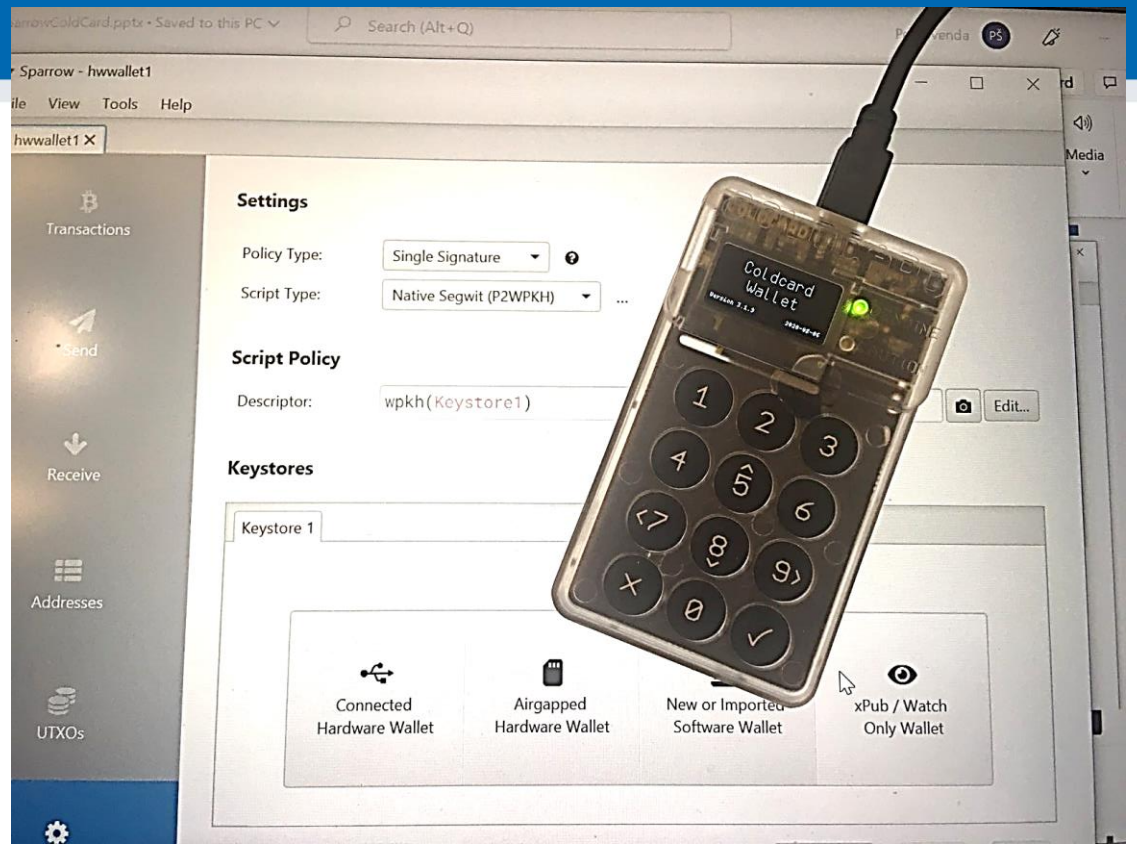

No deterministic link found among 25 for TX
100% TX efficiency with 1496 possible interpretations
5 inputs
0.0501฿
0.05฿
0.0501฿
0.05฿
0.0501฿
IN 0
IN 1
IN 2
IN 3
IN 4
OUT 0
OUT 1
OUT 2
OUT 3
OUT 4
5 outputs
0.05฿
0.05฿
0.05฿
0.05฿
0.05฿

Wasabi CoinJoin 1.0

- Equal output CoinJoin, mixed outputs all have same size, around 0.1btc
- Mixing performed in single round with larger number of participants (e.g., 100)
- Untrusted coordinator required
  - Operated by ZKSnacks company, but can be others

Wasabi CoinJoin 2.0 (WabiSabi protocol)

- Non-equal output CoinJoin
  - mixed outputs have different size, no (toxic) change
- Mixing performed in a single round with larger number of participants (e.g., 100)
- Untrusted coordinator required
  - Operated by ZKSnacks company, but can be others



SINGLE-SIGNATURE HARDWARE WALLET

Before we start…

- You have only one hardware wallet per group
  - Only one of you will have hardware wallet with Sparrow
  - All others will have software wallets

- **VERY IMPORTANT!!!**
  - ColdCard is real hardware wallet (~$100)
  - “Bricked” if correct PIN is forgotten unknown (no “reset” button)
  - For this tutorial, **always set PIN to 12 34 !!!**

Steps of hardware wallet usage

1. Prepare ColdCard hardware, generate and backup new wallet
   – No computer required, everything happens on ColdCard device
2. Prepare Sparrow on PC with private keys stored on ColdCard
   – Public information from ColdCard wallet is exported to Sparrow
3. Receive tBTC to ColdCard wallet (via Sparrow)
   – No ColdCard required, only public keys are required
4. Send tBTC from ColdCard wallet (via Sparrow)
   – Private keys on ColdCard required, checks and signing happens on ColdCard

Update firmware on all wallets, update demo pictures

1. PREPARE COLDCARD HARDWARE, GENERATE AND BACKUP NEW WALLET

Prepare your ColdCard device

1. Open sealed bag
2. Connect via USB cable
3. Read and accept conditions on small screen, press OK
4. Check the serial number match (screen, bag), press OK
   - What is security goal of this check?
5. ‘Choose PIN Code’ option, press OK
6. Enter PIN Prefix: **USE `12`!!!**, press OK
   - Write on paper shown words (what they are for?)
7. Enter rest of PIN: **USE `34`!!!**, press OK
8. Generate new wallet, write on paper 24 words, verify 24 words
9. State: ‘Ready to Sign’ option shall be displayed

---

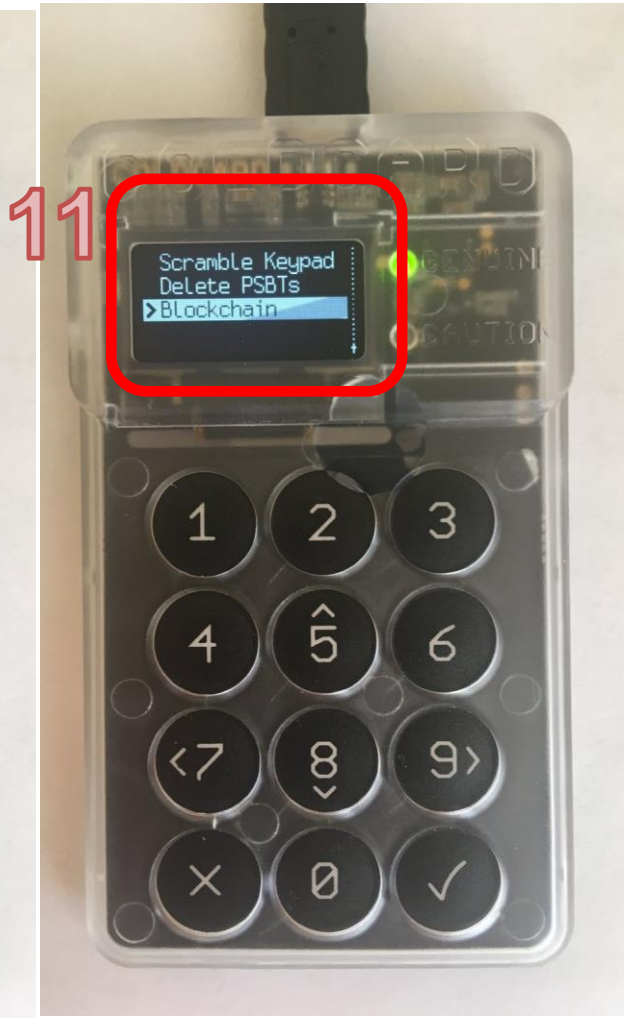
10. Move down to ‘Settings’
11. Move down to ‘Blockchain’
12. Change to ‘Testnet: BTC’

We will work with testnet BTC => need to tell wallet to use testnet addresses


1
Coinkite
293761048487
DONT TRUST.
VERIFY.
DONT TRUST.
VERIFY.
2
3
accept terms and
continue.
5
Choose PIN Code
Advanced
Bag Number
Help
THERE IS
ABSOLUTELY NO
WAY TO RECOVER A
FORGOTTEN PIN!
Write it down.
There is
ABSOLUTELY NO
WAY to 'reset
the PIN' or
'factory reset'
6
Enter PIN Prefix
7
Enter rest of PIN
8
New Wallet
Import Existing
Help
Advanced
Generating...
9
Ready To Sign
Logout
Advanced

Set wallet to use testnet BTC

If not set to testnet, then Sparrow wallet will (later) not detect the connected ColdCard during Sign operation

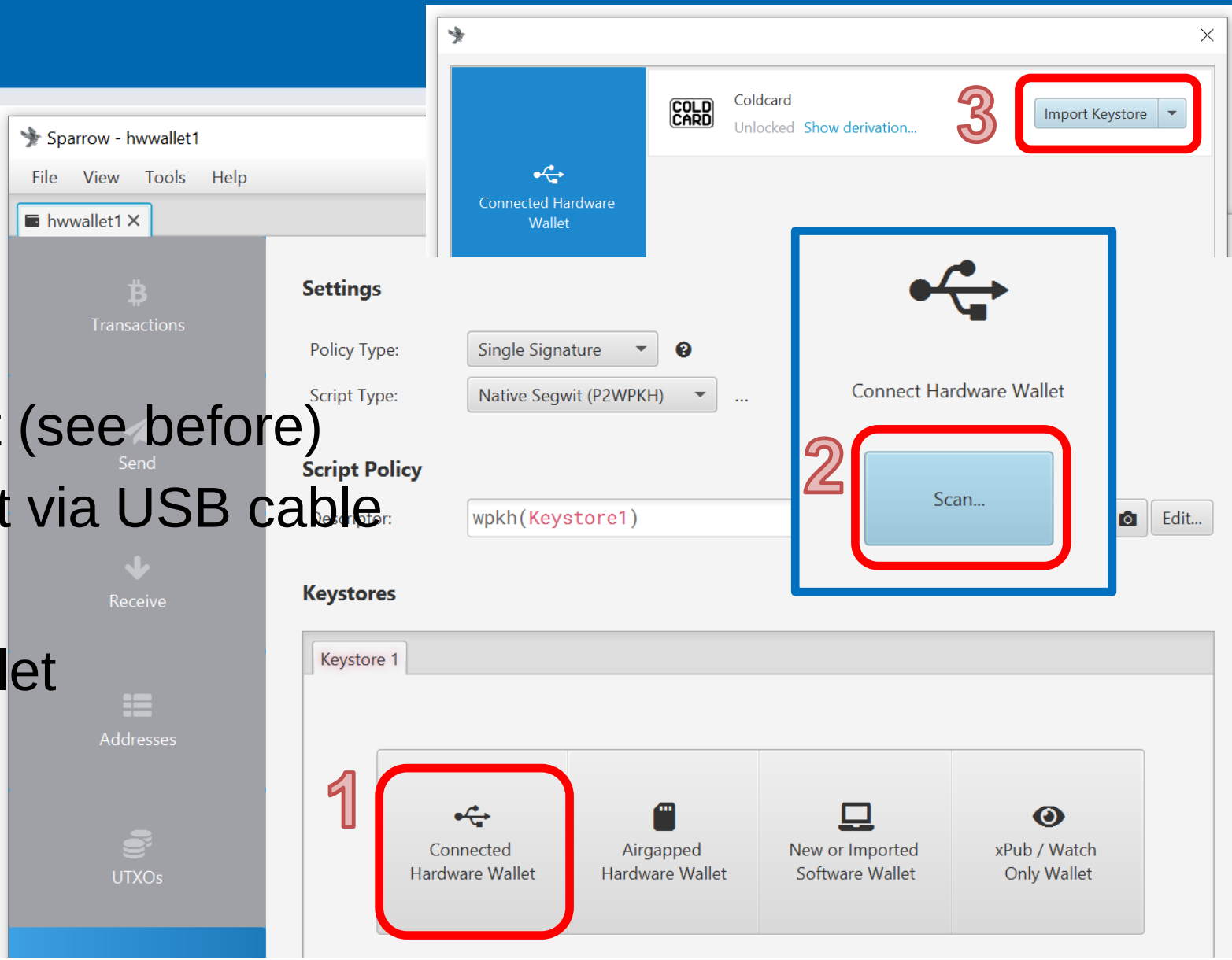


2. PREPARE SPARROW ON PC WITH PRIVATE KEYS STORED ON COLDCARD

Create wallet

- `sparrow -n testnet`
- Prepare ColdCard wallet (see before)
- Connect ColdCard wallet via USB cable
- File → New wallet

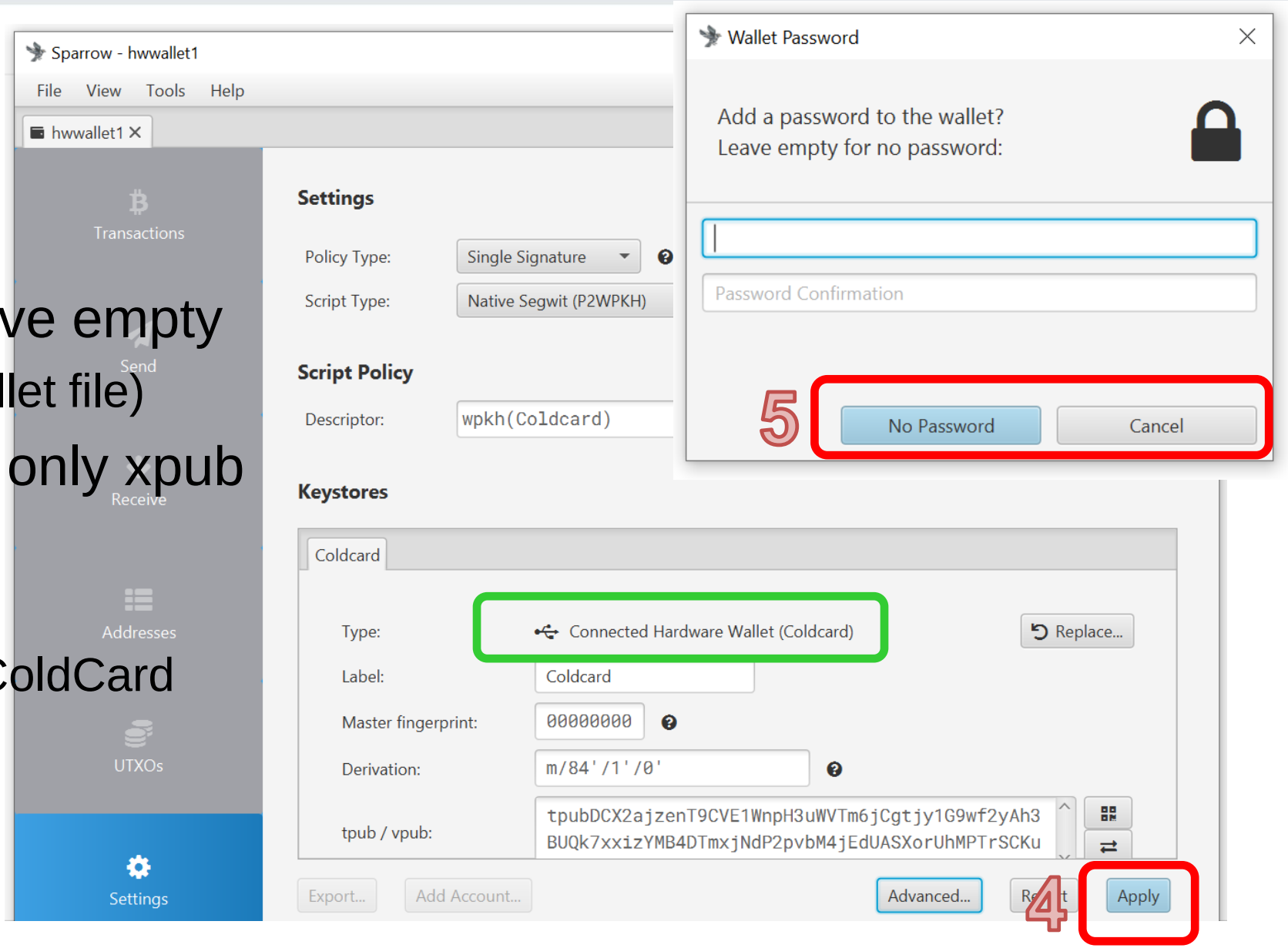
1. Connect Hardware Wallet
2. Scan
3. Import Keystore



Create wallet

6. Apply
7. Set password or leave empty
   – (encryption of local wallet file)

- Local wallet contains only xpub
  – *.mv.db file
  – File→Open wallet
  – Private key(s) are on ColdCard



3. RECEIVE TBTC TO COLDCARD WALLET (VIA SPARROW)

Task: send some tBTC from software to hardware wallet

- Exactly same procedure as for sending between software-only wallets
  - Hardware wallet's private key is not required for receiving
- Person with ColdCard shall receive one transaction from every other person (PC1 and CC)
- Obtain his/her receive address
  - Via messenger: CC → Receive tab → Copy address → send via Signal → PC1
  - Via QR: CC → Receive tab ; PC1 → Send → camera icon → scan address QR
- Enter some sats into Amount box
  - Observe visualized transaction below (more inputs may be added)

PC1

Sparrow - wallet1

File View Tools Help

wallet1 | to coldcard wallet

Transactions
Send
Receive
Addresses
UTXOs
Settings

Send

Pay to: tb1q7a777xe6jppnk2az43qqq5r856gm0dhcwl9y4g

Label: to coldcard wallet

Amount: 354,290 sats $ 72.13 Max

+ Add

Fee

Target Blocks | Mempool Size

Range: 1 2 4 8 16 32 64 128 256 512 1024

Rate: 1.01 sats/vB High Priority

Fee: 141 sats $ 0.03

0 kvB 16:04 17:30

internal send (cha...

Add Mix Partner?

Transaction

to coldcar...

tb1quj0t...

Fee

Optimize: Efficiency | Privacy Analysis... Clear Create Transaction »

CC

Sparrow - hwwallet1

File View Tools Help

hwwallet1

Transactions
Send
Receive
Addresses
UTXOs
Settings

Receive

Address: tb1q7a777xe6jppnk

Label: from swwallet wallet1

Derivation: m/84'/1'/0'/0/0

Last Used: Never

Required ScriptPubKey

Script:

Output Descriptor

Descriptor: wpkh(034bff5cbec46af1833f5e222bc66006f fcdbb94e222e67f5160c35d0cddb4df6b)

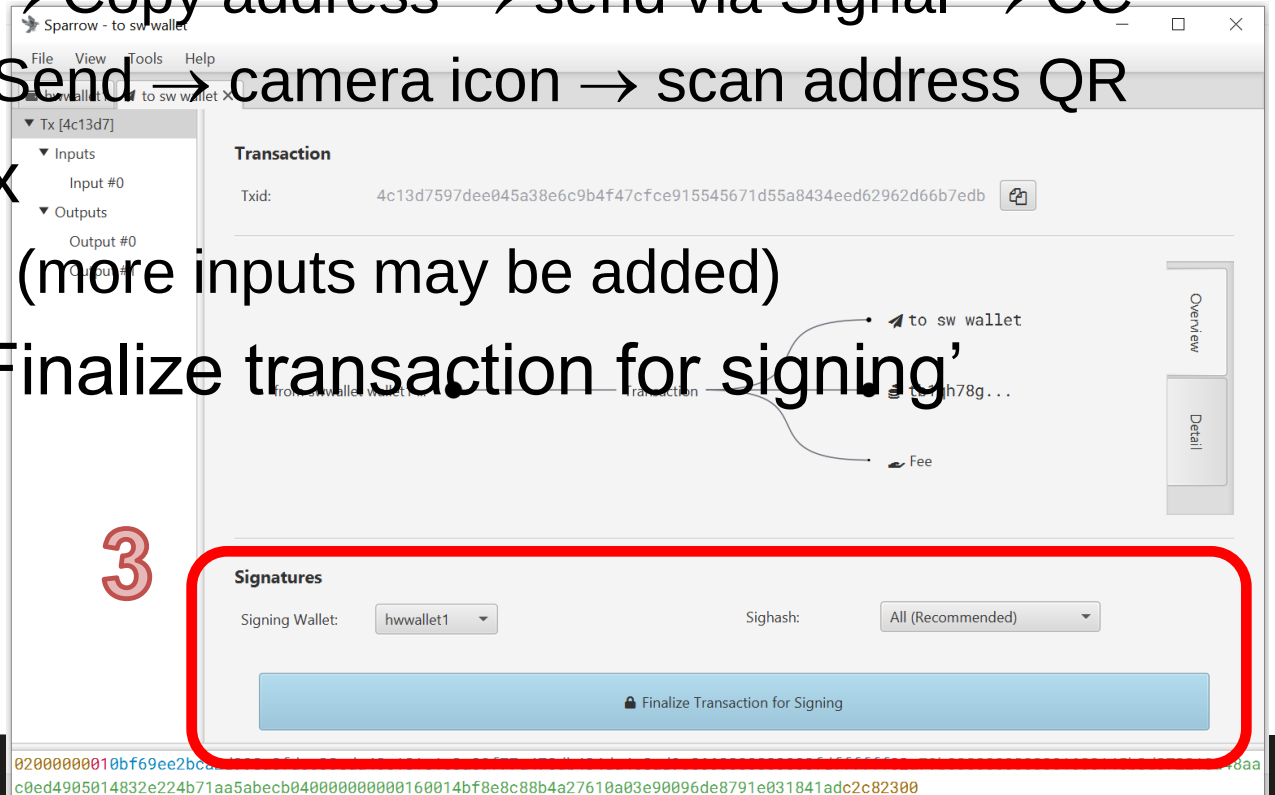
Display Address | Get Next Address

4. SEND TBTC FROM COLDCARD WALLET (VIA SPARROW)

Task: send some tBTC from hardware to software wallet

- Person with ColdCard sends to at least one other person (CC → PC1)

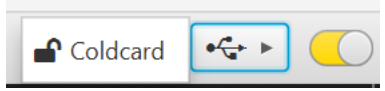
1. Obtain PC1's receive address
   - Via messenger: PC1 → Receive tab → Copy address → send via Signal → CC
   - Via QR: PC1 → Receive tab ; CC → Send → camera icon → scan address QR
2. Enter some sats into Amount box
   - Observe visualized transaction below (more inputs may be added)
3. Click 'Create transaction', click 'Finalize transaction for signing'



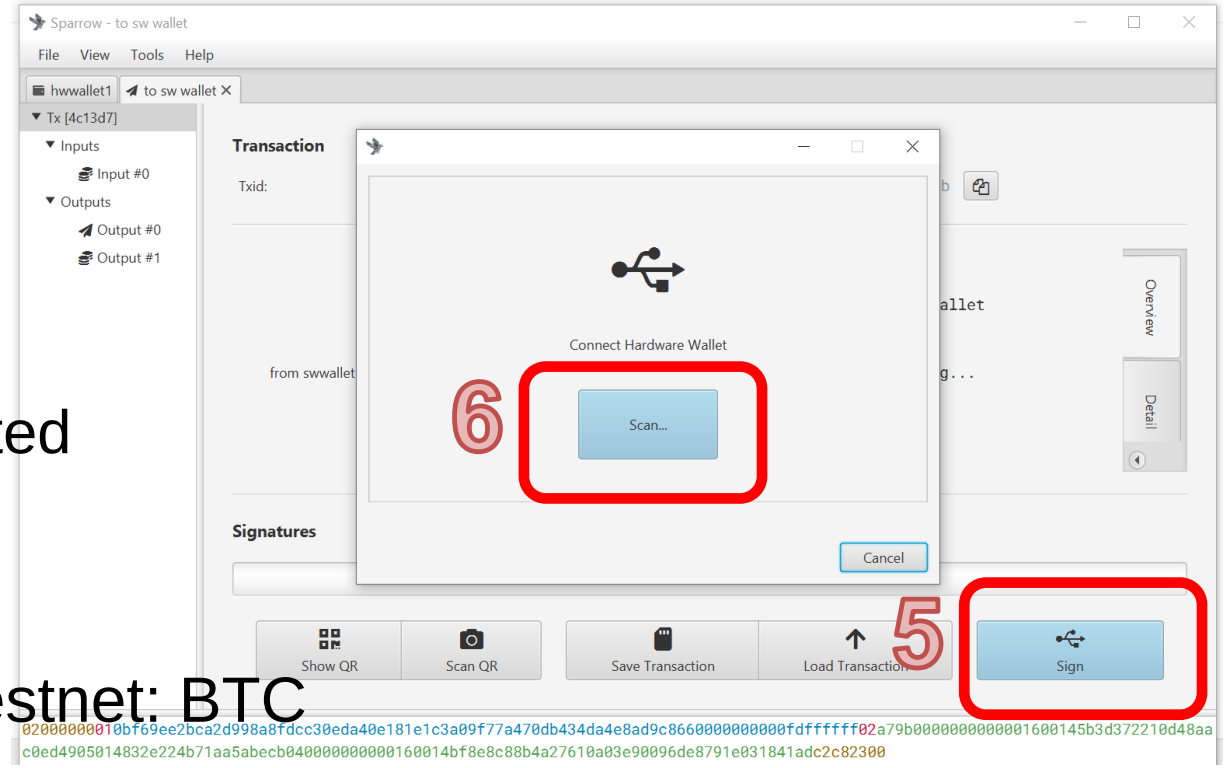
Send some tBTC from hardware to software wallet

4

4. Connect ColdCard via USB
   - Enter PIN Prefix, press OK
   - Enter rest of PIN => ‘Ready To Sign’
5. Click ‘Sign’ in Sparrow
6. Click ‘Scan’ in Sparrow

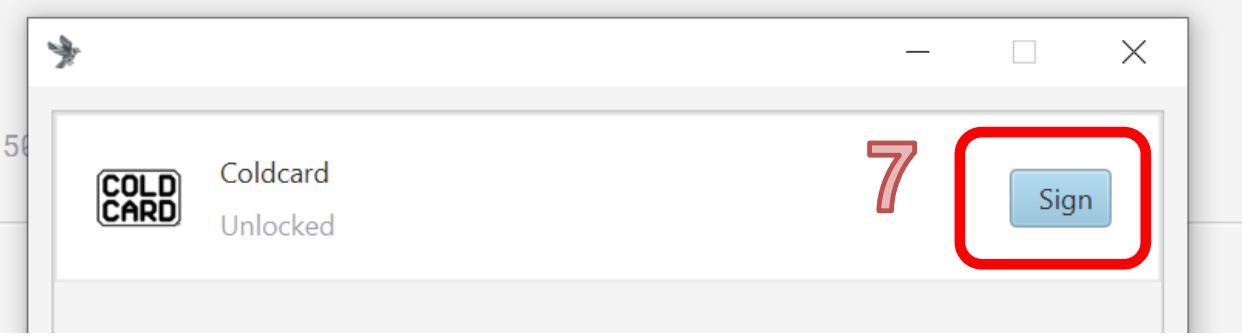
- Note: 

  - Look for icon after is ColdCard connected
  - If icon is not visible, try to reconnect
  - If icon is visible but Scan fails, check
    - ColdCard:Settings→Blockchain→Testnet: BTC



Send some tBTC from hardware to software wallet

7. Select ColdCard and click ‘Sign’



8. Verify on ColdCard’s screen (compare with your Sparrow)
   – Amount, address, fee, changeback, changeback address
   – Press OK if match
9. Click ‘Broadcast Transaction’
   – Transaction is now complete, broadcast to network

Task: attack your setup with hardware wallet! (15 min)

- Imagine five different ways how an attacker can steal your funds from your Sparrow single signature wallet with ColdCard
  - Continue in Miro: https://miro.com/app/board/uXjVPaI0Mp4=/?share_link_id=697987574971
    - Password: ‘fimunicz’
  - Compare to situation without hardware wallet
  - Discuss the cost and prerequisites of the different attacks
- Consider at least the following:
  - Phishing? Physical attack? Logical attack? Side-channel attack? Malware? Supply chain? …

Questions

- Is wallet owner an attacker against embedded secure element?
- What protection is offered by air-gapped mode with memory card?
- Why newer ColdCard Mk4 has 2 different secure elements?
- Would hardware wallet with secure element but without display provide same assurances?
- Can be hardware wallet firmware buggy? Can you find such example? Compare its Trusted Computing Base to notebook.
- How to securely update the ColdCard's firmware?
- How will you recognize fake ColdCard/secure element?

Questions

- What is stored on a ColdCard’s secure element?
- Where are private keys stored? Are they stored or generated on demand?
- What if you lose your ColdCard device?

LIGHTING NETWORK

Get some satoshi via Lighting network

- I will send some satoshi to one member of your project group
- She/he will send corresponding fraction to each of the remaining members
- Poor-man option: Custodial wallet (beware, is custodial!)
  - Wallet of Satoshi (Android, iOS), Setup time: instant installation and use
- Better option: Non-custodial wallet connected to hosted Lighting wallet
  - BlueWallet, you need to have at least some on-chain btc (at least 30k sats == 0.0003 btc)
  - Your wallet holds the private keys, but channels are opened by trusted service
  - Setup time: Takes up to several hours before ready (on-chain transactions)
- Best option: Setup your own full node and own Lighting node
  - E.g., Raspi4 + 1TB HDD + mynodebtc.com image + mobile wallet (BlueWallet, Zap, RTL…)
  - Similar to previous option, but Lighting wallet now connects to your Lighting node
  - Setup time: Days before your full node is synchronized, then several hours to open channel

Getting some bitcoins (in general)

- On mainnet (real bitcoins)
  - exchange, BTC ATM, beer for sats with friends, get paid in btc…
- Testnet (test bitcoins)
  - electrum.exe --testnet , generate new standard wallet, get testnet address (starts with m)
  - Go to https://coinfaucet.eu/en/btc-testnet/, ask for coins to your testnet address
  - Testnet explorer: https://blockstream.info/testnet/
- Regtest (local bitcoins)
  - Complete blockchain on your PC, you are sole miner => mine them
  - `bitcoin-cli -regtest getnewaddress`
  - `bitcoin-cli -regtest generatetoaddress 101 miner_address`

Note: This tutorial is achieving same results as tutorial with Sparrow wallet. Sparrow wallet is overall more capable, leaving it here for historical reasons

MULTIGNATURE WITH ELECTRUM WALLET

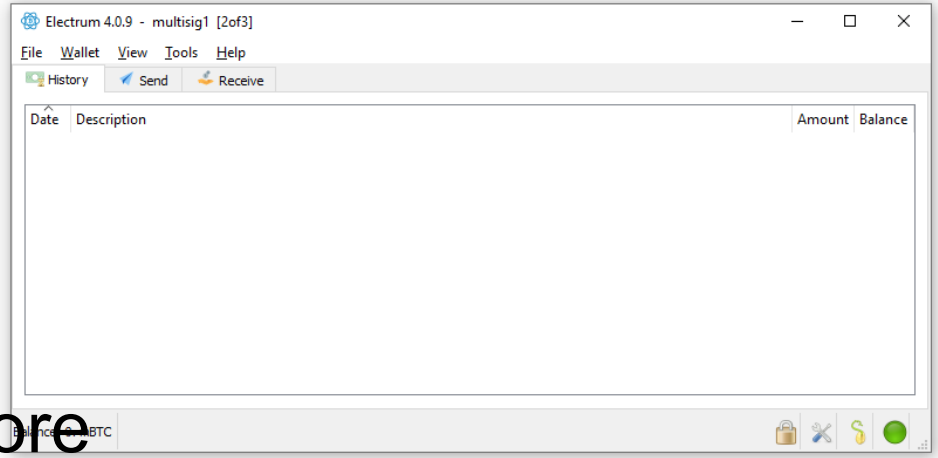
Task: using multisignature wallet (3ppl/room)

1. Create new 2-out-of-3 multisignature wallet in Electrum
   - All three people in the group are participants (separate machines)
2. Send some coins from last week to multisig wallet
   - Generate new receiving address
   - Wait till included in block
- Analyze TX (from normal to multisig) via chain explorer - How lock script looks like? Why?
  - Screenshot explorer, annotate
3. Send from multisig wallet back to standard one
   - Why you need to generate PSBT?
   - Is it safe to send PSBT via email?
   - Who can broadcast transaction when 1, 2 and 3 signatures are made?
- Analyze TX (from multisig to normal) via chain explorer - How unlock script looks like? Why?
  - Screenshot explorer, annotate

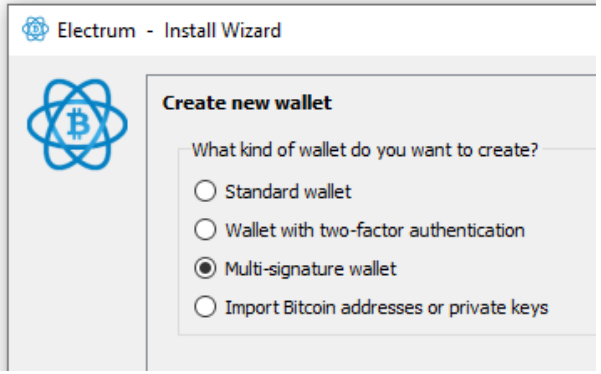
Important: Use Electrum 4.2.0 or higher

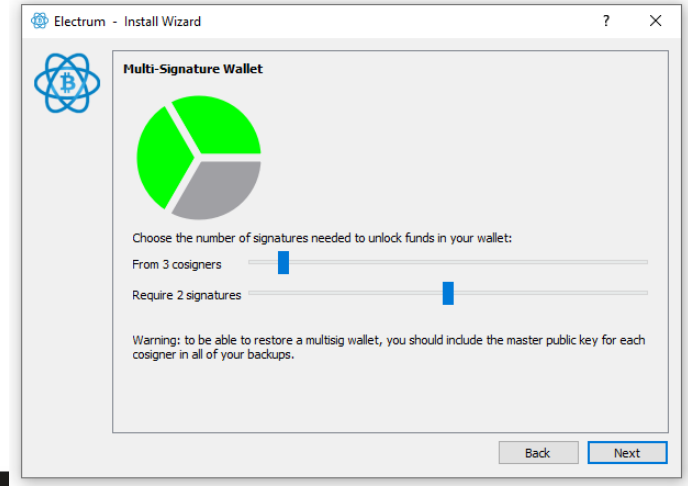
- You need to have same type of address
  - 4.2.0 is allowing only for segwit addresses
  - Older version may allow for legacy addresses – can’t be mixed with segwit

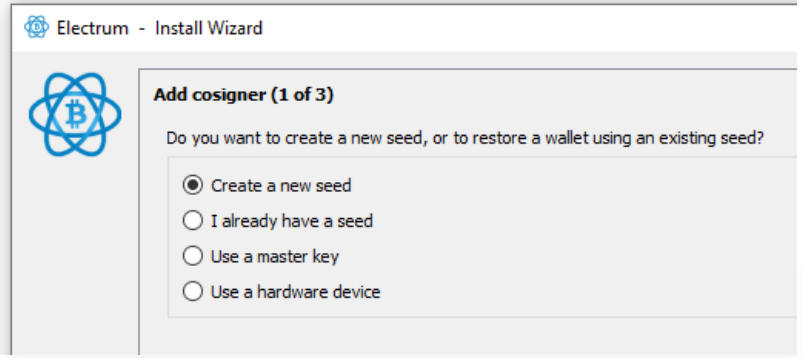
Creating multisig wallet (--testnet)

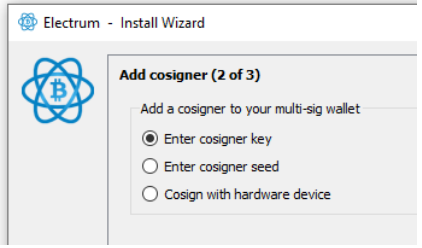


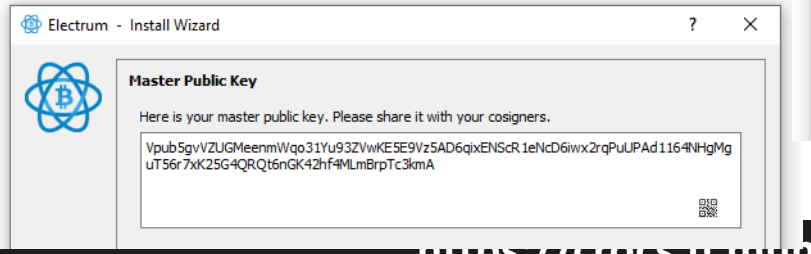
- If you already have wallet: File → New/Restore
  - All three people performs the same process
- Save seed and masterpub key for yourself (cosigner 1)
- Get masterpub key from others, Add cosigner (2 of 3), (3 of 3)
- Finish creation of multisig wallet

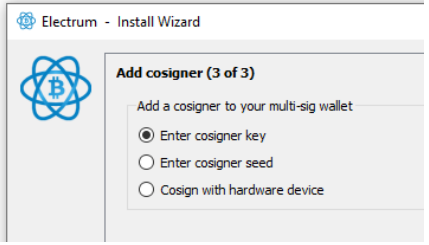






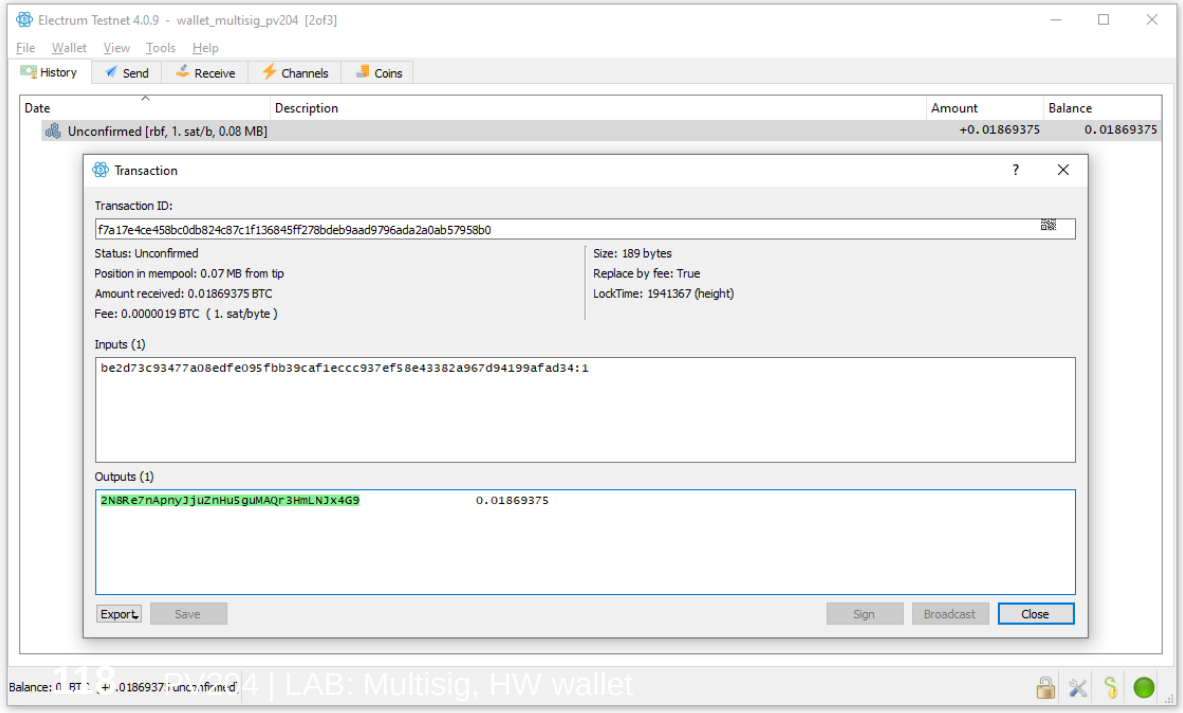


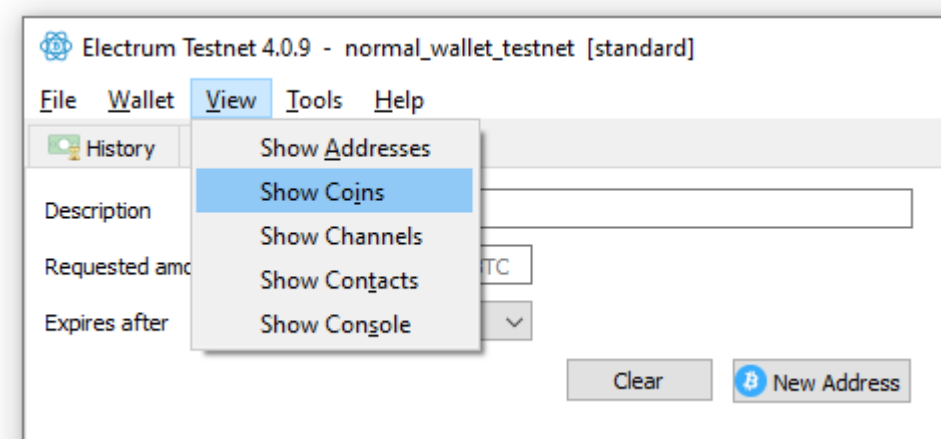




Send from normal wallet to multisig one

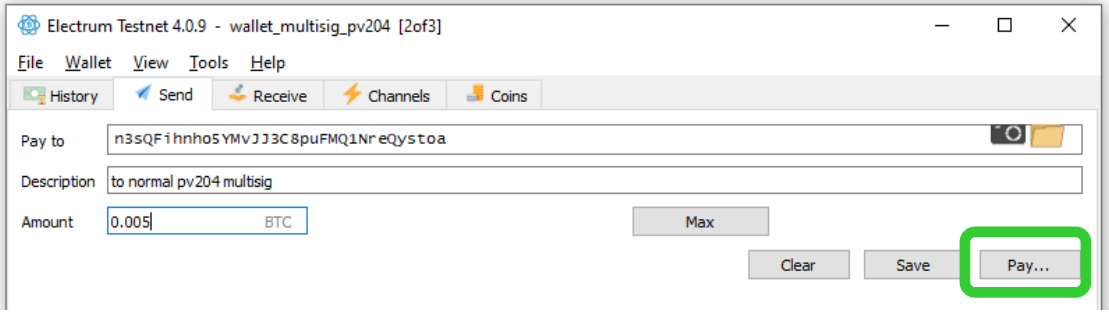
- Generate receive address on multisig, send to it from normal one
- Optional: try using coin control
  - View → Show coins, RClick on target coin → Spend
  - Max button in Send will only take marked coin(s)

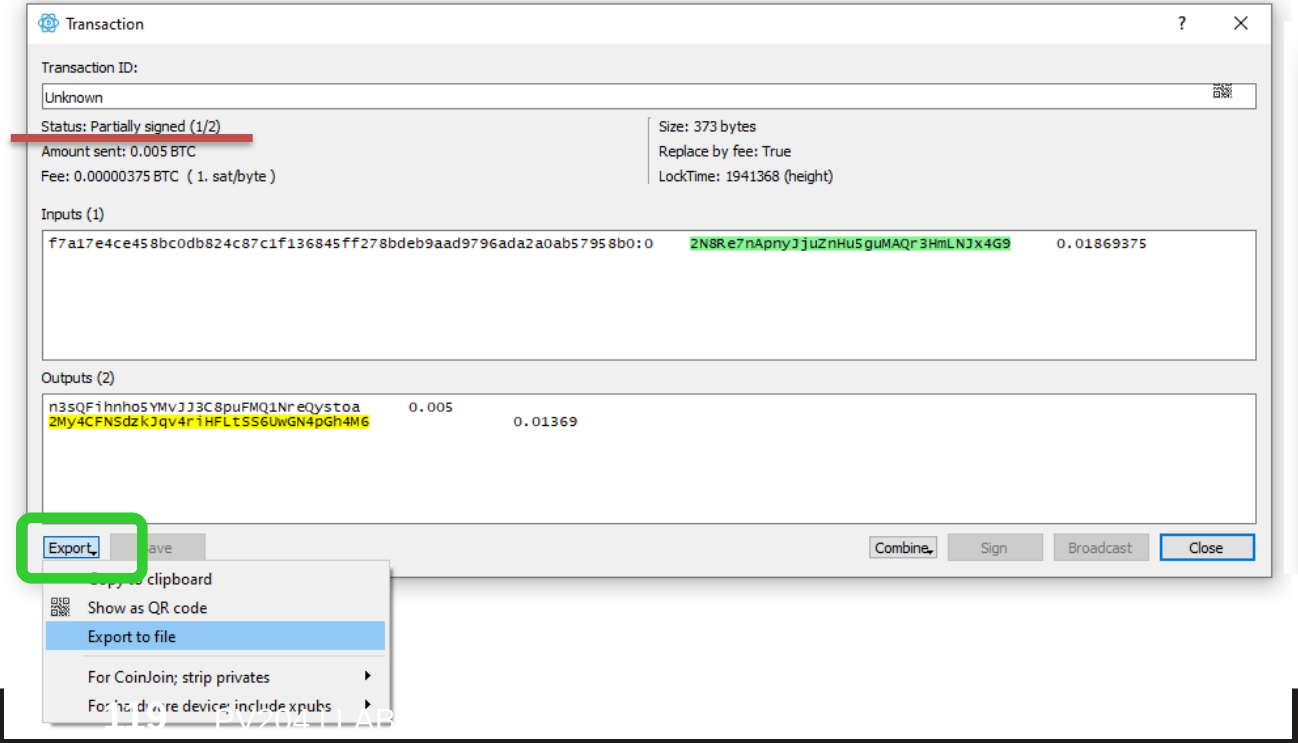


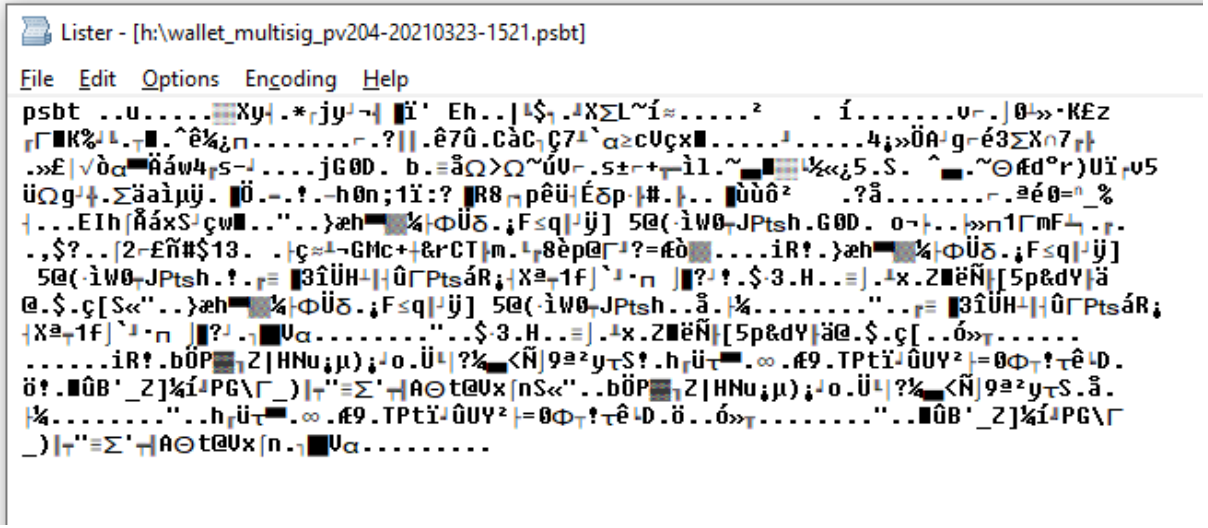


Send from multisig wallet to normal one – first signer

- Generate receive address on normal wallet
- One signer creates transaction
  - Save button saves partially prepared tx locally
  - Pay button signs (partially) transaction, allows to Export

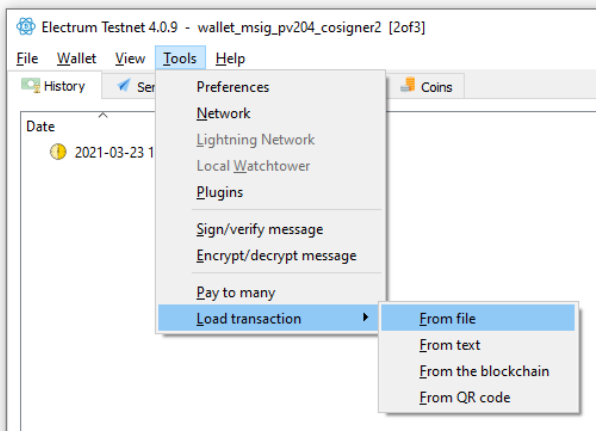


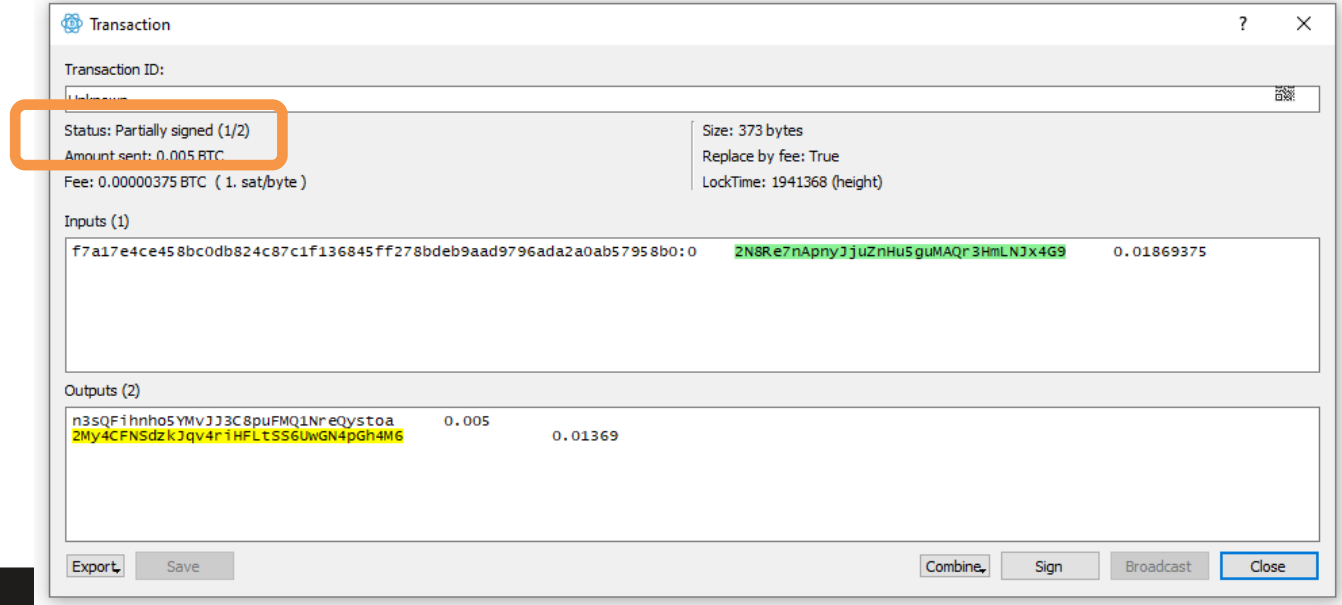


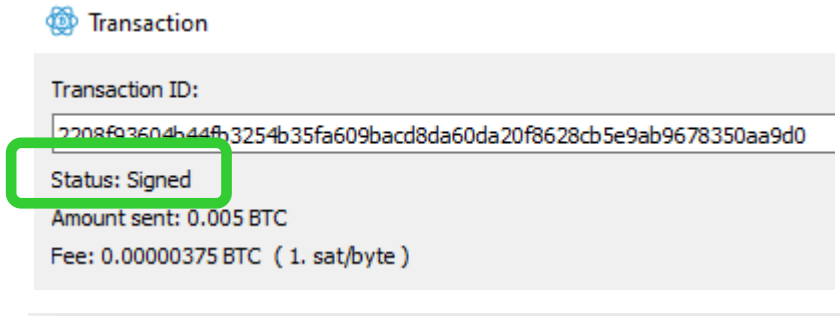


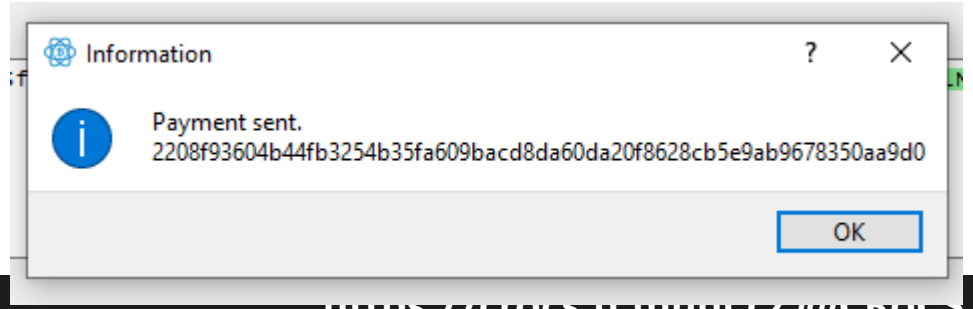
Send from multisig wallet to normal one – second signer

- Open cosigner's wallet
- Tools→Load transaction→From file
- Check target info and amount
- Sign loaded transaction
- Broadcast to network









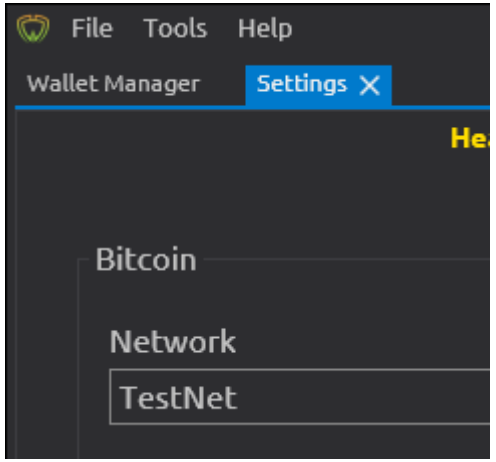
Questions

- Analyze your transactions via blockchain explorer
  - E.g., https://blockstream.info/testnet/
  - TX (from normal to multisig wallet)
    - Can you figure that transaction was from normal to multisig?
    - If yes/no – what is the advantage / disadvantage?
  - TX (from multisig to normal wallet)
    - Can you recognize that input was multisig? How and Why?
  - How much was possible to save in fees by using segwit instead of legacy address?
- Which option is better for backup (not loosing possibility to spend)? 1-of-3 or 3-of-3?
- Which option is better against and attacker (prevent him to spend your coins)? 1-of-3 or 3-of-3?
- What are advantages and disadvantages of 2-of-3 vs. 3-of-5?

Note: This tutorial is now outdated, Wallet Wasabi 1.0 and related coinjoin mix is going to be retired soon (2024). Wallet Wasabi 2.x is the main production replacement

WASABI WALLET 1.X

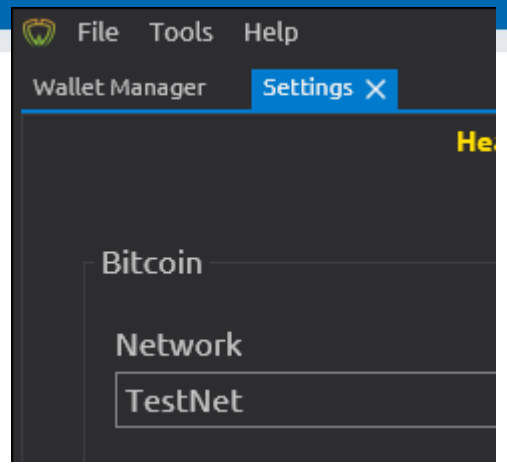
Wasabi wallet (testnet)



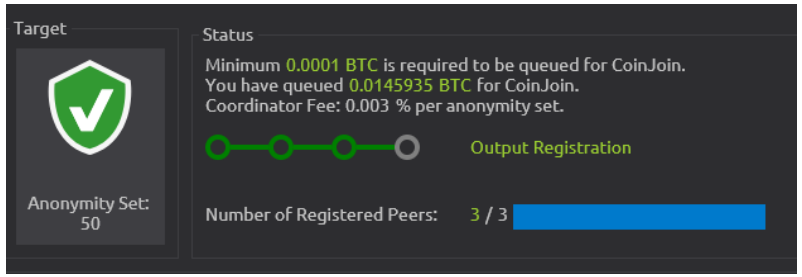
- Solo task (1 students / breakout room)
- Install Wasabi wallet from https://wasabiwallet.io/
  – For real use, verify PGP signature
- Start it, go to Settings and change Network to TestNet
- Restart application
- Generate new Wallet
  – Backup seed, password is used to encrypt seed (if none, what it means?)
- Wasabi forces you to set coin label (Why?)
- Send some sats to Wasabi wallet from your normal testnet wallet

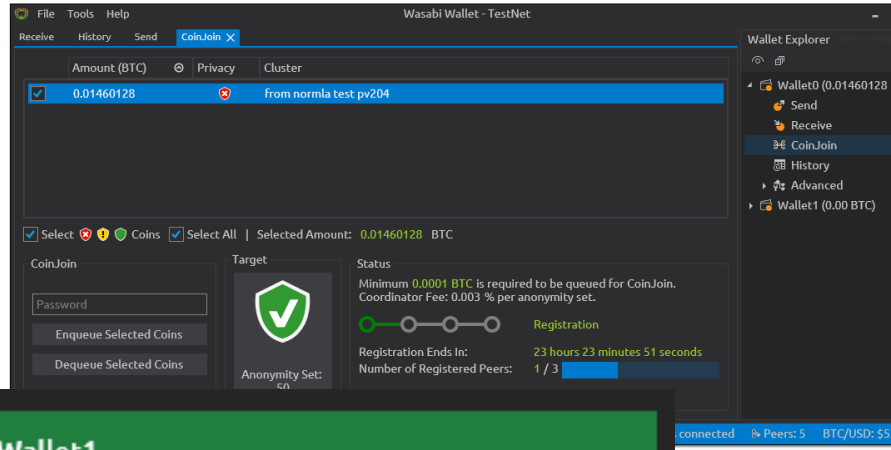
COINJOIN WITH WASABI WALLET

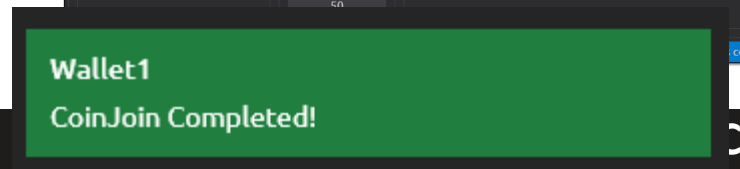
Wasabi wallet – participating in CoinJoin



- Visit CoinJoin option
  - Change Target to Anonymity Set: 2 (so mixing finish quickly)
    - For real use, keep it 50!
  - Enqueue Selected Coins into next round of CoinJoin
- Waits until registered and confirmed
- Keep your computer running
  - The protocol is interactive, requires several rounds
- What have you got at the end?
- Investigate txid on chain explorer
  - Use Tor, otherwise you will leak IP to TX mapping







File Tools Help — Wasabi Wallet - TestNet

Receive | History | Send | CoinJoin | CoinJoin | Receive

	Status	Amount (BTC)	Privacy	Cluster
☐	waiting for confirmation	0.00018	!	from normla test pv204
☐	waiting for confirmation	0.00018238	!	from normla test pv204
☐	waiting for confirmation	0.00036	!	from normla test pv204
☐	waiting for confirmation	0.00072	!	from normla test pv204
☐	waiting for confirmation	0.00144	!	from normla test pv204
☐	waiting for confirmation	0.01171112	✖	from normla test pv204

☐ Select ✖ ! ✓ Coins ☐ Select All

CoinJoin

Password

Enqueue Selected Coins

Dequeue Selected Coins

Target

Anonymity Set: 50

Status

Minimum 0.0001 BTC is required to be queued for CoinJoin.
You have queued 0.0145935 BTC for CoinJoin.
Coordinator Fee: 0.003 % per anonymity set.

Registration

Registration Ends In: 23 hours 6 minutes 51 seconds

Number of Registered Peers: 2 / 3

Wallet Explorer

- Wallet0 (0.0145935 BTC)
  - Send
  - Receive
  - CoinJoin
  - History
  - Advanced
- Wallet1 (0.00499033 BTC)
  - Send
  - Receive
  - CoinJoin
  - History
  - Advanced

Ready | Tor is running | Backend is connected | Peers: 7 | BTC/USD: $55671

BLOCKCHAIN EXPLORERS