21/Bitcoin meetup #22
“Bitcoin miniscript”

Prezentace

18:10 – 19:10

Pauza

19:10 – 19:30

Tombola + Diskuze

19:30

Přednášející

JEDNA DVACET Praha

Vít Obrusník
Firmware Tech Lead

Jan Šetina
Firmware Product Lead

⚡ombola – 2500 sats

1. Vexl merch
2. Trezor merch

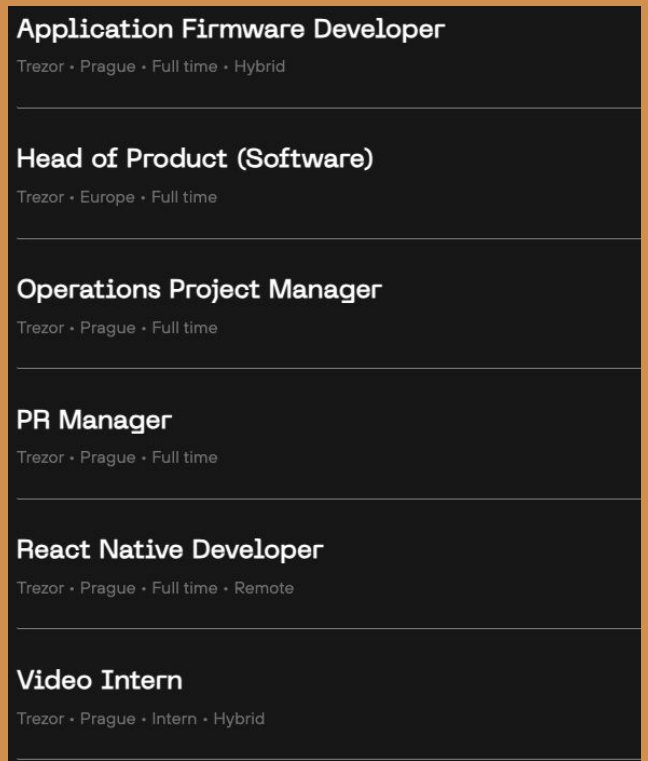
Nadcházející meetupy

- ST 4.3. Suite Sync + Dan Steigerwald
- Duben MeshCore + Stick
- ČT 21.5 Pizza & Grill Day 🍕

Komerční vložka

Vexl – Head of Marketing & Communications

Trezor –

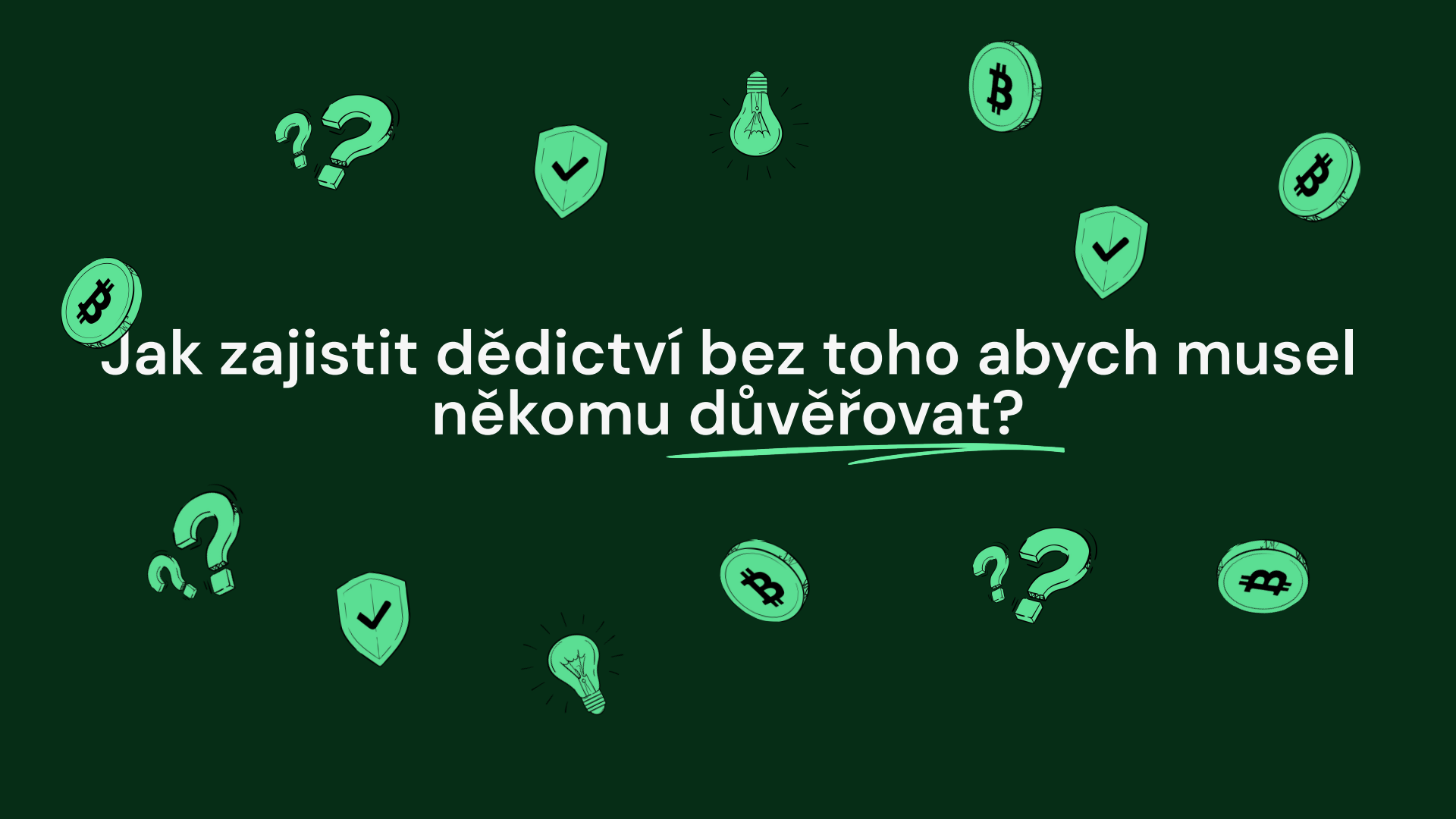


TREZOR

Miniscript

Jan Šetina, Vít Obrusnik

Co se stane s vašimi bitcoiny pokud zítra zmizíte?


Jak zajistit dědictví bez toho abych musel
někomu důvěřovat?

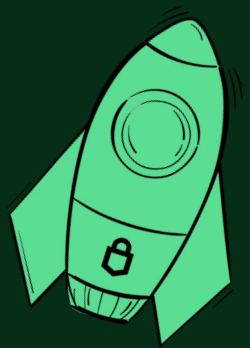
Já můžu utratit kdykoli

nebo

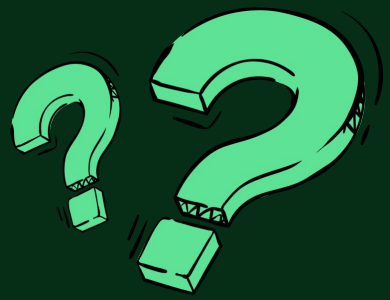
Druhý klíč může utratit po 6 měsících

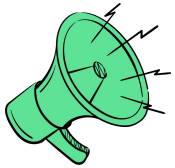
Lze i teď, ale

- velmi složité na napsání v nativní vrstvě bitcoinu (bitcoin script)
- obtížně auditovatelné; snadno se udělá chyba
- pro složitější věci násobně náročnější



CO VLASTNĚ CHCEME?





Chceme smart kontrakty (scénáře), které jsou:

1. auditovatelné
2. bezpečné
3. standardizované
4. srozumitelné uživateli

Adresa


bc1quxwuhgd97s95l6rcvm2uya25fsndvf8ru490vyahed6g2l9fx4jqt0xtq6
prefix
checksum
locking script

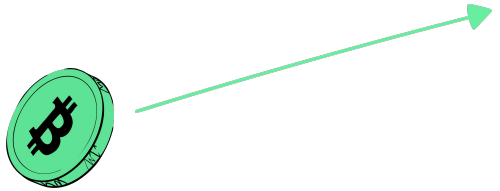
Evaluace scriptu

bc1quxw…qt0xtq6

Evaluace scriptu

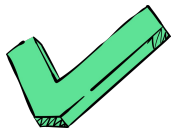
locking script

Evaluace scriptu

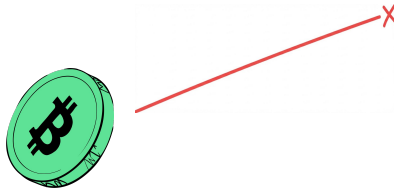


`unlocking script  locking script`

= True



Evaluace scriptu



`unlocking script  locking script`

= False

Script

7 se rovná 3+4

<OP_7> <OP_3 OP_4 OP_ADD OP_EQ>

Script - standardní

```
tady je důkaz že  vlastním tento klíč
```

Script - standardní

- Taproot (P2TR) (bc1p…)
- SegWit (P2WPKH) (bc1q…)
- Legacy SegWit (P2SH-P2WPKH) (3…)
- Legacy (P2PKH) (1…)

- `OP_1 <x_only_pubkey>`
- `OP_0 <pubkey_hash>`
- `OP_HASH160 <script-hash> OP_EQUAL`
- `OP_DUP OP_HASH160 <hash> OP_EQUALVERIFY OP_CHECKSIG`

Script - komplexní

```
Alice OP_CHECKSIG OP_IFDUP OP_NOTIF Bob OP_CHECKSIGVERIFY
f003 OP_CSV OP_ENDIF
```

Alice může poslat kdykoliv, ale Bob až po týdnu.

Miniscript

- bezpečný subset Bitcoin Scriptu
- deklarativní (ne imperativní)
- umožňuje snadnou kompozici
- 3 úrovně
  - Policy
  - Miniscript
  - Script

Miniscript - Policy

- klíče
  - `pk(NAME)`
- timelocks
  - `older(NUM)`
  - `after(NUM)`
- kompozice
  - `and(POL1, POL2)`
  - `or(POL1, POL2)`
  - `thresh(NUM, POL1, POL2, ...)`

Policy - standardní

```
pk(Alice)
```

```
Alice OP_CHECKSIG
```

Policy - Multisig

```
and(
  pk(Alice),
  pk(Bob)
)
```

```
Alice OP_CHECKSIGVERIFY
Bob OP_CHECKSIG
```

```
thresh(
  2,
  pk(Alice),
  pk(Bob),
  pk(Charlie)
)
```

```
OP_PUSHNUM_2 Alice
Bob Charlie
OP_PUSHNUM_3
OP_CHECKMULTISIG
```

Policy - jednoduché dědictví

```
or(
  pk(Alice),
  and(
    pk(Bob),
    older(1008)
  )
)
```

```
Alice OP_CHECKSIG
OP_IFDUP OP_NOTIF
Bob
OP_CHECKSIGVERIFY
f003 OP_CSV
OP_ENDIF
```

Alice může poslat kdykoliv, ale Bob až po týdnu.

Policy - dědictví s multisigem

```
or(
  pk(Alice),
  and(
    thresh(
      2,
      pk(Bob),
      pk(Charlie),
      pk(Eva)
    ),
    older(1008)
  )
)
```

```
Alice OP_CHECKSIG
OP_IFDUP OP_NOTIF
OP_PUSHNUM_2 Bob
Charlie Eva
OP_PUSHNUM_3
OP_CHECKMULTISIGVERIFY
f003 OP_CSV OP_ENDIF
```

Policy - dědictví s multisigem opt. pro Alici

```
or(
   95@pk(Alice),
   and(
      thresh(
         2,
         pk(Bob),
         pk(Charlie),
         pk(Eva)
      ),
      older(1008)
   )
)
```

Alice OP_CHECKSIG OP_IFDUP OP_NOTIF OP_DUP OP_HASH160 9a1c78a507689f6f54b847ad1cef1e614ee23f1e OP_EQUALVERIFY OP_CHECKSIG OP_TOALTSTACK OP_DUP OP_HASH160 4f99bbf75707e44bc2afa65337dece914e817aac OP_EQUALVERIFY OP_CHECKSIG OP_FROMALTSTACK OP_ADD OP_TOALTSTACK OP_DUP OP_HASH160 1e22c04e0df917ffa2a3bd456968430426886e11 OP_EQUALVERIFY OP_CHECKSIG OP_FROMALTSTACK OP_ADD OP_PUSHNUM_2 OP_EQUALVERIFY f003 OP_CSV OP_ENDIF

Reálné použití miniscriptu

Miniscript je způsob, jak standardně a bezpečně popsat:

- Kdo může utratit
- Za jakých podmínek
- S jakým recovery scénářem

1. Pokročilé Multisig & Recovery
   - např. vytvoření obnovovacích klíčů, které se aktivují pouze po nečinnosti původních klíčů
2. Dědictví (timelock recovery)
   - jasné definování dědičné cesty
   - bezpečné implementace timelocku
   - jednoznačné vysvětlení uživateli
3. Privacy u složitějších skriptů
   - Jednotný způsob konstrukce skriptů
   - možnost skrýt větve policy

problem No. 1

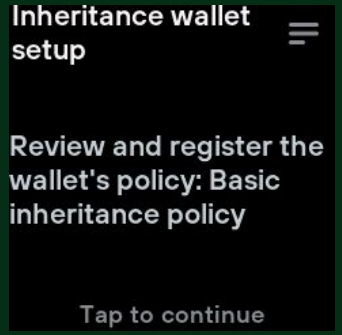
- nestandardnost scriptů
- generace adres dle daných instrukcí

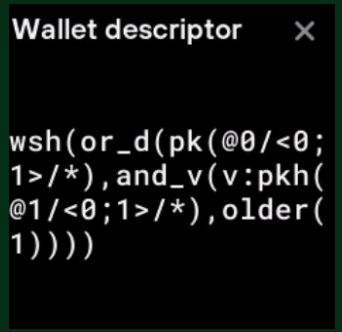
→ potřeba uchovávat policy

- na plnou obnovu již nestačí pouze seed!!!
  - potřeba zálohovat jednotlivé policy

→ potřeba ověřit policy na zařízení

- registrace policy na zařízení





problem No. 2

```
OP_IF
   OP_2
      <PubKeyA> <PubKeyB> <PubKeyC>
   OP_3
   OP_CHECKMULTISIG
OP_ELSE
   <25920> OP_CHECKSEQUENCEVERIFY
OP_DROP
   OP_1
      <PubKeyD> <PubKeyE>
   OP_2
   OP_CHECKMULTISIG
OP_ENDIF
```

```
or_i(
  multi(2, A, B, C),
  and_v(
    older(25920),
    multi(1, D, E)
  )
)
```

Wallet descriptor

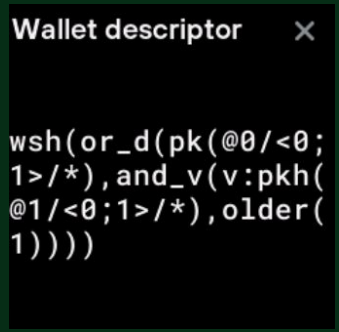
```
wsh(or_d(pk(@0/<0;
1>/*),and_v(v:pkh(
@1/<0;1>/*),older(
1))))
```

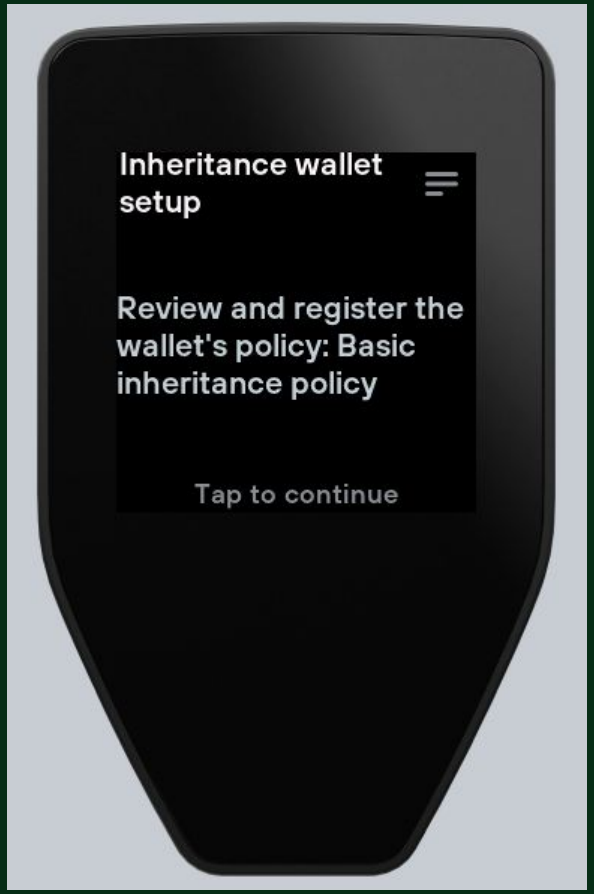
problem No. 2

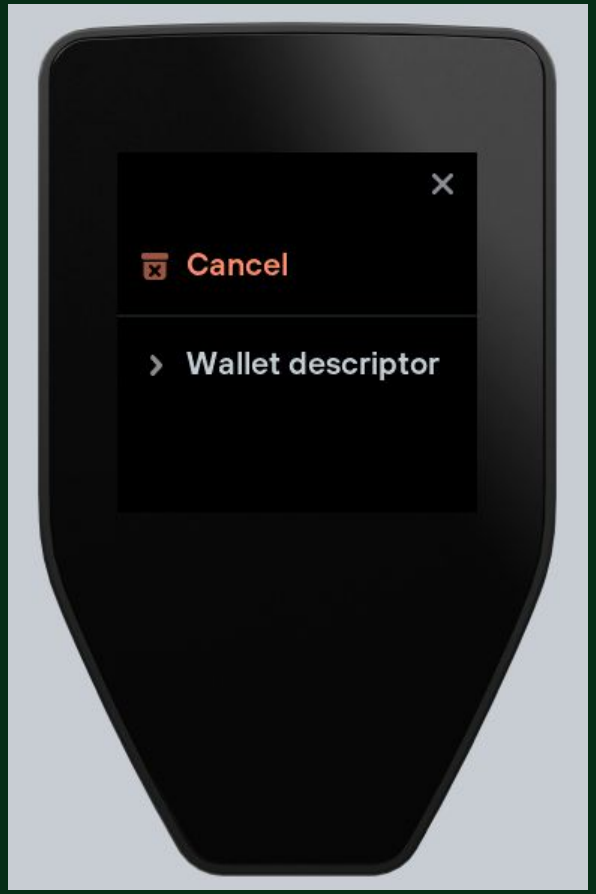
standardní spend: 2-of-3 multisig (A,B,C)
kdykoliv

recovery spend: po timelocku (6 měsíců)
stačí 1-of-2 (D,E)

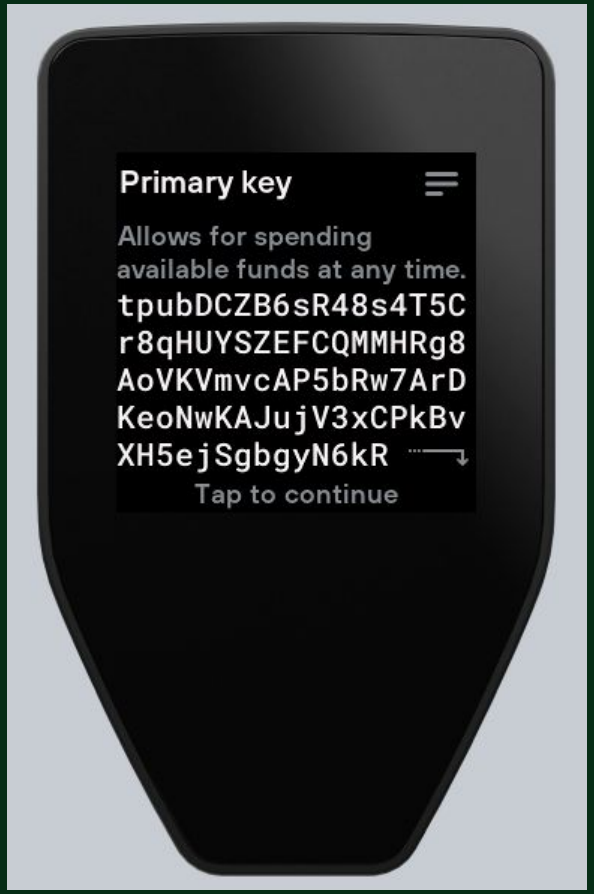
```
or_i(
  multi(2, A, B, C),
  and_v(
    older(25920),
    multi(1, D, E)
  )
)
```

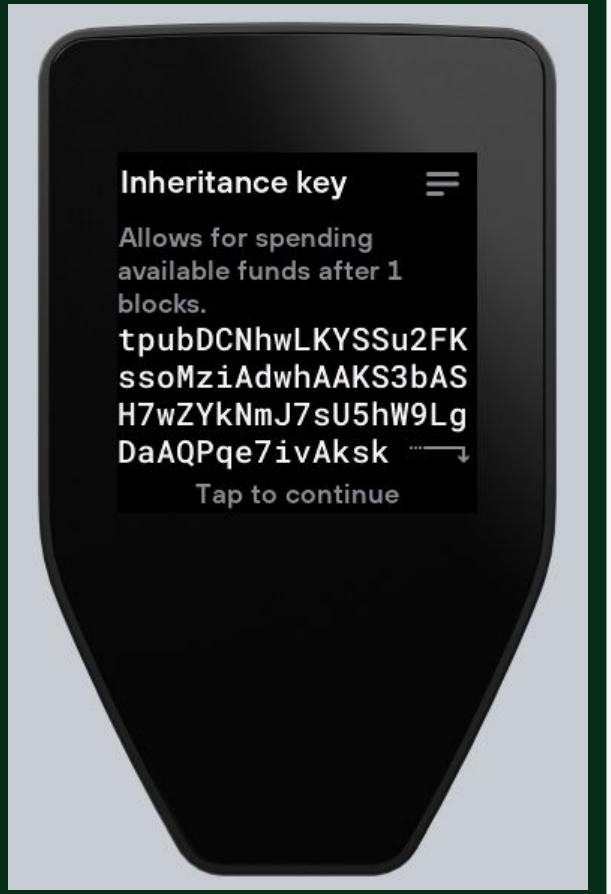



Inheritance wallet setup
Review and register the wallet's policy: Basic inheritance policy
Tap to continue


Cancel
Wallet descriptor


Wallet descriptor
wsh(or_d(pk(@0/<0;1>/*),and_v(v:pkh(@1/<0;1>/*),older(1))))


Primary key
Allows for spending available funds at any time.
tpubDCZB6sR48s4T5Cr8qHUYSZEFCQMMHRg8AoVKVmvcAP5bRw7ArDKeoNwKAJujV3xCPkBvXH5ejSgbgyN6kR
Tap to continue


Inheritance key
Allows for spending available funds after 1 blocks.
tpubDCNhwLKYSSu2FKssoMziAdwhAAKS3bASH7wZYkNmJ7sU5hW9LgDaAQPqe7ivAksk
Tap to continue


Basic inheritance policy
Testnet
tb1qzvr7ptes6kq2ee0745a7h2n639etfz43nsz9d2jn8u6wz8egx0hqnr5pza
Tap to continue

Díky