chaincode

Schnorr Signatures
Taproot Constructions

Elichai & James

Part 1- Bitcoin Core Taproot Branch

Part 2 - Interactive Toolkit Documentation (Jupyter)

chaincode

Schnorr vs ECDSA

ECDSA - NIST/ANSI (1997) - $sk = e + kG_x d$

Schnorr Signatures (1991) - $s = k + ed$

$$Sig(s, kG)$$

$$Sig(s, R)$$

Glossary

m - Message.

$e = H(m)$

d = Private Key.

k = Random nonce

G = Generator Point.

Point = scalar*G = (x,y)

Public key = dG

Multi Signatures (e.g. MuSig)

$$P_1 = d_1 G, \quad P_2 = d_2 G$$

$$s_1 = k_1 + e d_1, \quad s_2 = k_2 + e d_2$$

$$s_1 + s_2 = (k_1 + k_2) + e(d_1 + d_2)$$

$$s' = k' + e d'$$

$$P' = (d_1 + d_2) G$$

Glossary

m - Message.

$$e = H(m)$$

d = Private Key.

k = Random nonce

G = Generator Point.

Point = scalar*G = (x,y)

Public key = dG

Pay to Contract (e.g. Taproot)

$$P' = P + H(P||s)G$$

$$d' = d + H(P||s)$$

Glossary

m - Message.

$$e = H(m)$$

d = Private Key.

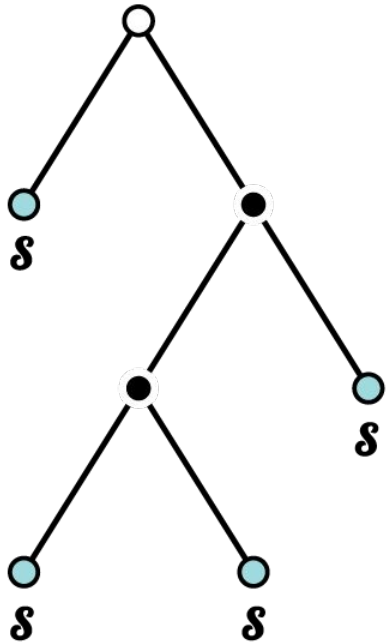
k = Random nonce

G = Generator Point.

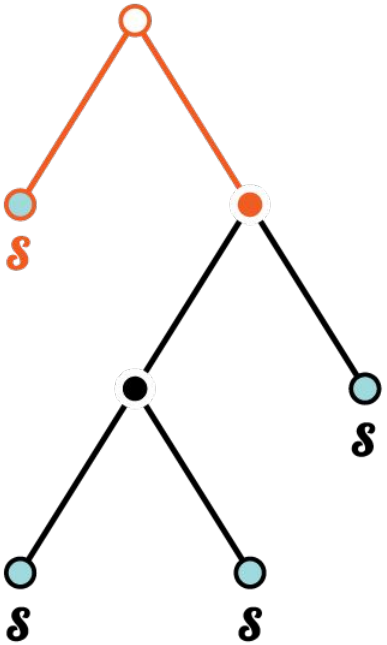
Point = scalar*G = (x,y)

Public key = dG

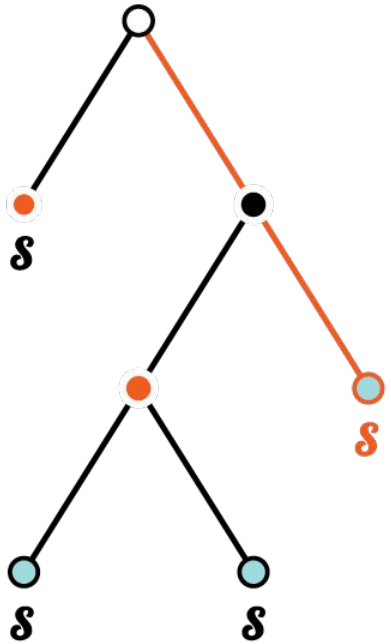
Merkle Branches



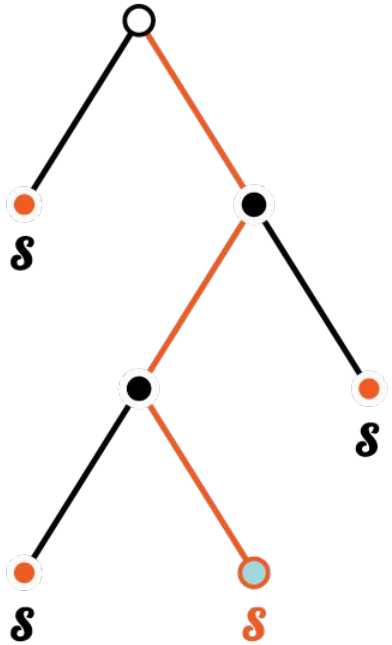
Merkle Branches



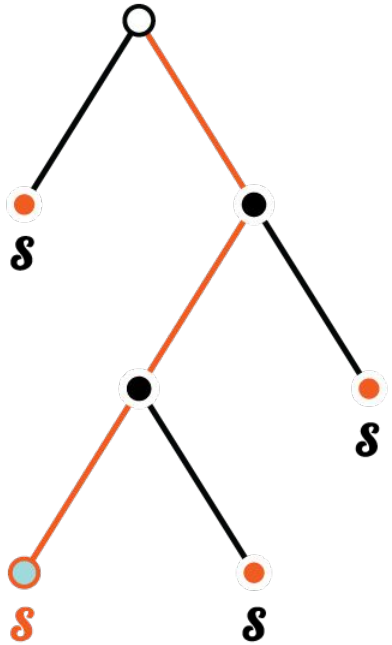
Merkle Branches



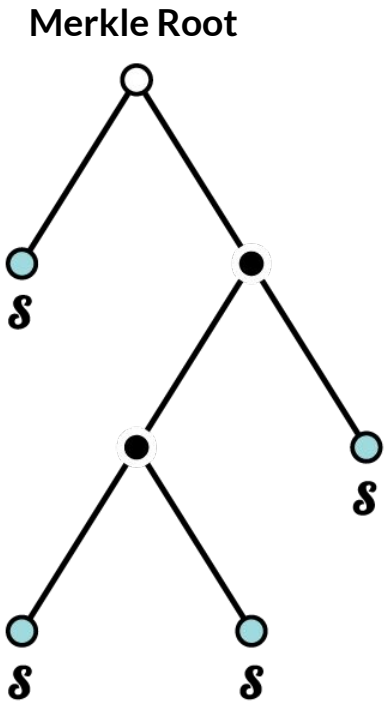
Merkle Branches



Merkle Branches



Taproot



$$P' = P + H(P||MerkleRoot)G$$

Glossary

m - Message.

d = Private Key.

k = Random nonce

G = Generator Point.

Point = scalar*G = (x,y)

Public key = dG

Part 1 - Introduction to Schnorr & Taproot

Part 2 - Optech Taproot Developer Toolkit

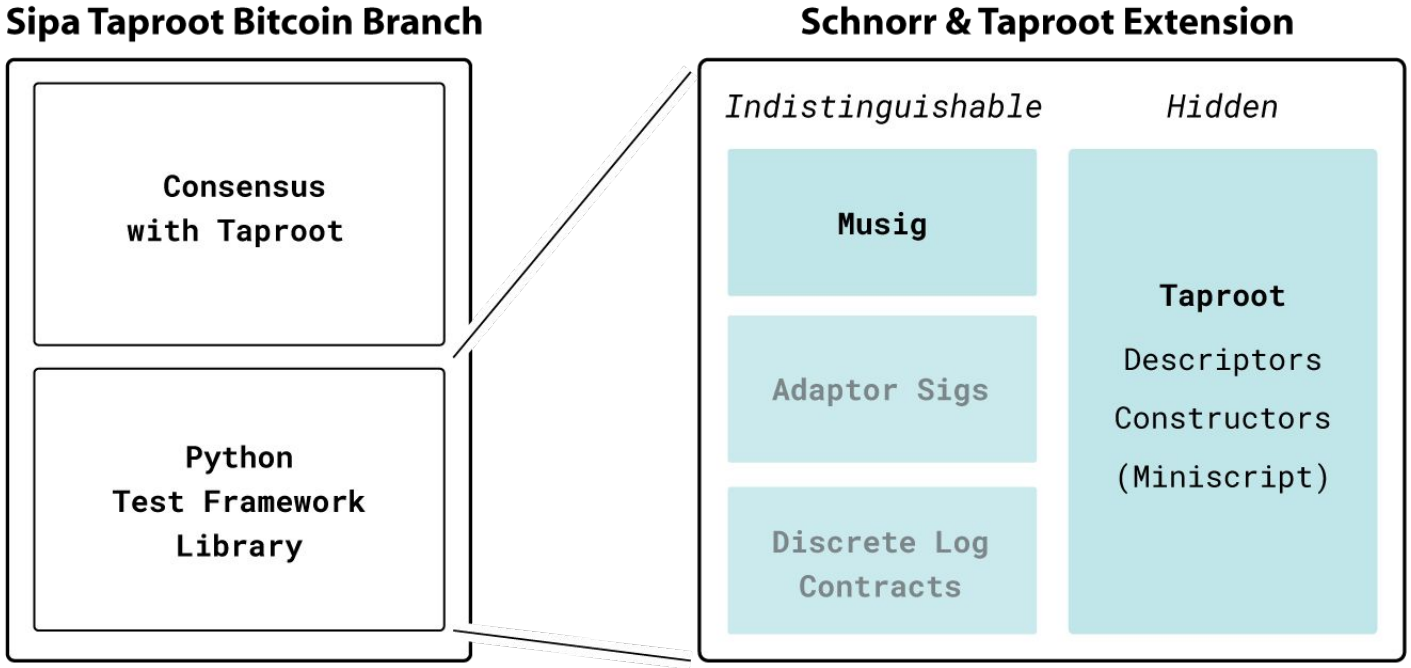
chaincode

Links to Taproot Developer Toolkit

Bitcoin Core Taproot Branch (v0.1)

Interactive Toolkit Documentation (Jupyter)

Bitcoin Core Taproot Branch



Bitcoin Core (Taproot) + Jupyter

bitcoinops/taproot-workshop

- 1.0-Workshop-Setup.ipynb
- 1.1-Introduction-to-Schnorr.ipynb
- 1.2-Introduction-to-Musig.ipynb
- ...
- **Solutions**
  - 1.1-Introduction-to-Schnorr-Solutions.ipynb
  - 1.2-Introduction-to-Musig-Solutions.ipynb
  - ...

bitcoinops/bitcoin/tree/optech-taproot

- src/bitcoind
- test/functional/test_framework
  - key.py
  - messages.py
  - script.py
  - util.py
  - ...

Jupyter Notebook —*imports*→ **Bitcoind + TestFramework**

chaincode

Examples:

Interactive Taproot Toolkit Documentation

chaincode

Become a Schnorr & Taproot Expert

You will learn:

- Bitcoin Core Python Library
- Schnorr & Taproot Development

chaincode

Become a Schnorr & Taproot Expert

You will learn:

- Bitcoin Core Python Library
- Schnorr & Taproot Development

I will help you (if you sign up now)

- Slack: James C.
- james@teachbitcoin.io

chaincode

Thank you - questions?

chaincode