21Lectures

Bitcoin Masterclass

Taproot & MuSig2

Day 2

Taproot

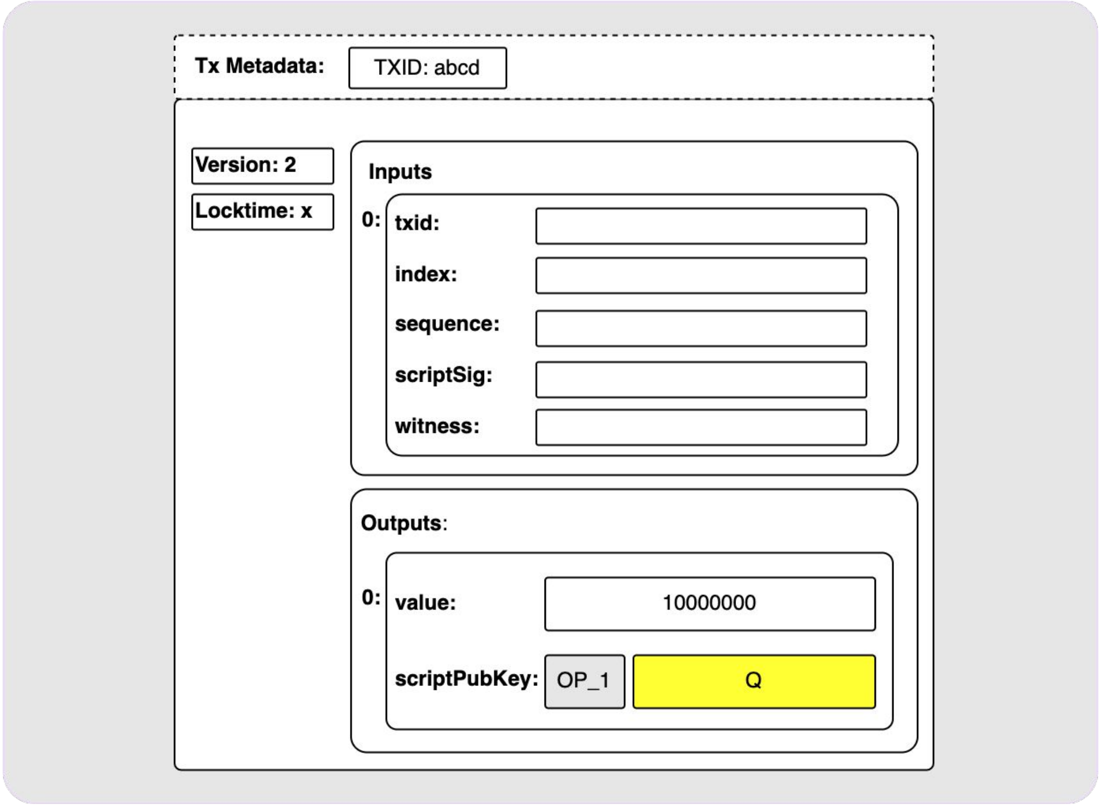
Was activated together with Schnorr signatures

Provides a way of

- Providing a multitude of spending conditions
- In a hidden way
- Such that only the one used is revealed
- Or none at all if a “default-pubkey-condition” can be used

https://ellemouton.com/posts/taproot-prelims/

A Taproot output



Tx Metadata: TXID: abcd

Version: 2

Locktime: x

Inputs

0: txid:

index:

sequence:

scriptSig:

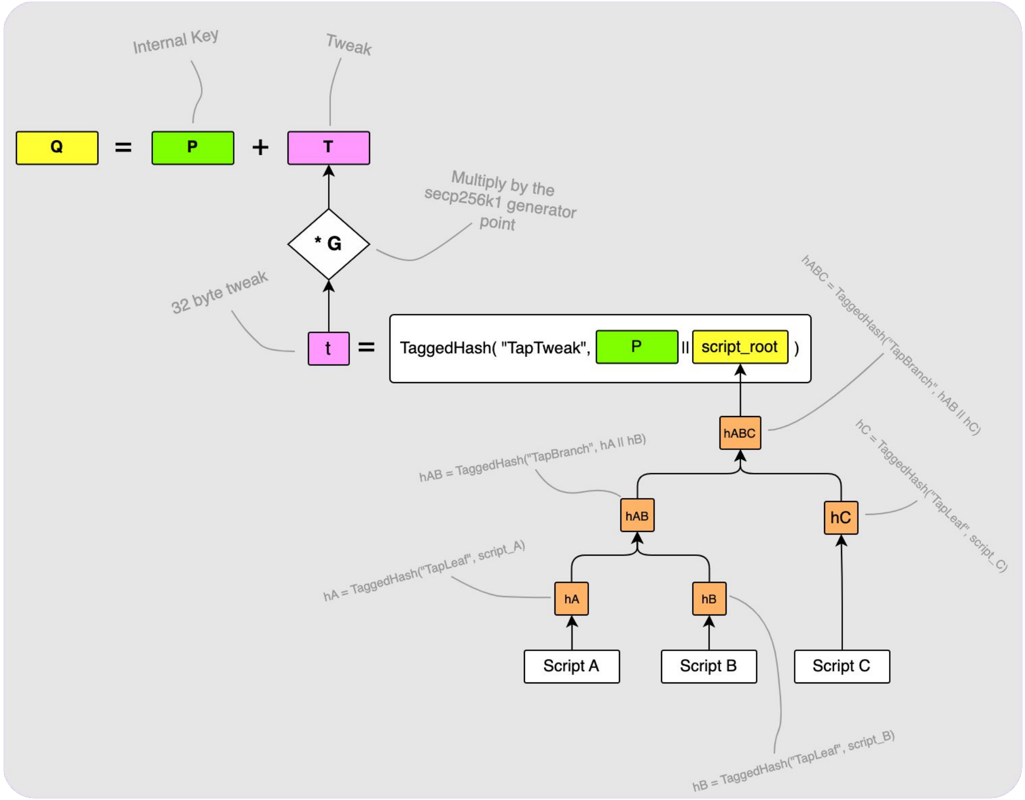
witness:

Outputs:

0: value: 10000000

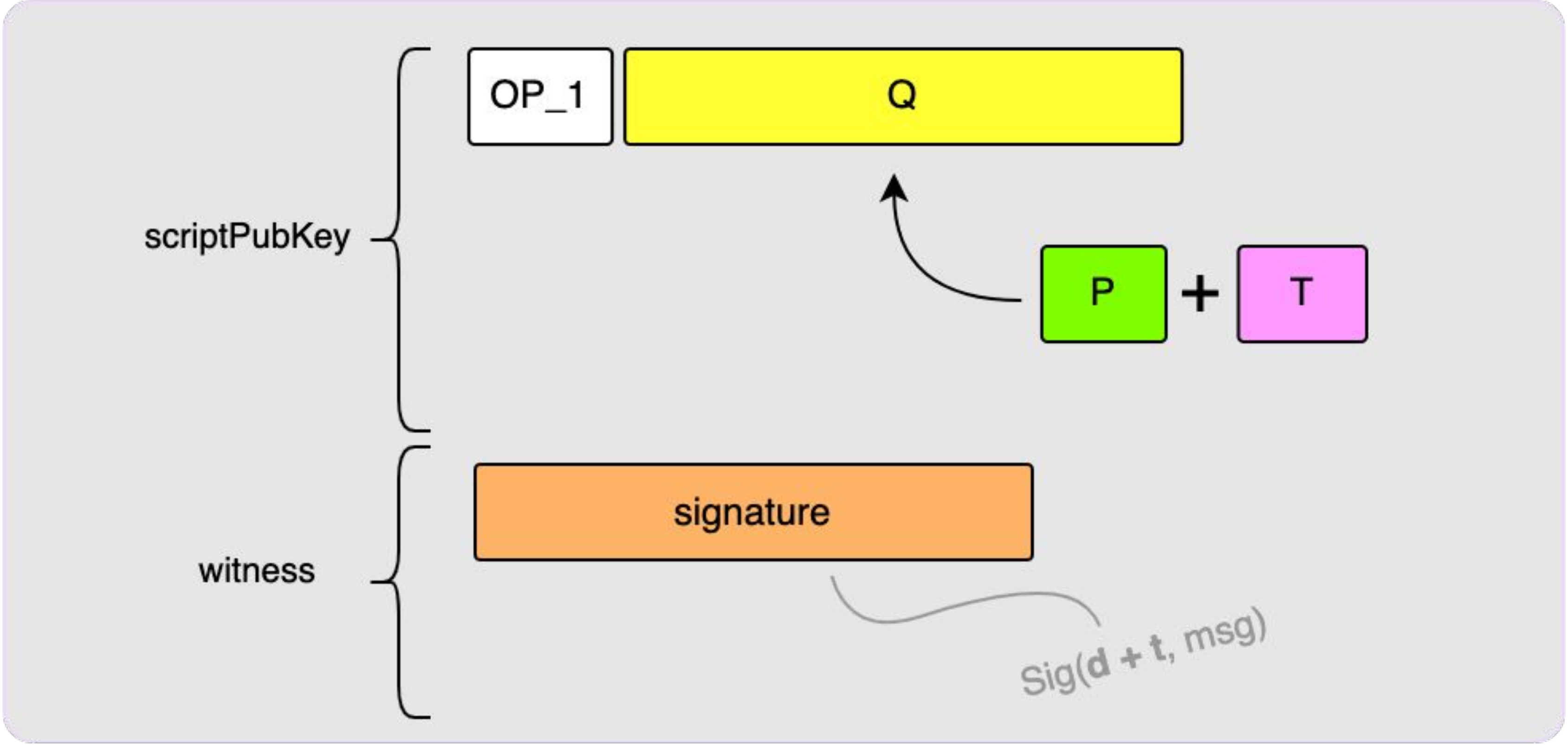
scriptPubKey: OP_1 Q

Constructing a Taproot Output


Internal Key
Tweak
Q
=
P
+
T
Multiply by the secp256k1 generator point
* G
32 byte tweak
t
=
TaggedHash("TapTweak", P || script_root)
hABC = TaggedHash("TapBranch", hAB || hC)
hABC
hAB = TaggedHash("TapBranch", hA || hB)
hAB
hC
hC = TaggedHash("TapLeaf", script_C)
hA = TaggedHash("TapLeaf", script_A)
hA
hB
Script A
Script B
Script C
hB = TaggedHash("TapLeaf", script_B)

Quiz: Why is P in the tagged hash?

Spending a Taproot output: Key-Path Spend

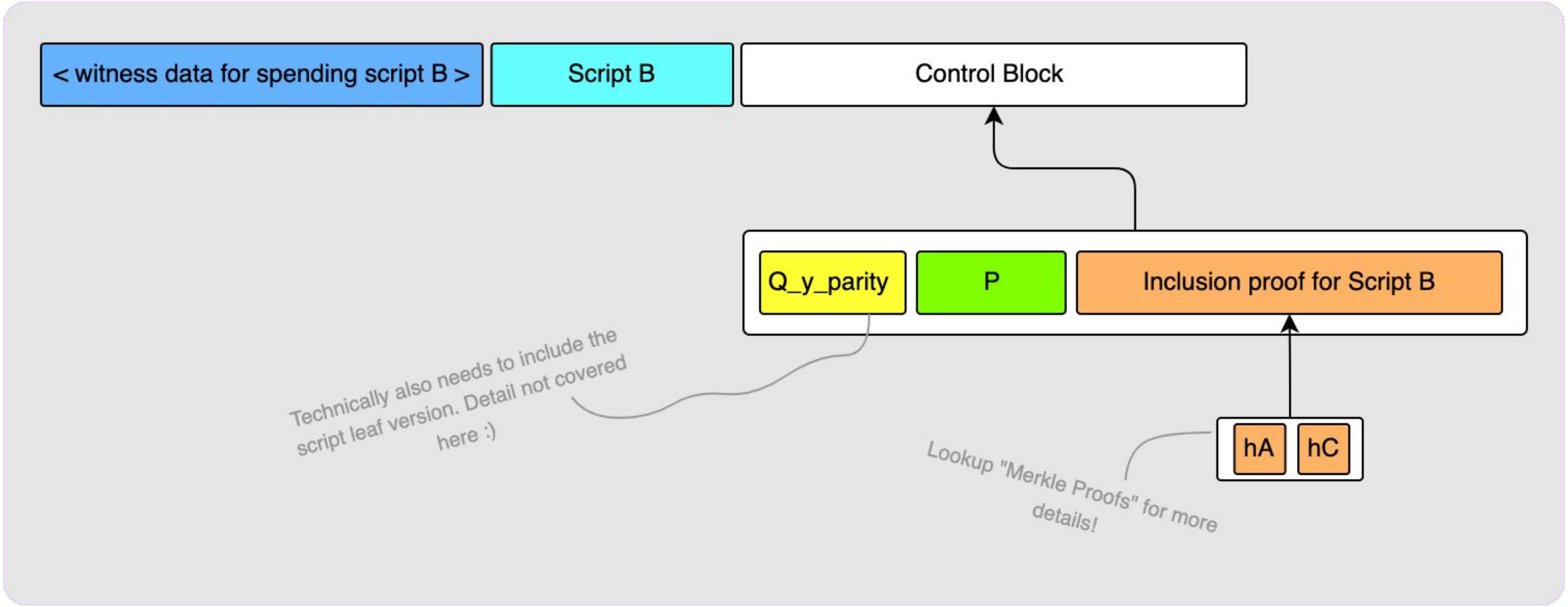


Why is the key path spend so useful?

Often, protocols like LN have a default “everybody-agrees” spending condition.

See Musig2 later!

Spending a Taproot output: Script-Path Spend



< witness data for spending script B >

Script B

Control Block

Q_y_parity

P

Inclusion proof for Script B

hA

hC

Technically also needs to include the script leaf version. Detail not covered here :)

Lookup "Merkle Proofs" for more details!

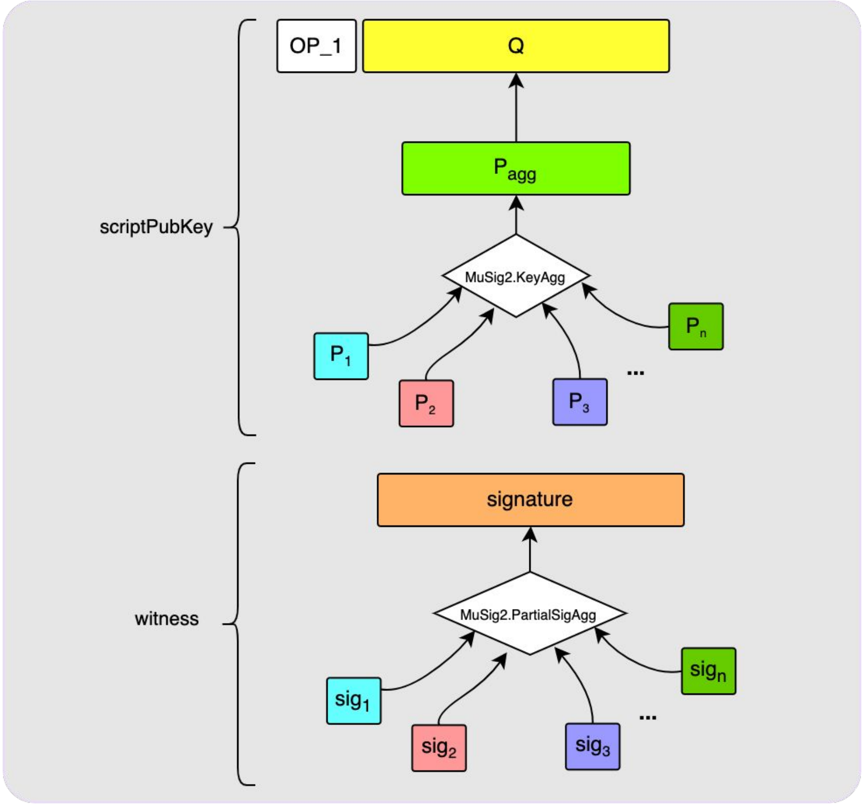
Musig2

Schnorr verification equation:

S = R + H (R | P | m) * P

Linear in all public keys, just sum everything up (use summed points in hash)! => Blackboard

Caution: Naïve approach is horribly broken!



Musig2

Multisig that is indistinguishable from a single-key signature.

Perfect for an “everybody-agrees” default path.

Details: https://github.com/bitcoin/bips/blob/master/bip-0327.mediawiki