Taproot Spending rules

Indice

- O problema
- Definicao
- Tree Organizacao e balanceamento.
- Tagged hash
- Tweak
- tweak(Estrutura merkle) + pub => address
- Spend by key path
- Spend by script path

O problema ❗

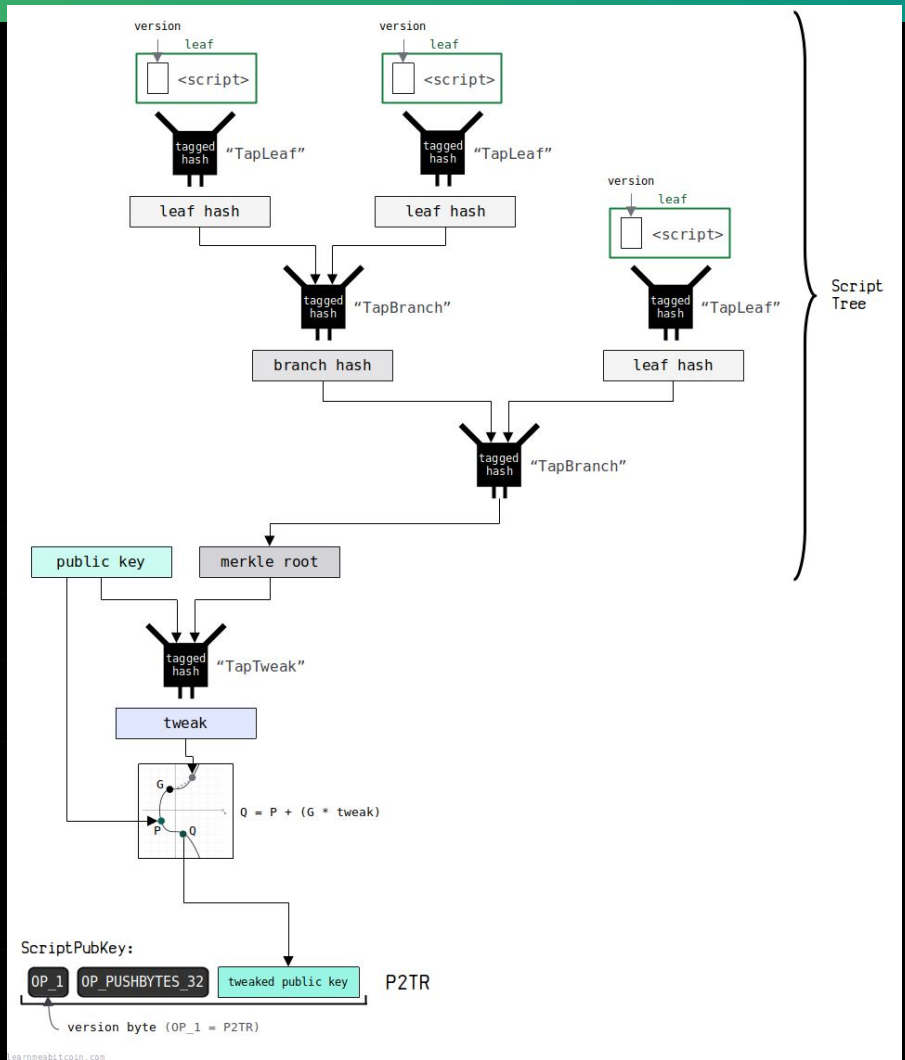
Anteriormente a nossa melhor opção era o Pwsh para construirmos multsigs, que nos causavam dois problemas, primeiro e não tínhamos privacidade com relação ao tipo e funcionamento do script, e segundo era limitado a apenas um corpo de script.

Taproot 

Em Síntese Taproot e uma forma de se colocar miniscripts em uma arvore organizada, sendo que cada ramo seria uma possibilidade de gasto, sendo necessário revelar este ramo apenas na hora de gastá-lo e somente o ramo usado no gasto mantendo os ramos alternativos ocultos.

Árvore🌳

Organização.



Tagged hash

O tagged hash consiste em um hash com uma tag de prefixo.

Tweak = tagged_hash(“TapTweak”, public_key,merkle_root)

Tweak point 🎯

Calcular o tweak point consiste em apenas realizar uma operação tagged hash com a public key e a merkle root

Tweak = tagged_hash(“TapTweak”, public_key,merkle_root)

Gerando a tweakpubkey

Q = P + (G x Tweak)

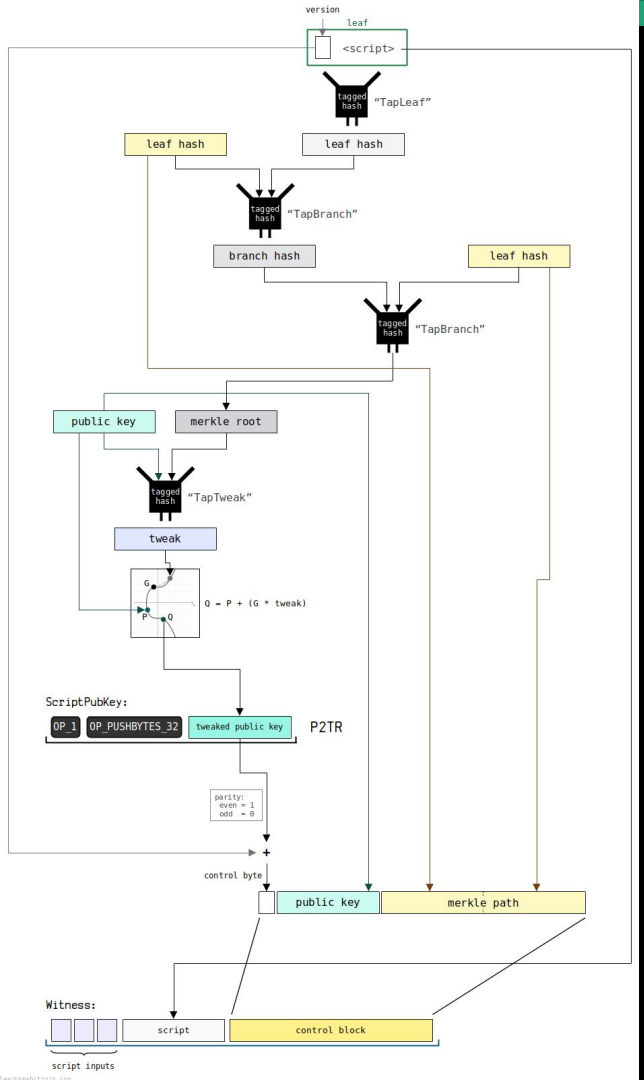
Sendo:

Q = o valor final

P = a chave publica(neste caso com a merkle root)

G = o ponto gerador

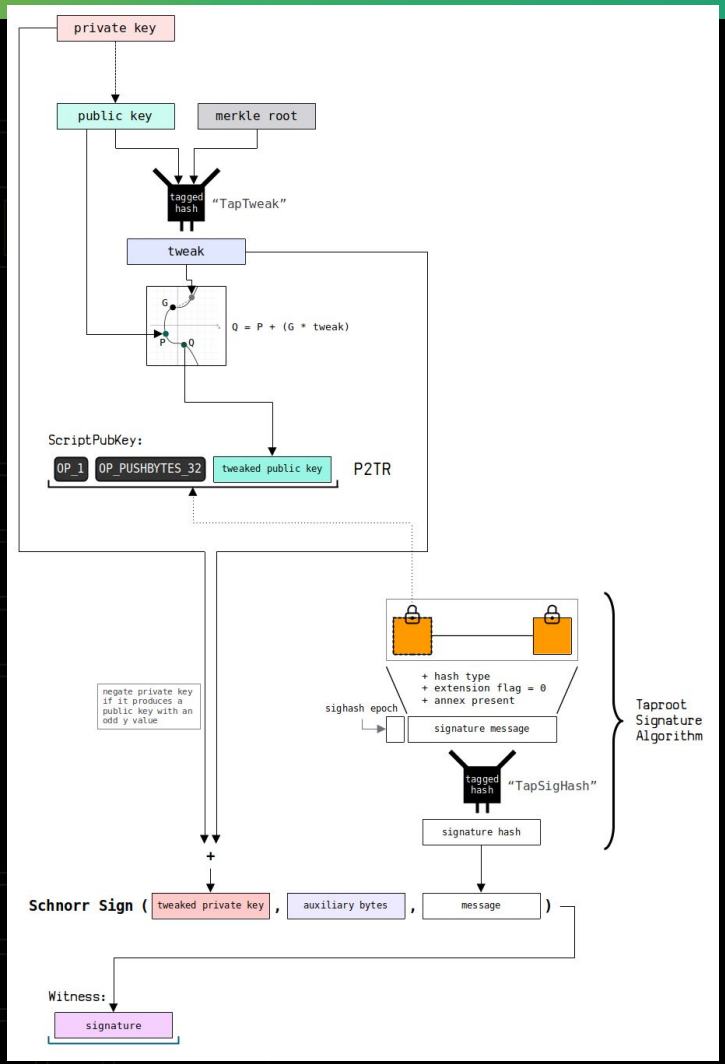
Tweak = O valor gerado anteriormente


version
leaf
<script>
tagged hash
"TapLeaf"
leaf hash
leaf hash
tagged hash
"TapBranch"
branch hash
leaf hash
tagged hash
"TapBranch"
public key
merkle root
tagged hash
"TapTweak"
tweak
G
P
Q
Q = P + (G * tweak)
ScriptPubKey:
OP_1
OP_PUSHBYTES_32
tweaked public key
P2TR
parity:
even = 1
odd = 0
+
control byte
public key
merkle path
Witness:
script
control block
script inputs

- Gerar a tweak private key

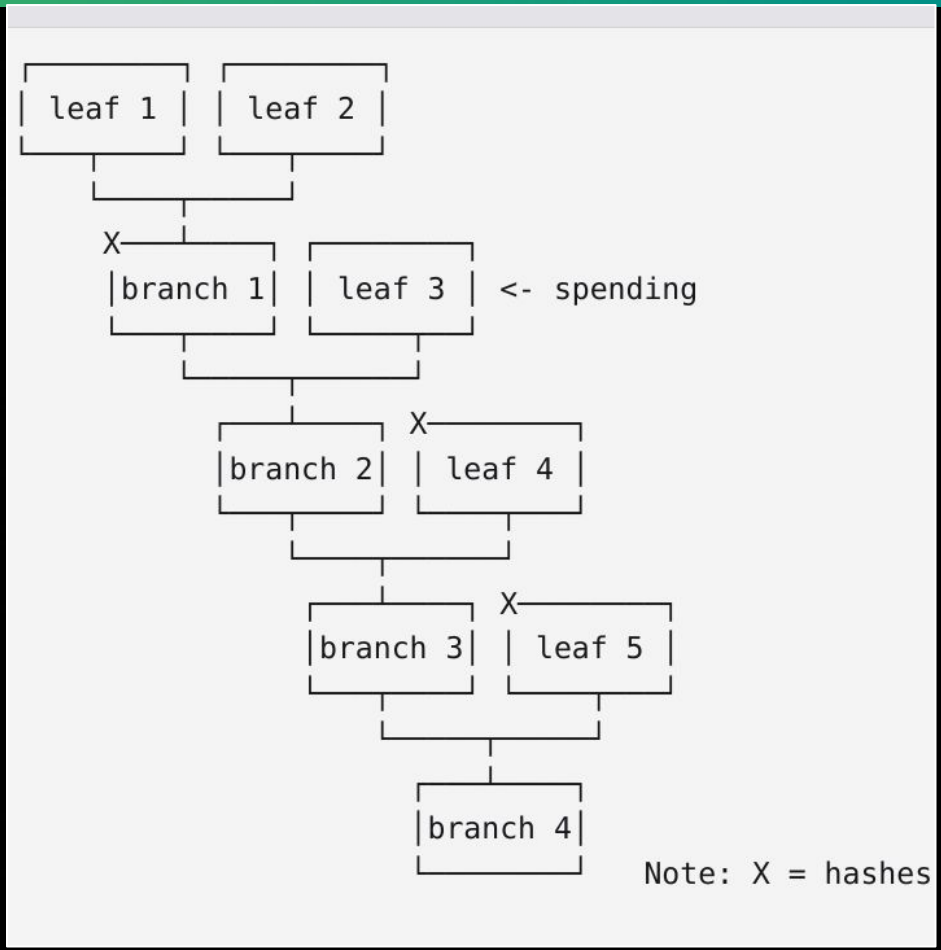
```
private key         = ce1fc7baa9db31c4ef9c6564f70d551f41fc479bb23fa844d50848220edaaf91

NOTE: If the private key produces a public key with an odd y coordinate, it needs to be negated s

n                   = fffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141
private key negated = n - private key
private key negated = 31e038455624ce3b10639a9b08f2aadf78b2954afd08f7f6eaca166ac15b91b0

tweak               = bf0094eae70ba67e2f9fc3c4b81f078c90931855a8d24c959619174c92060cde

tweaked private key = (private key negated + tweak) % n
tweaked private key = f0e0cd303d3074b940035e5fc111b26c0945ada0a5db448c80e32db753619e8e
```

NOTE: If the private key produces a public key with an odd y coordinate, it needs to be negated so that it produces the same public key but with an even y coordinate,


private key
public key
merkle root
tagged hash
"TapTweak"
tweak
G
P
Q
Q = P + (G * tweak)
ScriptPubKey:
OP_1
OP_PUSHBYTES_32
tweaked public key
P2TR
negate private key if it produces a public key with an odd y value
+ hash type
+ extension flag = 0
+ annex present
sighash epoch
signature message
tagged hash
"TapSigHash"
signature hash
Taproot Signature Algorithm
+
Schnorr Sign (
tweaked private key
,
auxiliary bytes
,
message
)
Witness:
signature

- Spend by script path 🛣️

Note o que alguns hashes são presumíveis e portanto não precisam ser informados no final.



Calculo 🧮

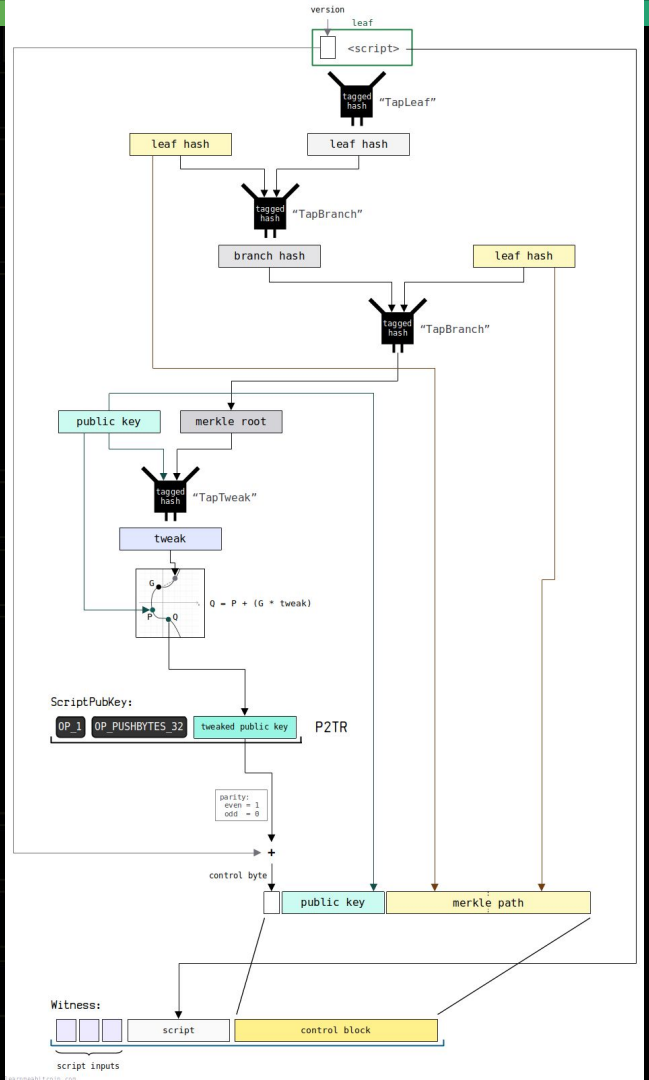
```
leaf 1 hash  = 6b13becdaf0eee497e2f304adcfa1c0c9e84561c9989b7f2b5fc39f5f90a60f6
leaf 2 hash  = ed5af8352e2a54cce8d3ea326beb7907efa850bdfe3711cef9060c7bb5bcf59e
leaf 3 hash  = 160bd30406f8d5333be044e6d2d14624470495da8a3f91242ce338599b233931
leaf 4 hash  = bf2c4bf1ca72f7b8538e9df9bdfd3ba4c305ad11587f12bbfafa00d58ad6051d
leaf 5 hash  = 54962df196af2827a86f4bde3cf7d7c1a9dcb6e17f660badefbc892309bb145f

branch 1     = 1324300a84045033ec539f60c70d582c48b9acf04150da091694d83171b44ec9
branch 2     = beec0122bddd26f642140bcd922e0264ce1e2be5808a41ae58d82e829bc913d7
branch 3     = a4e0d9cc12ce2f32069e98247581d5eb9ca0a4cf175771a8df2c53a93dcb0ebd
branch 4     = b5b72eea07b3e338962944a752a98772bbe1f1b6550e6fb6ab8c6e6adb152e7c

merkle path  = branch 1 + leaf 4 hash + leaf 5 hash
merkle path  = 1324300a84045033ec539f60c70d582c48b9acf04150da091694d83171b44ec9b
```

- Script inputs

Example [hide]

```
script inputs = 03

NOTE: This will unlock the simple "OP_EQUAL OP_3" upcoming leaf script
```

Neste campo temos os stack items, necessários para usar a folha escolhida.


version
leaf
<script>
tagged hash
"TapLeaf"
leaf hash
leaf hash
tagged hash
"TapBranch"
branch hash
leaf hash
tagged hash
"TapBranch"
public key
merkle root
tagged hash
"TapTweak"
tweak
G
P
Q
Q = P + (G * tweak)
ScriptPubKey:
OP_1
OP_PUSHBYTES_32
tweaked public key
P2TR
parity:
even = 1
odd = 0
+
control byte
public key
merkle path
Witness:
script
control block
script inputs

vinteum

Learn more:

Obrigado.

Joãozinho

@holand1antari

joaozinho.freefall704@passinbox.com