Covenants with BIP-448

a TapScript covenant proposal

authors:

- Antoine Poinsot
- Greg Sanders
- Steven Roose

Covenants

- now: conditions to spend the money
  - signature from public key: scriptSig & witness
  - time constraints: lockTime, sequence, opcodes
  - each input fulfills previous output constraint
- covenants: where can the money go
  - which new outputs can be created
  - each covenant output encumbers all new outputs

Why covenants?

- better programmability
  - security
- second layer protocols
  - Lightning
  - Ark
  - BitVM

Commit to next tx (outputs)

- I have two kids: Alice and Bob
- policy
  - I can spend with my keys using keyspend
  - after 5 years money gets divided into A&B’s wallets

Commit to next tx (outputs)

- I have two kids: Alice and Bob
- example: UTXO of 10 BTC
  - either you spend the 10 BTC directly
  - after 5 years can be spent by a tx that gives 5 BTC to Alice and 5 BTC to Bob

`OP_CHECKTEMPLATEVERIFY` (BIP-119)

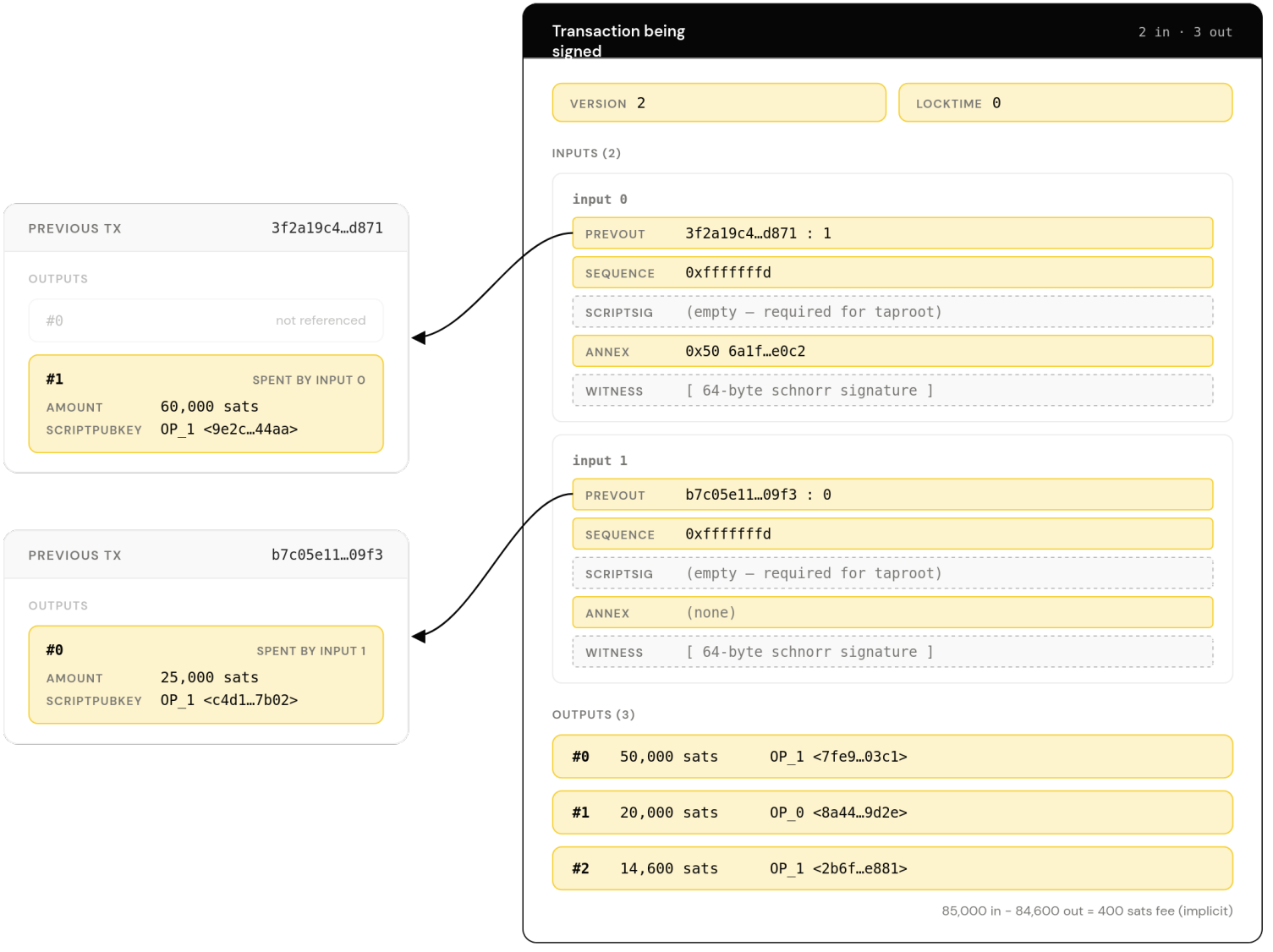
- check current transaction against “template”
  - transaction hash that does not commit prevouts
- pre-calculate the template
  - `<template> OP_CTV`

Commit to next tx (outputs)

- I have two kids: Alice and Bob
- example: UTXO of 10 BTC
  - either you spend the 10 BTC directly
  - after 5 years can be spent by a tx that gives 5 BTC to Alice and 5 BTC to Bob
- `<template> OP_CTV <5 yrs> OP_CSV`
  - template: 1-input-2-output tx to A & B

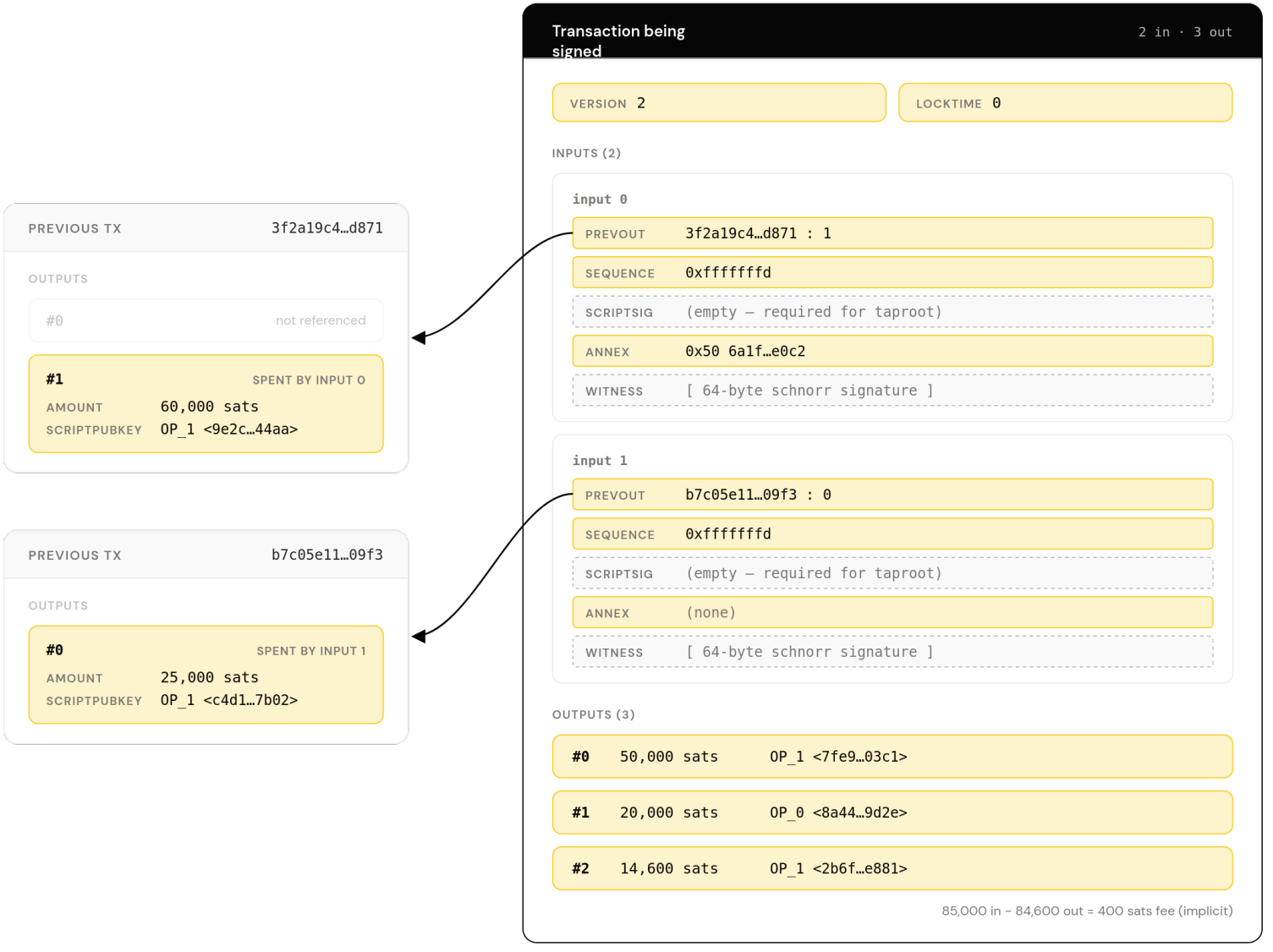
Tangent: Signatures

- signatures **do** encumber the transaction outputs
- `<pubkey> OP_CHECKSIG`
  - witness: `<signature>`


PREVIOUS TX
3f2a19c4…d871
OUTPUTS
#0
not referenced
#1
SPENT BY INPUT 0
AMOUNT
60,000 sats
SCRIPTPUBKEY
OP_1 <9e2c…44aa>
PREVIOUS TX
b7c05e11…09f3
OUTPUTS
#0
SPENT BY INPUT 1
AMOUNT
25,000 sats
SCRIPTPUBKEY
OP_1 <c4d1…7b02>
Transaction being signed
2 in · 3 out
VERSION 2
LOCKTIME 0
INPUTS (2)
input 0
PREVOUT
3f2a19c4…d871 : 1
SEQUENCE
0xfffffffd
SCRIPTSIG
(empty — required for taproot)
ANNEX
0x50 6a1f…e0c2
WITNESS
[64-byte schnorr signature]
input 1
PREVOUT
b7c05e11…09f3 : 0
SEQUENCE
0xfffffffd
SCRIPTSIG
(empty — required for taproot)
ANNEX
(none)
WITNESS
[64-byte schnorr signature]
OUTPUTS (3)
#0 50,000 sats OP_1 <7fe9…03c1>
#1 20,000 sats OP_0 <8a44…9d2e>
#2 14,600 sats OP_1 <2b6f…e881>
85,000 in − 84,600 out = 400 sats fee (implicit)

Tangent: Signatures

- signatures **do** encumber the transaction outputs
- `<pre-made sig> <pubkey> OP_CHECKSIG`


PREVIOUS TX
3f2a19c4…d871
OUTPUTS
#0
not referenced
#1
SPENT BY INPUT 0
AMOUNT
60,000 sats
SCRIPTPUBKEY
OP_1 <9e2c…44aa>
PREVIOUS TX
b7c05e11…09f3
OUTPUTS
#0
SPENT BY INPUT 1
AMOUNT
25,000 sats
SCRIPTPUBKEY
OP_1 <c4d1…7b02>
Transaction being signed
2 in · 3 out
VERSION 2
LOCKTIME 0
INPUTS (2)
input 0
PREVOUT
3f2a19c4…d871 : 1
SEQUENCE
0xfffffffd
SCRIPTSIG
(empty — required for taproot)
ANNEX
0x50 6a1f…e0c2
WITNESS
[64-byte schnorr signature]
input 1
PREVOUT
b7c05e11…09f3 : 0
SEQUENCE
0xfffffffd
SCRIPTSIG
(empty — required for taproot)
ANNEX
(none)
WITNESS
[64-byte schnorr signature]
OUTPUTS (3)
#0 50,000 sats OP_1 <7fe9…03c1>
#1 20,000 sats OP_0 <8a44…9d2e>
#2 14,600 sats OP_1 <2b6f…e881>
85,000 in − 84,600 out = 400 sats fee (implicit)

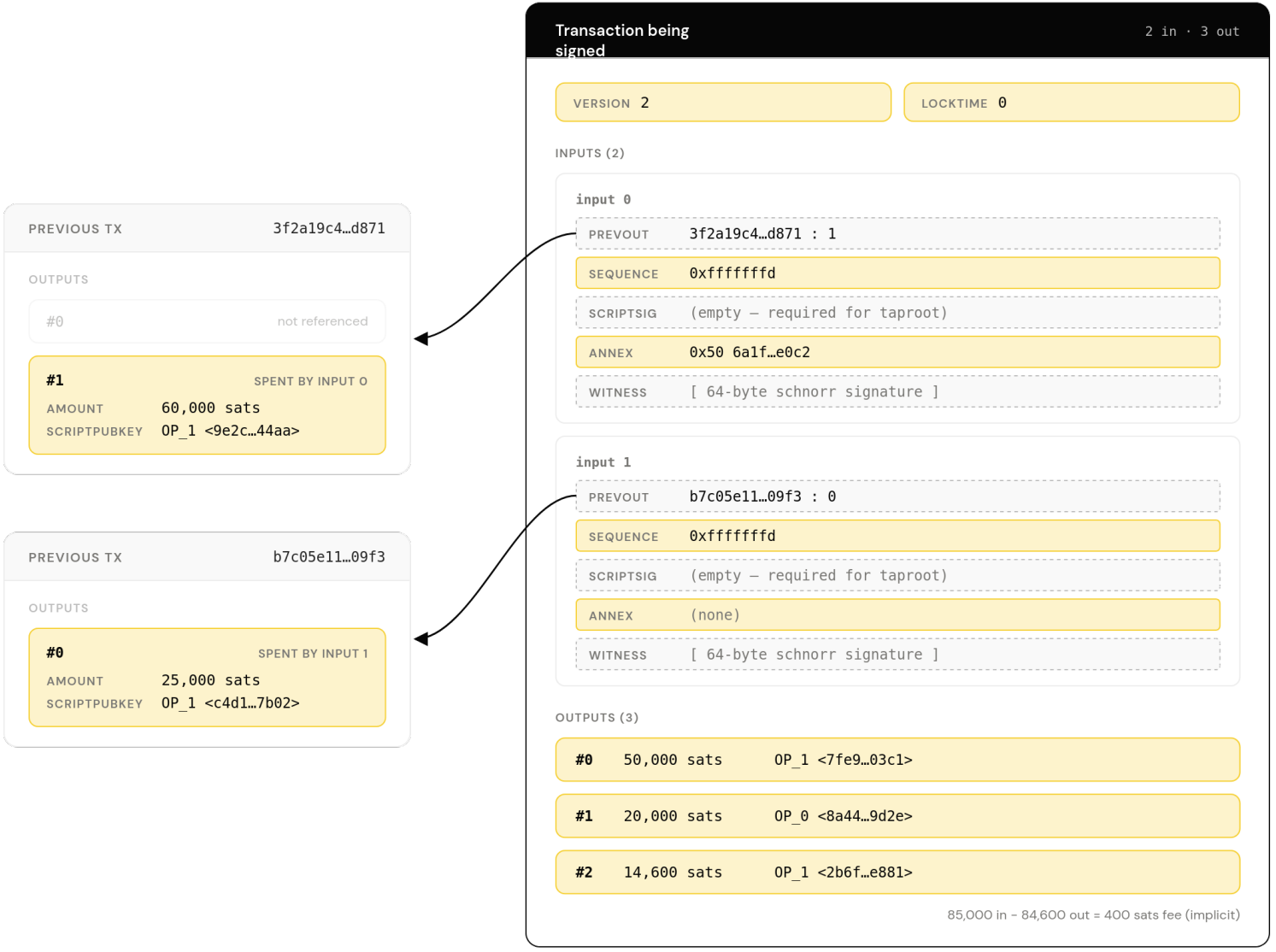
Tangent: Signatures

- signatures **do** encumber the transaction outputs
- `<pre-made sig> <pubkey> OP_CHECKSIG`
  - doesn’t work because of prevouts

SIGHASH_ANYPREVOUT (BIP-118)

(previously: SIGHASH_NOINPUT)

- re-bindable signatures
- pre-sign off-chain transactions
  - without knowing the txid of the input UTXO
  - Lightning channel commitment transactions (pre-segwit)
  - Lightning symmetry
- <pubkey> OP_CHECKSIG
  - witness: <APO-flag|signature>


PREVIOUS TX
3f2a19c4…d871
OUTPUTS
#0
not referenced
#1
SPENT BY INPUT 0
AMOUNT
60,000 sats
SCRIPTPUBKEY
OP_1 <9e2c…44aa>
PREVIOUS TX
b7c05e11…09f3
OUTPUTS
#0
SPENT BY INPUT 1
AMOUNT
25,000 sats
SCRIPTPUBKEY
OP_1 <c4d1…7b02>
Transaction being signed
2 in · 3 out
VERSION 2
LOCKTIME 0
INPUTS (2)
input 0
PREVOUT
3f2a19c4…d871 : 1
SEQUENCE
0xfffffffd
SCRIPTSIG
(empty — required for taproot)
ANNEX
0x50 6a1f…e0c2
WITNESS
[64-byte schnorr signature]
input 1
PREVOUT
b7c05e11…09f3 : 0
SEQUENCE
0xfffffffd
SCRIPTSIG
(empty — required for taproot)
ANNEX
(none)
WITNESS
[64-byte schnorr signature]
OUTPUTS (3)
#0
50,000 sats
OP_1 <7fe9…03c1>
#1
20,000 sats
OP_0 <8a44…9d2e>
#2
14,600 sats
OP_1 <2b6f…e881>
85,000 in - 84,600 out = 400 sats fee (implicit)

Commit to next tx (outputs)

- I have two kids: Alice and Bob
- example: UTXO of 10 BTC
  - either you spend the 10 BTC directly
  - after 5 years can be spent by a tx that gives 5 BTC to Alice and 5 BTC to Bob
- `<APO|sig> <pubkey> OP_CHECKSIG <5 yrs> OP_CSV`
  - signature pre-signs 1-input-2-output tx to A & B

Technical equivalence

- CTV using APO
  - `<APO|signature> <pubkey> OP_CHECKSIG`
  - “pre-commit the APO signature”

OP_CHECKSIGFROMSTACK (BIP-348)

- generic message signature verification
- `<signature> <message> <pubkey> OP_CSFS`
- `<message> <pubkey> OP_CSFS`
  - witness: `<signature>`

Technical equivalence

- CTV using APO
  - `<APO|signature> <pubkey> OP_CHECKSIG`
  - “pre-commit the APO signature”
- APO using CTV & CSFS
  - `OP_CTV <pubkey> OP_CSFS`
    - witness: `<template> <signature>`
  - “check signature of the template”

Summary APO vs CTV

- CTV + CSFS
  - `<template> OP_CTV` ~34wu
  - `OP_CTV <pubkey> OP_CSFS` ~133wu
    - witness: `<template> <signature>`
- APO
  - `<pubkey> OP_CHECKSIG` ~100wu
    - witness: `<APO|signature>` ~100wu
  - `<APO|signature> <pubkey> OP_CHECKSIG`

BIP-448

- re-bindable signatures
- next-transaction commitment
- three opcodes, tapscript only

OP_TEMPLATEHASH (BIP-446)

- push the template hash directly on the stack
  - TapScript push opcodes using OP_SUCCESS!
- next-tx commitment
  - <template> OP_TEMPLATEHASH OP_EQUALVERIFY
- re-bindable signature
  - OP_TEMPLATEHASH <pubkey> OP_CSFS
    - witness: <signature>

OP_TEMPLATEHASH (BIP-446)

- TEMPLATEHASH ~!= CTV
  - goal is txid stability given prevout
  - taproot only: no *scriptSigs* allowed
  - cfr taproot SIGHASH_ALL, but
    - no *hash_type* and *spend_type*: irrelevant
    - no *prevouts*: obvious
    - no *amounts*: avoid footgun in templates
  - commit *annex*: closer to SIGHASH modes

Comparison

	next-tx commit	re-bindable sig
APO	100 wu	100 wu
CTV + CSFS	34 wu	133 wu
TH + CSFS	35 wu	100 wu

OP_TEMPLATEHASH (BIP-446)

- push the template hash directly on the stack
  - TapScript push opcodes using OP_SUCCESS!
- next-tx commitment
  - <template> OP_TEMPLATEHASH OP_EQUALVERIFY
- re-bindable signature
  - OP_TEMPLATEHASH <pubkey> OP_CSFS
    - witness: <signature>

`OP_INTERNALKEY` (BIP-349)

- push the taproot internal key on the stack
  - the control block already contains a pubkey!
- re-bindable signature
  - `OP_TEMPLATEHASH OP_INTERNALKEY OP_CSFS`
    - witness: `<signature>`

~68wu

Comparison

	next-tx commit	re-bindable sig
APO + IK	68 wu	68 wu
CTV + CSFS + IK	34 wu	101 wu
TH + CSFS + IK	35 wu	68 wu

BIP-448

- re-bindable signatures
- next-transaction commitment
- three opcodes, tapscript only
  - `OP_TEMPLATEHASH` (BIP-446)
  - `OP_CHECKSIGFROMSTACK` (BIP-348)
  - `OP_INTERNALKEY` (BIP-349)
- bip448.dev

What’s next?

- learn: bip448.dev
- build proof-of-concepts!

Ark

- original Ark design used CTV
- “Covenant-less Ark”
  - workaround using interactive signing rounds
  - shipped by Second and Ark Labs

Ark

- TEMPLATEHASH support in Bark

- visit templatehash.com to try it out!

BIP-448

- re-bindable signatures
- next-transaction commitment
- three opcodes, tapscript only
  - `OP_TEMPLATEHASH` (BIP-446)
  - `OP_CHECKSIGFROMSTACK` (BIP-348)
  - `OP_INTERNALKEY` (BIP-349)
- **bip448.dev**