SIGHASH_NOINPUT (bip118)

Bitcoinedge initiative

Tel Aviv, Israel
September 2019

Bryan Bishop <kanzure@gmail.com>

0E4C A12B E16B E691 56F5 40C9 984F 10CC 7716 9FD2

Sighash flags

- Sighash type is a single byte appended to the DER-encoded signature
- SIGHASH_ALL - sign entire transaction except signature scripts
- SIGHASH_NONE - sign only inputs, anyone can change the outputs, anyone can add a SIGHASH_ALL signature
- SIGHASH_ANYONECANPAY - signs only the current input
- SIGHASH_SINGLE - signs this input, its corresponding output, and other inputs partially

Proposed exotic sighash flags

- SIGHASH_MULTIPLE
- SIGHASH_LIST
- SIGHASH_WITHINPUTVALUE
- SIGHASH_NOINPUT
- SIGHASH_NORMALIZED
- SIGHASH_WITHOUT_*
- SIGHASH_SUM
- SIGHASH_DANGEROUSLYPROMISCUOUS

https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-August/010759.html

SIGHASH_NOINPUT (bip118)

- Dissociate transactions from a specific input, while still requiring a valid signature
  - SIGHASH_NOINPUT signature does not cover/include the input's txid nor input's index
- Originally proposed in February 2015 as a malleability fix (before segwit) by Lightning Network authors
- February 2016 writeup (bip118 includes the amount!)
- bip118 by Christian Decker (cdecker)
- Not activated/deployed yet (October 2018)

Application-specific pubkeys only

"This also means that particular care has to be taken in order to avoid unintentionally enabling this rebinding mechanism. NOINPUT MUST NOT be used, unless it is explicitly needed for the application, e.g., it MUST NOT be a default signing flag in a wallet implementation. Rebinding is only possible when the outputs the transaction may bind to all use the same public keys. Any public key that is used in a NOINPUT signature MUST only be used for outputs that the input may bind to, and they MUST NOT be used for transactions that the input may not bind to. For example an application SHOULD generate a new key-pair for the application instance using NOINPUT signatures and MUST NOT reuse them afterwards."

Application-specific pubkeys only

- Also called "contract-specific key-pairs"
- Any input can be swapped on the bitcoin network, as long as it has a valid signature
- Adversaries may automatically do this based on their observations or knowledge
  - Discourages excessive pubkey reuse
- May inadvertedly burn excess fee to miners
- Great for situations where same BTC amounts on many different UTXOs that are intended to be spendable by same transaction

Lightning network's use case for SIGHASH_NOINPUT

- Third-party watchtowres can monitor for channel violations and broadcast pre-signed transactions to sweep the channel funds as a penalty/revocation transaction for all prior states
  - Probably also transfer more data each time more funds are added to the channel...
- Previously: data transferred to the third-party for each lightning channel/state update

More use cases

- Malleability fix (with some downsides about key reuse)
- Lightning watchtowers
- eltoo
  - Compress transaction history by using SIGHASH_NOINPUT to bind any prior state with the latest state, skipping intermediate states.
  - Don't need a penalty transaction for each state change

Alternative SIGHASH_NOINPUT proposals

- I think there was an opcode proposal at some point?
- 08:58 <cdecker> Sure
- 09:27 <cdecker> Not exactly sure what you mean with "May inadvertedly burn excess fee to miners" on slide 7
- 09:28 <cdecker> noinput still commits to the value that is being spent, so the fee will be identical to all rebindings. It's a restriction I put there to avoid rebindings as a way to change fees.
- 09:29 <cdecker> That's in my writeup of BIP118, could never find enough details on the original proposal
- 09:31 <cdecker> As for alternatives there is Johnson's SIGHASH2 proposal which is a strict superset of BIP118, with some added quirks, and there's Roasbeefs checksigfromstack, which could also be used in a similar fashion
- 09:31 <cdecker> noinput as a separate opcode was also proposed
-

SIGHASH_NOINPUT (bip118)

Bitcoinedge initiative

Tel Aviv, Israel
September 2019

Bryan Bishop <kanzure@gmail.com>

0E4C A12B E16B E691 56F5 40C9 984F 10CC 7716 9FD2

https://twitter.com/kanzure