IT486 v3.0: Blockchains and Cryptocurrencies

Bitcoin transaction malleability

What is malleability?

- it's when adversaries can modify ciphertexts, messages, signatures, etc. and things still work etc.
- in the case of Bitcoin, transactions can be changed and still be valid

tx asymmetry

recall the tx format; inputs and outputs don't look the same

txid:index (36B) signature (1000B)	script (25B) amount (8B)
txid:index signature	script (pubkey) amount

What gets signed

- sign the whole tx, inputs and output
- but inputs contain signatures
- and you can't sign the signature

What gets signed

- remove the signature fields, sign, then put signatures in
- change any bit of the signed message and the signature is invalid
- but txid is the hash of the message, including signatures

Signature malleability

- 3rd party malleability
- leading zeros
- low s can flip the sign of the signatures and it's still valid

Signature malleability

- 1st party
- recall signing uses a nonce k
- use a different k, different valid signature on the same message
- RFC 6979 defines deterministic k algo

So you've been malleated

- txid changes
- outputs are still the same
- which inputs also still the same
- no big deal?

So you've been malleated

- in most cases, some wallets have trouble

So you've been malleated

- in most cases, some wallets have trouble
- broadcast tx 2d5cac, which never got confirmed
- instead malleated to 9cba3e

So you've been malleated

- in most cases, some wallets have trouble
- broadcast tx 2d5cac, which never got confirmed
- instead malleated to 9cba3e
- wallet shows unconfirmed forever

dependent txs

- spending unconfirmed change output from tx1 7feec1. sign and broadcast tx2

dependent txs

- spending unconfirmed change output from tx1 7feec1. sign and broadcast tx2
- tx1 changes to b2068c1
- tx2 invalid, refers to txid which can never be confirmed

dependent txs

- txid change is annoying but can refer to malleated txids and re-sign
- what if you can't resign?
- multisign, pre-signed txs

different idea

- use non-malleable signatures?
- but many useful signature schemes are malleable

Segregated witness

- don't include signatures in txids
- signature changes but txid doesn't

Segregated witness

- don't include signatures in txids
- signature changes but txid doesn't
- how do you make this backwards compatible?

Segregated witness

- don't include signatures in txids
- signature changes but txid doesn't
- how do you make this backwards compatible?
  - make outputs which don't require signatures!

Segwit transaction

- output (locking) script:

  0 <pubkey hash>
- sig script (unlocking):

  (nothing)

Segwit transaction

- output (locking) script:

  `0 <pubkey hash>`
- sig script (unlocking):

  `(nothing)`
- old nodes: see blank signature

Segwit transaction

- output (locking) script:

  0 <pubkey hash>
- sig script (unlocking):

  (nothing)
- old nodes: see blank signature
  - push 0, followed by <pubkey hash>

Segwit transaction

- output (locking) script:

  0 <pubkey hash>
- sig script (unlocking):

  (nothing)
- old nodes: see blank signature
  - push 0, followed by <pubkey hash>
  - since the top of stack is non-zero, transaction counts as valid, and coins move!

Segwit transaction

- output script:

  `0 <pubkey hash>`
- sig script (unlocking):

  `(nothing)`

Segwit transaction

- output script:

  0 <pubkey hash>
- sig script (unlocking):

  (nothing)
- new nodes: treat the above sequence as a template

Segwit transaction

- output script:

  0 <pubkey hash>
- sig script (unlocking):

  (nothing)
- new nodes: treat the above sequence as a template
  - get the pubkey and signature from the "witness" field, which old nodes never see

Segwit transaction

- output script:

  0 <pubkey hash>
- sig script (unlocking):

  (nothing)
- new nodes: treat the above sequence as a template
  - get the pubkey and signature from the "witness" field, which old nodes never see
  - execute the same sequence as with p2pkh

    <signature> <pub key> OP_DUP OP_HASH160 <pubkey hash> OP_EQUALVERIFY OP_CHECKSIG

new tx type

recall old tx format

txid:index (36B) signature (1000B)	script (25B) amount (8B)
txid:index signature	script (pubkey) amount

new tx type

new tx format

txid:index (36B) signature (0B) [witness]	script (25B) amount (8B)
txid:index signature [witness]	script (pubkey) amount

omit to old nodes

- when people ask for witness txs, include the witness
- when they just ask for txs, give it to them without the witness field

omit to old nodes

- old nodes: signature can't change; there isn't one!
- new nodes: signature can change, but doesn't affect txid