```
BIP: 143
Layer: Consensus (soft fork)
Title: Transaction Signature Verification for Version 0 Witness Program
Author: Johnson Lau <jl2012@xbt.hk>
        Pieter Wuille <pieter.wuille@gmail.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0143
Status: Final
Type: Standards Track
Created: 2016-01-03
License: PD
```

## Abstract

This proposal defines a new transaction digest algorithm for signature verification in version 0 witness program, in order to minimize redundant data hashing in verification, and to cover the input value by the signature.

## Motivation

There are 4 ECDSA signature verification codes in the original Bitcoin script system: `CHECKSIG`, `CHECKSIGVERIFY`, `CHECKMULTISIG`, `CHECKMULTISIGVERIFY` ("sigops"). According to the sighash type (`ALL`, `NONE`, `SINGLE`, `ANYONECANPAY`), a transaction digest is generated with a double SHA256 of a serialized subset of the transaction, and the signature is verified against this digest with a given public key. The detailed procedure is described in a Bitcoin Wiki article. [1]

Unfortunately, there are at least 2 weaknesses in the original SignatureHash transaction digest algorithm:

- For the verification of each signature, the amount of data hashing is proportional to the size of the transaction. Therefore, data hashing grows in $O(n^2)$ as the number of sigops in a transaction increases. While a 1 MB block would normally take 2 seconds to verify with an average computer in 2015, a 1MB transaction with 5569 sigops may take 25 seconds to verify. This could be fixed by optimizing the digest algorithm by introducing some reusable "midstate", so the time complexity becomes O(n). [2][3][4]
- The algorithm does not involve the amount of Bitcoin being spent by the input. This is usually not a problem for online network nodes as they could request for the specified transaction to acquire the output value. For an offline transaction signing device ("cold wallet"), however, the unknowing of input amount makes it impossible to calculate the exact amount being spent and the transaction fee. To cope with this problem a cold wallet must

[1] 1
[2] CVE-2013-2292
[3] New Bitcoin vulnerability: A transaction that takes at least 3 minutes to verify
[4] The Megatransaction: Why Does It Take 25 Seconds?

also acquire the full transaction being spent, which could be a big obstacle in the implementation of lightweight, air-gapped wallet. By including the input value of part of the transaction digest, a cold wallet may safely sign a transaction by learning the value from an untrusted source. In the case that a wrong value is provided and signed, the signature would be invalid and no funding might be lost. [5]

Deploying the aforementioned fixes in the original script system is not a simple task. That would be either a hardfork, or a softfork for new sigops without the ability to remove or insert stack items. However, the introduction of segregated witness softfork offers an opportunity to define a different set of script semantics without disrupting the original system, as the unupgraded nodes would always consider such a transaction output is spendable by arbitrary signature or no signature at all. [6]

## Specification

A new transaction digest algorithm is defined, but only applicable to sigops in version 0 witness program:

```
 Double SHA256 of the serialization of:
1. nVersion of the transaction (4-byte little endian)
2. hashPrevouts (32-byte hash)
3. hashSequence (32-byte hash)
4. outpoint (32-byte hash + 4-byte little endian)
5. scriptCode of the input (serialized as scripts inside CTxOuts)
6. value of the output spent by this input (8-byte little endian)
7. nSequence of the input (4-byte little endian)
8. hashOutputs (32-byte hash)
9. nLocktime of the transaction (4-byte little endian)
10. sighash type of the signature (4-byte little endian)
```

Semantics of the original sighash types remain unchanged, except the followings:

1. The way of serialization is changed;
2. All sighash types commit to the amount being spent by the signed input;
3. `FindAndDelete` of the signature is not applied to the `scriptCode`;
4. `OP_CODESEPARATOR`(s) after the last executed `OP_CODESEPARATOR` are not removed from the `scriptCode` (the last executed `OP_CODESEPARATOR` and any script before it are always removed);
5. `SINGLE` does not commit to the input index. When `ANYONECANPAY` is not set, the semantics are unchanged since `hashPrevouts` and `outpoint` together implictly commit to the input index. When `SINGLE` is used with `ANYONECANPAY`, omission of the index commitment allows permutation of the input-output pairs, as long as each pair is located at an equivalent index.

[5] SIGHASH_WITHINPUTVALUE: Super-lightweight HW wallets and offline data
[6] BIP141: Segregated Witness (Consensus layer)

The items 1, 4, 7, 9, 10 have the same meaning as the original algorithm. [7]

The item 5:

- For `P2WPKH` witness program, the `scriptCode` is `0x1976a914{20-byte-pubkey-hash}88ac`.
- For `P2WSH` witness program,
  - if the `witnessScript` does not contain any `OP_CODESEPARATOR`, the `scriptCode` is the `witnessScript` serialized as scripts inside `CTxOut`.
  - if the `witnessScript` contains any `OP_CODESEPARATOR`, the `scriptCode` is the `witnessScript` but removing everything up to and including the last executed `OP_CODESEPARATOR` before the signature checking opcode being executed, serialized as scripts inside `CTxOut`. (The exact semantics is demonstrated in the examples below)

The item 6 is a 8-byte value of the amount of bitcoin spent in this input.

`hashPrevouts`:

- If the `ANYONECANPAY` flag is not set, `hashPrevouts` is the double SHA256 of the serialization of all input outpoints;
- Otherwise, `hashPrevouts` is a `uint256` of `0x0000......0000`.

`hashSequence`:

- If none of the `ANYONECANPAY`, `SINGLE`, `NONE` sighash type is set, `hashSequence` is the double SHA256 of the serialization of `nSequence` of all inputs;
- Otherwise, `hashSequence` is a `uint256` of `0x0000......0000`.

`hashOutputs`:

- If the sighash type is neither `SINGLE` nor `NONE`, `hashOutputs` is the double SHA256 of the serialization of all output amount (8-byte little endian) with `scriptPubKey` (serialized as scripts inside CTxOuts);
- If sighash type is `SINGLE` and the input index is smaller than the number of outputs, `hashOutputs` is the double SHA256 of the output amount with `scriptPubKey` of the same index as the input;
- Otherwise, `hashOutputs` is a `uint256` of `0x0000......0000`.[8]

The `hashPrevouts`, `hashSequence`, and `hashOutputs` calculated in an earlier verification may be reused in other inputs of the same transaction, so that the time complexity of the whole hashing process reduces from $O(n^2)$ to $O(n)$.

Refer to the reference implementation, reproduced below, for the precise algorithm:

[7]

[8] In the original algorithm, a `uint256` of `0x0000......0001` is committed if the input index for a `SINGLE` signature is greater than or equal to the number of outputs. In this BIP a `0x0000......0000` is committed, without changing the semantics.

```
uint256 hashPrevouts;
uint256 hashSequence;
uint256 hashOutputs;

if (!(nHashType & SIGHASH_ANYONECANPAY)) {
    CHashWriter ss(SER_GETHASH, 0);
    for (unsigned int n = 0; n < txTo.vin.size(); n++) {
        ss << txTo.vin[n].prevout;
    }
    hashPrevouts = ss.GetHash();
}

if (!(nHashType & SIGHASH_ANYONECANPAY) && (nHashType & 0x1f) != SIGHASH_SINGLE && (nHashT
    CHashWriter ss(SER_GETHASH, 0);
    for (unsigned int n = 0; n < txTo.vin.size(); n++) {
        ss << txTo.vin[n].nSequence;
    }
    hashSequence = ss.GetHash();
}

if ((nHashType & 0x1f) != SIGHASH_SINGLE && (nHashType & 0x1f) != SIGHASH_NONE) {
    CHashWriter ss(SER_GETHASH, 0);
    for (unsigned int n = 0; n < txTo.vout.size(); n++) {
        ss << txTo.vout[n];
    }
    hashOutputs = ss.GetHash();
} else if ((nHashType & 0x1f) == SIGHASH_SINGLE && nIn < txTo.vout.size()) {
    CHashWriter ss(SER_GETHASH, 0);
    ss << txTo.vout[nIn];
    hashOutputs = ss.GetHash();
}

CHashWriter ss(SER_GETHASH, 0);
// Version
ss << txTo.nVersion;
// Input prevouts/nSequence (none/all, depending on flags)
ss << hashPrevouts;
ss << hashSequence;
// The input being signed (replacing the scriptSig with scriptCode + amount)
// The prevout may already be contained in hashPrevout, and the nSequence
// may already be contain in hashSequence.
ss << txTo.vin[nIn].prevout;
ss << static_cast<const CScriptBase&>(scriptCode);
ss << amount;
ss << txTo.vin[nIn].nSequence;
// Outputs (none/one/all, depending on flags)
```

```
ss << hashOutputs;
// Locktime
ss << txTo.nLockTime;
// Sighash type
ss << nHashType;

return ss.GetHash();
```

## Restrictions on public key type

As a default policy, only compressed public keys are accepted in `P2WPKH` and `P2WSH`. Each public key passed to a sigop inside version 0 witness program must be a compressed key: the first byte MUST be either `0x02` or `0x03`, and the size MUST be 33 bytes. Transactions that break this rule will not be relayed or mined by default.

Since this policy is preparation for a future softfork proposal, to avoid potential future funds loss, users MUST NOT use uncompressed keys in version 0 witness programs.

## Example

To ensure consistency in consensus-critical behaviour, developers should test their implementations against all the tests below. More tests related to this proposal could be found under https://github.com/bitcoin/bitcoin/tree/master/src/test/data .

### Native P2WPKH

```
 The following is an unsigned transaction:
0100000002fff7f7881a8099afa6940d42d1e7f6362bec38171ea3edf433541db4e4ad969f0000000000eeffffff

nVersion:  01000000
txin:      02 fff7f7881a8099afa6940d42d1e7f6362bec38171ea3edf433541db4e4ad969f 00000000 00 e
ef51e1b804cc89d182d279655c3aa89e815b1b309fe287d9b2b55d57b90ec68a 01000000 00 ffffffff
txout:     02 202cb20600000000 1976a9148280b37df378db99f66f85c95a783a76ac7a6d5988ac
9093510d00000000 1976a9143bde42dbee7e4dbe6a21b2d50ce2f0167faa815988ac
nLockTime: 11000000

The first input comes from an ordinary P2PK:
scriptPubKey : 2103c9f4836b9a4f77fc0d81f7bcb01b7f1b35916864b9476c241ce9fc198bd25432ac value:
private key  : bbc27228ddcb9209d7fd6f36b02f7dfa6252af40bb2f1cbc7a557da8027ff866

The second input comes from a P2WPKH witness program:
scriptPubKey : 00141d0f172a0ecb48aee1be1f2687d2963ae33f71a1, value: 6
private key  : 619c335025c7f4012e556c2a58b2506e30b8511b53ade95ea316fd8c3286feb9
public key   : 025476c2e83188368da1ff3e292e7acafcdb3566bb0ad253f62fc70f07aeee6357
```

```
To sign it with a nHashType of 1 (SIGHASH_ALL):

hashPrevouts:
dSHA256(fff7f7881a8099afa6940d42d1e7f6362bec38171ea3edf433541db4e4ad969f00000000ef51e1b804cc
= 96b827c8483d4e9b96712b6713a7b68d6e8003a781feba36c31143470b4efd37

hashSequence:
dSHA256(eeffffffffffffff)
= 52b0a642eea2fb7ae638c36f6252b6750293dbe574a806984b8e4d8548339a3b

hashOutputs:
dSHA256(202cb206000000001976a9148280b37df378db99f66f85c95a783a76ac7a6d5988ac9093510d00000000
= 863ef3e1a92afbfdb97f31ad0fc7683ee943e9abcf2501590ff8f6551f47e5e5

hash preimage: 0100000096b827c8483d4e9b96712b6713a7b68d6e8003a781feba36c31143470b4efd3752b0a

nVersion:     01000000
hashPrevouts: 96b827c8483d4e9b96712b6713a7b68d6e8003a781feba36c31143470b4efd37
hashSequence: 52b0a642eea2fb7ae638c36f6252b6750293dbe574a806984b8e4d8548339a3b
outpoint:     ef51e1b804cc89d182d279655c3aa89e815b1b309fe287d9b2b55d57b90ec68a01000000
scriptCode:   1976a9141d0f172a0ecb48aee1be1f2687d2963ae33f71a188ac
amount:       0046c32300000000
nSequence:    ffffffff
hashOutputs:  863ef3e1a92afbfdb97f31ad0fc7683ee943e9abcf2501590ff8f6551f47e5e5
nLockTime:    11000000
nHashType:    01000000

sigHash:      c37af31116d1b27caf68aae9e3ac82f1477929014d5b917657d0eb49478cb670
signature:    304402203609e17b84f6a7d30c80bfa610b5b4542f32a8a0d5447a12fb1366d7f01cc44a022057

The serialized signed transaction is: 01000000000102fff7f7881a8099afa6940d42d1e7f6362bec3817

nVersion:  01000000
marker:    00
flag:      01
txin:      02 fff7f7881a8099afa6940d42d1e7f6362bec38171ea3edf433541db4e4ad969f 00000000 4948
ef51e1b804cc89d182d279655c3aa89e815b1b309fe287d9b2b55d57b90ec68a 01000000 00 ffffffff
txout:     02 202cb20600000000 1976a9148280b37df378db99f66f85c95a783a76ac7a6d5988ac
9093510d00000000 1976a9143bde42dbee7e4dbe6a21b2d50ce2f0167faa815988ac
witness    00
02 47304402203609e17b84f6a7d30c80bfa610b5b4542f32a8a0d5447a12fb1366d7f01cc44a0220573a954c451
nLockTime: 11000000
```

### P2SH-P2WPKH

```
 The following is an unsigned transaction: 0100000001db6b1b20aa0fd7b23880be2ecbd4a98130974cf

nVersion:  01000000
txin:      01 db6b1b20aa0fd7b23880be2ecbd4a98130974cf4748fb66092ac4d3ceb1a5477 01000000 00 f
txout:     02 b8b4eb0b00000000 1976a914a457b684d7f0d539a46a45bbc043f35b59d0d96388ac
0008af2f00000000 1976a914fd270b1ee6abcaea97fea7ad0402e8bd8ad6d77c88ac
nLockTime: 92040000

The input comes from a P2SH-P2WPKH witness program:
scriptPubKey : a9144733f37cf4db86fbc2efed2500b4f4e49f31202387, value: 10
redeemScript : 001479091972186c449eb1ded22b78e40d009bdf0089
private key  : eb696a065ef48a2192da5b28b694f87544b30fae8327c4510137a922f32c6dcf
public key   : 03ad1d8e89212f0b92c74d23bb710c00662ad1470198ac48c43f7d6f93a2a26873

To sign it with a nHashType of 1 (SIGHASH_ALL):

hashPrevouts:
dSHA256(db6b1b20aa0fd7b23880be2ecbd4a98130974cf4748fb66092ac4d3ceb1a547701000000)
= b0287b4a252ac05af83d2dcef00ba313af78a3e9c329afa216eb3aa2a7b4613a

hashSequence:
dSHA256(feffffff)
= 18606b350cd8bf565266bc352f0caddcf01e8fa789dd8a15386327cf8cabe198

hashOutputs:
dSHA256(b8b4eb0b000000001976a914a457b684d7f0d539a46a45bbc043f35b59d0d96388ac0008af2f0000000
= de984f44532e2173ca0d64314fcefe6d30da6f8cf27bafa706da61df8a226c83

hash preimage: 01000000b0287b4a252ac05af83d2dcef00ba313af78a3e9c329afa216eb3aa2a7b4613a18606

nVersion:     01000000
hashPrevouts: b0287b4a252ac05af83d2dcef00ba313af78a3e9c329afa216eb3aa2a7b4613a
hashSequence: 18606b350cd8bf565266bc352f0caddcf01e8fa789dd8a15386327cf8cabe198
outpoint:     db6b1b20aa0fd7b23880be2ecbd4a98130974cf4748fb66092ac4d3ceb1a547701000000
scriptCode:   1976a91479091972186c449eb1ded22b78e40d009bdf008988ac
amount:       00ca9a3b00000000
nSequence:    feffffff
hashOutputs:  de984f44532e2173ca0d64314fcefe6d30da6f8cf27bafa706da61df8a226c83
nLockTime:    92040000
nHashType:    01000000

sigHash:      64f3b0f4dd2bb3aa1ce8566d220cc74dda9df97d8490cc81d89d735c92e59fb6
signature:    3044022047ac8e878352d3ebbde1c94ce3a10d057c24175747116f8288e5d794d12d482f022021
```

```
The serialized signed transaction is: 01000000000101db6b1b20aa0fd7b23880be2ecbd4a98130974cf4
nVersion:  01000000
marker:    00
flag:      01
txin:      01 db6b1b20aa0fd7b23880be2ecbd4a98130974cf4748fb66092ac4d3ceb1a5477 01000000 1716
txout:     02 b8b4eb0b00000000 1976a914a457b684d7f0d539a46a45bbc043f35b59d0d96388ac
0008af2f00000000 1976a914fd270b1ee6abcaea97fea7ad0402e8bd8ad6d77c88ac
witness    02 473044022047ac8e878352d3ebbde1c94ce3a10d057c24175747116f8288e5d794d12d482f0220
nLockTime: 92040000
```

### Native P2WSH

This example shows how `OP_CODESEPARATOR` and out-of-range `SIGHASH_SINGLE` are processed:

```
 The following is an unsigned transaction:
0100000002fe3dc9208094f3ffd12645477b3dc56f60ec4fa8e6f5d67c565d1c6b9216b36e0000000000ffffffff

nVersion:  01000000
txin:      02 fe3dc9208094f3ffd12645477b3dc56f60ec4fa8e6f5d67c565d1c6b9216b36e 00000000 00 f
0815cf020f013ed6cf91d29f4202e8a58726b1ac6c79da47c23d1bee0a6925f8 00000000 00 ffffffff
txout:     01 00f2052a01000000 1976a914a30741f8145e5acadf23f751864167f32e0963f788ac
nLockTime: 00000000

The first input comes from an ordinary P2PK:
scriptPubKey: 21036d5c20fa14fb2f635474c1dc4ef5909d4568e5569b79fc94d3448486e14685f8ac value:
private key:  b8f28a772fccbf9b4f58a4f027e07dc2e35e7cd80529975e292ea34f84c4580c
signature:    304402200af4e47c9b9629dbecc21f73af989bdaa911f7e6f6c2e9394588a3aa68f81e9902204f

The second input comes from a native P2WSH witness program:
scriptPubKey : 00205d1b56b63d714eebe542309525f484b7e9d6f686b3781b6f61ef925d66d6f6a0, value:
witnessScript: 21026dccc749adc2a9d0d89497ac511f760f45c47dc5ed9cf352a58ac706453880aeadab21025
<026dccc749adc2a9d0d89497ac511f760f45c47dc5ed9cf352a58ac706453880ae> CHECKSIGVERIFY CODESEPA

To sign it with a nHashType of 3 (SIGHASH_SINGLE):

hashPrevouts:
dSHA256(fe3dc9208094f3ffd12645477b3dc56f60ec4fa8e6f5d67c565d1c6b9216b36e000000000815cf020f01
= ef546acf4a020de3898d1b8956176bb507e6211b5ed3619cd08b6ea7e2a09d41

nVersion:     01000000
hashPrevouts: ef546acf4a020de3898d1b8956176bb507e6211b5ed3619cd08b6ea7e2a09d41
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     0815cf020f013ed6cf91d29f4202e8a58726b1ac6c79da47c23d1bee0a6925f800000000
scriptCode:   (see below)
amount:       0011102401000000
```

```
nSequence:    ffffffff
hashOutputs:  0000000000000000000000000000000000000000000000000000000000000000 (this is the
nLockTime:    00000000
nHashType:    03000000

scriptCode:  4721026dccc749adc2a9d0d89497ac511f760f45c47dc5ed9cf352a58ac706453880aeadab21025
^^
(please note that the not-yet-executed OP_CODESEPARATOR is not removed from the scriptCode)
preimage:    01000000ef546acf4a020de3898d1b8956176bb507e6211b5ed3619cd08b6ea7e2a09d41000000
sigHash:     82dde6e4f1e94d02c2b7ad03d2115d691f48d064e9d52f58194a6637e4194391
public key:  026dccc749adc2a9d0d89497ac511f760f45c47dc5ed9cf352a58ac706453880ae
private key: 8e02b539b1500aa7c81cf3fed177448a546f19d2be416c0c61ff28e577d8d0cd
signature:   3044022027dc95ad6b740fe5129e7e62a75dd00f291a2aeb1200b84b09d9e3789406b6c002201a9

scriptCode:  23210255a9626aebf5e29c0e6538428ba0d1dcf6ca98ffdf086aa8ced5e0d0215ea465ac
(everything up to the last executed OP_CODESEPARATOR, including that OP_CODESEPARATOR, are r
preimage:    01000000ef546acf4a020de3898d1b8956176bb507e6211b5ed3619cd08b6ea7e2a09d41000000
sigHash:     fef7bd749cce710c5c052bd796df1af0d935e59cea63736268bcbe2d2134fc47
public key:  0255a9626aebf5e29c0e6538428ba0d1dcf6ca98ffdf086aa8ced5e0d0215ea465
private key: 86bf2ed75935a0cbef03b89d72034bb4c189d381037a5ac121a70016db8896ec
signature:   304402200de66acf4527789bfda55fc5459e214fa6083f936b430a762c629656216805ac020396

The serialized signed transaction is: 01000000000102fe3dc9208094f3ffd12645477b3dc56f60ec4fa8
```

This example shows how unexecuted `OP_CODESEPARATOR` is processed, and `SINGLE|ANYONECANPAY` does not commit to the input index:

```
 The following is an unsigned transaction:
0100000002e9b542c5176808107ff1df906f46bb1f2583b16112b95ee5380665ba7fcfc0010000000000ffffffff

nVersion:  01000000
txin:      02 e9b542c5176808107ff1df906f46bb1f2583b16112b95ee5380665ba7fcfc001 00000000 00 f
80e68831516392fcd100d186b3c2c7b95c80b53c77e77c35ba03a66b429a2a1b 00000000 00 ffffffff
txout:     02 8096980000000000 1976a914de4b231626ef508c9a74a8517e6783c0546d6b2888ac
8096980000000000 1976a9146648a8cd4531e1ec47f35916de8e259237294d1e88ac
nLockTime: 00000000

The first input comes from a native P2WSH witness program:
scriptPubKey: 0020ba468eea561b26301e4cf69fa34bde4ad60c81e70f059f045ca9a79931004a4d value: 0.
witnessScript:0063ab68210392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98ac
0 IF CODESEPARATOR ENDIF <0392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98

The second input comes from a native P2WSH witness program:
scriptPubKey: 0020d9bbfbe56af7c4b7f960a70d7ea107156913d9e5a26b0a71429df5e097ca6537 value: 0.
witnessScript:5163ab68210392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98ac
1 IF CODESEPARATOR ENDIF <0392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98
```

```
To sign it with a nHashType of 0x83 (SINGLE|ANYONECANPAY):

nVersion:     01000000
hashPrevouts: 0000000000000000000000000000000000000000000000000000000000000000
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     (see below)
scriptCode:   (see below)
amount:       ffffff0000000000
nSequence:    ffffffff
hashOutputs:  (see below)
nLockTime:    00000000
nHashType:    83000000

outpoint:    e9b542c5176808107ff1df906f46bb1f2583b16112b95ee5380665ba7fcfc00100000000
scriptCode:  270063ab68210392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98a
(since the OP_CODESEPARATOR is not executed, nothing is removed from the scriptCode)
hashOutputs: b258eaf08c39fbe9fbac97c15c7e7adeb8df142b0df6f83e017f349c2b6fe3d2
preimage:    0100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
sigHash:     e9071e75e25b8a1e298a72f0d2e9f4f95a0f5cdf86a533cda597eb402ed13b3a
public key:  0392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98
private key: f52b3484edd96598e02a9c89c4492e9c1e2031f471c49fd721fe68b3ce37780d
signature:   3045022100f6a10b8604e6dc910194b79ccfc93e1bc0ec7c03453caaa8987f7d6c341356600220

outpoint:    80e68831516392fcd100d186b3c2c7b95c80b53c77e77c35ba03a66b429a2a1b00000000
scriptCode:  2468210392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98ac
(everything up to the last executed OP_CODESEPARATOR, including that OP_CODESEPARATOR, are r
hashOutputs: 91ea93dd77f702b738ebdbf3048940a98310e869a7bb8fa2c6cb3312916947ca
preimage:    0100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
sigHash:     cd72f1f1a433ee9df816857fad88d8ebd97e09a75cd481583eb841c330275e54
public key:  0392972e2eb617b2388771abe27235fd5ac44af8e61693261550447a4c3e39da98
private key: f52b3484edd96598e02a9c89c4492e9c1e2031f471c49fd721fe68b3ce37780d
signature:   30440220032521802a76ad7bf74d0e2c218b72cf0cbc867066e2e53db905ba37f130397e022077

The serialized signed transaction is:
01000000000102e9b542c5176808107ff1df906f46bb1f2583b16112b95ee5380665ba7fcfc0010000000000fff
nVersion:  01000000
marker:    00
flag:      01
txin:      02 e9b542c5176808107ff1df906f46bb1f2583b16112b95ee5380665ba7fcfc001 00000000 00 f
80e68831516392fcd100d186b3c2c7b95c80b53c77e77c35ba03a66b429a2a1b 00000000 00 ffffffff
txout:     02 8096980000000000 1976a914de4b231626ef508c9a74a8517e6783c0546d6b2888ac
8096980000000000 1976a9146648a8cd4531e1ec47f35916de8e259237294d1e88ac
witness    02 483045022100f6a10b8604e6dc910194b79ccfc93e1bc0ec7c03453caaa8987f7d6c3413566002
02 473044022032521802a76ad7bf74d0e2c218b72cf0cbc867066e2e53db905ba37f130397e02207709e2188ec
nLockTime: 00000000
```

```
Since SINGLE|ANYONECANPAY does not commit to the input index, the signatures are still valid
0100000000010280e68831516392fcd100d186b3c2c7b95c80b53c77e77c35ba03a66b429a2a1b0000000000fff
nVersion:  01000000
marker:    00
flag:      01
txin:      02 80e68831516392fcd100d186b3c2c7b95c80b53c77e77c35ba03a66b429a2a1b 00000000 00 f
e9b542c5176808107ff1df906f46bb1f2583b16112b95ee5380665ba7fcfc001 00000000 00 ffffffff
txout:     02 8096980000000000 1976a9146648a8cd4531e1ec47f35916de8e259237294d1e88ac
8096980000000000 1976a914de4b231626ef508c9a74a8517e6783c0546d6b2888ac
witness    02 47304402200325218 02a76ad7bf74d0e2c218b72cf0cbc867066e2e53db905ba37f130397e0220
02 483045022100f6a10b8604e6dc910194b79ccfc93e1bc0ec7c03453caaa8987f7d6c3413566002206216229ec
nLockTime: 00000000
```

### P2SH-P2WSH

This example is a P2SH-P2WSH 6-of-6 multisig witness program signed with 6 different `SIGHASH` types.

```
 The following is an unsigned transaction: 010000000136641869ca081e70f394c6948e8af409e18b619

nVersion:  01000000
txin:      01 36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e 01000000 00 f
txout:     02 00e9a43500000000 1976a914389ffce9cd9ae88dcc0631e88a821ffdbe9bfe2688ac
c0832f0500000000 1976a9147480a33f950689af511e6e84c138dbbd3c3ee41588ac
nLockTime: 00000000

The input comes from a P2SH-P2WSH 6-of-6 multisig witness program:
scriptPubKey : a9149993a429037b5d912407a71c252019287b8d27a587, value: 9.87654321
redeemScript : 0020a16b5755f7f6f96dbd65f5f0d6ab9418b89af4b1f14a1bb8a09062c35f0dcb54
witnessScript: 56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b28

hashPrevouts:
dSHA256(36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000)
= 74afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31f7aaa0

hashSequence:
dSHA256(ffffffff)
= 3bb13029ce7b1f559ef5e747fcac439f1455a2ec7c5f09b72290795e70665044

hashOutputs for ALL:
dSHA256(00e9a435000000001976a914389ffce9cd9ae88dcc0631e88a821ffdbe9bfe2688acc0832f050000000
= bc4d309071414bed932f98832b27b4d76dad7e6c1346f487a8fdbb8eb90307cc

hashOutputs for SINGLE:
dSHA256(00e9a435000000001976a914389ffce9cd9ae88dcc0631e88a821ffdbe9bfe2688ac)
= 9efe0c13a6b16c14a41b04ebe6a63f419bdacb2f8705b494a43063ca3cd4f708
```

```
hash preimage for ALL: 0100000074afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31f7a
nVersion:     01000000
hashPrevouts: 74afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31f7aaa0
hashSequence: 3bb13029ce7b1f559ef5e747fcac439f1455a2ec7c5f09b72290795e70665044
outpoint:     36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000
scriptCode:   cf56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b2
amount:       b168de3a00000000
nSequence:    ffffffff
hashOutputs:  bc4d309071414bed932f98832b27b4d76dad7e6c1346f487a8fdbb8eb90307cc
nLockTime:    00000000
nHashType:    01000000
sigHash:      185c0be5263dce5b4bb50a047973c1b6272bfbd0103a89444597dc40b248ee7c
public key:   0307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba3
private key:  730fff80e1413068a05b57d6a58261f07551163369787f349438ea38ca80fac6
signature:    304402206ac44d672dac41f9b00e28f4df20c52eeb087207e8d758d76d92c6fab3b73e2b022036

hash preimage for NONE: 0100000074afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31f7
nVersion:     01000000
hashPrevouts: 74afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31f7aaa0
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000
scriptCode:   cf56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b2
amount:       b168de3a00000000
nSequence:    ffffffff
hashOutputs:  0000000000000000000000000000000000000000000000000000000000000000
nLockTime:    00000000
nHashType:    02000000
sigHash:        e9733bc60ea13c95c6527066bb975a2ff29a925e80aa14c213f686cbae5d2f36
public key:     03b28f0c28bfab54554ae8c658ac5c3e0ce6e79ad336331f78c428dd43eea8449b
private key:    11fa3d25a17cbc22b29c44a484ba552b5a53149d106d3d853e22fdd05a2d8bb3
signature:      3044022068c7946a43232757cbdf9176f009a928e1cd9a1a8c212f15c1e11ac9f2925d900220

hash preimage for SINGLE: 0100000074afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31
nVersion:     01000000
hashPrevouts: 74afdc312af5183c4198a40ca3c1a275b485496dd3929bca388c4b5e31f7aaa0
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000
scriptCode:   cf56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b2
amount:       b168de3a00000000
nSequence:    ffffffff
hashOutputs:  9efe0c13a6b16c14a41b04ebe6a63f419bdacb2f8705b494a43063ca3cd4f708
nLockTime:    00000000
nHashType:    03000000
sigHash:        1e1f1c303dc025bd664acb72e583e933fae4cff9148bf78c157d1e8f78530aea
public key:     034b8113d703413d57761b8b9781957b8c0ac1dfe69f492580ca4195f50376ba4a
```

```
private key:    77bf4141a87d55bdd7f3cd0bdccf6e9e642935fec45f2f30047be7b799120661
signature:      3044022059ebf56d98010a932cf8ecfec54c48e6139ed6adb0728c09cbe1e4fa0915302e022

hash preimage for ALL|ANYONECANPAY: 0100000000000000000000000000000000000000000000000000000000000000
nVersion:     01000000
hashPrevouts: 0000000000000000000000000000000000000000000000000000000000000000
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000
scriptCode:   cf56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b2
amount:       b168de3a00000000
nSequence:    ffffffff
hashOutputs:  bc4d309071414bed932f98832b27b4d76dad7e6c1346f487a8fdbb8eb90307cc
nLockTime:    00000000
nHashType:    81000000
sigHash:        2a67f03e63a6a422125878b40b82da593be8d4efaafe88ee528af6e5a9955c6e
public key:     033400f6afecb833092a9a21cfdf1ed1376e58c5d1f47de74683123987e967a8f4
private key:    14af36970f5025ea3e8b5542c0f8ebe7763e674838d08808896b63c3351ffe49
signature:      3045022100fbefd94bd0a488d50b79102b5dad4ab6ced30c4069f1eaa69a4b5a763414067e02

hash preimage for NONE|ANYONECANPAY: 0100000000000000000000000000000000000000000000000000000000000000
nVersion:     01000000
hashPrevouts: 0000000000000000000000000000000000000000000000000000000000000000
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000
scriptCode:   cf56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b2
amount:       b168de3a00000000
nSequence:    ffffffff
hashOutputs:  0000000000000000000000000000000000000000000000000000000000000000
nLockTime:    00000000
nHashType:    82000000
sigHash:        781ba15f3779d5542ce8ecb5c18716733a5ee42a6f51488ec96154934e2c890a
public key:     03a6d48b1131e94ba04d9737d61acdaa1322008af9602b3b14862c07a1789aac16
private key:    fe9a95c19eef81dde2b95c1284ef39be497d128e2aa46916fb02d552485e0323
signature:      3045022100a5263ea0553ba89221984bd7f0b13613db16e7a70c549a86de0cc0444141a40702

hash preimage for SINGLE|ANYONECANPAY: 0100000000000000000000000000000000000000000000000000000000000000
nVersion:     01000000
hashPrevouts: 0000000000000000000000000000000000000000000000000000000000000000
hashSequence: 0000000000000000000000000000000000000000000000000000000000000000
outpoint:     36641869ca081e70f394c6948e8af409e18b619df2ed74aa106c1ca29787b96e01000000
scriptCode:   cf56210307b8ae49ac90a048e9b53357a2354b3334e9c8bee813ecb98e99a7e07e8c3ba32103b2
amount:       b168de3a00000000
nSequence:    ffffffff
hashOutputs:  9efe0c13a6b16c14a41b04ebe6a63f419bdacb2f8705b494a43063ca3cd4f708
nLockTime:    00000000
nHashType:    83000000
```

```
sigHash:        511e8e52ed574121fc1b654970395502128263f62662e076dc6baf05c2e6a99b
public key:     02d8b661b0b3302ee2f162b09e07a55ad5dfbe673a9f01d9f0c19617681024306b
private key:    428a7aee9f0c2af0cd19af3cf1c78149951ea528726989b2e83e4778d2c3f890
signature:      30440220525406a1482936d5a21888260dc165497a90a15669636d8edca6b9fe490d309c022(

The serialized signed transaction is: 0100000000010136641869ca081e70f394c6948e8af409e18b619c
```

### No FindAndDelete

These examples show that `FindAndDelete` for the signature is not applied. The transactions are generated in an unconventional way. Instead of signing using a private key, the signatures are pre-determined as part of `witnessScript`. The public keys are generated with key recovery, using the fixed signatures and the `sighash` defined in this proposal. Therefore, the private keys are unknown.

```
 The following is an unsigned transaction: 010000000169c12106097dc2e0526493ef67f21269fe888ef

nVersion:  01000000
txin:      01 69c12106097dc2e0526493ef67f21269fe888ef05c7a3a5dacab38e1ac8387f1 4c1d0000 00 f
txout:     01 0100000000000000 00
nLockTime: 00000000

The input comes from a P2WSH witness program:
scriptPubKey : 00209e1be07558ea5cc8e02ed1d80c0911048afad949affa36d5c3951e3159dbea19, value:
redeemScript : OP_CHECKSIGVERIFY <0x30450220487fb382c4974de3f7d834c1b617fe15860828c7f9645449
ad4830450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc9022100baf95feb48

To sign it with a nHashType of 1 (SIGHASH_ALL):

hashPrevouts:
dSHA256(69c12106097dc2e0526493ef67f21269fe888ef05c7a3a5dacab38e1ac8387f14c1d0000)
= b67c76d200c6ce72962d919dc107884b9d5d0e26f2aea7474b46a1904c53359f

hashSequence:
dSHA256(ffffffff)
= 3bb13029ce7b1f559ef5e747fcac439f1455a2ec7c5f09b72290795e70665044

hashOutputs:
dSHA256(0100000000000000000)
= e5d196bfb21caca9dbd654cafb3b4dc0c4882c8927d2eb300d9539dd0b934228

hash preimage: 01000000b67c76d200c6ce72962d919dc107884b9d5d0e26f2aea7474b46a1904c53359f3bb13

nVersion:     01000000
hashPrevouts: b67c76d200c6ce72962d919dc107884b9d5d0e26f2aea7474b46a1904c53359f
hashSequence: 3bb13029ce7b1f559ef5e747fcac439f1455a2ec7c5f09b72290795e70665044
```

```
outpoint:     69c12106097dc2e0526493ef67f21269fe888ef05c7a3a5dacab38e1ac8387f14c1d0000
scriptCode:   4aad4830450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc9
amount:       400d030000000000
nSequence:    ffffffff
hashOutputs:  e5d196bfb21caca9dbd654cafb3b4dc0c4882c8927d2eb300d9539dd0b934228
nLockTime:    00000000
nHashType:    01000000

sigHash:      71c9cd9b2869b9c70b01b1f0360c148f42dee72297db312638df136f43311f23
signature:    30450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc902210
pubkey:       02a9781d66b61fb5a7ef00ac5ad5bc6ffc78be7b44a566e3c87870e1079368df4c

The serialized signed transaction is: 0100000000010169c12106097dc2e0526493ef67f21269fe888ef0

   nVersion:  01000000
marker:    00
flag:      01
txin:      01 69c12106097dc2e0526493ef67f21269fe888ef05c7a3a5dacab38e1ac8387f1 4c1d0000 00 f
txout:     01 0100000000000000 00
witness:   03 4830450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc90221
2102a9781d66b61fb5a7ef00ac5ad5bc6ffc78be7b44a566e3c87870e1079368df4c
4aad4830450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc9022100baf95feb
nLockTime: 00000000


 The following transaction is a OP_CHECKMULTISIGVERIFY version of the FindAndDelete examples

redeemScript:  OP_2 OP_CHECKMULTISIGVERIFY <30450220487fb382c4974de3f7d834c1b617fe15860828c7
hash preimage: 0100000039283953eb1e26994dde57b7f9362a79a8c523e2f8deba943c27e826a005f1e63bb13
sighash:       c1628a1e7c67f14ca0c27c06e4fdeec2e6d1a73c7a91d7c046ff83e835aebb72
witness:       07 00
4830450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc9022100baf95feb48f8
48304502205286f726690b2e9b0207f0345711e63fa7012045b9eb0f19c2458ce1db90cf43022100e89f17f86abc
0102
2102966f109c54e85d3aee8321301136cedeb9fc710fdef58a9de8a73942f8e567c0
21034ffc99dd9a79dd3cb31e2ab3e0b09e0e67db41ac068c625cd1f491576016c84e
9552af4830450220487fb382c4974de3f7d834c1b617fe15860828c7f96454490edd6d891556dcc9022100baf95f
```

The new serialization format is described in BIP144 [9]

## Deployment

This proposal is deployed with Segregated Witness softfork (BIP 141)

[9] BIP144: Segregated Witness (Peer Services)

## Backward compatibility

As a soft fork, older software will continue to operate without modification. Non-upgraded nodes, however, will not see nor validate the witness data and will consider all witness programs, including the redefined sigops, as anyone-can-spend scripts.

## Reference Implementation

https://github.com/bitcoin/bitcoin/pull/8149

## References

## Copyright