```
BIP: 19
Layer: Applications
Title: M-of-N Standard Transactions (Low SigOp)
Author: Luke Dashjr <luke+bip17@dashjr.org>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0019
Status: Rejected
Type: Standards Track
Created: 2012-01-30
License: BSD-2-Clause
```

Abstract

This BIP proposes M-of-N-signatures required transactions as a new 'standard' transaction type using the existing scripting system without significant modifications.

Copyright

Motivation

Enable secured wallets, escrow transactions, and other use cases where redeeming funds requires more than a single signature.

A couple of motivating use cases:

- A wallet secured by a "wallet protection service" (WPS). 2-of-2 signatures required transactions will be used, with one signature coming from the (possibly compromised) computer with the wallet and the second signature coming from the WPS. When sending protected bitcoins, the user's bitcoin client will contact the WPS with the proposed transaction and it can then contact the user for confirmation that they initiated the transaction and that the transaction details are correct. Details for how clients and WPS's communicate are outside the scope of this BIP. Side note: customers should insist that their wallet protection service provide them with copies of the private key(s) used to secure their wallets that they can safely store off-line, so that their coins can be spent even if the WPS goes out of business.
- Three-party escrow (buyer, seller and trusted dispute agent). 2-of-3 signatures required transactions will be used. The buyer and seller and agent will each provide a public key, and the buyer will then send coins into a 2-of-3 CHECKMULTISIG transaction and send the seller and the agent the transaction id. The seller will fulfill their obligation and then ask the buyer to co-sign a transaction (already signed by seller) that sends the tied-up coins to him (seller).

If the buyer and seller cannot agree, then the agent can, with the cooperation of either buyer or seller, decide what happens to the tied-up coins. Details of how buyer, seller, and agent communicate to gather signatures or public keys are outside the scope of this BIP.

Specification

Two new standard transaction types (scriptPubKey) that are relayed by clients and included in mined blocks.

N-of-N (all signatures required):

```
( {pubkey} OP_CHECKSIGVERIFY )*n
```

N-of-M (some signatures required):

```
{pubkey} OP_CHECKSIG  ( OP_SWAP {pubkey} OP_CHECKSIG OP_ADD )*(n-1) n OP_EQUAL
```

But only for n less than or equal to 3.

These transactions are redeemed using a standard scriptSig:

```
...signatures...
```

The current Satoshi bitcoin client does not relay or mine transactions with scriptSigs larger than 200 bytes; to accommodate 3-signature transactions, this will be increased to 500 bytes.

Templates

scriptPubKey:

```
{pubkey} OP_CHECKSIGVERIFY {pubkey} OP_CHECKSIGVERIFY
```

```
{pubkey} OP_CHECKSIGVERIFY {pubkey} OP_CHECKSIGVERIFY {pubkey} OP_CHECKSIGVERIFY
```

```
{pubkey} OP_CHECKSIG OP_SWAP {pubkey} OP_CHECKSIG OP_ADD {n} OP_EQUAL
```

```
{pubkey} OP_CHECKSIG OP_SWAP {pubkey} OP_CHECKSIG OP_ADD OP_SWAP {pubkey} OP_CHECKSIG OP_AI
```

scriptSig:

```
...signatures... up to 500 bytes
```

Rationale

OP_CHECKMULTISIG is already an enabled opcode, and is the most straightforward way to support several important use cases. This is already specified in BIP 0011. However, each OP_CHECKMULTISIG counts toward the block limit as 20 sigops, which only allows 1000 total multisig transactions in a block. Using OP_CHECKSIG only counts as 1 per signature, so can scale better.

Implementation

All used operations are already supported by old clients and miners as a non-standard transaction type.