SV开发中的那些坑

aaron67

aaron67@aaron67.cc

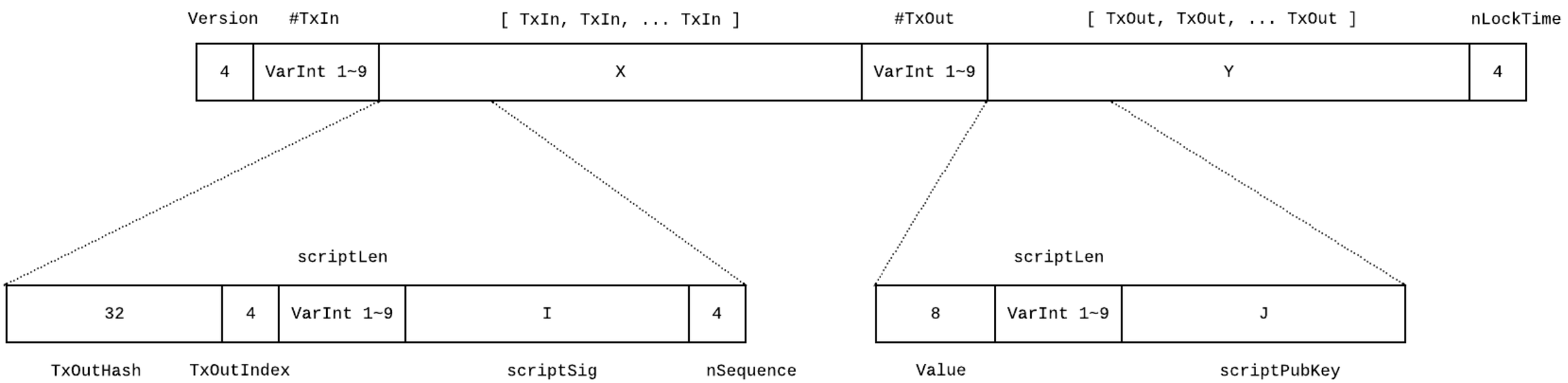
https://aaron67.cc/tags/bitcoin

192

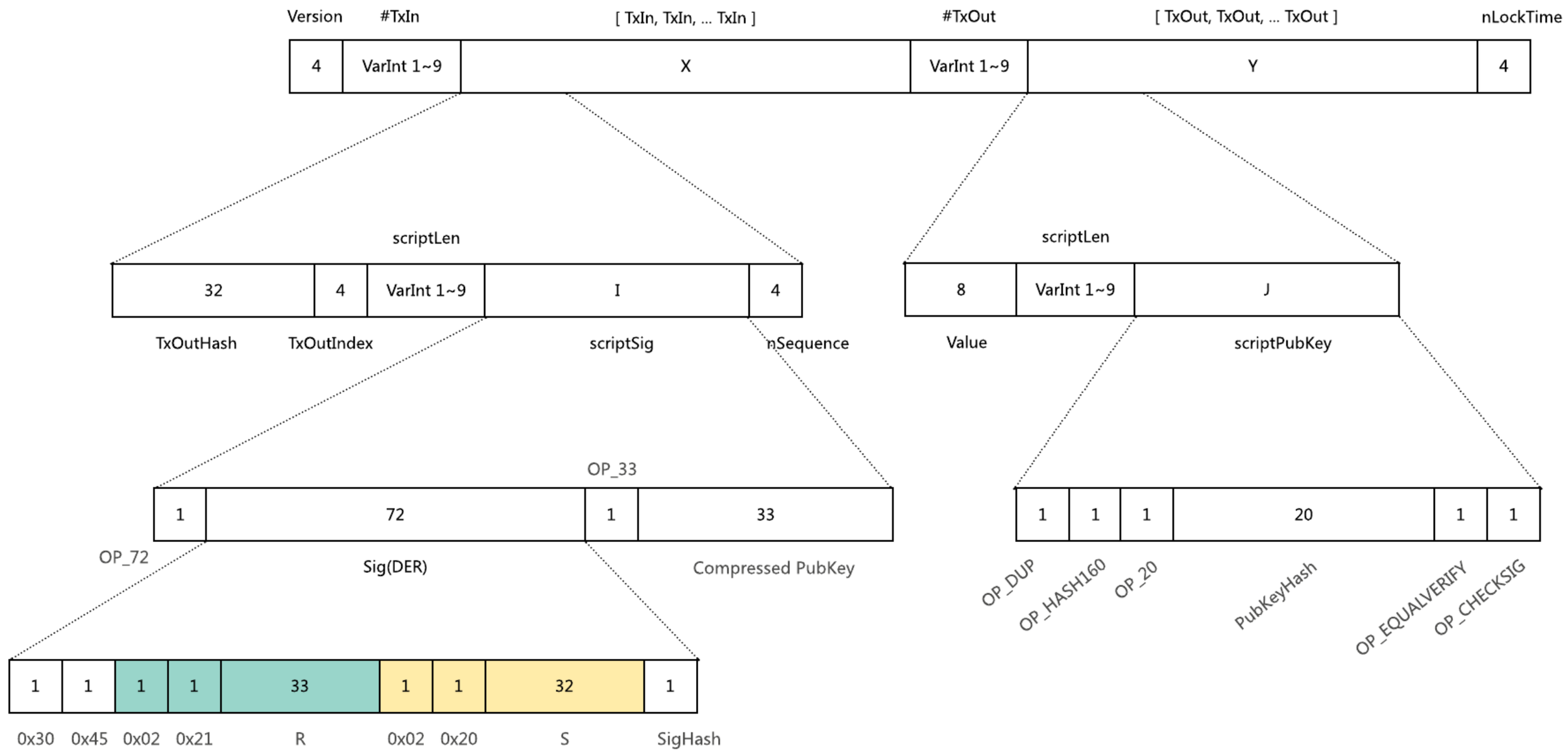
Tx

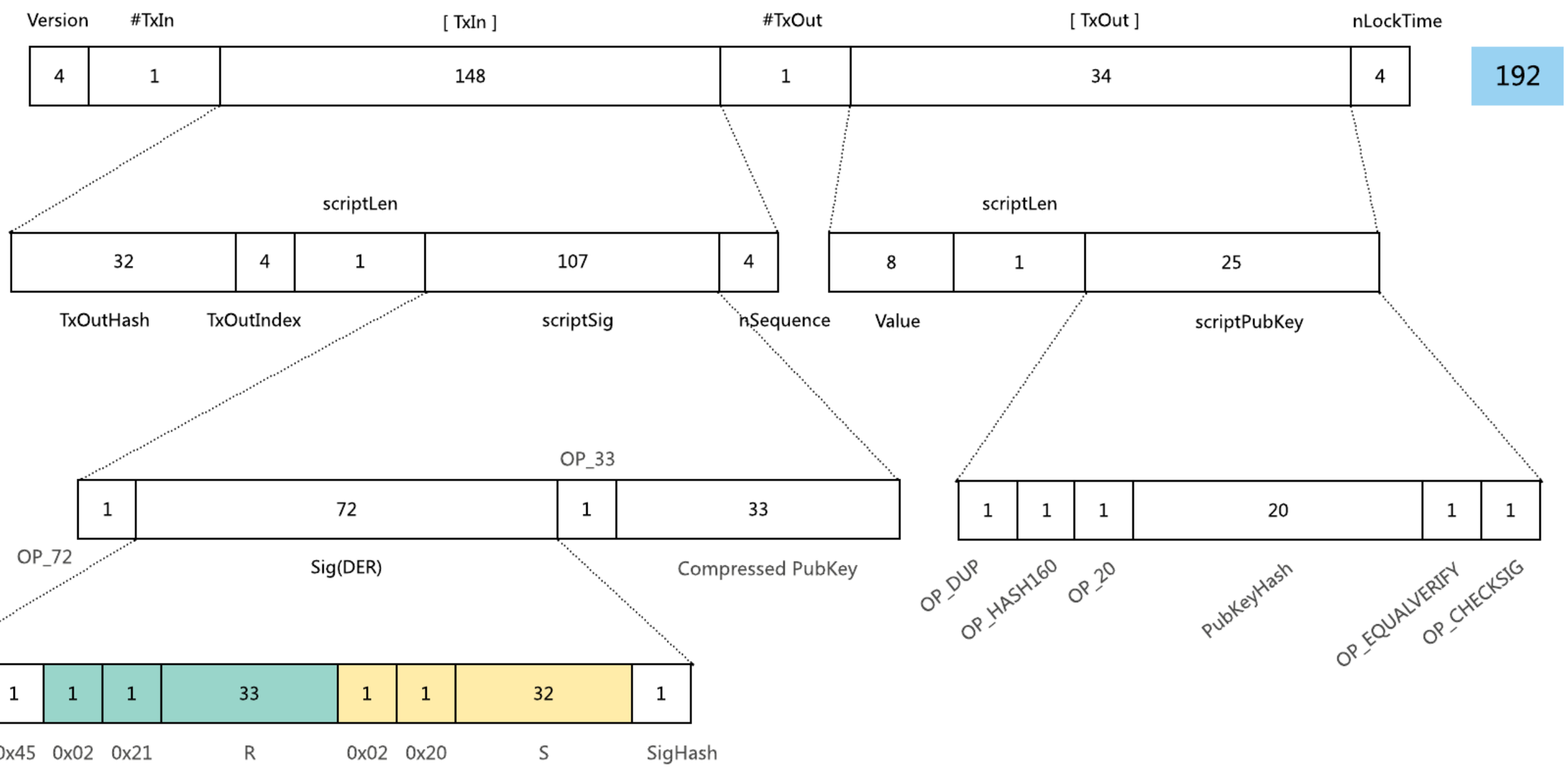
Version	#TxIn	[TxIn, TxIn, ... TxIn]	#TxOut	[TxOut, TxOut, ... TxOut]	nLockTime
4	VarInt 1~9	X	VarInt 1~9	Y	4

VarInt


Version
#TxIn
[TxIn, TxIn, ... TxIn]
#TxOut
[TxOut, TxOut, ... TxOut]
nLockTime
4
VarInt 1~9
X
VarInt 1~9
Y
4
scriptLen
32
4
VarInt 1~9
I
4
TxOutHash
TxOutIndex
scriptSig
nSequence
scriptLen
8
VarInt 1~9
J
Value
scriptPubKey

P2PKH


Version
#TxIn
[TxIn, TxIn, ... TxIn]
#TxOut
[TxOut, TxOut, ... TxOut]
nLockTime
4
VarInt 1~9
X
VarInt 1~9
Y
4
scriptLen
32
4
VarInt 1~9
I
4
TxOutHash
TxOutIndex
scriptSig
nSequence
OP_33
1
72
1
33
OP_72
Sig(DER)
Compressed PubKey
1
1
1
1
33
1
1
32
1
0x30
0x45
0x02
0x21
R
0x02
0x20
S
SigHash
scriptLen
8
VarInt 1~9
J
Value
scriptPubKey
1
1
1
20
1
1
OP_DUP
OP_HASH160
OP_20
PubKeyHash
OP_EQUALVERIFY
OP_CHECKSIG


Version
#TxIn
[TxIn]
#TxOut
[TxOut]
nLockTime
4
1
148
1
34
4
192
scriptLen
scriptLen
32
4
1
107
4
8
1
25
TxOutHash
TxOutIndex
scriptSig
nSequence
Value
scriptPubKey
OP_33
1
72
1
33
OP_72
Sig(DER)
Compressed PubKey
1
1
1
1
33
1
1
32
1
0x30
0x45
0x02
0x21
R
0x02
0x20
S
SigHash
1
1
1
20
1
1
OP_DUP
OP_HASH160
OP_20
PubKeyHash
OP_EQUALVERIFY
OP_CHECKSIG

Transaction

Transaction ID:

6495237c8716a1f4965f81ffc76a76c0b267f4b9c5b8a2f49274b9ad642ceaec

Status: 6 confirmations (in block 615,816)

Date: 2020-01-02 21:31

Amount sent: 0. BSV

Size: 192 bytes

Fee: 0.00000192 BSV (1. sat/B)

LockTime: 615815

Inputs (1)

0250c7b5...f9e38d50:0	1GVaDhhBPfFEvKFDb7suHrCiVk9LDhWNXS

Outputs (1)

12A8JRK1Wvazx713Q2LBsybvy32guXsefM	3.40789937

Copy | Save | Sign | Broadcast | Close

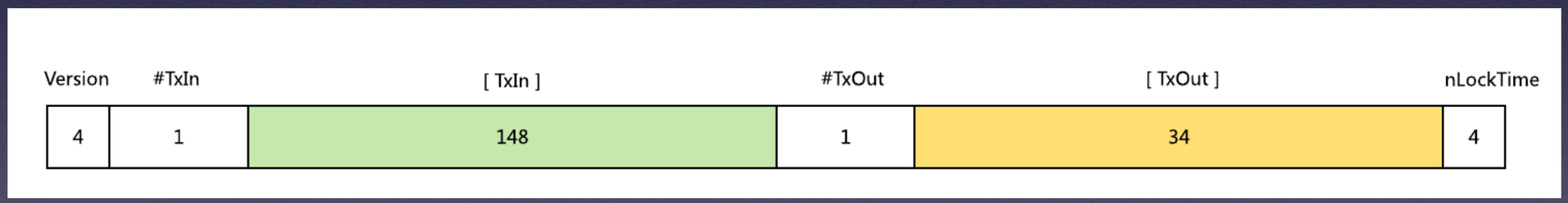
546

```
bool IsStandardTx(const Config &config, const CTransaction &tx, std::string &reason) {
    if (tx.nVersion > CTransaction::MAX_STANDARD_VERSION || tx.nVersion < 1) {
        reason = "version";
        return false;
    }

    // Extremely large transactions with lots of inputs can cost the network
    // almost as much to process as they cost the sender in fees, because
    // computing signature hashes is O(ninputs*txsize). Limiting transactions
    // to MAX_STANDARD_TX_SIZE mitigates CPU exhaustion attacks.
    unsigned int sz = tx.GetTotalSize();
    if (sz >= MAX_STANDARD_TX_SIZE) {
        reason = "tx-size";
        return false;
    }

    for (const CTxIn &txin : tx.vin) {
        // Biggest 'standard' txin is a 15-of-15 P2SH multisig with compressed
        // keys (remember the 520 byte limit on redeemScript size). That works
        // out to a (15*(33+1))+3=513 byte redeemScript, 513+1+15*(73+1)+3=1627
        // bytes of scriptSig, which we round off to 1650 bytes for some minor
        // future-proofing. That's also enough to spend a 20-of-20 CHECKMULTISIG
        // scriptPubKey, though such a scriptPubKey is not considered standard.
        if (txin.scriptSig.size() > 1650) {
            reason = "scriptsig-size";
            return false;
        }
        if (!txin.scriptSig.IsPushOnly()) {
            reason = "scriptsig-not-pushonly";
            return false;
        }
    }

    unsigned int nDataOut = 0;
    txnouttype whichType;
    for (const CTxOut &txout : tx.vout) {
        if (!::IsStandard(config, txout.scriptPubKey, whichType)) {
            reason = "scriptpubkey";
            return false;
        }

        if (whichType == TX_NULL_DATA) {
            nDataOut++;
        } else if ((whichType == TX_MULTISIG) && (!fIsBareMultisigStd)) {
            reason = "bare-multisig";
            return false;
        } else if (txout.IsDust(dustRelayFee)) {
            reason = "dust";
            return false;
        }
    }

    // only one OP_RETURN txout is permitted
    if (nDataOut > 1) {
        reason = "multi-op-return";
        return false;
    }

    return true;
}
```

```
153 class CTxOut {
154 public:
155     Amount nValue;
156     CScript scriptPubKey;
157
158     CTxOut() { SetNull(); }
159
160     CTxOut(Amount nValueIn, CScript scriptPubKeyIn)
161         : nValue(nValueIn), scriptPubKey(scriptPubKeyIn) {}
162
163     ADD_SERIALIZE_METHODS;
164
165     template <typename Stream, typename Operation>
166     inline void SerializationOp(Stream &s, Operation ser_action) {
167         READWRITE(nValue);
168         READWRITE(scriptPubKey);
169     }
170
171     void SetNull() {
172         nValue = Amount(-1);
173         scriptPubKey.clear();
174     }
175
176     bool IsNull() const { return (nValue == Amount(-1)); }
177
178     Amount GetDustThreshold(const CFeeRate &minRelayTxFee) const {
179         /**
180          * "Dust" is defined in terms of CTransaction::minRelayTxFee, which has
181          * units satoshis-per-kilobyte. If you'd pay more than 1/3 in fees to
182          * spend something, then we consider it dust. A typical spendable
183          * non-segwit txout is 34 bytes big, and will need a CTxIn of at least
184          * 148 bytes to spend: so dust is a spendable txout less than
185          * 546*minRelayTxFee/1000 (in satoshis). A typical spendable segwit
186          * txout is 31 bytes big, and will need a CTxIn of at least 67 bytes to
187          * spend: so dust is a spendable txout less than 294*minRelayTxFee/1000
188          * (in satoshis).
189          */
190         if (scriptPubKey.IsUnspendable()) return Amount(0);
191
192         size_t nSize = GetSerializeSize(*this, SER_DISK, 0);
193
194         // the 148 mentioned above
195         nSize += (32 + 4 + 1 + 107 + 4);
196
197         return 3 * minRelayTxFee.GetFee(nSize);
198     }
199
200     bool IsDust(const CFeeRate &minRelayTxFee) const {
201         return (nValue < GetDustThreshold(minRelayTxFee));
202     }
203
204     friend bool operator==(const CTxOut &a, const CTxOut &b) {
205         return (a.nValue == b.nValue && a.scriptPubKey == b.scriptPubKey);
206     }
207
208     friend bool operator!=(const CTxOut &a, const CTxOut &b) {
209         return !(a == b);
210     }
211
212     std::string ToString() const;
213 };
```

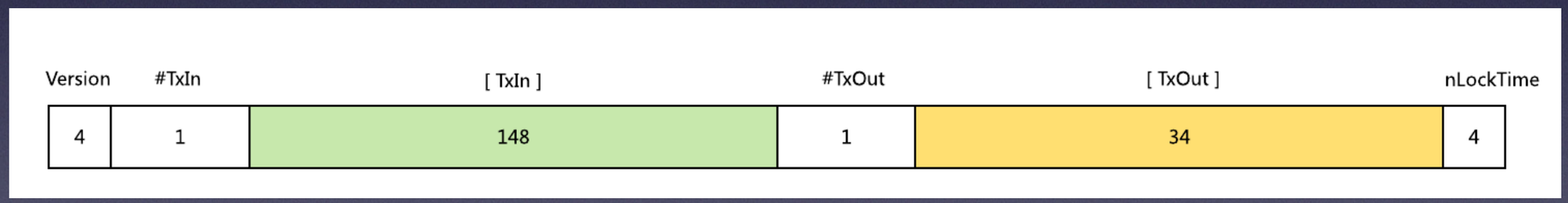
```
/**
 * "Dust" is defined in terms of CTransaction::minRelayTxFee, which has
 * units satoshis-per-kilobyte. If you'd pay more than 1/3 in fees to
 * spend something, then we consider it dust. A typical spendable
 * non-segwit txout is 34 bytes big, and will need a CTxIn of at least
 * 148 bytes to spend: so dust is a spendable txout less than
 * 546*minRelayTxFee/1000 (in satoshis). A typical spendable segwit
 * txout is 31 bytes big, and will need a CTxIn of at least 67 bytes to
 * spend: so dust is a spendable txout less than 294*minRelayTxFee/1000
 * (in satoshis).
 */
```

```
/**
 * "Dust" is defined in terms of CTransaction::minRelayTxFee, which has
 * units satoshis-per-kilobyte. If you'd pay more than 1/3 in fees to
 * spend something, then we consider it dust. A typical spendable
 * non-segwit txout is 34 bytes big, and will need a CTxIn of at least
 * 148 bytes to spend: so dust is a spendable txout less than
 * 546*minRelayTxFee/1000 (in satoshis). A typical spendable segwit
 * txout is 31 bytes big, and will need a CTxIn of at least 67 bytes to
 * spend: so dust is a spendable txout less than 294*minRelayTxFee/1000
 * (in satoshis).
 */
```



$$fee = 148 + 34 = 182 <= A_{utxo} / 3$$

```
/**
 * "Dust" is defined in terms of CTransaction::minRelayTxFee, which has
 * units satoshis-per-kilobyte. If you'd pay more than 1/3 in fees to
 * spend something, then we consider it dust. A typical spendable
 * non-segwit txout is 34 bytes big, and will need a CTxIn of at least
 * 148 bytes to spend: so dust is a spendable txout less than
 * 546*minRelayTxFee/1000 (in satoshis). A typical spendable segwit
 * txout is 31 bytes big, and will need a CTxIn of at least 67 bytes to
 * spend: so dust is a spendable txout less than 294*minRelayTxFee/1000
 * (in satoshis).
 */
```



$$fee = 148 + 34 = 182 <= A_{utxo} / 3$$

$$A_{utxo} >= 546$$

```
    if combine:
        # calculated_fee is in total satoshis.
        calculated_fee = estimate_tx_fee(len(unspents), num_outputs, fee, compressed, total_op_return_size)
        total_out = sum_outputs + calculated_fee
        unspents = unspents.copy()
        total_in += sum(unspent.amount for unspent in unspents)

    else:
        unspents = sorted(unspents, key=lambda x: x.amount)

        index = 0

        for index, unspent in enumerate(unspents):
            total_in += unspent.amount
            calculated_fee = estimate_tx_fee(len(unspents[:index + 1]), num_outputs, fee,
                                             compressed, total_op_return_size)
            total_out = sum_outputs + calculated_fee

            if total_in >= total_out:
                break

        unspents[:] = unspents[:index + 1]

    remaining = total_in - total_out

    # If the uxto less than dust (546) the miner will not relay that tx, even the service can successful return.
    # Here we put all the remnant (<546) to the miner in this case.
    # We could adjust here when new dust agreement reached in future.
    if remaining > DUST:
        outputs.append((leftover, remaining))
    elif remaining < 0:
        raise InsufficientFunds('Balance {} is less than {} (including '
                                'fee).'.format(total_in, total_out))

    outputs.extendleft(messages)

    return unspents, list(outputs)
```

BitSV 0.10.2

UTXO


14:17
70%
BSVRUN
钥匙 13
天选 0.279
挖矿 0.887
0
第1赛道 GO
1
0
Unexpected Error
An unexpected error has ocurred.
Please, try again in a few minutes.

可转账金额：0.00122853 BTC

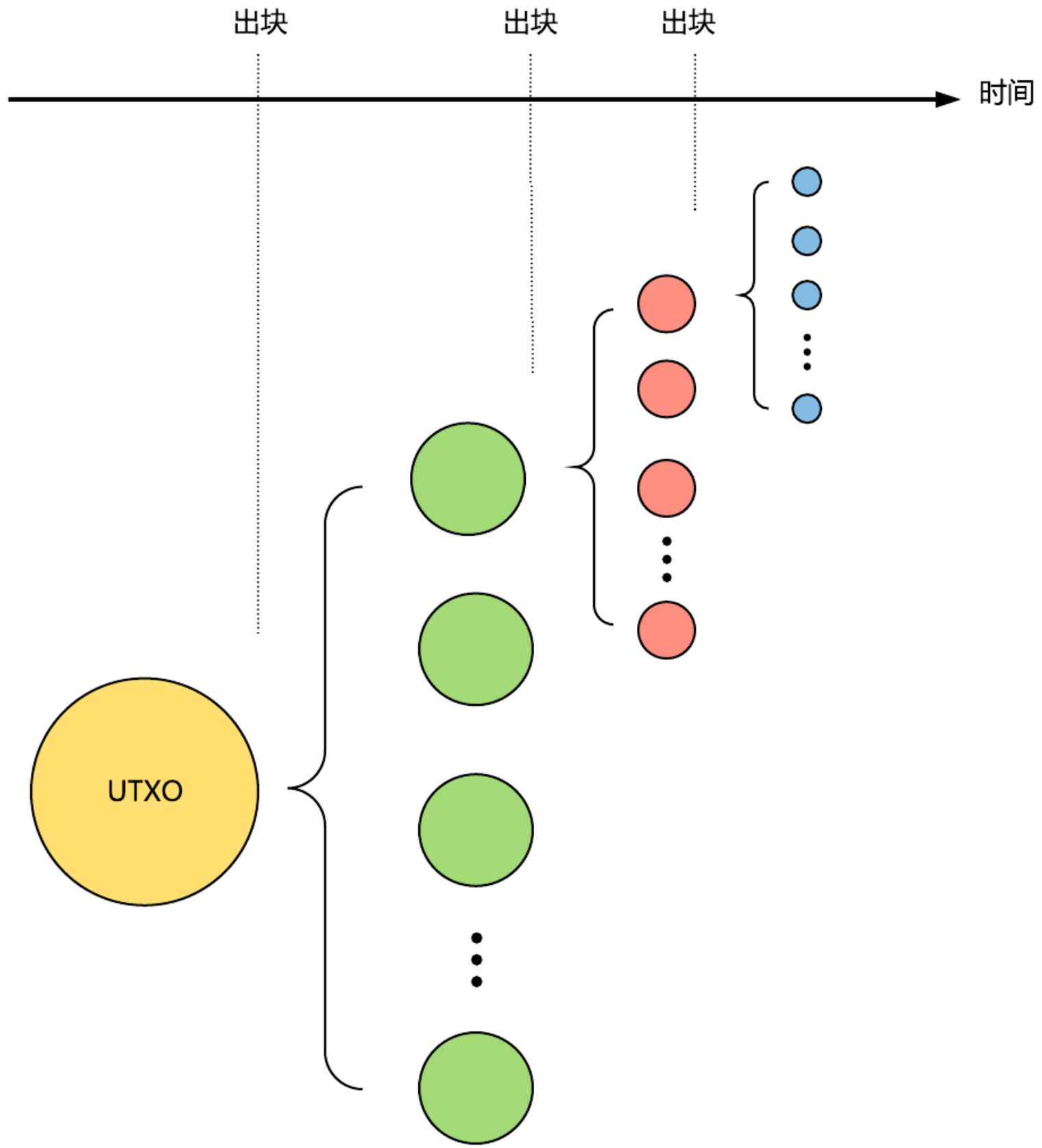
全部转账

-26: 64: too-long-mempool-chain

添加转账地址

转 账

拆分UTXO


出块
出块
出块
时间
UTXO

Your transaction history

Dec 29th

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

1EUR[redacted]VJWNnt	0.000007	606137	a2113f751b...:1
1EUR[redacted]VJWNnt	0.000007	606137	a017915002...:1
1EUR[redacted]VJWNnt	0.000007	606137	4b8a1a8f7c...:1
1EUR[redacted]VJWNnt	0.000007	606137	bd831915ba...:1
1EUR[redacted]VJWNnt	0.000007	606137	390b833ced...:1
1EUR[redacted]VJWNnt	0.000007	606137	40c343491e...:1
1EUR[redacted]VJWNnt	0.000007	606137	c34678994b...:1
1EUR[redacted]VJWNnt	0.000007	606137	602d9e53b3...:1
1EUR[redacted]VJWNnt	0.000007	606137	d6e0a074e5...:1
1EUR[redacted]VJWNnt	0.000007	606137	63fe6d8a6f...:1

Block 000000000000000005a05620a379faeda0759180b98c0a444637964d33b2a032 (#606137)

Timestamp 2019-10-27 05:19:38 utc

Version 1

Size 262 B

Confirmations 9,105

Fee Paid 0.000003 BSV ⇄ (0.00001 BSV - 0.000007 BSV)

Fee Rate 1.146 sat/B

1 Input, 2 Outputs

1	13U7PfoxtY[redacted] via 4698e9068c418fdc8bd4...[103]	0.00001 BSV ⇄	1	OP_RETURN	0
		0.00001 BSV ⇄			

Decode Report

ASCII Script Hex

j□gatling
bitcoinblocks.live"1EUR[redacted]aVJWNnt

合并UTXO

Your transaction was not sent: transaction is too large.

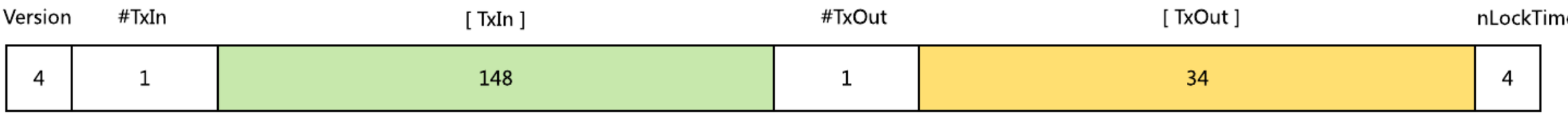
The server returned the following message, which may or may not help describe the problem. A malicious server may return misleading messages, so act on it at your own risk. In particular, do not download software from any links provided; the official ElectrumSV website is only https://electrumsv.io/.

(1, 'the transaction was rejected by network rules.
\n\n64: tx-
size\n[01000000fdd9022f6fd32f965a34727f25e6ef30b9bbf10e
17903016054047b65c6f5c6216c700010000006b483045022100d87
b17412ca965f26af84e54495f87a471a2127be56cf92b536b8d20cd

General info

Address	[redacted]
Balance	[redacted]
Last seen receiving	a minute ago
Transaction count	3,192,219
Output count / Unspent output count	3,192,216 / 29,491

Click to see more ↓



fee = 4 + 1 + 148 * 100 + 1 + 34 + 4 = 14844

Transaction

Transaction ID:

Unknown

Description:

Status: Unsigned

Amount sent: 0. BSV

Size: 14844 bytes

Fee: 0.00014844 BSV (1. sat/B)

LockTime: 615844

Inputs (100)

52068dd0...b95657dd:96	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ	0
52068dd0...b95657dd:97	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ	0
52068dd0...b95657dd:98	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ	0
52068dd0...b95657dd:99	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ	0

Outputs (1)

1QJcN55ExXXAHJjKxyt9omasAA3qC9Jjrg	3.40771535

Transaction

Transaction ID:

d608216b1c82f92262aecb01c13ed6ae922648081880b7c564e18d2e00f148d8

Description:

Status: Signed

Amount sent: 0. BSV

Size: 14793 bytes

Fee: 0.00014844 BSV (1.003448 sat/B)

LockTime: 615844

Inputs (100)

52068dd0...b95657dd:96	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ
52068dd0...b95657dd:97	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ
52068dd0...b95657dd:98	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ
52068dd0...b95657dd:99	1PzifurYAWQ99x1MMzAxFh7cPqQmsvTmAQ

Outputs (1)

1QJcN55ExXXAHJjKxyt9omasAA3qC9Jjrg	3.40771535

Transaction

Transaction ID:

9822c8fc11279a7953dd763a5c2b79bd8dff61e5da074cc0397940583098477 2

Description:

Status: Signed

Amount sent: 0. BSV

Size: 191 bytes

Fee: 0.00000192 BSV (1.005236 sat/B)

LockTime: 615845

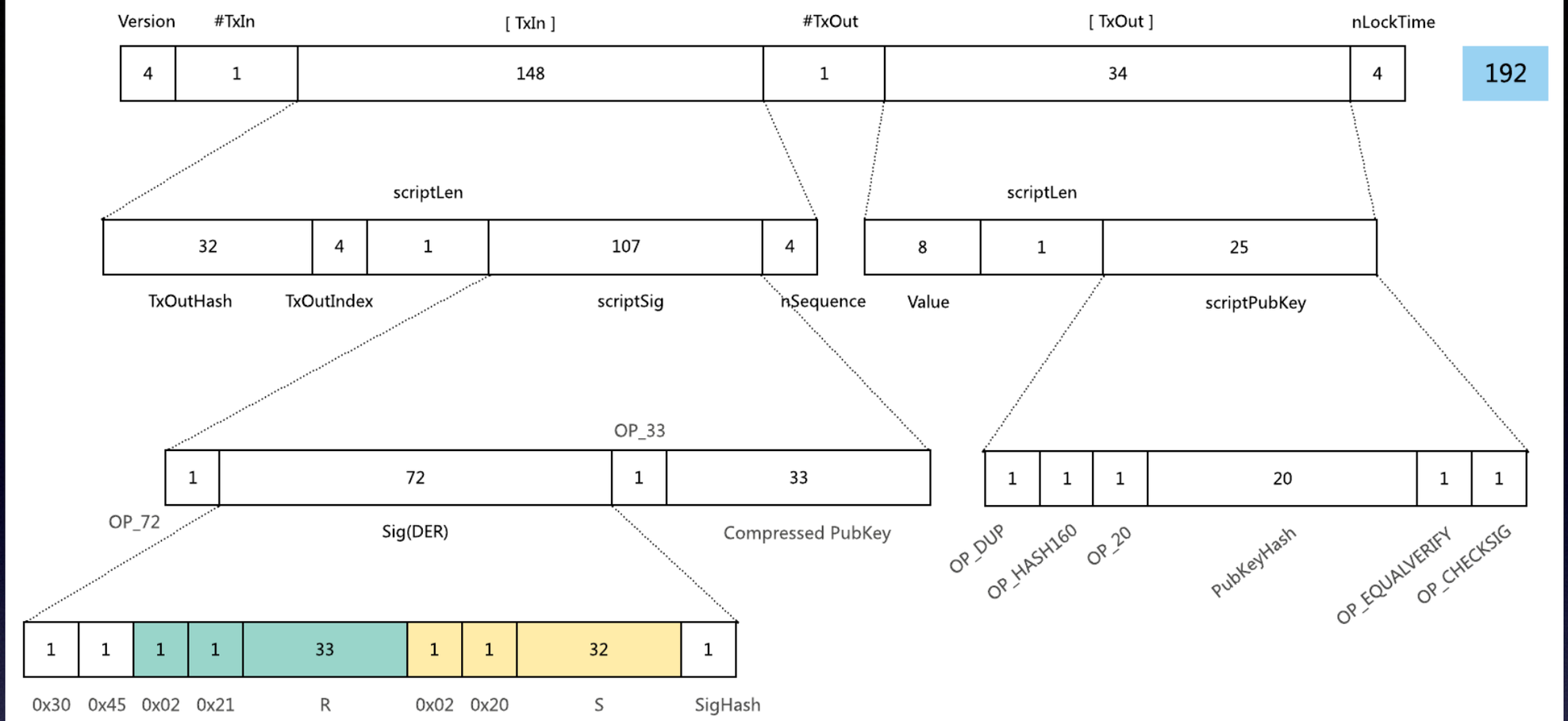
Inputs (1)

d608216b...00f148d8:0	1QJcN55ExXXAHJjKxyt9omasAA3qC9Jjrg	3.40771535

Outputs (1)

12RtKfRBHMhz164zBiDCLrHLf3aE66c6KJ	3.40771343

Copy Save Sign Broadcast Close



传送门1

传送门2

```
INPUT # 2:
hash:  ee 59 5b 71 cc 5a 1f b9 80 a7 7a f8 b9 62 53 4a e0 4 9b 29 6 fb e0 79 e 48 fa 71 e9 f0 22 99
index: 0
script length: 139
scriptSig:  48 30 45 2 21 0 fa f8 4d 50 e9 9d ee ef 7d 2c f9 f8 b4 60 b 8d e5 6f d7 88 d9 f6 65 21 a0 43 78 f9 b 49 a8 7
6 2 20 46 8 cd e7 76 fd a7 54 ce 87 1a bf f8 73 dc 4d 29 aa 34 c0 3a 50 d9 60 85 20 e 82 5b ae 73 a4 1 41 4 56 84 d9 b3
83 46 de b7 f9 3b 6b 82 82 dc f8 22 7b fc b7 29 13 d8 f4 c5 fa d9 98 7e 38 77 4 67 fe e2 6b 5a b 57 d0 ae f4 df 40 2 46
3e c1 f1 93 46 40 d6 90 5e ed e1 ed 28 bb 7e 43 2b cb d1
sequence:  ff ff ff ff

INPUT # 3:
hash:  5e 32 74 71 c5 bd be cc a4 5f cc bd 69 8a 47 94 24 a3 1c 99 a9 70 4b 68 e6 19 f6 b3 e3 b9 29 55
index: 1
script length: 138
scriptSig:  47 30 44 2 20 6f 36 1f b4 b9 7a ae a0 4f 18 cf 9f f8 1a 18 6e 48 ed 44 78 37 9e 6e 93 e0 ab 28 d4 d4 82 26 c
9 2 20 15 65 b7 3f 2f 3 ef fa cf 42 9e 7a 84 5 43 c5 77 34 75 18 f2 56 dc c3 de f8 55 8f a8 ef fc ef 1 41 4 d c0 d6 2b d
8 7d 2e 54 be 3f 95 c4 18 7f a3 d 48 d6 cd 0 43 19 78 40 1d d7 98 b7 45 7 d9 ad b4 57 fd 94 34 87 6f 56 27 35 ba ef b b9
 d6 e3 53 66 c2 c1 81 b6 d9 61 b8 36 2a d9 5a f7 26 aa
sequence:  ff ff ff ff
```

Your transaction history

Dec 29th

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Error
There was an problem during the broadcast of this transaction.

From: To:

Spam Wallet

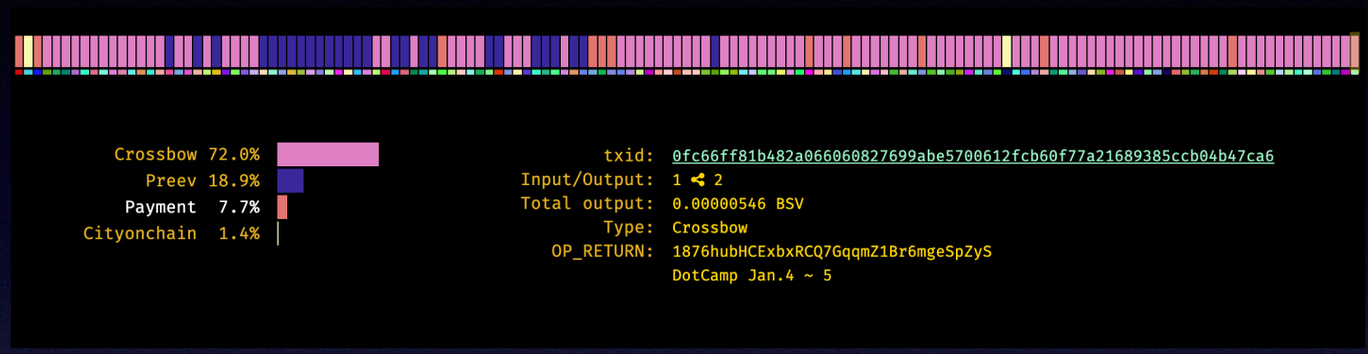
广播交易

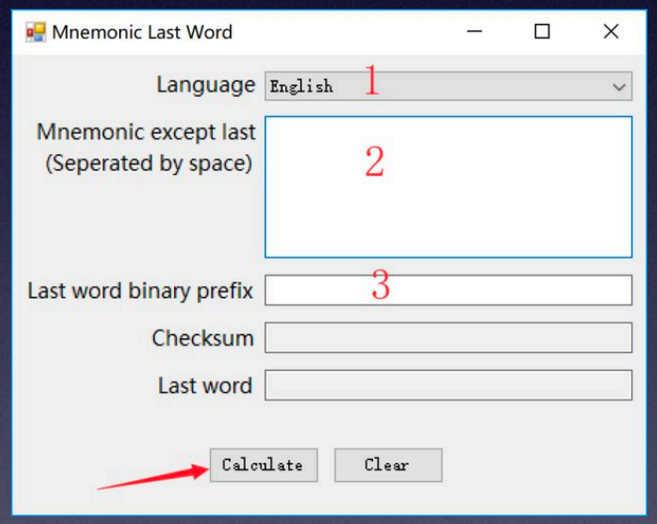
广播交易

- 不要直接认可接口返回的广播结果
- 关键场景一定要自己做验证

广告时间

- Cannon & Crossbow
- webot
- Mnemonic Last Word
- BlueSV
- 梦想咖啡馆





SV开发中的那些坑

aaron67

aaron67@aaron67.cc

https://aaron67.cc/tags/bitcoin