BITCOIN TRANSACTION MALLEABILITY THEORY IN PRACTICE

Daniel Chechik
Security Researcher
Twitter: @DanielChechik

Ben Hayak
Security Researcher
Twitter: @BenHayak
Agenda

• What is Bitcoin

• Bitcoin Transactions

• Transaction Malleability Vulnerability

• What Happened in MT.Gox

• Live Demo
WHAT IS BITCOIN?
Currently ~ $600

CoinDesk BPI: $90.50

www.coindesk.com
What is Bitcoin?

• Bitcoin is a payment system introduced as open-source software in 2009 by a developer known as Satoshi Nakamoto

• P2P network – Trust is a result of data transparency

• Decentralization – No institution is controlling your money/coins.

• Anonymous Virtual currency.
What is a Block?

• Container of Transactions

• Chained to all other valid blocks and shared among all peers
Block Chain

- The network data history

- Block
  - Transactions

- PreviousBlockHash

- Block
  - Transactions

- PreviousBlockHash

- Block
  - Transactions

- PreviousBlockHash
What is a Block?

- Bitcoin Blocks – structure

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magic No</td>
<td>Value Always 0xD9B4BEF9</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Blocksize</td>
<td>Number of bytes following up to end of block</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Blockheader</td>
<td>Consists of 6 items</td>
<td>80 bytes</td>
</tr>
<tr>
<td>Transaction counter</td>
<td>Positive integer VI = VarInt</td>
<td>1 - 9 bytes</td>
</tr>
<tr>
<td>Transactions</td>
<td>The (non empty) list of transactions</td>
<td>&lt;Transaction counter&gt;-many transactions</td>
</tr>
</tbody>
</table>
What is a Block?

• Bitcoin Blocks – structure

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magic No</td>
<td>Number of bytes following up to end of block</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Blocksize</td>
<td>Number of bytes following up to end of block</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Blockheader</td>
<td>Consists of 6 items</td>
<td>80 bytes</td>
</tr>
<tr>
<td>Transaction counter</td>
<td>Positive integer ( VI = \text{VarInt} )</td>
<td>1 - 9 bytes</td>
</tr>
<tr>
<td>Transactions</td>
<td>The (non empty) list of transactions</td>
<td>(&lt;\text{Transaction counter}&gt;)-many transactions</td>
</tr>
</tbody>
</table>
What is a Block?

- **Bitcoin Blocks – structure**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magic No</td>
<td>Value Always 0xD9B4BEF9</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Blocksize</td>
<td>Number of bytes following up to end of block</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Blockheader</td>
<td>Consists of 6 items</td>
<td>80 bytes</td>
</tr>
<tr>
<td>Transaction counter</td>
<td>Positive integer VI = VarInt</td>
<td>1 - 9 bytes</td>
</tr>
<tr>
<td>Transactions</td>
<td>The (non empty) list of transactions</td>
<td>&lt;Transaction counter&gt;-many transactions</td>
</tr>
</tbody>
</table>
## Block Header Structure

<table>
<thead>
<tr>
<th>Field</th>
<th>Purpose</th>
<th>Updated when...</th>
<th>Size (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Block version number</td>
<td>You upgrade the software and it specifies a new version</td>
<td>4</td>
</tr>
<tr>
<td>hashPrevBlock</td>
<td>256-bit hash of the previous</td>
<td>A new block comes in</td>
<td>32</td>
</tr>
<tr>
<td>hashMerkleRoot</td>
<td>256-bit hash based on all of the transactions in the block</td>
<td>A transaction is accepted</td>
<td>32</td>
</tr>
<tr>
<td>Time</td>
<td>Current timestamp as seconds since 1970-01-01T00:00 UTC</td>
<td>Every few seconds</td>
<td>4</td>
</tr>
<tr>
<td>Bits</td>
<td>Current target in compact format</td>
<td>The difficulty is adjusted</td>
<td>4</td>
</tr>
<tr>
<td>Nonce</td>
<td>32-bit number (starts at 0)</td>
<td>A hash is tried increments</td>
<td>4</td>
</tr>
</tbody>
</table>
What is Mining?

1. Collect all Pending Transactions to memory
2. Build a theoretic Block – with the transactions
3. Use computing power to “Solve” your Blockhash
4. Broadcast the block to the network

Diagram:
- Transactions (Tx) flow into Memory
- Memory produces a Block
- Block is broadcast to the network
LET’S SIMULATE MINING RIGHT NOW!
Additional Mining Goals

- Keep a steady network
- Record all coin data
What is Bitcoin - Summary

• Block – container of transactions

• Block chain - record of all coin data from the beginning

• Block “Solving” – a process used to keep the network steady and to generate blocks.
TRANSACTIONS
Transactions

100 BTC
Alice → Bob

Broadcasted to network

Confirmed (Block Solved)

Collected by miners
Transactions

100 MYC
Alice ➔ Bob

Pay To: YYuvP3K461RaKNaCT3X56MaFmkpWKWKHRH ➔ Bob’s Wallet

Label: Enter a label for this address to add it to your address book
Amount: 100 MYC
Transactions

100 MYC
Alice -> Bob

Broadcasted to network

100 MYC

Status: 0/unconfirmed, broadcast through 1 node
Transactions

100 MYC

Broadcasted to network Alice à  Bob

Transactions

100 MYC

Collected by miners

NTY-SL-U0F1G3:CPUMINER dchechik$ ./minerd --algo scrypt --url http://127.0.0.1:30000 --userpass mycoin:12345 --threads 4
[2014-06-22 13:38:40] thread 0: 4184 hashes, 6.78 khash/s
[2014-06-22 13:38:45] thread 0: 33924 hashes, 6.70 khash/s
[2014-06-22 13:38:53] accepted: 1/1 (100.00%), 26.06 khash/s (yay!!!)
Transactions

- Transactions confirmed (Block Solved)
- Sent from Alice to Bob
- 100 MYC
- Status: 6 confirmations, broadcast through 1 node
- Transaction ID: 35b56afcd8df60bd7707efb12908265c9abae30311ccd5e02841eb474ef20c75
- Date: 6/22/2014 13:36
- Amount: 50.10 MYC
Transactions

Jeff to Daniel

Daniel to Ben
Transactions

Transactions are built from two main components:

**Inputs**
- Source of coins (Ref to Txout in block chain)

**Outputs**
- Redeemer’s Bitcoin address
- Amount
Transactions

• Prove you have the coins (by including a reference)

• Include a Public Key of the recipient

• Sign the transaction
**P2P Lottery**

<table>
<thead>
<tr>
<th>MessageID (sha256)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
| From: Lottery  
Prize: You won a Car! |
| ... | Length |
| To: “Ben” |
P2P Lottery

MessageID (sha256)

---

From: Lottery
Prize: You won a Vacation

To: “Ben”
P2P Lottery

MessageID (sha256)

... Length

From: Lottery
Prize: You won a Car!

Signature (DER)

... Length

To: “Ben”
Standard Transaction

TxId (sha256*2)

<table>
<thead>
<tr>
<th>...</th>
<th>Length (1 Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Source of coins</td>
</tr>
<tr>
<td></td>
<td>sender</td>
</tr>
</tbody>
</table>

| ... | Amount | Length |

ScriptPubkey
### PushData Opcode

<table>
<thead>
<tr>
<th>TxId (sha256*2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

- **ScriptSig**

<table>
<thead>
<tr>
<th>...</th>
<th>Amount</th>
<th>Length</th>
</tr>
</thead>
</table>

- **ScriptPubkey**
Standard Transaction

TxId (sha256*2)

... 0x32

ScriptSig

... 40BTC 0x19

ScriptPubkey
Mutated Transaction

- **TxId (sha256*2)**
- **ScriptSig**
  - **PUSHDATA2**
  - `0x32`
  - `0x32`
  - `0x4D`
  - `0x00`
- **ScriptPubkey**
  - `0x32 == 0x0032`
  - `40BTC`
  - `0x19`
Standard Vs Mutated

**Standard**

- **TxId (sha256*2)**
- Length (1 Byte)
- ScriptSig
- Amount
- Length
- ScriptPubkey

- **TxId** = c6cfe6e4f129a34671d10c1bbe158eff05197d388727e331951b0ec2637c194e

**Mutated**

- **TxId (sha256*2)**
- **PUSHDATA2** Length 0x00
- ScriptSig
- Amount
- Length
- ScriptPubkey

- **Mutated TxId** = dc34efd49ed738bf4500db367292164166989cb15773026e9e185b78292bbbc89
Transaction Malleability

- Two different transactions
  - Same amount of coins
  - Same destination and source
- Mutated wins and gets in a Block

RACE!
Rejected Transactions

- Invalid transaction data
- Already spent out-point
- Identical transactions
- Invalid signature
WHAT HAPPENED IN MT.GOX?
Dear MtGox Customers and Bitcoiners,

As you are aware, the MtGox team has been working hard to address an issue with the way that bitcoin withdrawals are processed. By "bitcoin withdrawal" we are referring to transactions from a MtGox bitcoin wallet to an external bitcoin address. Bitcoin transactions to any MtGox bitcoin address, and currency withdrawals (Yen, Euro, etc) are not affected by this issue.

The problem we have identified is not limited to MtGox, and affects all transactions where Bitcoins are being sent to a third party. We believe that the changes required for addressing this issue will be positive over the long term for the whole community. As a result we took the necessary action of suspending bitcoin withdrawals until this technical issue has been resolved.

Addressing Transaction Malleability
MtGox has detected unusual activity on its Bitcoin wallets and performed investigations during the past weeks. This confirmed the presence of transactions which need to be examined more closely.
Mt.Gox

30BTC -> Attacker’s Wallet

B330.......5088

P2P Bitcoin

Attacker

Attacker’s Wallet
30BTC -> Attacker’s Wallet

Mt.Gox

Attacker

ScriptSig

ScriptPubkey

B330......5088

0x19

0x30

30BTC

Attacker’s Wallet
Mt.Gox

30BTC -> Attacker’s Wallet

B330........5088

P2P Bitcoin

30BTC

0x19

ScriptSig

0x30

B330........5088

ScriptPubkey

Attacker

Attacker’s Wallet

black hat USA 2014
Mt.Gox

30BTC -> Attacker’s Wallet

P2P Bitcoin

Attacker’s Wallet

B330.........5088

Mutated Transaction

Valid Signature

C3a8........03f8
30BTC -> Attacker’s Wallet

Mt.Gox

Mutated Transaction

Valid Signature

Attacker

Attacker’s Wallet

black hat USA 2014
Mt.Gox

30BTC -> Attacker’s Wallet

P2P Bitcoin

B330.......5088

Attacker

Attacker’s Wallet

C3a8.......03f8

30BTC -> Attacker’s Wallet
30BTC -> Attacker’s Wallet

Mt.Gox

Transaction (B330........5088) Failed?!?

Unconfirmed

30BTC -> Attacker’s Wallet

Attacker’s Wallet
A transaction of 30BTC failed on Mt.Gox. The transaction was intended to go to Attacker's Wallet but was marked as unconfirmed. The attacker tried to generate another transaction using a different wallet address. However, the transaction failed again. The attacker might need to generate another transaction to resolve the issue.
30BTC -> Attacker's Wallet

B330........5088
Unconfirmed

Transaction (B330........5088)
Failed?!?

Generate Another Transaction!

Mt.Gox

P2P Bitcoin

A$acker

A$acker's Wallet

Mt.Gox

P2P Bitcoin

A$acker

A$acker's Wallet

Attacker's Wallet
30BTC -> Attacker’s Wallet

B330······5088
Unconfirmed

Transaction (B330······5088)
Failed?!?

Generate Another Transaction!

I’m off to the Caribbean
MALLEABILITY FIX
bool IsStandardTx(const CTransaction& tx, string& reason) {
    [...snip...]
    if (!txin.scriptSig.HasCanonicalPushes()) {
        reason = "scriptsig-non-canonical-push";
        return false;
    }
    [...snip...]
    return true;
}
Transaction Malleability Fix

```cpp
bool CScript::HasCanonicalPushes() const
{
    const_iterator pc = begin();
    while (pc < end())
    {
        opcode_type opcode;
        std::vector<unsigned char> data;
        if (!GetOp(pc, opcode, data))
            return false;
        if (opcode > OP_16)
            continue;
        if (opcode < OP_PUSHDATA1 && opcode > OP_0 && (data.size() == 1 && data[0] <= 16))
            // Could have used an OP_n code, rather than a 1-byte push.
            return false;
        if (opcode == OP_PUSHDATA1 && data.size() < OP_PUSHDATA1)
            // Could have used a normal n-byte push, rather than OP_PUSHDATA1.
            return false;
        if (opcode == OP_PUSHDATA2 && data.size() <= 0xFF)
            // Could have used an OP_PUSHDATA1.
            return false;
        if (opcode == OP_PUSHDATA4 && data.size() <= 0xFFFF)
            // Could have used an OP_PUSHDATA2.
            return false;
    }
    return true;
}
```
Thank You!

Daniel Chechik – Daniel.chechik@gmail.com
Ben Hayak - Ben.hayak@gmail.com

BTC: 12qPtFhw9UPL8HvfSsSjvqxeFXp4hRiWym
References

Github - https://github.com/sipa/bitcoin/commit/87fe71e1fc810ee120a10063fdd26c3245686d54
Spiderlabs – http://www.spiderlabs.com
Bitcoin Wiki - https://en.bitcoin.it/wiki