



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

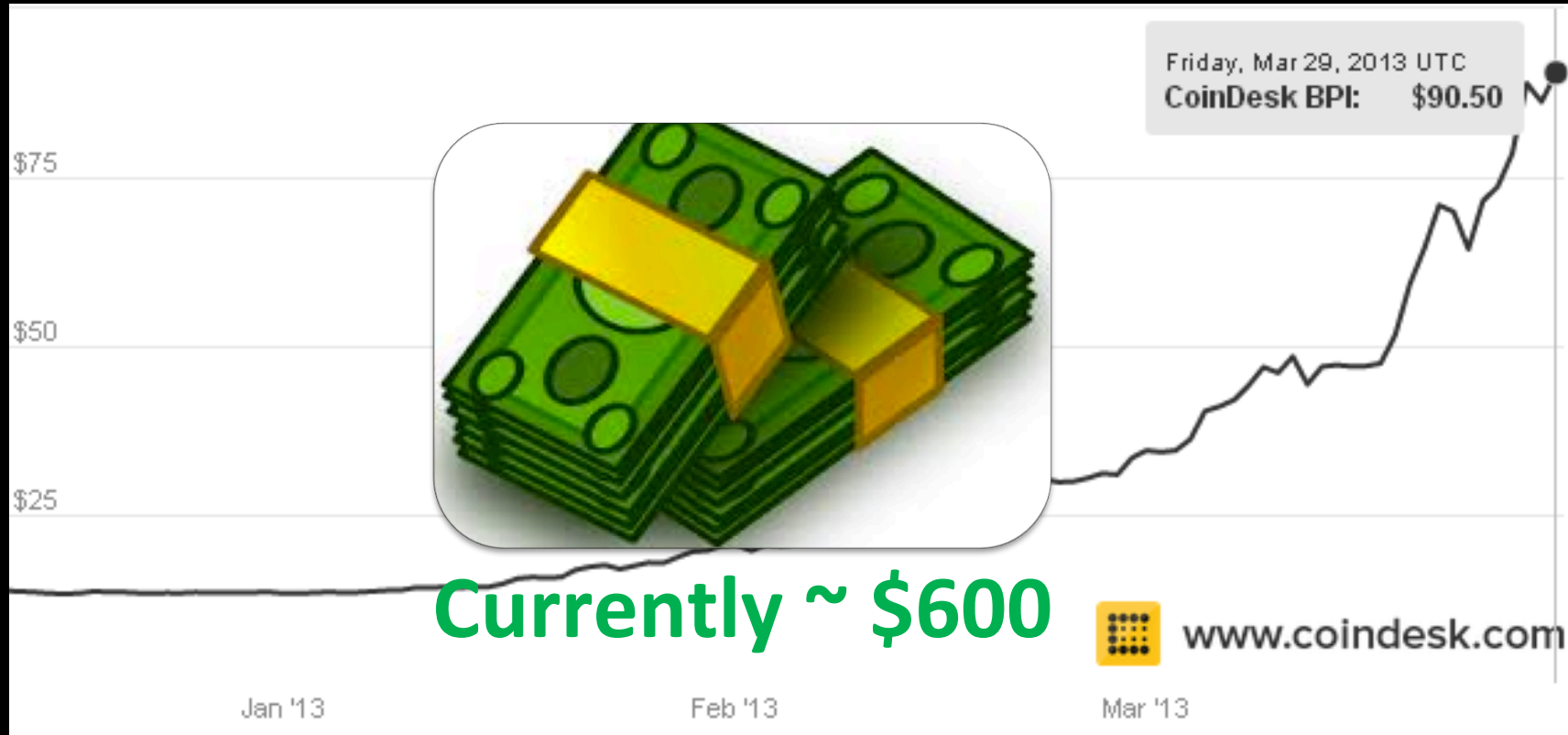
The logo for Black Hat USA 2014. It features a white silhouette of a person wearing a black hat inside a white circle. Below this icon, the text "black hat" is written in a bold, lowercase, sans-serif font. Underneath "black hat", the text "USA 2014" is written in a smaller, uppercase, sans-serif font. The entire logo is set against a background of a glowing blue and purple globe of the Earth.

black hat
USA 2014

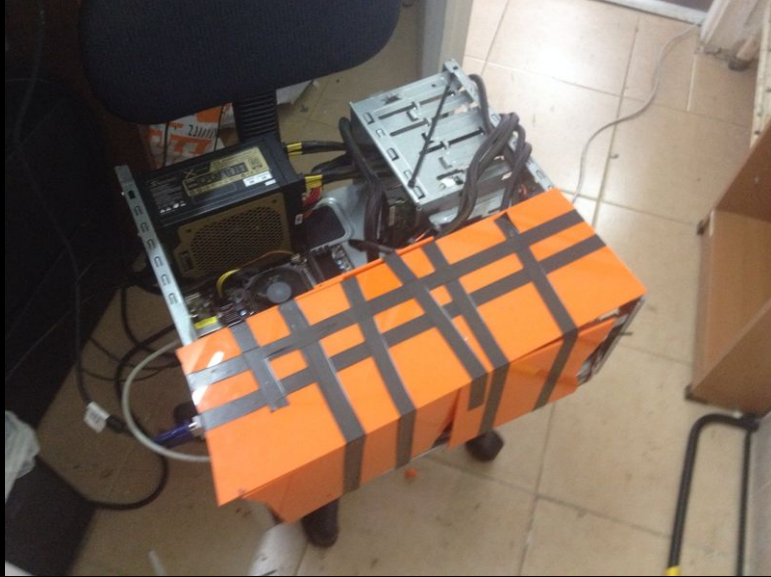
WHAT IS BITCOIN?













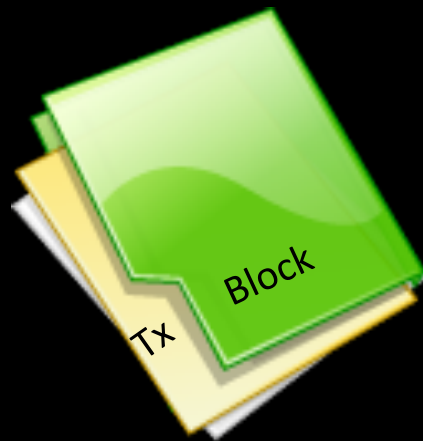
What is Bitcoin?



- 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.

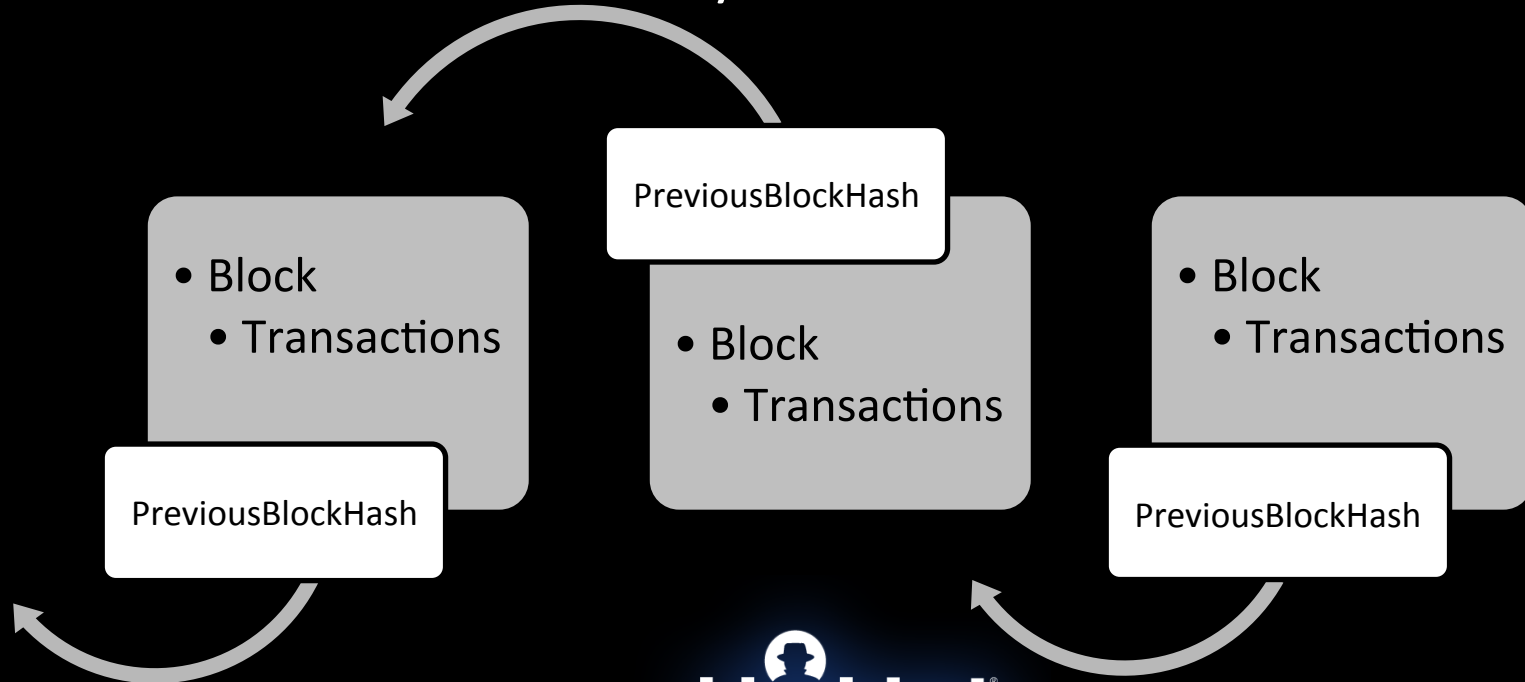
What is a Block?

- Container of Transactions
- Chained to all other valid blocks and shared among all peers



Block Chain

- The network data history




What is a Block?

- Bitcoin Blocks – structure

Field	Description	Size
Magic No	Value Always 0xD9B4BEF9	4 bytes
Blocksize	Number of bytes following up to end of block	4 bytes
Blockheader	Consists of 6 items	80bytes
Transaction counter	Positive integer VI = VarInt	1 - 9 bytes
Transactions	The (non empty) list of transactions	<Transaction counter>-many transactions

What is a Block?

- Bitcoin Blocks – structure

Field		Size
Magic No		4 bytes
Blocksize	Number of bytes following up to end of block	4 bytes
Blockheader	Consists of 6 items	80bytes
Transaction counter	Positive integer VI = VarInt	1 - 9 bytes
Transactions	The (non empty) list of transactions	<Transaction counter>-many transactions

What is a Block?

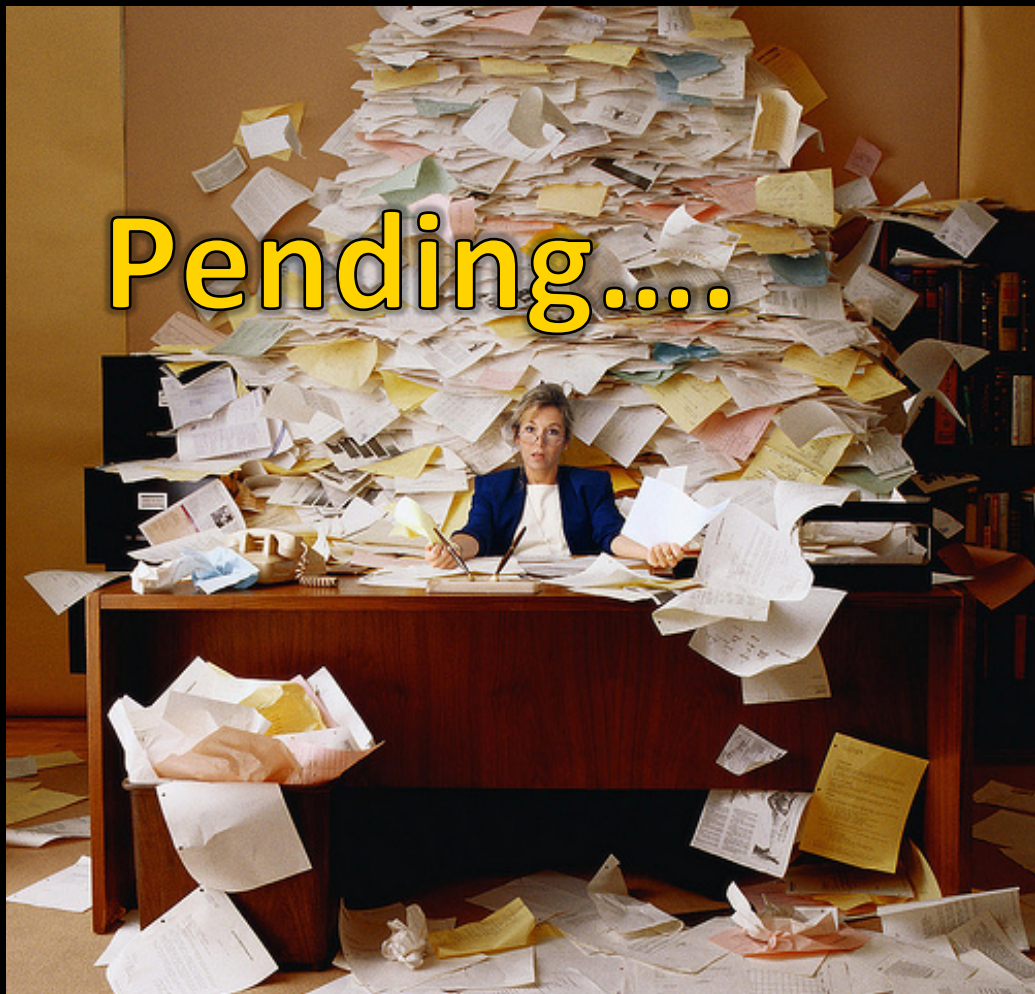
- Bitcoin Blocks – structure

Field	Description	Size
Magic No	Value Always 0xD9B4BEF9	4 bytes
Blocksize	Number of bytes following up to end of block	4 bytes
Blockheader	Consists of 6 items	80bytes
Transaction counter	Positive integer VI = VarInt	1 - 9 bytes
Transactions	The (non empty) list of transactions	<Transaction counter>-many transactions

Block Header Structure

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

Pending....



What is Mining?

Collect all Pending Transactions to memory



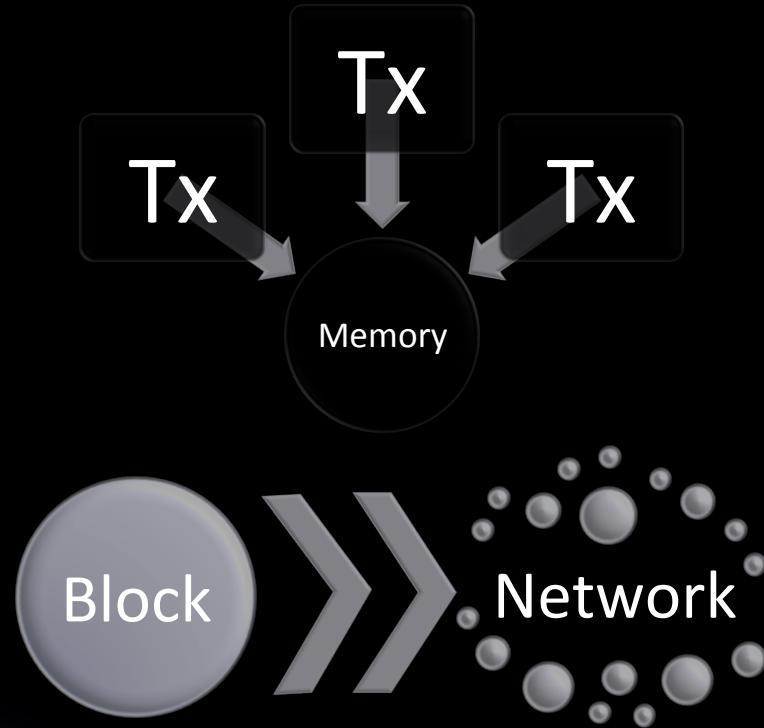
Build a theoretic Block – with the transactions



Use computing power to “Solve” **your** Blockhash



Broadcast the block to the network





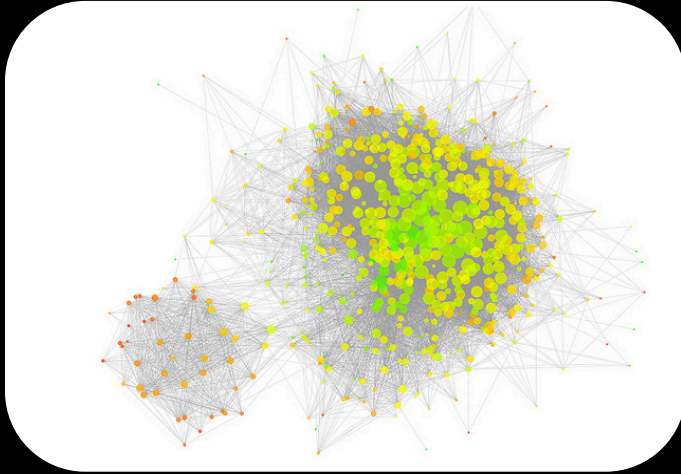
**LET'S SIMULATE
MINING RIGHT NOW!**



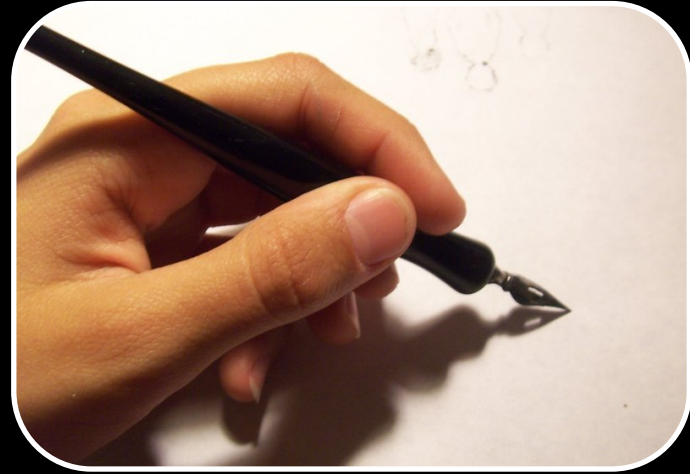
black hat[®]
USA 2014

0x02000

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.

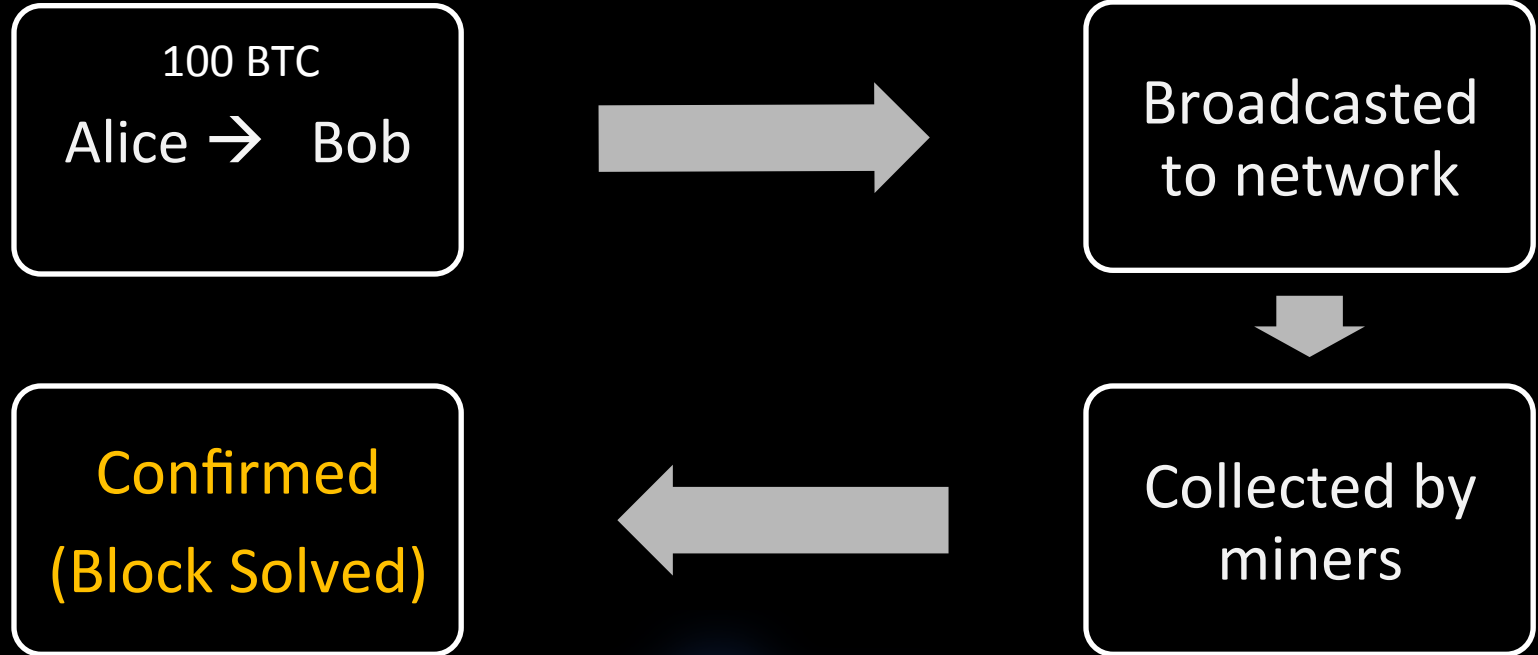


black hat[®]
USA 2014

TRANSACTIONS

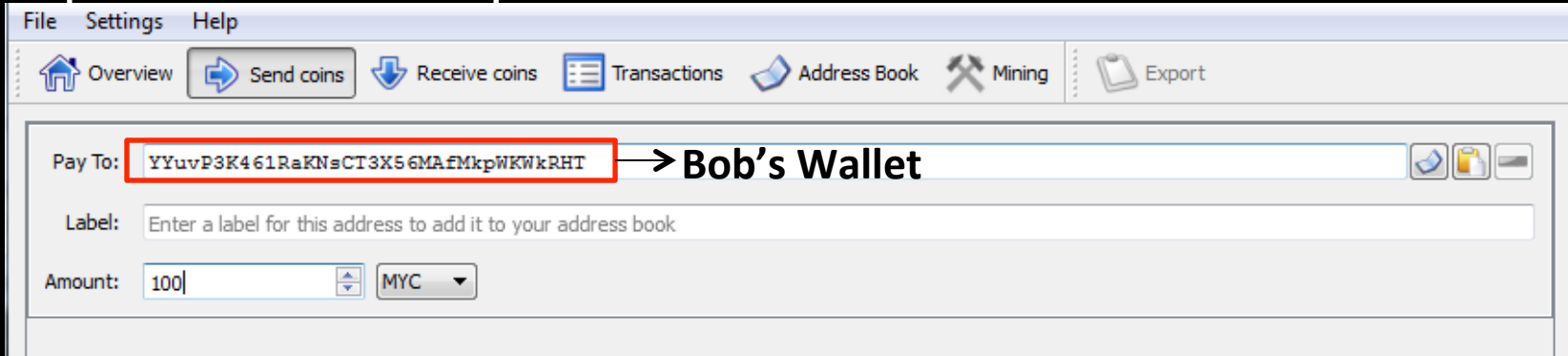


Transactions



Transactions

100 MYC
Alice → Bob



The screenshot shows a wallet application window with a menu bar (File, Settings, Help) and a toolbar with icons for Overview, Send coins, Receive coins, Transactions, Address Book, Mining, and Export. The main area contains a transaction form with the following fields:

- Pay To:** A text field containing the address `YYuvP3K461RaKNsCT3X56MAfMkpWKWkRHT`, which is highlighted with a red box. To its right is a dropdown menu currently showing `→ Bob's Wallet`.
- Label:** A text input field with the placeholder text "Enter a label for this address to add it to your address book".
- Amount:** A text input field containing the value `100`, followed by a spin button and a dropdown menu set to `MYC`.

Transactions

100 MYC

Broadcasted

The screenshot shows a Bitcoin wallet interface with a menu bar (File, Settings, Help) and a toolbar (Overview, Send coins, Receive coins, Transactions, Address Book, Mining, Export). Below the toolbar is a search bar and a table of transactions. The table has columns for Date, Type, Address, and Amount. A transaction is highlighted with a red box around the question mark icon in the Date column. A 'Transaction details' dialog box is open over the transaction, showing the following information:

Date	Type	Address	Amount
6/22/2014 13:36	Sent to	(YYuvP3K461RaKNsCT3X56MAfMkpWKWkRHT)	-100.00
6/22/2014 13:36			[25.00]
6/22/2014 13:36			[50.10]
6/22/2014 13:36			20.00
6/22/2014 13:36			10.00
6/22/2014 13:36			[50.00]

Transaction details

Status: 0/unconfirmed, broadcast through 1 node

Date: 6/22/2014 13:36

To: YYuvP3K461RaKNsCT3X56MAfMkpWKWkRHT

Debit: -100.00 MYC

Net amount: -100.00 MYC

Transaction ID: 35b56afcd8df60bd7707efb12908265c9abae30311ccd5e02841eb474ef20c75

Transactions

100 MYC

Broadcasted

```
NTY-SL-UUF1G3:CPUMINER dchechik$ ./minerd --algo scrypt --url http://127.0.0.1:38000 --userpass mycoin:12345 --threads 4
[2014-06-22 13:38:39] 4 miner threads started, using 'scrypt' algorithm.
[2014-06-22 13:38:40] thread 3: 4104 hashes, 7.13 khash/s
[2014-06-22 13:38:40] thread 0: 4104 hashes, 6.78 khash/s
[2014-06-22 13:38:40] thread 2: 4104 hashes, 6.75 khash/s
[2014-06-22 13:38:40] thread 1: 4104 hashes, 6.75 khash/s
[2014-06-22 13:38:45] thread 1: 33744 hashes, 6.82 khash/s
[2014-06-22 13:38:45] thread 2: 33756 hashes, 6.69 khash/s
[2014-06-22 13:38:45] thread 0: 33924 hashes, 6.70 khash/s
[2014-06-22 13:38:45] thread 3: 35664 hashes, 6.74 khash/s
[2014-06-22 13:38:50] thread 2: 33468 hashes, 6.79 khash/s
[2014-06-22 13:38:50] thread 0: 33504 hashes, 6.79 khash/s
[2014-06-22 13:38:50] thread 1: 34092 hashes, 6.66 khash/s
[2014-06-22 13:38:50] thread 3: 33720 hashes, 6.76 khash/s
[2014-06-22 13:38:51] thread 0: 8292 hashes, 6.76 khash/s
[2014-06-22 13:38:53] accepted: 1/1 (100.00%), 26.96 khash/s (yay!!!)
[2014-06-22 13:38:55] thread 2: 33936 hashes, 6.98 khash/s
[2014-06-22 13:38:55] thread 1: 33300 hashes, 6.88 khash/s
[2014-06-22 13:38:55] thread 3: 33804 hashes, 6.83 khash/s
```

Transactions

File Settings Help

Overview Send coins Receive coins Transactions Address Book Mining Export

All All Enter address or label to search Min amount

Date	Type	Address	Amount
6/19/2014 15:22	Received with	(YSanSyuaUeuFcdxjaQHxUFKffwLf9UsMjD)	[25.00]
6/22/2014 13:36	Sent to	(YYuvP3K461RaKNsCT3X56MAfMkpWKWkRHT)	-100.00
			[50.10]
			20.00
			10.00
			[50.00]
			[50.00]

Transaction details

Status: 6 confirmations, broadcast through 1 node
Date: 6/22/2014 13:36
To: YYuvP3K461RaKNsCT3X56MAfMkpWKWkRHT
Debit: -100.00 MYC
Net amount: -100.00 MYC
Transaction ID: 35b56afcd8df60bd7707efb12908265c9abae30311ccd5e02841eb474ef20c75

Transactions



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





TRANSACTION MALLEABILITY

P2P Lottery

MessageID (sha256)

...	Length
From: Lottery Prize: You won a Car!	
...	Length
To: "Ben"	

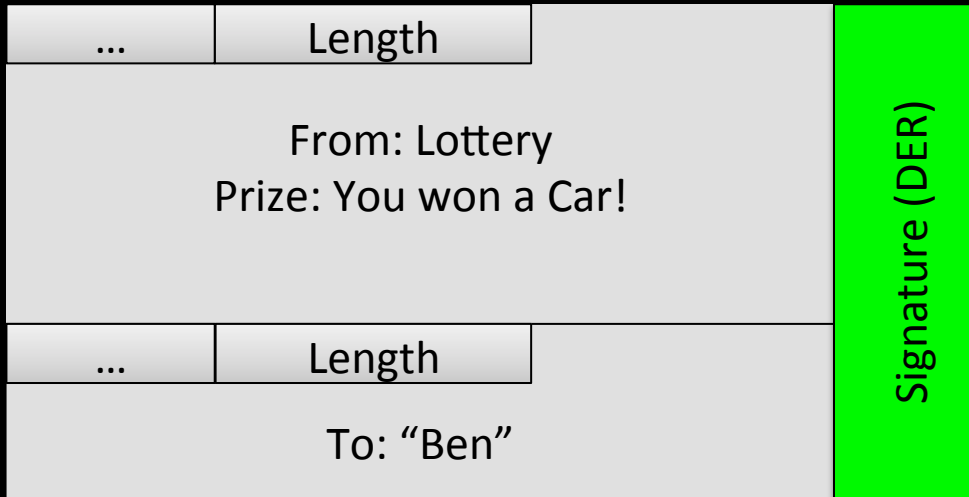
P2P Lottery

MessageID (sha256)

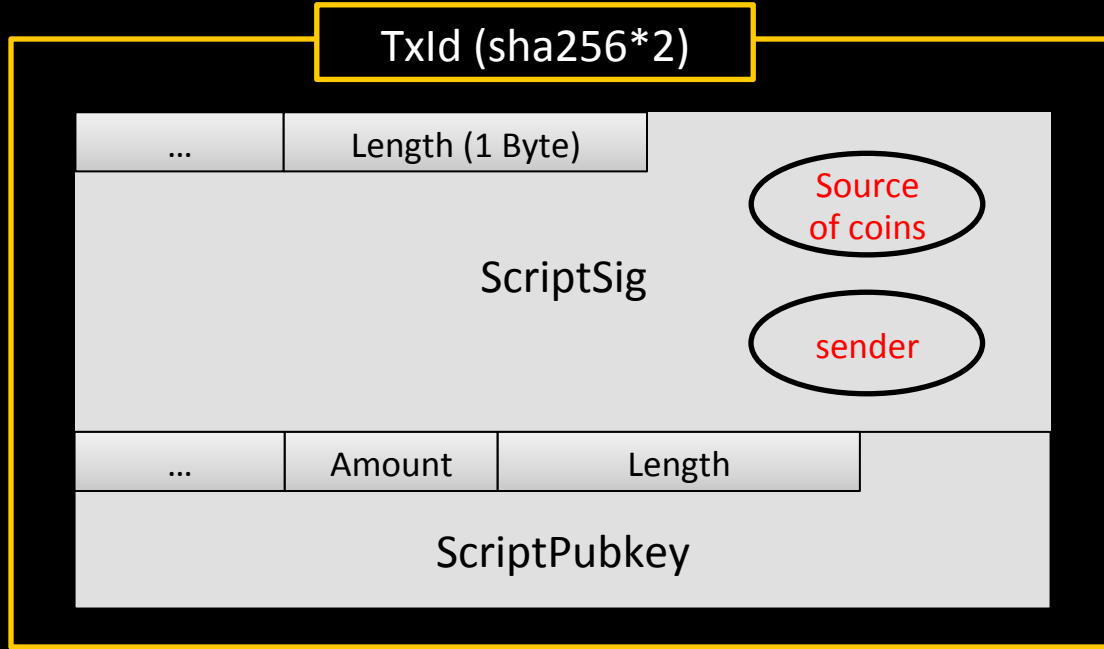
...	Length
From: Lottery Prize: You won a Vacation	
...	Length
To: "Ben"	

P2P Lottery

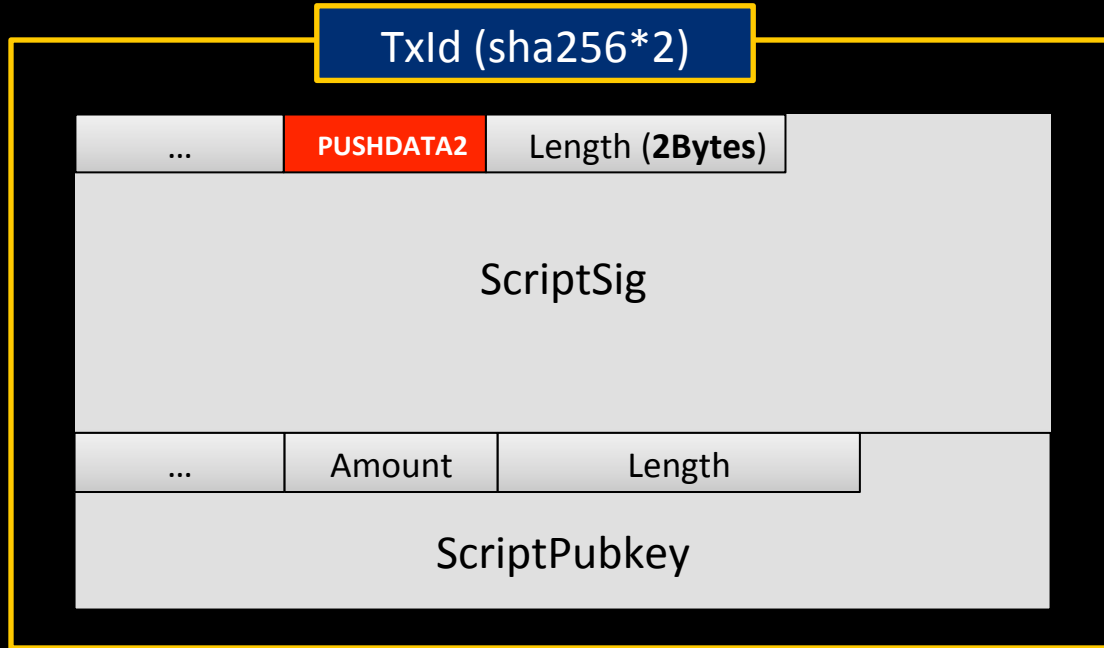
MessageID (sha256)



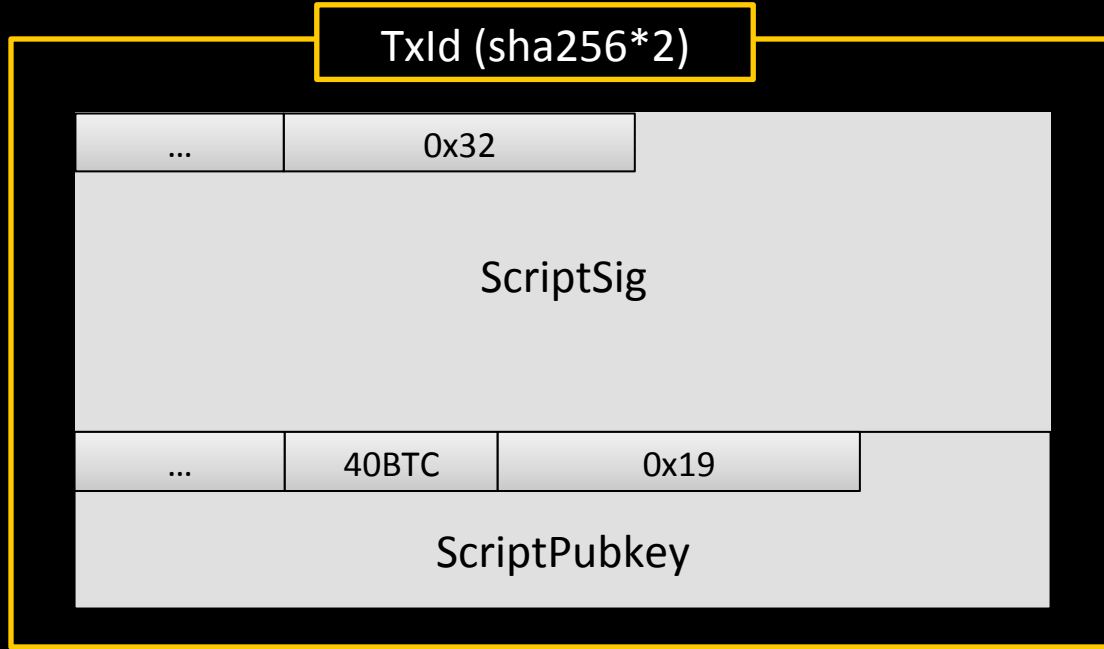
Standard Transaction



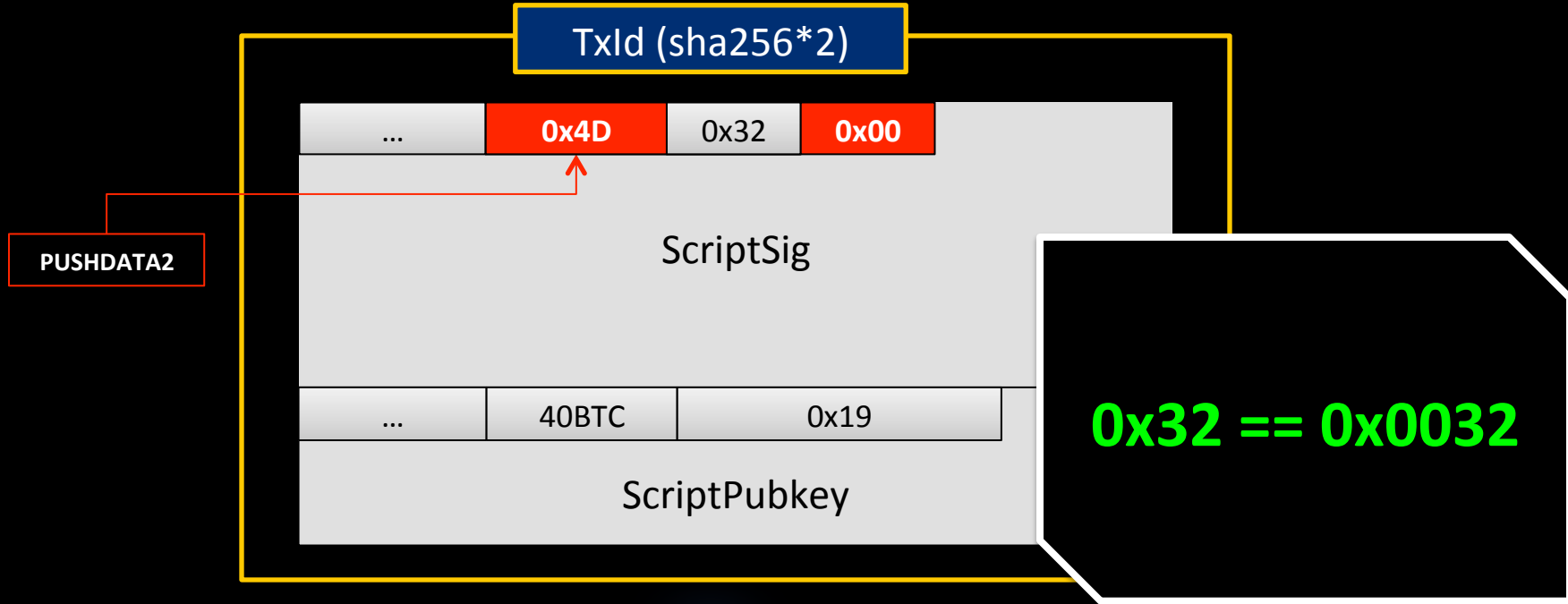
PushData Opcode



Standard Transaction



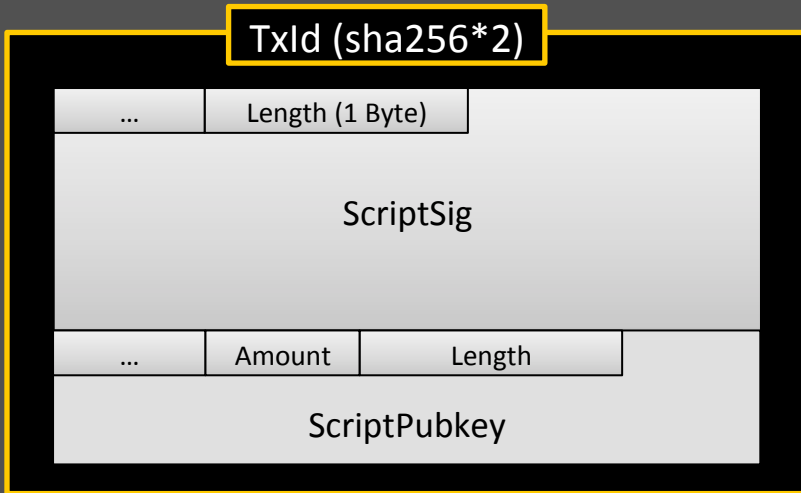
Mutated Transaction



Standard

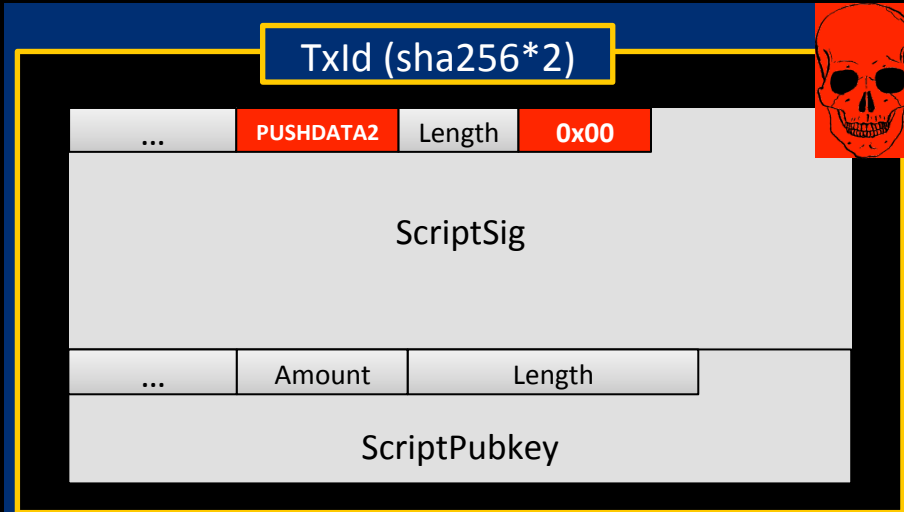
Vs

Mutated



TxId =

c6cfe6e4f129a34671d10c1bbe158eff05197d38872
7e331951b0ec2637c194e

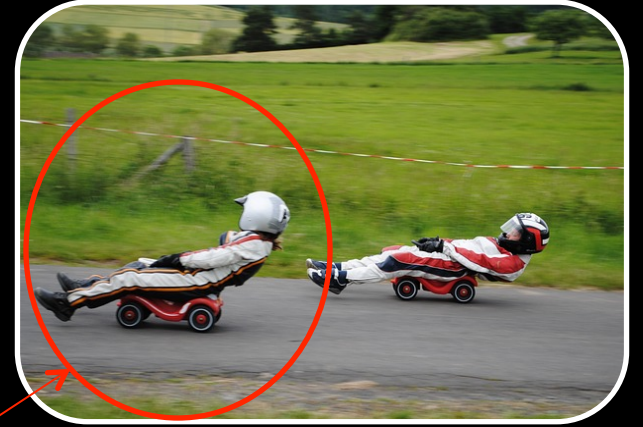


Mutated TxId =

dc34efd49ed738bf4500db367292164166989cb1577302
6e9e185b78292bbc89

Transaction Malleability

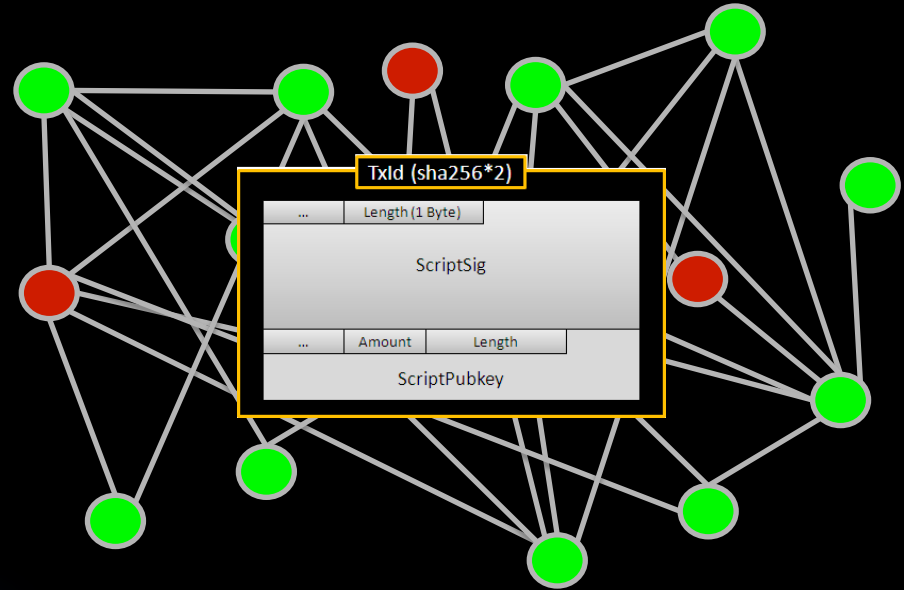
- Two different transactions
 - Same amount of coins
 - Same destination and source
- Mutated wins and gets in a Block



RACE!

Rejected Transactions

- Invalid transaction data
- Already spent out-point
- Identical transactions
- Invalid signature





WHAT HAPPENED IN MT.GOX?

Mt.Gox Announcement

[HOME](#)[TRADE](#)[MERCHANT TOOLS](#)[SECURITY CENTER](#)[SETTINGS](#)[FAQ](#)[NEWS](#)

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

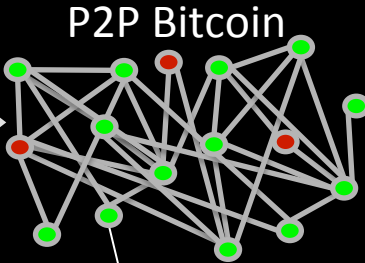
MtGox has detected unusual activity on its Bitcoin wallets and performed investigations during the past weeks. This confirmed the presence of transactions which need to be examined more closely.



Mt.Gox

30BTC -> Attacker's Wallet

B330.....5088



Attacker



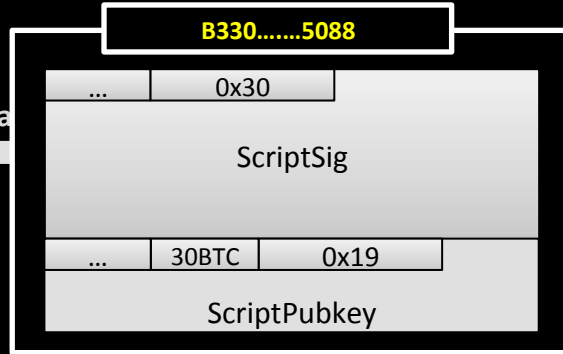
Attacker's Wallet



Mt.Gox

30BTC -> Attacker's Wa

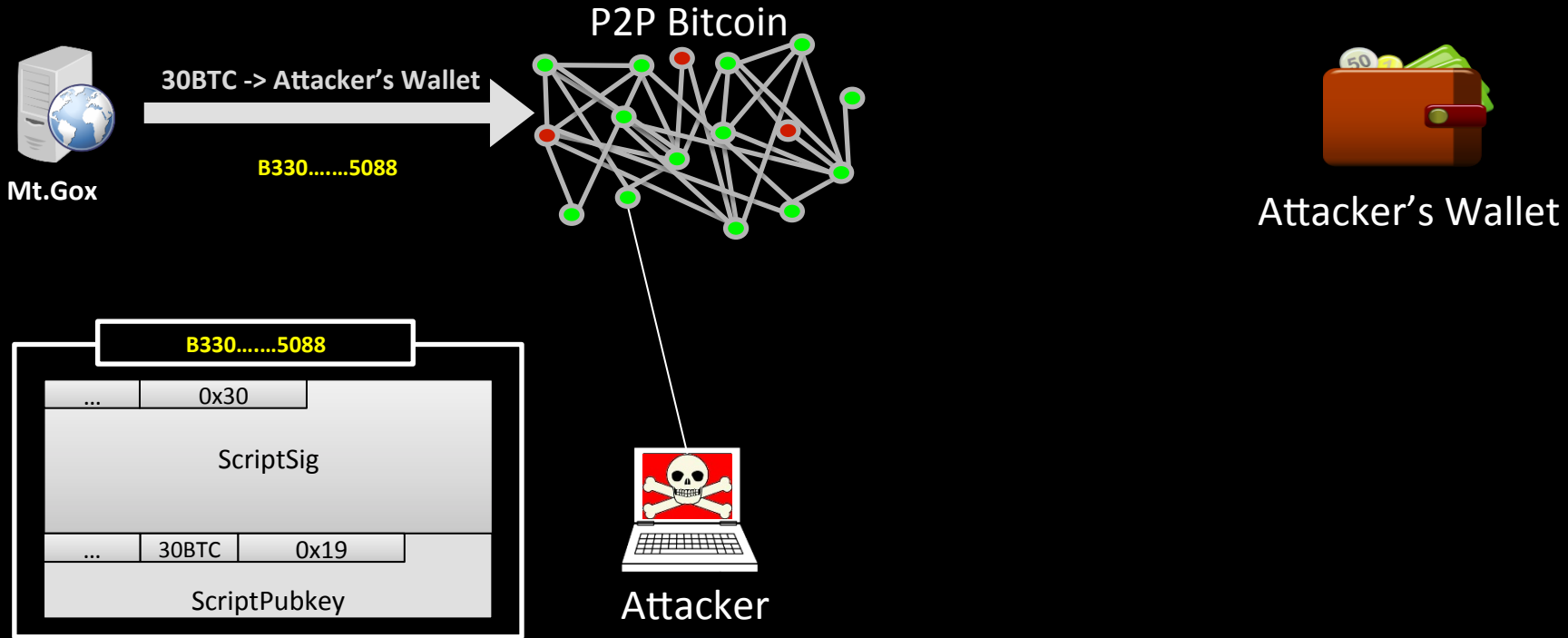
B330.....5088



Attacker's Wallet



Attacker

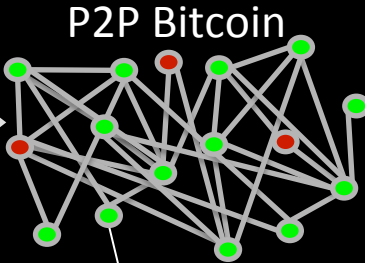




Mt.Gox

30BTC -> Attacker's Wallet

B330.....5088



Attacker's Wallet

C3a8.....03f8

0x30

Mutated Transaction



Valid Signature



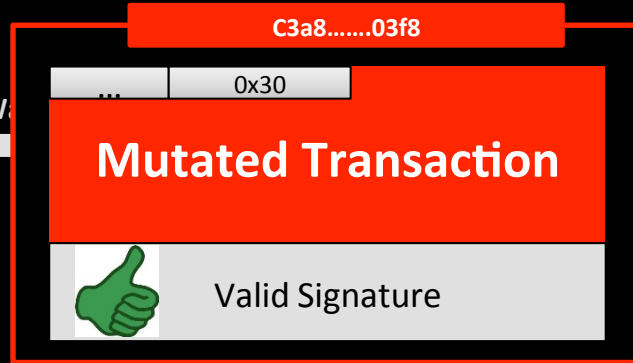
Attacker



Mt.Gox

30BTC -> Attacker's W

B330.....5088



Attacker's Wallet

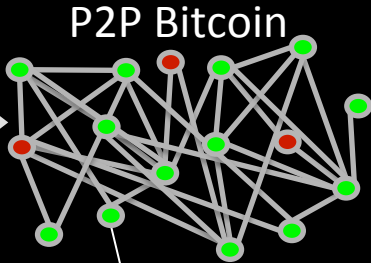


Attacker



30BTC -> Attacker's Wallet

B330.....5088



30BTC -> Attacker's Wallet

C3a8.....03f8



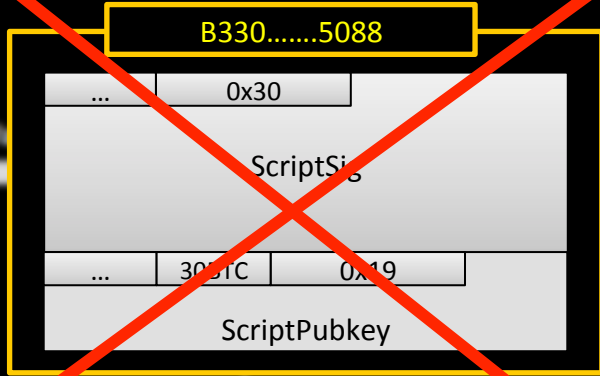
Attacker's Wallet



Attacker

Unconfirmed Tx

B330.....5088



30BTC -> Attacker's Wallet

B330.....5088

-> Attacker's Wallet

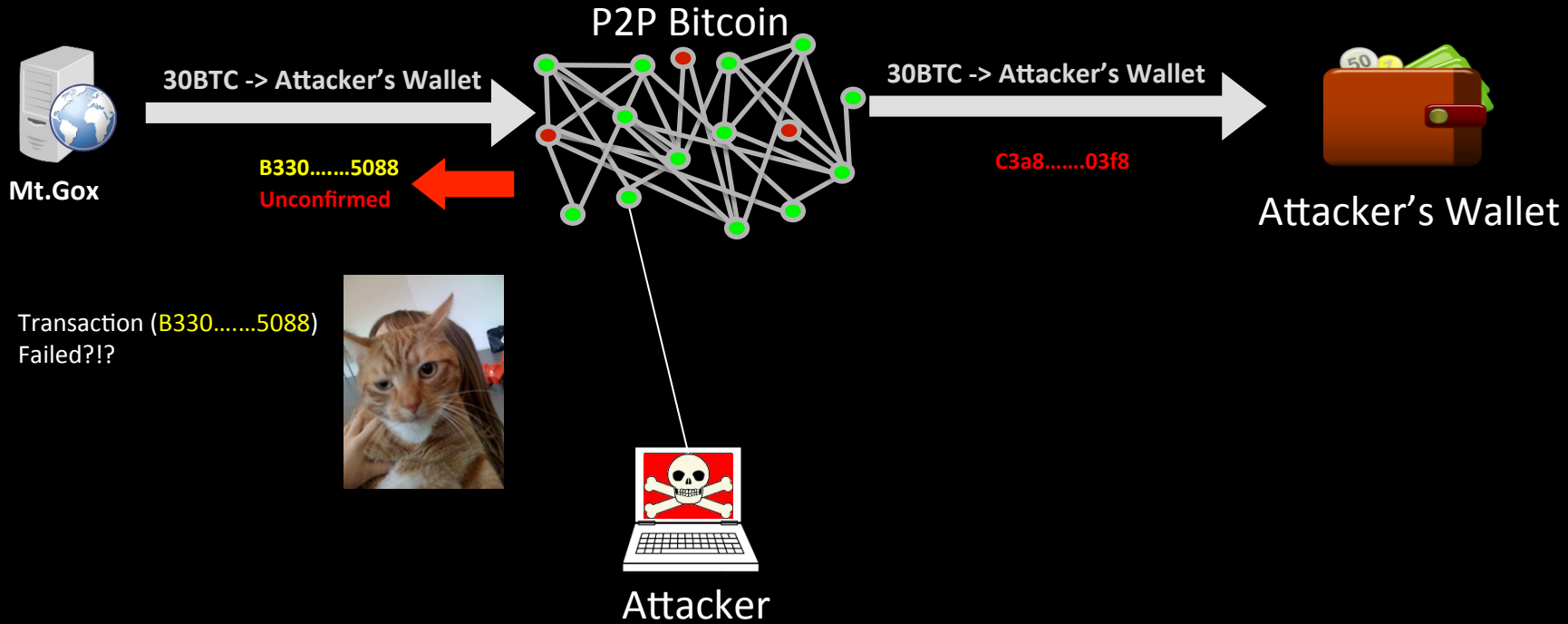
C348.....0200



Attacker's Wallet



Attacker



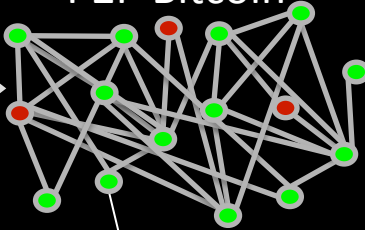


Mt.Gox

30BTC -> Attacker's Wallet

B330.....5088
Unconfirmed

P2P Bitcoin



30BTC -> Attacker's Wallet

C3a8.....03f8



Attacker's Wallet



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

30BTC -> Attacker's Wallet
A213.....1234

Generate Another
Transaction!



Attacker

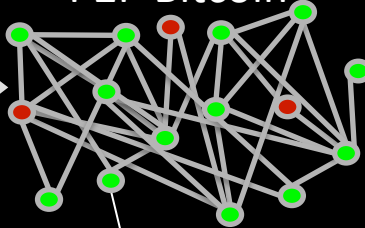


Mt.Gox

30BTC -> Attacker's Wallet

B330.....5088
Unconfirmed

P2P Bitcoin



30BTC -> Attacker's Wallet

C3a8.....03f8



Attacker's Wallet



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

30BTC -> Attacker's Wallet
A213.....1234

Generate Another
Transaction!



Attacker

P2P Bitcoin



Mt.Gox

30BTC -> Attacker's Wallet

B330.....5088
Unconfirmed



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

30BTC -> Attacker's Wallet
A213.....1234

Generate Another
Transaction!





black hat[®]
USA 2014

DEMO



MALLEABILITY FIX

Transaction 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

```
1885 bool CScript::HasCanonicalPushes() const
1886 {
1887     const_iterator pc = begin();
1888     while (pc < end())
1889     {
1890         opcode_t opcode;
1891         std::vector<unsigned char> data;
1892         if (!GetOp(pc, opcode, data))
1893             return false;
1894         if (opcode > OP_16)
1895             continue;
1896         if (opcode < OP_PUSHDATA1 && opcode > OP_0 && (data.size() == 1 && data[0] <= 16))
1897             // Could have used an OP_n code, rather than a 1-byte push.
1898             return false;
1899         if (opcode == OP_PUSHDATA1 && data.size() < OP_PUSHDATA1)
1900             // Could have used a normal n-byte push, rather than OP_PUSHDATA1.
1901             return false;
1902         if (opcode == OP_PUSHDATA2 && data.size() <= 0xFF)
1903             // Could have used an OP_PUSHDATA1.
1904             return false;
1905         if (opcode == OP_PUSHDATA4 && data.size() <= 0xFFFF)
1906             // Could have used an OP_PUSHDATA2.
1907             return false;
1908     }
1909     return true;
1910 }
```



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 official document - <https://bitcoin.org/bitcoin.pdf>

Bitcoin Wiki - <https://en.bitcoin.it/wiki>

Bitcoin Transaction Malleability Wiki - https://en.bitcoin.it/wiki/Transaction_Malleability

Ken Shirriff - <http://www.righto.com/2014/02/bitcoin-transaction-malleability.html>