UNIVERSITY OF NICOSIA
ΠΑΝΕΠΙΣΤΗΜΙΟ ΛΕΥΚΩΣΙΑΣ

DFIN-511
Introduction to Digital Currencies

Session 3
Basics of Crypto-currencies, Part I: Cryptography, transactions, and mining

Objectives of Session 3

- Go through major events in the history of Bitcoin
- Understand how Bitcoin and cryptography are related
- Gain a first idea of how Bitcoin transactions work
- Get introduced to Bitcoin mining

We are about to step into the more technical aspects of Bitcoin, as they are crucial in understanding how this system is enabled and how the Bitcoin network operates.

For the non-technical people, familiarization with several new concepts will be required, but we guarantee it will be a very rewarding experience in the long term!

Agenda

1. A brief history of Bitcoin
2. Basics of Bitcoin Cryptography
3. Transactions and the blockchain
4. Mining
5. Conclusions
6. Self-Assessment Exercises and Further Reading

1. A brief history of Bitcoin

A brief history of Bitcoin

2008

- The bitcoin.org domain name is registered
- Satoshi Nakamoto's original Bitcoin paper is published
- The Bitcoin Project is registered at sourceforge.net

2009

- The Genesis block is established on January 3rd at 18:15:05 GMT
- The first Bitcoin client (*bitcoind* v0.1) is released
- The first Bitcoin transaction is performed in block 170, from Satoshi Nakamoto to Hal Finney
- The first difficulty increase occurs on December 30th, from 1.00 to 1.18289953 in block 32256

2010

- The first Bitcoin currency exchange site (Bitcoin Market) is established
- OpenGL GPU hash farm is established by ArtForz and GPU mining begins
- A vulnerability in Bitcoin is exploited to generate 184 billion bitcoins – the bug is quickly fixed
- Pooled mining begins (discussed later in this session)

A brief history of Bitcoin

2011

- Bitcoin reaches parity with the US dollar for the first time (that is, 1 USD = 1 BTC)
- Bitcoin generation difficulty passes 1 million for the first time in June 2011
- The first Bitcoin Conference and World Expo is held in New York City

2012

- The Bitcoin Foundation is established in September 2012
- The First Bitcoin Halving Day is observed on November 28th, with block 210000 having a block reward of 25 BTC. The next scheduled Halving Day will be observed in 2016, when block reward halves to 12.5 BTC in block 420000.

2013

- A hard fork of reference client v0.8.0 occurs; network updates within a few hours
- The total Bitcoin market capitalization exceeds US $1 billion
- University of Nicosia becomes the world's first university to accept bitcoins for tuition
- Bitcoin mining difficulty passes 1 billion in December 2013

2014

- Bitcoin mining difficulty passes 35 billion
- University of Nicosia launches the world's first Master's Degree in Digital Currency

2. Basics of Bitcoin cryptography

Bitcoin and Cryptography

Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem, including:

- A decentralized peer-to-peer network (enabled by **the Bitcoin protocol**)
- A public transaction ledger (**the blockchain**)
- A decentralized mathematical and deterministic currency issuance mechanism **(distributed mining and the "Proof-of-Work" concept)**
- A decentralized transaction verification system **(transaction script)**

(From "Mastering Bitcoin")

The Bitcoin system is based on decentralized trust, thus it heavily relies on cryptographic technologies, such as:

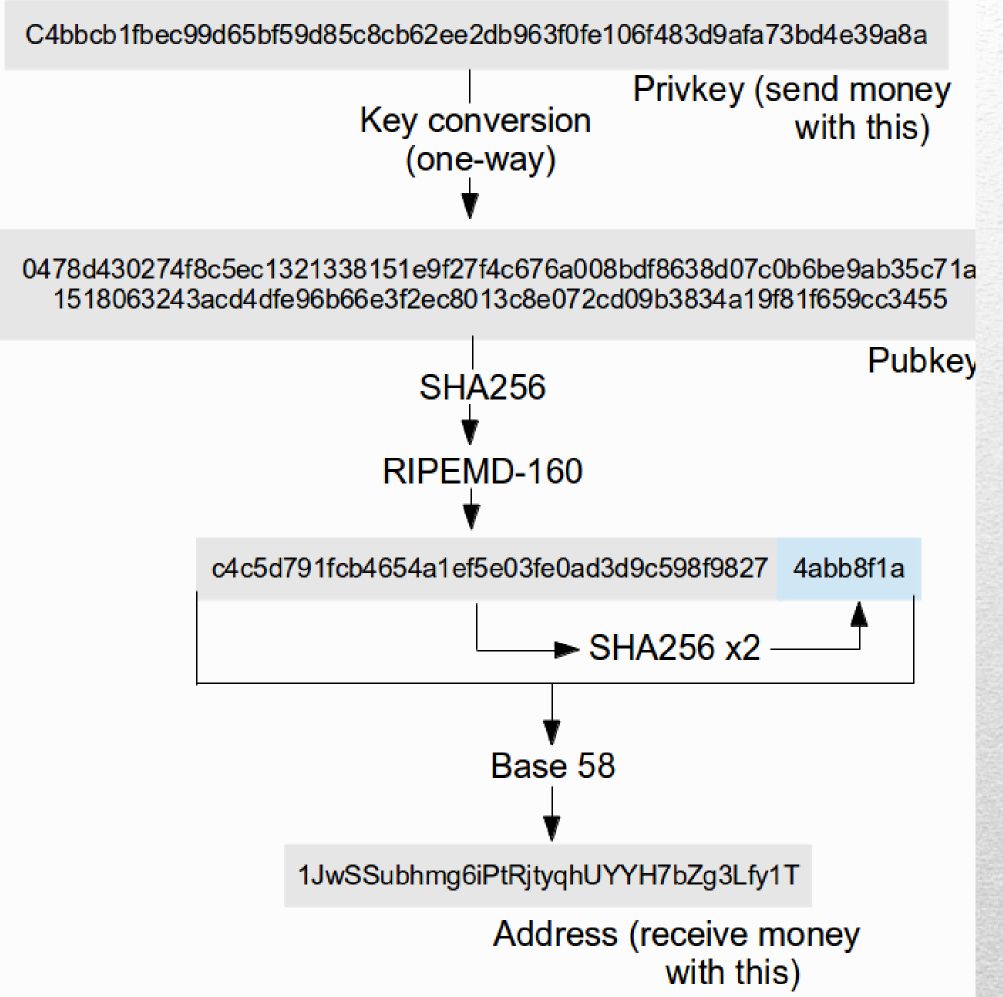
- Cryptographic hash functions (i.e. **SHA-256** and **RIPEMD-160**)
- Public Key Cryptography (i.e. **ECDSA** – the **E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm)

Bitcoin and Cryptography

In Bitcoin, a **transaction** is a record informing the network of a transfer of bitcoins from one owner to another owner.

Ownership of bitcoins is established through digital keys, Bitcoin addresses, and digital signatures.

Digital keys are created and stored offline and consist of a mathematically-related Private-Public key-pair, created using the **E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm (**ECDSA**).

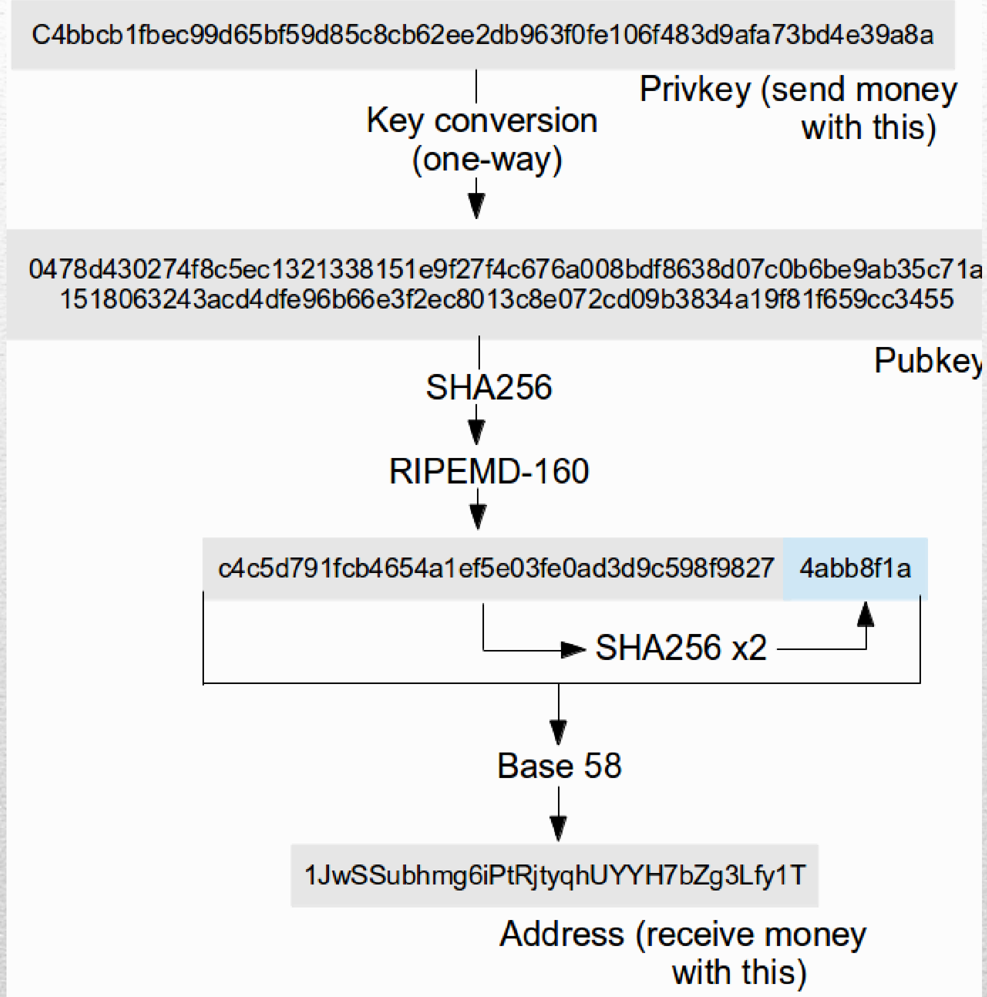


source

Bitcoin and Cryptography

The **Private key (Privkey)** is initially generated at random, and is kept secret at all times. It is used by the current owner of bitcoins to digitally sign a Bitcoin transaction, when he authorizes the transfer to the new owner. A transaction's **digital signature** confirms ownership, and can be used to verify that the transaction is authentic.

The **Public key (Pubkey)** is generated from the Private Key using a one-way cryptographic hash function. It is used by the new owner to validate a transaction's digital signature.

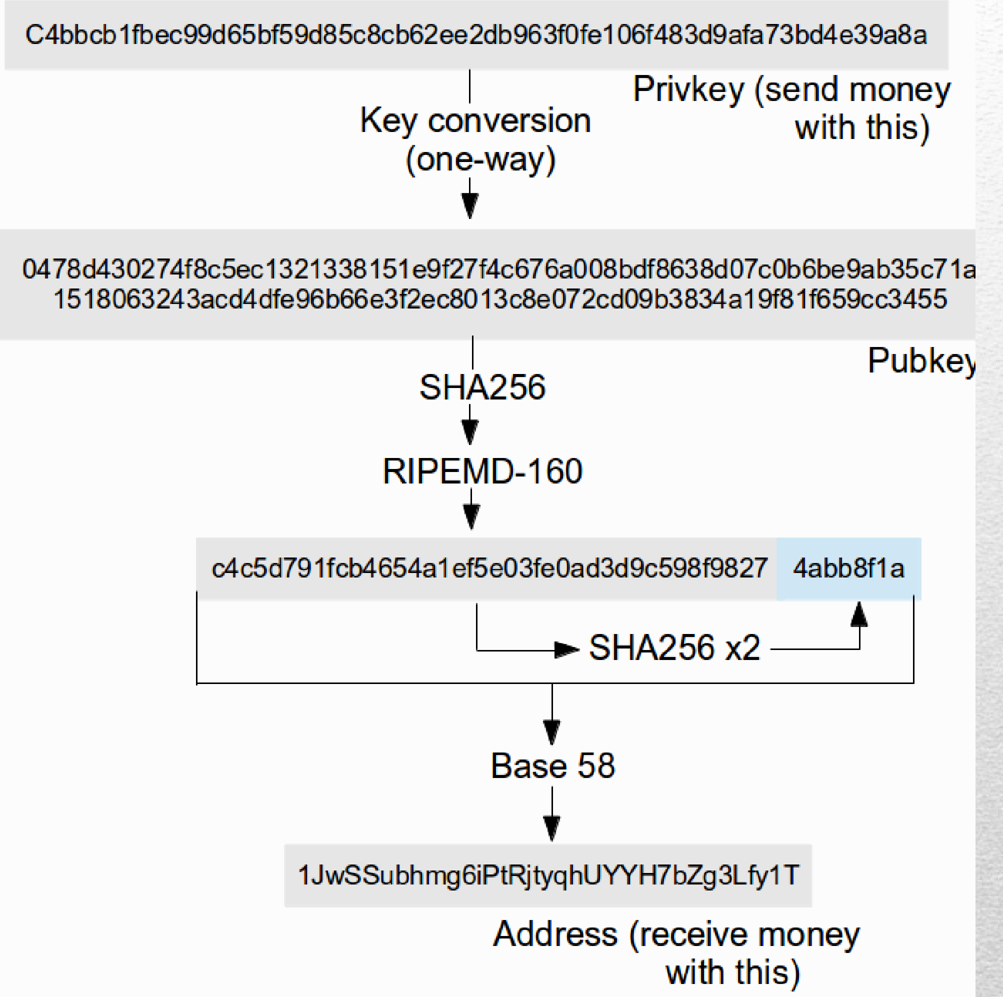


source

Bitcoin and Cryptography

A **Bitcoin address**, which is a participant's unique identifier on the Bitcoin network, is *usually* generated by applying the SHA-256 and RIPEMD-160 cryptographic hash functions (discussed later), in series, on the Public key.

Finally, Bitcoin addresses are encoded using Base58 encoding, which represents an address in a human-readable form of 58 alphanumeric characters.



Source

Bitcoin and Cryptography

When transactions are broadcast over the network, the SHA-256 hash function is used to verify data integrity (i.e. to establish that data was not corrupted or modified during transmission).

All Bitcoin transactions are stored in blocks, which are linked (or "chained") together in sequence to form the ***blockchain***. Cryptographic hash functions are generally used to:

- verify block integrity, and
- establish the chronological order of the blockchain

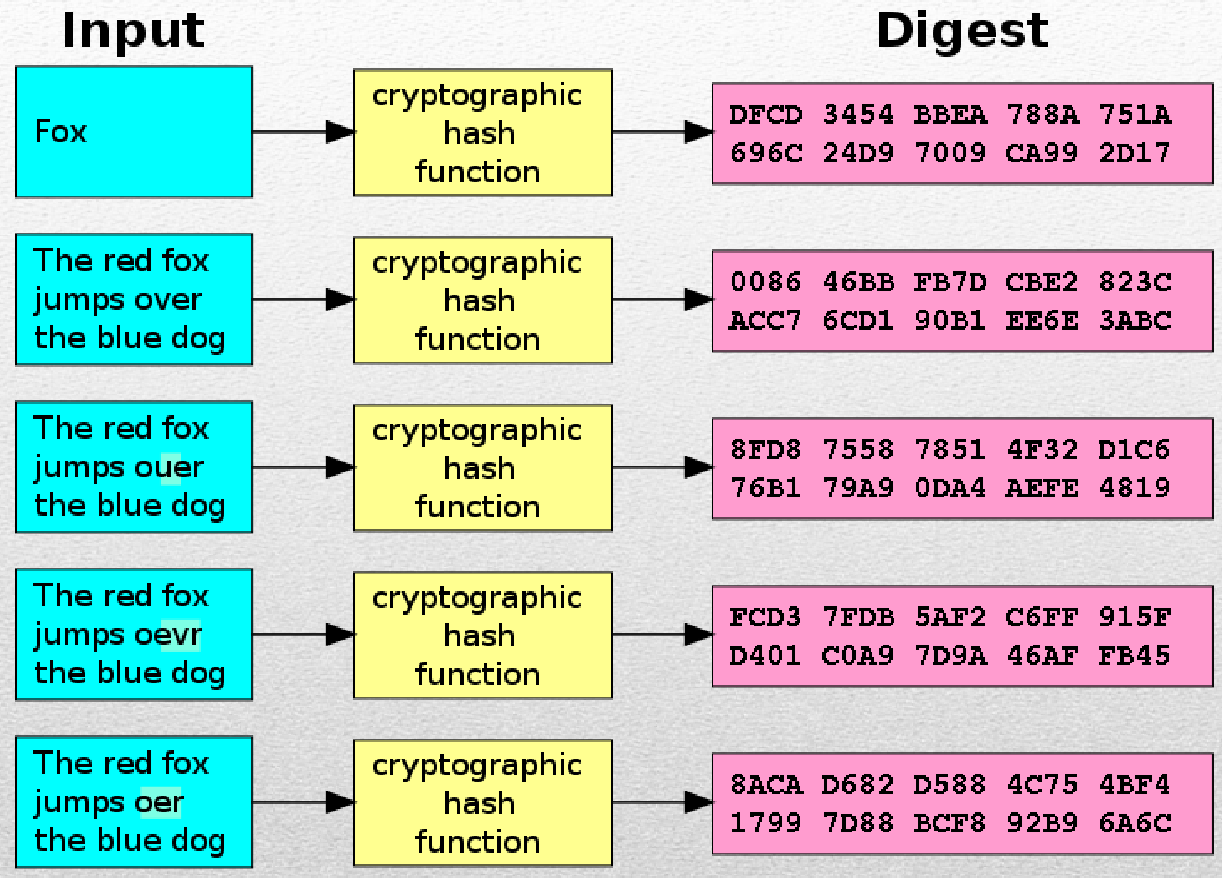
Furthermore, hash functions are used as part of the ***Proof-of-Work (PoW)*** *algorithm*, which is a prominent part of the Bitcoin mining algorithm (discussed later in this session).

The following pages will explain the *basics* of cryptographic hash functions and public key cryptography, as used by Bitcoin.

Hash Functions

A cryptographic **hash function** is a mathematical function commonly used to verify the integrity of data, by transforming *identical* data to a unique, representative, fixed-size code. For example, depending on the hash function applied, *"John Smith"* could always map to the same number, e.g. 10.

Any accidental or intentional modification to the data will change the hash code (e.g. if *"John Smith"* were changed to *"Jon Smith"*, *"Jon Smith"* would map to a different number, e.g. 15, etc.)



Source

Hash Functions

Bitcoin uses the SHA-256 hash function, where the hash code is 256 bits (or 32 bytes) long.

The SHA-256 hash is usually presented as a string of 64 *hexadecimal* characters (i.e. each one of the 32 bytes is represented by 2 hexadecimal characters).

For example, the word "*Bitcoin*" produces the SHA-256 hash shown in the screenshot below (generated below using the *sha256sum* Linux command).

```
# sha256sum
Bitcoin
deb10ca6fd85a5eba792ea8561da390635242f0c37c376f8eb7d7859adbffca9
```

Public Key Cryptography

In Bitcoin, all transaction information is publicly visible to everyone in the network, and transactions are not encrypted. However, Bitcoin relies heavily on digital signatures (one of the main uses of Public Key Cryptography) to verify transactions in the network.

In Public Key Cryptography, two keys are used:

K_{priv}, the **Private key** which must be always kept secret by the owner, and
K_{pub}, the **Public key** which is visible by everyone

Public Key Cryptography

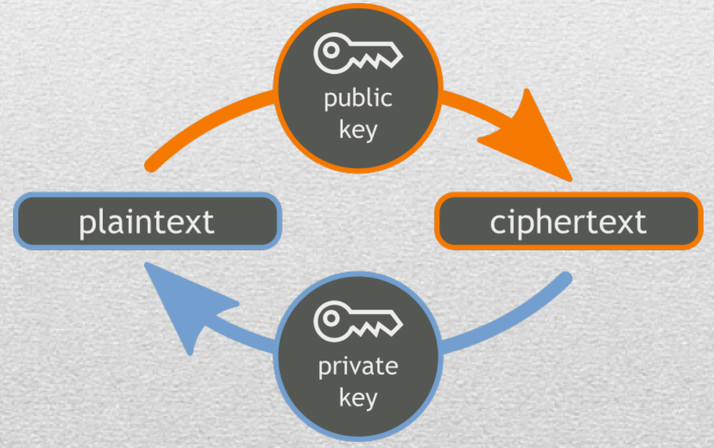
The sender encrypts the message **M** using the recipient's public key: C = encrypt(M, K_{pub})

The recipient decrypts the encrypted message **C** using his own private key: M = decrypt(C, K_{priv})

Where: **C** is the result of encryption (also known as "ciphertext"), and **M** is the unencrypted/ decrypted message (also known as "plaintext").

There is an asymmetric mathematical relationship between the public and private keys:

- The public key can be easily derived from the private key
- The private key is nearly impossible (or computationally infeasible) to derive from the public key



Source: Wikimedia Commons

Public Key Cryptography

At the same time, the public and private keys can also be swapped (i.e. are interchangeable):

For example, if one encrypts a message **M** with his private key: C = encrypt(M, $\mathbf{K}_{priv}$),

then this message can be decrypted using the corresponding public key: M = decrypt(C, $\mathbf{K}_{pub}$)

The above property is used by Bitcoin for digital signing, using **ECDSA (Elliptic Curve Digital Signature Algorithm)** to implement transaction ownership verification:

- The private key is kept secret by the owner, and is used to sign a plaintext message
- The public key (which is visible to everyone) can be used to verify the validity of the plaintext message’s digital signature
- If the plaintext message is long, the corresponding ciphertext might be long as well. Thus, a hash function is applied on the plaintext message first, and the result is used to produce the digital signature, instead.

Digital Signatures

To make a Bitcoin payment, a Bitcoin transaction **T** is constructed. A subset **M** of the information in transaction **T**, is signed as follows:

1. Create transaction T

2. Select subset M of transaction T (e.g. Transaction identifier, transaction instructions, etc.)

3. Compute hash H of M:
   H = sha256(M)

4. Compute a signature **S** by encrypting hash **H** with the sender's private key:
   S = encrypt(H, $\mathbf{K}_{priv}$)

5. Send the signature S and the public key $\mathbf{K}_{pub}$ along with the transaction T to Bitcoin miners.

Digital Signatures

To verify a transaction T received with signature S and public key $\mathbf{K}_{pub}$, a receiver will:

1. Select subset M of the information in transaction T

2. Compute hash H of M:

   H = sha256(M)

3. Decrypt signature S with the public key $\mathbf{K}_{pub}$:

   H' = decrypt(S, $\mathbf{K}_{pub}$)

4. Compare H and H'. If they match, then the signature is valid and the transaction is valid

Digital Signatures

In summary:

- Each transaction associates an amount of bitcoins with a bitcoin address, which is usually produced from a hash of the owner's public key
- When bitcoins are sent to someone, the transaction records the transfer of bitcoins from the current owner’s Bitcoin address to the new owner's Bitcoin address, and includes a valid transaction signature
- When this transaction is broadcast to the Bitcoin network, every peer knows that the new owner of these bitcoins is the owner of the receiving Bitcoin address
- The current owner’s signature verifies for everyone that the transaction is authentic
- The complete history of transactions is kept by every peer in the Bitcoin network, so anyone can verify who is the current owner of any particular amount of bitcoins

Digital Signatures

Both the Public and Private keys are stored in a Bitcoin wallet.

```
>getaddressesbyaccount ""
[
"1aavsnddTKS3fWFiW83t9vwqHCZYrpFAd"
]
>dumpprivkey 1aavsnddTKS3fWFiW83t9vwqHCZYrpFAd
L3nzYUMrpMua59tqgnR7Gk37nvr5458auGzXRWQnUWY5fuZCu6ab
```

A Bitcoin wallet, similarly to a credit card, does not contain any bitcoins, but only the *Private-Public key-pairs* as *tokens* that allow you to access your funds. The output above was produced by the *bitcoind* program and reveals the Private key:

L3nzYUMrpMua59tqgnR7Gk37nvr5458auGzXRWQnUWY5fuZCu6ab

which is used to derive its corresponding Public key, and then the Bitcoin address:

1aavsnddTKS3fWFiW83t9vwqHCZYrpFAd

Some of *bitcoind*'s features will be covered in Session 5(Bitcoin in practice - part 2)

3. Transactions and the blockchain

From Digital Signatures to Bitcoin

We saw how Digital Signatures, a crucial part of many systems involving digital transactions, can be used to convey private information using Public Key Cryptography.

When we need to transfer a digital title of ownership without a central authority, we need a ledger that records these changes in ownership, so that these changes cannot be refuted or altered by malicious activity.

Next, **we will see how Digital Signatures and hash functions are used to form the blockchain** of the Bitcoin protocol, i.e. its distributed ledger of transactions.

P2P Network and Ownership

- Bitcoin is run over a peer-to-peer (P2P) network of computers, called **nodes**
- Nodes are responsible for processing transactions and maintaining all records of ownership
- Anyone can download the free open-source Bitcoin software and become a node
- All nodes are treated equally, and **no single node is trusted**. However, **the system is based on the assumption that the majority of computing power (i.e. at least 51%) will come from honest nodes**
- Ownership records are replicated on every node
- Bitcoin users possess digital keys that allow control over bitcoins recorded in a public ledger (the ***blockchain***)
- The public ledger records transactions transferring ownership of a quantity of bitcoins from one owner to the another, like a double-entry bookkeeping ledger

Addresses

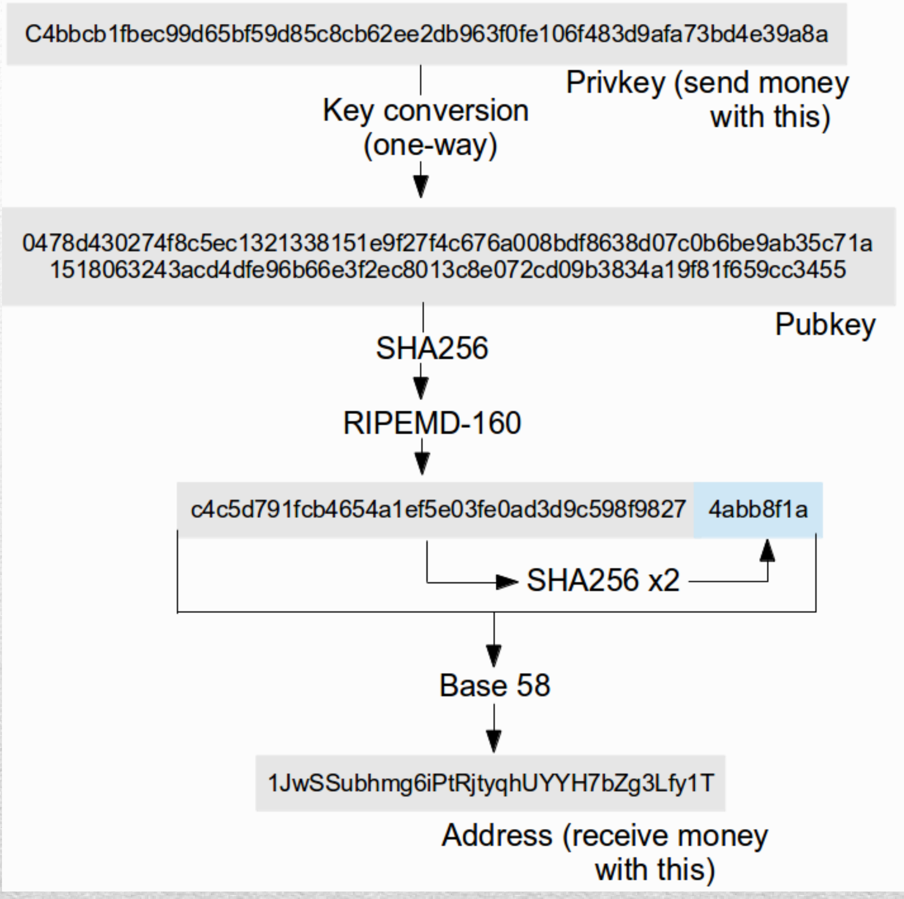
- Transactions in the blockchain do not record the public keys of recipients, but instead use an abstraction called a “Bitcoin address” to record the beneficiary of each amount, allowing for greater flexibility
- To create a Bitcoin address, the Bitcoin client software first generates an ECDSA Public-Private key-pair from a random number
- The Bitcoin address is then generated by applying the following algorithm, in order:

```
version  = (1 byte version number)
keyHash  = RIPEMD-160(SHA-256(publicKey))
data     = version + keyHash
dataHash = SHA-256(SHA-256(data))
checksum = (first 4 bytes of dataHash)
address  = Base58Encode(data + checksum)
```

Addresses

As noted earlier, a Bitcoin address is a computation based on the user's Public key:

- The ***keyHash*** is produced by applying the SHA-256 and RIPEMD-160 hash functions, in series, on the *Pubkey*
- ***Data*** is a concatenation of **keyHash** and an *address **version** number*
- The ***dataHash*** is produced by applying the SHA-256 algorithm twice on ***Data***
- However, only the first 4 bytes of the ***dataHash*** are used as a **checksum**
- The bitcoin ***address*** is a concatenation of ***Data*** and ***checksum*** encoded in Base58 encoding.
- ***Base58Encode*** is a function that encodes binary as text using the Base58 encoding.



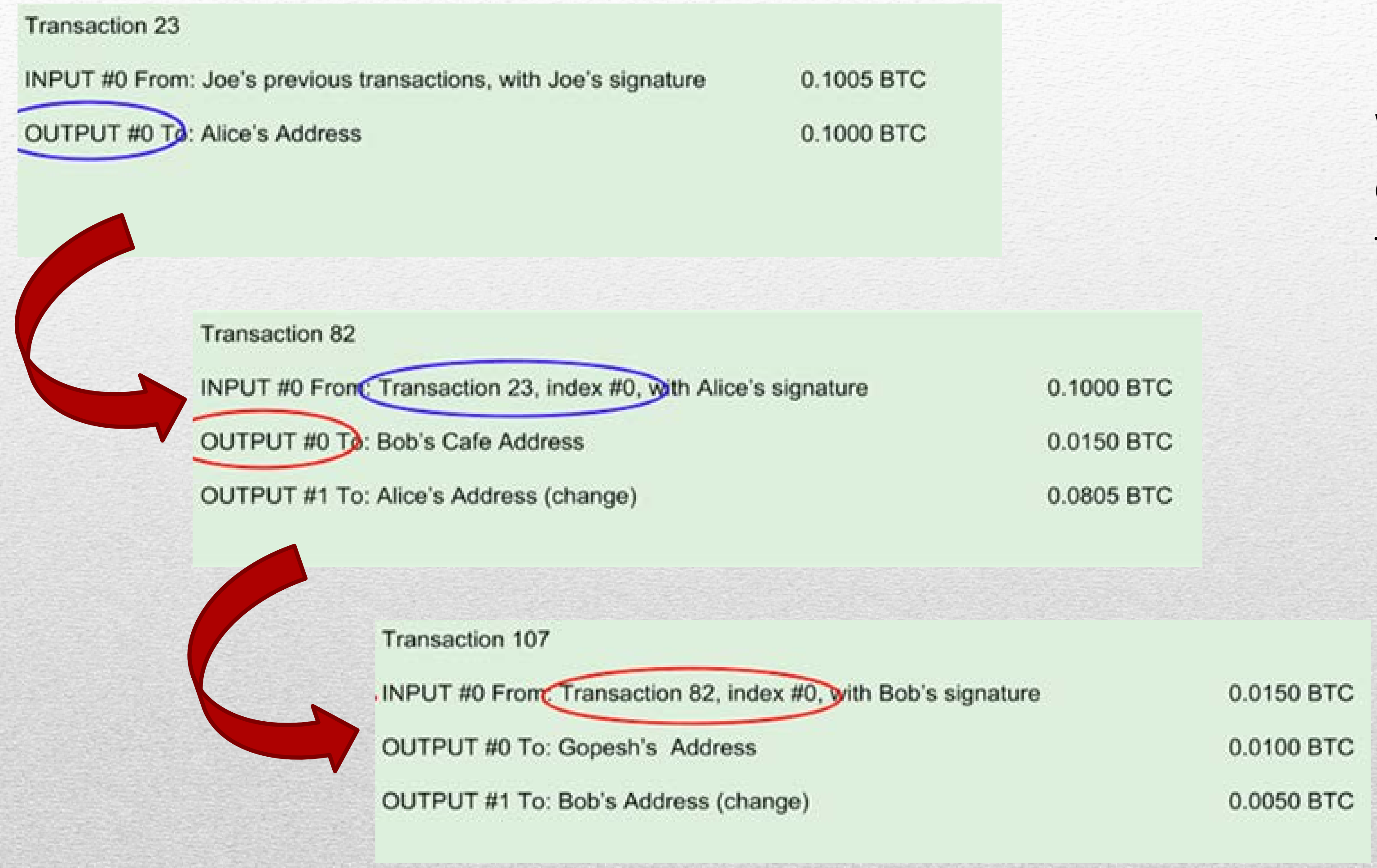
source

Fun fact: While there are 62 characters if we take all small and capital letters and numbers, Satoshi wanted to avoid confusion in Bitcoin addresses over commonly mistaken characters, so he removed 4 of them: O0lI – which is which ?!?

Transactions

- A Bitcoin **transaction** tells the network that the owner of a number of bitcoins has authorized the transfer of some of these bitcoins to another owner
- The new owner can now spend these bitcoins by creating another transaction that authorizes transfer to another owner, and so on, in a chain of ownership
- Transactions are like lines in a double-entry bookkeeping ledger. Each transaction contains one or more **inputs**, which are debits against a Bitcoin account
- On the other side of the transaction, there are one or more **outputs**, which are credits added to a Bitcoin account
- The inputs and outputs (debits and credits) do not necessarily add up to the same amount. Instead, outputs add up to slightly less than inputs and the difference represents an implied **transaction fee**, a small payment collected by the miner who includes the transaction in the ledger
- The transaction contains proof of ownership for an amount of bitcoins (inputs) whose value is transferred, in the form of a digital signature from the owner, that can be independently validated by anyone in the Bitcoin network

Transactions

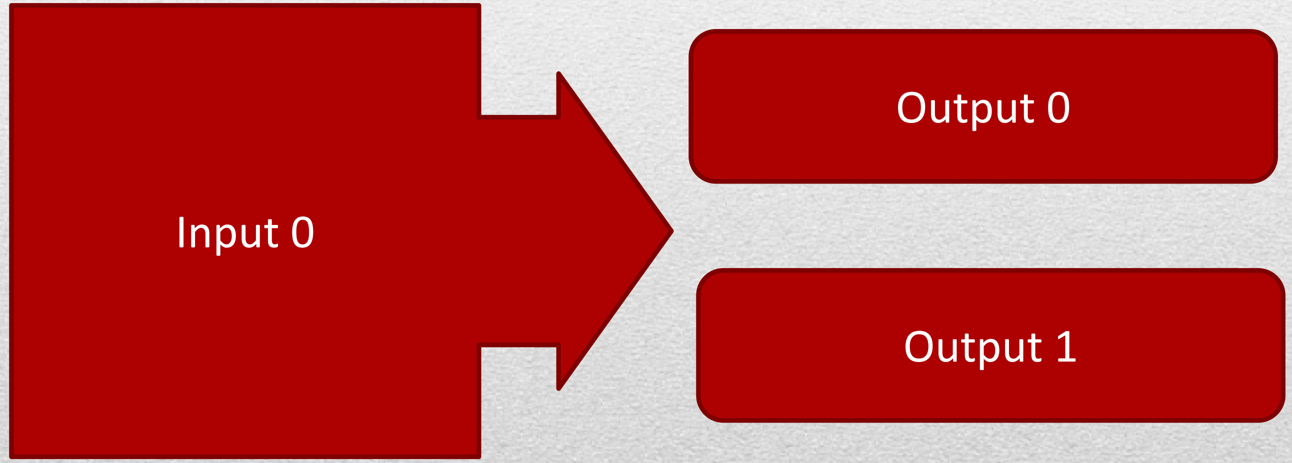


We can see that outputs of one transaction are inputs for the next transaction

(From "Mastering Bitcoin")

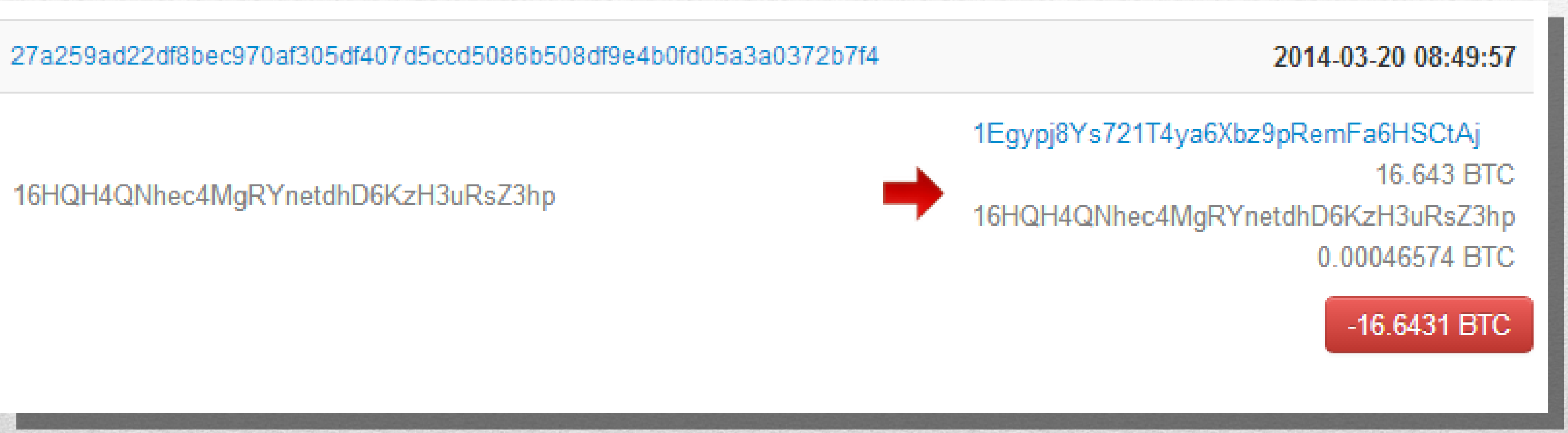
Transactions

The most common form of transaction is a simple payment from one Bitcoin address to another, which often includes some “change” to be returned to the original owner. This type of transaction has one input and two outputs and is shown below:



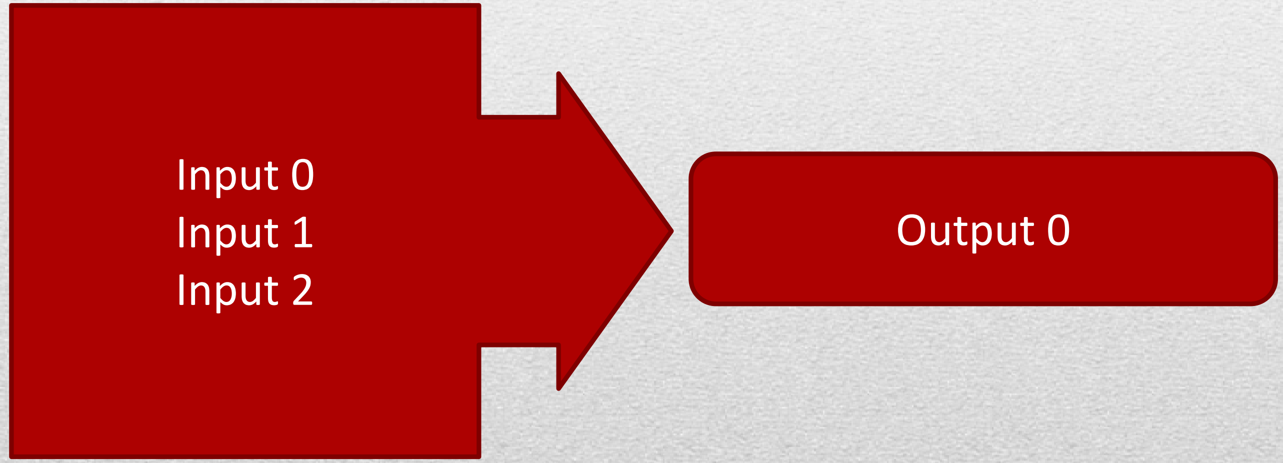
Transactions

This is an example of a real transaction of this type:



Transactions

Another common form of transaction is a transaction that aggregates several inputs into a single output. This represents the real-world equivalent of exchanging a pile of coins and currency notes for a single larger note. Transactions like these are sometimes generated by wallet applications to cleanup lots of smaller amounts that were received as change for payments.



Transactions

This is an example of a real transaction of this type:

a2d7f24a020d2c1c4d1abaec07a4ae8d7fa04a9ec9e1d0230834efd9d48ffccf 2014-01-14 00:36:37

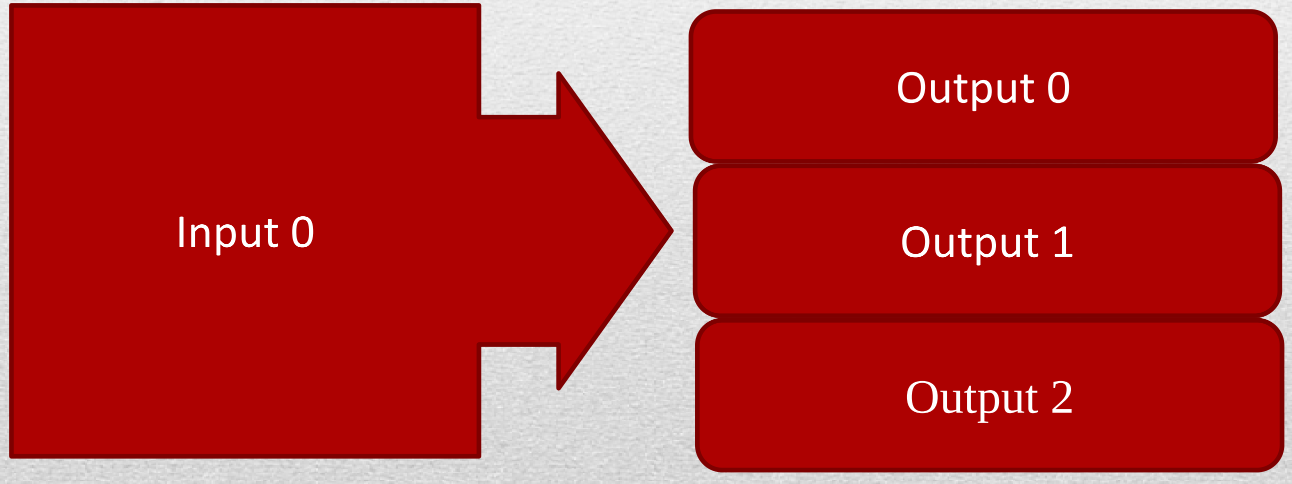
1CXyk23Sy3pnVz8G9EN95HbHzpT9WXXhaB
1HbRuiWGBayVsCcz4goYFypHNUR7huAPbr
1P4mBUEUaZnDpREZD8Tk8BAFMKPnPpP97S
1CdoNnxx3A6QvMqJuuy9ER5MW7MiVjovFH
19BPriDhWRpmPaMVEja42tbgACjXHAbBYA
1FVC7eFrTQiBPUg7jv9AJF3oc4wDvud3GK
1MHAfAXefHKxbjEQWWTajmNSb2wqSXAoTA
18KTTPULXw5NTQQj3b6HrfAMRz6Jh4YQ8f
1QCV36hs1yuYJNSzjkxfwfiW7NhdYGJp3D
19wpPopMDWySLeMhP8LVA71GtUDzPN668x

→ 114RkSMy4q2deixQStcru7pbQFU9hwczEH
16.47174935 BTC

16.47174935 BTC

Transactions

Finally, another transaction form that is often observed in the Bitcoin ledger is a transaction that distributes one input to multiple outputs representing multiple recipients. This type of transaction is sometimes used by commercial entities to distribute funds, such as when processing payroll payments to multiple employees.



Transactions

This is en example of a real transaction of this type:

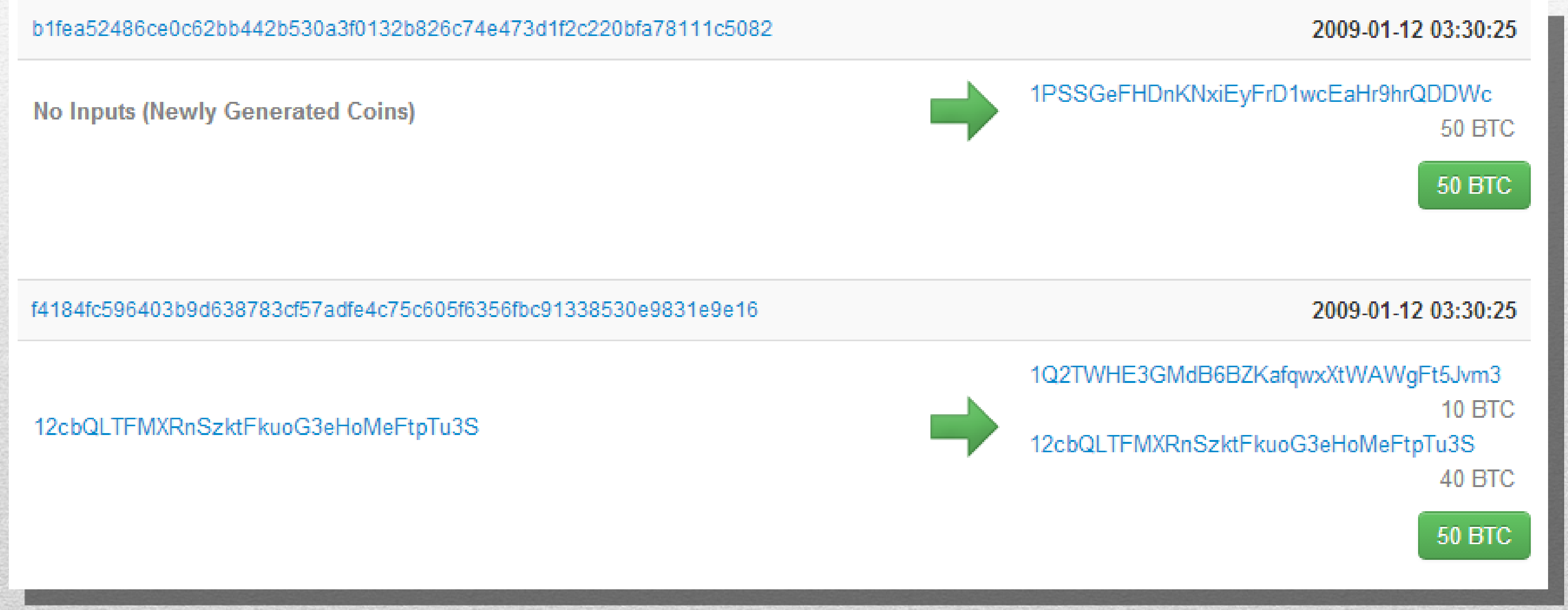
18346e5611acd88c154ac5255ab2c9ec14635f8a8db5b46257ce208f6802f1b0 **2014-05-23 20:19:17**

1HEoyBe7tFefWLXVAXxzXMiMJVY5BM4CDL	16k1sKLppU6Hhfx5XepjxR7HSjWouLcG5r 0.1655 BTC 1HJtBanfm1aP5abD3uuD2eGf5U8mDoVR9x 1.3187741 BTC 1JxJfeXLrnGisgx9cnXTJMbSxH5WbLSYEr 0.01106288 BTC 1FynGED6mqRTXwG2uHnjjPTVY5j1S22u1u 0.49993333 BTC 14RmyxMZSPWooHAv8BrPt75wphyXJRwksD 0.00025 BTC 1Nm9uDTps87VkMvZvCbhHoZwfEiExKu6rn 0.30738106 BTC

-2 BTC

Transactions

Below is the first transaction ever performed between Satoshi and Hal Finney. We can also see the coinbase transaction that rewarded that block's discovery with 50BTC without having any inputs.



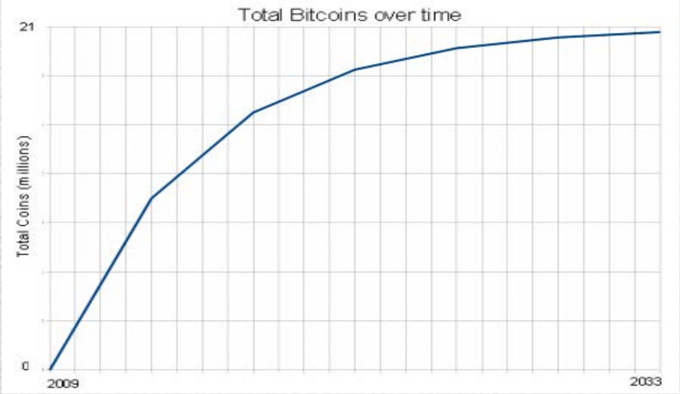
4. Mining

Mining

The Bitcoin system of trust is based on computation. **Transactions** are bundled into **blocks**, which require an enormous amount of computation to “prove” (or “confirm”), but only a small amount of computation to verify as “proven”, in a process called **mining**.

Mining serves two purposes in Bitcoin:

- **Mining creates new bitcoins in each block**, almost like a central bank printing new money. The amount of bitcoins to be created is fixed and diminishes with time (see Session 2)
- **Mining creates trust** by ensuring that transactions are confirmed only when enough computational power was devoted to the block that contains them. More blocks mean more computation, which means more trust.



Bitcoin production over time

Mining algorithm

Mining consists of the following steps, which are performed in a continuous loop:

1. **Bundling transactions** that were broadcast on the peer-to-peer network **into a block**. Each miner can arbitrarily decide which transactions to include in their block
2. **Verifying that** all **transactions** in the block **are valid**
3. **Selecting the most recent block** on the longest path in the blockchain **and inserting a hash of its header** into the new block
4. **Trying to solve the *Proof of Work (PoW) problem*** for the new block and simultaneously watching for new blocks coming from other nodes
    - If a solution is found to the Proof of Work problem, the new block is added to the local blockchain and broadcast to the peer-to-peer network

Proof of work

Miners search for acceptable blocks using the following procedure, performed in a loop:

1. Increment (add 1 to) an arbitrary number in the block header called a **nonce**
2. Take the hash of the resulting block header
3. Check if the hash of the block header, when expressed as a number, is less than a predetermined target value

If the hash of the block header is not less than the target value, the block will be rejected by the network. Finding a block that has a sufficiently small hash value is the Proof of Work problem.

Mining performance, therefore, is measured in hashes/sec. Currently the performance of miners is measured in GH/s (billions of hashes per second) or TH/s (trillions of hashes per second)

H/ s = Hashes per second
KH/ s = Kilo Hashes per second
MH/ s = Mega Hashes per second
GH/ s = Giga Hashes per second
TH/ s = Tera Hashes per second
PH/ s = Peta Hashes per second

1,000 H/ s = 1 KH/ s
1,000 KH/ s = 1 MH/ s
1,000 MH/ s = 1 GH/ s
1,000 GH/ s = 1 TH/ s
1,000 TH/ s = 1 PH/ s

Mining Difficulty

- Bitcoin nodes actively regulate the rate of creation of new blocks
- As more miners join, the rate of block creation will go up. As the rate of block creation goes up, the **mining difficulty** rises to compensate, which pushes the rate of block creation back down
- **The creation of new blocks must take an average of 10 minutes**
(Ten minutes was specifically chosen by Satoshi Nakamoto as a tradeoff between fast confirmation time and the amount of work wasted due to chain splits. Read more about this here)
- The regulation is done by periodically adjusting the hash target value for blocks
- Every 2,016 blocks (which ideally spans every 2 weeks, with each block taking 10 minutes to confirm) Bitcoin nodes calculate a new difficulty accordingly, based on the time it took to mine the last 2,016 blocks

Mining Reward

- Solving the Proof of Work problem requires a lot of computing power and that power costs money. To encourage people to invest their resources in mining, Bitcoin provides a **reward** in each successfully mined block (plus the **transaction fees** of the transactions contained in the new block)
- When a block is discovered, the discoverer will award themselves a certain number of bitcoins, which is agreed-upon by everyone in the network
  - Currently this bounty is 25 bitcoins
  - Based on Bitcoin's algorithm, this bounty halves every 210,000 blocks (i.e. approximately every 4 years)
  - Eventually, the reward will be removed entirely when the limit of 21 million bitcoins is reached by the year 2140
  - After that, transaction processing will be rewarded solely by transaction fees

Solo Mining

- **Solo mining** is when you use your computer (or specialized mining hardware) to search for blocks. In this case, you are getting paid only when you personally find a block, receiving the full amount of the reward, plus any transaction fees
- This type of mining is efficient only when mining difficulty is low enough to expect to find new blocks often
- Today (May 1, 2015), the hashrate of the Bitcoin network is 334,248,521 GH/s, or 334 PH/s. In order to mine one block per day, your mining rig should perform with the speed of:

  334,248,521 *GH / 24 Hours / 6 Blocks per Hour = 2,321,170 GH/s ≈ 2,3 PH/s*

- In this case, you are getting paid only when you personally find a block, receiving the full amount of it, plus any transaction fees

Pool mining

- **Mining pools** collect all of the hashing from miners and basically run them off of one account
- When a block is found, the mining pool's wallet is the one that gets the payment, and then the payments are split and distributed into each miner's site account based on their personal contribution towards finding the block
- For example, if a miner contributed half of the pool's shares into finding the new block, they would get half of the block reward

Mining Pools

Most pools (with very rare exceptions, like P2pool) work using the following algorithm:

- The pool server prepares a block with the coinbase transaction pointing to the pool's address
- Miners in the pool contact the pool server and make a *getwork* request to get the block to work on
- Each miner tries to solve the Proof of Work problem for the block, by incrementing the nonce and hashing the block header
- Whenever a miner finds a hash value that is below the easier target, it submits the solution to the server for a share
- The mining server verifies submitted shares and tracks how many each miner has
- When a miner finds a solution to the Proof of Work problem, the server pays out the reward in proportion to the number of shares each miner earned since the last payout
- Miners periodically contact the pool server for updates on what to work on, in case a new block was discovered

Pool Mining - Rewards

Mining pools use different distribution schemes, the most popular of which are:

- **PPS (Pay Per Share)**: Each miner gets paid a guaranteed amount for every share they submit. Pools that use this method often employ custom pool difficulties as well, rather than allowing for variable difficulties. This makes the calculations a lot easier and ensures every miner is fairly treated
- **PPLNS (Pay Per Last Number of Shares)**: Each miner gets paid based on the last x number of shares after a block is found. For example, if it is set to pay at 5,000 shares and a miner has contributed 2,500 of the last 5,000, this miner would get half of the block's payment
- **Proportional**: Each miner gets paid based on the proportion of shares since the last block. This is a lot like PPLNS, but instead of only counting n shares, it counts every share between each block and then calculates the payments based on each person's proportional amount

Mining Hardware

- **CPU mining**: Initially, Satoshi's Bitcoin client software did mining on a user's PC (i.e. CPU mining), but now CPUs have been eclipsed by more efficient mining hardware
- **GPU mining**: GPUs (i.e. Graphics Processing Units on Graphics cards) are designed for doing lots of mathematical calculations in parallel and are orders of magnitude faster than CPUs
- **FPGA (Field Programmable Gate Arrays) mining**: An intermediate step between a fast processor and a dedicated ASIC, FPGAs were used until ASICs emerged and dominated Bitcoin mining
- **ASIC mining**: ASICs (Application-Specific Integrated Circuits) are custom built for a particular application and are thus orders of magnitude faster than GPUs, which are general-purpose. In Bitcoin, these chips are customized to perform SHA-256 hashing. Today, ASIC mining is the only economically efficient mining technique

5. Conclusions

Conclusions

- The Bitcoin protocol relies on the following cryptographic techniques:
  - Hash functions (i.e. SHA-256, RIPEMD-160)
  - Digital signatures (i.e. using ECDSA)

- The Bitcoin network is:
  - Distributed (i.e. it does not have any central points of trust/failure)
  - Secure (as long as more than 50% of nodes are trustworthy)
  - Reliable (transaction ledgers are massively replicated by all network nodes)

- Bitcoin generation is self-regulated and based on a mathematical algorithm

6. Self-Assessment Exercises and Further Reading

Self-Assessment Exercises

1. Navigate your web browser to http://blockchain.info. Find the list of the "Latest Transactions" and choose one. Click the transaction identifier (e.g "6e270b2cdb854b8a997be9bb7...") in order to get that transaction's details. Answer the following questions:
    i. How many inputs and outputs does the transaction have?
    ii. What is the transaction type?
    iii. What is the Bitcoin address(es) of the sender(s)?
    iv. What is the Bitcoin address(es) of the recipient(s)?
    v. What is the amount of bitcoins spent?
    vi. What is the amount of bitcoins received?
    vii. What is the fee for this transaction?
    viii. How is the transaction fee connected with the inputs and outputs?

    Make your own interpretation of the transaction.

Self-Assessment Exercises

2. Navigate your web browser to https://blockchain.info/. Enter the following tx hash into the search textbox: 6b6c09f7cbe509ef6cb7515ac8619abbbcc4362170c8b56142ca95f8c0df6021

    i. What is the block height?
    ii. What is the hash of the previous block?
    iii. What is the hash of the next block?
    iv. How many transactions does the block contain?
    v. What is the block *nonce*?
    vi. What is the address of the miner?
    vii. What was the miner's reward?
    viii. Why does the block hash start with zeroes?
    ix. How much did the miner collect in fees?

Self-Assessment Exercises

3. Perform the following actions and answer the questions:
   i. Navigate your web browser to http://blockexplorer.com/q/getdifficulty. What is the current difficulty?
   ii. Navigate your web browser to http://blockexplorer.com/q/estimate.What is the estimated next difficulty?
   iii. Navigate your web browser to http://bitcoin.sipa.be/. Examine the first chart. How does difficulty increase?
   iv. Examine the block used in the Assignment #2. What was the difficulty? Has the difficulty changed since the block was created?

Self-Assessment Exercises

4. Navigate your web browser to https://blockchain.info/pools .
    i. Examine the pie-chart as well as block statistics. Which is the most powerful pool?
    ii. What is the performance of that pool?
    iii. Navigate your web browser to http://mining.bitcoin.cz/. Examine http://mining.bitcoin.cz/stats/. Compare these two pools.

Further Reading

Bitcoin: A Peer-to-Peer Electronic Cash System, Satoshi Nakamoto https://bitcoin.org/bitcoin.pdf

(Satoshi Nakamoto's original paper)

Mastering Bitcoin, Andreas M. Antonopoulos, http://shop.oreilly.com/product/0636920032281.do

(A text book introducing the Bitcoin concept though storytelling)

Bitcoin Wiki https://en.bitcoin.it/

Bitcoin Internals, Chris Clark
http://www.amazon.com/Bitcoin-Internals-Technical-Guide-ebook/dp/B00DG8EPT0

Triple Entry Accounting, Ian Grigg, December 25, 2005. http://iang.org/papers/triple_entry.html

Hashcash - a denial of service counter-measure , Adam Back, August 1, 2002.
http://www.hashcash.org/papers/hashcash.pdf
(Introduces the Proof-of-Work concept)

b-money, Wei Dai, November 27, 1998. http://www.weidai.com/bmoney.txt
(Introduces the idea of a public ledger)

How Bitcoin transactions work http://visual.ly/bitcoin-infographic

How Bitcoin Works Under the Hood https://www.youtube.com/watch?v=Lx9zgZCMqXE

Questions?

Contact Us

Email:
dc@unic-online.com

Twitter:
@MScDigital