INTRODUCTION TO BITCOIN

Paper Cash

- *Coins* (including *paper bills*) are issued by the Bank of Canada in accordance with an economic policy.
- Suppose that Alice wishes to give a coin to Bob (in return for some goods or services).
- Bob can examine the coin to ensure that it is *valid* (i.e., not counterfeit).
- *Double spending* is not a concern because Alice cannot give the same (valid) coin to two different parties.
- *Payer anonymity* and *payment untraceability* are provided.

Features of Paper Cash

- *Recognizable* (as legal tender)
- *Portable* (easily carried)
- *Transferable* (without involvement of the financial network)
- *Divisible* (has the ability to make change)
- *Unforgeable* (difficult to duplicate)
- *Untraceable* (difficult to keep a record of where money is spent)
- *Anonymous* (no record of who spent the money)

Note: Many of these features are not available with credit cards.

Chaum's E-Cash

Outline of the *online* e-cash scheme:

1. Alice visits the Bank and requests a $100 withdrawal.
2. Alice gives the Bank the blinded hash of M, where $M =$ "This is $100 bill, #serial_number".
3. The Bank signs the blinded hash after which Alice obtains the *coin* (M, s), where s is the Bank's signature on s using its $100 dollar RSA private key.
4. Alice gives the coin (M, s) to Bob.
5. Bob forwards (M, s) to the Bank, who *verifies* the signature and that it is not in the *spent-coin database*.
6. The Bank informs Bob whether or not the coin is valid.
7. If valid, Bob completes the transaction with Alice.

Bitcoin

- An electronic cash scheme invented by *Satoshi Nakamoto* (a pseudonym) in 2008.
- Bitcoin is *decentralized*, i.e., there is no "Bank".
  - How can the creation of coins be regulated?
  - How does the recipient of a coin ensure it has not been previously spent?
  - Note: Payer anonymity and payment untraceability are *not* primary goals of Bitcoin.
- Anyone can use Bitcoin:
  - Download a *wallet* from `bitcoin.org`.
  - Obtain bitcoins by "*mining*" or from an exchange such as *kraken* or *BTC China*.

Bitcoins

- The first bitcoins were generated by Satoshi Nakamoto on *Jan 3 2009*.
- The basic unit of bitcoin currency is *1 BTC*.
- Each BTC can be divided into 100 million pieces, the smallest of which, i.e., *0.00000001 BTC*, is a *satoshi*.
- Bitcoins can be generated (i.e., *mined*) by anyone.
- They are generated at the rate of R BTC every 10 minutes (approximately).
- Initially, $R = 50$. On Nov 28 2012, R was lowered to 25. On Jul 9 2016, R was lowered to 12.5.
- R will be halved over time, until the year *2140*, when a total of *21 million BTC* will have been generated.
- By March 2017, over *16 million BTC* had been generated.

Value of a Bitcoin

The US dollar value of 1 BTC has fluctuated widely:
(see `coinbase.com/charts`)

†May 22 2010:	$0.0025	Jul 6 2013:	$69.31
Jul 17 2010:	$0.08	Oct 31 2013:	$127.25
Jan 1 2011:	$0.30	Nov 30 2013:	$1126.82
Feb 9 2011:	$1.00	Jan 1 2014:	$747.56
Jun 8 2011:	$31.91	Jan 3 2015:	$289.86
Jan 1 2013:	$13.30	Jan 2 2016:	$433.23
Apr 9 2013:	$ 223.10	Mar 11 2017:	$1183.65

†10,000 BTC for a $25 pizza order

See: `bitcointalk.org/index.php?topic=137.0`

Why Use Bitcoin?

- It's *decentralized*.
  - Not under the control of any government.
  - Not under the control of any bank, credit card company, or other financial institution.
  - Anyone can use it (even if you don't have a credit history).
  - It's (relatively) easy to use.
- Transactions are *irreversible*.
- Transaction *fees are low* (even across borders).
- Of course, there are many reasons *not* to use Bitcoin. We will not dwell on those for now....

Elements of Bitcoin

1. *Transaction*: The transferring of a coin from one user to another. All transactions are public and are broadcast to all users.
2. *Peer-to-peer network*: The users of Bitcoin are organized in a peer-to-peer network.
3. *Blocks*: Every 10 minutes or so, the latest transactions are verified and collected in a block. This block is hashed and (cryptographically) linked with other blocks. The block is broadcast to the peer-to-peer network.
4. *Block chain*: The list of blocks is called the block chain. It contains a record of *all* past transactions.
5. *Mining*: The process of verifying transactions and compiling a block is called mining. A successful miner receives a *reward* (new BTCs).
6. *Proof-of-work*: To successfully compile a block and receive a reward, the miner has to solve a cryptographic challenge.

Main Cryptographic Ingredients

1. *SHA-256* hash function.
2. *ECDSA* with the *secp256k1* elliptic curve:

$$E : Y^2 = X^3 + 7 \text{ over } \mathbb{Z}_p,$$

where

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$$

$q = \#E(\mathbb{Z}_p)$ is a 256-bit prime.

$P \neq \infty$ is a fixed point in $E(\mathbb{Z}_p)$.

The hash function used is SHA-256.

Key Pairs (for ECDSA)

- Each user selects $a \in_R [1, q-1]$ and computes the elliptic curve point $A = aP$.
- The user's ECDSA *private key* is a; the user's ECDSA *public key* is A.
- We will denote Alice's key pair by (a, A), Bob's key pair by (b, B), Chris's key pair by (c, C), etc.
- In Bitcoin, a user's public key is used to identify the user.
- More generally, a user can select a *different key pair* for each transaction.

Transactions

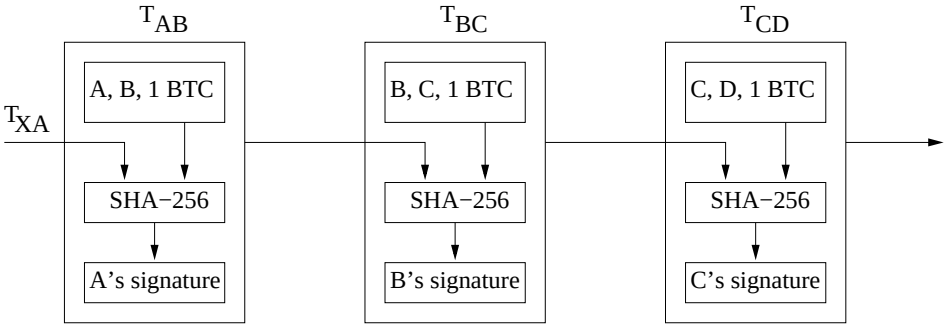
- A *transaction* is the transfer of a coin from one user to another user.
- Suppose that Alice has a coin, say of value 1 BTC.
- The transaction in which Alice obtained this bitcoin is represented by T_{XA}.
- Suppose Alice wishes to give this coin to Bob. The transaction is represented as follows:

$$T_{AB} = \{\widetilde{T_{XA}}, A, B, \text{ 1 BTC}\}_A,$$

  where $\widetilde{m}$ denotes the hash of m, and $\{M\}_A$ denotes a message M and its ECDSA signature with respect to the public key A.
- This transaction is *broadcast* to the entire network.
- Note: The transaction contains Alice's and Bob's public keys, but not their names.

Transactions (2)

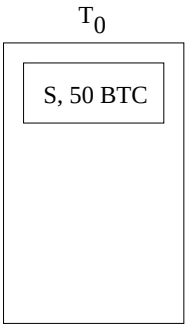
- Similarly, if Bob then gives the coin to Chris, the transaction is represented as: $T_{BC} = \{\widetilde{T_{AB}}, B, C,$ 1 BTC$\}_B$.
- The 1 BTC coin can be thought of as the *chain* of transactions:



- Anyone can verify the signatures to *verify* the chain of ownership.
- Questions: How was this coin generated in the first place? How can the recipient verify that a coin has not been double-spent?

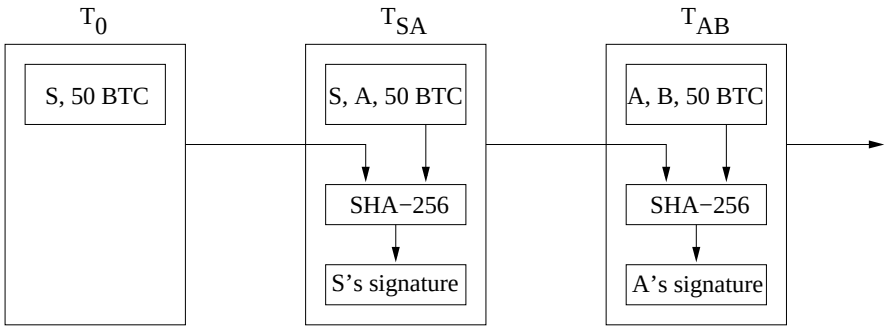
The First Bitcoins

- The first bitcoins were generated by Satoshi Nakamoto on *Jan 3 2009*.



  Here, S is Satoshi Nakamoto's public key.
- This transaction is embedded in the Bitcoin software.

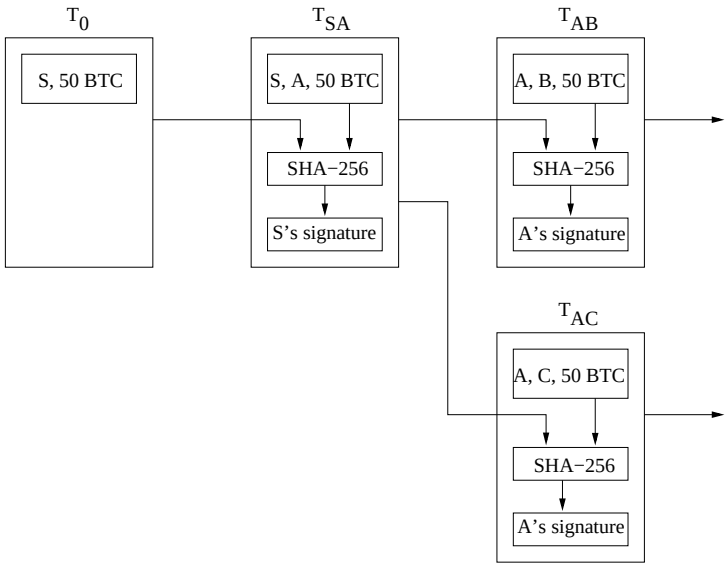
Transaction Chain for the First Bitcoins



Questions: How are the other bitcoins generated?

How can the recipient verify that a coin has not been *double-spent*? (Without using a trusted central authority.)

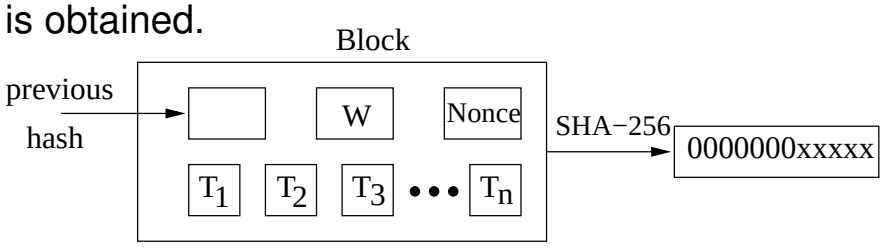
Transaction Chain for the First Bitcoins (2)



Only one of the transactions T_{AB}, T_{AC} should be accepted as valid; the other transaction should be rejected.

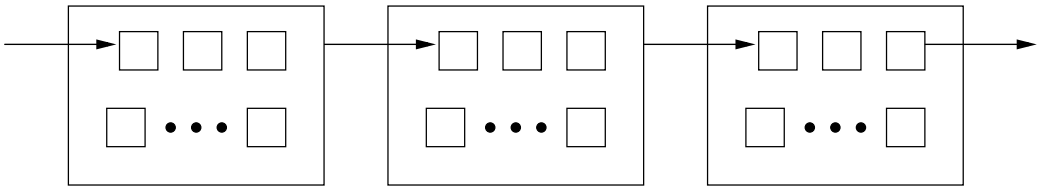
Proof-of-Work

- Recall that all transactions are broadcast to all users.
- Any user (with public key) W can volunteer to collect all transactions $T_1, T_2, \ldots, T_n$ that it received in an interval of time, say the previous 10 minutes.
- The user W verifies that these transactions are valid and that the corresponding coins have not been previously spent.
- The user forms a *block* consisting of the hash of the previous block, the user's public key W, a *nonce*, and $T_1, \ldots, T_n$.
- The nonce is incremented until a hash value that begins with t zeros is obtained.



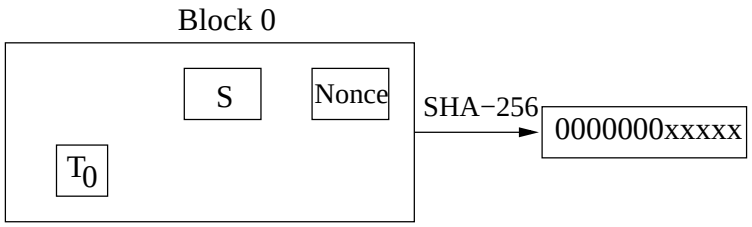
- The block is broadcast to the network.

The Block Chain



- Users will accept a block if all the transactions in it are valid, if the coins have not been previously spent, and if the hash value begins with t zeros.
- Users show their acceptance of the block by using its hash as the "previous hash" for the next block, thereby growing the block chain.
- The block chain serves as a *ledger* that records all transactions.

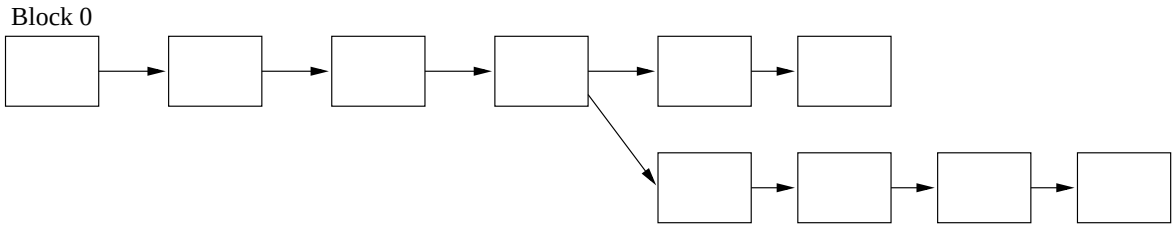
The Genesis Block



Created by Satoshi Nakamoto (S) on Jan 3 2009.

Block 0 is embedded in the Bitcoin software.

Forks in the Block Chain



- There is a possibility that two blocks are created around the same time by two different users.
- This causes a *fork* in the block chain.
- To remedy the fork, users will trust the longest chain and continue to grow that chain.
  - More precisely, users will trust the chain that that was most difficult to generate.
- The blocks that are not part of the longest chain are dropped and the transactions in them are returned to the miners' memory pool of unverified transactions.

Mining

- *Incentive*: The block creator (W) is awarded R BTC (currently, $R = 12.5$) [*mining*].
- *Work factor*: The target t is updated every 2016 blocks (2 weeks) to ensure that the average time it takes to generate a block is 10 minutes.
- Currently, the bitcoin network is generating hashes at the rate of approximately $2^{61.6}$ per second. The hash difficulty is approximately $t = 70$.
- A PC can do approximately 2^{23} hashes per second. So, one PC will take about 4,000,000 years to generate a block.
- *Mining pools*: Users form mining pools and share an award.

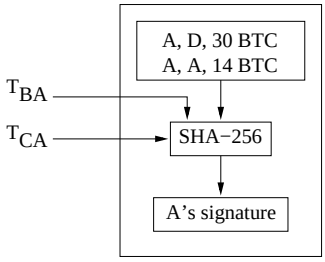
Security Notes

- If A gives B a coin, then B should complete the transaction with A only after the transaction T_{AB} appears in the block chain, perhaps followed by several more blocks.
  - Transactions are *not* instantaneous.
  - If the transaction is accepted instantaneously, B has to accept the risk that A might double spend the coin.
- Bitcoin is "secure" as long as honest users collectively control more CPU power than any cooperating group of users.
- Since all transactions are public, *payer anonymity* and *payment untraceability* are not guaranteed.

Transactions with Multiple Inputs/Outputs

Bitcoins can be combined and split.

- Suppose that Alice (A):
  - received 25 BTC from Bob (B) in Transaction T_{BA}
  - received 20 BTC from Chris (C) in Transaction T_{CA}.
- Suppose that Alice wishes to:
  - give 30 BTC to David (D)
  - leave 14 BTC to herself as change
  - give 1 BTC as a *transaction fee*.
- Here is the corresponding transaction:
- The transaction fee is claimed by the miner who validates this transaction.



Transactions with Multiple Inputs/Outputs (2)

Suppose that Bob owns two public keys, B and H.

Suppose that Bob received 9 BTC in transactions T_1 and T_2:

T_1: 1 BTC from A to B
3 BTC from D to E
2 BTC from C to H

T_2: 5 BTC from C to F
6 BTC from G to B

Suppose Bob wishes to give 2.5 BTC to F, 3 BTC to I, 1.5 BTC as change to B, 1.75 BTC as change to H, and offer a transaction fee of 0.25 BTC.

He forms the transaction T_3:
Inputs are: $(\widetilde{T_1}, 1)$, $(\widetilde{T_1}, 3)$, $(\widetilde{T_2}, 2)$
Outputs are: $(F, 2.5)$, $(I, 3.0)$, $(B, 1.5)$, $(H, 1.75)$.
T_3 has 3 signatures, with public keys B, H, B.

Miscellaneous Notes

- *Secure storage*: Suppose Alice has some unspent coins, and that these coins were paid to her public key A.
  - If an attacker obtains a copy of Alice's wallet, then the attacker can spend the coins corresponding to A.
  - If Alice deletes (or loses) the private key a corresponding to A, then all the coins corresponding to A are lost forever.
- *Mt. Gox*: A Bitcoin exchange based in Tokyo. It "lost" 850,000 BTC and declared bankruptcy in February 2014. Later, it "found" 200,000 BTC.
- *Mining costs*: Mining requires *hardware* and *electricity*.
- Several technical details have been omitted including:
  - A public key is identified by its 160-bit hash value.
  - The use of Merkle hash trees to minimize the size of a block.

Exploring Bitcoin

- Bitcoin magazine: `bitcoinmagazine.com`
- Download a wallet: `bitcoin.org`
- Live block chain: `blockchain.info`
- Bitcoin Block Explorer: `blockexplorer.com`
- Genesis Block: `blockexplorer.com/b/0`
- Block 1: `blockexplorer.com/b/1`
- Block 100,000: `blockexplorer.com/b/100000`

Ethereum

- Invented by *Vitalik Buterin* in 2013.
- Blockchain-based decentralized computing platform.
- Underlying cryptocurrency is called *ether*.
- Supports a Turing-complete programming language.
- Permits full *smart contract* functionality.
- Potential decentralized applications include:
  - Lotteries.
  - Mobile payment services.
  - Crowdfunding (a.k.a. Kickstarter).
  - Democratic autonomous organizations (DAOs).