From Timestamping to Blockchain

Introduction to Computer Security

week 14

Timestamping servies

- People need to certify a document existed on a certain date (e.g., in a dispute)
  - copyright
  - patent

Timestamping servies

- People need to certify a document existed on a certain date (e.g., in a dispute)
  - copyright
  - patent
- Notary or lawyers testify that a letter or document existed on a certain date

Timestamping servies

- People need to certify a document existed on a certain date (e.g., in a dispute)
  - copyright
  - patent
- Notary or lawyers testify that a letter or document existed on a certain date
- What about in the digital world?
- Document can be copied and modified endlessly
- It is trivial to change the date stamp on a computer file

Timestamping requirements

- The data must be timestamped regardless of the physical medium on which it is stored
- It must be impossible to change data without the change being apparent
- It must be impossible to change timestamp of a document with a different date and time from the present one

Arbitrated solution

- Alice transmits a copy of the document to Trent
- Trent records the date and time he received the document and retains a copy of the document for safekeep

Arbitrated solution

- Alice transmits a copy of the document to Trent
- Trent records the date and time he received the document and retains a copy of the document for safekeep
- Problems
  - Privacy: Alice has to send a copy of the document to Trent
  - Trent requires a huge database to store the copies, and the bandwidth requirement is high
  - The database may be attacked by an electromagnetic bomb
  - Trent may not be that honest, or he may collude with Alice?

An improved arbitrated solution (credit. B. Schneider's book — Applied Cryptography)

- $A \rightarrow T : hash(M)$
- $T \rightarrow A : [hash(M)||timestamp]_{sk(T)}$

An improved arbitrated solution (credit. B. Schneider's book — Applied Cryptography)

- $A \rightarrow T : hash(M)$
- $T \rightarrow A : [hash(M)||timestamp]_{sk(T)}$
- Seems to have solved almost everything
  - No need to reveal the document M
  - Even hash values are not required to be stored
  - Signature of Trent proves the time of the document

An improved arbitrated solution (credit. B. Schneider's book — Applied Cryptography)

- $A \rightarrow T : hash(M)$
- $T \rightarrow A : [hash(M)||timestamp]_{sk(T)}$
- Seems to have solved almost everything
  - No need to reveal the document M
  - Even hash values are not required to be stored
  - Signature of Trent proves the time of the document
- Trent may still be dishonest or collude with Alice

Linking protocol

- We may link Alice's timestamp with previously generated timestamp (probably for other people)
- This *partially* proves that Alice's timestamp is likely to be generated after someone else's timestamp

Linking protocol

- We may link Alice's timestamp with previously generated timestamp (probably for other people)
- This *partially* proves that Alice's timestamp is likely to be generated after someone else's timestamp
- Suppose $H_n = hash(M_A)$ is the timestamp for Alice, and the previous timestamp $H_{n-1} = hash(M_B)$ for Bob (i.e., Trent certified for Bob immediately before Alice)
- $A \to T : H_n$
- $T \to A : T_n = [n, Alice, H_n, t_n, Bob, H_{n-1}, T_{n-1}, L_n]_{sk(T)}$ where $L_n = hash(Bob, H_{n-1}, T_{n-1}, L_{n-1})$
- After Trent certifies someone after Alice, he sends Alice T_{n+1}.

Linking protocol

- We may link Alice's timestamp with previously generated timestamp (probably for other people)
- This *partially* proves that Alice's timestamp is likely to be generated after someone else's timestamp
- Suppose $H_n = hash(M_A)$ is the timestamp for Alice, and the previous timestamp $H_{n-1} = hash(M_B)$ for Bob (i.e., Trent certified for Bob immediately before Alice)
- $A \to T : H_n$
- $T \to A : T_n = [n, Alice, H_n, t_n, Bob, H_{n-1}, T_{n-1}, L_n]_{sk(T)}$ where $L_n = hash(Bob, H_{n-1}, T_{n-1}, L_{n-1})$
- After Trent certifies someone after Alice, he sends Alice T_{n+1}.
- If Someone challenges Alice, she may contact the originators of the previous and following documents.

Advantages of the linking protocol

- If Trent colludes with Alice on t_n (i.e., to certify it to an earlier or later date), most likely he needs to collude with the one before and after Alice.

Advantages of the linking protocol

- If Trent colludes with Alice on t_n (i.e., to certify it to an earlier or later date), most likely he needs to collude with the one before and after Alice.
- The same logic applies, and Trent needs to collude with almost every people linked on the chain (everyone after Alice, and someone before Alice)
- Note that $L_n = hash(Bob, H_{n-1}, T_{n-1}, L_{n-1})$ will be different if either t_{n-1} or L_{n-1} is modified.

To strengthen the linking protocol

- People die; Stamps get lost

To strengthen the linking protocol

- People die; Stamps get lost
- One possible way is to embed the previous 10 people's timestamps into Alice's, and then sending Alice the identities of the next 10 people.

To strengthen the linking protocol

- People die; Stamps get lost
- One possible way is to embed the previous 10 people's timestamps into Alice's, and then sending Alice the identities of the next 10 people.
- Another way is to publish the ongoing chain on a public web site. (So that the entire chain is verifiable by everyone)

Blockchain and Public Name services

- Namecoin is one of the earliest Altcoins
- Distributed DNS (domain $\rightarrow$ ip_addr)

Blockchain and Public Name services

- Namecoin is one of the earliest Altcoins
- Distributed DNS (domain $\rightarrow$ ip_addr)
- Benefits from the Blockchain architecture
  - tamper-resistent (cryptographic hash function)
  - availability (millions of copies on the Internet)
  - (probabilistic) distributed consensus

Blockchain and Public Name services

- Namecoin is one of the earliest Altcoins
- Distributed DNS (domain $\rightarrow$ ip_addr)
- Benefits from the Blockchain architecture
  - tamper-resistent (cryptographic hash function)
  - availability (millions of copies on the Internet)
  - (probabilistic) distributed consensus
- Let's start with the first successful Blockchain application as a crypto-currency

A centralized crypto-currency protocol

- If Jinan University issues a crypto-currency `JN-coin` for students

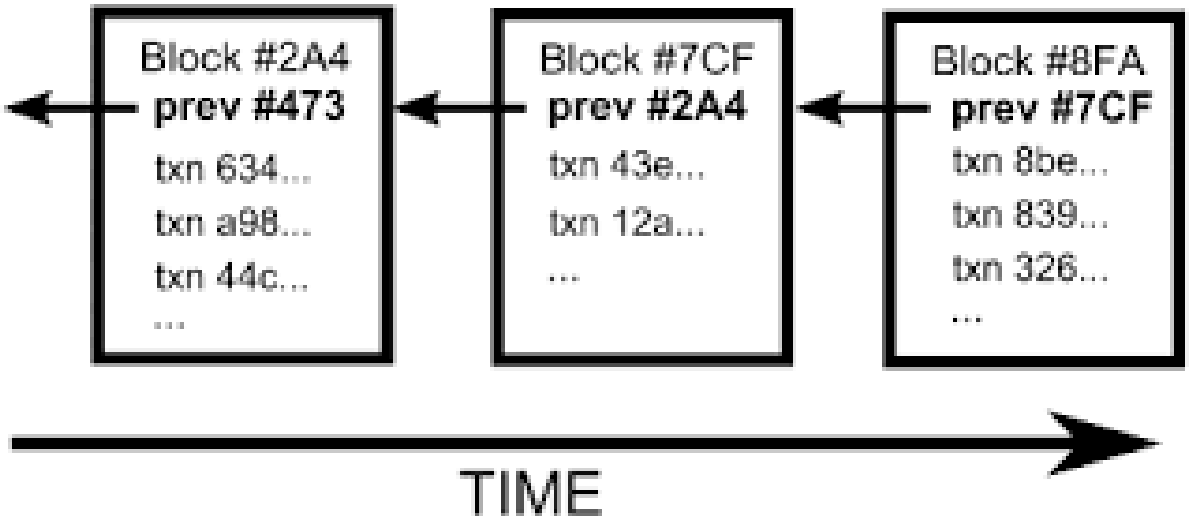
A centralized crypto-currency protocol

- If Jinan University issues a crypto-currency `JN-coin` for students
- The coin movement can be published on a centralized blockchain
  - $1\texttt{JN} = 1\texttt{RMB}$
  - Students buy credit from the university
  - Students pay each other using their `JN` credit

A centralized crypto-currency protocol

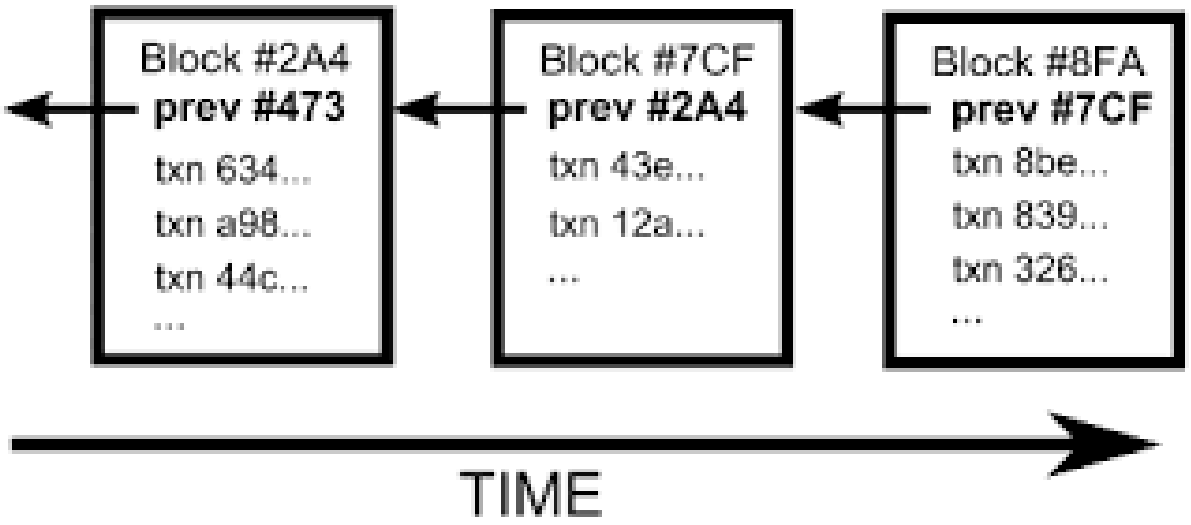
- If Jinan University issues a crypto-currency `JN-coin` for students
- The coin movement can be published on a centralized blockchain
  - $1\texttt{JN} = 1\texttt{RMB}$
  - Students buy credit from the university
  - Students pay each other using their `JN` credit
- If we crypto-lized the entire procedure
  - The University has its public key and private key pair
  - Let $pk(A)$, $pk(B)$ be Alice and Bob's public key
  - $sk(A)$, $sk(B)$ be Alice and Bob's private key
  - $TX_i = [\text{issue to } pk(A)\ 10\texttt{JN}]_{sk(U)}$
  - $TX_j = [pk(A) \text{ pays } pk(B)\ 5\texttt{JN}]_{sk(A)}$ if Alice pays Bob $5\texttt{JN}$
  - Every 10 seconds the university publish a new block including all transactions that have happened during the last 10 seconds

A centralized crypto-currency protocol



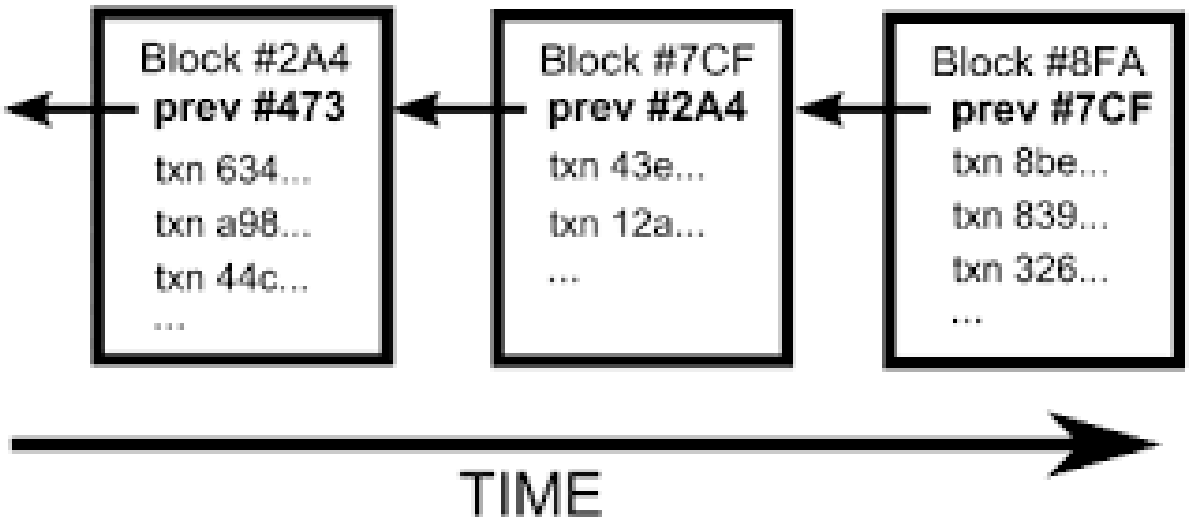
- It is viable to check which person (represented by his public key) owns how many coins at each moment of time
- People are anonymized (pseudonymized)

A centralized crypto-currency protocol



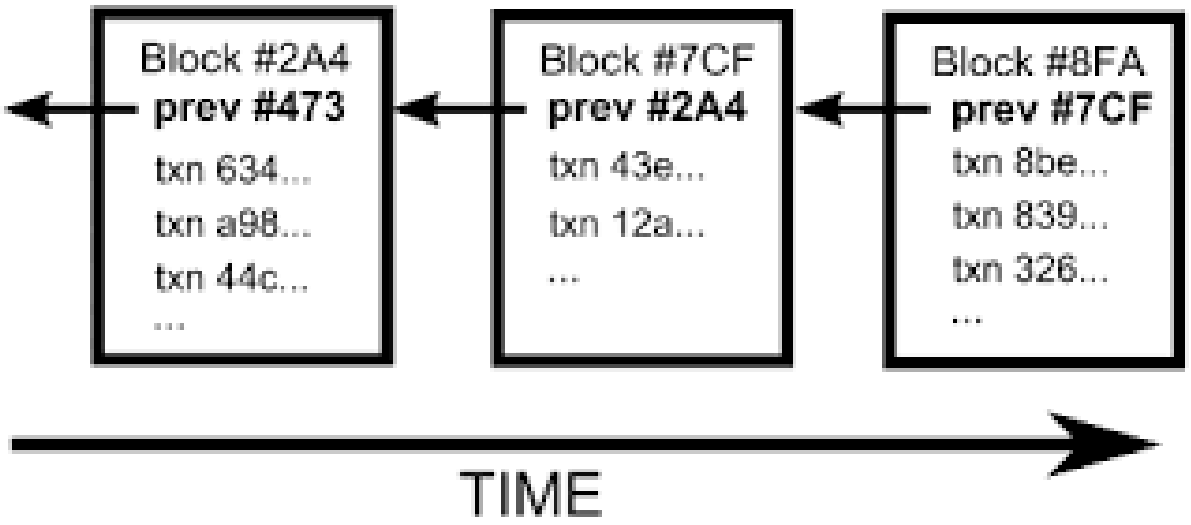
- Trusting the university is essential

A centralized crypto-currency protocol



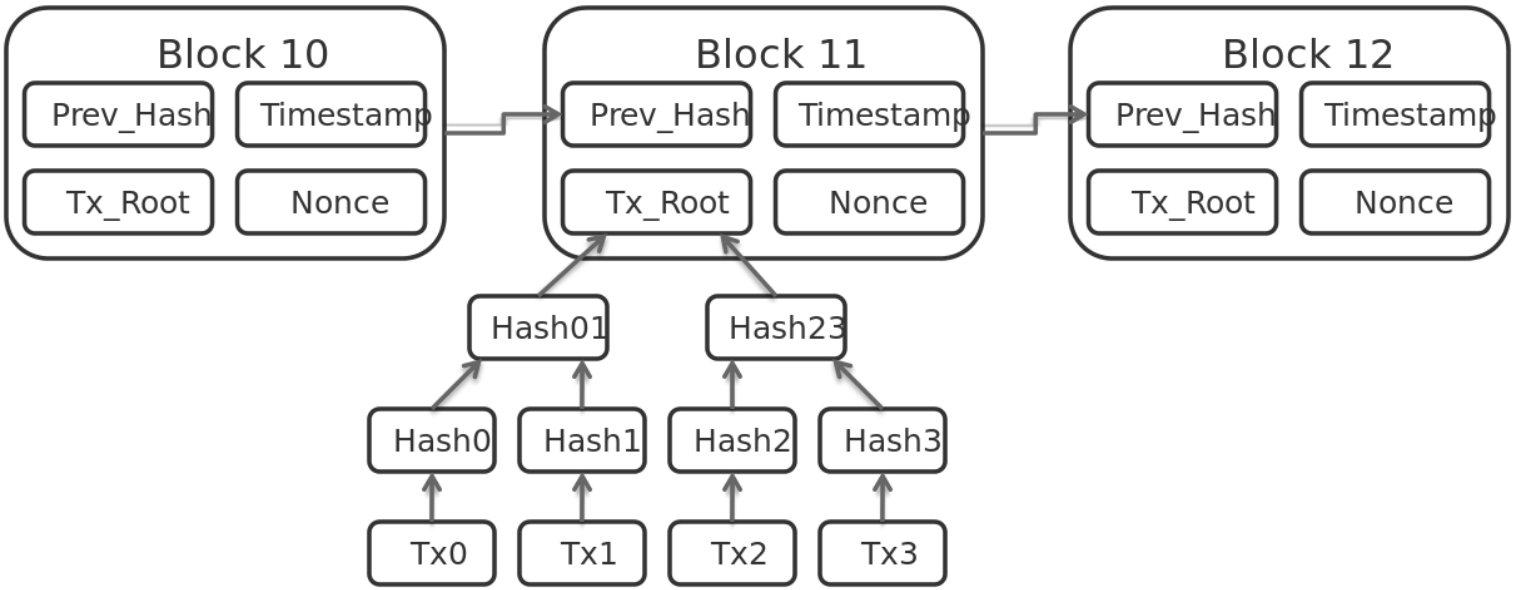
- Trusting the university is essential
- What if the university `JN-coin` server is down
- What if the university system admin is corrupted

A centralized crypto-currency protocol



- Trusting the university is essential
- What if the university `JN-coin` server is down
- What if the university system admin is corrupted
- Actually we have the same problem with Alipay, wechat-pay (and every centralized system)

Block organization



- Prev_Hash of Block n is $hash(\texttt{Block}_{n-1})$
- Timestamp records the time when the block is generated
- Merkle tree inside each block bundle transactions to Tx_Root

Bitcoin as a centralized crypto-currency

- Bitcoin: A Peer-to-Peer Eletronic Cash System (Satoshi Nakamoto, 2009)
- A distributed ledger made of Blocks
- Mining and Proof of Work (PoW)
- Distributed consensus (to avoid double spending)

Hashcash

- Hashcash — a denial of service counter-measure (A. Back, 2002)
- To introduce a cost to spammers by setting up rules for a client's mailbox
  - Given an email M, it needs to of the form $[M, I, nonce]$
  - I is the receiver's Id (email address)
  - needs to satisfy that $hash([M, I, nonce]) = 0^k\{0,1\}^{128-k}$ (let's say, md5 hash)
- For each recipient, the spammer needs to invest a tiny amount of computation time
- Based on the principle that the hash function is secure (pre-image resistant, and well distributed)

Proof of Work (PoW) Mining in Bitcoin

- Miners get a bonus by checking consistency of the existing chain and adding a new block to the chain (1M to 2M in size)
- The reward is 50 BTC in block #1 and halves every $210,000$ blocks (against inflation)

Proof of Work (PoW) Mining in Bitcoin

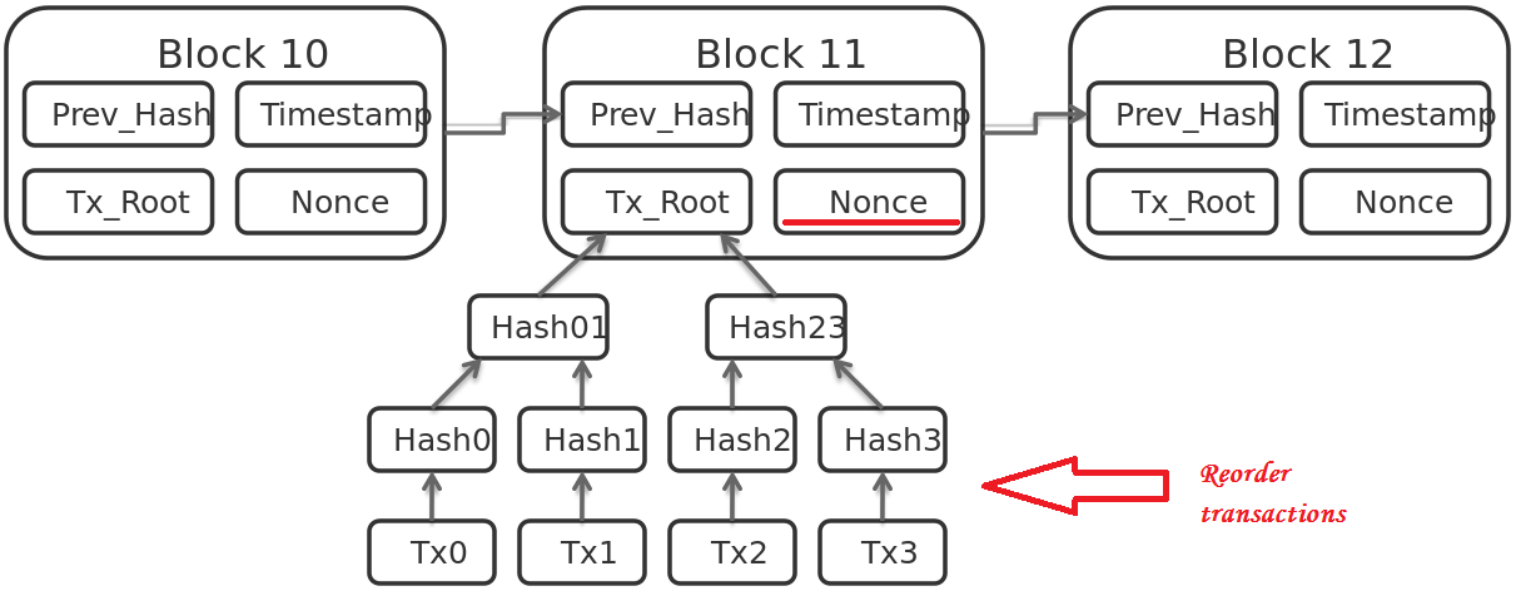
- Miners get a bonus by checking consistency of the existing chain and adding a new block to the chain (1M to 2M in size)
- The reward is 50 BTC in block #1 and halves every $210,000$ blocks (against inflation)
- By design blocks are mined on average every 10 minutes, 144 blocks are mined per day on average.

Proof of Work (PoW) Mining in Bitcoin

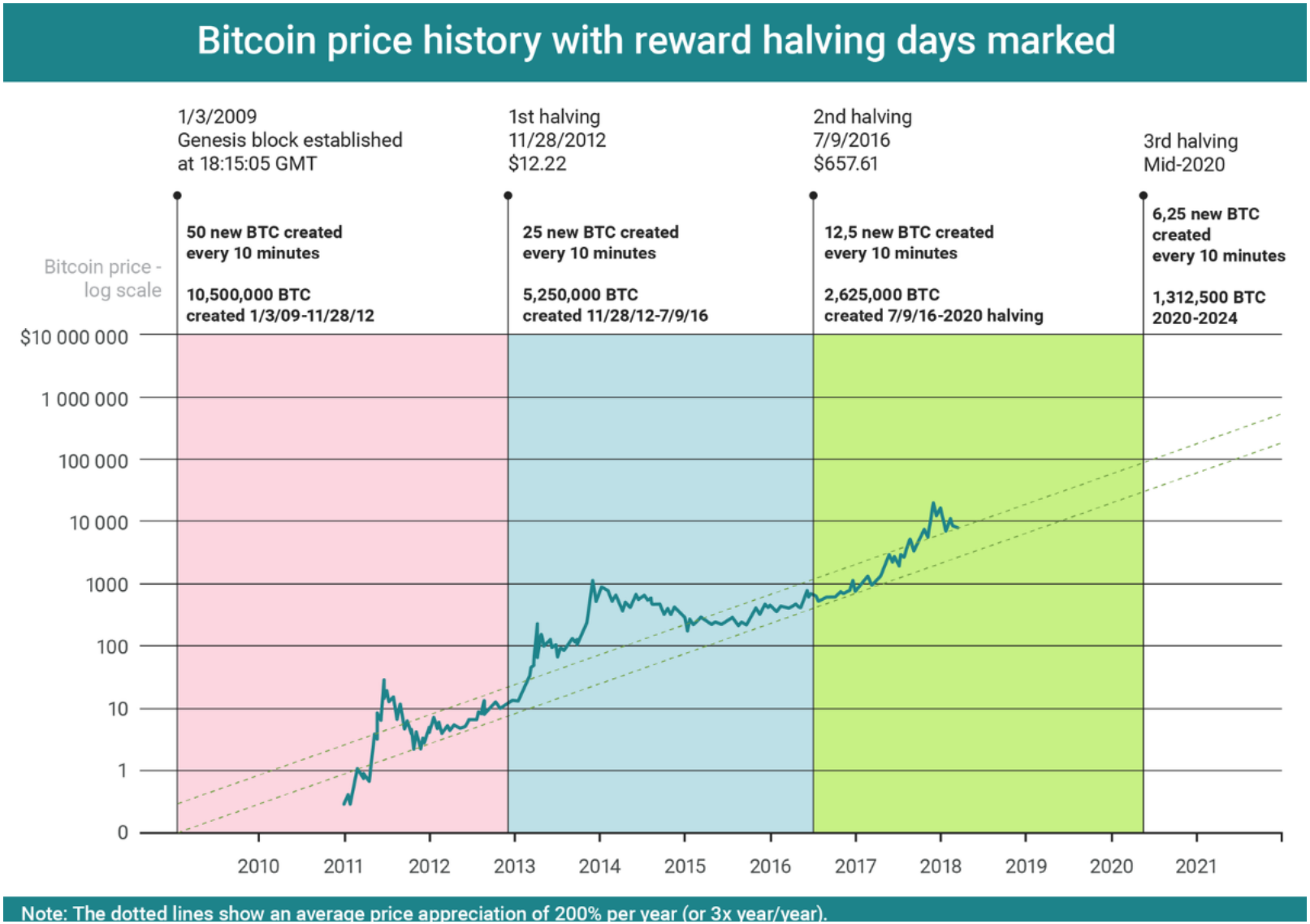
- Miners get a bonus by checking consistency of the existing chain and adding a new block to the chain (1M to 2M in size)
- The reward is 50 BTC in block #1 and halves every $210,000$ blocks (against inflation)
- By design blocks are mined on average every 10 minutes, 144 blocks are mined per day on average.
- The required k-bits of preceding 0's needs to maintain 10 minutes rates

Proof of Work (PoW) Mining in Bitcoin

- Miners get a bonus by checking consistency of the existing chain and adding a new block to the chain (1M to 2M in size)
- The reward is 50 BTC in block #1 and halves every 210,000 blocks (against inflation)
- By design blocks are mined on average every 10 minutes, 144 blocks are mined per day on average.
- The required k-bits of preceding 0's needs to maintain 10 minutes rates



Proof of Work (PoW) Mining in Bitcoin

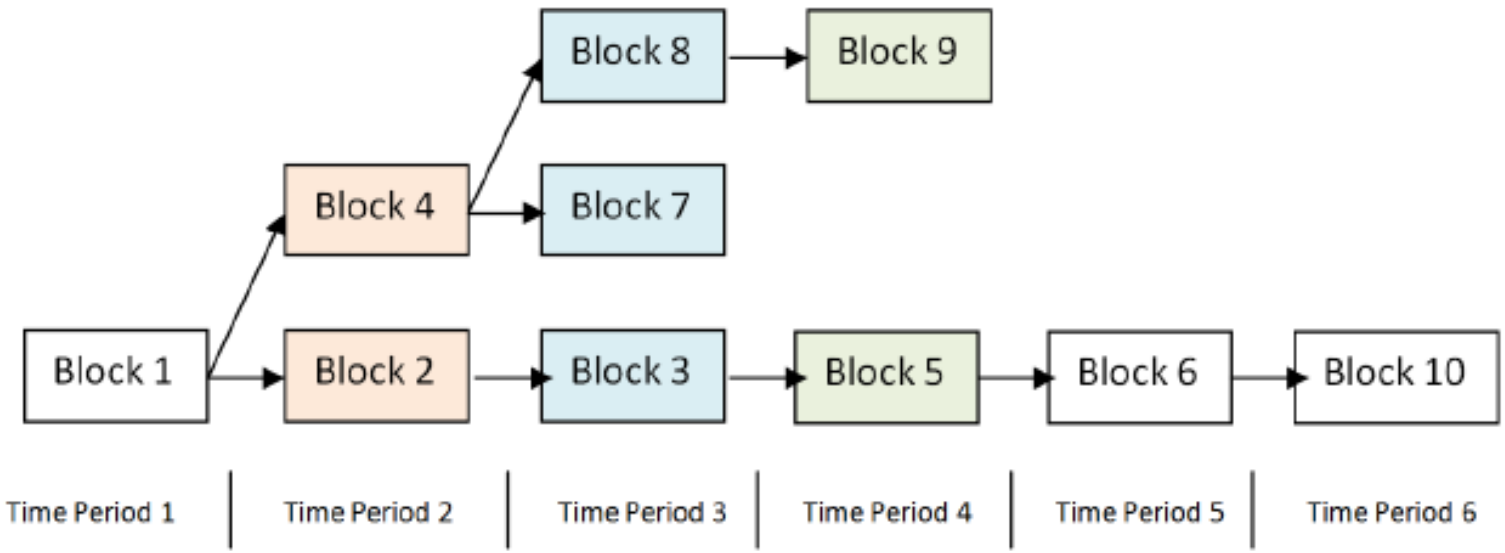


Proof of Work (PoW) Mining— Miner Competition

- Several miners successfully generate a next candidate block simultaneously within 10 minutes.
- Who wins the block is up to all miners in the Bitcoin network (the longest chain survives, over time)

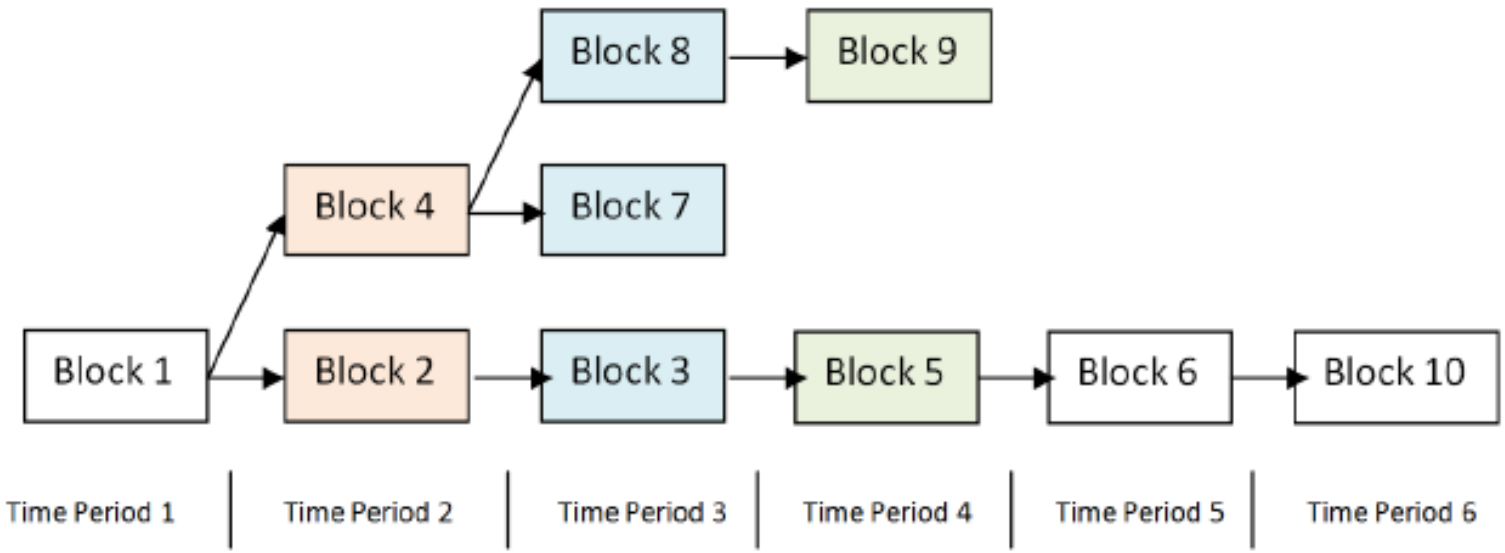
Proof of Work (PoW) Mining— Miner Competition

- Several miners successfully generate a next candidate block simultaneously within 10 minutes.
- Who wins the block is up to all miners in the Bitcoin network (the longest chain survives, over time)



Proof of Work (PoW) Mining— Miner Competition

- Several miners successfully generate a next candidate block simultaneously within 10 minutes.
- Who wins the block is up to all miners in the Bitcoin network (the longest chain survives, over time)



- Security guaranteed by the assumption that the majority of miners are honest

Possible double spending in Bitcoin

- Every coin is unique — traceable to where it is generated
- If you own a coin, it is either because you mined a block or someone paid you

Possible double spending in Bitcoin

- Every coin is unique — traceable to where it is generated
- If you own a coin, it is either because you mined a block or someone paid you
- Suppose Alice owns a coin P, and plan to pay P to Bob to buy some stuff

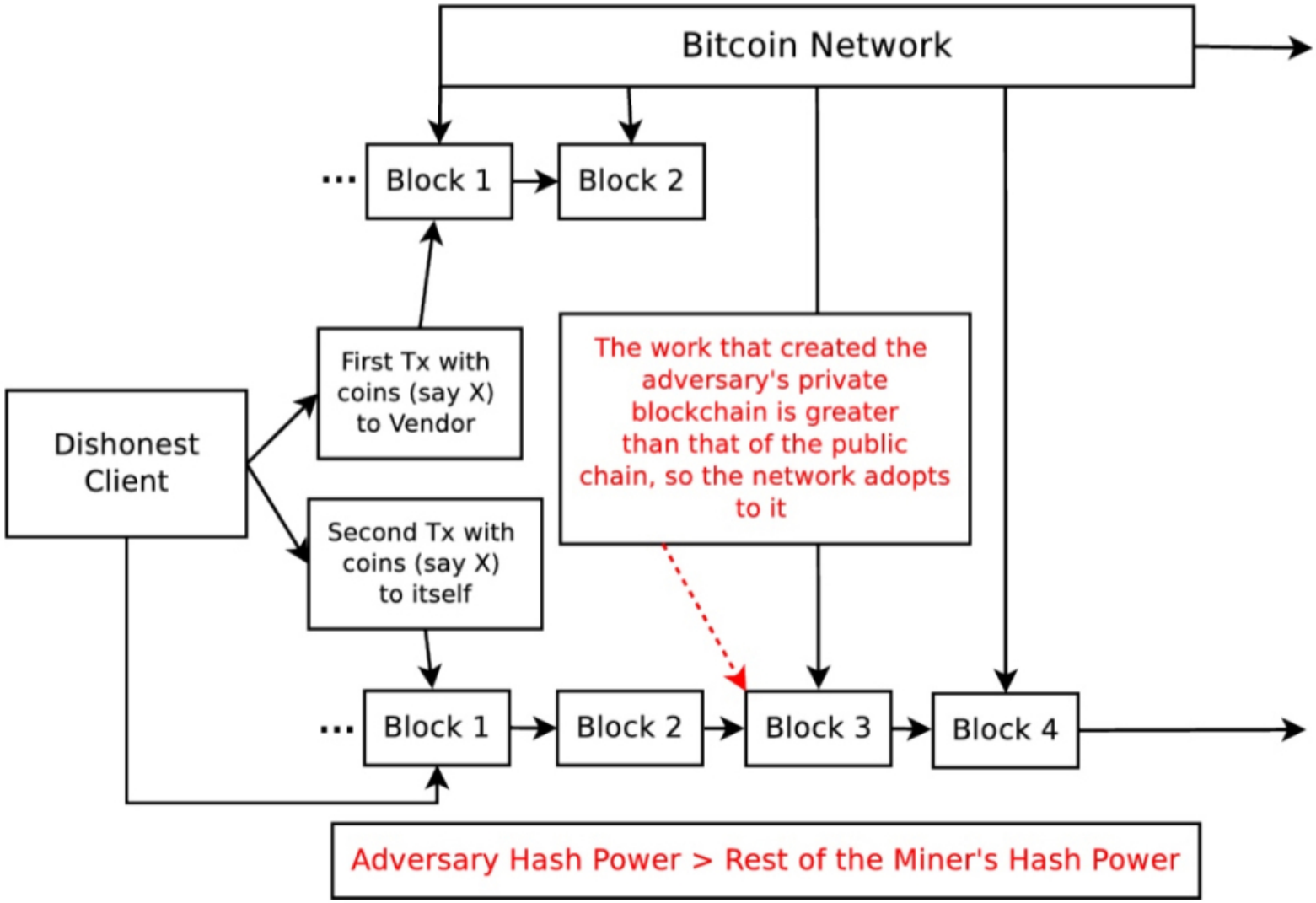
Possible double spending in Bitcoin

- Every coin is unique — traceable to where it is generated
- If you own a coin, it is either because you mined a block or someone paid you
- Suppose Alice owns a coin P, and plan to pay P to Bob to buy some stuff
- After $TX_A = ([pk(A) \rightarrow pk(B), P]_{sk(A)})$ is evidenced on a block, Bob believes that and delivers goods

Possible double spending in Bitcoin

- Every coin is unique — traceable to where it is generated
- If you own a coin, it is either because you mined a block or someone paid you
- Suppose Alice owns a coin P, and plan to pay P to Bob to buy some stuff
- After $TX_A = ([pk(A) \rightarrow pk(B), P]_{sk(A)})$ is evidenced on a block, Bob believes that and delivers goods
- Then Alice tries to generate a new block that retains her coin, and attaches it to the block immediately before the block that contains TX_A.
- Alice has a good chance to succeed if her hashing power is more than the rest of the Bitcoin network (why?)

Double spending attack in Bitcoin

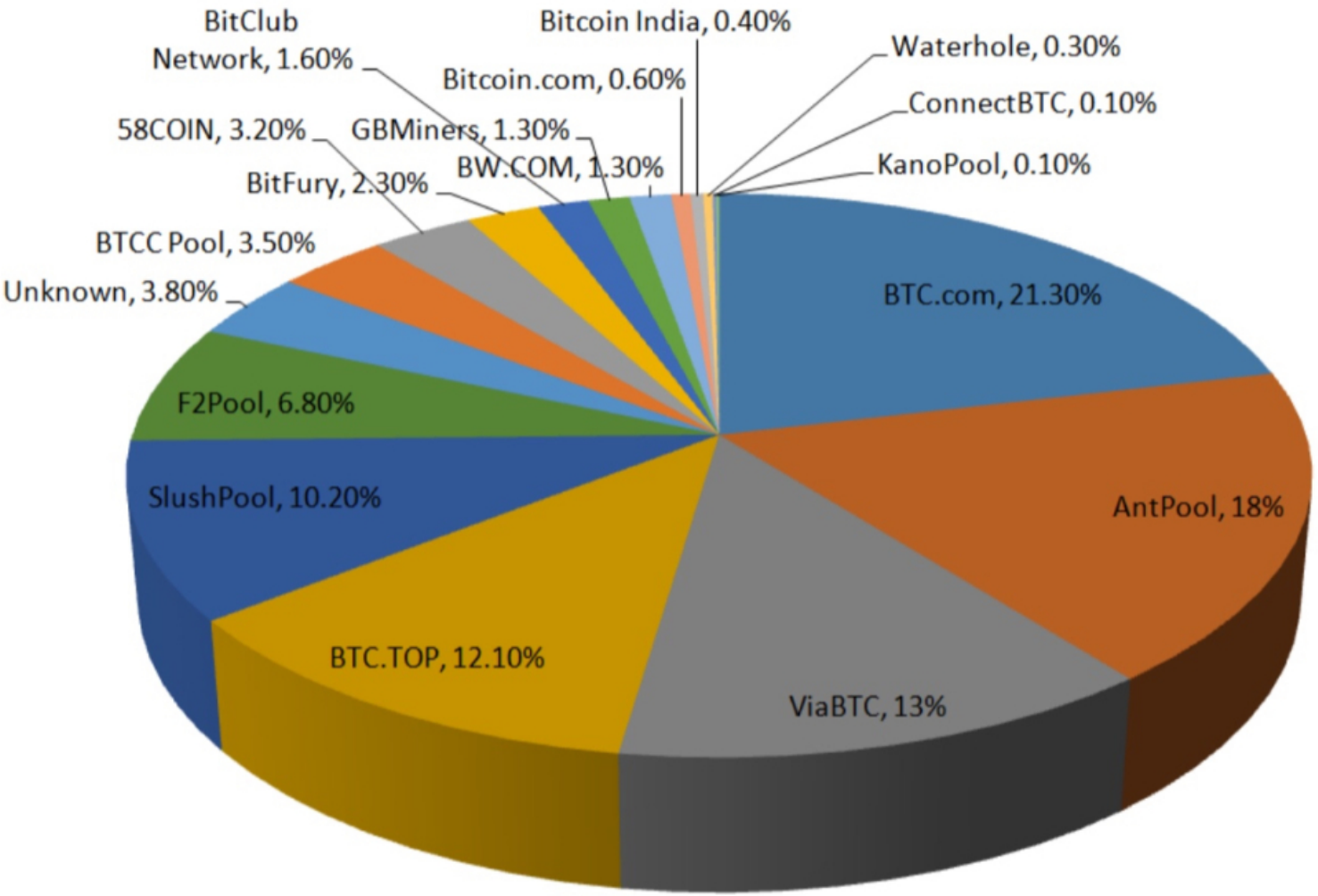


Other attacks in Bitcoin

- **Finney attack**: a form of double spending attack in which a dishonest client pre-mines a block (or a sequence of blocks) containing a transaction paying the coins to an address under their control.
- **Goldfinger attack**: if a miner (mining pool) can get to 51% of the overall hashrate, then all bets are off. This is known as a 51% attack (it enables you to double spend at will for example). Now the motivation is not to profit directly through Bitcoin, but instead to bring down the currency or network.
- **Block discarding / selfish mining**: In block discarding, a dishonest miner (or colluding set of miners) working in a pool withholds a block once found. They keep working on the private chain, and publish their mined blocks immediately before honest forks get back to the same length.
- **Bribery attacks, Wallet theft, DDoS to miners**

Mining pools in the Bitcoin network

- The market share of hashrate for mining pools as of December 2017.



Other cryptocurrency systems than Bitcoin

- Litecoin, Namecoin ...
- Ethereum (Crypto-currency plus smart contract)
- IOTA (Based on a Directed Acyclic Graph)

Other cryptocurrency systems than Bitcoin

- Litecoin, Namecoin ...
- Ethereum (Crypto-currency plus smart contract)
- IOTA (Based on a Directed Acyclic Graph)
- Some interesting places to check (price, exchange rates, market cap etc.)
  - `https://coinmarketcap.com/`
  - `https://www.coinbase.com/`

Top 100 Cryptocurrencies by Market Capitalization

Cryptocurrencies ▾ | Exchanges ▾ | Watchlist | USD ▾ | Next 100 → | View All

#	Name	Market Cap	Price	Volume (24h)	Circulating Supply	Change (24h)	Price Graph (7d)
1	Bitcoin	$60,762,890,529	$3,488.73	$4,900,331,896	17,416,912 BTC	-3.26%	
2	XRP	$12,639,509,054	$0.308831	$401,981,861	40,926,963,305 XRP *	-0.63%	
3	Ethereum	$9,483,071,106	$91.42	$1,655,144,473	103,725,395 ETH	-2.79%	
4	Stellar	$2,262,128,286	$0.118032	$98,315,870	19,165,371,791 XLM *	-3.93%	
5	Tether	$1,881,827,942	$1.01	$3,145,400,537	1,856,421,736 USDT *	-0.32%	
6	Bitcoin Cash	$1,855,246,091	$105.99	$68,797,089	17,503,913 BCH	-1.69%	
7	EOS	$1,741,449,091	$1.92	$857,349,601	906,245,118 EOS *	-3.26%	
8	Bitcoin SV	$1,621,506,908	$92.64	$57,653,631	17,503,611 BSV	-7.12%	
9	Litecoin	$1,477,317,823	$24.82	$417,556,570	59,519,217 LTC	-2.79%	
10	TRON	$885,486,196	$0.013367	$53,903,239	66,246,286,486 TRX *	-1.04%	

Summary of the day

- Timestamping services (proof of time and integrity)
- Arbitration and Linking
- Centralized control vs de-centralized control
- Block as a techniques used for cryptocurrencies
  - Block + Hash $\Rightarrow$ Integrity and tamper-resistance
  - PoW Mining
  - (Probabilistic) distributed consensus (double spending attack)
  - Altcoins