香港城市大學
City University of Hong Kong

CS 6290

Privacy-enhancing Technologies

Department of Computer Science

Slides credit in part from E. Ben-Sasson, D. Dziembowski, I. Eyal, R. Geambasu, and F. Greenspan, A. Judmayer, A. Juels, A. Miller, J. Poon, V. Shmatikov, D. Song, and F. Zhang

Lecture 1 – Introduction

Prof. Cong WANG

CS Department

City University of Hong Kong

When people talk about privacy…

1999

"Get over it"

2010

"No one cares"

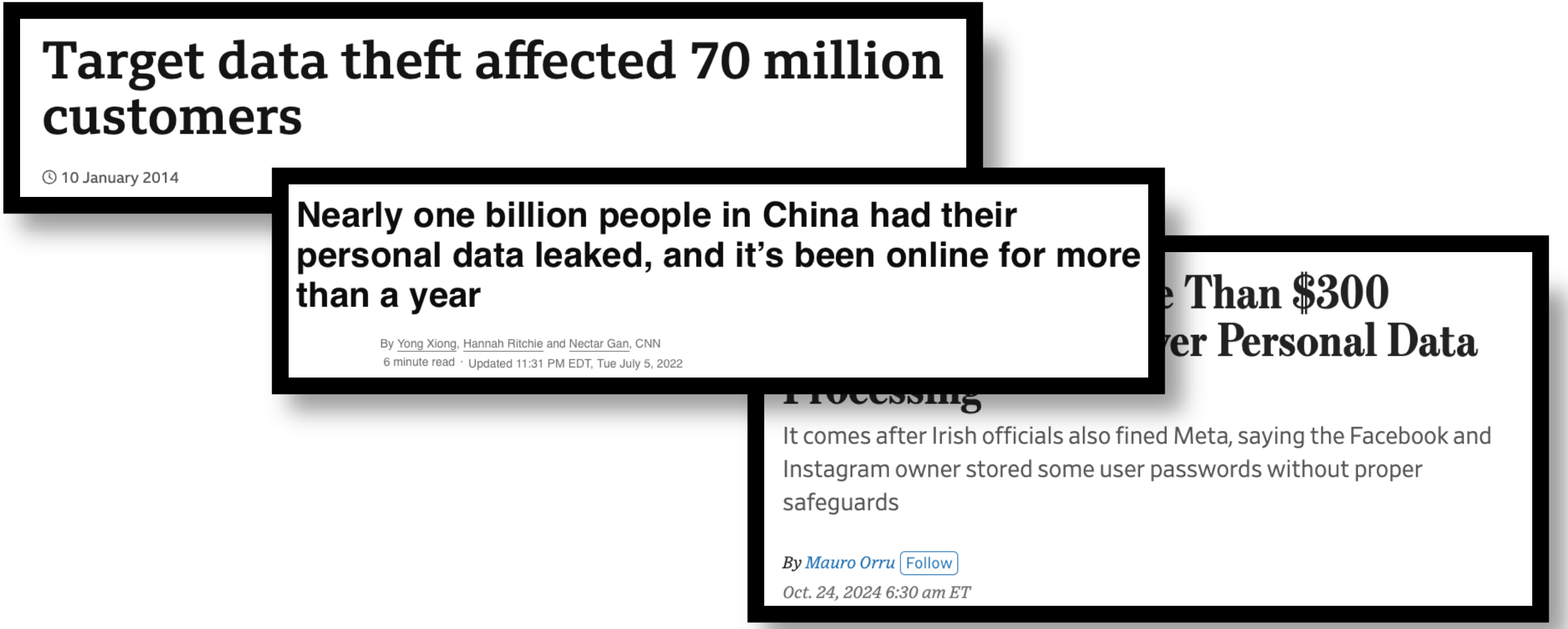
2009

"If you've done nothing wrong…"


Privacy
is King.
That's iPhone.

(Facebook’s F8 developer conference in 2019)

Rising privacy concerns



Major data breaches

Entity	Year	Records	Organization type	Method	Sources
Yahoo	2013	3,000,000,000	web	hacked	[640][641]
National Public Data	2024	2,900,000,000+ (claimed), including names, email addresses, phone numbers, Social Security numbers, and mailing addresses	data broker	hacked	[471]
Verifications.io (total leaks)	2019	2,000,000,000	online marketing	poor security	[620]
First American Corporation	2019	885,000,000	financial	poor security	[343]
Verifications.io (first leak)	2019	809,000,000	online marketing	poor security	[619]
Collection No. 1	2019	773,000,000	various	compilation of multiple data breaches	[256]
Ticketmaster	2024	560,000,000	ticket distribution	hacked third party service	[579][580]
Facebook	2019	540,000,000	social network	poor security	[333][334]
Marriott International	2018	500,000,000	hotel/casino	hacked	[444]
Yahoo	2014	500,000,000	web	hacked	[642][643][644] [645][646]
Friend Finder Network	2016	412,214,295	web	poor security / hacked	[347][348]
Myspace	2016	360,000,000+, including usernames, passwords email addresses	social network	poor security/account recovery	[467][468][469]

https://en.wikipedia.org/wiki/List_of_data_breaches

Reasons?

Big data and machine learning technologies **push away** from privacy principles.

E.g., Foundational Fair Information Practices (FIPs):

- Collection limitation ↘
- Data quality ↘
- Purpose specification ↘
- Use limitation ↘
- Security ↘
- Openness/notice ↘
- Individual participation ↘
- Accountability ↘

Privacy-enhancing technologies to answer:

- How can we guarantee that the collected user data are not misused and privacy policies not violated?
- How can we protect user privacy while simultaneously allowing effective data sharing and utilization?
- When the servers are not fully trusted, how can we still provide desirable services to users and respect their privacy?

Teaching team

- Instructor:
  - Prof. **Cong WANG**
  - Tel(O): +852 3442 2010
  - Email: congwang@cityu.edu.hk
- Teaching assistant:
  - Dr. **Yufei CHEN**
  - Email: yufeichen8@cityu.edu.hk

Teaching materials

- Weekly lecture notes, tutorials, reading materials
  - usually provided before class
- No required textbook; several recommended books:
  - Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder
  - Introduction to Modern Cryptography, by Jonathan Katz and Yehuda Lindell
  - The Algorithmic Foundations of Differential Privacy, by Cynthia Dwork and Aaron Roth
  - A Pragmatic Introduction to Secure Multi-Party Computation, by David Evans, Vladimir Kolesnikov, and Mike Rosulek

Teaching pattern

- Lecture and Tutorials (3 hours per week)
  - Information sessions
  - Discussions on commonly asked question
  - Discussions based on lecture materials/assigned readings

Assessment

- 60% Continuous Assessment:
  - 30% assignments
    - **3 programming-based assignments** to be submitted in total
    - Per assignment: implementation code + experiment report
  - 30 % group project + oral presentation
    - **3 students** per group
    - System projects, theoretical projects, or survey projects
- 40% Final Examination
- Plagiarism will not be tolerated

Announcements and Q&A

- You have to check Canvas and emails!
  - Announcements, material, assignments, etc.
- Ask questions:
  - Use canvas->discussions
    - Serve like a public bulletin so everyone can see the discussions
    - TA and I will check the discussions
  - Email to me or TA
    - Use CityU mailbox, i.e., congwang@cityu.edu.hk
    - Do **NOT** use Canvas->mailbox (inbox)!
      - Mails could get lost…

Tentative course overview

- An introduction to four advanced topics and their respective privacy-enhancing technologies, including:
  - **Part I: Cryptocurrency and blockchain systems**
    - Bitcoin and how it works, smart contracts, dApps, and DeFi
  - **Part II: Confidential computing technologies**
    - Focus on zero-knowledge proof (ZKP), secure multi-party computation (MPC), and secure hardware
  - **Part III: Data privacy**
    - Data attacks, k-anonymization, differential privacy
  - **Part IV: Privacy in the wild**
    - Online tracking and countermeasures, private machine learning, and other trending technologies

Tentative course overview

- An introduction to four advanced topics and their respective privacy-enhancing technologies, including:
  - **Part I: Cryptocurrency and blockchain systems**
    - Bitcoin and how it works, smart contracts, dApps, and DeFi
  - **Part II: Confidential computing technologies**
    - Focus on zero-knowledge proof (ZKP), secure multi-party computation (MPC), and secure hardware
  - **Part III: Data privacy**
    - Data attacks, k-anonymization, differential privacy
  - **Part IV: Privacy in the wild**
    - Online tracking and countermeasures, private machine learning, and other trending technologies

The rise of cryptocurrencies

Bitcoin Price (USD) – Source : coinbase.com

- Bitcoin sparked research into multiple challenging areas and applications
  - More than 2000+ cryptocurrency startups

Bitcoin use today

Online sites

1. Get Bitcoin

There are several ways to get Bitcoins, but the easiest is to exchange them for currency at your bank or a Bitcoin exchange. You can also buy Bitcoins from friends, accept them as payment for goods or services, or generate new Bitcoins through a process called "mining."

Sign Up at Coinbase.com

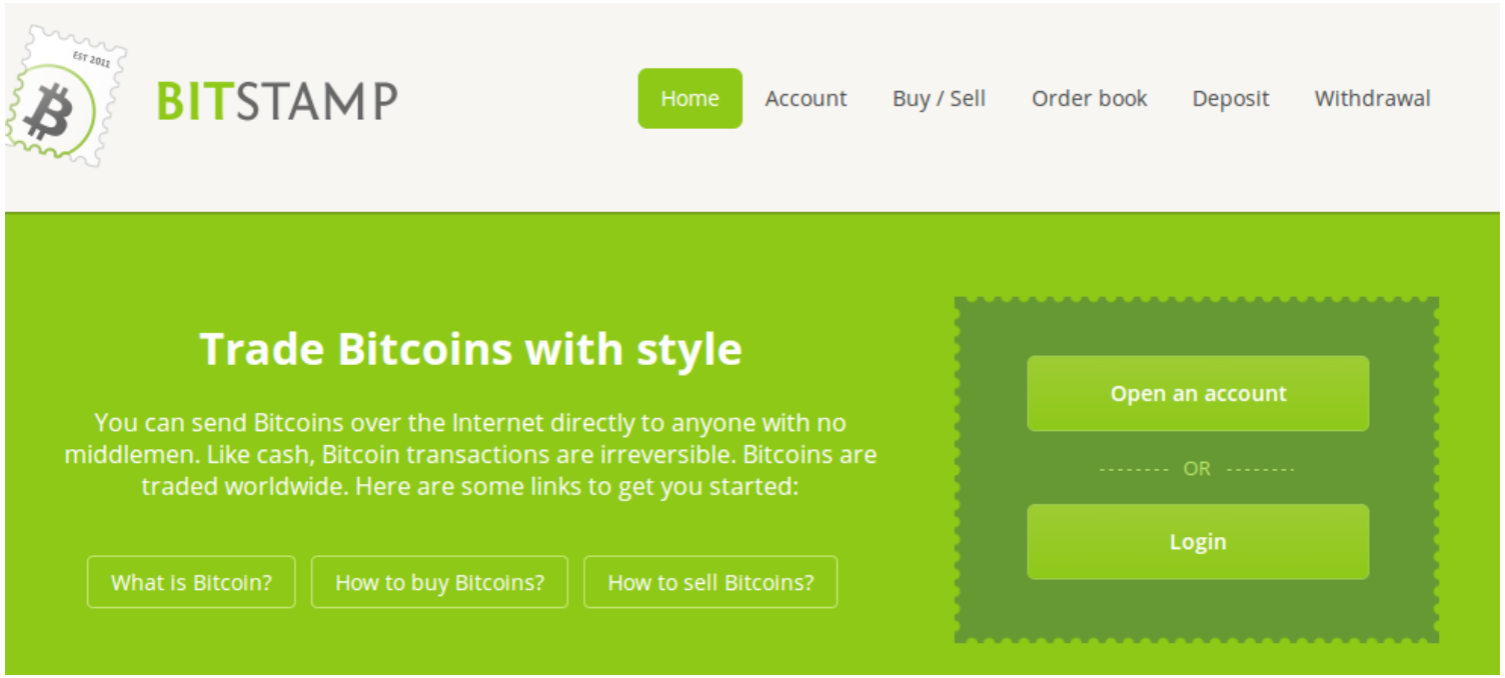
2. Shop Overstock.com

You can now pay for all your favorite products on Overstock.com using Bitcoins! As the first major retailer to accept Bitcoins, Overstock.com is expanding the possibilities of Bitcoin purchases by offering thousands of products to the Bitcoin community.

ATMs

Sell your Bitcoins!

Bitcoin exchanges



Tyler Moore and Nicolas Christin. Beware the middleman: Empirical analysis of Bitcoin-exchange risk. Financial Cryptography, 2013.

Bitcoins has its dark sides

Tor + Bitcoin = End-to-end anonymity for commercial transactions

Bitcoins has its dark sides

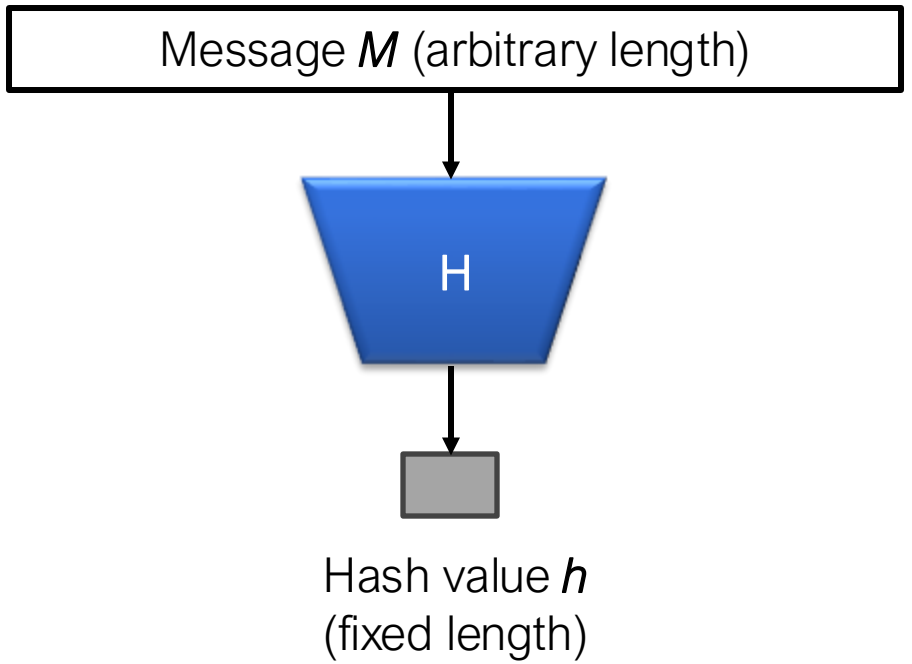
Bitcoin has stimulated

- Money laundering
- Theft of Bitcoin wallets
- Illicit marketplaces (Silk Road)
- Rogue mining
  - E.g., ZeroAccess botnet
- Ransomware

How bitcoin works?

Background: Hash function

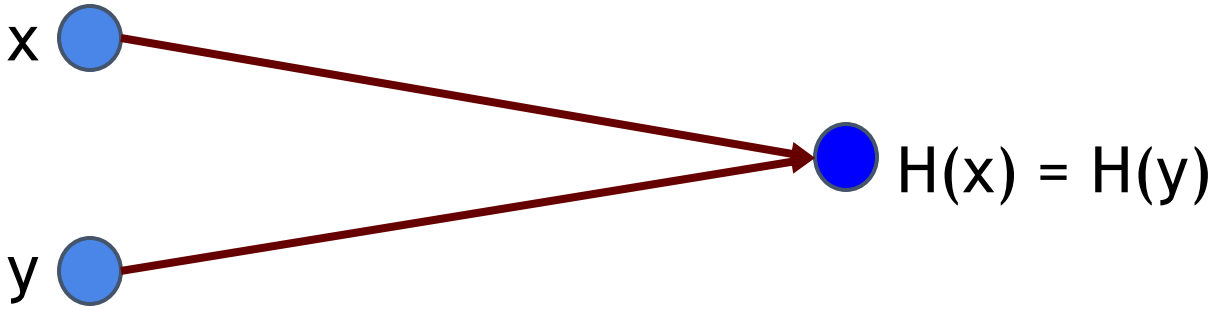
- Map data of arbitrary size to data of a fixed size
  - E.g., takes any string as input and generates fixed-size output (**we'll use 256 bits**)
  - Efficiently computable
- Security properties:
  - Collision-resistant
  - Hiding
  - Puzzle-friendly



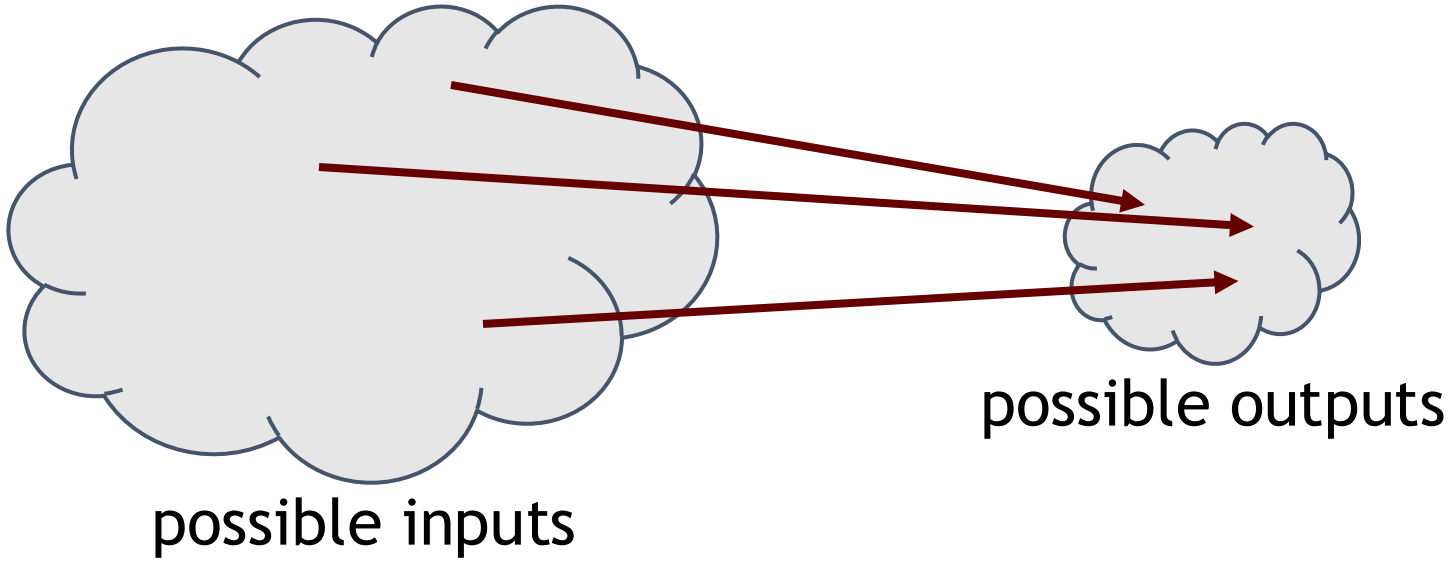
Property 1: Collision-resistant

- Difficult to find **x** and **y** such that

  x != **y** and H(**x**)=H(**y**)



Collisions do exist ...



… but can anyone find them?

How to find a collision

- A cryptographic hash is expected to have a collision resistance strength of **n/2** bits (lower due to birthday paradox)
  - If you can randomly generate a sequence 2^{130} inputs

    99.97% chance that two of them will collide

- **This works no matter what H is …**
- **But it takes too long to matter**

Is there a faster way to find collisions?

For some possible H's, yes.

For others, we don't know of one.

No H has been proven collision-free.

Application: Hash as message digest

- If we know H(x) = H(y),

  it's safe to assume that x = y.

- To recognize a file that we saw before, just remember its hash.

- Useful because the hash is small for network transmission.

Property 2: Hiding

- We want something like this:

  Given H(x), it is infeasible to find x.

H(“heads”)

H(“tails”)

easy to find x!

Hiding property

- If **r** is sampled uniformly at random from a high-entropy domain, then given an outcome

  H(**r** | | **x**), it is infeasible to find **x**.

Note: r is a secret and is NOT given here.

No particular value is chosen with more than negligible probability.

Commitment API

- (*com, key*) := commit(*msg*)
- *match* := verify(*com, key, msg*)

- Security properties:
  - Hiding: Given *com*, infeasible to find *msg*.
  - Binding: Infeasible to find ***msg != msg'*** such that

    verify(commit(*msg*), *msg'*) == true

Concrete construction sample

- commit(*msg)* := (H(*key* |*msg*), H(*key*))

  where *key* is a random 256-bit value
- verify(*com, key, msg*) := (H(*key* |*msg*) == *com*)

- Security properties:
  - Hiding: Given H(*key* |*msg*), infeasible to find *msg*.
  - Binding: Infeasible to find *msg != msg'* such that

    H(*key* |*msg)* == H(*key* | *msg'*)

Property 3: Puzzle-friendly

Puzzle-friendly:

- For every possible output value y,

if k is chosen uniformly at random and public to all,

then it is infeasible to find x such that H(k | x) = y.

Application: Search puzzle

- Given a "puzzle ID" *k*,
  and a target set *Y*:

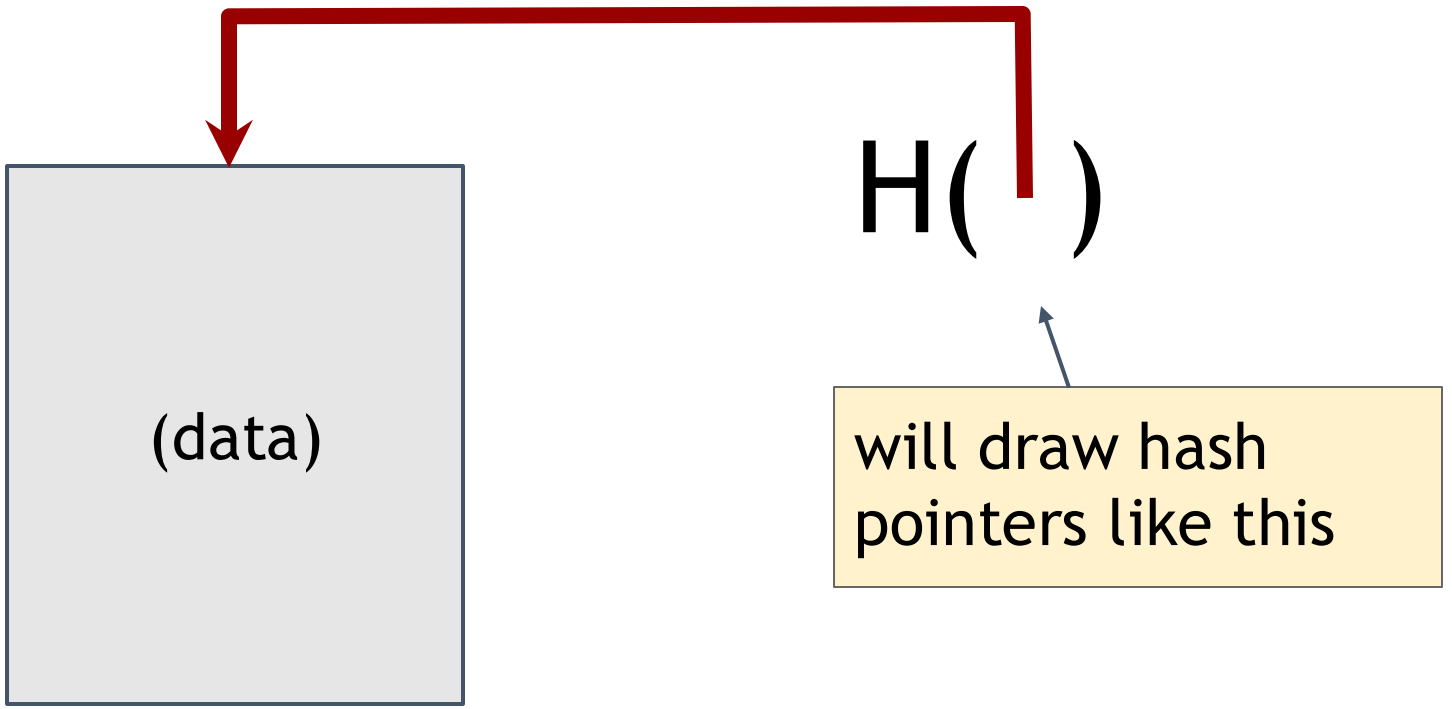
Try to find a "solution" *x* such that
$H(k \mid x) \in Y$.

- Puzzle-friendly property implies that no solving strategy is much better than trying random values of *x*.

Hash pointers

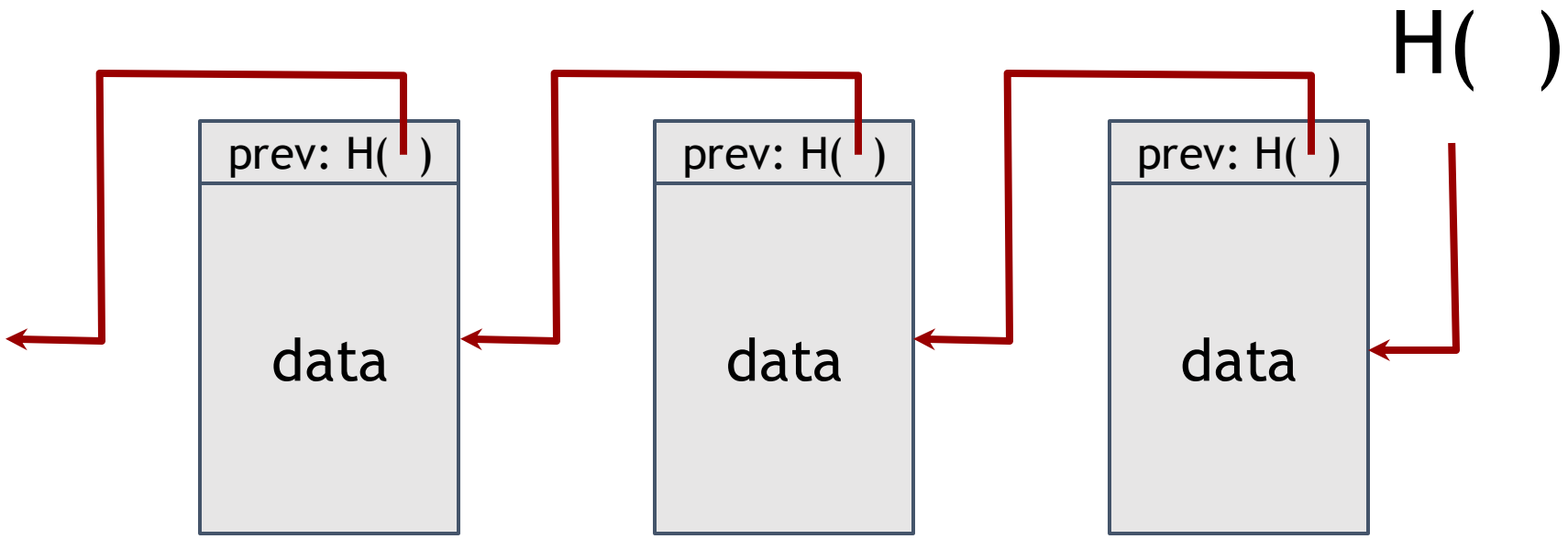
- Hash pointer is:
  - pointer to where some info is stored, and
  - (cryptographic) hash of the info

- If we have a hash pointer, we can
  - ask to get the info back, and
  - verify that it hasn't changed


H( )
(data)
will draw hash pointers like this

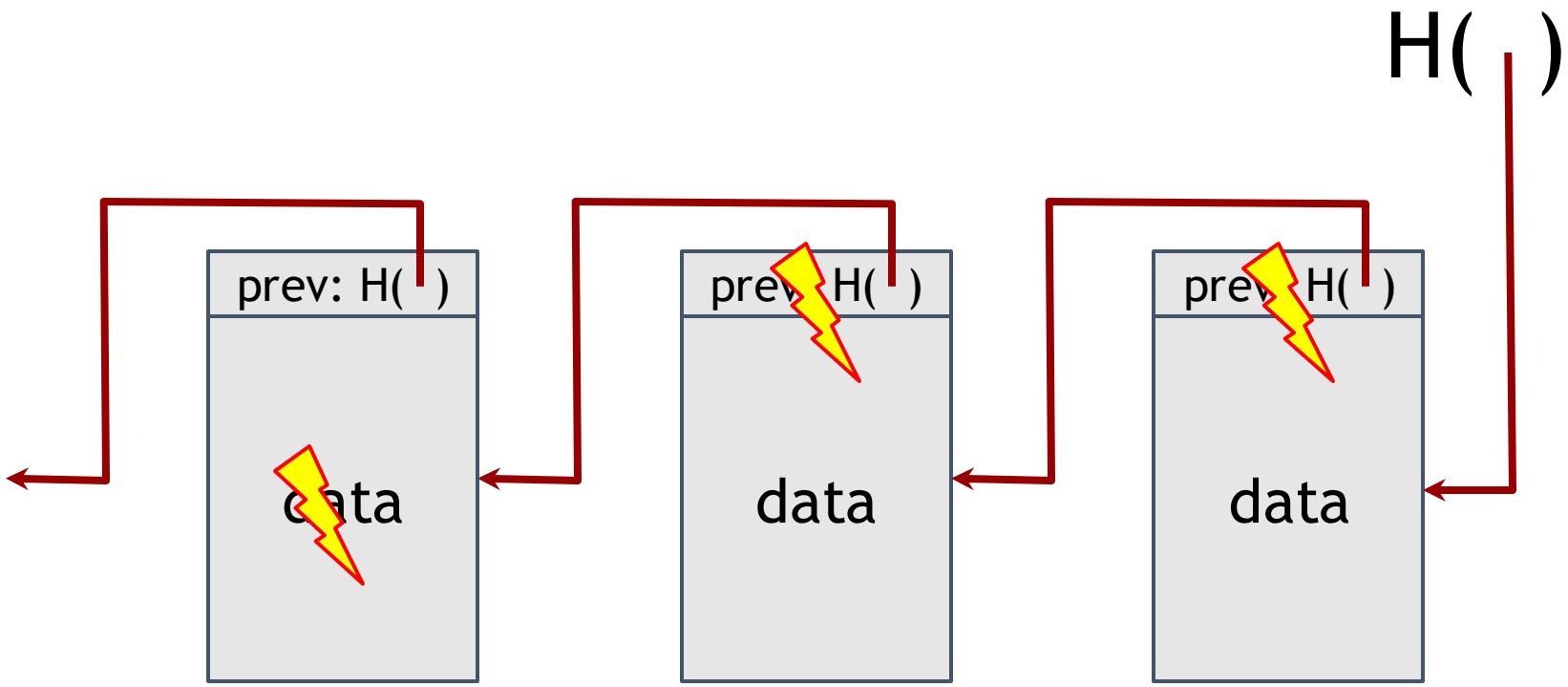
“Blockchain”

- key idea: **build data structures with hash pointers**
- linked list with hash pointers = “block chain”



use case: tamper-evident log

Detecting tampering



use case: tamper-evident log

Background: Digital signatures

- What we want from signatures:
  - Only you can sign, but anyone can verify
  - Signature is tied to a particular document

(*sk*, *pk*) := generateKeys(keysize)

sk: secret signing key

pk: public verification key

sig := sign(*sk*, message)

isValid := verify(*pk*, message, sig)

API for digital signatures

Requirements for signatures

- “valid signatures verify”
  - verify(pk, message, sign(sk, message)) == true

- “can’t forge signatures”

  adversary who:

  knows pk

  gets to see signatures on messages of his choice

  can’t produce a verifiable signature on another message

Useful trick: public key == an identity

if you see sig such that verify(**pk**, msg, sig)==true,
think of it as

pk says, “[msg]”.

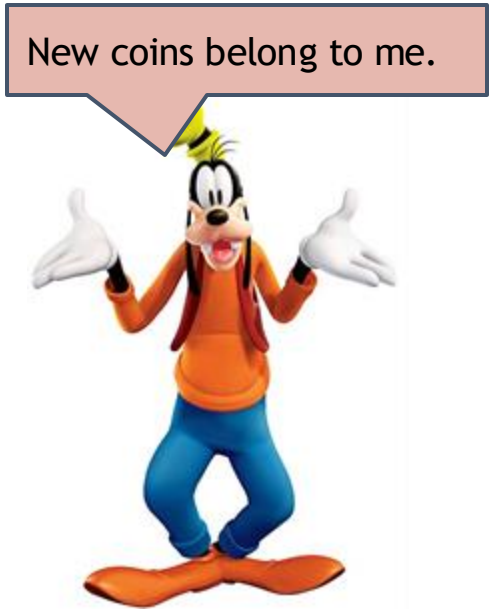
to “speak for” **pk**, you must know matching secret key **sk**

GoofyCoin

Goofy can create new coins

signed by pk_{Goofy}

CreateCoin [uniqueCoinID]



A coin's owner can spend it.

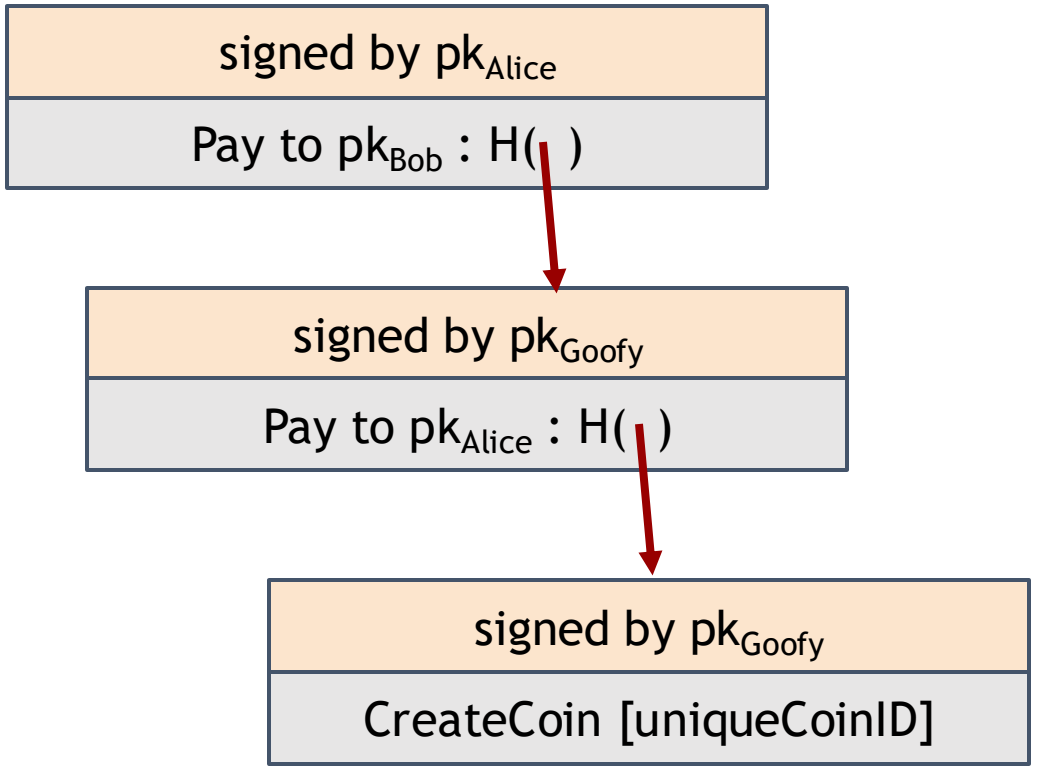
signed by pk_{Goofy}

Pay to pk_{Alice} : H( )

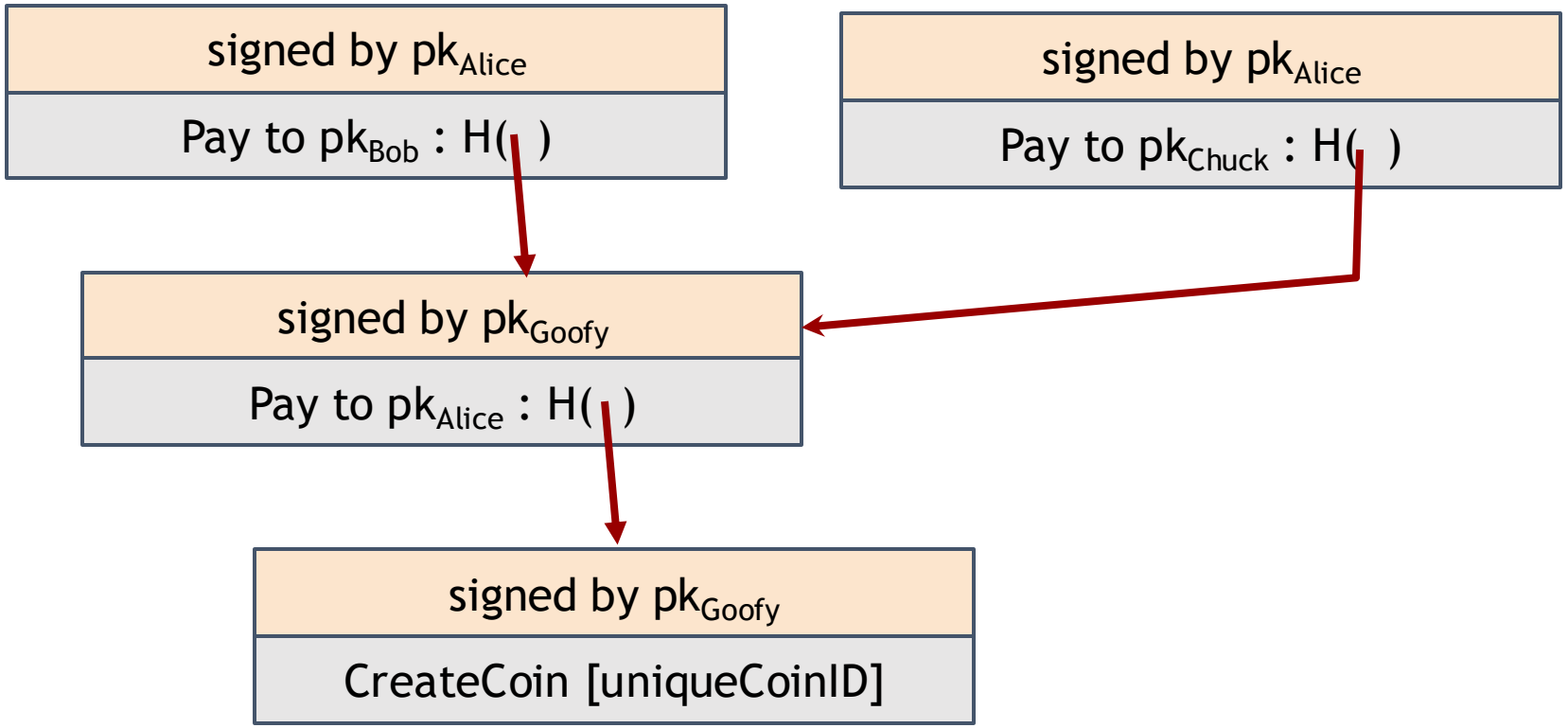
signed by pk_{Goofy}

CreateCoin [uniqueCoinID]

The recipient can pass on the coin again.



double-spending attack



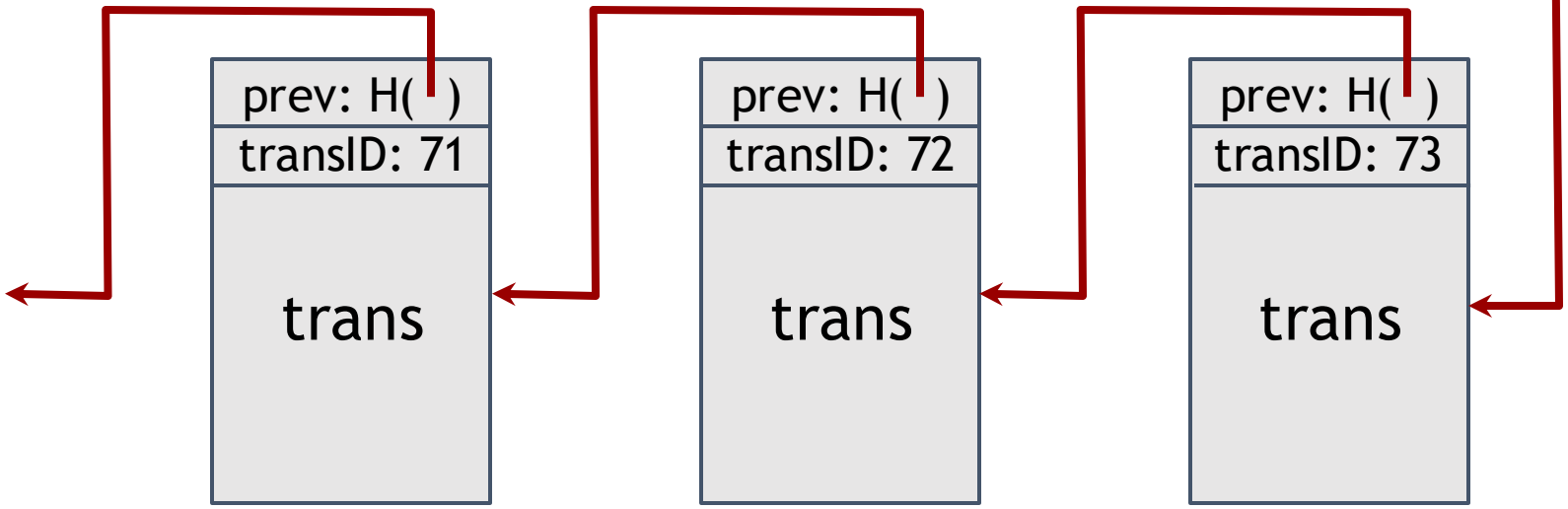
double-spending attack

the main design challenge in digital currency

ScroogeCoin

Scrooge publishes a history of all transactions
(a block chain, signed by Scrooge)

H( )



optimization: put multiple transactions in the same block

CreateCoins transaction creates new coins

transID: 73	type:CreateCoins	
coins created		
num	*value*	*recipient*
0	3.2	0x…
1	1.4	0x…
2	7.1	0x…

coinID 73(0)

coinID 73(1)

coinID 73(2)

PayCoins transaction consumes (and destroys) some coins, and creates new coins of the same total value

transID: 73 type:PayCoins		
consumed coinIDs: 68(1), 42(0), 72(3)		
coins created		
num	*value*	*recipient*
0	3.2	0x…
1	1.4	0x…
2	7.1	0x…
signatures		

Valid if:
-- consumed coins valid,
-- not already consumed,
-- total value out = total value in, and
-- signed by owners of all consumed coins

What are the problems with ScroogeCoin?

Can we use ScroogeCoin in real life?

Can we use ScroogeCoin in real life?

"How to Time-Stamp a Digital Document", Haber and Stornetta, 1991

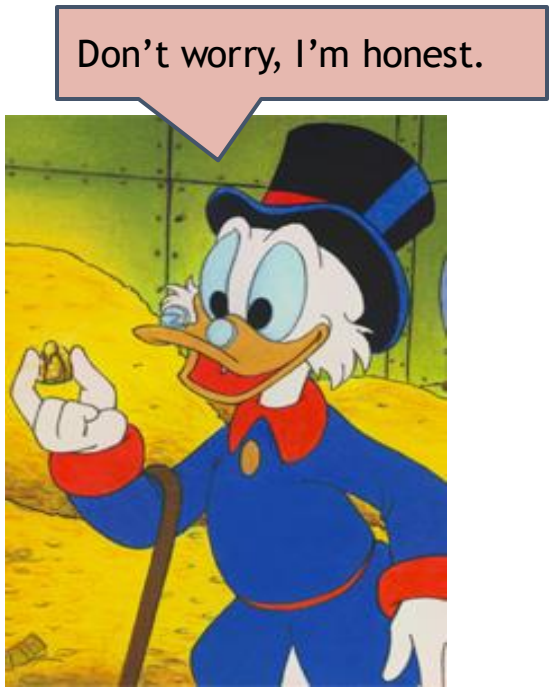
Surety: Linked-timestamping service, 1995

Publishes hash of the tail on New York Times, Lost and Found section


helped Southeast Asian refugees.

NOTICES & LOST AND FOUND
(5100-5102)

Universal Registry Entries:
Zone 2 -
HJUTngMYGtltH3hRqdcuh2sUS
pxZW29Znmlm14n+cqgIkRN/
seRf9WzBSL13Cks6UZ5TUg==
Zone 3 -
AZ/v3O2W/WRjiHG/QpXR4Mj
RNGUg+99b1R9lpzuHxbkE+OcK
WOYGrALk/a3rbZseDgiyBQ==
These base64-encoded values represent the combined fingerprints of all digital records notarized by Surety between 20170125Z - 20170131Z.
www.surety.com 239-436-2790



Crucial question:

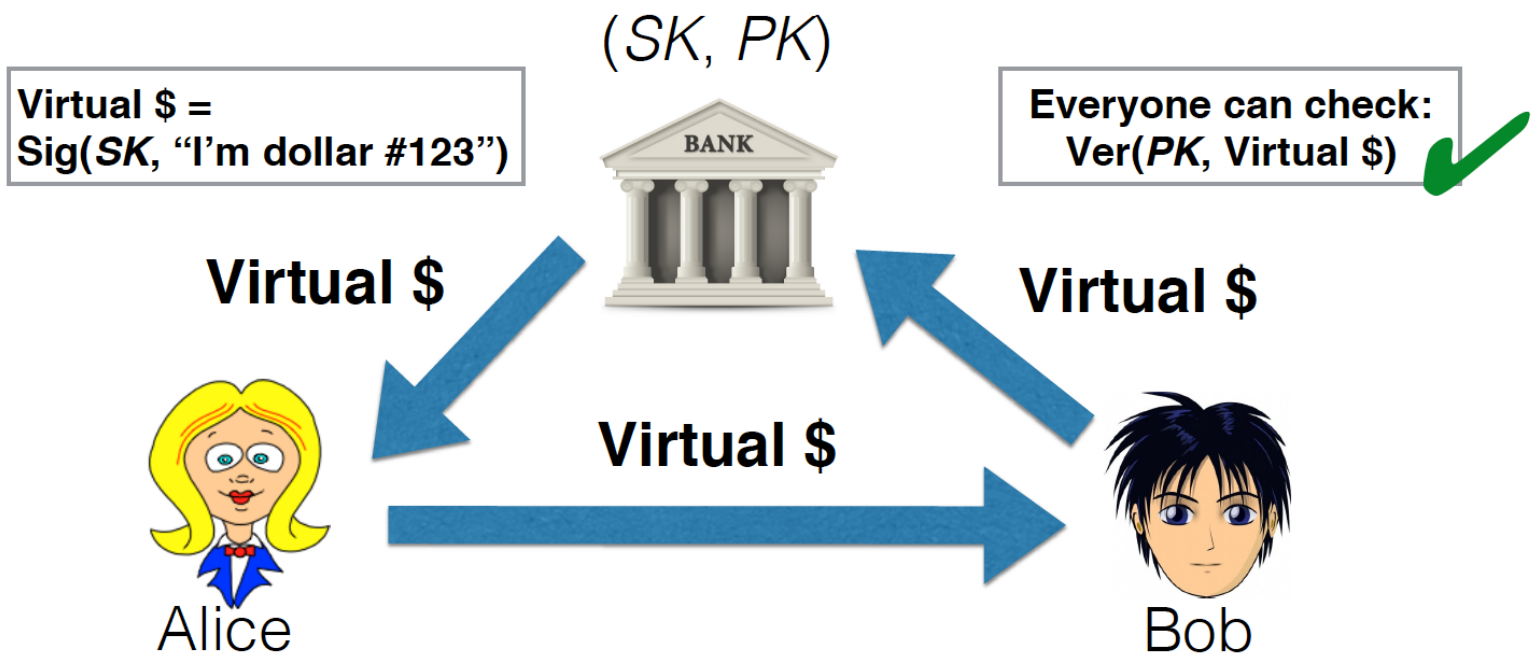
Can we descroogify the currency, and operate without any central, trusted party?

Bitcoin: A Peer to Peer Electronic Cash System

Historical backdrop

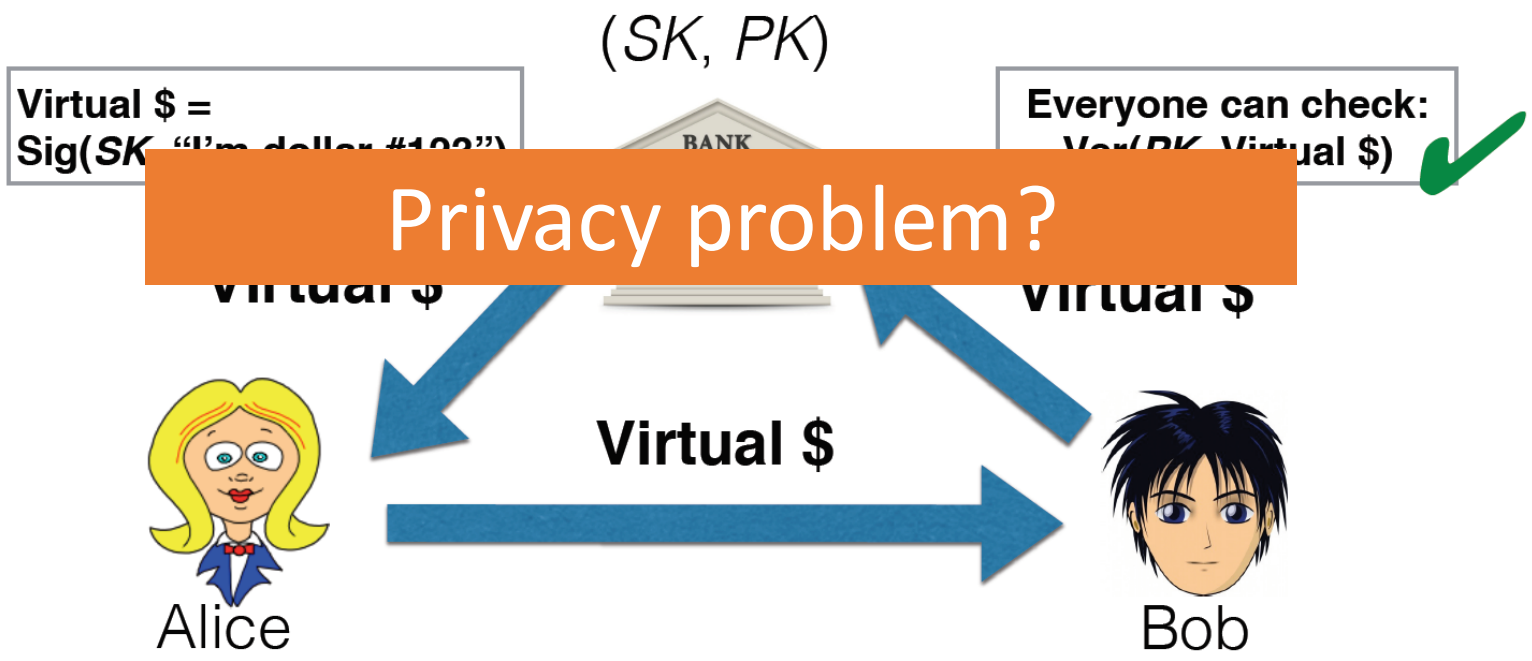
One of the earliest proposed uses of digital signatures (RSA) was to create virtual currency (in Ireland)

- Idea: A bank creates coins consisting of digital signatures
- Simplified version…



E-cash

- Bank can also record spent serial numbers
- Bob can verify validity of coin online



Privacy problem?

Blind digital signatures (Chaum, 1982)

- Ideas:
  - Alice chooses a serial number *z* for virtual $
  - Bank digitally signs *z* without seeing it
- RSA setup:
  - Public key: (*N*, *e*); private key (*N*, *d*)
  - Full-domain hash: *H*: {0,1}* -> Z_N

RSA blind signature

serial num. $z \leftarrow \$$

$c = H(z)$

$r \leftarrow \$ \ Z_N$

$t = r^e c$

$s' = t^d$

$= rc^d$

s'

$s = s' / r$

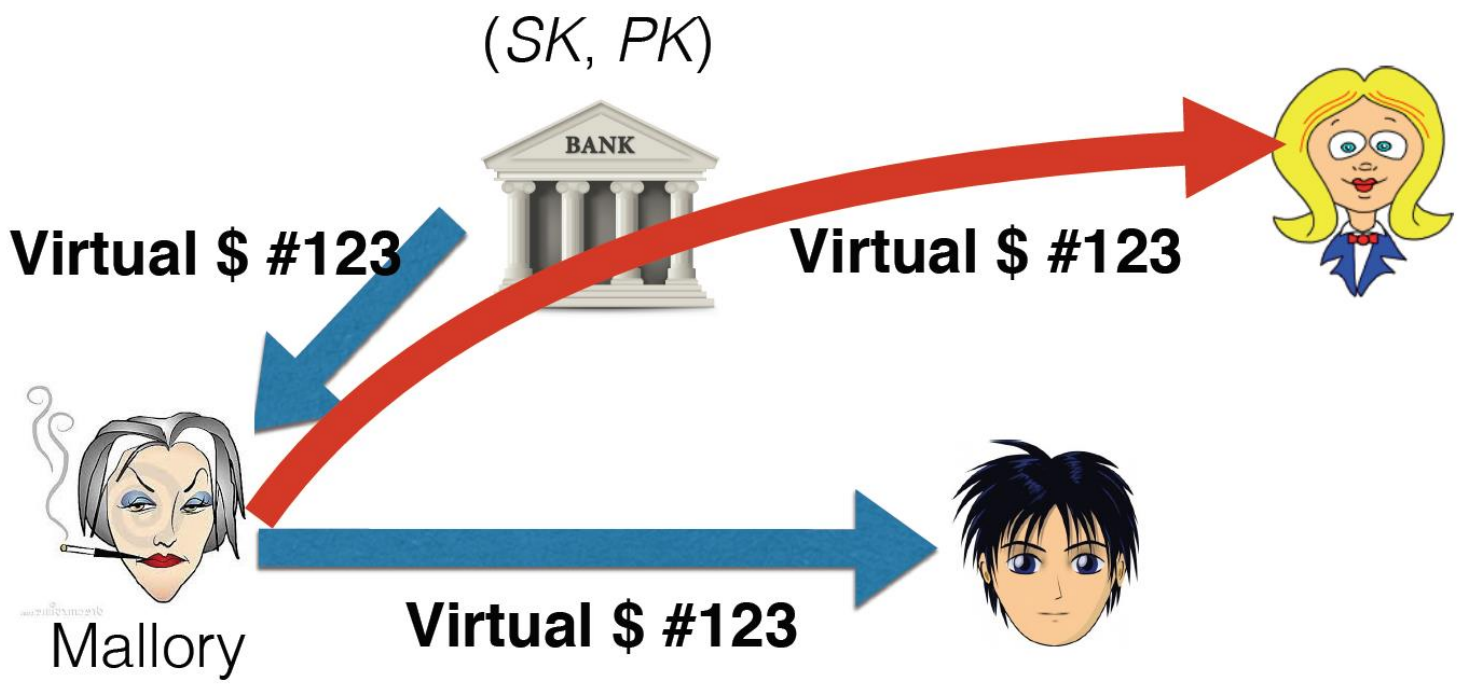
$= c^d$

RSA public key (N,e); private key: (N, d)

Operations mod N

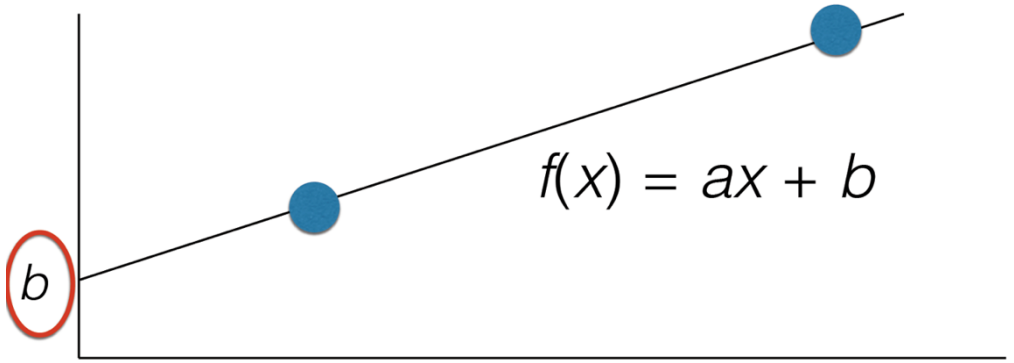
What if we want to go offline?

- Suppose Mallory double-spends in blinded scheme?



Conditional anonymity

- Intuition:
  - Identity encoded as value *b*
  - Coin randomness encoded as value *a*
  - Receiver challenges coin spender to reveal point *f(x)* when spending
  - Double-spending: two points uniquely specify a line



Despite promising, we still need a *trusted* third party.

The Times 03/Jan/2009 *Chancellor on brink of second bailout for banks*.

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of

```
bitcoin-0.1.0.rar
bitcoin-0.1.0.tgz
```

Paper published in 2008
Source code in 2009

Wait…But who is Nakamoto?

Bitcoin's face

And another …

Bitcoin design -- from basic principles

Key property #1
Bitcoin is pseudonymous

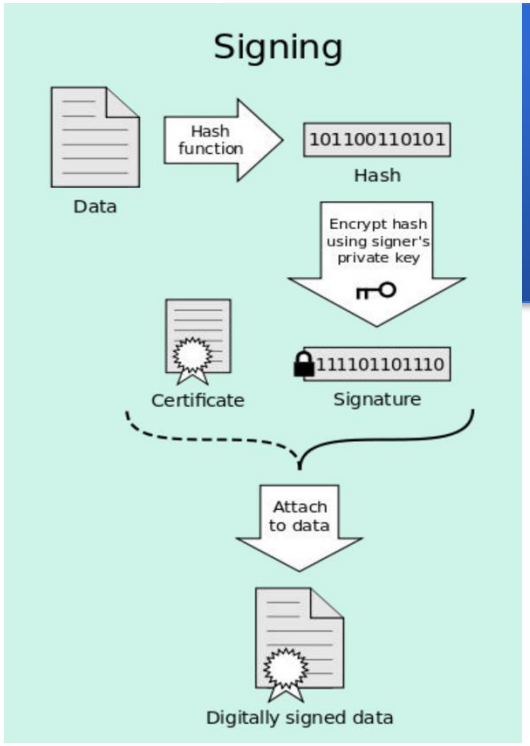
1. Each entity X has an (ECDSA) key pair (PK_X, SK_X)
2. No association between X and real-world identity

PK: hUK67H9fyg

SK: z4Pxc2kKn3

Recall: digital signature

- First, create a message digest using a cryptographic hash
- Then, sign the message digest with your private key



Authentication

Integrity

Non-repudiation

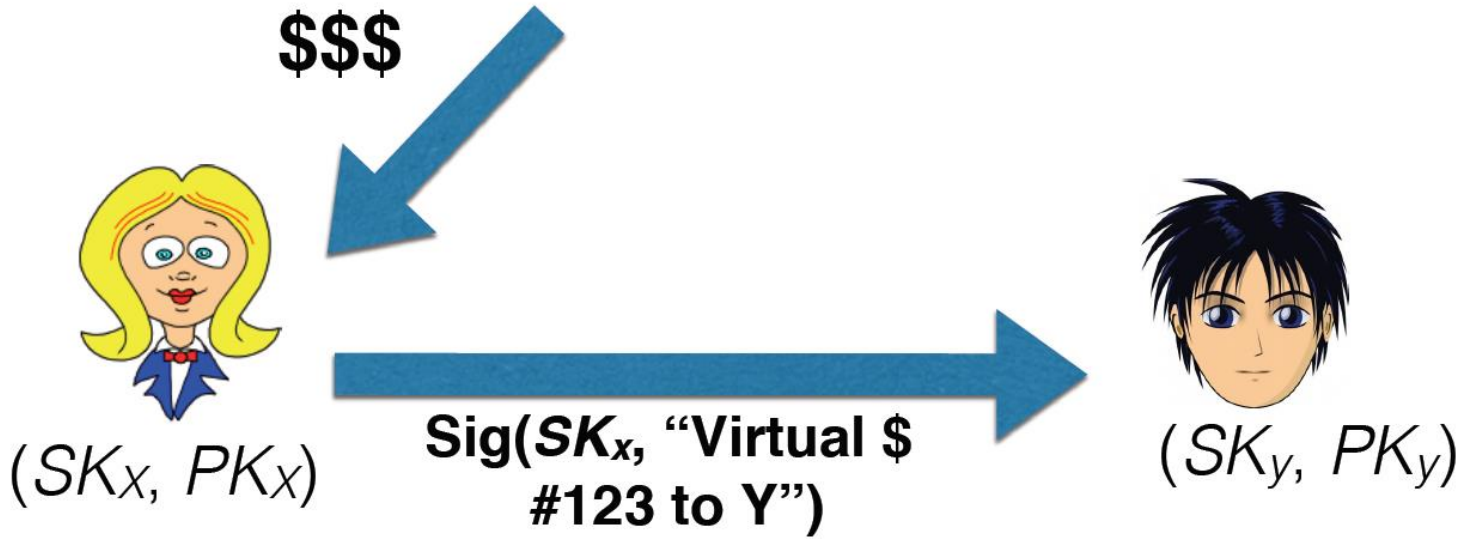
Digital signatures are used in bitcoin

Bitcoin uses ECDSA

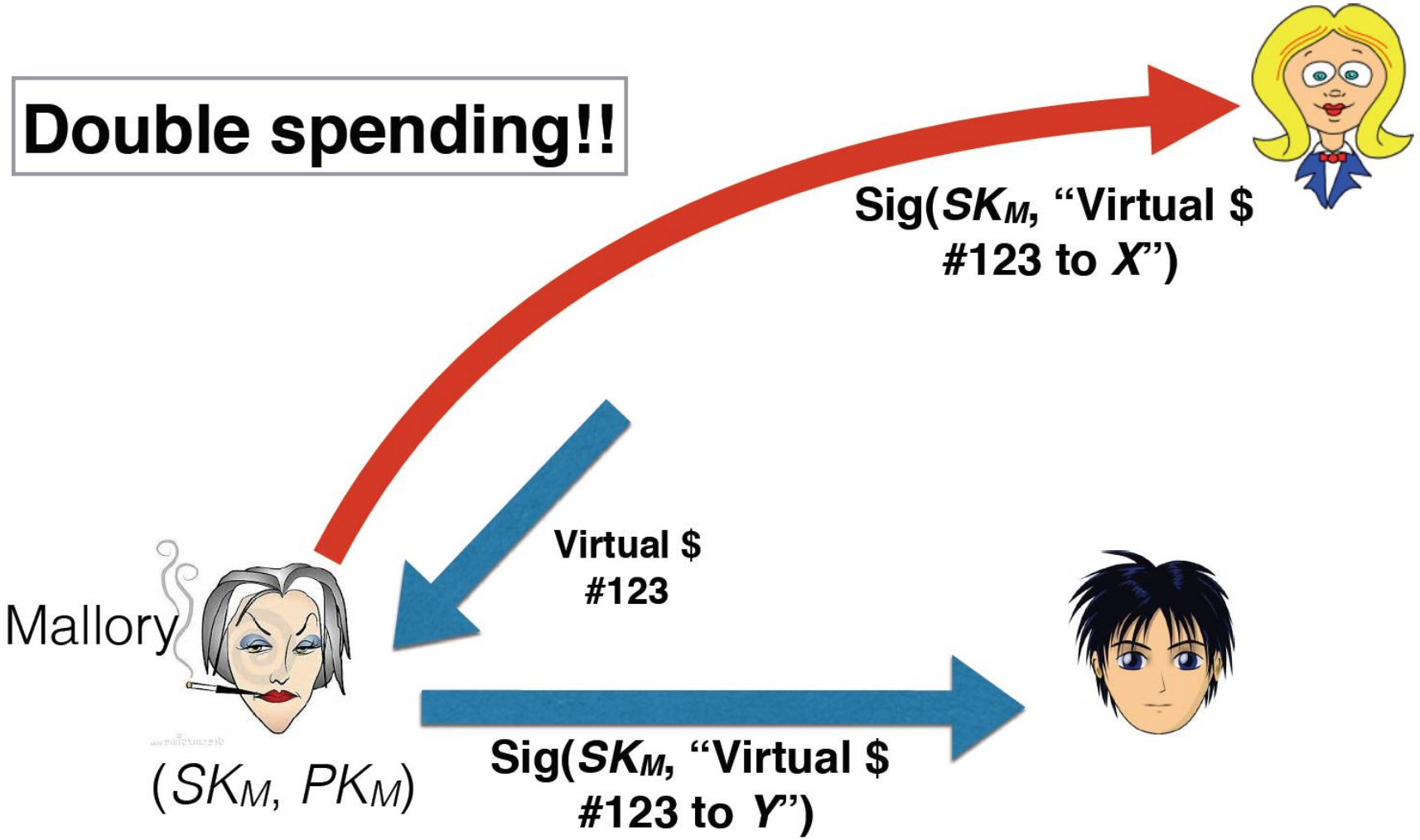
- "Elliptic-Curve Digital Signature Algorithm"
- Concretely, uses secp256k1 (slightly nonstandard) curve
  - Private key *SK* is 256 bits; (uncompressed) public key *PK* is 512 bits

Could build naïve system…

- Idea: Coins *and* transactions, i.e., flow of money, can be authenticated—neither is forgeable
- Thanks to public-key crypto, everyone can verify all coins and transactions (if public keys are distributed throughout system)



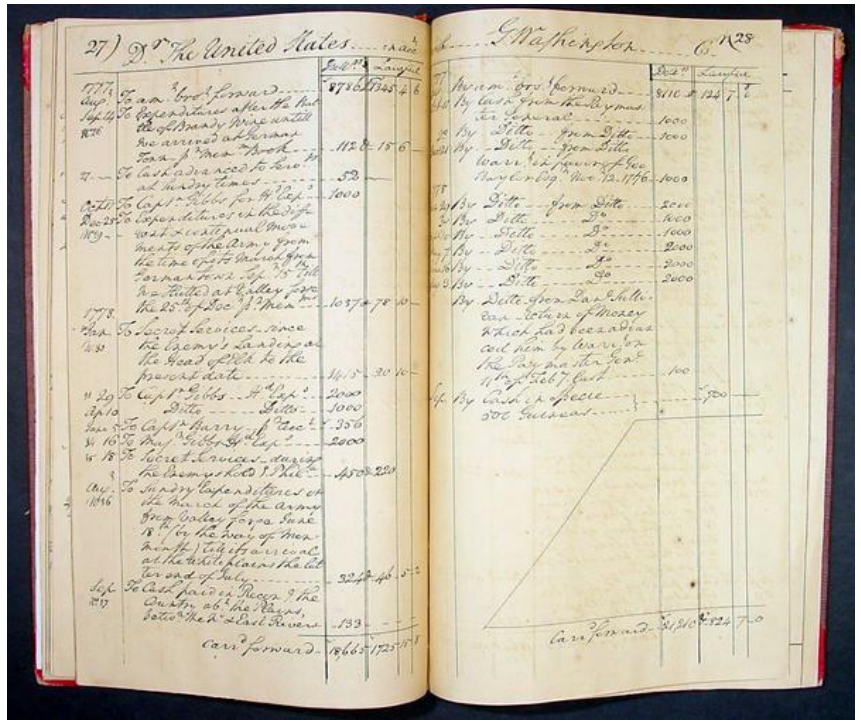
But there is a problem



The double spending problem

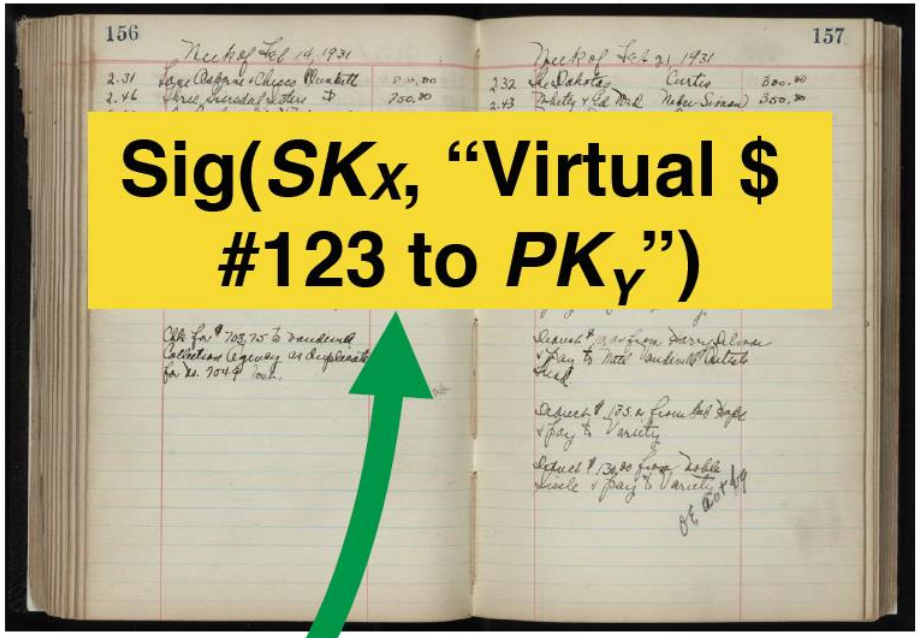
- Suppose Mallory gives the same dollar (dollar #123) to *both* Alice *and* Bob.
- The good news:
  - She can be caught after the fact.
  - Alice and Bob have (cryptographic) *proof* that Mallory cheated: Mallory's signatures!
- The bad news:
  - Either Alice or Bob is out a dollar.
  - Who's going to prosecute Mallory?
  - E.g., suppose Mallory is halfway around the world?

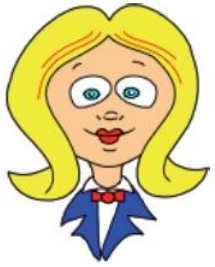
What a bank will do to prevent double spending?



Maintain a ledger to record every transaction!

Bitcoin entities emulate a *public* trusted bulletin-board (ledger)





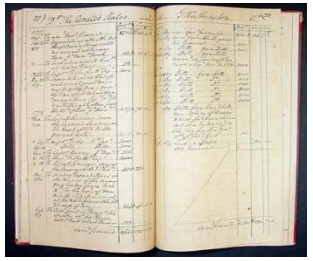
(SK_X, PK_X)

Sig(SK_X, "Virtual $ #123 to PK_Y")

(SK_Y, PK_Y)

Bob checks Sig *and* ledger ✔

The *public* ledger prevents double spending



Ledger

Sig(SK_M, "Virtual $ #123 to *X*")

The ledger shows that you've already spent this coin to *Y*!

Virtual $ #123

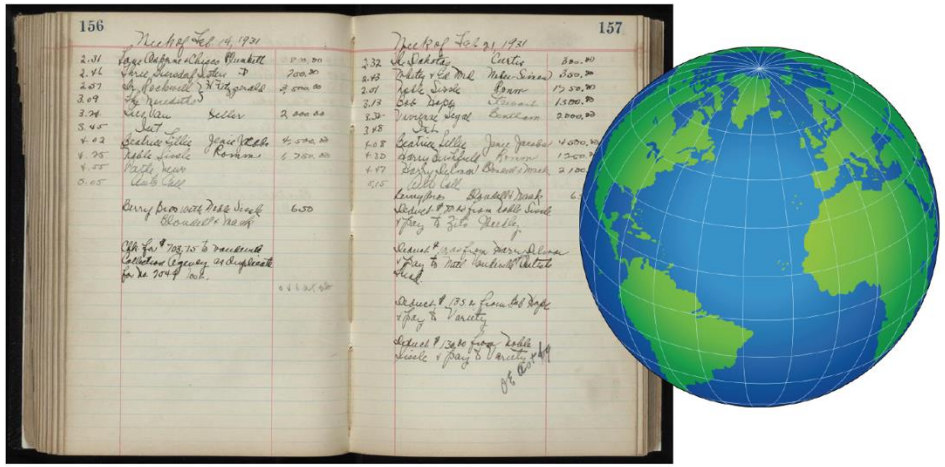
Mallory

(SK_M, PK_M)

Sig(SK_M, "Virtual $ #123 to *Y*")

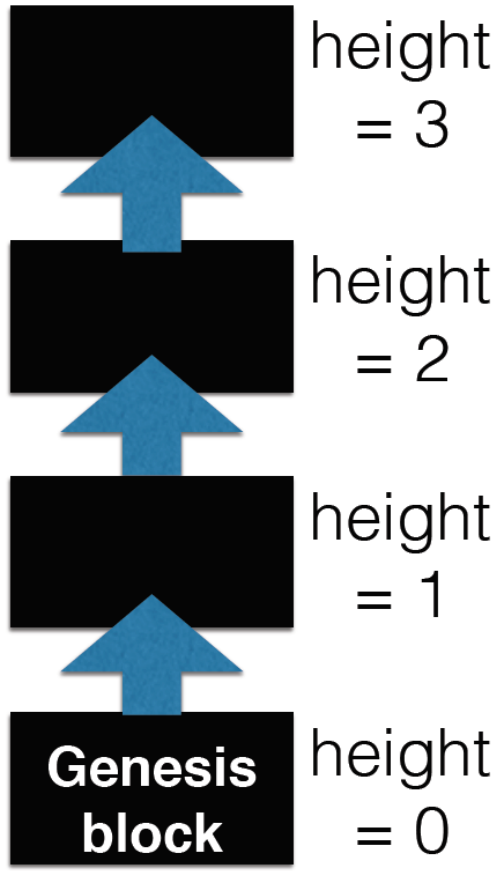
Key property #2
Bitcoin is decentralized

1. Has a ledger, but with **no Bank**!
2. The ledger is *agreed upon* and *distributed* among many entities
3. The ledger is called **blockchain** in Bitcoin

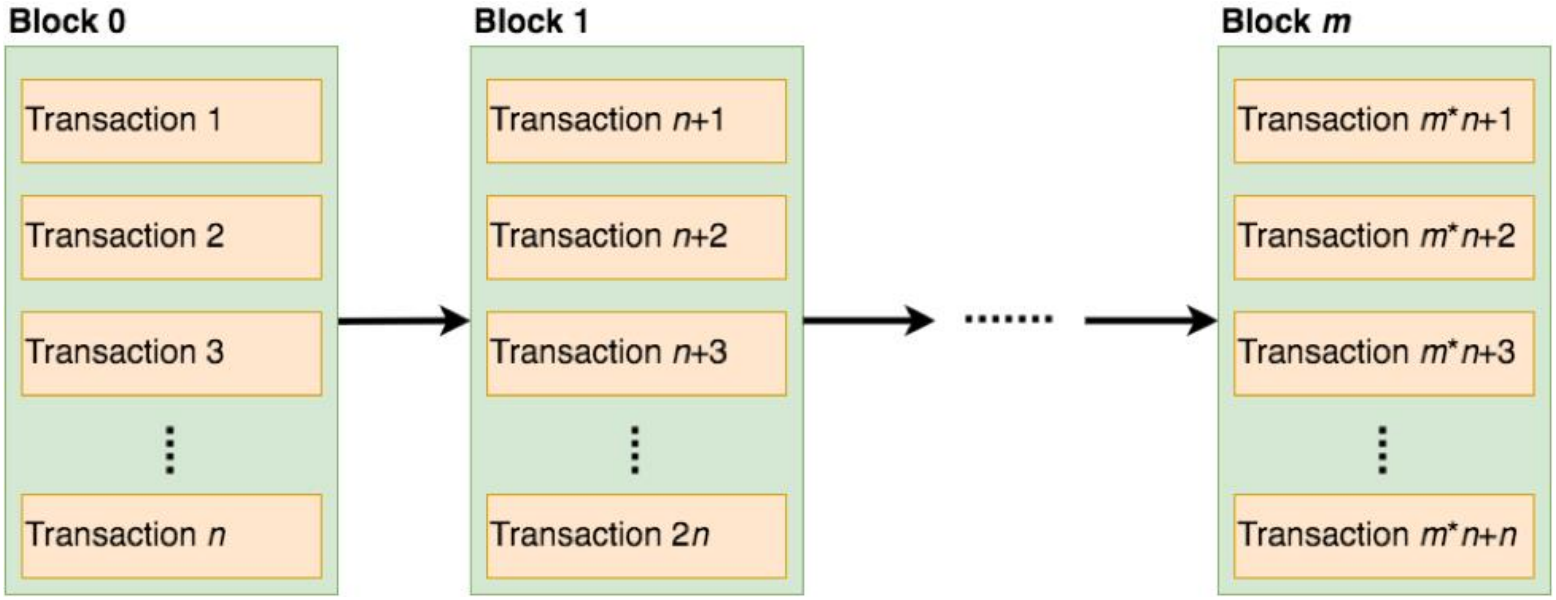


Blockchain

- Record of *every* transaction in Bitcoin system
- Maintained as append-only data structure
- New block added every 10 minutes (on average)
- Each block contains a bundle of latest transactions.
  - E.g., SIG_{SKA} [“Alice sends 0.4 BTC to Bob”]
  - (Actually, there’s a scripting language, but we’ll gloss over it…)



Block of transactions



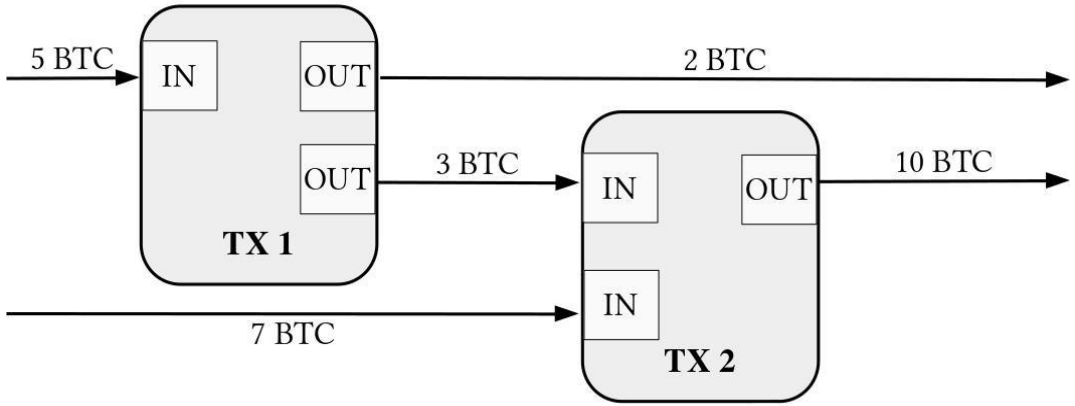
One block every 10 minutes
Around 2000 transactions per block

Q: what's the average transaction rate? (transactions/second)

https://ycharts.com/indicators/bitcoin_average_transactions_per_block Assessed on Jan. 5 2023.

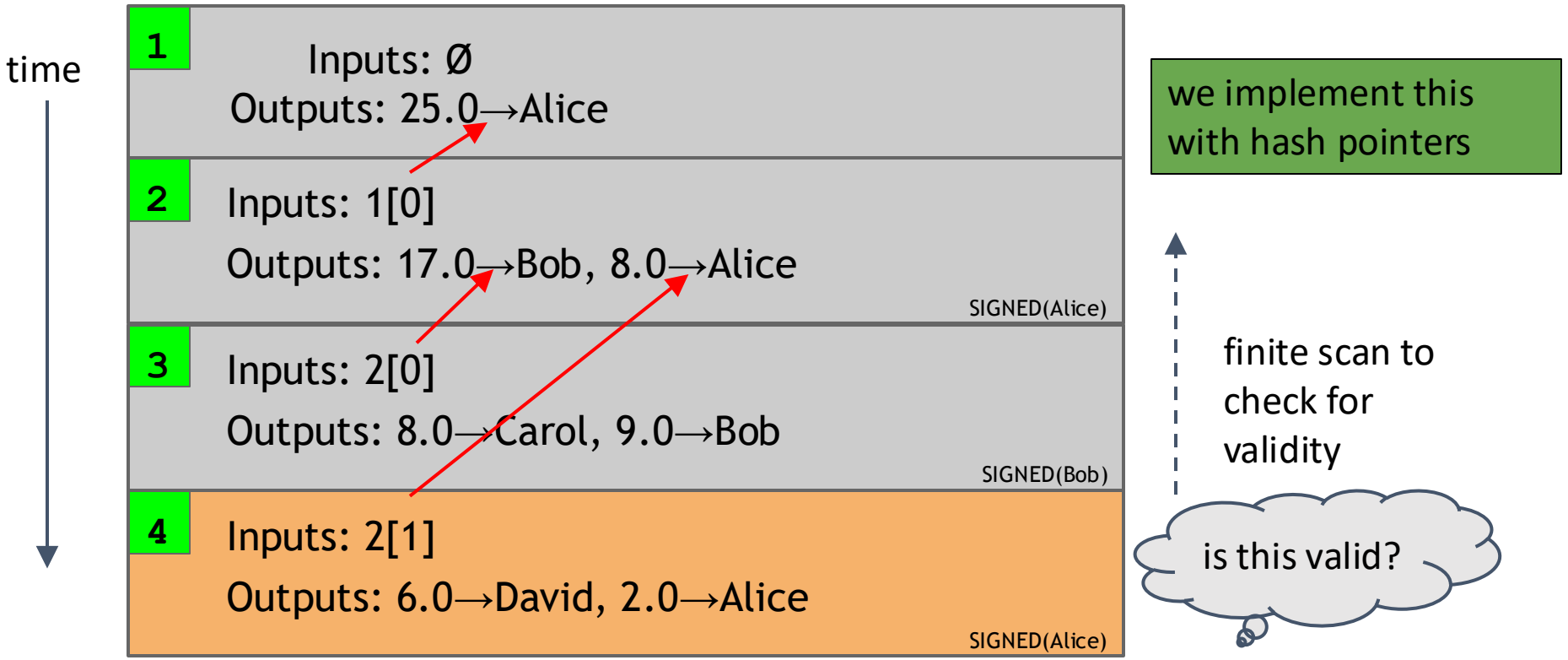
UTXO model

- Unspent transactions output (UTXO)
  - No explicit “account balances”
  - Structured in terms of transactions
- Your account balance is the combined value of all UTXOs you control
- Each transactions consumes at least one UTXO and creates at least one UTXO



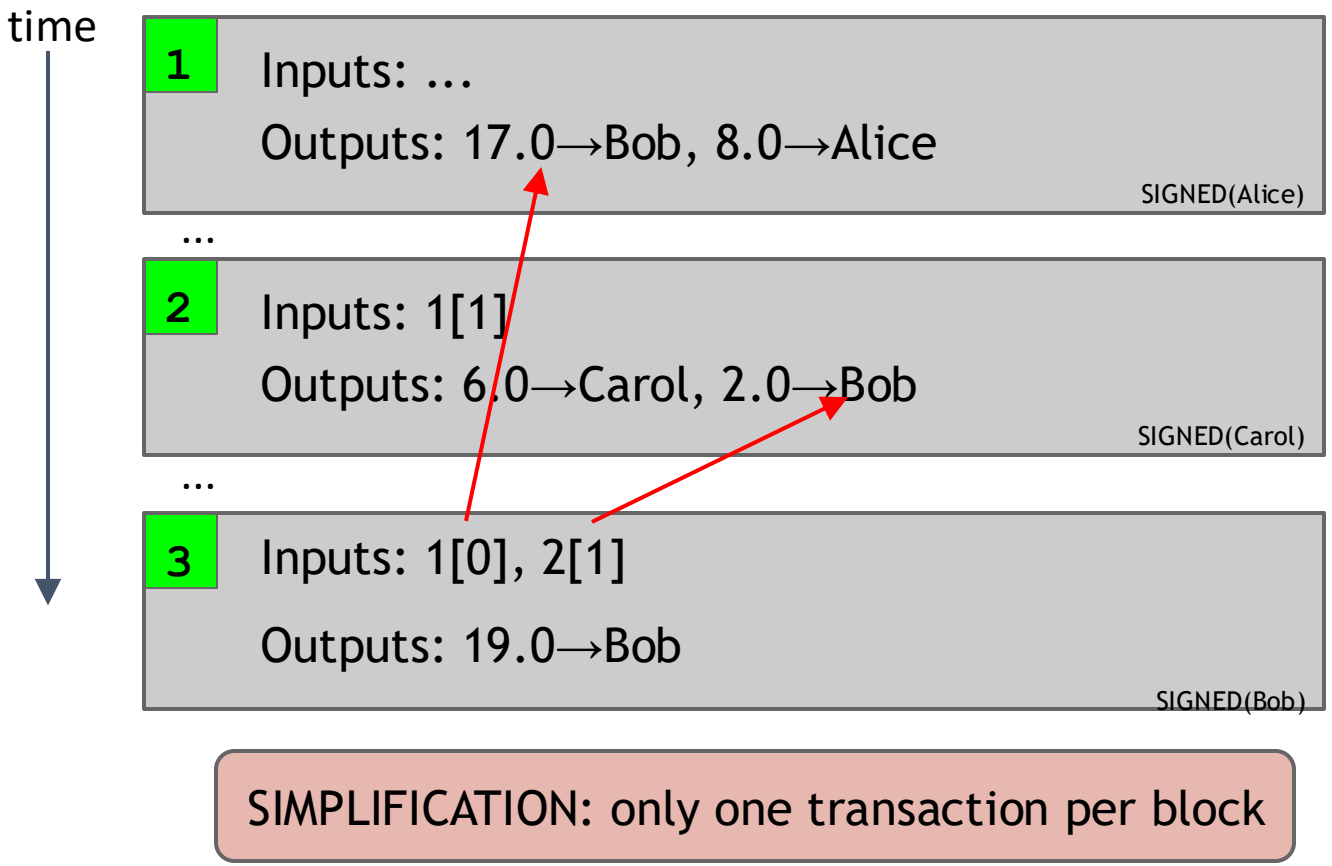
- Each UTXO can only be consumed at most once and only in its entirety
- The sum of a transactions inputs must be greater or equal to the sum of its outputs
  - The difference is the transaction fee

Demo



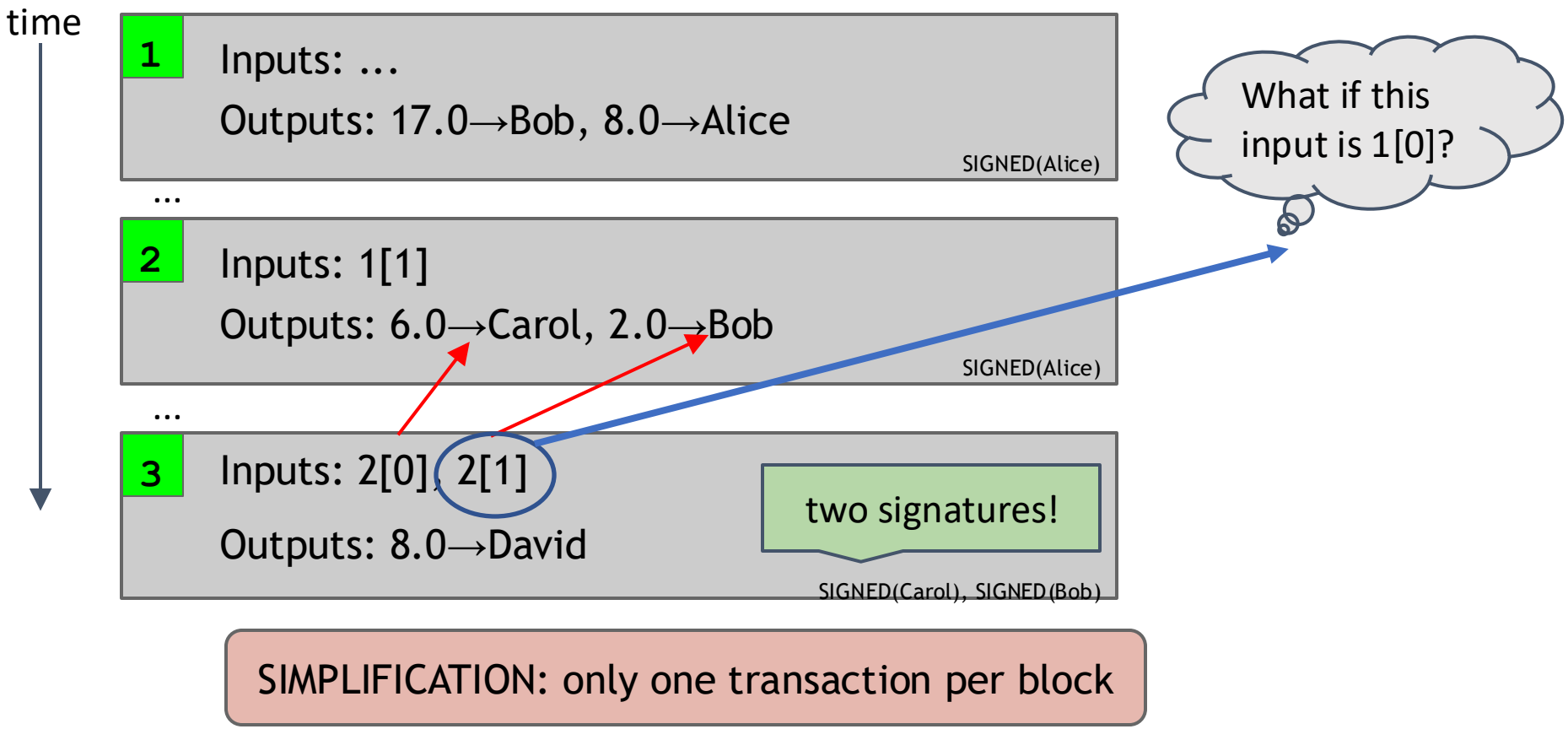
SIMPLIFICATION: only one transaction per block

Merging value



SIMPLIFICATION: only one transaction per block

Joint payments



Key challenge in bitcoin

- **Distributed consensus:**

  All “correct” nodes decide on the **same** value

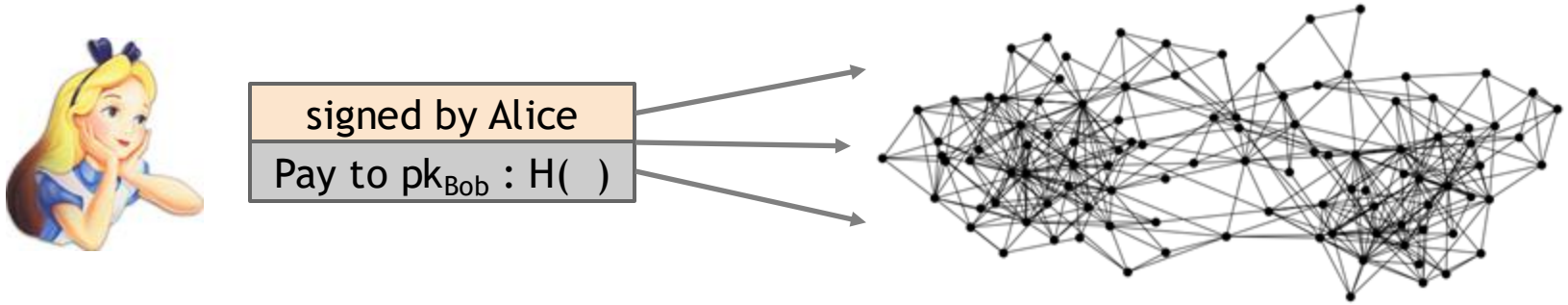
  This value must have been proposed by some **correct** node

  The protocol terminates

(more about this in the next class…)

Bitcoin is a peer-to-peer system

When Alice wants to pay Bob:
she broadcasts the transaction to all Bitcoin nodes

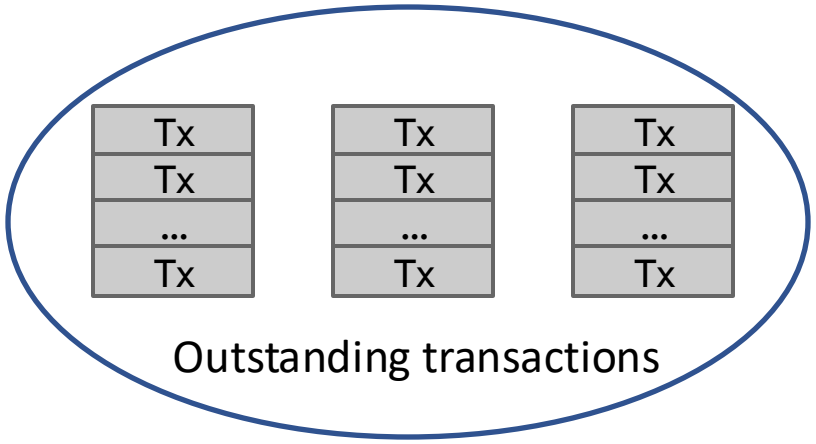


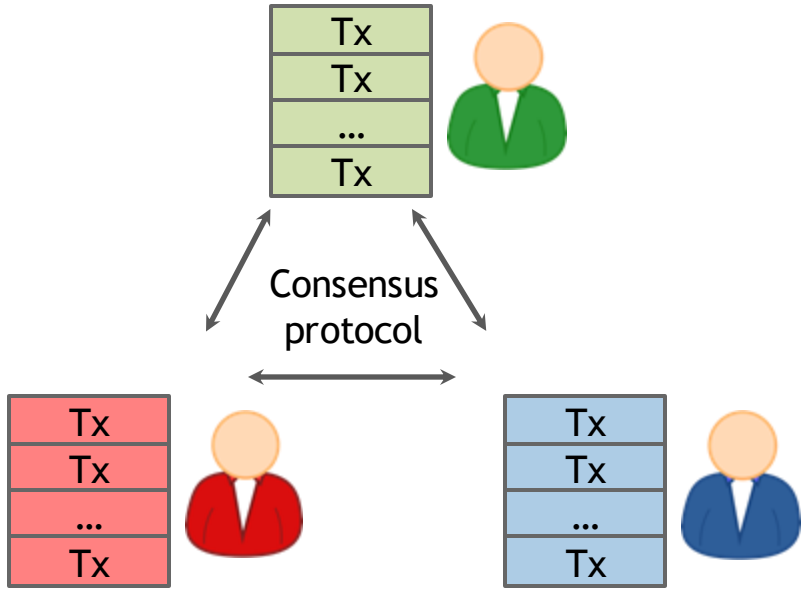
Note: Bob's computer is not in the picture

How concensus could work in bitcoin

At any given time:

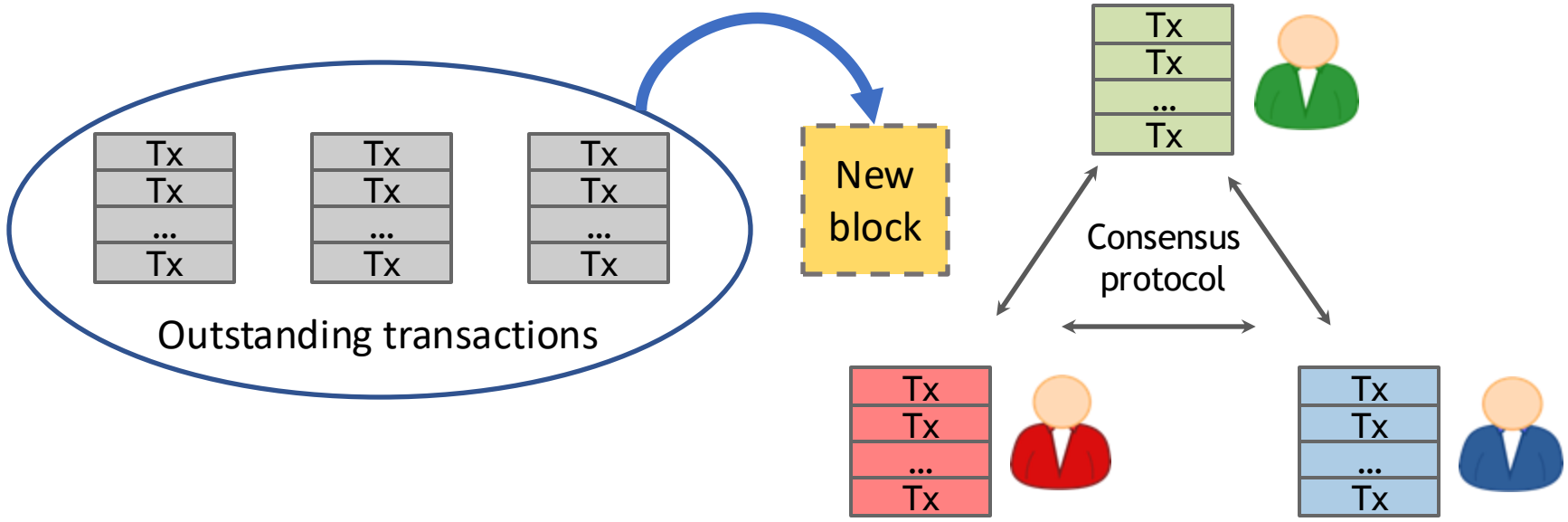
- All nodes have a sequence of **blocks of transactions** they've **reached consensus on**
- Each node has a set of outstanding transactions it's heard about





How concensus could work in bitcoin

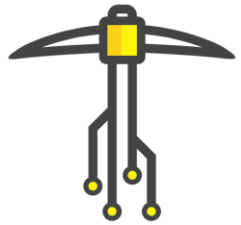
- Now, how does system decide **which block to be extended?**
- Ideal for P2P system: All clients in the world *vote* on the correct new block



How concensus could work in bitcoin

- **But** it's hard to ensure one vote per machine in a P2P system
  - E.g., there's the problem of "Sybil" attacks
    - one user creates multiple identities

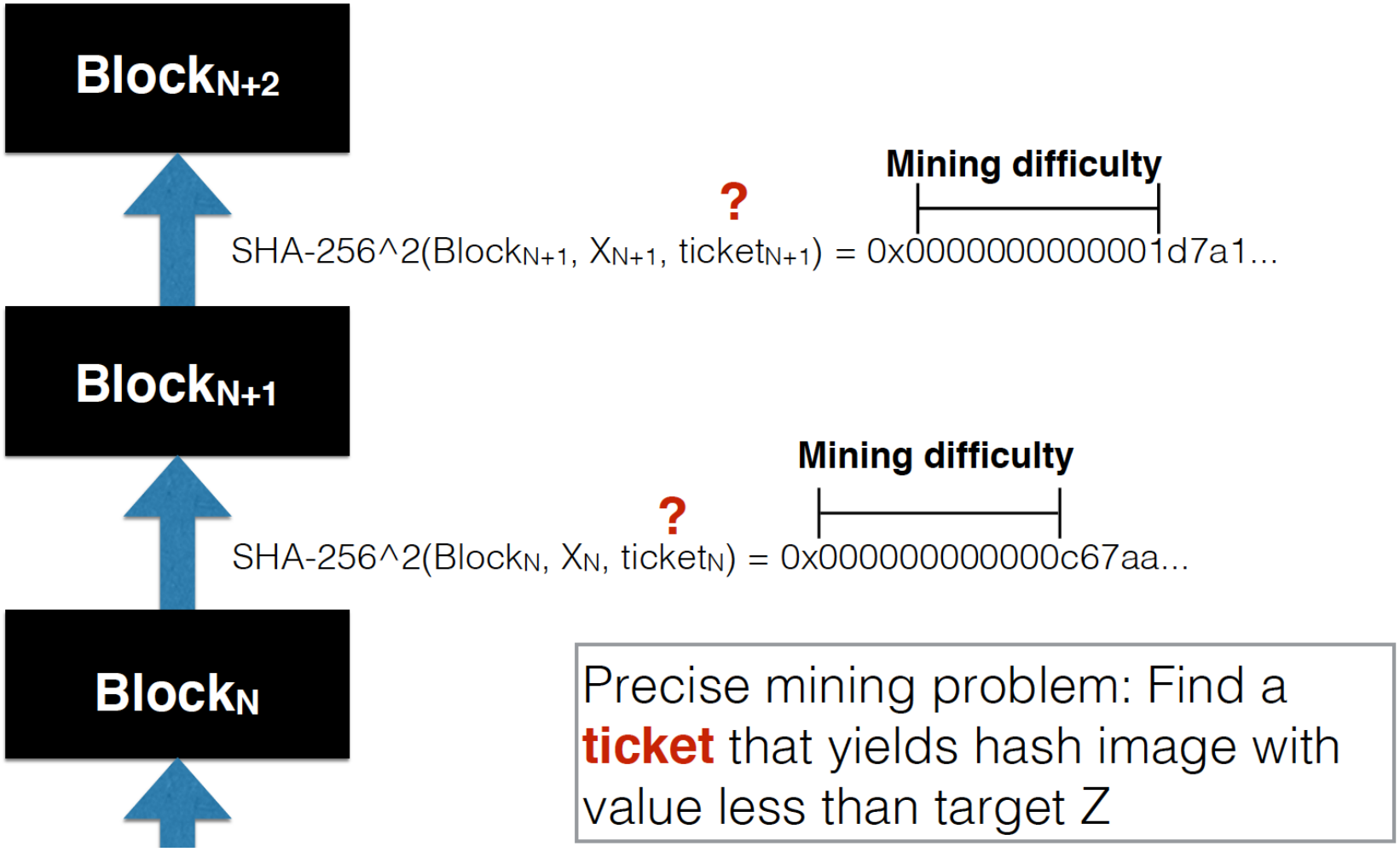
- So "voting" (cleverly) in Bitcoin takes the form of hash power.
  - I.e., one vote per CPU (roughly speaking)



“Mining” in bitcoin

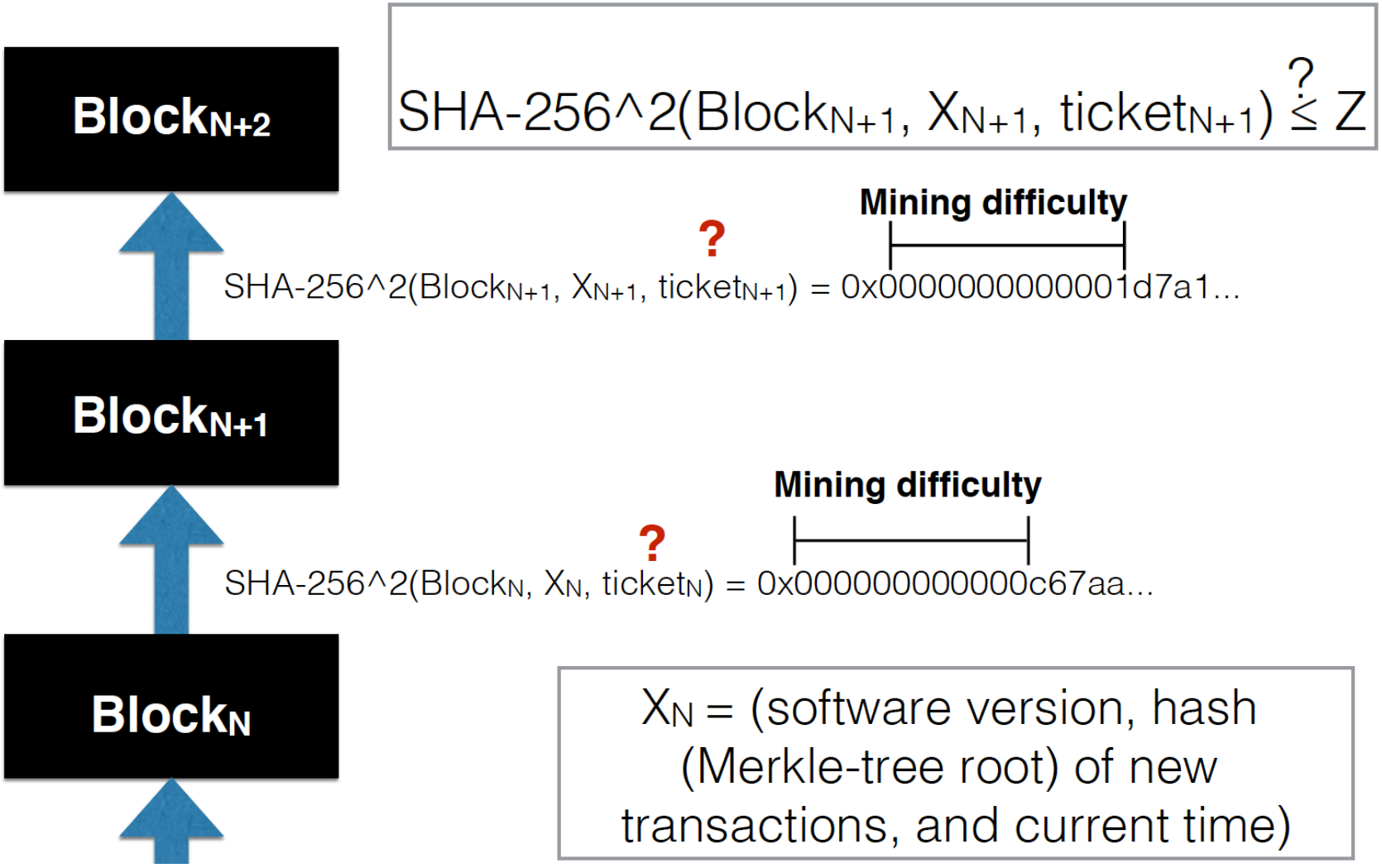
- All miners execute communal, computationally–intensive process called **mining**.
  - Together, mining community defines blockchain
- Intuition:
  - All miners collectively search for hard-to-compute “signature” on new block (solve a puzzle)
  - Attacker with little computing power unlikely to mine new valid block faster than honest ones
    - Security: assume less than 50% malicious

Block mining

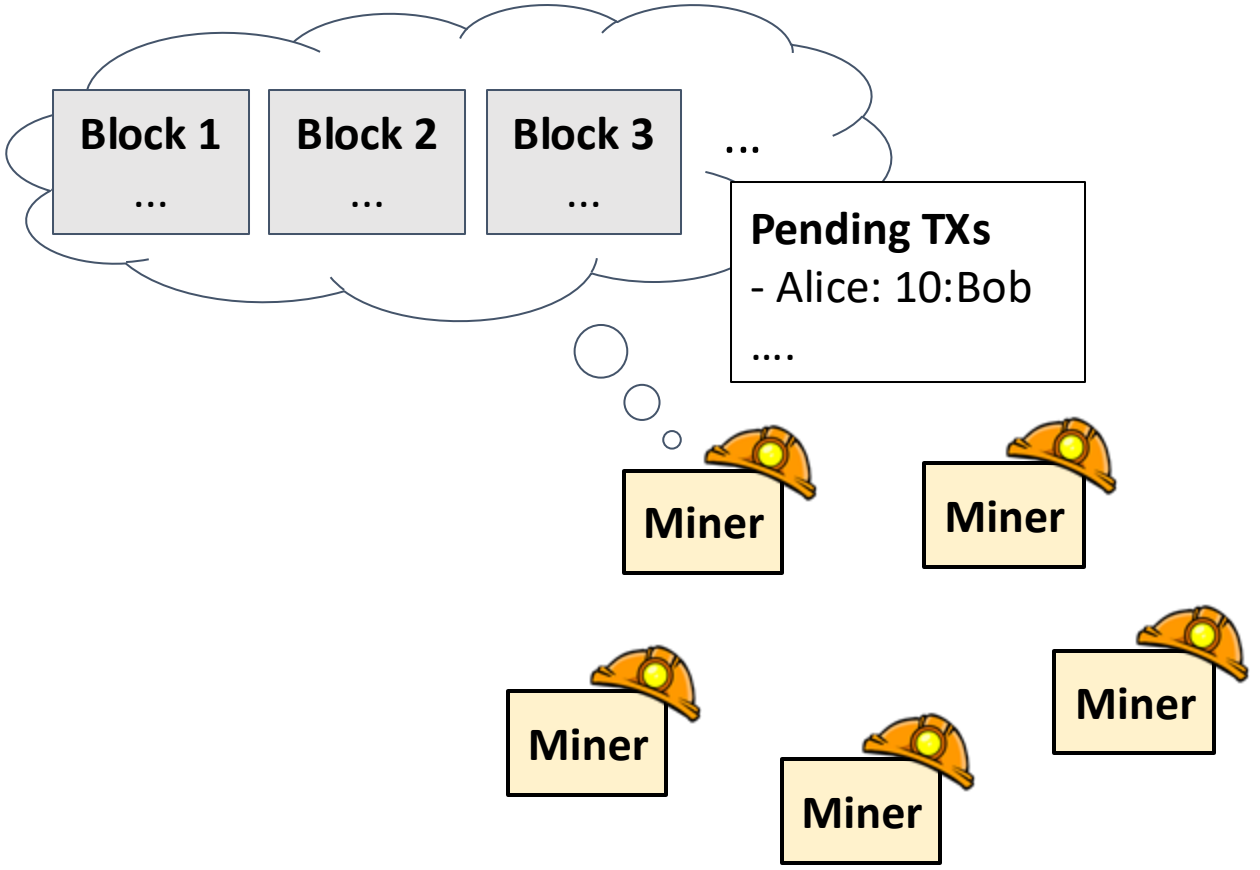


Precise mining problem: Find a **ticket** that yields hash image with value less than target Z

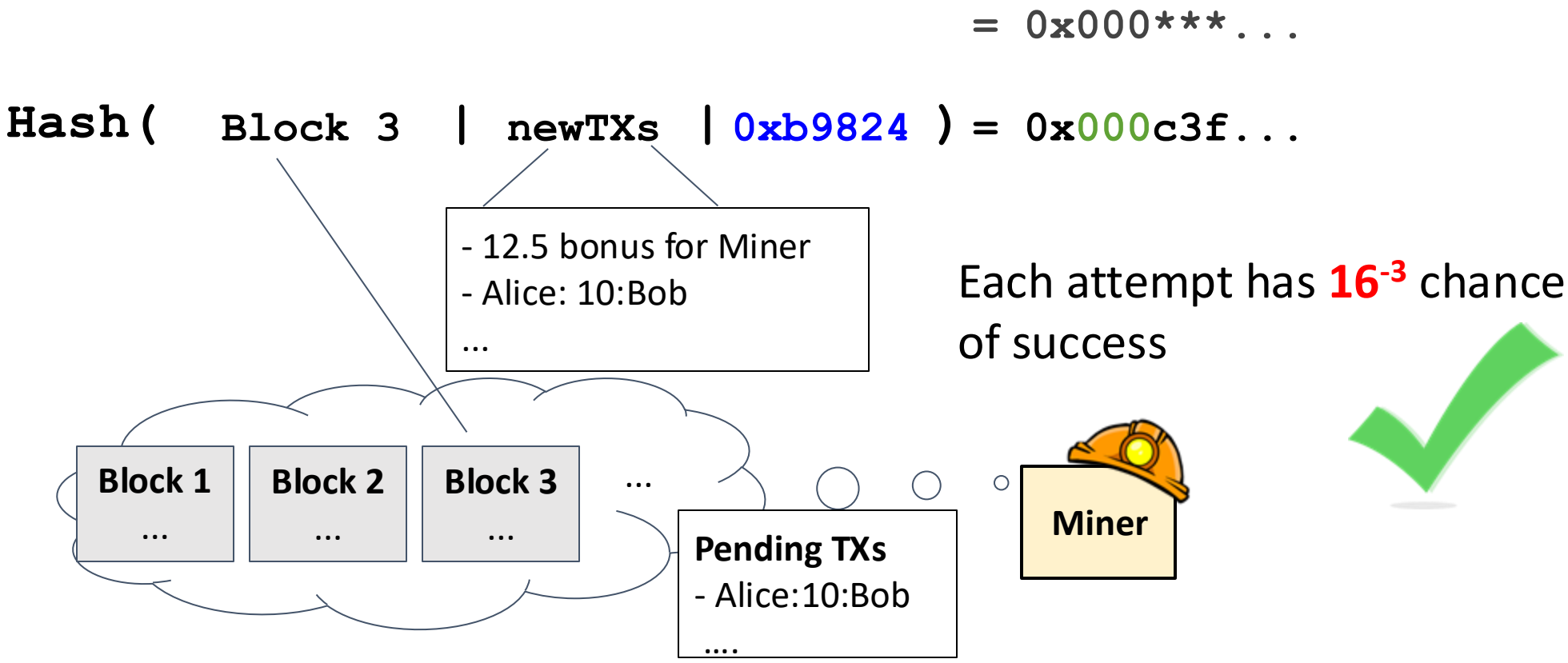
Block mining

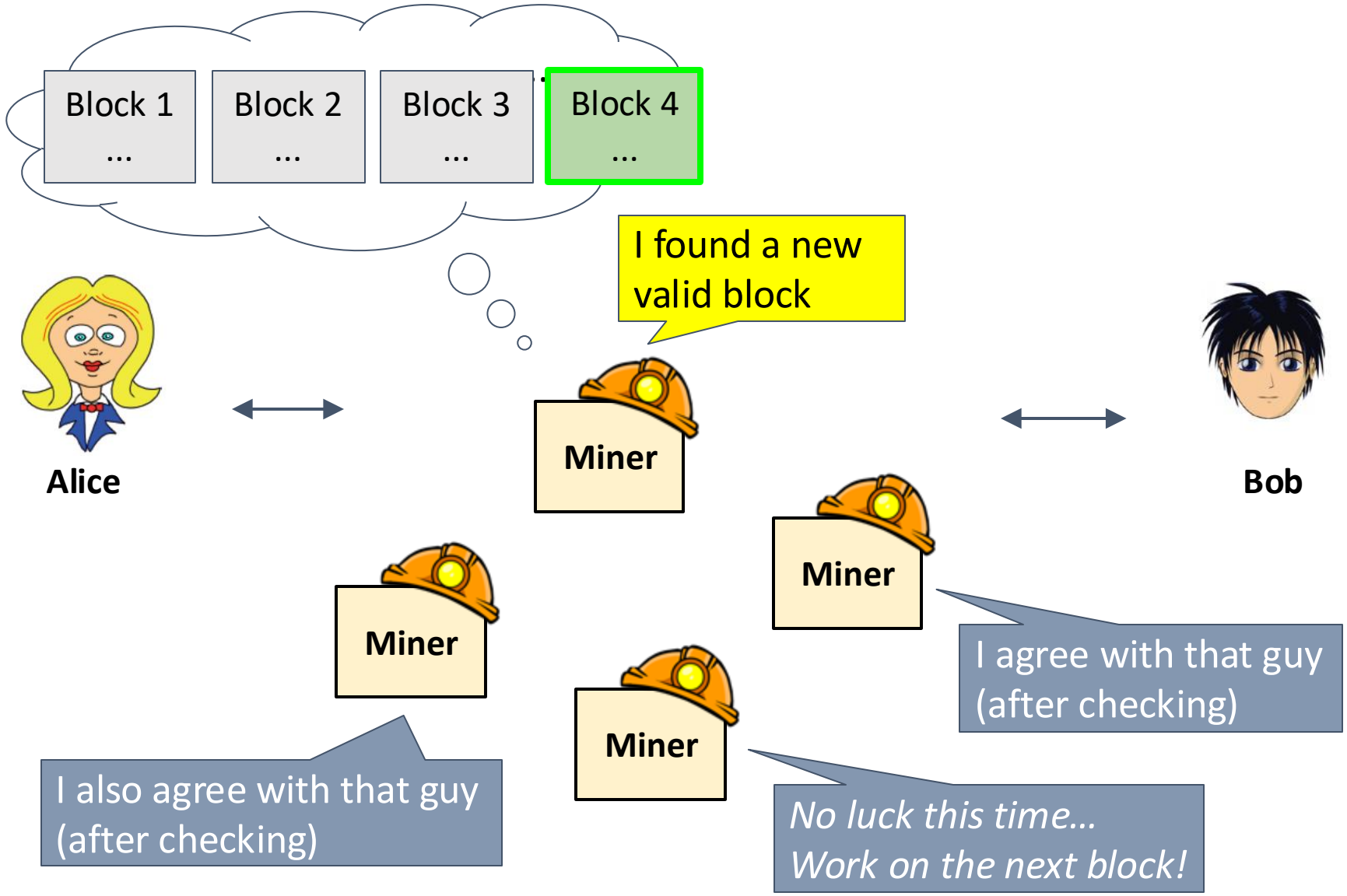


Demo


Block 1
...
Block 2
...
Block 3
...
...
Pending TXs
- Alice: 10:Bob
....
Miner
Miner
Miner
Miner
Miner

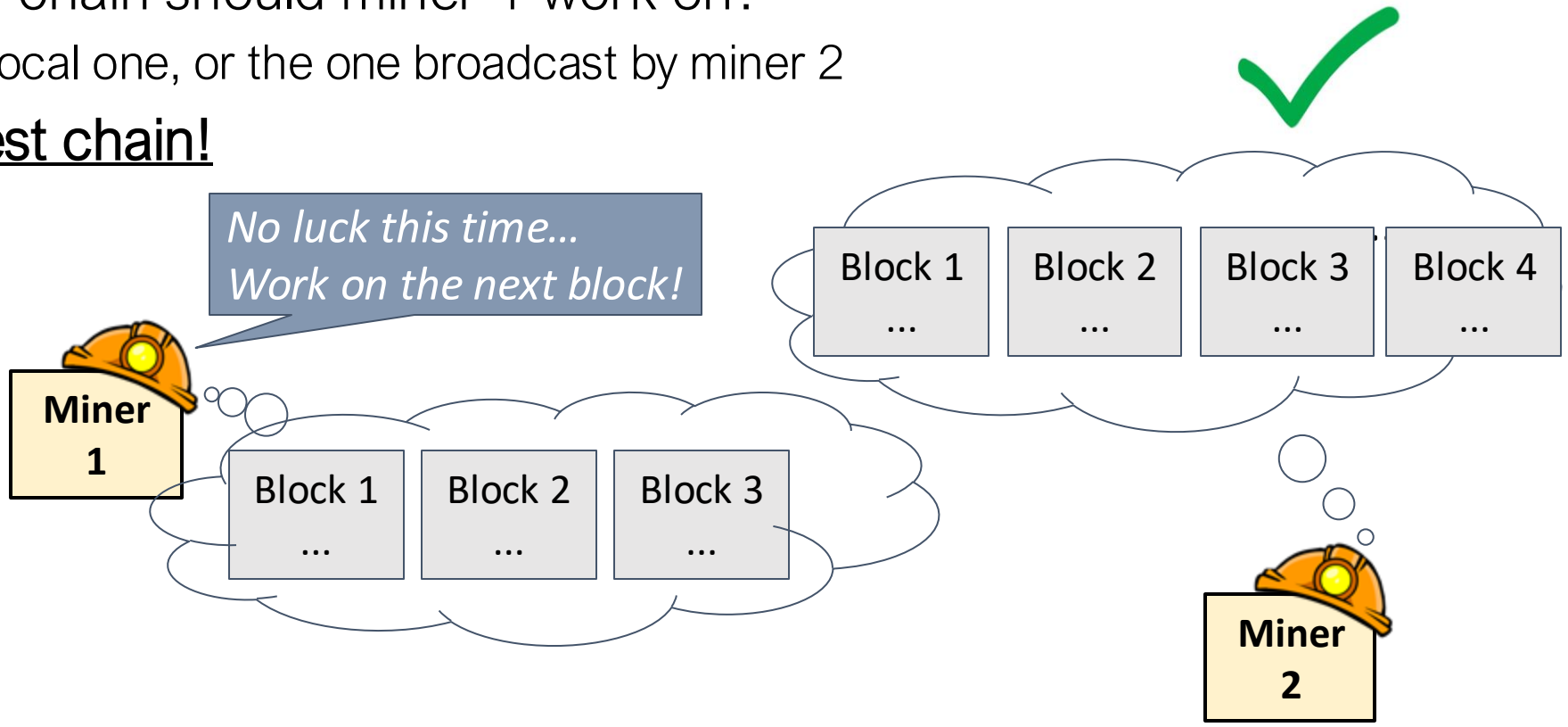
Miners commit new transactions by solving puzzles




Block 1
...
Block 2
...
Block 3
...
Block 4
...
I found a new valid block
Alice
Miner
Bob
Miner
Miner
I agree with that guy (after checking)
I also agree with that guy (after checking)
Miner
No luck this time…
Work on the next block!

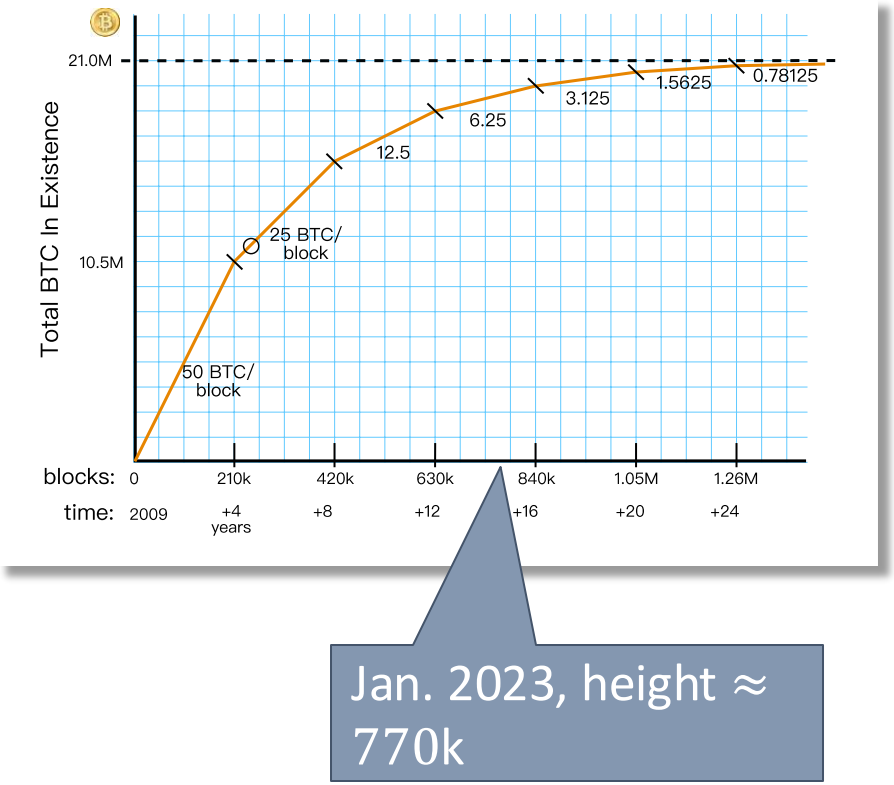
Longest chain rule

- One subtle question
  - Which chain should miner 1 work on?
    - His local one, or the one broadcast by miner 2
  - <u>Longest chain!</u>



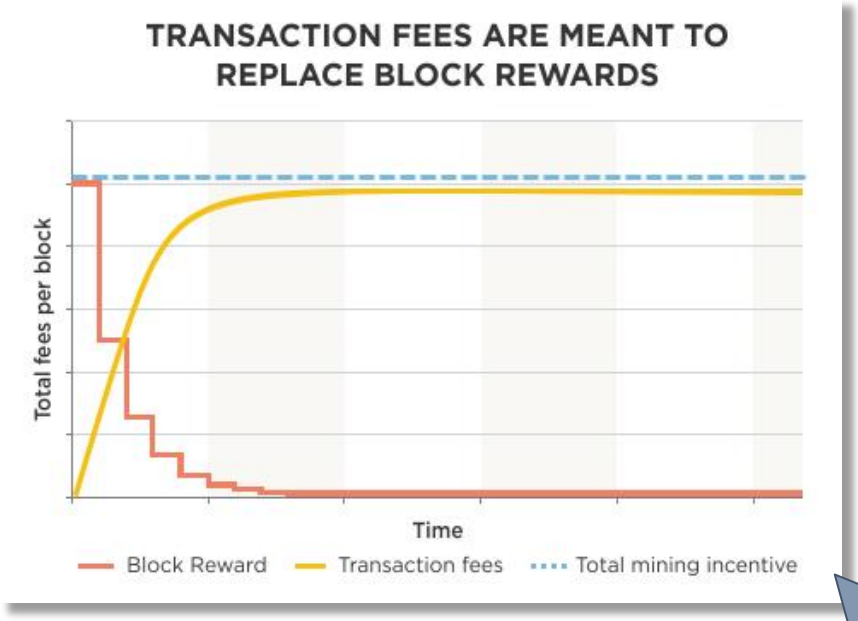
What's the incentive for miners to mine?

- Key idea: Bitcoin is a lottery
- Every miner tries tickets until a "winning" one is found
- The reward for the winner: Bitcoin!
  - Special transaction (*coinbase*) in block assigns BTC to winner
  - Originally, 50 BTC/block; today, 6.25 BTC/block
  - **Winner also gets *transaction fees***
- 21 million BTC will be produced over the lifetime of the system



What's the incentive for miners to mine?

- Key idea: Bitcoin is a lottery
- Every miner tries tickets until a "winning" one is found
- The reward for the winner: Bitcoin!
  - Special transaction (*coinbase*) in block assigns BTC to winner
  - Originally, 50 BTC/block; today, 6.25 BTC/block
  - **Winner also gets *transaction fees***
- 21 million BTC will be produced over the lifetime of the system



transaction fee = transaction value – spent value (transaction fee ≥ 0)

Miner: I am more willing to include your transaction as I can earn more BTC from it!

What's the incentive for miners to mine?

- In principle, Bitcoin is democratic
- *Anyone* can mine
- *Reward* is proportional to computational investment
- But…

How long will it take to find a block?

- In the early days, with an PC (CPU mining).

```
while (1){
    HDR[kNoncePos]++;
    IF (SHA256(SHA256(HDR)) < (65535 << 208)/ DIFFICULTY)
        return;
}
```

A typical PC can crank out about *one* bitcoin every $4\frac{1}{2}$ years. -- WSJ Oct. 2017

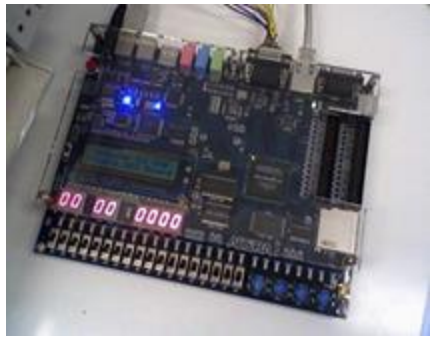
https://www.wsj.com/articles/hackers-latest-move-using-your-computer-to-mine-bitcoin-1509102002. By Robert McMillan.

GPU mining

- GPUs designed for high-performance graphics
  - High parallelism
  - High throughput
- First used for Bitcoin ca. October 2010
- Implemented in OpenCL

Source: LeonardH, cryptocurrenciestalk.com

FPGA mining



- **F**ield **P**rogrammable **G**ate **A**rea
- First used for Bitcoin ca. June 2011
- Implemented in Verilog

Bob Buskirk, thinkcomputers.org

Bitcoin ASICs

TerraMiner™ IV – 2TH/s Networked ASIC Miner

$5,999

Shipping June 2014

300 GH Bitcoin Mining Card

The Monarch BPU 300 C

$1,497.00

Qty: 1 ADD TO CART

Pre-Order Terms: This is a pre-order. 28nm ASIC bitcoin mining hardware products are shipped according to placement in the order queue, and delivery may take 3 months or more after order. All sales are final.

Special purpose!

Professional mining centers

Needs:

cheap power

good network

cool climate

Evolution of mining

CPU

GPU

FPGA

ASIC

gold pan

sluice box

placer mining

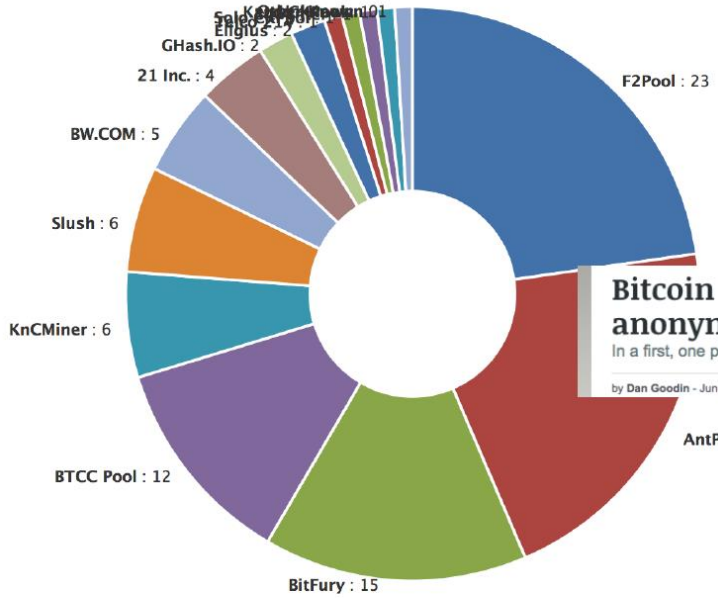
pit mining

Mining pools

Hashrate Distribution An estimation of hashrate distribution amongst the largest mining pools

The graph below shows the market share of the most popular bitcoin mining pools. It should only be used as a rough estimate and for various reasons will not be 100% accurate. A large portion of Unknown blocks does not mean an attack on the network, it simply means we have been unable to determine the origin.

24 hours - 48 hours - 4 Days



Bitcoin security guarantee shattered by anonymous miner with 51% network power

In a first, one player got a monopoly of Bitcoin's total computational power.

by Dan Goodin - June 15 2014, 3:15pm EDT

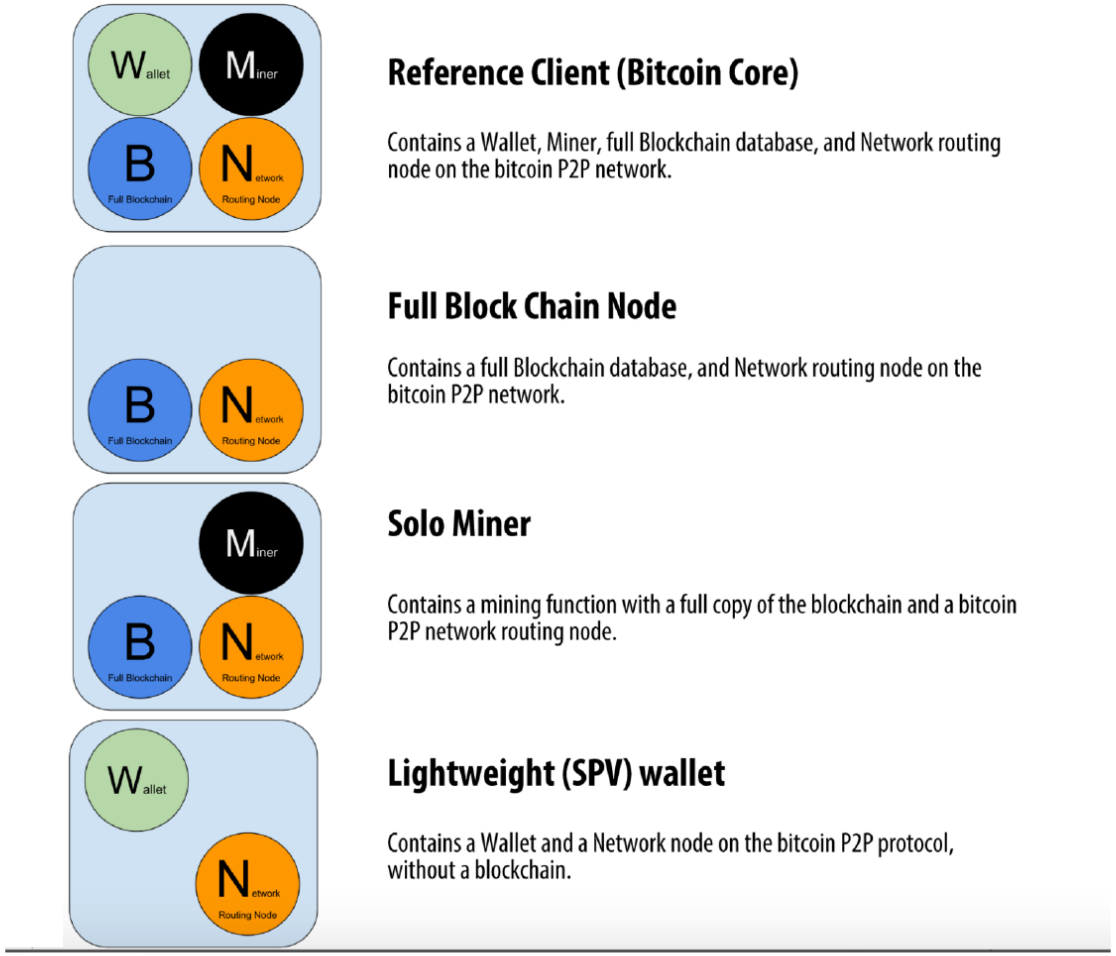
ars technica

Researchers from Cornell University say that on multiple occasions, a single mining pool repeatedly contributed more than 51 percent of Bitcoin's total cryptographic hashing output for spans as long as 12 hours. The contributor was GHash, which bills itself as the "#1 Crypto & Bitcoin Mining Pool." During

Mining blocks isn't enough

- What else is needed to make a working monetary system?

Node types in bitcoin network



[Source: http://chimera.labs.oreilly.com/books/1234000001802/ch06.html]

Routing functionality

- Transactions and blocks are broadcast to *entire network of nodes*
- Rebroadcast protocol
  - Each node transmits to 8 other (randomly selected) nodes
  - TCP on port 8333

Storage

- Full nodes:
  - Store entire blockchain
  - Enforce consensus rules, ensuring blocks in blockchain adhere to
    - 6.25 BTC reward
    - Correct signatures on transactions
    - BTC not double-spent
    - Etc…

Full node distribution

GLOBAL BITCOIN NODES DISTRIBUTION

Reachable nodes as of Thu Jan 07 2021 15:54:21 GMT+0800 (Hong Kong Standard Time).

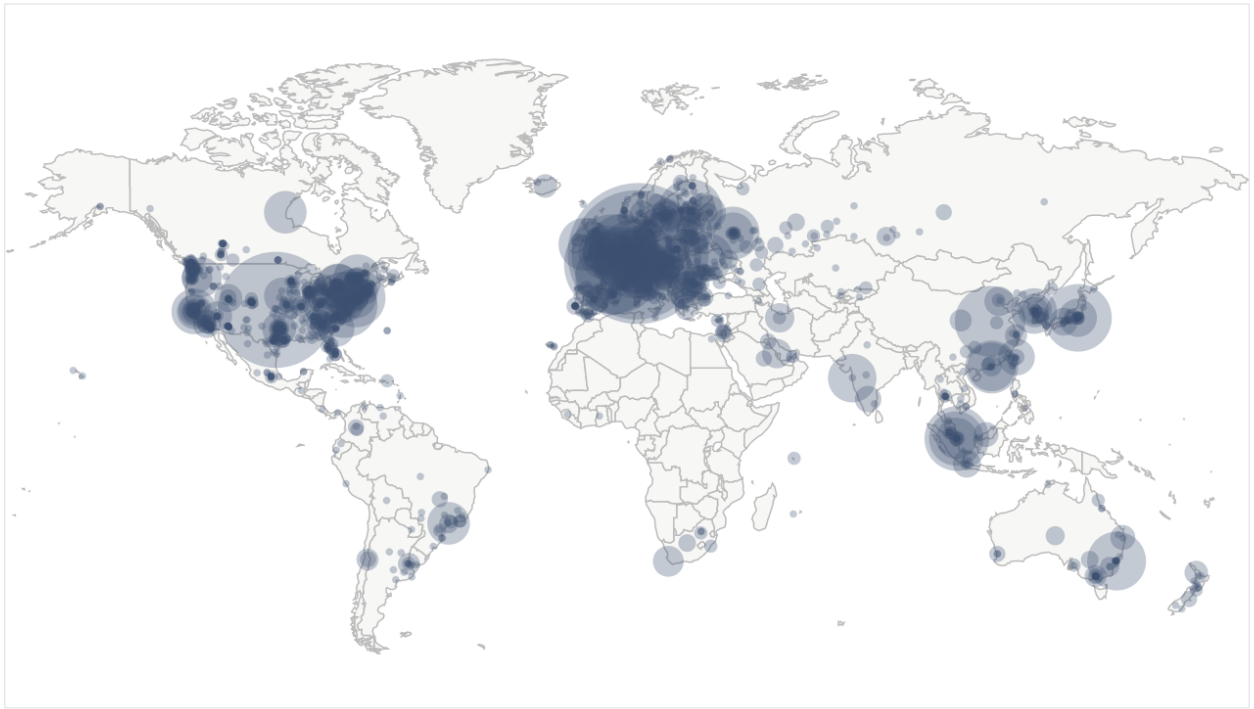
9812 NODES

24-hour charts »

Top 10 countries with their respective number of reachable nodes are as follow.

RANK	COUNTRY	NODES
1	United States	1956 (19.93%)
2	Germany	1794 (18.28%)
3	n/a	1232 (12.56%)
4	France	603 (6.15%)
5	Netherlands	464 (4.73%)
6	Canada	368 (3.75%)
7	United Kingdom	342 (3.49%)
8	Singapore	232 (2.36%)
9	Russian Federation	212 (2.16%)
10	Japan	207 (2.11%)

More (98) »

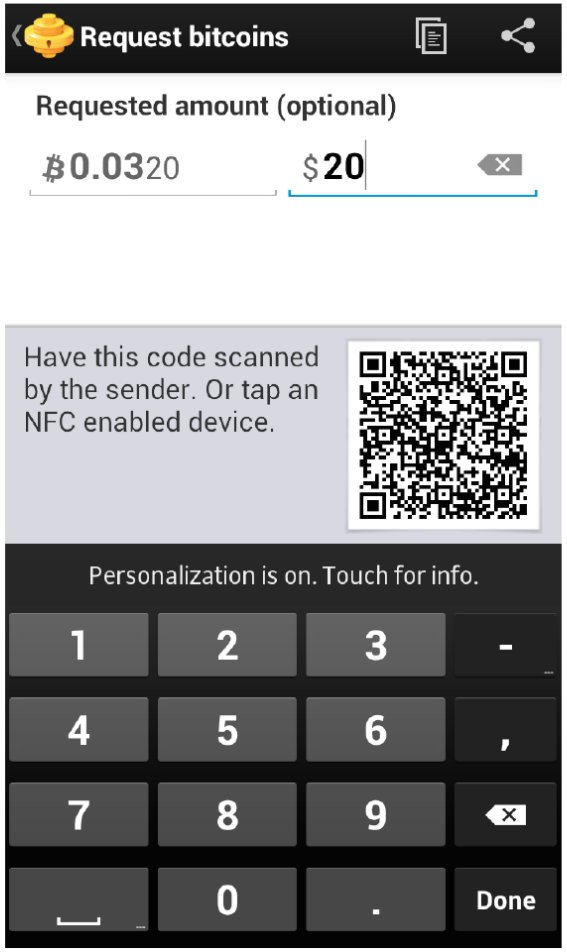


Map shows concentration of reachable Bitcoin nodes found in countries around the world.

LIVE MAP

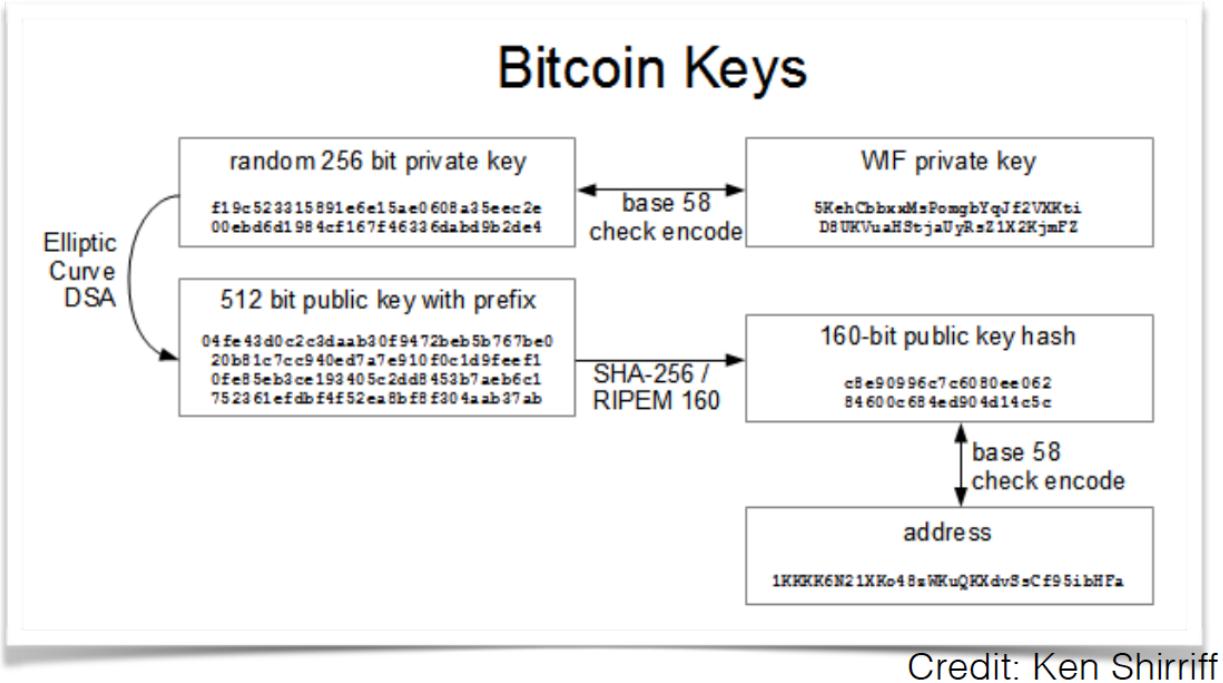
Bitcoin wallets

- You don't need to mine or run full node to use Bitcoin
- Wallet are applications that permit easy management of a Bitcoins.
- What's going on under the hood?



Bitcoin wallets: Under the hood

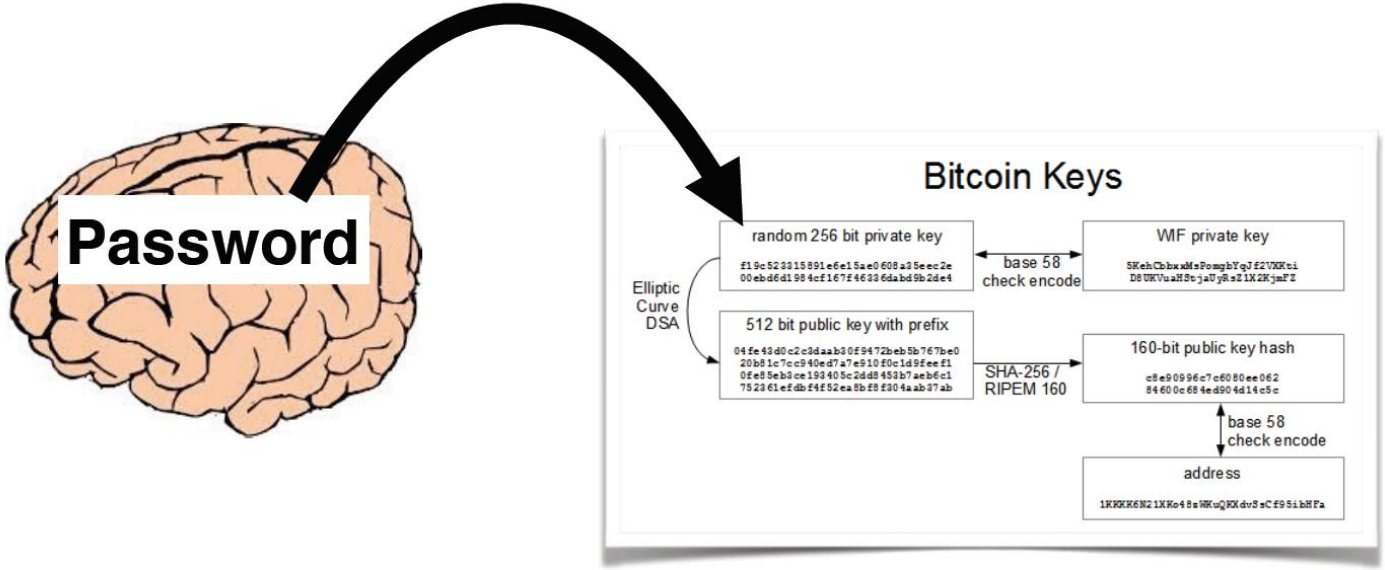
- Remember: identity associated with ECDSA digital signature key pair
  - *SK* used to sign / authorize transactions.
  - *PK* used to identify users and verify transactions.
- Bitcoin wallet stores, protects, and allows use of *SK* to make transactions.



Credit: Ken Shirriff

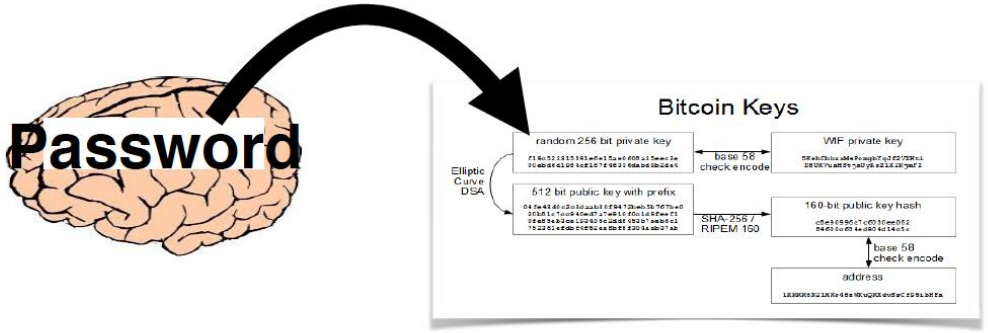
Brain wallets

- You can generate *SK* from a password



- Your Bitcoin are then completely portable.

Brain wallets



- Unfortunately, human brains are poor password stores…
- Cracking brainwallets at one point rumored more profitable than mining…

Finders keepers? I found an address with 50 BTC via brain wallet!
January 18, 2014, 04:58:04 PM #1

I was playing around with the brain wallet and checking the addresses with blockchain. I found a wallet with a balance of 50 BTC! The coins were put in the wallet in 2011 and there hasn't been any activity since. I don't want to steal someones coins but if they are "lost" I don't want to have them just sitting there. It's a lot of money! I was thinking of sending a small amount into the wallet with a message letting the person know the situation. If nothing happens after a while I guess it's "Finders Keepers, Losers Weepers." What's the right thing to do in this situation?

Related references

- Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan , Joshua A. Kroll, and Edward W. Felten. “SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies”, in Proc. of IEEE S&P 2015.
- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder. “Bitcoin and Cryptocurrency Technologies”, in Princeton University Press, 2016 (first two chapters)
- Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”
- Bitcoin Wiki, online at https://en.bitcoin.it/wiki/Main_Page
- Maurice Herlihy. “Blockchains from a Distributed Computing Perspective ”, 2018

Playground demo

- Start from hash:
  - https://andersbrownworth.com/blockchain/hash
- Till tokens:
  - https://andersbrownworth.com/blockchain/tokens