Bitcoin – part 1

CS 161: Computer Security

Prof. Raluca Ada Popa

April 27, 2020

Announcements

- Started recording
- TA Nick on chat
- Project 3 part 2 is released and will be due on **May 3 at 11:59 pm**

What is Bitcoin?

- Bitcoin is **a cryptocurrency**: a digital currency whose rules are enforced by cryptography and not by a trusted party (e.g., bank)
- **Core ideal:** avoid trust in institutions (e.g., banks, governments)
  - Reasons: Ideological, financial (avoid fees), peudo-anonymity
- Bitcoin is also a **ledger.** Its protocol is built on a technique called a blockchain, which has applications beyond Bitcoin
- Created by Satoshi Nakamoto, an anonymous identity, in 2009

Satoshi Nakamoto

- Wrote beautiful white paper on Bitcoin, in the syllabus
- No one knows who he is, online presence only
- Name stands for clear/wise medium; most likely not Japanese, but pseudonym
- He is very rich! [But hasn't changed yet]

Cryptocurrencies have supporters and opposers

Supporters say…

- No need to trust or depend on banks or the government
- Low transaction fees when transferring
- Helps disadvantaged areas (due to the fee but also because you don’t need a bank account or official identity)
- Anybody can partake
- Cross nation trading easier
- Transparent
- Anonymity
- “Digital gold”

Digital gold

- Used to be thought to replace cash but that seems less likely today
- People associate it to digital gold, a way to invest in a currency that does not belong to one government

Will do better in the pandemic?

Critics say…

- A major criticism is that it brings waste (e.g., proof of work, many large copies of blockchain)
- Not as decentralized as we wished
- Not scalable
- Market fluctuations
- Anonymity
- No security in case of loss
- Helps criminals
- Not a replacement for cash, not all vendors accept it

Why I am excited about it…

- I think they have brought about some very interesting and creative techniques at the intersection of cryptography and systems, and stirred much innovation in the field beyond Bitcoin and blockchains (e.g., smart contracts, consensus protocols, ledgers like Certificate Transparency)
  - I think the Bitcoin protocol is a strike of genius, because of the very creative way of combining different techniques.
  - You can understand the core of it using what you learned in class.

- They also increased the public's awareness towards the power of cryptography

Bitcoin technical design

Let’s work it out together!

Replacing banks

"IN BANKS WE DISTRUST"

Basic notions a bank provides:

- Identity management
- Transactions
- Prevents double spending

How can we enforce these properties cryptographically?

Two components

1. **Ledger:**
    1. publicly-visible,
    2. append-only, and
    3. immutable,

    log
2. **Cryptographic transactions**

Cryptographic transactions

- For now, assume the existence of a trusted ledger (append-only, immutable, everyone can see what is on it)

Identity

Q: How can we give a person a cryptographic identity?

- Each user has a PK and SK
- User referred to by PK

Transactions

Q: How can Alice transfer 10 ฿ (bitcoins) to Bob in a secure way?

- Idea: Alice signs transaction using her SK_A
- $sign_{SK_A}$("PK_A transfers *10 ฿ to PK_B*")
- Anyone can check Alice intended the transaction

Q: Problems?

- Alice can spend more money than she has. She can sign as much as she wants.

Q: Ideas how to solve this still assuming a "trusted ledger owner"?

Include only correct transactions in the public ledger

- **For now only**: assume there is a trustworthy ledger owner, assume initial budgets for each PK

Q: how would you prevent double spending?

- Assume all signatures/transactions are sorted in order of creation; include previous transaction where money came from

$TX = (PK_{sender}\text{->}PK_{receiver};X\ ฿;\quad PK_{sender}\text{->}PK_{sender};R\ ฿;$
list of transactions L where money came from)

time

Initial budgets: PK_A has 10 ฿	$TX_1 = (PK_A\text{->}PK_B;10\ ฿;$ from initial budgets) $sign_{SK_A}(TX_1)$	$TX_2 = (PK_B\text{->}PK_C;5\ ฿;$ $PK_B\text{->}PK_B;5\ ฿;$ from TX_1) $sign_{SK_B}(TX_2)$

How does the ledger owner check a transaction?

Verify TX:

1. The signature on TX verifies with the PK of the sender
2. The transactions in L have PK of sender as their recipient
   (that is, the sender receives Bitcoins in the transactions in L)
3. The transactions in L have not been spent before by sender
   (each transaction A->B can only be spent once by B, and once by A if there were remaining bitcoins in it)
4. Sender had X+R Bitcoins in L: the sum of the amounts received in the transactions in L total to X+R.

Two components

1. **Ledger:**
    1. publicly-visible,
    2. append-only, and
    3. immutable,

    log
2. **Cryptographic transactions**

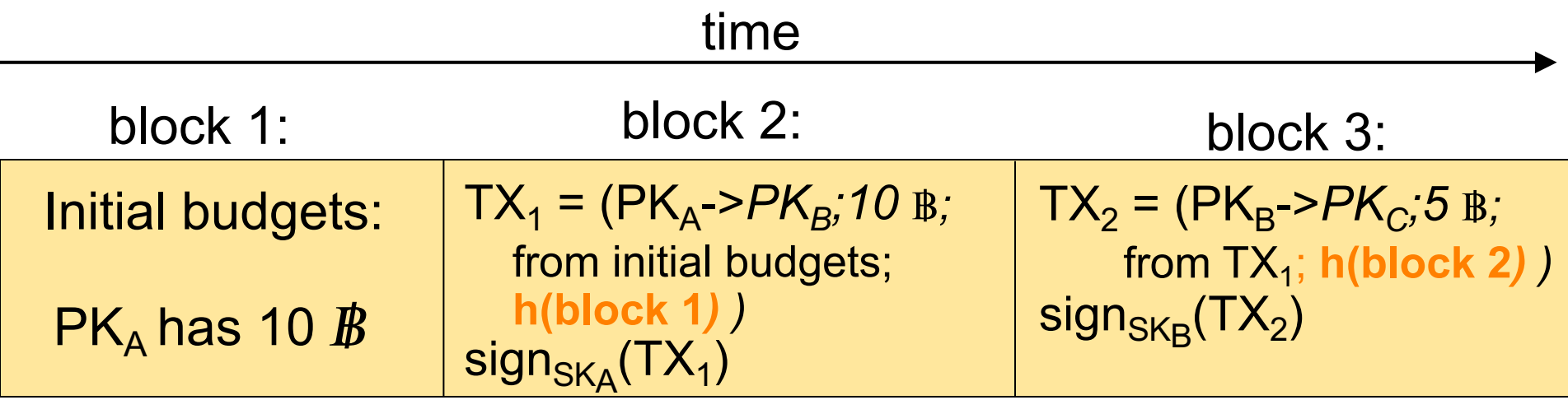
Bitcoin's ledger

1. Hash chain / blockchain

2. Consensus via proof of work

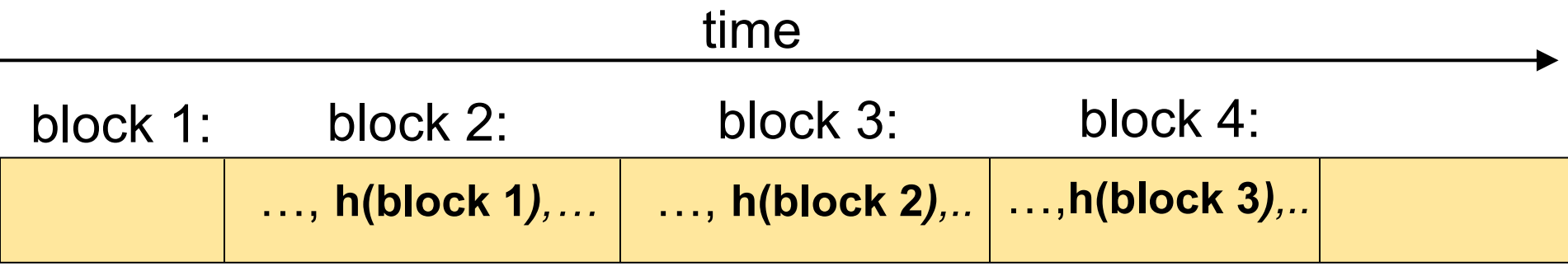
Blockchain

- Chain transactions using their hashes => hashchain
- Each transaction contains hash of previous transaction **(which contains the hash of its own previous transaction, and so on)**



block i refers to the entire block (transaction description and signature), so the hash is over all of this

Properties of the hashchain

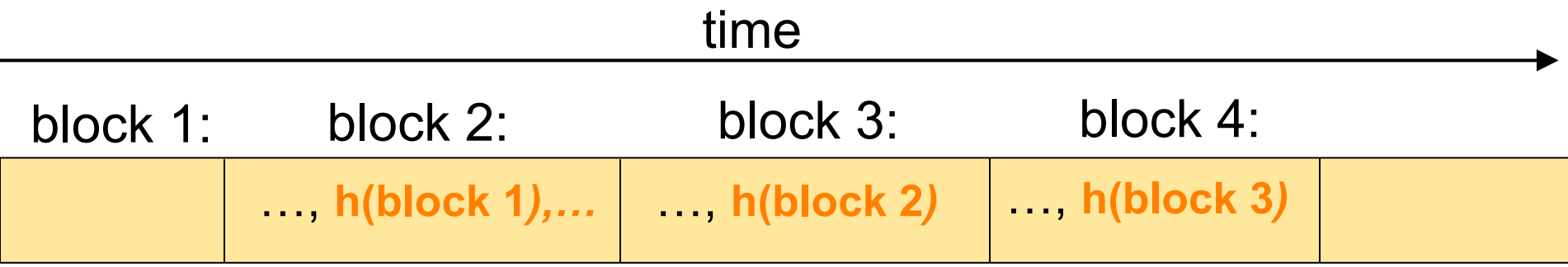


Given h(block i) from a trusted source and all the blocks *1 … i* from an untrusted source, Alice can verify that blocks 1 … i are not compromised using h(block i)

Q: How?

A: Alice recomputes the hashes of each block, checks it matches the hash in the next block, and so on, until the last block, which she checks it matches the hash from the trusted source

Why can't attacker cheat?

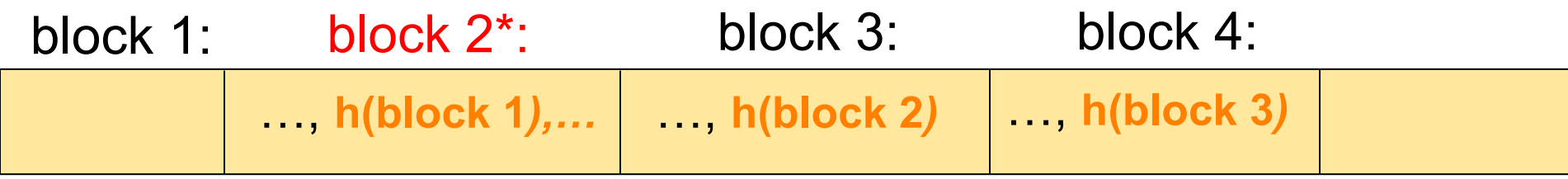


Say Alice obtains h(block 4) from somewhere **trusted**

She fetches the entire blockchain from a compromised server.

Q: Why can't the attacker give Alice an incorrect chain?

Say block 2 is incorrect.



A: because the hash is collision resistant

She fetches the entire blockchain from a compromised server.

Q: Why can't the attacker give Alice an incorrect chain?

Say block 2 is incorrect.

block 1:	block 2*:	block 3:	block 4:	
	…, h(block 1),…	…, h(block 2)	…, h(block 3)	

- If block 2* is incorrect, then hash(block 2*) ≠ hash(block 2)
- Then the third block is block 3* ≠ block 3 because it includes hash(block 2*)
- So hash(block 3*) ≠ hash(block 3)
- Then the fourth block is block 4* ≠ block 4 because it includes hash(block 3*)
- So hash(block 4*) ≠ hash(block 4)
- Hence, the hash of the block chain from the server will not match the trusted hash, detecting misbehavior
- If the hash does match, the attacker supplied the correct block chain

In Bitcoin:

- Every participant stores the blockchain
- There is no central party storing it
- When someone wants to create a new transaction, they broadcast the transaction to everyone
- Every node checks the transaction, and if it is correct, it creates a new block including this transaction and adds it to its local blockchain

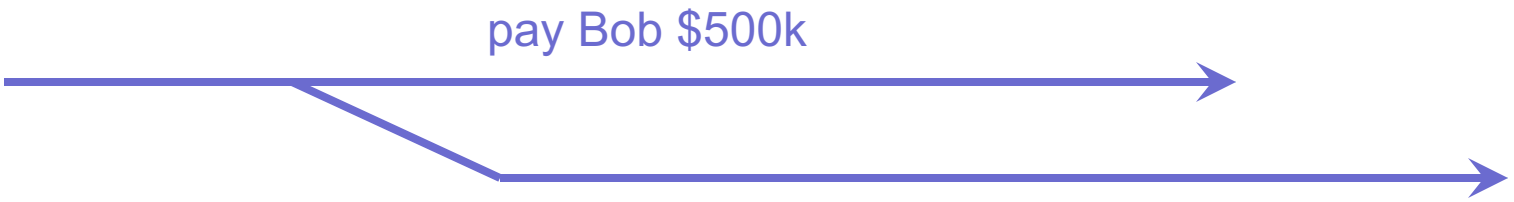
- Some participants can be malicious
- The majority are assumed to be honest

Why is the hash chain not enough?

- People can choose to truncate blockchain or not include certain transactions
- So we need a way for everyone to agree on the content of the blockchain: consensus

Example

- Mallory can fork the hash chain
- Say she buys Bob's house from him for $500K in Bitcoins. Then, she goes back in time and, starting from the block chain just before this transaction was added to it, she starts appending new entries from there. Can she get others to accept this forked chain, so she gets her $500K back? Yes.



Bitcoin's ledger

1. Hash chain / blockchain

2. Consensus via proof of work

Next lecture!