Lab 6: Bit-Coin Puzzles and Cryptographic Hash

Monday, October 22

CompSci 531, Fall 2018

Credits: Some slides taken from Bruce Maggs' CompSci 590 course in Spring 2018. Those slides are themselves based in part on Based on a Bitcoin Tutorial presentation by Joseph Bonneau, from Princeton University.

Outline

- Bitcoin and cryptocurrencies at a High Level
- Preview: Public Key Cryptography
- Review: Cryptographic Hash Functions as Proof of Work
- How bitcoin works

Bitcoin Value

Bitcoin and Cryptocurrencies

- Bitcoin isn’t a single thing. It is a currency, a payment system, a lot of cryptographic algorithms, and software implementations.

- The goal of bitcoin is to enable “trustless” payments with low transaction costs in an “anonymous” distributed network.

- What does that mean?

A Brief History of Currency

- Two people might want to *trade* with one another.

- In which case we don't need currency.

A Brief History of Currency

- But in reality, the situation is more complicated. What if you want something someone else has, but don’t have anything to trade that they want?

- Idea: a universal good, that exists in limited quantities, that can be traded for anything. Currency!

- Originally gold or silver. Today, things are a little different.

Online Purchase

Cryptocurrency Problem

- How do we accomplish the following transaction:
    - Without any trusted parties (cryptography), and
    - Without any “hard money” being moved around, while still guaranteeing that only amazon has the money afterward? (Cryptographic hash as proof of work and the blockchain).

Outline

- ~~Bitcoin and cryptocurrencies at a High Level~~
- Preview: Public Key Cryptography
- Review: Cryptographic Hash Functions as Proof of Work
- How bitcoin works

Preview: Public Key Cryptography

- In class tomorrow, you will dive into how to design and analyze these algorithms. For our purposes today, we just need to know what they do and guarantee.

- We want to be able to send secure message (encryption), and to be able to prove that the person who claims to have sent a message really sent it (digital signature).

- The challenge for public key cryptography is how to do this *without* having to exchange a secret key beforehand (that would be *private* key cryptography).

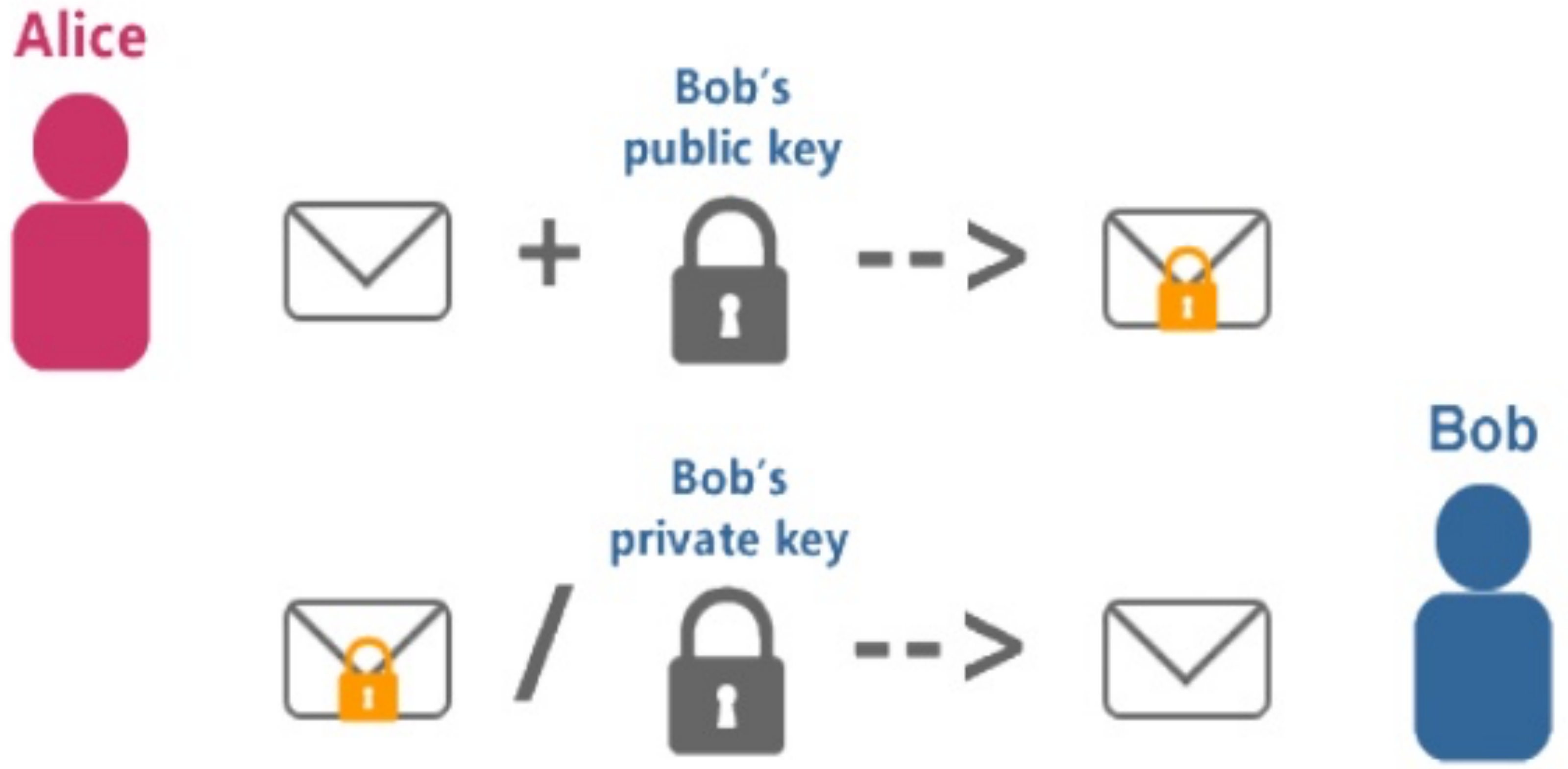
Preview: Public Key Cryptography

- Alice and Bob have public keys P_A, P_B and private keys S_A, S_B, which you should think of as 4,000 bits.

- *Everyone* knows the public keys (just post them "in the clear"), but only Alice knows S_A, and only Bob knows S_B. Let M be a message that Alice wants to send to Bob. She encrypts M with Bob's public key, and then Bob decrypts with his private key.

- We want two functions, both of which are easy to compute:
  - Encrypt(M, P)
  - Decrypt(M, S)
- Such that Decrypt(Encrypt(M, P_B), S_B) == M

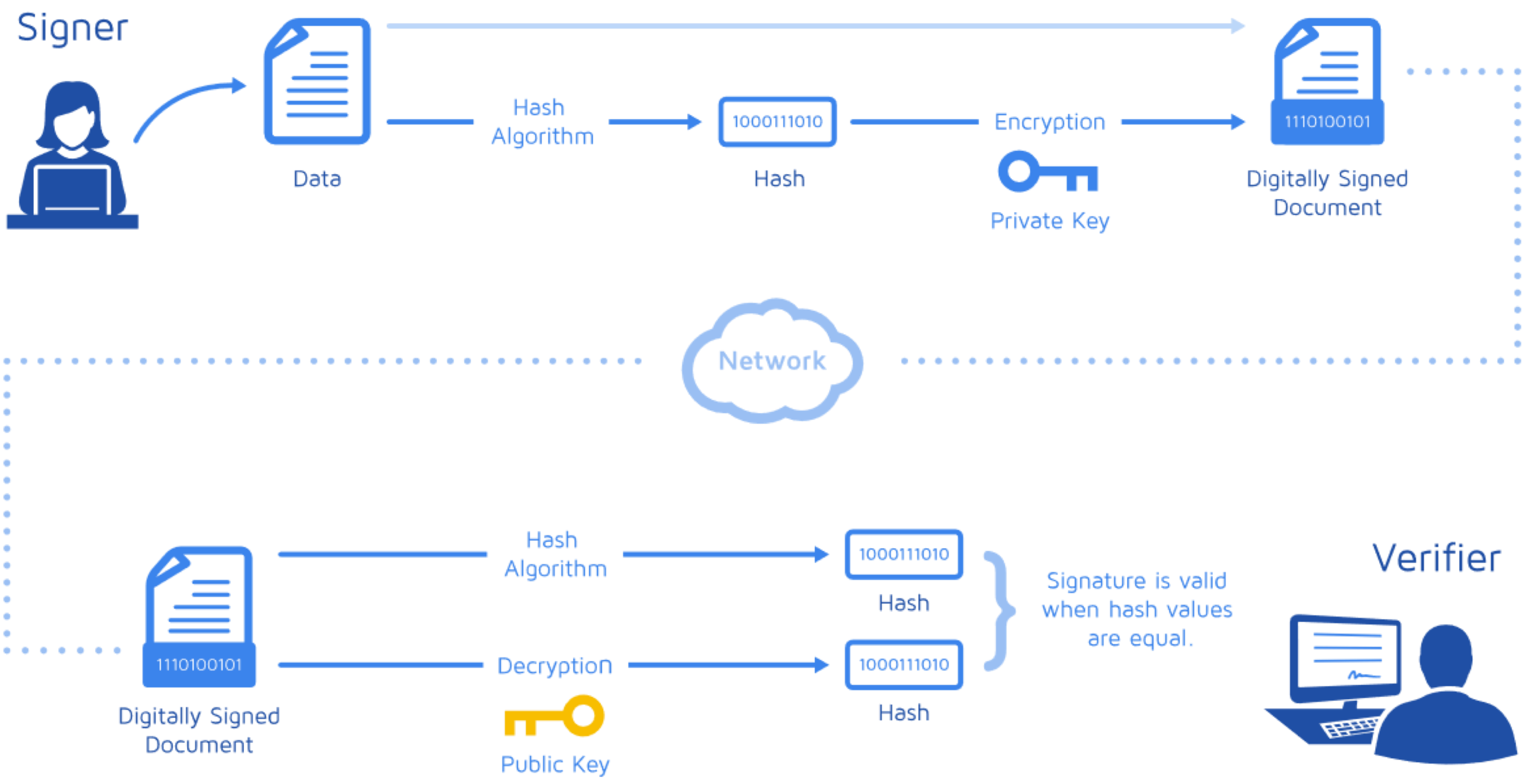
Working of RSA



Preview: Public Key Cryptography

- We can use the same basic idea for a *digital signature* scheme. Here, if Alice wants to prove to Bob that *she* sent the message, she can sign it using her private key, and he can verify with her public key.

- We want two functions, both of which are easy to compute:
  - Sign(M, S)
  - Verify(M, P)
- Such that Verify(Sign(M, S_A), P_A) == M


Signer
Data
Hash
Algorithm
1000111010
Hash
Encryption
Private Key
1110100101
Digitally Signed
Document
Network
1110100101
Digitally Signed
Document
Hash
Algorithm
1000111010
Hash
Decryption
Public Key
1000111010
Hash
Signature is valid
when hash values
are equal.
Verifier

Preview: Public Key Cryptography

- The crucial property for security is that while Encrypt/Decrypt and Sign/Verify are all highly efficient functions, computing their *inverses*, is computationally challenging (in RSA, it requires you to factor large numbers).

- These schemes will allow us to perform bitcoin transactions in a decentralized way, without the need for so many trusted entities.

Outline

- ~~Bitcoin and cryptocurrencies at a High Level~~
- ~~Preview: Public Key Cryptography~~
- Review: Cryptographic Hash Functions as Proof of Work
- How bitcoin works

Cryptographic Hash Functions

A related idea we need is that of a cryptographic hash function, that is, a hash function that has the following properties:

1. Deterministic (same message always gives the same hash)
2. Efficient (computationally)
3. Extremely difficult to reverse engineer the input from the output.
4. Small change to input → large change in output
5. Extremely difficult to find two input with the same output.

Proof of Work

- The idea of proof of work was introduced in the 90’s, originally with anti-spam applications.

- In order to allow a transaction to go through, you give give a cryptographic puzzle consisting of some input x and a cryptographic hash function h(). To solve the puzzle, one must find a number called a *nonce*, such that h(x+nonce) == 0.

- The properties of a cryptographic hash function ensure that (practically speaking), the only way to find such a nonce is brute force search.

Outline

- ~~Bitcoin and cryptocurrencies at a High Level~~
- ~~Preview: Public Key Cryptography~~
- ~~Review: Cryptographic Hash Functions as Proof of Work~~
- How bitcoin works

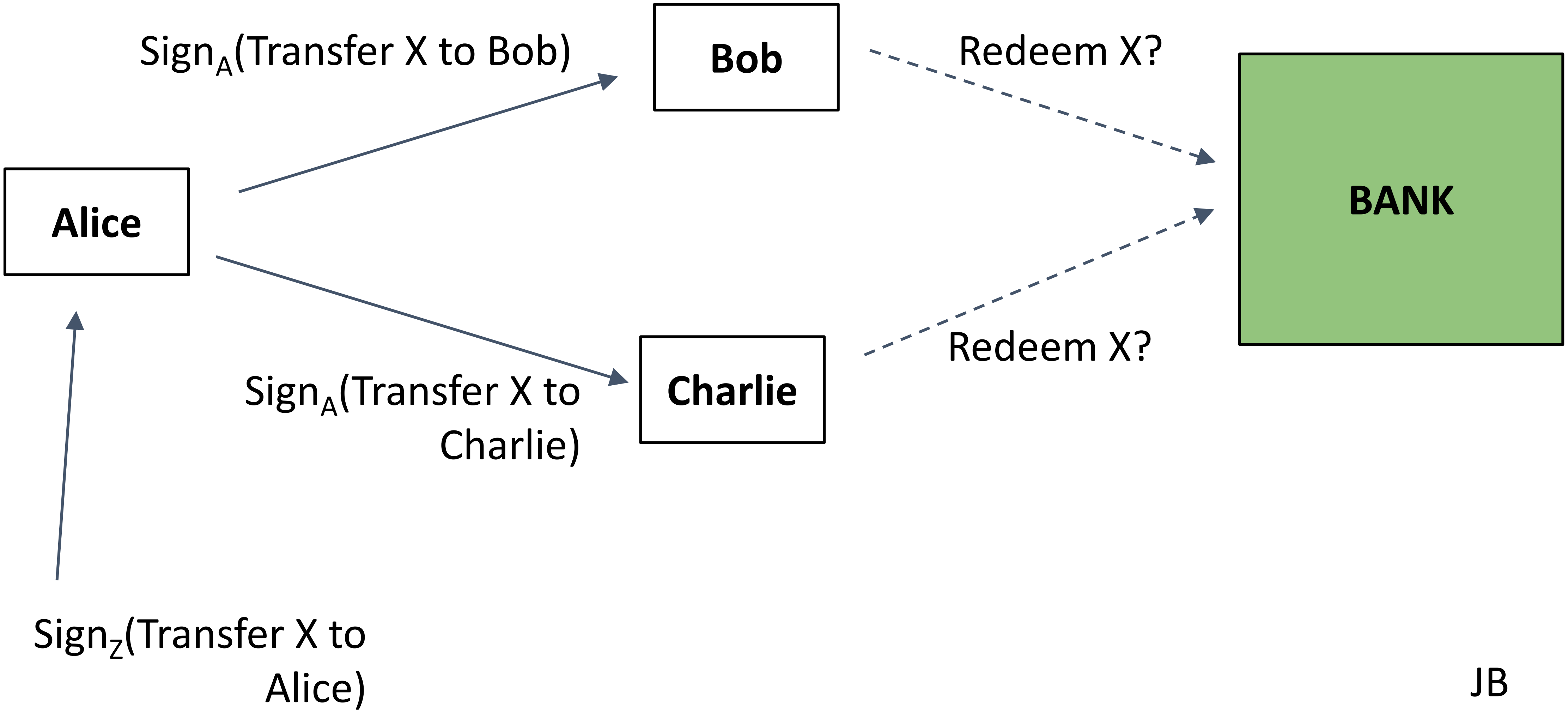
How bitcoin works (roughly)

- The generation/amount of currency is limited by the ability to (approximately) compute the inverse of hash functions as proof of work.

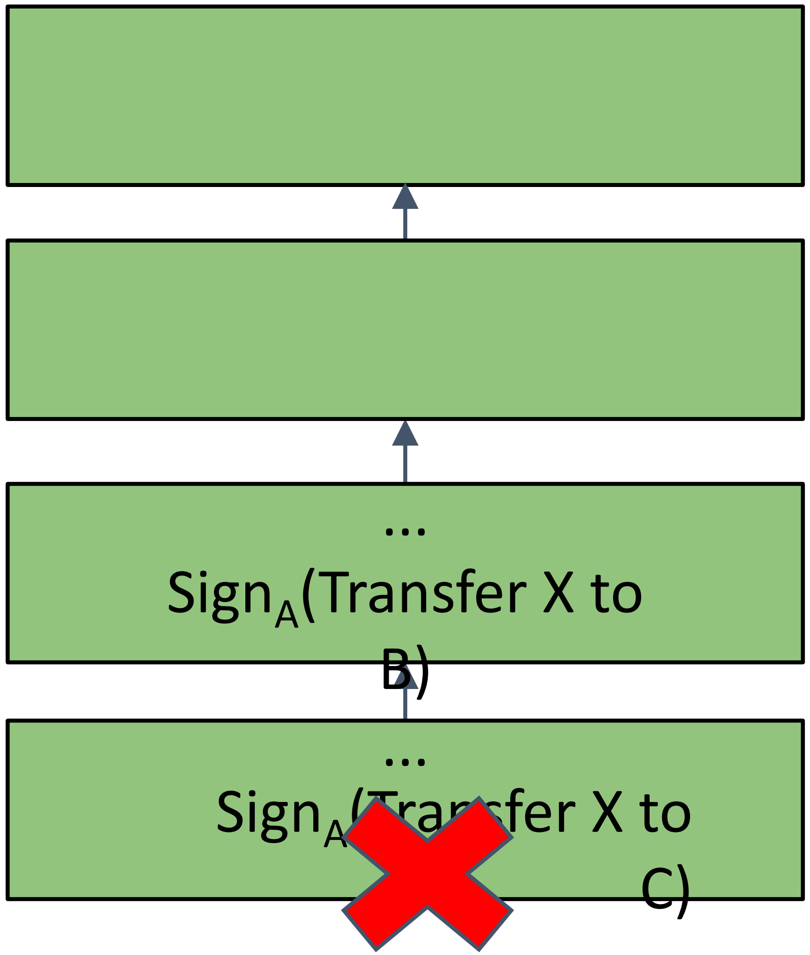
- Once you have a bitcoin, you can send bitcoins to someone in a transaction by digitally signing.

- But wait, what *is* a bitcoin really? And why can’t I just tell multiple people that I’m sending them bitcoin at the same time?

Double spending: why ecash is hard



Solution: Maintain a global public append-only log



The block chain –a public ledger of all transactions.

(In Bitcoin, the log is extended in increments of blocks, each of which may contain thousands of transactions.*)*

bmm

Spending a Bitcoin

- A transaction is of the form “send these Bitcoins from address Y to address Z”
- Specific Bitcoins are described as outputs of previous transactions.
- The transaction is signed with the private key of address Y and broadcast, along with the public key of Y, to the payment network
- A transaction might also include a transaction fee.

Bitcoin mining

- Approximately every ten minutes, one lucky Bitcoin miner earns a reward for extending the block chain by one block.
- Mining reward: 12.5 Bitcoin
- Mining is the only mechanism for creating new bitcoins. The total number of Bitcoins will never exceed 21M. (Bitcoins in circulation: https://blockchain.info/charts/total-bitcoins)
- The rewarded miner also receives all (optional) transaction fees in the block.

How is a new block created?

- A Bitcoin miner creates a block by

(1) Gathering a set of pending transactions, prioritizing those with transaction fees
(2) Verifying the transactions
(3) Gives the reward and transaction fees to himself/herself
(4) Solving a hashing problem

How is a transaction verified?

- “send these Bitcoins from address Y to address Z”

- The miner first checks the signature using the public key for address Y.
- compute hash of public key for Y, which should be Y
- check signature of transaction using public key for Y

- Then the miner checks the public ledger to verify that Y hasn’t already sent these Bitcoins to someone else.

The Hashing Problem

- To extend the blockchain, a miner creates a new block, which has a block header. The block header contains:

(1) block version number
(2) SHA-256 hash of previous block header
(3) SHA-256 hash of new transactions to include in the blockchain, including creation of reward bitcoins (e.g., 12.5 new BTC)
(4) current target / difficulty
(5) timestamp
(6) nonce

- Block is valid if SHA-256(SHA-256(header)) leads in enough zeros, as determined by current difficulty. Miner has to find the right nonce by trial and error!

- Difficulty chosen so that the time until the first miner wins is about ten minutes, on average.

Target and Difficulty

- A miner can win if his/her hash value is below the current 256-bit target, i.e., the hash value has enough leading zeros.

- Probability that a given nonce will produce a winning hash value is target / 2^{256}

- difficulty = difficulty_1_target / target, where difficulty_1_target = 2^{224}

- Expected time (in seconds) to mine a block = 2^{256} / (target * hashrate)
- = difficulty * 2^{32} / hashrate

- Difficulty was 440,779,902,286 on March 3, 2017.

- Difficulty is adjusted every 2016 blocks. If a new block is added in ten minutes, 2016 blocks are added in exactly two weeks.

Conclusion

- Bitcoin is a complicated system of currency, payment, algorithms, and software. There are abundant technical, legal, and ethical dilemmas.

- But it fundamentally rests on the power of public key cryptography and cryptographic hashing.