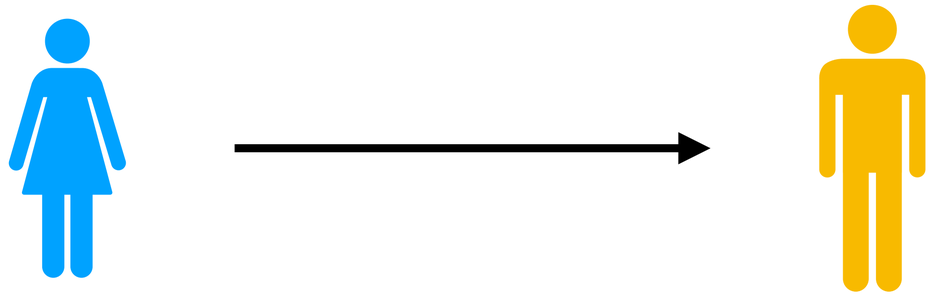
Bitcoin & Cryptocurrency

Material taken from here, here

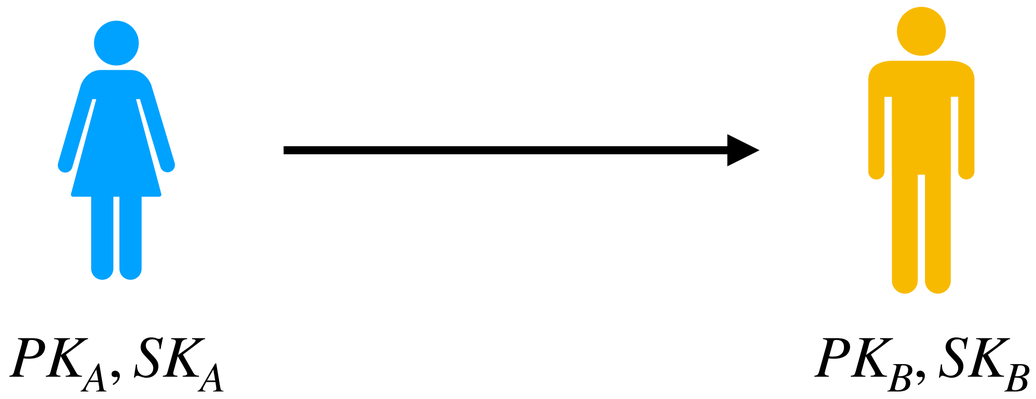
How to design a digital currency?

- Alice wants to digitally send some money to Bob
- Need identities
  - Each user has a public-private key pair
  - Each user referred to by their public key



How to design a digital currency?

- Alice simply signs a message saying that she is giving Bob a coin
- Good properties:
  - Publicly verifiable
  - Alice cannot renege
  - No one can forge this transaction



"I, Alice, am giving Bob one coin", σ

How to design a digital currency?

- Problem: what if Alice keeps on sending the same message over and over again, even if she doesn't have enough coins?
  - Double spending
- Coins need to be unique
- Banks can handle this as a centralized authority using a ledger

"I, Alice, am giving Bob one coin", σ

"I, Alice, am giving Bob one coin", σ

"I, Alice, am giving Bob one coin", σ

How to design a digital currency?

- Ledger keeps track of initial budgets, and all transactions
- Ledger owner checks that
  - Signature verifies for sender with PK
  - Transaction is indeed sending from a sender with the corresponding PK
  - Checks for any double spending

"I, Alice, am giving Bob one coin", σ

PK_A: 5 coins PK_B: 0 coins	$TX_1 = (PK_A \rightarrow PK_B, 1)$: 2 coins $\sigma_{SK_A}(TX_1)$	$TX_2 = (PK_A \rightarrow PK_B, 1)$: 5 coins $\sigma_{SK_A}(TX_2)$

Making everyone collectively the bank

- Everyone keeps a copy of the ledger using a blockchain
  - Each transaction contains a hash of the previous transaction
  - Given h(block i) from a trusted source, and preceding blocks from an untrusted source, can verify that the previous blocks are not compromised
- Updates need to be propagated to everyone
- Everyone checks the transaction, and if it's correct, creates a new block including this transaction and add to the local blockchain

PK_A: 5 coins PK_B: 0 coins	$TX_1 = (PK_A \rightarrow PK_B, 1)$: 2 coins $\sigma_{SK_A}(TX_1)$ h(block1)	$TX_2 = (PK_A \rightarrow PK_B, 1)$: 5 coins $\sigma_{SK_A}(TX_2)$ h(block2)

Making everyone collectively the bank

- Problem: Alice purchases an item from Bob for 100 coins. Then she creates a fork from the block just before the transaction, and she can create new blocks of transactions.
- Solution: proof-of-work
  - Make it computational costly for network users to validate transactions
  - Reward them for trying to help validate transactions

Proof-of-work

- Only miners can add blocks to the blockchain
- All miners try to solve a computationally hard (but solvable) problem: find a nonce such that $h(\text{block})$ starts with a number of zeros
  - Bitcoin protocol actually requires the hash to be less than or equal to a "target": every 2016 blocks, every Bitcoin client compares the actual time it took to generate these blocks and modifies the target
- Once a miner solves a proof-of-work, it creates a block that includes a set of received transactions
- Miner receives a free coin if mining is successful

Consensus

- Longest correct chain wins; everyone checks all blocks and all transactions
- Can Alice create a fork?
  - Bob waits for a number of blocks to be appended after the said transaction. If Alice does not have a majority of the computing power, then the probability that she can catch up drops exponentially.
- What happens if Alice and Bob are mining blocks, and both solve a proof-of-work and append two different blocks, creating a fork?
  - The next miner that appends onto one of these chains invalidates the other chain.

Consensus

- If a miner has included Alice's transaction in the latest block, is it guaranteed that her transaction is forever in the blockchain?
  - No, there could be a legitimate miner appending a different block, so it's safe to wait for a few blocks.
- What happens if a miner refuses to include Alice's transaction?
  - It's unfortunate, but there could be another miner that will do so. Each transaction includes a fee that goes to the miner so a miner is incentivized to include as many transactions as possible.

Bitcoin is anonymous?

- Silk Road: online black market
- SoK on this topic
  - Common input ownership
  - Change addresses
  - Transaction graph analysis
  - Side-channel information

Ethereum

Next time: guest lecture!

Title: Proof-of-stake blockchains and compact certificates

Abstract: In this lecture I will sketch the big ideas behind proof-of-stake blockchains, using Algorand as the running example. Then I will define a cryptographic primitive called a compact certificate and discuss how compact certificates can be used to take a snapshot of a proof-of-stake blockchain. At the end, I will show how these snapshots can be used both to help clients quickly catch up to the current state of the chain, and to implement cross-blockchain interoperation.