PRIVACY, SECURITY AND USABILITY

Blockchain

What is Blockchain?

- "To understand the power of blockchain systems, and the things they can do, it is important to distinguish between three things that are commonly muddled up, namely the bitcoin currency, the specific blockchain that underpins it and the idea of blockchains in general."

The Trust Machine,
THE ECONOMIST, Oct. 31, 2015

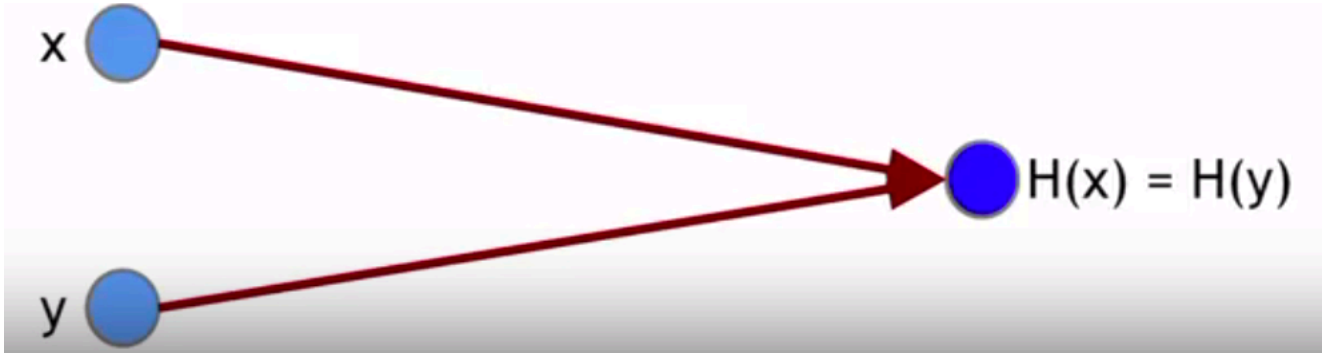
What is Blockchain?

- We may refer to different things:
  - The idea of blockchain
  - The specific blockchain that underlies Bitcoin or another coin offering
    - Ethereum, Hyperledger Fabric, etc.
  - Bitcoin or another cryptocurrency

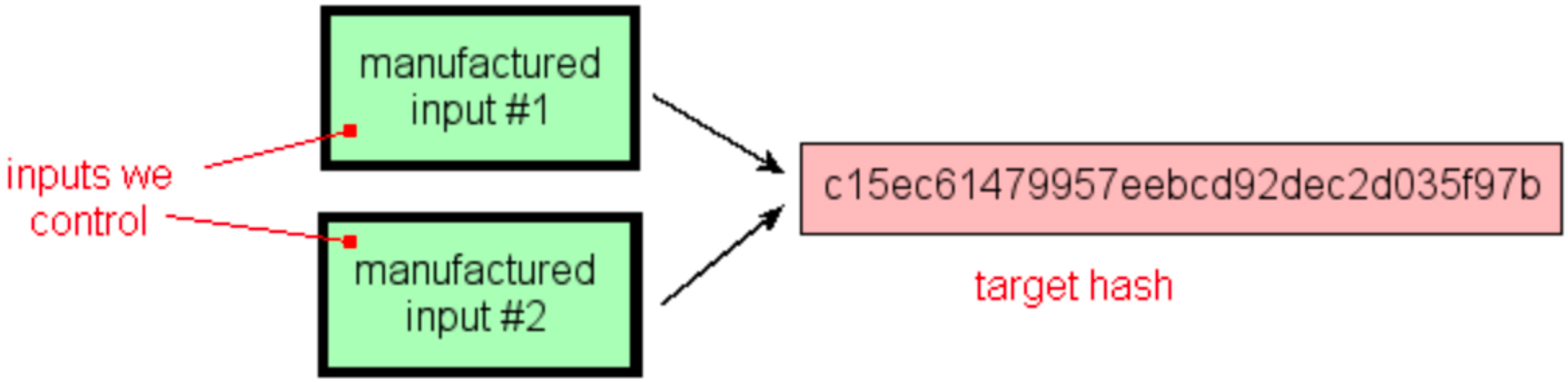
CRYPTO CONCEPTS

Cryptographic Hash Functions

- Takes a string as input
- Creates fixed size output
- Efficiently computable
- Collision resistant:
  - it is hard to find two inputs that hash to the same output
    - E.g. Find x,y, s.t.
      - $H(a) x H(y)$ AND $x != y$



Collision Resistant

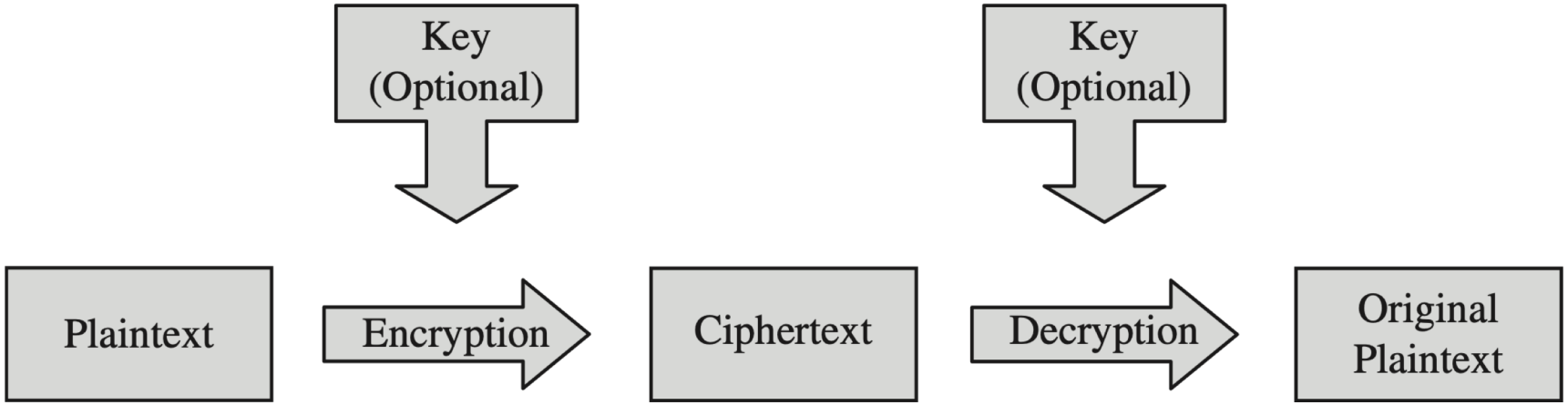


Collision Resistant

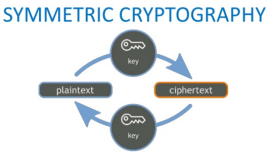
- Collisions do exist
  - Input size typically larger than hash size
    - => Number of possible inputs > number of possible outputs
- But computationally hard to find

PUBLIC KEY (ASYMMETRIC) ENCRYPTION

Encryption/Decryption Process



Reminder - Symmetric Cryptosystem

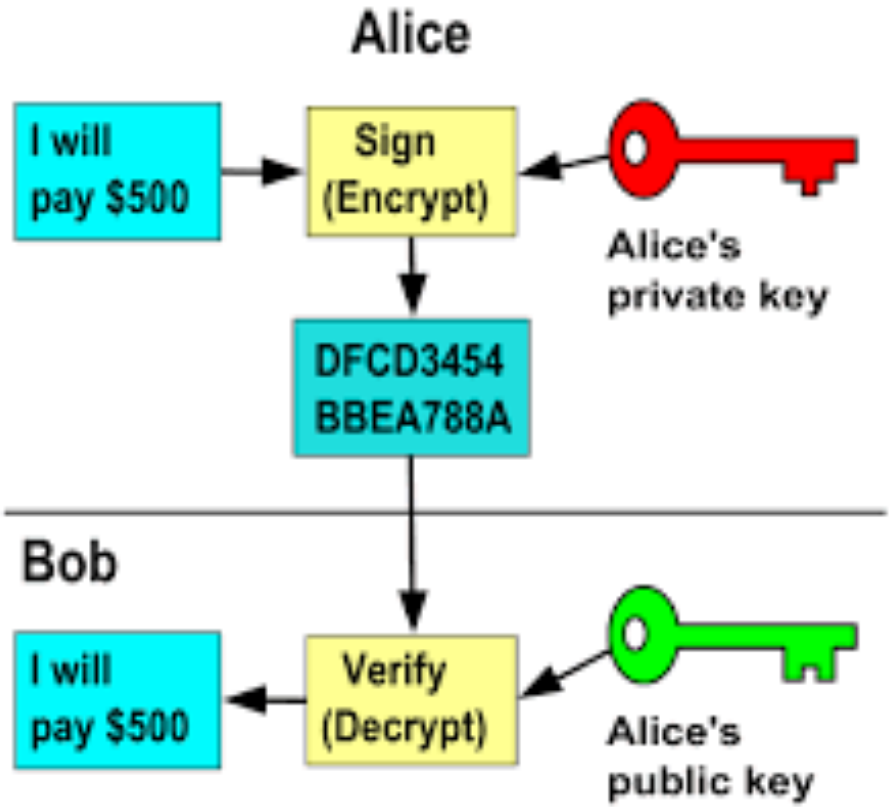


- Scenario
  - Alice wants to send a message (plaintext P) to Bob.
  - The communication channel is insecure and can be eavesdropped
  - If Alice and Bob have previously agreed on a symmetric encryption scheme and a secret key K, the message can be sent encrypted (ciphertext C)

Public Key (Asymmetric) Cryptography

- Instead of two users sharing one secret key, each user has two keys: one public and one private
- Messages encrypted using the user's public key can only be decrypted using the user's private key, and vice versa
  - P = D(Kpriv,E(Kpub,P)) or
  - P = D(Kpub,E(Kpriv,P))

Public Key Encryption



https://commons.wikimedia.org/wiki/File:Public_key_signing.svg

Public Key Encryption

- A cryptographic system that uses pairs of keys
  - public keys which may be disseminated widely
    - Any person can encrypt the message
  - private keys which are known only to the owner
    - Message can only be decrypted with this key
- Analogous to a self-closing door
  - Need key to open it
  - Closing it shuts it automatically

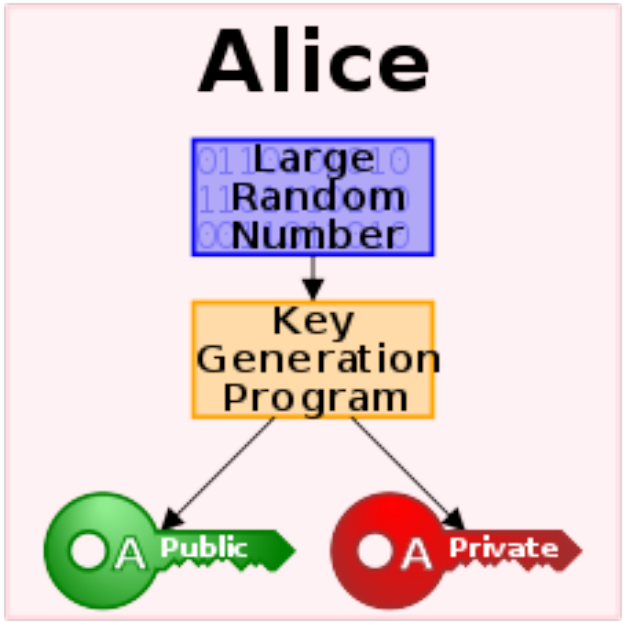
Public Key Encryption

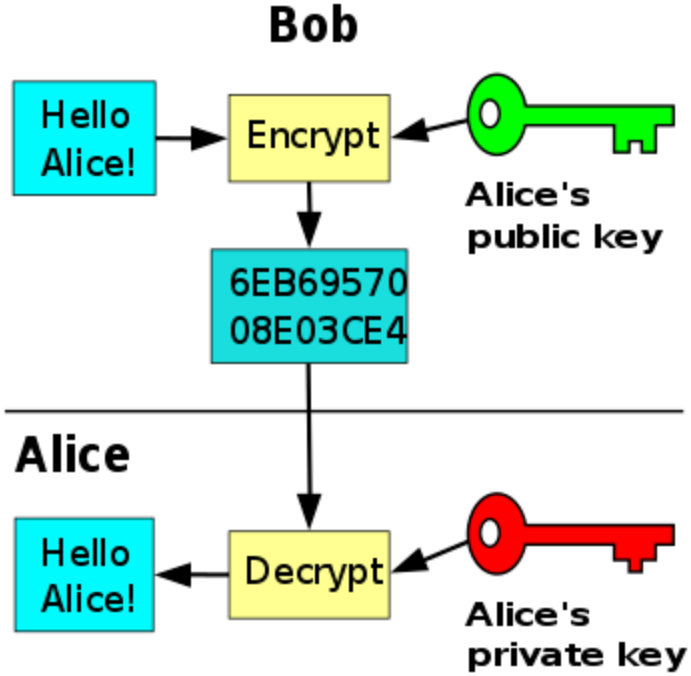
- Accomplishes two functions: authentication and encryption
- Authentication: the public key verifies that a holder of the paired private key sent message
  - Decryption will fail otherwise
- Encryption: only the paired private key holder can decrypt the encrypted message

Public Key Encryption

- Why does it work?
- It is not feasible to compute the private key
  - From knowledge of its paired public key
- Therefore, only the private key is kept private
  - The public key can be openly distributed without compromising security
- Public key cryptography relies on math problems that have no efficient solution

Public Key Encryption





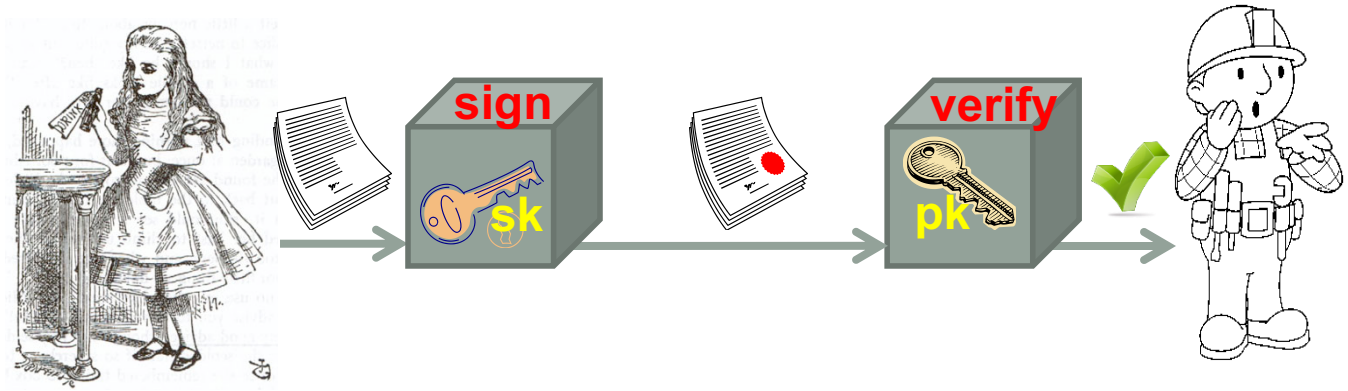
https://en.wikipedia.org/wiki/Public-key_cryptography

Public Key Encryption

- Often used to secure communication
  - Over the internet, open networks
  - Typically used for key exchange

Digital Signatures

- Alice wants to sign a document for Bob
  - She has a (secret, public) key pair
  - Bob knows Alice's public key



- A public verification procedure

$$Verify(pk, doc, sig) = Yes/No$$

- Can't generate signatures without secret-key

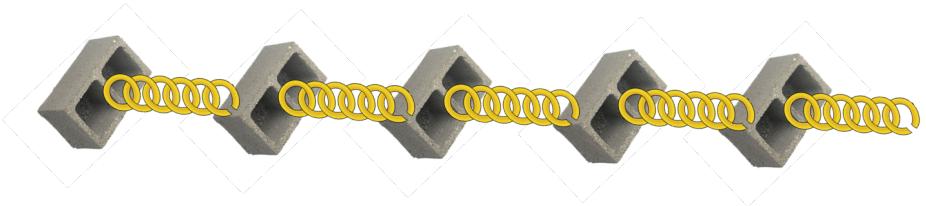
BLOCKCHAIN TECHNOLOGY AND PRIVACY

Challenges and solutions in distributed systems

What is a Blockchain?

- A list of records, called blocks, chained using cryptography
  - The entire list is called a ledger
- Each block contains:
  - Transaction data (the data that we really care about)
  - Timestamp
  - A cryptographic hash of the previous block
- Hashing ensures that once we agree on a block, all previous transactions are verified

What is a Blockchain?



- Participants use a *consensus protocol*, to agree on each new block which is added to the ledger
- Hence a blockchain is:
  - Distributed: used/modified by many different participants
  - Consistent: everyone sees the same ledger
  - Immutable: once a block is there, it never changes
  - Verifiable: anyone can check the above
- Without having to trust a single centralized entity

What is a Blockchain Good For?

- Recording transactions in an agreed chronological order

What is a Blockchain Good For?

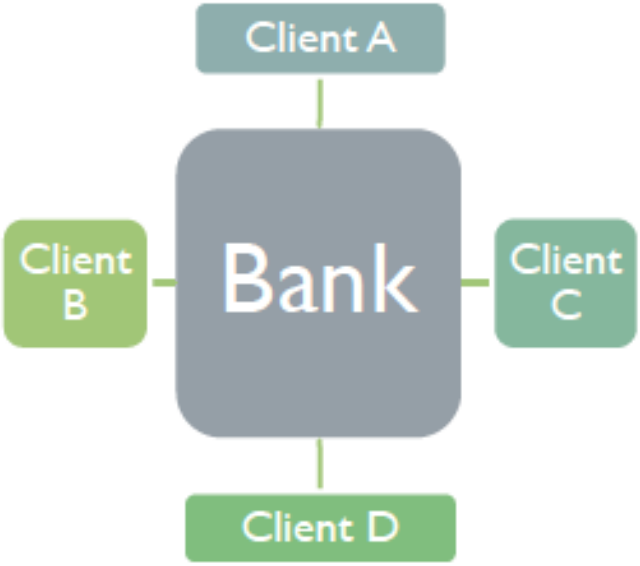
- Recording transactions in an agreed chronological order
- In a payment system, a trusted ledger is needed to keep track of who has how much money
  - Traditionally, we all trust a bank to ensure that if I buy something for $100, this $100 is deducted from my account
  - Cryptocurrency uses a blockchain for the same purpose
    - The ledger tells us how much money is in each account
- The ledger is a distributed database
  - No need for a single trusted entity, no single point of authority

What is a Blockchain Good For?

- Distribution provides extra security
  - Consensus ensures that it is hard to insert invalid data
  - By design, a blockchain is resistant to modification of past data
- The ledger is distributed among thousands of parties
  - Changing a transaction would require hacking all of them
- Resulting in both accountability and transparency
  - Suitable for many commerce applications (not just bitcoin)

What is a Distributed Ledger?

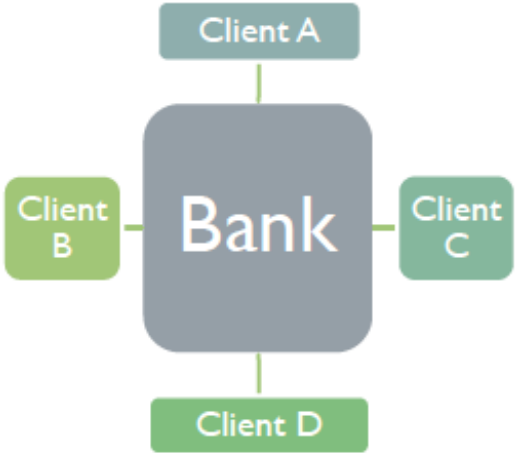
- Centralized Ledger:



https://www.sec.gov/spotlight/investor-advisory-committee-2012/slides-nancy-liao-brief-intro-to-blockchain-iac-101217.pdf

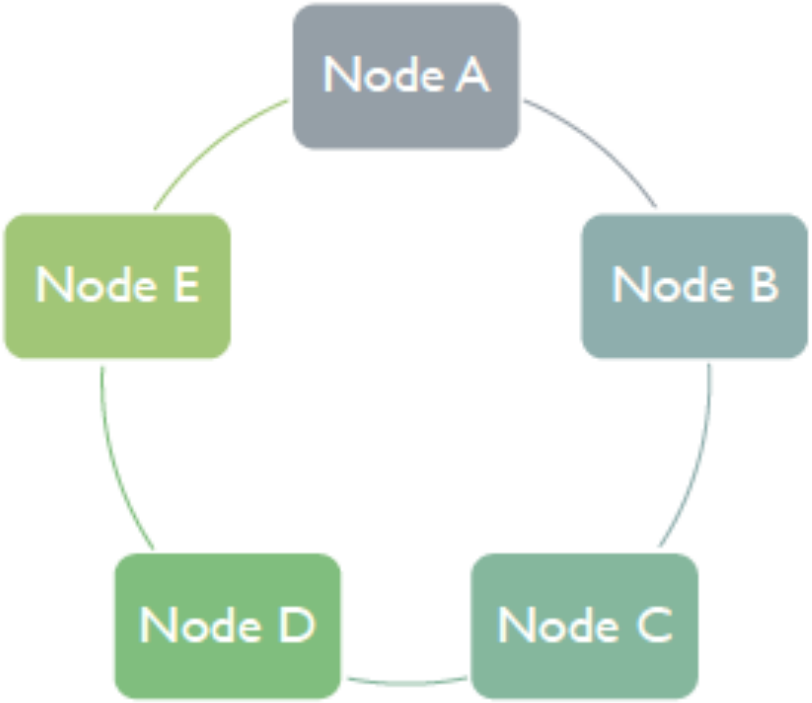
What is a Distributed Ledger?

- Centralized Ledger:
  - There are multiple ledgers, but Bank holds the “golden record”
  - Client B must reconcile its own ledger against that of Bank, and must convince Bank of the “true state” of the Bank ledger if discrepancies arise



https://www.sec.gov/spotlight/investor-advisory-committee-2012/slides-nancy-liao-brief-intro-to-blockchain-iac-101217.pdf

What is a Distributed Ledger?



https://www.sec.gov/spotlight/investor-advisory-committee-2012/slides-nancy-liao-brief-intro-to-blockchain-iac-101217.pdf

What is a Distributed Ledger?

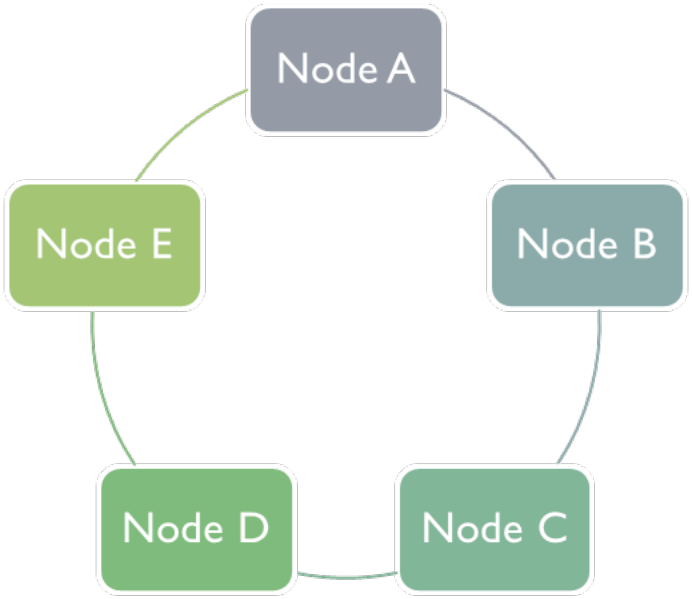
- There is one ledger
- All nodes have some access level to that ledger
- All nodes agree to a protocol that determines the “true state” of the ledger at any point in time.
- The application of this protocol is sometimes called “achieving consensus.”

What is a Distributed Ledger?

- We may have one or more entities
  - Each entity may have one or more nodes

What is a Distributed Ledger?

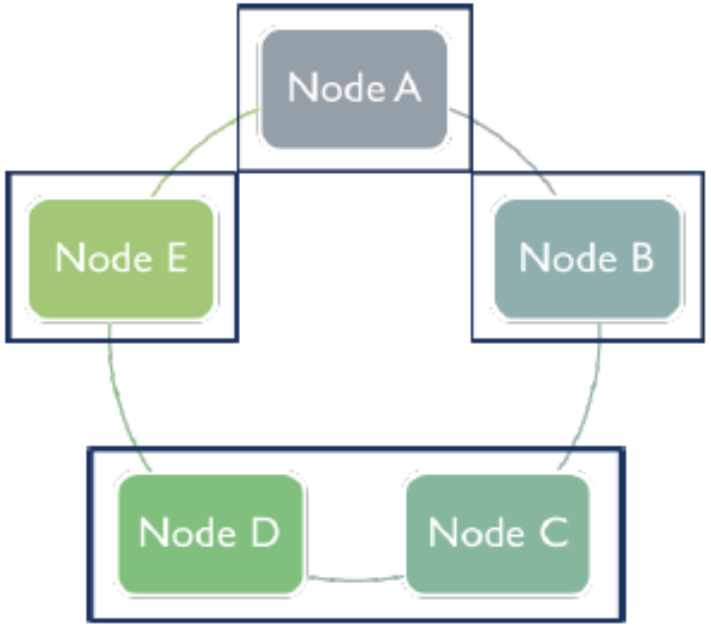
- Single Entity:



https://www.sec.gov/spotlight/investor-advisory-committee-2012/slides-nancy-liao-brief-intro-to-blockchain-iac-101217.pdf

What is a Distributed Ledger?

- Multiple entities:



https://www.sec.gov/spotlight/investor-advisory-committee-2012/slides-nancy-liao-brief-intro-to-blockchain-iac-101217.pdf

Nodes

- The role of a **node** is to support the network by maintaining a copy of a **blockchain** and, in some cases, to process transactions
- **Nodes** are often arranged in the structure of trees, known as binary trees.
- Each cryptocurrency has its own **nodes**, maintaining the transaction records of that particular token

Using a Distributed Ledger

- User initiates a transaction using her digital signatures
- User broadcasts her transaction to nodes
- One or more nodes begin validating transaction
- Nodes validate aggregated transactions into blocks
- Nodes broadcast blocks to each other
- Consensus protocol used to agree on next block
- Newly created block chained to previous blocks

Distributed Ledger Advantages

- Can be used without a trusted authority
- Can be used to create or transfer value or ownership
  - Can be used to record these transfers

Distributed Ledgers

- The degree of trust between users determines the technological configuration of a distributed ledger

The Consensus Problem

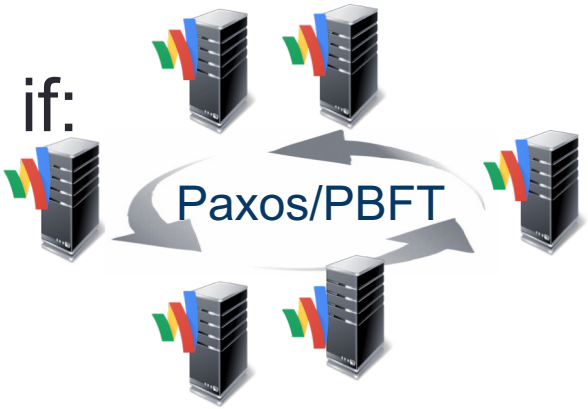
- This is the heart of any blockchain
  - How to reach agreement on the next block to add
- Consensus was studied in the literature since the 80's:
  - Multiple parties, each with its own value
    - In our case, "what I think the next block should be"
  - They want to agree on a single value to use
    - By sending messages to each other
  - Some of them may misbehave
    - Send the wrong messages, or not send anything at all

Traditional “permissioned” Consensus

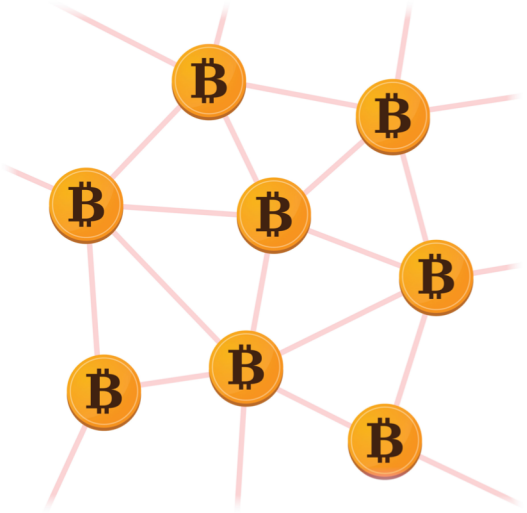
- A fixed set of parties, they all know each other
- The consensus protocol ensures:
  - Liveness: they will eventually finish the protocol
  - Consistency: once they do, they all finish with the same value

Traditional “permissioned” Consensus

- A fixed set of parties, they all know each other
- The consensus protocol ensures:
  - Liveness: they will eventually finish the protocol
  - Consistency: once they do, they all finish with the same value
- Traditional protocols ensure this, if:
  - only minority of them misbehave
  - “honest” participants can reliably send and receive messages

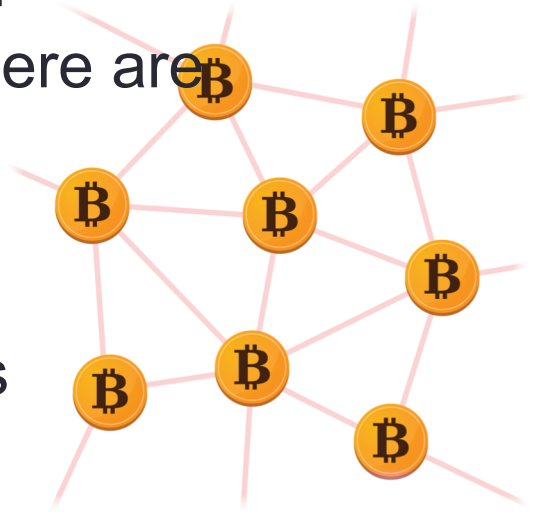


The new “Permissionless” Model



The new "Permissionless" Model

- No one knows all the participates
  - Not even how many of them there are
- Participants come and go
- ANYONE can join
  - No authentication mechanisms



- Calls for a very different type of consensus protocols

The new "Permissionless" Model

- Old consensus protocols cannot possibly work here
- The main issue: "Sybil attacks"
  - An attacker can create many different identities
  - Uses them to gain a disproportionately large influence

  ➔ Assuming "honest majority" is no longer justified

Nakamoto's Blockchain [Nak'08]

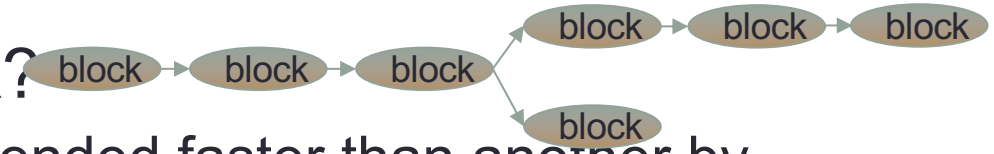
- Prevents Sybil attacks with Proofs-of-Work Puzzles [DN'92]
  - To propose a new block you must solve a puzzle
  - Puzzles are moderately hard to solve, very easy to check
  - A block must contain a solved puzzle to be added to the chain
- Instead of "honest majority", this mechanism assumes "honest majority of computing power"
  - The majority of computing power used to solve the puzzles is controlled by honest participants

Nakamoto's Blockchain [Nak'08]

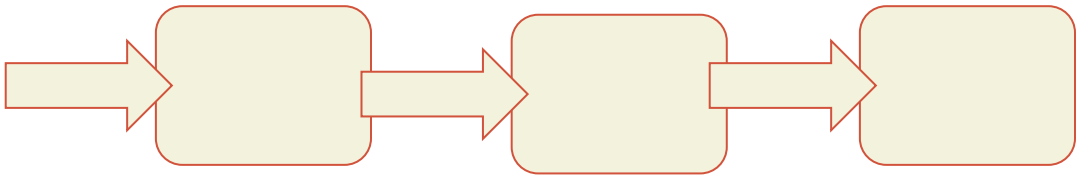
- How to handle multiple solved puzzles?
  - A puzzle is built on top of an existing chain
    - Includes in it the hash of the last known block
  - If you see multiple blocks with solved puzzles, choose the one which is built on top of the longest chain
  - Add the block, solve new puzzles on top of this longer chain

Nakamoto's Blockchain [Nak'08]

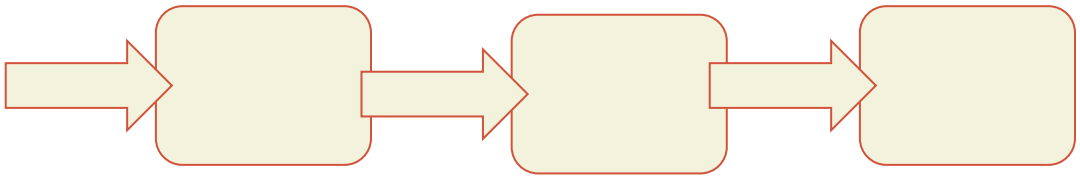
- How to handle multiple solved puzzles?
  - A puzzle is built on top of an existing chain
    - Includes in it the hash of the last known block
  - If you see multiple blocks with solved puzzles, choose the one which is built on top of the longest chain
  - Add the block, solve new puzzles on top of this longer chain
- What happens in a fork?

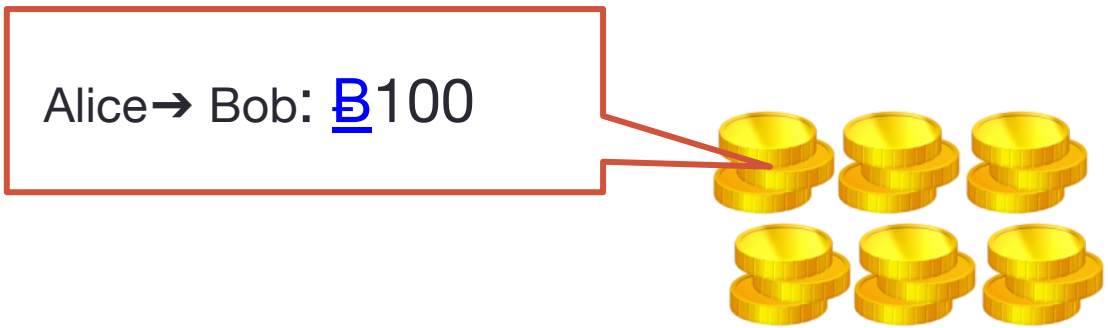


  - One extension will be extended faster than another by chance
  - Everyone will keep working on this one, the other will die out

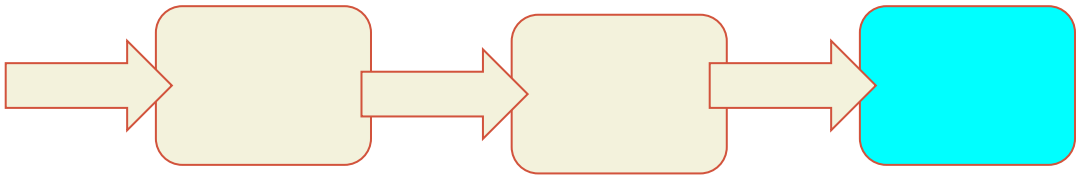


How to build a “blockchain”





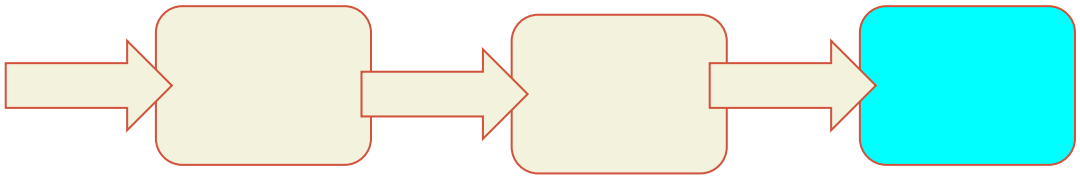
How to build a “blockchain”



"Hash function"

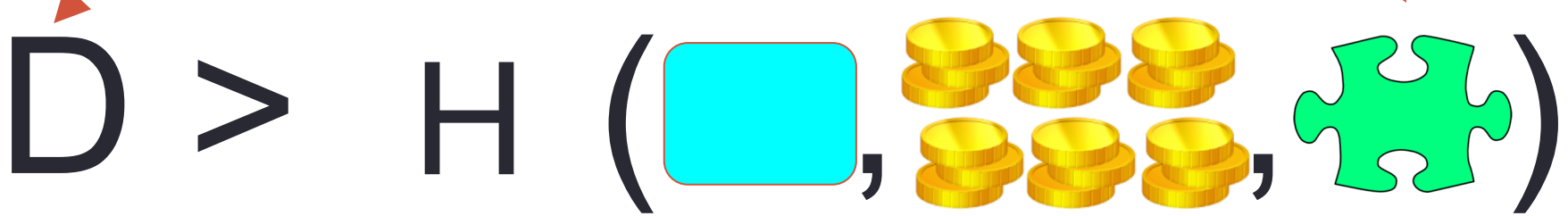
D > H ()

How to build a "blockchain"

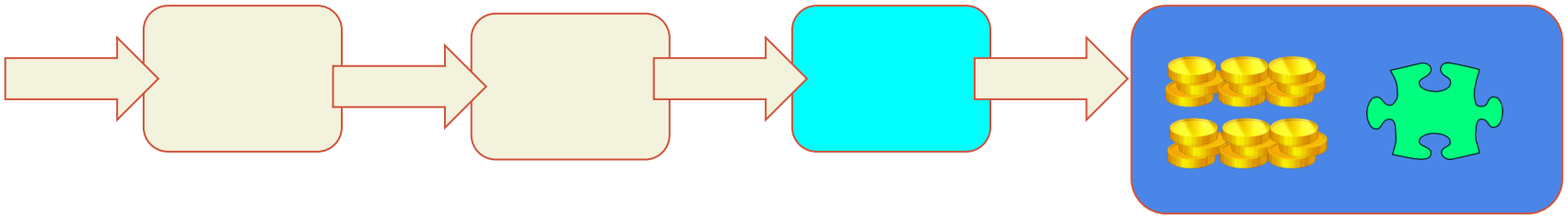


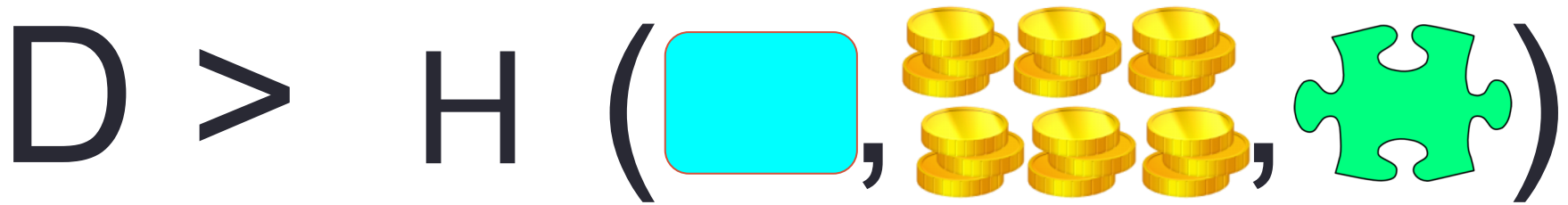
puzzle solution

Difficulty

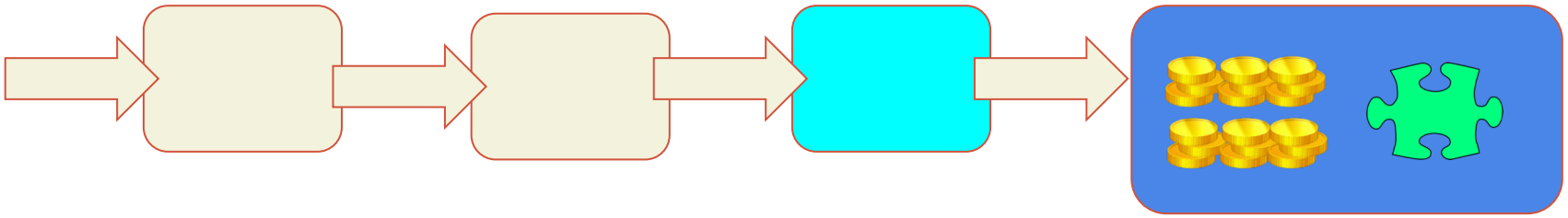


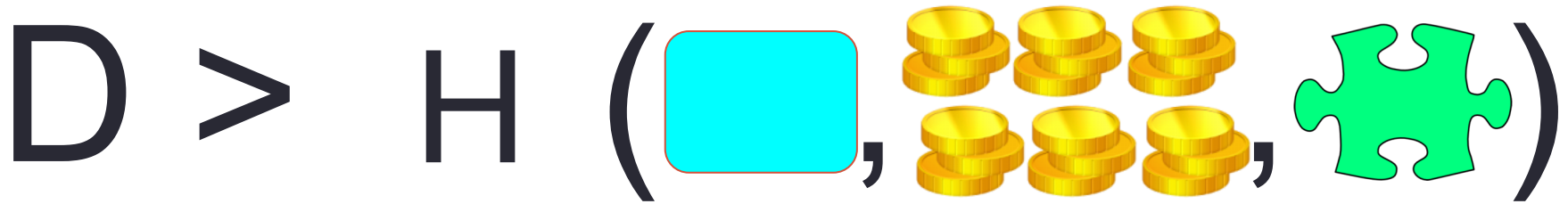
Search for a puzzle solution





We found a new block





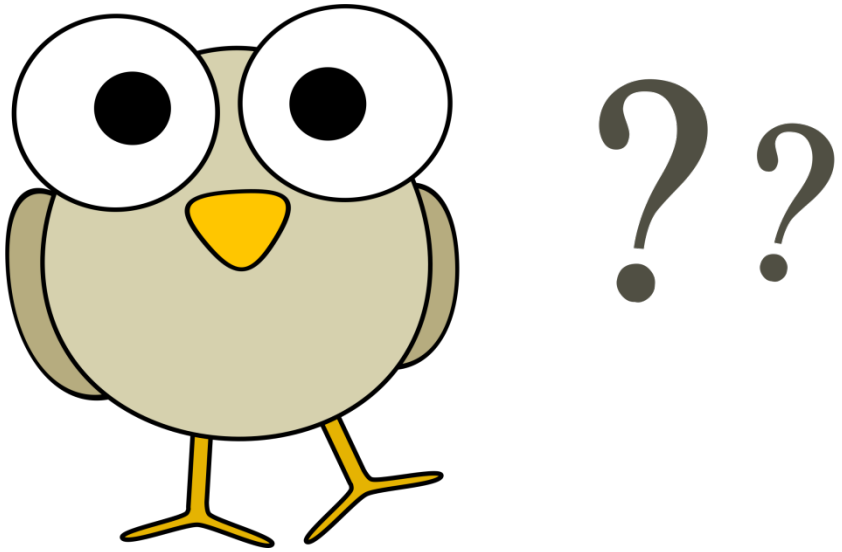
Best way to find a solution is brute-force search

Cryptography Use in Blockchain

- Initiation and broadcasting of transactions:
  - Digital signatures
  - Private/public keys
- Validation of transaction:
  - Proof of work
- Chaining blocks:
  - Hash functions

Blockchain

- Questions?



Public and Private Blockchains

- There are two classes of blockchains
  - Corresponding to the two type of consensus protocols
- Public/Permissionless
  - Most cryptocurrencies: Bitcoin, Ethereum, etc.
  - Transactions are typically pseudonymous
    - Accounts are public, but account holders are hidden
- Private/Permissioned
  - E.g., Hyperledger Fabric, JPMorgan
  - Participation requires credentials from administrator(s)

Blockchain and Privacy

- The basic tenant of any blockchain: everyone sees the same ledger
  - Does this mean that we cannot deal with private information?
- A partial solution: store an encryption/hash of the data
  - The data itself is kept somewhere off chain

Blockchain and Privacy

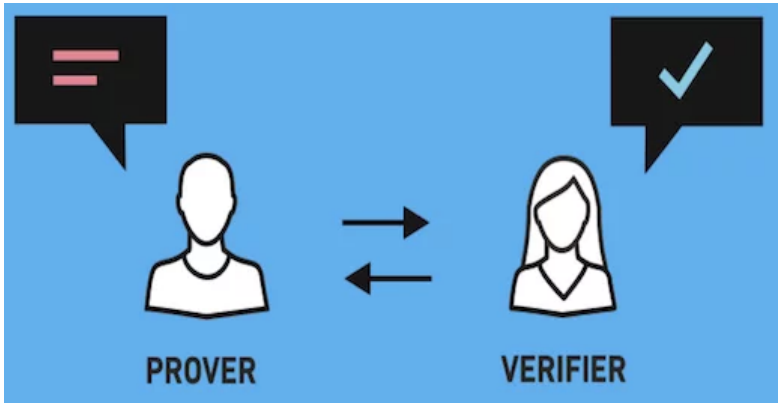
- A partial solution: store an encryption/hash of the data
  - The data itself is kept somewhere off chain
- What happens when we want to use that data?
  - The data owner can provide it, but it violates privacy
  - Moreover, how would anyone check it after the fact?
- A better solution, use *"advanced cryptography"*
  - Techniques such as zero knowledge, secure multiparty computation, homomorphic encryption, etc.
  - Capable of protecting computation, not just data

ADVANCED CRYPTO TOOLS

- Zero-Knowledge (ZK)
- Secure Multi-Party Computation (MPC)

Zero Knowledge Proofs

- I have a secret
  - I can convince you of some properties of my secret
  - Without revealing it



- Available (in principle) since the 80's [GMR'85]

Zero Knowledge Proofs

- I have a secret
  - I can convince you of some properties of my secret
  - Without revealing it
- Example: my secret is my purchase his

Zero Knowledge Proofs

- I have a secret
  - I can convince you of some properties of my secret
  - Without revealing it
- Example: my secret is my purchase his
  - I can prove to Hood that I bought 10 gallons of milk this month
    - so I can get a coupon
  - Without revealing anything else

Zero-Knowledge Proofs [GMR'85]

- Alice proves to Bob that a statement is true
  - Without revealing anything about why it is true

- Illustration: proving to a color-blind person that two balls have different colors

Zero-Knowledge Proofs - Example

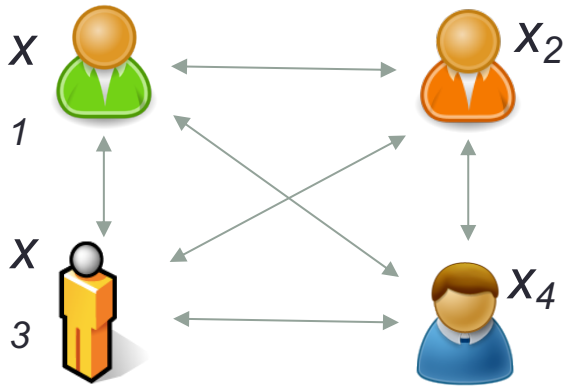
- Example: Alice proves to Bob, who is color-blind, that two balls have a different color
- Procedure: Give Bob the two balls
- Bob puts balls behind his back

Zero-Knowledge Proofs - Example

- Repeat multiple times:
  - Bob shuffles balls an odd or even number each time
  - Shows the ball in his right hand
  - Alice tell Bob if he switched the number of balls an even or odd times
- After multiple times, Bob verifies Alice is always right
  - = Therefore, the balls indeed have different colors

Secure Multi-Party Computation

- We all have our individual secrets
  - We can compute a function of these secrets
  - Without revealing them to each other (or anyone else)



Goal:
Correctness: Everyone computes $y = f(x_1,\ldots,x_n)$
Privacy: Nothing but the output is revealed

- Available (in principle) since the 80's [Yao'86, GMW'86]

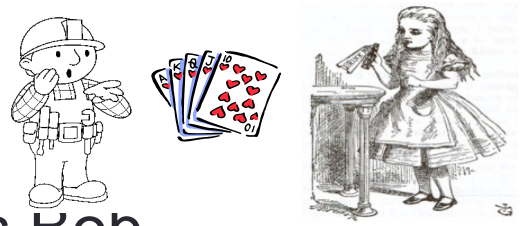
Secure Multi-Party Computation

- We all have our individual secrets
  - We can compute a function of these secrets
  - Without revealing them to each other (or anyone else)
- Example: medical data
  - Evaluating the effectivness of a treatment

    f **(patient1Data, patient2Data, ...) = effective/not-effective**
  - Data for different patients held by different clinics
  - Can compute this without revealing any private data

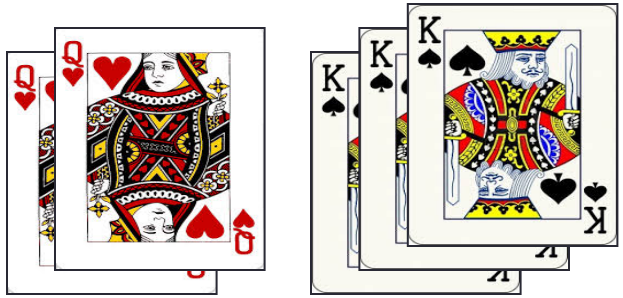
EXAMPLE: GAME OF LIKE

Illustration: Alice and Bob's First Date



- Alice & Bob plan their first date:
- After the date
  - Alice will know whether or not she likes Bob
  - Bob will know whether or not he likes Alice
  - But neither will know (yet) what the other feels
- Then they plan to play a game
  - Game only reveals if they both like each other
    - The logical-AND function
  - But if Alice doesn't like Bob, then she does not learn whether Bob likes her (and vice versa)

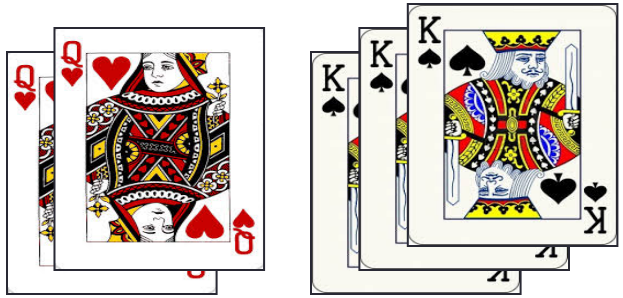
The “Game of Like” [den Boer ’89]



- Alice and Bob use five cards:
  - Two identical queen of hearts
  - Three identical king of spades
- Each of then gets one queen and one king
- Third king is left on the table, face down

The “Game of Like”

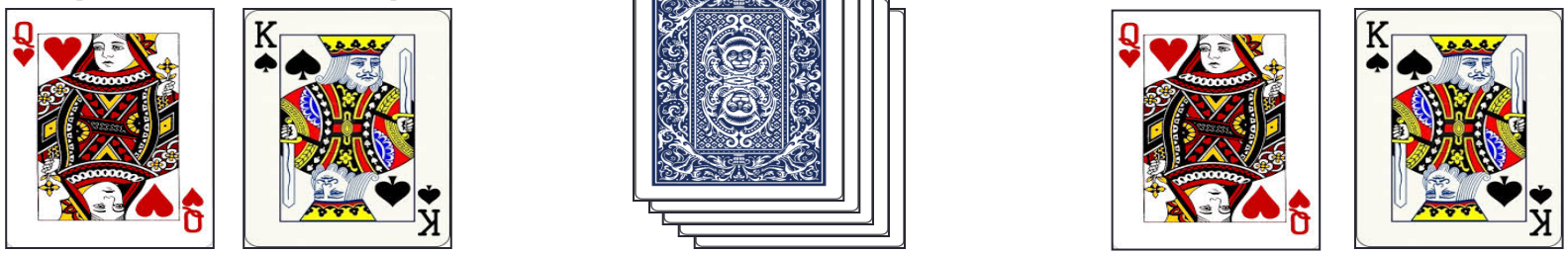
- Alice and Bob use five cards:
  - Two identical queen of hearts
  - Three identical king of spades



- Each of then gets one queen and one king
- Third king is left on the table, face down

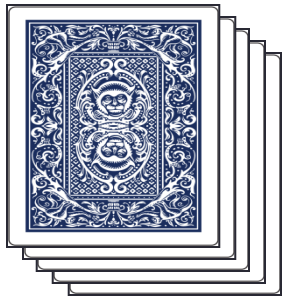
The "Game of Like"

- Bob puts his cards face down on top
  - Queen on top means he likes Alice, king on top means he does not
- Alice puts her cards face down on top
  - King on top means she likes Bob, queen on top means she does not



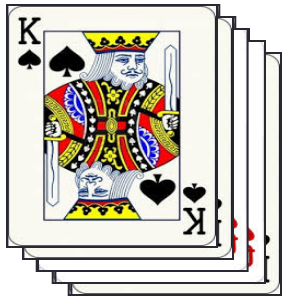
The “Game of Like”

- Alice and Bob take turn cutting the deck
  - Result is a cyclic shift of the deck



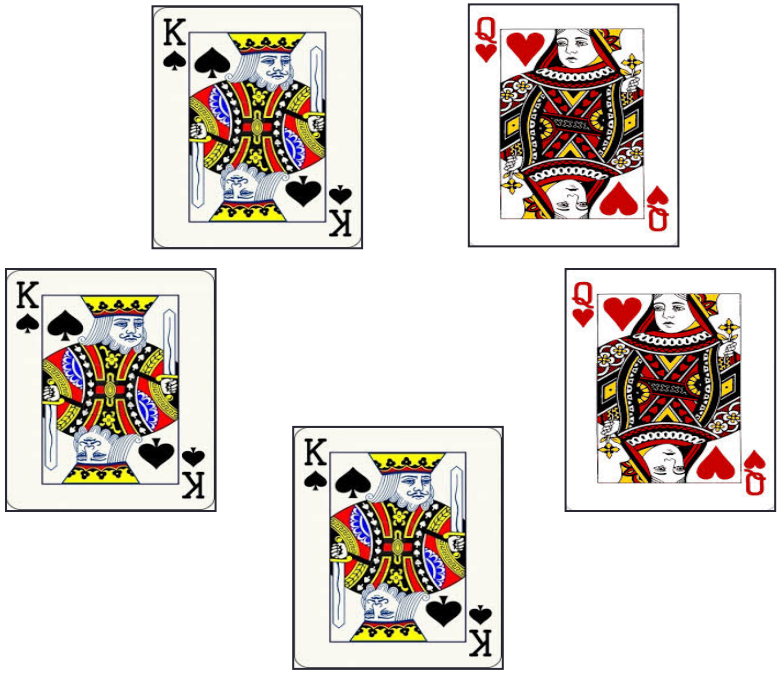
The “Game of Like”

- Alice and Bob take turn cutting the deck
  - Result is a cyclic shift of the deck
- Then they open the cards in order (on a circle)
  - If queens are adjacent they like each other

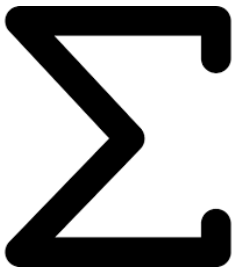


The "Game of Like"

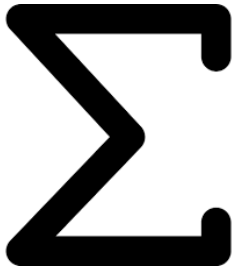
- Alice and Bob take turn cutting the deck
  - Result is a cyclic shift of the deck
- Then they open the cards in order (on a circle)
  - If queens are adjacent they like each other
- Theorem: nothing is revealed when the queens are not adjacent



EXAMPLE #2 – SUM OF THREE NUMBERS

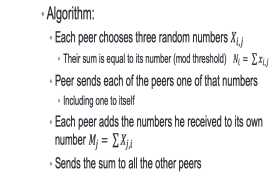


MPC Example 2



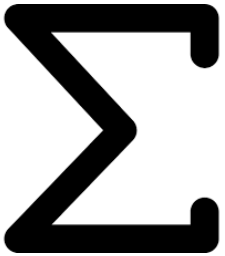
- Goal: calculate the sum of three secret r
  - Without sharing the numbers themselves
- Scenario:
  - Three peers, each has its own secret number N_i
  - Peers want to figure out the sum
    - Without sharing their secret number
  - Number is assumed to be less then some threshold
    - Say one Million

MPC Example 2



- Algorithm:
  - Each peer chooses three random numbers $X_{i,j}$
    - Their sum is equal to its number (mod threshold) $N_i = \sum x_{i,j}$
  - Peer sends each of the peers one of that numbers
    - Including one to itself
  - Each peer adds the numbers he received to its own number $M_j = \sum X_{j,i}$
  - Sends the sum to all the other peers

MPC Example 2



- Algorithm (cont.):
  - The peers summarize the numbers they received

$$S = \sum M_j$$

    - Each of the peers now has the sum, but did not learn anything about the number of the other peers

MPC Example 2

- N_i – secret number of each peer, $X_{i,j}$ - n

peer i sends to peer j

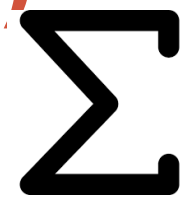
$$N_i = \sum_j X_{i,j}$$

- M_j - sum of all numbers peer j gets from all peers (including its own)

$$M_j = \sum_i x_{i,j}$$

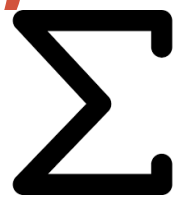
- $S = \sum_j M_j = \sum_{i,j} X_{i,j} = \sum_j N_j$

MPC Example 2 (cont.)



- Numbers each peer has: $N = \{7,20,53\}$
- Peer 1 chooses $X_{1,j} = \{0,3,4\}$, Peer 2 - $X_{2,j} = \{1,9,10\}$, Peer 3 - $X_{3,j} = \{15,15,23\}$
- Peer 1 adds numbers he received $\mathrm{M}_1 = \sum_i X_i = 0 + 1 + 15 = 16$
- Similarly, peer 2 and 3 receive $M_2 = 3 + 9 + 15 = 27; M_3 = 4 + 10 + 23 = 37$

MPC Example 2 (cont.)



- All peers send their respective M_i's to th er peers
- Each peer will now summarize the M_j's it received from all peers (including its own number)

$$S = \sum_j M_j = 16 + 27 + 37 = 80$$

- All peers have now the correct sum
  - But none learned the number of the other two peers

Secure Computation

- Theorem [GMW'86]: For any multi-party function $f \in Poly$, there exists a protocol to securely compute f

- The moral: anything that can be computed can be computed securely
  - But cost could be high

Theorem [GMW'86]: For any multi-party function $f \in Poly$, there exists a protocol to securely compute f

- The moral: **anything that can be computed can be computed securely**
  - But cost could be high

Real-World Secure Computation

- Prices of Sugar Beets in Denmark are determined using secure computation
  - For over five years now
- Some universities and other organizations are using cryptographic voting protocols
- Extensive research over last decade into improving efficiency and usability
  - Some start-ups, code libraries, etc.

BACK TO BLOCKCHAIN

Blockchain and Privacy

- A very active research direction

- Privacy achieved through different cryptographic means:
  - Zero-knowledge: Zcash
  - MPC: Pre-IPO training demo, bidding proof-of-concept

Z-Cash

- A distributed technology
  - Using zero-knowledge proofs
  - Introduced in 2016
    - New development
- A digital coin currency
- Technology hides details of transaction
  - The sender, receiver and payment amount

Zero-Knowledge Proofs

- Technology allows to hide certain data
  - But can not hide two (or more) data items
    - While calculating a function on both
  - => Can MPC help?

SUPPORTING PRIVATE DATA ON HYPERLEDGER FABRIC WITH SECURE MULTIPARTY COMPUTATION [BHH18]

HYPERLEDGER

Introduction - Hyperledger Fabric

- A “permissioned” blockchain architecture
  - Participants need to get permission to participate in transaction
- Originally introduced by Data Assets and IBM in the first Hackathon
  - Maintained by a consortium of companies, among them IBM
- Open source project, hosted  HYPERLEDGER foundation
  - Supported by many different organizations

Introduction - Hyperledger Fabric

- Intended to be used to develop new modular applications
- Provides support for consensus and membership services
  - Making them “plug and play”
- Logic based on “smart contracts” called “chaincode”

HYPERLEDGER

Dealing with Private Data

- Approach: some data on the ledger is encrypted
  - Everyone still sees the same ledger, ciphertext, and all
- My private data stored on the ledger, encrypted with my key
  - How to use it when trying to endorse transaction?
- Solution: use secure Multi-Party Computation (MPC)
  - An interactive protocol with multiple parties, each with private input
  - Computing the correct output, learning nothing more
- Many ways to enable auditability
  - Future work

Project Demo

- Utilizes MPC
- As part of the project a demo was created to demonstrate feasibility of approach
  - SmartBid demo
  - Created on top of Hyperledger, uses Hyperledger framework and technology

SmartBid Demo

- Demonstrates “eBay-like” auctions
  - But without the trusted intermediary
- Different entities with private data
  - Sellers have items with secret reserve price
  - Buyers have secret bid amount
- Item sold to the highest bidder above reserve price
  - If successful: winner, winning price revealed
  - If failed: nothing is revealed
- Current demo supports up to two bidders

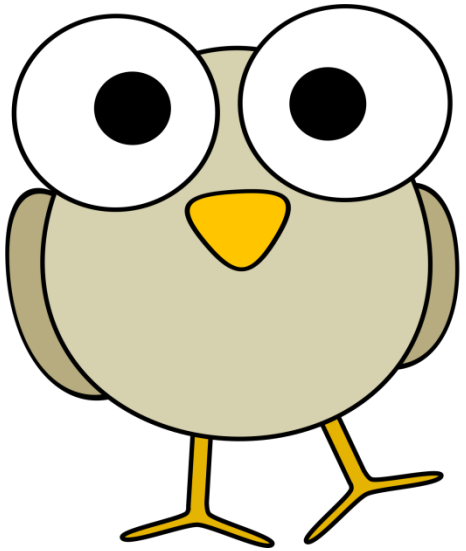
Pre-IPO Trading Prototype

- As a second proof-of-concept, a Pre-IPO trading prototype was created
  - Each user is entering a list of tuples of price-volume at which he is willing to purchase shares
    - A lower price would imply higher volume
  - Data is not shared on ledger
    - But resulting selling price is calculated based on participants data

Conclusions

- Blockchain is important influential
- Pose new questions, technical, economic ,and social
- Lots of opportunities for research

- Questions?



Bitcoin

- Bitcoin is a cryptocurrency, a form of electronic cash
- It is a decentralized digital currency
  - without a central bank or single administrator
- Bitcoins can be sent from user-to-user on the peer-to-peer bitcoin network
  - without the need for intermediaries

Bitcoin

- Transactions are verified by network nodes through cryptography
  - recorded in a public distributed ledger called a blockchain
- Bitcoin was invented by an unknown person or group of people
  - using the name Satoshi Nakamoto
- Released as open-source software in 2009

Bitcoin

- Transactions are verified by network nodes through cryptography
  - recorded in a public distributed ledger called a blockchain
  - Bitcoin creators are called “Miners”

Bitcoin

- Transactions are verified by network nodes through cryptography
  - Miners check if the issuer is the rightful owner (holder of private key) of coins to associated bitcoin address
  - Miners perform the following:
    - collect all pending transactions
    - verifies them
    - Bundles them into a block
    - Solve a cryptographic puzzle for this block
    - If it is successful, block is added to the existing blockchain
      - miner is rewarded with a bitcoin

What is Bitcoin?

- What is Bitcoin?

Trust and Bitcoins

- We've stopped trusting institutions and started trusting strangers

USABILITY OF BITCOIN

Usability of Bitcoin

- A First Look at the Usability of Bitcoin Key Management [Eskanadri et Al, 2015]
  - Surveyed 6 Bitcoin key management techniques
    - cover the vast majority of deployed Bitcoin software
  - Propose an evaluation and comparison framework for Bitcoin key management techniques
    - based on 10 security, usability and deployability criteria
  - Perform a cognitive walkthrough of six distinct Bitcoin clients and tools
    - identify usability issues while performing basic Bitcoin tasks
      - e.g.,viewing account balance, sending funds,etc.

Usability of Bitcoin

- Main findings:
  - Metaphors and abstractions used in the surveyed clients subject to misinterpretations
  - Clients do not do enough to support their users

Usability of Bitcoin

Category	*Example*	Malware Resistant	Key(s) Kept Offline	No Trusted Third Party	Resistant to Physical Theft	Resistant to Physical Observation	Resilient to Password Loss	Resilient to Key Churn	Immediate Access to Funds	No New User Software	Cross-device Portability
Keys in Local Storage	Bitcoin Core			●		●	●	●	●		
Password-protected Wallets	MultiBit		○	●	○	●		●	●		
Offline Storage	Bitaddress	○	●	●			●				●
Air-gapped Storage	Armory	○	●	●		●	●	●			
Password-derived Keys	Brainwallet		●	●	○			●	●	●	●
Hosted Wallet (Hot)	`Coinbase.com`						●	●	●	●	●
Hosted Wallet (Cold)		○	●				●	●		●	●
Hosted Wallet (Hybrid)	Blockchain.info		○	○			●	●	●	●	●
Cash		●	●	●		●	●	●	●	●	●
Online Banking							●	●	●	●	●

Usability of Bitcoin

- The Other Side of the Coin: User Experiences with Bitcoin Security and Privacy [Krombholz, 2017]

Usability of Bitcoin

- Online survey restricted to bitcoin users only
  - Hosted on soscisurvey.de
- Surveyed 990 Bitcoin users to determine Bitcoin management strategies
  - identified how users deploy security measures to protect their keys and bitcoins

Usability of Bitcoin

- Conclusions:
  - Managing bitcoins is still a major challenge for many users
  - Many users do not apply sufficient security measures such as encryption and backups
  - Many participants were not even aware of security features provided by their system
  - 22.5% of the study participants have already experienced security breaches and lost bitcoin
    - half of them mentioned a self-induced error as the reason
      - => users find it still difficult to manage their bitcoins in a secure way

Usability of Bitcoin

- Questions?