CSE 107: Introduction to Cryptography

Special Topics: Cryptocurrencies and Zero-Knowledge Proofs

Nadia Heninger

UCSD

Fall 2021

Some material from Eric Wustrow and George Danezis

Historical background: Cypherpunks

"We the Cypherpunks are dedicated to building anonymous systems. We are defending our privacy with cryptography, with anonymous mail forwarding systems, with digital signatures, and with electronic money." – Eric Hughes, A Cypherpunk's Manifesto, 1993

- 1985: David Chaum "Security without identification: Transaction systems to make Big Brother obsolete"
- In the 1990s, the Cypherpunks mailing list was extremely active; many influential members
- Software: PGP, Tor, anonymous remailers, Off-the-record messaging...
- Cypherpunk ideas: Anonymous digital currency, WikiLeaks, assassination markets, pseudonymity...
- These ideas encode libertarian-to-anarchist politics

How do you build digital currency?

A central authority can keep a balance ledger and update with each transaction.

Account	Amount
Dave	$342.87
Fred	$32,944.09
Eve	$89,218.87
Charlie	$429,718.90
Alice	$1,000.00
Bob	$0.00

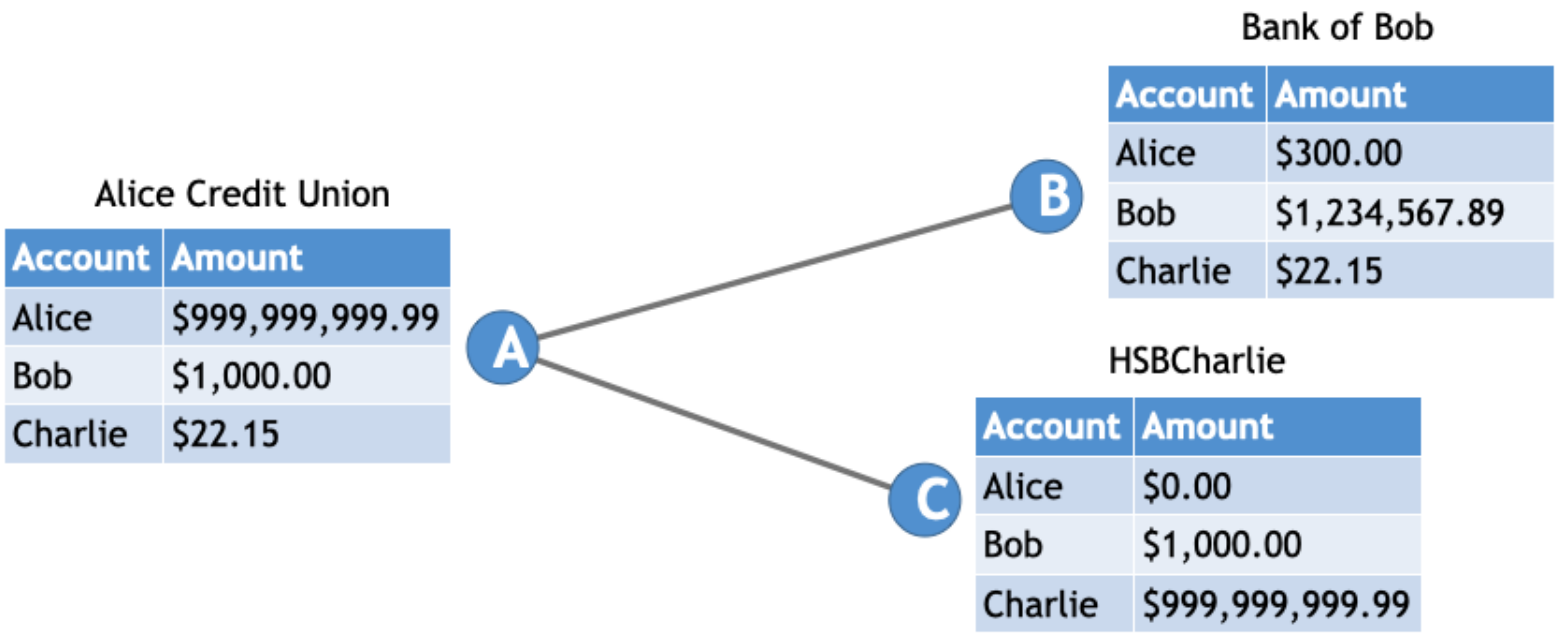
Alice pays Bob $200

Account	Amount
Dave	$342.87
Fred	$32,944.09
Eve	$89,218.87
Charlie	$429,718.90
Alice	$800.00
Bob	$200.00

How do you build a decentralized digital currency?

Without a central authority, different entities need to agree on transactions and balances.

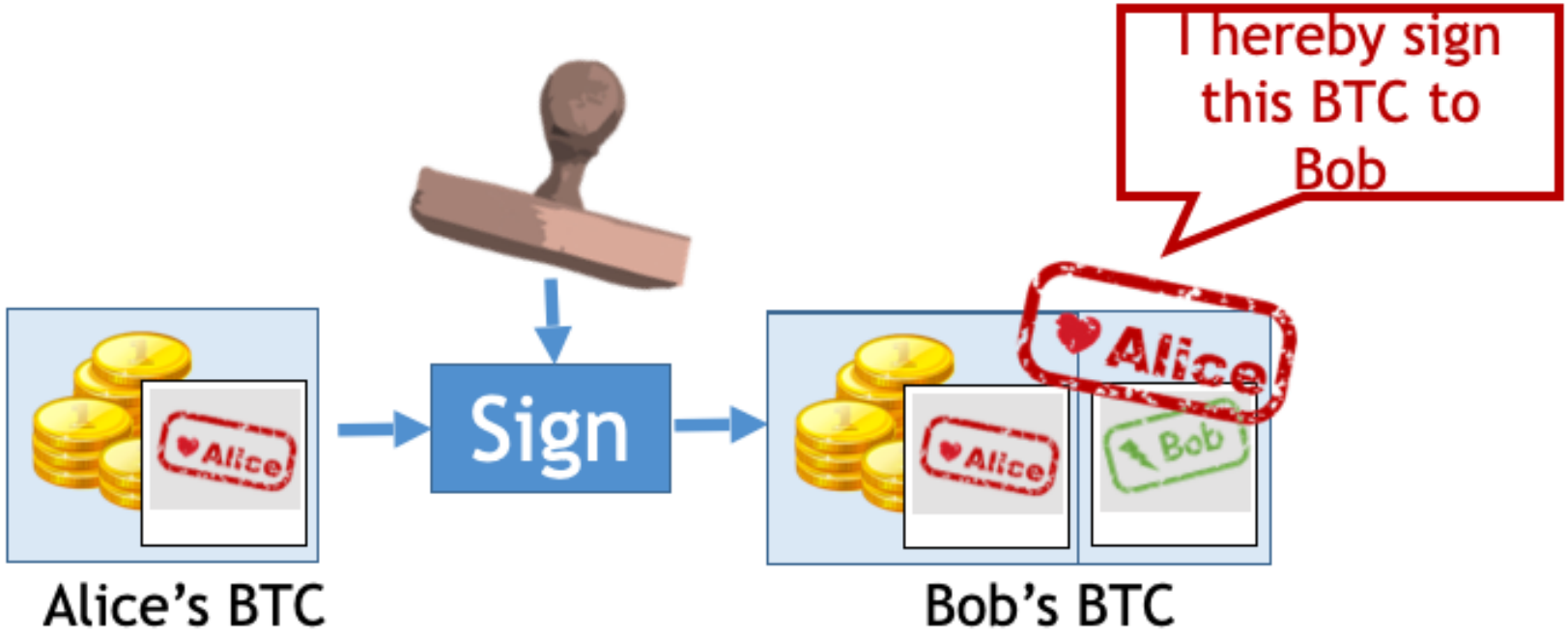
How do you keep someone from sending someone else's money to themselves?



Transactions: Use digital signatures to authenticate

A digital signature gives guarantees:

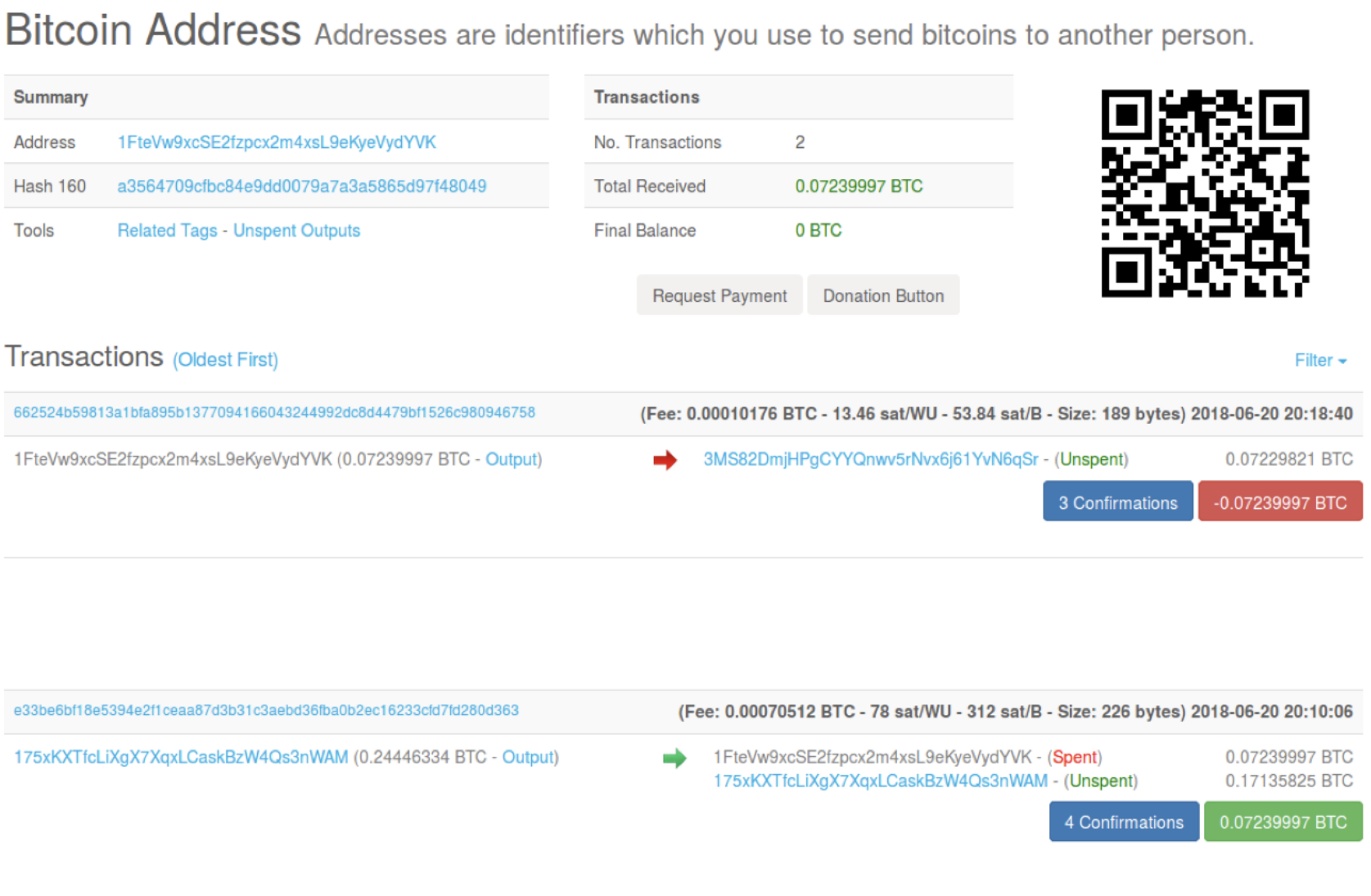
- The transaction has not been altered
- Only the entity with the private key can generate a valid signature
- Anyone can validate a signature with the public key



Pseudonymous identity: Derive from public key

Bitcoins are associated with an address.

The address is a hash of a public key.



Bitcoin Address Addresses are identifiers which you use to send bitcoins to another person.

Summary	
Address	1FteVw9xcSE2fzpcx2m4xsL9eKyeVydYVK
Hash 160	a3564709cfbc84e9dd0079a7a3a5865d97f48049
Tools	Related Tags - Unspent Outputs

Transactions	
No. Transactions	2
Total Received	0.07239997 BTC
Final Balance	0 BTC

Request Payment | Donation Button

Transactions (Oldest First) Filter

662524b59813a1bfa895b13770941660432449 92dc8d4479bf1526c980946758 (Fee: 0.00010176 BTC - 13.46 sat/WU - 53.84 sat/B - Size: 189 bytes) 2018-06-20 20:18:40

1FteVw9xcSE2fzpcx2m4xsL9eKyeVydYVK (0.07239997 BTC - Output) → 3MS82DmjHPgCYYQnwv5rNvx6j61YvN6qSr - (Unspent) 0.07229821 BTC

3 Confirmations | -0.07239997 BTC

e33be6bf18e5394e2f1ceaa87d3b31c3aebd36fba0b2ec16233cfd7fd280d363 (Fee: 0.00070512 BTC - 78 sat/WU - 312 sat/B - Size: 226 bytes) 2018-06-20 20:10:06

175xKXTfcLiXgX7XqxLCaskBzW4Qs3nWAM (0.24446334 BTC - Output) → 1FteVw9xcSE2fzpcx2m4xsL9eKyeVydYVK - (Spent) 0.07239997 BTC

175xKXTfcLiXgX7XqxLCaskBzW4Qs3nWAM - (Unspent) 0.17135825 BTC

4 Confirmations | 0.07239997 BTC

Problem: Double-spending

1. Alice has 1 token.
2. Alice sends 1 token to Bob and 1 token to Charlie.
3. Synchronization issue: each of Bob and Charlie is able to validate that Alice had a token to send, but doesn't know about the others' tokens.

A decentralized system needs some way to achieve consensus before transactions are accepted to prevent double-spending.

Bitcoin distributed ledger consensus

We would like to record all transactions in a public ledger.

Use some kind of consensus protocol to ensure everyone has same view of ledger.

Bitcoin uses a hash chain: every block of transactions includes cryptographic hash of previous block.

This means that once people agree on a block, they must agree on previous blocks.

Chaining blocks together with hash functions

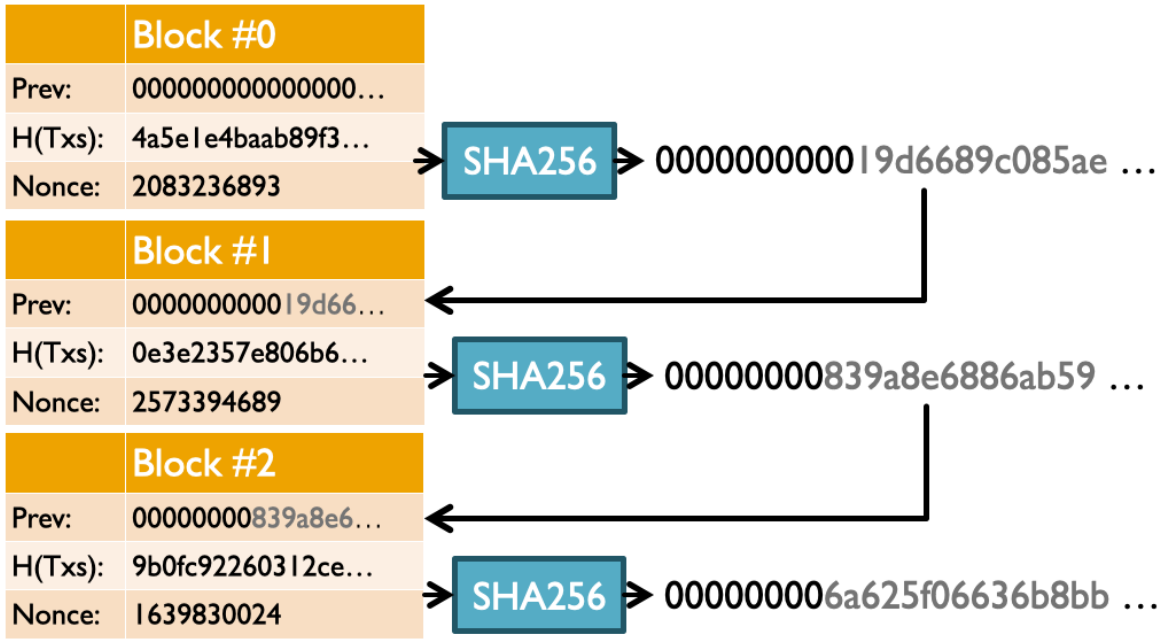
Network participants receive blocks from other nodes.

Which blockchain do you trust? The longest one.

How do you keep someone from making up a new super long blockchain?

Bitcoin uses “Hashcash” proof-of-work scheme to rate limit block creation.

Bitcoin consensus: Proof of work



- A block includes a set of transactions. "Miners" search for a nonce value that results in k leading 0s in the SHA256 hash of the block.
- We expect this to take 2^k hash function evaluations.
- The first miner to find such a value sends it to the network and work continues on the next block.
- The longest chain represents the most work: an attacker can't outcompete an honest majority.

Bitcoin Summary

Three main ideas:

- Public cryptographic keys for pseudonymous identifiers and transaction validation.
- Hash chain to ensure integrity of intermediate blocks.
- Proof-of-work-based distributed consensus scheme.

Bitcoin: Putting it all together

1. To generate an address, generate an ECDSA public key and hash it. This is your public address.
2. To receive money, another participant generates a transaction (actually a small executable script) sending bitcoin to this address and distributes it on the network.
3. Miners aggregate transactions from the network into a block and race to finish the proof of work first on that block.
4. The winning miner sends the block with proof of work on the network.
5. Once most nodes agree that the block with your transaction is part of the longest chain, you now have bitcoin.

“Smart contracts”: Ethereum

Idea: Include an expressive scripting language and have all nodes execute these scripts.

Pro: Replace governments, lawyers, accountants, and regulators with executable code.

“Smart contracts”: Ethereum

Idea: Include an expressive scripting language and have all nodes execute these scripts.

Pro: Replace governments, lawyers, accountants, and regulators with executable code.

Con: Basically nobody can write secure code. (See CSE 127.)

- An attacker stole $50 million of Ether from the DAO (decentralized autonomous organization) by exploiting a vulnerability in the DAO’s smart contract code.

- The Ethereum community decided to fork the blockchain to roll back the transaction.

Bitcoin and cryptocurrency criticisms

- Proof of work mining is environmentally wasteful. Bitcoin is now consuming more electricity than Argentina.

Bitcoin and cryptocurrency criticisms

- Proof of work mining is environmentally wasteful. Bitcoin is now consuming more electricity than Argentina.
  - Can replace proof of work with “proof of stake”: consensus based on coin ownership.

Bitcoin and cryptocurrency criticisms

- Proof of work mining is environmentally wasteful. Bitcoin is now consuming more electricity than Argentina.
  - Can replace proof of work with “proof of stake”: consensus based on coin ownership.

- Bitcoin is not anonymous. Transactions are all public, and addresses are only pseudonymous and can be linked to real-world identities when used.

Bitcoin and cryptocurrency criticisms

- Proof of work mining is environmentally wasteful. Bitcoin is now consuming more electricity than Argentina.
  - Can replace proof of work with “proof of stake”: consensus based on coin ownership.

- Bitcoin is not anonymous. Transactions are all public, and addresses are only pseudonymous and can be linked to real-world identities when used.
  - Zcash uses fancy crypto (zk-SNARKs) to validate transactions without publishing transactions publicly to network.

Bitcoin and cryptocurrency criticisms

- Proof of work mining is environmentally wasteful. Bitcoin is now consuming more electricity than Argentina.
  - Can replace proof of work with “proof of stake”: consensus based on coin ownership.

- Bitcoin is not anonymous. Transactions are all public, and addresses are only pseudonymous and can be linked to real-world identities when used.
  - Zcash uses fancy crypto (zk-SNARKs) to validate transactions without publishing transactions publicly to network.

- Bitcoin does not scale: The transaction rate will never be high enough to be a real payment network.

Bitcoin and cryptocurrency criticisms

- Proof of work mining is environmentally wasteful. Bitcoin is now consuming more electricity than Argentina.
  - Can replace proof of work with “proof of stake”: consensus based on coin ownership.

- Bitcoin is not anonymous. Transactions are all public, and addresses are only pseudonymous and can be linked to real-world identities when used.
  - Zcash uses fancy crypto (zk-SNARKs) to validate transactions without publishing transactions publicly to network.

- Bitcoin does not scale: The transaction rate will never be high enough to be a real payment network.
  - Various proposals (Lighting Network). Bitcoin will never be a payment network.

Bitcoin and cryptocurrency criticisms

- The cryptocurrency space is full of charlatans, ponzi schemes, and tulip mania.

Bitcoin and cryptocurrency criticisms

- The cryptocurrency space is full of charlatans, ponzi schemes, and tulip mania.
- A public blockchain is not what most people want for real-world applications.

Bitcoin and cryptocurrency criticisms

- The cryptocurrency space is full of charlatans, ponzi schemes, and tulip mania.

- A public blockchain is not what most people want for real-world applications.
  - A blockchain is just an append-only linked list.
  - Many proposed applications (healthcare? supply chain management?) better suited to a trusted third party with a database, an API, and maybe some digital signatures.
  - There are better distributed consensus algorithms for closed groups. (Computer scientists worked this stuff out in the 1980s...)

Bitcoin and cryptocurrency criticisms

- The cryptocurrency space is full of charlatans, ponzi schemes, and tulip mania.

- A public blockchain is not what most people want for real-world applications.
  - A blockchain is just an append-only linked list.
  - Many proposed applications (healthcare? supply chain management?) better suited to a trusted third party with a database, an API, and maybe some digital signatures.
  - There are better distributed consensus algorithms for closed groups. (Computer scientists worked this stuff out in the 1980s…)

- Irreversible transactions are not what consumers actually want in a payment system.
  - Cryptocurrencies are “speedrunning 500 years of bad economics”–Nick Weaver

Cryptocurrencies: The positives

- Renewed excitement in CS research like Byzantine fault tolerance, consensus protocols, programming language design for smart contracts, exotic cryptographic primitives...
- In a gold rush, the people who get rich are not the miners following the crowds, but the people selling equipment to the miners.

Zero-Knowledge Proofs

A zero-knowledge proof is

- A protocol between a prover and a verifier
- That allows the prover to convince the verifier of a statement about secrets

Properties of proof systems:

- Completeness: True statements can be proven by honest provers to honest verifiers
- Soundness: False statements can’t be proven to honest verifiers by cheating provers
- Zero-knowledge: The verifier only learns that the statement is true, and learns no information about the secret

Example Motivations

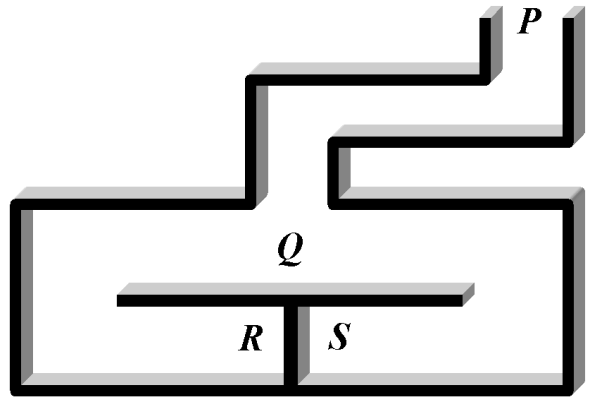
A zero-knowledge (ZK) proof allows you to

- Convince Bob your claim is true
- Without revealing anything beyond the fact that your claim is true

Bob is	Your claim is	What is not revealed is
Another student	You can solve the homework problem	Your solution
A server	You have a valid password authorizing your access	Your password
The Clay Institute	You have a proof that P is different from NP	Your proof

Ali Baba's Cave

Alice wants to prove that she knows the secret words to open the R-S portal without revealing the words to Bob.



The protocol:

1. Bob goes to P and waits there.
2. Alice goes to either R or S, chosen at random.
3. Bob goes to Q and randomly says either "R" or "S".
4. Alice appears from the side chosen by Bob.

Conviction: If Alice does not know the secret words, then with probability $1/2$ she will not be able to appear on the side requested by Bob.

Zero-knowledge: If Alice knows the secret words, Bob cannot hear them from his position at Q.

Identification

Alice		**Server**
Has password s		Has hashed password $P = H(s)$
	$\xrightarrow{s}$	If $H(s) = P$ then accept else reject.

Problem: Server learns s, even if sent over TLS

Identification

Let G be a cyclic group of order m generated by g.

Assume discrete logarithm problem relative to g is hard.

Alice		**Server**
Has secret key s		Has public key $P = g^s$
	$\xrightarrow{s}$	If $g^s = P$ then accept else reject.

Same Problem: Server learns s, even if sent over TLS

Schnorr Identification Protocol

Let G be a cyclic group of order m generated by g.

Assume discrete logarithm problem relative to g is hard.

Alice		**Server**
Has secret key s		Has public key $P = g^s$
$r \xleftarrow{\$} \mathbb{Z}_m$; $R \leftarrow g^r$	$\xrightarrow{R}$	$c \xleftarrow{\$} \mathbb{Z}_m$
	$\xleftarrow{c}$	
$a \leftarrow r + sc \bmod m$	$\xrightarrow{a}$	If $g^a = RP^c$ then accept else reject

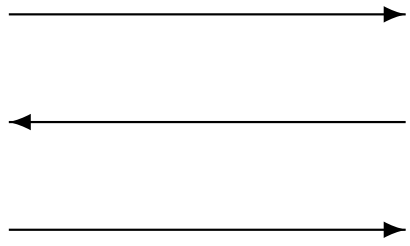
Conviction: If Alice does not know the secret s, she will be unable to find R, a satisfying the verification equation.

Zero-knowledge: Server does not learn s

The general setting

The prover is claiming to know s such that $R(P, s) = 1$ where R is some public relation. The verifier has P.

Prover **Verifier**



Completeness: If Prover has s such that $R(P, s) = 1$ and Prover follows protocol then Verifier accepts with probability 1.

Proof of knowledge: If Prover does not "know" s such that $R(P, s) = 1$ then it cannot make Verifier accept with probability greater than $1/2$.

Zero knowledge: If Prover knows s such that $R(P, s) = 1$ and follows protocol then verifier learns nothing beyond this.

How to turn Schnorr into a signature

Signatures should not be interactive.

Alice signs message and sends message and signature to Bob.

The “Fiat-Shamir Heuristic” turns Schnorr into a Non-Interactive Zero-Knowledge (NIZK) Proof of Knowledge

Let G be a cyclic group of order m generated by g.

Alice

Has secret key s

$r \xleftarrow{\$} \mathbb{Z}_m$; $R \leftarrow g^r$

$c = H(P, R, m)$

$a \leftarrow r + sc \bmod m$

$\xrightarrow{m, c, a}$

Server

Has public key $P = g^s$

If $c = H(P, R, m)$ and $g^a = RP^c$ then accept else reject

Applications of Zero Knowledge

Can generate proofs for NP languages, boolean circuits.

Improvements allow more efficient proofs.

Lots of extensions: zk-SNARK (Zero Knowledge Succint Non-Interactive Argument of Knowledge), zk-STARK (Zero Knowledge Scalable Transparent ARgument of Knowledge) etc.

Application: Efficient verification of outsourced computation.