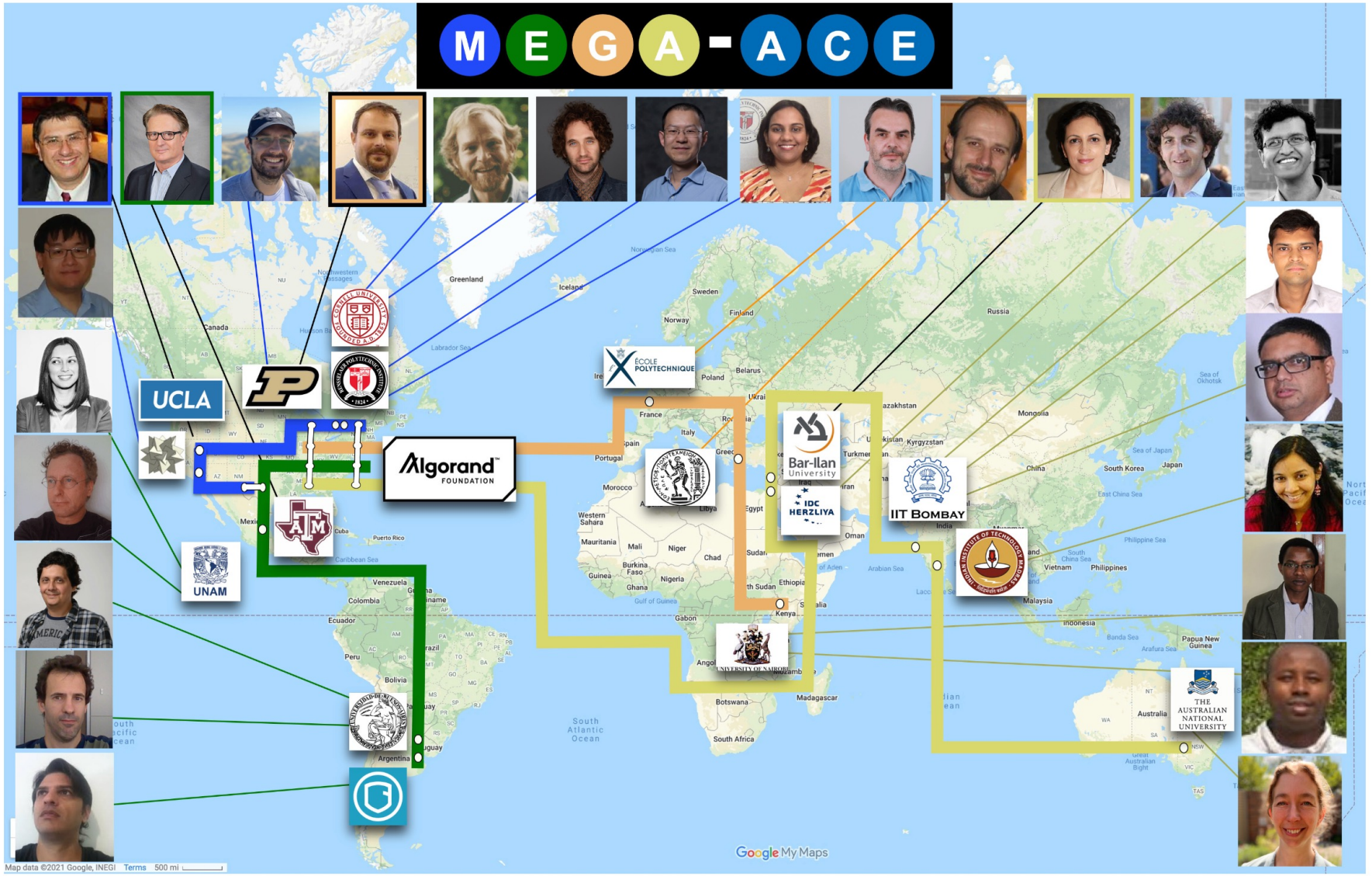
Inaugural Texas A&M Blockchain Day

Welcome & Blockchain Warm-Up

Juan Garay
Department of Computer Science & Engineering
Texas A&M University
garay@tamu.edu
https://jagaray.com

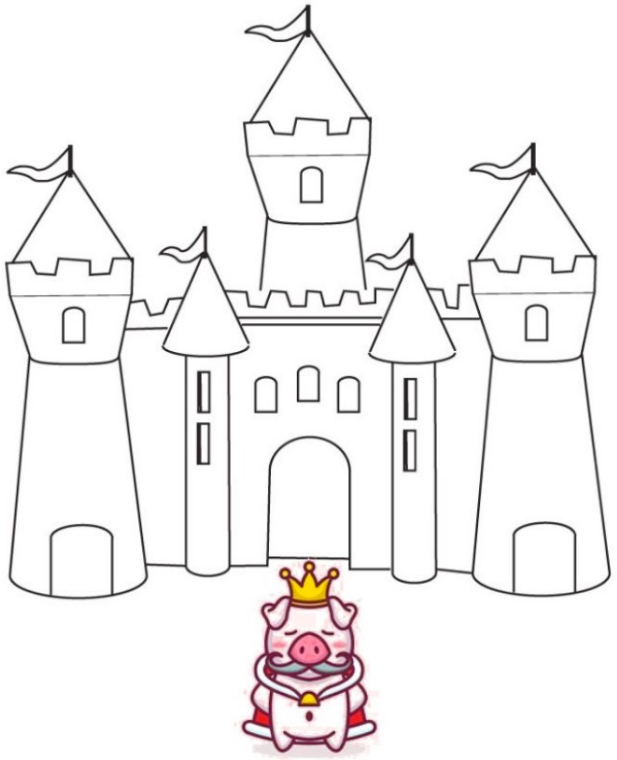
Event Sponsors

MEGA-ACE: Multidisciplinary Educational Global Alliance for Algorand Center of Excellence



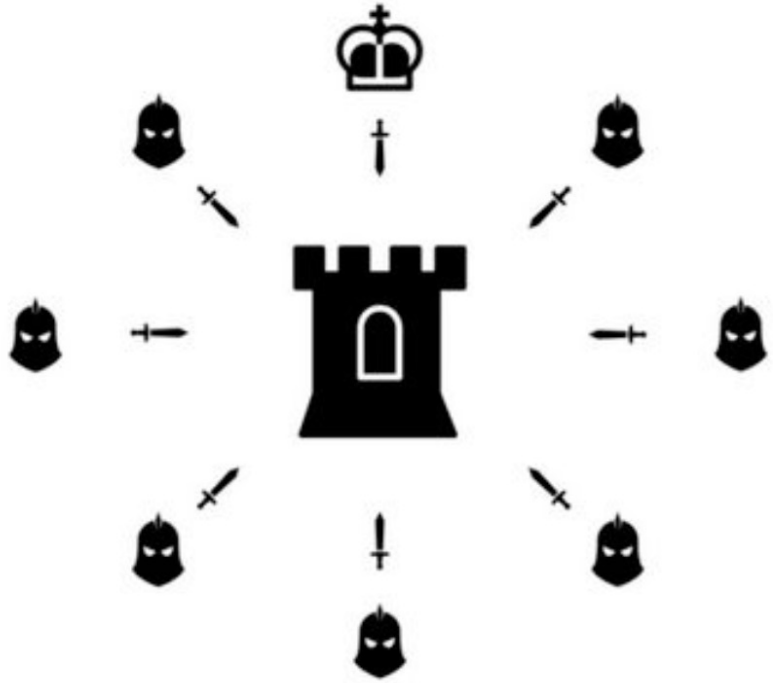
Centrally Controlled (Computer) Systems

- A single person (party/node) controls who can read/write/delete data
- If the person/party/node dies/is dishonest/crashes, the system crashes



Controlled-Access Distributed Systems

- Nodes collectively control the system
- If only few nodes are faulty, system remains operational
- Controlled participation — only authorized parties



Open-Access Distributed Systems?

- Nodes collectively control the system
- If only few nodes are faulty, system remains operational
- Anyone can participate, join or leave as they please

What is a Blockchain?

- A *blockchain* is a distributed database that satisfies a unique set of *safety* and *liveness* properties
- To understand it, we will focus on its first application
- *Distributed ledgers* use a blockchain as one possible means of implementation

Distributed Ledger

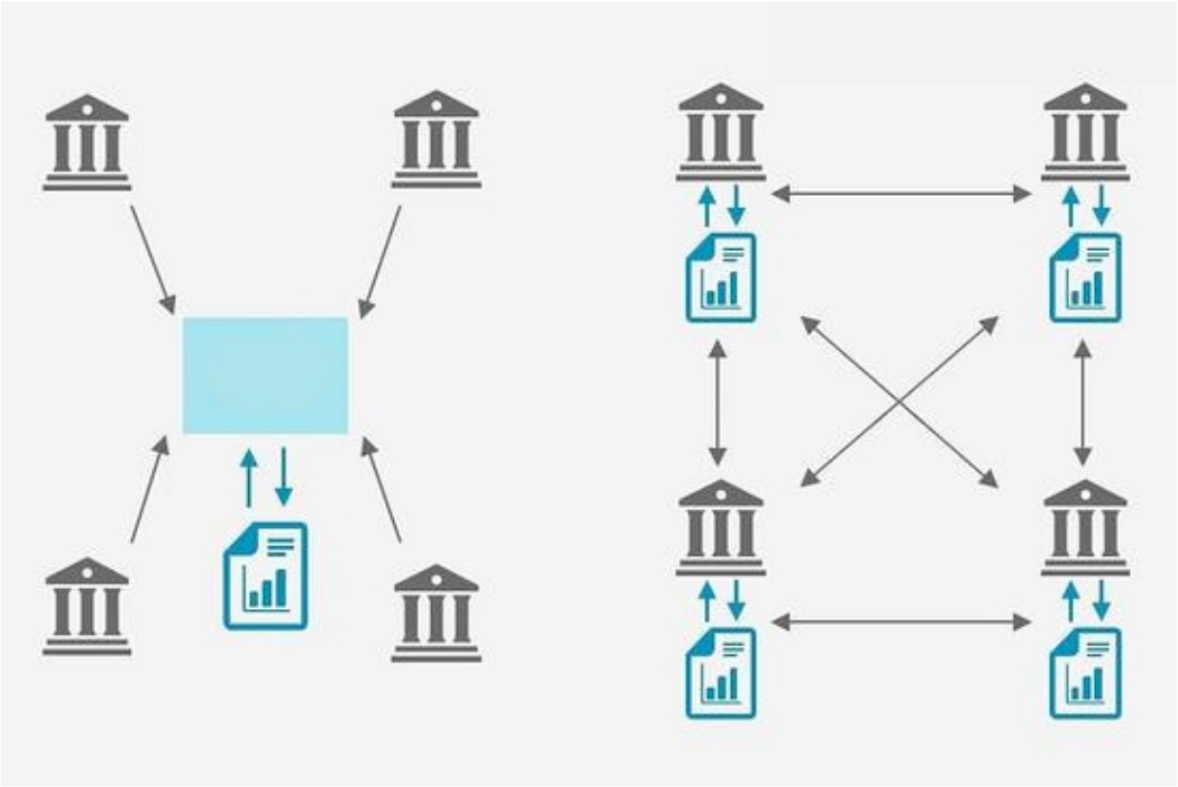


Figure: http://blogs.wsj.com/cio/2016/02/02/cio-explainer-what-is-blockchain/

- Consistency: Everyone sees the same history
- Liveness: Everyone can add new transactions

The Never-Ending-Book Parable

A Book of Data

- Anyone can be a scribe and produce a page
- New pages are produced indefinitely, as long as scribes are interested in doing so
- Each new page requires some effort to produce

Importance of *Consensus*

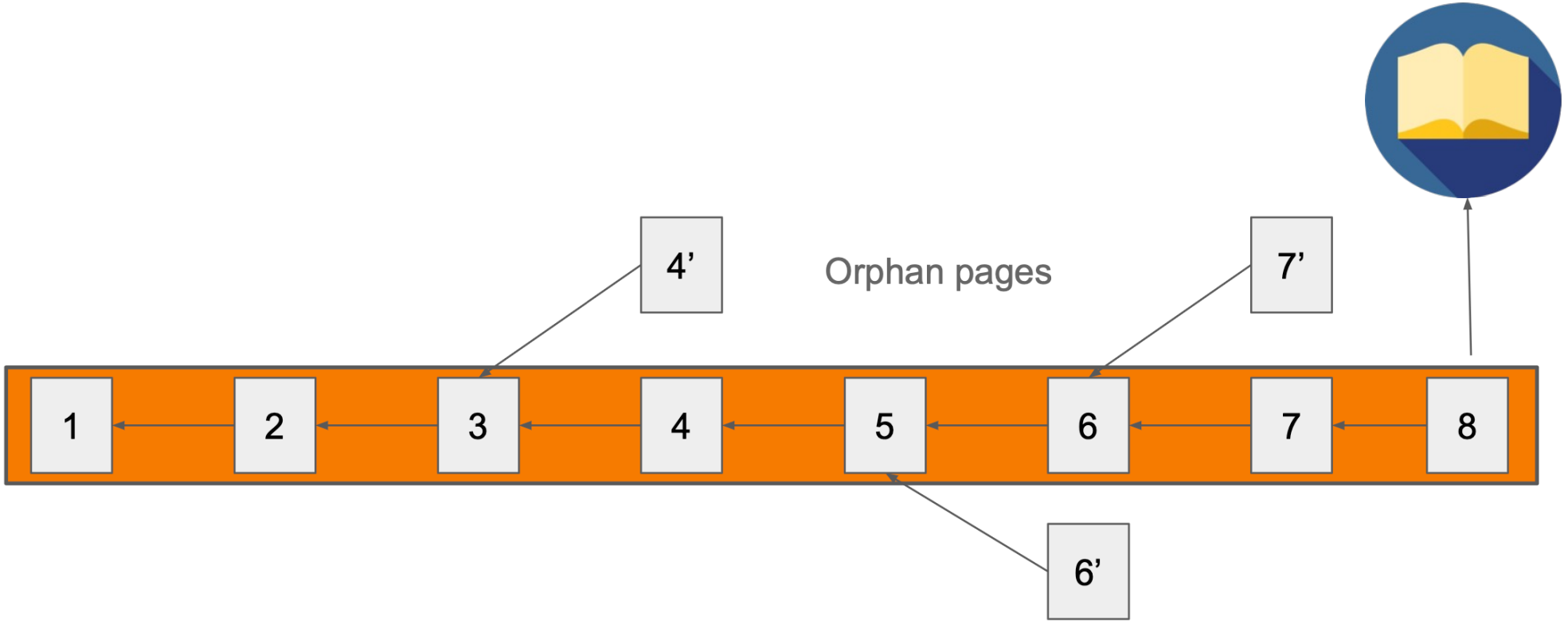
If multiple conflicting books exist, which one is the right one?

Choosing the Correct Book

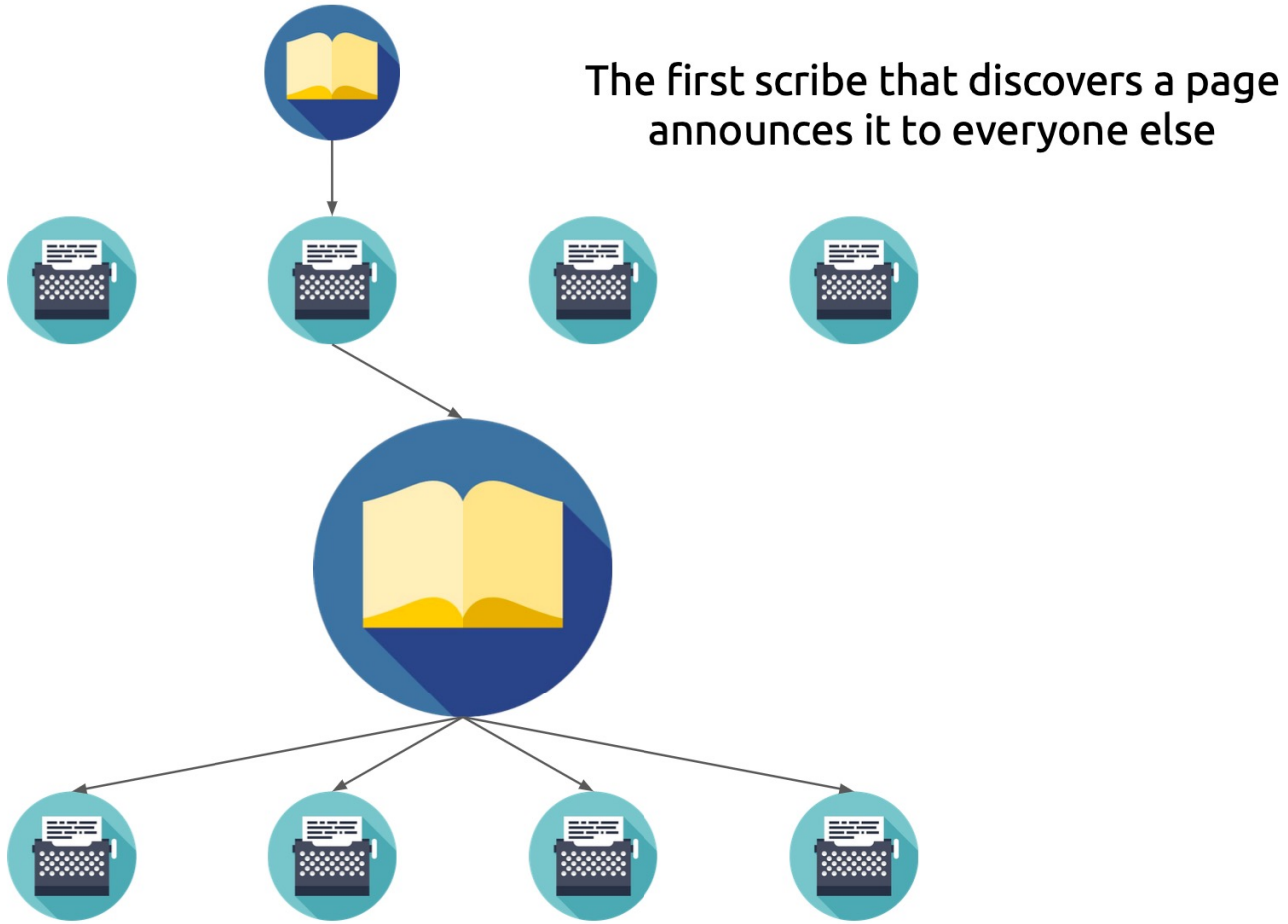
The correct book to work on and refer to is the book with the most pages. If multiple books exist, just pick one at random

Assembling the Current Book

- Each page refers to the previous one
- Assembled by stringing together the longest sequence of pages

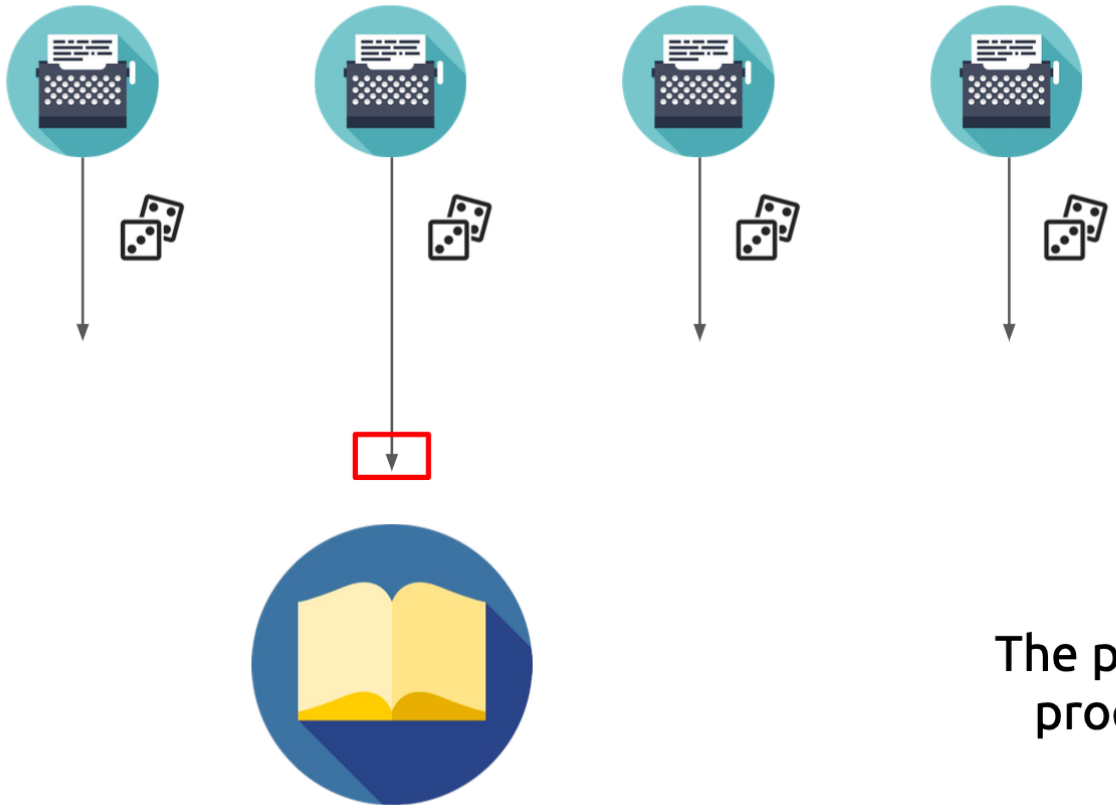


Rules for Adding Pages to (Extend) the Book



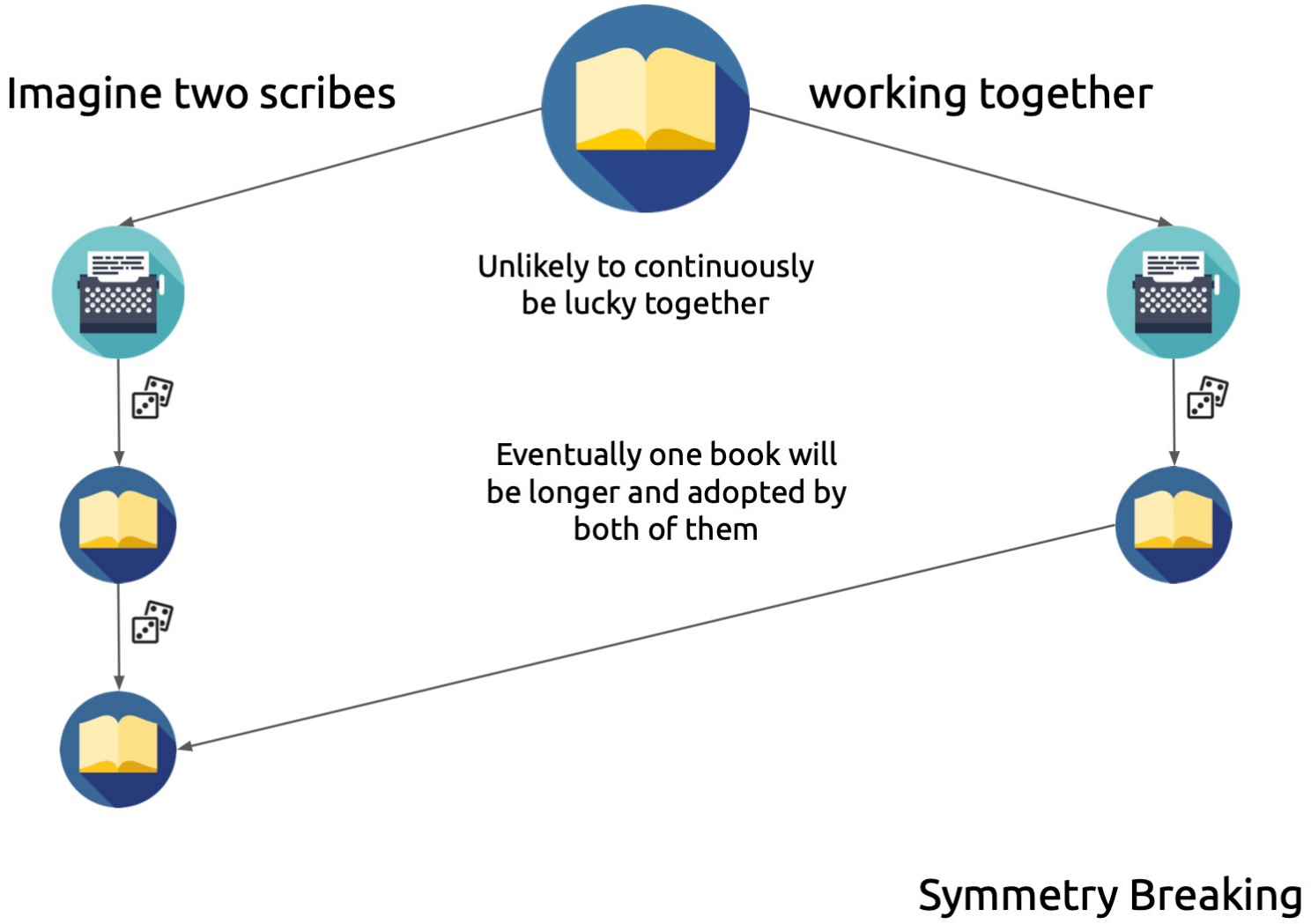
Effort is Needed to Produce a Page

Each page needs a special combination from a set of dice to be rolled



The probabilistic nature of the process is paramount to its security

The Benefits of Randomness



Being a Scribe

- Anyone can be a scribe for the book…
- … as long as they have a set of dice
- The more dice a scribe has, the higher the likelihood to produce the winning combination to add a page to the book

Parable and Reality

	The “blockchain”
	“Miners” / Computer systems that organize transactions in blocks
	Solving a cryptographic puzzle that is moderate hard to solve
	Using a computer to test for a solution from a large space of candidate solutions

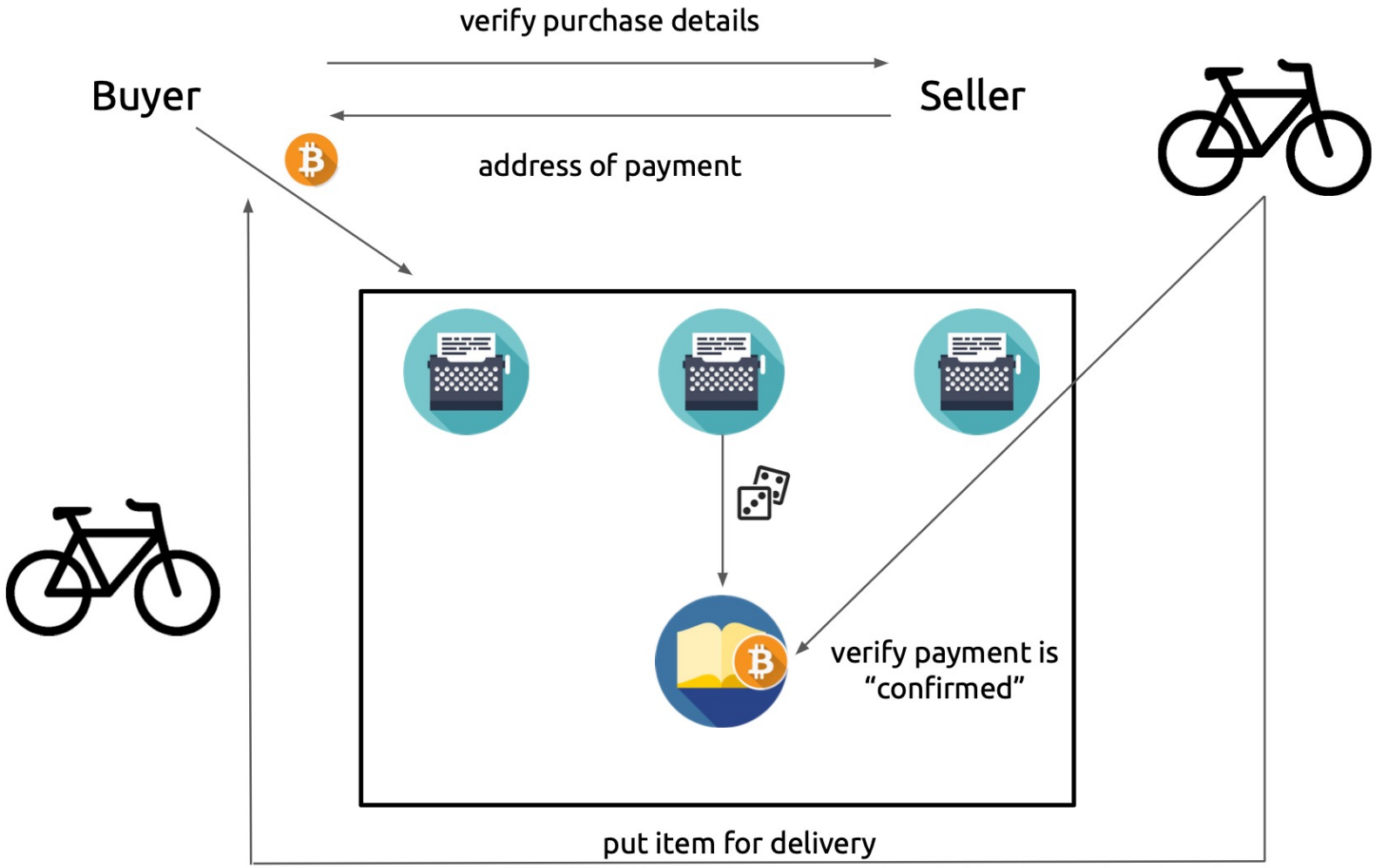
The First Blockchain Application: Bitcoin

What is Bitcoin (Trying to Be?)

- Money
  - Competing with UDS $, GBP £, EUR €, etc.
  - Medium of exchange: Give money to get goods and viceversa
  - Unit of account: Means to price goods, for accounting/debt purposes
  - Short/Medium term store of value: Can be exchanged for the same amount of goods in the (not so distant) future
- Payment system
  - Competing with cash, Visa, Mastercard, etc.
  - High throughput (large no. of transactions/sec)
  - Low latency (fast transaction settlement)
  - Uninterrupted operation
- Commodity
  - Competing with gold, silver, oil, etc.
  - A (useful) “material” that can be bought/sold

Person-to-Person Funds Transfer

Using the Bitcoin “Book”



Advantages

- Resilience
  - The book is shared across the network
  - Even if some of the nodes crash or are corrupted, the system remains operational
- Censorship resistance
  - Anyone can participate
  - Geographical disparity of nodes
  - Good alternative for borderline (or beyond) legal transactions
- Digital and open
  - New applications can easily be built on top of it
  - Programs can be hosted (and executed) on the ledger (“smart contracts”)

Disadvantages

- Bad as money
  - Price fluctuations and circulation do not follow economic growth (bad store of value)
  - Nothing is priced in Bitcoin (bad unit of account)
  - Slow and expensive (bad medium of exchange)
    - Low throughput (~ 5 tx/sec)
    - High latency (~ 60 mins)
    - High fees*** (~ $3)
- Irreversibility
  - If a transaction is processed, it cannot be deleted/reversed
  - If user's Btc's are stolen or user loses key, no recovery mechanism exists
- Environmentally damaging
  - Bitcoin CO_2 footprint*: 76.44 Mt (~Colombia's)
  - *Single* Btc tx CO_2 footprint*: 820.84 kg (~1.8M Visa transactions)
  - Single Btx tx e-waste**: 242 g (1.5 iPhones)

* https://digiconomist.net/bitcoin-energy-consumption

** https://www.sciencedirect.com/science/article/pii/S0921344921005103

*** https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

Smart Contracts

From Money to Smart Contracts

- Since we have created the book, why stop at recording monetary transactions?
- We can encode in the book arbitrary relations between accounts
- Scribes can perform tasks and take action, like verifying that stakeholders comply/adhere to contractual obligations

Questions to Consider

- How are the pages created? Since at the beginning the book is empty, where does the “money” come from? $\Longrightarrow$ Proof of Work (PoW)
- How is it possible to sign something digitally? $\Longrightarrow$ Digital Signatures
- How does a page properly refer to a previous page? $\Longrightarrow$ Hash Functions

Proofs of Work

Proofs of Work (aka “Cryptographic Puzzles”)

- Main objective: Given some *data,* ensure that some effort has been invested in computation involving the data
- “Moderately hard functions” [DN92,RSW96,Back97,JB99,GMPY06, BGJPVW16, BRSV17/18…]

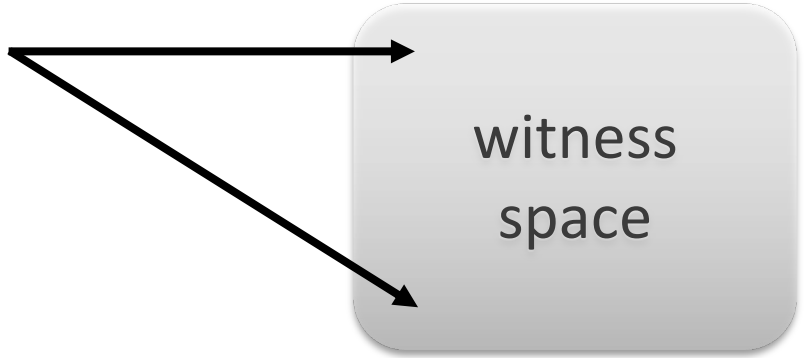
Prover **Verifier**

Proofs of Work (aka "Cryptographic Puzzles") (2)

More formally:

$Q(\cdot,\cdot)$: Polynomial-time predicate

$Q(x,\cdot)$



Challenge determines work level d

Search for witness takes long time (e.g., exp(d)) (a complexity lower bound needs to be assumed)

Successful only with some (small) probability!

Verification is easy!

Proofs of Work (aka "Cryptographic Puzzles") (3)

- One concrete instantiation:

```
int counter;
counter = 0
while Hash(data, counter) > Target
        increment counter
return counter
```

- In this case: PoW of *data* equals a value *w* with the property

$$\text{Hash}(data, w) \leq \text{Target}$$

- In Bitcoin:
  - Hash = SHA-256
  - Example of Target:

Target = 0000000000000000171A8B0000000000000000000000000000000000000000

Questions to Consider

- How are the pages created? Since at the beginning the book is empty, where does the “money” come from? ⟹ Proof of Work (PoW)
- How is it possible to sign something digitally? ⟹ Digital Signatures
- How does a page properly refer to a previous page? ⟹ Hash Functions

Bitcoin Address/Account (Security)

- Based on *elliptic curve cryptography* (ECC, curve *secp256k1*)
- Account: (PrivKey, PubKey)

- Bitcoin address:

  Base58(RIPEMD160(SHA256(PubKey)))

  E.g., *37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP*

- PrivKey used to **sign** outgoing transactions
- Wallet: many (PrivKey, PubKey)
- Transaction:

Bitcoin Transactions

Example:
Alice sends Bob 1 BTC, Bob uses it to send another payment.

When Alice sends Bob a payment of 1 BTC, she signs a transaction that deducts 1 BTC from her funds and creates a new transaction output that is worth 1 BTC and can only be spent by Bob, the owner of the recipient address.

Bob now wants to send 0.4 BTC to Charles. The transaction output from Alice's transaction is now used to fund this new transaction. The transaction creates two new outputs: One with 0.4 BTC that is associated with Charles' address, and one with 0.6 BTC associated with Bob's address (it is the change). The first transaction output (from Alice's transaction) is consumed by the transaction.

Putting it All Together

PoW-based Blockchain Protocols (Bitcoin)

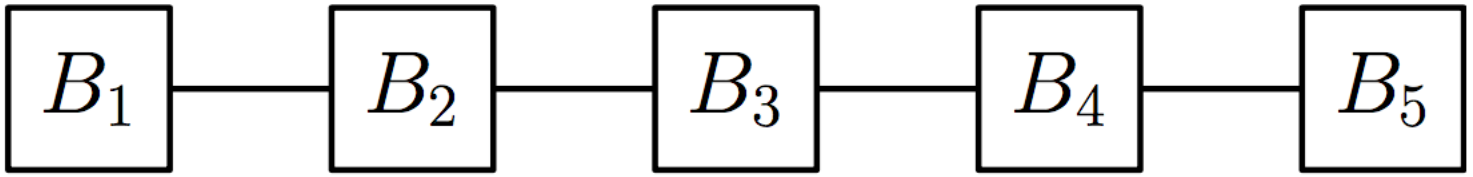
- Parties (“miners”) have to do work in order to install a transaction

PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks

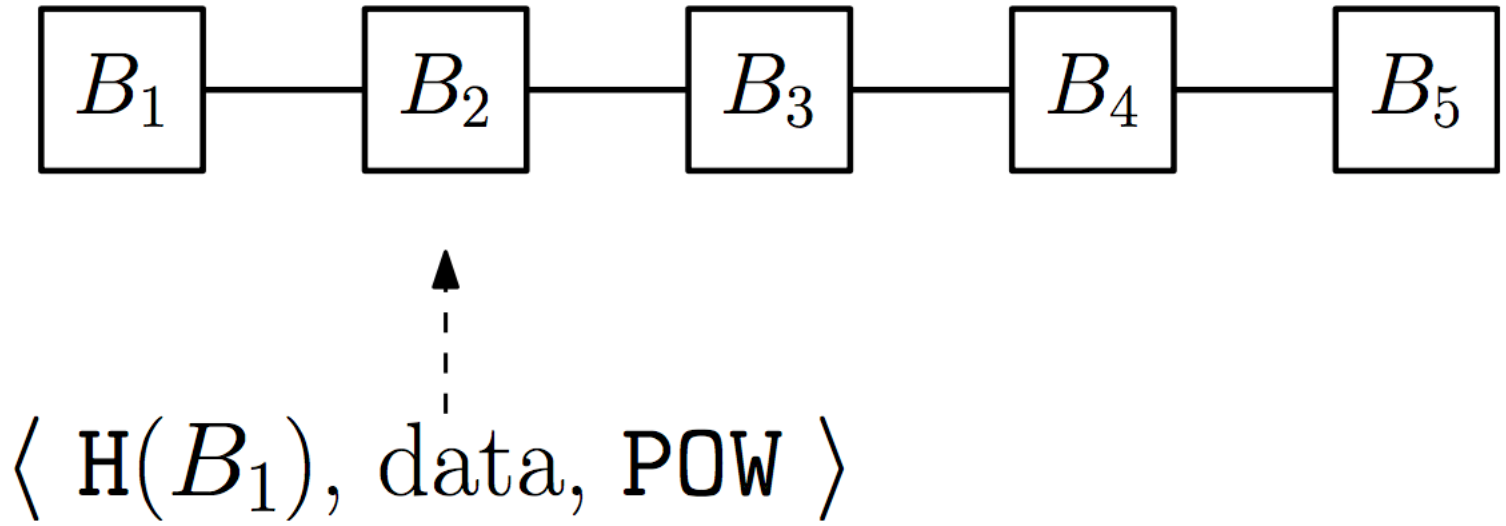
PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks



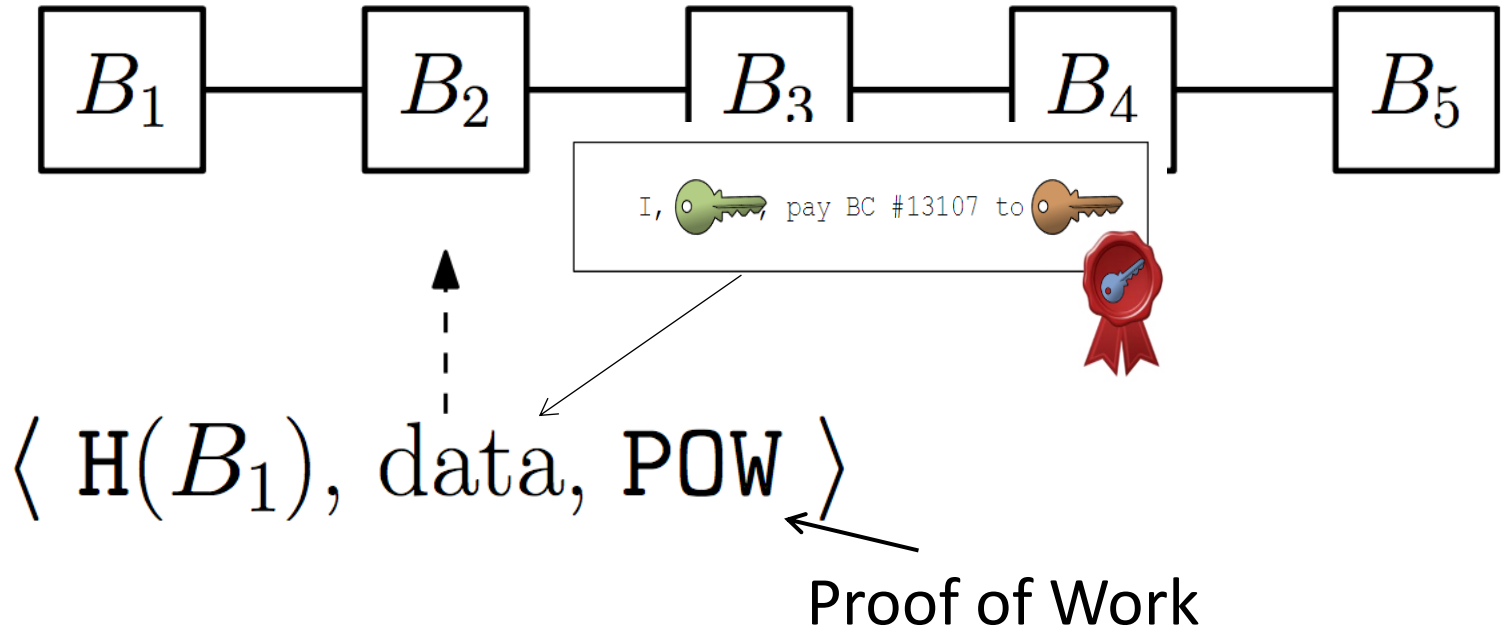
PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks



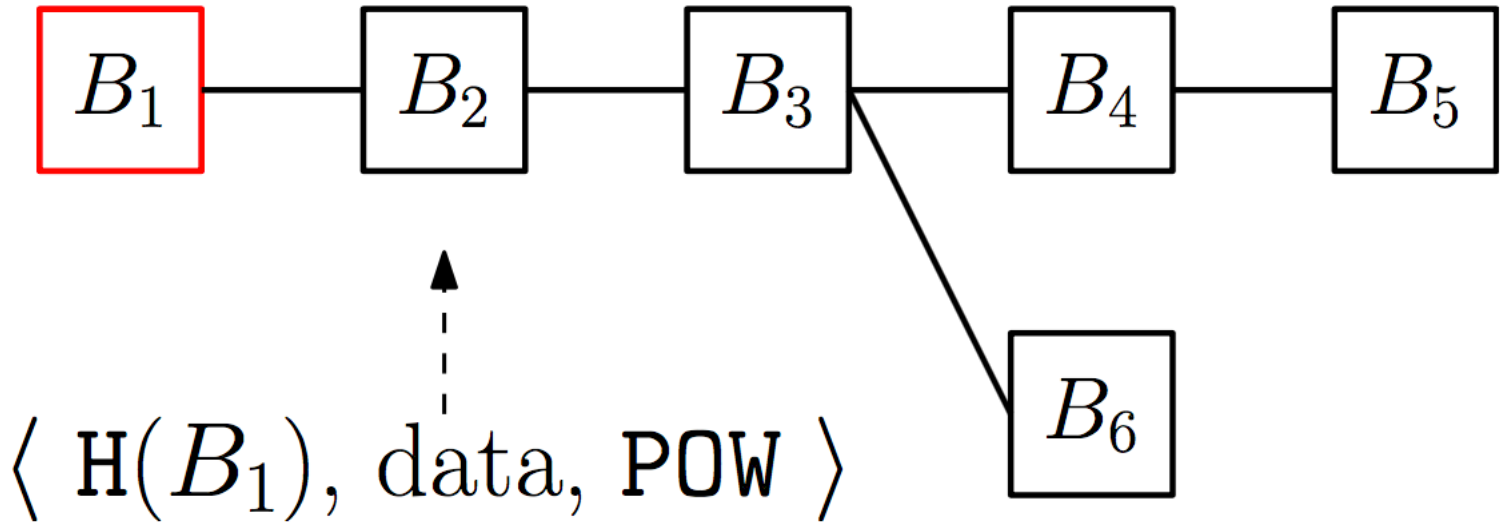
PoW-based Blockchain Protocols (Bitcoin)

- Parties ("miners") have to do work in order to install a transaction
- Transactions are organized in chains of blocks



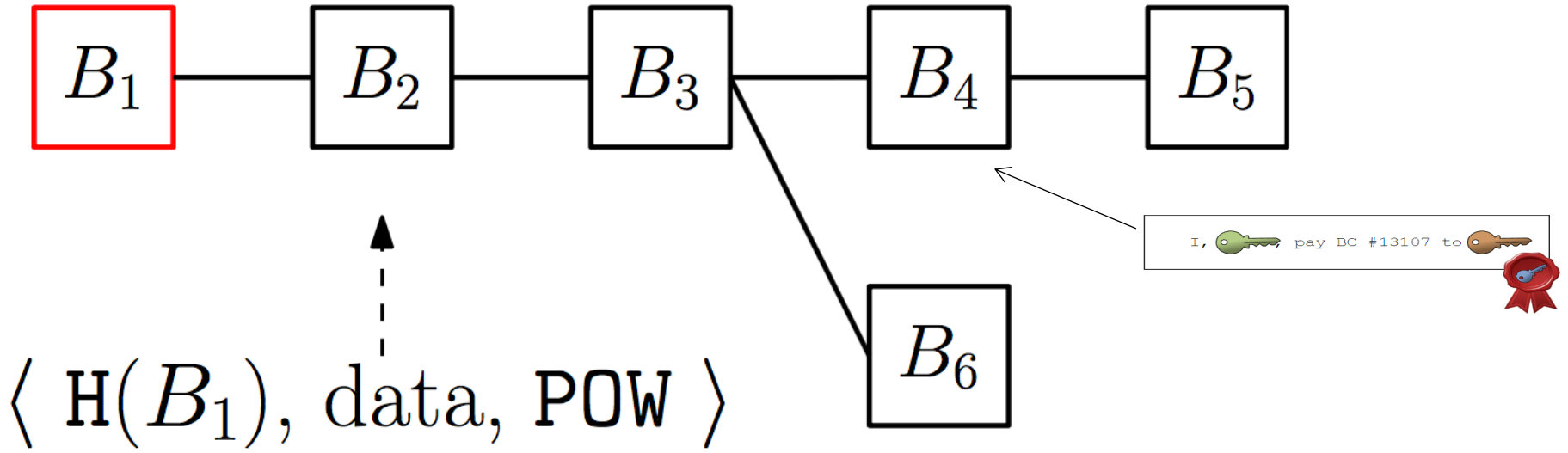
PoW-based Blockchain Protocols (Bitcoin) (2)

- Parties ("miners") always choose the *longest* chain they received



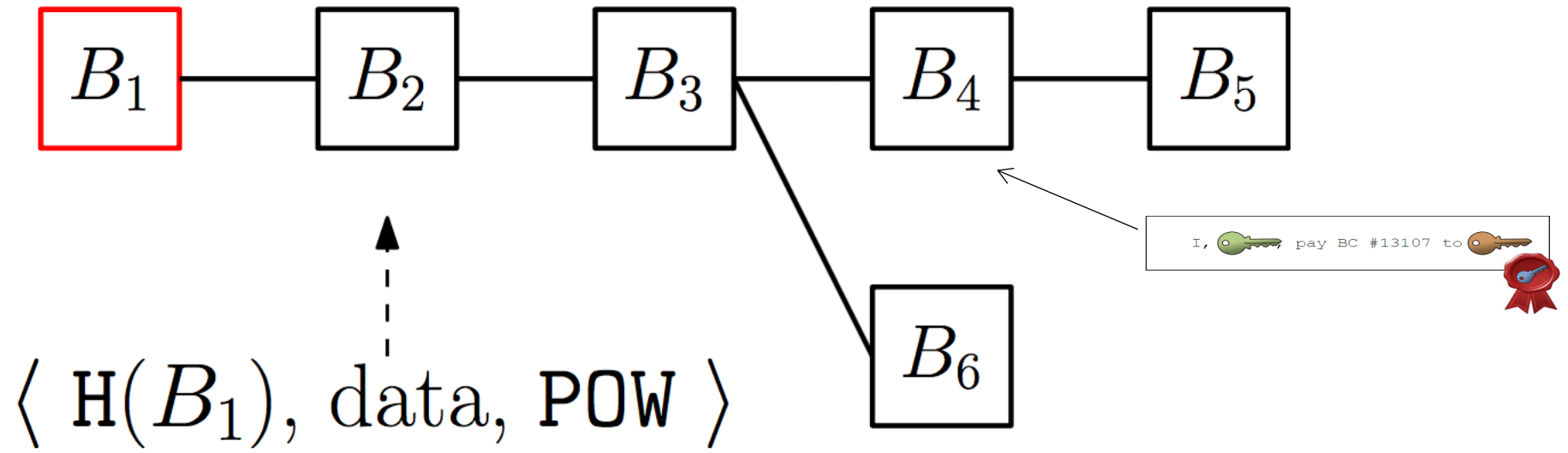
PoW-based Blockchain Protocols (Bitcoin) (3)

- Parties ("miners") always choose the *longest* chain they received
- If party wants to **erase** a transaction, it has to find a longer chain!



PoW-based Blockchain Protocols (Bitcoin) (4)

- Parties ("miners") always choose the *longest* chain they received
- If party wants to **erase** it transaction, it has to find a longer chain!



- If transaction is "sufficiently deep," it cannot do this unless it has a "majority of hashing power"!

The Intriguing “Permissionless” Model

The “Permissionless” Model [Nak08]

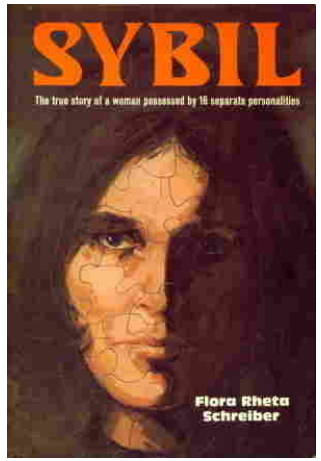
- Not a traditional distributed system
  - Nodes known *a priori* and authenticated

- The “permissionless” model
  - Nodes do *not* know each other (not even their exact number!)
  - Nodes come and go
  - *Anyone* can join

- And yet, realize a distributed ledger!

The "Permissionless" Model [Nak08] (2)

- Strong impossibility results w/o authentication [Oku05, BCLPR05]
- *Sybil* attacks are unavoidable [Dou02,…]
- ⅓ barrier in no. of misbehaving parties [LSP82, Bor96, Fit03]

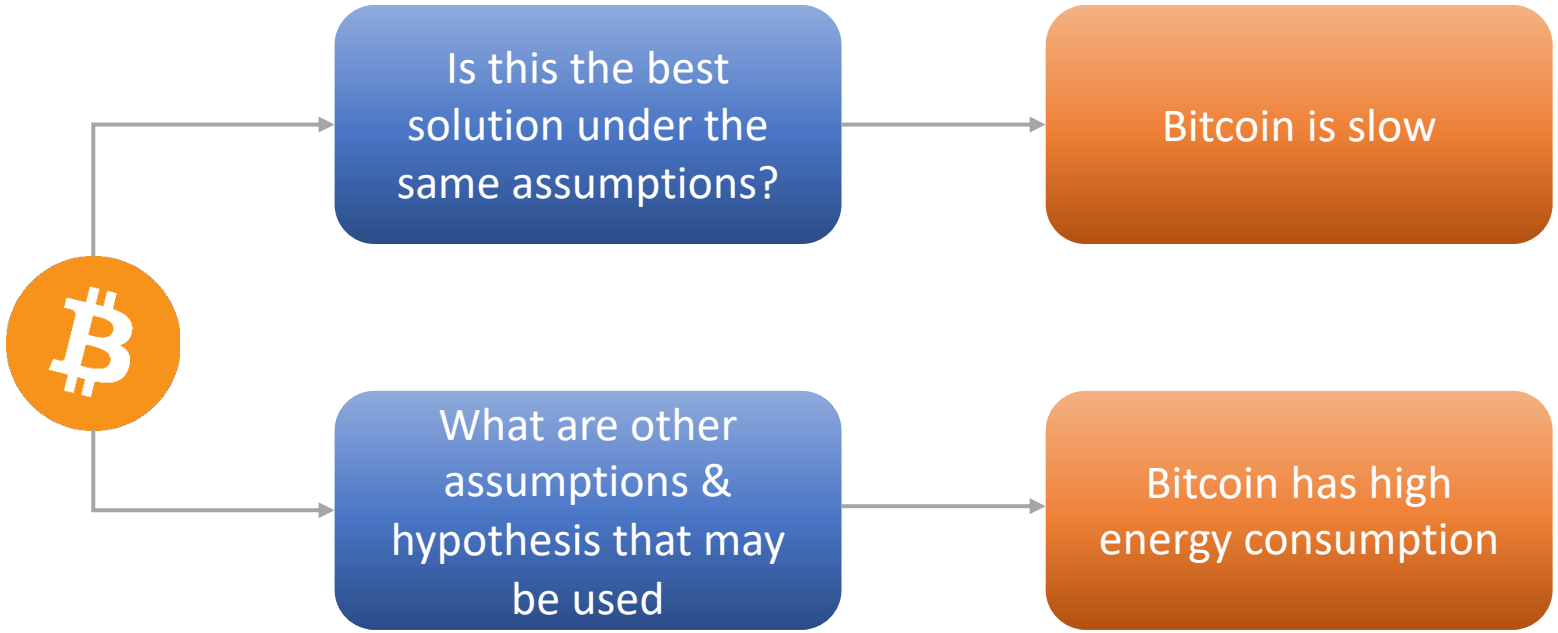


(Informal) Answer [Nak08]:

- The "*blockchain,*" based on Proofs of Work
- **Claim:** The blockchain realizes a "public ledger," assuming an *honest majority*
  - Consistency: Everyone sees the same history
  - Liveness: Everyone can add new transactions

Proofs of Stake

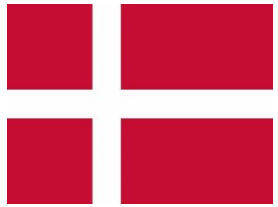
Bitcoin Challenges



~ 2000 tx/s

~ 100 tx/s

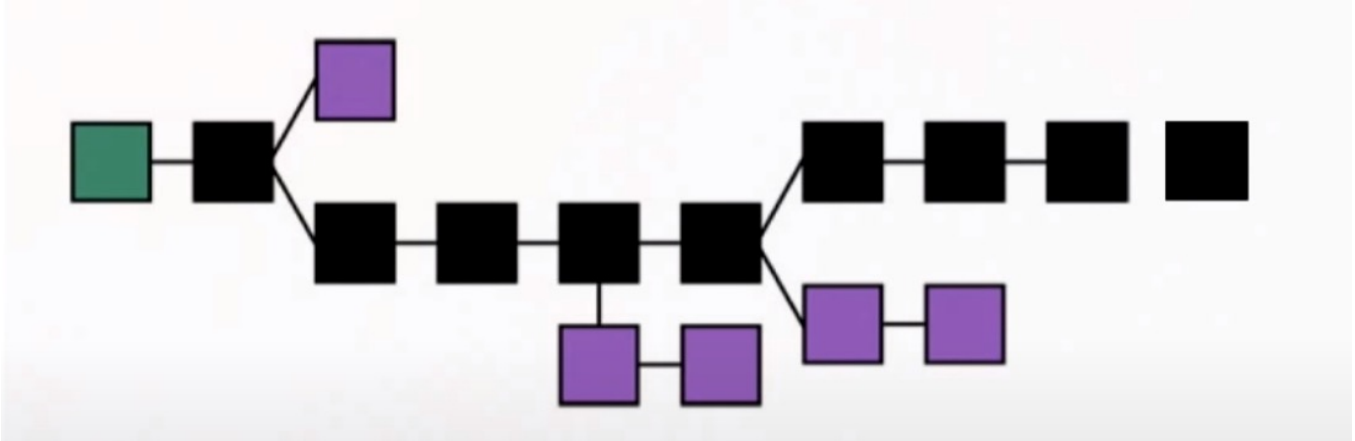
~ 15 tx/s



In 2020

Proof-of-Stake Background

- Generating the next block in Bitcoin is like an election



- A miner is elected with probability proportional to its hashing power
- "Collisions" may occur but they can be solved by the longest chain rule or a similar concept

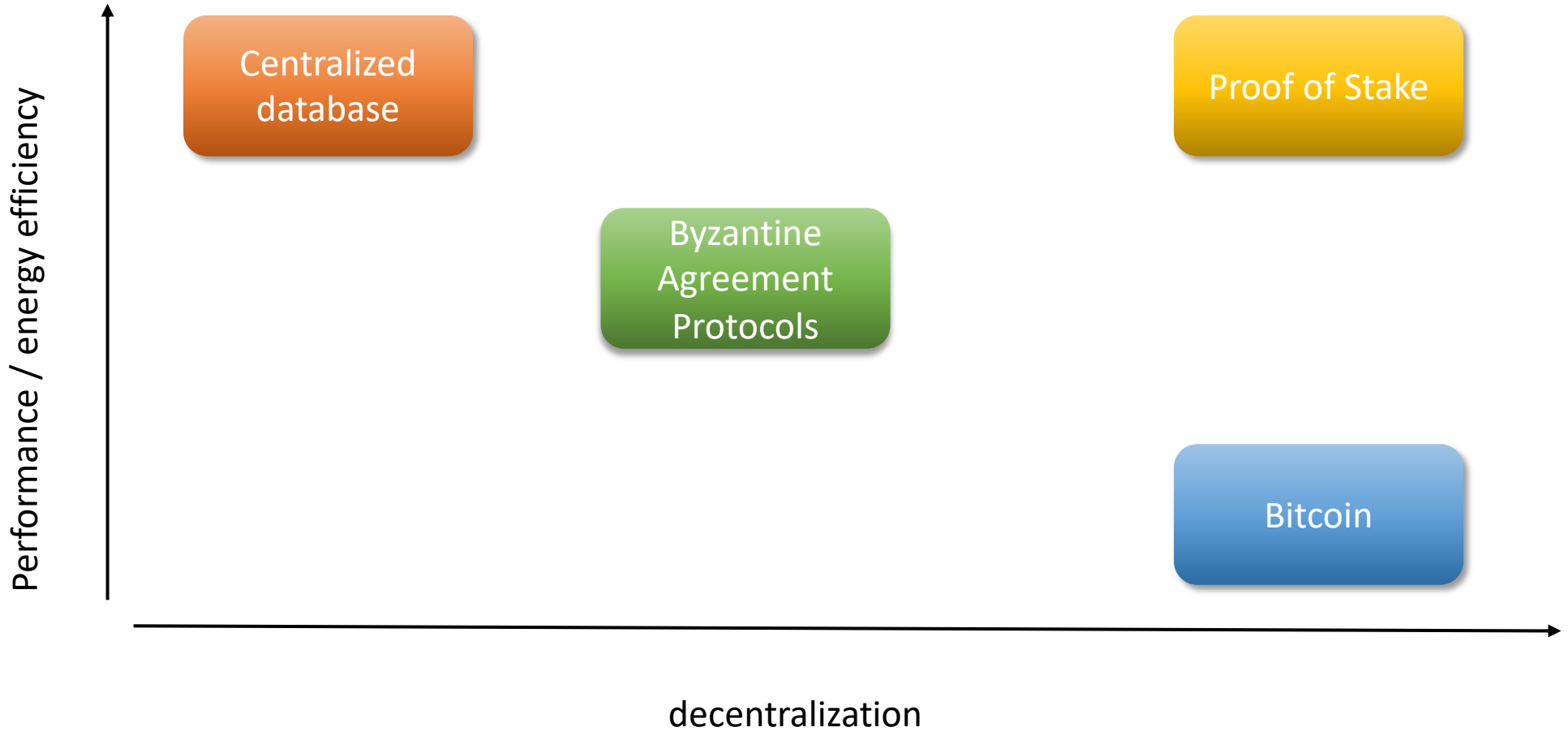
Proof of Stake

- Use stake (a ***virtual*** resource) instead of hashing power (a **physical** resource)

CARDANO

Algorand

Performance vs Decentralization



Proof-of-Stake Approaches

- PoS blockchains. Employ hash chains, digital signatures and some form of longest chain rule
  - E.g., Ouroboros, Snow White, Nxt
- PoS BFT. Adapt classical Byzantine fault-tolerant protocols to operate in the PoS setting
  - E.g., Algorand
- Both approaches are classified as PoS since protocol participation is based on the "proof of stake" primitive
- Fundamental enabling tool: ***Verifiable Random Functions*** (VRFs)

Verifiable Random Functions (VRFs) [MRS99]

- Cryptographic primitive that maps inputs to *verifiable* pseudorandom outputs
  - I.e., once a key pair (public key, secret key) and an input *X* are fixed, a VRF produces a unique pseudorandom **verifiable** output
- VRFs are the public-key version of a *keyed* cryptographic hash
  - Only the holder of the private key can compute the hash, but anyone with public key can verify the correctness of the hash


Thank You

Program

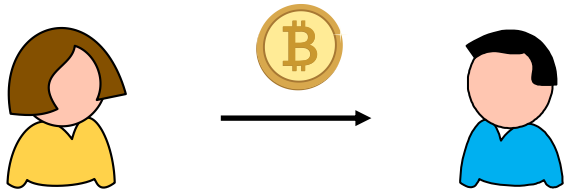
Monday, May 1st		
10:00 AM - 10:30 AM	Coffee	
10:30AM - 11:00 AM	Juan Garay, Texas A&M	Welcome & Blockchain Warm Up
11:00 AM - 11:30 PM	Le Xie, Texas A&M	Blockchain and Energy: Understanding the Impact of Cryptomining on the Electric Grid
11:30 PM - 12:00 PM	Korok Ray, Texas A&M	Banking in Bitcoin
12:00 PM - 1:15 PM	Lunch (provided)	
1:15 PM - 2:15 PM	Oshani Seneviratne, RPI	Empowering Decentralization through Algorand's Smart Contracts
2:15 PM - 2:30 PM	Ishan Dhanani, TAMU Blockchain Club	RevPass: Revolutionizing the Sports Passes
2:30 PM - 3:00 PM	Coffee Break	
3:00 PM - 4:00 PM	Tal Rabin, UPenn	YOSO: You Only Speak Once – Secure MPC with Stateless Ephemeral Roles

The Bitcoin Model (3)

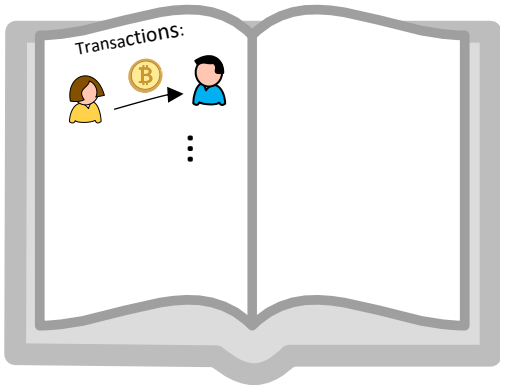
- Not a traditional distributed system
  - Nodes known *a priori* and authenticated
  - Paxos [Lam78], Raft [OO14] , Byzantine Fault Tolerance, Secure Multi-party Computation [Yao82, GMW87, BGW88,…]
- The “permissionless” model
  - Nodes do *not* know each other (not even their exact number!)
  - Nodes come and go
  - *Anyone* can join
- Can anything meaningful be done in such model?

Blockchains

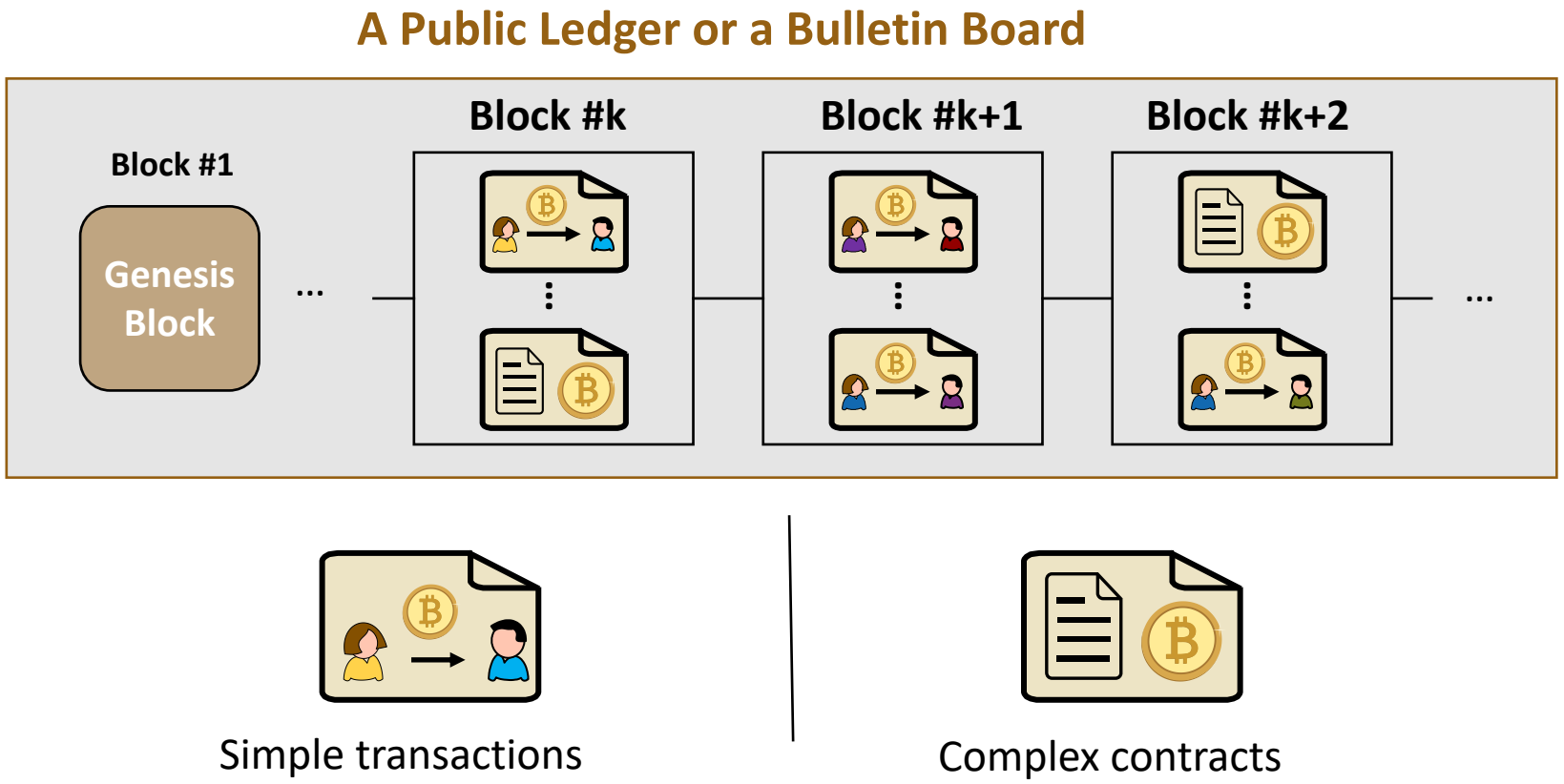
1) The problem: digital money transfer



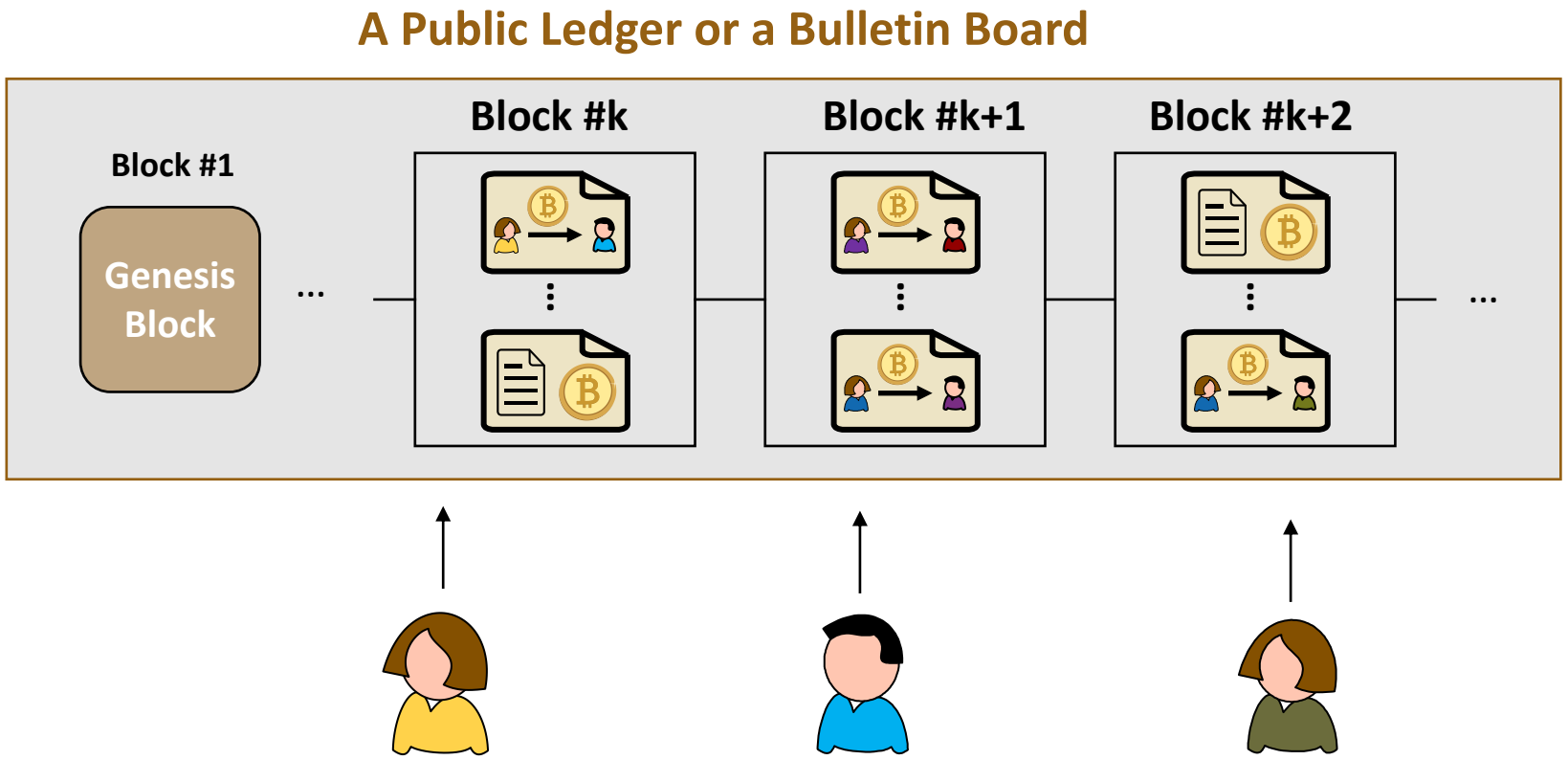
2) The accounting: implement a ledger!



Blockchains

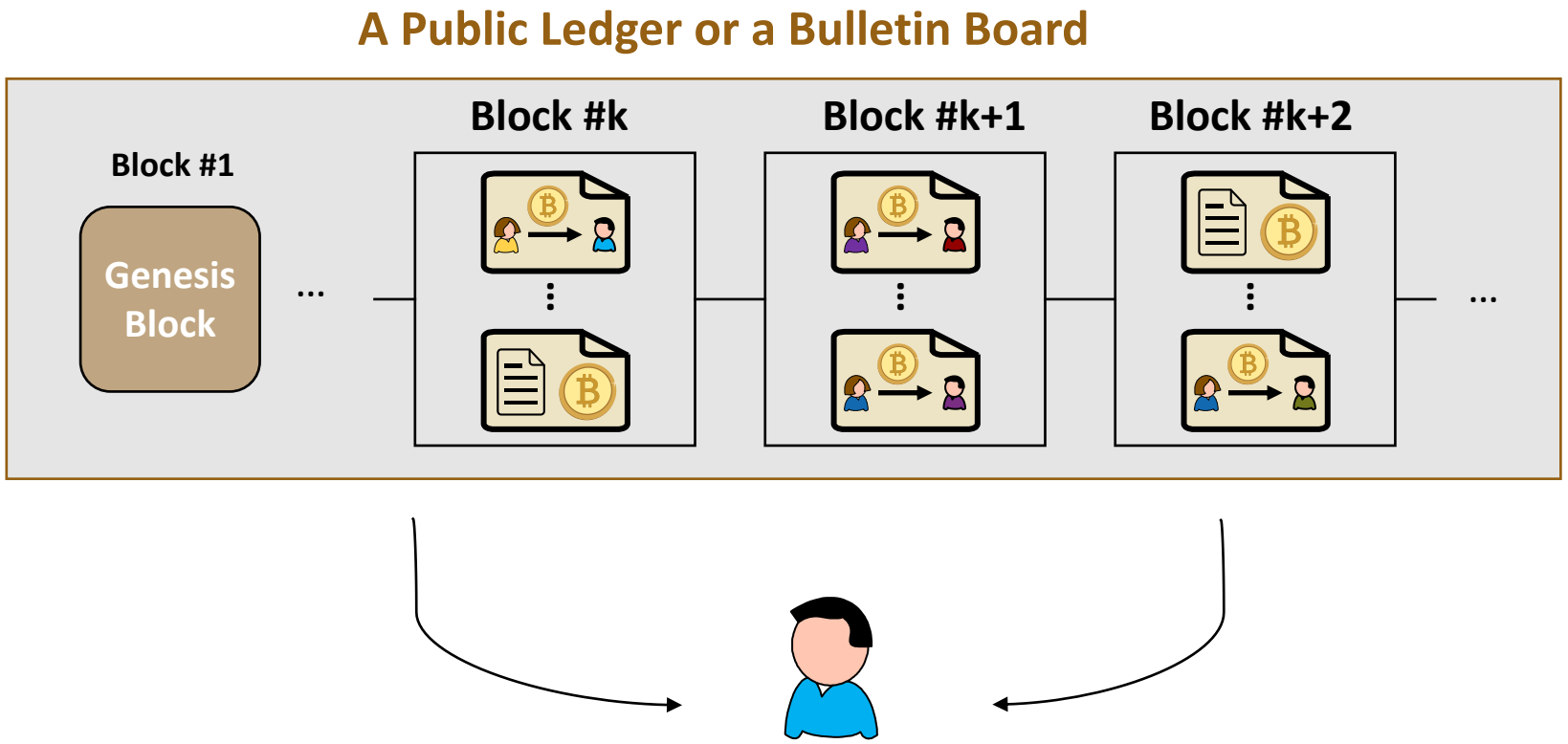


Blockchains



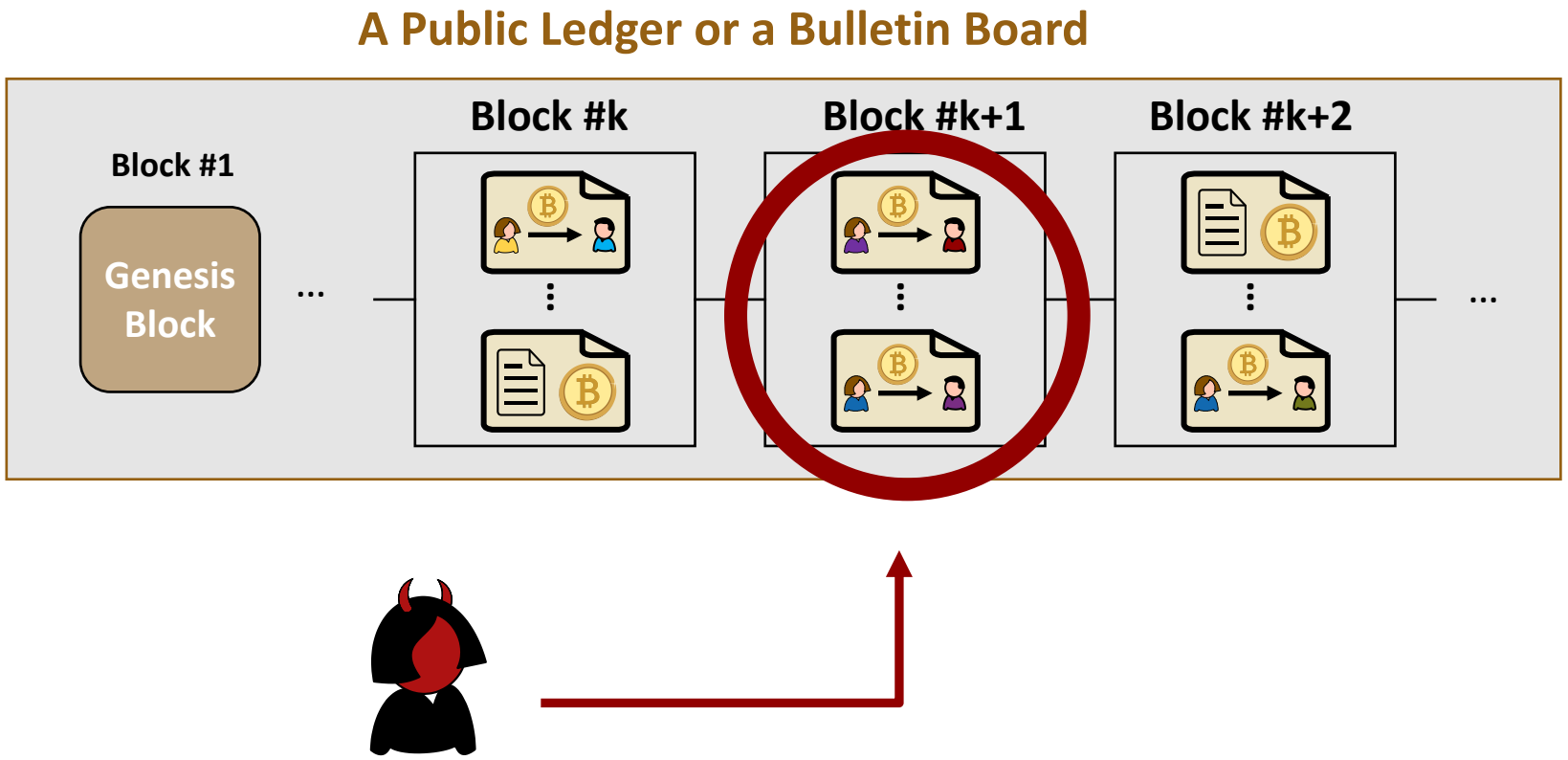
1) Content is provided by any user with sufficient funds
2) Users remain anonymous (*pseudonymous)*

Blockchains



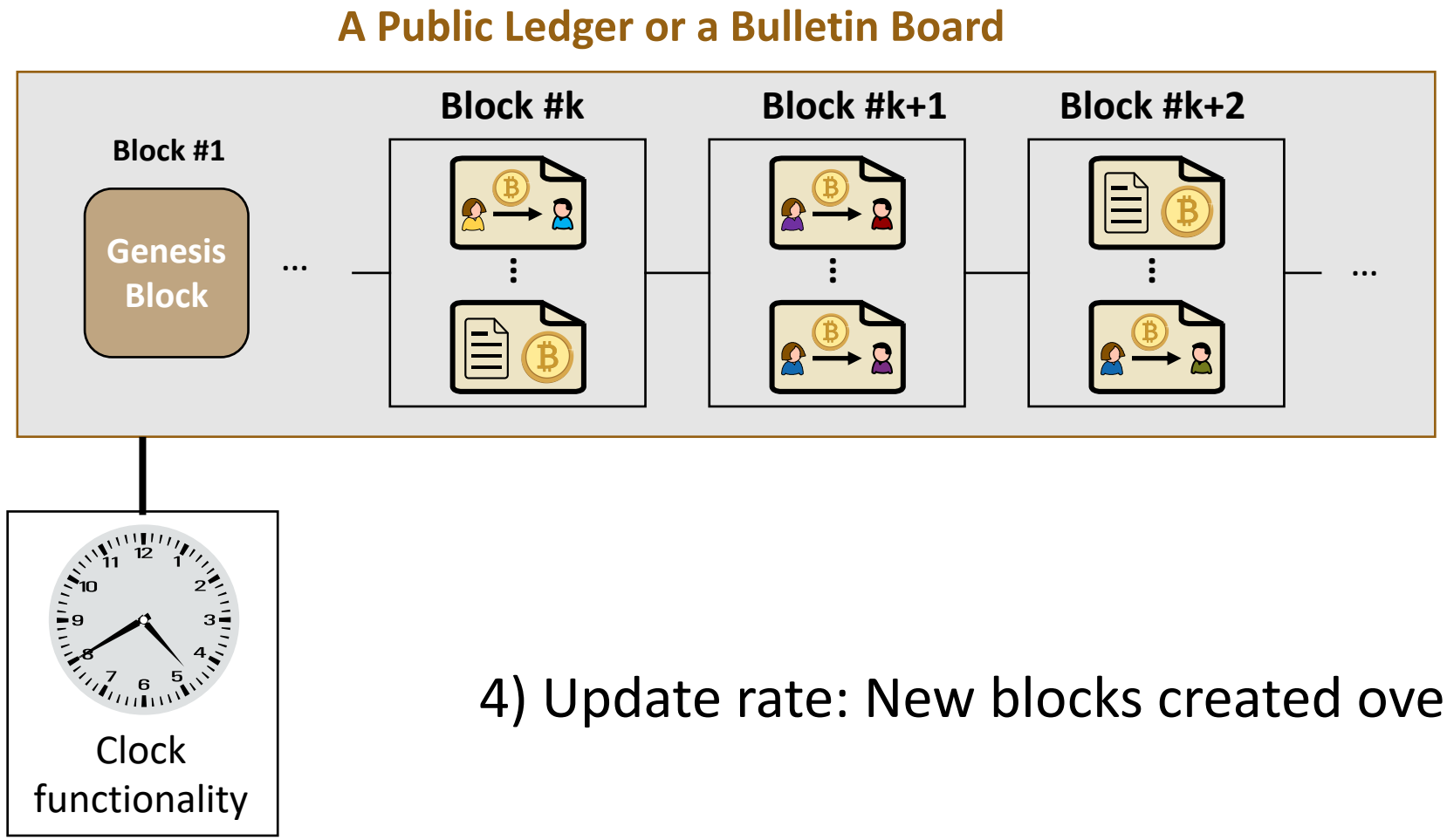
3) *Anyone* can read the current content of the ledger

Blockchains



3) No modifications of blockchain blocks possible

Blockchains



4) Update rate: New blocks created over time

Public *Distributed* Ledger

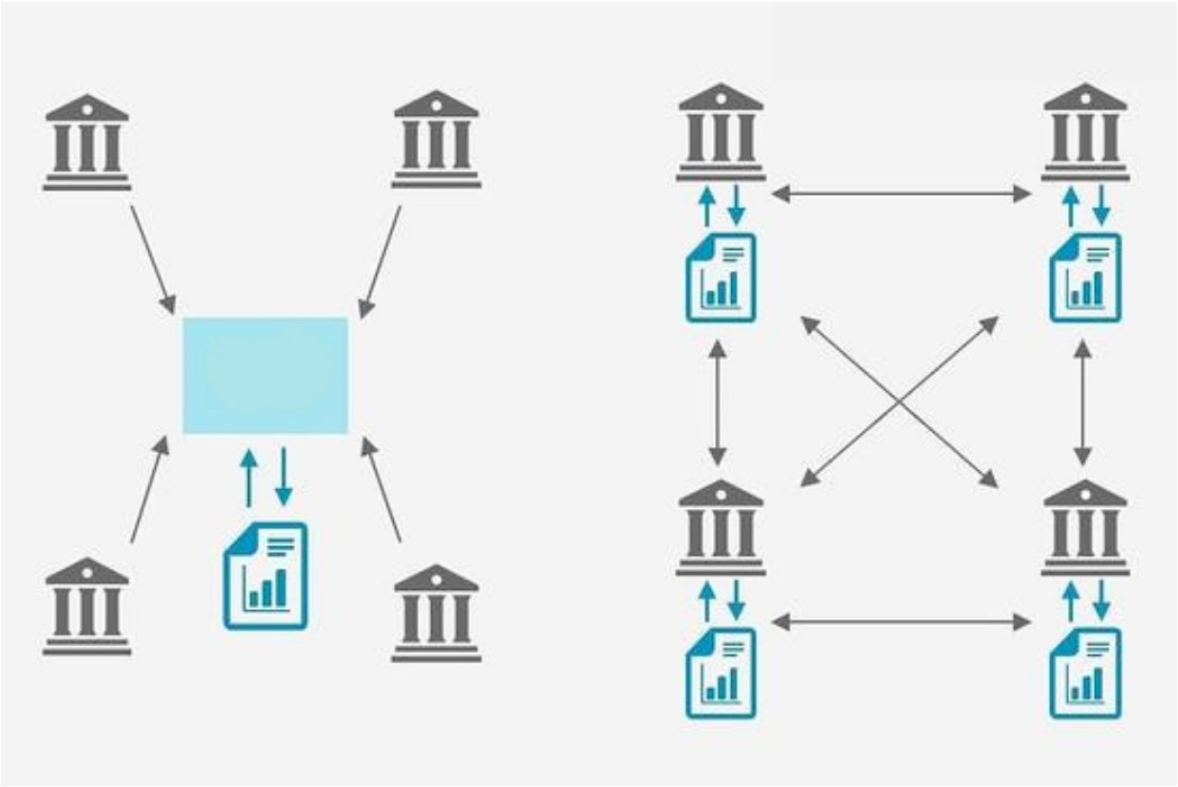


Figure: http://blogs.wsj.com/cio/2016/02/02/cio-explainer-what-is-blockchain/

- Consistency: Everyone sees the same history
- Liveness: Everyone can add new transactions

Talk Outline

- Proof-of-Work (PoW)-based Blockchains
- *Resource-Restricted Cryptography*
- PoWs in the “Standard” Model

PoW-based Blockchain Protocols (Bitcoin)

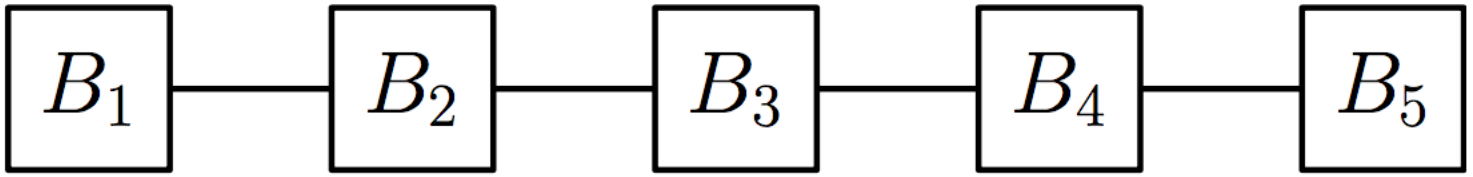
- Parties (“miners”) have to do work in order to install a transaction

PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks

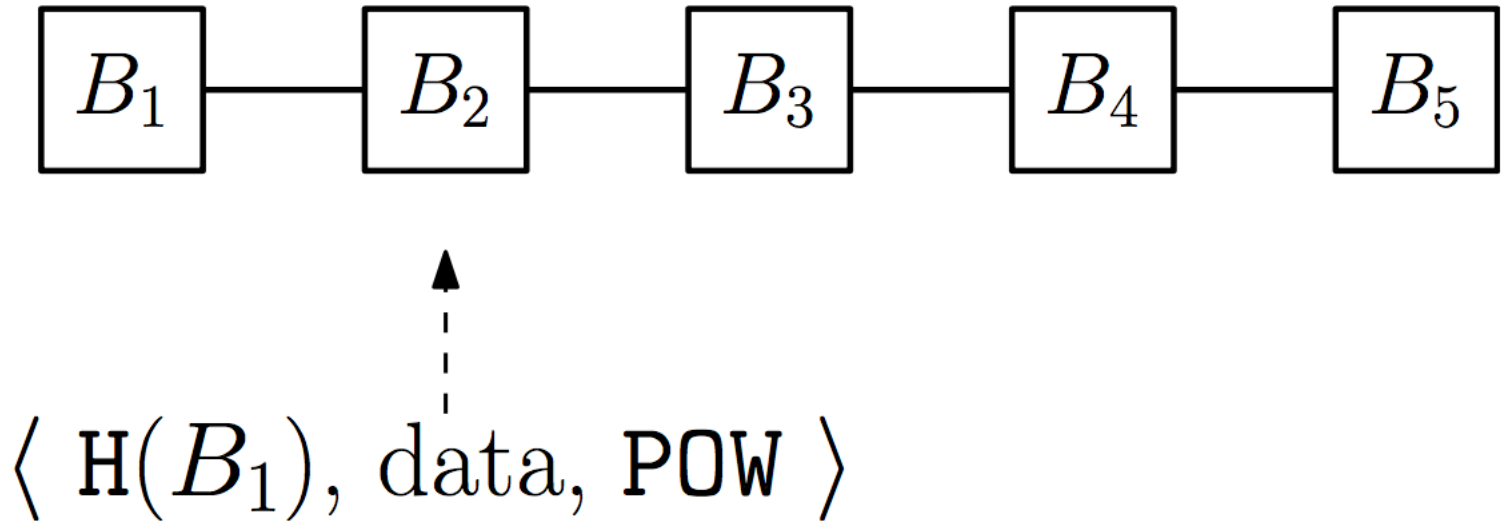
PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks



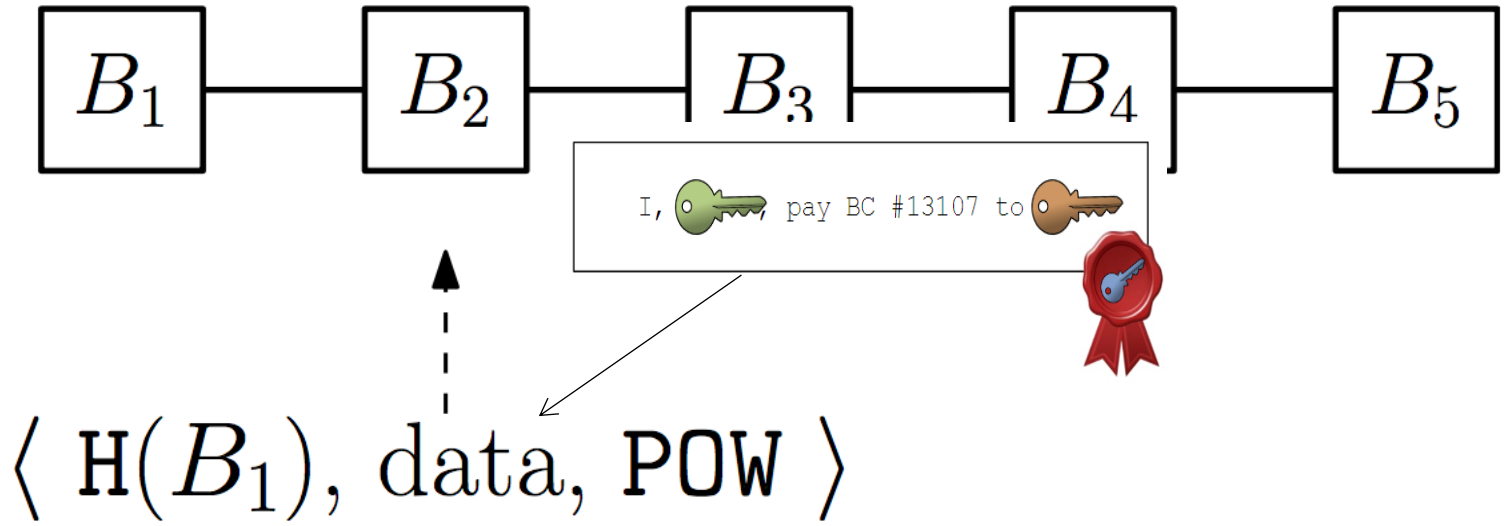
PoW-based Blockchain Protocols (Bitcoin)

- Parties ("miners") have to do work in order to install a transaction
- Transactions are organized in chains of blocks



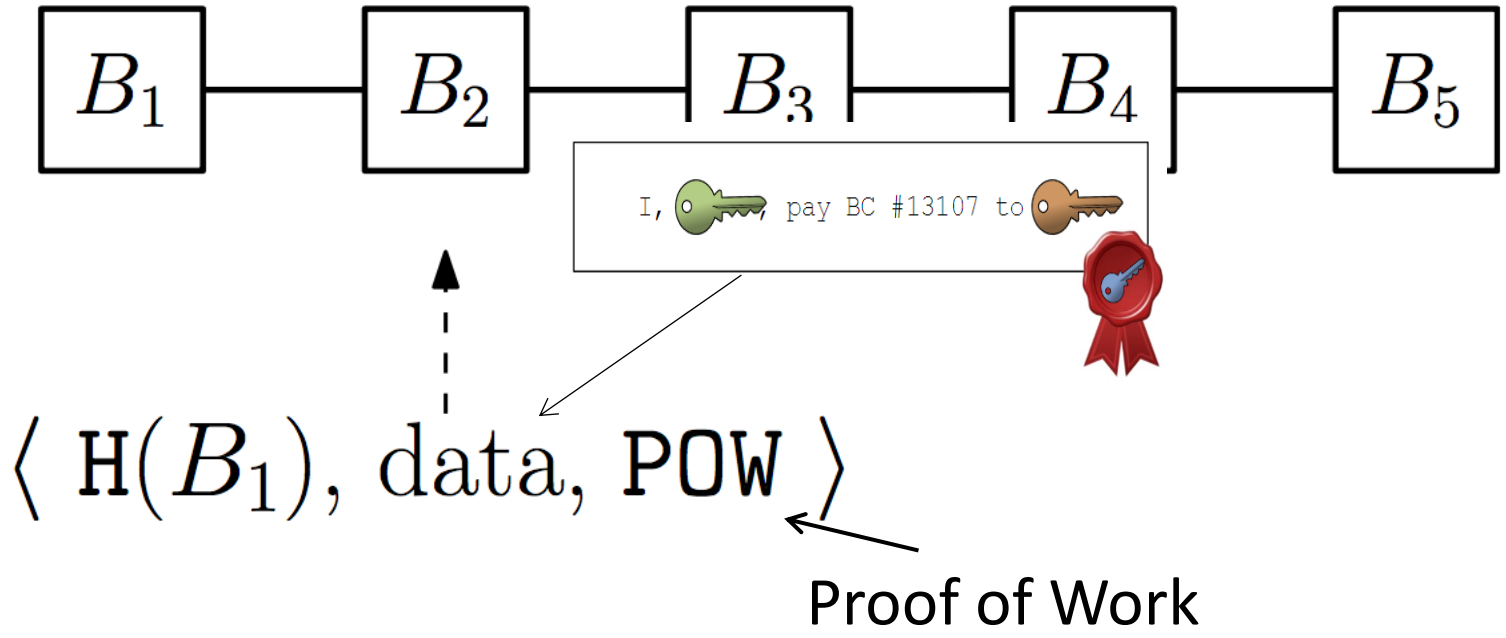
PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks

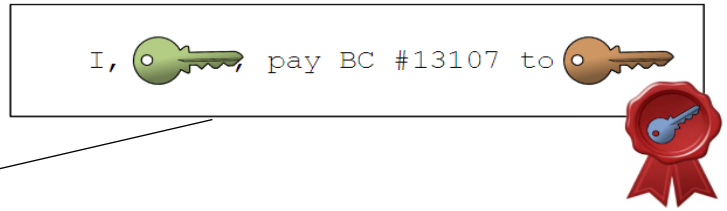


PoW-based Blockchain Protocols (Bitcoin)

- Parties (“miners”) have to do work in order to install a transaction
- Transactions are organized in chains of blocks



PoW-based Blockchain Protocols (Bitcoin) (2)



- Miners collect a set of transactions ("data")

  $\mathbf{tx} = (tx_1, tx_2, \ldots, tx_m)$

- Then do "work"

  ctr := 0; **while** Hash(ctr; Hash(τ,**tx**)) > T **do** ctr++

  T: block's "target" (*difficulty level*)

  (T = 0000000000000000171A8B0000000000000000000000000000000000000000)

- If **while** loop terminates "broadcast" (τ,ctr,**tx**)

  (new "block": state, counter, set of transactions)

SHA-256(·)

The Proof of Work Era

Last ~15 years:

- PoW-based blockchain protocols (e.g., Bitcoin [Nak08])

Proofs of Work (aka “Crypto Puzzles”)

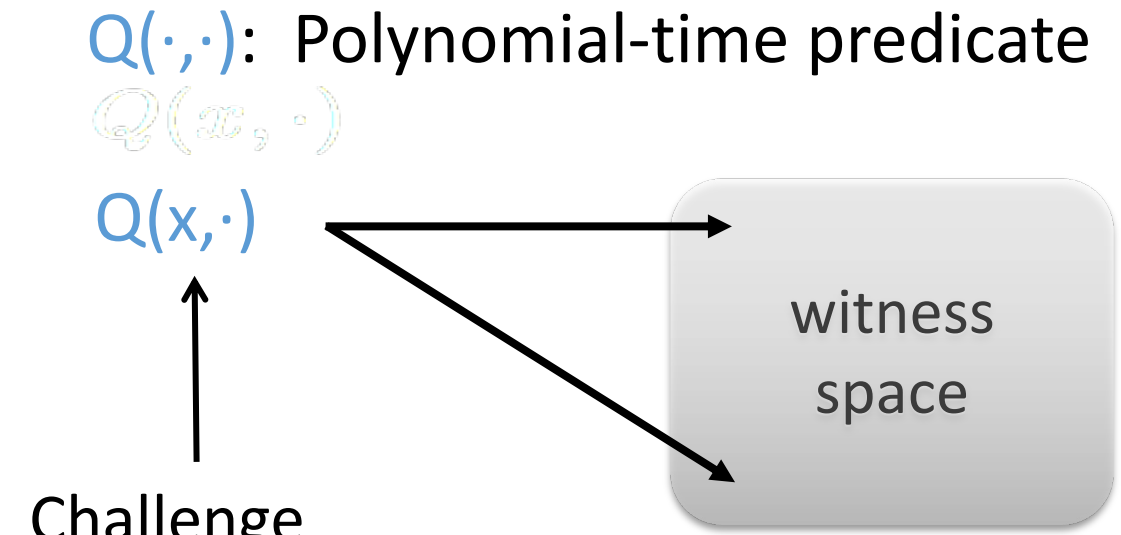
- “Moderately hard functions” [DN92,RSW96,Back97,JJ99,BN00,GMPY06,BGJPVW16, BRSV17/18…]

Prover **Verifier**

Image credit: Marshall Ball

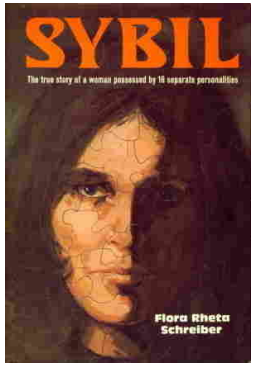
Proofs of Work (aka "Crypto Puzzles")

- "Moderately hard functions" [DN92,RSW96,Back97,JJ99,BN00,GMPY06,BGJPVW16, BRSV17/18…]

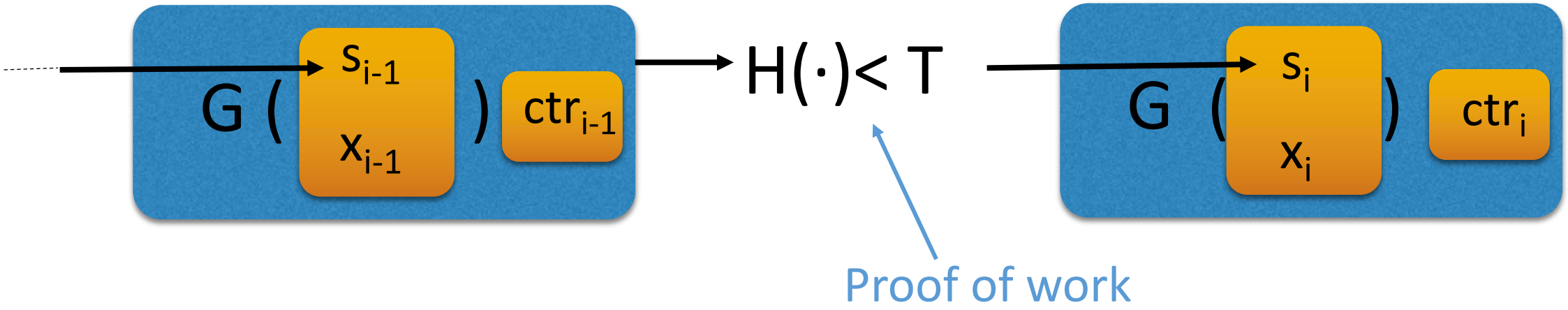


Successful only with some (small) probability!

Proofs of Work (aka “Crypto Puzzles”)



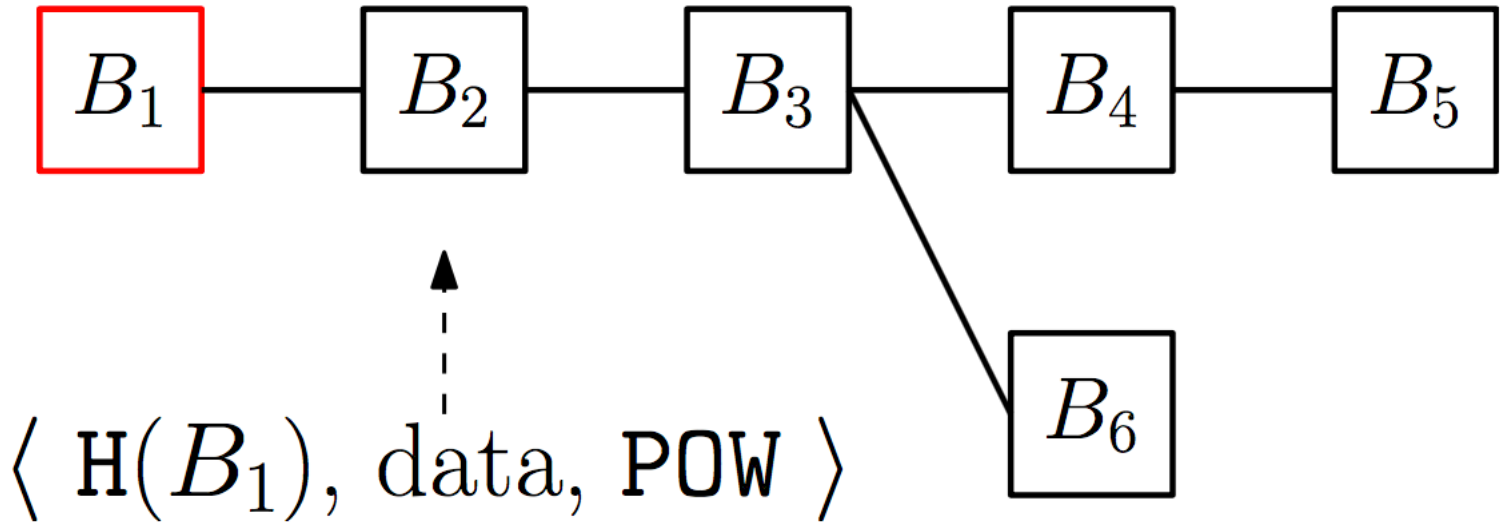
- Spam mitigation, Sybil attacks, denial of service protection, …
- Most impactful application: Design of blockchain protocols such as Bitcoin



$$\text{Hash}(ctr_{i-1}; \text{Hash}(s_{i-1}, x_{i-1})) < T$$

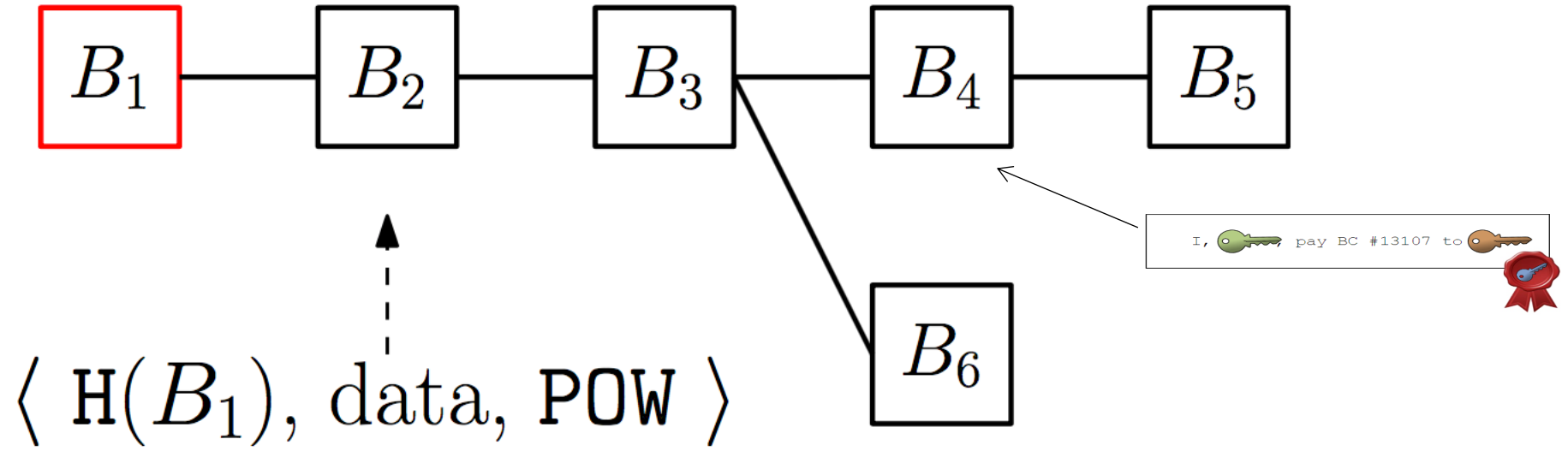
PoW-based Blockchain Protocols (Bitcoin) (3)

- Parties ("miners") always choose the *longest* chain they received



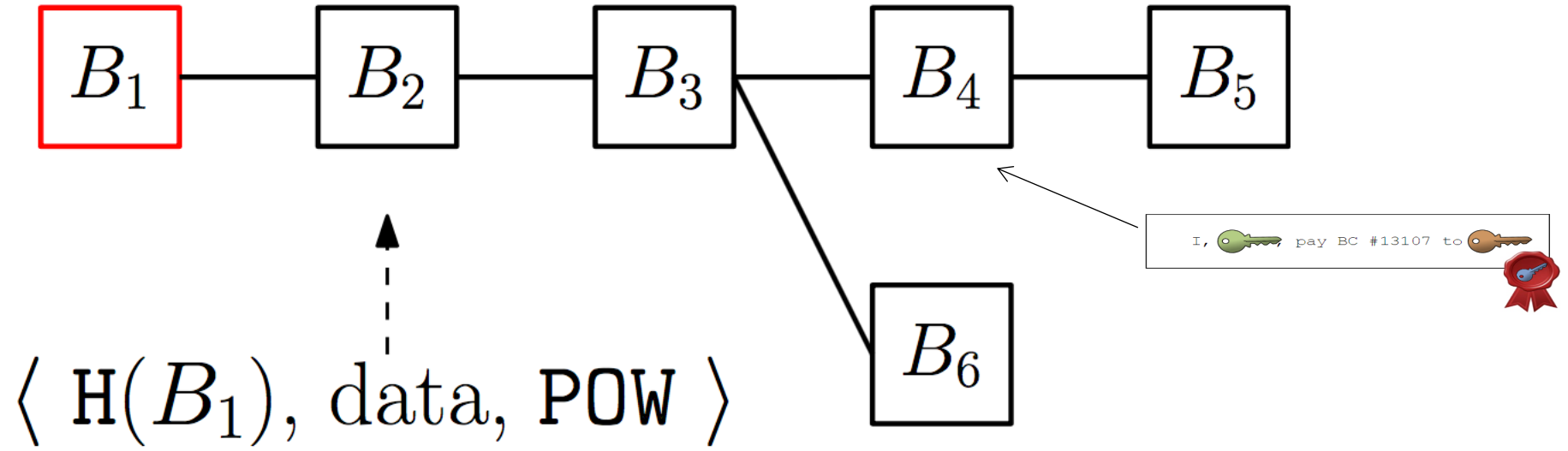
PoW-based Blockchain Protocols (Bitcoin) (3)

- Parties ("miners") always choose the *longest* chain they received
- If party wants to **erase** a transaction, it has to find a longer chain!



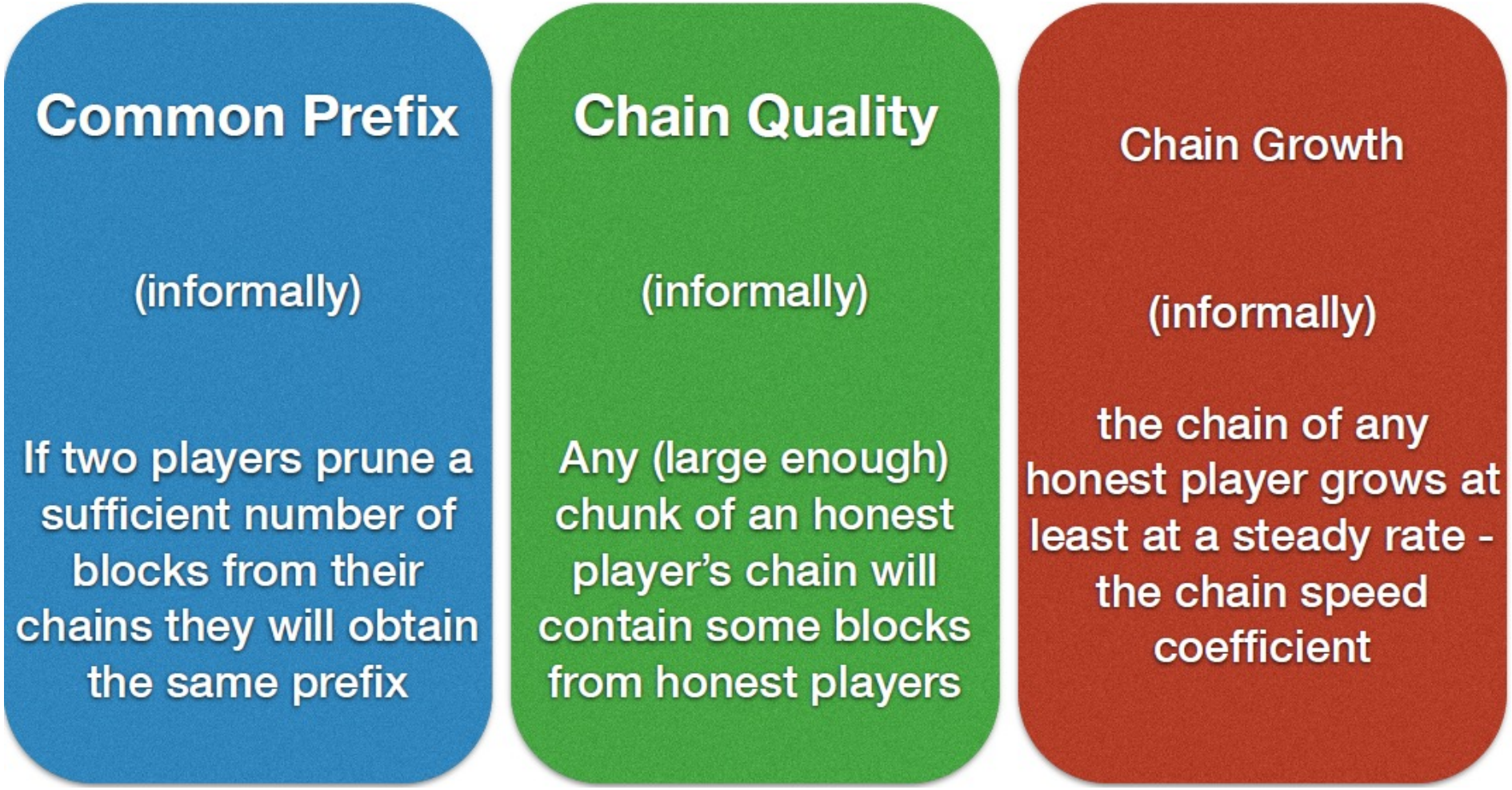
PoW-based Blockchain Protocols (Bitcoin) (4)

- Parties ("miners") always choose the *longest* chain they received
- If party wants to **erase** it transaction, it has to find a longer chain!



- If transaction is "sufficiently deep," it cannot do this unless it has a "majority of hashing power"

Basic Properties of the Blockchain [GKL15]



From Blockchain Properties to Applications [GKL15]

- Determine the parameters for which above properties (Common Prefix, Chain Quality, Chain Growth) hold with overwhelming probability (in the security parameter)
- Then show how an application's properties can be proven (in a black-box manner) using these properties

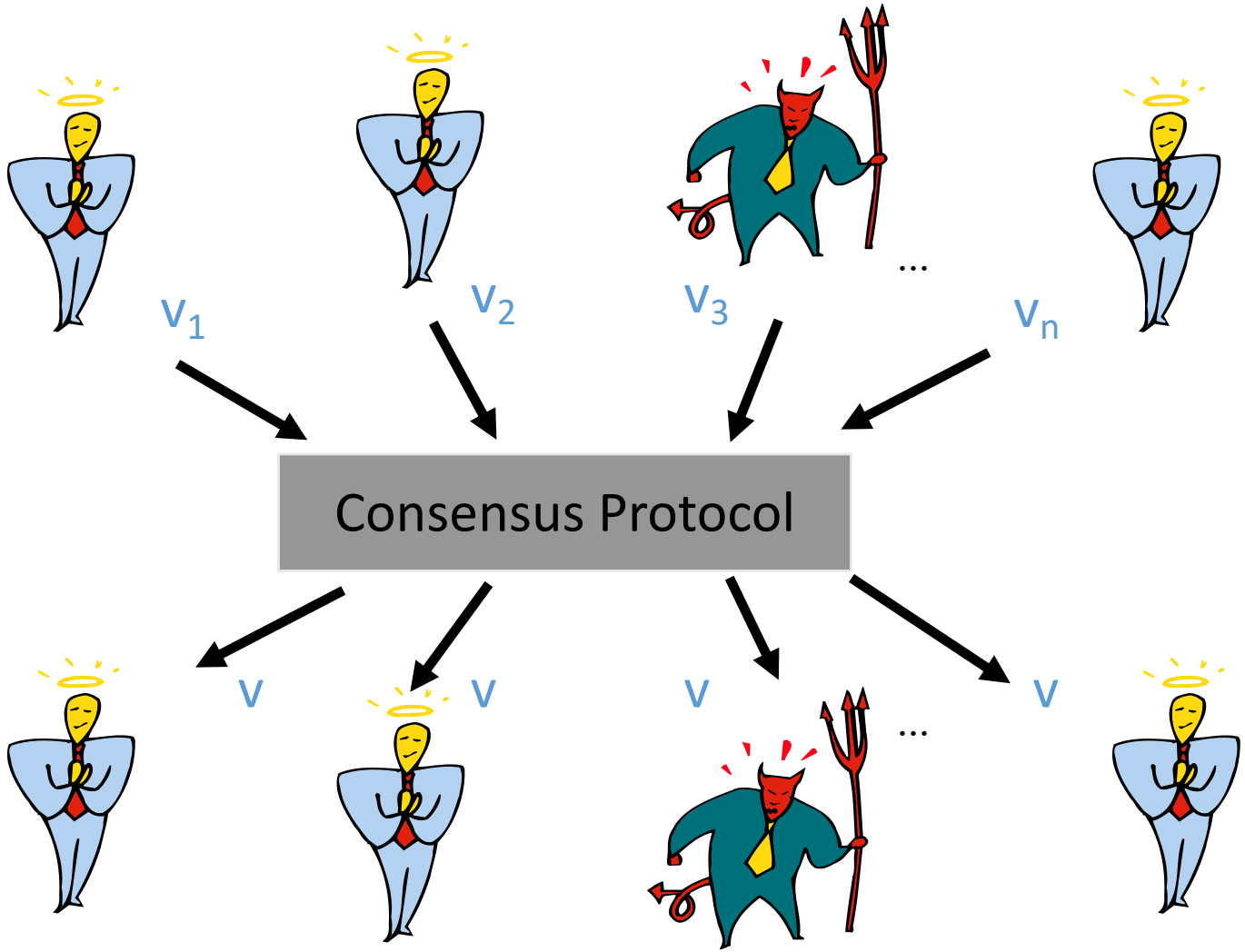
From Blockchain Properties to Applications [GKL15]

- Consensus
- Robust transaction ledger (Bitcoin)
  - Aka “ledger consensus,” “Nakamoto consensus”

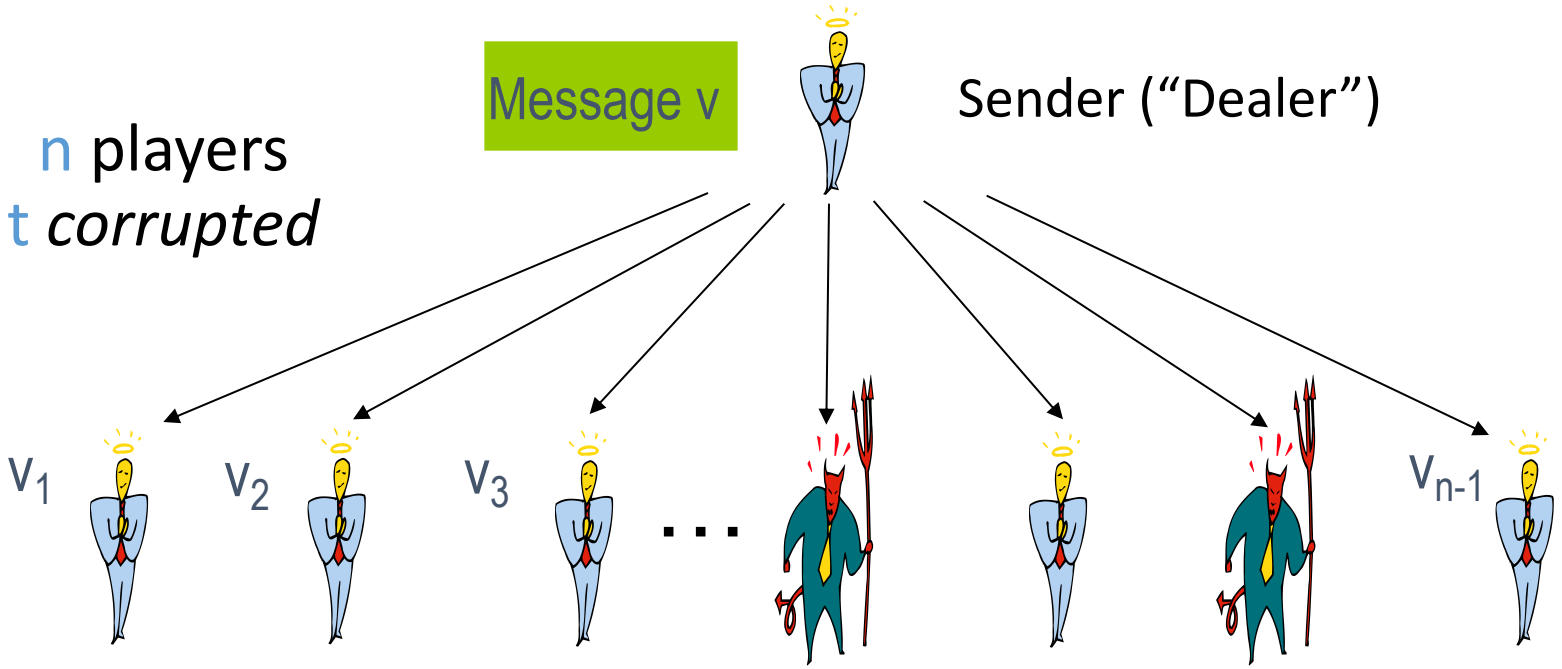
Consensus (*Byzantine Agreement*) [PSL80, LSP82]

n parties
t *corrupted*

Adversary



Broadcast (aka *Byzantine Generals*) [PSL80, LSP82]



- Validity: If dealer is honest, $v_i = v$
- Agreement: $v_i = v_j$
- Termination: Every player eventually outputs a value

Secure Multiparty Computation (MPC) [Yao86, GMW87,…]

Secure Multiparty Computation (MPC) [GMW87]:

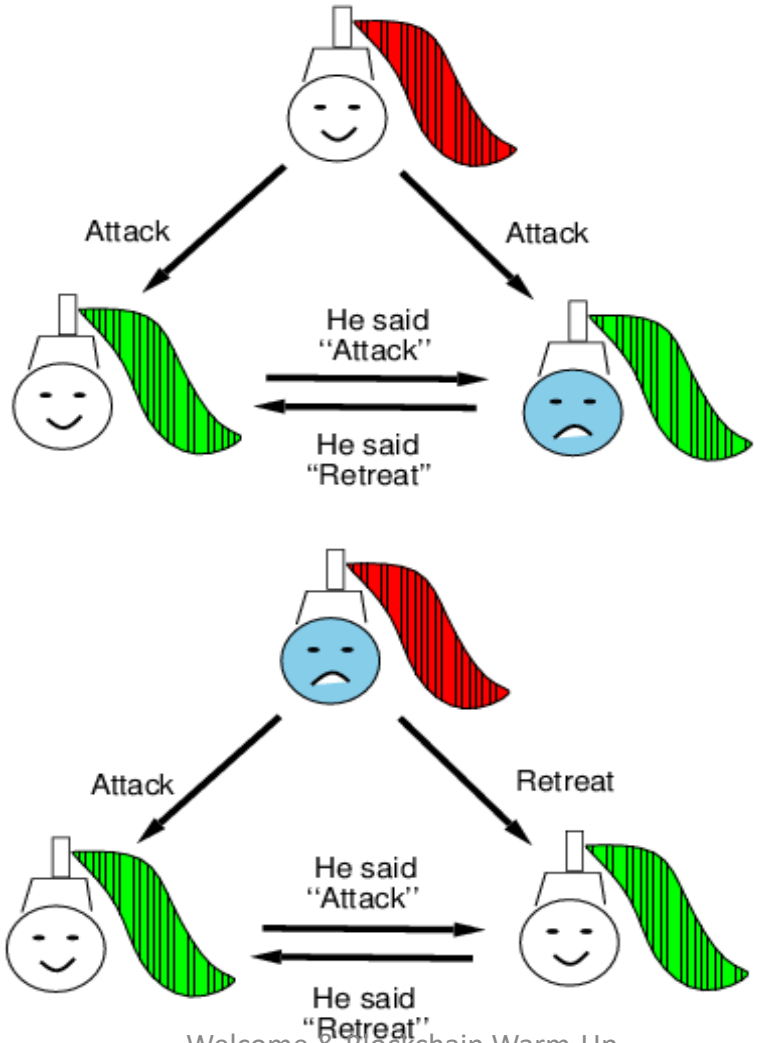
- n parties $\{P_1, P_2, \ldots, P_n\}$: each P_i holds a private input x_i;
  $t < n$ (maliciously) corrupted
- One public function $f(x_1,x_2,\ldots,x_n)$
- All want to learn $y = f(x_1,x_2,\ldots,x_n)$ (*Correctness*)
- Nobody wants to disclose his private input (*Privacy*)

Secure 2-Party Computation (2PC) [Yao 82, Yao 86]: $n=2$

Consensus/Broadcast

- One of the most fundamental problems in fault-tolerant distributed computing
- Fundamental MPC instance; important role in cryptographic protocols
- Renewed interest with the advent of blockchain protocols like Bitcoin
  - New protocol paradigms
  - Wider research community
  - Applications expanded to novel settings

Impossibility of Broadcast with t ≥ n/3 (and no crypto) [PSL80, LSP82]



On the Necessity of an Honest Majority for Consensus

- Above impossibility ($t \geq n/3$) assumes no cryptography (i.e., digital signatures) is used
- Can the bound on no. of corruptions be improved using cryptography? (And in particular, a *Public Key Infrastructure* (PKI) — called *"private (state) setup"* in the sequel?)

- Yes, but can't do better than

$$t < n/2$$

regardless of the resources available to the parties

Public vs Private State Setup

We know that:

- Private setup + $t < n/2 \Rightarrow$ MPC [GMW87,RB89,PW92,CDDH99]
- Public setup + $t \geq n/3 \Rightarrow$ **Broadcast impossible** [LSP82, Bor96]

On the Necessity of a PKI ("Private-State Setup") [Bor96]

Without a PKI, broadcast (consensus) is ***impossible*** when $t \geq n/3$
even if using cryptography

Public vs Private State Setup (2)

We know that:

- Private setup + $t < n/2 \Rightarrow$ MPC [GMW87,RB89,PW92,CDDH99]
- Public setup + $t \geq n/3 \Rightarrow$ **Broadcast impossible** [LSP82, Bor96]
- Broadcast impossible $\Rightarrow$ **MPC impossible**

In general:

Theorem

Assuming a public setup, $t \geq n/3$, private channels, enhanced TDP, a RO, then MPC is impossible.

These results were established over 20 years ago…

…but then Nakamoto showed up!

First PoW-based Consensus Protocol [GKL15]

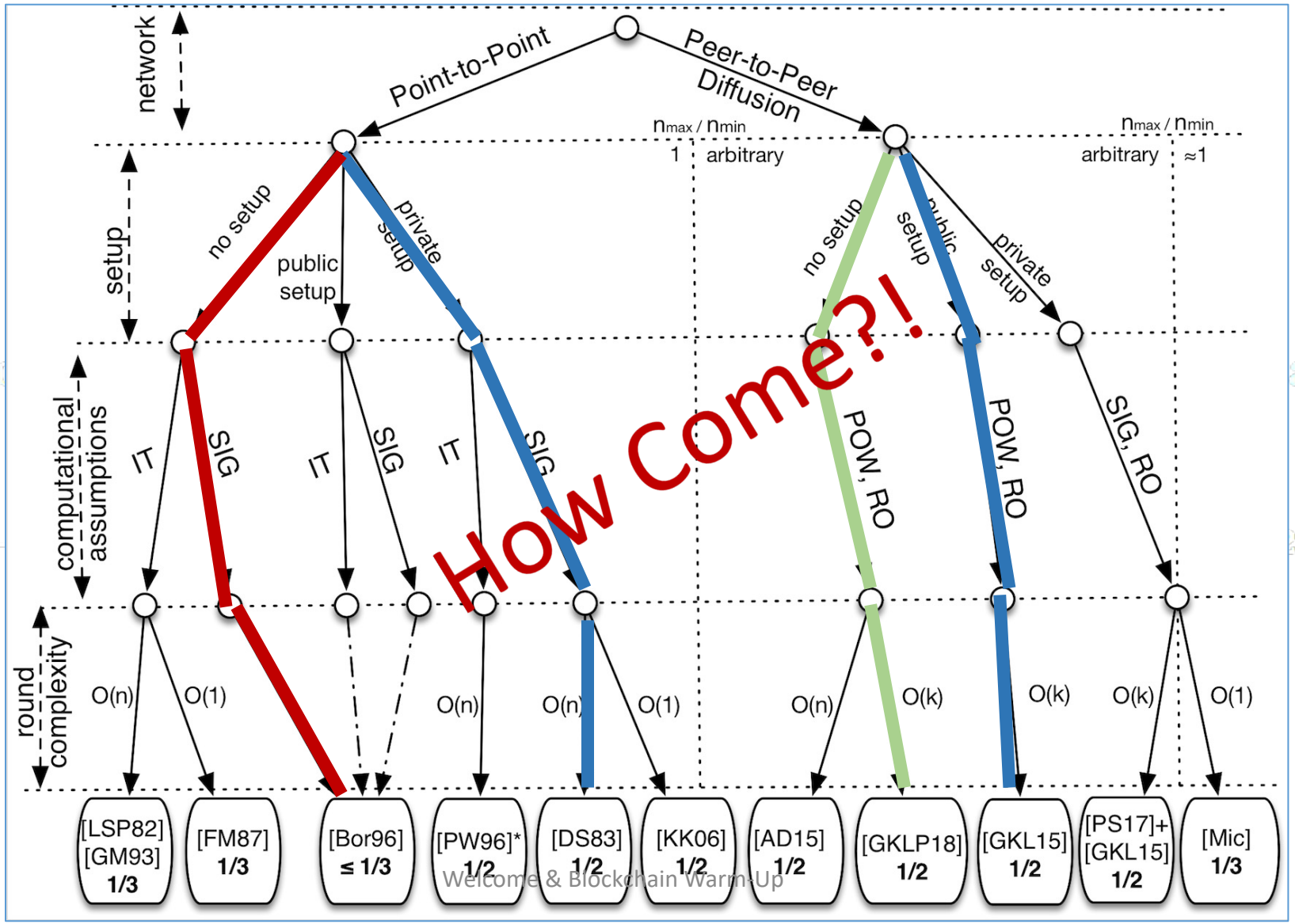
- The *n* parties start building a blockchain inserting their inputs
- If a party receives a longer blockchain switches to that one but *keeps the same input*
- Once the blockchain is long enough (2k) the parties prune the last k blocks and output the *majority value* in the prefix

- We get:
  - Agreement from the Common Prefix property
  - Validity as long as adv. controls < ⅓ of the parties (tight, due to the Chain Quality property)

Observations [GKL15]

- Based on the basic Bitcoin backbone properties – CP, CQ, CG – we obtained a *probabilistic solution* for the consensus problem tolerating a 1/3 fraction of corrupted parties
- 1/3 is *suboptimal*
  - **Main obstacle:** The blockchain (backbone) protocol does not provide sufficient *chain quality*
  - We cannot guarantee we have enough blocks originating from honest parties $(1-\beta)\gamma$
- 1/2 can be achieved, using a more elaborate protocol – a technique we call ***2-for-1 PoWs***

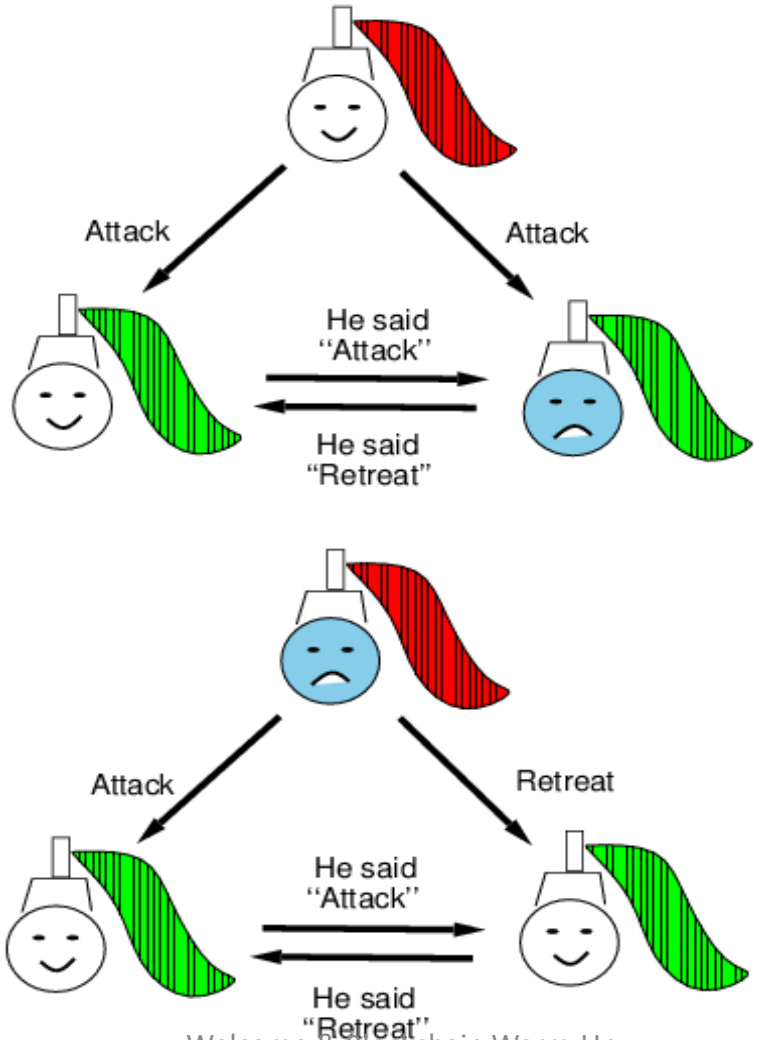
A Consensus Taxonomy [GK20]



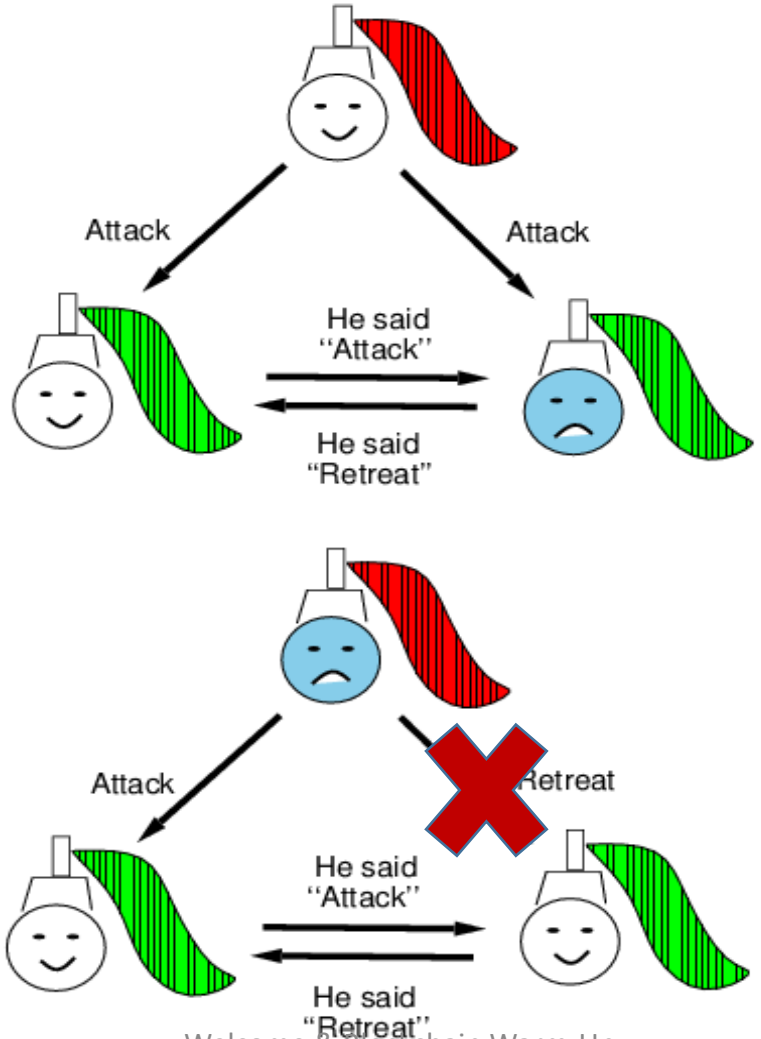
Resource-Restricted Cryptography

Impossibility with n = 3t [PSL80, LSP82]

RECALL



Resource-restricted Cryptography [GKOPZ20]



Resource-Restricted Cryptography

Long history:

- Moderately hard functions, key exchange, spam mitigation, time-released crypto, fair computation, PoWs [Mer76, DN92, RSW96, Back97, BN00, GMPY06, AD15, BGJPVW16, BRSV17/18,…]
- Many different *resources* considered: (sequential) computational power, space, stake, ...

Abstract resource: Network access

Secure Multiparty Computation (MPC) [Yao86, GMW87,…]

RECALL

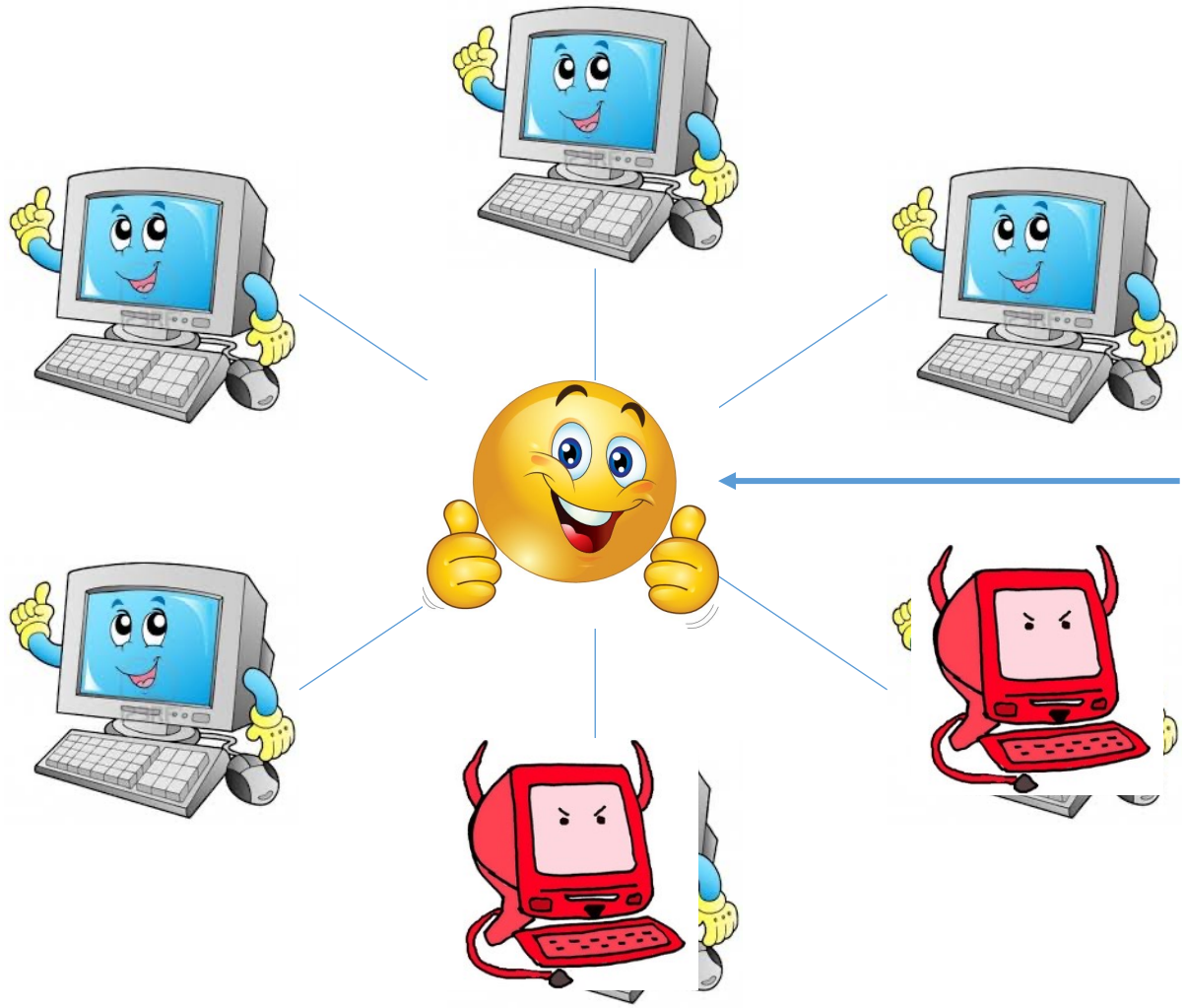
Secure Multiparty Computation (MPC) [GMW87]:

- n parties $\{P_1, P_2, \ldots, P_n\}$: each P_i holds a private input x_i;
  $t < n$ (maliciously) corrupted
- One public function $f(x_1,x_2,\ldots,x_n)$
- All want to learn $y = f(x_1,x_2,\ldots,x_n)$ (*Correctness*)
- Nobody wants to disclose his private input (*Privacy*)

Secure 2-Party Computation (2PC) [Yao 82, Yao 86]: n=2

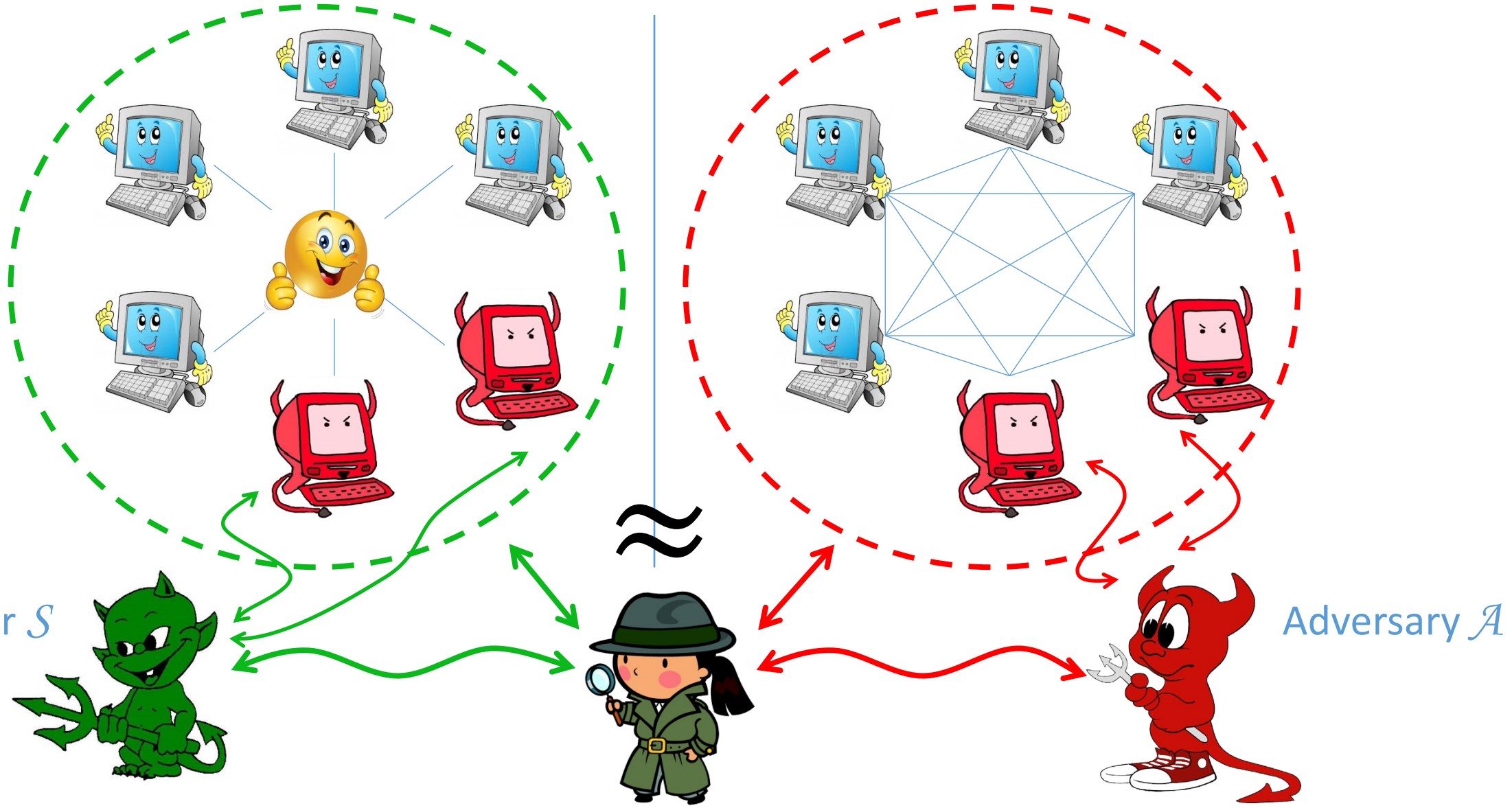
Detour:
Simulation-based Security

Simulation-based Security: Ideal World [GMW87, C01,…]

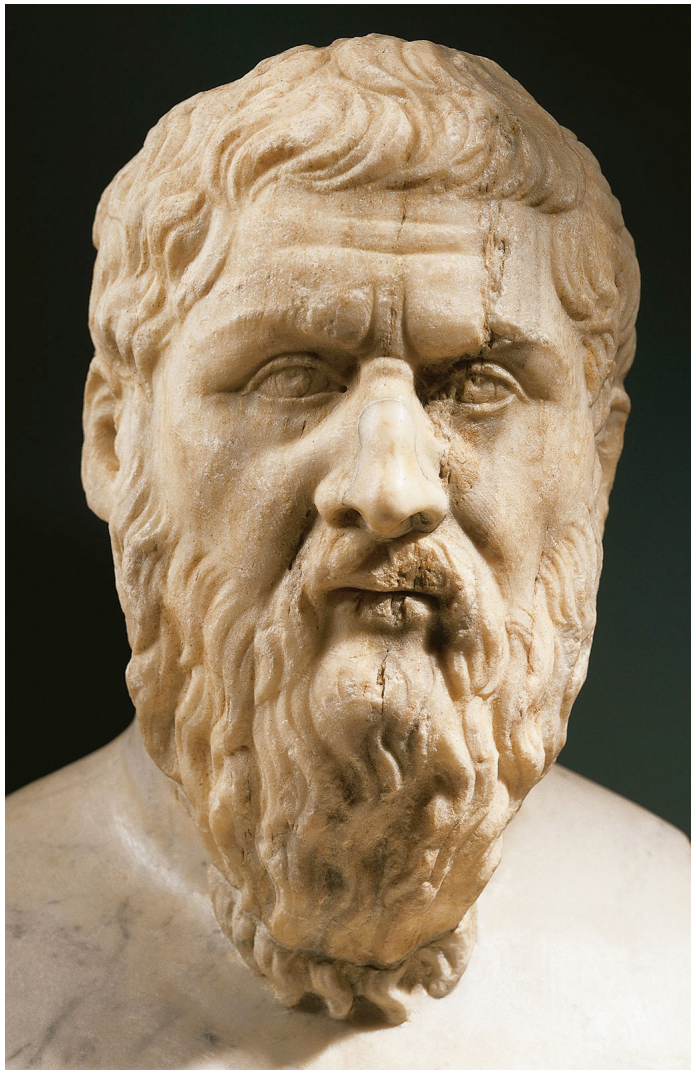


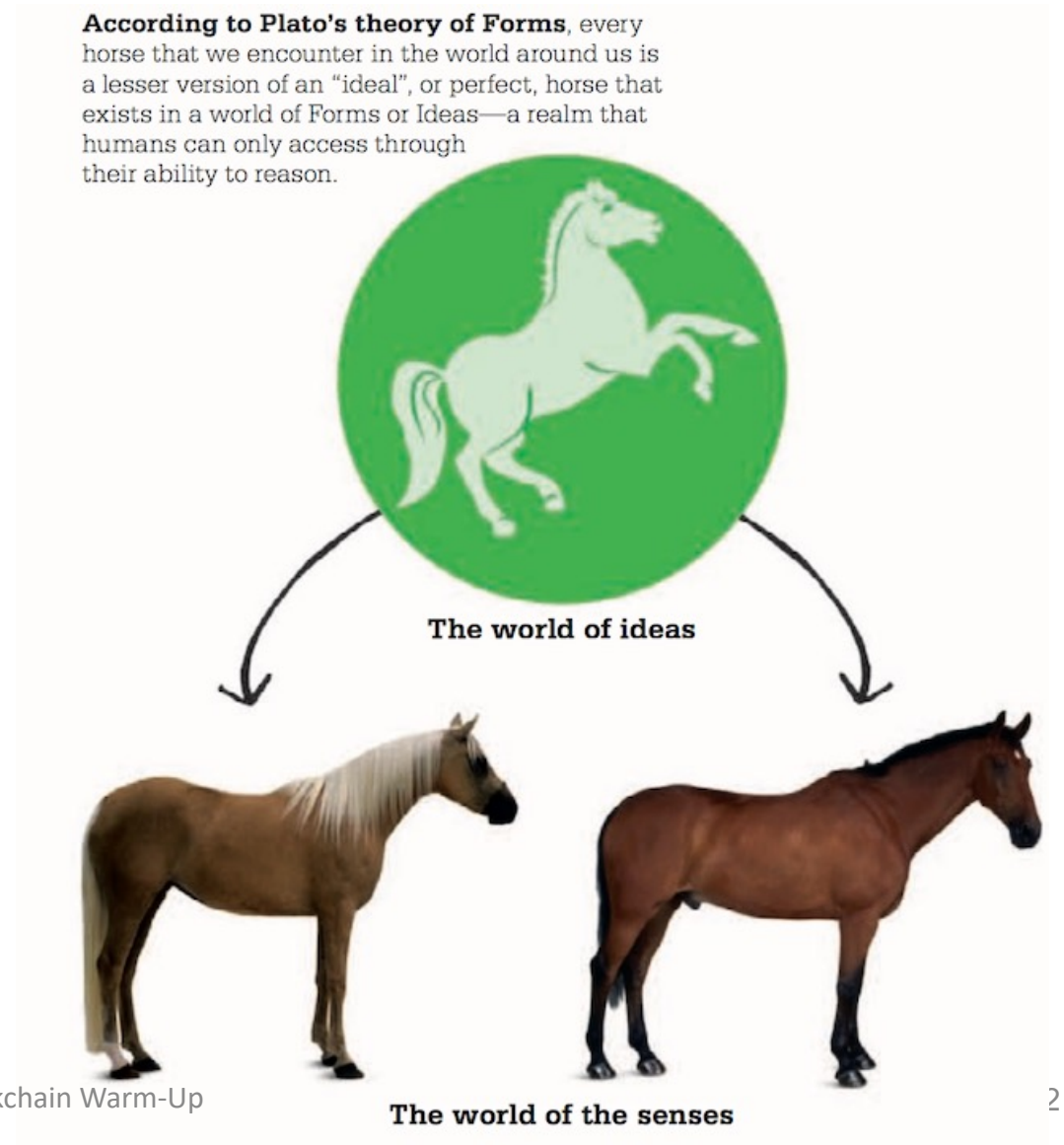
"Ideal functionality" $\mathcal{F}$ (arbitrary distributed computation)

Simulation-based Security: Real vs Ideal [GMW87, C01,…]



Plato’s Theory of Forms

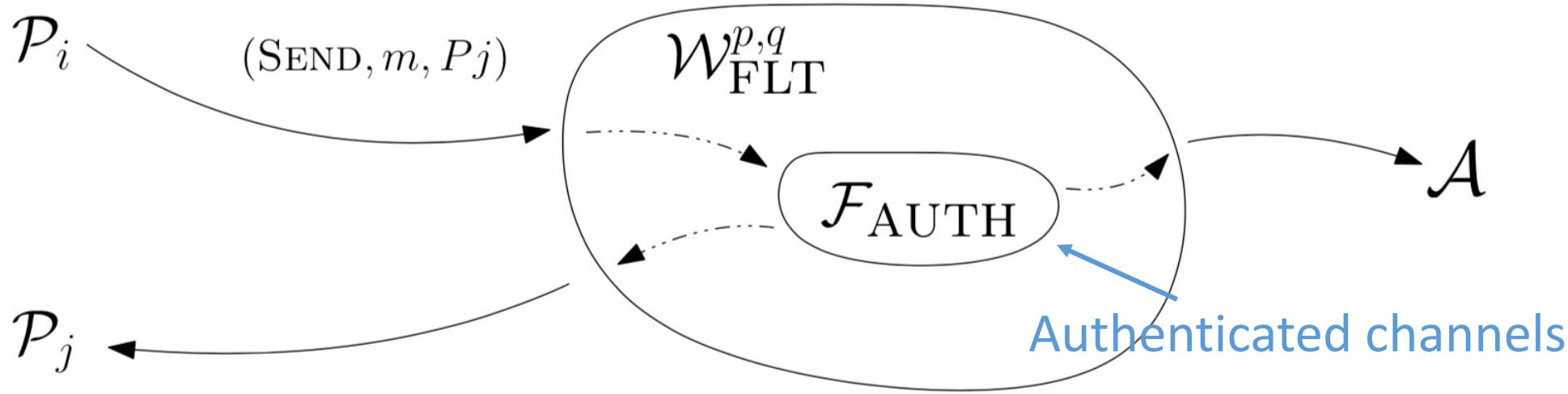




Simulation-based Security: Real vs Ideal [GMW87, C01,…] (2)

- Enables modular security proofs
  - Can prove secure a sub-task/protocol independently (i.e., the protocol realizes the corresponding ideal functionality), and then assume that the main protocol has access to this ideal resource
  - This is called the *hybrid model*
- Enables (universal) composition with other protocols [Canetti01,…]

The Filtering *Wrapper* Functionality



Model restricted network access as a functionality *wrapper* $\mathcal{W}_{\text{FLT}}^{p,q}(\cdot)$

- It models ability to "speak" only if party produces a PoW
- *Probabilistic access*: New messages sent with probability *p*
- *Bounded access*: *q* Send attempts per round
- *Free forwarding*
- Can wrap different types of networks: authenticated, private,…

MPC Feasibility

Does the RR Crypto paradigm allow for MPC with an **honest majority** and **public setup**?

MPC Feasibility

Does the RR Crypto paradigm allow for MPC with an **honest majority** and ***public*** setup? **Yes!**

2. $\mathcal{W}_{\mathrm{FLT}}^{p,q}(\mathcal{F}_{\mathrm{AUTH}}) + \mathcal{F}_{\mathrm{SIG}} \Rightarrow \mathcal{F}_{\mathrm{REG}}$ [this work]
3. $\mathcal{F}_{\mathrm{REG}} + \mathcal{F}_{\mathrm{SIG}} \Rightarrow \mathcal{F}_{\mathrm{CERT}}$ [Can04]
4. $\mathcal{F}_{\mathrm{CERT}} \Rightarrow \mathcal{F}_{\mathrm{BA}}$ [HZ10]
5. $\mathcal{F}_{\mathrm{BA}} + \mathcal{F}_{\mathrm{SC}} \Rightarrow \mathcal{F}_{\mathrm{MPC}}$ [GMW87,RB89,CDDHR98]

(* all implications hold against *adaptive* adversaries)

Theorem

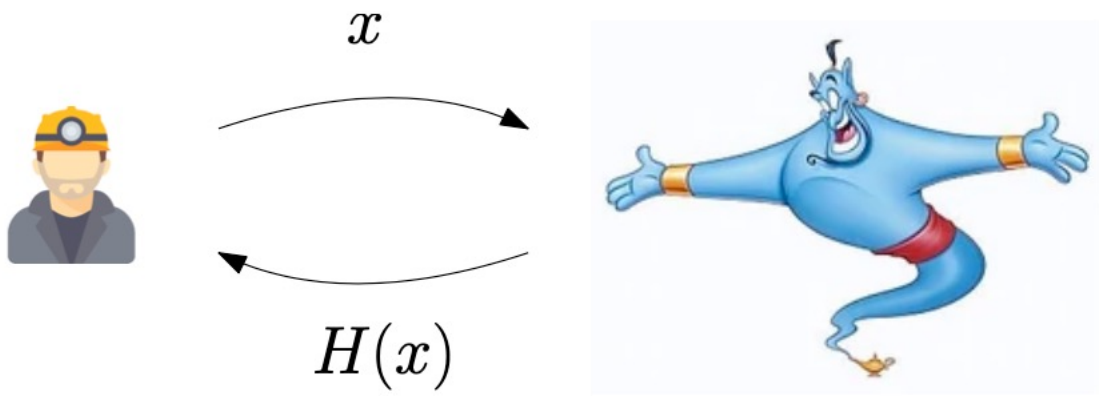
$\mathcal{F}_{\mathrm{MPC}}$ can be UC-implemented in the $(\mathcal{W}_{\mathrm{FLT}}^{p,q}(\mathcal{F}_{\mathrm{AUTH}}), \mathcal{F}_{\mathrm{SIG}}, \mathcal{F}_{\mathrm{SC}}, \mathcal{G}_{\mathrm{CLOCK}})$-hybrid model assuming $t < n/2$

Talk Outline

- PoW-based Blockchains
- *Resource-Restricted Cryptography*
- PoWs in the “Standard” Model
  - Based on hash-function properties
  - Based on fine-grained complexity

The ”Random Oracle” Methodology [BR93]

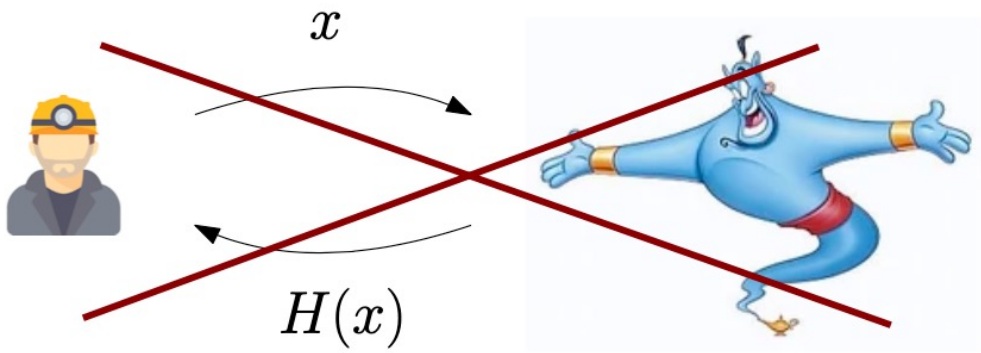
- Many blockchain protocols use hash-based PoWs
- **“Random-oracle”** methodology: Replace hash function with an *ideal* random function

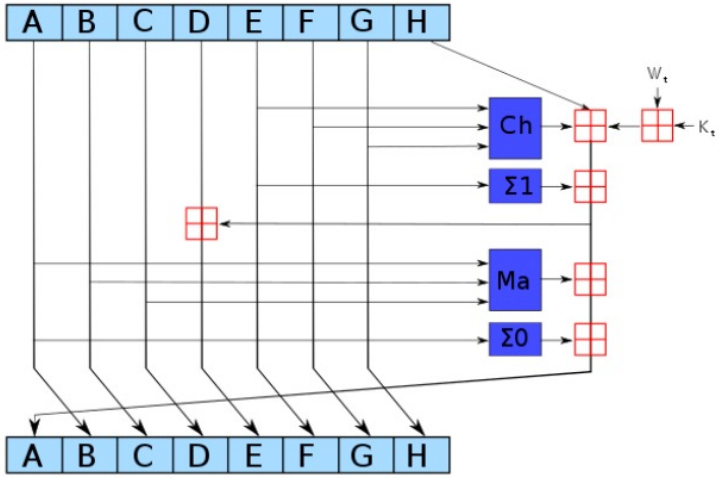


Blockchain PoWs Formalization (Informal) [GKP20]

- Two algorithms: Prove, Verify
- Basic properties:
  - Prove is expensive (moderately hard)
  - Verify is (much) easier
  - Completeness
- "Blockchain-friendly" properties:
  - Amortization resistance
  - High rate of success for unique (i.e., just one) honest prover
  - Run-time independence
  - …
- Challenging to achieve (in the standard model)

Blockchains from Non-Idealized Hash Functions [GKP20]





- Secure blockchain protocol assuming:
  - Hash function satisfies 3 simple properties: collision resistance, computational randomness extraction, *iterated hardness*
  - NIZKs (Non-Interactive Zero-Knowledge Proofs)
  - Adversary controls 1/3 of the computational power

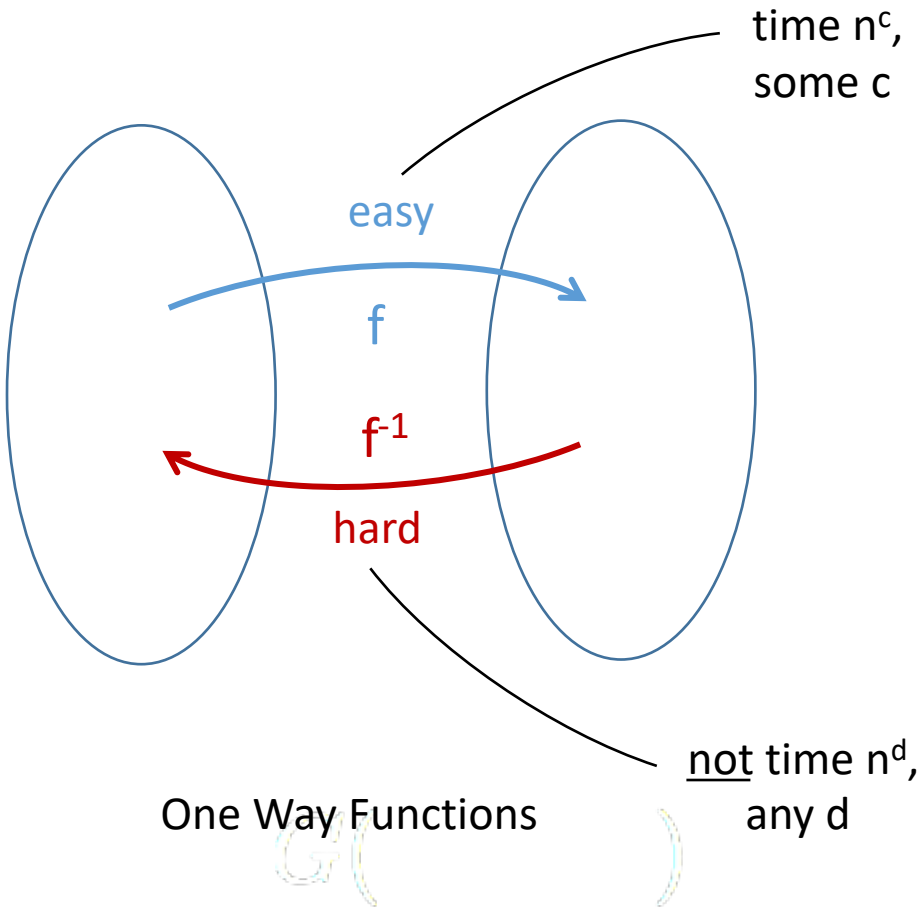
Topics

- PoW-based Blockchains
- *Resource-Restricted Cryptography*
- PoWs in the “Standard” Model
  - Based on hash-function properties
  - Based on fine-grained complexity

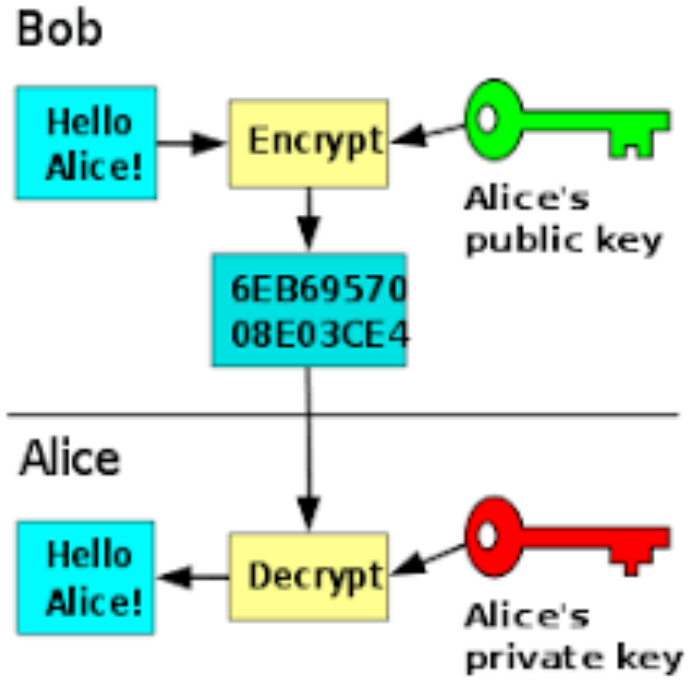
Cryptography

RECALL

Procedures that mitigate adversarial behavior



One Way Functions



Public Key Cryptography

Refined Adversarial Model

- "Classical" adversary: Runs in: $n, n^2, n^3, \ldots, n^{100}, \ldots$

arbitrary constant

- "Fine-grained" adversary: Runs in n, n^2, n^3
  - Cf. *fine-grained* complexity theory

fixed constant

- *"How to do cryptography when cryptography is not possible?"*

Fine-Grained Complexity

- **Observation:** For many natural problems, “brute force” (time T), is essentially state of the art ($\Omega(T^{1-\varepsilon}), \forall \varepsilon > 0$)

- **Conjecture:** For many natural problems, brute force is essentially optimal

- **Examples:** Orthogonal Vectors, All-Pairs Shortest Paths, 3Sum,…

PoWs/Blockchains from Fine-Grained Complexity

- [BRSV17/18]: Assuming Orthogonal Vectors take $n^{2-o(1)}$ in the worst case, then n^2-PoWs exist

- [BGKP23]: Assuming Orthogonal Vectors conjecture, **(permissionless) consensus** is achievable with public setup tolerating a constant fraction of corruptions

Summary

- PoWs (basic blockchain enabler): Powerful primitive
  - PoW-based protocols challenge long-established impossibility results
- The *Resource-Restricted Cryptography* framework
  - Consensus/MPC with public setup tolerating $t < n/2$
- PoW realizations in the “standard” model

References

- **[GKL15]** J. Garay, A. Kiayias, R. Ostrovsky, G. Panagiotakos and V. Zikas, “Resource-restricted Cryptography: Revisiting MPC Bounds in the Proof-of-Work Era.” *Eurocrypt 2020*. Full version at Cryptology ePrint Archive: Report 2019/1264, https://eprint.iacr.org/2020/1264
- **[GKP20]** J. Garay, A. Kiayias and G. Panagiotakos, “Blockchains From Non-Idealized Hash Functions.” *TCC 2020*. Full version at Cryptology ePrint Archive: Report 2019/315, https://eprint.iacr.org/2019/315
- **[GK20]** J. Garay and A. Kiayias, “A Consensus Taxonomy in the Blockchain Era.” *CT-RSA 2020*. Full version at Cryptology ePrint Archive: Report 2018/754, https://eprint.iacr.org/2018/754
- **[BGKP23]** M. Ball, J. Garay, A. Kiayias and G. Panagiotakos, “Permissionless Consensus from Proofs of Work in the Standard Model.” In preparation

Inaugural Texas A&M Blockchain Day

May 1, 2023

Location: PETR 118

Program and registration: tamublockchain.github.io


Thank You

Backup Slides

On the Necessity of an Honest Majority for Consensus (2)

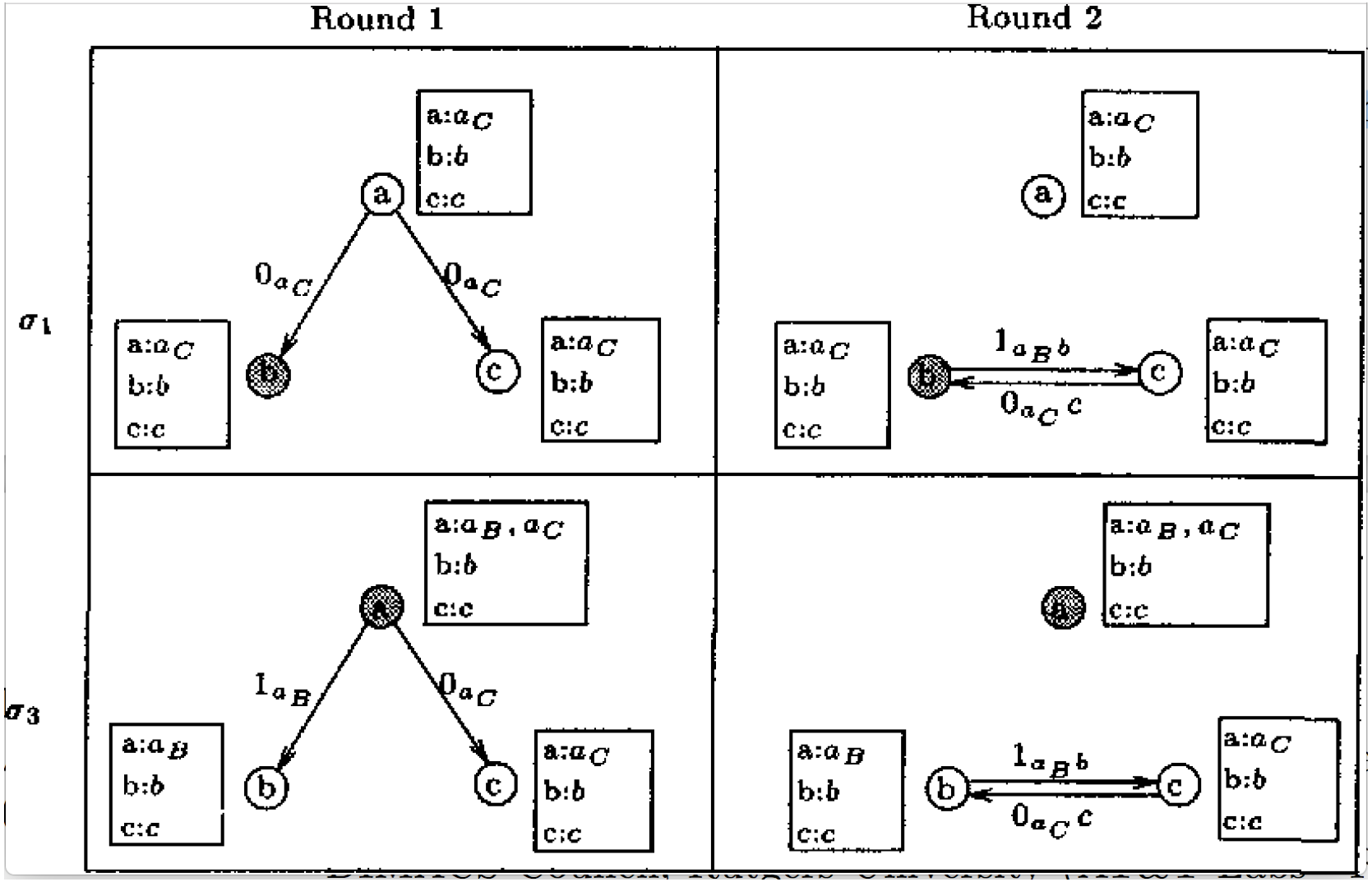
- **Scenario** [Fit03]: n parties equally divided with respect to their initial values $\in \{0,1\}$. Adv. corrupts $\emptyset$, P_0 and P_1 uniformly at random:

  1. With $1/3$ prob. adversary corrupts no one
  2. With $1/3$ prob. adversary corrupts parties with input 0
  3. With $1/3$ prob. adversary corrupts parties with input 1

  In any case, the corrupted parties follow the protocol

- Case 1 requires honest parties to converge to common output (**Agreement**)
- Case 2 & 3: Honest parties should output 1 (resp., 0) (**Validity**)
- But the three cases are indistinguishable in the view of the honest parties

On the Necessity of a PKI ("Private-State Setup") [Bor96] (2)



Nakamoto's Consensus Protocol [Nak08]

- "The proof-of-work chain is a solution to the Byzantine Generals' Problem…"

The Bitcoin developer Satoshi Nakamoto described the problem this way:

> A number of Byzantine Generals each have a computer and want to attack the King's wi-fi by brute forcing the password, which they've learned is a certain number of characters in length. Once they stimulate the network to generate a packet, they must crack the password within a limited time to break in and erase the logs, lest they be discovered. They only have enough CPU power to crack it fast enough if a majority of them attack at the same time.
> They don't particularly care when the attack will be, just that they agree. It has been decided that anyone who feels like it will announce an attack time, which we'll call the "plan", and whatever plan is heard first will be the official plan. The problem is that the network is not instantaneous, and if two generals announce different plans at close to the same time, some may hear one first and others hear the other first.

Nakamoto's Consensus Protocol [Nak08] (2)

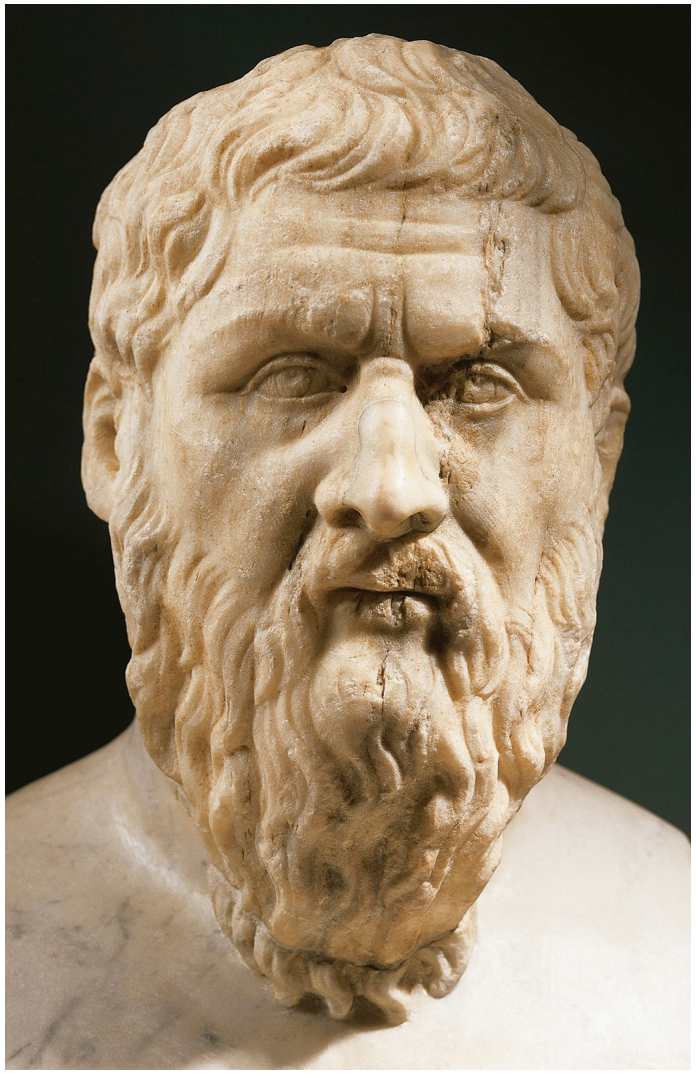
- Assume a public setup (e.g., "genesis" block) and an honest majority of computing power (i.e., $t < n/2$)
- The n parties start building a blockchain inserting their input (that would be transaction included in a block)
- If a party receives a longer blockchain, it switches to that one and switches its input
- When the blockchain is long enough the party outputs the (unique) value that it contains

Nakamoto's Consensus Protocol [Nak08] (3)

- Assume a public setup (e.g., "genesis" block) and an honest majority of computing power (i.e., $t < n/2$)
- The n parties start building a blockchain inserting their input (that would be transaction included in a block)
- If a party receives a longer blockchain, it switches to that one and switches its input
- When the blockchain is long enough the party outputs the (unique) value that it contains

- **Issue:** If adv. finds a solution first, then honest parties will extend adv.'s solution and switch to adv.'s input → protocol doesn't guarantee **Validity** (with non-negligible probability)

Plato’s Theory of Forms

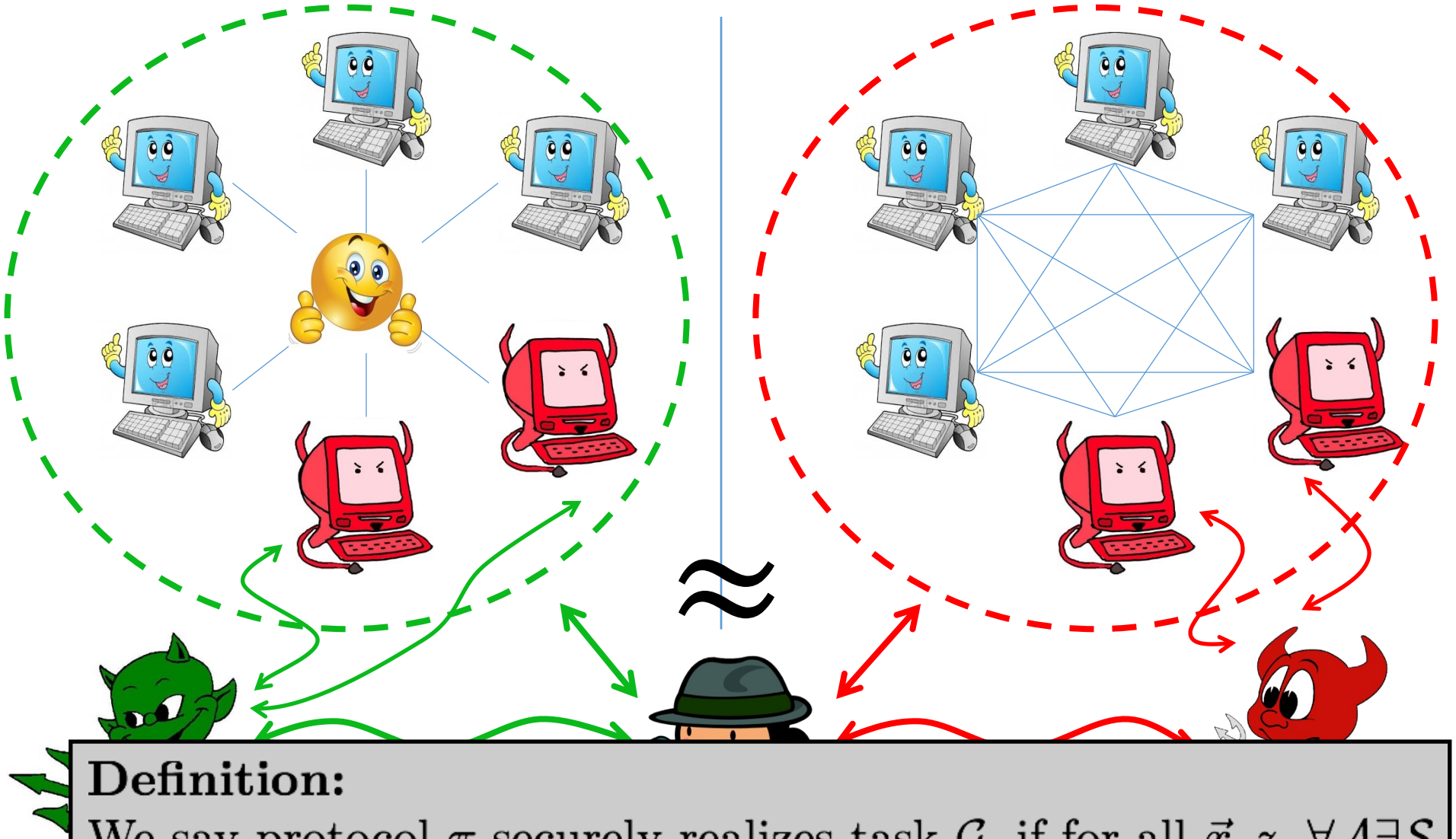


According to Plato's theory of Forms, every horse that we encounter in the world around us is a lesser version of an “ideal”, or perfect, horse that exists in a world of Forms or Ideas—a realm that humans can only access through their ability to reason.

The world of ideas

The world of the senses

Simulation-based Security: Real vs Ideal [GMW87, C01,…]



Definition:
We say protocol π securely realizes task $\mathcal{G}$, if for all $\vec{x}, z$, $\forall \mathcal{A} \exists \mathcal{S}$ such that $\{\mathsf{REAL}_{\pi,\mathcal{A}}\}_{\vec{x},z} \approx \{\mathsf{IDEAL}_{\mathcal{G},\mathcal{S}}\}_{\vec{x},z}$

RECALL

Resource-Restricted Cryptography

Long history:

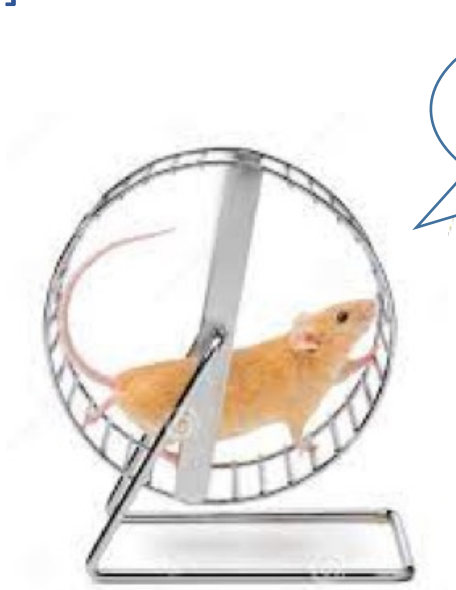
- Moderately hard functions, key exchange, spam mitigation, time-released crypto, fair computation, PoWs [Mer76,DN92,RSW96,BN00,GMPY06,AD15,BGJPVW16, BRSV17/18,…]
- Many different *resources* considered: (sequential) computational power, space, stake, …

Abstract resource: Network access

Proofs of Work (aka "Crypto Puzzles")

RECALL

- "Moderately hard functions" [DN92,RSW96,Back97,JJ99,BN00,GMPY06,BGJPVW16, BRSV17/18…]



I ran for a thousand steps!

Yeah, sure…

Prover

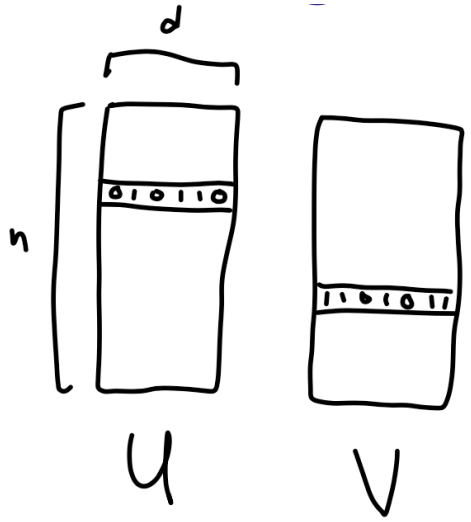
Verifier

Image credit: Marshall Ball

Blockchain PoWs Formalization (Informal) [GKP17/20]

- Two algorithms: Prove, Verify
- Basic properties:
  - Prove is expensive (moderately hard)
  - Verify is (much) easier
  - Completeness
- "Blockchain-friendly" properties:
  - Amortization resistance
  - High rate of success for unique (i.e., just one) honest prover
  - Run-time independence
  - …
- Challenging to achieve (in the standard model)

Orthogonal Vectors



is there $u \in U, v \in V$:

$\langle u, v \rangle = 0$?

(u, v represent disjoint subsets of $[d]$?)

SETH

Best Worst case Algorithm

$O(n^{2-\Theta(1/\log(d/\log n))})$

[AbboudWilliamsYu'15, ChanWilliams'16]

Orthogonal Vectors Conjecture

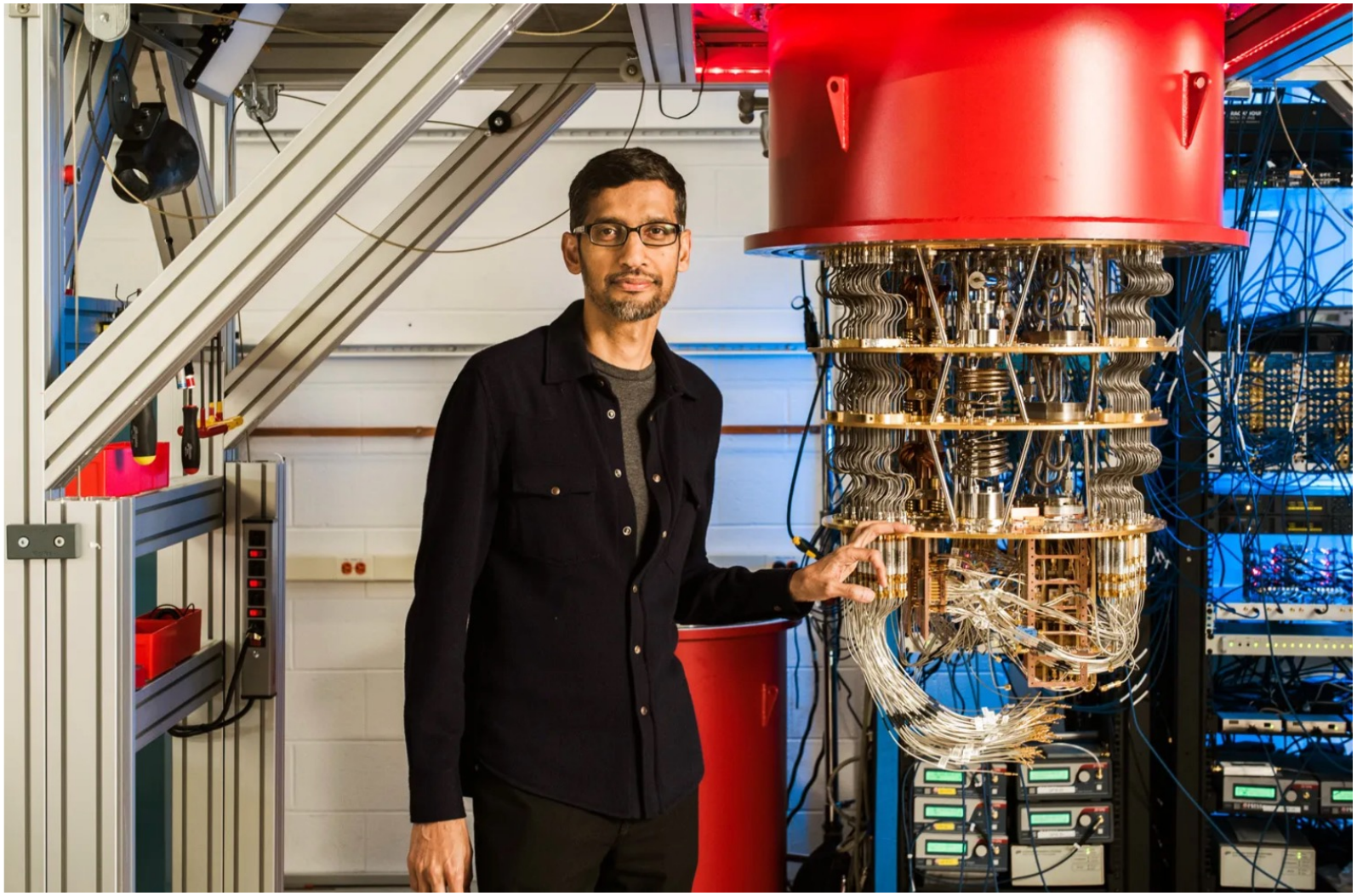
for $d = \omega(\log n)$, $\forall \varepsilon > 0$,

OV cannot be solved in time

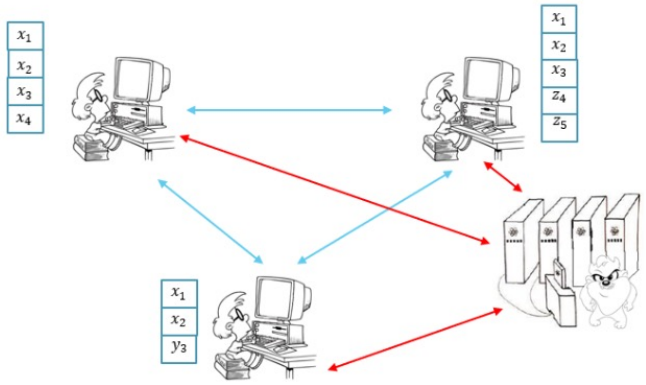
$O(n^{2-\varepsilon})$

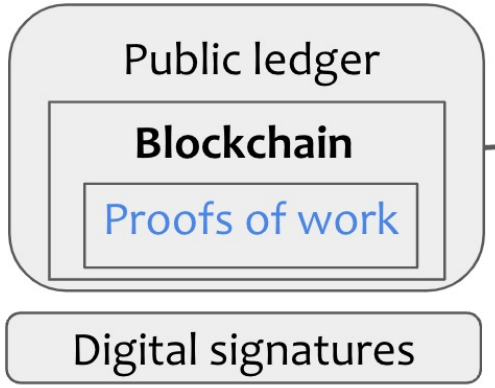
[Williams'05]

“Quantum is Coming”



Post-Quantum Security



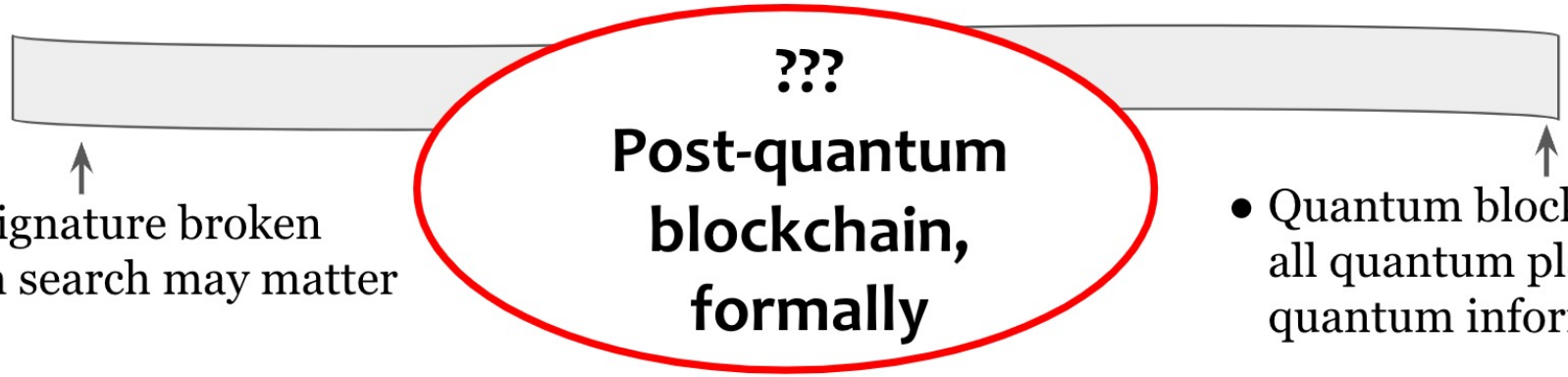


[GKL15]: (1st) Classical formal study

- Modular model of blockchain; define & prove security properties.
  - Common prefix, Chain quality, Chain growth, …
- Consensus/ledger from blockchain.
  - Consistency (Persistence) & Liveness

Concretely_Efficient_MPC_062822.pptx

The impact of quantum computing



- ECDSA signature broken
- Quantum search may matter

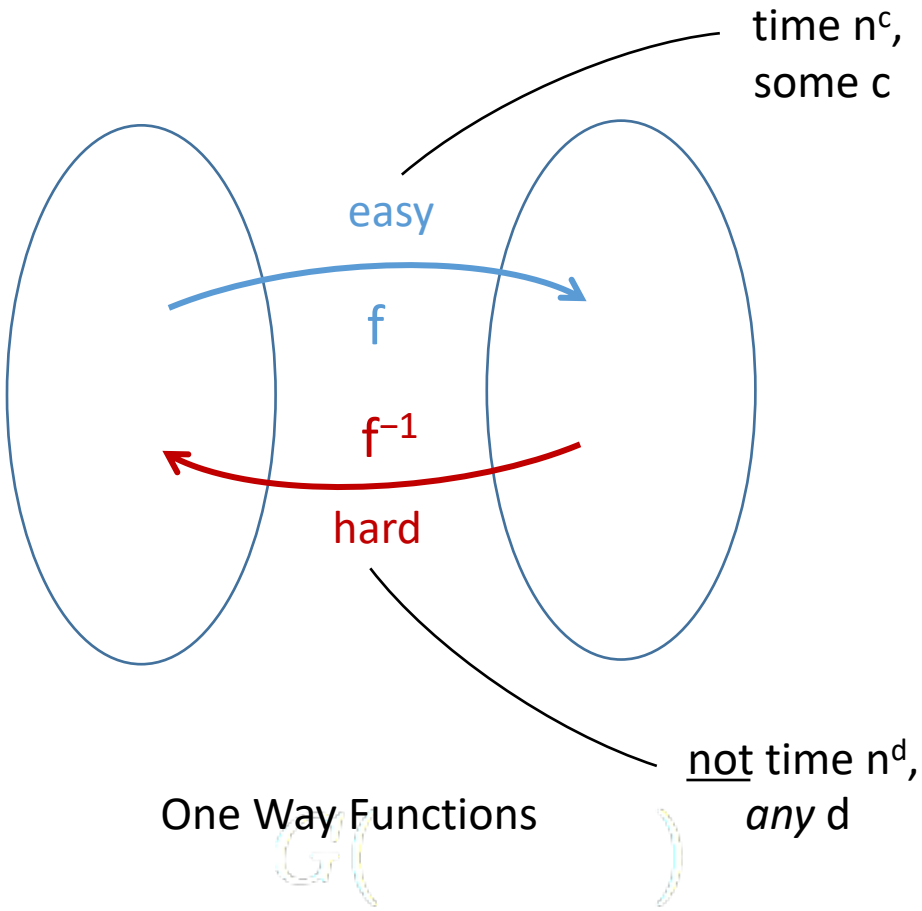
- Quantum blockchain/Bitcoin: all quantum players process quantum information.

References (2)

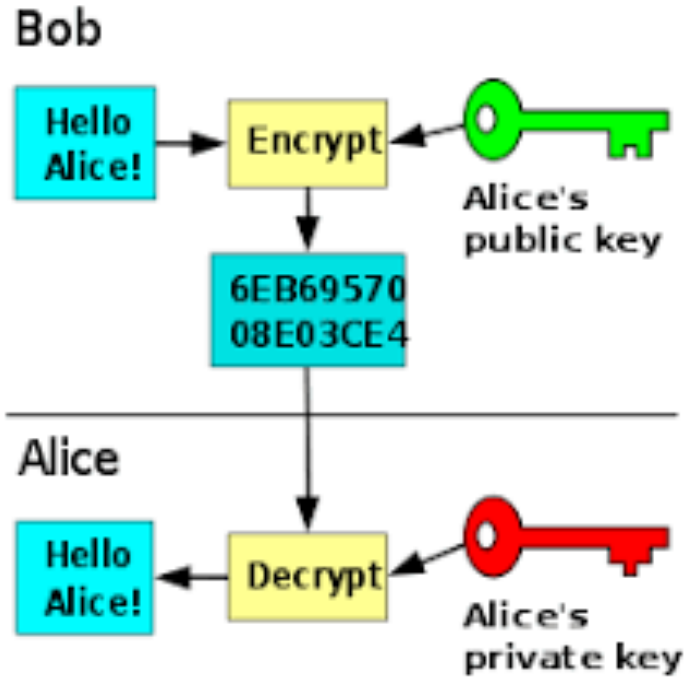
- A. Cojocaru, J. Garay, A. Kiayias, F. Song and P. Wallden, “Post-Quantum Blockchain Proofs of Work.” arXiv.org, https://arxiv.org/abs/2012.15254

Cryptography

Procedures that mitigate adversarial behavior



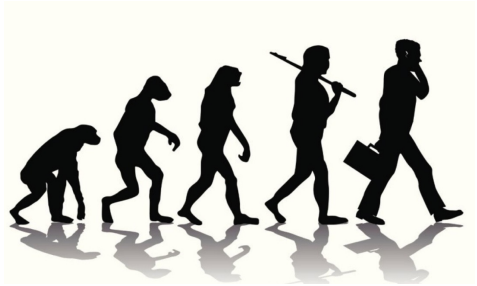
One Way Functions

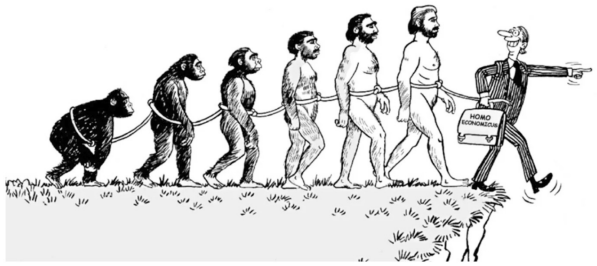


Public Key Cryptography

Why Would People Participate?

- *Utility*: “That property in any object, whereby it tends to produce benefit, advantage, pleasure, good, or happiness.” [Jeremy Bentham, 1870]
- *Homo Economicus*:
  - Constantly acts rationally and optimally
  - Aims to maximize its financial gain





- People would participate in Bitcoin if they are financially rewarded
- Users would pay others to run the system
- Free market (everywhere)

The “Permissionless” Model [Nak08]

- Not a traditional distributed system
  - Nodes known *a priori* and authenticated

- The “permissionless” model