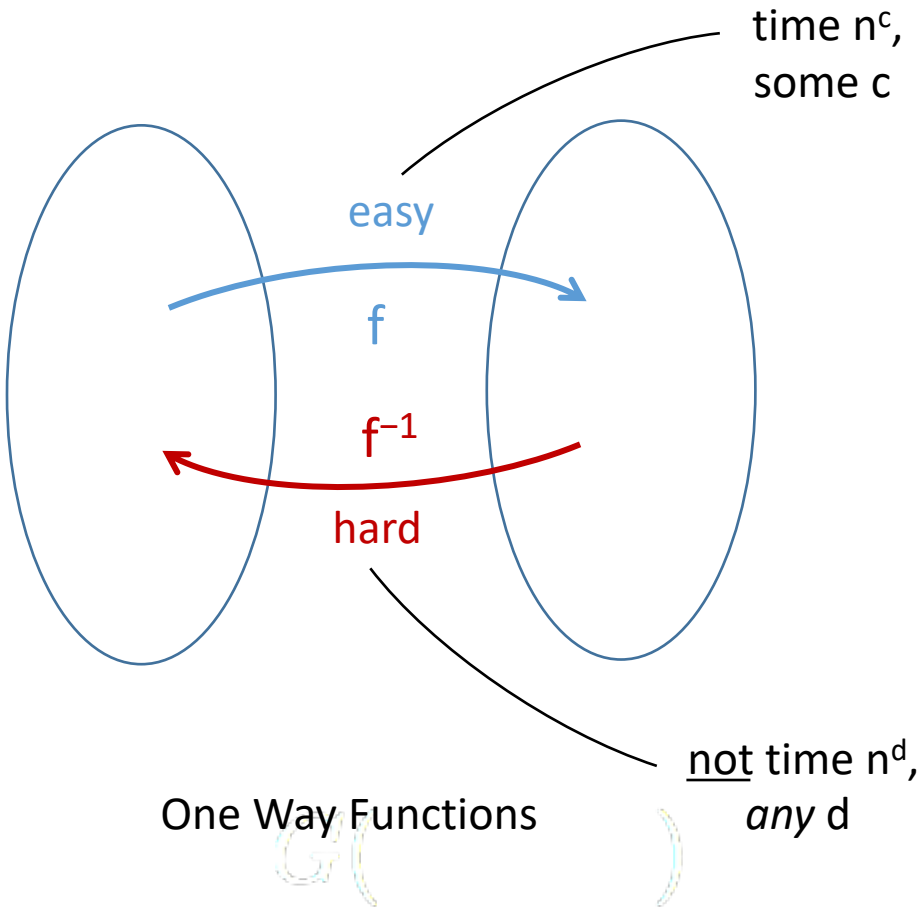
Cryptography in the ~~Blockchain~~ Proof-of-Work Era

Juan A. Garay
Texas A&M University
garay@tamu.edu
https://jagaray.com

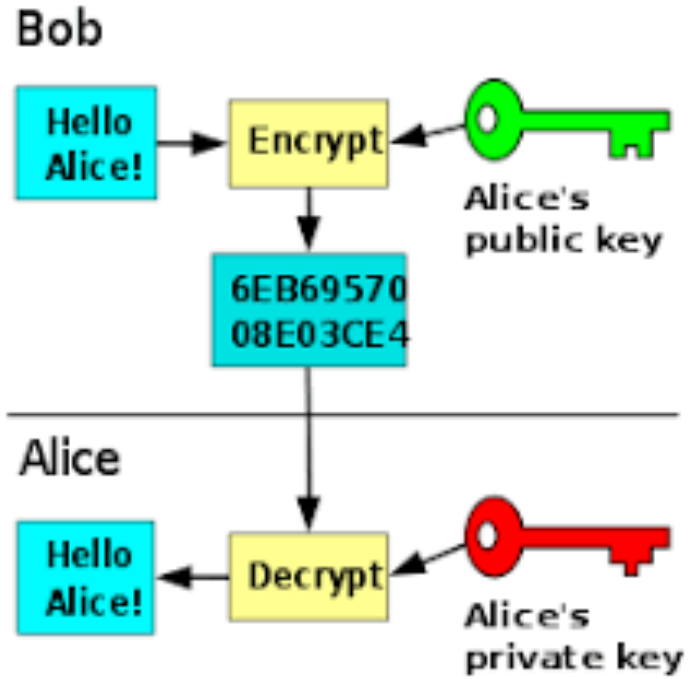
Joint work with Marshall Ball (NYU), Nikos Leonardos (U. of Athens), Aggelos Kiayias (U. of Edinburgh), Rafail Ostrovsky (UCLA), Giorgos Panagiotakos (IOHK) and Vassilis Zikas (Purdue University)

Cryptography

Procedures that mitigate adversarial behavior



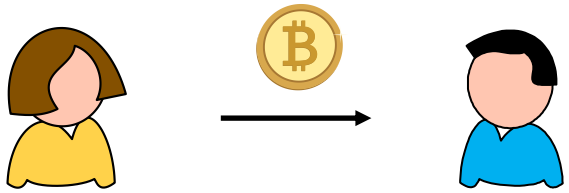
One Way Functions



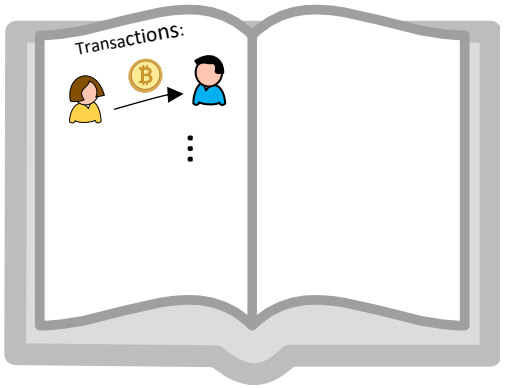
Public Key Cryptography

Blockchains

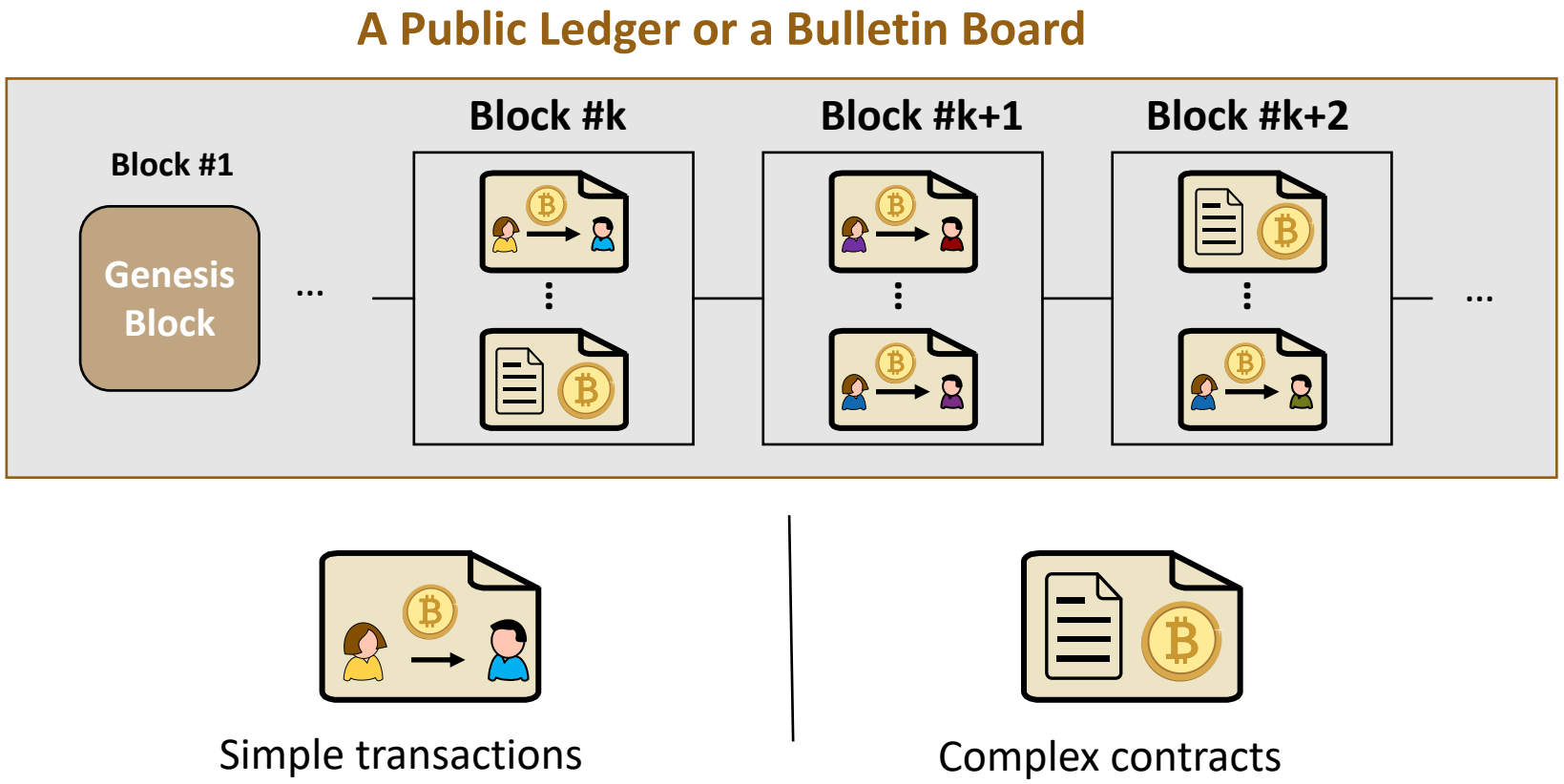
1) The problem: digital money transfer



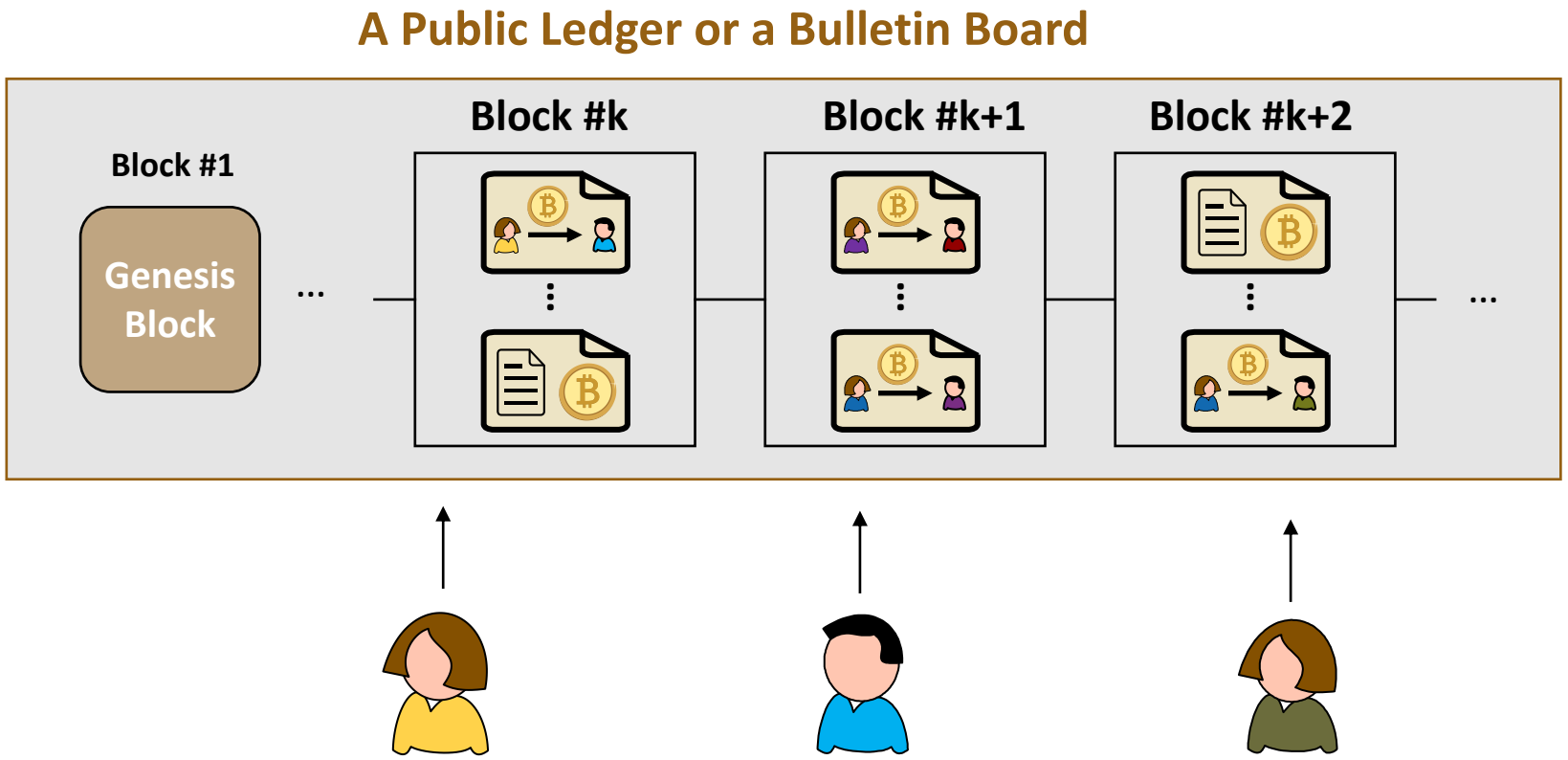
2) The accounting: implement a ledger!



Blockchains

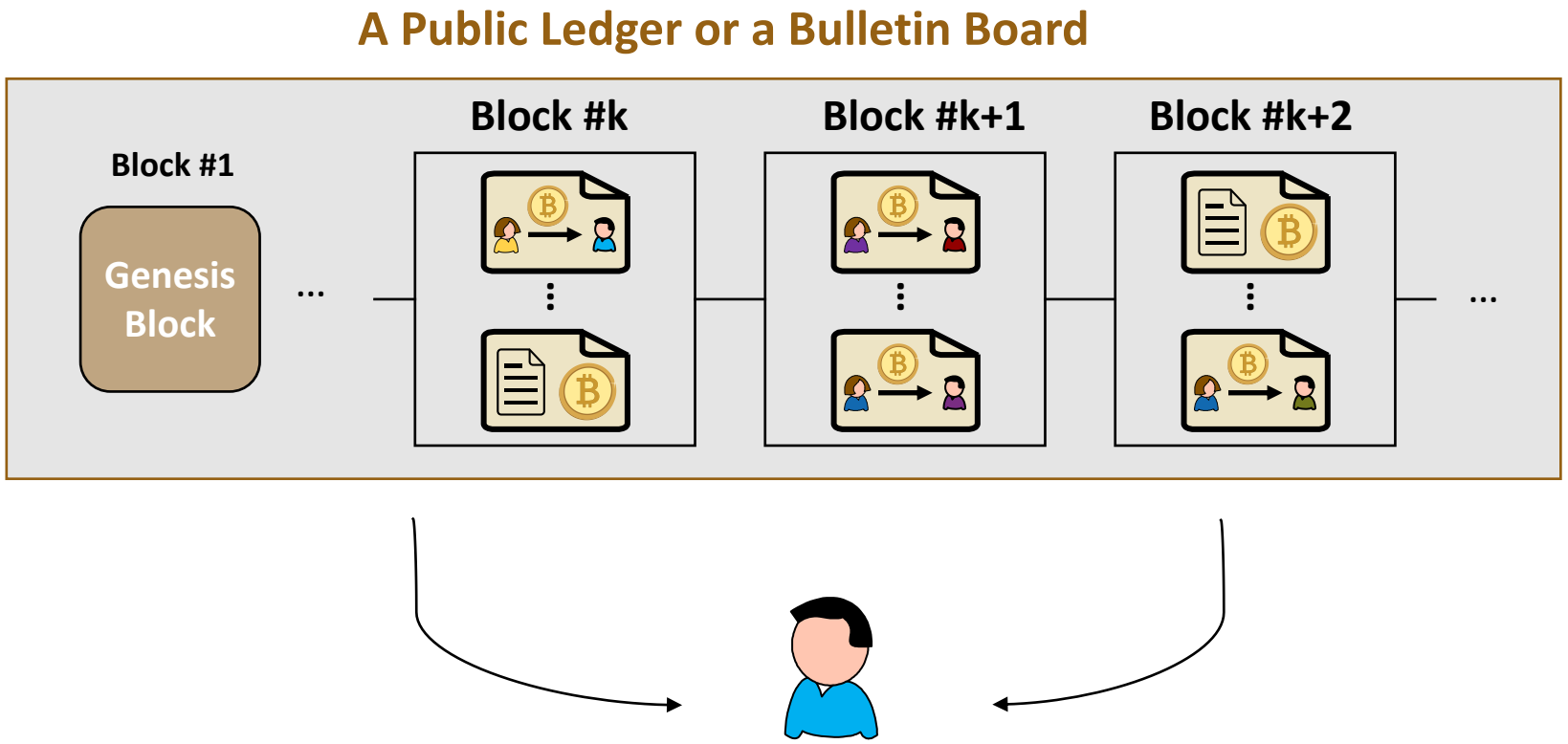


Blockchains



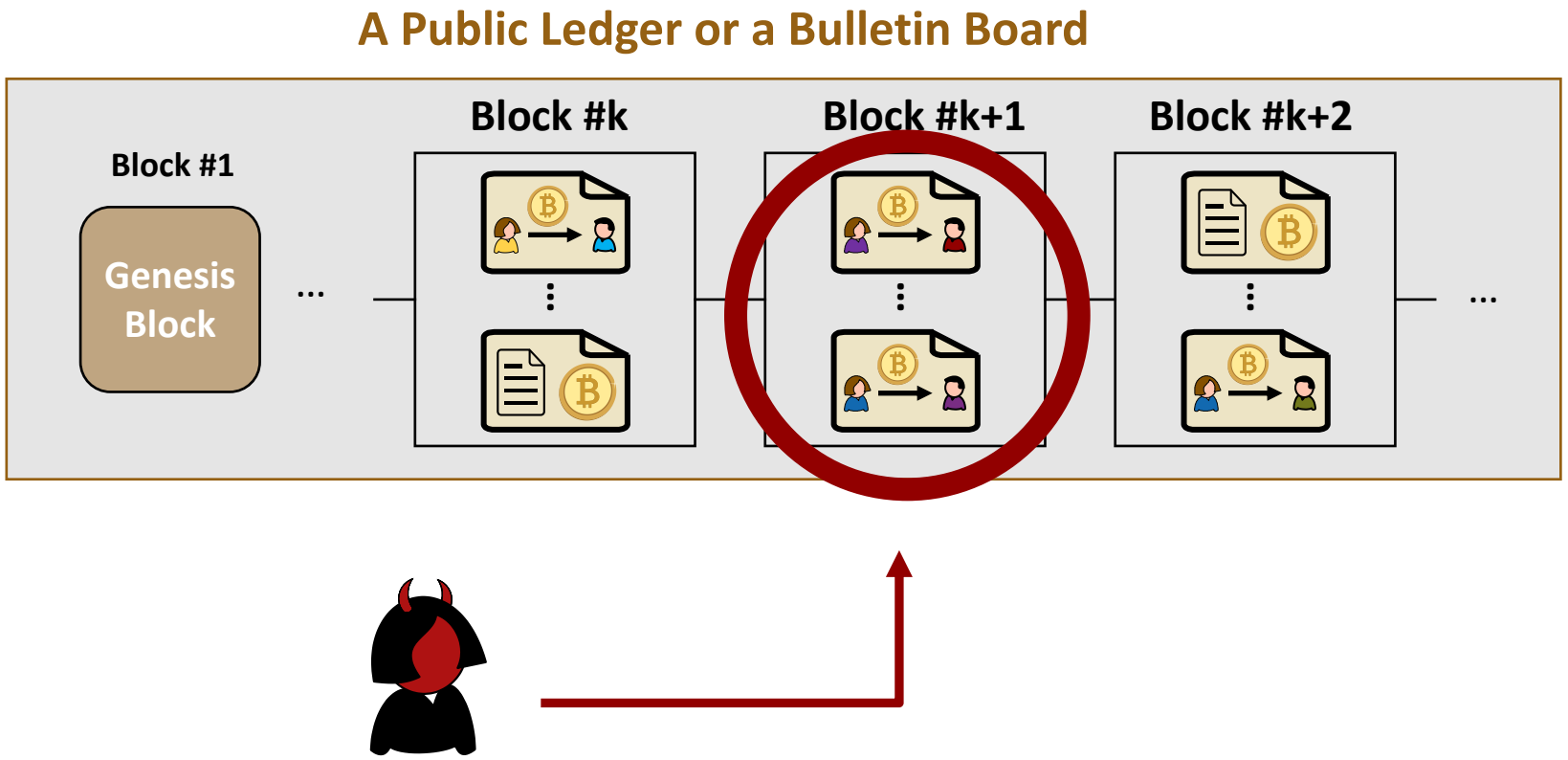
1) Content is provided by any user with sufficient funds

Blockchains



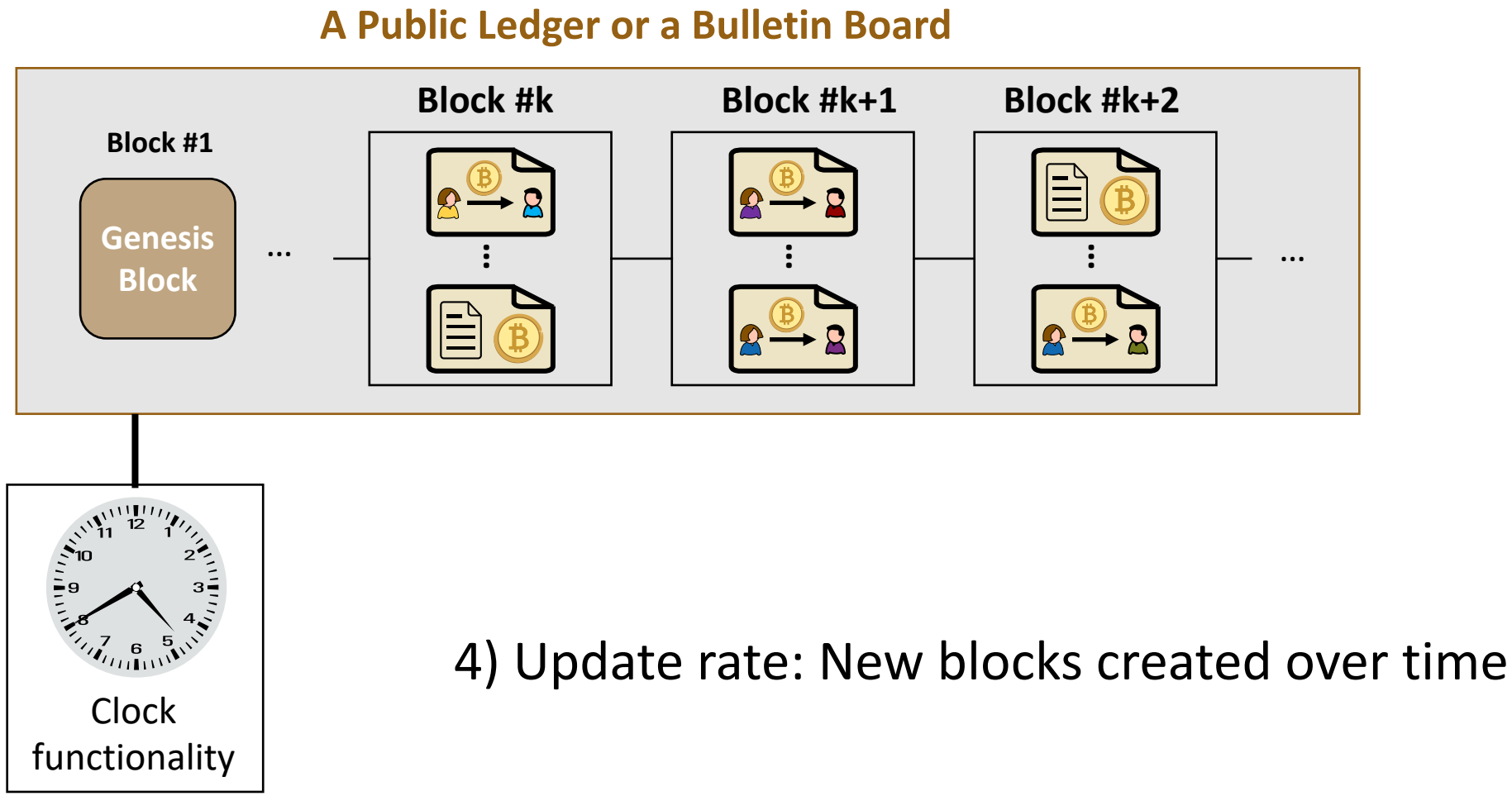
2) Anyone can read the current content of the ledger

Blockchains



3) No modifications of blockchain blocks possible

Blockchains



4) Update rate: New blocks created over time

Distributed Ledger

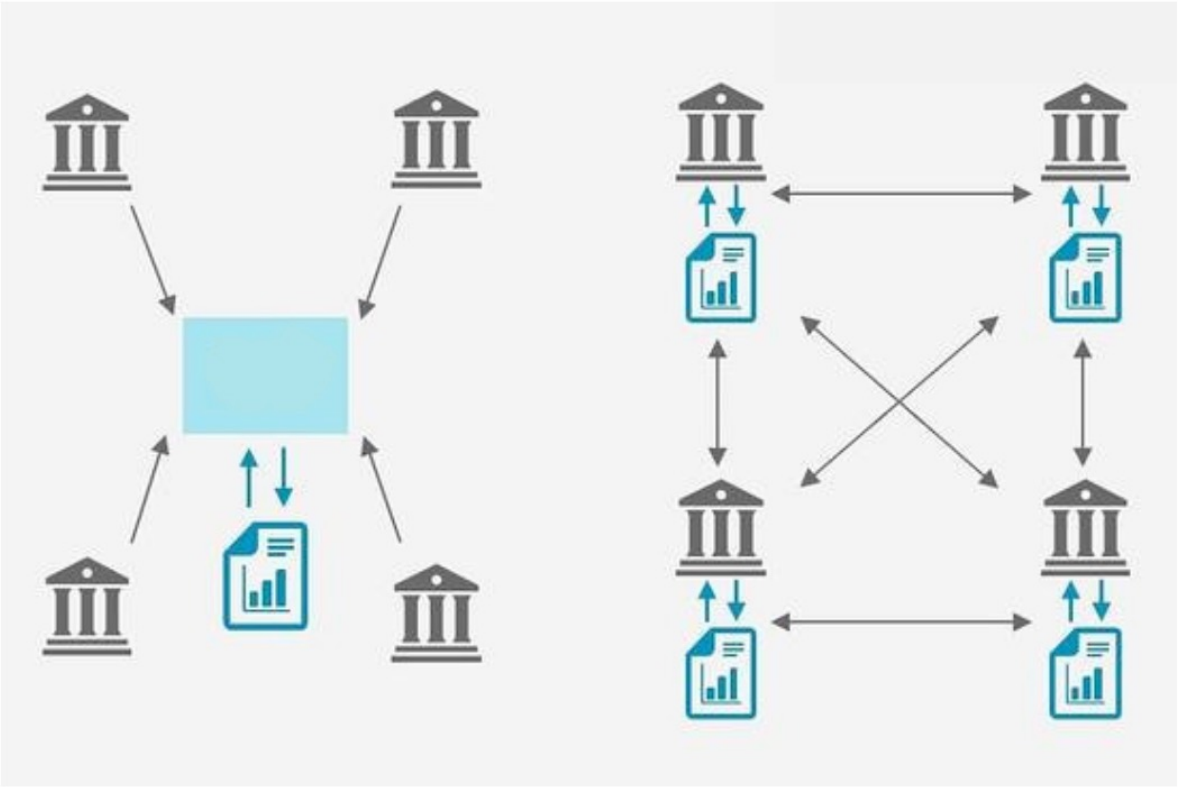


Figure: http://blogs.wsj.com/cio/2016/02/02/cio-explainer-what-is-blockchain/

- 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)

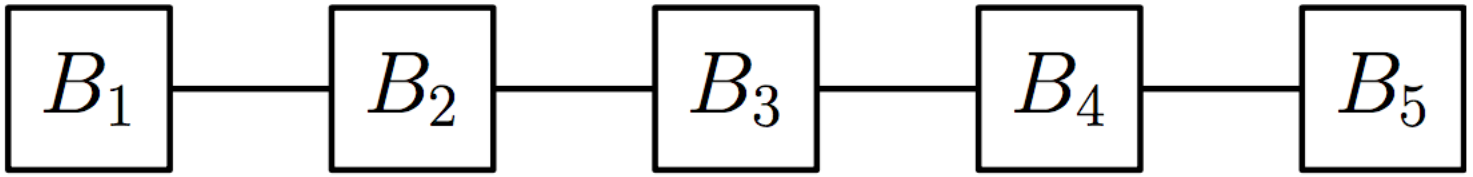
- Parties (“miners”) have to do work in order to install a transaction

PoW-based Blockchain Protocols (Bitcoin)

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

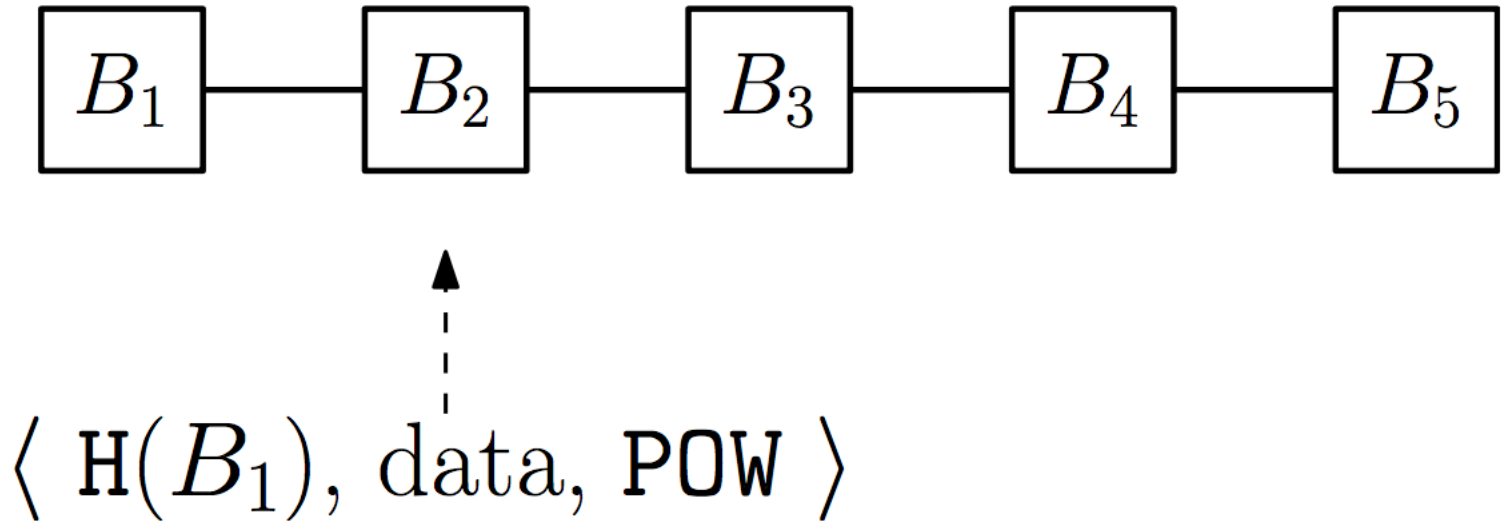
PoW-based Blockchain Protocols (Bitcoin)

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



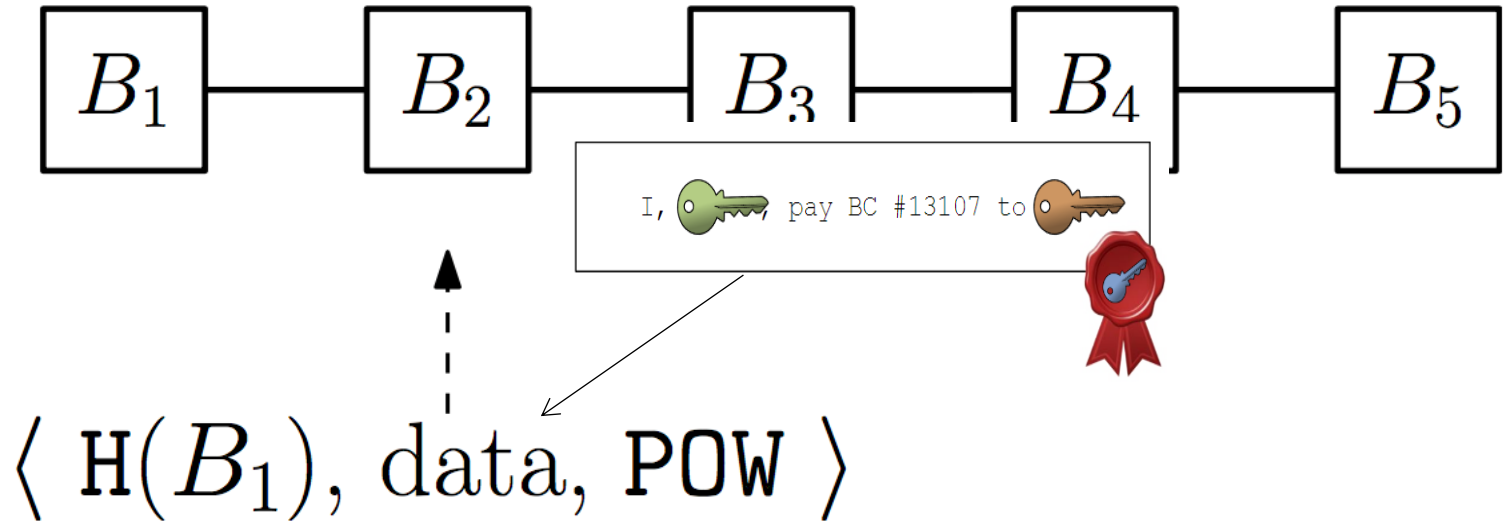
PoW-based Blockchain Protocols (Bitcoin)

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



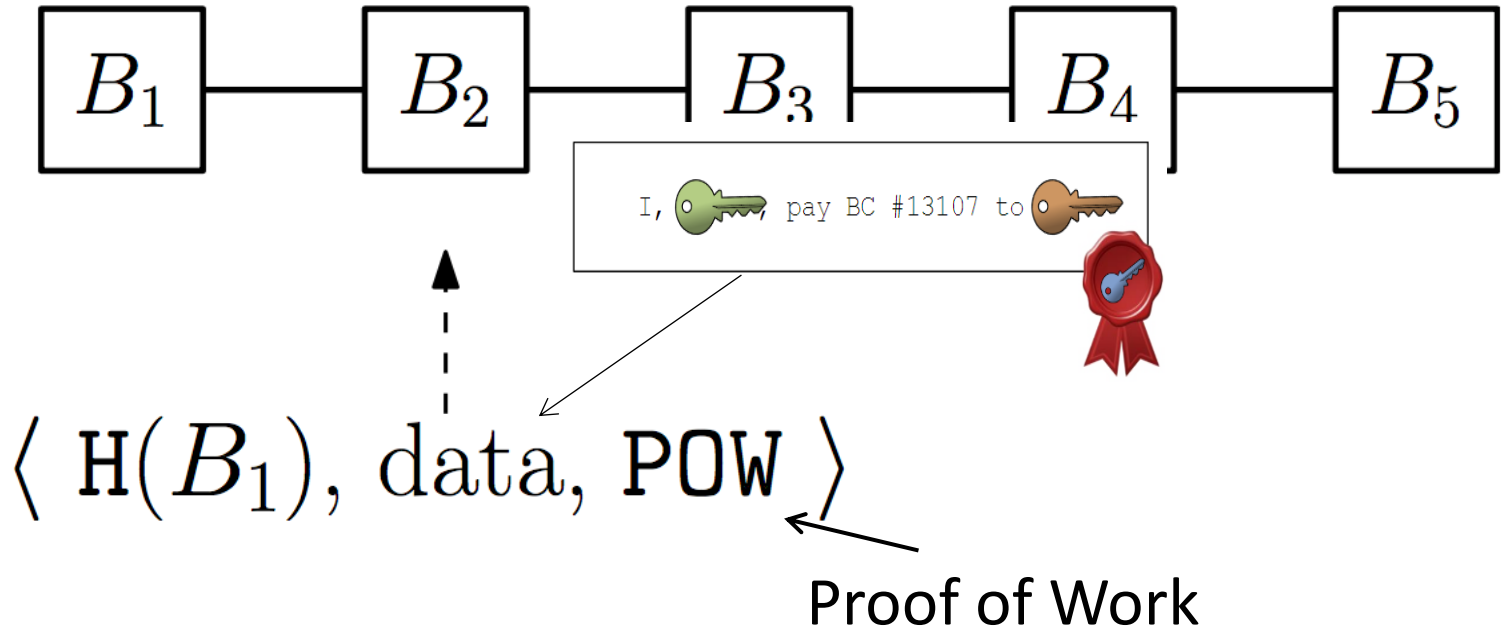
PoW-based Blockchain Protocols (Bitcoin)

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

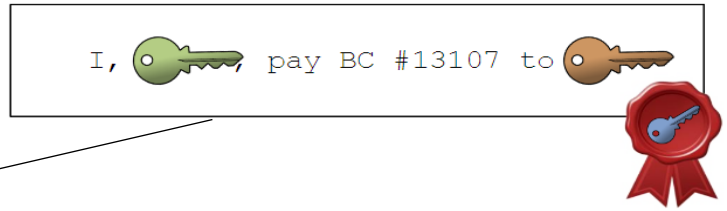


PoW-based Blockchain Protocols (Bitcoin)

- 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 10+ 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…]

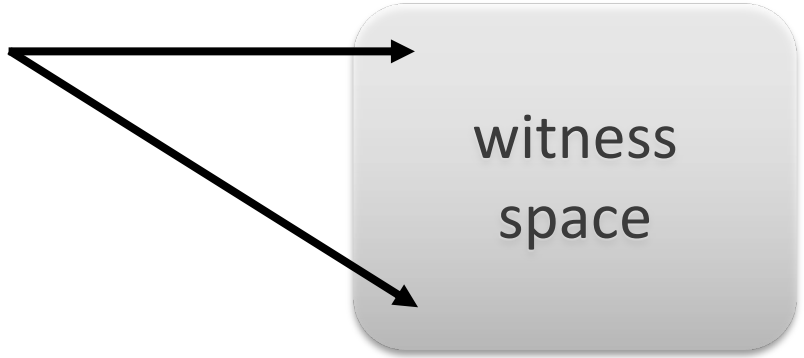
Prover **Verifier**

Proofs of Work (aka "Crypto Puzzles")

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

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

$Q(x,\cdot)$



Challenge determines work level d

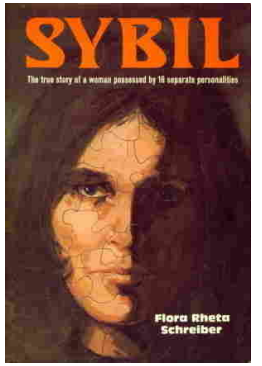
Search for witness takes long time (e.g., big-poly(d)) (a complexity lower bound needs to be assumed/proved)

Verification is easy!

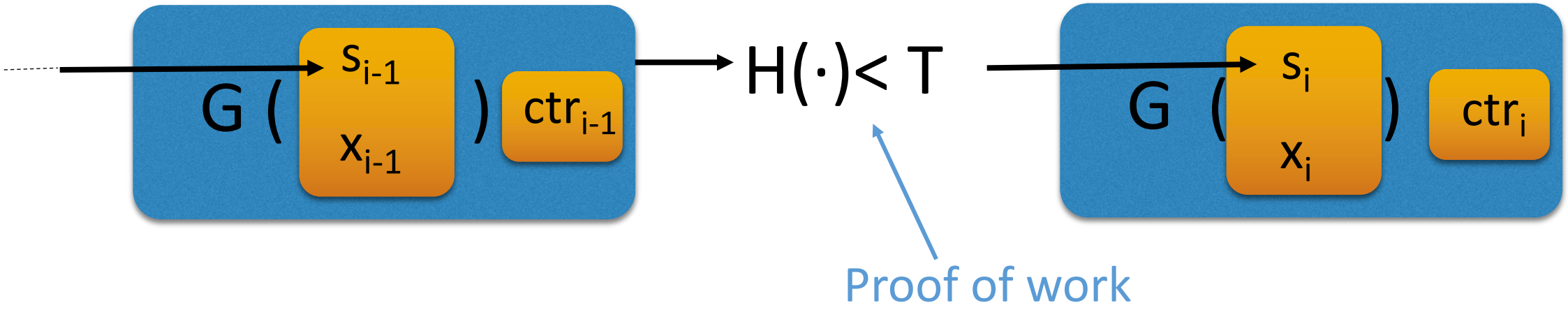
Successful only with some (small) probability!

cf. "fine-grained complexity"

Proofs of Work (aka “Crypto Puzzles”)



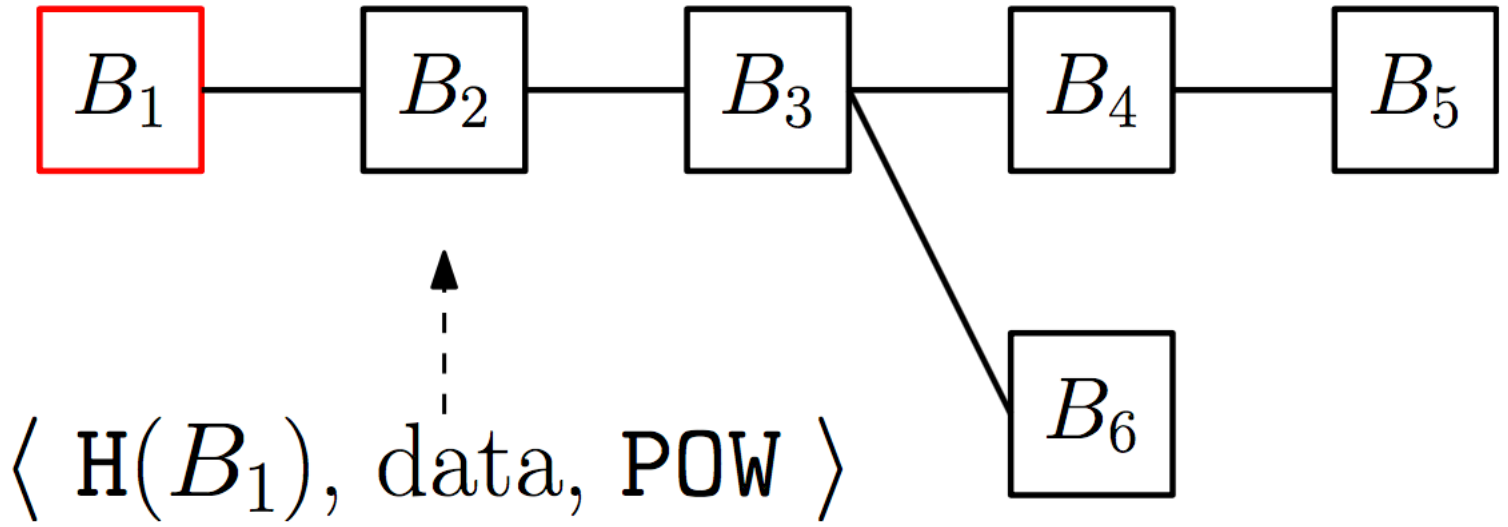
- Spam mitigation, Sybil attacks, denial of service protection, …
- Most impactful application: Design of blockchain protocols such as Bitcoin



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

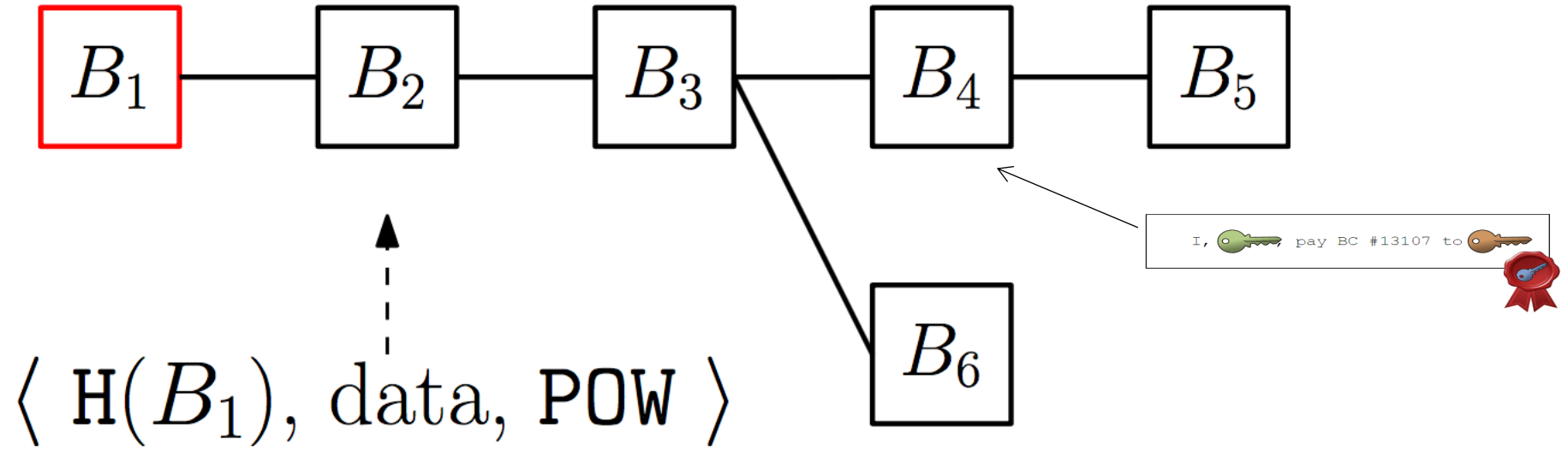
PoW-based Blockchain Protocols (Bitcoin) (3)

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



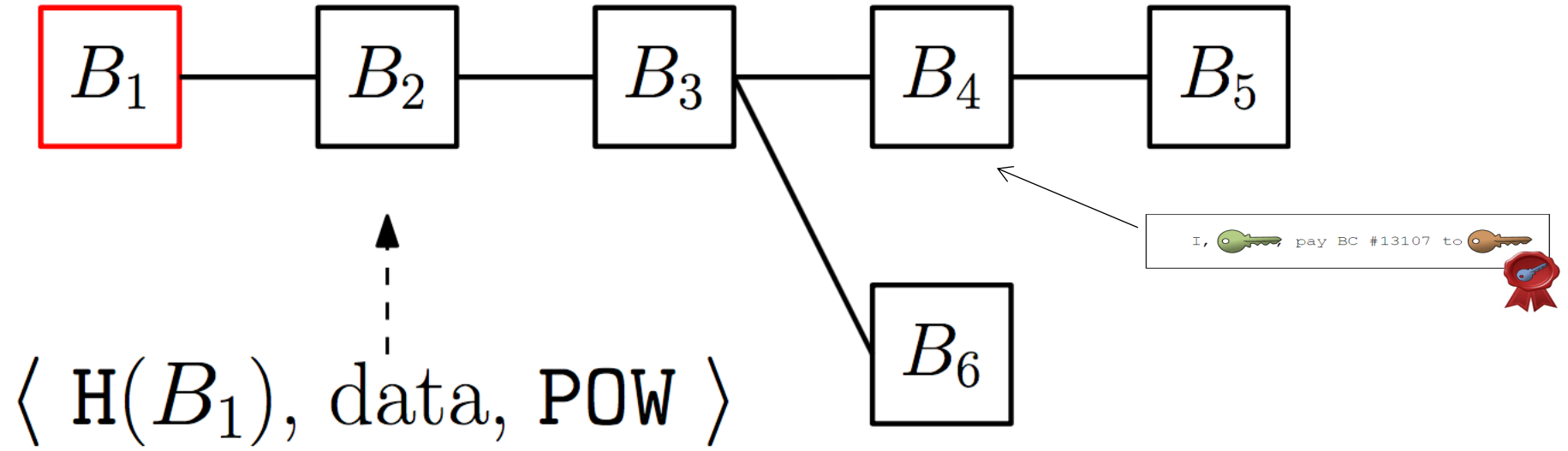
PoW-based Blockchain Protocols (Bitcoin) (3)

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



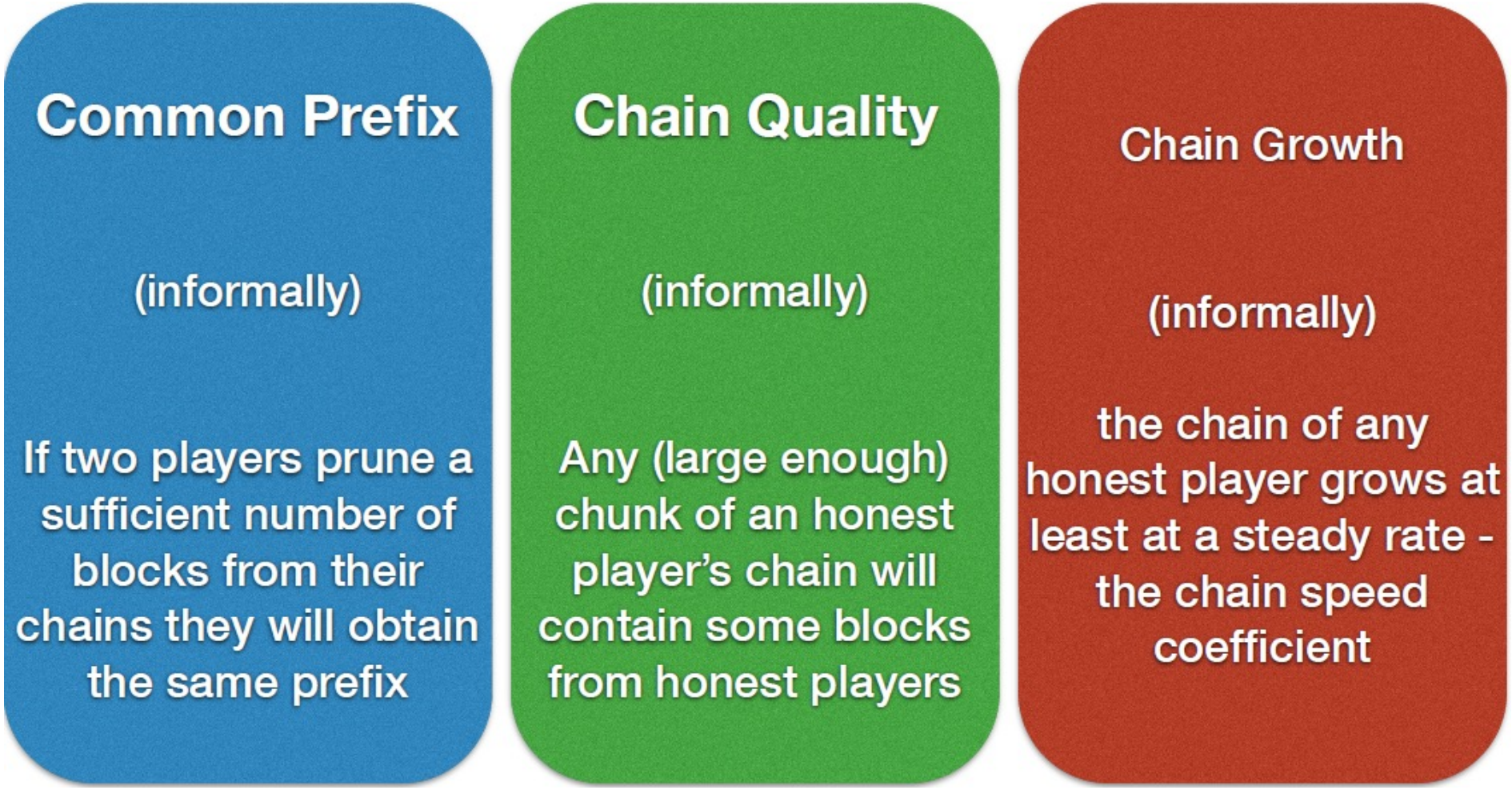
PoW-based Blockchain Protocols (Bitcoin) (4)

- 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]

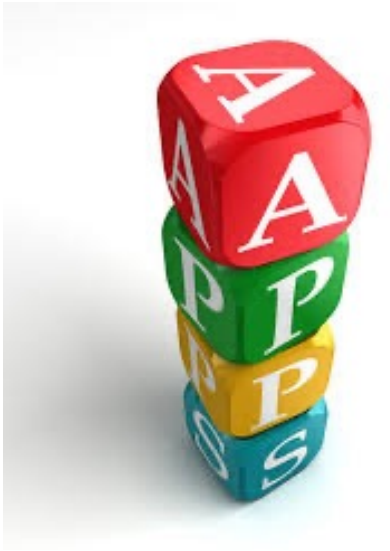


From Blockchain Properties to Applications [GKL15]

- 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”

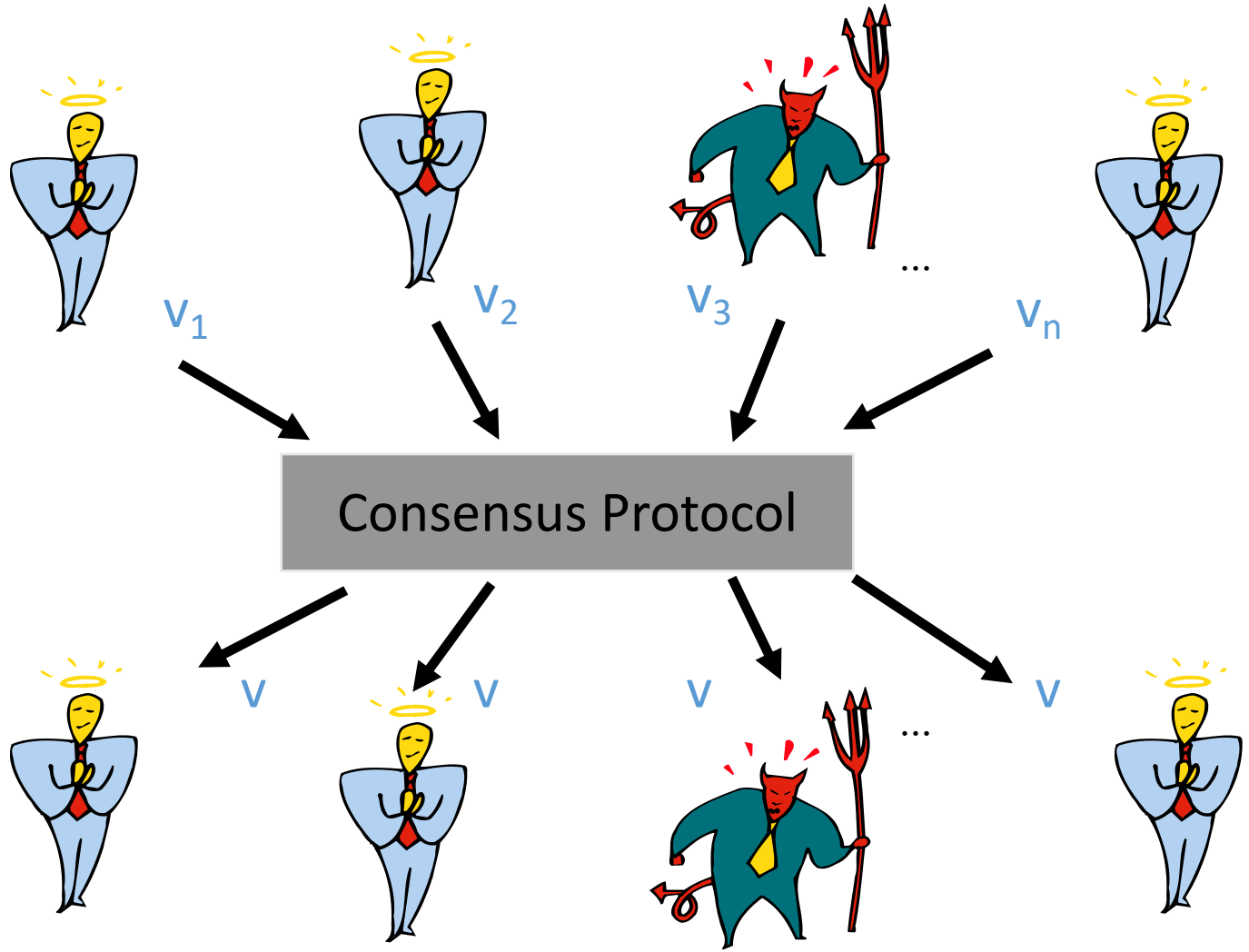


Consensus (*Byzantine Agreement*) [PSL80, LSP82]

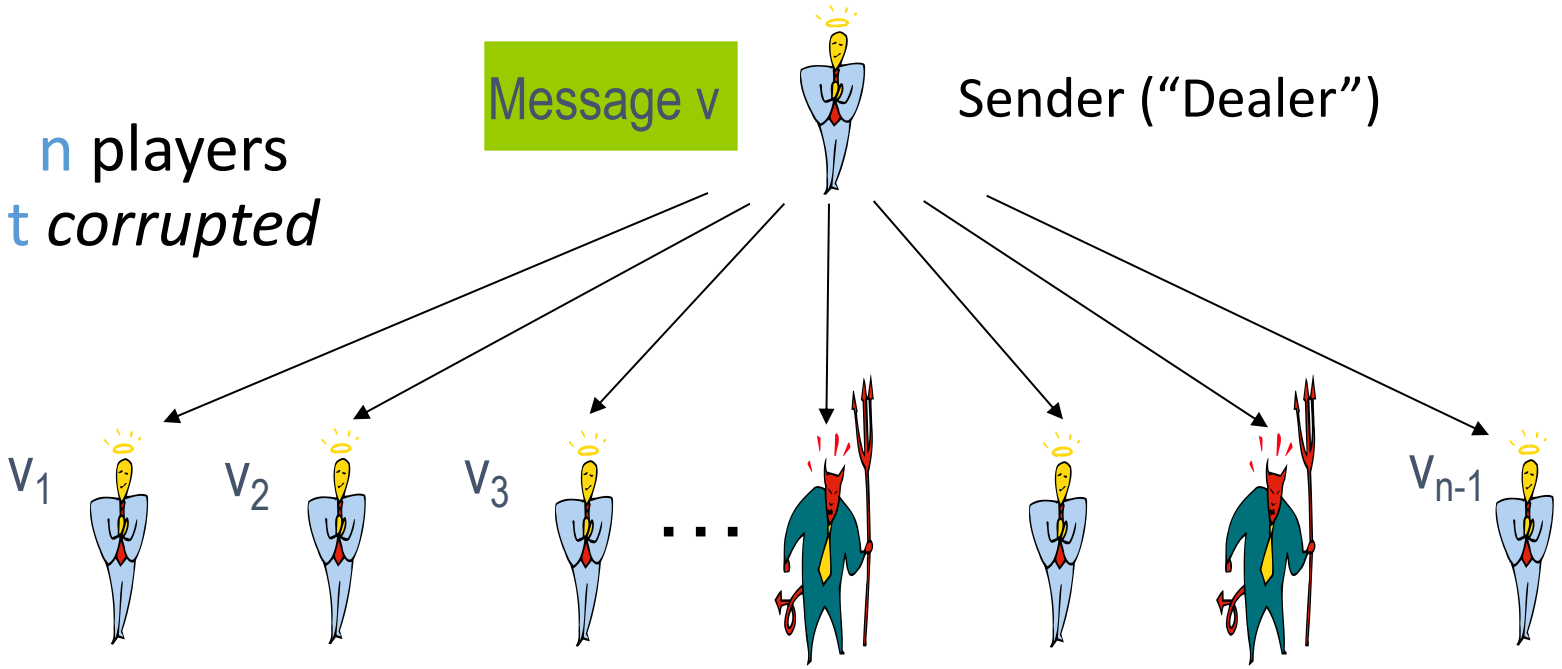
n parties
t *corrupted*

Adversary

$v_i \in V = \{0,1\}$



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

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

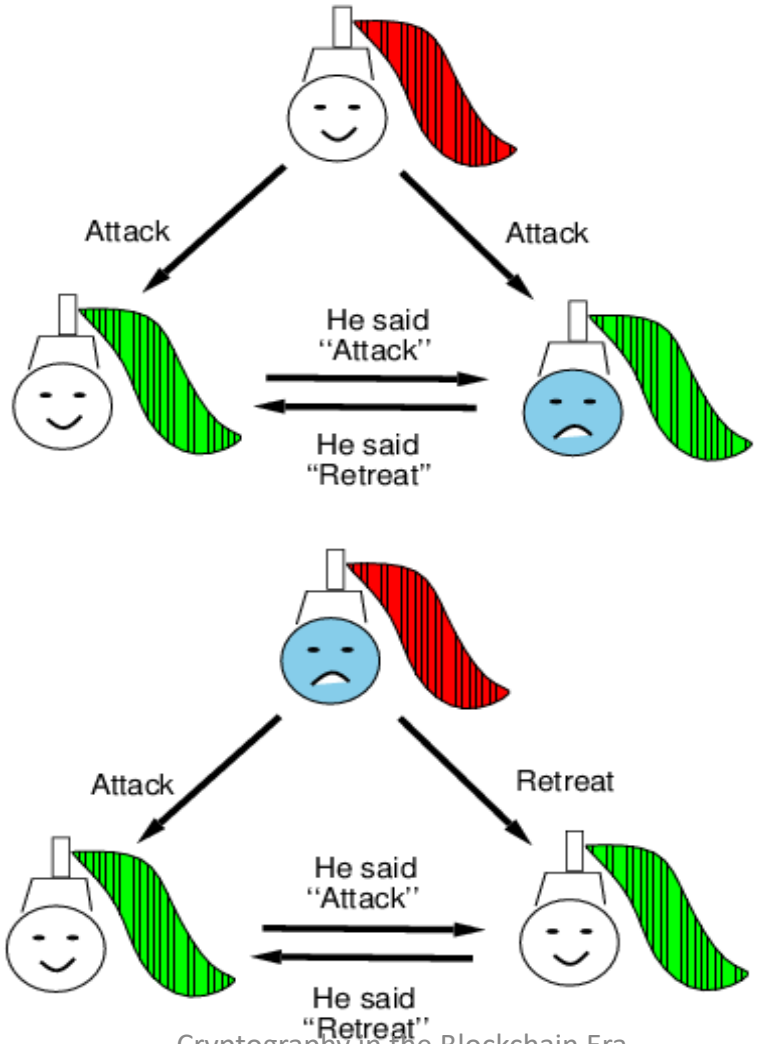
Secure Multiparty Computation (MPC) [Yao86, GMW87,…]

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

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 next slide?)

- Yes, but can't do better than

$$t < n/2$$

regardless of the resources available to the parties

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

Public vs Private State Setup

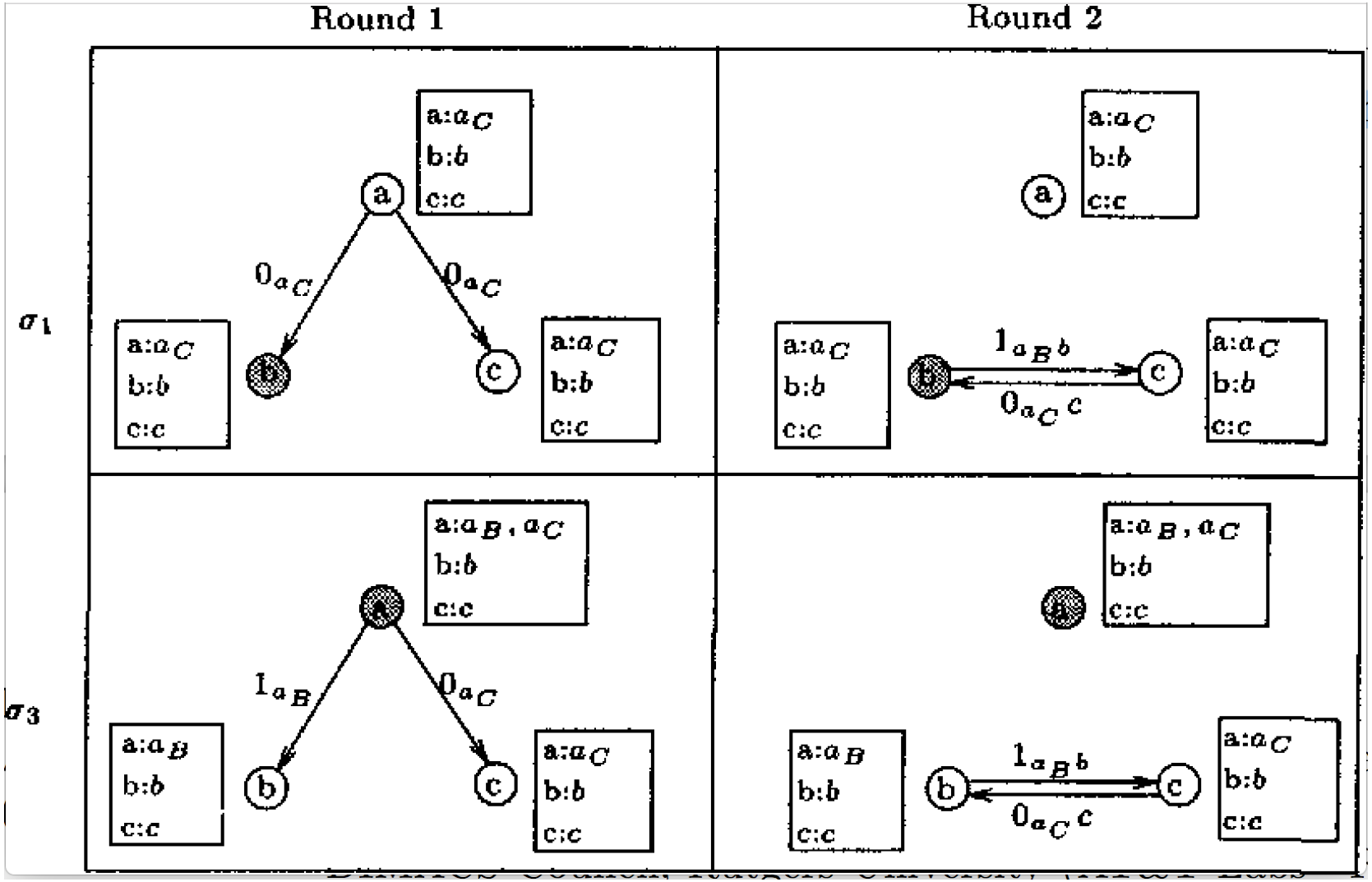
We know that:

- 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

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



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

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)

First PoW-based Consensus Protocol [GKL15]

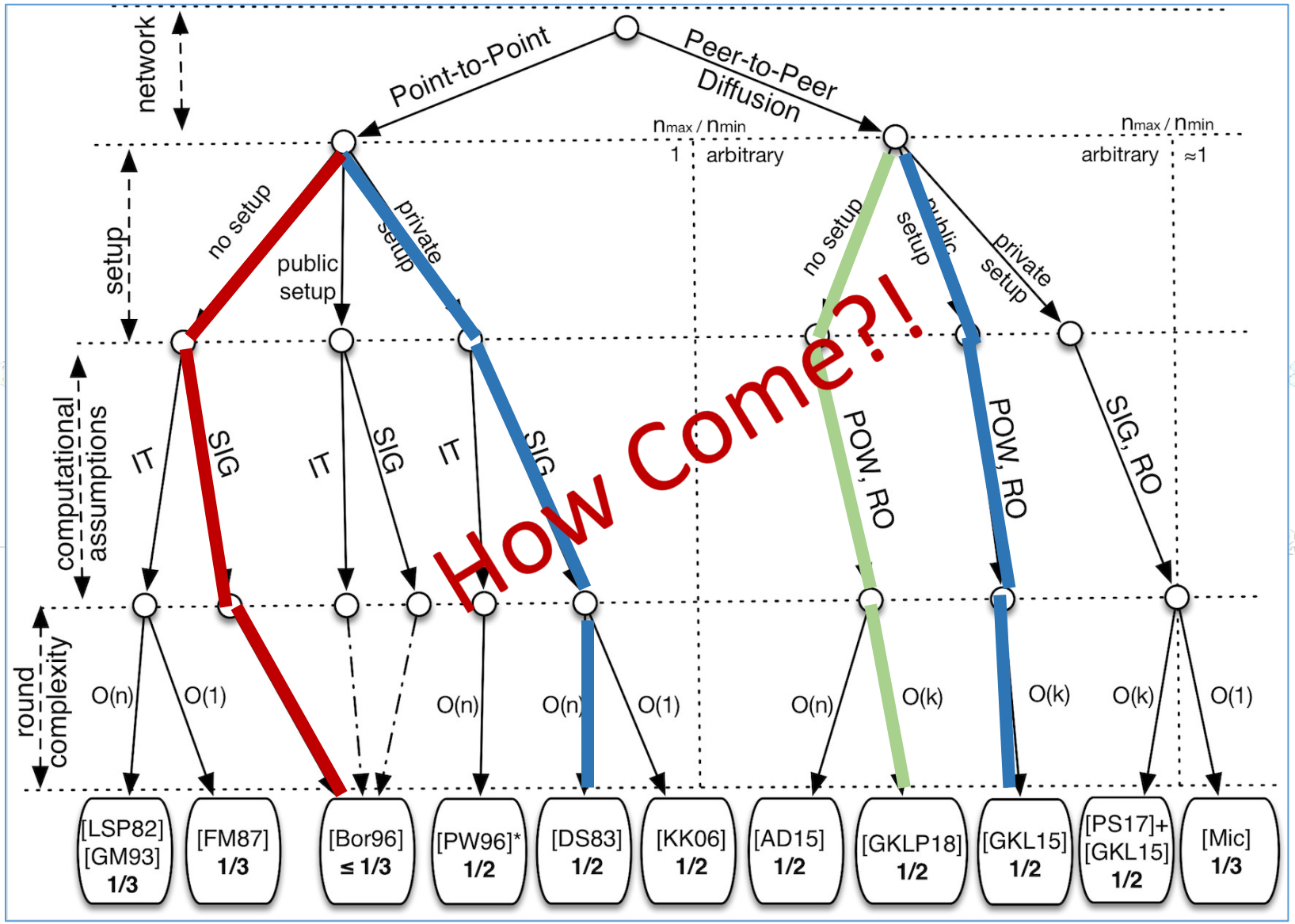
- The *n* parties start building a blockchain inserting their inputs
- If a party receives a longer blockchain switches to that one but *keeps the same input*
- Once the blockchain is long enough (2k) the parties prune the last k blocks and output the *majority value* in the prefix

- 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***

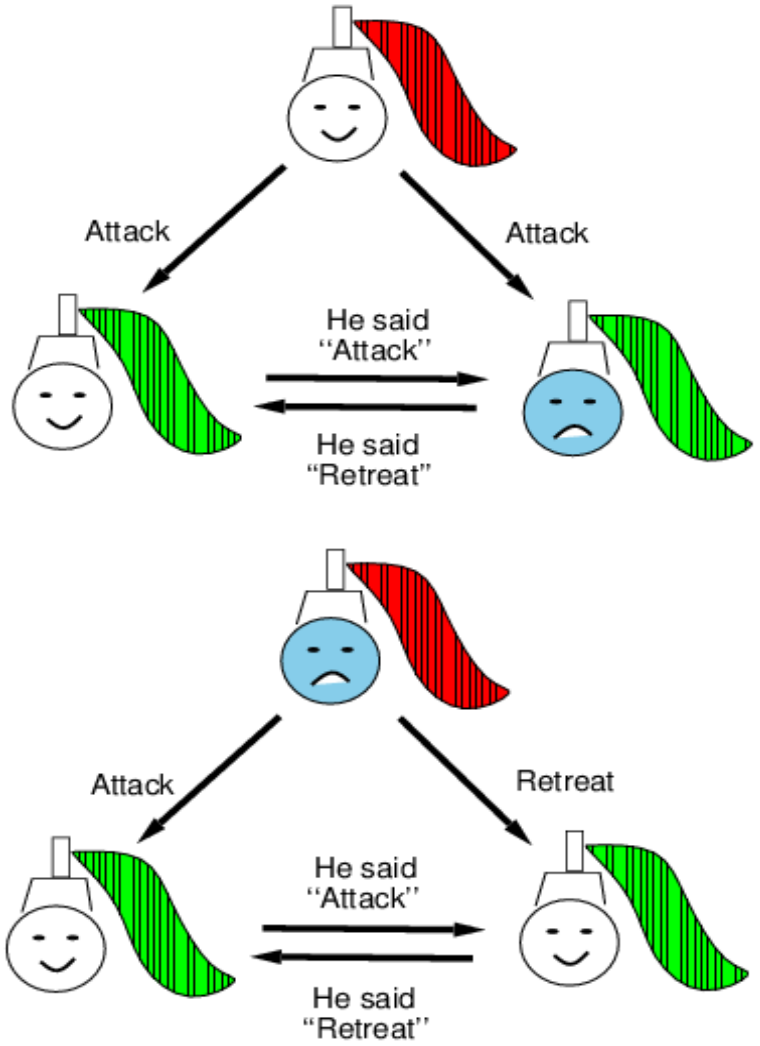
A Consensus Taxonomy [GK20]



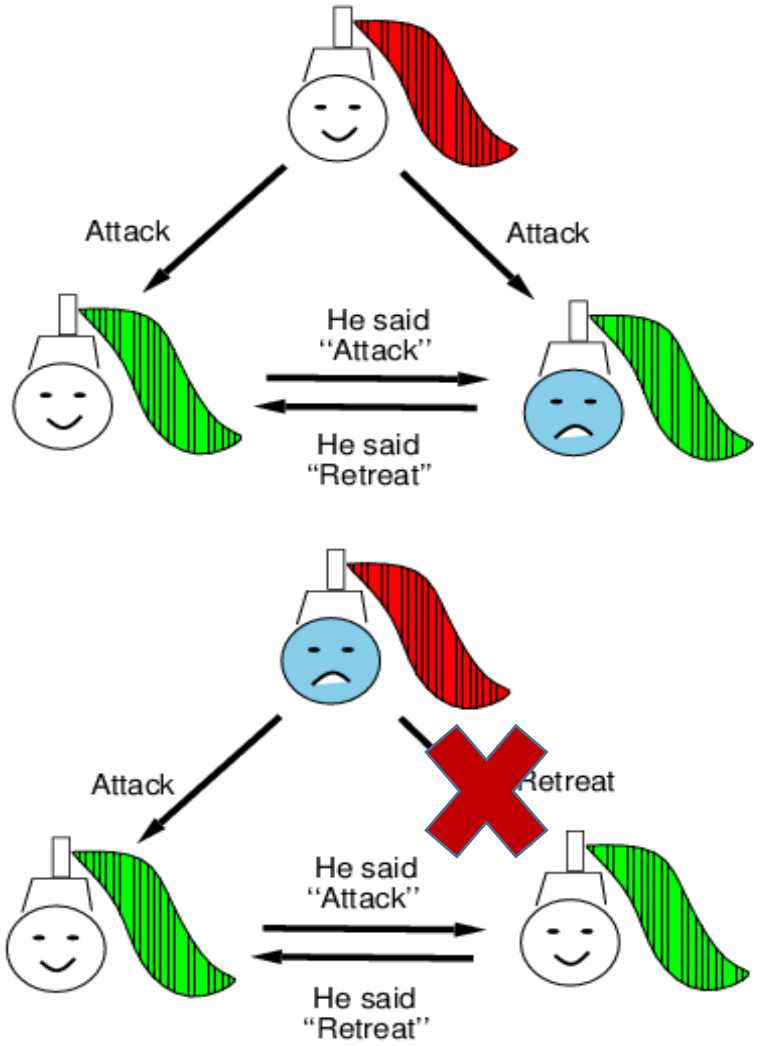
Resource-Restricted Cryptography

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,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

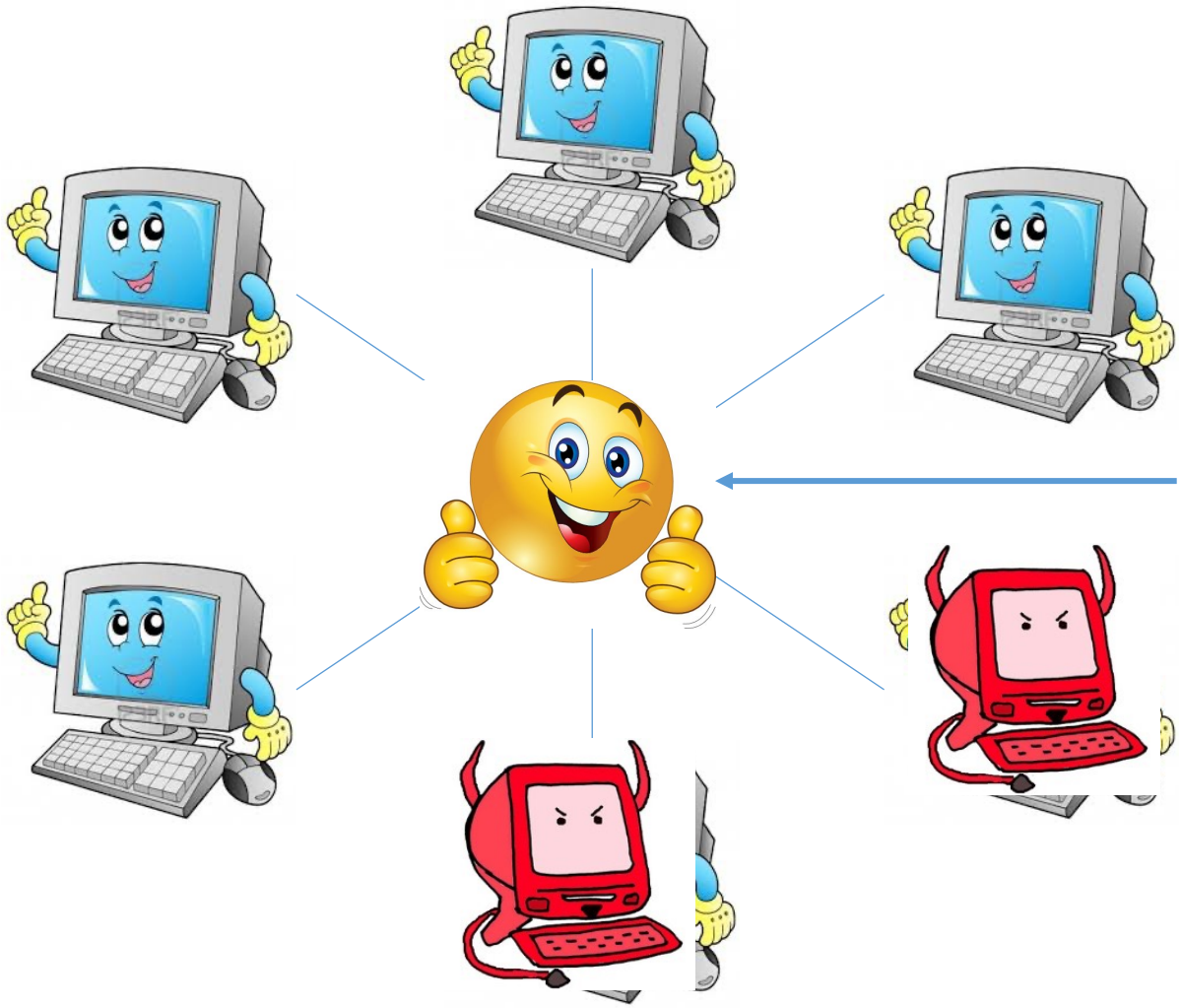
Secure Multiparty Computation (MPC) [GMW87]:

- 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

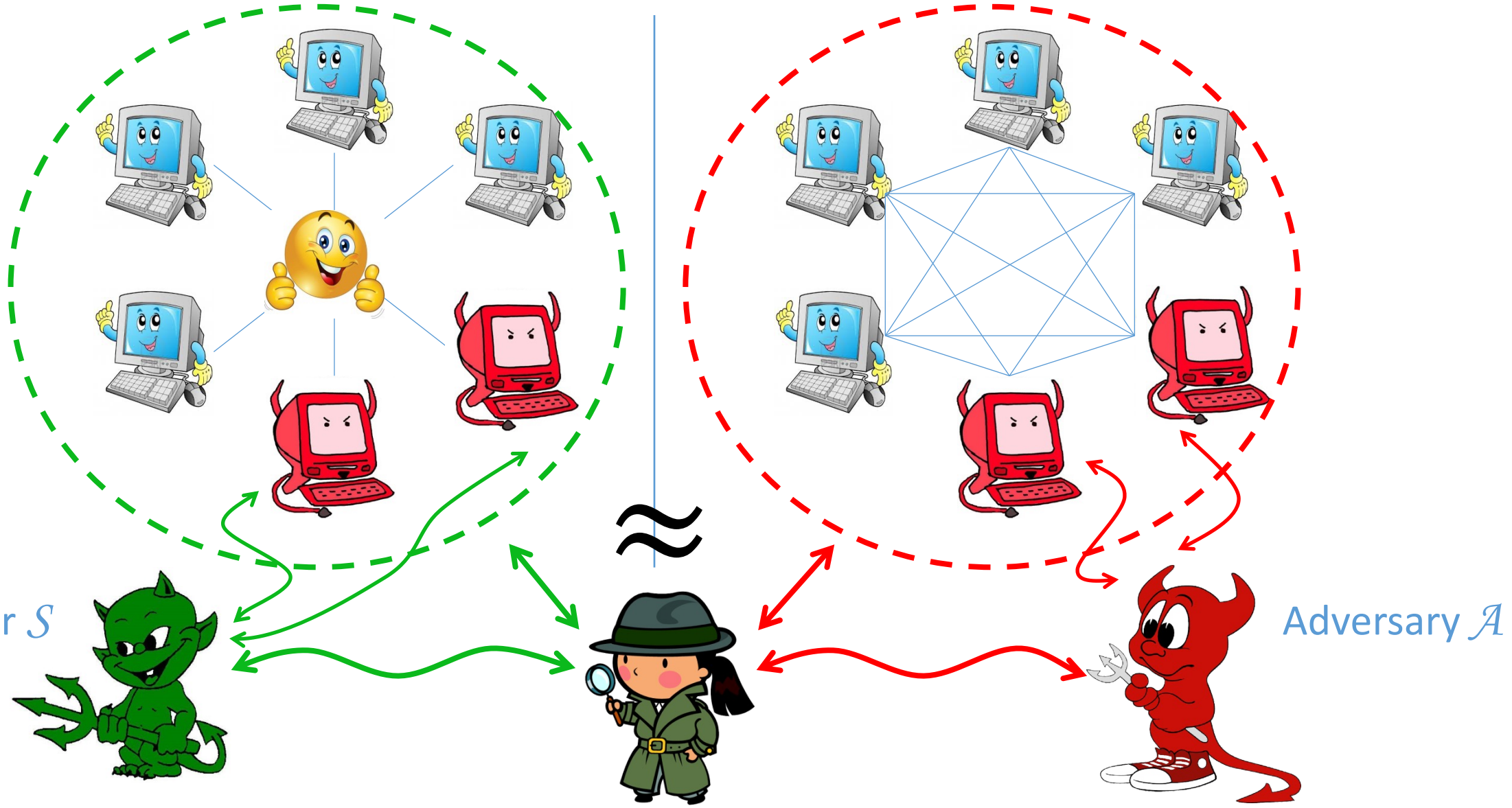
Detour:
Simulation-based Security

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

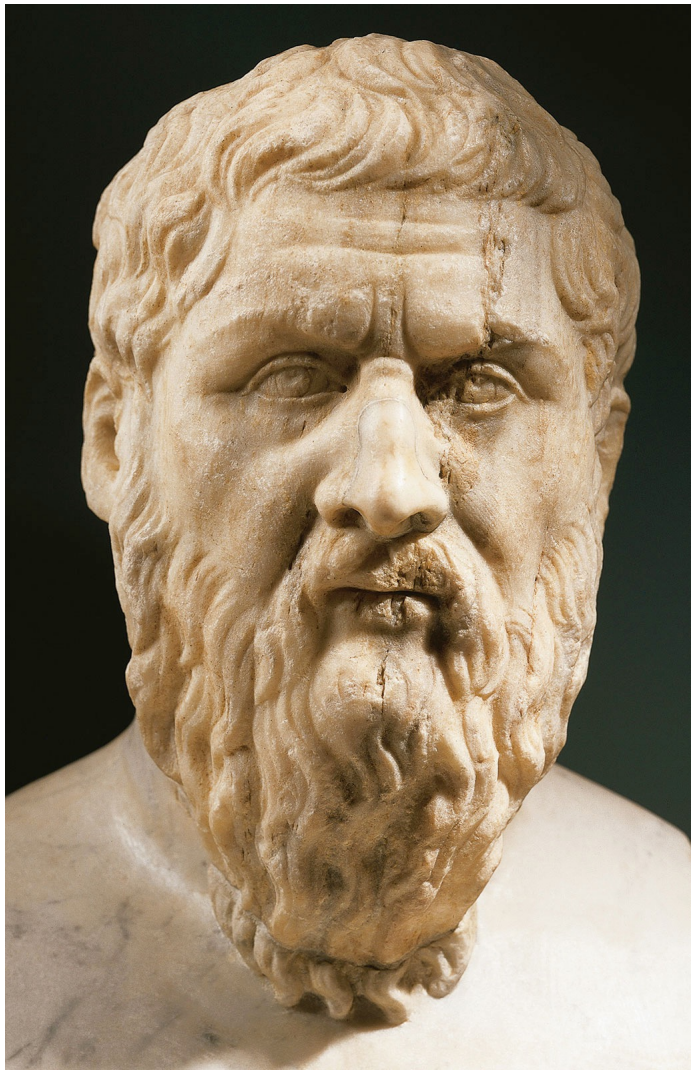


"Ideal functionality" $\mathcal{F}$ (arbitrary distributed computation)

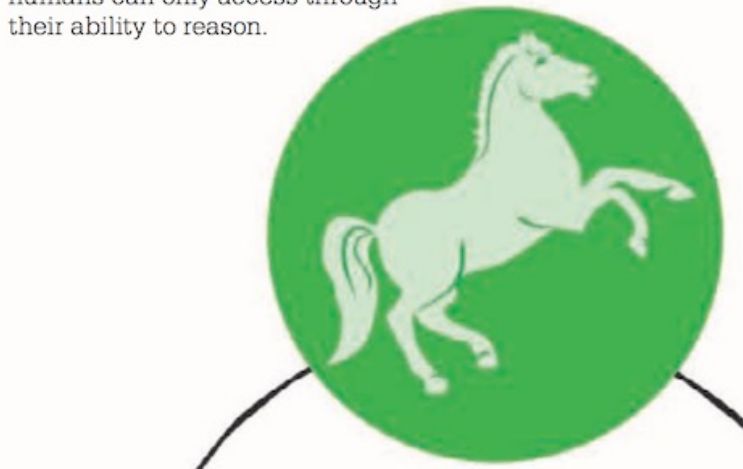
Simulation-based Security: Real vs Ideal [GMW87, C01,…]



Plato’s Theory of Forms



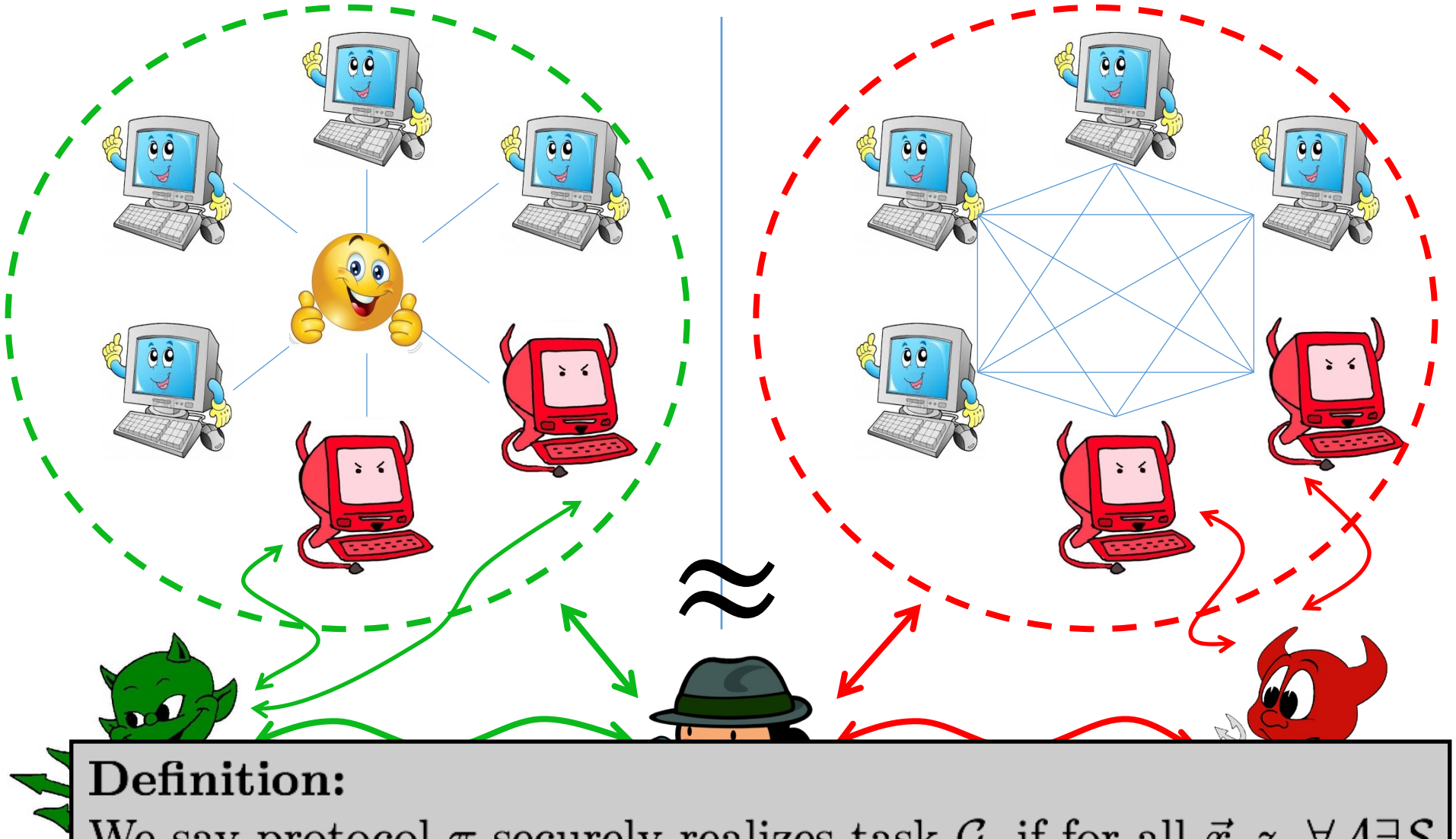
According to Plato's theory of Forms, every horse that we encounter in the world around us is a lesser version of an “ideal”, or perfect, horse that exists in a world of Forms or Ideas—a realm that humans can only access through their ability to reason.



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 $\{\mathsf{REAL}_{\pi,\mathcal{A}}\}_{\vec{x},z} \approx \{\mathsf{IDEAL}_{\mathcal{G},\mathcal{S}}\}_{\vec{x},z}$

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* (more later)
- Enables (universal) composition with other protocols [Canetti01,…]

RECALL

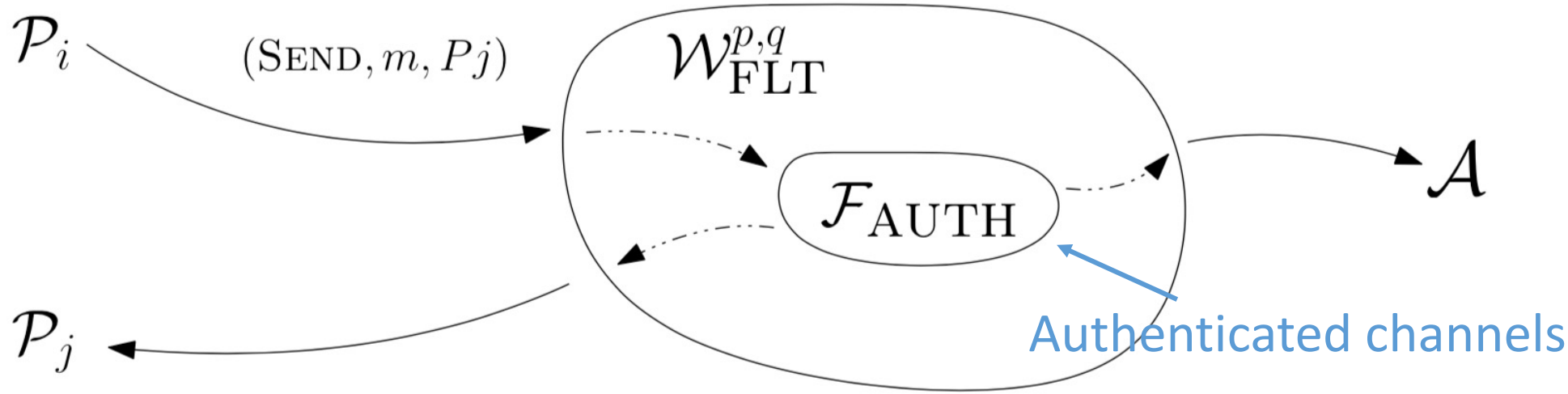
Resource-Restricted Cryptography

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

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?

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

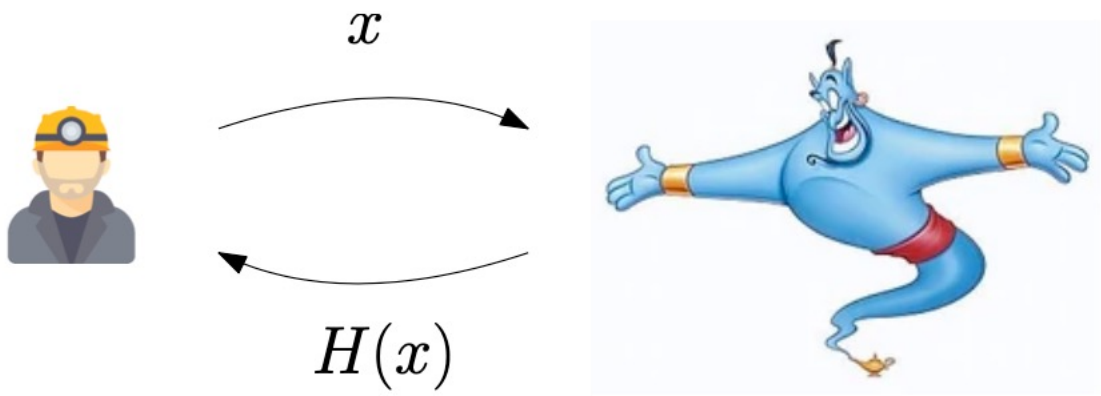
$\mathcal{F}_{\mathrm{MPC}}$ can be UC-implemented in the $(\mathcal{W}_{\mathrm{FLT}}^{p,q}(\mathcal{F}_{\mathrm{AUTH}}), \mathcal{F}_{\mathrm{SIG}}, \mathcal{F}_{\mathrm{SC}}, \mathcal{G}_{\mathrm{CLOCK}})$-hybrid model assuming $t < n/2$

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]

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



Proofs of Work (aka "Crypto Puzzles")

RECALL

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

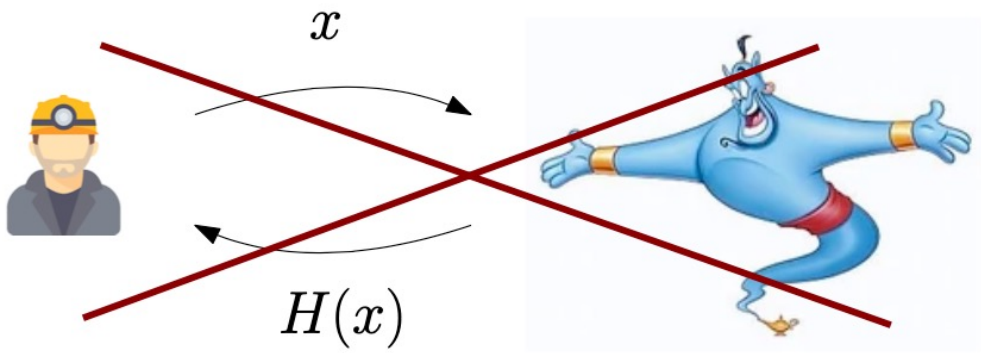
Prover **Verifier**

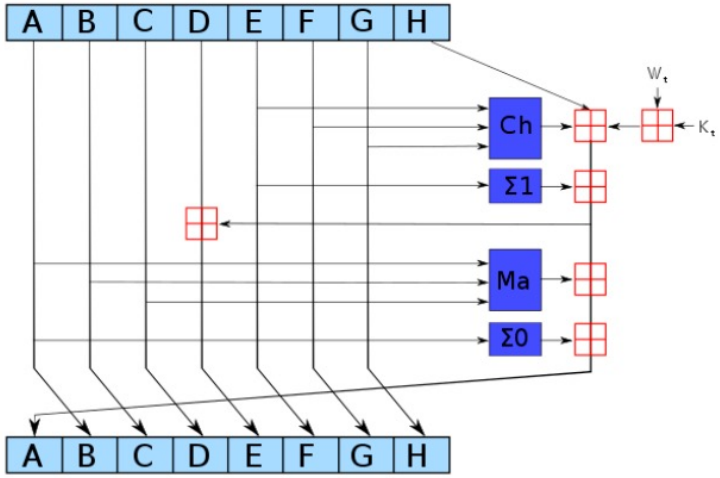
Image credit: Marshall Ball

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 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

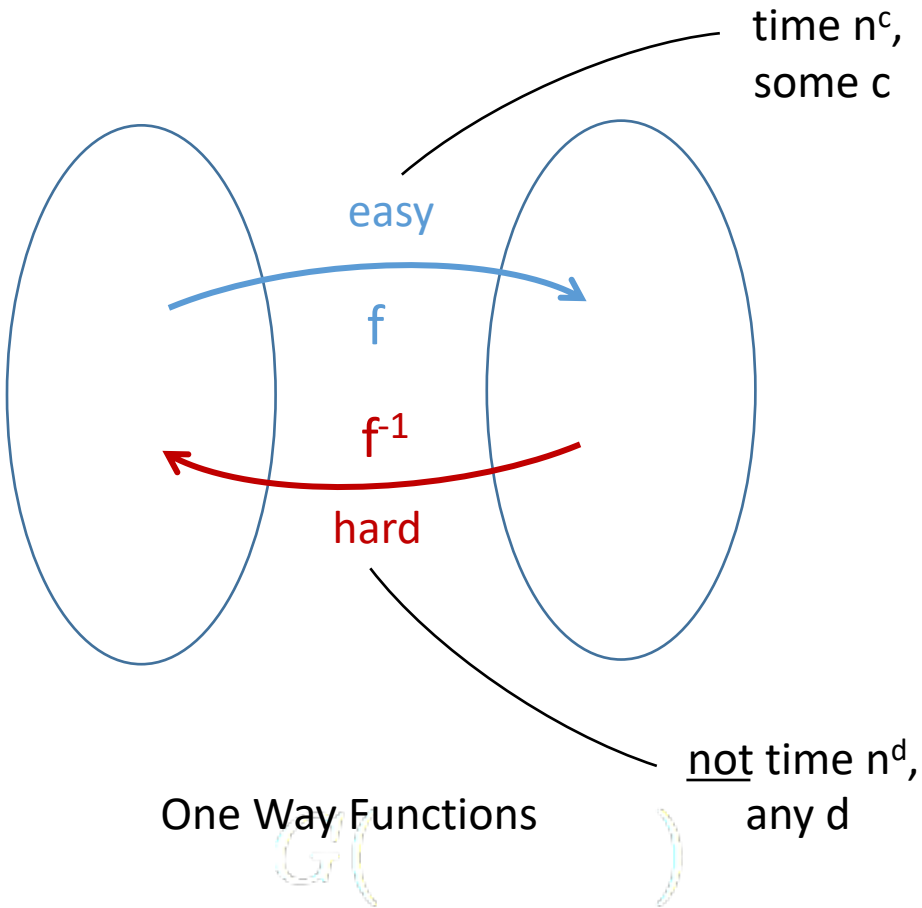
Topics

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

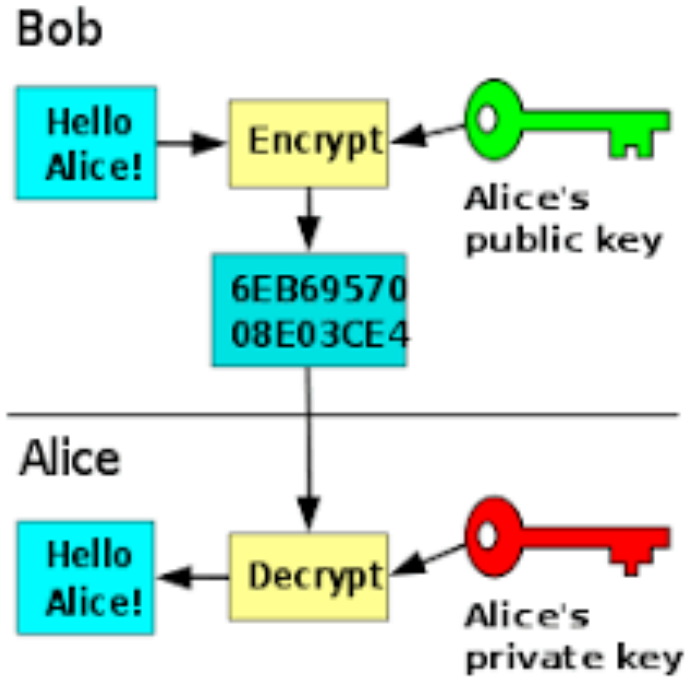
Cryptography

RECALL

Procedures that mitigate adversarial behavior



One Way Functions



Public Key Cryptography

Refined Adversarial Model

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

arbitrary constant

- "Fine-grained" adversary: Runs in n, n^2, n^3

fixed constant

  - Cf. fine-grained complexity theory

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

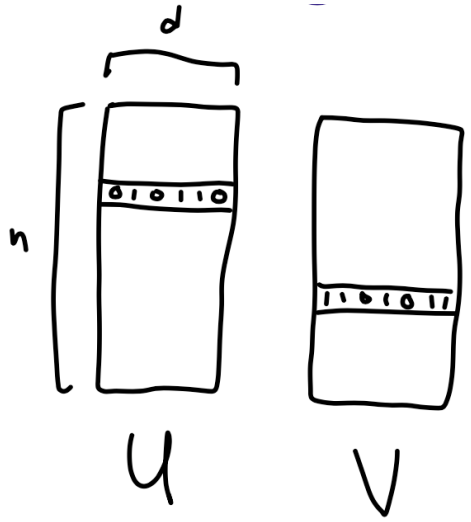
Fine-Grained Complexity

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

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

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

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]

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

- [BGKP21]: Assuming Orthogonal Vectors conjecture, (permissionless) consensus is achievable with public setup tolerating a constant fraction of corruptions
  - Cf. [AD15,GKLP18]

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
- Future/On-going work:
  - Revisit other lower bounds/impossibility results
  - Implement “filtering wrappers” by restricting other resources

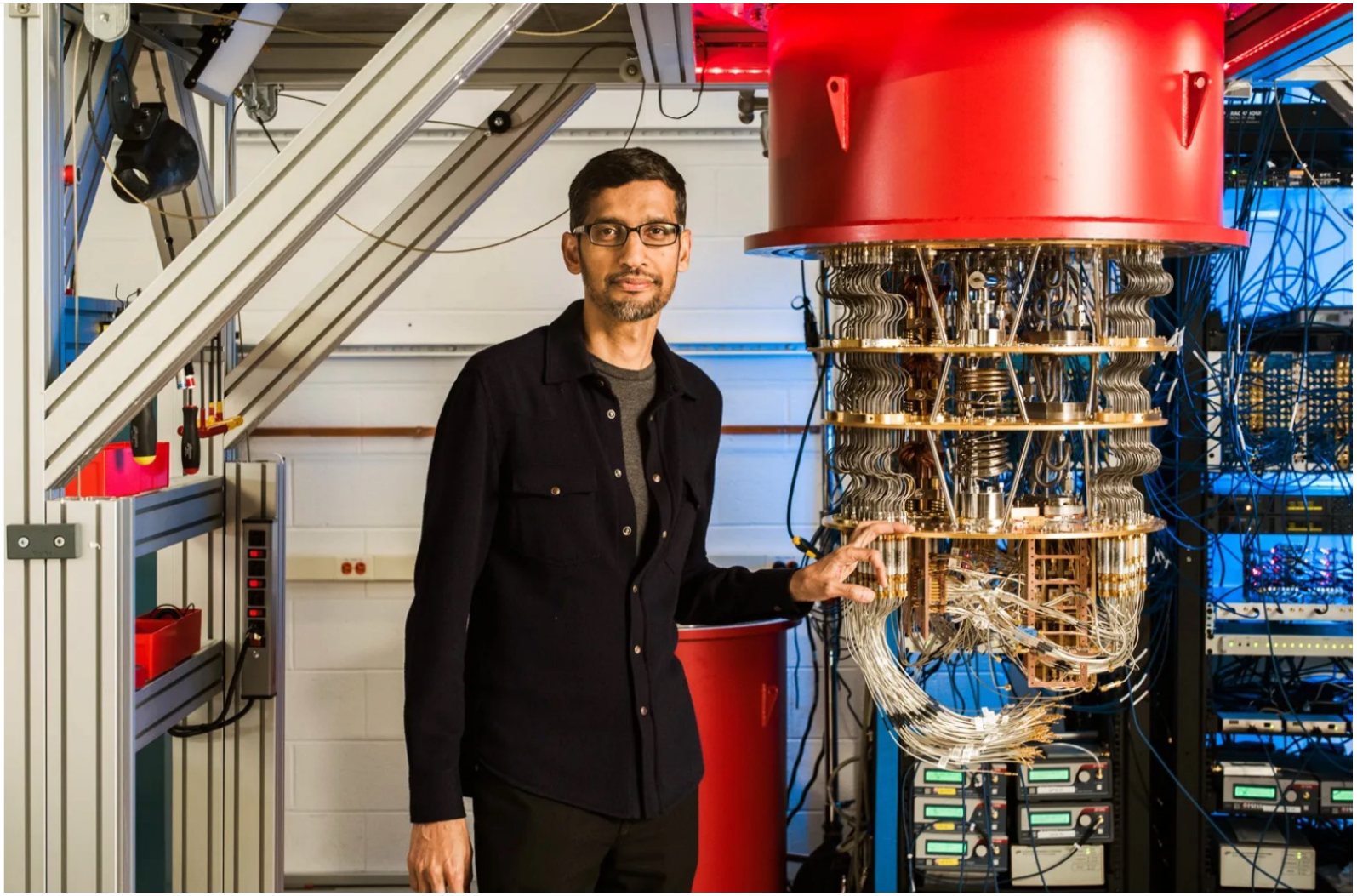
References

- 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
- 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
- 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
- R. Cohen, J. Garay and V. Zikas, “Completeness Theorems for Adaptively Secure Broadcast.” Full version at Cryptology ePrint Archive: Report 2021/775, https://eprint.iacr.org/2021/775
- M. Ball, J. Garay, A. Kiayias and G. Panagiotakos, “Permissionless Consensus from Proofs of Work in the Standard Model.” In preparation

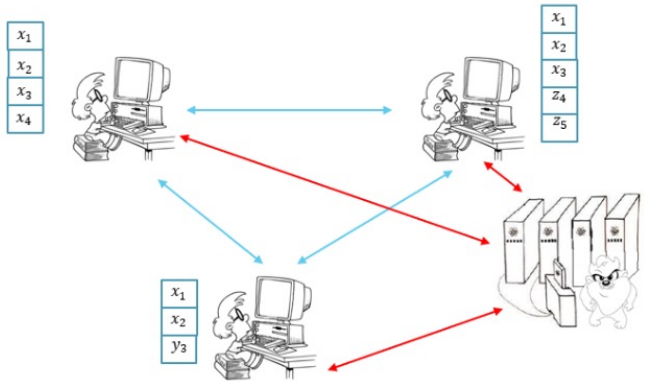

Thank You

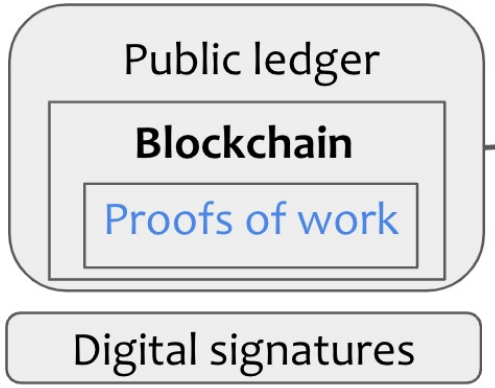
Backup Slides

“Quantum is Coming”



Post-Quantum Security



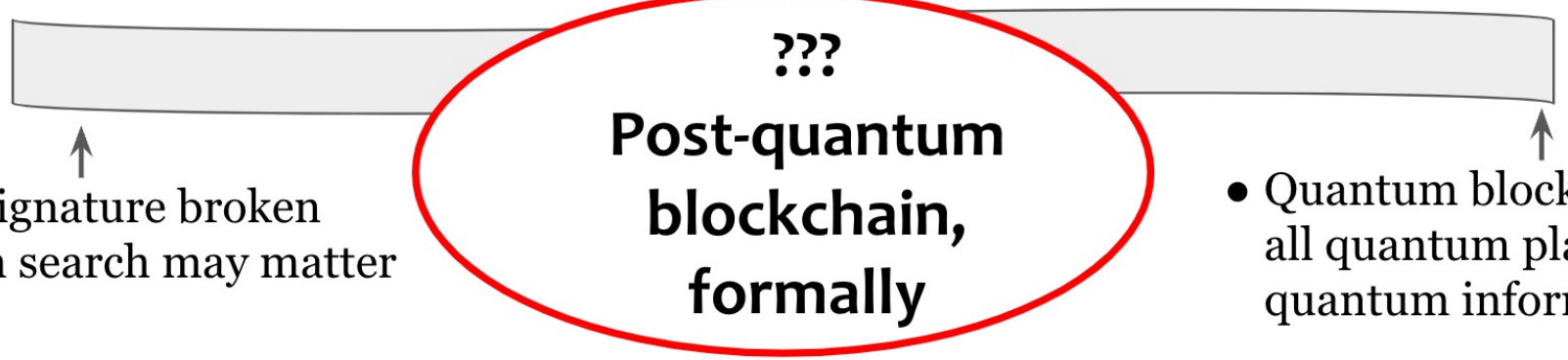


[GKL15]: (1st) Classical formal study

- 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 Security of the Bitcoin Backbone and Quantum Multi-Solution Bernoulli Search.” arXiv.org, December 2020, https://arxiv.org/abs/2012.15254