Pursuing the Limits of Cryptography

Arka Rai Choudhuri

Advised by Abhishek Jain

5th November 2021

Modern Cryptography

Digital Signatures

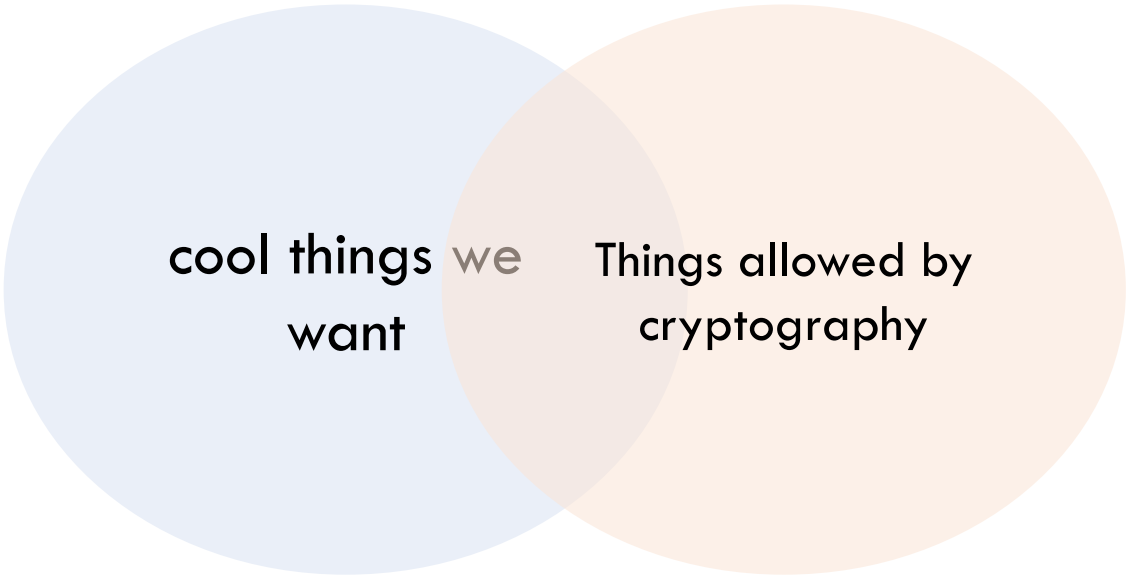
Digital Watermarking

Software Obfuscation

Computing over Encrypted Data

Limits of Cryptography

Limits of Cryptography



Focus of this work

Focus of this work

Interactive Zero-Knowledge Proofs

Interactive Zero-Knowledge Proofs

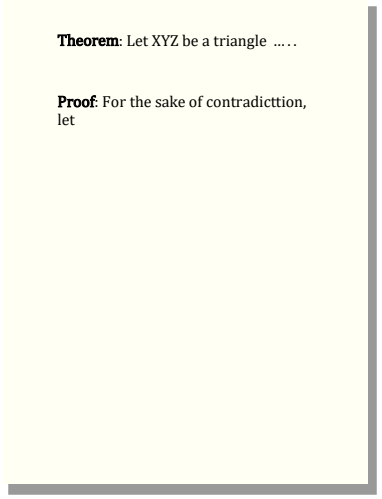
Interactive Zero-Knowledge Proofs

Theorem: Let XYZ be a triangle ….…

Proof: For the sake of contradicttion, let

Mathematical Proofs

Interactive Zero-Knowledge Proofs



Mathematical Proofs

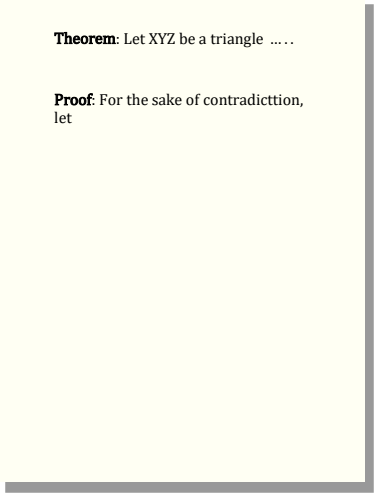
Interactive Zero-Knowledge Proofs

Theorem: Let XYZ be a triangle ….

Proof: For the sake of contradicttion, let

Mathematical Proofs

Interactive Zero-Knowledge Proofs



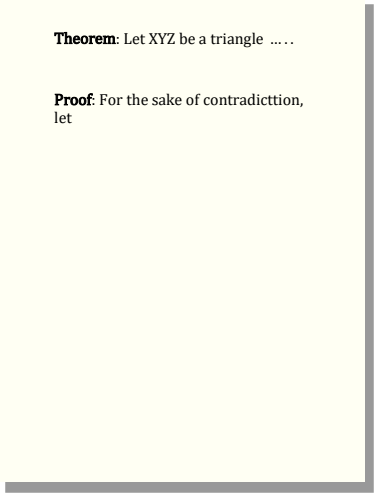
accept

Mathematical Proofs

Interactive Zero-Knowledge Proofs

Completeness

If the Theorem is true, Alice should be able to convince Bob.



accept

Mathematical Proofs

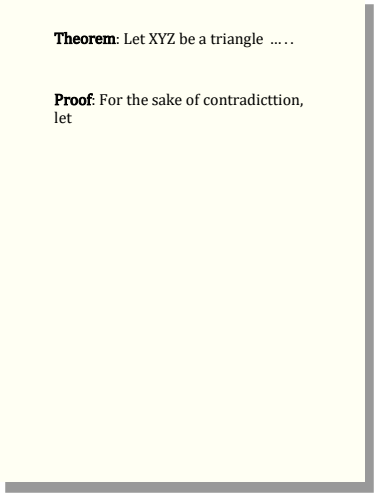
Interactive Zero-Knowledge Proofs

Completeness

If the Theorem is true, Alice should be able to convince Bob.

Soundness

If the Theorem is false, Alice should not be able to convince Bob.



accept

Mathematical Proofs

Interactive Zero-Knowledge Proofs

[Goldwasser-Micali-Rackoff'85, Babai-Moran'88]

Completeness

If the Theorem is true, Alice should be able to convince Bob.

Soundness

If the Theorem is false, Alice should not be able to convince Bob.

Interactive Zero-Knowledge Proofs

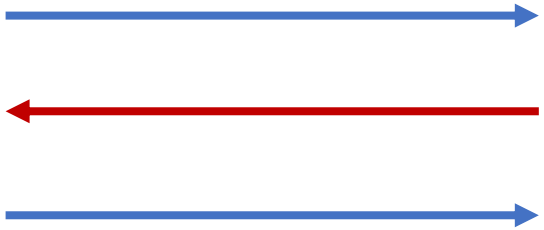
[Goldwasser-Micali-Rackoff'85, Babai-Moran'88]

Completeness

If the Theorem is true, Alice should be able to convince Bob.

Soundness

If the Theorem is false, Alice should not be able to convince Bob.



1 Interaction

accept

Interactive Zero-Knowledge Proofs

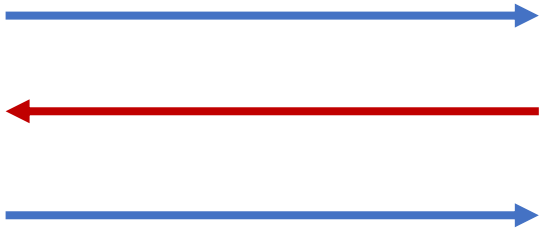
[Goldwasser-Micali-Rackoff'85, Babai-Moran'88]

Completeness

If the Theorem is true, Alice should be able to convince Bob.

Soundness

If the Theorem is false, Alice has a **small** chance to convince Bob.



1 Interaction

accept/reject

2 Soundness error

Interactive Zero-Knowledge Proofs

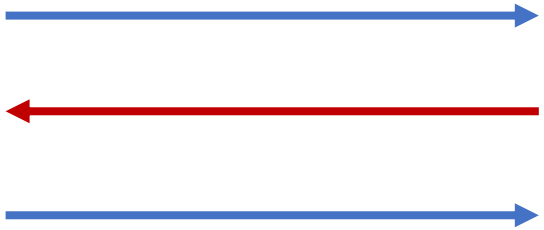
[Goldwasser-Micali-Rackoff'85]

Completeness

If the Theorem is true, Alice should be able to convince Bob.

Soundness

If the Theorem is false, Alice has a small chance to convince Bob.



Zero-Knowledge

Interaction reveals nothing beyond the validity of the proposition.

Interactive Zero-Knowledge Proofs

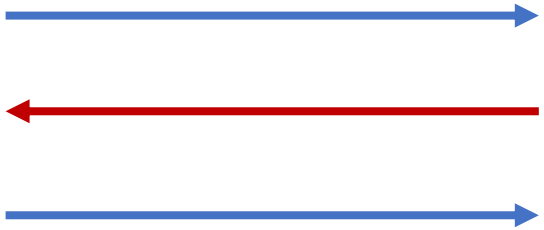
[Goldwasser-Micali-Rackoff'85]

Completeness

If the Theorem is true, Alice should be able to convince Bob.

Soundness

If the Theorem is false, Alice has a small chance to convince Bob.



If the proposition is true, Bob might as well have generated the interaction on their own.

Interactive Zero-Knowledge Proofs

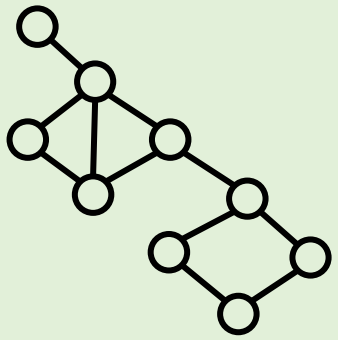
Graph 3-Coloring

Given a graph, can the vertices be colored such that no two vertices connected by an edge have the same color?

Interactive Zero-Knowledge Proofs

Graph 3-Coloring

Given a graph, can the vertices be colored such that no two vertices connected by an edge have the same color?

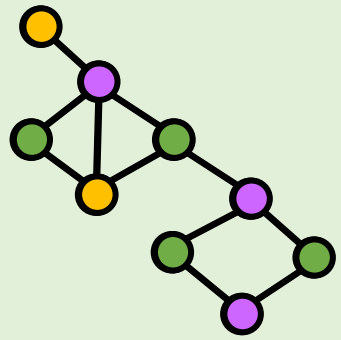


3 colorable

Interactive Zero-Knowledge Proofs

Graph 3-Coloring

Given a graph, can the vertices be colored such that no two vertices connected by an edge have the same color?

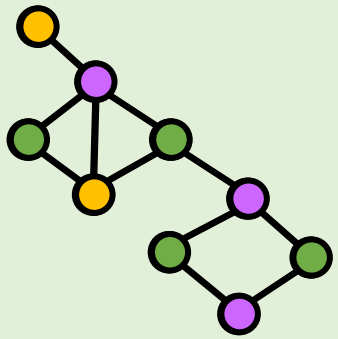


3 colorable

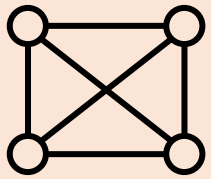
Interactive Zero-Knowledge Proofs

Graph 3-Coloring

Given a graph, can the vertices be colored such that no two vertices connected by an edge have the same color?



3 colorable

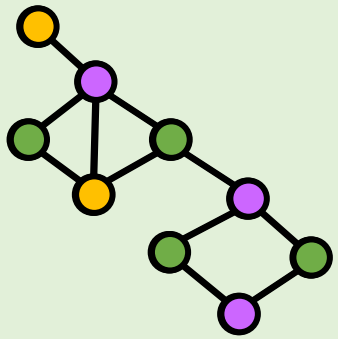


Not 3 colorable

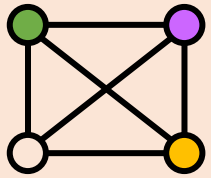
Interactive Zero-Knowledge Proofs

Graph 3-Coloring

Given a graph, can the vertices be colored such that no two vertices connected by an edge have the same color?



3 colorable



Not 3 colorable

Interactive Zero-Knowledge Proofs

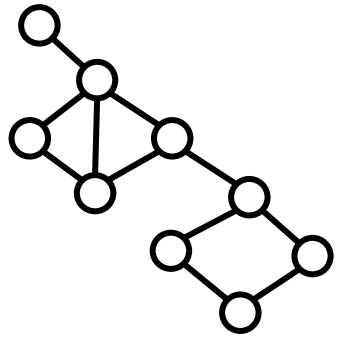
Graph 3-Coloring

Given a graph, can the vertices be colored such that no two vertices connected by an edge have the same color?

Known to be NP-Complete

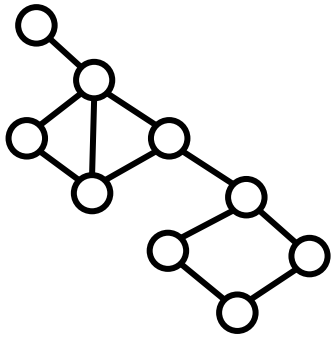
Hardest among the set of problems NP, whose solutions are easy to verify.

Interactive Zero-Knowledge Proofs

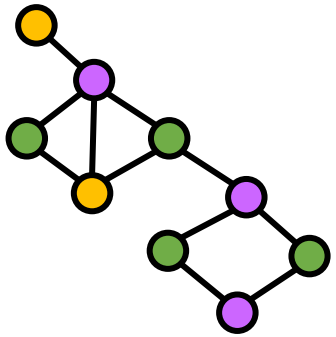


Alice wants to convince Bob that the graph is 3-colorable.

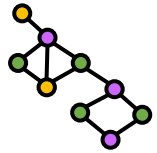
Interactive Zero-Knowledge Proofs



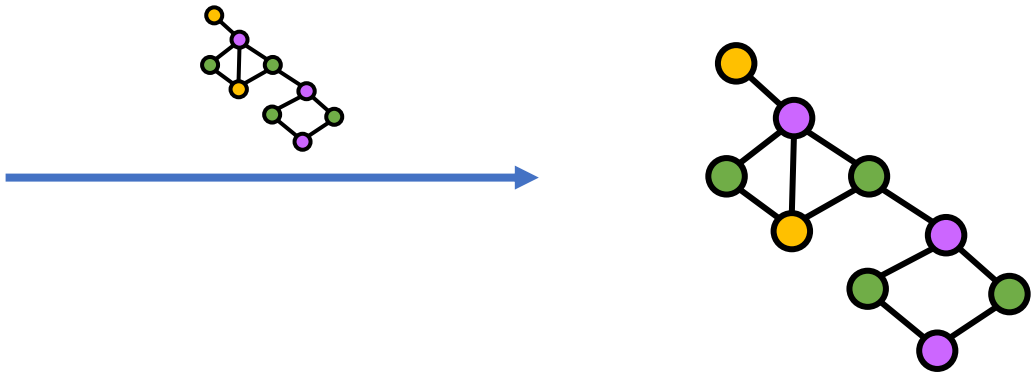
Interactive Zero-Knowledge Proofs



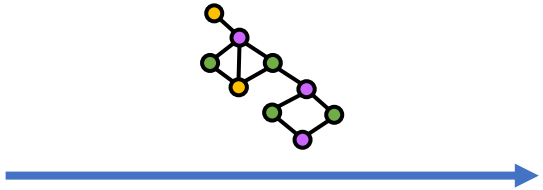
Interactive Zero-Knowledge Proofs

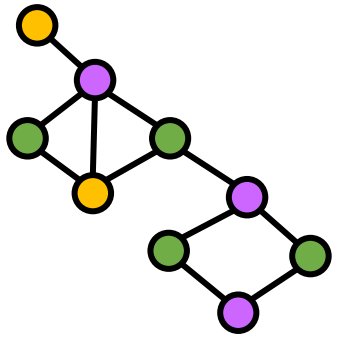


Interactive Zero-Knowledge Proofs



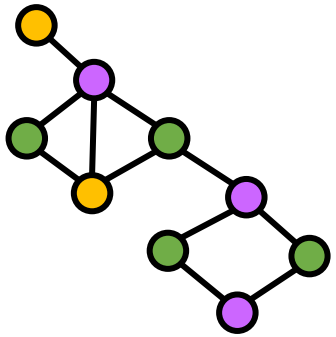
Interactive Zero-Knowledge Proofs





Bob learns the coloring, not zero-knowledge.

Interactive Zero-Knowledge Proofs



Interactive Zero-Knowledge Proofs

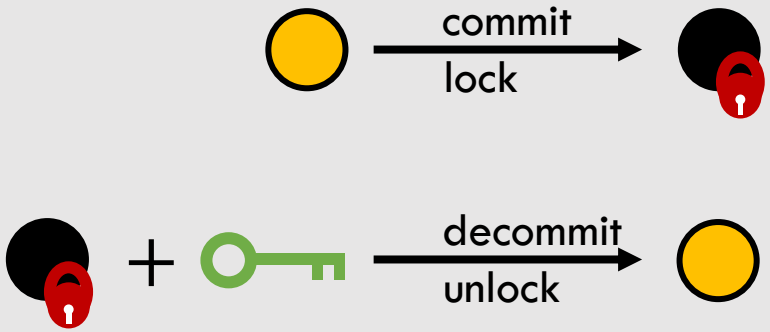
Digital Analogue of Locked Boxes: Commitment schemes

Interactive Zero-Knowledge Proofs

Digital Analogue of Locked Boxes: Commitment schemes

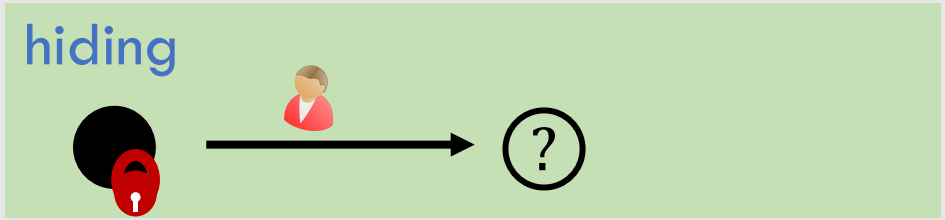
Interactive Zero-Knowledge Proofs

Digital Analogue of Locked Boxes: Commitment schemes



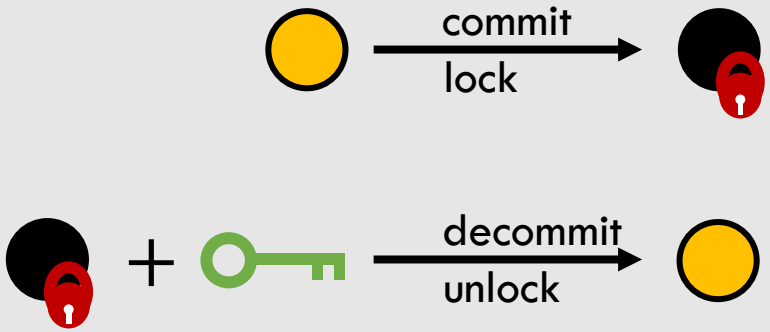
Interactive Zero-Knowledge Proofs

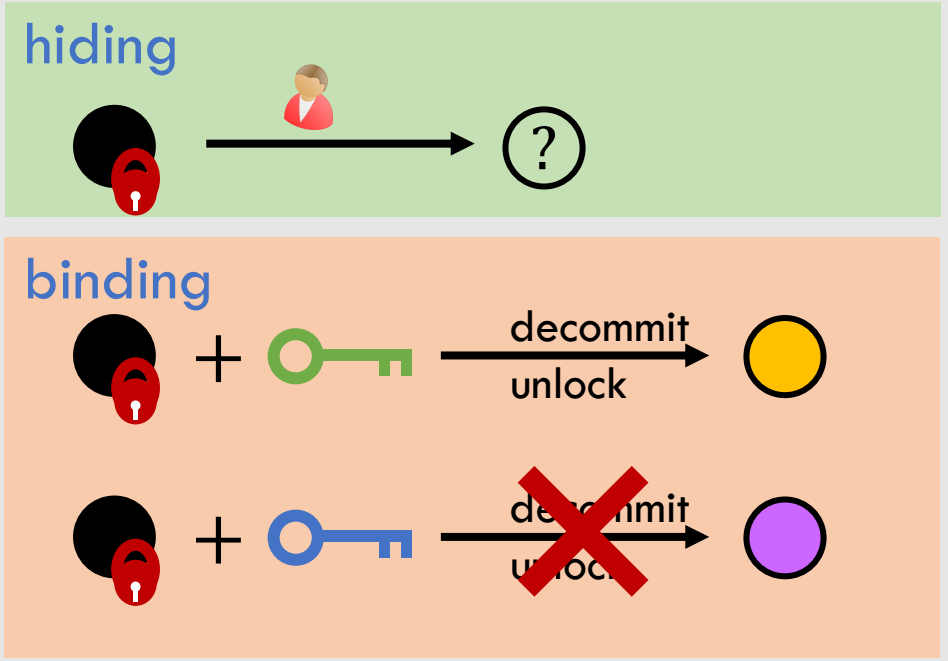
Digital Analogue of Locked Boxes: Commitment schemes



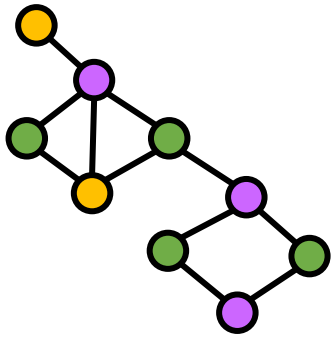
Interactive Zero-Knowledge Proofs

Digital Analogue of Locked Boxes: Commitment schemes



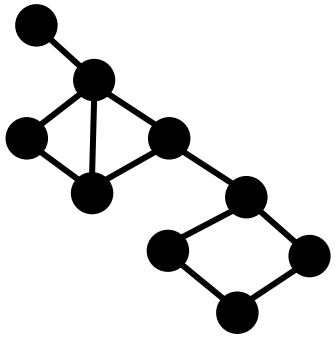


Interactive Zero-Knowledge Proofs

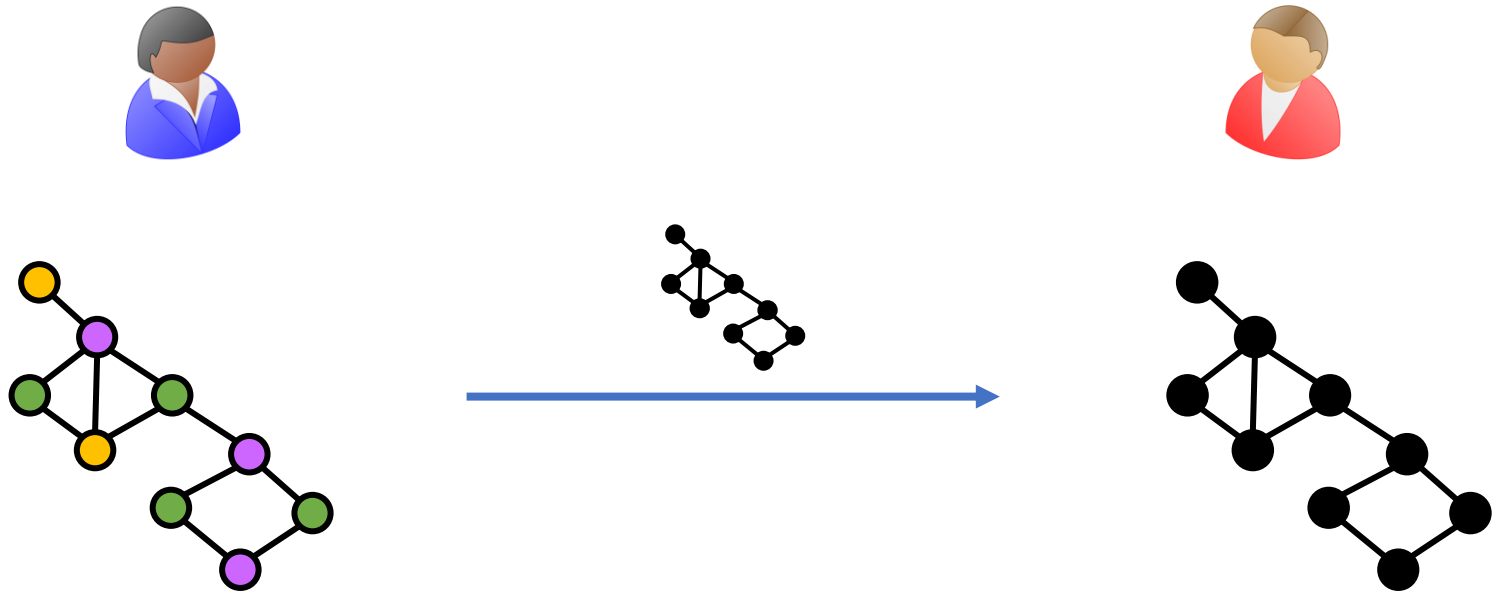


Interactive Zero-Knowledge Proofs

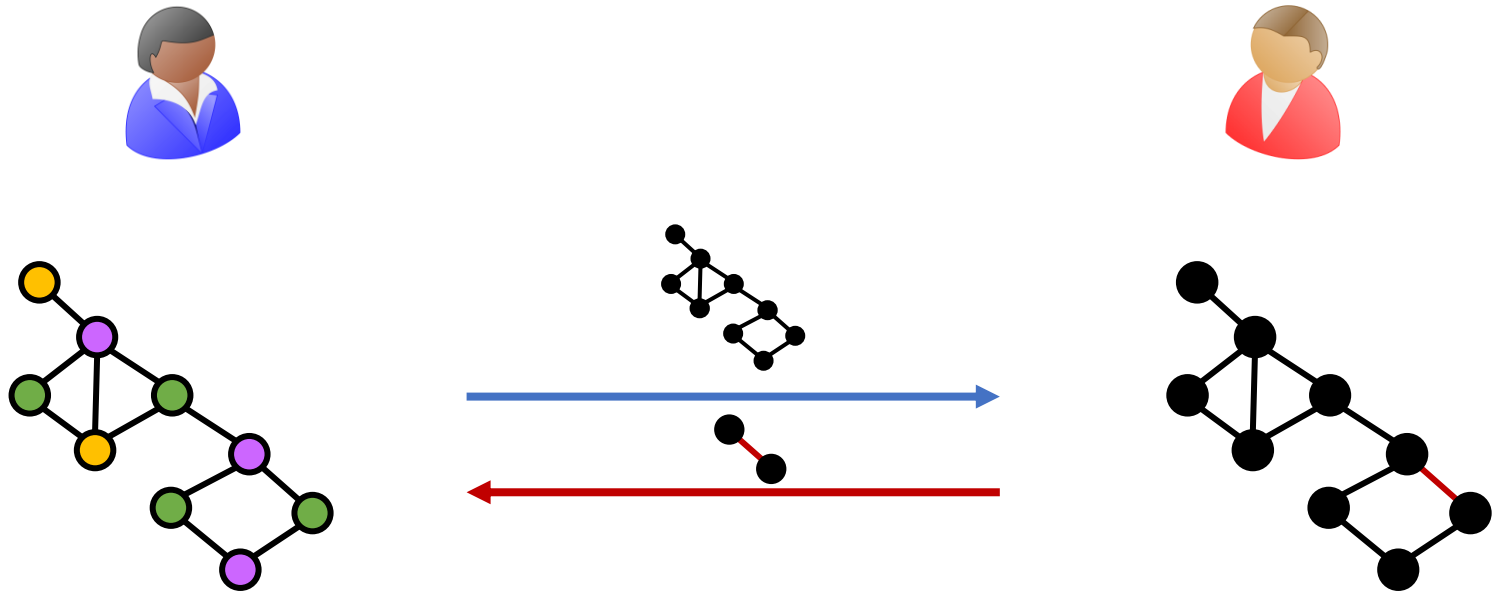
Lock/commit to the vertex colors



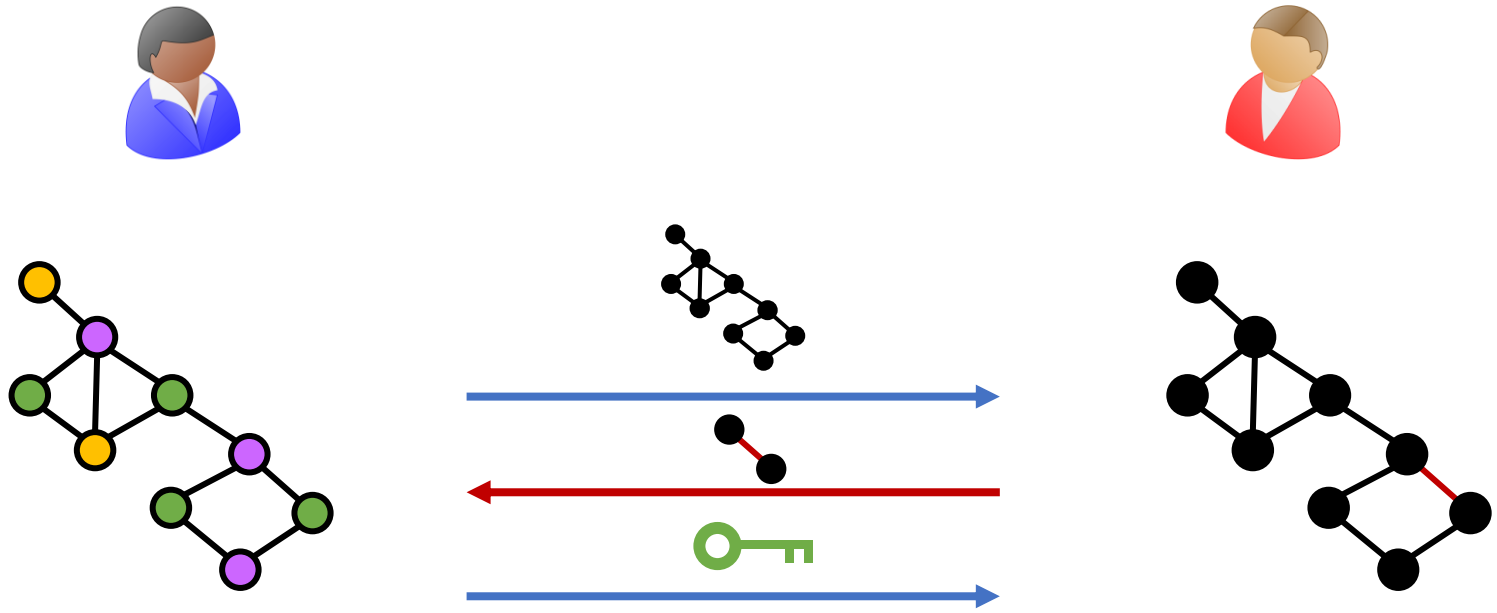
Interactive Zero-Knowledge Proofs



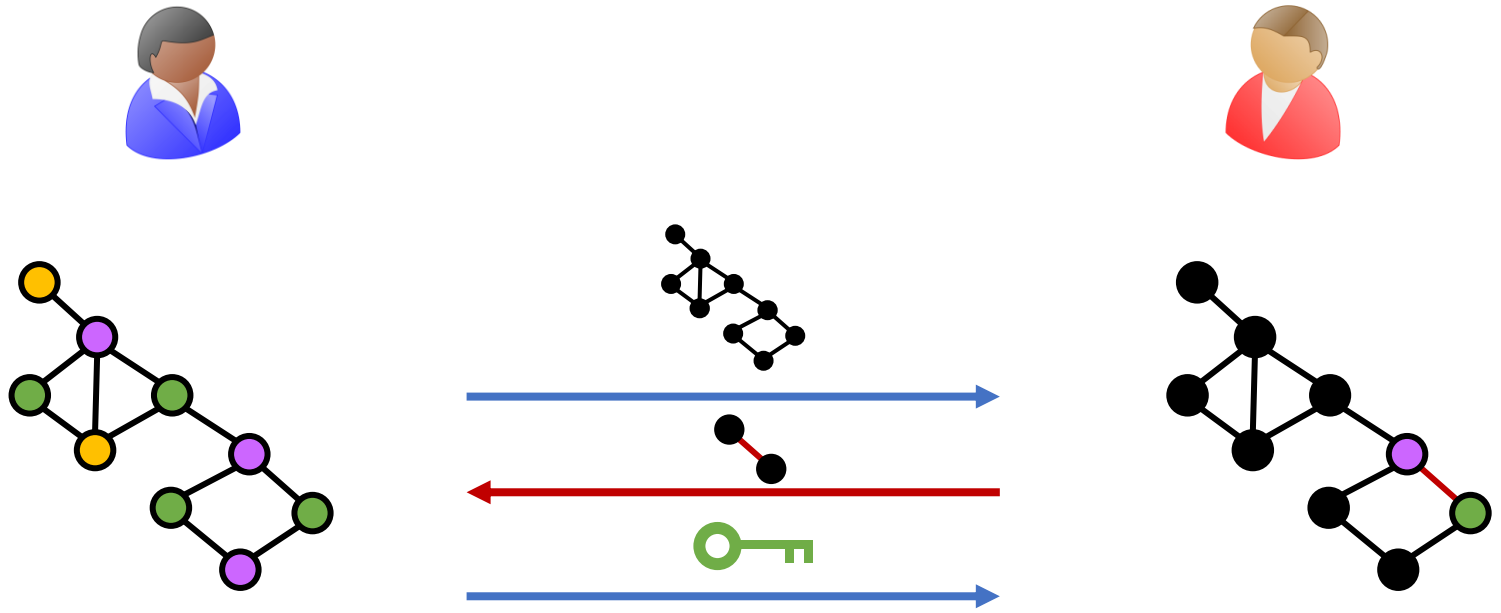
Interactive Zero-Knowledge Proofs



Interactive Zero-Knowledge Proofs

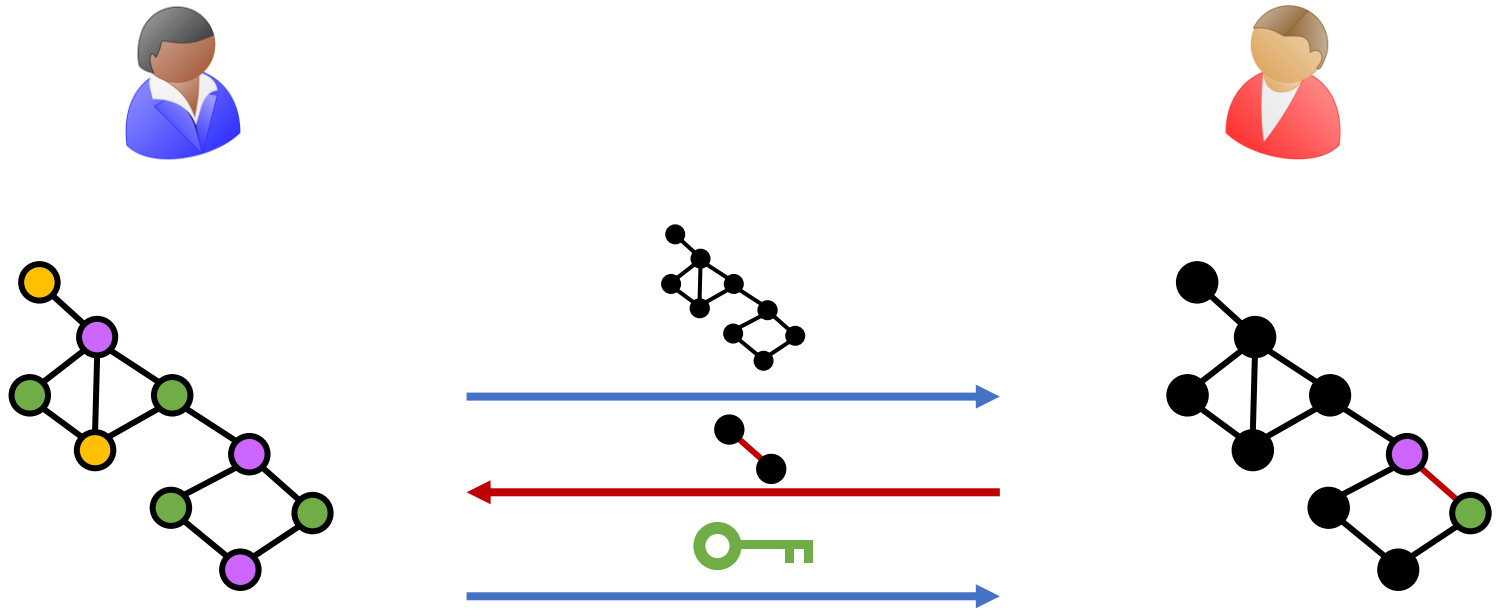


Interactive Zero-Knowledge Proofs



Bob checks if the colors are different.

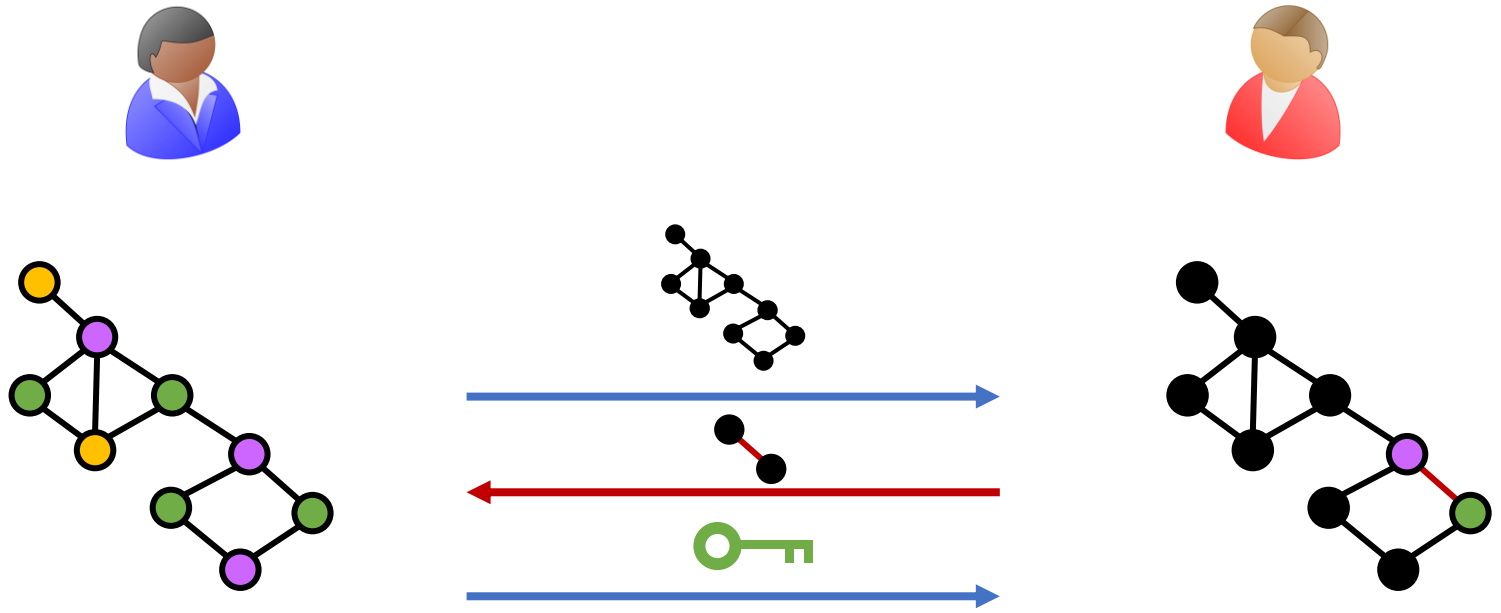
Interactive Zero-Knowledge Proofs



Bob checks if the colors are different.

Bob only learns that vertices connected to the chosen edge have different colors.

Interactive Zero-Knowledge Proofs

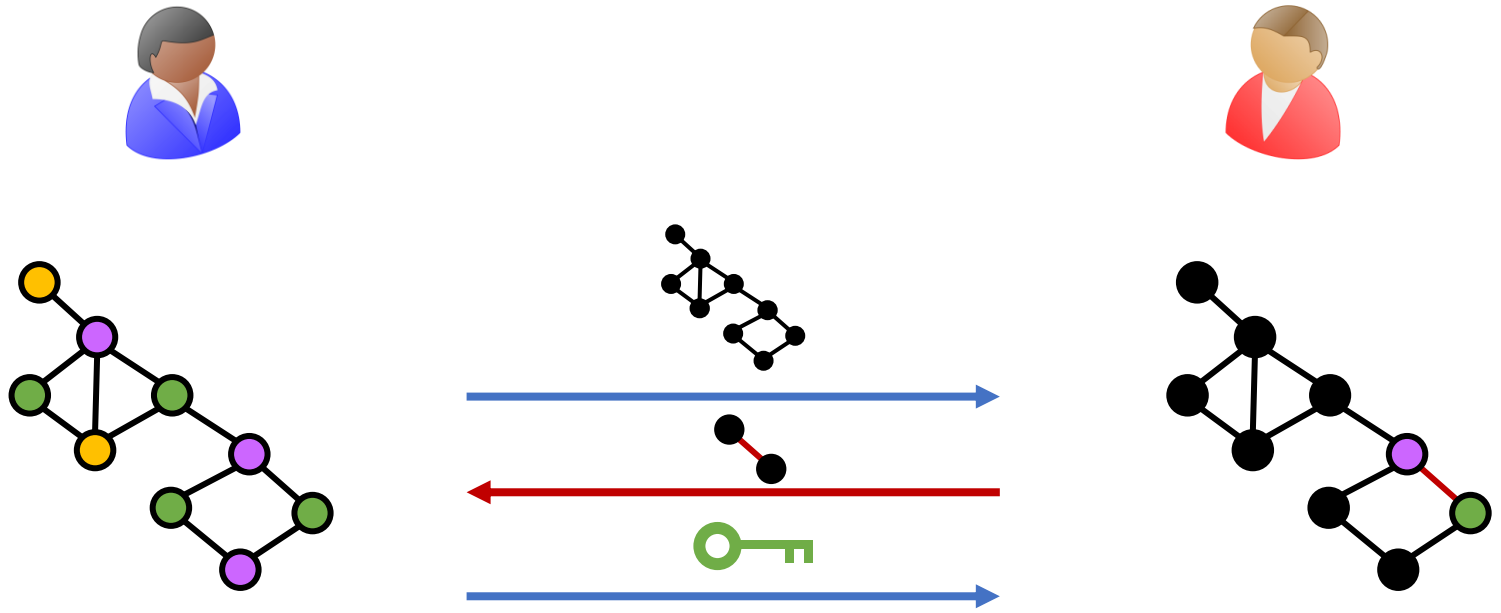


Bob checks if the colors are different.

If graph is not 3-colorable, Bob picks an edge with adjacent vertices of the same color with probability $\frac{1}{\#\text{Edges}}$

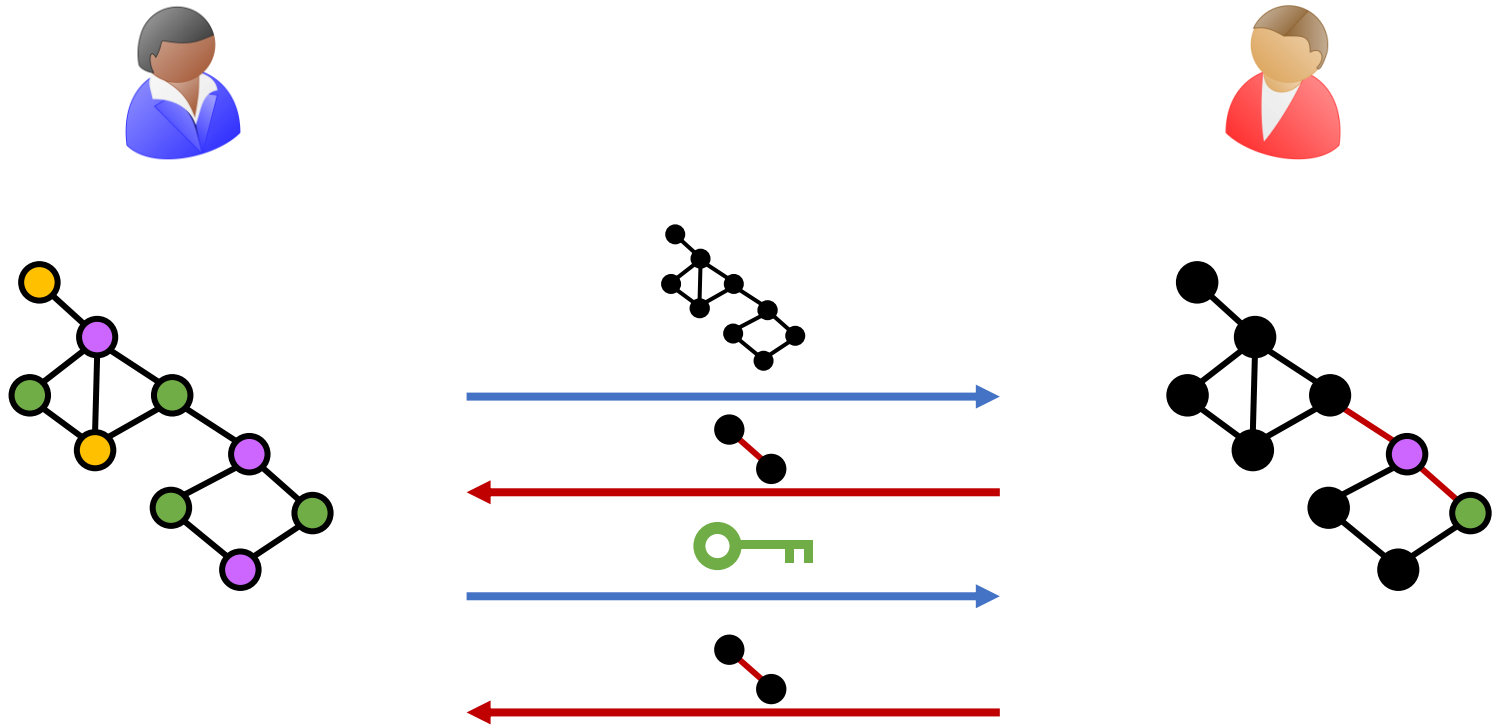
Repeat for improved confidence.

Interactive Zero-Knowledge Proofs



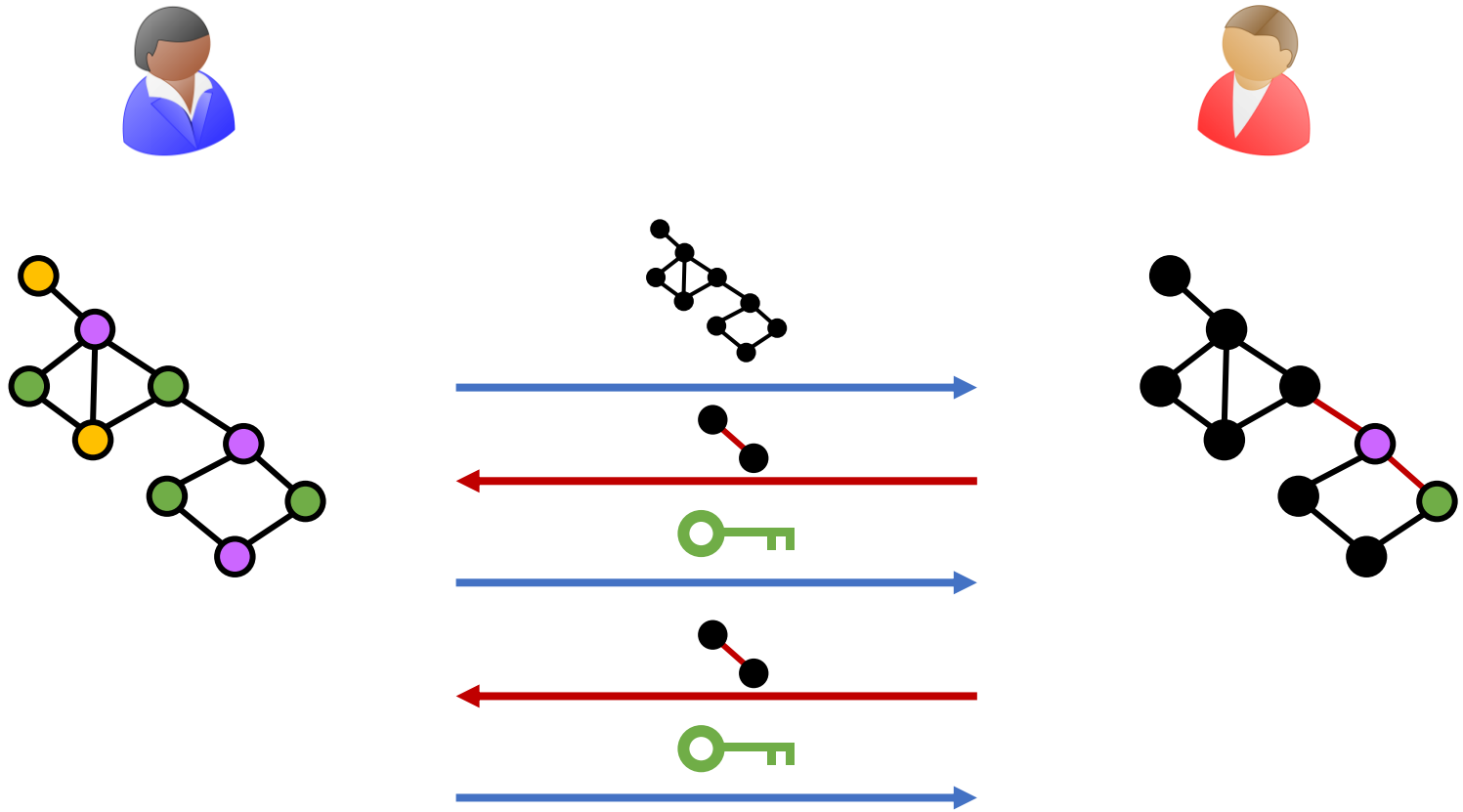
Bob checks if the colors are different.

Interactive Zero-Knowledge Proofs



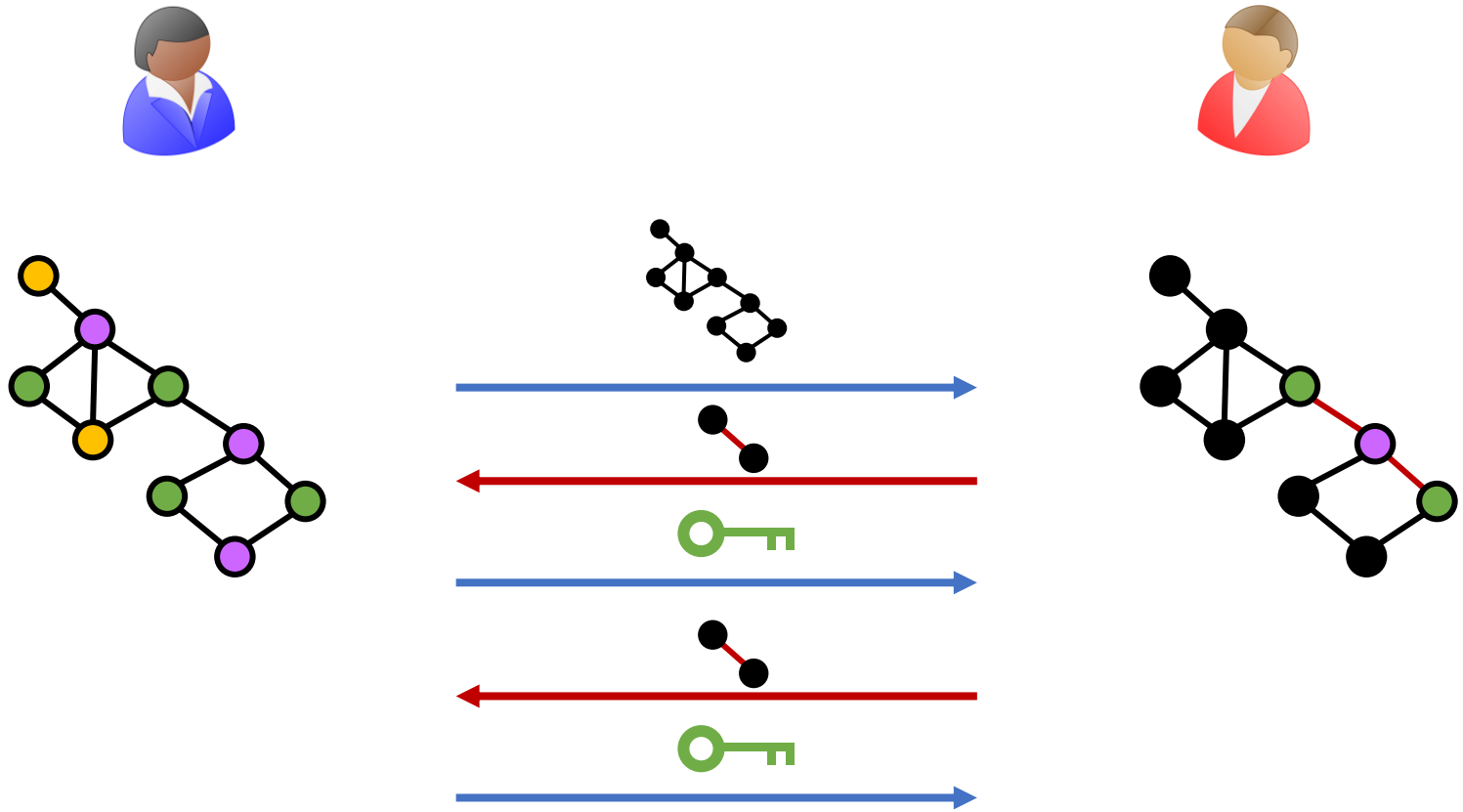
Bob checks if the colors are different.

Interactive Zero-Knowledge Proofs



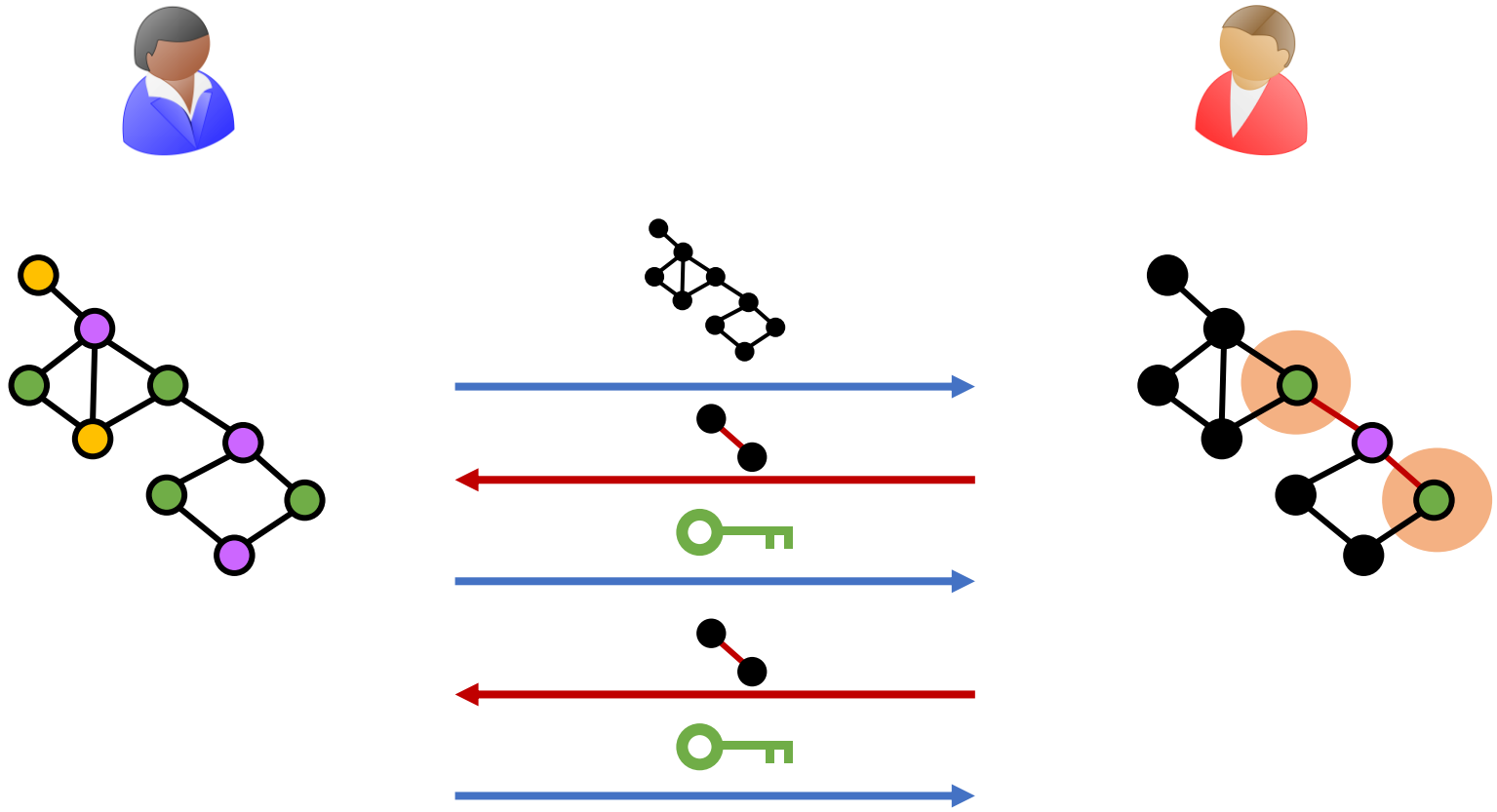
Bob checks if the colors are different.

Interactive Zero-Knowledge Proofs



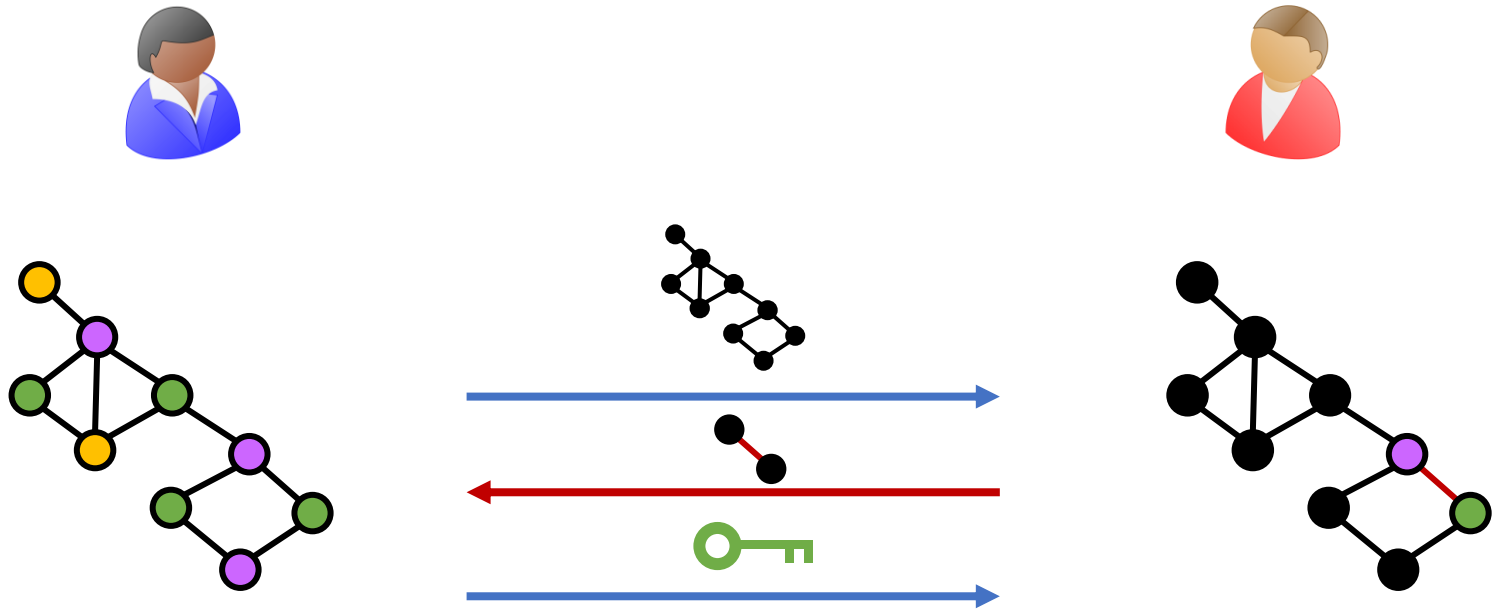
Bob checks if the colors are different.

Interactive Zero-Knowledge Proofs

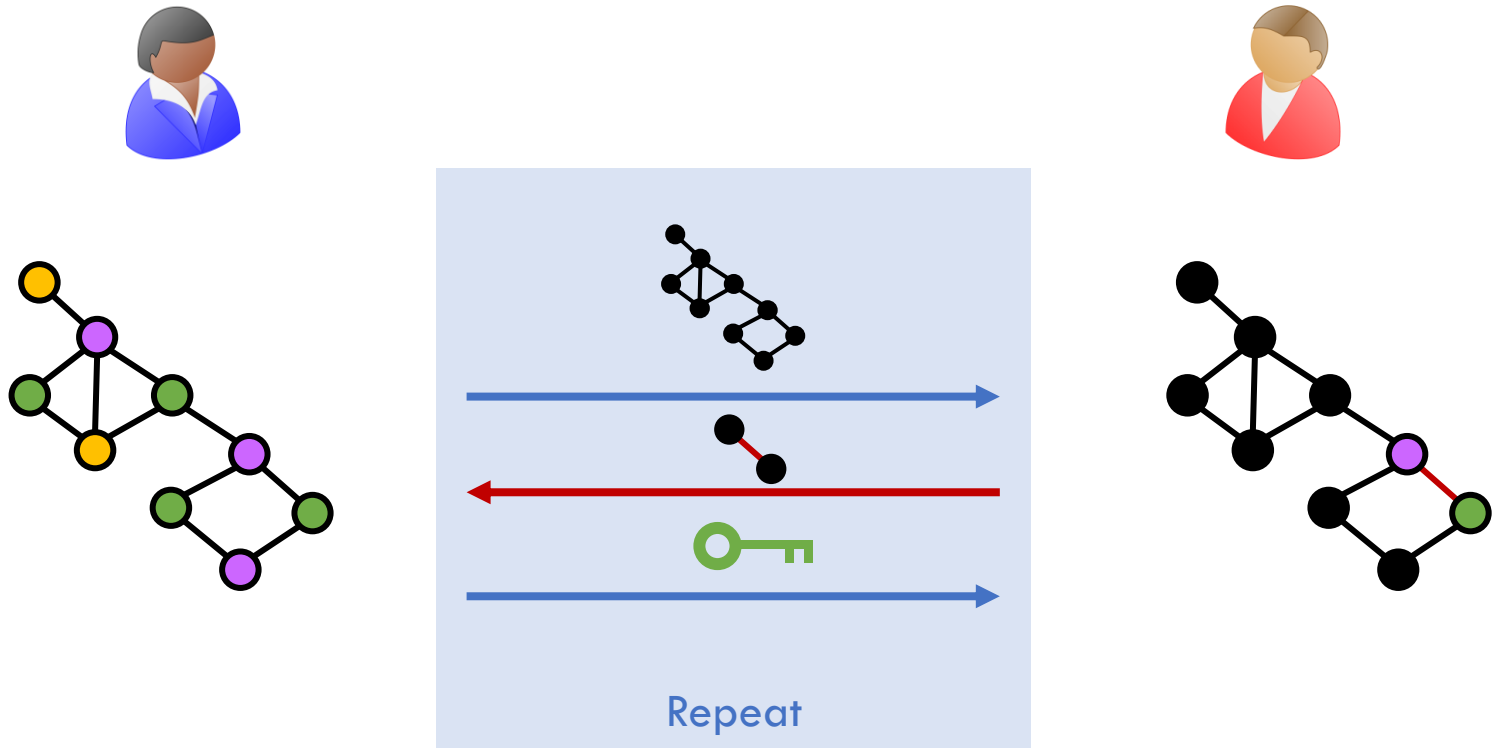


Bob checks if the colors are different.

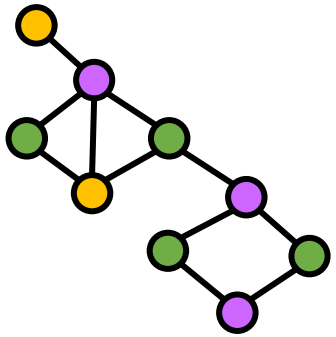
Interactive Zero-Knowledge Proofs

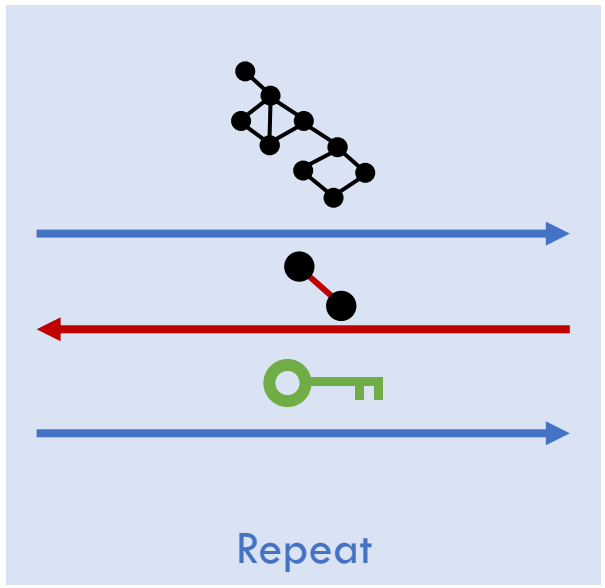


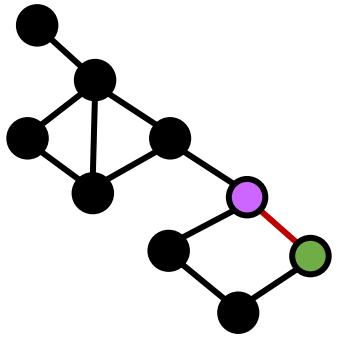
Interactive Zero-Knowledge Proofs



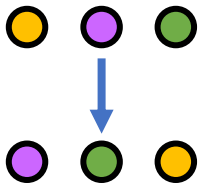
Interactive Zero-Knowledge Proofs



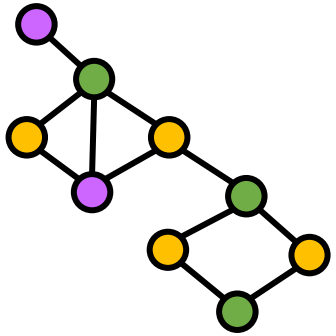


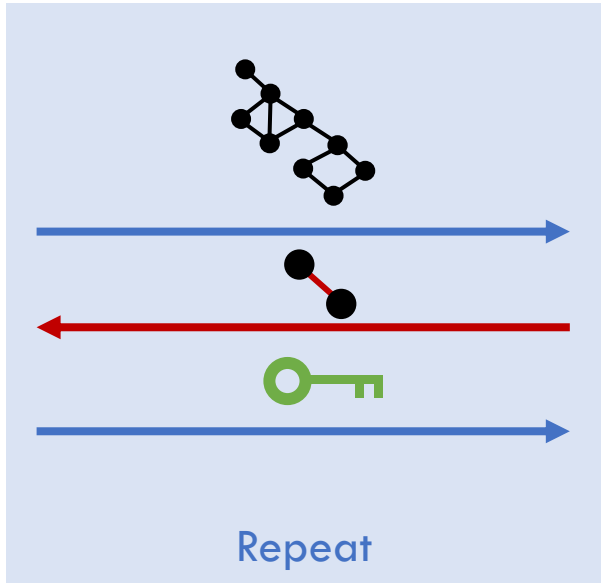


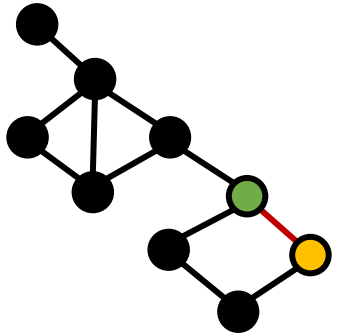
For each repetition randomly permute the colors



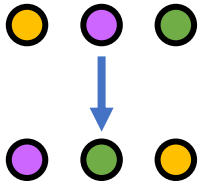
Interactive Zero-Knowledge Proofs



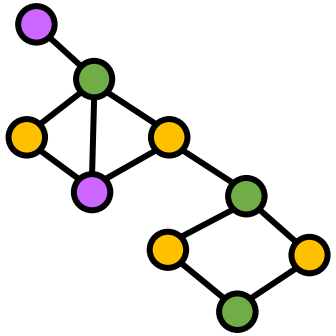


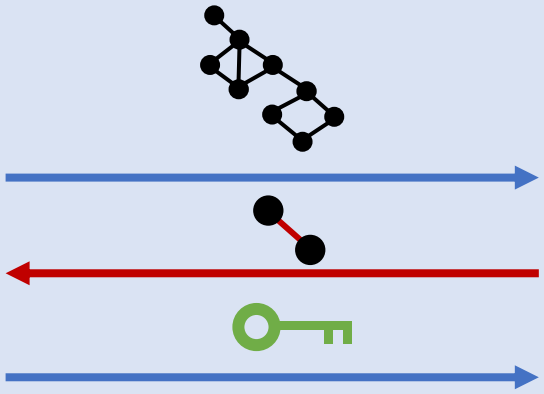


For each repetition randomly permute the colors

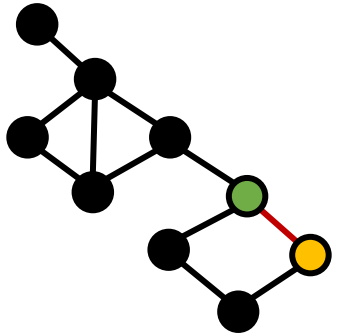


Interactive Zero-Knowledge Proofs

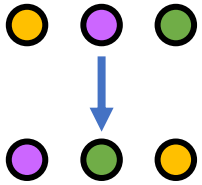




Repeat

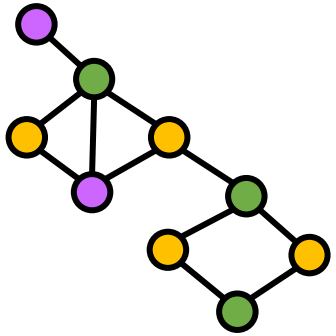


For each repetition randomly permute the colors

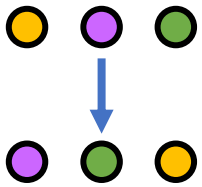


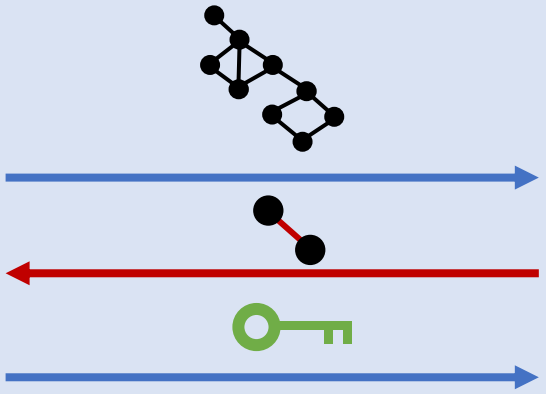
In each repetition,  sees two (independently) random colors for chosen edge.

Interactive Zero-Knowledge Proofs



For each repetition randomly permute the colors

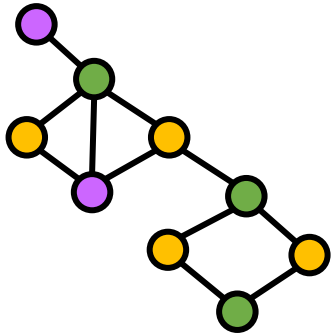




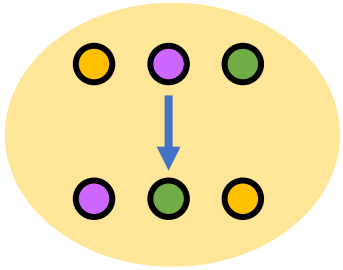
Repeat

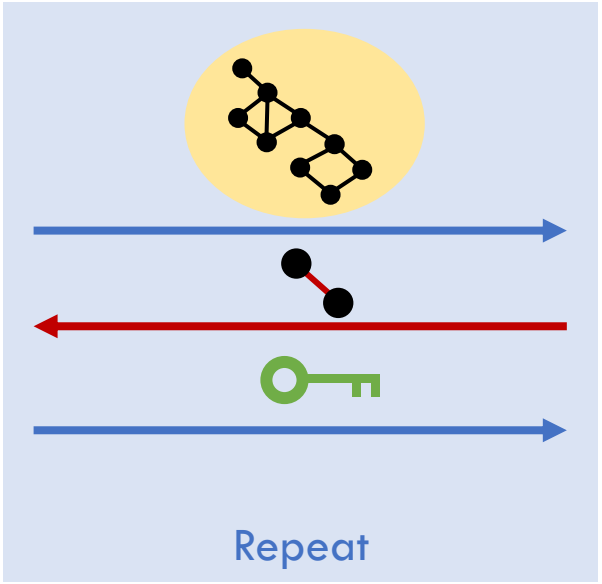
Prover requires randomness to "hide" the coloring.

Interactive Zero-Knowledge Proofs



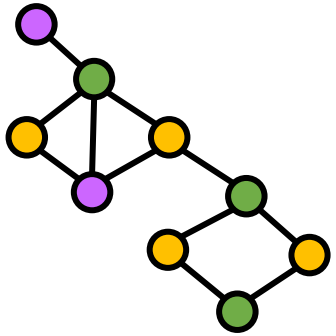
For each repetition randomly permute the colors



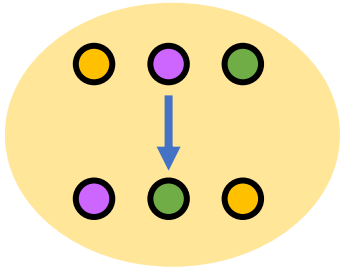


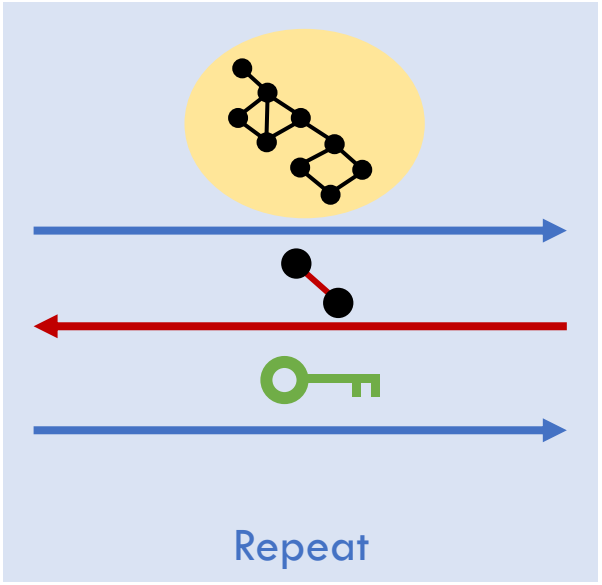
Prover requires randomness to "hide" the coloring.

Interactive Zero-Knowledge Proofs



For each repetition randomly permute the colors



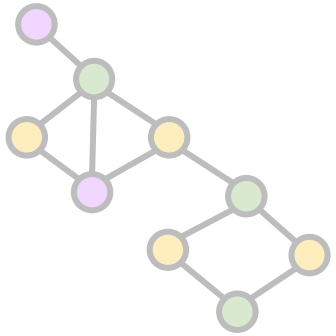


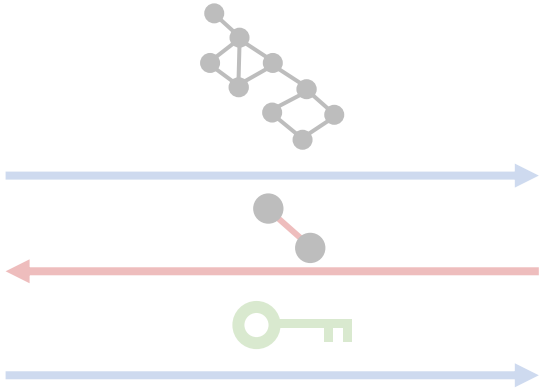
Prover requires randomness to "hide" the coloring.

Randomness is an expensive resource – we want to minimize its usage.

Interactive Zero-Knowledge Proofs

For each repetition randomly permute the colors





Prover requires randomness to "hide" the coloring.

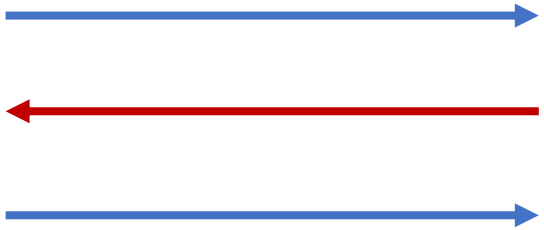
Randomness is an expensive resource – we want to minimize its usage.

Can we construct zero knowledge proofs where the prover doesn't need any randomness?

Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness



Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness

Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness

Turing Machine

Deterministic Prover Zero-Knowledge Proofs

input

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness

Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

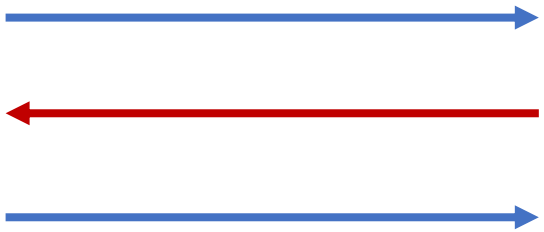
Zero-knowledge requires the prover to have access to randomness

advice string captures prior knowledge of the world.

Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness

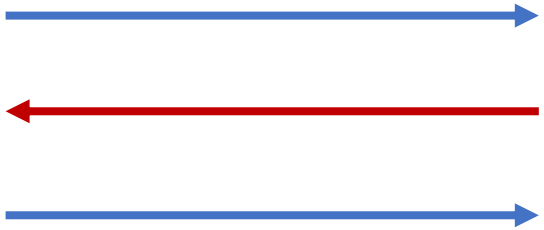


advice string captures prior knowledge of the world.

Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness when has an unbounded advice string.



advice string captures prior knowledge of the world.

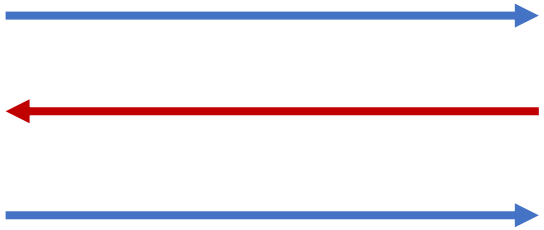
Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge requires the prover to have access to randomness when has an unbounded advice string.

Theorem

For every bound on  's advice string, we construct a deterministic prover zero-knowledge protocol.

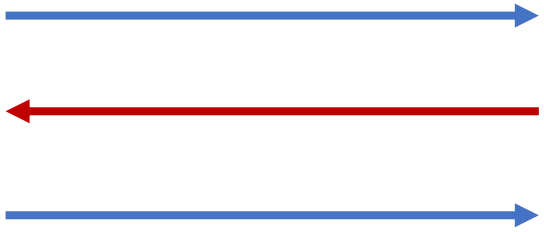


advice string captures prior knowledge of the world.

Deterministic Prover Zero-Knowledge Proofs

[Goldreich-Oren'94]

Zero-knowledge with deterministic  impossible.



Focus of this work

Focus of this work

Focus of this work

Interactive Zero-Knowledge Proofs

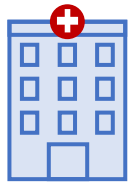
Is prover randomness essential for zero-knowledge?

Secure Computation

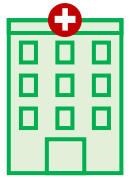
Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]

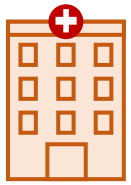
x_1



x_2



x_3

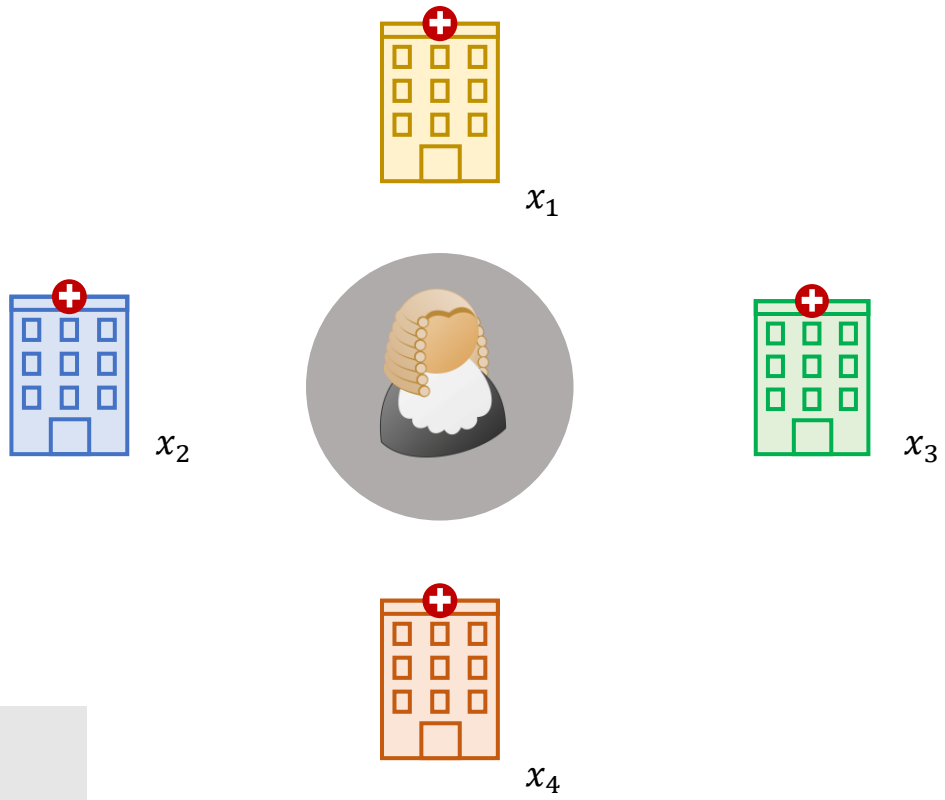


x_4

$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

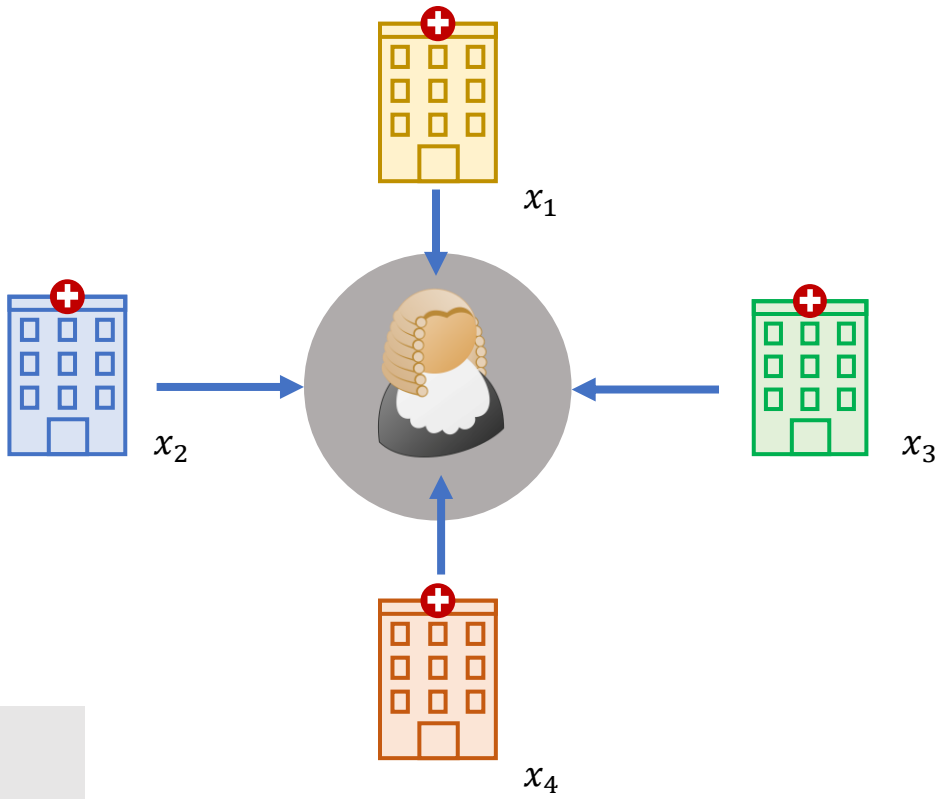
[Yao'86, Goldreich-Micali-Wigderson'87]



$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

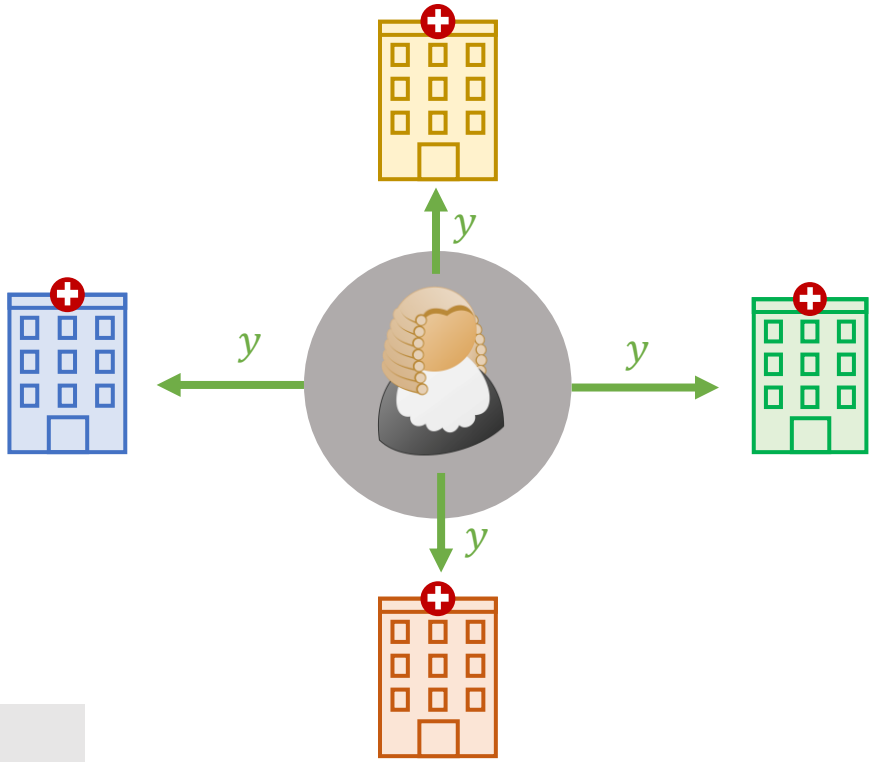
[Yao'86, Goldreich-Micali-Wigderson'87]



$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

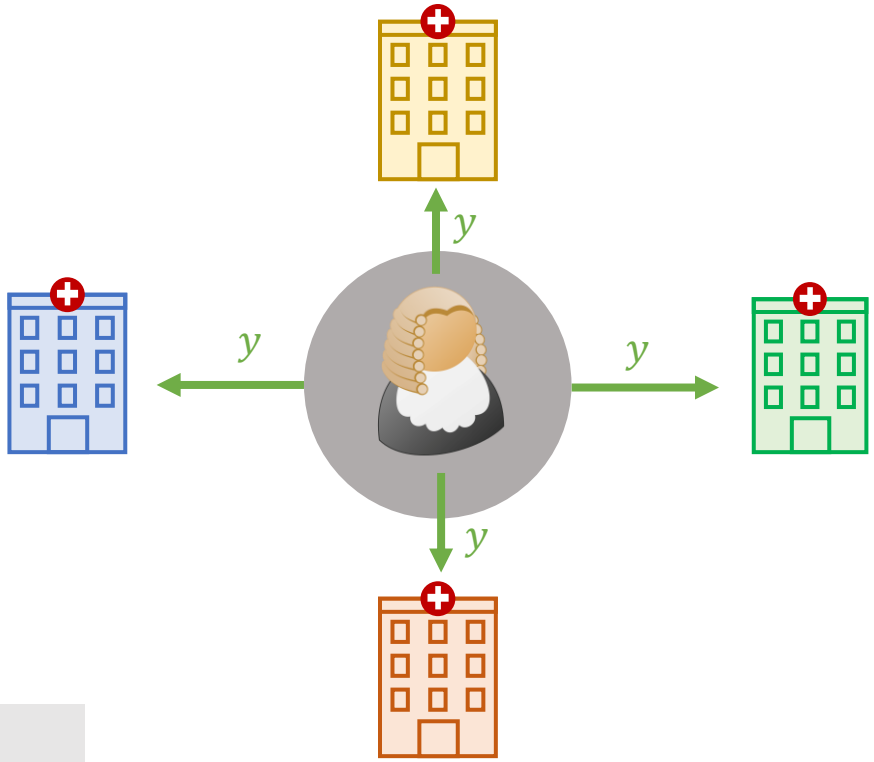
[Yao'86, Goldreich-Micali-Wigderson'87]



$y = f(x_1, x_2, x_3, x_4)$

Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]

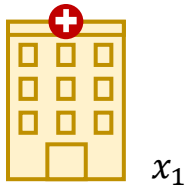


$y = f(x_1, x_2, x_3, x_4)$

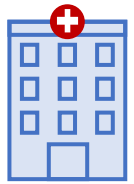
In Cryptography, the goal is to minimize trust assumptions.

Secure Computation

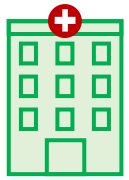
[Yao'86, Goldreich-Micali-Wigderson'87]



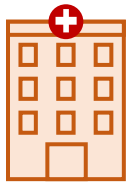
x_1



x_2



x_3



x_4

$y = f(x_1, x_2, x_3, x_4)$

In Cryptography, the goal is to minimize trust assumptions.

Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]

x_1

x_2

x_3

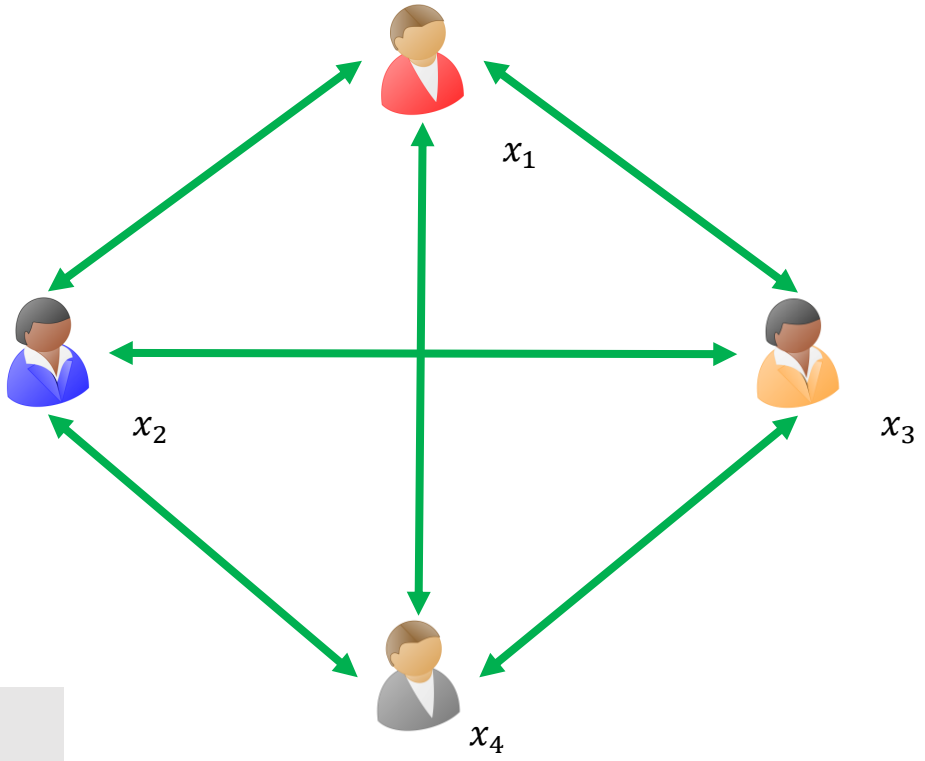
x_4

$y = f(x_1, x_2, x_3, x_4)$

In Cryptography, the goal is to minimize trust assumptions.

Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]



Run a protocol by exchanging messages.

$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]

x_1

x_2

x_3

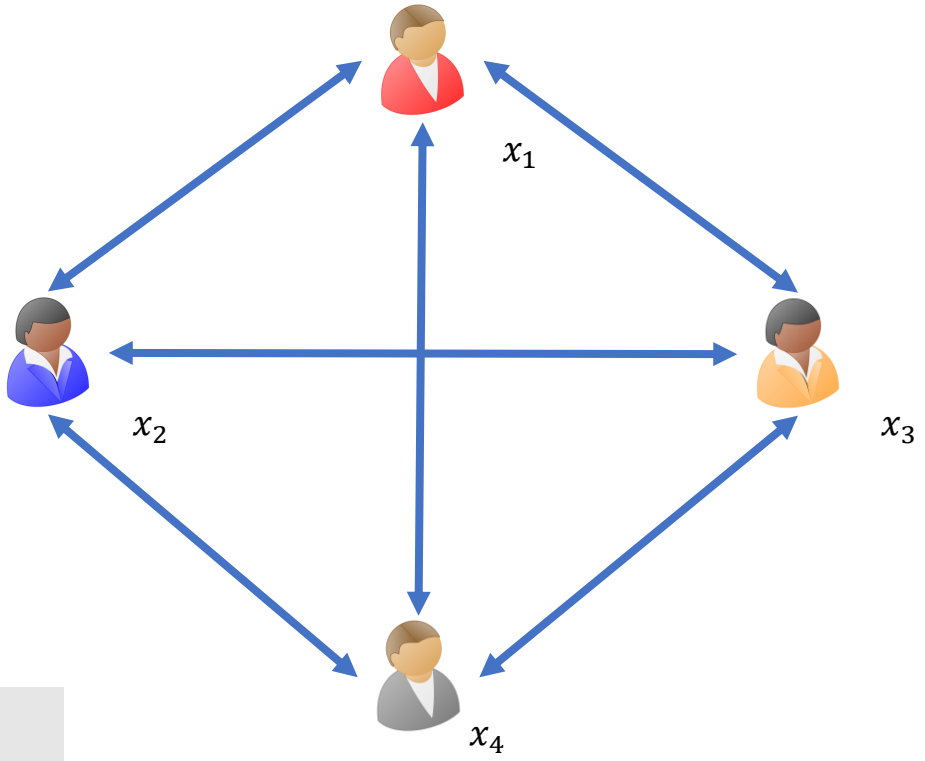
x_4

Run a protocol by exchanging messages.

$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]

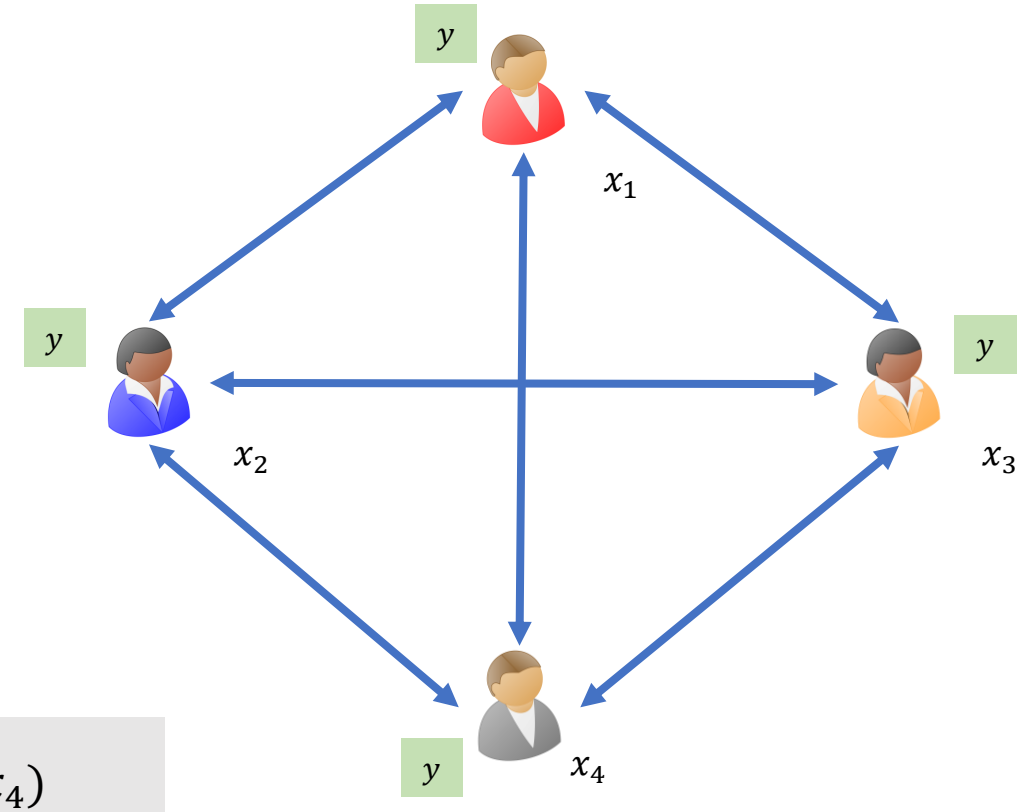


Run a protocol by exchanging messages.

$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

[Yao'86, Goldreich-Micali-Wigderson'87]

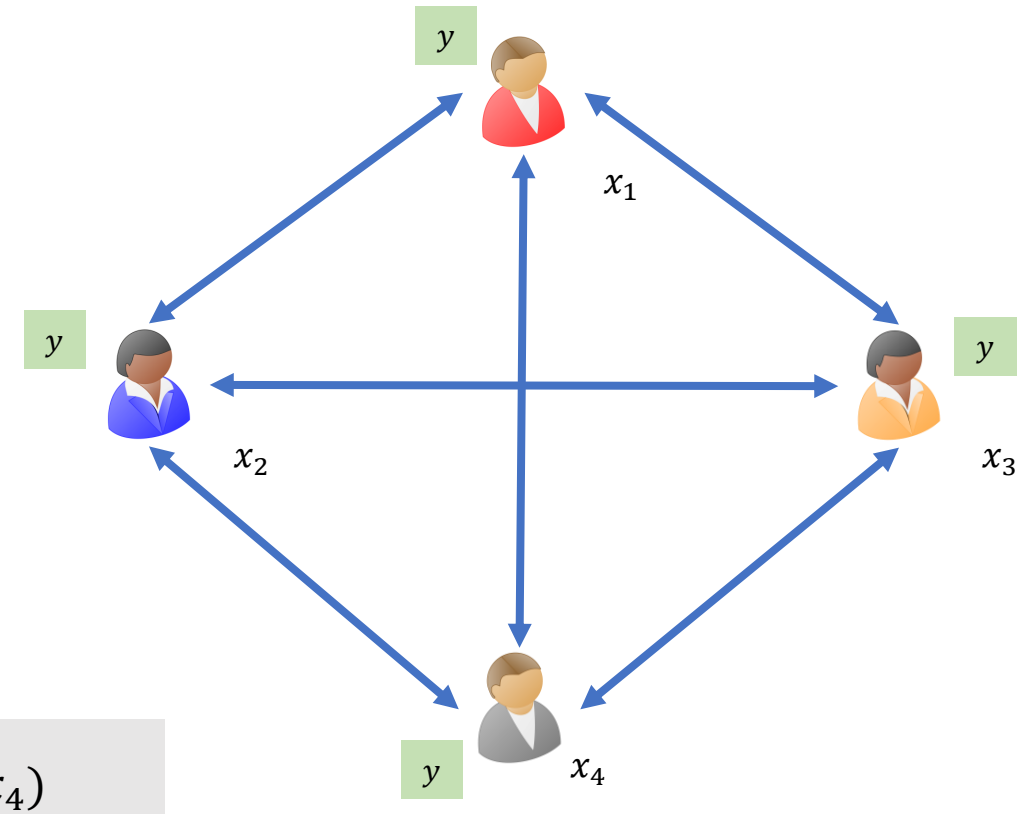


Run a protocol by exchanging messages.

$$y = f(x_1, x_2, x_3, x_4)$$

Secure Computation

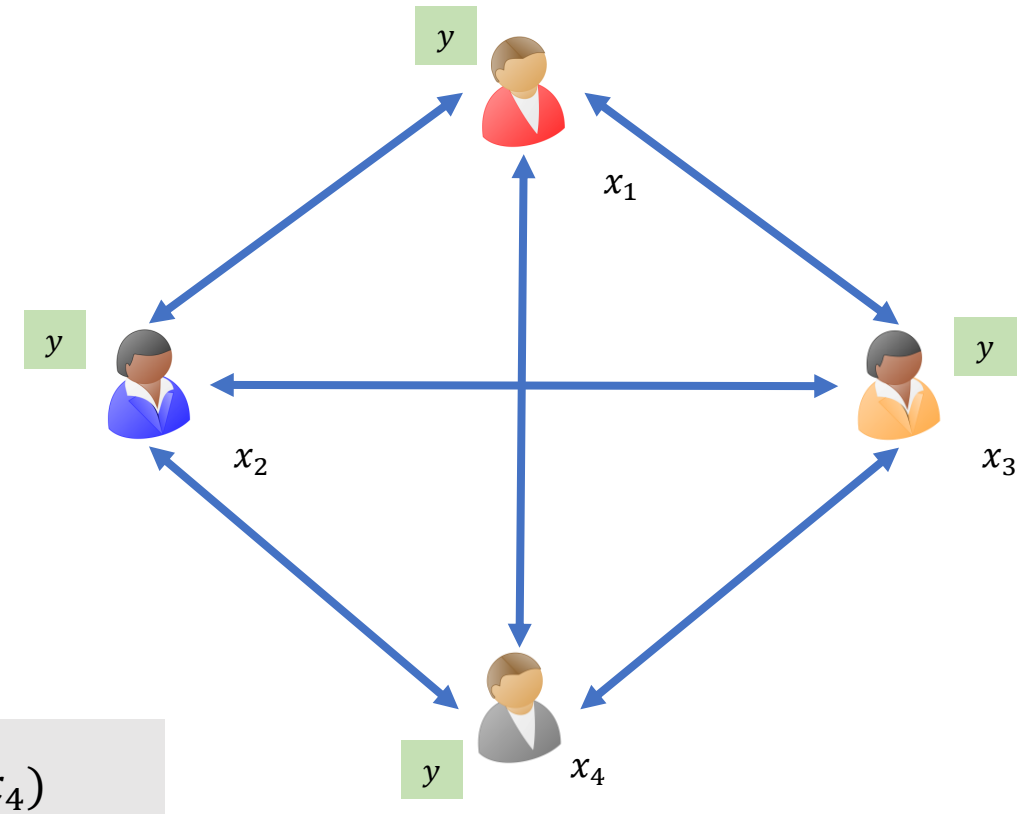
[Yao'86, Goldreich-Micali-Wigderson'87]



$y = f(x_1, x_2, x_3, x_4)$

Misbehaving participants should not learn anything beyond the output of the function.

Secure Computation Interaction



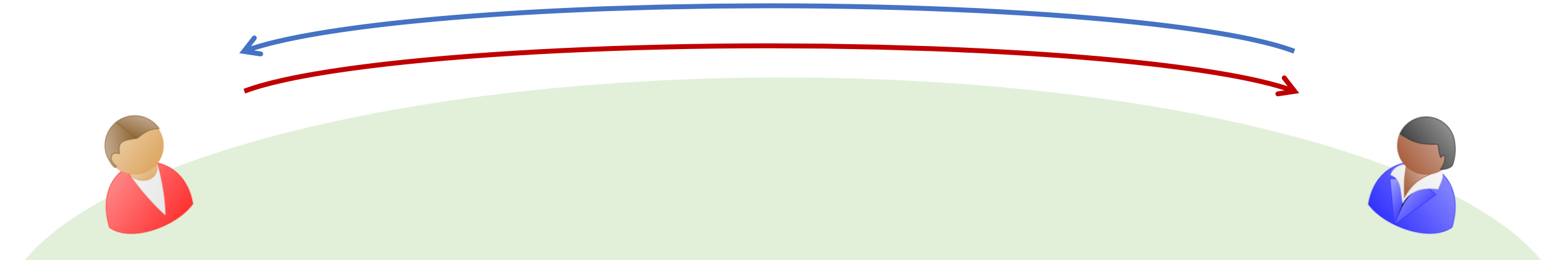
$y = f(x_1, x_2, x_3, x_4)$

Misbehaving participants should not learn anything beyond the output of the function.

A round constitutes of every participant sending a message.

Network Latency

Network Latency

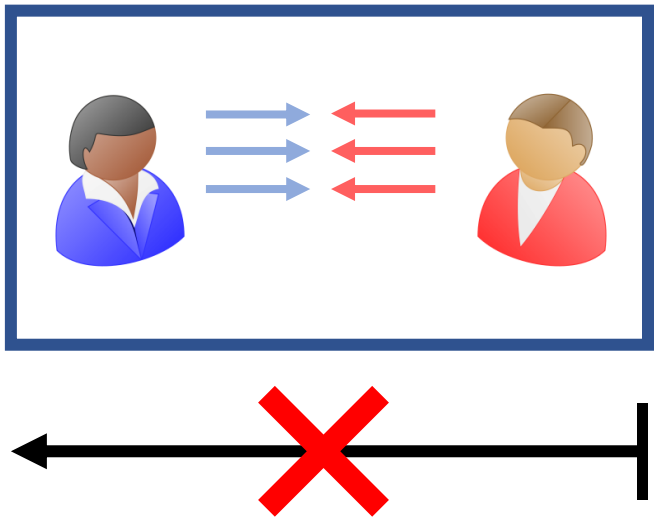


Network Latency

To minimize the effect of network latency, minimize the number of communication rounds.

Known bounds for interaction

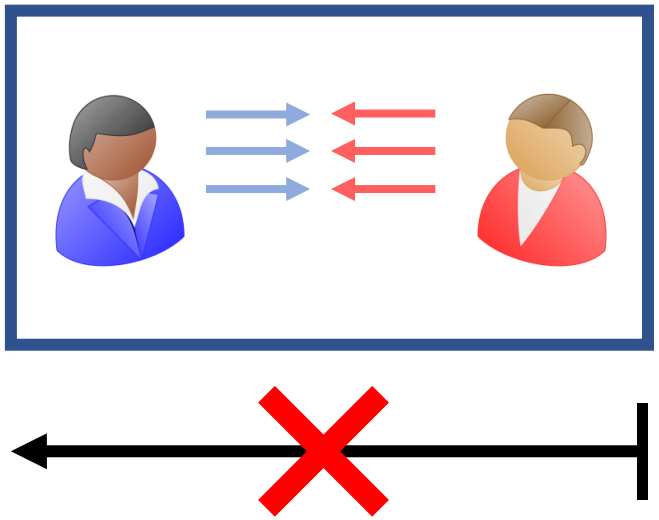
[Garg-Mukherjee-Pandey-Polychroniadou'16]



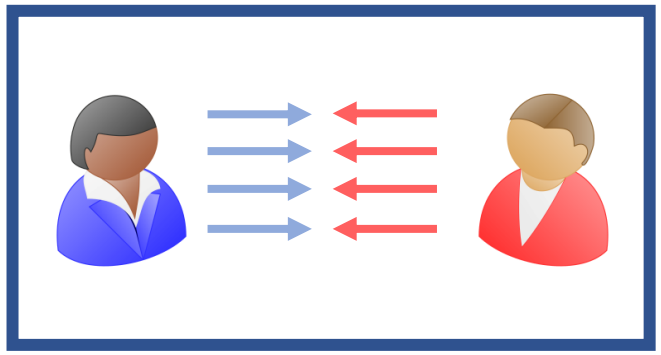
Impossible

Known bounds for interaction

[Garg-Mukherjee-Pandey-Polychroniadou'16]



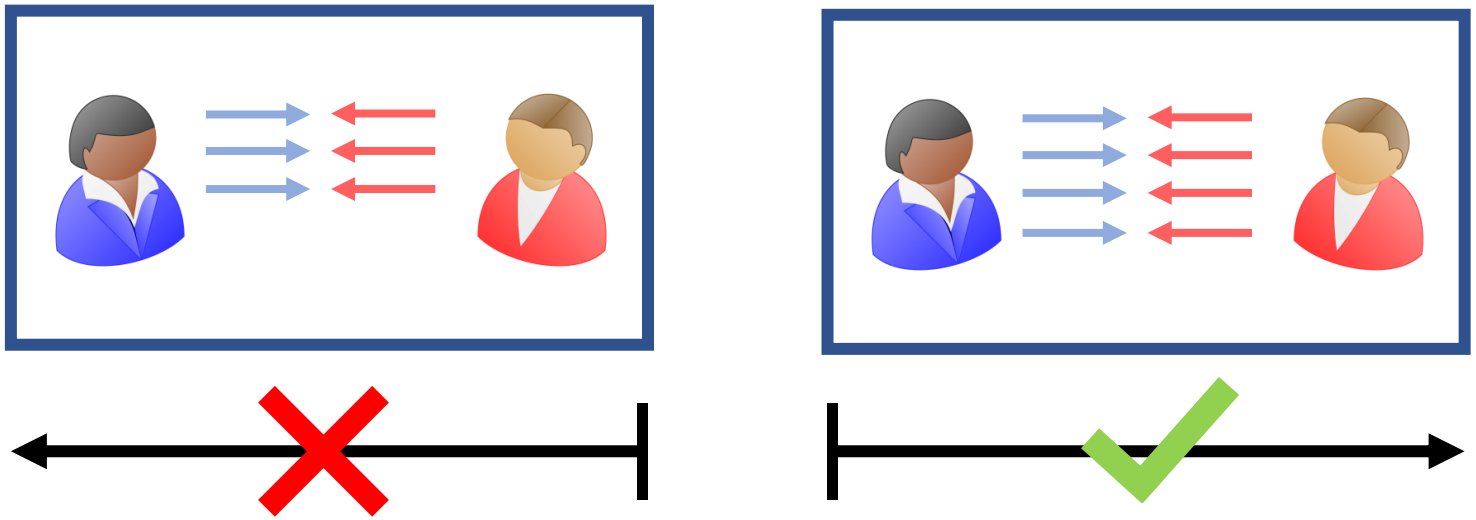
Impossible



?

Is the lower bound tight?

Round Optimal Protocol



Impossible

Theorem

There are four round protocols under optimal assumptions.

Focus of this work

Interactive Zero-Knowledge Proofs

Is prover randomness essential for zero-knowledge?

Secure Computation

Focus of this work

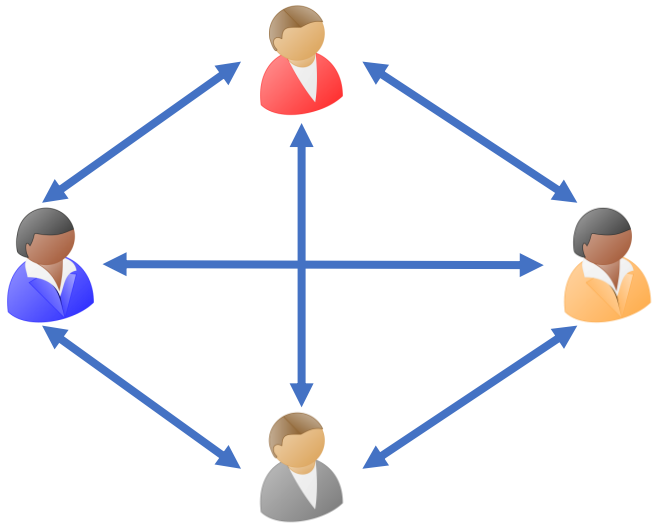
Interactive Zero-Knowledge Proofs

Is prover randomness essential for zero-knowledge?

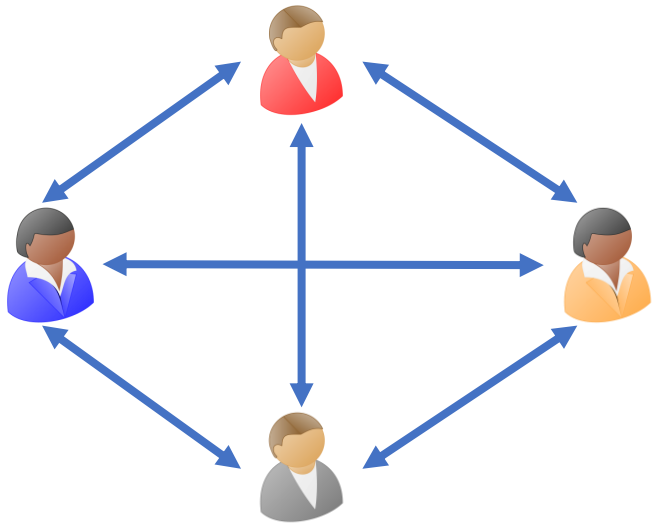
Secure Computation

Can we construct secure computation protocols in minimal rounds from minimal assumptions?

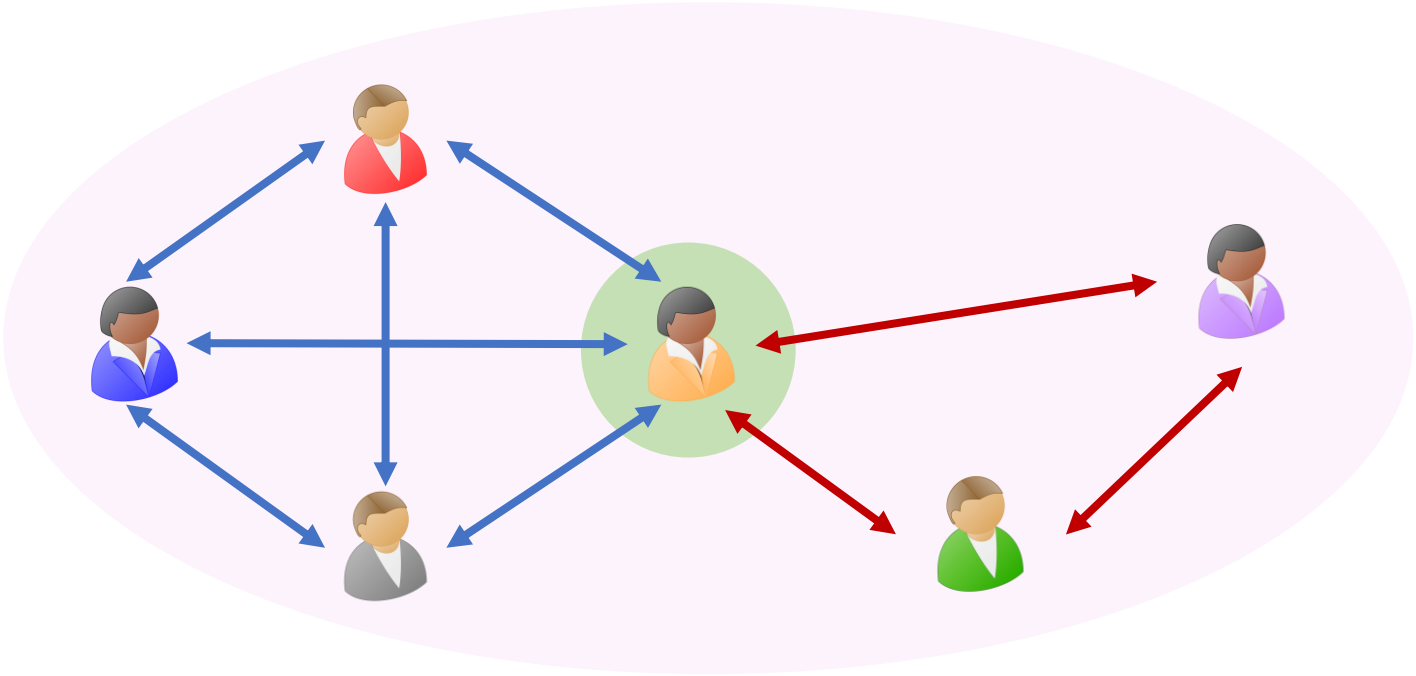
Protocols



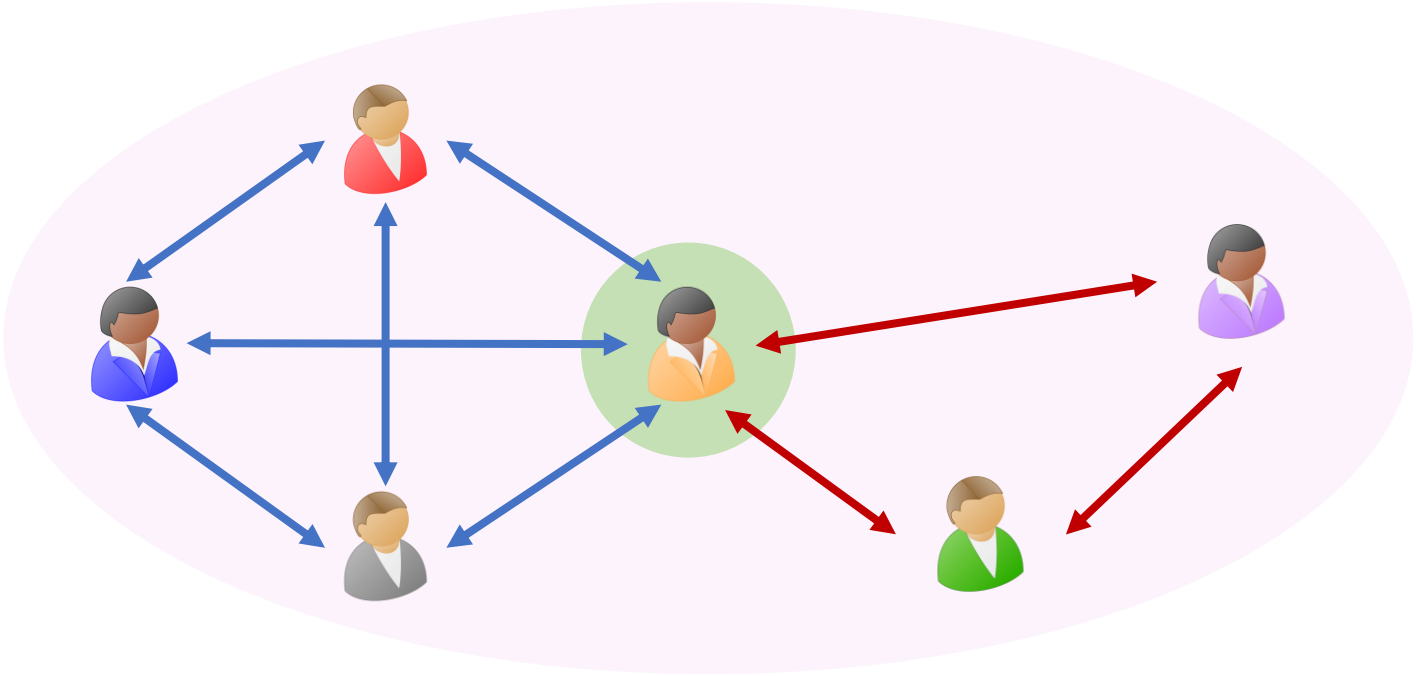
Protocols on the Internet



Protocols on the Internet

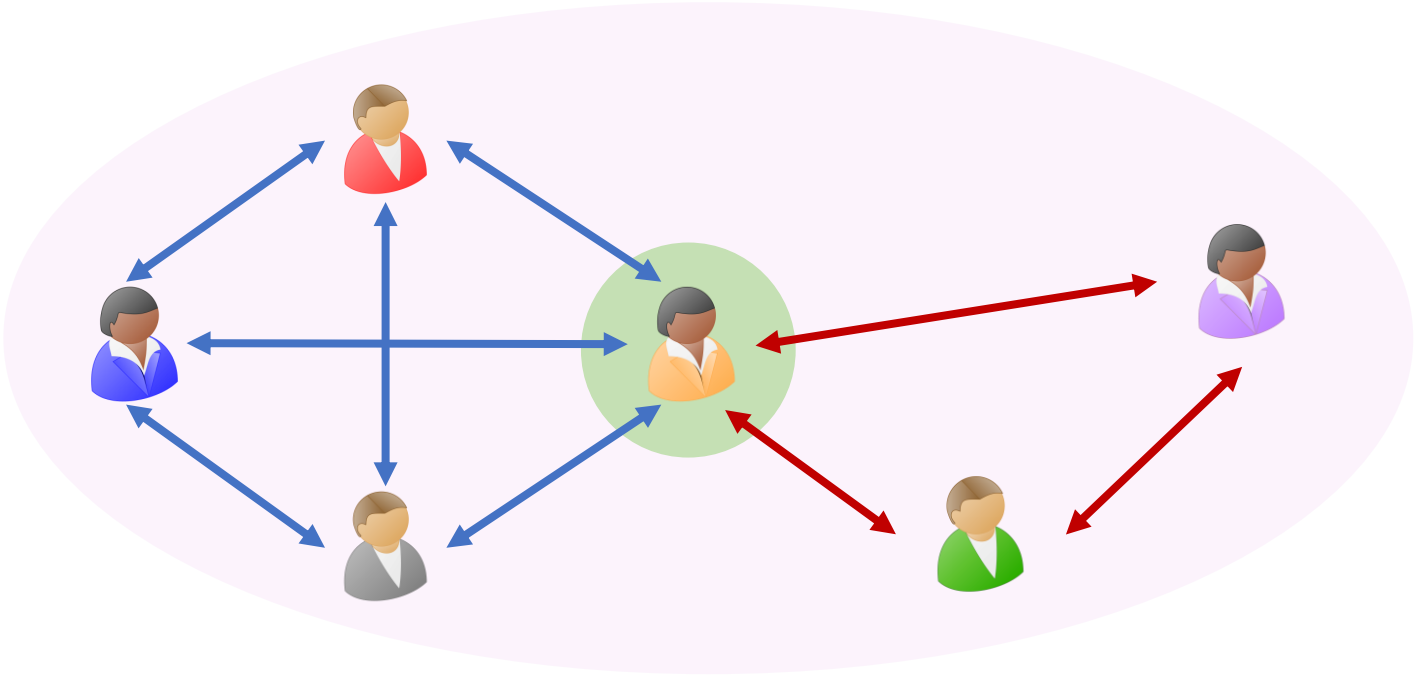


Protocols on the Internet



Existing protocols can no longer be proven secure when multiple concurrent copies are running.

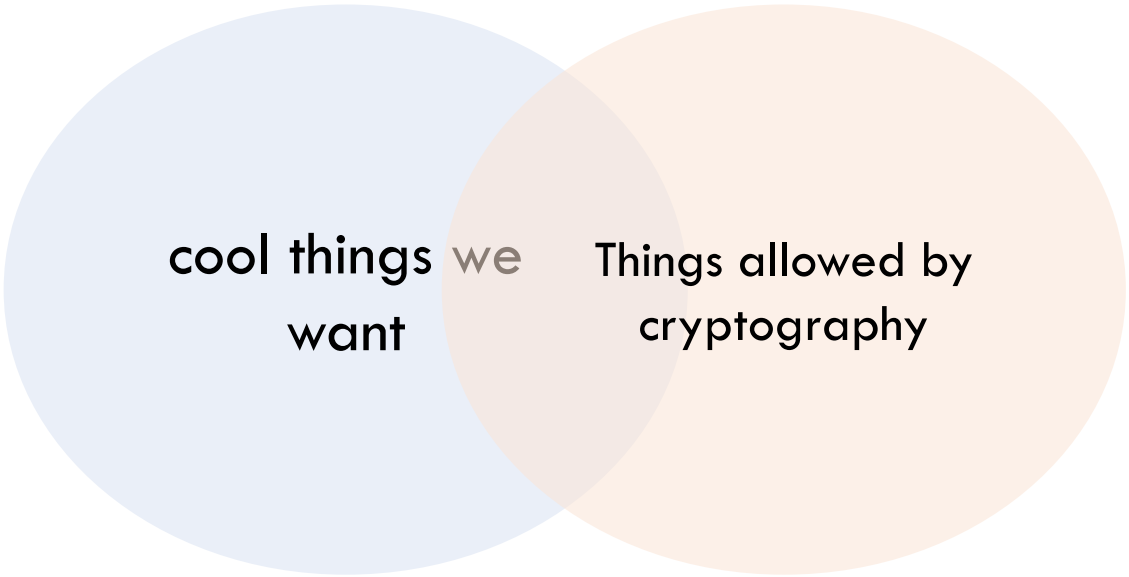
Protocols on the Internet



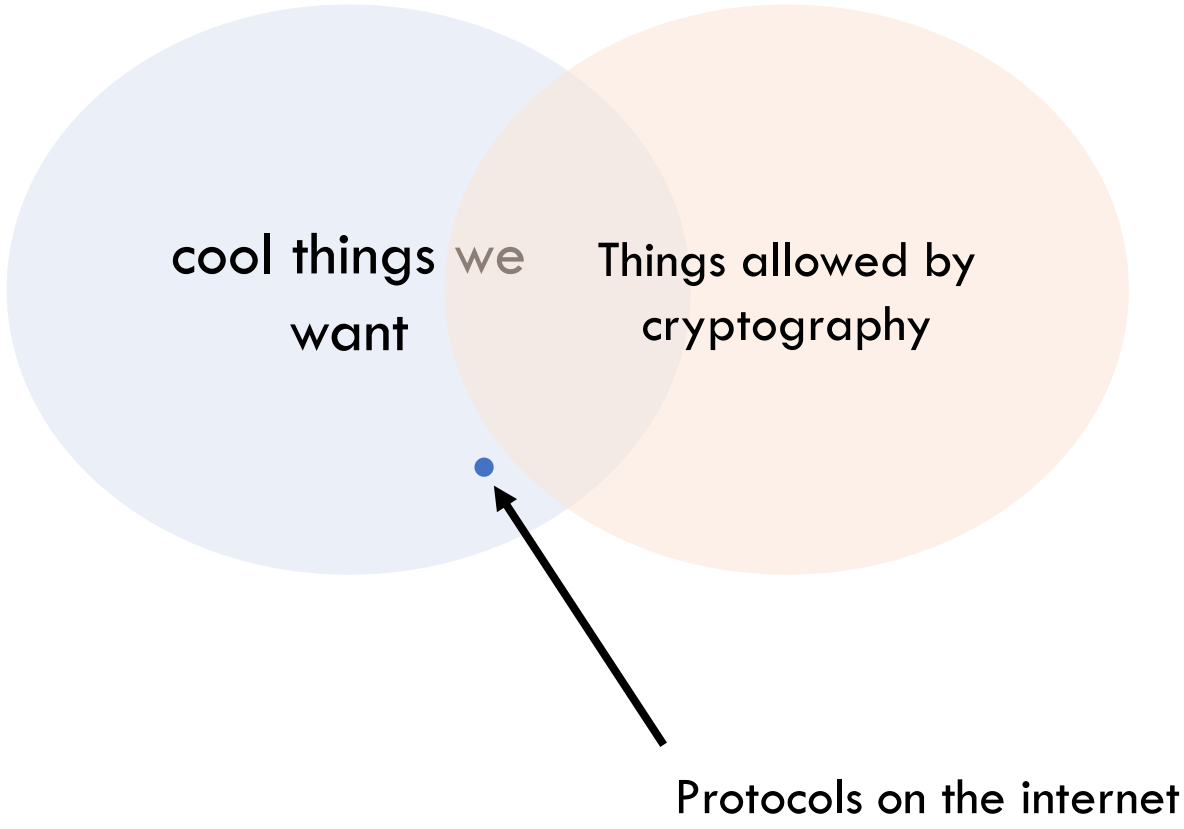
Existing protocols can no longer be proven secure when multiple concurrent copies are running.

In fact, impossible to construct secure protocols in this setting without trust assumption.

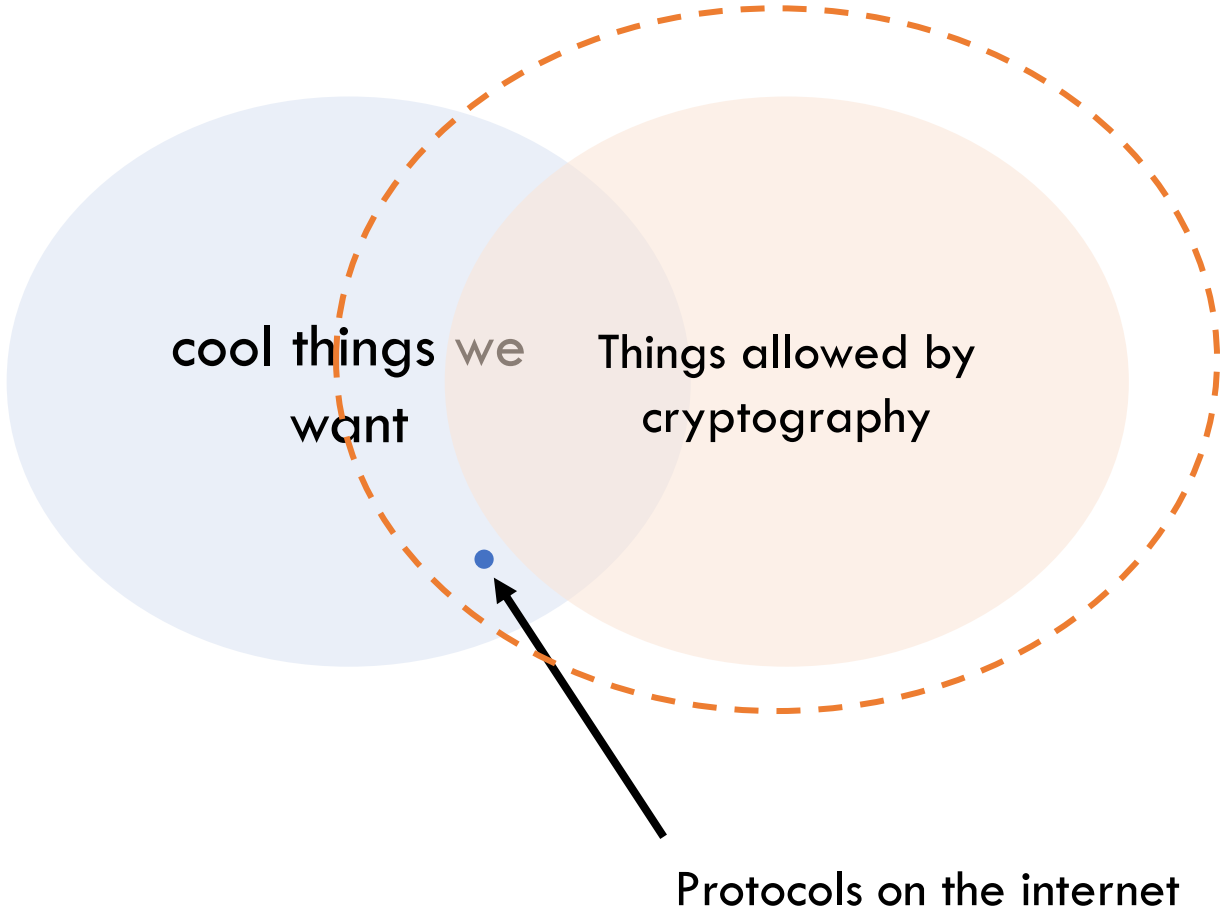
Circumventing Impossibilities



Circumventing Impossibilities



Circumventing Impossibilities

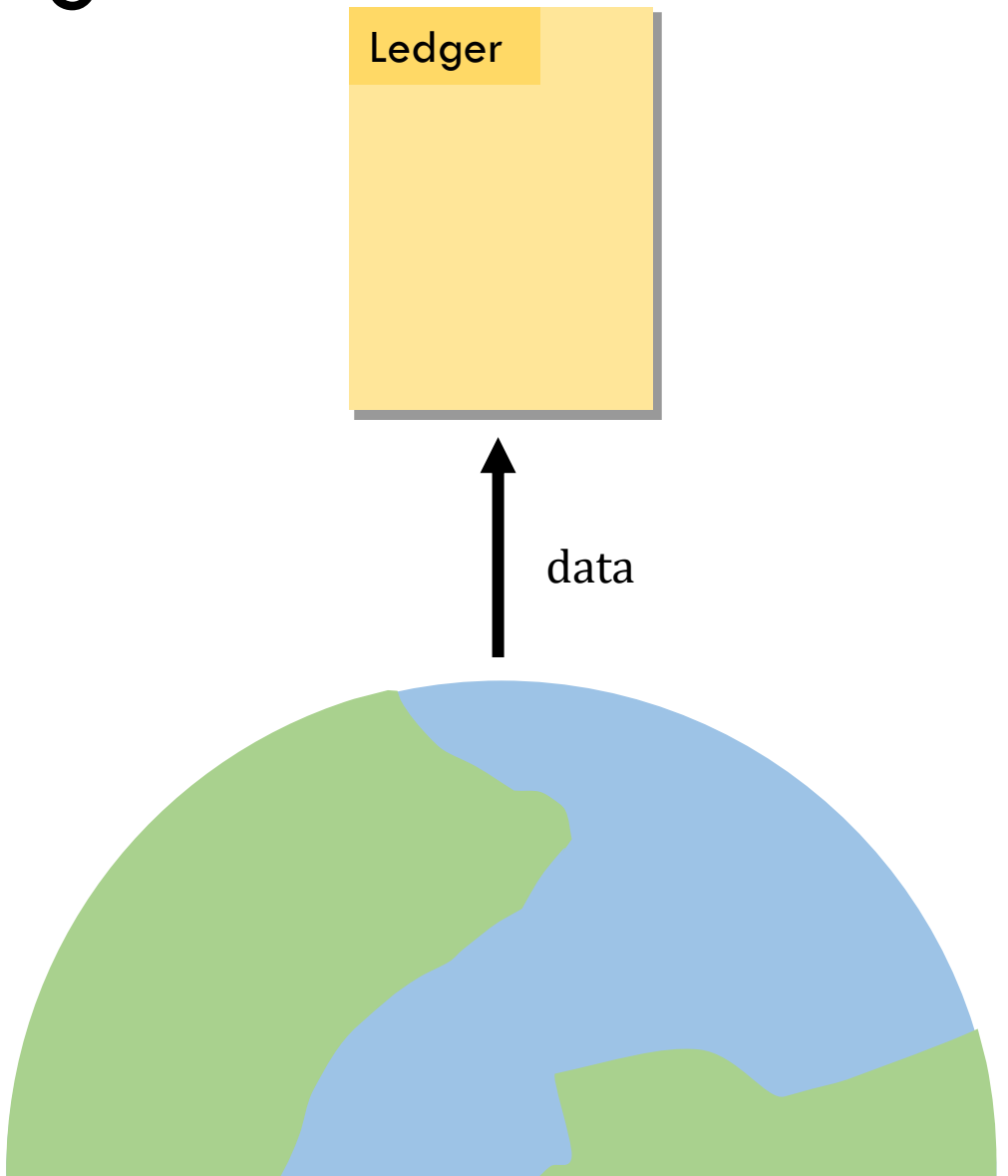


Relax trust assumptions.

Global Ledger

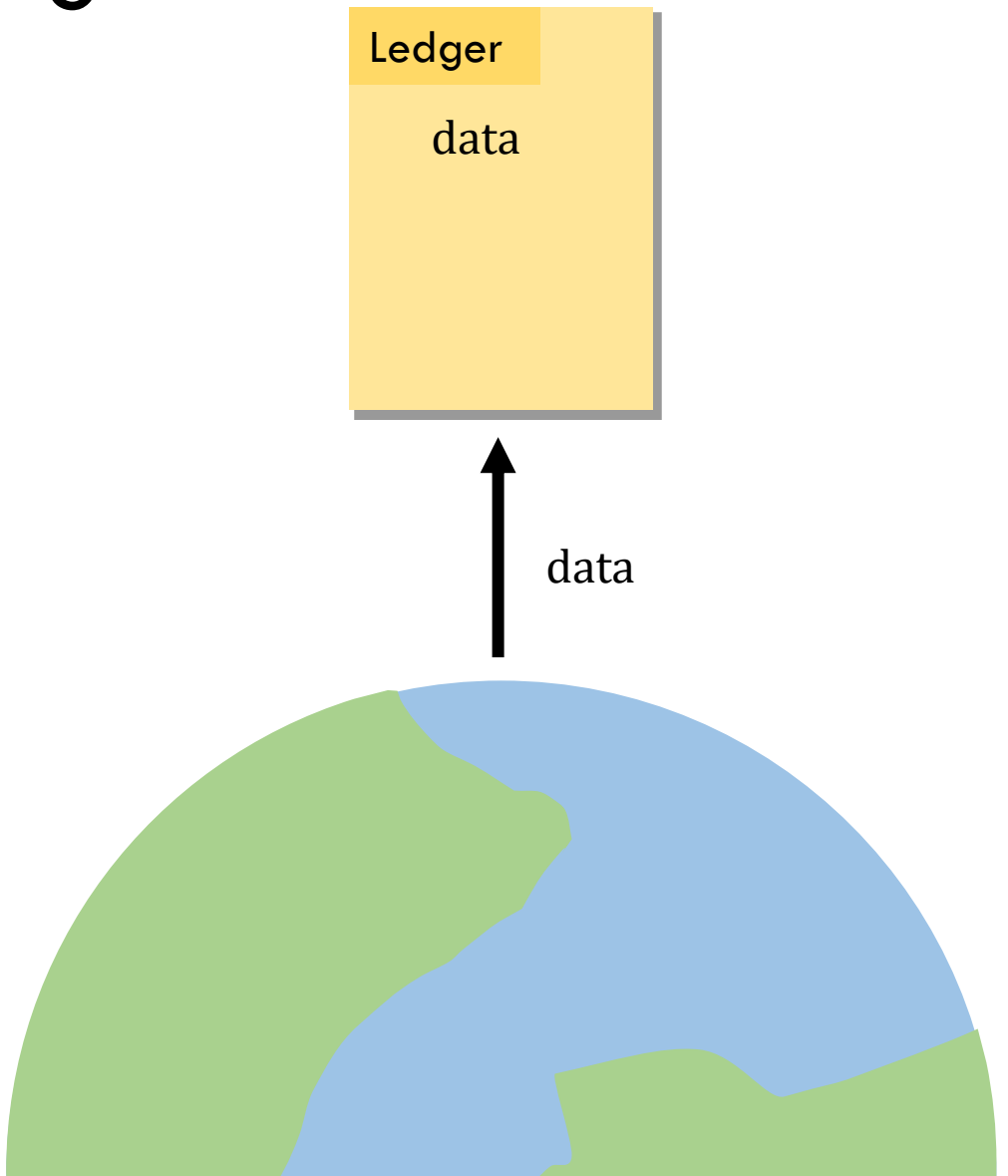
Ledger

Global Ledger



Anybody can post data to the ledger.

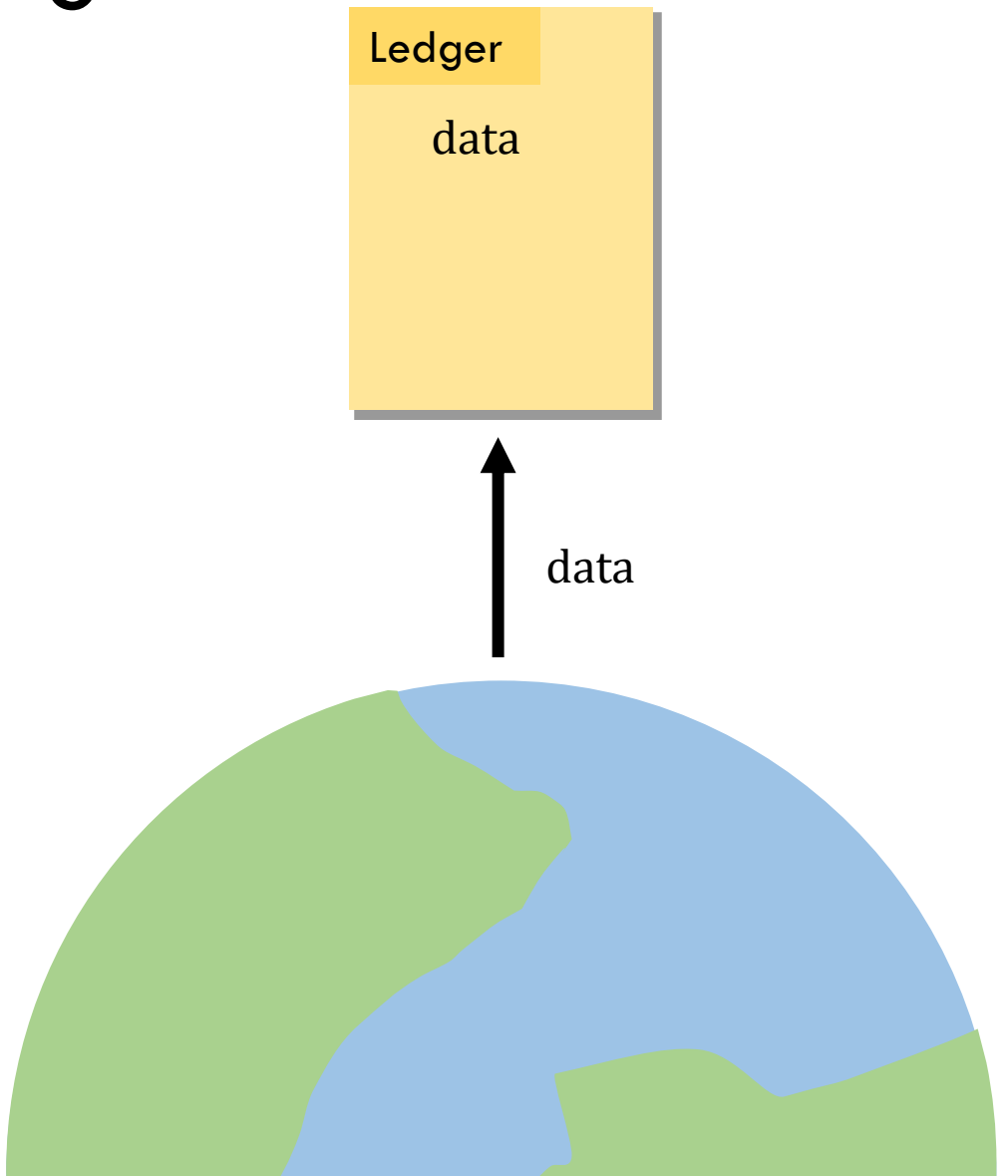
Global Ledger



Anybody can post data to the ledger.

Ledger publicly accessible.

Global Ledger



Anybody can post data to the ledger.

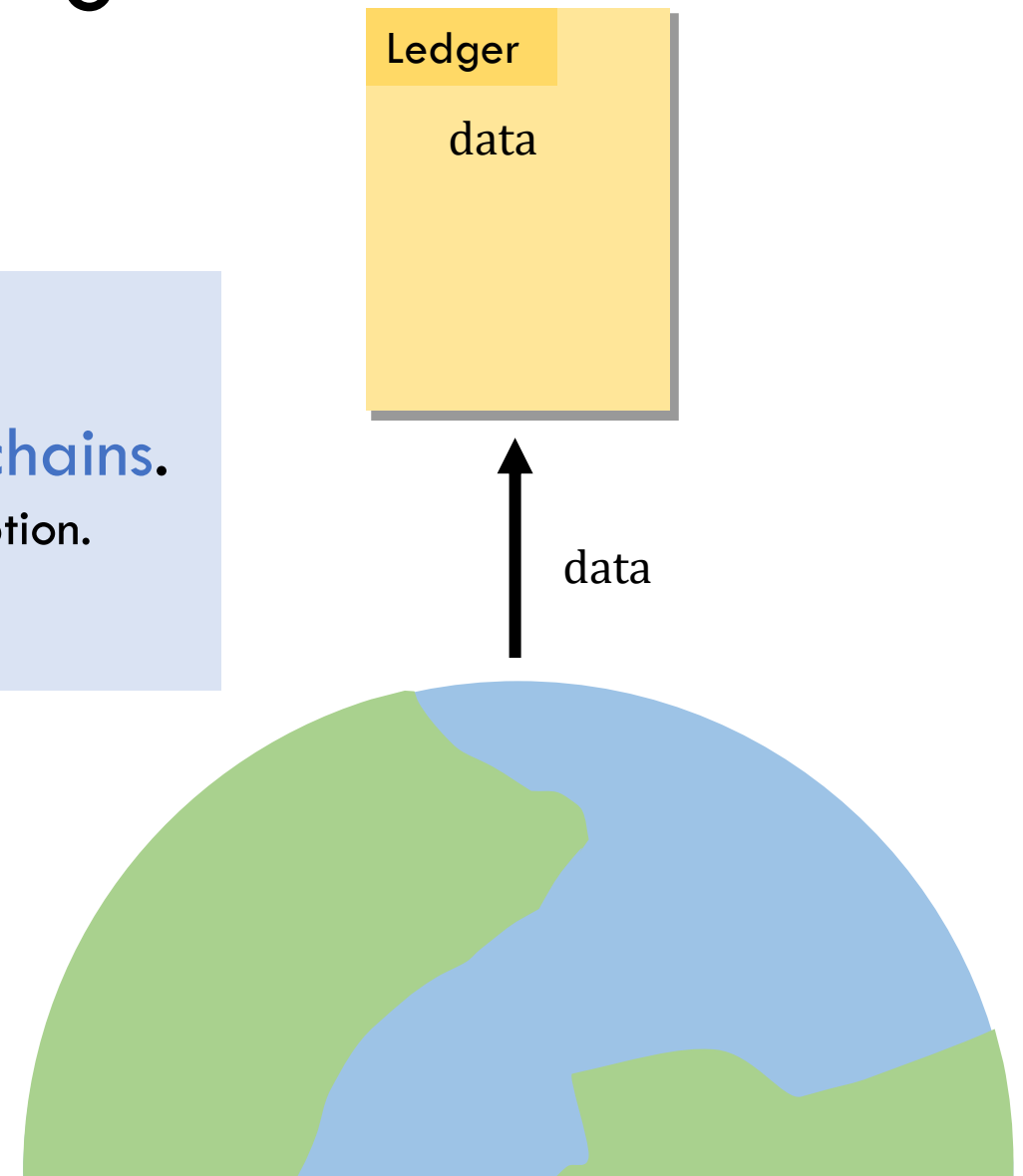
Ledger publicly accessible.

Posted data is permanent.

Global Ledger

Instantiated via Blockchains.

Decentralized trust assumption.



Anybody can post data to the ledger.

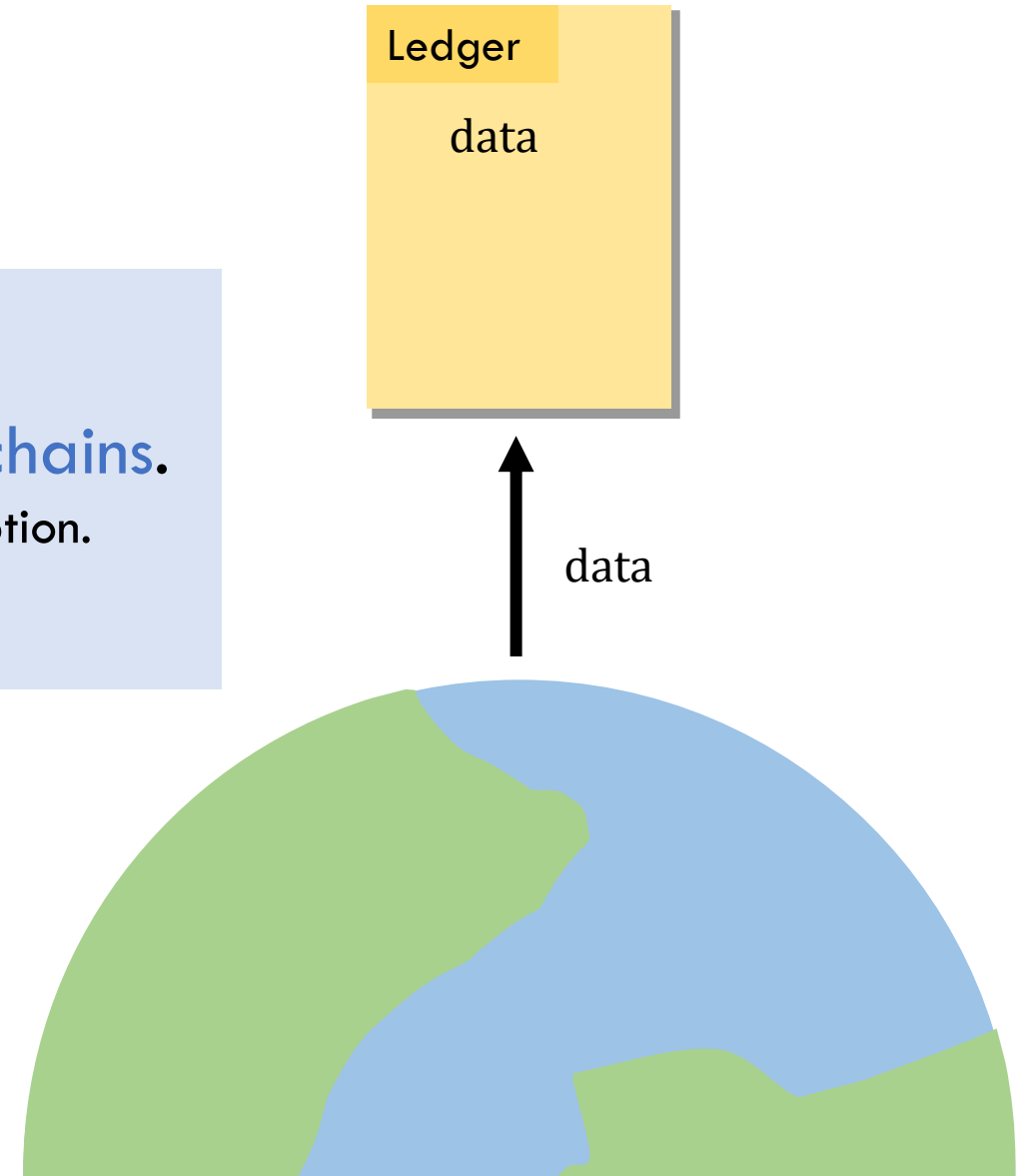
Ledger publicly accessible.

Posted data is permanent.

Blockchain

Instantiated via Blockchains.

Decentralized trust assumption.

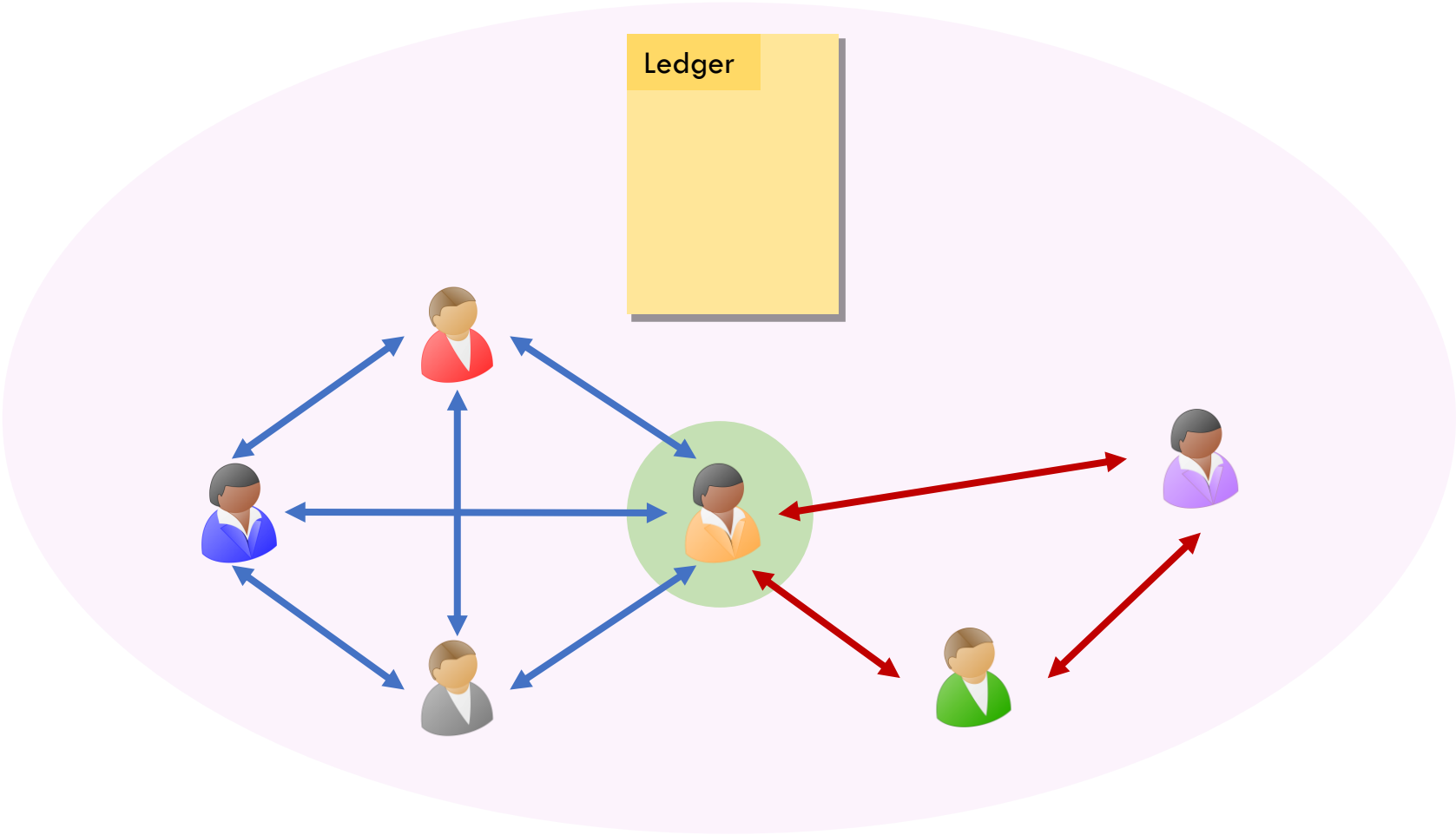


Anybody can post data to the ledger.

Ledger publicly accessible.

Posted data is permanent.

Protocols on the Internet – Blockchain model

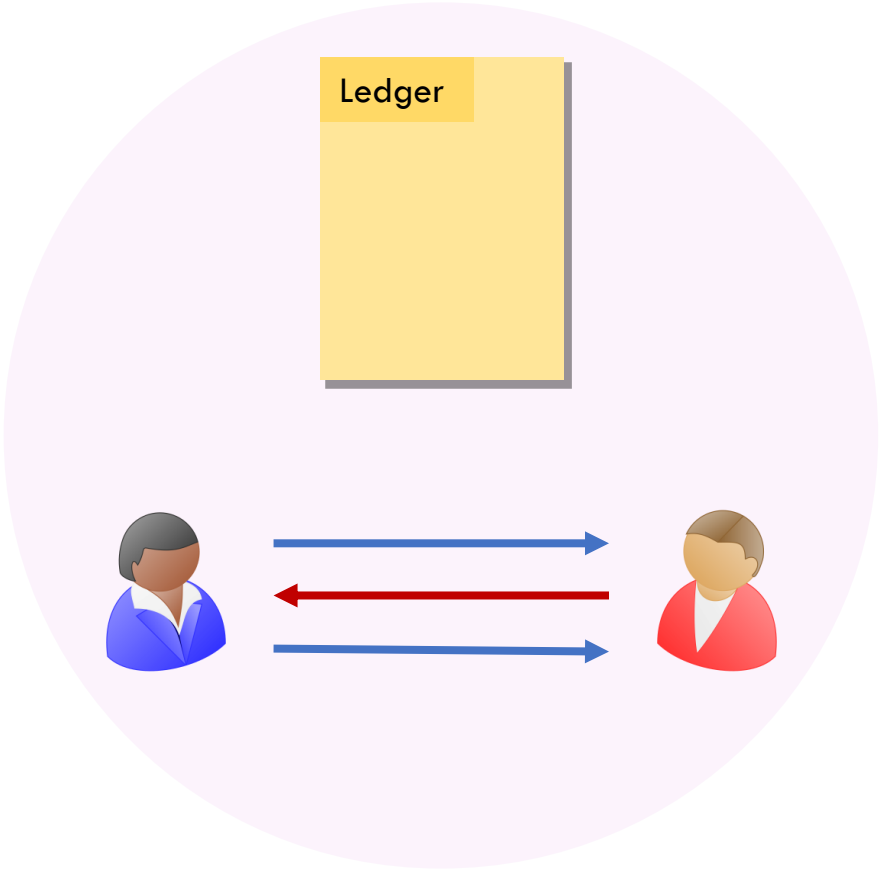


Each participant has access to the blockchain.

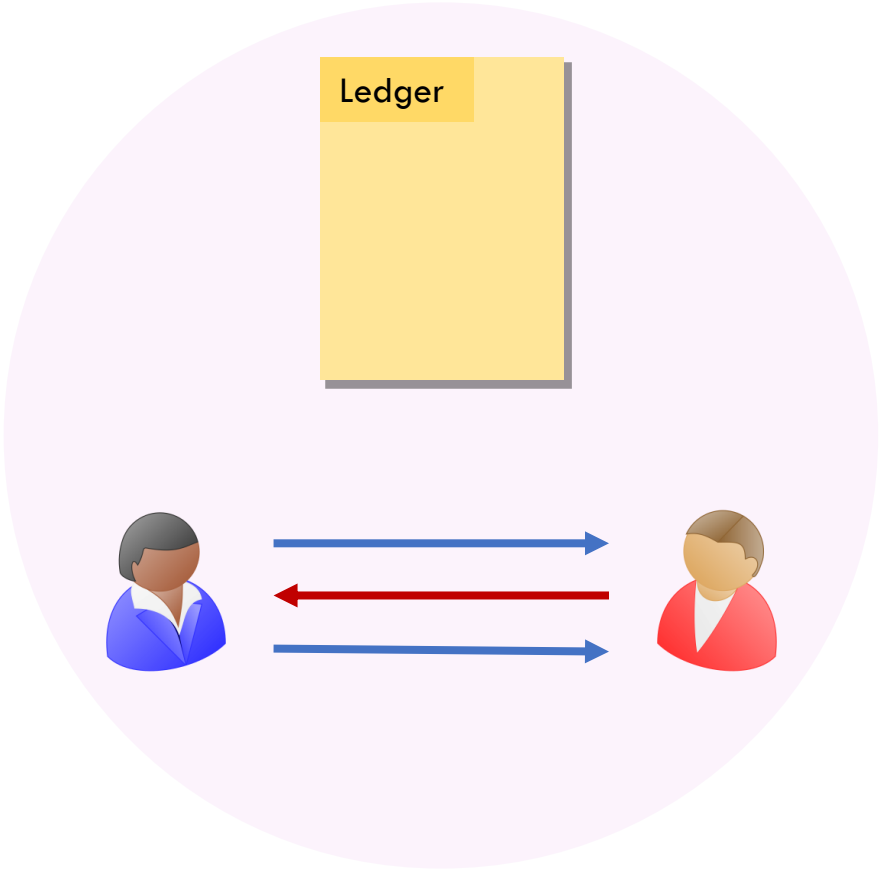
Theorem

We construct new protocols in the blockchain model that are secure when multiple concurrent instances are run.

Zero-Knowledge in the Blockchain Model



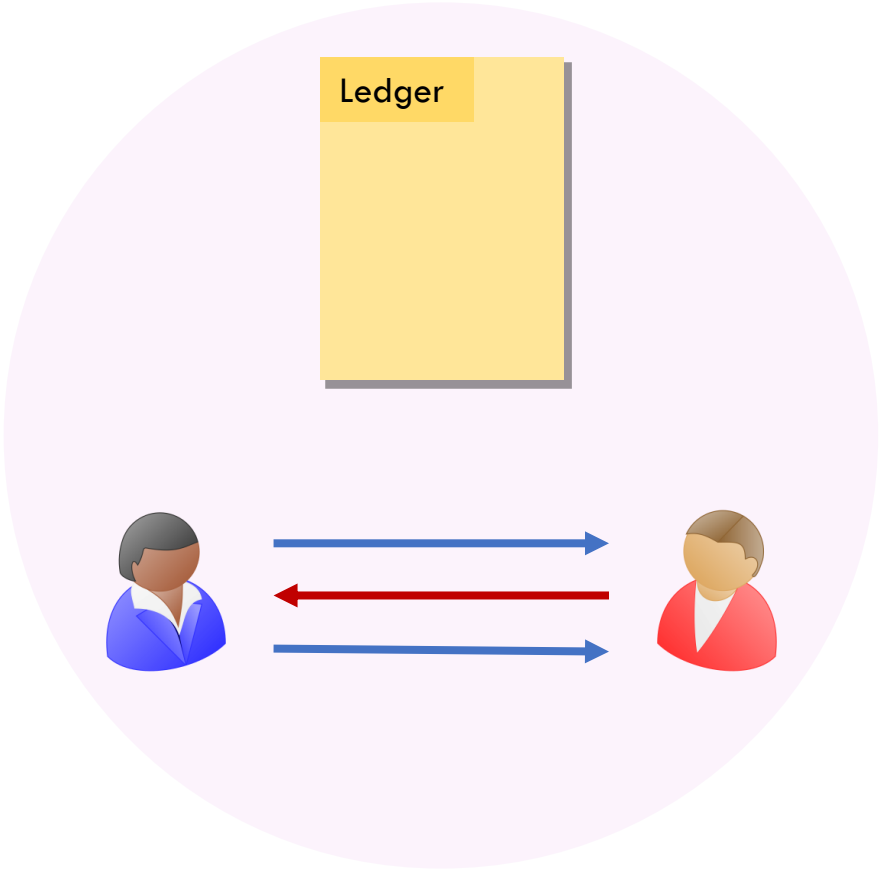
Zero-Knowledge in the Blockchain Model



Theorem

Proof techniques for existing protocols do not work in the blockchain model.

Zero-Knowledge in the Blockchain Model



Theorem

Proof techniques for existing protocols do not work in the blockchain model.

Theorem

We construct new zero-knowledge protocols in the blockchain model.

Focus of this work

Interactive Zero-Knowledge Proofs

Is prover randomness essential for zero-knowledge?

Secure Computation

Can we construct secure computation protocols in minimal rounds from minimal assumptions?

Focus of this work

Interactive Zero-Knowledge Proofs

Is prover randomness essential for zero-knowledge?

Secure Computation

Can we construct secure computation protocols in minimal rounds from minimal assumptions?

Can we make reasonable relaxations to the trust assumptions in order to circumvent barriers in secure computation?

Necessity of Randomness in Zero-knowledge

Founding Secure Computation on Blockchains

Round Optimal Secure Computation

Necessity of Randomness in Zero-knowledge

Founding Secure Computation on Blockchains

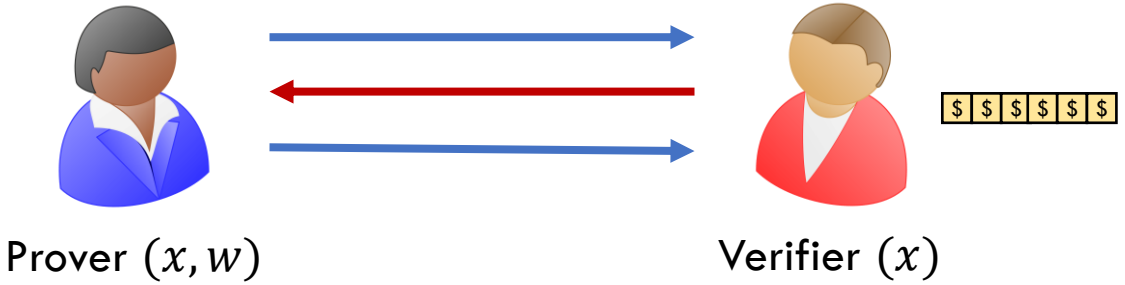
Round Optimal Secure Computation

Characterizing Deterministic Prover Zero-Knowledge

[Bitansky-C'20]

Zero Knowledge

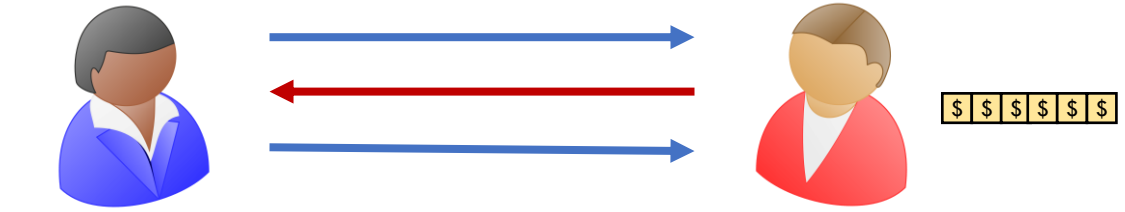
[Goldwasser-Micali-Rackoff'85]



wants to prove $x \in \mathcal{L}$ for some NP language $\mathcal{L}$.

Zero Knowledge

[Goldwasser-Micali-Rackoff'85]

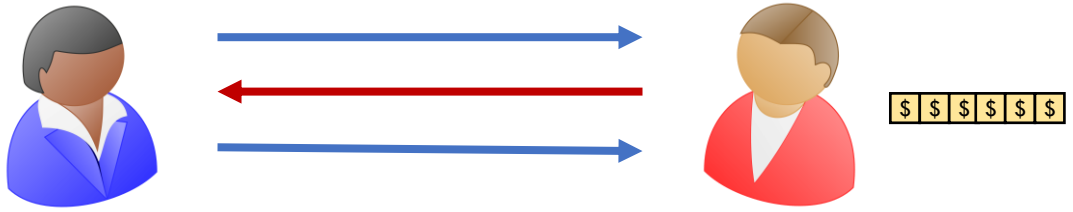


 wants to prove $x \in \mathcal{L}$ for some NP language $\mathcal{L}$.

There is a witness w of membership that can be verified easily.

Zero Knowledge

[Goldwasser-Micali-Rackoff'85]



Prover (x, w) Verifier (x)

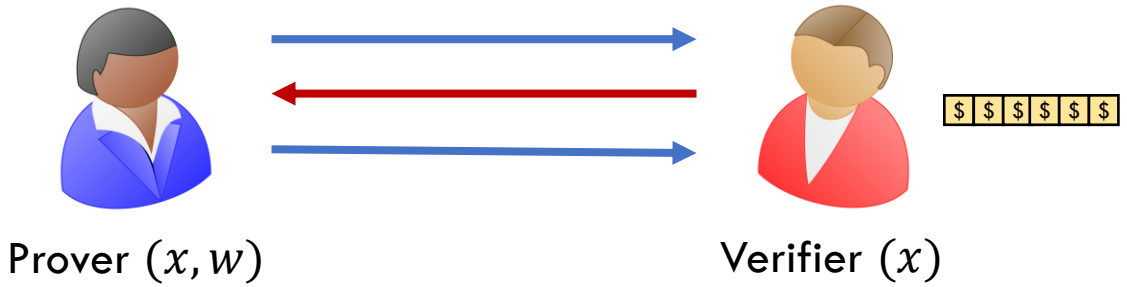
Completeness: $\forall x \in \mathcal{L}$, verifier accepts.

(Computational) Soundness

Zero Knowledge

Zero Knowledge

[Goldwasser-Micali-Rackoff'85]



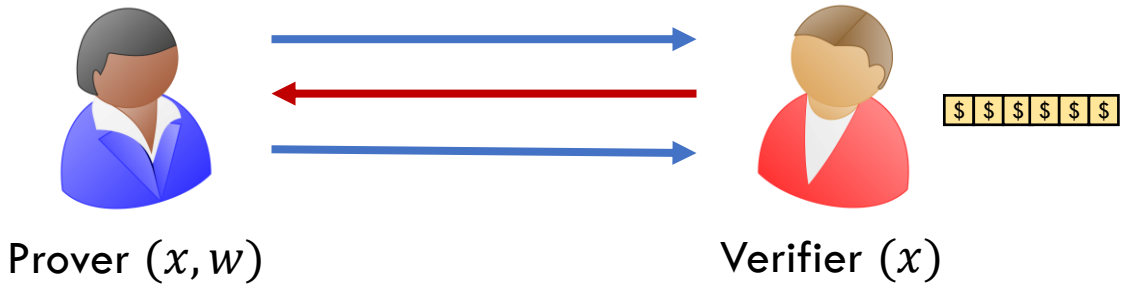
Completeness

(Computational) Soundness: $\forall x \notin \mathcal{L}$, no PPT prover can make the verifier accept.

Zero Knowledge

Zero Knowledge

[Goldwasser-Micali-Rackoff'85]



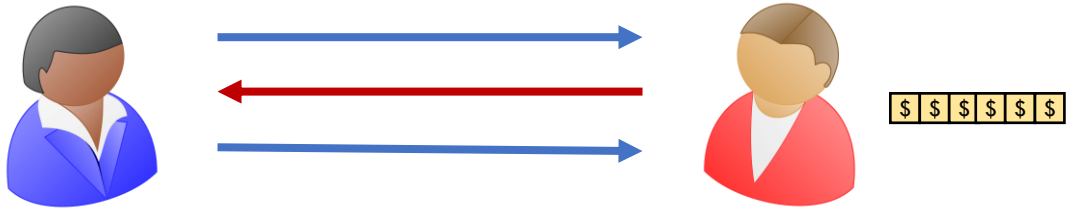
Completeness

(Computational) Soundness

Zero Knowledge: ∀ Verifiers 😈 ∃ Simulator 😈

Zero Knowledge

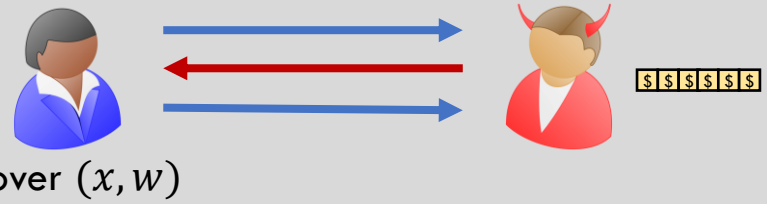
[Goldwasser-Micali-Rackoff'85]



Completeness

(Computational) Soundness

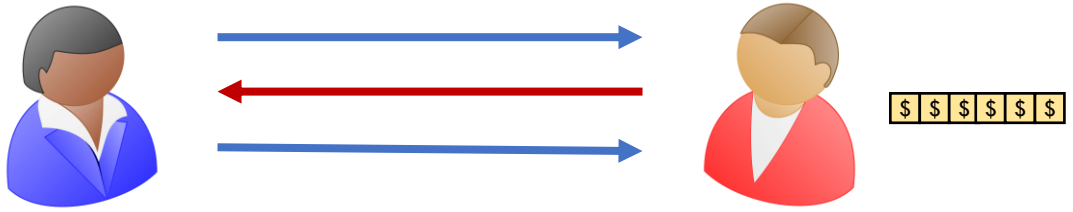
Zero Knowledge: ∀ Verifiers ∃ Simulator



Verifier's view in an execution with the prover

Zero Knowledge

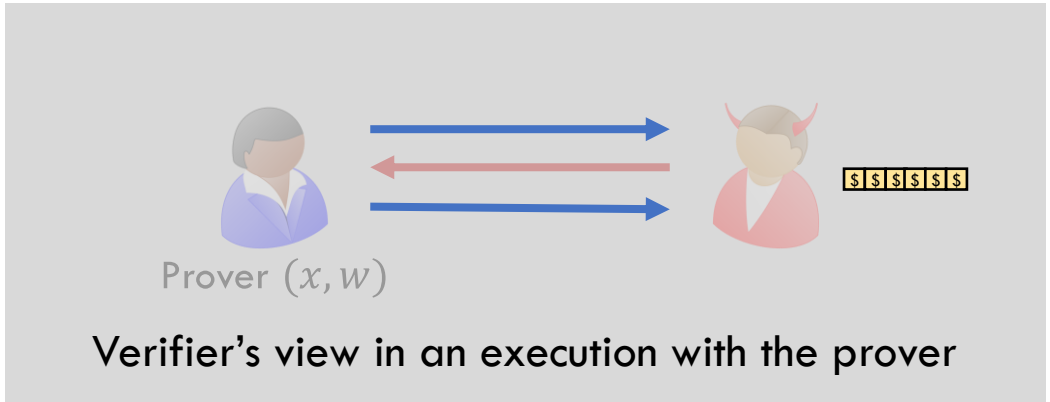
[Goldwasser-Micali-Rackoff'85]



Completeness

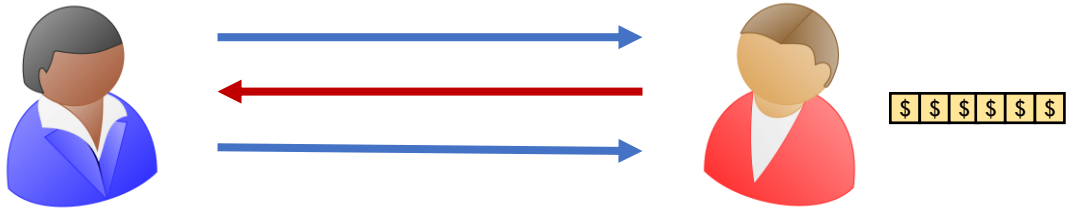
(Computational) Soundness

Zero Knowledge: ∀ Verifiers ∃ Simulator



Zero Knowledge

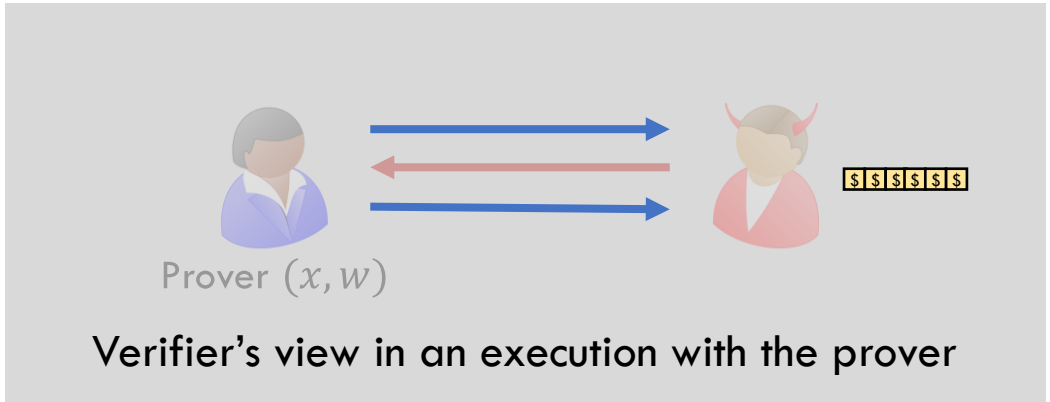
[Goldwasser-Micali-Rackoff'85]



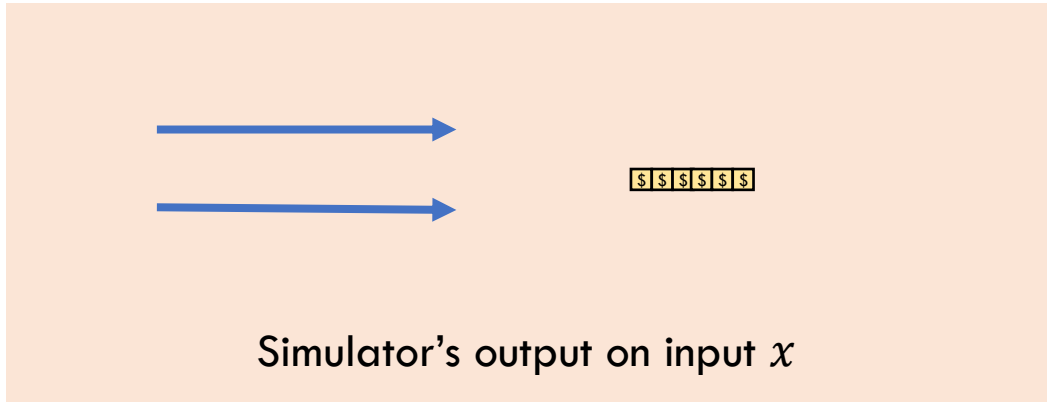
Completeness

(Computational) Soundness

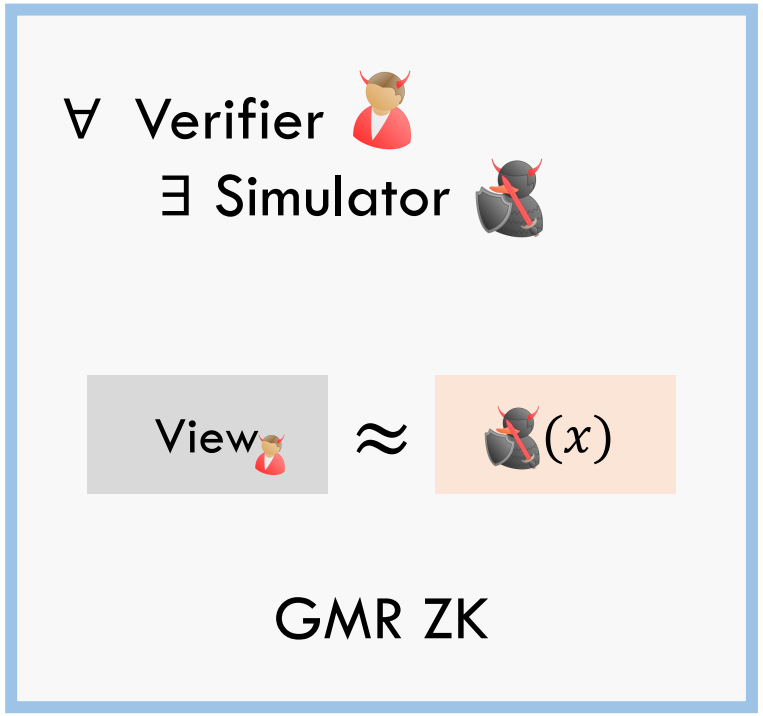
Zero Knowledge: ∀ Verifiers ∃ Simulator



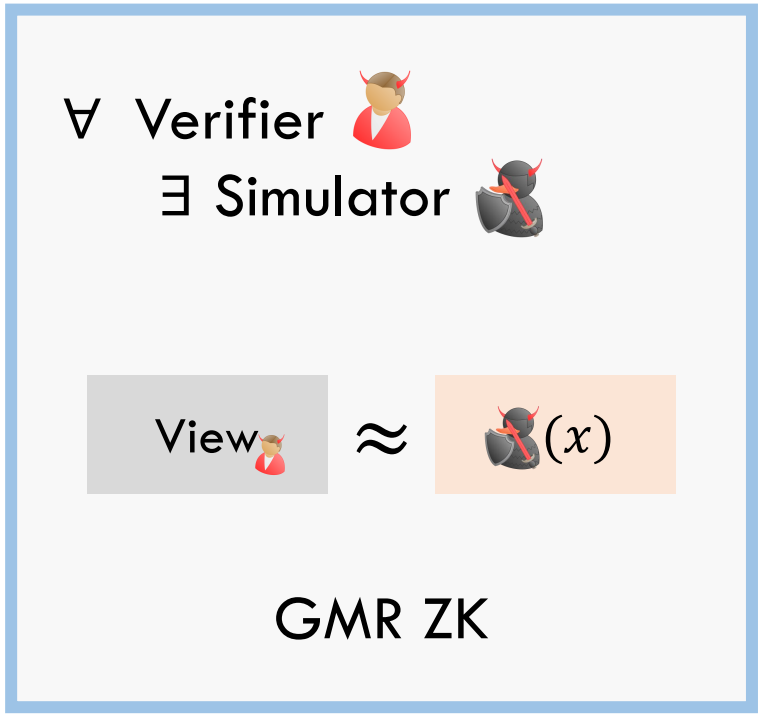
$\approx$

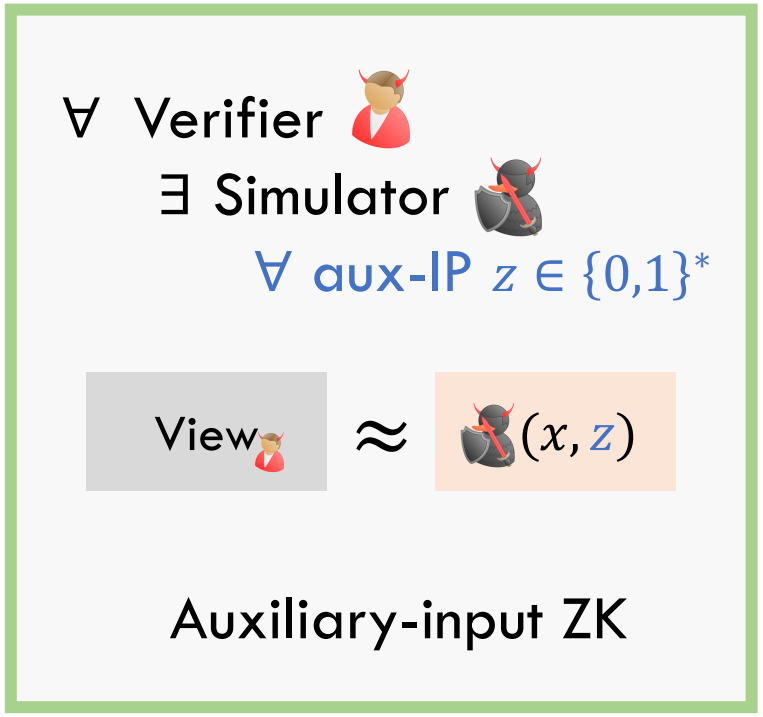


Many Flavors of Zero-Knowledge (ZK)



Many Flavors of Zero-Knowledge (ZK)





Auxiliary input captures protocol context for the verifier.

Many Flavors of Zero-Knowledge (ZK)

∀ Verifier

∃ Simulator

GMR ZK

∀ Verifier

∃ Simulator

∀ aux-IP $z \in \{0,1\}^*$

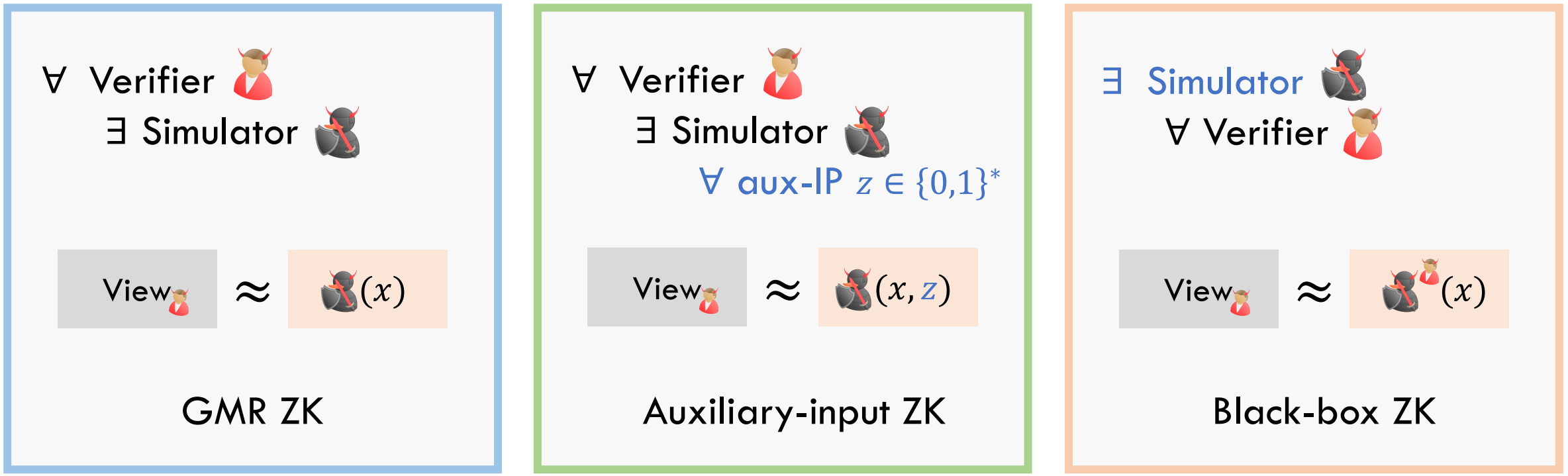
Auxiliary-input ZK

∃ Simulator

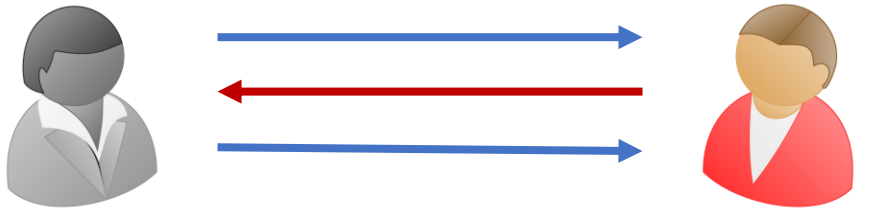
∀ Verifier

Black-box ZK

Many Flavors of Zero-Knowledge (ZK)



Deterministic Prover Zero Knowledge (DPZK)



Deterministic Prover (x, w) Verifier (x)

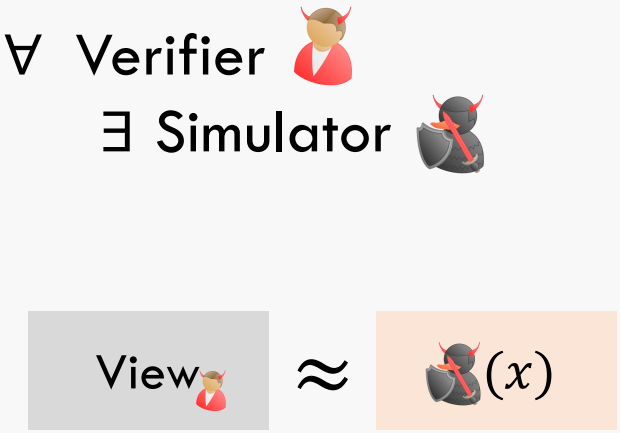
Is prover randomness essential for zero knowledge?

Limitations of DPZK

[Goldreich-Oren'94]

∀ Verifier

∃ Simulator

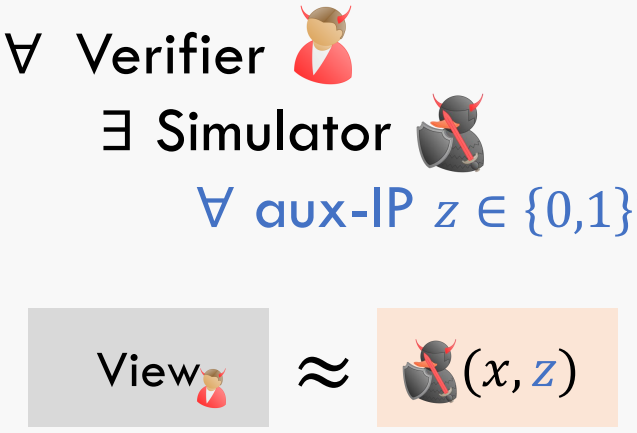


GMR ZK

∀ Verifier

∃ Simulator

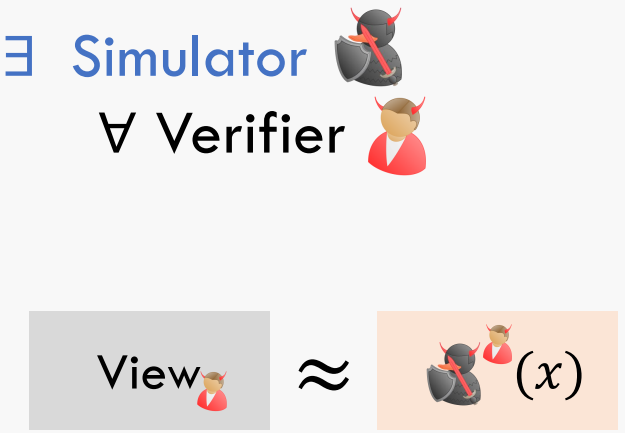
∀ aux-IP $z \in \{0,1\}^*$



Auxiliary-input ZK

∃ Simulator

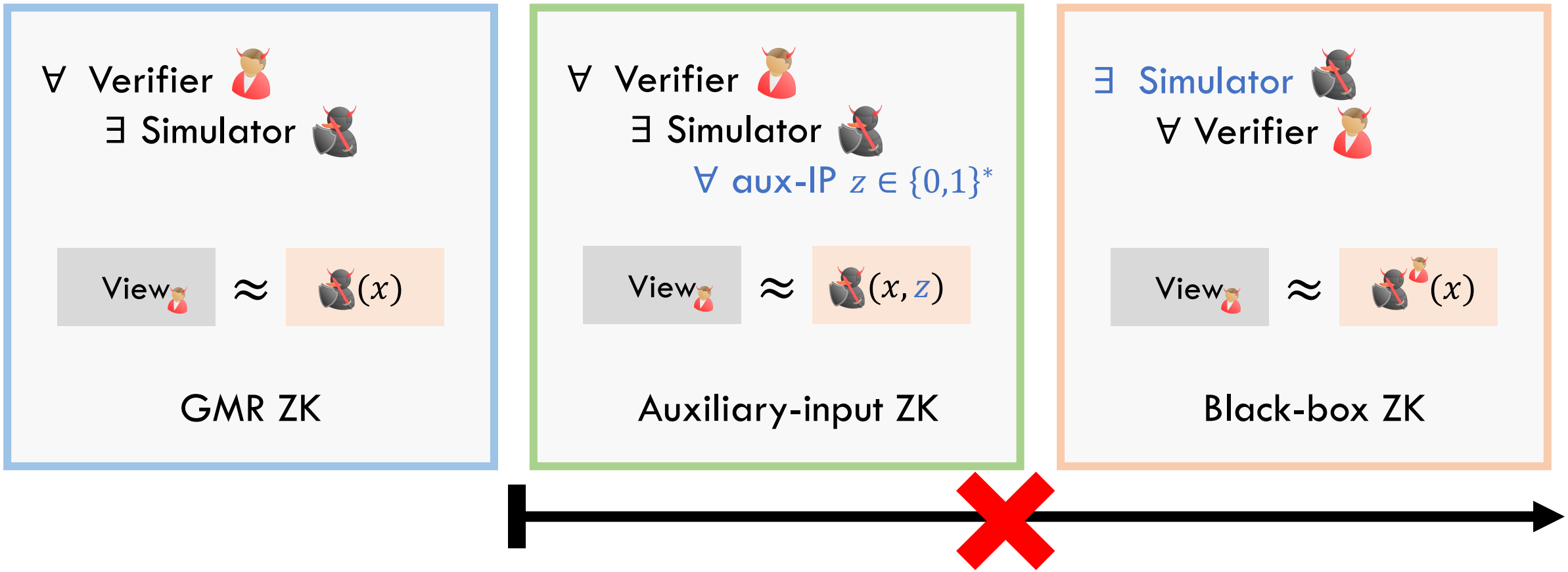
∀ Verifier



Black-box ZK

Limitations of DPZK

[Goldreich-Oren'94]



Prior Work

[Faonio-Nielsen-Venturi'17]

Witness encryption for $\mathcal{L} \Longrightarrow$ Honest-verifier DPZK for $\mathcal{L}$

Hash proof system for $\mathcal{L} \Longrightarrow$ Honest-verifier DPZK proofs for $\mathcal{L}$

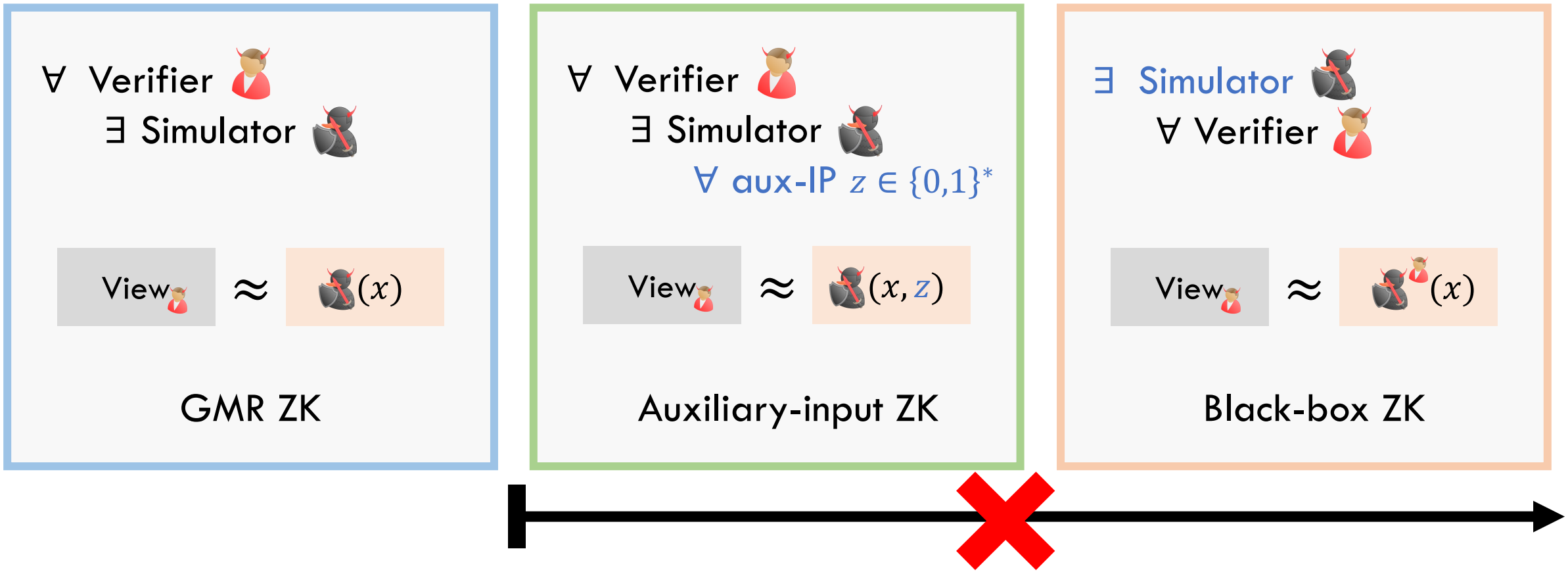
[Dahari-Lindell'20]

Doubly enhanced injective OWFs $\Longrightarrow$ Honest-verifier DPZK proofs for NP

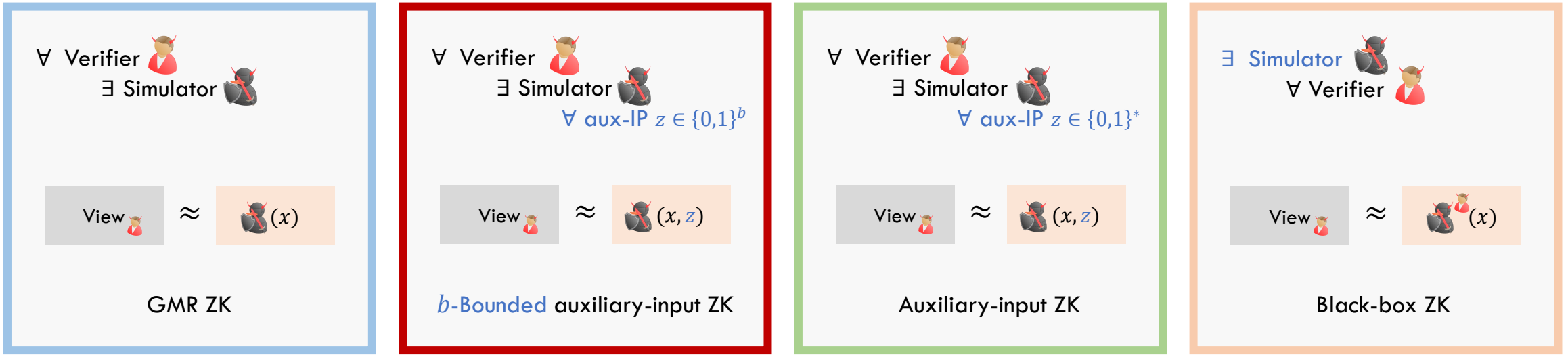
Inefficient honest prover.

Malicious-verifier DPZK for languages that have an entropy guarantee from witnesses.

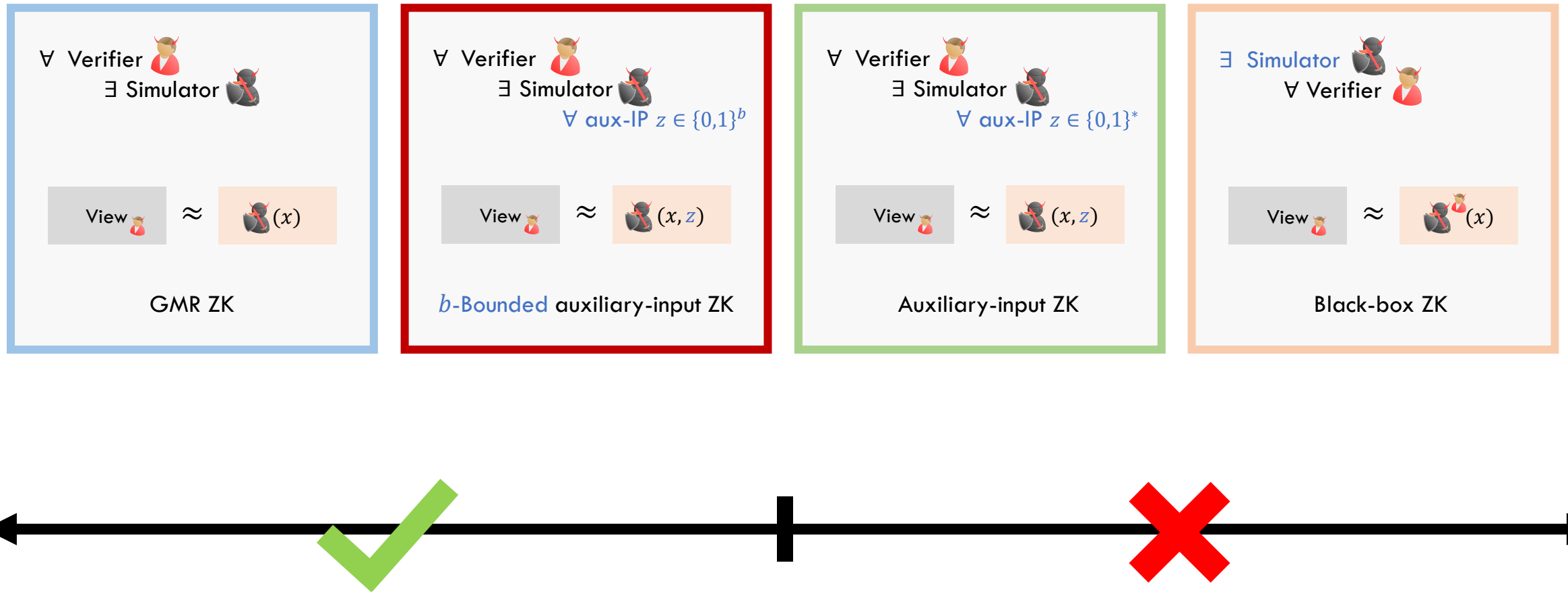
Our Results


∀ Verifier
∃ Simulator
View ≈ (x)
GMR ZK
∀ Verifier
∃ Simulator
∀ aux-IP z ∈ {0,1}*
View ≈ (x, z)
Auxiliary-input ZK
∃ Simulator
∀ Verifier
View ≈ (x)
Black-box ZK

Our Results


∀ Verifier
∃ Simulator
View ≈ (x)
GMR ZK
∀ Verifier
∃ Simulator
∀ aux-IP z ∈ {0,1}^b
View ≈ (x, z)
b-Bounded auxiliary-input ZK
∀ Verifier
∃ Simulator
∀ aux-IP z ∈ {0,1}*
View ≈ (x, z)
Auxiliary-input ZK
∃ Simulator
∀ Verifier
View ≈ (x)
Black-box ZK

Our Results


∀ Verifier
∃ Simulator
View ≈ (x)
GMR ZK
∀ Verifier
∃ Simulator
∀ aux-IP $z \in \{0,1\}^b$
View ≈ (x, z)
b-Bounded auxiliary-input ZK
∀ Verifier
∃ Simulator
∀ aux-IP $z \in \{0,1\}^*$
View ≈ (x, z)
Auxiliary-input ZK
∃ Simulator
∀ Verifier
View ≈ (x)
Black-box ZK

Our Results

Assuming NIWIs + sub-exponentially secure iO + OWF + sub-exponentially secure keyless CRHF, there exist two message DPZK arguments for all of NP against bounded auxiliary-input verifiers.

NIWI – Non interactive witness indistinguishable proofs
iO – Indistinguishability obfuscation
OWF – One-way functions
CRHF – Collision resistant hash functions

Our Results

Assuming NIWIs + sub-exponentially secure iO + OWF + sub-exponentially secure keyless CRHF, there exist two message DPZK arguments for all of NP against bounded auxiliary-input verifiers.

Any DPZK argument for a language $\mathcal{L}$ implies a witness encryption for $\mathcal{L}$.

Honest Verifier DPZK

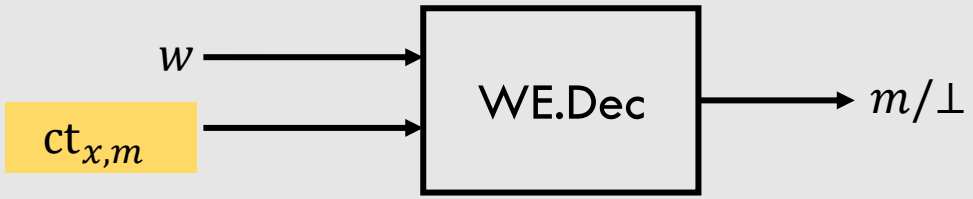
[Faonio-Nielsen-Venturi'17]

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Witness Encryption for $\mathcal{L}$

Deterministic Decryption

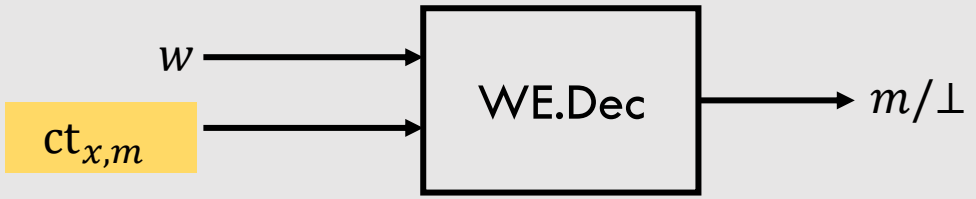


Honest Verifier DPZK

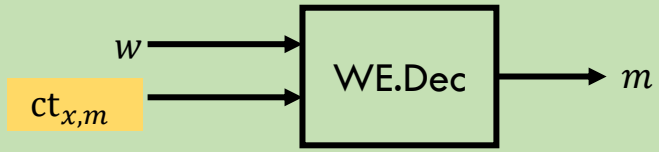
[Faonio-Nielsen-Venturi'17]

Witness Encryption for $\mathcal{L}$

Deterministic Decryption



For $(x, w) \in \mathrm{Rel}_{\mathcal{L}}$



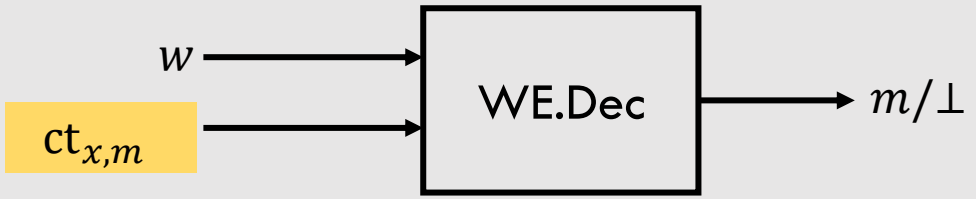
Correctness

Honest Verifier DPZK

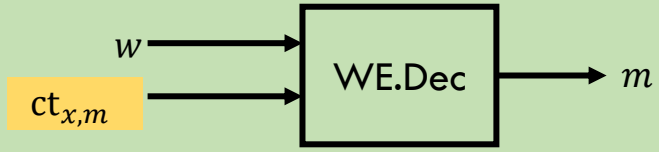
[Faonio-Nielsen-Venturi'17]

Witness Encryption for $\mathcal{L}$

Deterministic Decryption



For $(x, w) \in \mathrm{Rel}_{\mathcal{L}}$



Correctness

For $x \notin \mathcal{L}$

Security

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Deterministic Prover (x, w)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Deterministic Prover (x, w)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

$\tilde{u} := \mathsf{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Deterministic Prover (x, w)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

$\tilde{u} := \mathsf{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness: From correctness of WE.

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Cheating Prover (x)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness: From WE security when $x \notin \mathcal{L}$

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Cheating Prover (x)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

?

Output 1 iff $u = \tilde{u}$

Completeness

Soundness: From WE security when $x \notin \mathcal{L}$

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Simulator (x)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness

Honest Verifier Zero Knowledge:

Honest Verifier DPZK

[Faonio-Nielsen-Venturi'17]

Simulator (x)

Verifier (x)

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness

Honest Verifier Zero Knowledge: Simulator knows u

Explainable Verifier DPZK

Explainable Verifier

There exist honest verifier coins that explains verifier messages as honest messages.

Explainable Verifier DPZK

Explainable Verifier

There exist honest verifier coins that explains verifier messages as honest messages.

Simulator no longer “knows” the message that an explainable verifier encrypts via the Witness Encryption.

Aux-I/P DPZK for explainable verifiers also ruled out by [Goldreich-Oren’94]

Explainable Verifier DPZK

Explainable Verifier

There exist honest verifier coins that explains verifier messages as honest messages.

Simulator no longer "knows" the message that an explainable verifier encrypts via the Witness Encryption.

Aux-I/P DPZK for explainable verifiers also ruled out by [Goldreich-Oren'94]

Idea: Use additional trapdoor statement that only the simulator can use.

Malicious Verifier DPZK

Verifier proves honest behavior

Explainable Verifier DPZK

Deterministic Prover (x, w)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$\mathrm{ct}_{x,u}$

$\tilde{u} := \mathsf{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Explainable Verifier DPZK

Deterministic Prover (x, w)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathrm{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathrm{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$

$\tilde{u} := \mathrm{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Statement: R

N

Witness: Turing Machine

b

TM

R

Explainable Verifier DPZK

Deterministic Prover (x, w)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathrm{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathrm{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$ $\widetilde{\mathrm{ct}}_{R,u}$ R

$\tilde{u} := \mathrm{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

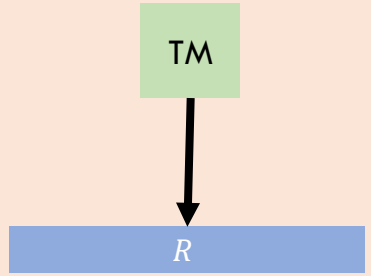
Output 1 iff $u = \tilde{u}$

Statement: R

N

Witness: Turing Machine

b



Explainable Verifier DPZK

Deterministic Prover (x, w)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathrm{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathrm{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$ $\widetilde{\mathrm{ct}}_{R,u}$ R

$\tilde{u} := \mathrm{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness: Same as HVZK

Statement: R

N

Witness: Turing Machine

b

TM

R

Explainable Verifier DPZK

Cheating Prover (x, w)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathsf{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$ $\widetilde{\mathrm{ct}}_{R,u}$ R

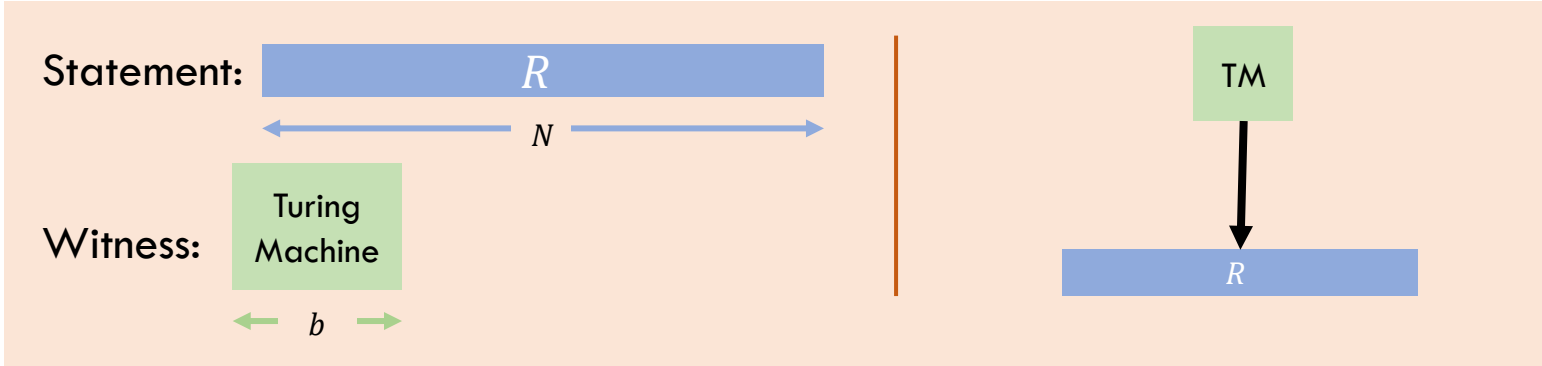
$\tilde{u} := \mathsf{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness: w.h.p. no short machine exists that can output a random R



Explainable Verifier DPZK

Simulator (x)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathrm{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathrm{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$ $\widetilde{\mathrm{ct}}_{R,u}$ R

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness

Zero Knowledge:

Statement: R

N

Witness: Turing Machine

b

TM

R

Explainable Verifier DPZK

Simulator (x)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathrm{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathrm{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$ $\widetilde{\mathrm{ct}}_{R,u}$ R

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness

Zero Knowledge: Simulator uses the verifier's code as witness; verifier's randomness simulated by a PRG.

Statement: R

N

Witness: Turing Machine

b

TM

R

Explainable Verifier DPZK

Simulator (x)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathrm{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathrm{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$ $\widetilde{\mathrm{ct}}_{R,u}$ R

$\tilde{u} := \widetilde{\mathrm{WE.Dec}}(\widetilde{\mathrm{ct}}_{R,u}$, )

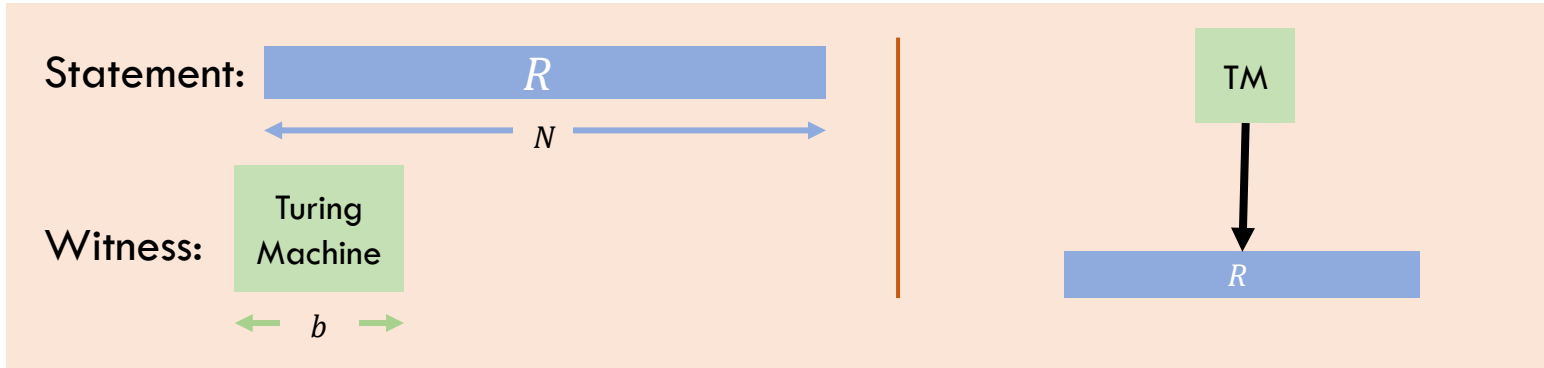
$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Completeness

Soundness

Zero Knowledge: Simulator uses the verifier's code as witness; verifier's randomness simulated by a PRG.



Explainable Verifier DPZK

Deterministic Prover (x, w)

Verifier (x)

size $\leq b$

includes auxiliary input

$u \leftarrow \{0,1\}^n$

$\mathrm{ct}_{x,u} \leftarrow \mathsf{WE.Enc}_x(u)$

$R \leftarrow \{0,1\}^N$

$\widetilde{\mathrm{ct}}_{R,u} \leftarrow \widetilde{\mathsf{WE.Enc}}_R(u)$

$N \gg b$

$\mathrm{ct}_{x,u}$

$\tilde{u} := \mathsf{WE.Dec}(\mathrm{ct}_{x,u}, w)$

$\tilde{u}$

Output 1 iff $u = \tilde{u}$

Necessity of Randomness in Zero-knowledge

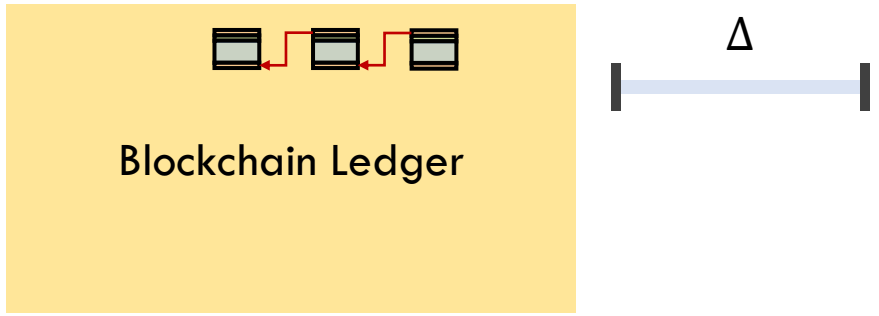
Founding Secure Computation on Blockchains

Round Optimal Secure Computation

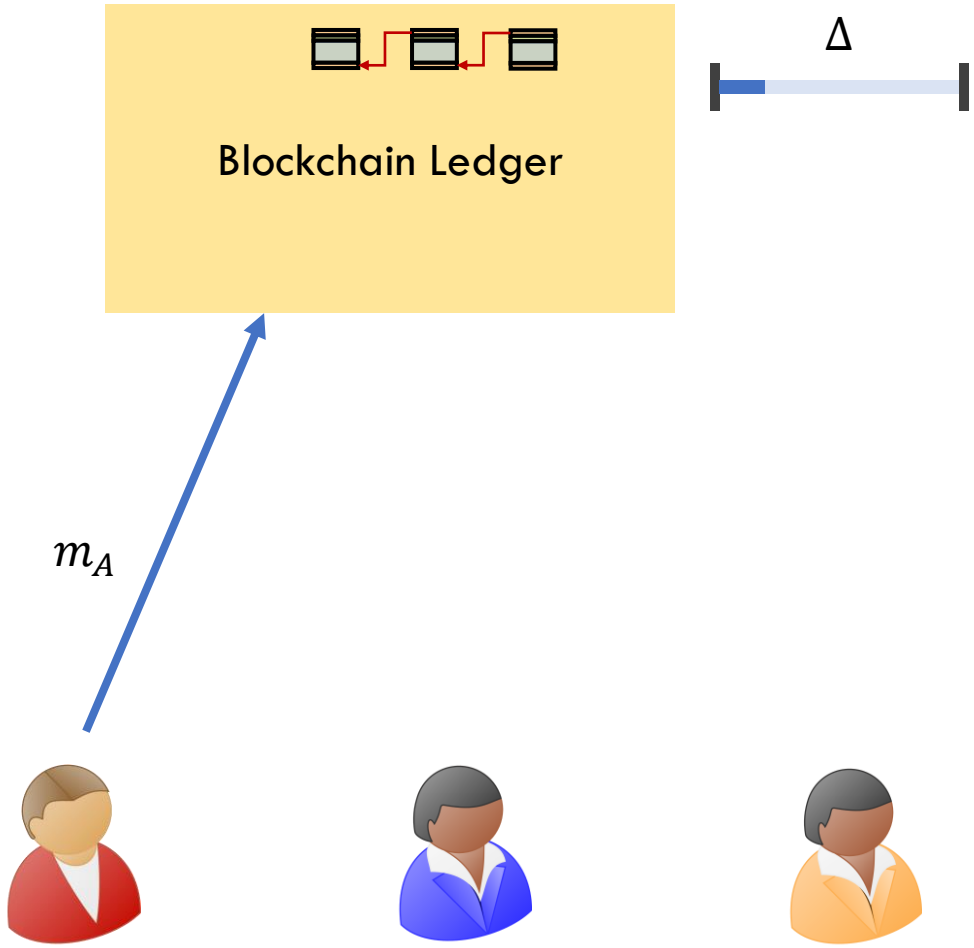
Founding Secure Computation on Blockchains

[C-Goyal-Jain'19]

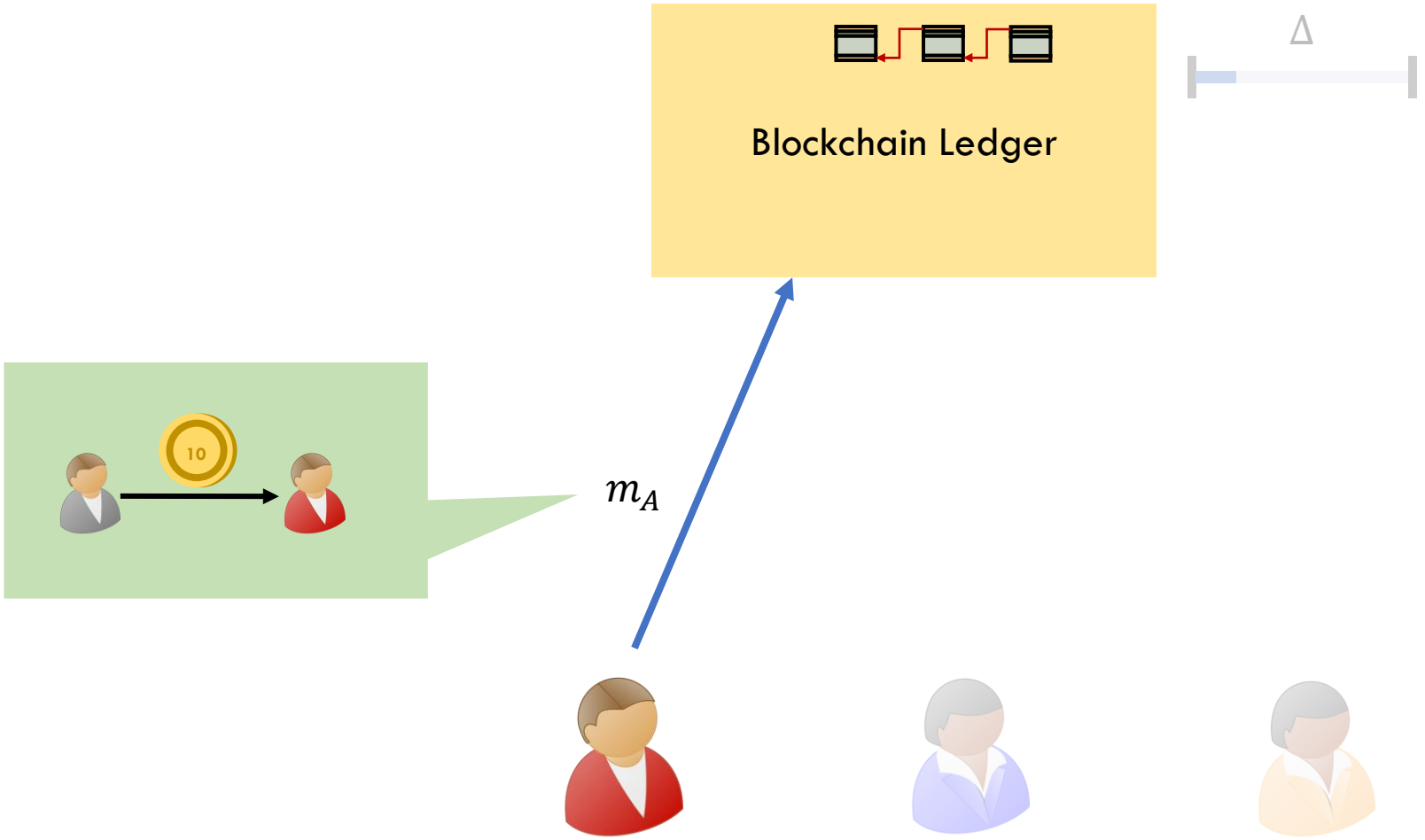
Blockchain Model



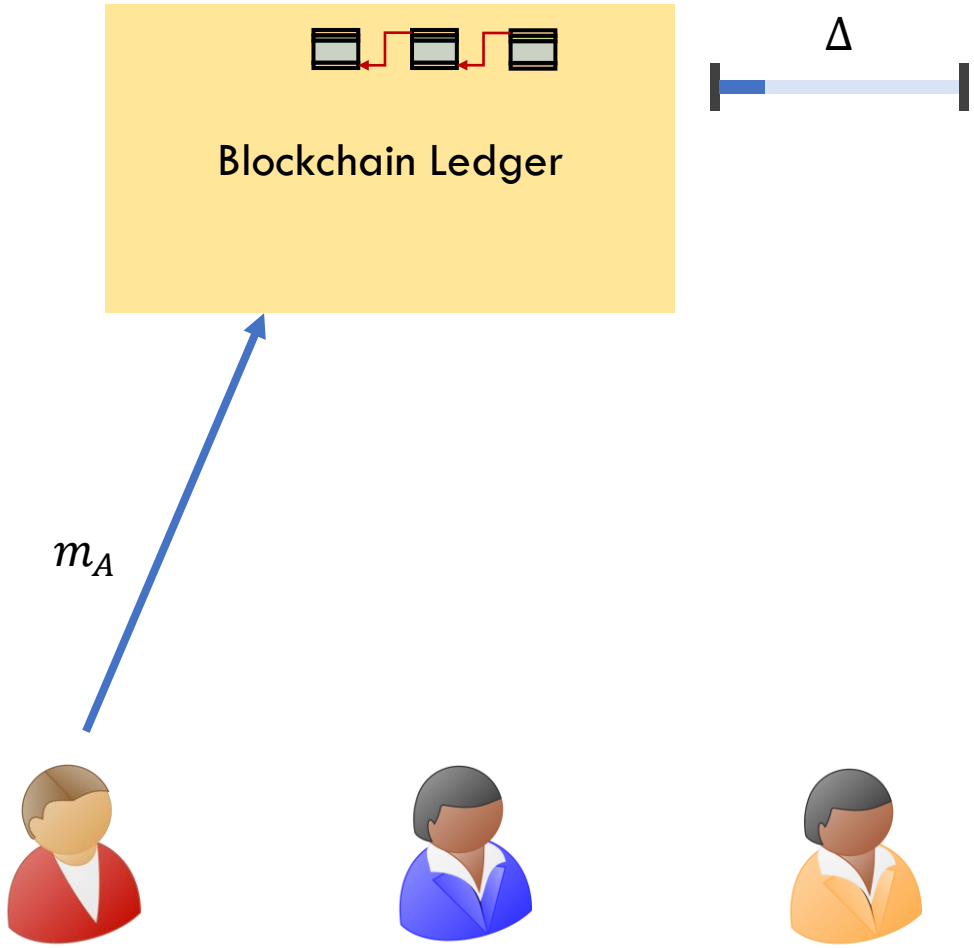
Blockchain Model



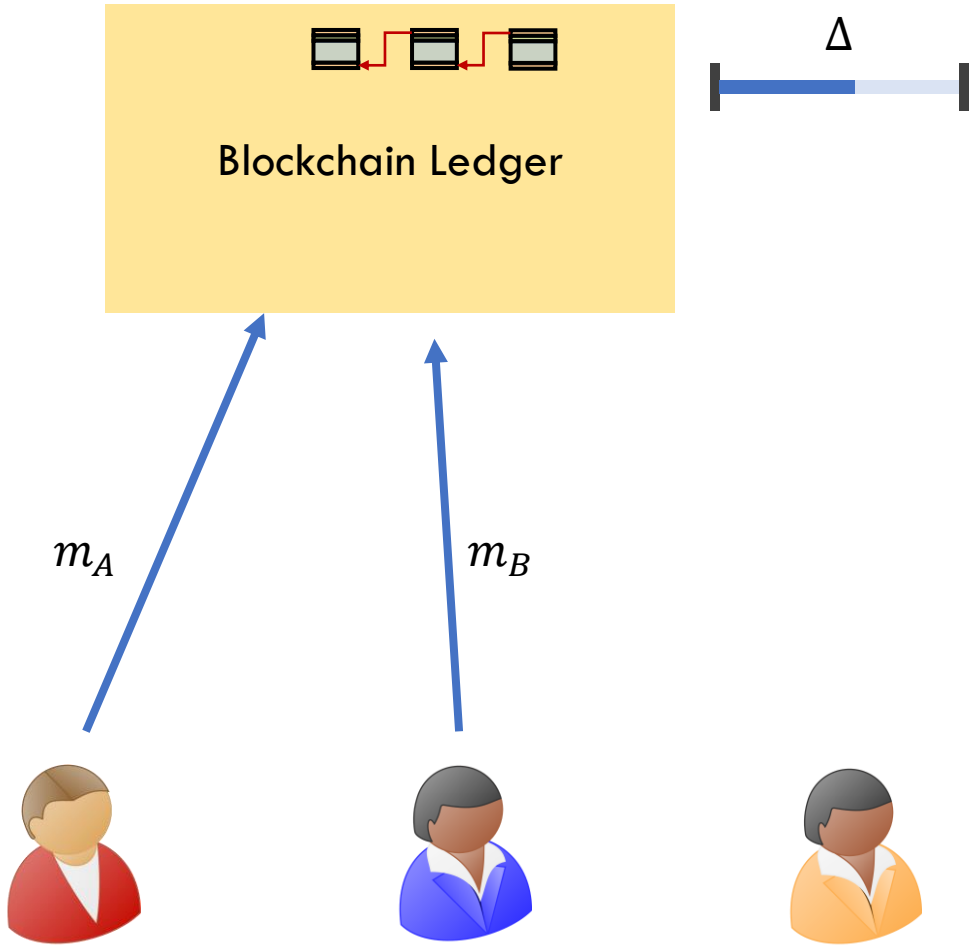
Blockchain Model



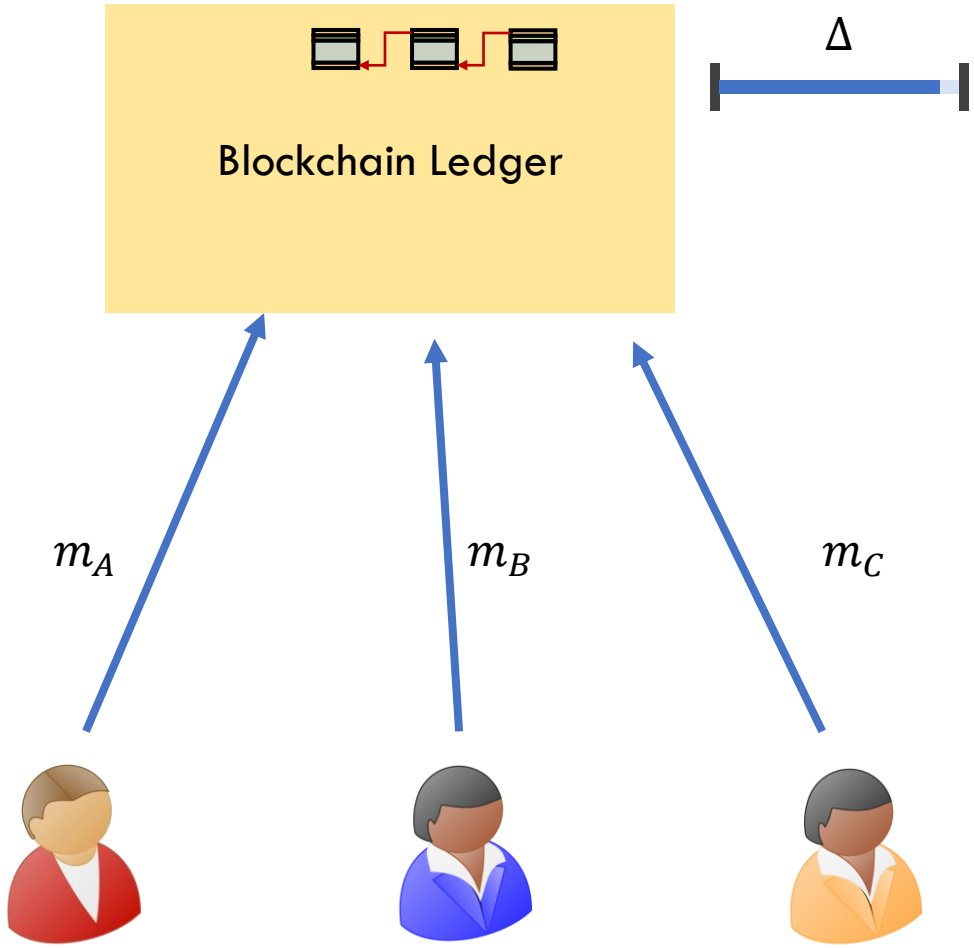
Blockchain Model



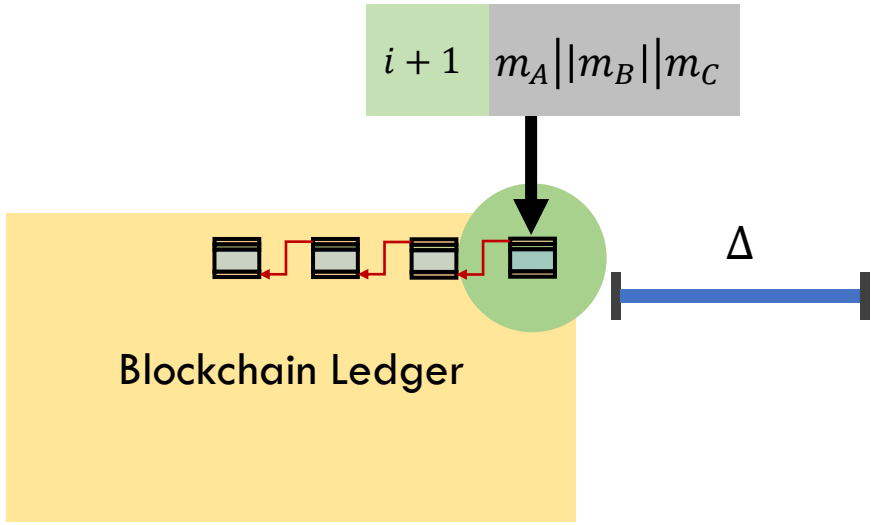
Blockchain Model



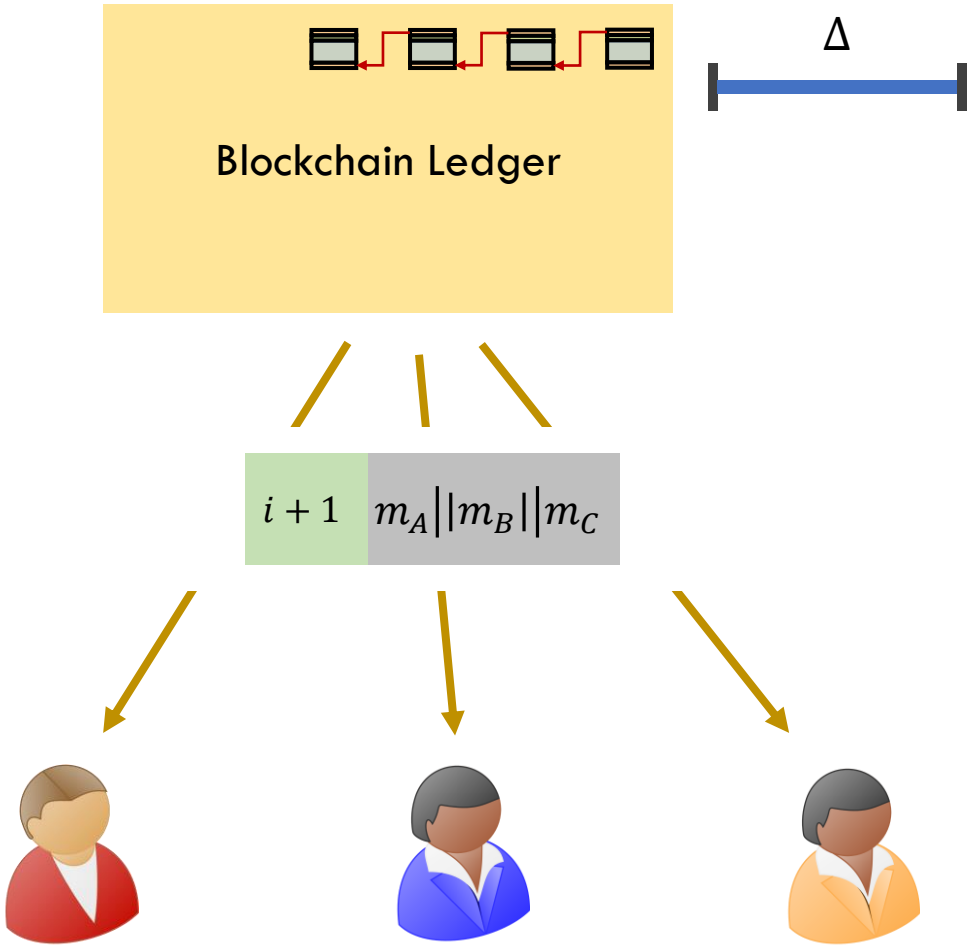
Blockchain Model



Blockchain Model

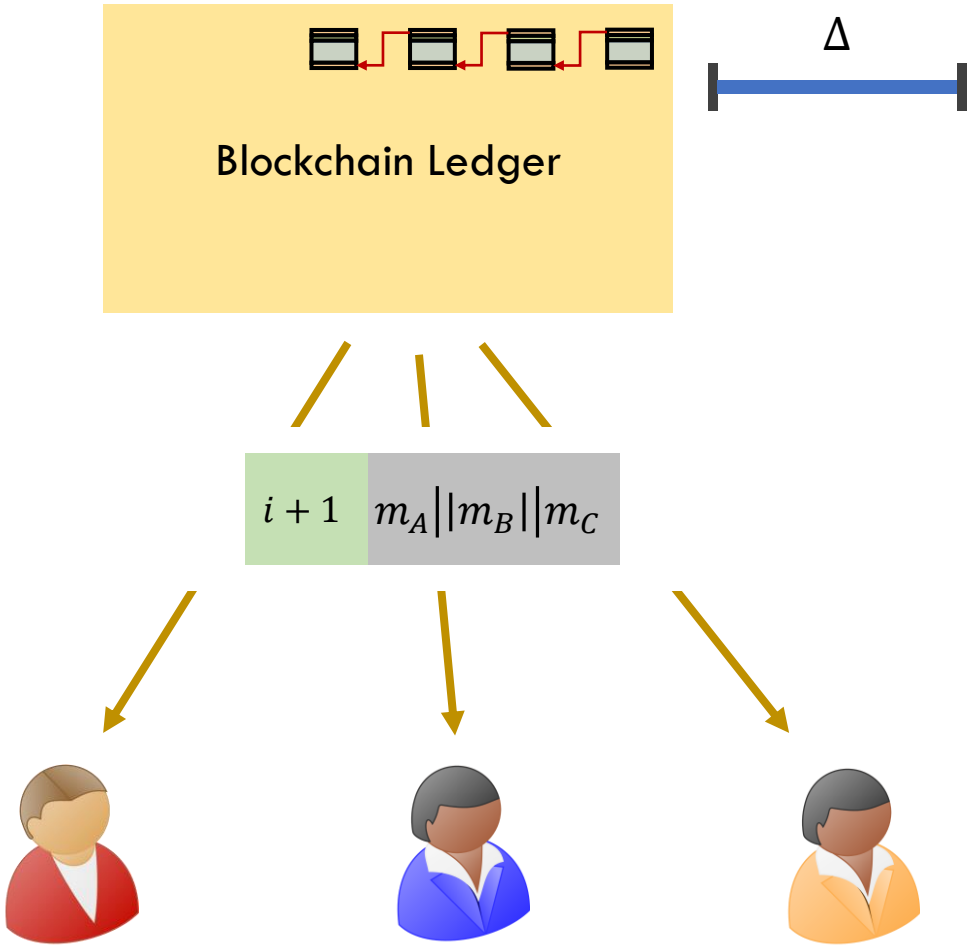


Blockchain Model



Blockchain Model

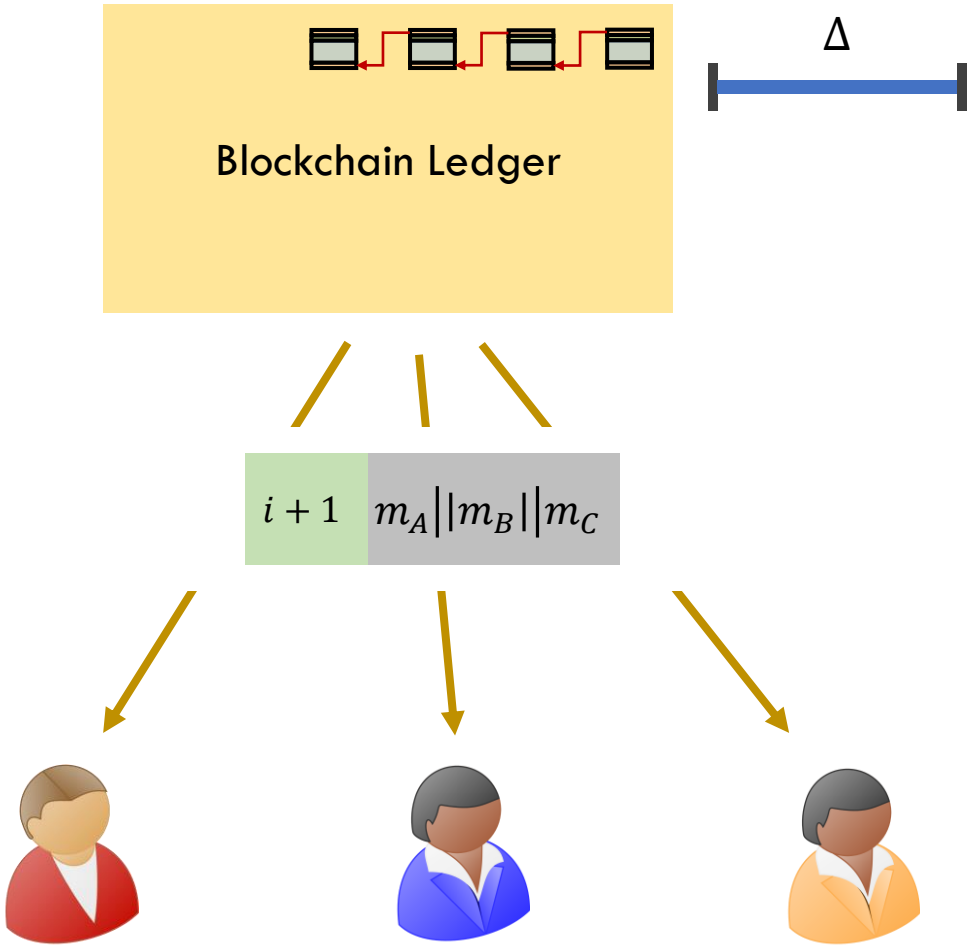
All parties have a consistent view of the blockchain



Blockchain Model

All parties have a consistent view of the blockchain

A message sent to the oracle is guaranteed to appear on the next block

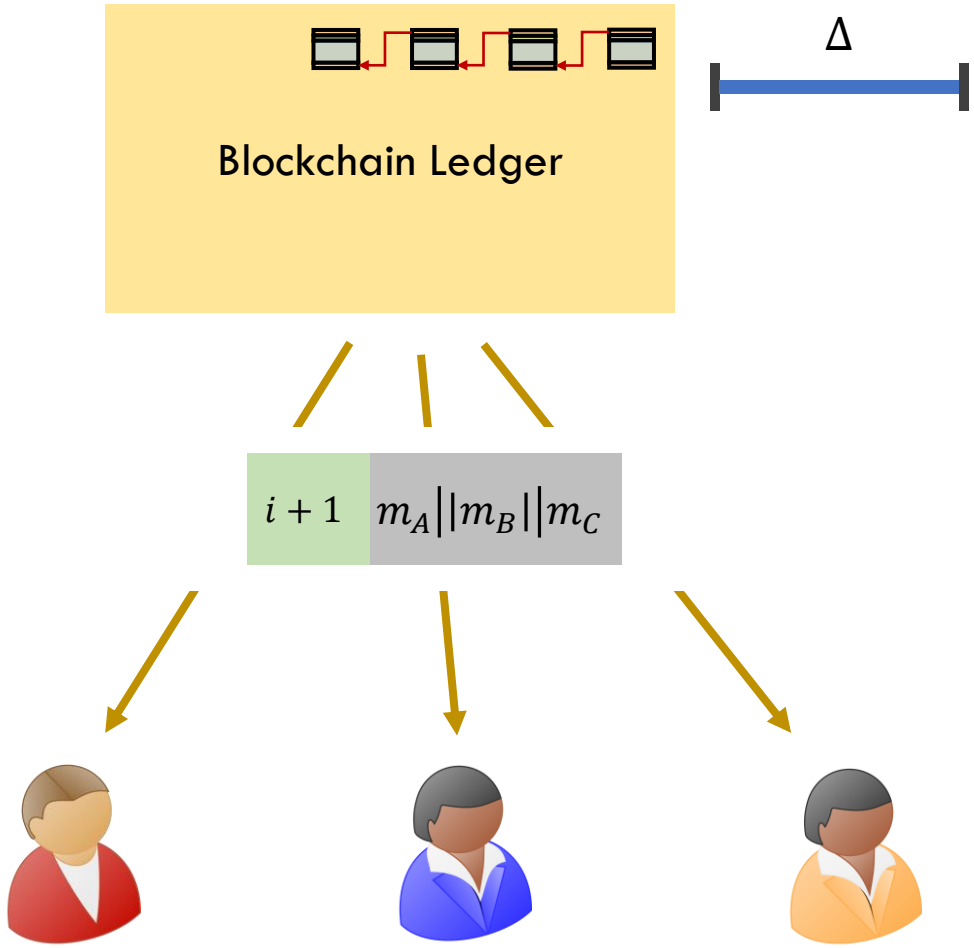


Blockchain Model

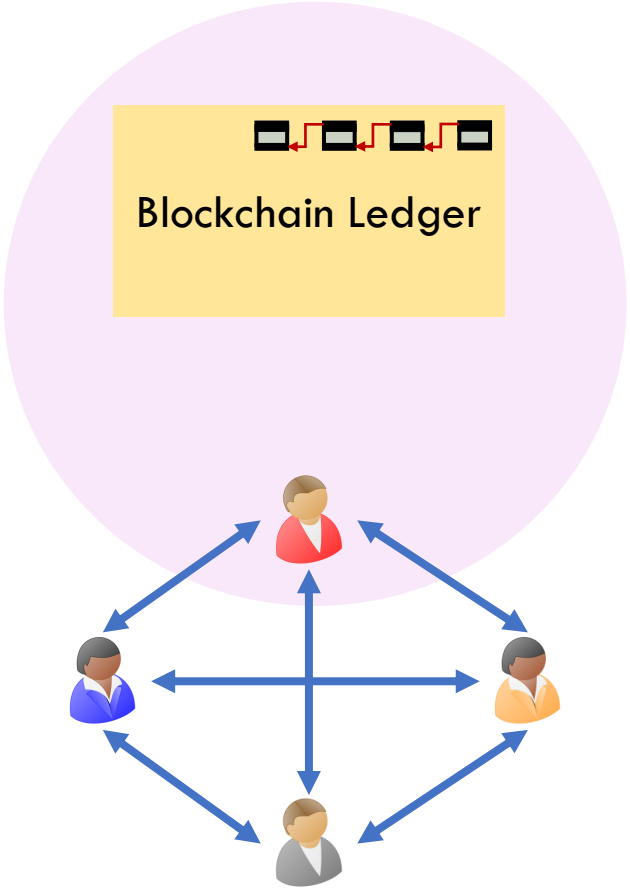
All parties have a consistent view of the blockchain

A message sent to the oracle is guaranteed to appear on the next block

Only the oracle can create blocks

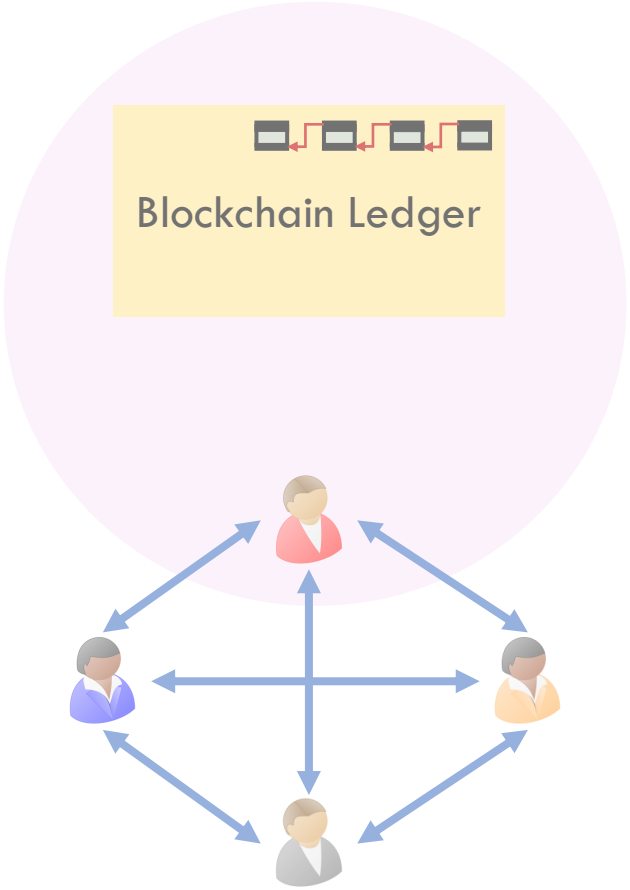


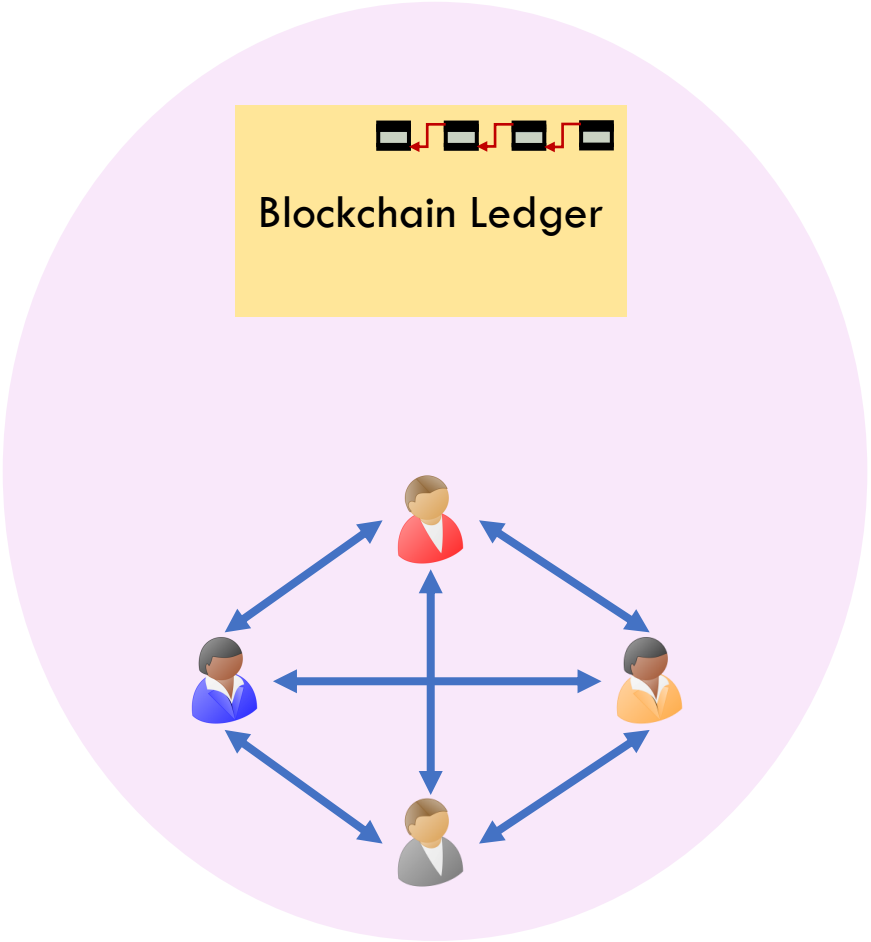
Blockchains and Protocols



 is blockchain-active

Blockchains and Protocols

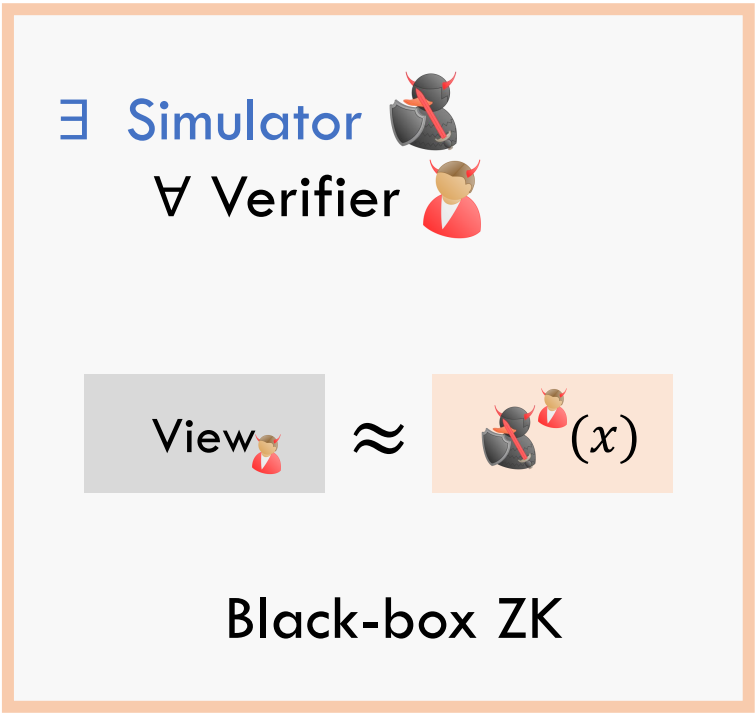




is blockchain-active

Protocol is in the blockchain hybrid model.

Black-box Zero-Knowledge



Black-box Zero-Knowledge

$\exists$ Simulator 

$\forall$ Verifier 

View  $\approx$  (x)

Black-box ZK

Black-box Zero-Knowledge

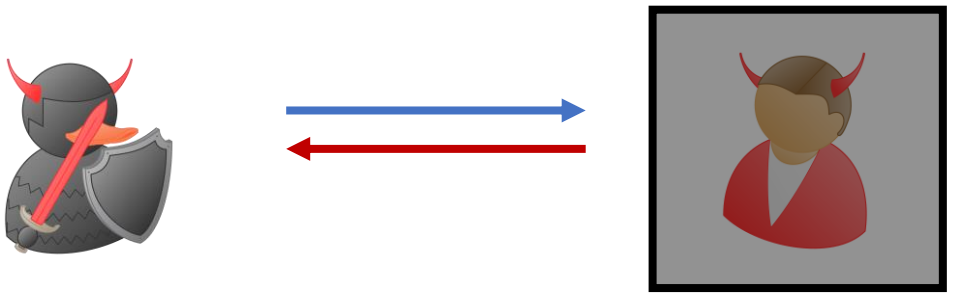
$\exists$ Simulator 

$\forall$ Verifier 

View  $\approx$  (x)

Black-box ZK

Black-box Zero-Knowledge



$\exists$ Simulator

$\forall$ Verifier

View $\approx$ (x)

Black-box ZK

Black-box Zero-Knowledge

∃ Simulator 

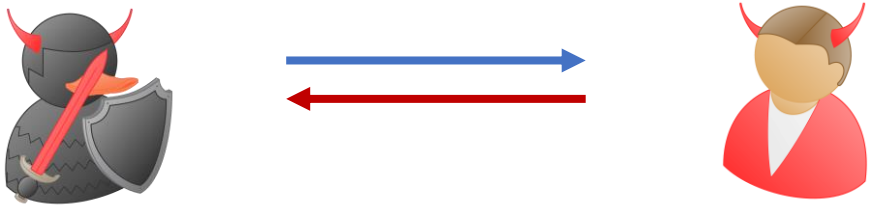
∀ Verifier 

View  ≈  (x)

Black-box ZK

Black-box simulator works by rewinding the cheating verifier.

Black-box Zero-Knowledge



$\exists$ Simulator

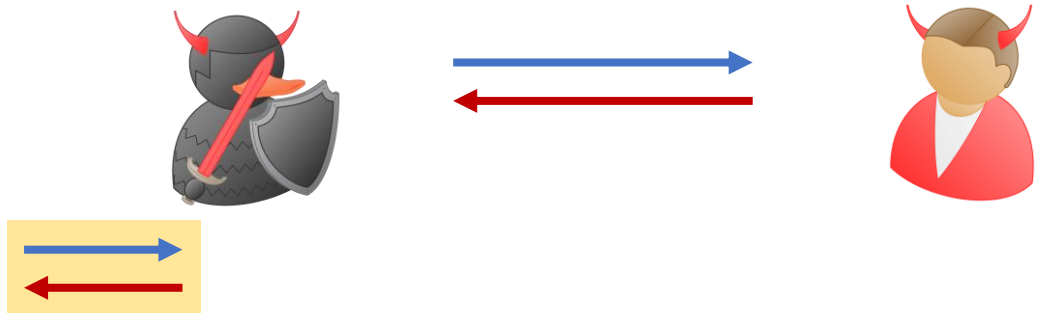
$\forall$ Verifier

View $\approx$ (x)

Black-box ZK

Black-box simulator works by rewinding the cheating verifier.

Black-box Zero-Knowledge



$\exists$ Simulator

$\forall$ Verifier

View $\approx$ (x)

Black-box ZK

Black-box simulator works by rewinding the cheating verifier.

Black-box Zero-Knowledge

$\exists$ Simulator

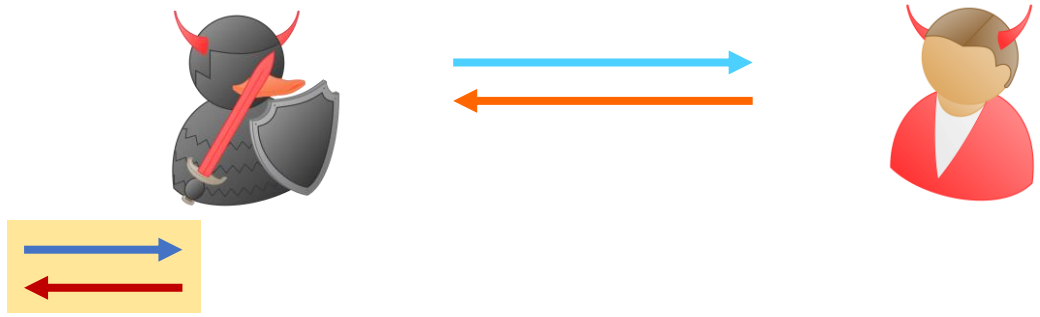
$\forall$ Verifier

View $\approx$ (x)

Black-box ZK

Black-box simulator works by rewinding the cheating verifier.

Black-box Zero-Knowledge



$\exists$ Simulator

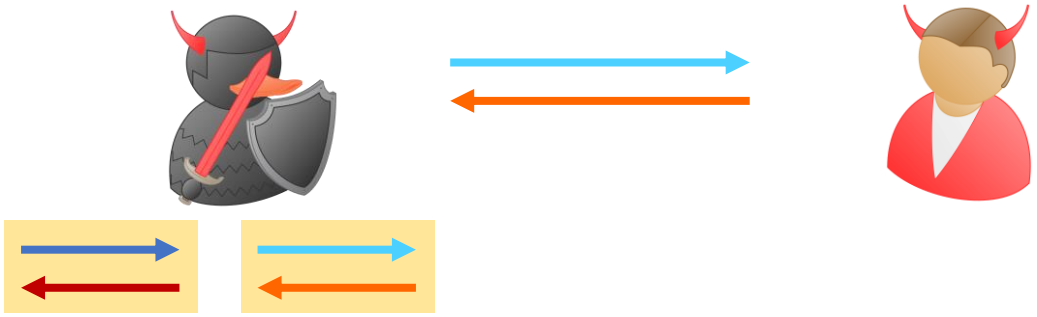
$\forall$ Verifier

View $\approx$ (x)

Black-box ZK

Black-box simulator works by rewinding the cheating verifier.

Black-box Zero-Knowledge



$\exists$ Simulator

$\forall$ Verifier

View $\approx$ (x)

Black-box ZK

Black-box simulator works by rewinding the cheating verifier.

Black-box ZK Impossible with Blockchain-active verifier

Blockchain Ledger

Prevent Simulator from rewinding the verifier.

prover

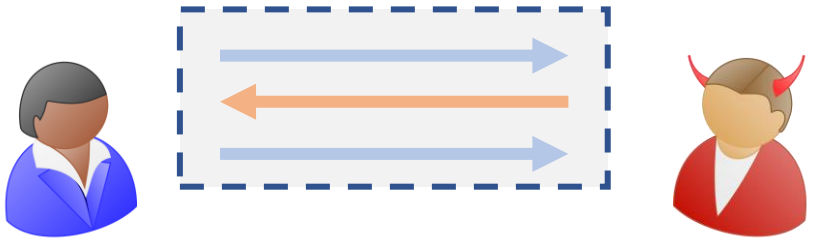
verifier

Black-box simulator works by rewinding the cheating verifier.

Black-box ZK Impossible with Blockchain-active verifier

Blockchain Ledger

Prevent Simulator from rewinding the verifier.



prover

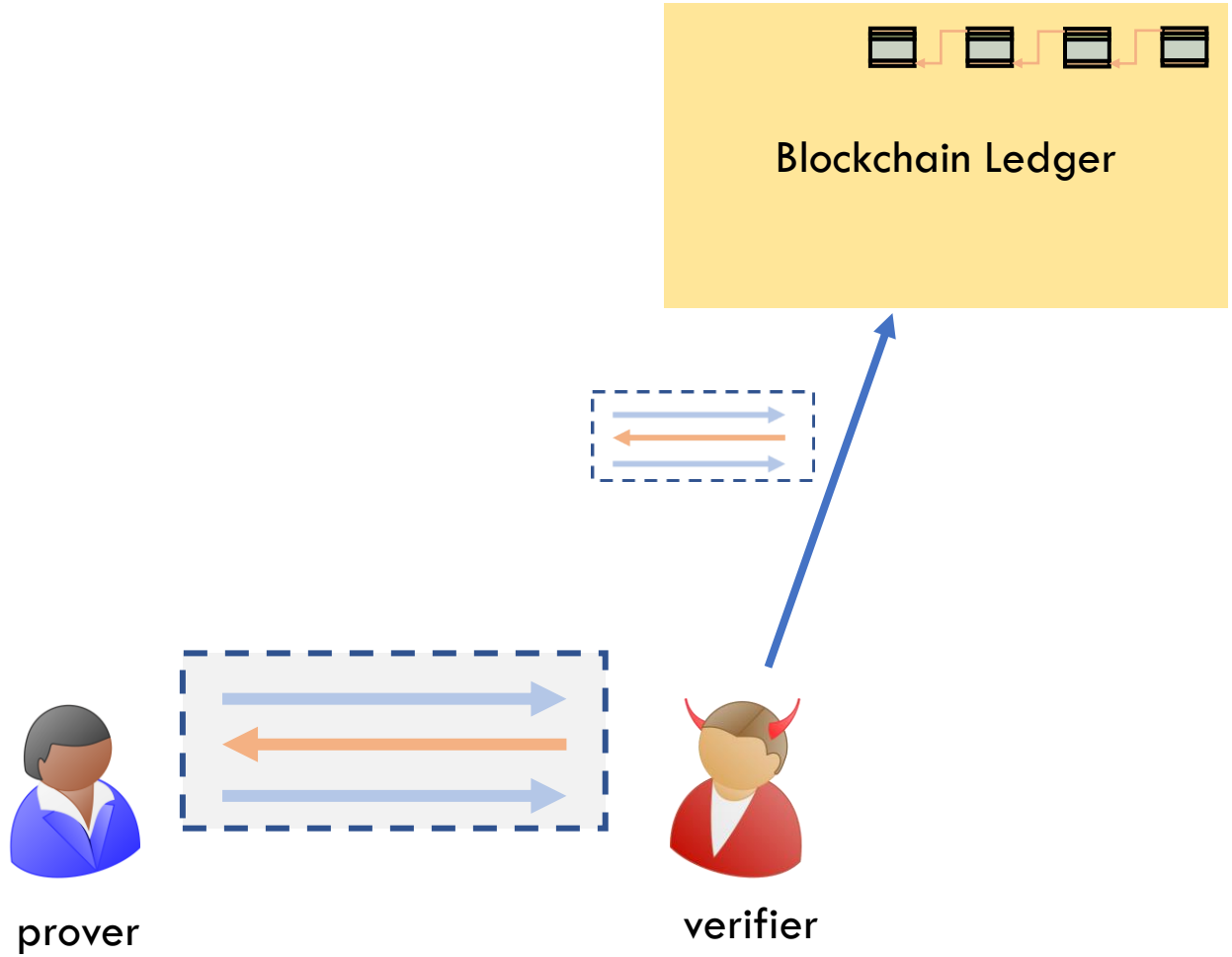
verifier

Black-box simulator works by rewinding the cheating verifier.

Black-box ZK Impossible with Blockchain-active verifier

Prevent Simulator from rewinding the verifier.

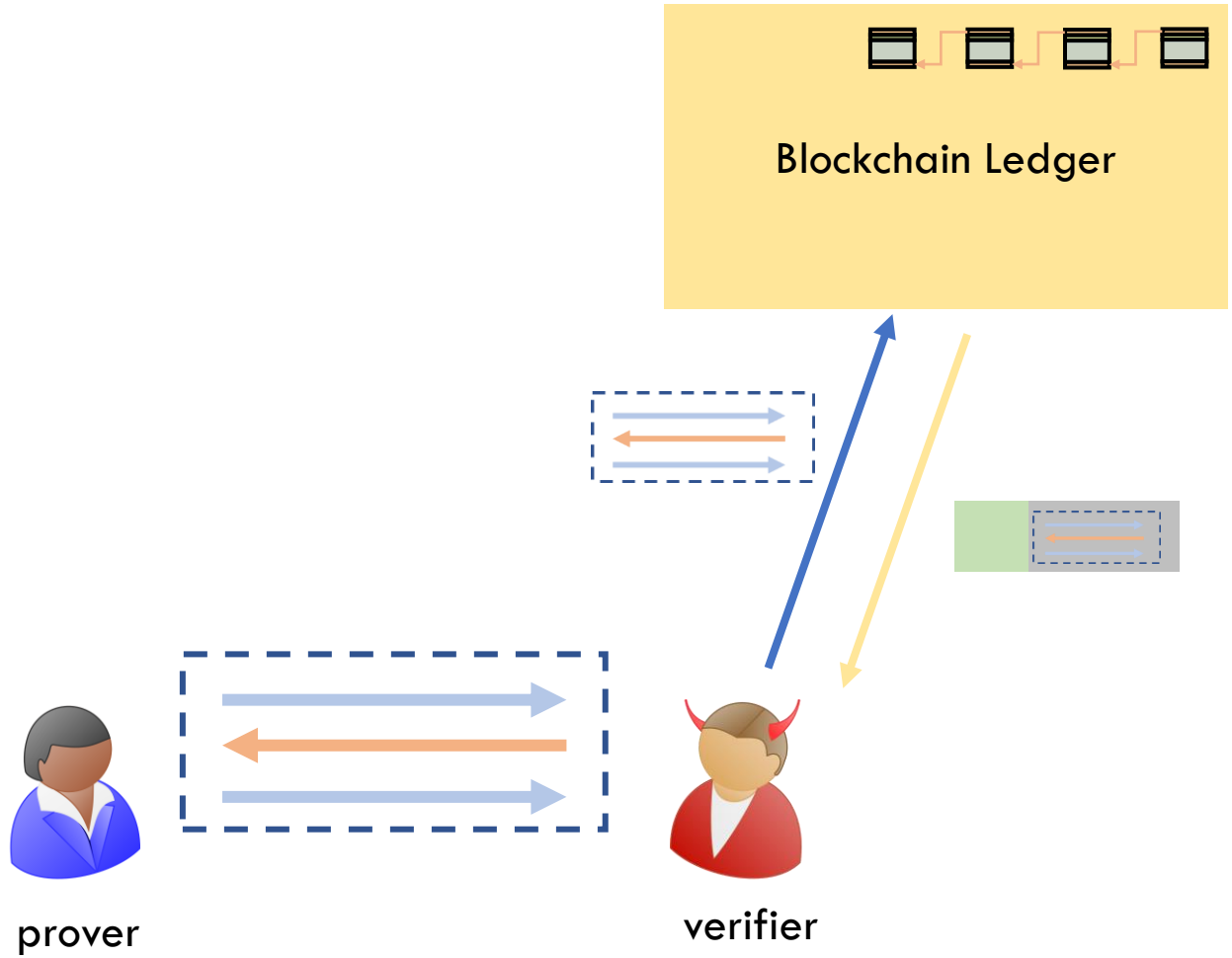
Black-box simulator works by rewinding the cheating verifier.



Black-box ZK Impossible with Blockchain-active verifier

Prevent Simulator from rewinding the verifier.

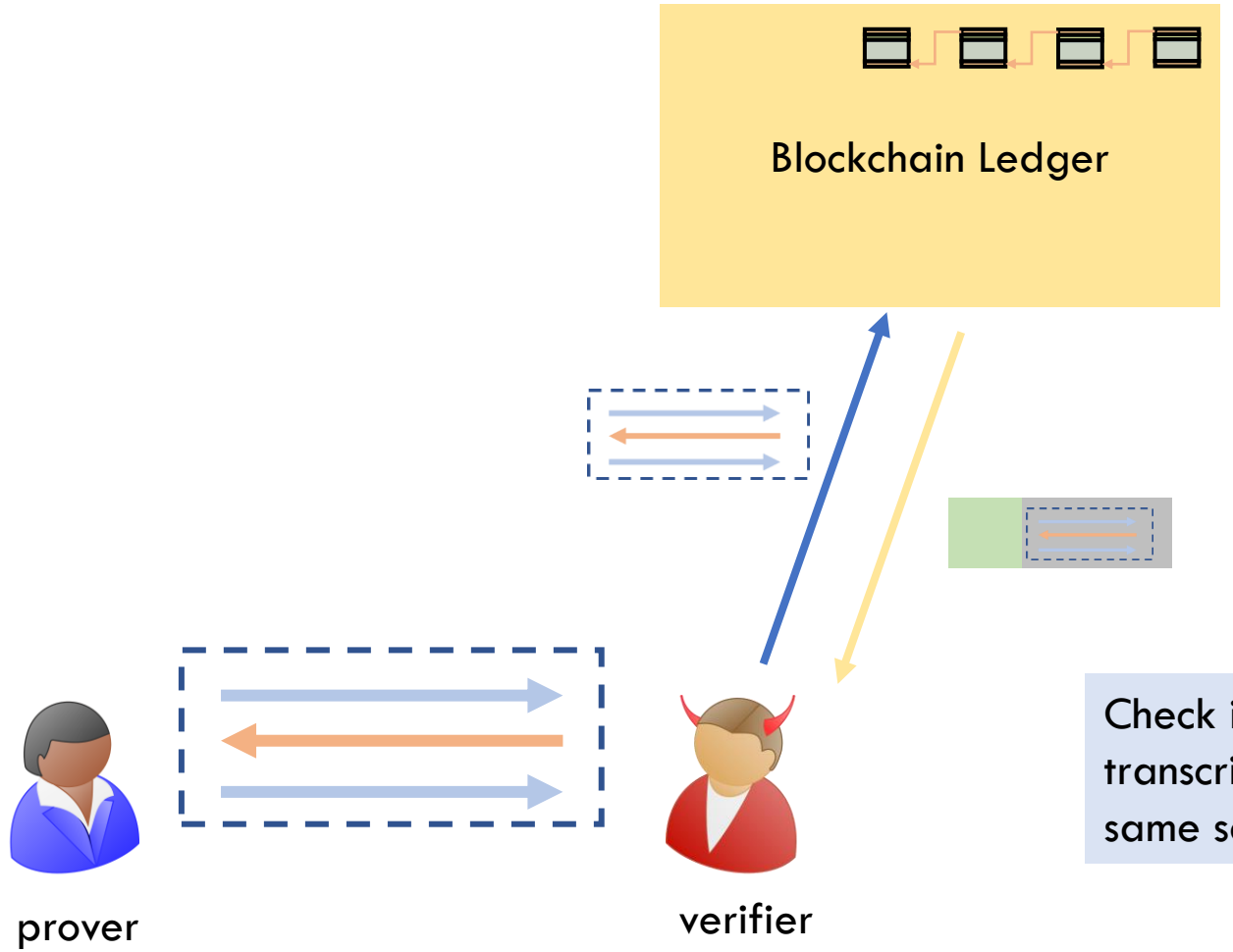
Black-box simulator works by rewinding the cheating verifier.



Black-box ZK Impossible with Blockchain-active verifier

Prevent Simulator from rewinding the verifier.

Black-box simulator works by rewinding the cheating verifier.

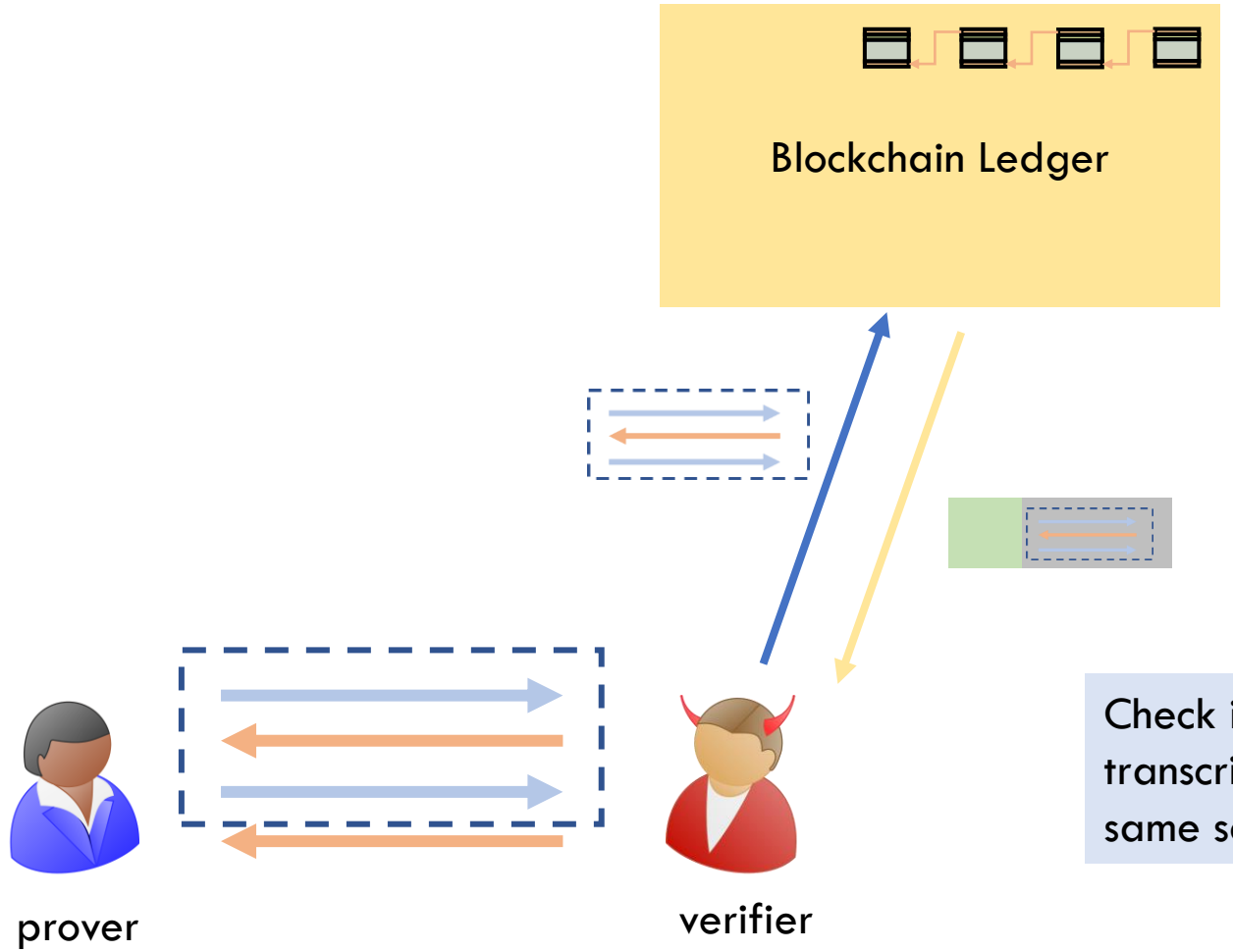


Check if different transcript for the same session.

Black-box ZK Impossible with Blockchain-active verifier

Prevent Simulator from rewinding the verifier.

Black-box simulator works by rewinding the cheating verifier.

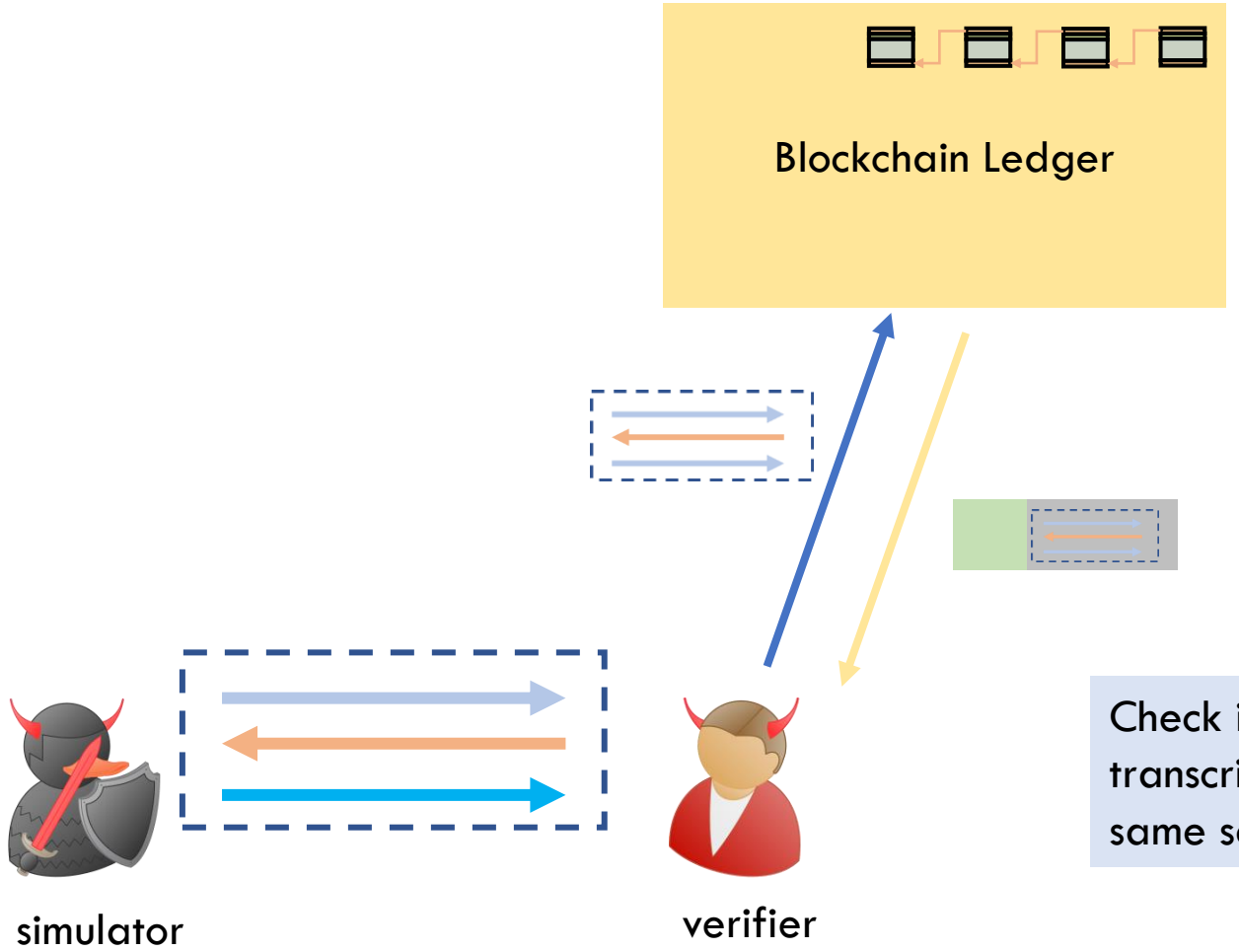


Check if different transcript for the same session.

Black-box ZK Impossible with Blockchain-active verifier

Prevent Simulator from rewinding the verifier.

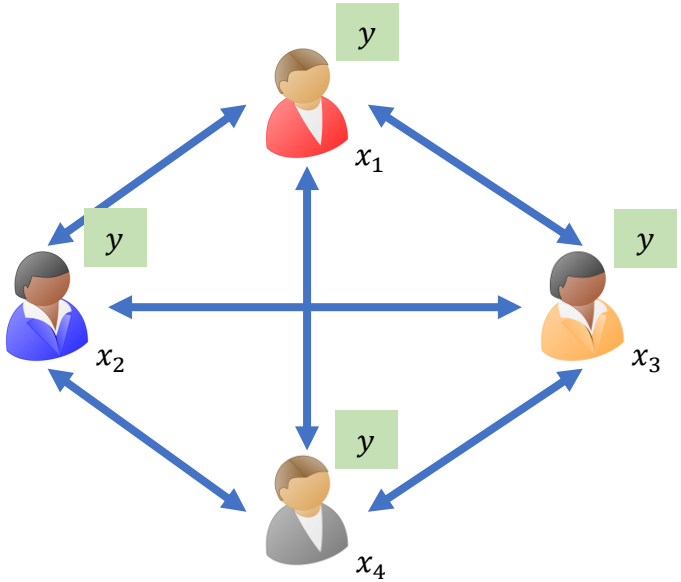
Black-box simulator works by rewinding the cheating verifier.



Check if different transcript for the same session.

Secure Computation

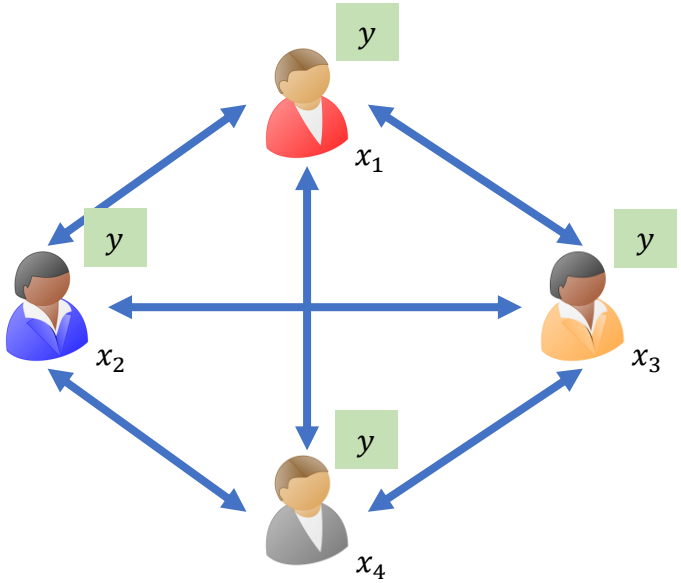
Security



$$y = f(x_1, x_2, x_3, x_4)$$

Security

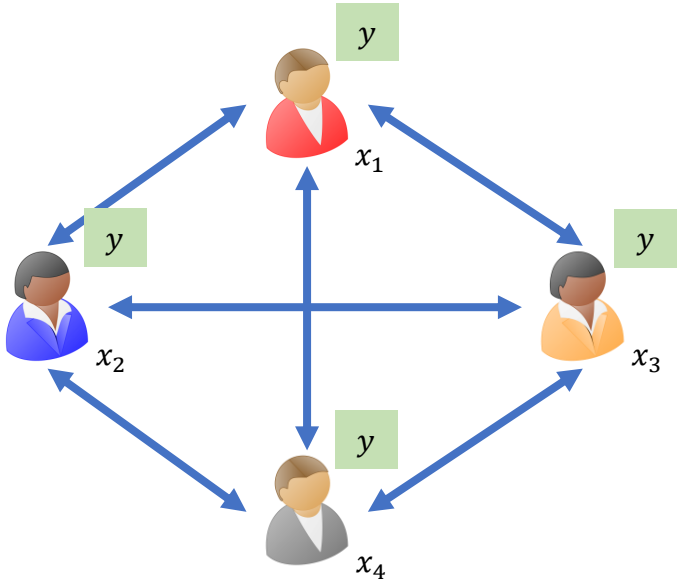
real world



$$y = f(x_1, x_2, x_3, x_4)$$

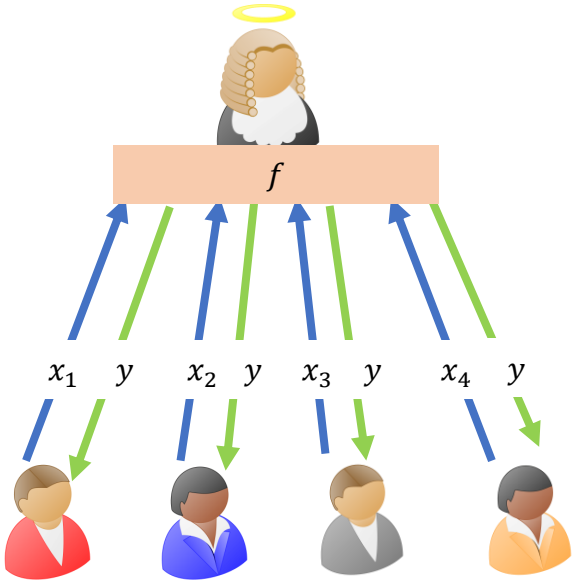
Security

real world



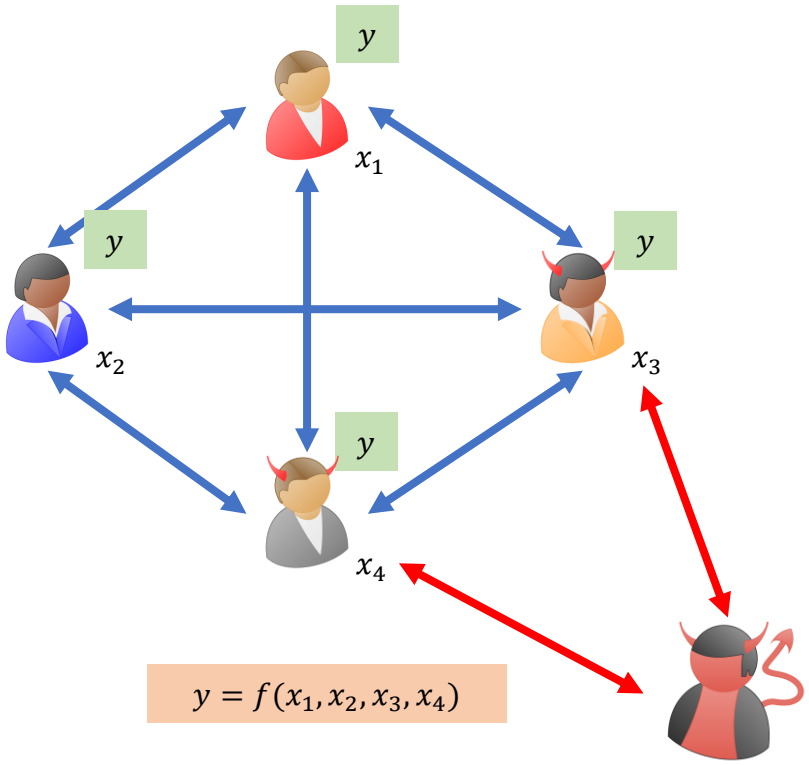
$y = f(x_1, x_2, x_3, x_4)$

ideal world

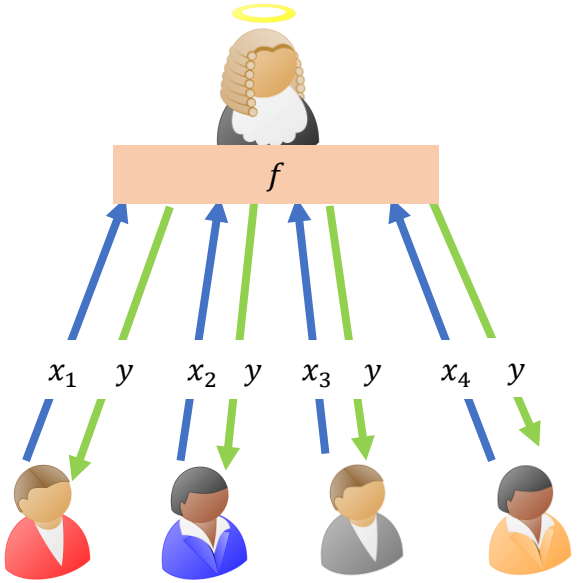


Security

real world

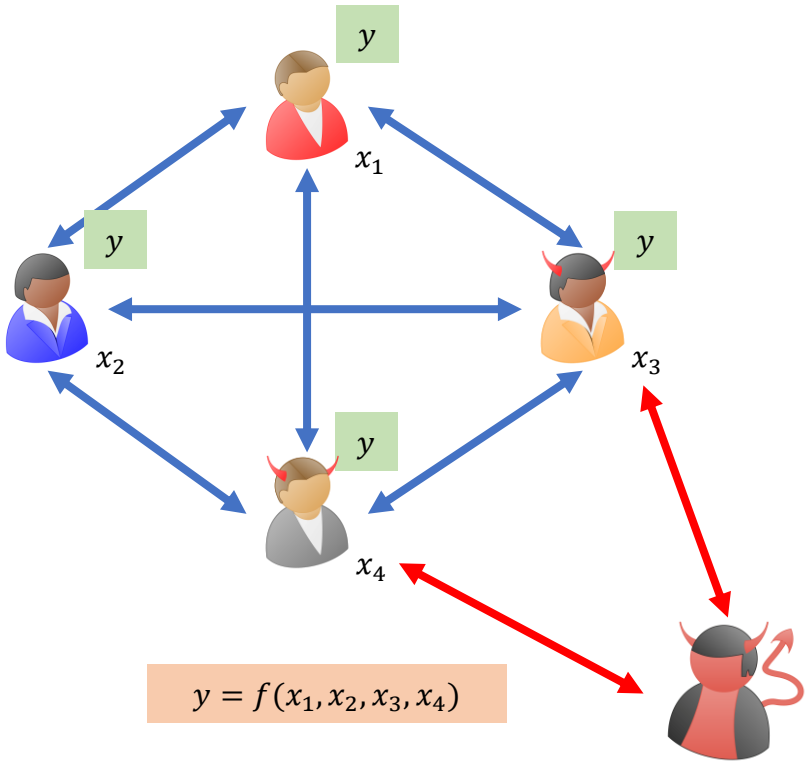


ideal world

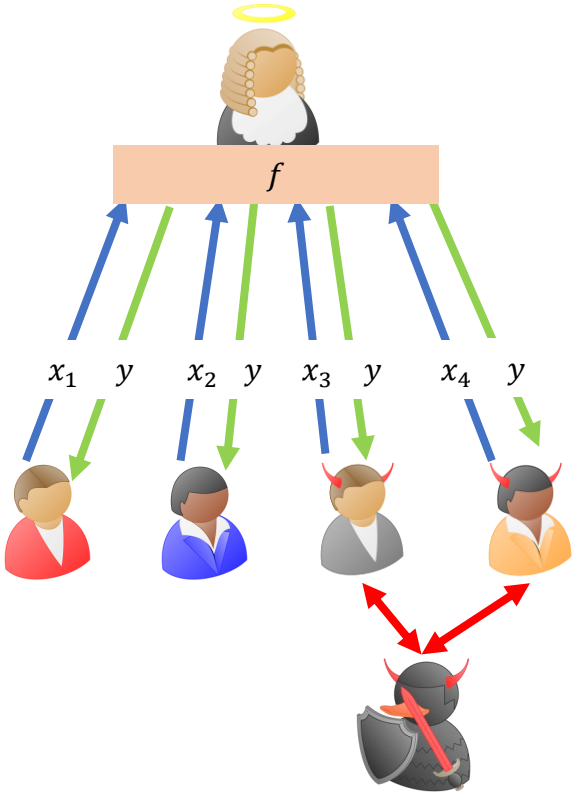


Security

real world



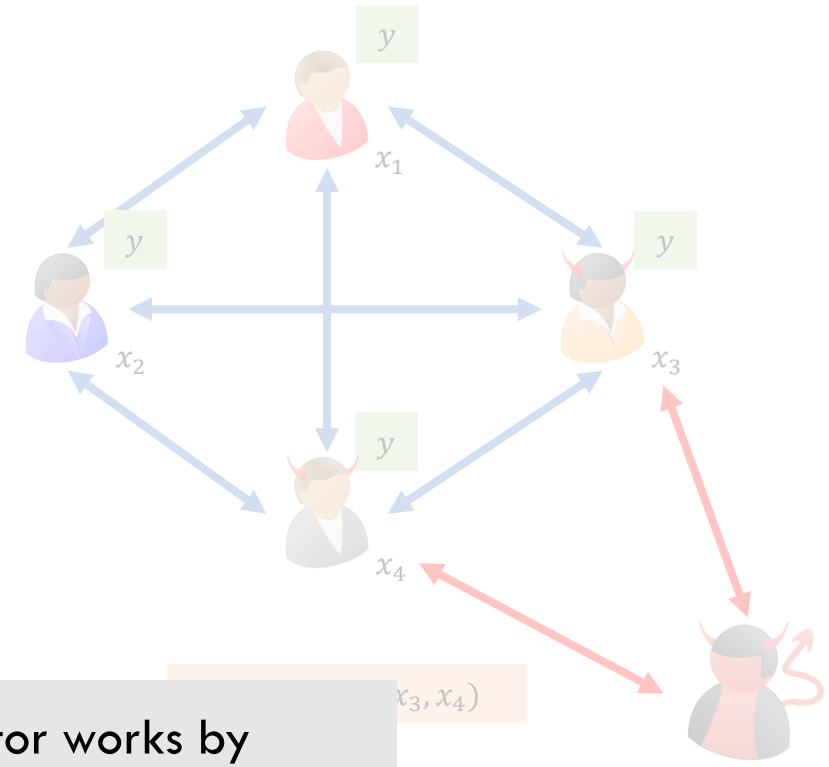
ideal world



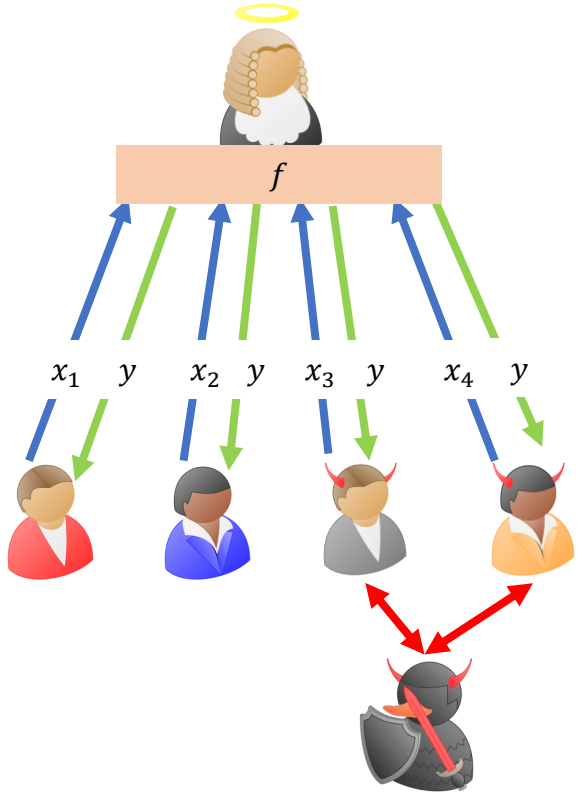
Black-box Simulator

Security

real world



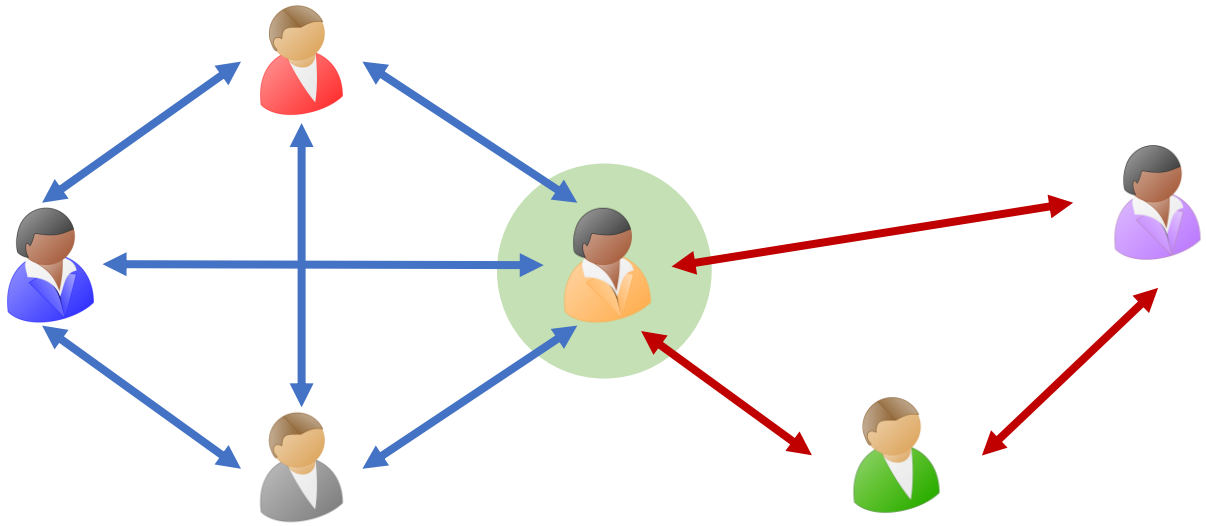
ideal world



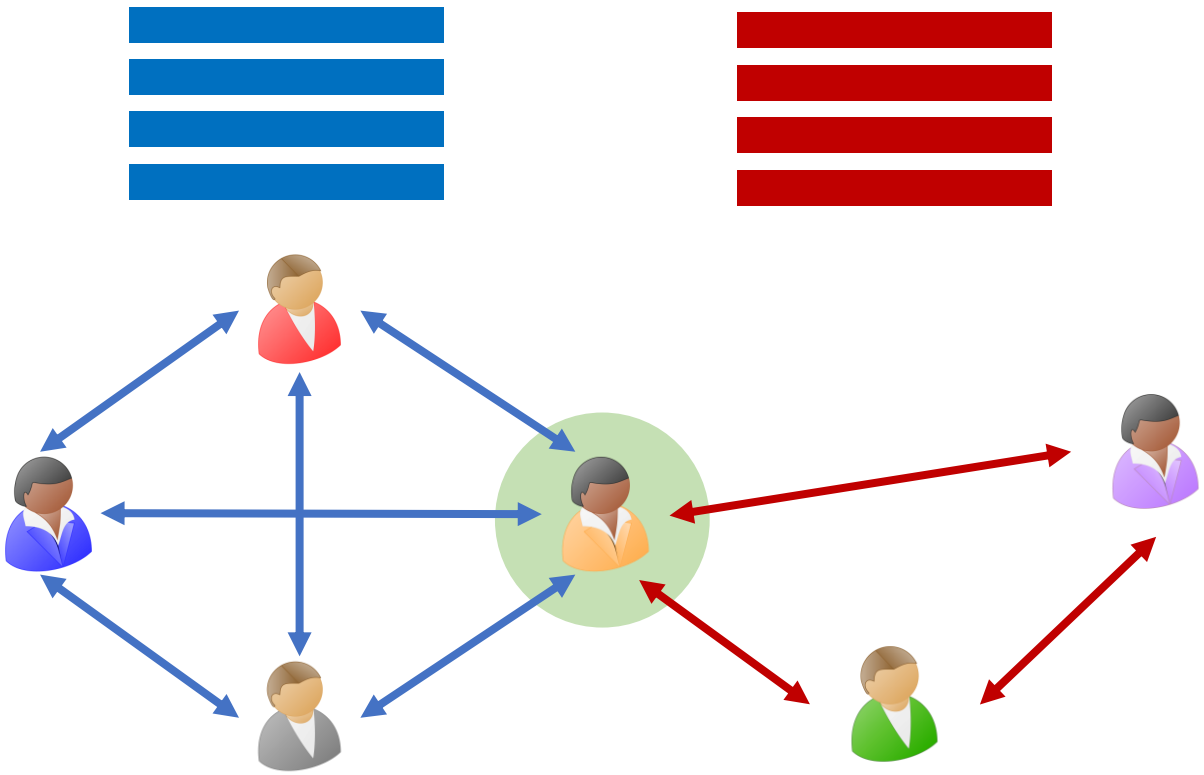
Black-box simulator works by rewinding the misbehaving participants.

Black-box Simulator

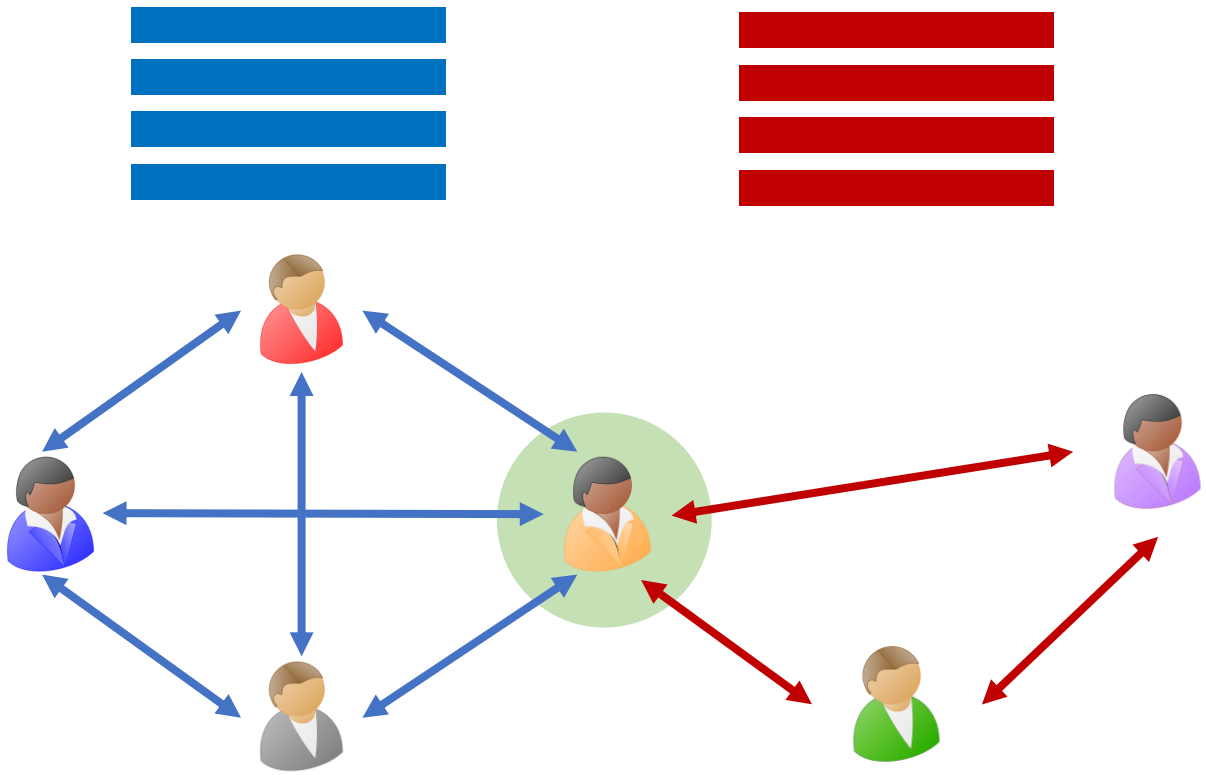
Protocols on the internet

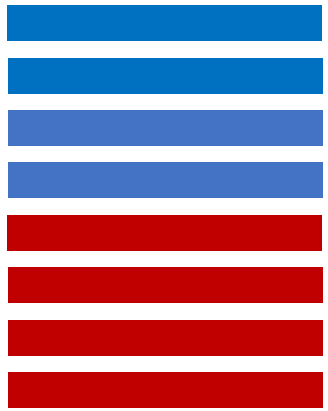


Protocols on the internet

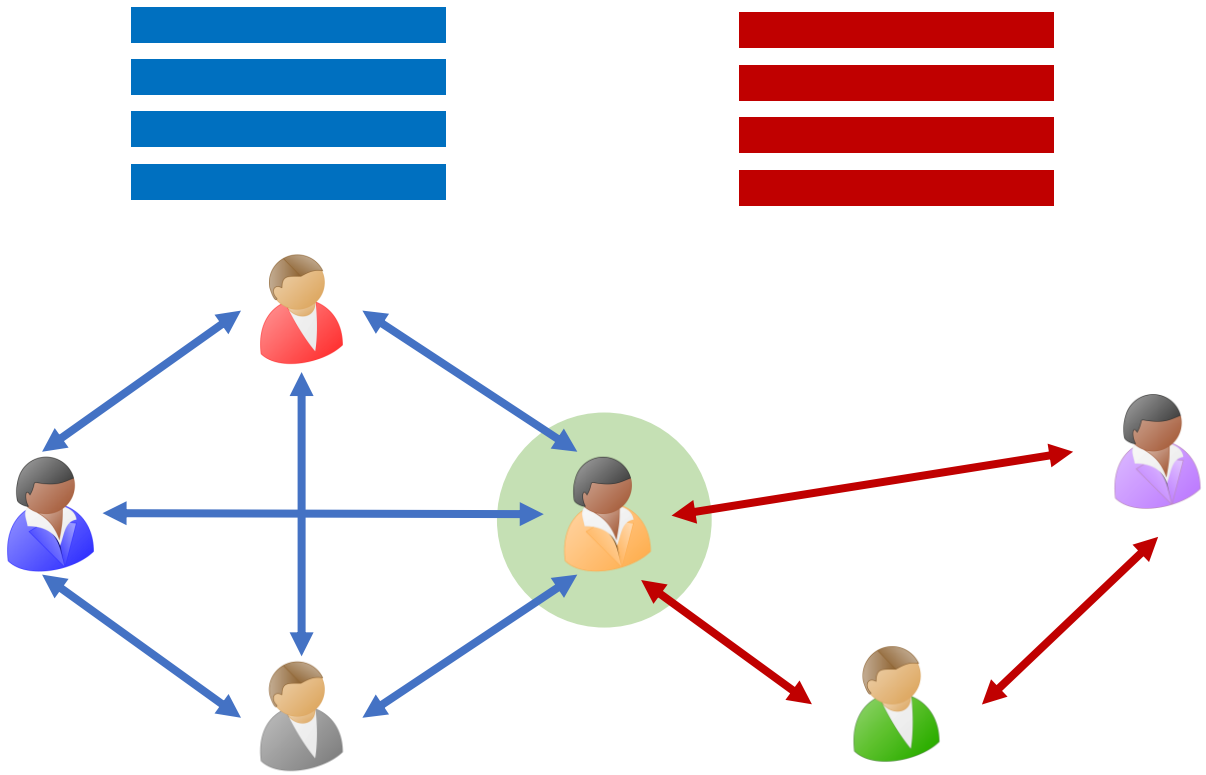


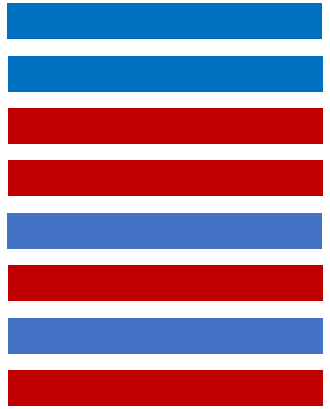
Protocols on the internet





Protocols on the internet





Arbitrary interleaving of protocol messages.

Protocols on the internet

simulator

Arbitrary interleaving of protocol messages.

Protocols on the internet

simulator

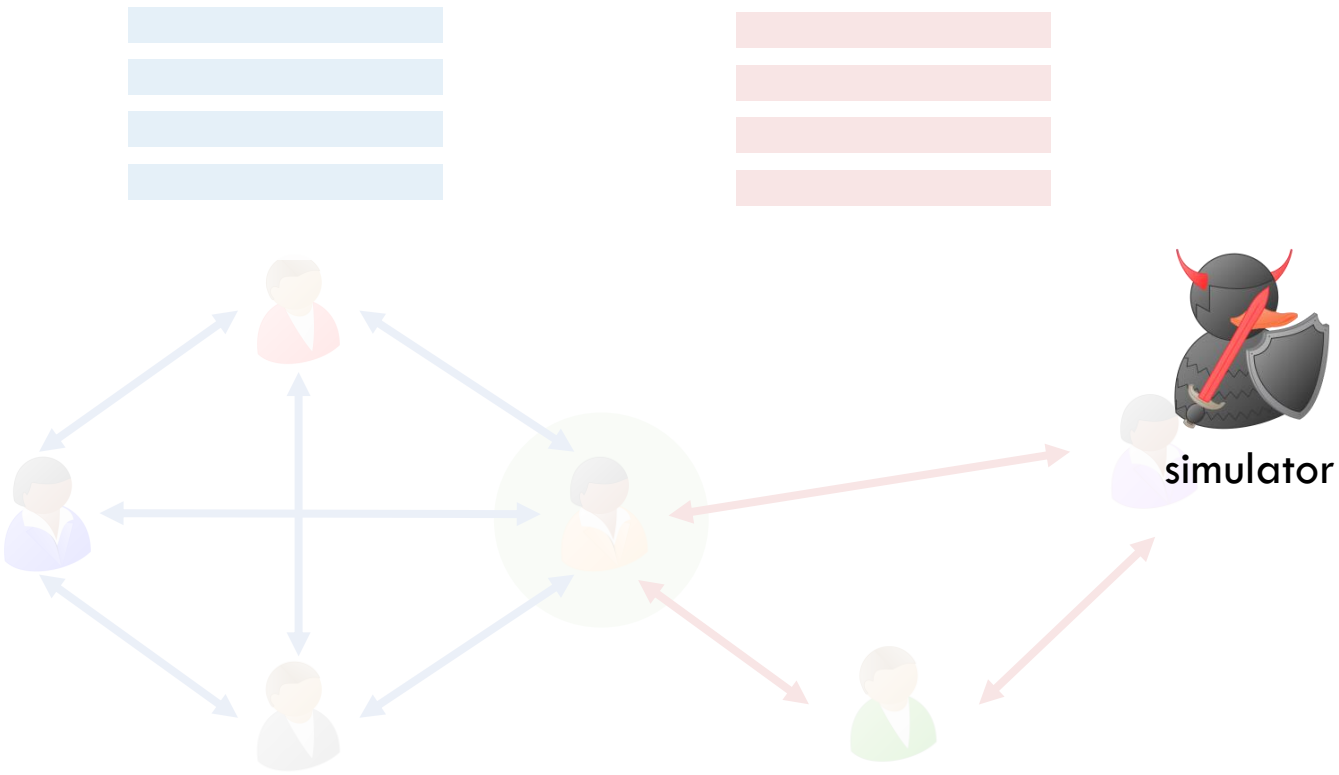
Arbitrary interleaving of protocol messages.

Protocols on the internet

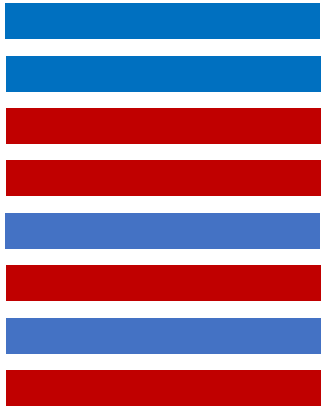
simulator

Arbitrary interleaving of protocol messages.

Protocols on the internet

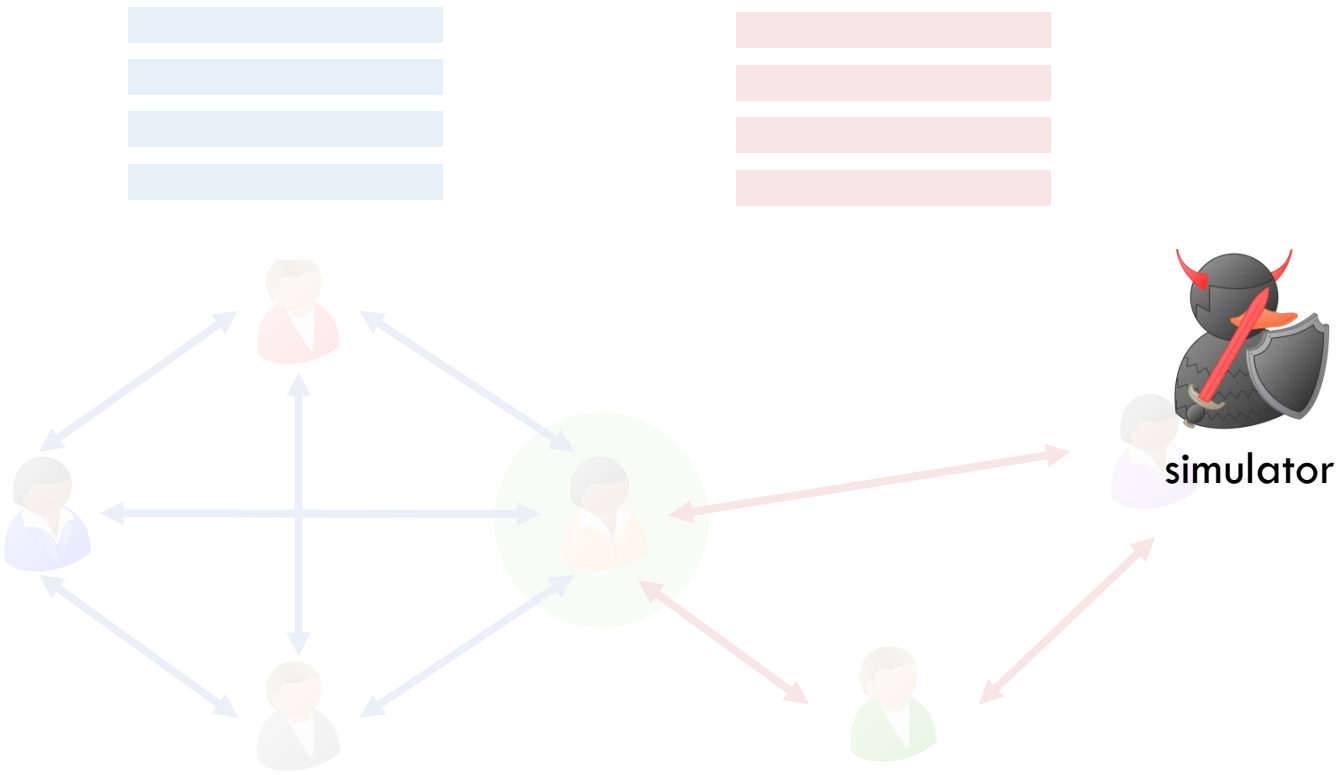


simulator

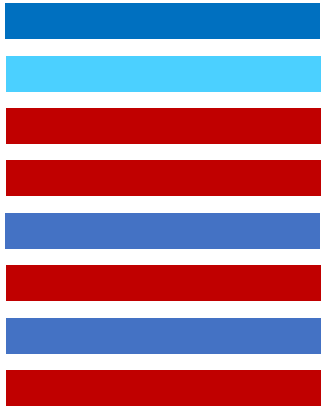


Arbitrary interleaving of protocol messages.

Protocols on the internet

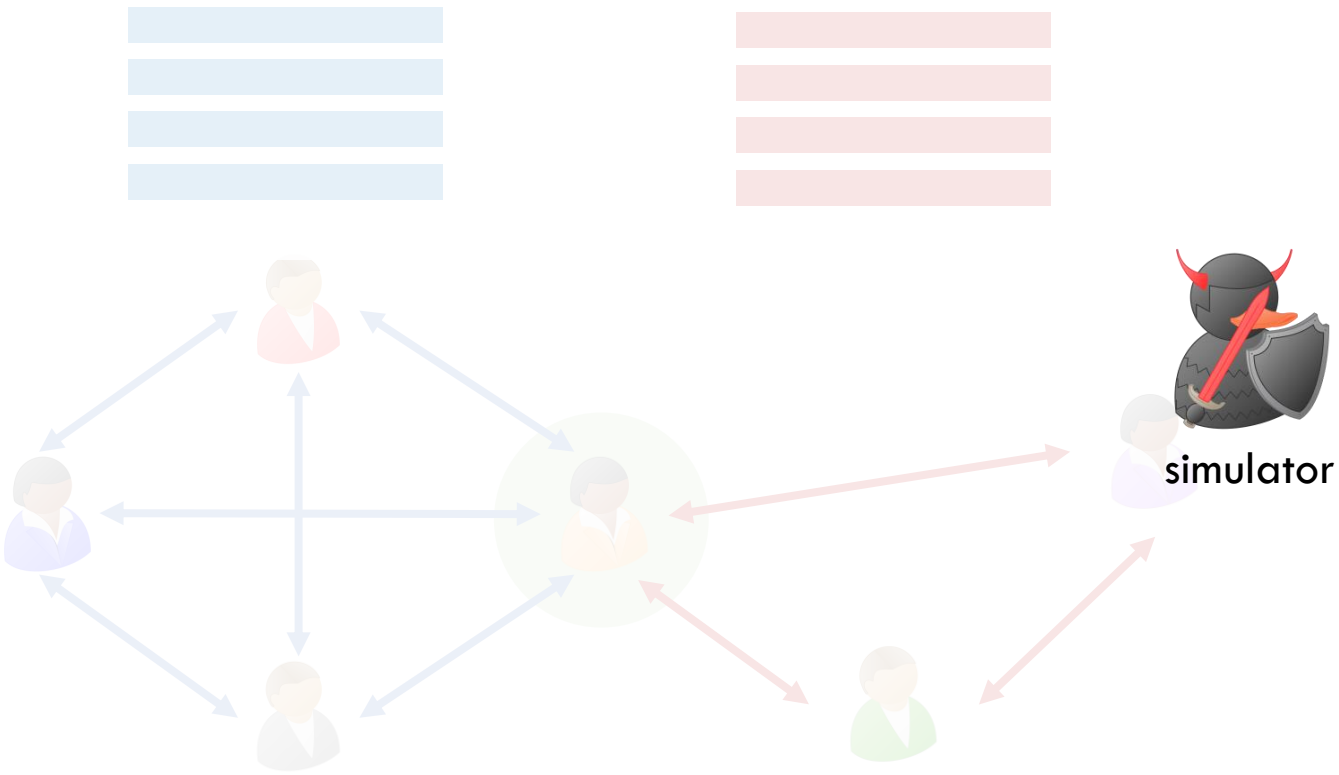


simulator

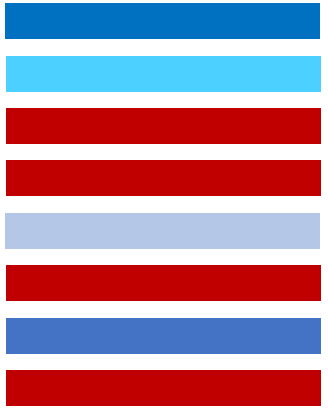


Arbitrary interleaving of protocol messages.

Protocols on the internet

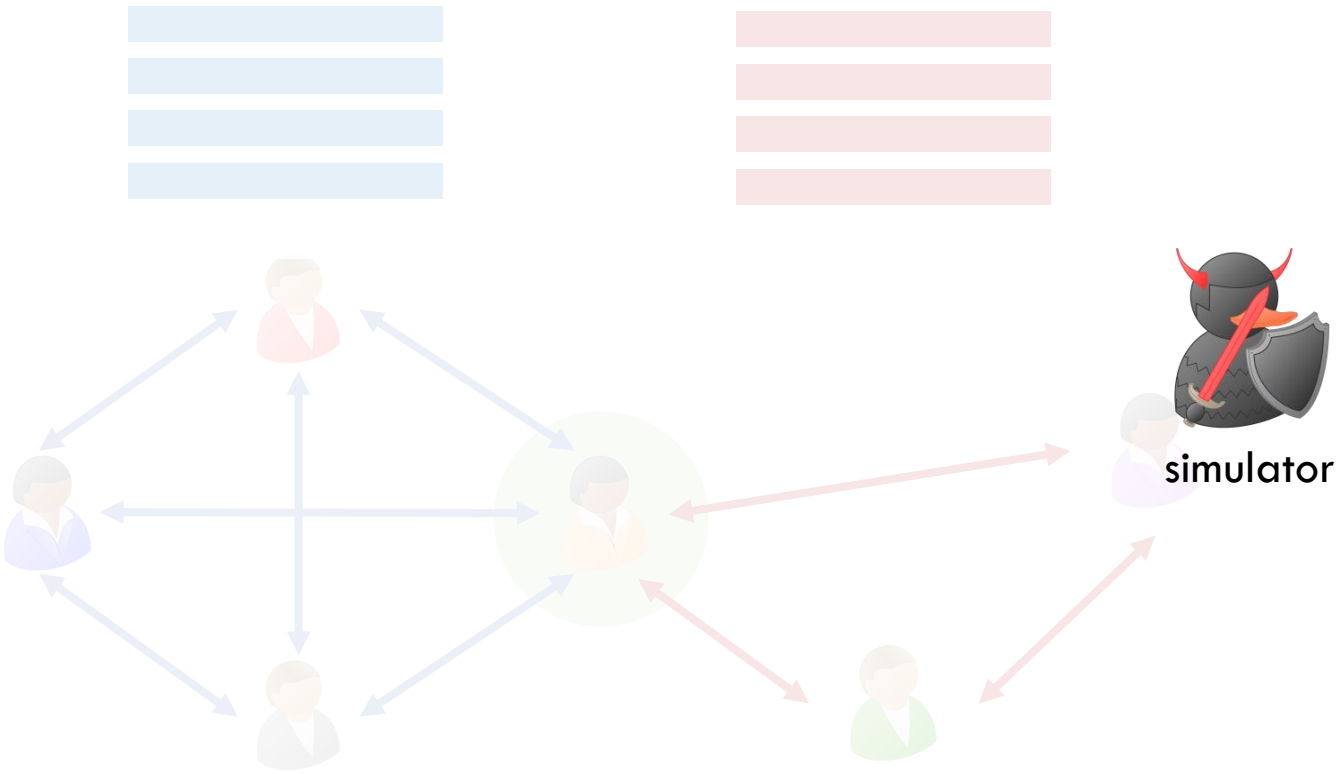


simulator

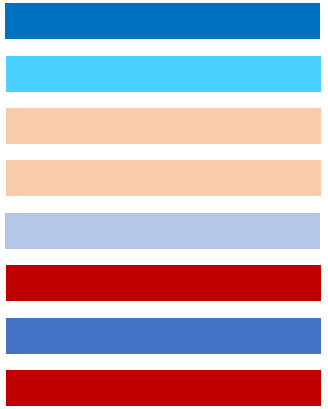


Arbitrary interleaving of protocol messages.

Protocols on the internet

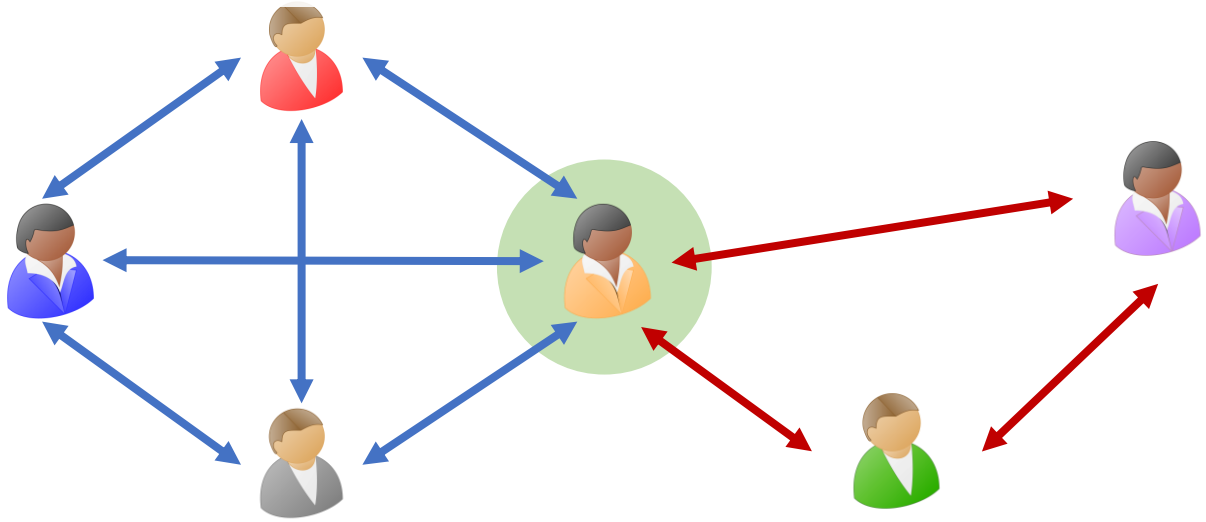


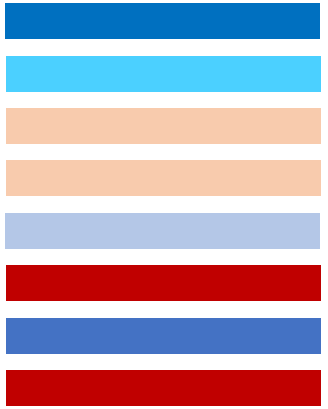
simulator



Arbitrary interleaving of protocol messages.

Protocols on the internet



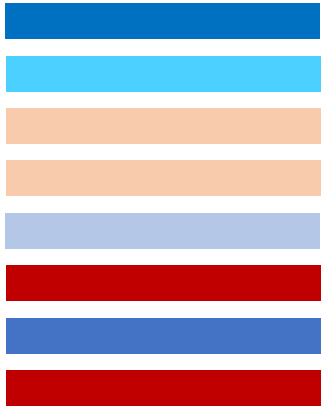


Arbitrary interleaving of protocol messages.

 can change their protocol inputs on the right on being rewound.

Protocols on the internet

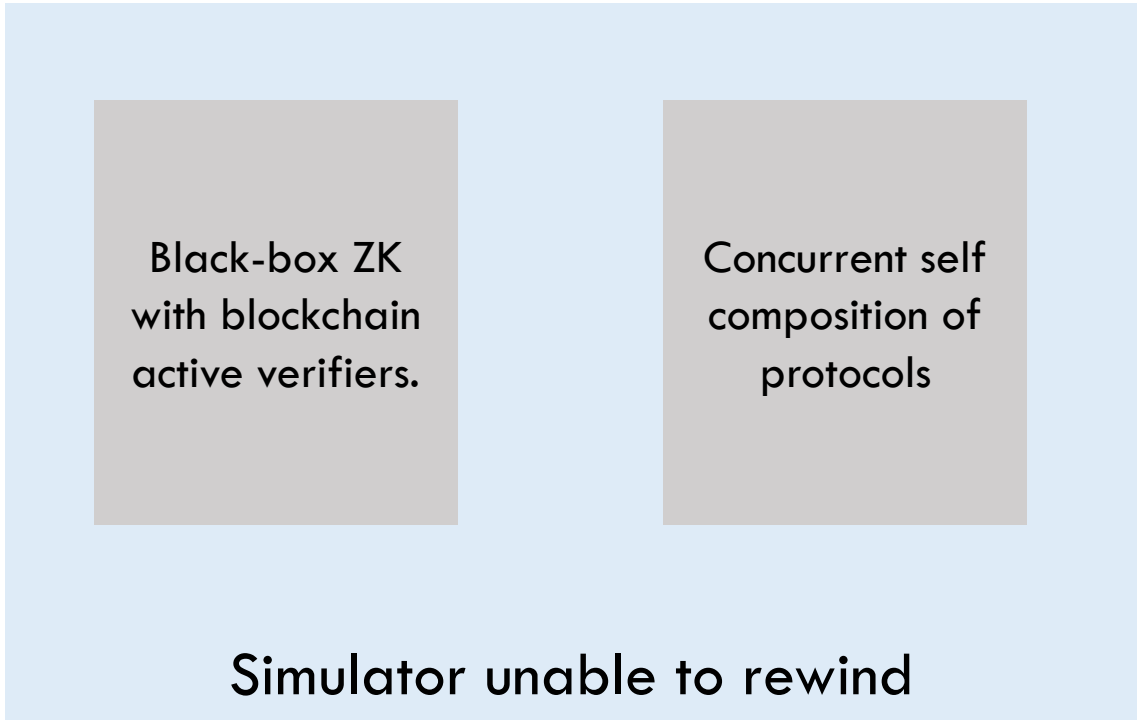
Black-box impossible [Lindell'04]



Arbitrary interleaving of protocol messages.

can change their protocol inputs on the right on being rewound.

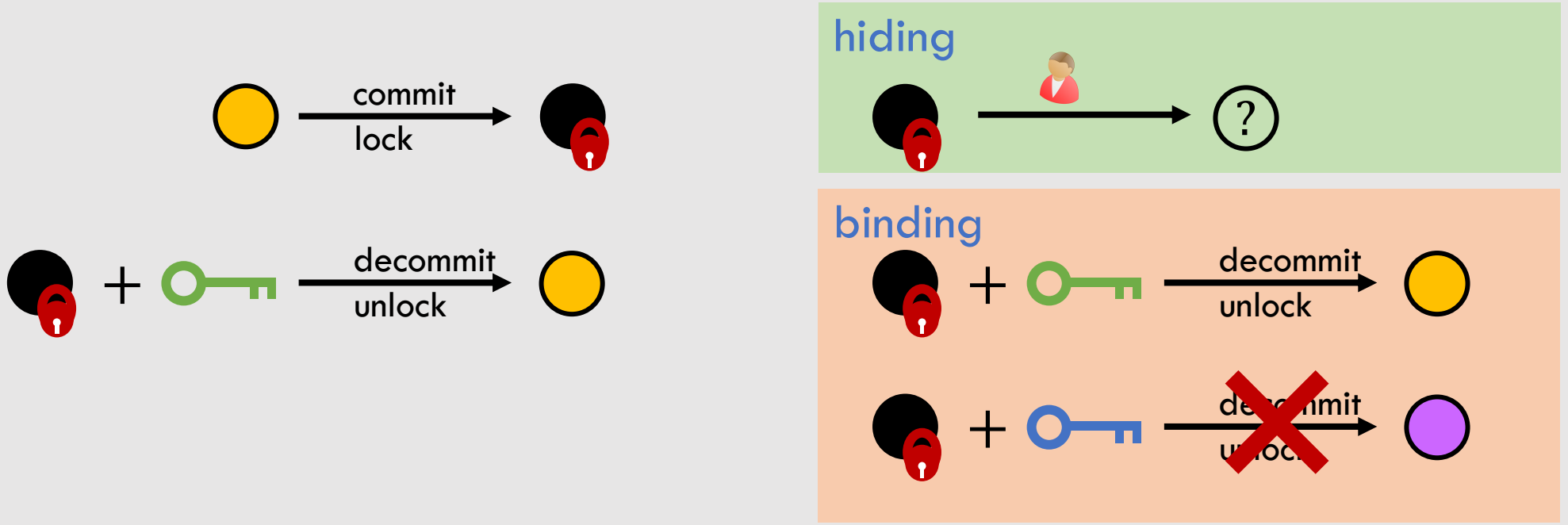
Rewinding Issues



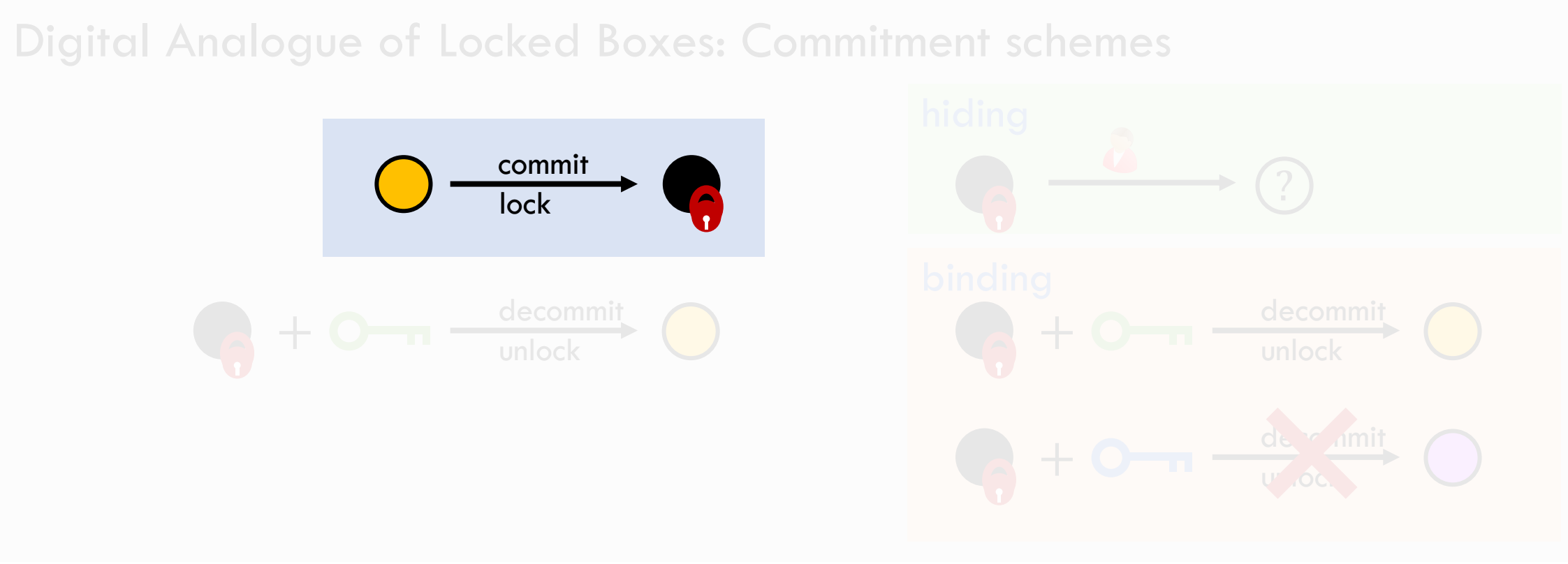
Extractable Commitments Blockchain Hybrid Model

Extractable Commitments Blockchain Hybrid Model

Digital Analogue of Locked Boxes: Commitment schemes

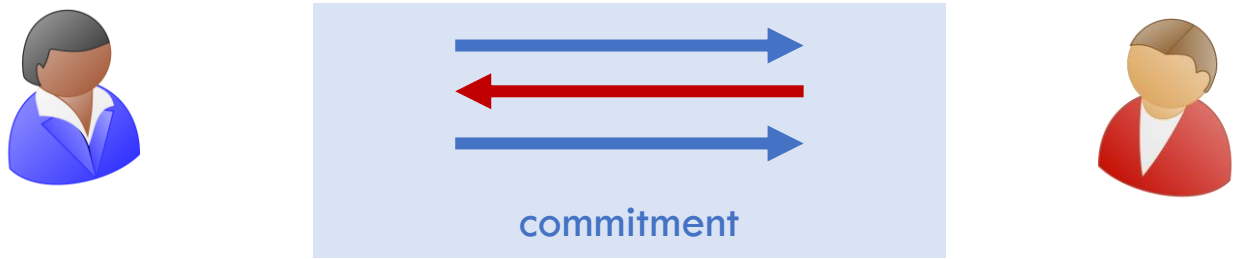


Extractable Commitments Blockchain Hybrid Model



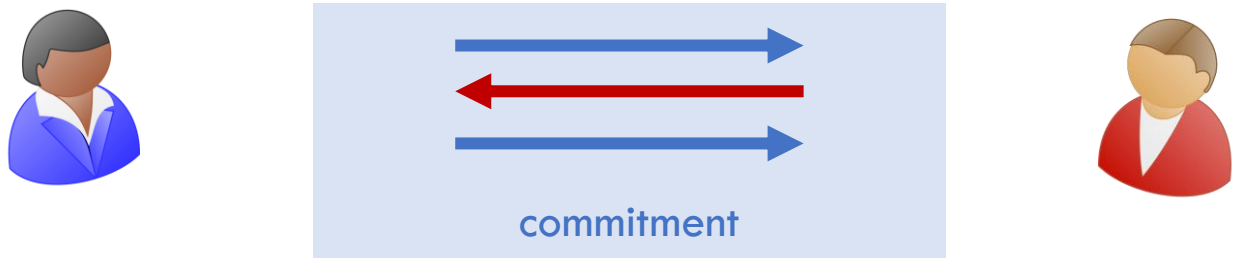
Extractable Commitments Blockchain Hybrid Model

Extractable Commitments Blockchain Hybrid Model



Extractable Commitments Blockchain Hybrid Model

[Prabhakaran-Rosen-Sahai'02]

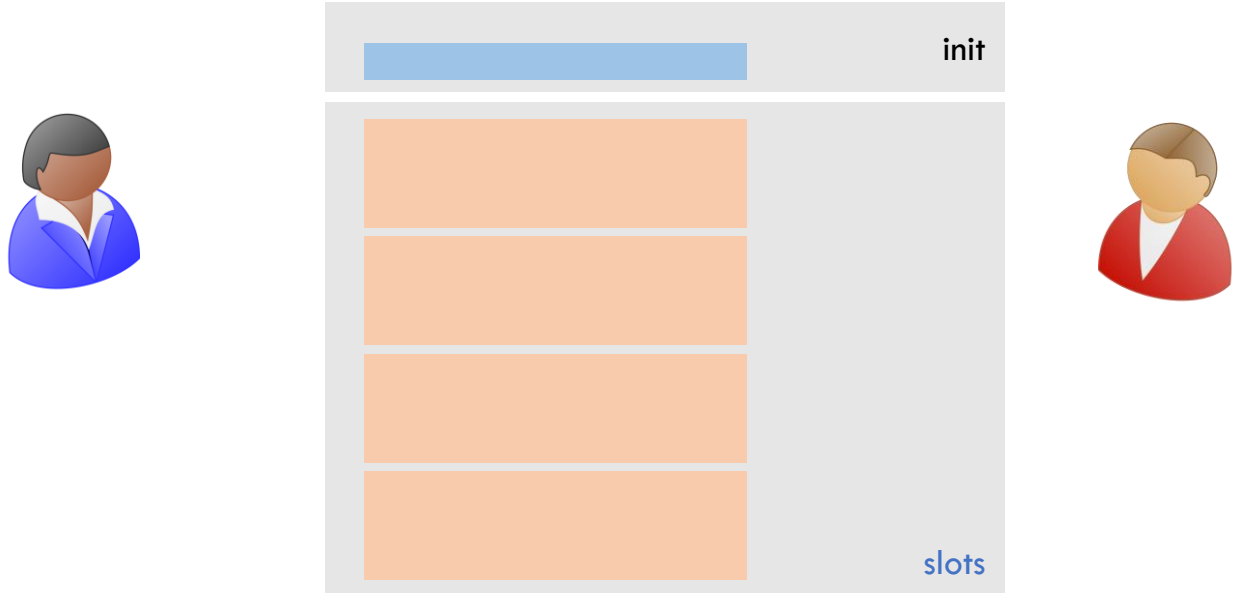


Extractable Commitments Blockchain Hybrid Model

[Prabhakaran-Rosen-Sahai'02]

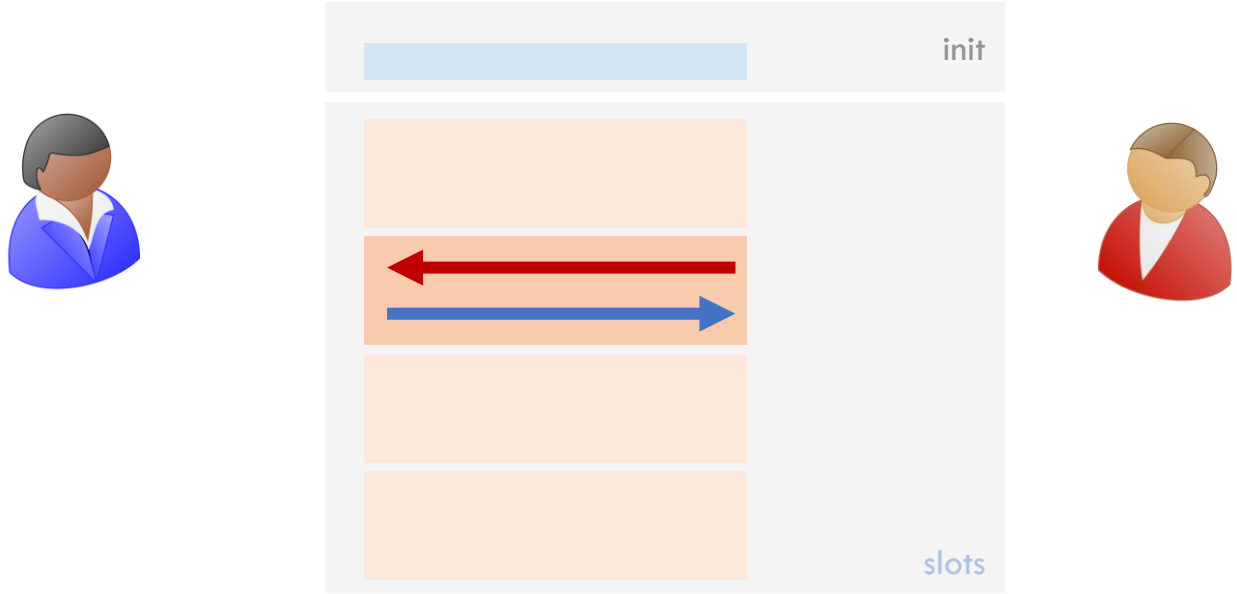
Extractable Commitments Blockchain Hybrid Model

[Prabhakaran-Rosen-Sahai'02]



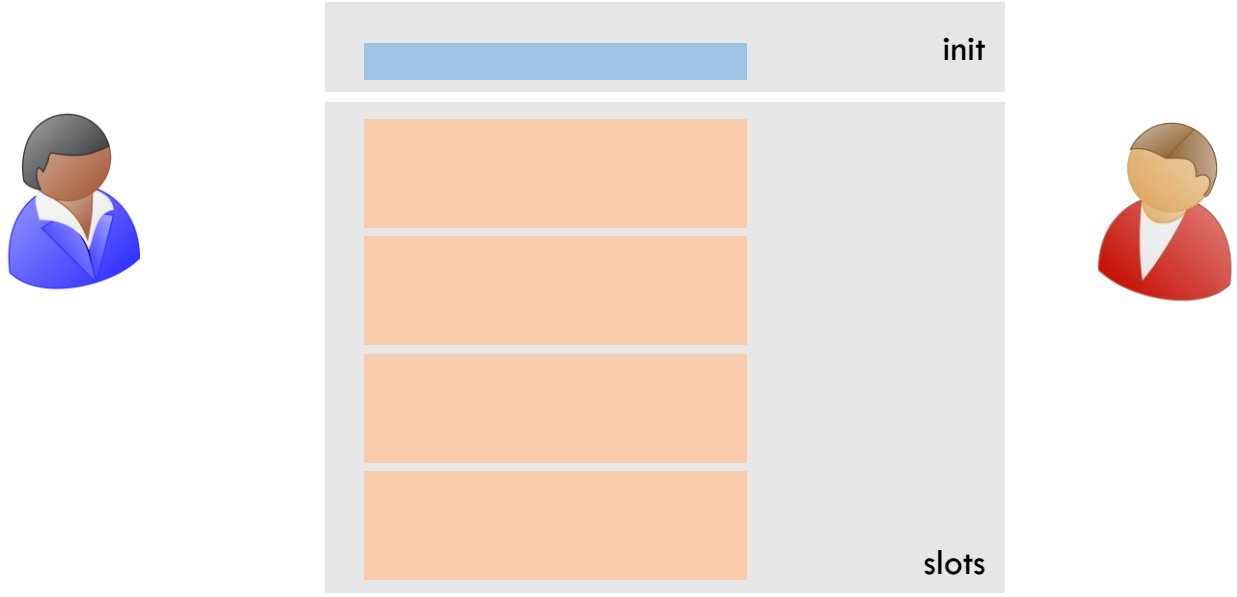
Extractable Commitments Blockchain Hybrid Model

[Prabhakaran-Rosen-Sahai'02]



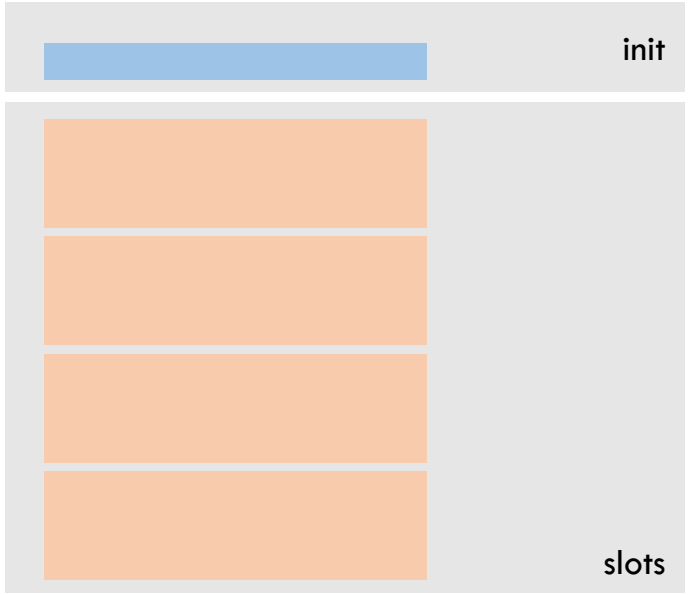
Extractable Commitments Blockchain Hybrid Model

[Prabhakaran-Rosen-Sahai'02]



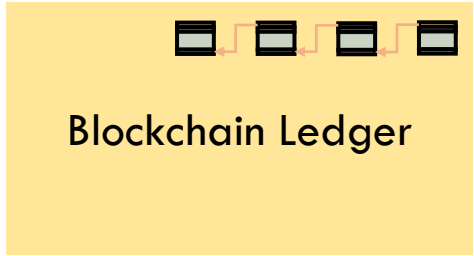
Extractable Commitments Blockchain Hybrid Model

[Prabhakaran-Rosen-Sahai'02]



If the simulator can rewind in any one of the slots, then the simulator can extract the committed value.

Extractable Commitments Blockchain Hybrid Model



Extractable Commitments Blockchain Hybrid Model

Blockchain Ledger

Use the blockchain as a coarse timer.

Extractable Commitments Blockchain Hybrid Model

Blockchain Ledger

Use the blockchain as a coarse timer.

Time-out k

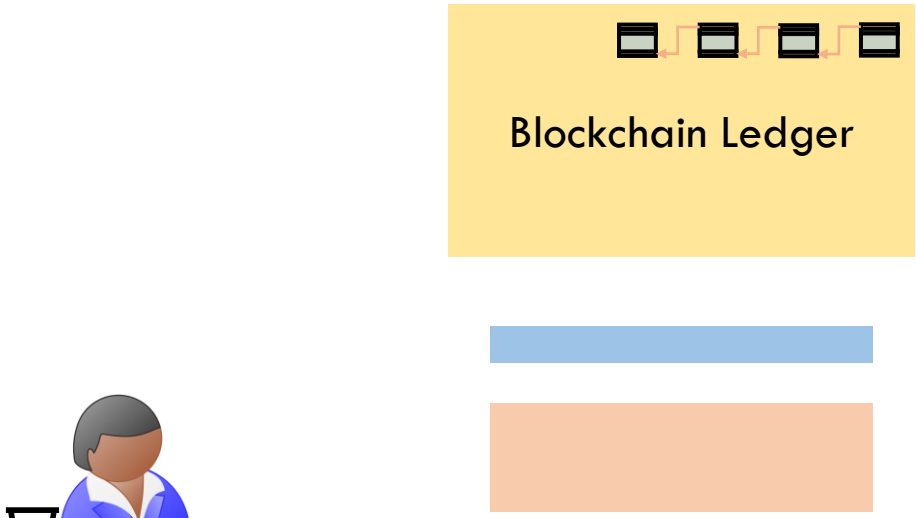
Extractable Commitments Blockchain Hybrid Model

Blockchain Ledger

Use the blockchain as a coarse timer.

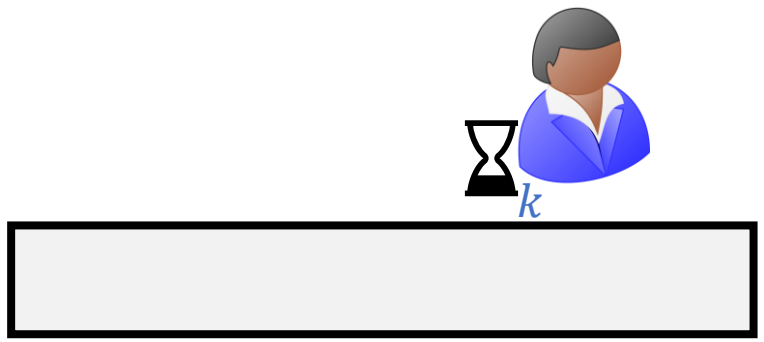
Time-out k

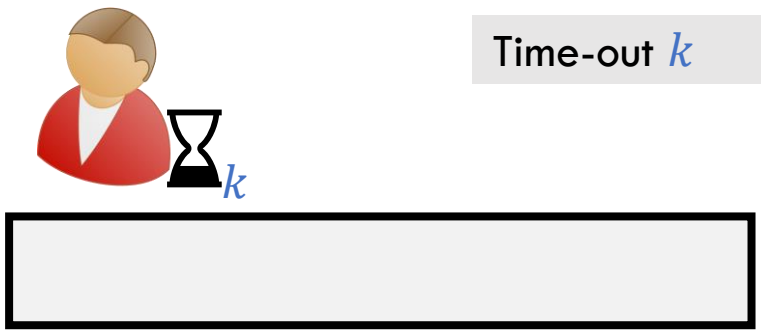
Extractable Commitments Blockchain Hybrid Model



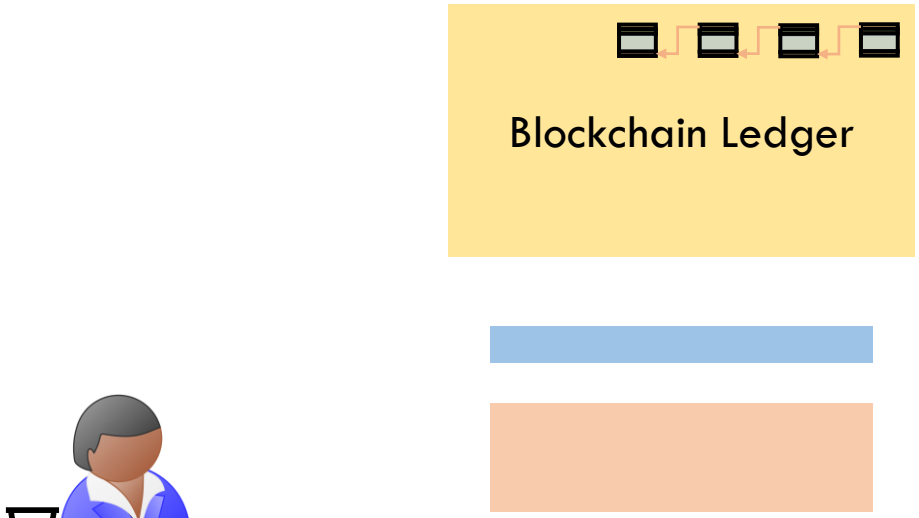
Use the blockchain as a coarse timer.

Time-out k



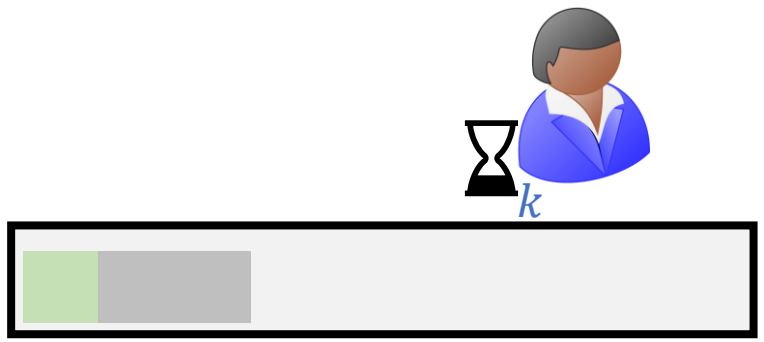


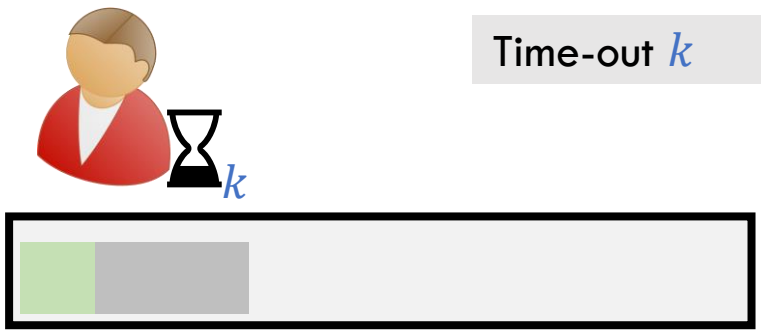
Extractable Commitments Blockchain Hybrid Model



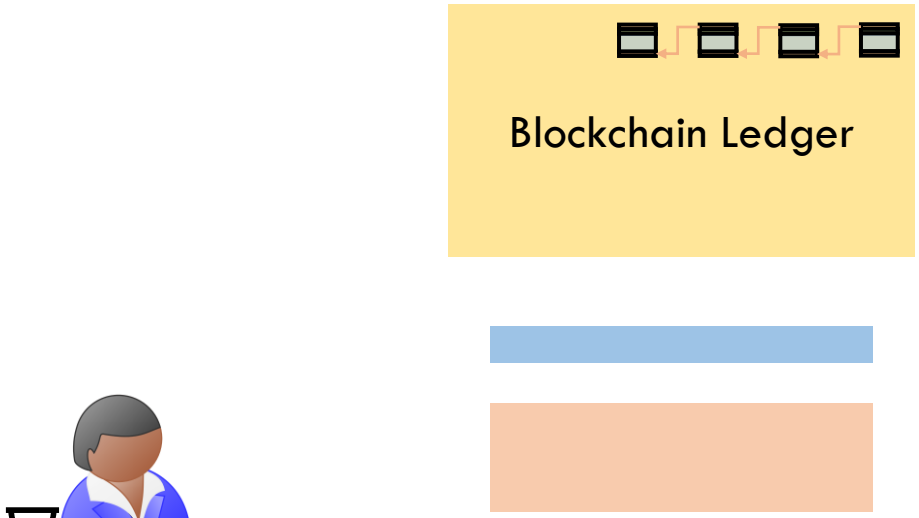
Use the blockchain as a coarse timer.

Time-out k



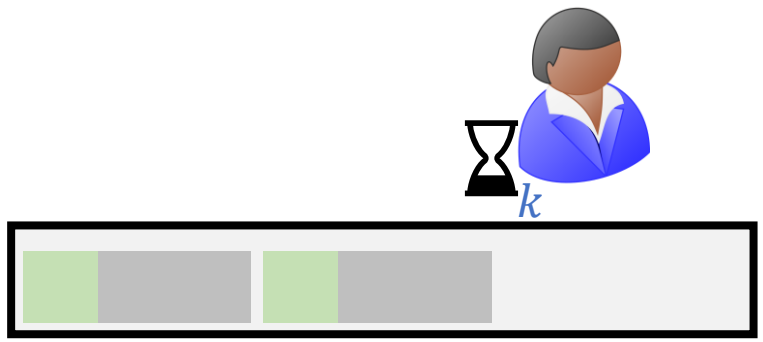


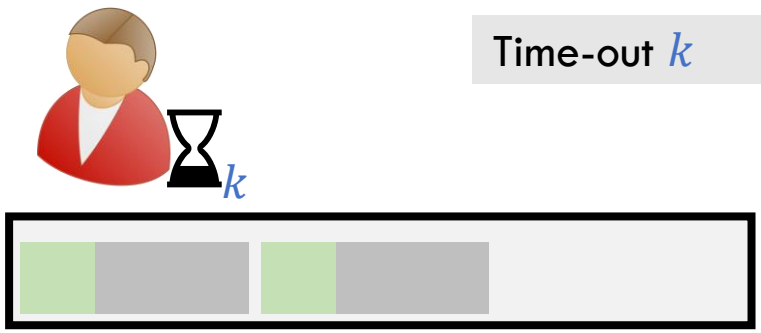
Extractable Commitments Blockchain Hybrid Model



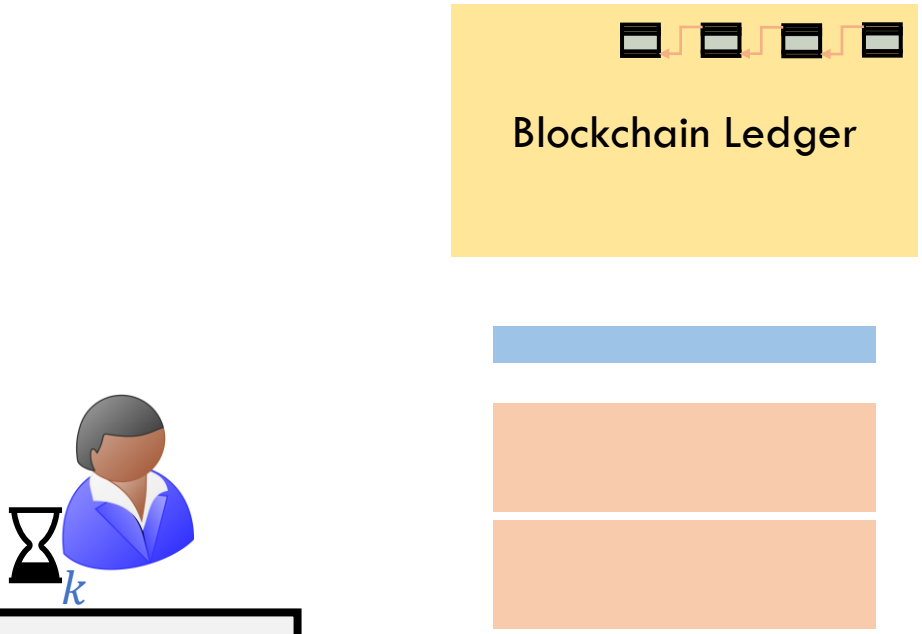
Use the blockchain as a coarse timer.

Time-out k



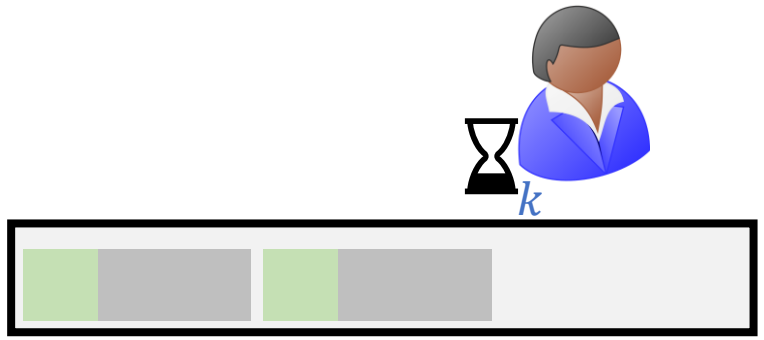


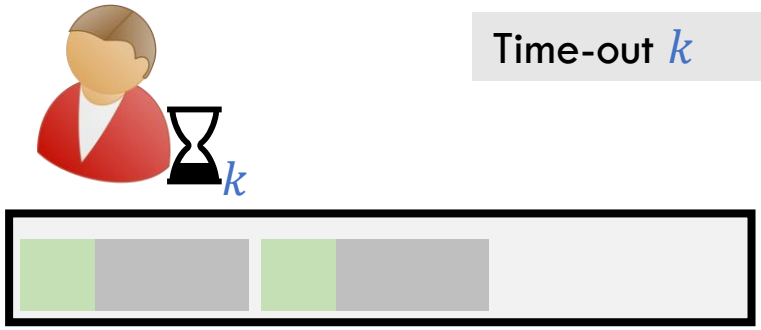
Extractable Commitments Blockchain Hybrid Model



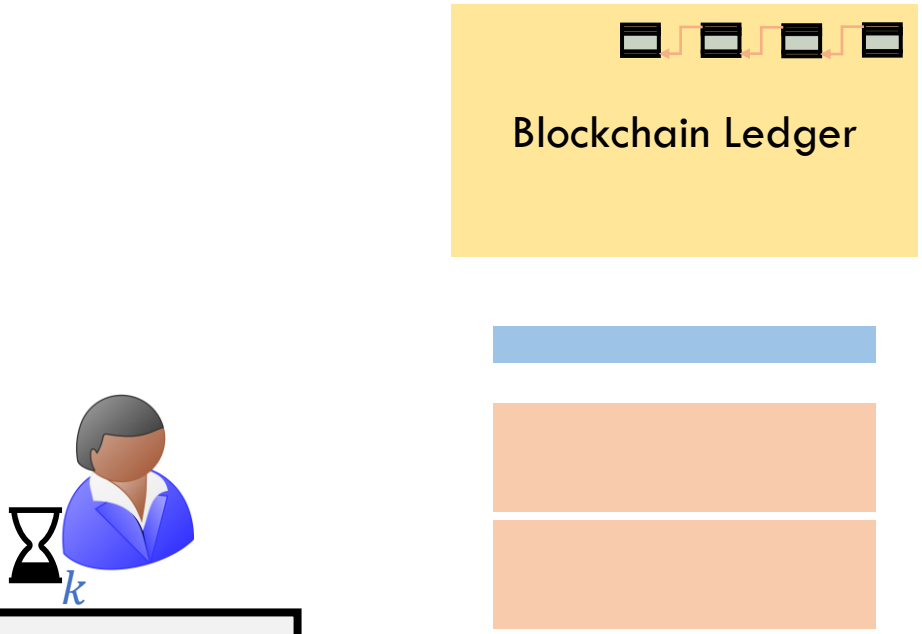
Use the blockchain as a coarse timer.

Time-out k



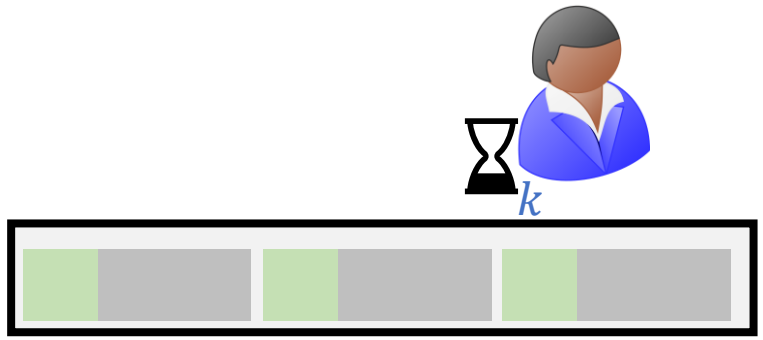


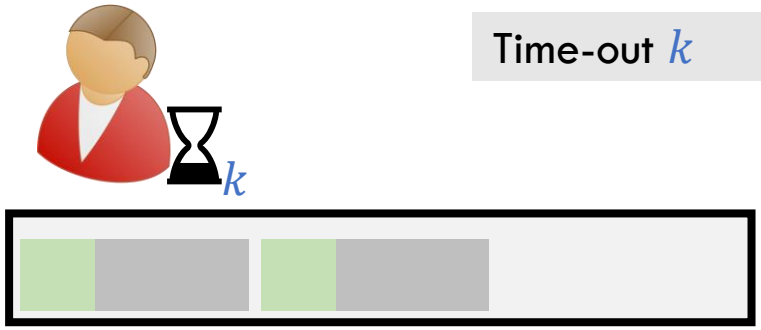
Extractable Commitments Blockchain Hybrid Model



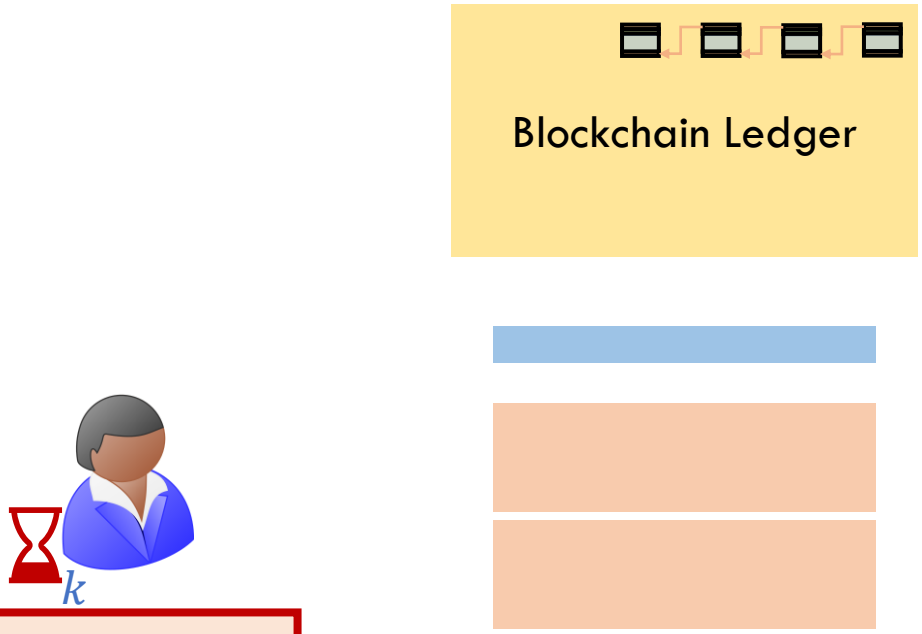
Use the blockchain as a coarse timer.

Time-out k



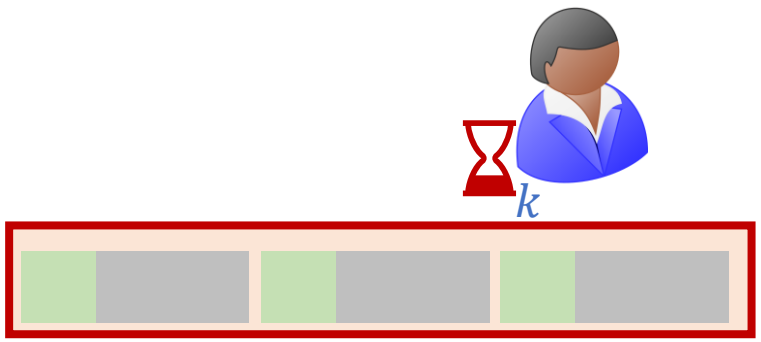


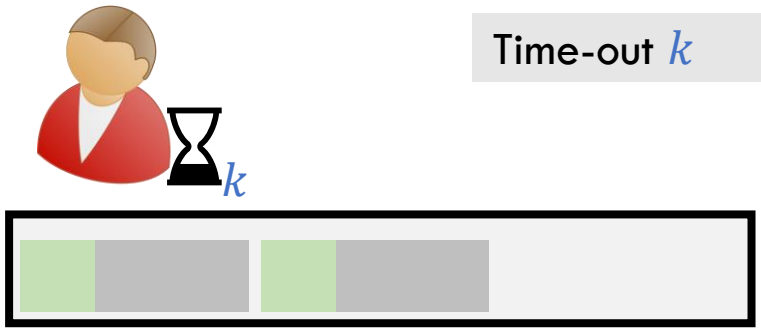
Extractable Commitments Blockchain Hybrid Model



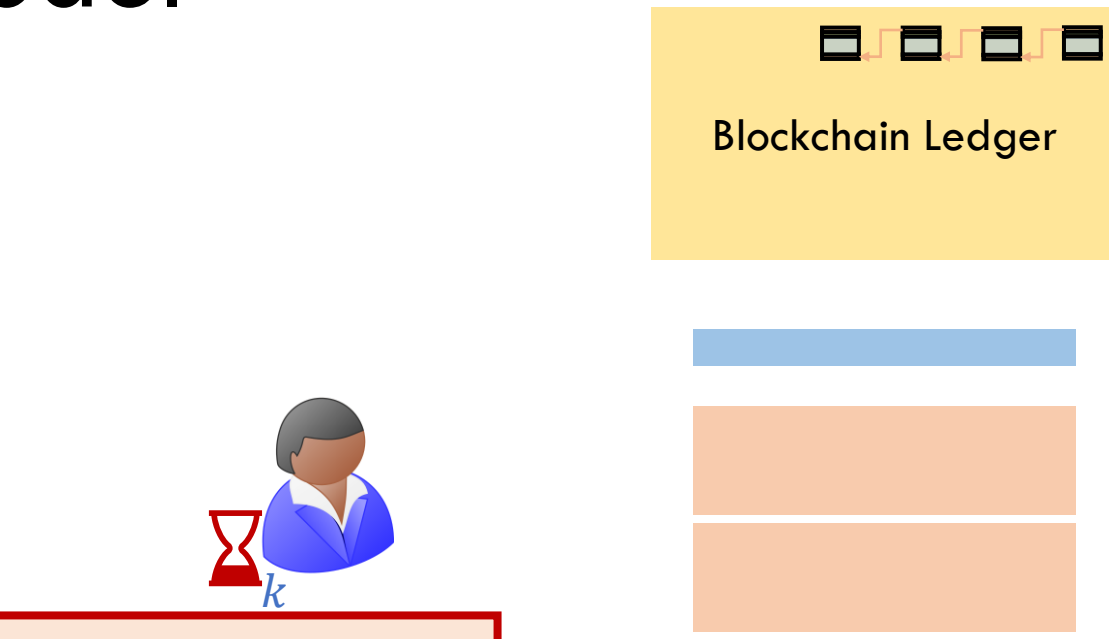
Use the blockchain as a coarse timer.

Time-out k



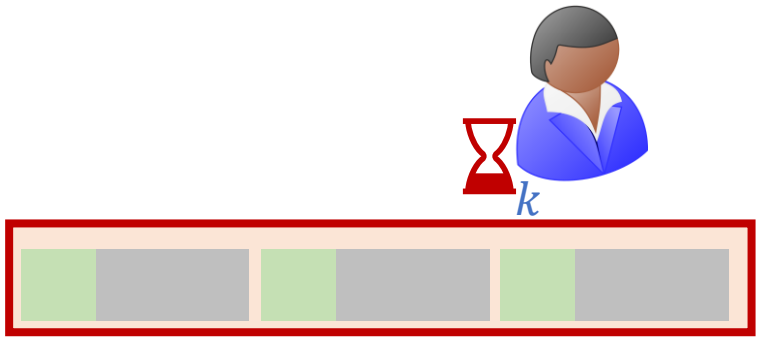


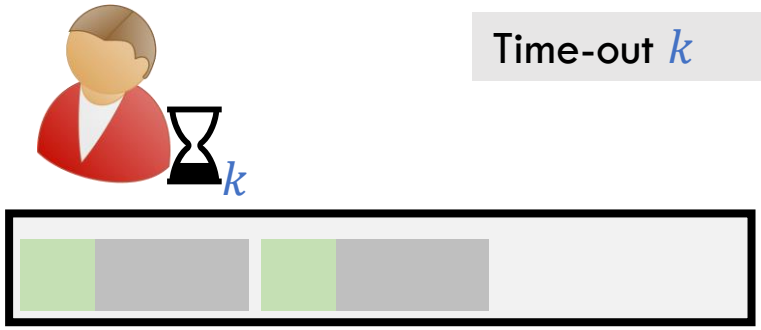
Extractable Commitments Blockchain Hybrid Model



Use the blockchain as a coarse timer.

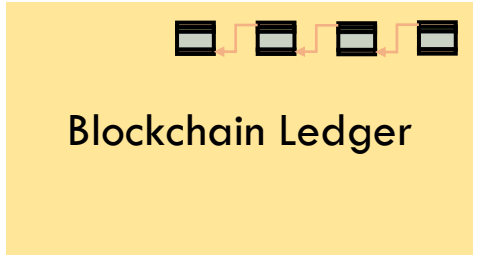
Time-out k

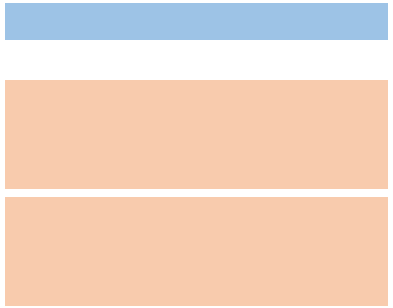


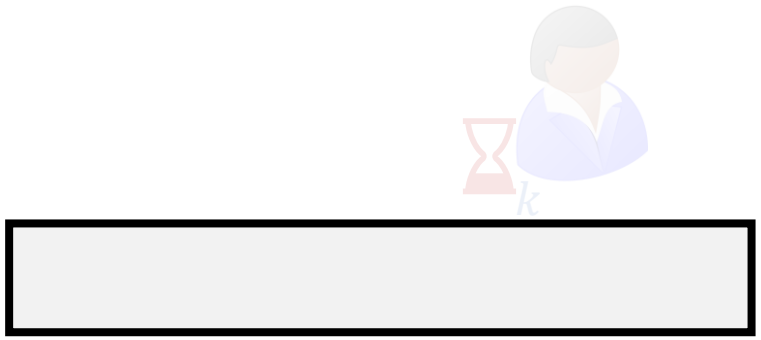


quits protocol if timer runs out.

Extractable Commitments Blockchain Hybrid Model





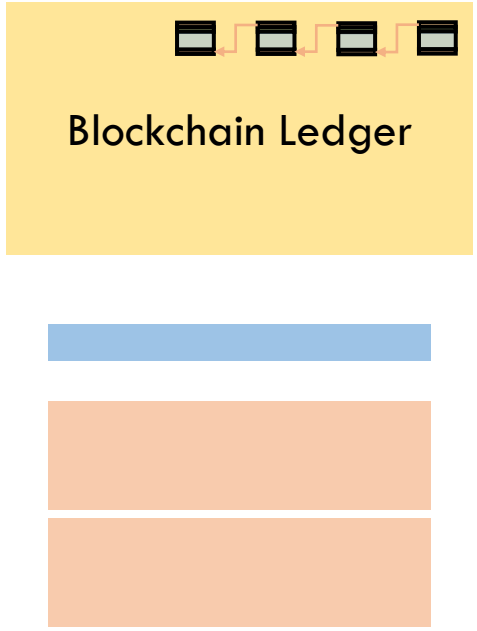


Use the blockchain as a coarse timer.

Time-out k

quits protocol if timer runs out.

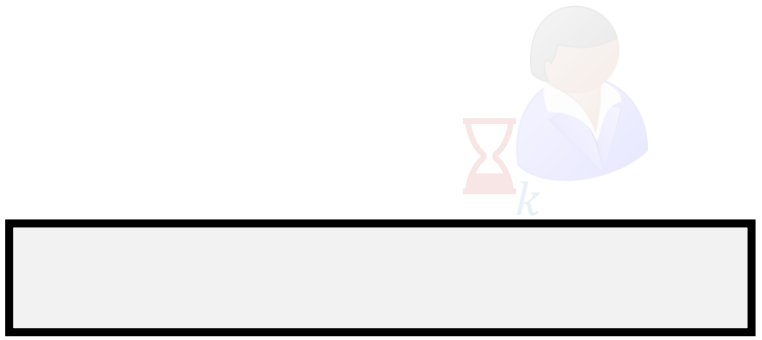
Extractable Commitments Blockchain Hybrid Model



Commitment valid only if timer doesn't run out.

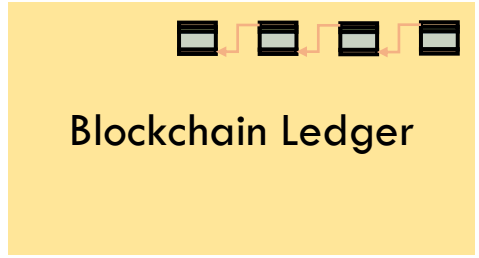
a coarse timer.

Time-out k



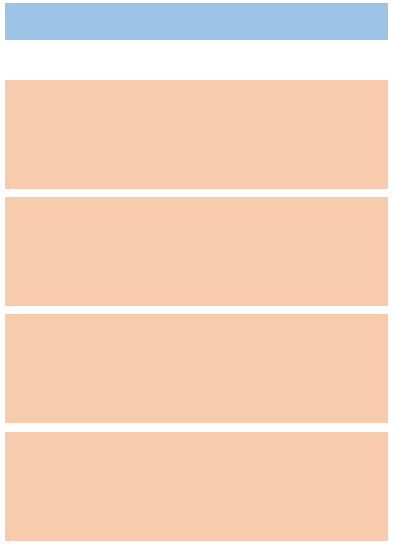
quits protocol if timer runs out.

Extractable Commitments Blockchain Hybrid Model

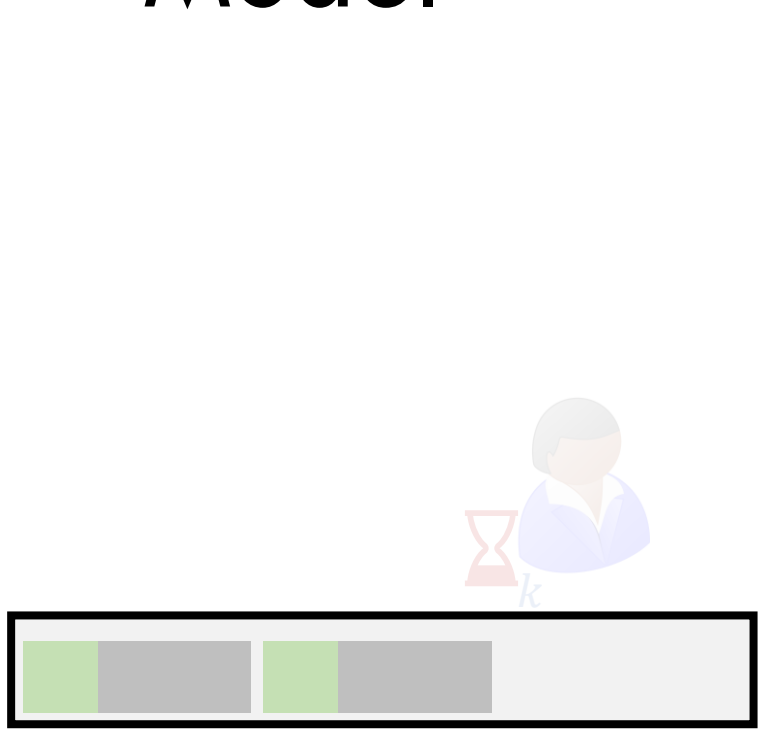


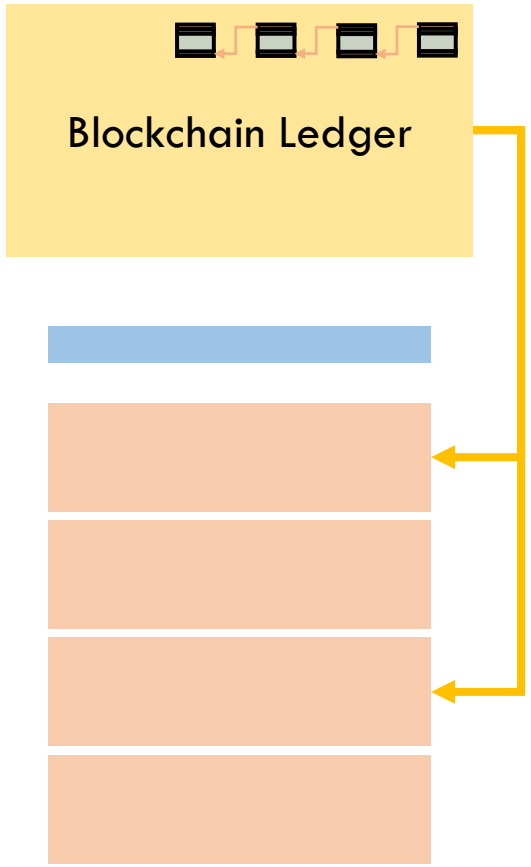
Commitment valid only if timer doesn't run out.

Example: 4 slots, time-out $k = 3$



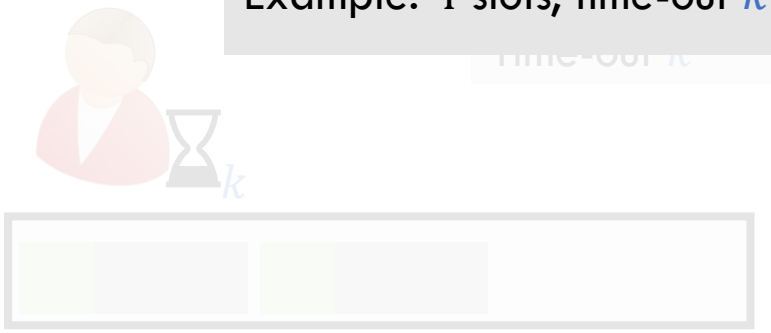
Extractable Commitments Blockchain Hybrid Model



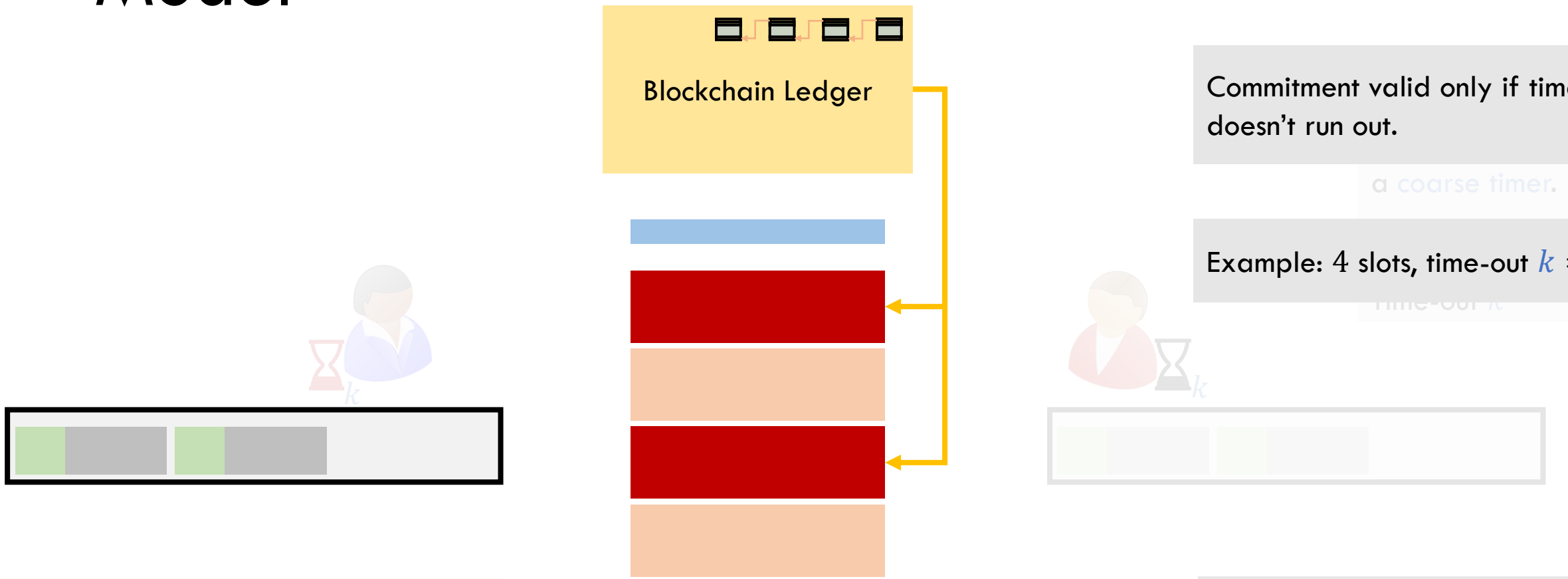


Commitment valid only if timer doesn't run out.

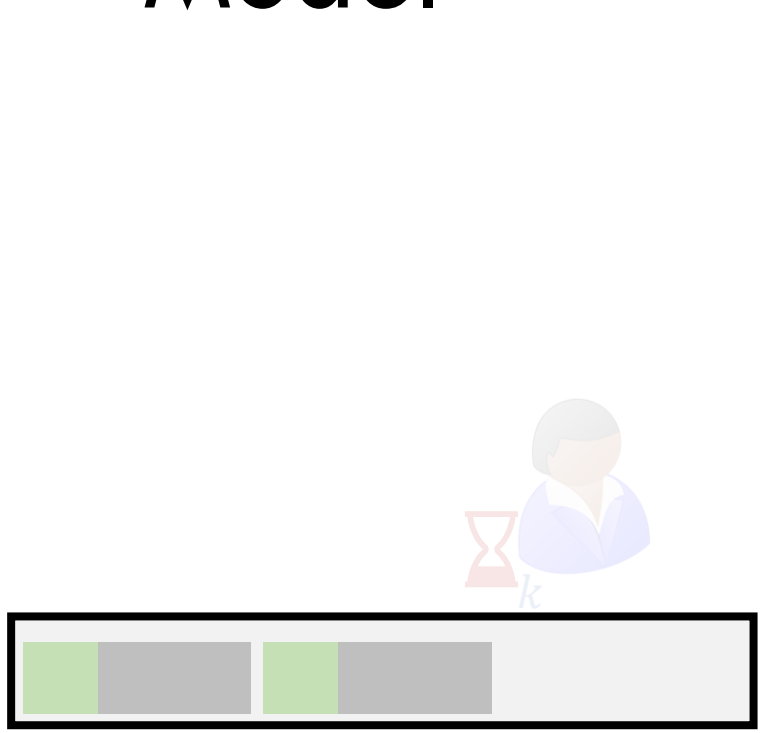
Example: 4 slots, time-out $k = 3$



Extractable Commitments Blockchain Hybrid Model

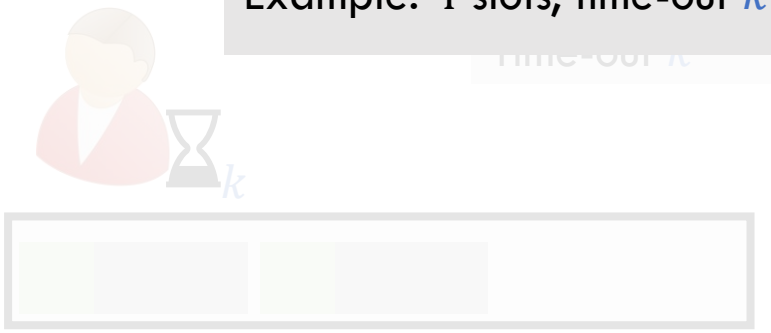


Extractable Commitments Blockchain Hybrid Model



Commitment valid only if timer doesn't run out.

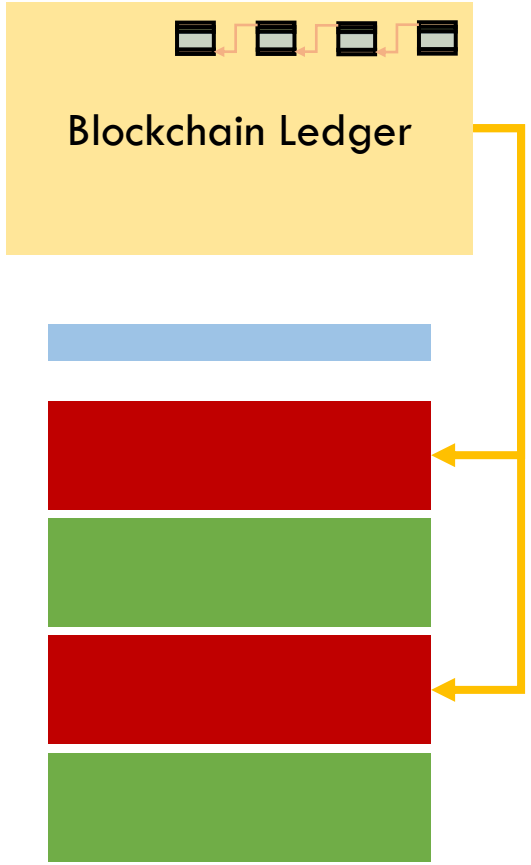
Example: 4 slots, time-out $k = 3$



Bad slot

Good slot

Extractable Commitments Blockchain Hybrid Model



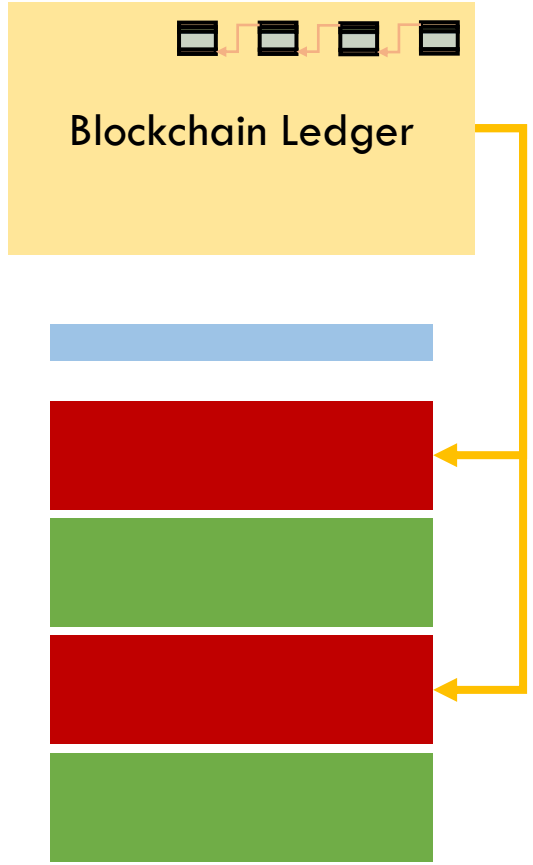
Commitment valid only if timer doesn't run out.

Example: 4 slots, time-out $k = 3$

Bad slot

Good slot

Extractable Commitments Blockchain Hybrid Model



Commitment valid only if timer doesn't run out.

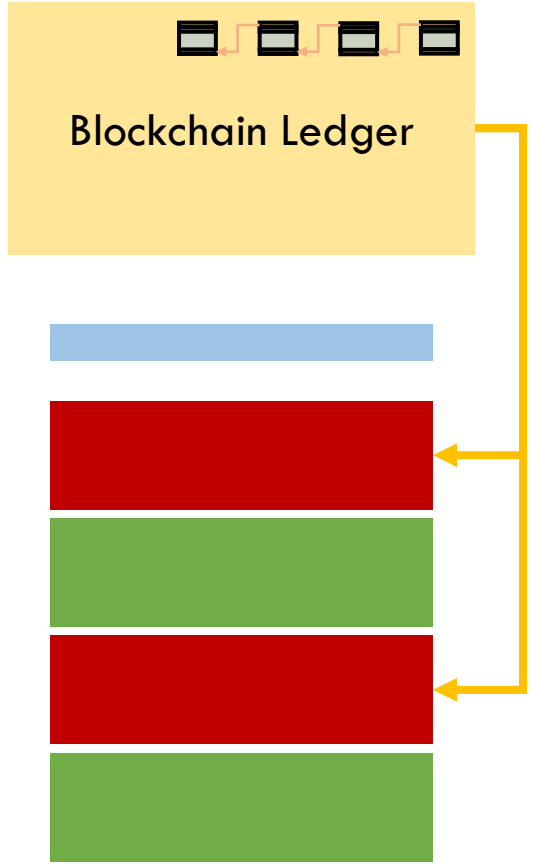
Example: 4 slots, time-out $k = 3$

Simulator can rewind any good slot to extract.

Bad slot

Good slot

Extractable Commitments Blockchain Hybrid Model



Commitment valid only if timer doesn't run out.

Example: 4 slots, time-out $k = 3$

Simulator can rewind any good slot to extract.

Guaranteed if $\#\text{slots} > k$

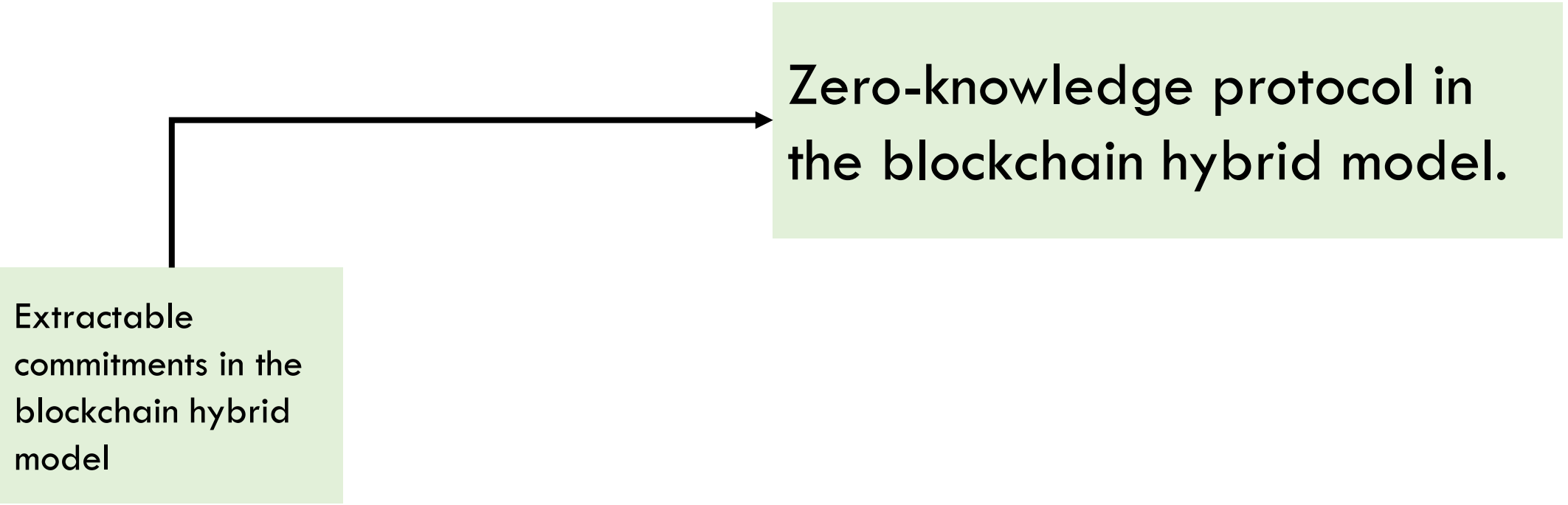
Bad slot

Good slot

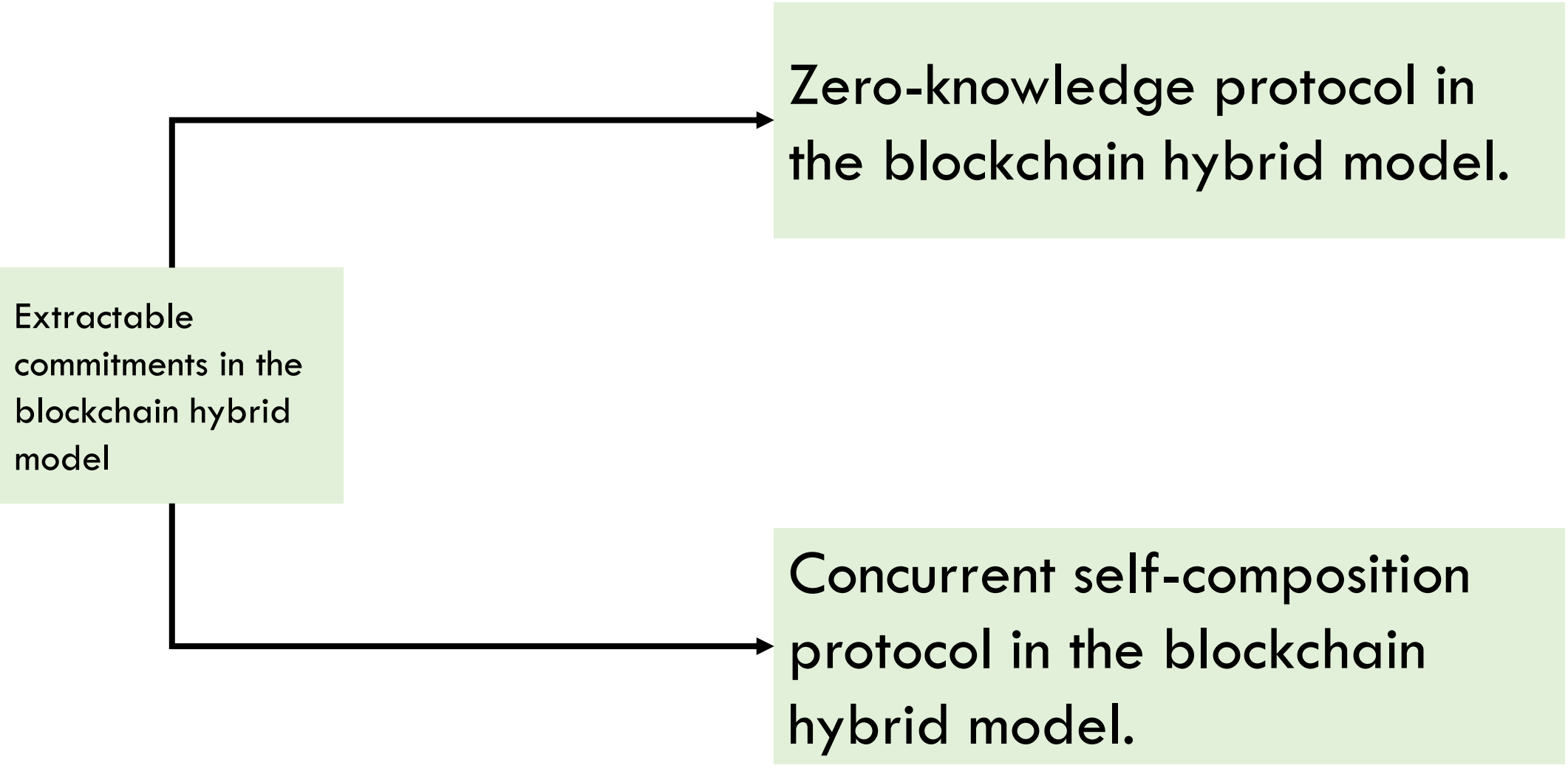
Results

Extractable commitments in the blockchain hybrid model

Results


Zero-knowledge protocol in the blockchain hybrid model.
Extractable commitments in the blockchain hybrid model

Results


Extractable commitments in the blockchain hybrid model
Zero-knowledge protocol in the blockchain hybrid model.
Concurrent self-composition protocol in the blockchain hybrid model.

Results

Zero-knowledge protocol in the blockchain hybrid model.

Extractable commitments in the blockchain hybrid model

Concurrent self-composition protocol in the blockchain hybrid model.

Results

Zero-knowledge
the blockchain h

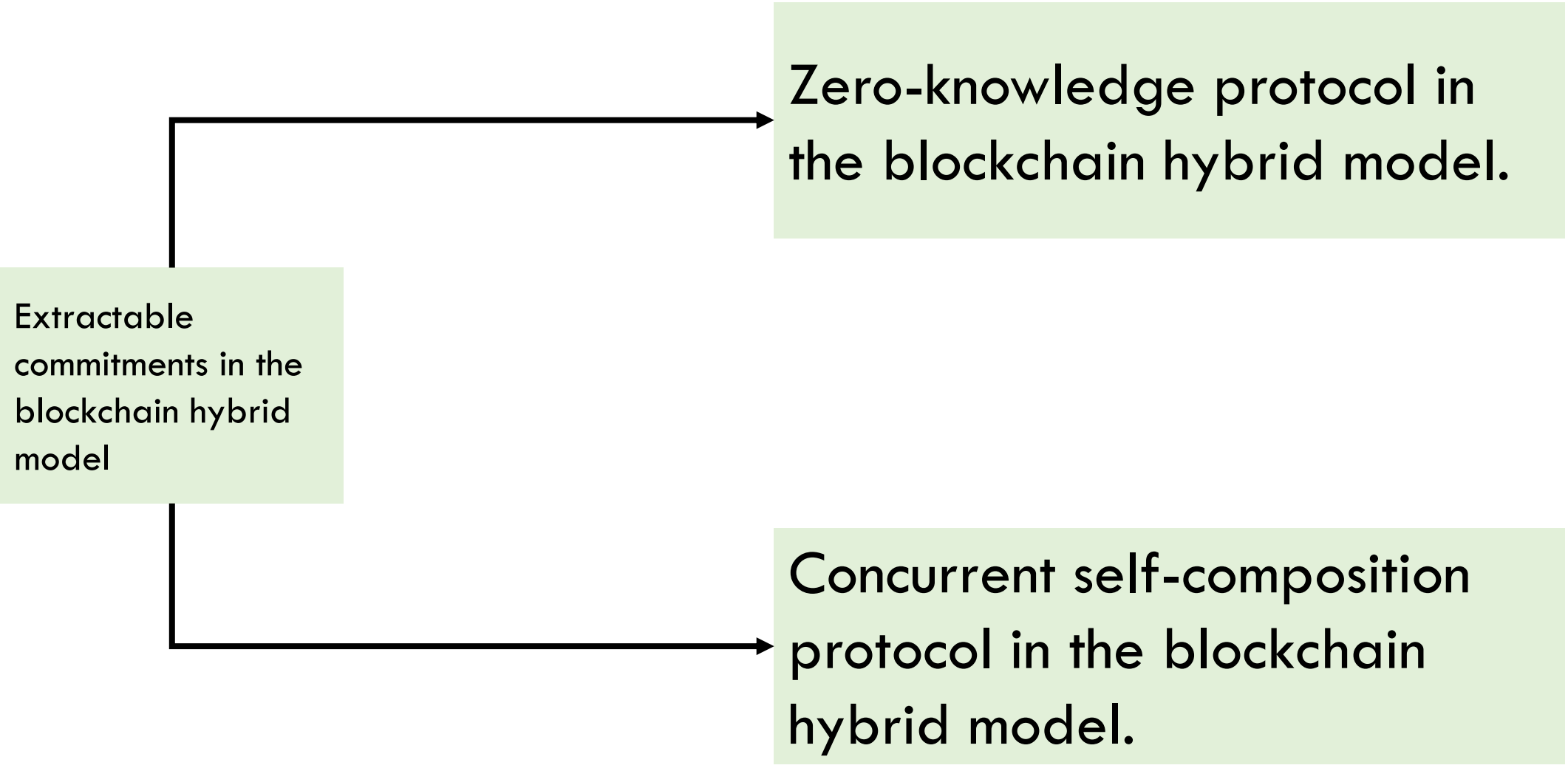
An extractable commitment slot is good if no new session started during the slot.

New session only when new block created.

Extractable commitments in the blockchain hybrid model

Concurrent self-composition protocol in the blockchain hybrid model.

Results


Extractable commitments in the blockchain hybrid model
Zero-knowledge protocol in the blockchain hybrid model.
Concurrent self-composition protocol in the blockchain hybrid model.

Necessity of Randomness in Zero-knowledge

Founding Secure Computation on Blockchains

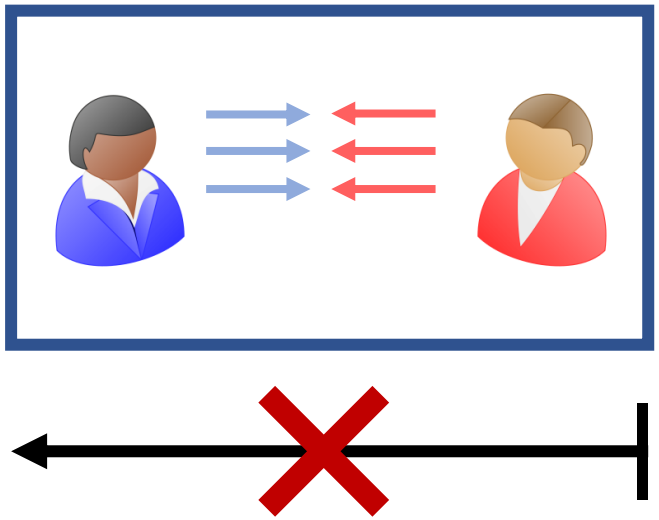
Round Optimal Secure Computation

Round Optimal Secure Multiparty Computation from Minimal Assumptions

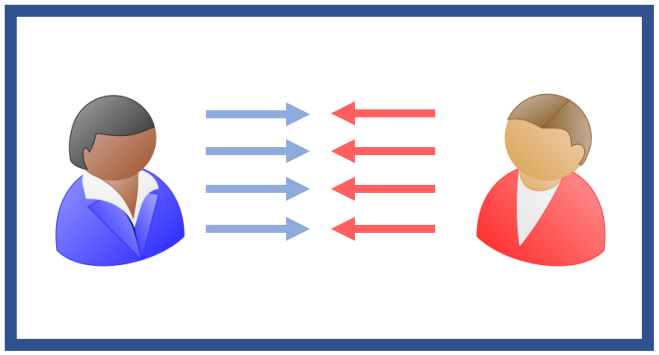
[C-Ciampi-Goyal-Jain-Ostrovsky'20]

Known bounds for interaction

[Garg-Mukherjee-Pandey-Polychroniadou'16]



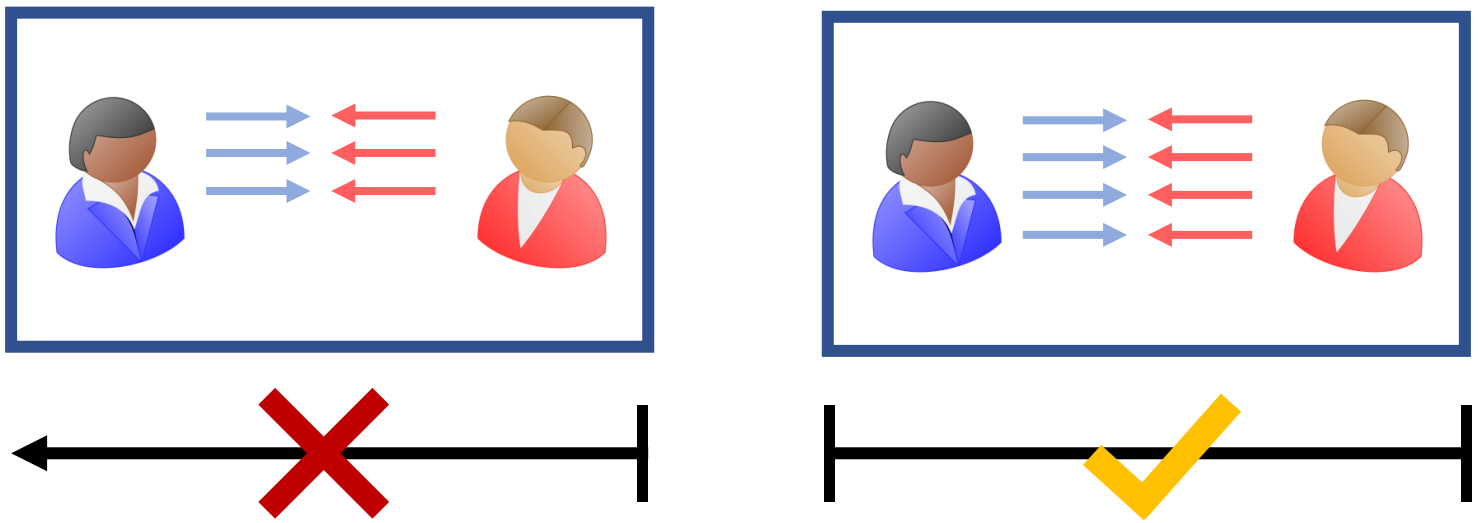
Impossible



?

Known bounds for interaction

[Garg-Mukherjee-Pandey-Polychroniadou'16]



Impossible

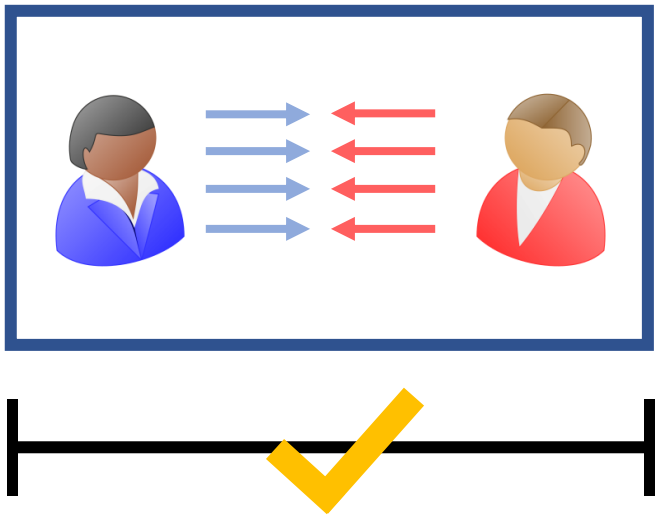
Four Round Protocols

[Ananth-**C**-Jain'17, Brakerski-Halevi-Polychroniadou'17]

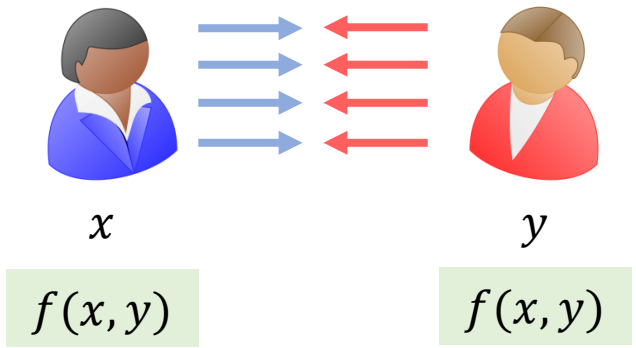
4 round protocol from subexponential hardness assumptions.

[Badrinarayanan-Goyal-Jain-Kalai-Khurana-Sahai'18, Halevi-Hazay-Polychroniadou-Venkitasubramaniam'18]

4 round protocol from strong number theoretic assumptions

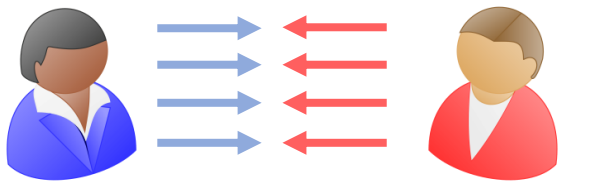


Necessary and Sufficient Functionality for Secure Computation



Necessary and Sufficient Functionality for Secure Computation

Oblivious Transfer (OT) Functionality



x

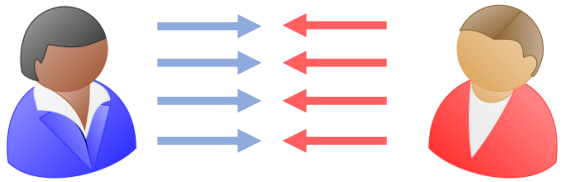
y

$f(x, y)$

$f(x, y)$

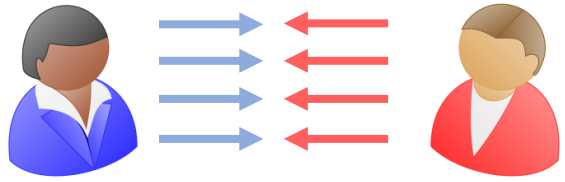
Necessary and Sufficient Functionality for Secure Computation

Oblivious Transfer (OT) Functionality



x y

$f(x,y)$ $f(x,y)$

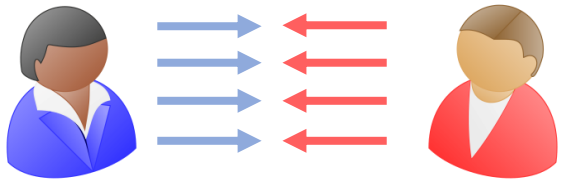


b a_0, a_1

$f(x,y)$ $f(x,y)$

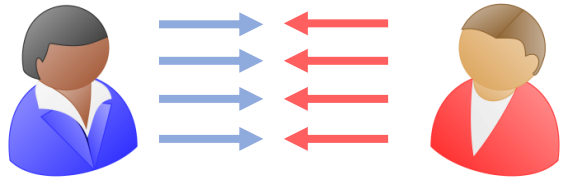
Necessary and Sufficient Functionality for Secure Computation

Oblivious Transfer (OT) Functionality



x y

$f(x,y)$ $f(x,y)$

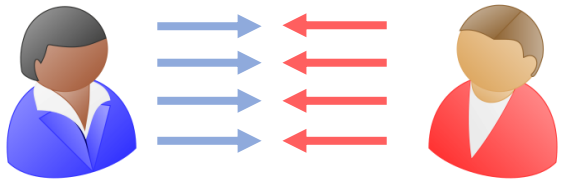


b a_0, a_1

$f(x,y)$ $\perp$

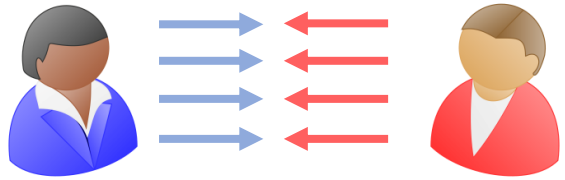
Necessary and Sufficient Functionality for Secure Computation

Oblivious Transfer (OT) Functionality



x y

$f(x,y)$ $f(x,y)$

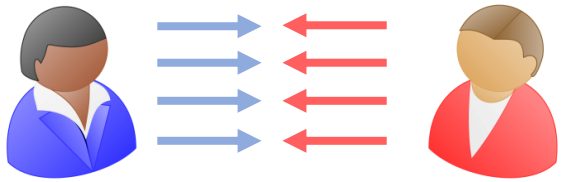


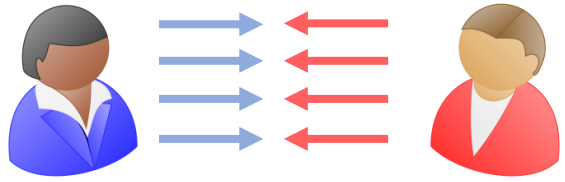
b a_0, a_1

a_b $\perp$

Necessary and Sufficient Functionality for Secure Computation

Oblivious Transfer (OT) Functionality



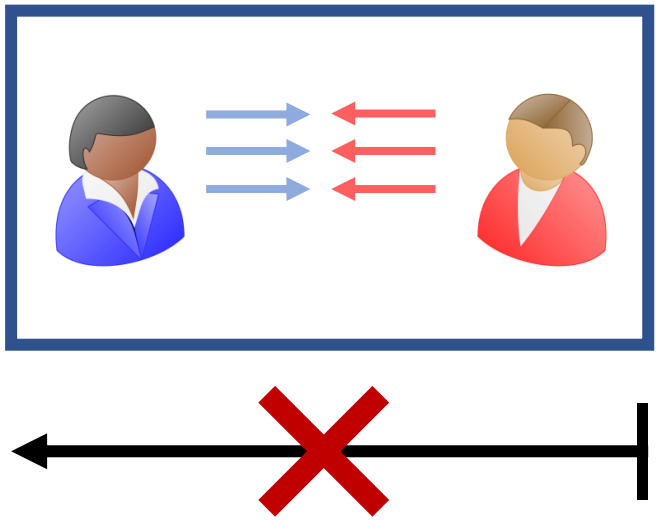


[Kilian'88]

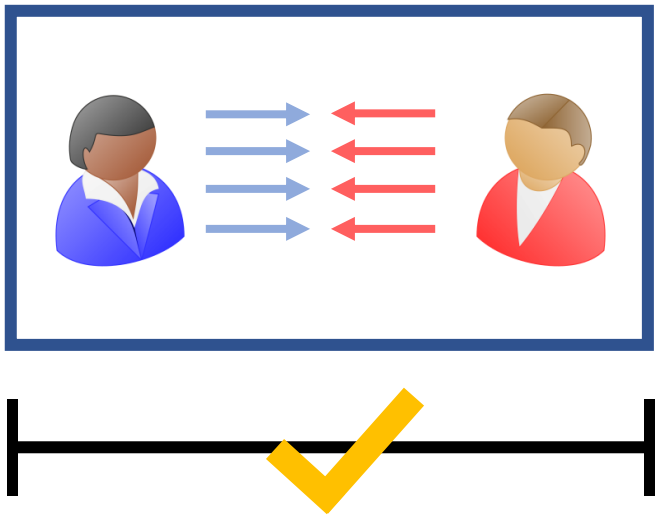
Oblivious Transfer both necessary and sufficient for secure computation.

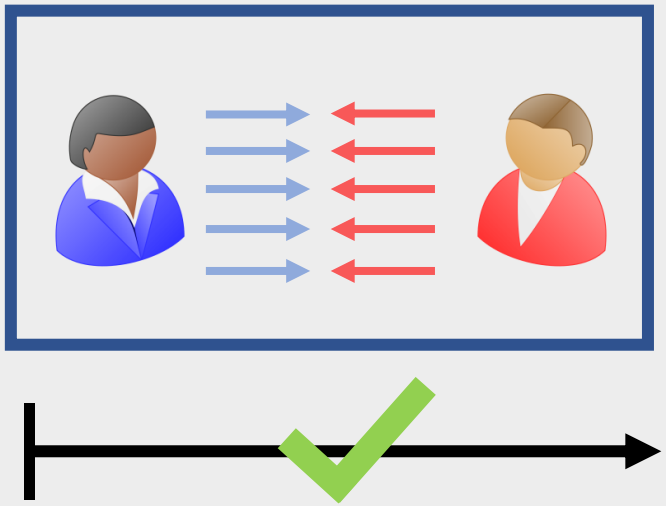
Known bounds for interaction

[Garg-Mukherjee-Pandey-Polychroniadou'16, Benhamouda-Lin'18]



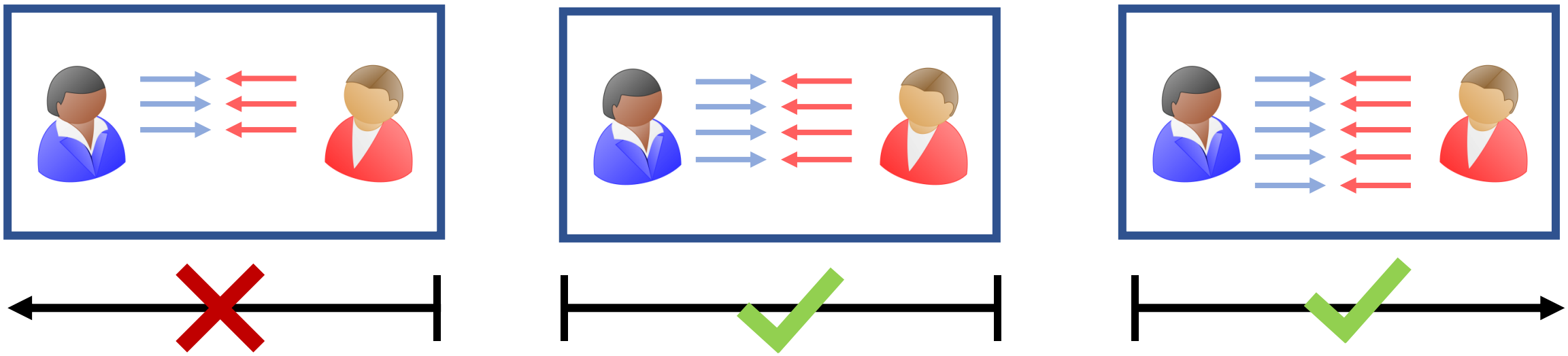
Impossible





k round OT $\Rightarrow$ k round Protocol $\forall k \geq 5$

Four Round Protocol from Minimal Assumptions



There exist four round secure computation protocols assuming four round oblivious transfer protocol.

Final Thoughts

Interactive Zero-Knowledge Proofs

Is prover randomness essential for zero-knowledge?

Secure Computation

Can we construct secure computation protocols in minimal rounds from minimal assumptions?

Can we make reasonable relaxations to the trust assumptions in order to circumvent barriers in secure computation?

Thanks to all my collaborators.









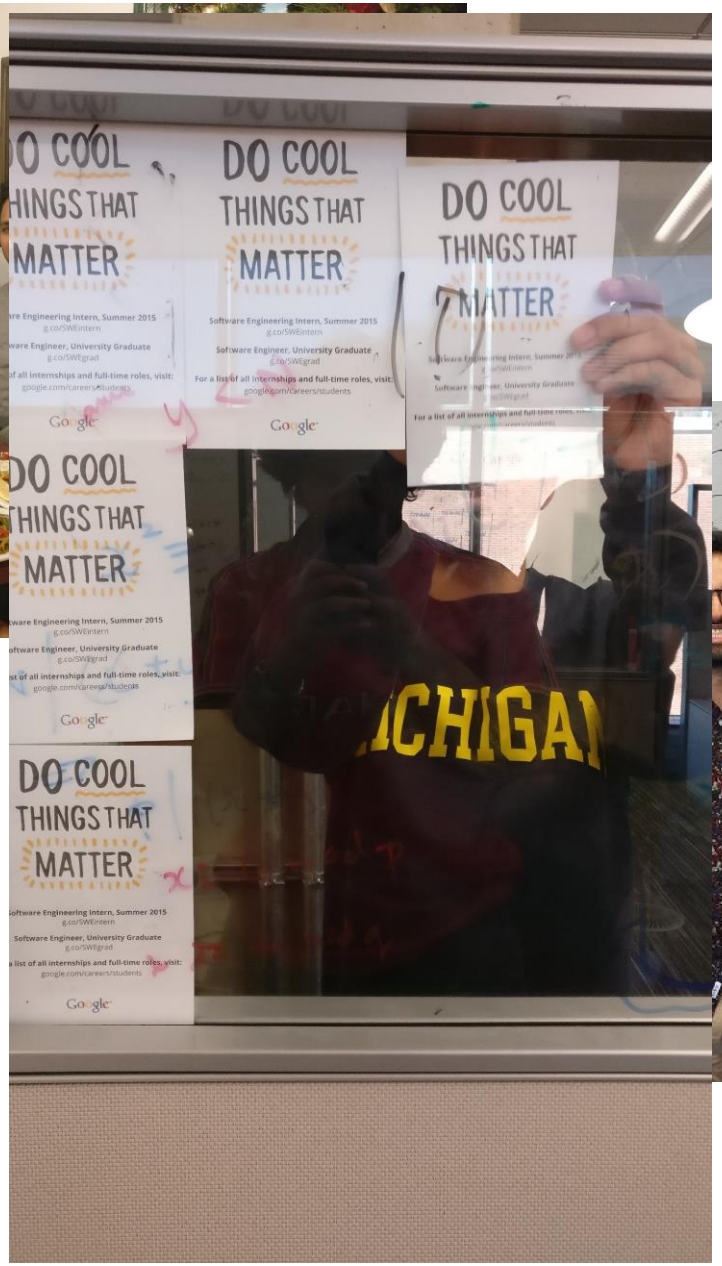


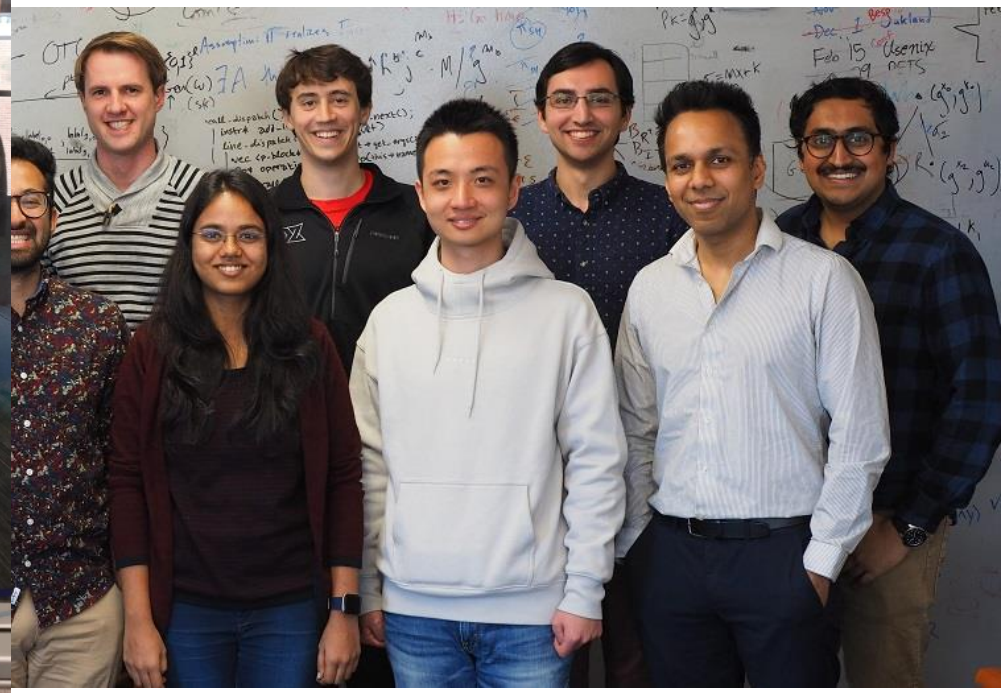


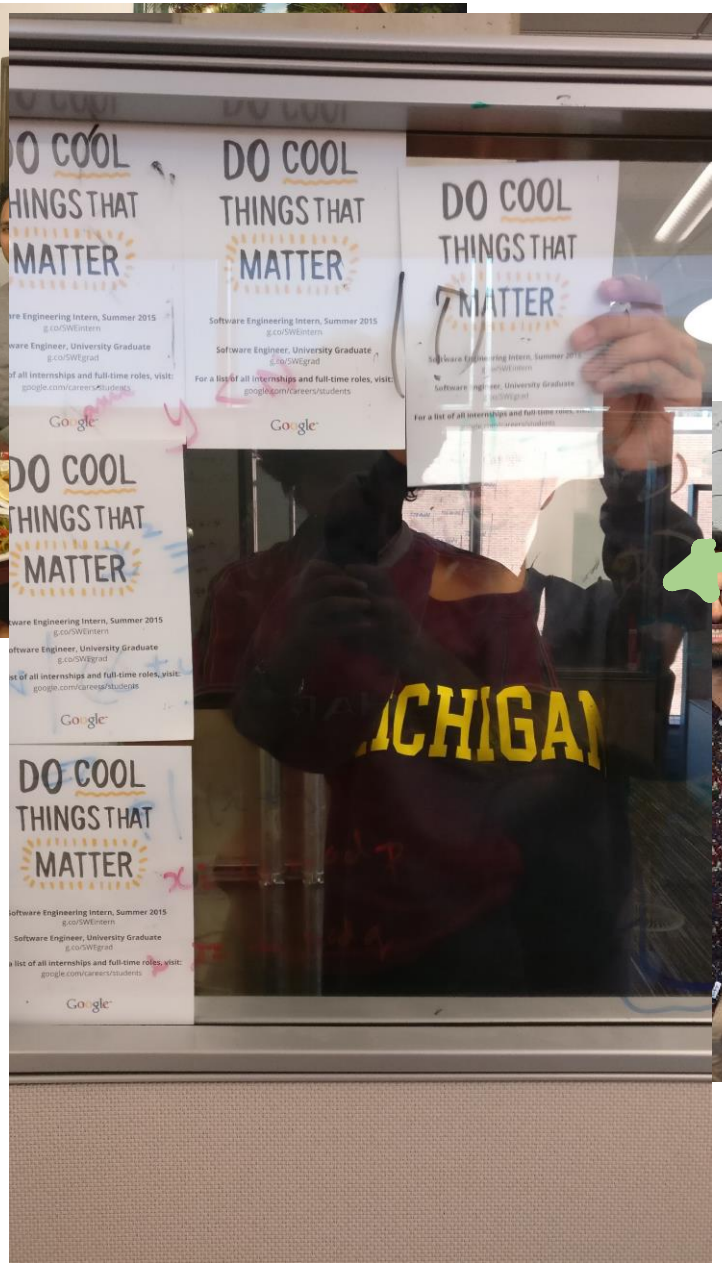




APAKS


DO COOL
THINGS THAT
MATTER
Software Engineering Intern, Summer 2015
g.co/SWEintern
Software Engineer, University Graduate
g.co/SWEgrad
For a list of all internships and full-time roles, visit:
google.com/careers/students
Google





Alishah

Thank you.

Questions?