

Relativized Succinct Arguments in the ROM Do Not Exist

Annalisa Barbara, Alessandro Chiesa, Ziyi Guan

Bocconi

EPFL



<https://eprint.iacr.org/2024/728>

Succinct non-interactive arguments

SNARGs in the ROM

Succinct non-interactive arguments

SNARGs in the ROM

Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

Succinct non-interactive arguments

SNARGs in the ROM

Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Succinct non-interactive arguments

SNARGs in the ROM

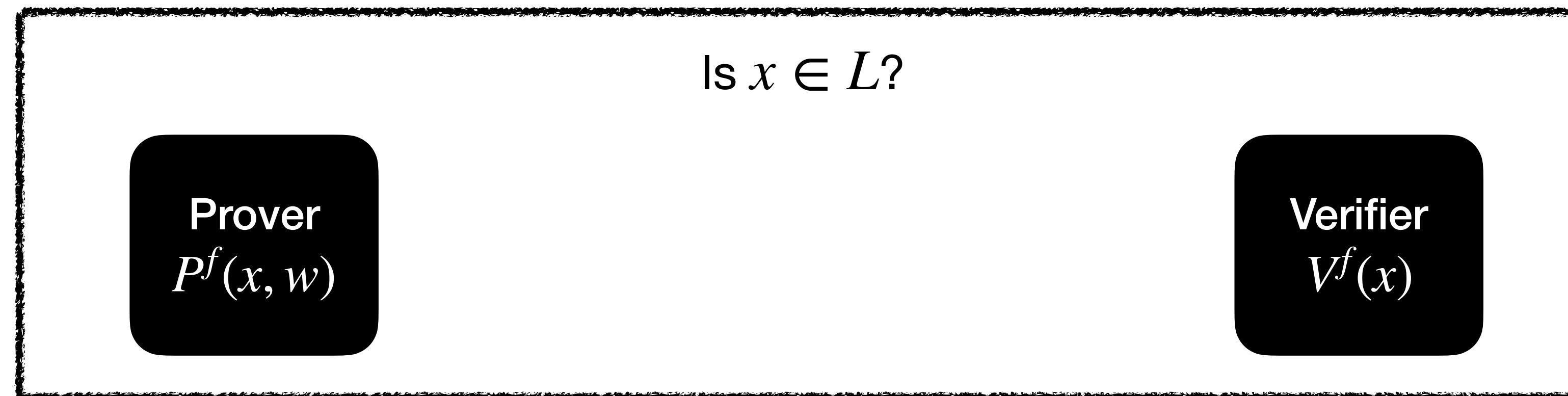


Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

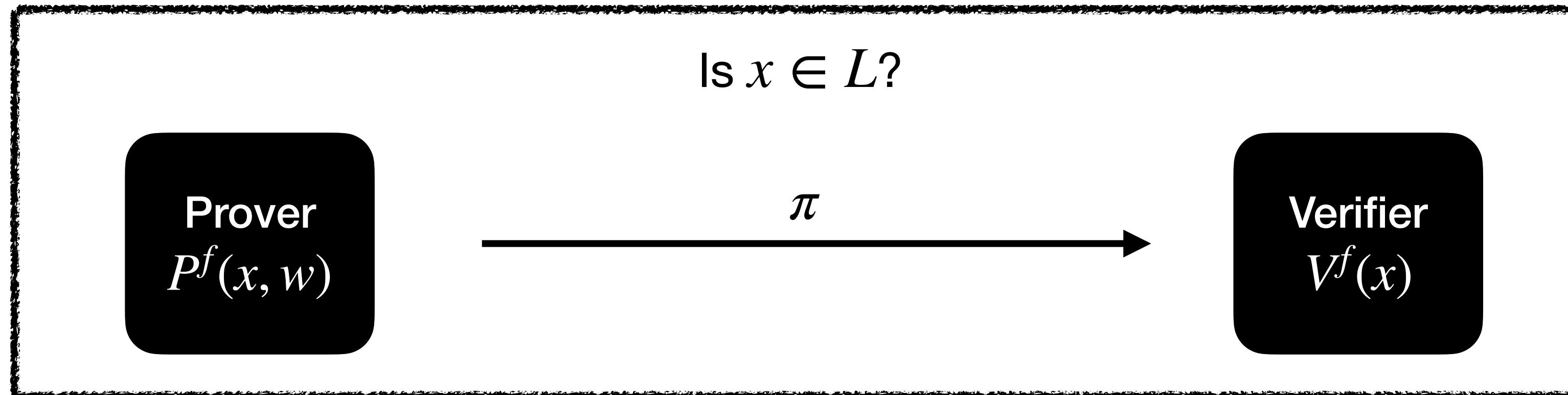
Succinct non-interactive arguments

SNARGs in the ROM



Succinct non-interactive arguments

SNARGs in the ROM

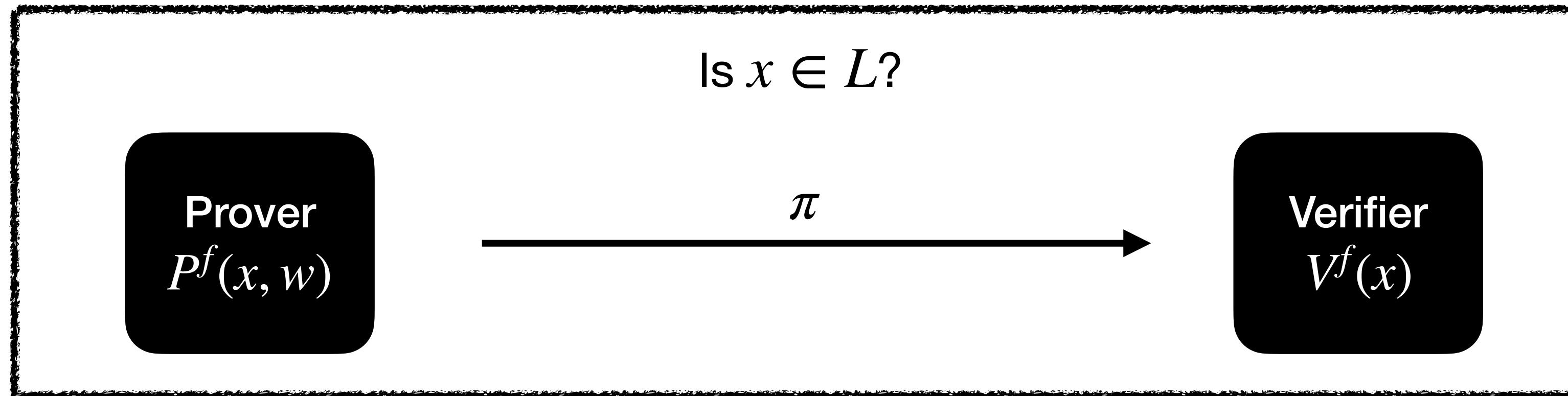


Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Succinct non-interactive arguments

SNARGs in the ROM



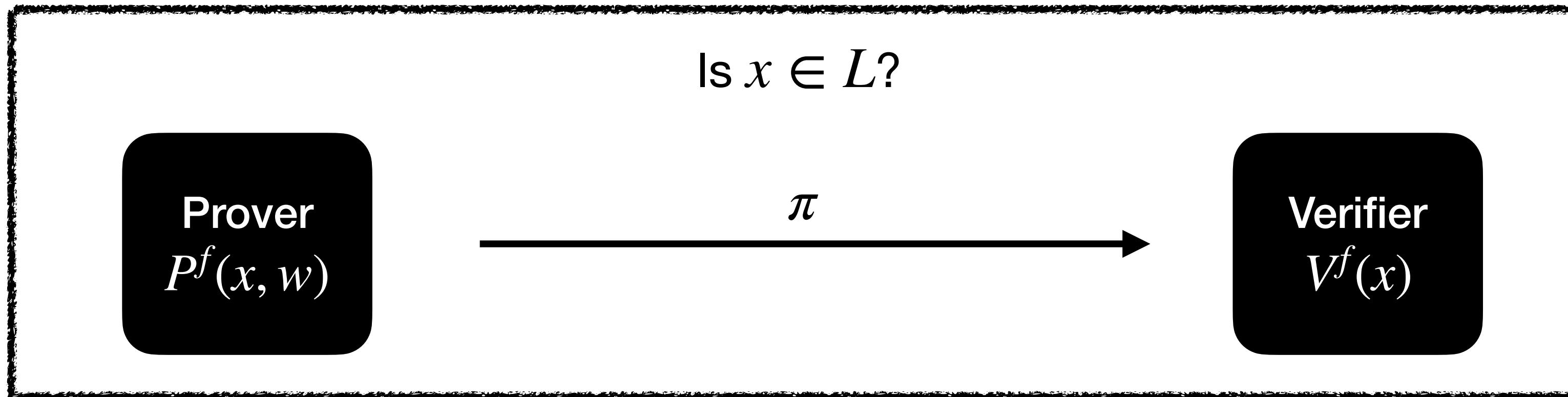
Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Completeness: \forall instance-generating adversary A ,

Succinct non-interactive arguments

SNARGs in the ROM



Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

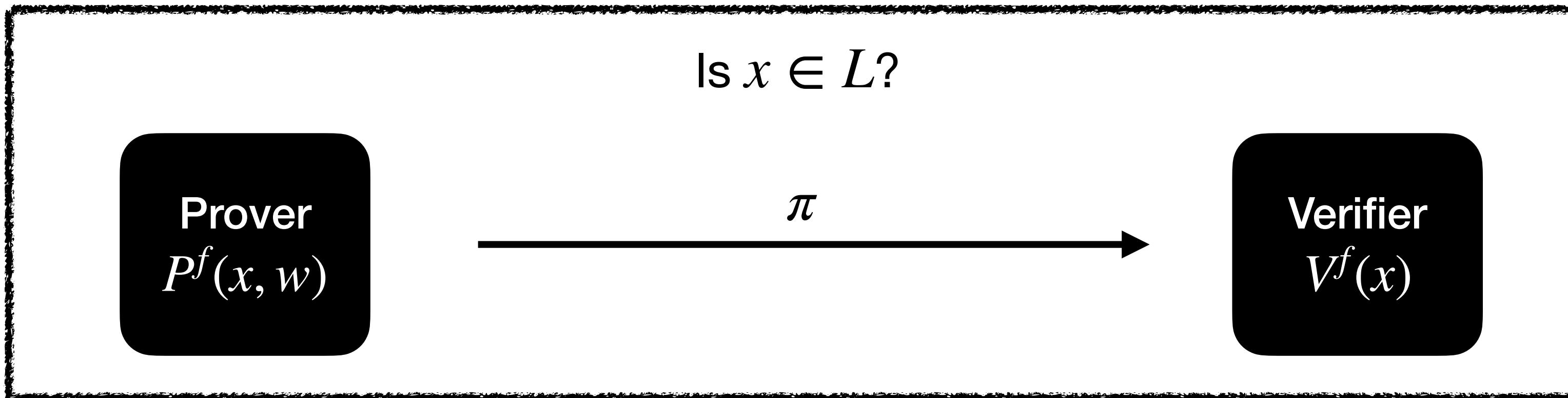
uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Completeness: \forall instance-generating adversary A ,

$$\Pr \left[x \in L \wedge V^f(x, \pi) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ x \leftarrow A^f \\ \pi \leftarrow P^f(x) \end{array} \right] = 1.$$

Succinct non-interactive arguments

SNARGs in the ROM



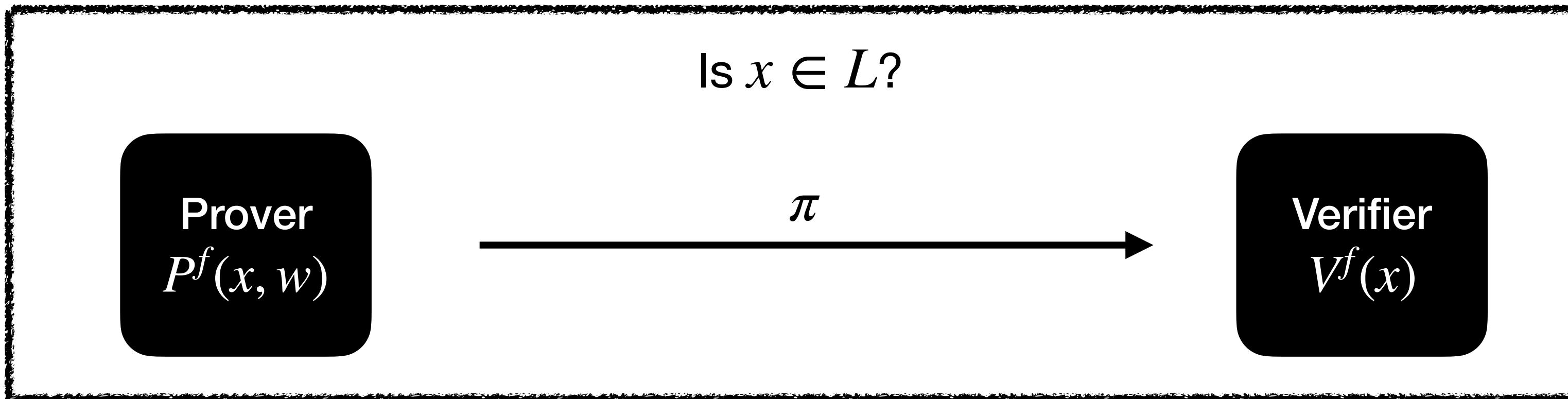
Completeness: \forall instance-generating adversary A ,

$$\Pr \left[x \in L \wedge V^f(x, \pi) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ x \leftarrow A^f \\ \pi \leftarrow P^f(x) \end{array} \right] = 1.$$

Soundness: \forall query-bounded and time-bounded adversary \tilde{P} ,

Succinct non-interactive arguments

SNARGs in the ROM



Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Completeness: \forall instance-generating adversary A ,

$$\Pr \left[x \in L \wedge V^f(x, \pi) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ x \leftarrow A^f \\ \pi \leftarrow P^f(x) \end{array} \right] = 1.$$

Soundness: \forall query-bounded and time-bounded adversary \tilde{P} ,

$$\Pr \left[x \notin L \wedge V^f(x, \tilde{\pi}) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ (x, \tilde{\pi}) \leftarrow \tilde{P}^f \end{array} \right] \leq \epsilon.$$

What is a relativized argument in the ROM?

Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

What is a relativized argument in the ROM?

Relativization: The language L is **relativized**, $L = \{L_f : f \in \mathcal{O}\}$.

Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

What is a relativized argument in the ROM?

Relativization: The language L is **relativized**, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$

Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

What is a relativized argument in the ROM?

Relativization: The language L is **relativized**, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$

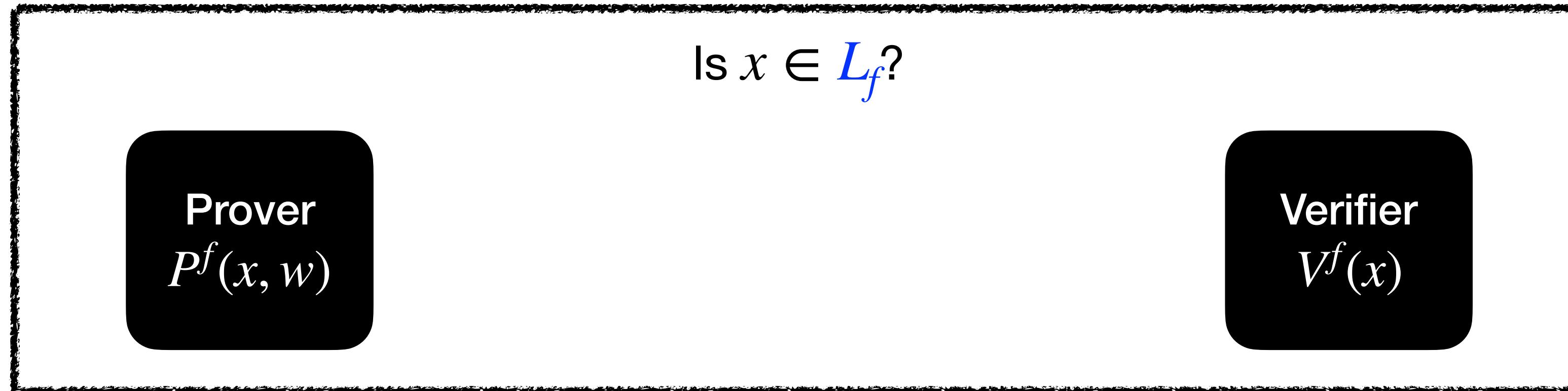


Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

What is a relativized argument in the ROM?

Relativization: The language L is **relativized**, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$

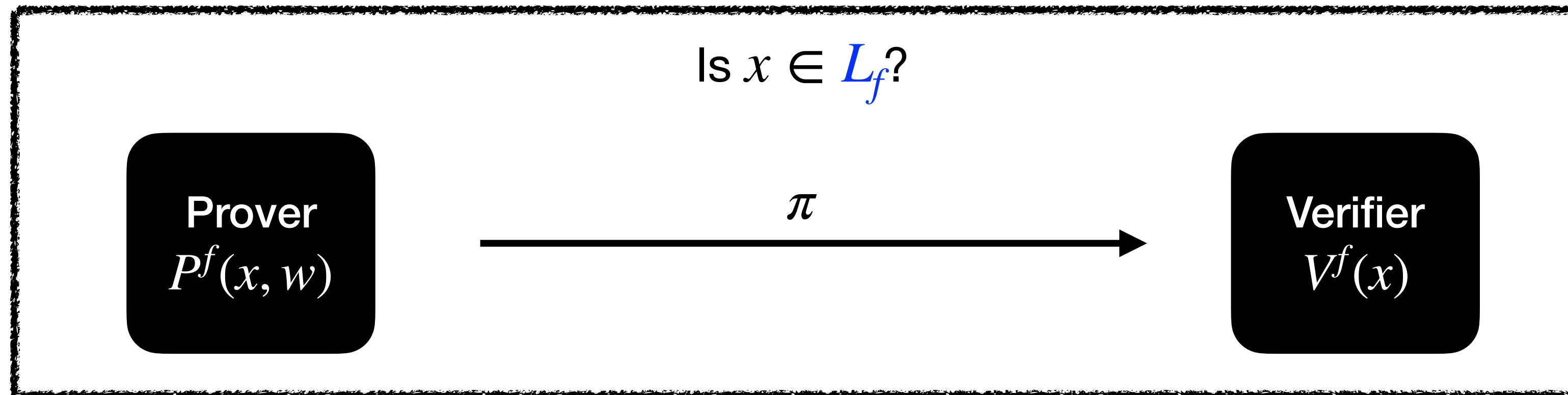


Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

What is a relativized argument in the ROM?

Relativization: The language L is relativized, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$

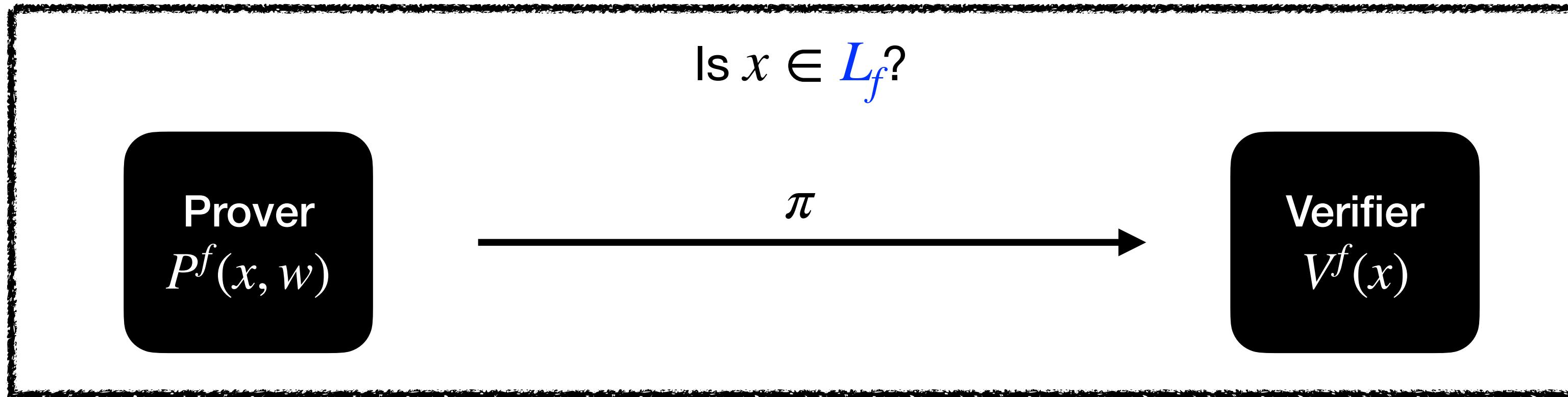


Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

What is a relativized argument in the ROM?

Relativization: The language L is relativized, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$



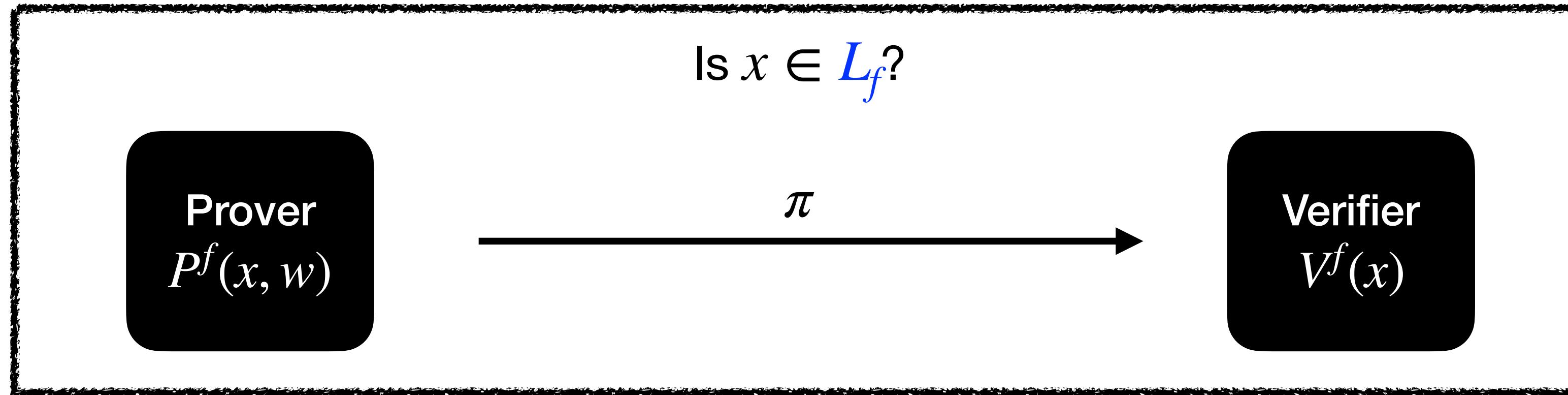
Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Completeness: \forall instance-generating adversary A ,

What is a relativized argument in the ROM?

Relativization: The language L is relativized, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$



Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

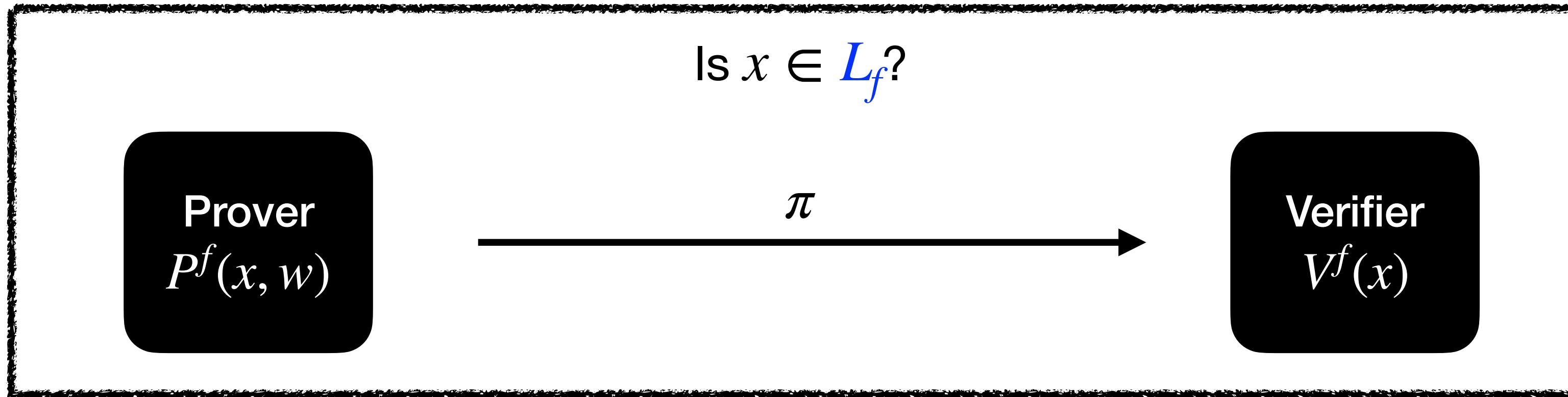
uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Completeness: \forall instance-generating adversary A ,

$$\Pr \left[x \in L_f \wedge V^f(x, \pi) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ x \leftarrow A^f \\ \pi \leftarrow P^f(x) \end{array} \right] = 1.$$

What is a relativized argument in the ROM?

Relativization: The language L is relativized, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$



Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all
functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

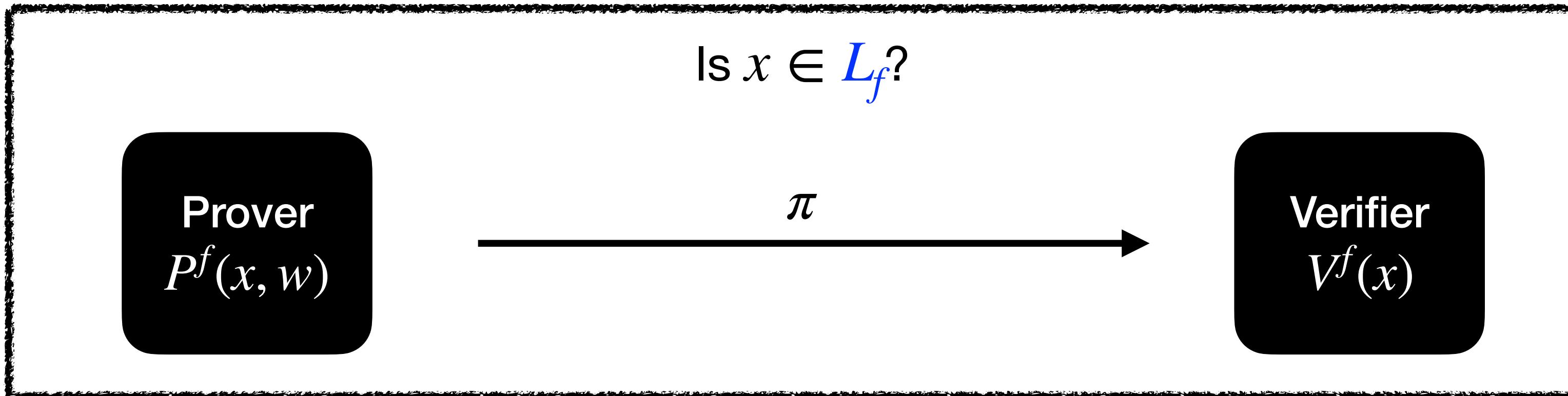
Completeness: \forall instance-generating adversary A ,

$$\Pr \left[x \in L_f \wedge V^f(x, \pi) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ x \leftarrow A^f \\ \pi \leftarrow P^f(x) \end{array} \right] = 1.$$

Soundness: \forall query-bounded and time-bounded adversary \tilde{P} ,

What is a relativized argument in the ROM?

Relativization: The language L is **relativized**, $L = \{L_f : f \in \mathcal{O}\}$. e.g. $L_f := \{(x, y) : y = f(x)\}$



Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

uniform distribution over all functions $f: \{0,1\}^* \rightarrow \{0,1\}^\ell$

Completeness: \forall instance-generating adversary A ,

$$\Pr \left[x \in L_f \wedge V^f(x, \pi) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ x \leftarrow A^f \\ \pi \leftarrow P^f(x) \end{array} \right] = 1.$$

Soundness: \forall query-bounded and time-bounded adversary \tilde{P} ,

$$\Pr \left[x \notin L_f \wedge V^f(x, \tilde{\pi}) = 1 \mid \begin{array}{l} f \leftarrow \mathcal{O} \\ (x, \tilde{\pi}) \leftarrow \tilde{P}^f \end{array} \right] \leq \epsilon.$$

Why study relativized arguments? [1/2]

Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation

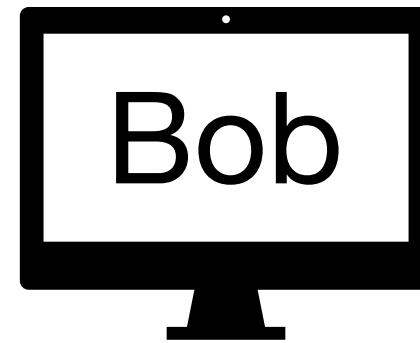
Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation



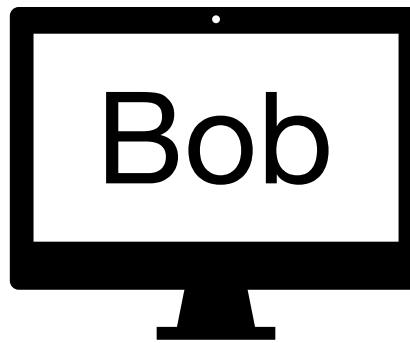
Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation



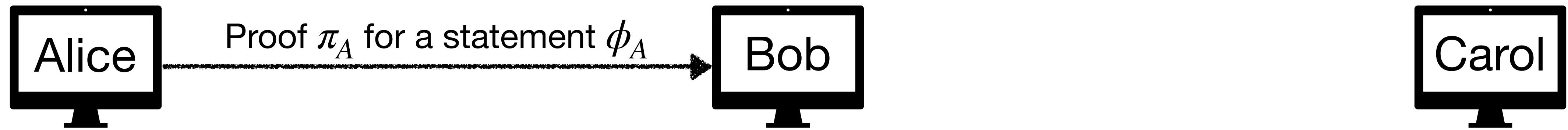
Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation



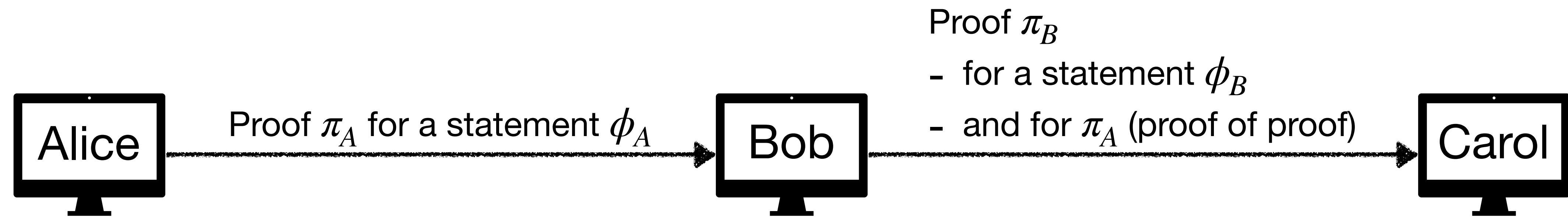
Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation



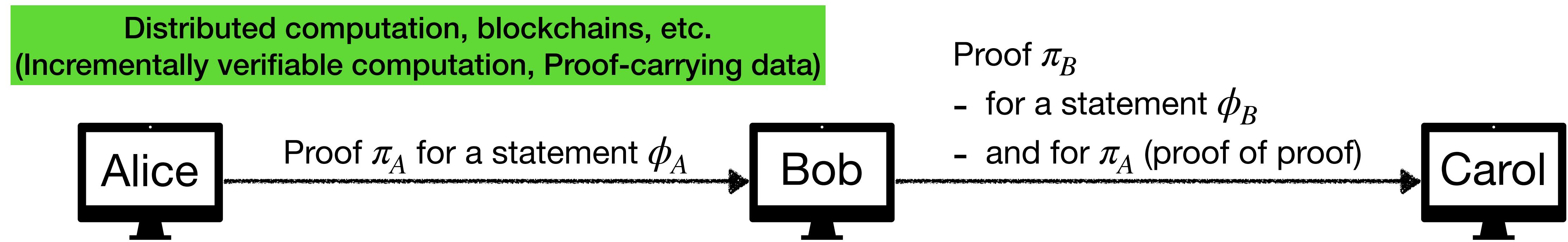
Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation



Why study relativized arguments? [1/2]

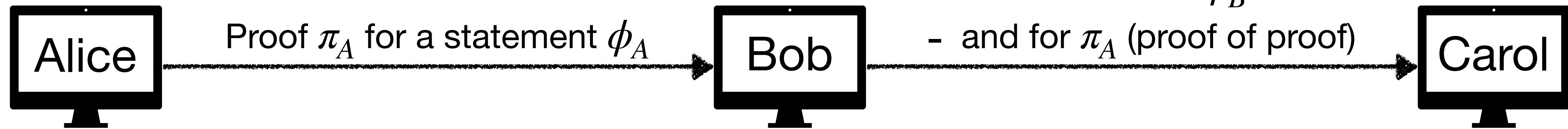
Motivation 1: Verifiable distributed computation



Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation

Distributed computation, blockchains, etc.
(Incrementally verifiable computation, Proof-carrying data)



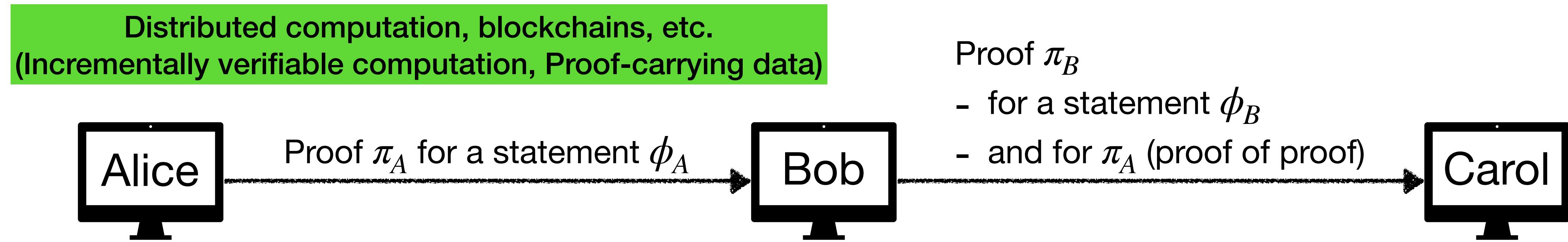
Proof π_B

- for a statement ϕ_B
- and for π_A (proof of proof)

How does Bob produce π_B ?

Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation



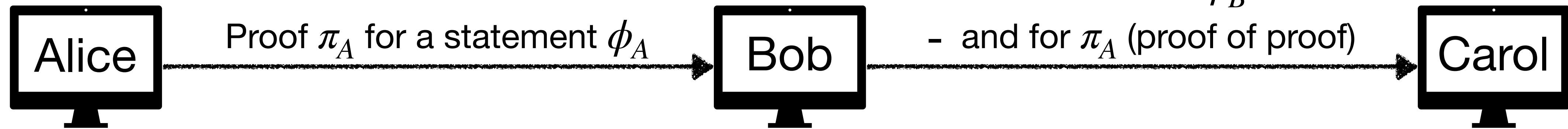
How does Bob produce π_B ?

Let ARG = (P, V) be a SNARG for relativized CSAT:

Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation

Distributed computation, blockchains, etc.
(Incrementally verifiable computation, Proof-carrying data)



Proof π_B

- for a statement ϕ_B
- and for π_A (proof of proof)

$$\text{CSAT}_f := \{(C, x) : \exists w, C^f(x, w) = 1\}$$

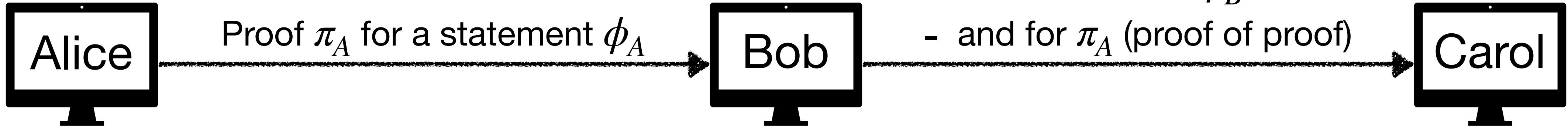
How does Bob produce π_B ?

Let ARG = (P, V) be a SNARG for relativized CSAT:

Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation

Distributed computation, blockchains, etc.
(Incrementally verifiable computation, Proof-carrying data)



Proof π_B

- for a statement ϕ_B
- and for π_A (proof of proof)

$$\text{CSAT}_f := \{(C, x) : \exists w, C^f(x, w) = 1\}$$

How does Bob produce π_B ?

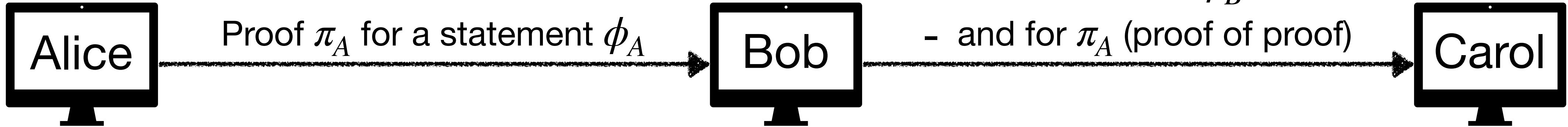
Let ARG = (P, V) be a SNARG for relativized CSAT:

Oracle recursive circuit $\mathcal{C}^f(\phi_B, (\phi_A, \pi_A))$

Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation

Distributed computation, blockchains, etc.
(Incrementally verifiable computation, Proof-carrying data)



Proof π_B

- for a statement ϕ_B
- and for π_A (proof of proof)

$$\text{CSAT}_f := \{(C, x) : \exists w, C^f(x, w) = 1\}$$

How does Bob produce π_B ?

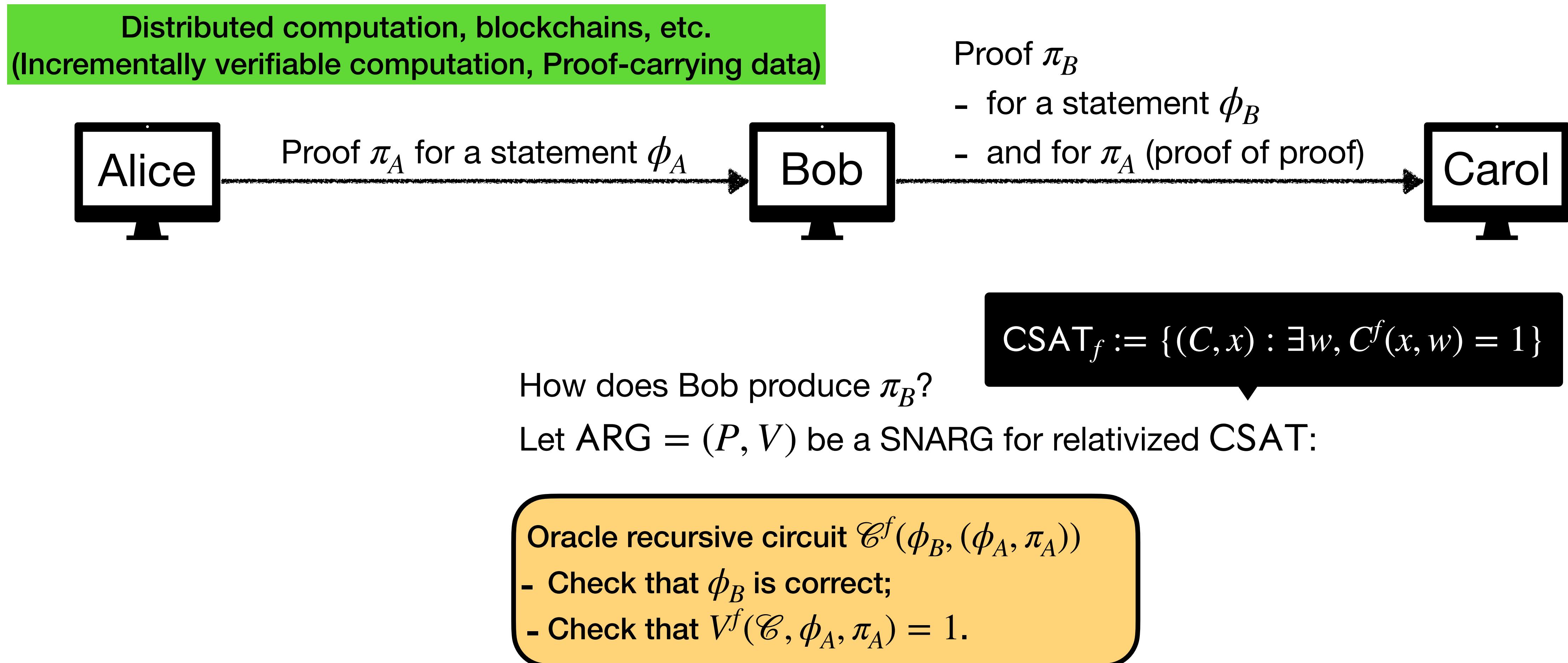
Let ARG = (P, V) be a SNARG for relativized CSAT:

Oracle recursive circuit $\mathcal{C}^f(\phi_B, (\phi_A, \pi_A))$

- Check that ϕ_B is correct;

Why study relativized arguments? [1/2]

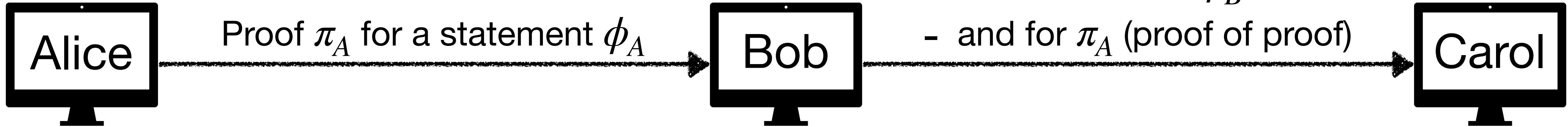
Motivation 1: Verifiable distributed computation



Why study relativized arguments? [1/2]

Motivation 1: Verifiable distributed computation

Distributed computation, blockchains, etc.
(Incrementally verifiable computation, Proof-carrying data)



Proof π_B

- for a statement ϕ_B
- and for π_A (proof of proof)

$$\text{CSAT}_f := \{(C, x) : \exists w, C^f(x, w) = 1\}$$

How does Bob produce π_B ?

Let ARG = (P, V) be a SNARG for relativized CSAT:

Oracle recursive circuit $\mathcal{C}^f(\phi_B, (\phi_A, \pi_A))$

- Check that ϕ_B is correct;
- Check that $V^f(\mathcal{C}, \phi_A, \pi_A) = 1$.

$$\pi_B \leftarrow P^f(\mathcal{C}, \phi_B, (\phi_A, \pi_A))$$

Why study relativized arguments? [2/2]

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

hash function

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

hash function

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

- Treat the hash function as an oracle.

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

- Treat the hash function as an oracle.
- Relativized arguments do not depend on complexity of the hash functions. 😊

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

- Treat the hash function as an oracle.
- Relativized arguments do not depend on complexity of the hash functions. 😊

If $\text{NP}^{H_s} \subseteq \text{ARG}^{H_s}$, SNARGs for L_s do not depend on $|H_s|$

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

- Treat the hash function as an oracle.
- Relativized arguments do not depend on complexity of the hash functions. 😊

If $\text{NP}^{H_s} \subseteq \text{ARG}^{H_s}$, SNARGs for L_s do not depend on $|H_s|$

More generally, relativization removes the need for optimizing the recurring sub-computation.

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

- Treat the hash function as an oracle.
- Relativized arguments do not depend on complexity of the hash functions. 😊

If $\text{NP}^{H_s} \subseteq \text{ARG}^{H_s}$, SNARGs for L_s do not depend on $|H_s|$

More generally, relativization removes the need for optimizing the recurring sub-computation.

Do relativized SNARGs exist in oracle models?

Why study relativized arguments? [2/2]

Motivation 2: Efficiency

Recurring cryptographic computations show up a lot:

- Correctness proof of encryption/decryption, signature verification, **hash function**, etc.

e.g. $L_s := \{(n, y) \in \mathbb{N} \times \{0,1\}^{|s|} : \exists x \in \{0,1\}^s, H_s^{(n)}(x) = y\}$

hash function

Necessary to construct hash function with small size??

SNARGs for L_s are expensive (|circuit that iteratively applies H_s for n times| = $\Omega(n |H_s|)$).

Potential alternative route:

- Treat the hash function as an oracle.
- Relativized arguments do not depend on complexity of the hash functions. 😊

If $\text{NP}^{H_s} \subseteq \text{ARG}^{H_s}$, SNARGs for L_s do not depend on $|H_s|$

More generally, relativization removes the need for optimizing the recurring sub-computation.

Do relativized SNARGs exist in oracle models? Yes!

Existing relativized SNARGs

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

How about the random oracle model?

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

How about the random oracle model?

Popular belief: No.

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

How about the random oracle model?

Popular belief: No.

Popular intuition: Relativized PCPs/IOPs do not exist in the ROM [CL20].

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

How about the random oracle model?

Popular belief: No.

Popular intuition: Relativized PCPs/IOPs do not exist in the ROM [CL20].

Counterexample to popular belief:

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

How about the random oracle model?

Popular belief: No.

Popular intuition: Relativized PCPs/IOPs do not exist in the ROM [CL20].

Counterexample to popular belief:

- Relativized PCPs/IOPs **do not exist** in the LDROM [CL20].

Existing relativized SNARGs

Relativized SNARGs exist in some oracle models:

- Signed random oracle model (SROM) [CT10]
- Low-degree random oracle (LDROM) [CCS22]
- Arithmetized random oracle model (AROM) [CCGOS23]

Hard to instantiate!

How about the random oracle model?

Popular belief: No.

Popular intuition: Relativized PCPs/IOPs do not exist in the ROM [CL20].

Counterexample to popular belief:

- Relativized PCPs/IOPs **do not exist** in the LDROM [CL20].
- Relativized SNARGs **exist** in the LDROM [CCS22].

Our results

Our results

Relativized arguments in the random oracle model do not exist.

Our results

Relativized arguments in the random oracle model do not exist.

verifier query complexity to the RO
Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Theorem 2. $\text{NTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Theorem 2. $\text{NTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

Corollary. Relativized IVC/PCD does not exist in the ROM!

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Theorem 2. $\text{NTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

Existence of IVC/PCD in the ROM still remains open.

Corollary. Relativized IVC/PCD does not exist in the ROM!

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Theorem 2. $\text{NTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

Existence of IVC/PCD in the ROM still remains open.

Corollary. Relativized IVC/PCD does not exist in the ROM!

Note.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Theorem 2. $\text{NTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

Existence of IVC/PCD in the ROM still remains open.

Corollary. Relativized IVC/PCD does not exist in the ROM!

Note.

- The results hold for SNARGs secure against **query-bounded** and **time-bounded** adversaries.

Our results

Relativized arguments in the random oracle model do not exist.

Trivial Baseline 1. $\text{DTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = t]$.

verifier query complexity to the RO

Verifier computes everything itself.

Theorem 1. $\text{DTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{vq} = o(t)]$.

argument proof size

Trivial Baseline 2. $\text{NTIME}^{\mathcal{O}}[t] \subseteq \text{ARG}^{\mathcal{O}}[\text{as} = t]$.

Prover sends the entire witness.

Theorem 2. $\text{NTIME}^{\mathcal{O}}[t] \not\subseteq \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

Existence of IVC/PCD in the ROM still remains open.

Corollary. Relativized IVC/PCD does not exist in the ROM!

Note.

- The results hold for SNARGs secure against **query-bounded** and **time-bounded** adversaries.
- Similar results hold for interactive arguments.

Separation between NTIME and ARG

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

$$L_{\mathcal{O}} := \{L_f : f \in \mathcal{O}\}$$

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

$$L_{\mathcal{O}} := \{L_f : f \in \mathcal{O}\}$$

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

$$L_{\mathcal{O}} := \{L_f : f \in \mathcal{O}\}$$

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f	1000	0101	1111	0000	0010
f	0001	0111	0110	0111	0110

$x \notin L_f$

$x \in L_f$

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

$$L_{\mathcal{O}} := \{L_f : f \in \mathcal{O}\}$$

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f	1000	0101	1111	0000	0010
f	0001	0111	0110	0111	0110

$x \notin L_f$
 $x \in L_f$

Why is L_f hard?

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

$$L_{\mathcal{O}} := \{L_f : f \in \mathcal{O}\}$$

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f	1000	0101	1111	0000	0010
f	0001	0111	0110	0111	0110

$x \notin L_f$
 $x \in L_f$

Why is L_f hard?

- Needs $t(n)$ queries to be sure that $x \in L_f$ or not.

Hard language in $\text{NTIME}^{\mathcal{O}}[t]$

Lemma.

There exists $L_{\mathcal{O}}$ such that $L_{\mathcal{O}} \in \text{NTIME}^{\mathcal{O}}[t]$ and $L_{\mathcal{O}} \notin \text{ARG}^{\mathcal{O}}[\text{as} = o(t)]$.

argument proof size

$$L_{\mathcal{O}} := \{L_f : f \in \mathcal{O}\}$$

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f	1000	0101	1111	0000	0010
f	0001	0111	0110	0111	0110

$x \notin L_f$
 $x \in L_f$

Why is L_f hard?

- Needs $t(n)$ queries to be sure that $x \in L_f$ or not.
- Flipping even one bit of f could change the membership of x .

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f	1000	0101	0100	0000	0010
f'	0000	0101	0100	0000	0010

$x \notin L_f$
 $x \in L_f$

Proof outline

Proof outline

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

Proof outline

1. Fix $x := 0^n$ for some n .

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

Proof outline

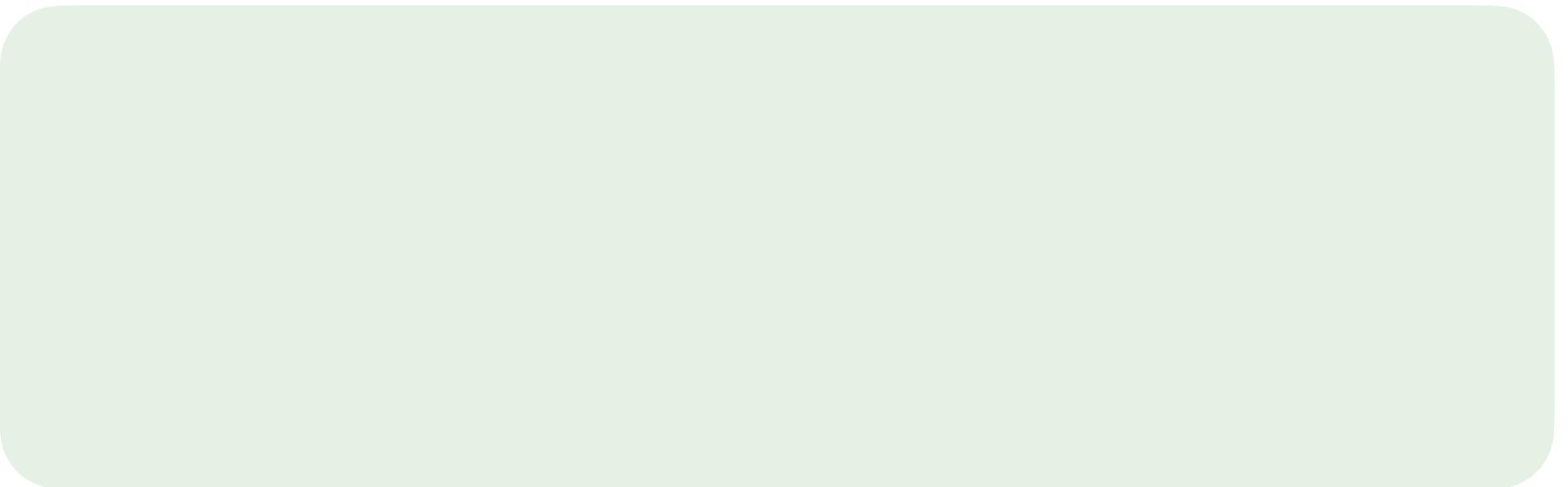
1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$



Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

4. **Claim*:** For every $f \in \mathcal{O}$, there exists a large set $Q_f \subseteq \{0,1\}^{t(n)}$ such that

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

4. **Claim*:** For every $f \in \mathcal{O}$, there exists a large set $Q_f \subseteq \{0,1\}^{t(n)}$ such that
 $\forall w \in Q_f, \forall i \in [t(n)], \Pr[V(x) \text{ queries } f \text{ at } w\|i]$ is small.

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Intuition: without a long argument string, argument verifier cannot make meaningful queries!

4. **Claim*:** For every $f \in \mathcal{O}$, there exists a large set $Q_f \subseteq \{0,1\}^{t(n)}$ such that
 $\forall w \in Q_f, \forall i \in [t(n)], \Pr[V(x) \text{ queries } f \text{ at } w\|i] \text{ is small.}$

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Intuition: without a long argument string, argument verifier cannot make meaningful queries!

4. **Claim*:** For every $f \in \mathcal{O}$, there exists a large set $Q_f \subseteq \{0,1\}^{t(n)}$ such that

$$\forall w \in Q_f, \forall i \in [t(n)], \Pr[V(x) \text{ queries } f \text{ at } w\|i] \text{ is small.}$$
5. Soundness of ARG + $x \notin L_{f^\star} \implies \Pr[V^{f^\star}(x, \pi_{f^\star}) = 1]$ is small for efficiently generated π_{f^\star} .

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^\star \in \mathcal{O}$ such that $x \notin L_{f^\star}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^\star , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^\star	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Intuition: without a long argument string, argument verifier cannot make meaningful queries!

4. **Claim*:** For every $f \in \mathcal{O}$, there exists a large set $Q_f \subseteq \{0,1\}^{t(n)}$ such that

$$\forall w \in Q_f, \forall i \in [t(n)], \Pr[V(x) \text{ queries } f \text{ at } w\|i] \text{ is small.}$$
5. Soundness of ARG + $x \notin L_{f^\star} \implies \Pr[V^{f^\star}(x, \pi_{f^\star}) = 1]$ is small for efficiently generated π_{f^\star} .
6. Point 4 $\implies \forall w \in Q_{f^\star}, \Pr[V^{f_w}(x, \pi_{f^\star}) = 1] \approx \Pr[V^{f^\star}(x, \pi_{f^\star}) = 1]$.

Proof outline

1. Fix $x := 0^n$ for some n .
2. Consider $f^* \in \mathcal{O}$ such that $x \notin L_{f^*}$.
3. For every $w \in \{0,1\}^{t(n)}$, define f_w to be f^* , except that $f_w(w\|i)_1 = 0$ for every $i \in [t(n)]$.

- $f_w \in \mathcal{O}$.
- $x \in L_{f_w}$.

$$L_f := \left\{ x \in \{0,1\}^n : \begin{array}{l} x = 0^n \\ \wedge \exists w \in \{0,1\}^{t(n)}, \forall i \in [t(n)], f(w\|i)_1 = 0 \end{array} \right\}$$

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Intuition: without a long argument string, argument verifier cannot make meaningful queries!

4. **Claim*:** For every $f \in \mathcal{O}$, there exists a large set $Q_f \subseteq \{0,1\}^{t(n)}$ such that

$$\forall w \in Q_f, \forall i \in [t(n)], \Pr[V(x) \text{ queries } f \text{ at } w\|i] \text{ is small.}$$
5. Soundness of ARG + $x \notin L_{f^*} \implies \Pr[V^{f^*}(x, \pi_{f^*}) = 1]$ is small for efficiently generated π_{f^*} .
6. Point 4 $\implies \forall w \in Q_{f^*}, \Pr[V^{f_w}(x, \pi_{f^*}) = 1] \approx \Pr[V^{f^*}(x, \pi_{f^*}) = 1]$.
7. Point 5 + 6 $\implies \forall w \in Q_{f^*}, \Pr[V^{f_w}(x, \pi_{f^*}) = 1]$ is small, contradicting completeness of ARG.

Discussion and open problems

Low-degree random oracle model

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM

secure against **query-bounded and time-bounded** adversaries

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Our proof:

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Our proof:

Can't generalize, no guarantee that $f_w \in \mathcal{P}[q, d]$.

	$w 1$	$w 2$	$w 3$	$w 4$	$w 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

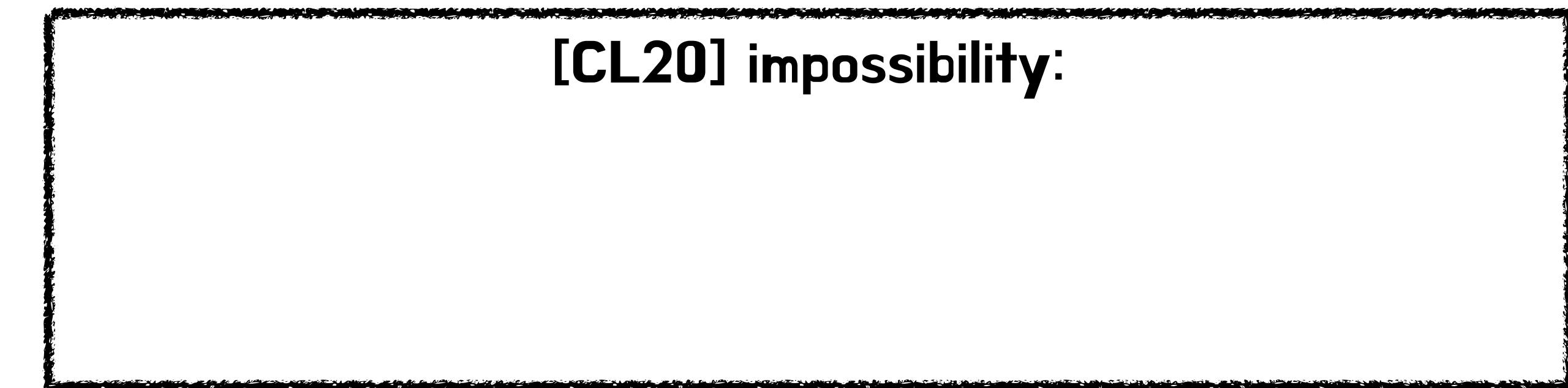
Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Our proof:

Can't generalize, no guarantee that $f_w \in \mathcal{P}[q, d]$.

	$w 1$	$w 2$	$w 3$	$w 4$	$w 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

[CL20] impossibility:



Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Our proof:

Can't generalize, no guarantee that $f_w \in \mathcal{P}[q, d]$.

	$w 1$	$w 2$	$w 3$	$w 4$	$w 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

[CL20] impossibility:

No relativized PCPs in the LDROM

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Our proof:

Can't generalize, no guarantee that $f_w \in \mathcal{P}[q, d]$.

	$w\ 1$	$w\ 2$	$w\ 3$	$w\ 4$	$w\ 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

[CL20] impossibility:

No relativized PCPs in the LDROM
(PCPs are common subroutines in SNARGs constructions)

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Our proof:

Can't generalize, no guarantee that $f_w \in \mathcal{P}[q, d]$.

	$w 1$	$w 2$	$w 3$	$w 4$	$w 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

[CL20] impossibility:

No relativized PCPs in the LDROM
(PCPs are common subroutines in SNARGs constructions)

Caveat: only proved it for specific $f \in \mathcal{P}[q, d]$,
instead of a uniformly sampled $f \leftarrow \mathcal{P}[q, d]$

Low-degree random oracle model

Low-degree random oracle (LDROM) $\mathcal{P} := \{\mathcal{P}_\ell\}_{\ell \in \mathbb{N}}$

\mathcal{P}_ℓ is the uniform distribution over all polynomials $f: \mathbb{F}_{q(\ell)}^{n(\ell)} \rightarrow \mathbb{F}_{q(\ell)}$ of individual degree at most $d(\ell)$.

Open problem 1. Rule out relativized SNARGs in the LDROM, secure against **query-bounded** adversaries.

[CCS22] construction:

Relativized SNARGs in the LDROM
secure against **query-bounded and time-bounded** adversaries

Do they exist or not??

Our proof:

Can't generalize, no guarantee that $f_w \in \mathcal{P}[q, d]$.

	$w 1$	$w 2$	$w 3$	$w 4$	$w 5$
f^*	1001	0111	1110	0000	1010
f_w	0001	0111	0110	0000	0010

[CL20] impossibility:

No relativized PCPs in the LDROM
(PCPs are common subroutines in SNARGs constructions)

Caveat: only proved it for specific $f \in \mathcal{P}[q, d]$,
instead of a uniformly sampled $f \leftarrow \mathcal{P}[q, d]$

Characterization

Characterization

Easy to learn/predict

Hard to learn/predict

Characterization

Easy to learn/predict | Structured Oracle

Hard to learn/predict

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Hard to learn/predict

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Low-Degree Random Oracle

Hard to learn/predict

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Hard to learn/predict

Low-Degree Random Oracle

Possible to construct relativized SNARGs
secure against query-bounded and time-bounded adversaries

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Hard to learn/predict

Low-Degree Random Oracle

Possible to construct relativized SNARGs
secure against query-bounded and time-bounded adversaries

Conjecture: no relativized SNARGs
secure against query-bounded adversaries

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Hard to learn/predict

Random Oracle

Low-Degree Random Oracle

Possible to construct relativized SNARGs
secure against query-bounded and time-bounded adversaries

Conjecture: no relativized SNARGs
secure against query-bounded adversaries

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Hard to learn/predict

Random Oracle

No relativized SNARGs

Low-Degree Random Oracle

Possible to construct relativized SNARGs
secure against query-bounded and time-bounded adversaries

Conjecture: no relativized SNARGs
secure against query-bounded adversaries

Characterization

Easy to learn/predict

Structured Oracle

Easy to construct relativized SNARGs:
Learn the oracle and use non-relativized SNARGs

Hard to learn/predict

Random Oracle

No relativized SNARGs

Low-Degree Random Oracle

Possible to construct relativized SNARGs
secure against query-bounded and time-bounded adversaries

Conjecture: no relativized SNARGs
secure against query-bounded adversaries

Open problem 2.

Give a sufficient and necessary condition for an oracle that separates DTIME/NTIME and relativized arguments.

Insights into Fiat-Shamir

Insights into Fiat-Shamir

Interactive protocol
in the standard model

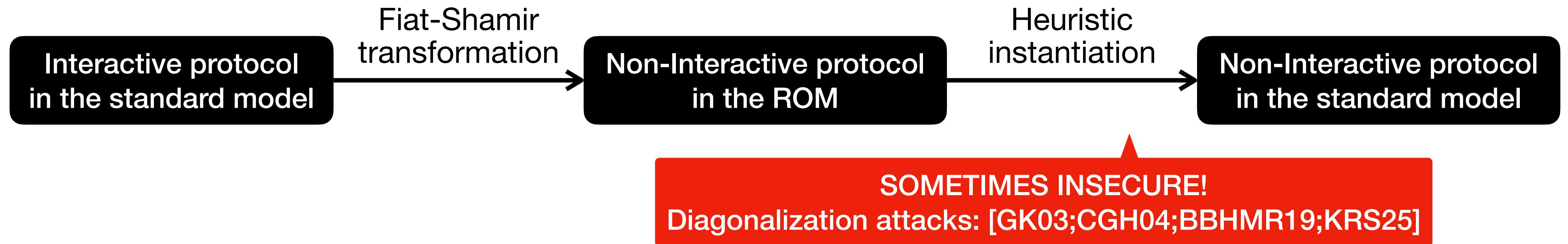
Insights into Fiat-Shamir



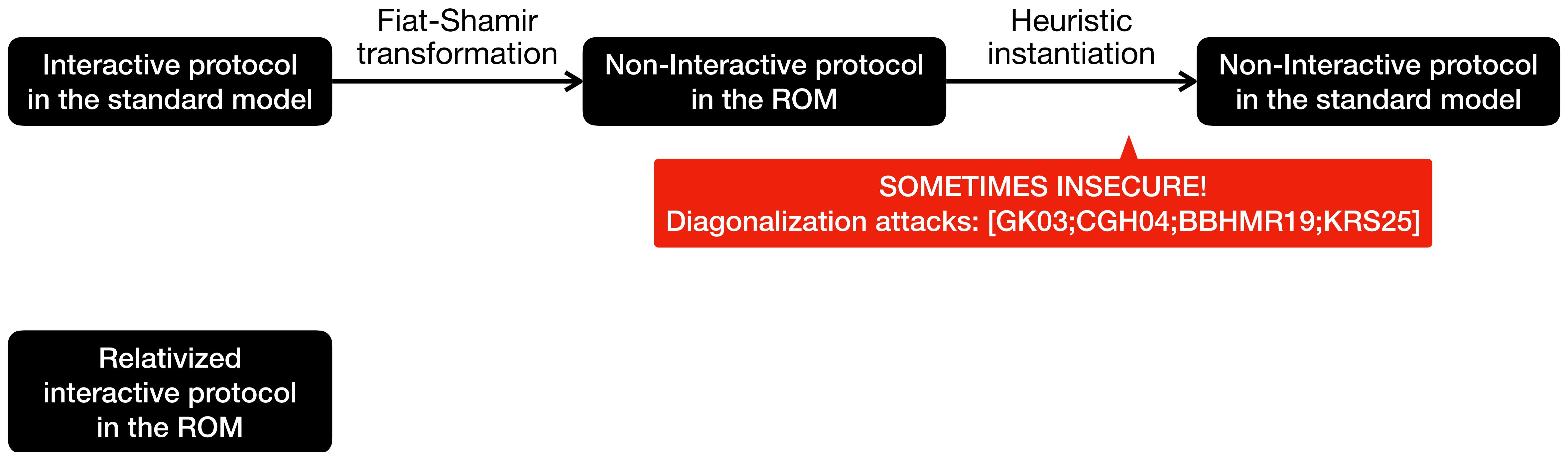
Insights into Fiat-Shamir



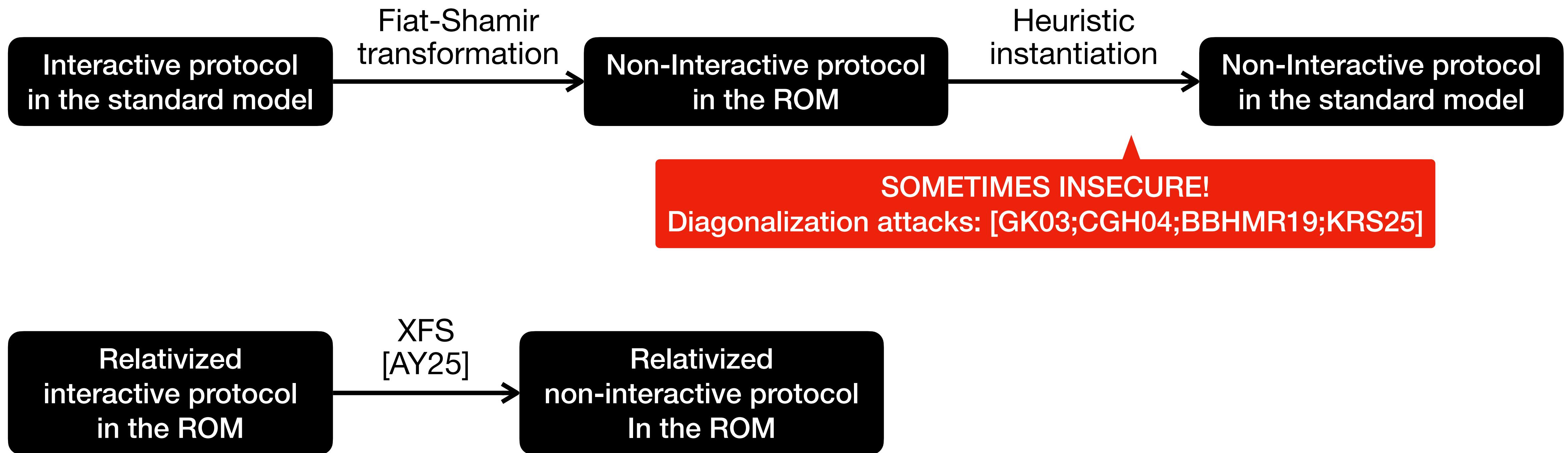
Insights into Fiat-Shamir



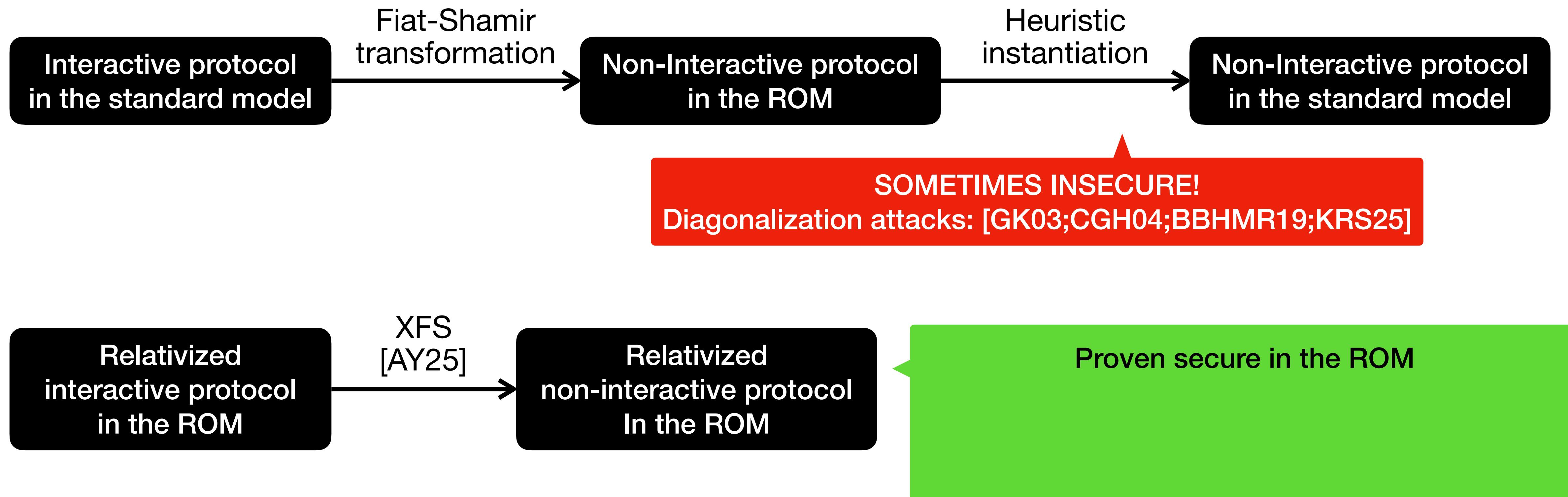
Insights into Fiat-Shamir



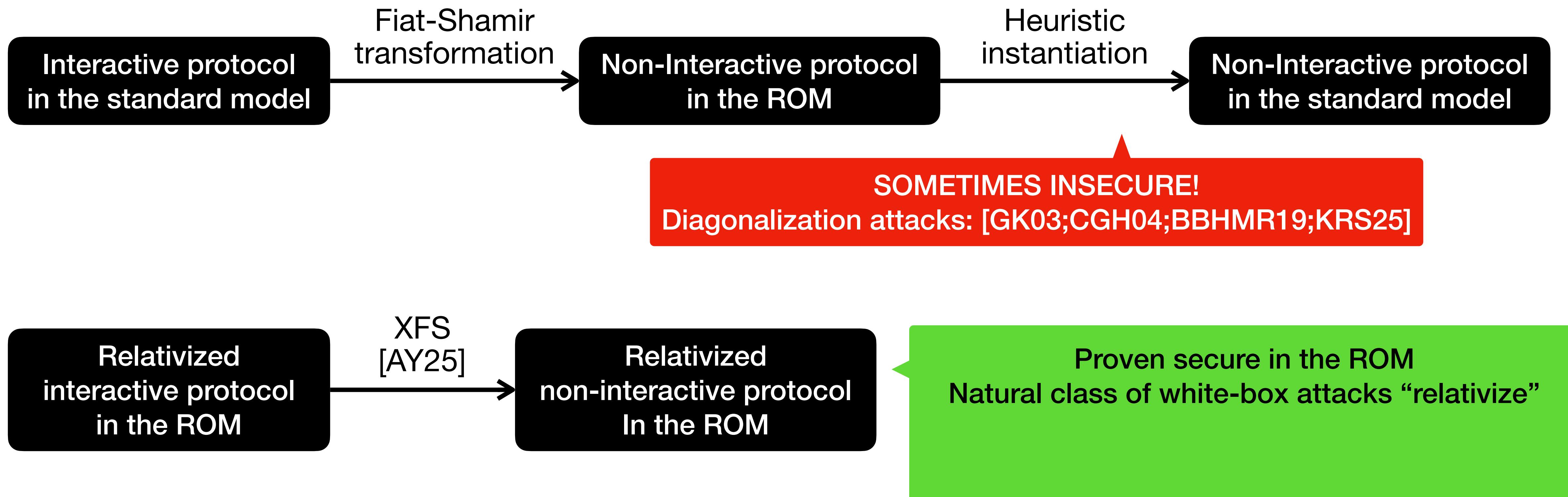
Insights into Fiat-Shamir



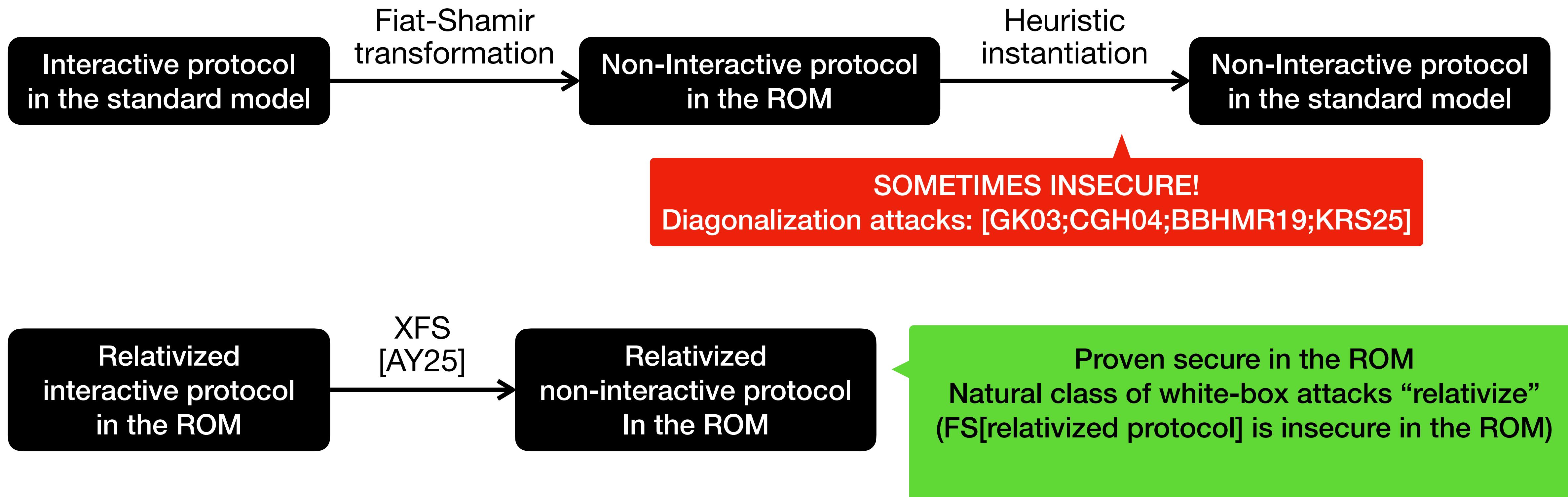
Insights into Fiat-Shamir



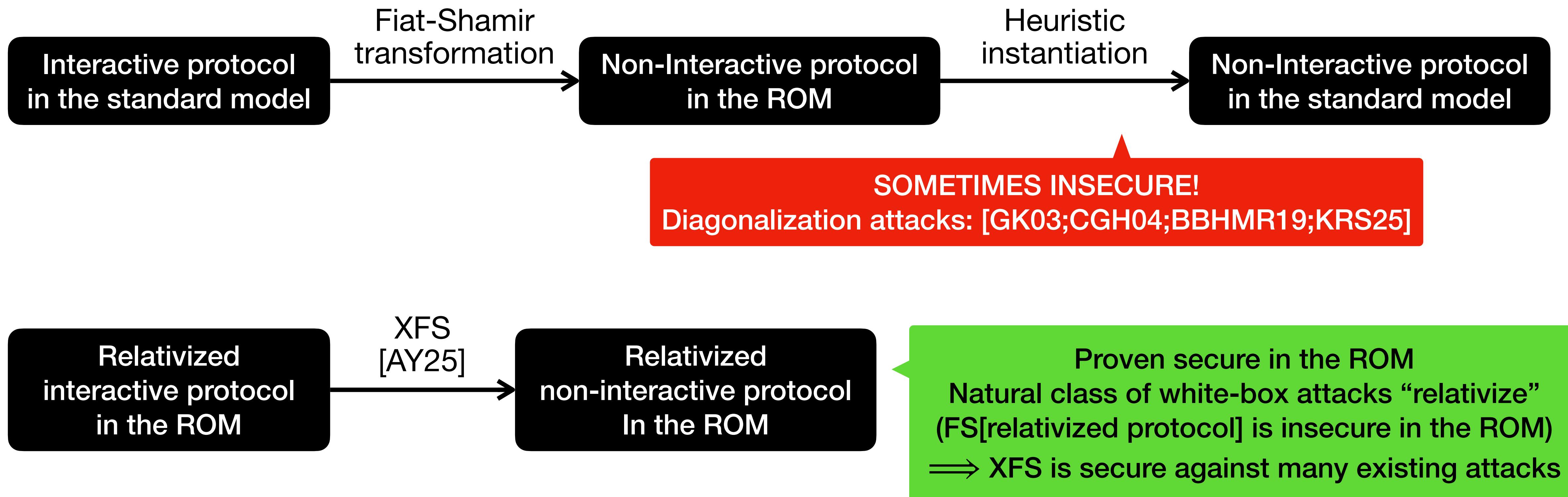
Insights into Fiat-Shamir



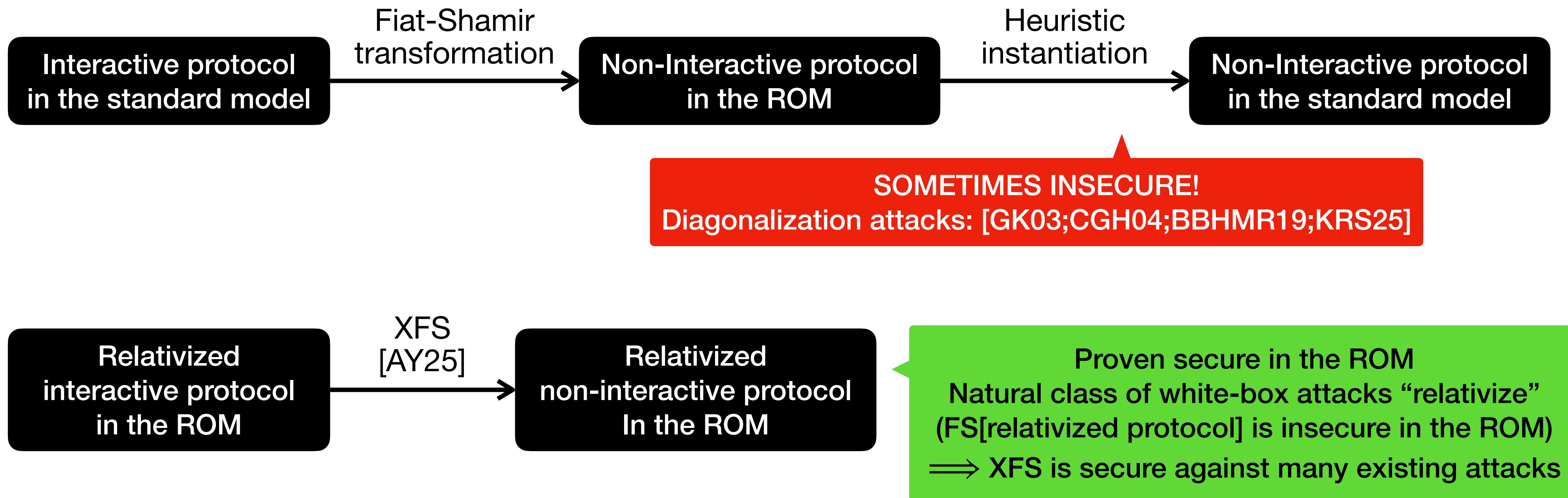
Insights into Fiat-Shamir



Insights into Fiat-Shamir



Insights into Fiat-Shamir



Is Fiat-Shamir transformation secure in other oracle models? LDROM? AROM?

Thank you!

<https://eprint.iacr.org/2024/728>

