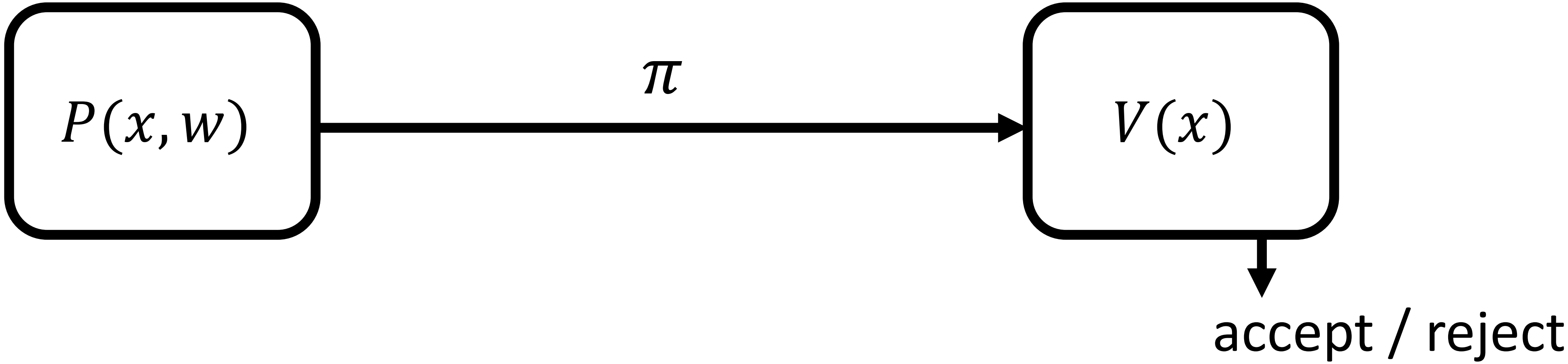
Quasi-Optimal SNARGs via Linear Multi-Prover Interactive Proofs

Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu

Non-Interactive Arguments for NP

$$\mathcal{L}_C = \{x : C(x, w) = 1 \text{ for some } w\}$$



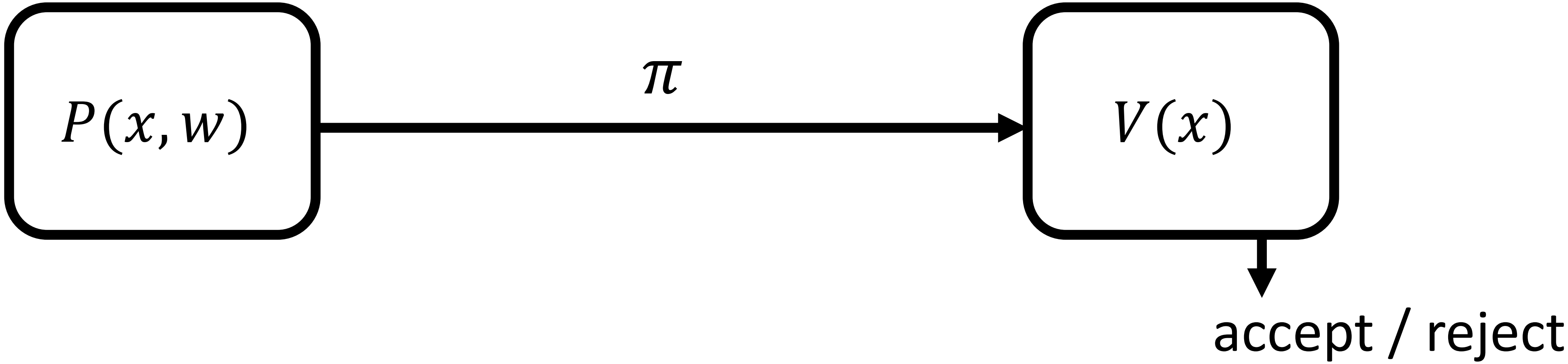
Completeness: $C(x, w) = 1 \Longrightarrow \Pr[\langle P(x, w), V(x)\rangle = 1] = 1$

Soundness: for all provers $P^\star$ of size 2^λ (λ is a security parameter):

$$x \notin \mathcal{L}_C \Longrightarrow \Pr[\langle P^\star(x), V(x)\rangle = 1] \leq 2^{-\lambda}$$

Succinct Non-Interactive Arguments (SNARGs)

$$\mathcal{L}_C = \{x : C(x,w) = 1 \text{ for some } w\}$$



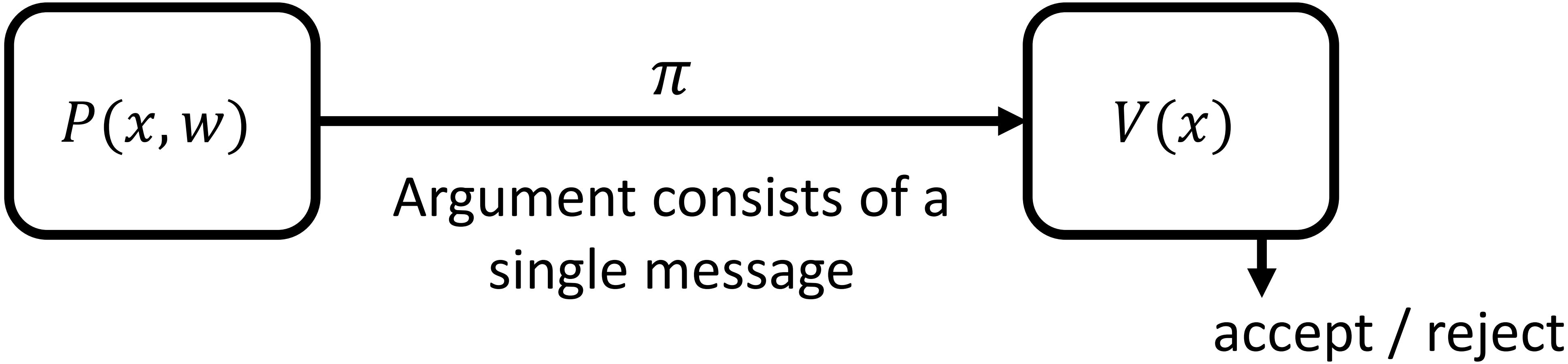
Argument system is *succinct* if:

- Prover communication is $\text{poly}(\lambda + \log|C|)$
- V can be implemented by a circuit of size $\text{poly}(\lambda + |x| + \log|C|)$

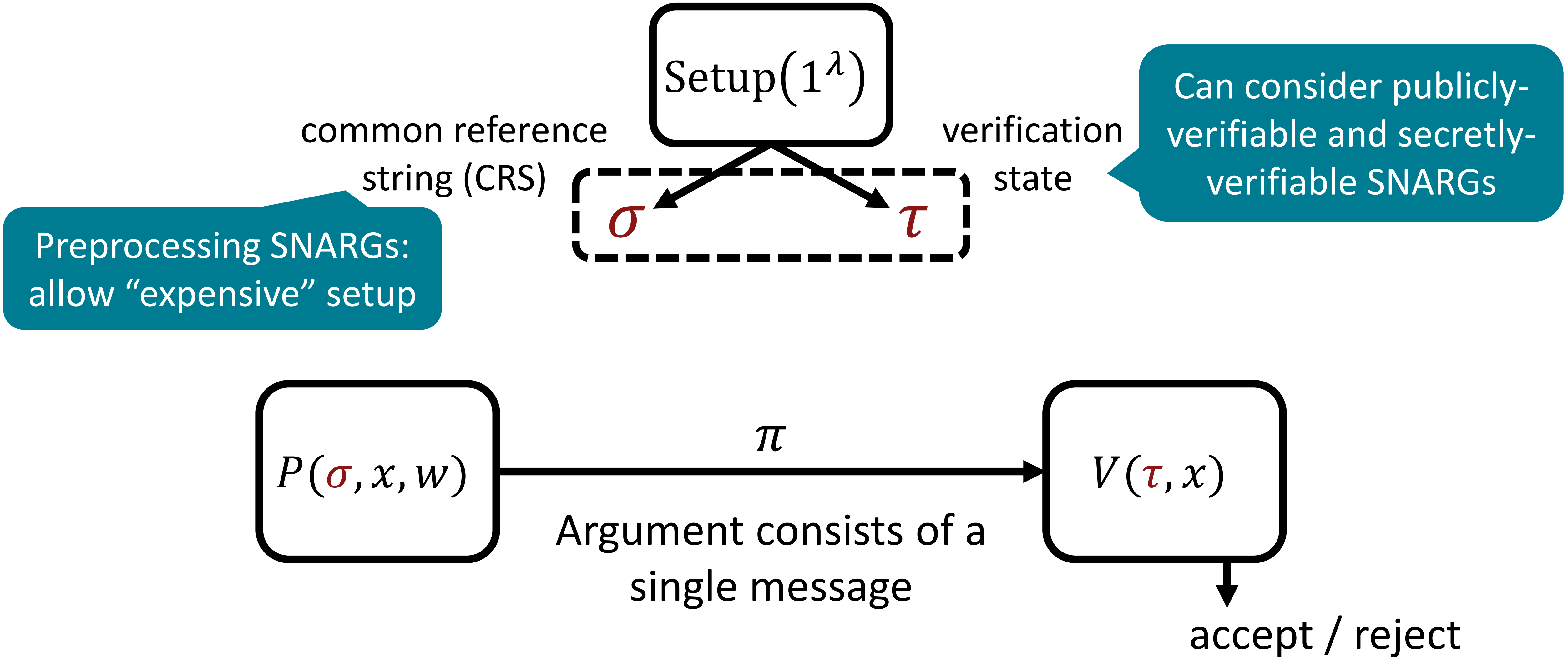
Verifier complexity significantly smaller than classic NP verifier

Succinct Non-Interactive Arguments (SNARGs)

Instantiation: "CS proofs" in the random oracle model [Mic94]



Succinct Non-Interactive Arguments (SNARGs)



Complexity Metrics for SNARGs

Soundness: for all provers $P^\star$ of size 2^λ:

$$x \notin \mathcal{L}_C \Longrightarrow \Pr[\langle P^\star(x), V(x)\rangle = 1] \le 2^{-\lambda}$$

How short can the proofs be?

$$|\pi| = \Omega(\lambda)$$

Even in the designated-verifier setting

[See paper for details]

How much work is needed to generate the proof?

$$|P| = \Omega(|C|)$$

Quasi-Optimal SNARGs

Soundness: for all provers $P^\star$ of size 2^λ:

$$x \notin \mathcal{L}_C \Longrightarrow \Pr[\langle P^\star(x), V(x)\rangle = 1] \le 2^{-\lambda}$$

A SNARG (for Boolean circuit satisfiability) is quasi-optimal if it satisfies the following properties:

- Quasi-optimal succinctness:

$$|\pi| = \lambda \cdot \mathrm{polylog}(\lambda, |C|) = \tilde{O}(\lambda)$$

- Quasi-optimal prover complexity:

$$|P| = \tilde{O}(|C|) + \mathrm{poly}(\lambda, \log|C|)$$

Quasi-Optimal SNARGs

Construction	Prover Complexity	Proof Size	Assumption
CS Proofs [Mic94]	$\tilde{O}(\|C\|)$	$\tilde{O}(\lambda^2)$	Random Oracle
Groth [Gro16]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Generic Group
Groth [Gro10]	$\tilde{O}(\lambda\|C\|^2 + \|C\|\lambda^2)$	$\tilde{O}(\lambda)$	Knowledge of Exponent
GGPR [GGPR12]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	
BCIOP (Pairing) [BCIOP13]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Encryption
BISW (LWE/RLWE) [BISW17]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Vector Encryption

For simplicity, we ignore low order terms $\text{poly}(\lambda, \log|C|)$

Construction	Prover Complexity	Proof Size	Assumption
CS Proofs [Mic94]	$\tilde{O}(\|C\|)$	$\tilde{O}(\lambda^2)$	Random Oracle
Groth [Gro16]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Generic Group
Groth [Gro10]	$\tilde{O}(\lambda\|C\|^2 + \|C\|\lambda^2)$	$\tilde{O}(\lambda)$	Knowledge of Exponent
GGPR [GGPR12]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	
BCIOP (Pairing) [BCIOP13]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Encryption
BISW (LWE/RLWE) [BISW17]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Vector Encryption

For simplicity, we ignore low order terms $\text{poly}(\lambda, \log|C|)$

Construction	Prover Complexity	Proof Size	Assumption
CS Proofs [Mic94]	$\tilde{O}(\|C\|)$	$\tilde{O}(\lambda^2)$	Random Oracle
Groth [Gro16]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Generic Group
Groth [Gro10]	$\tilde{O}(\lambda\|C\|^2 + \|C\|\lambda^2)$	$\tilde{O}(\lambda)$	Knowledge of Exponent
GGPR [GGPR12]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	
BCIOP (Pairing) [BCIOP13]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Encryption
BISW (LWE/RLWE) [BISW17]	$\tilde{O}(\lambda\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Vector Encryption
This work	$\tilde{O}(\|C\|)$	$\tilde{O}(\lambda)$	Linear-Only Vector Encryption

This Work

New framework for building preprocessing SNARGs (following [BCIOP13, BISW17])

Step 1 (information-theoretic):

- Linear multi-prover interactive proofs (linear MIPs)
- This work: first construction of a quasi-optimal linear MIP

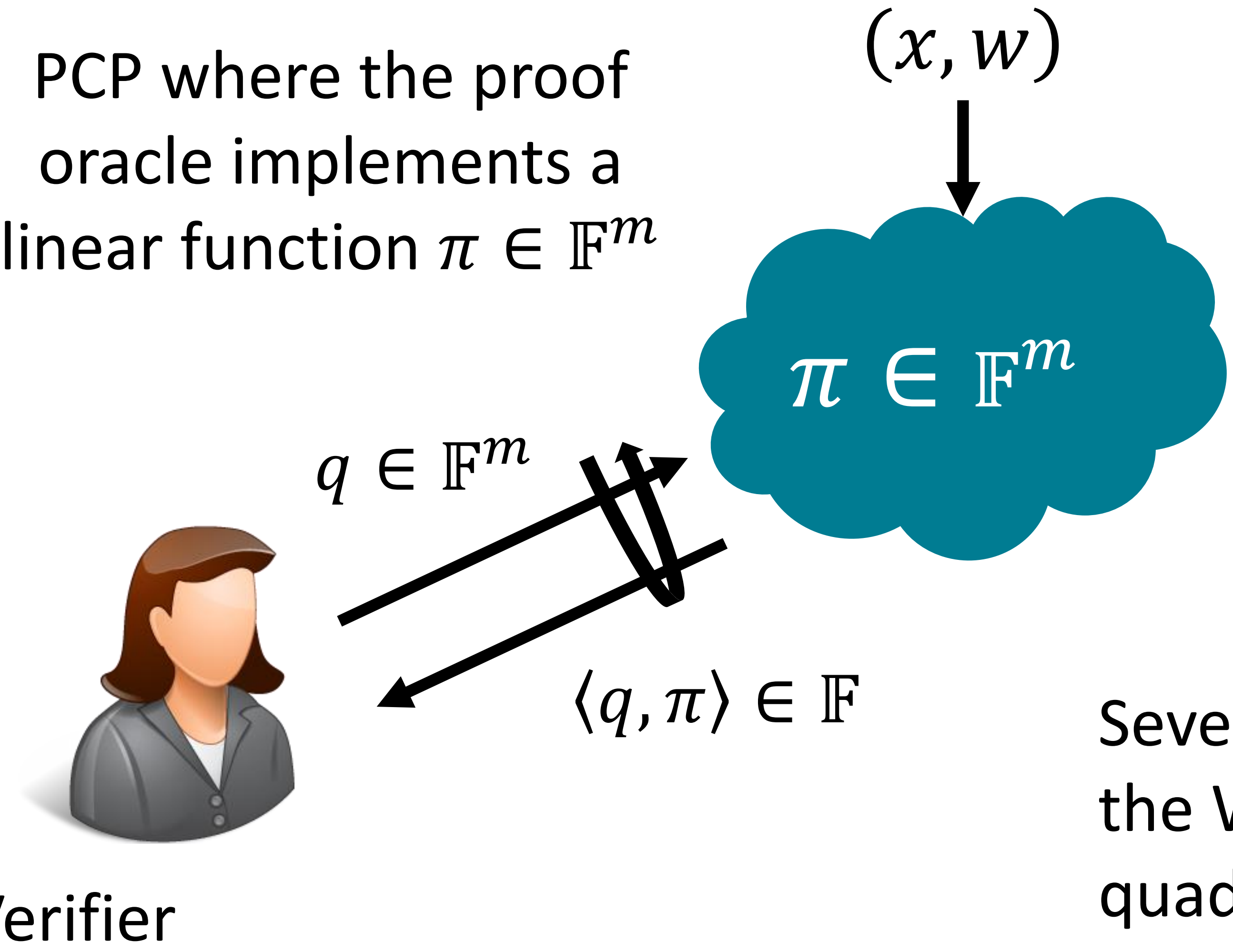
Step 2 (cryptographic):

- Linear-only vector encryption to simulate linear MIP model
- This work: linear MIP $\Longrightarrow$ preprocessing SNARG

Results yield the first quasi-optimal SNARG (from linear-only vector encryption over rings)

Linear PCPs [IKO07]

PCP where the proof oracle implements a linear function $\pi \in \mathbb{F}^m$

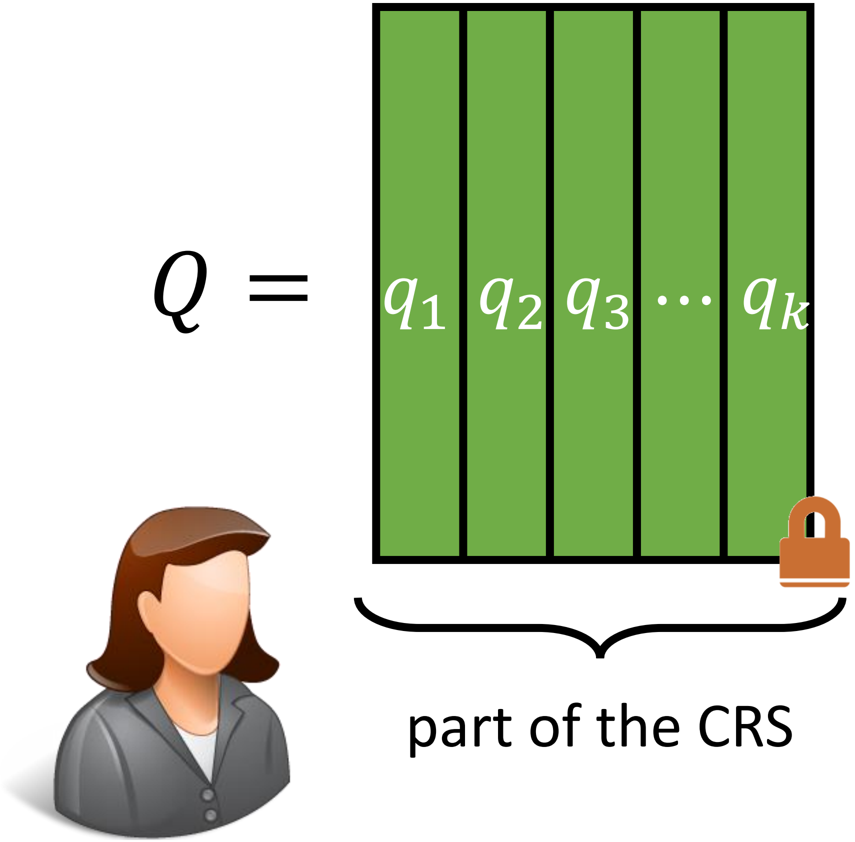


In these instantiations, verifier is <u>oblivious</u> (queries independent of statement)

Several possible instantiations: based on the Walsh-Hadamard code [ALMSS92] or quadratic span programs [GGPR13]

From Linear PCPs to SNARGs [BCIOP13]

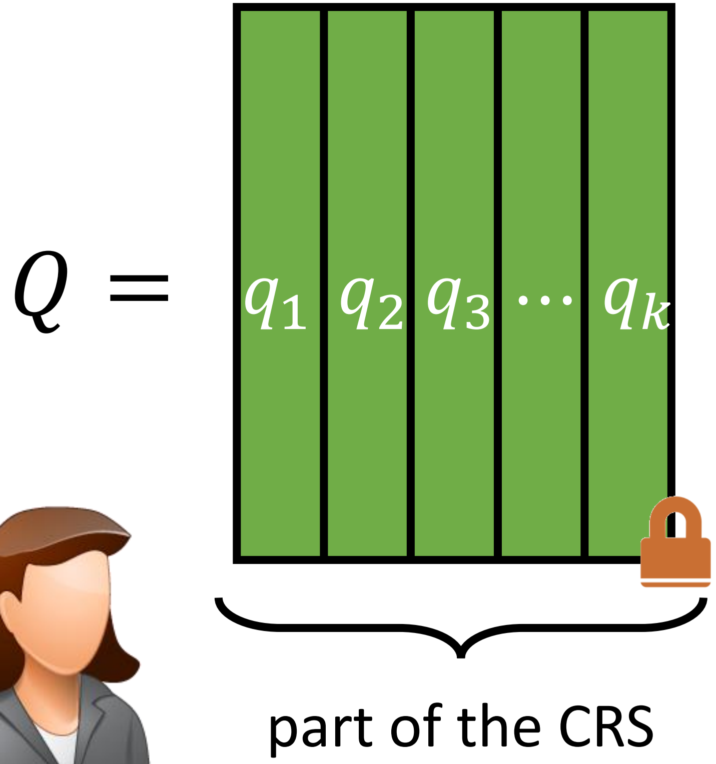
Verifier encrypts its queries using a linear-only encryption scheme



...CPs to SNARGs [BCIOP13]

Encryption scheme that only supports linear homomorphism

Verifier encrypts its queries using a linear-only encryption scheme



From Linear PCPs to SNARGs [BCIOP13]

Verifier encrypts its queries using a linear-only encryption scheme

$$Q = \begin{bmatrix} q_1 & q_2 & q_3 & \cdots & q_k \end{bmatrix}$$

part of the CRS

Prover constructs linear PCP π from (x, w)

$$(x, w) \downarrow \pi \in \mathbb{F}^m$$

Prover homomorphically computes responses to linear PCP queries

$\langle \pi, q_1 \rangle$	$\langle \pi, q_2 \rangle$	$\cdots$	$\langle \pi, q_k \rangle$

SNARG proof

From Linear PCPs to SNARGs [BCIOP13]

Evaluating inner product requires $\Omega(|C|)$ homomorphic operations; prover complexity:
$\Omega(\lambda) \cdot \Omega(|C|) = \Omega(\lambda |C|)$

Prover constructs linear PCP π from (x, w)

(x, w)

We pay $\Omega(\lambda)$ for each homomorphic operation. Can we reduce this?

$Q = \begin{bmatrix} q_1 & q_2 & q_3 & \cdots & q_k \end{bmatrix}$

Prove ... es

responses ... eries

$\langle \pi, q_1 \rangle$	$\langle \pi, q_2 \rangle$	$\cdots$	$\langle \pi, q_k \rangle$

SNARG proof

Proof consists of a constant number of ciphertexts: total length $O(\lambda)$ bits

Linear-Only Encryption over Rings

Consider encryption scheme over a polynomial ring $R_p = \mathbb{Z}_p[x]/\Phi_\ell(x) \cong \mathbb{F}_p^\ell$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_\ell \end{bmatrix} + \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ \vdots \\ x_\ell' \end{bmatrix} = \begin{bmatrix} x_1 + x_1{}' \\ x_2 + x_2' \\ x_3 + x_3' \\ \vdots \\ x_\ell + x_\ell' \end{bmatrix}$$

Homomorphic operations correspond to <u>component-wise</u> additions and scalar multiplications

Plaintext space can be viewed as a vector of field elements

Using RLWE-based encryption schemes, can encrypt $\ell = \tilde{O}(\lambda)$ field elements ($p = \mathrm{poly}(\lambda)$) with ciphertexts of size $\tilde{O}(\lambda)$

Linear-Only Encryption over Rings

Consider encryption scheme over a polynomial ring $R_p = \mathbb{Z}_p[x]/\Phi_\ell(x) \cong \mathbb{F}_p^\ell$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_\ell \end{bmatrix} + \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ \vdots \\ x_\ell' \end{bmatrix} = \begin{bmatrix} x_1 + x_1{}' \\ x_2 + x_2' \\ x_3 + x_3' \\ \vdots \\ x_\ell + x_\ell' \end{bmatrix}$$

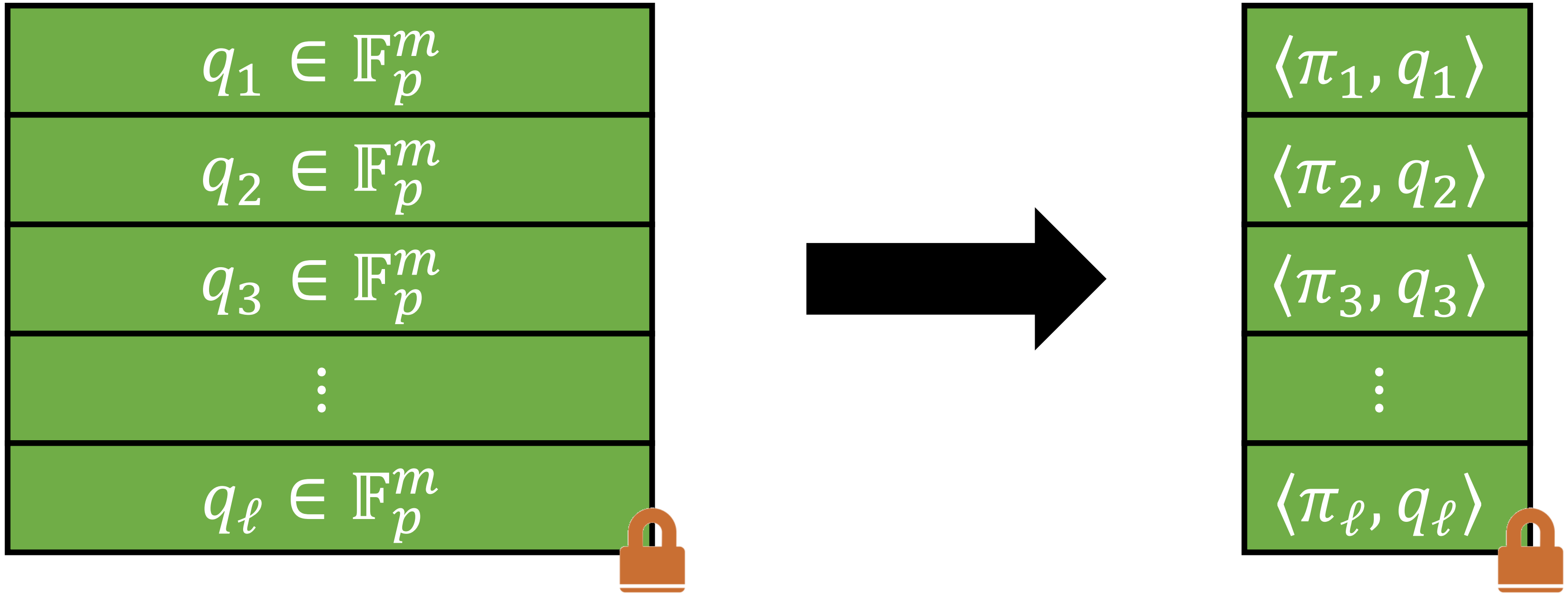
Homomorphic operations

Amortized cost of homomorphic operation on a single field element is $\text{polylog}(\lambda)$

Plaintext space can be viewed as a vector of field elements

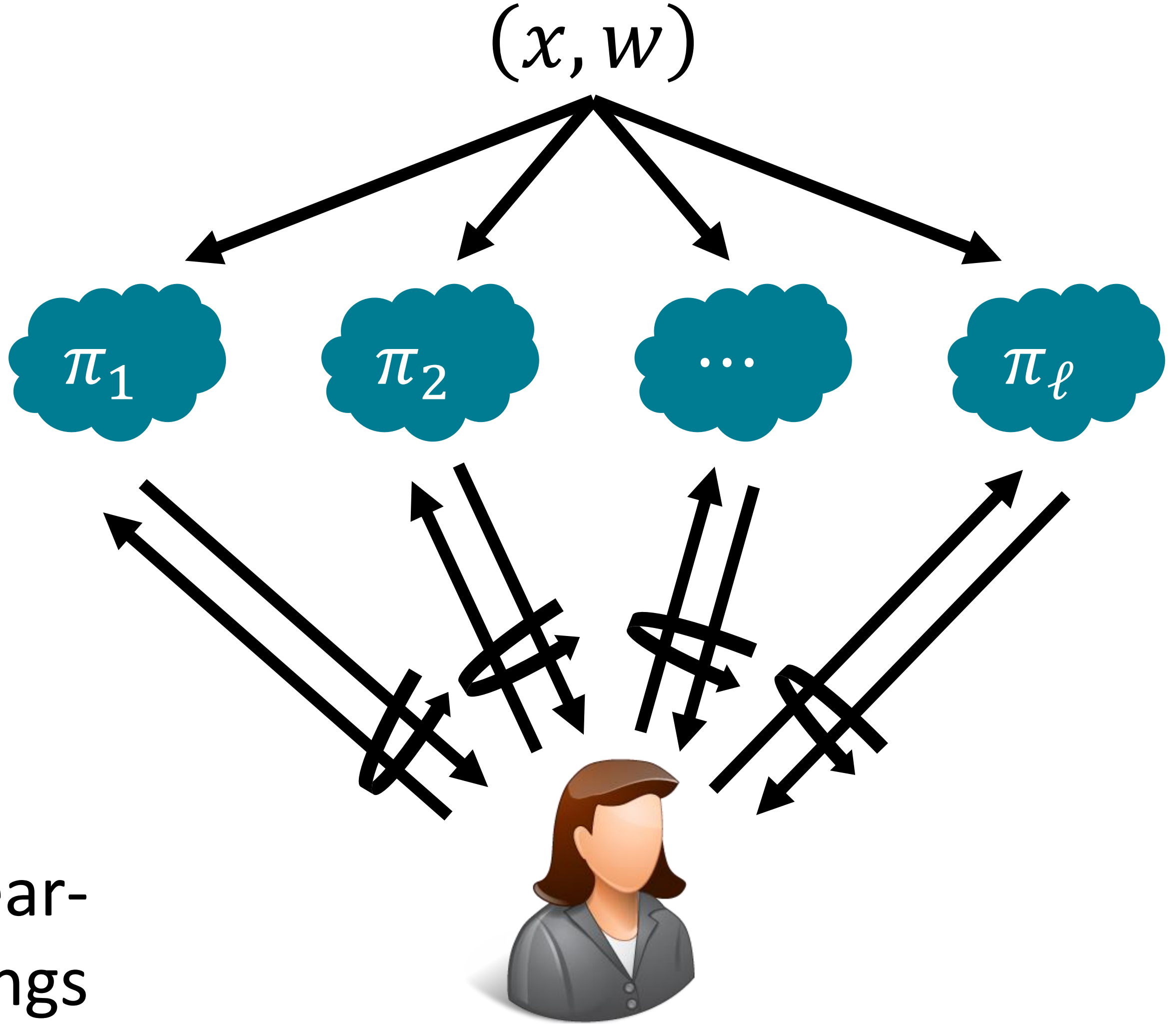
Using RLWE-based encryption schemes, can encrypt $\ell = \tilde{O}(\lambda)$ field elements ($p = \text{poly}(\lambda)$) with ciphertexts of size $\tilde{O}(\lambda)$

Linear-Only Encryption over Rings



Given encrypted set of query vectors, prover can homomorphically apply <u>independent</u> linear functions to each slot

Linear Multi-Prover Interactive Proofs (MIPs)

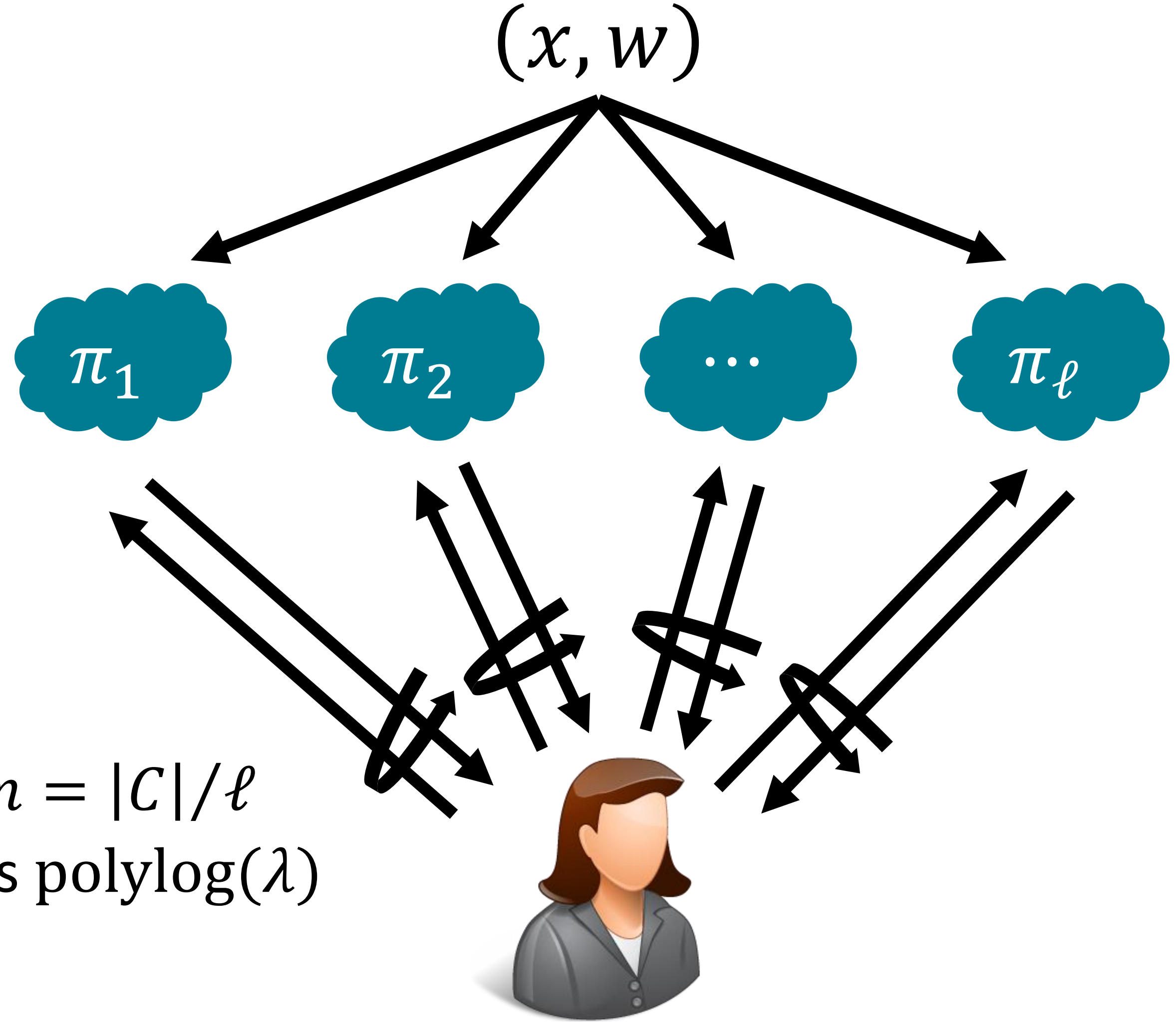


Verifier has oracle access to <u>multiple</u> linear proof oracles

[Proofs may be correlated]

Can convert linear MIP to preprocessing SNARG using linear-only (vector) encryption over rings

Linear Multi-Prover Interactive Proofs (MIPs)



Suppose

- Number of provers $\ell = \tilde{O}(\lambda)$
- Proofs $\pi_1, \ldots, \pi_\ell \in \mathbb{F}_p^m$ where $m = |C|/\ell$
- Number of queries to each π_i is $\text{polylog}(\lambda)$

Then, linear MIP is quasi-optimal

Linear Multi-Prover Interactive Proofs (MIPs)

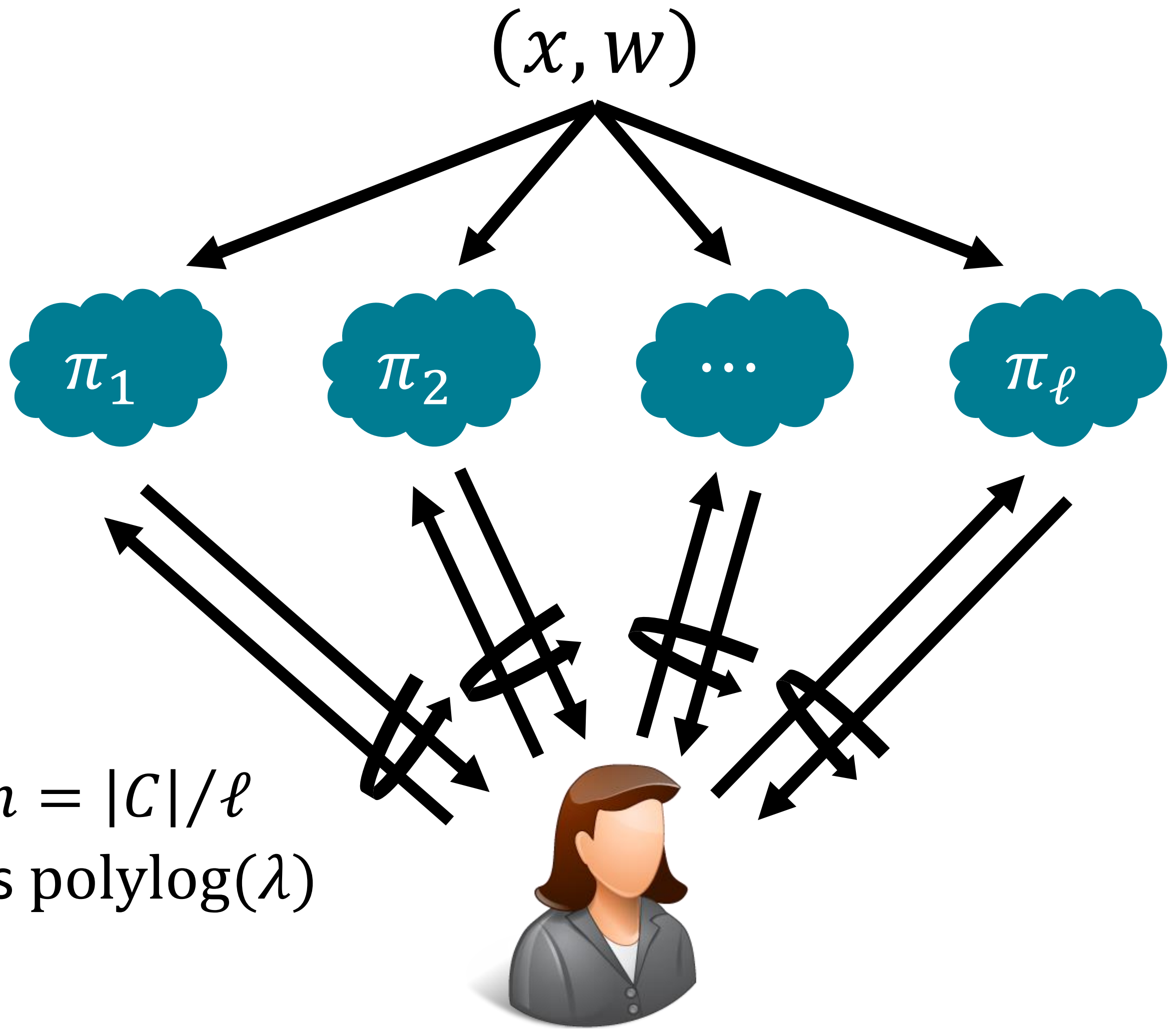
Prover complexity:
$\tilde{O}(\ell m) = \tilde{O}(|C|)$

Linear MIP size:
$O(\ell \cdot \text{polylog}(\lambda)) = \tilde{O}(\lambda)$

Suppose

- Number of provers $\ell = \tilde{O}(\lambda)$
- Proofs $\pi_1, \dots, \pi_\ell \in \mathbb{F}_p^m$ where $m = |C|/\ell$
- Number of queries to each π_i is $\text{polylog}(\lambda)$

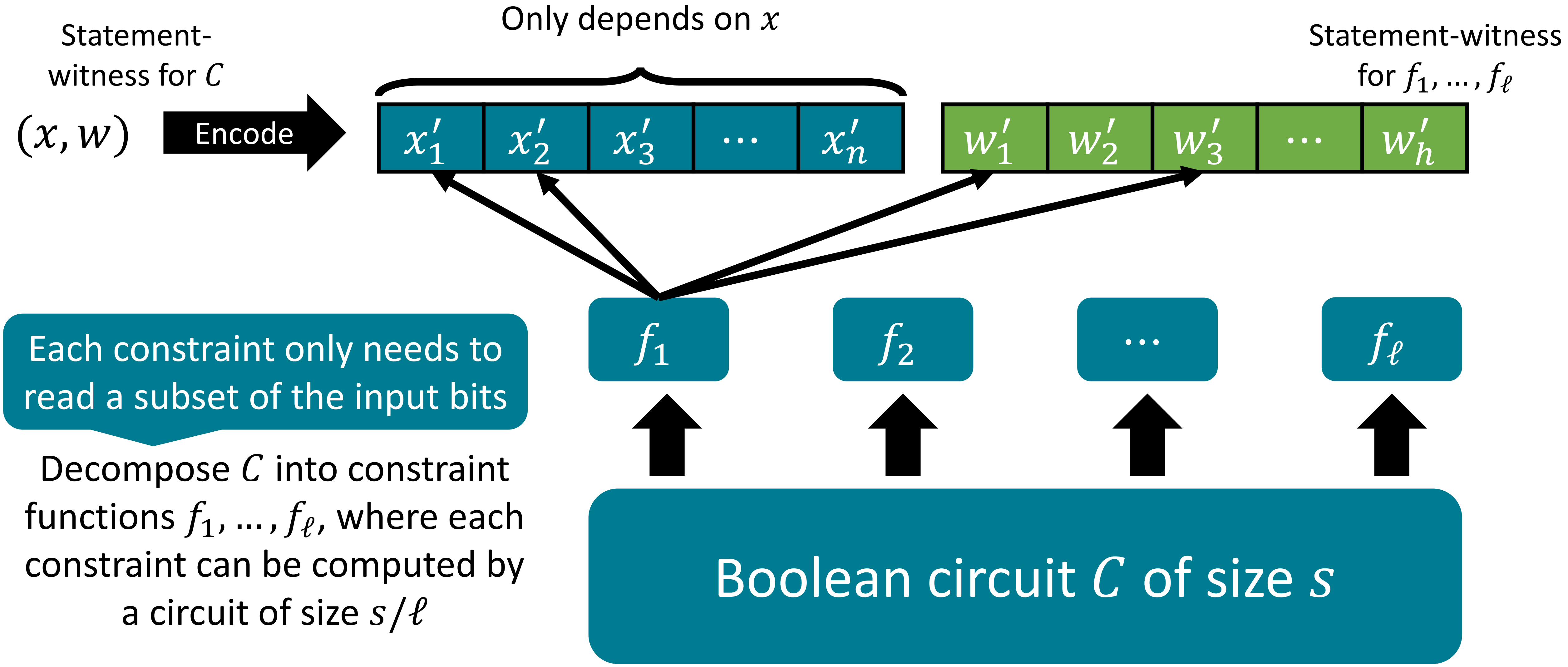
Then, linear MIP is quasi-optimal



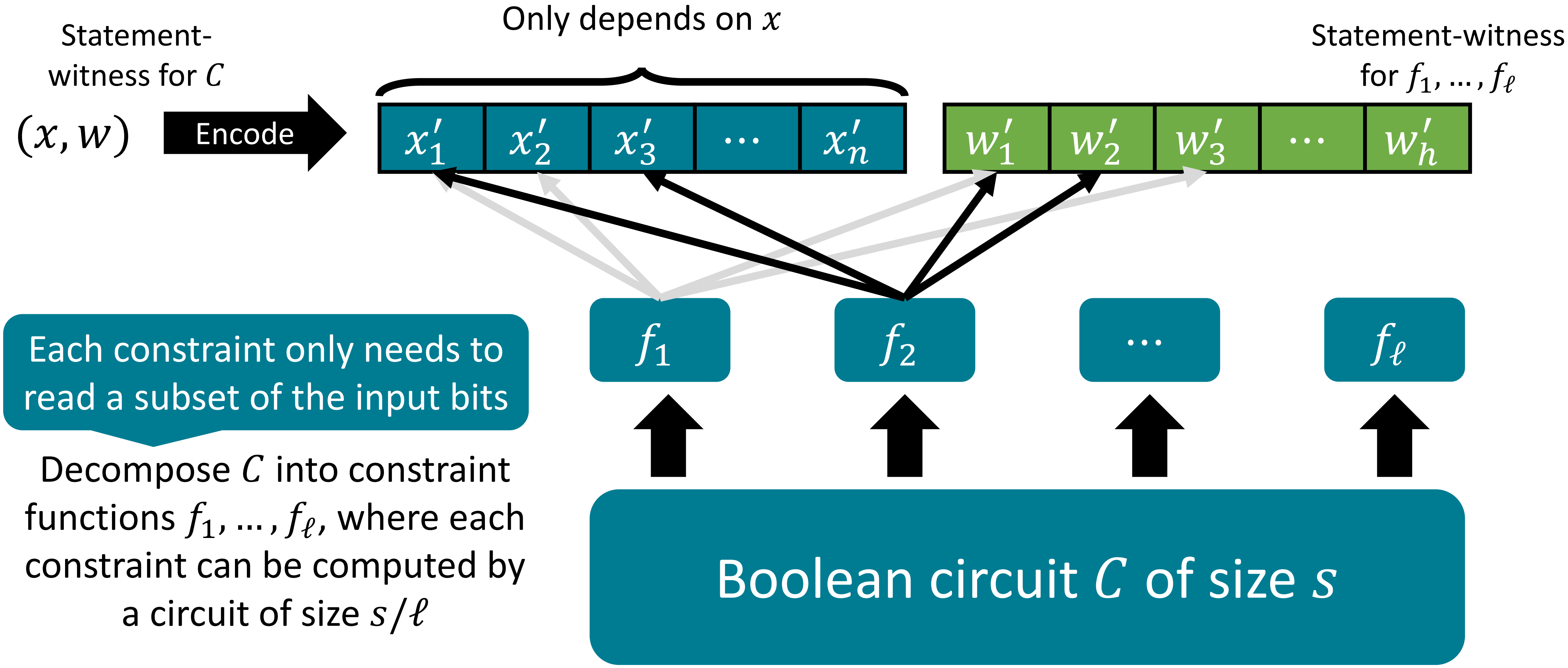
Quasi-Optimal Linear MIPs

This work: Construction of a quasi-optimal linear MIP for Boolean circuit satisfiability

Robust Decomposition



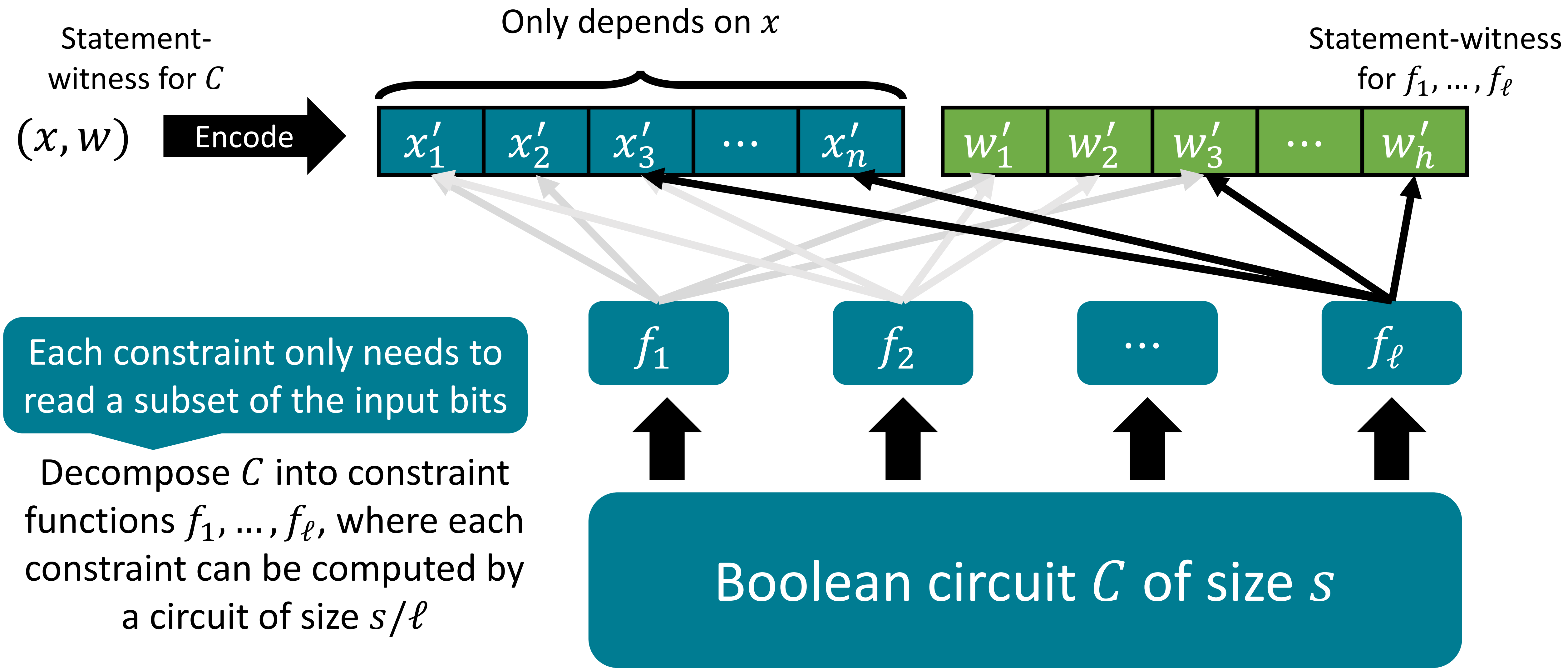
Robust Decomposition



Each constraint only needs to read a subset of the input bits

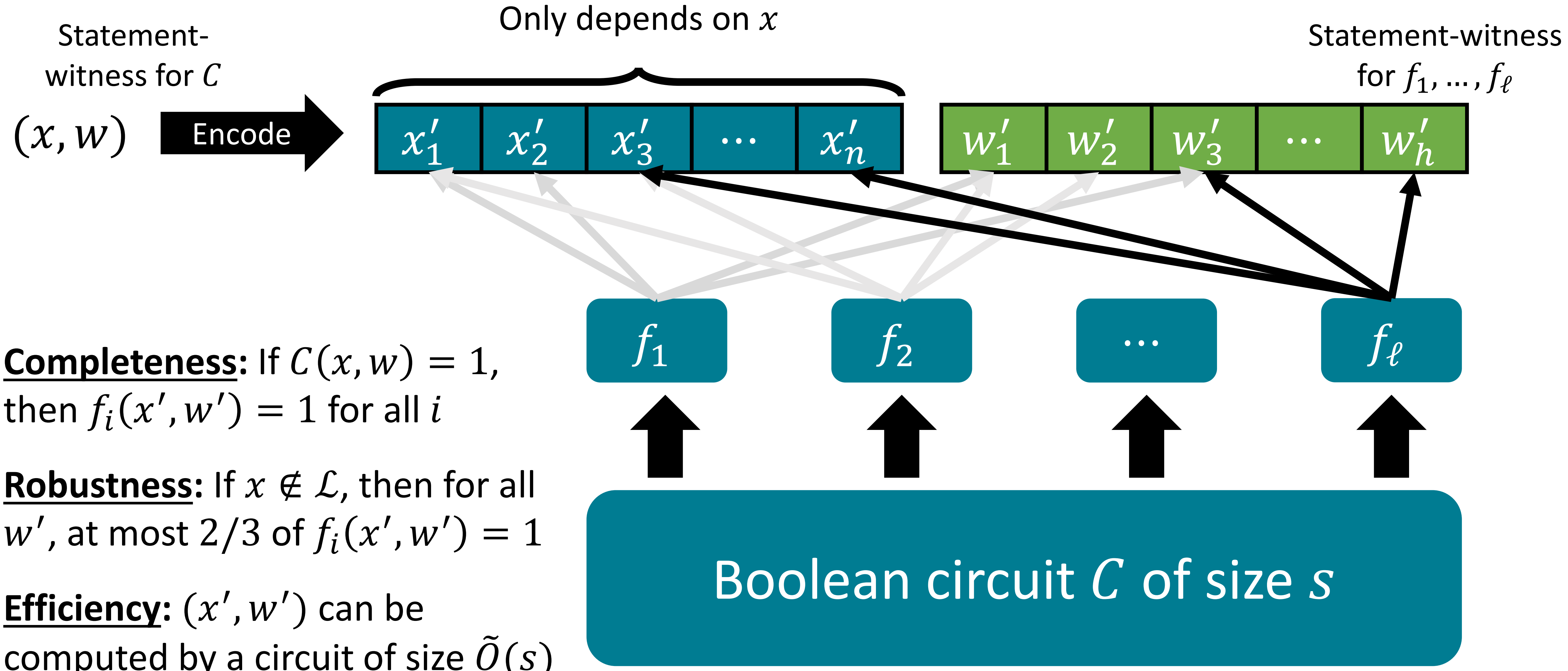
Decompose C into constraint functions $f_1, \ldots, f_\ell$, where each constraint can be computed by a circuit of size s/ℓ

Robust Decomposition



Decompose C into constraint functions $f_1, \ldots, f_\ell$, where each constraint can be computed by a circuit of size s/ℓ

Robust Decomposition

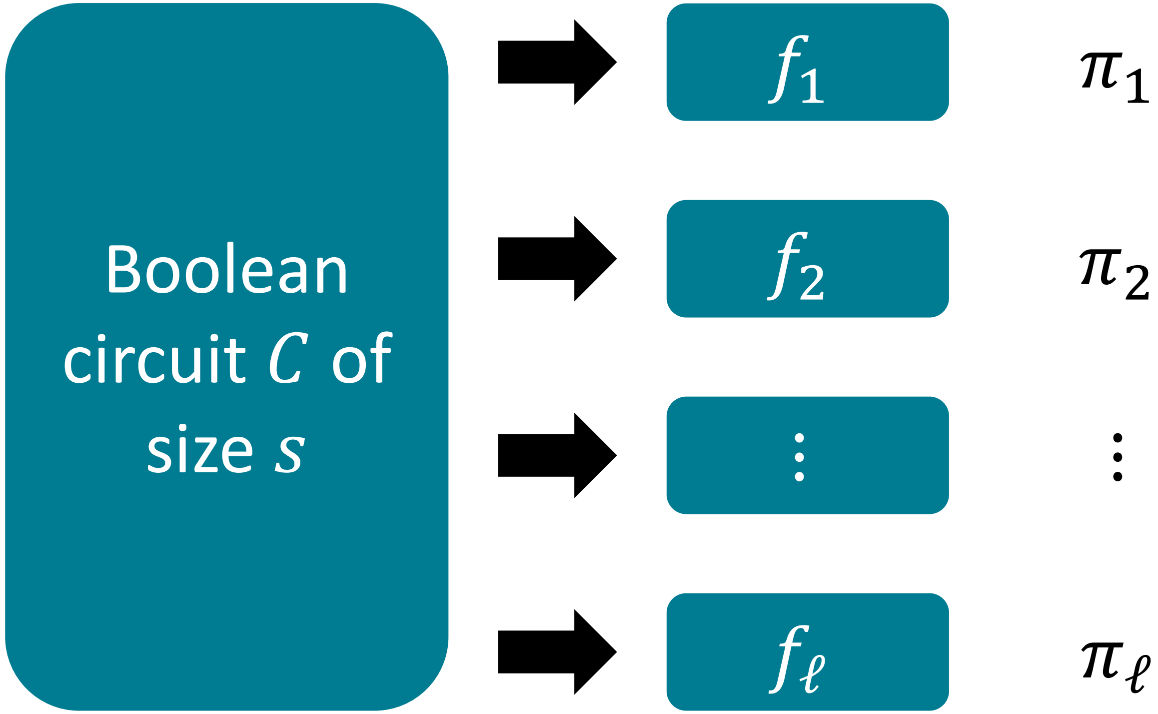


Completeness: If $C(x, w) = 1$, then $f_i(x', w') = 1$ for all i

Robustness: If $x \notin \mathcal{L}$, then for all w', at most 2/3 of $f_i(x', w') = 1$

Efficiency: (x', w') can be computed by a circuit of size $\tilde{O}(s)$

Robust Decomposition



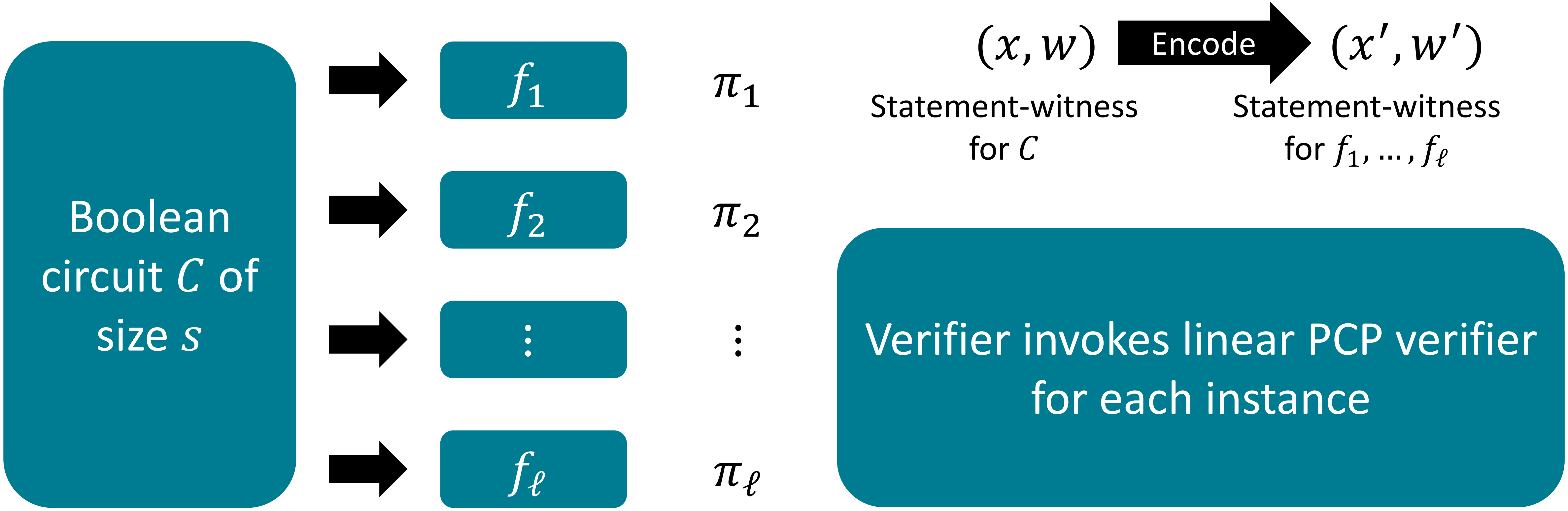
(x, w) **Encode** → (x', w')

Statement-witness for C → Statement-witness for $f_1, \ldots, f_\ell$

Using linear PCP based on QSPs [GGPR13], $|\pi_i| = O(|C|/\ell)$ and provides soundness $1/\mathrm{poly}(\lambda)$

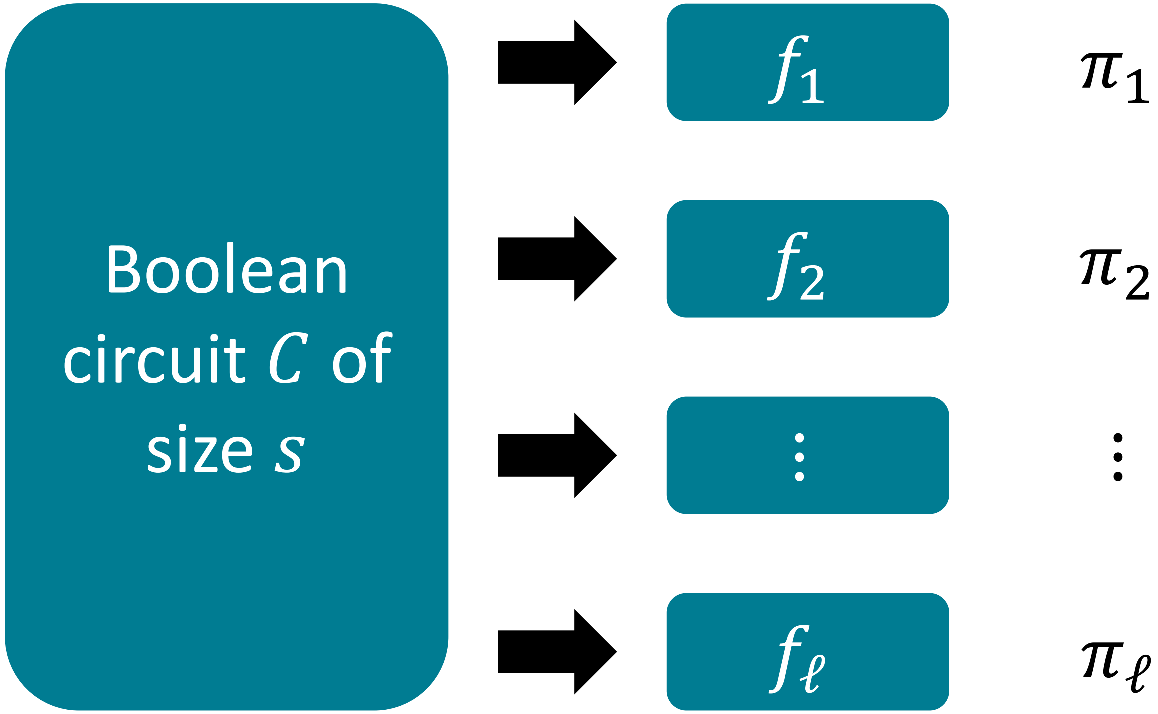
π_i: linear PCP that $f_i(x', \cdot)$ is satisfiable
(instantiated over $\mathbb{F}_p$ where $p = \mathrm{poly}(\lambda)$)

Robust Decomposition



π_i: linear PCP that $f_i(x', \cdot)$ is satisfiable
(instantiated over $\mathbb{F}_p$ where $p = \mathrm{poly}(\lambda)$)

Robust Decomposition



Completeness: Follows by completeness of decomposition and linear PCPs

Soundness: Each linear PCP provides $1/\text{poly}(\lambda)$ soundness and for false statement, at least $1/3$ of the statements are false, so if $\ell = \Omega(\lambda)$, verifier accepts with probability $2^{-\Omega(\lambda)}$

π_i: linear PCP that $f_i(x', \cdot)$ is satisfiable
(instantiated over $\mathbb{F}_p$ where $p = \text{poly}(\lambda)$)

Robust Decomposition

Robustness: If $x \notin \mathcal{L}$, then for all w', at most $2/3$ of $f_i(x', w') = 1$

For false x, no single w' can simultaneously satisfy $f_i(x', \cdot)$; however, all of the $f_i(x', \cdot)$ could individually be satisfiable

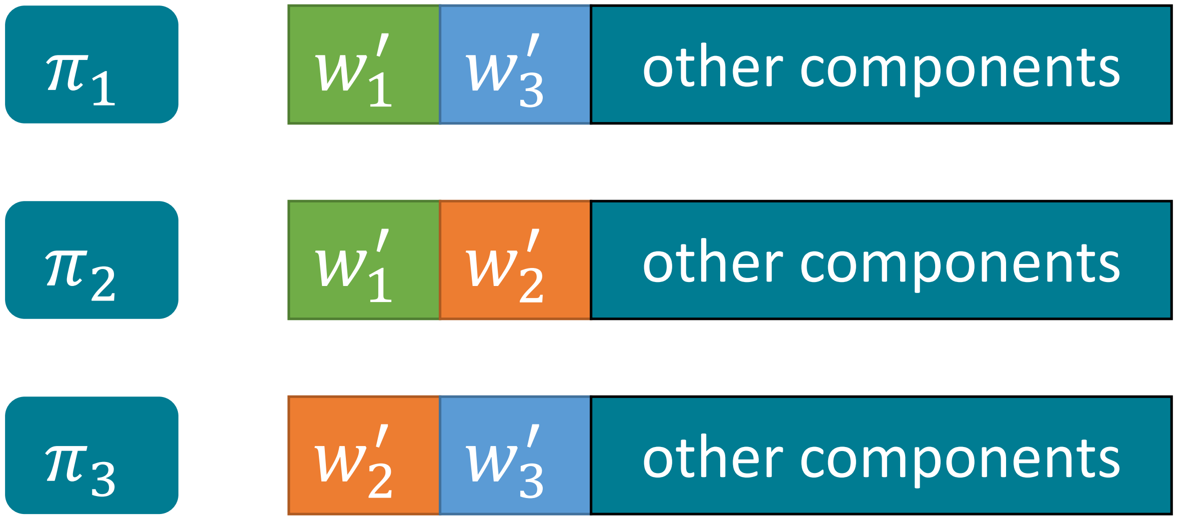
Completeness: Follows by completeness of decomposition and linear PCPs

Soundness: Each linear PCP provides $1/\mathrm{poly}(\lambda)$ soundness and for false statement, at least $1/3$ of the statements are false, so if $\ell = \Omega(\lambda)$, verifier accepts with probability $2^{-\Omega(\lambda)}$

Problematic however if prover uses different (x', w') to construct proofs for different f_i's

Consistency Checking

Require that linear PCPs are systematic: linear PCP π contains a copy of the witness:



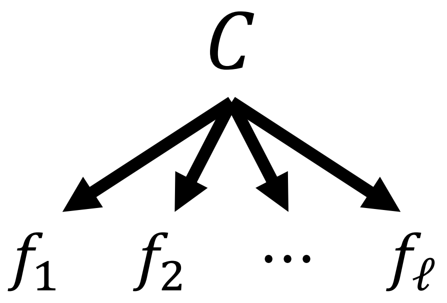
Goal: check that assignments to w' are consistent via linear queries to π_i

First few components of proof correspond to witness associated with the statement → Each proof induces an assignment to a few bits of the common witness w'

[See paper for details]

Quasi-Optimal Linear MIP

<u>Robust Decomposition</u>



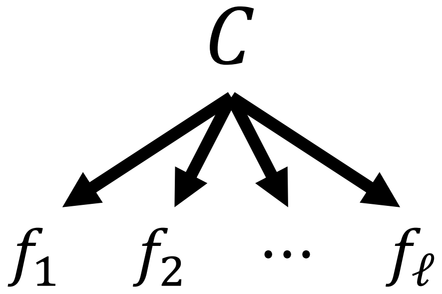
- Checking satisfiability of C corresponds to checking satisfiability of $f_1, \ldots, f_\ell$ (each of which can be checked by a circuit of size $|C|/\ell$)
- For a false statement, no single witness can simultaneously satisfy more than a constant fraction of f_i

Robust decomposition can be instantiated by combining "MPC-in-the-head" paradigm [IKOS07] with a robust MPC protocol with polylogarithmic overhead [DIK10]

[See paper for details]

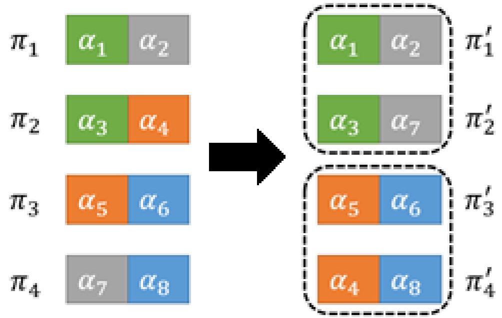
Quasi-Optimal Linear MIP

Robust Decomposition



- Checking satisfiability of C corresponds to checking satisfiability of $f_1, \ldots, f_\ell$ (each of which can be checked by a circuit of size $|C|/\ell$)
- For a false statement, no single witness can simultaneously satisfy more than a constant fraction of f_i

Consistency Check



- Check that consistent witness is used to prove satisfiability of each f_i
- Relies on pairwise consistency checks and permuting the entries to obtain a "nice" replication structure

Conclusions

A SNARG is quasi-optimal if it satisfies the following properties:

- Quasi-optimal succinctness: $|\pi| = \tilde{O}(\lambda)$
- Quasi-optimal prover complexity: $|P| = \tilde{O}(|C|) + \mathrm{poly}(\lambda, \log|C|)$

New framework for building quasi-optimal SNARGs by combining quasi-optimal linear MIP with linear-only vector encryption

- Construction of a quasi-optimal linear MIP possible by combining robust decomposition and consistency check

What if we had a 1-bit SNARG? Implies a form of witness encryption

- Highlights connection between soundness and confidentiality; see also [BDRV18] which shows laconic zero-knowledge implies PKE

[See paper for details]

Open Problems

Publicly-verifiable quasi-optimal SNARGs

- Or: multi-theorem designated-verifier SNARGs

Quasi-optimal zero-knowledge SNARGs

Thank you!

https://eprint.iacr.org/2018/133