Sumcheck Arguments and Lattice-based Succinct arguments

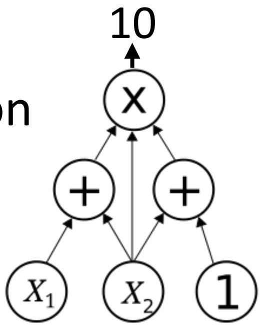
Jonathan Bootle (IBM Research – Zurich)
Alessandro Chiesa (EPFL)
Katerina Sotiraki (Yale University)

https://ia.cr/2021/333
https://ia.cr/2023/930

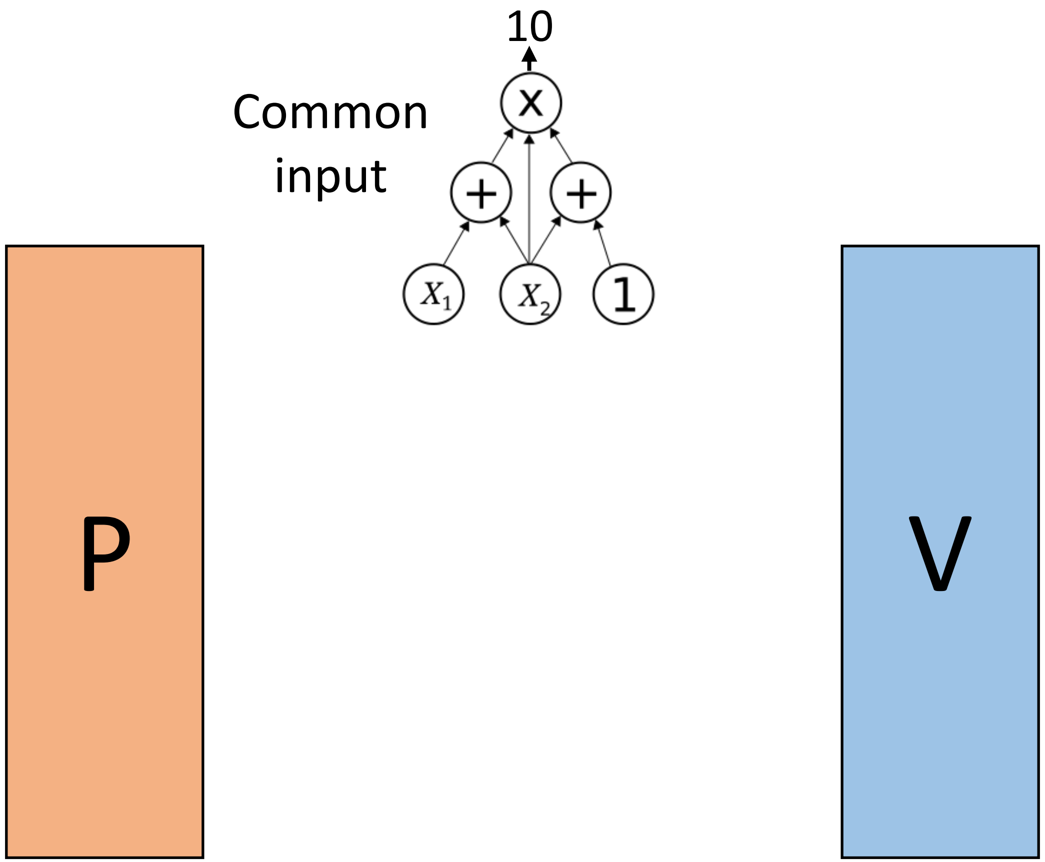
Slides by Jonathan Bootle

Succinct arguments

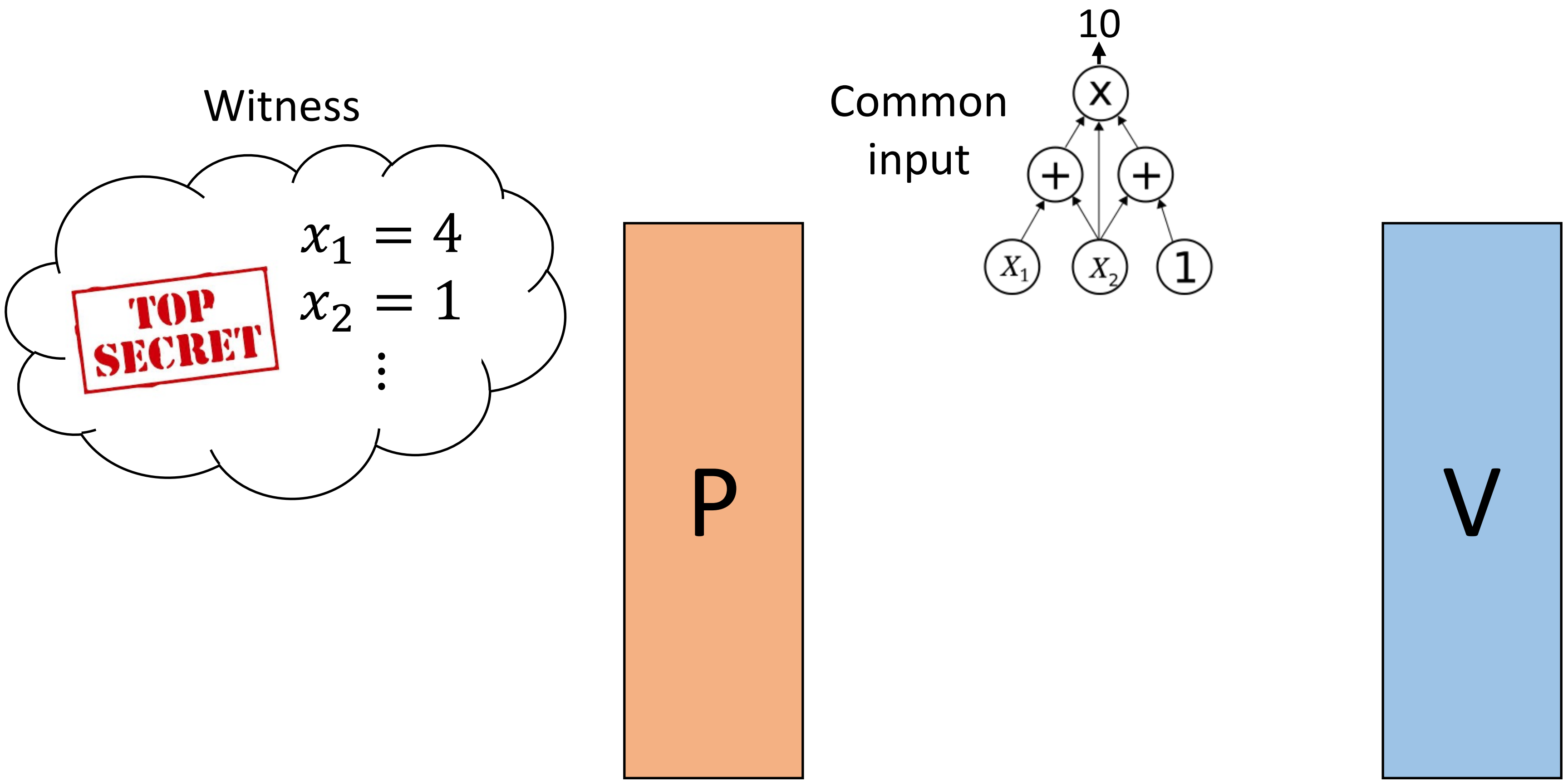
Common input



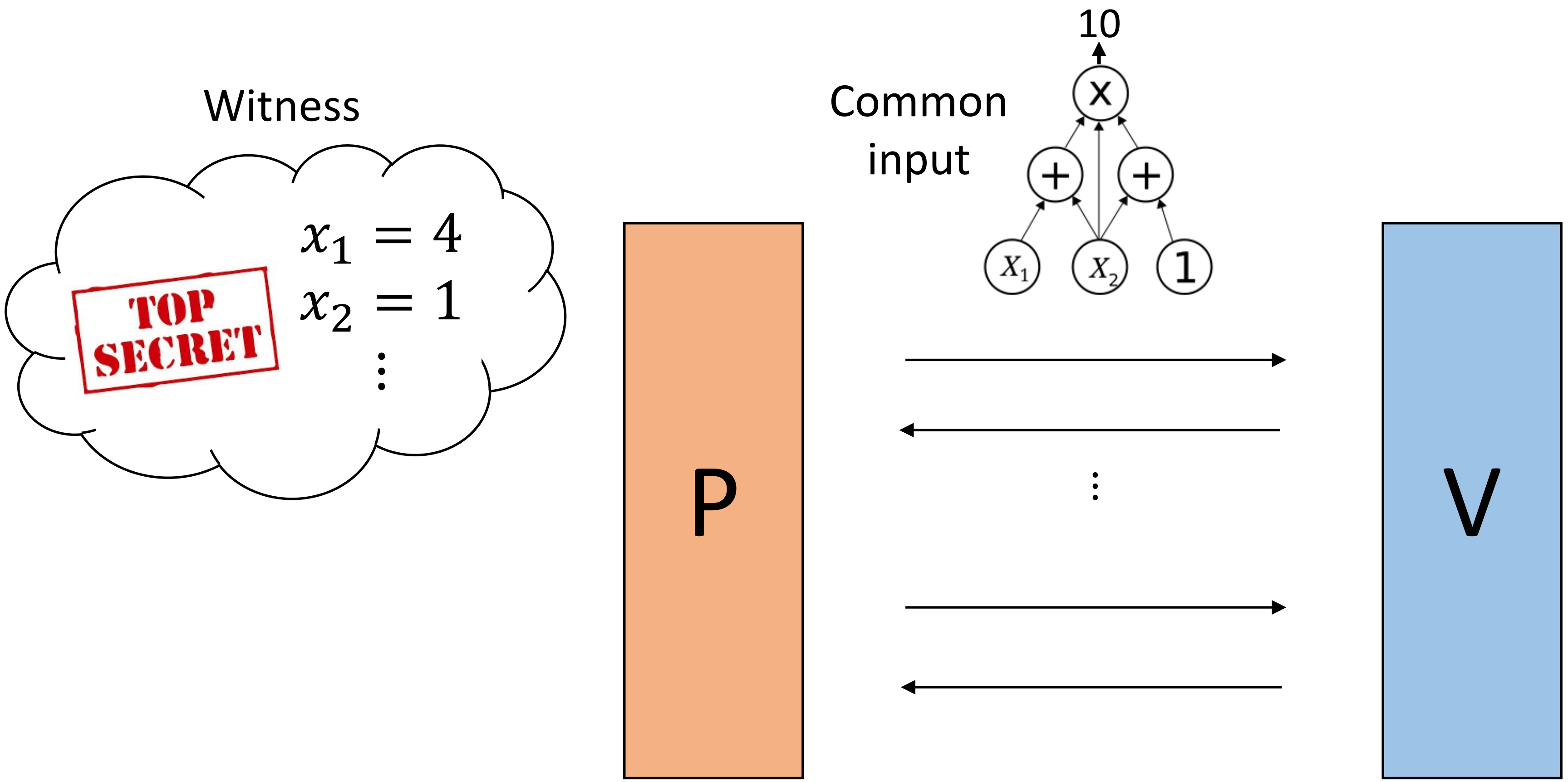
Succinct arguments



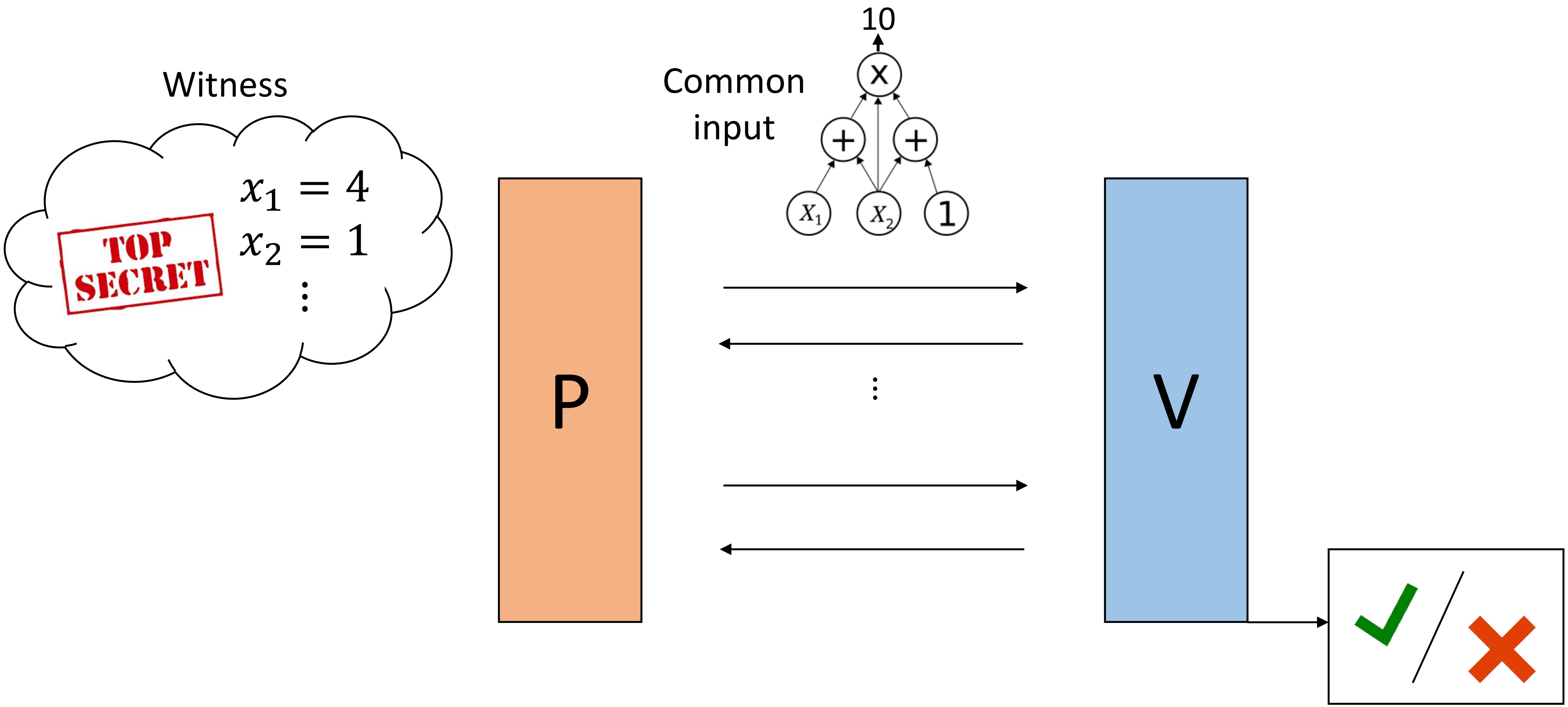
Succinct arguments



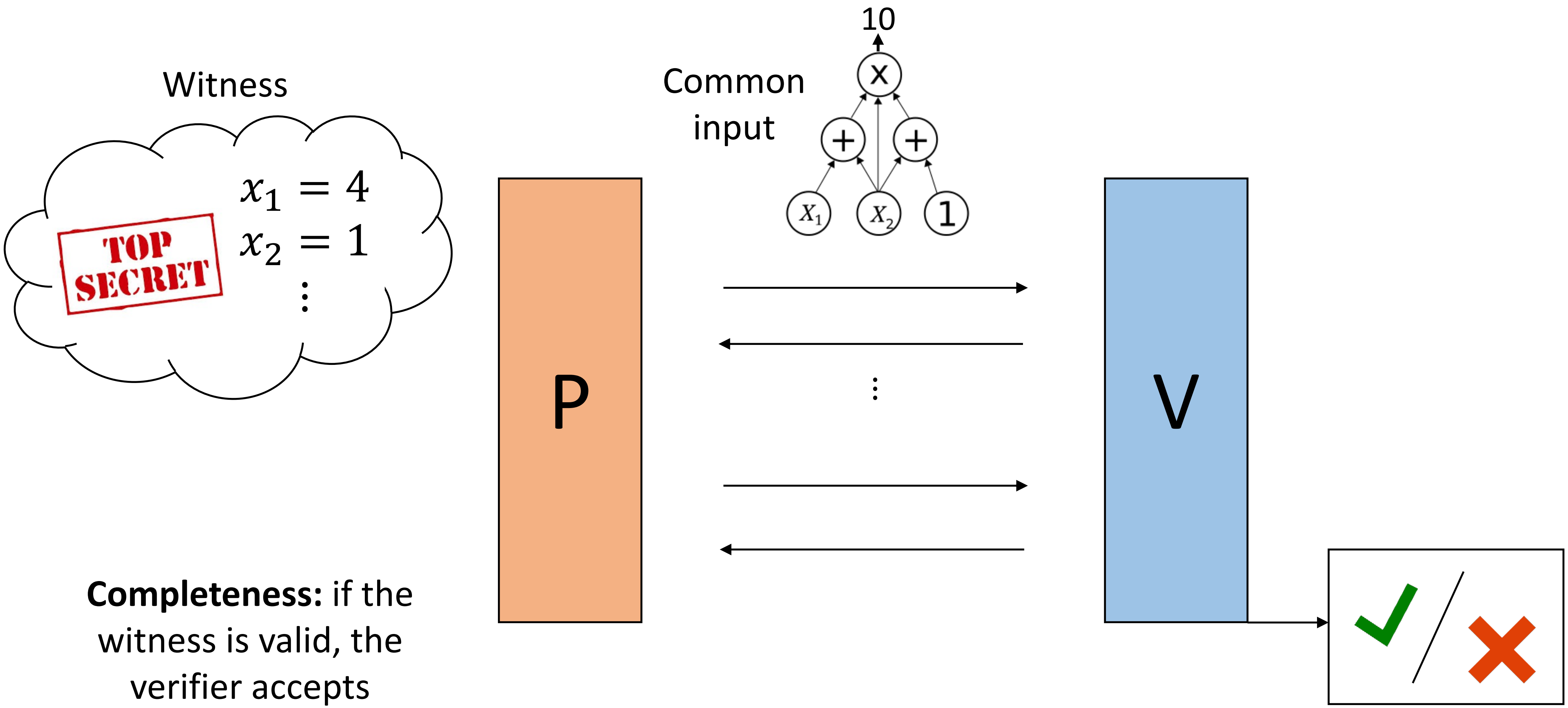
Succinct arguments



Succinct arguments



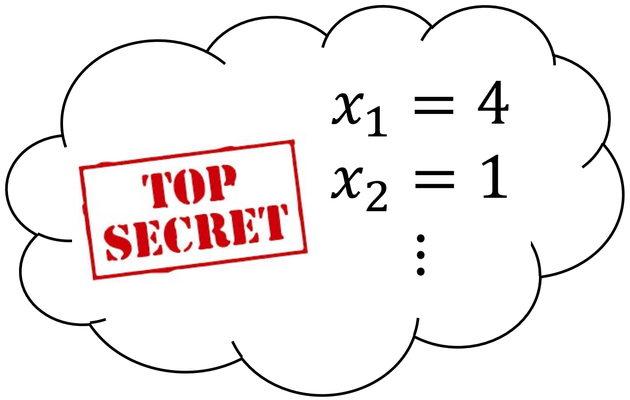
Succinct arguments



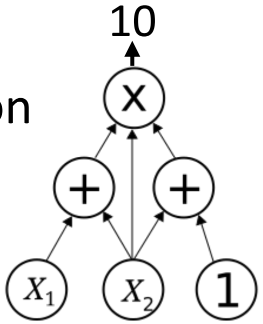
Completeness: if the witness is valid, the verifier accepts

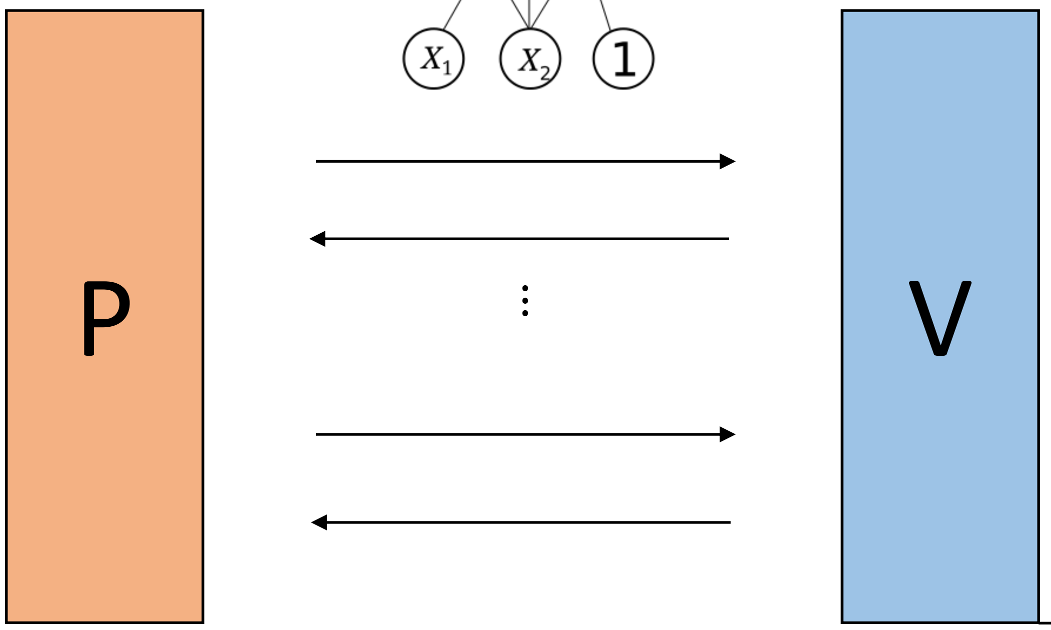
Succinct arguments

Witness



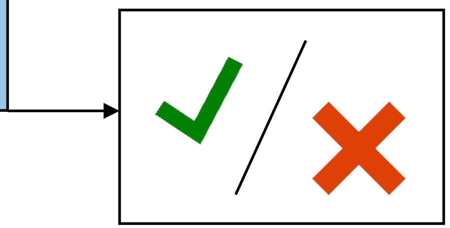
Common input





Soundness: if there is no witness, the verifier rejects

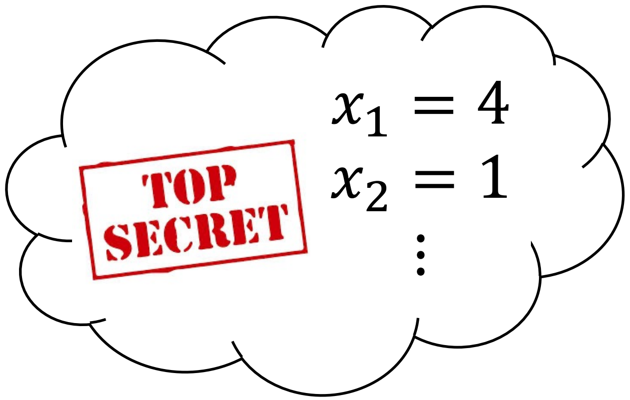
Knowledge soundness: if the prover does not know a witness, the verifier rejects



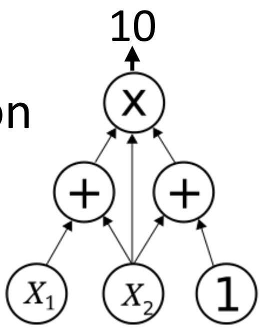
Completeness: if the witness is valid, the verifier accepts

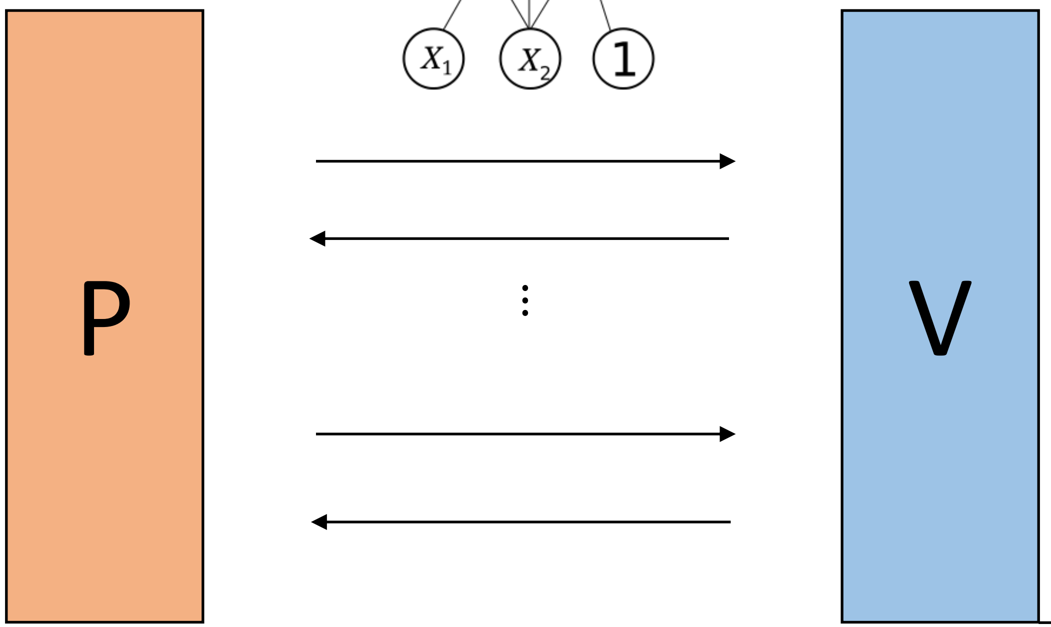
Succinct arguments

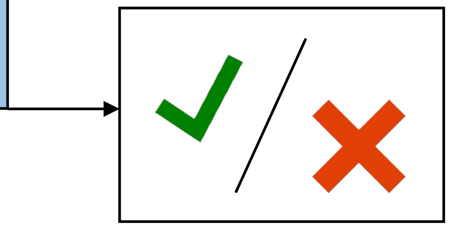
Witness



Common input







Soundness: if there is no witness, the verifier rejects

Knowledge soundness: if the prover does not know a witness, the verifier rejects

Completeness: if the witness is valid, the verifier accepts

Succinctness: the messages are much smaller than the witness

Building post-quantum succinct arguments

Building post-quantum succinct arguments

Hash-based

e.g. Aurora [BSCRSVW19]
Orion [XZS22]
Large proofs (~1MB)
Transparent

Building post-quantum succinct arguments

Pre-quantum, non-standard assumptions

e.g. [Groth16]

Tiny proofs (~1KB)
Trusted setup

Hash-based

e.g. Aurora [BSCRSVW19]
Orion [XZS22]
Large proofs (~1MB)
Transparent

Building post-quantum succinct arguments

Pre-quantum, non-standard assumptions

e.g. [Groth16]

Tiny proofs (~1KB)
Trusted setup

Pre-quantum, standard assumptions

e.g. Dory [Lee21]

Small proofs (~20KB)
Transparent

Hash-based

e.g. Aurora [BSCRSVW19]
Orion [XZS22]

Large proofs (~1MB)
Transparent

Building post-quantum succinct arguments

Pre-quantum, non-standard assumptions

e.g. [Groth16]

Tiny proofs (~1KB)
Trusted setup

Pre-quantum, standard assumptions

e.g. Dory [Lee21]

Small proofs (~20KB)
Transparent

Lattice-based, non-standard assumptions

e.g. [ACLMT22], [FLV23],[CLM23]

Large proofs (~1MB)***
Trusted setup

Hash-based

e.g. Aurora [BSCRSVW19]
Orion [XZS22]

Large proofs (~1MB)
Transparent

Building post-quantum succinct arguments

Pre-quantum, non-standard assumptions e.g. [Groth16] Tiny proofs (~1KB) Trusted setup	Pre-quantum, standard assumptions e.g. Dory [Lee21] Small proofs (~20KB) Transparent	
Lattice-based, non-standard assumptions e.g. [ACLMT22], [FLV23],[CLM23] Large proofs (~1MB)*** Trusted setup	Lattice-based, standard assumptions ? Small proofs Transparent	Hash-based e.g. Aurora [BSCRSVW19] Orion [XZS22] Large proofs (~1MB) Transparent

Building post-quantum succinct arguments

Pre-quantum, non-standard assumptions

e.g. [Groth16]

Tiny proofs (~1KB)
Trusted setup

Pre-quantum, standard assumptions

e.g. Dory [Lee21]

Small proofs (~20KB)
Transparent

Without succinct verification

e.g. Labrador [BS23]

Quite small proofs (~50KB)
Transparent

Lattice-based, non-standard assumptions

e.g. [ACLMT22], [FLV23],[CLM23]
Large proofs (~1MB)***
Trusted setup

Lattice-based, standard assumptions

?

Small proofs
Transparent

Hash-based

e.g. Aurora [BSCRSVW19]
Orion [XZS22]
Large proofs (~1MB)
Transparent

Building post-quantum succinct arguments

Pre-quantum, non-standard assumptions e.g. [Groth16] Tiny proofs (~1KB) Trusted setup	Pre-quantum, standard assumptions e.g. Dory [Lee21] Small proofs (~20KB) Transparent	Without succinct verification e.g. Labrador [BS23] Quite small proofs (~50KB) Transparent
Lattice-based, non-standard assumptions e.g. [ACLMT22], [FLV23],[CLM23] Large proofs (~1MB)*** Trusted setup	Lattice-based, standard assumptions ? Small proofs Transparent	Hash-based e.g. Aurora [BSCRSVW19] Orion [XZS22] Large proofs (~1MB) Transparent

Homomorphic cryptography

Question: can we construct **transparent**, succinct arguments from **standard** lattice assumptions?

The sumcheck protocol [LFKN92]

The sumcheck protocol [LFKN92]

Given a polynomial $p(X_1, \dots, X_\ell)$ over a field $\mathbb{F}$ and a value $u \in \mathbb{F}$,
prove that $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \dots, \omega_\ell) = u$

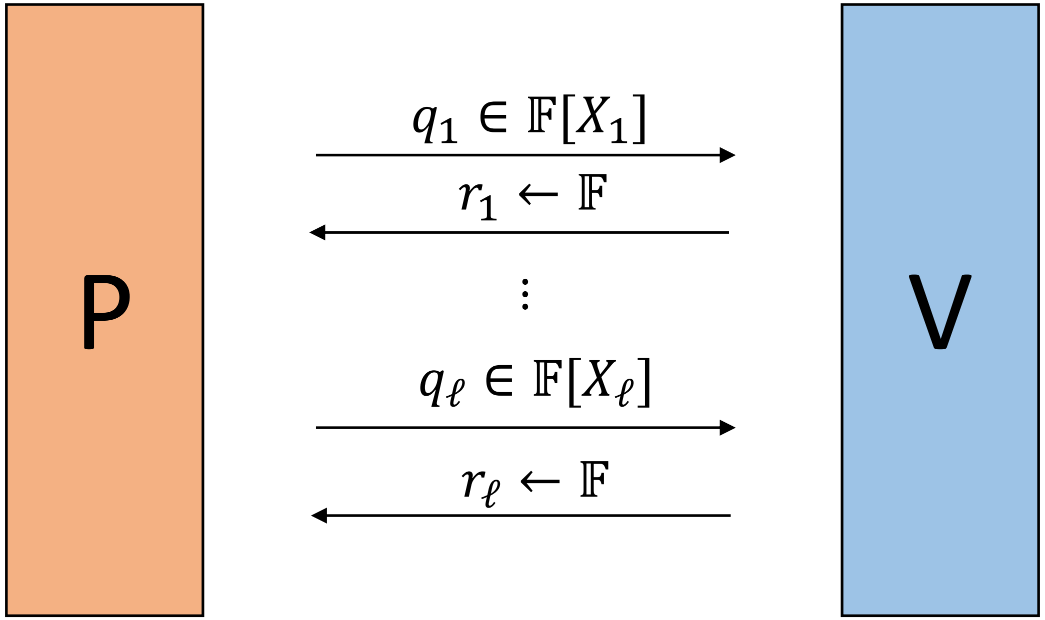
The sumcheck protocol [LFKN92]

Given a polynomial $p(X_1, \dots, X_\ell)$ over a field $\mathbb{F}$ and a value $u \in \mathbb{F}$,
prove that $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \dots, \omega_\ell) = u$

The sumcheck protocol [LFKN92]

Given a polynomial $p(X_1, \ldots, X_\ell)$ over a field $\mathbb{F}$ and a value $u \in \mathbb{F}$,
prove that $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \ldots, \omega_\ell) = u$

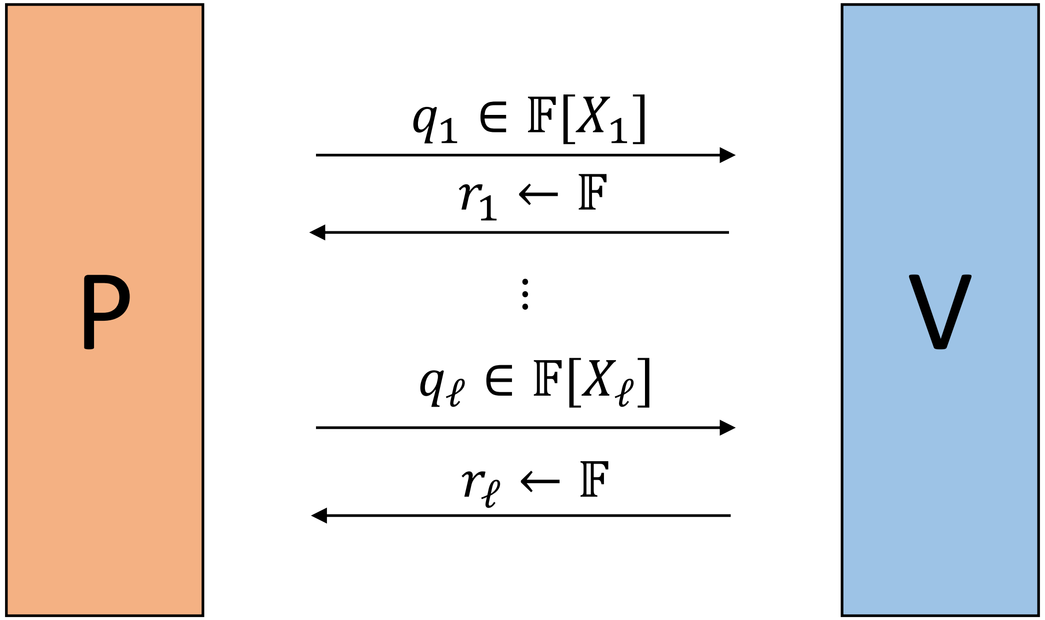
Computes polynomials
$q_i(X_i) = \sum_{\underline{\omega} \in H^{\ell-i}} p(r_1, \ldots, r_{i-1}, X_i, \omega_{i+1}, \ldots, \omega_\ell)$



The sumcheck protocol [LFKN92]

Given a polynomial $p(X_1, \dots, X_\ell)$ over a field $\mathbb{F}$ and a value $u \in \mathbb{F}$,
prove that $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \dots, \omega_\ell) = u$

Computes polynomials
$q_i(X_i) = \sum_{\underline{\omega} \in H^{\ell-i}} p(r_1, .., r_{i-1}, X_i, \omega_{i+1}, .., \omega_\ell)$

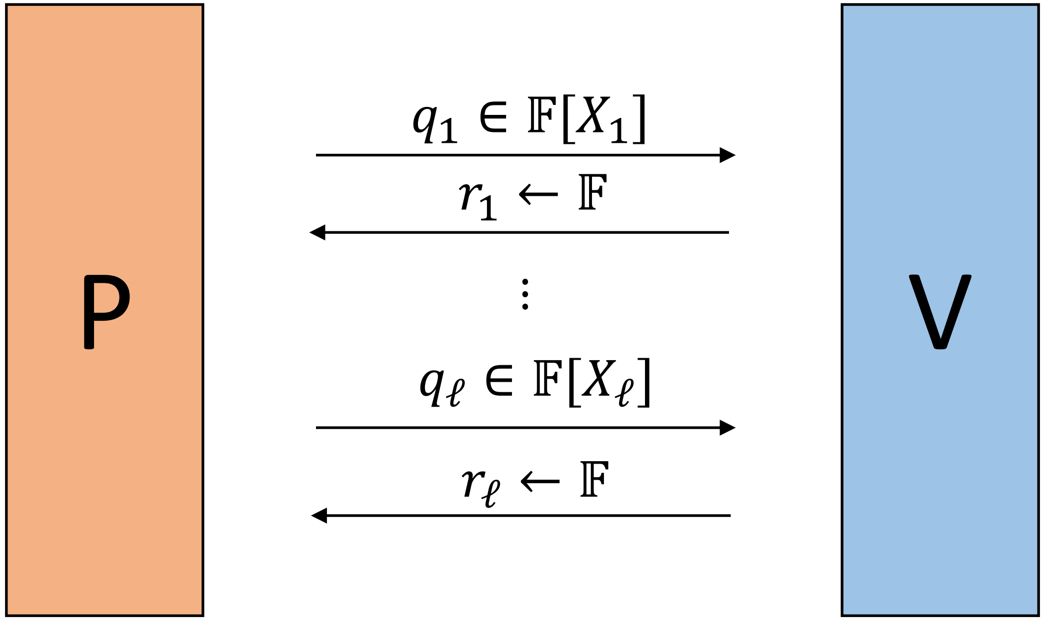


Checks that
$\sum_{\omega_1 \in H} q_1(\omega_1) = u$
$\sum_{\omega_2 \in H} q_2(\omega_2) = q_1(r_1)$
$\vdots$
$\sum_{\omega_\ell \in H} q_\ell(\omega_\ell) = q_{\ell-1}(r_{\ell-1})$

The sumcheck protocol [LFKN92]

Given a polynomial $p(X_1, \dots, X_\ell)$ over a field $\mathbb{F}$ and a value $u \in \mathbb{F}$,
prove that $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \dots, \omega_\ell) = u$

Computes polynomials
$q_i(X_i) = \sum_{\underline{\omega} \in H^{\ell-i}} p(r_1, .., r_{i-1}, X_i, \omega_{i+1}, .., \omega_\ell)$



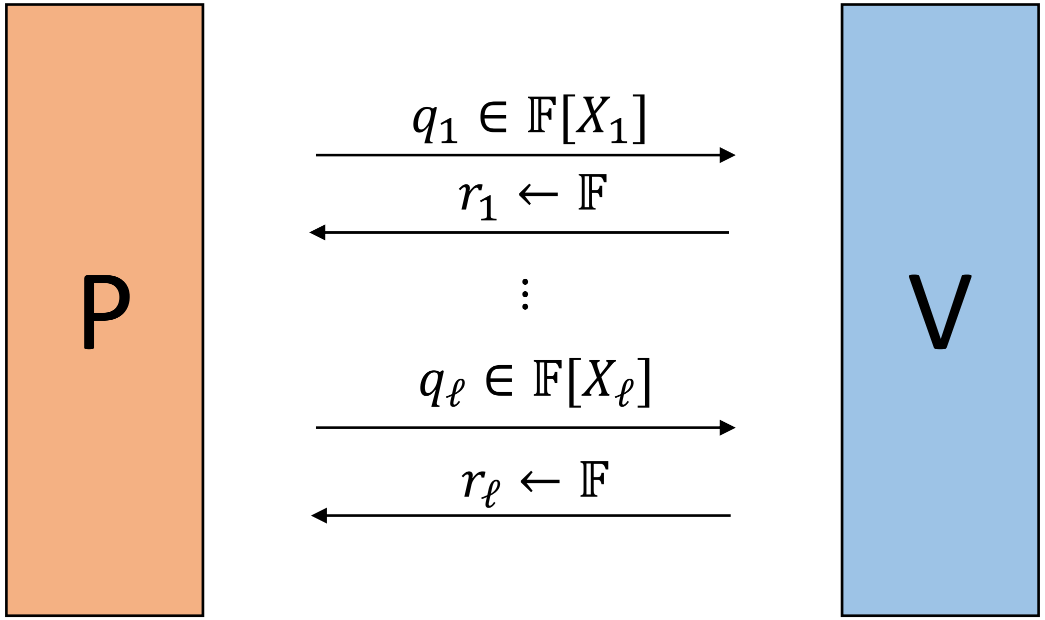
Checks that
$\sum_{\omega_1 \in H} q_1(\omega_1) = u$
$\sum_{\omega_2 \in H} q_2(\omega_2) = q_1(r_1)$
⋮
$\sum_{\omega_\ell \in H} q_\ell(\omega_\ell) = q_{\ell-1}(r_{\ell-1})$

Evaluates p to check that
$p(r_1, \dots, r_\ell) = q_\ell(r_\ell)$

The sumcheck protocol [LFKN92]

Given a polynomial $p(X_1, \dots, X_\ell)$ over a field $\mathbb{F}$ and a value $u \in \mathbb{F}$,
prove that $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \dots, \omega_\ell) = u$

Computes polynomials
$q_i(X_i) = \sum_{\underline{\omega} \in H^{\ell-i}} p(r_1, \dots, r_{i-1}, X_i, \omega_{i+1}, \dots, \omega_\ell)$



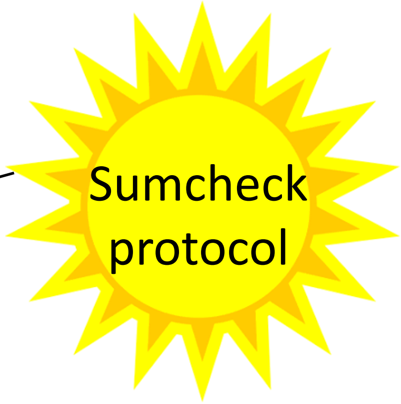
Checks that
$\sum_{\omega_1 \in H} q_1(\omega_1) = u$
$\sum_{\omega_2 \in H} q_2(\omega_2) = q_1(r_1)$
$\vdots$
$\sum_{\omega_\ell \in H} q_\ell(\omega_\ell) = q_{\ell-1}(r_{\ell-1})$

Evaluates p to check that
$p(r_1, \dots, r_\ell) = q_\ell(r_\ell)$

Soundness: If $\sum_{\underline{\omega} \in H^\ell} p(\omega_1, \dots, \omega_\ell) \neq u$ then V accepts with probability at most $\frac{\ell \cdot \deg(p)}{|\mathbb{F}|}$.

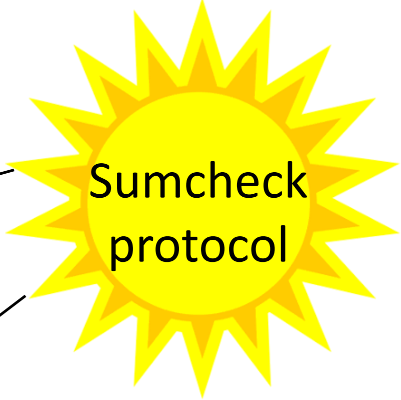
The sumcheck protocol is everywhere!

The sumcheck protocol is everywhere!



Probabilistic proofs
[BFL91,BFLS91,GKR08]

The sumcheck protocol is everywhere!



Probabilistic proofs
[BFL91,BFLS91,GKR08]

Sumcheck-based
succinct arguments
[Thaler13]

[CMT13], [VSBW13],
[W+17], [ZGKPP17],
[WTSTW18],
[XZZPS19], [Set20]

The sumcheck protocol is everywhere!

Sumcheck protocol

Probabilistic proofs
[BFL91,BFLS91,GKR08]

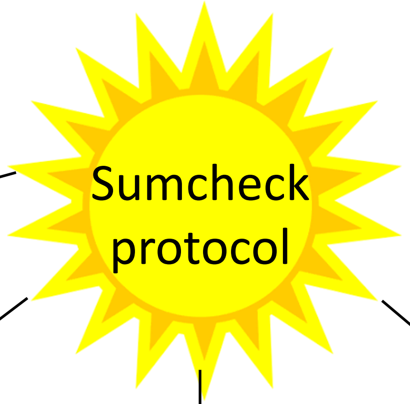
Sumcheck-based
succinct arguments
[Thaler13]

[CMT13], [VSBW13], [W+17], [ZGKPP17], [WTSTW18], [XZZPS19], [Set20]

Univariate-sumcheck-based arguments
[BCRSVS19]

[BCGGRS19], [ZXZS20], [CHMVW20], [COS20], [CFQR20], [BFHVXZ20]

The sumcheck protocol is everywhere!



Probabilistic proofs
[BFL91,BFLS91,GKR08]

Sumcheck-based
succinct arguments
[Thaler13]

[CMT13], [VSBW13],
[W+17], [ZGKPP17],
[WTSTW18],
[XZZPS19], [Set20]

Univariate-sumcheck-
based arguments
[BCRSVS19]

[BCGGRS19], [ZXZS20],
[CHMVW20], [COS20],
[CFQR20], [BFHVXZ20]

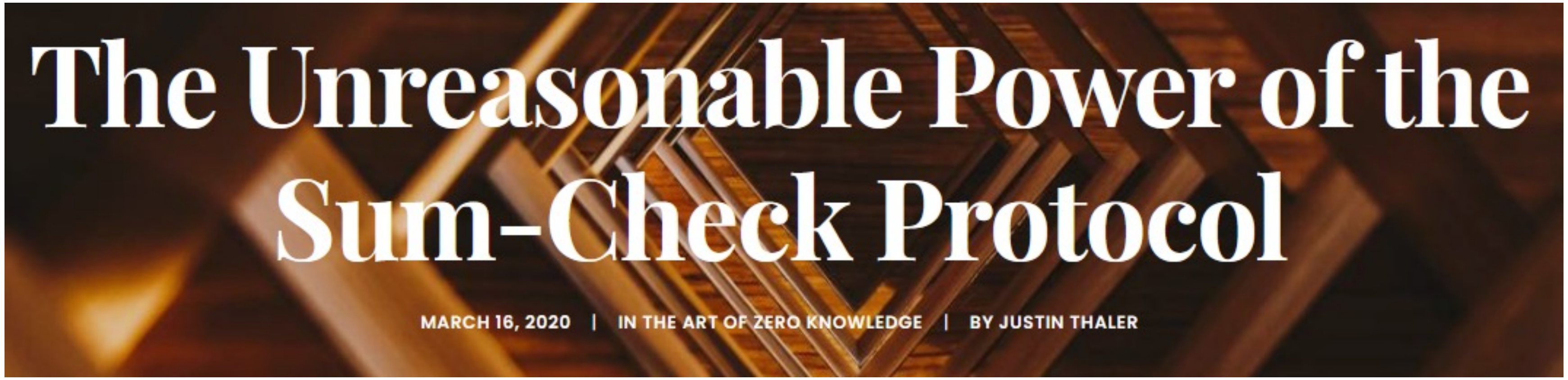
Sumchecks for
tensor codes
[Meir13]

[RR20],
[BCG20],
[BCL20]

The sumcheck protocol is everywhere!

Sumcheck protocol

Probabilistic proofs [BFL91,BFLS91,GKR08]



https://zkproof.org/2020/03/16/sum-checkprotocol/

Folding technique based on homomorphic enc: a separate body of work?

Folding technique based on homomorphic enc: a separate body of work?

Discrete-log arguments
[BBBPWM18], [PLS19],
[HKR19], [BHRRS20]

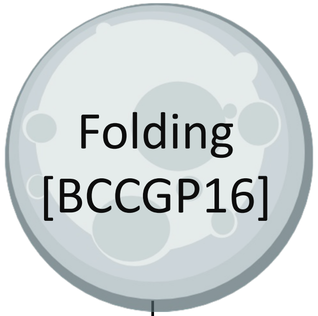
Folding technique based on homomorphic enc: a separate body of work?

Discrete-log arguments
[BBBPWM18], [PLS19],
[HKR19], [BHRRS20]

Pairing-group
arguments

[LMR19], [ZGKPP17],
[XZZPS19]

Folding technique based on homomorphic enc: a separate body of work?



Discrete-log arguments
[BBBPWM18], [PLS19],
[HKR19], [BHRRS20]

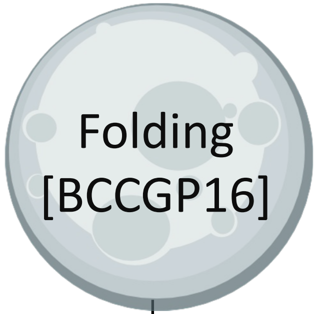
Pairing-group
arguments

[LMR19], [ZGKPP17],
[XZZPS19]

Unknown-order-group
arguments

[BFS20],
[BHRRS21]

Folding technique based on homomorphic enc: a separate body of work?



Discrete-log arguments
[BBBPWM18], [PLS19], [HKR19], [BHRRS20]

Pairing-group arguments

[LMR19], [ZGKPP17], [XZZPS19]

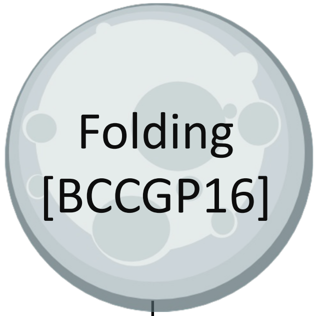
Unknown-order-group arguments

[BFS20], [BHRRS21]

Lattice arguments

[BLNS20], [ACK21], [LA20]

Folding technique based on homomorphic enc: a separate body of work?



Discrete-log arguments
[BBBPWM18], [PLS19],
[HKR19], [BHRRS20]

Pairing-group arguments

[LMR19], [ZGKPP17],
[XZZPS19]

Unknown-order-group arguments

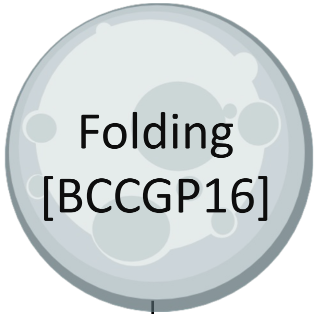
[BFS20],
[BHRRS21]

Lattice arguments

[BLNS20],
[ACK21], [LA20]

Some unifying abstractions: [BMMTV19,AC20,BDFG21]

Folding technique based on homomorphic enc: a separate body of work?



Discrete-log arguments
[BBBPWM18], [PLS19],
[HKR19], [BHRRS20]

[BBBPWM18] implemented in Rust, Haskell, Javascript, and deployed by Blockstream, and in Monero, Mimblewimble and more…


Aim, Fire: Bulletproofs Is a Crypto Privacy Breakthrough

https://www.coindesk.com/aim-fire-bulletproofs-breakthrough-privacy-blockchains

Some unifying abstractions: [BMMTV19,AC20,BDFG21]

Results

From two bodies of work…

Sumcheck protocol

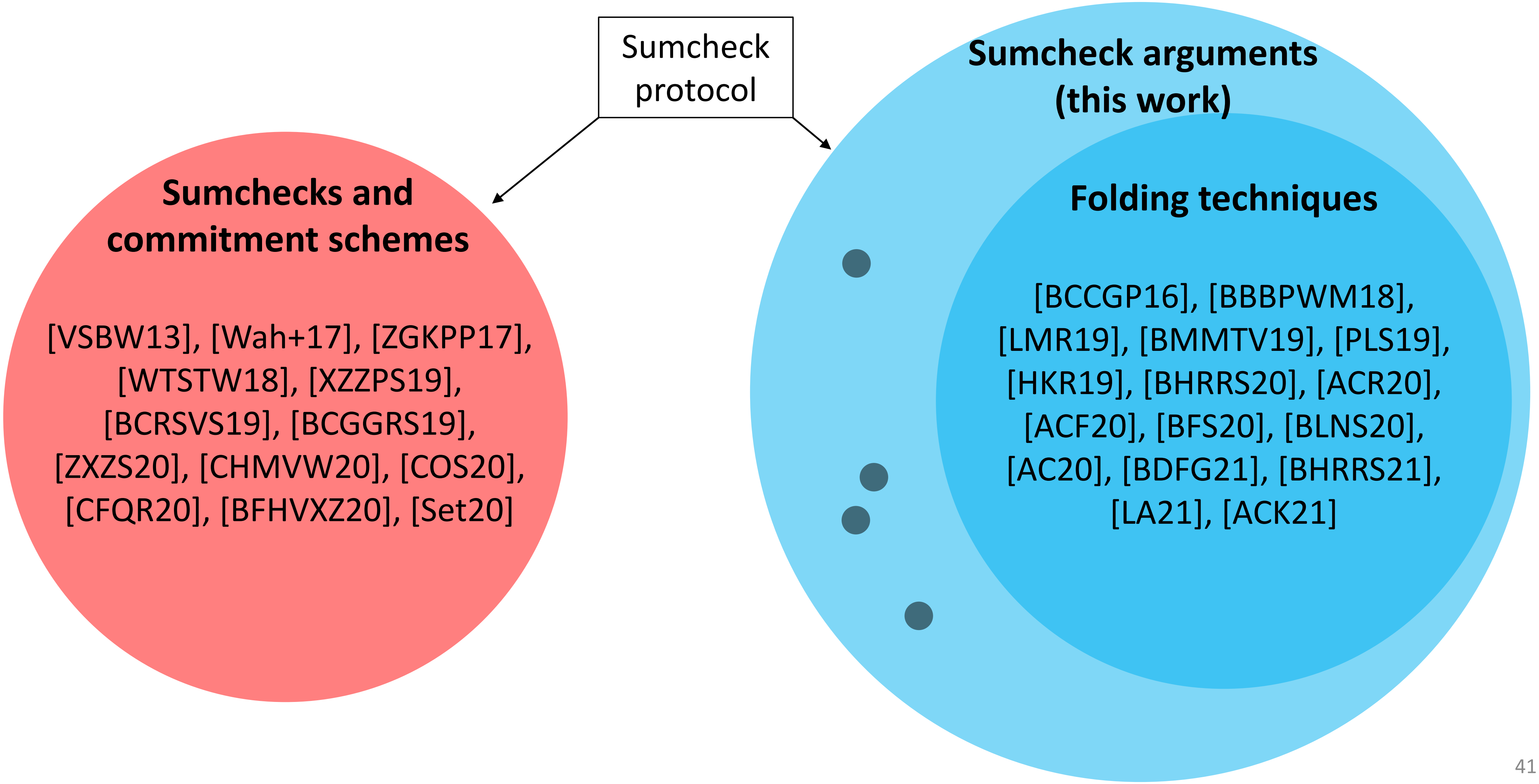
Sumchecks and commitment schemes

[VSBW13], [Wah+17], [ZGKPP17], [WTSTW18], [XZZPS19], [BCRSVS19], [BCGGRS19], [ZXZS20], [CHMVW20], [COS20], [CFQR20], [BFHVXZ20], [Set20]

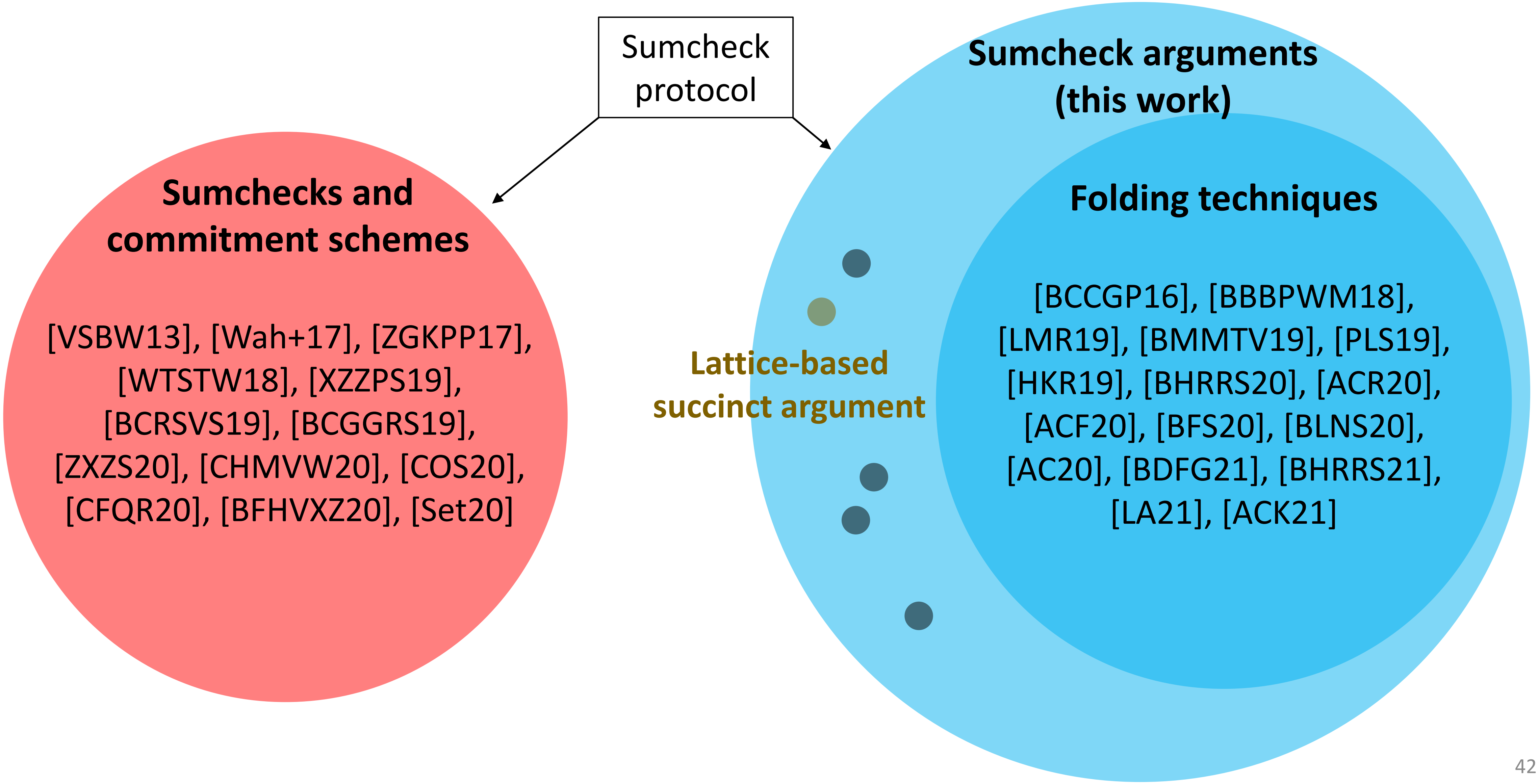
Folding techniques

[BCCGP16], [BBBPWM18], [LMR19], [BMMTV19], [PLS19], [HKR19], [BHRRS20], [ACR20], [ACF20], [BFS20], [BLNS20], [AC20], [BDFG21], [BHRRS21], [LA21], [ACK21]

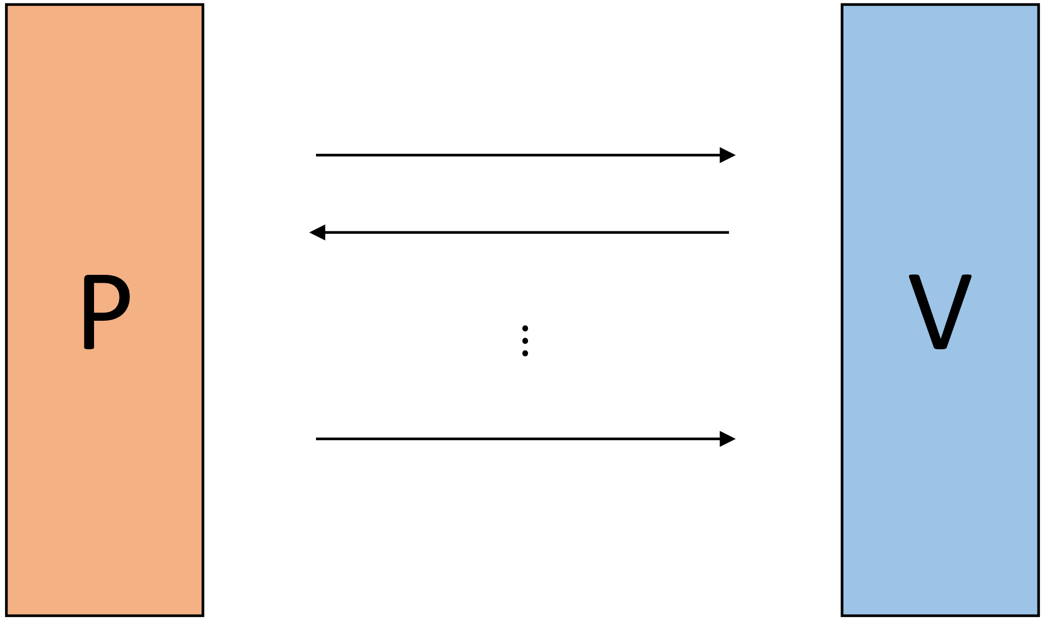
…to a unified perspective



…to a unified perspective

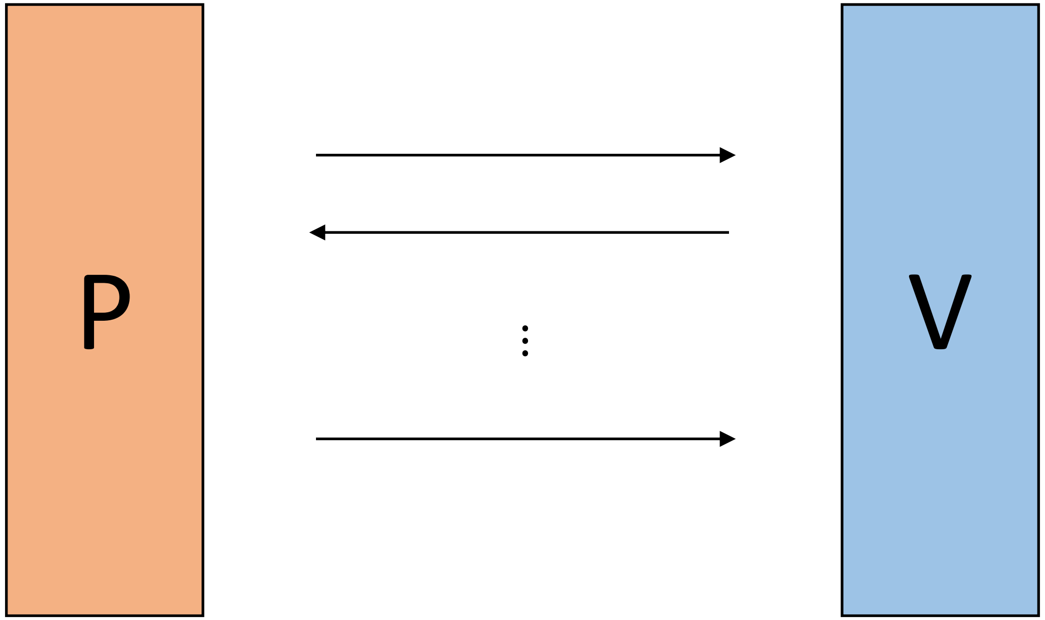


General goal:
succinct arguments for commitment openings



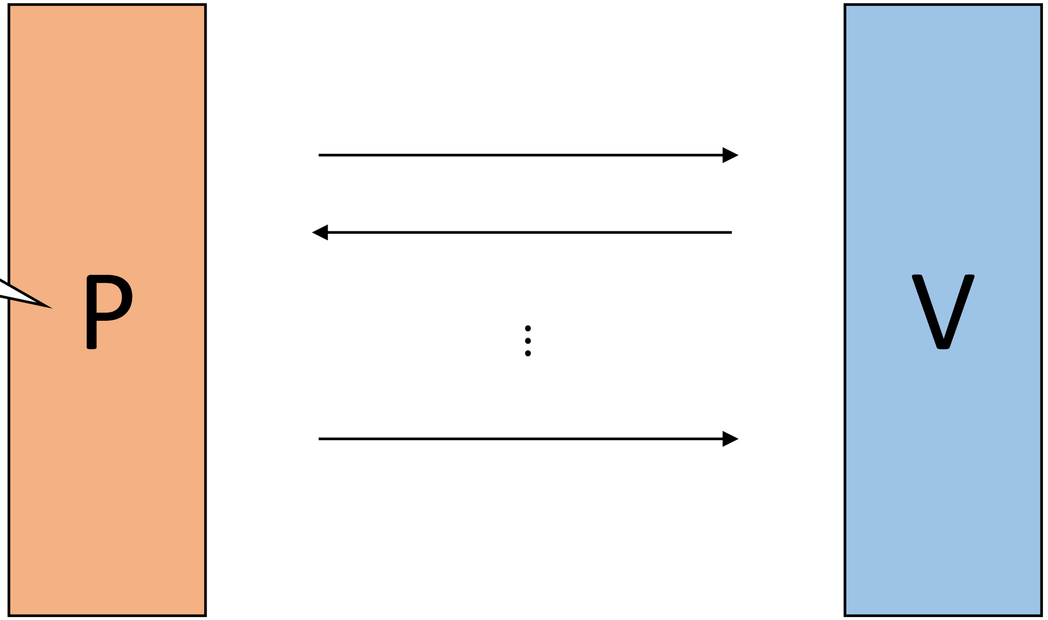
General goal: succinct arguments for commitment openings

Common input:
- commitment C
- commitment key ck



General goal:
succinct arguments for commitment openings

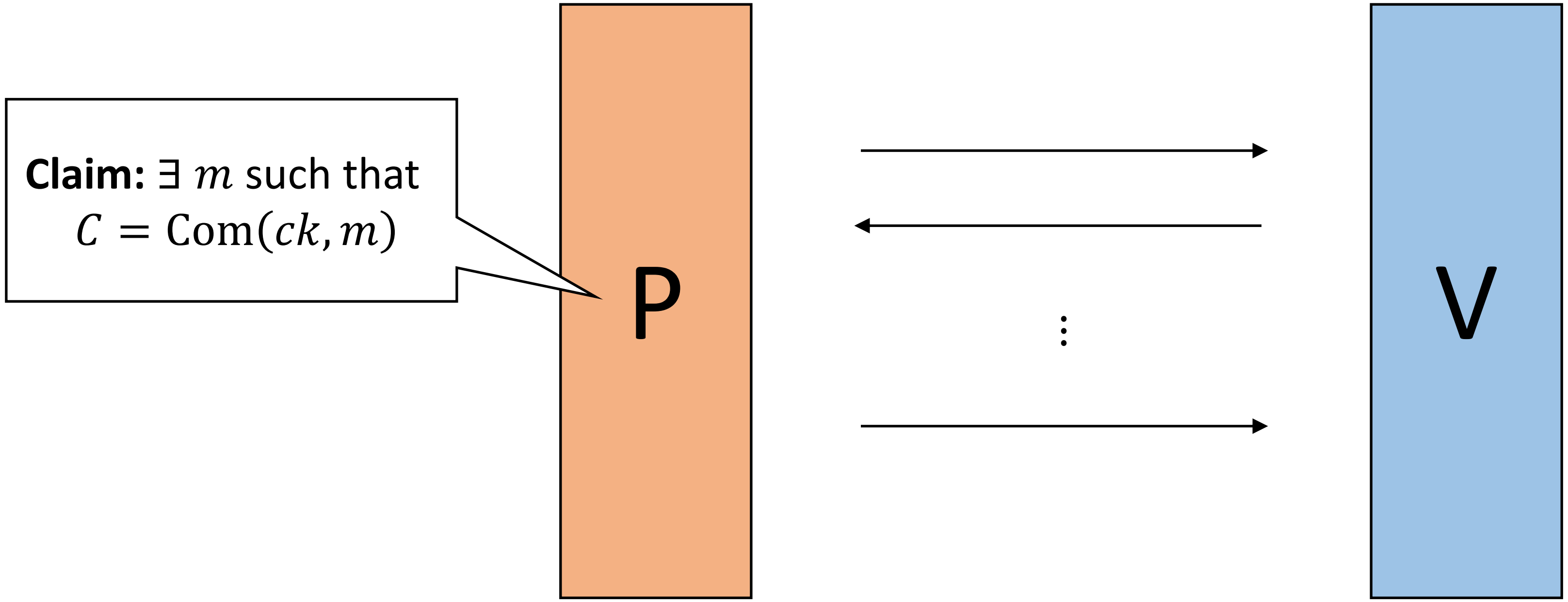
Common input:
- commitment C
- commitment key ck



Claim: $\exists\ m$ such that $C = \mathrm{Com}(ck, m)$

General goal:
succinct arguments for commitment openings

Common input:
- commitment C
- commitment key ck

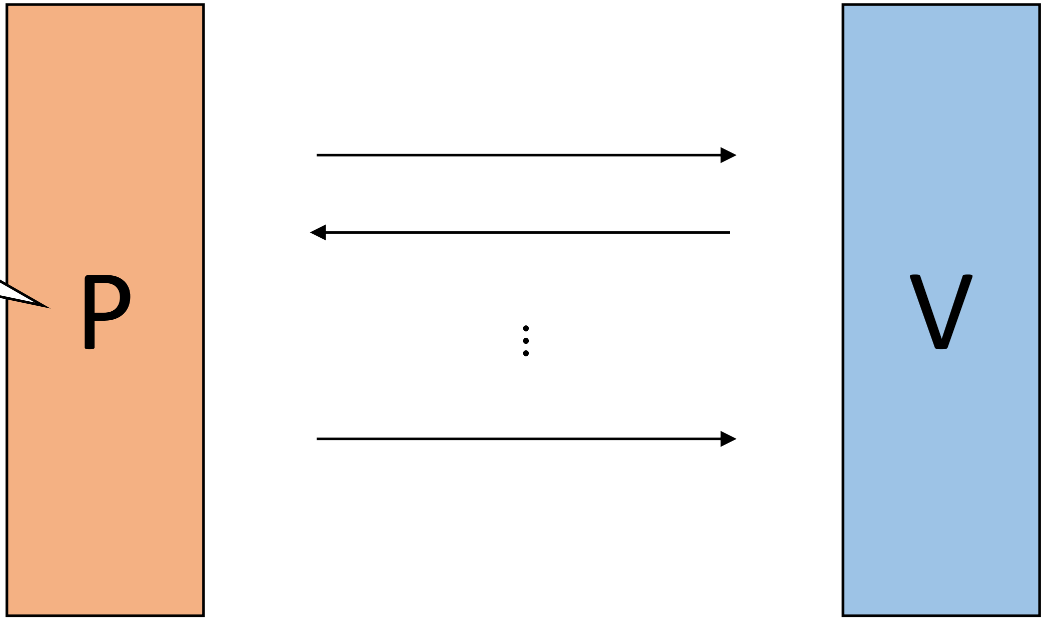


Succinctness goal:
communication $\ll |m|$

General goal:
succinct arguments for commitment openings

Common input:
- commitment C
- commitment key ck

Claim: $\exists\ m$ such that $C = \mathrm{Com}(ck, m)$



Succinctness goal:
communication $\ll |m|$

A new notion : sumcheck-friendly commitments

Definition: A commitment scheme CM is sumcheck friendly if

$$\mathrm{Com}(ck, m) = \sum_{\omega_1,\ldots,\omega_\ell \in H} f(p_m(\omega_1, \ldots, \omega_\ell), p_{ck}(\omega_1, \ldots, \omega_\ell))$$

A new notion : sumcheck-friendly commitments

Definition: A commitment scheme CM is sumcheck friendly if

$$\mathrm{Com}(ck, m) = \sum_{\omega_1,\dots,\omega_\ell \in H} f(p_m(\omega_1, \dots, \omega_\ell), p_{ck}(\omega_1, \dots, \omega_\ell))$$

evaluation points from $H \subseteq R$, R a ring

A new notion : sumcheck-friendly commitments

Definition: A commitment scheme CM is sumcheck friendly if

$$\mathrm{Com}(ck, m) = \sum_{\omega_1,\ldots,\omega_\ell \in H} f\left(p_m(\omega_1, \ldots, \omega_\ell), p_{ck}(\omega_1, \ldots, \omega_\ell)\right)$$

commitment space $\mathbb{C}$ is an R-module

evaluation points from $H \subseteq R$, R a ring

A new notion : sumcheck-friendly commitments

Definition: A commitment scheme CM is sumcheck friendly if

$$\mathrm{Com}(ck, m) = \sum_{\omega_1,\ldots,\omega_\ell \in H} f\left(p_m(\omega_1, \ldots, \omega_\ell), p_{ck}(\omega_1, \ldots, \omega_\ell)\right)$$

commitment space $\mathbb{C}$ is an R-module

evaluation points from $H \subseteq R$, R a ring

message polynomial in $\mathbb{M}[X_1, \ldots, X_\ell]$, $\mathbb{M}$ an R-module

A new notion : sumcheck-friendly commitments

Definition: A commitment scheme CM is sumcheck friendly if

$$\mathrm{Com}(ck, m) = \sum_{\omega_1,\ldots,\omega_\ell \in H} f(p_m(\omega_1, \ldots, \omega_\ell), p_{ck}(\omega_1, \ldots, \omega_\ell))$$

commitment space $\mathbb{C}$ is an R-module

evaluation points from $H \subseteq R$, R a ring

message polynomial in $\mathbb{M}[X_1, \ldots, X_\ell]$, $\mathbb{M}$ an R-module

key polynomial in $\mathbb{K}[X_1, \ldots, X_\ell]$, $\mathbb{K}$ an R-module

A new notion : sumcheck-friendly commitments

Definition: A commitment scheme CM is sumcheck friendly if

combiner function $f : \mathbb{M} \times \mathbb{K} \to \mathbb{C}$

$$\mathrm{Com}(ck, m) = \sum_{\omega_1, \dots, \omega_\ell \in H} f\left(p_m(\omega_1, \dots, \omega_\ell), p_{ck}(\omega_1, \dots, \omega_\ell)\right)$$

commitment space $\mathbb{C}$ is an R-module

evaluation points from $H \subseteq R$, R a ring

message polynomial in $\mathbb{M}[X_1, \dots, X_\ell]$, $\mathbb{M}$ an R-module

key polynomial in $\mathbb{K}[X_1, \dots, X_\ell]$, $\mathbb{K}$ an R-module

Main result: sumcheck arguments

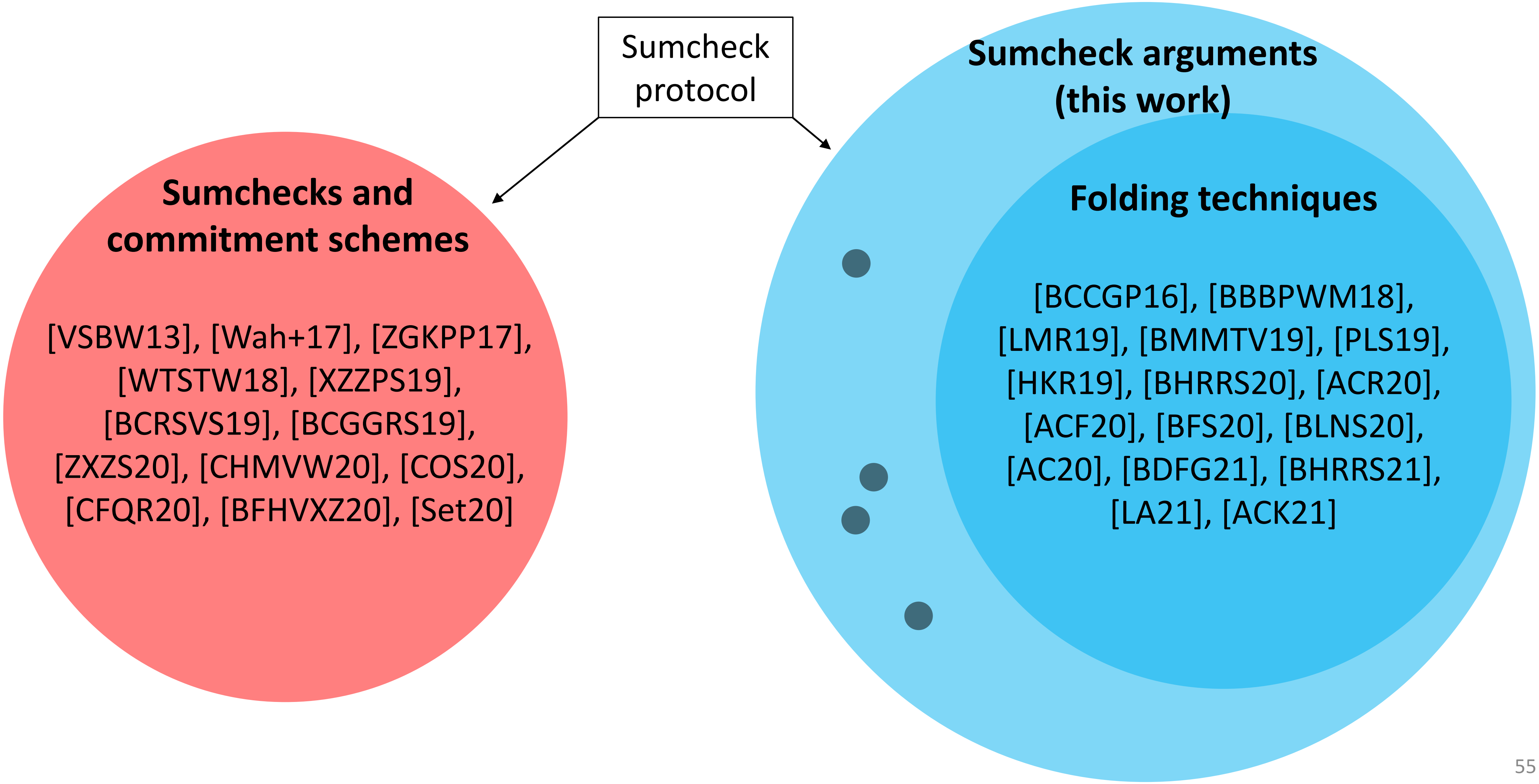
Theorem 1:

If CM is **sumcheck-friendly** and **invertible**. The sumcheck protocol applied to

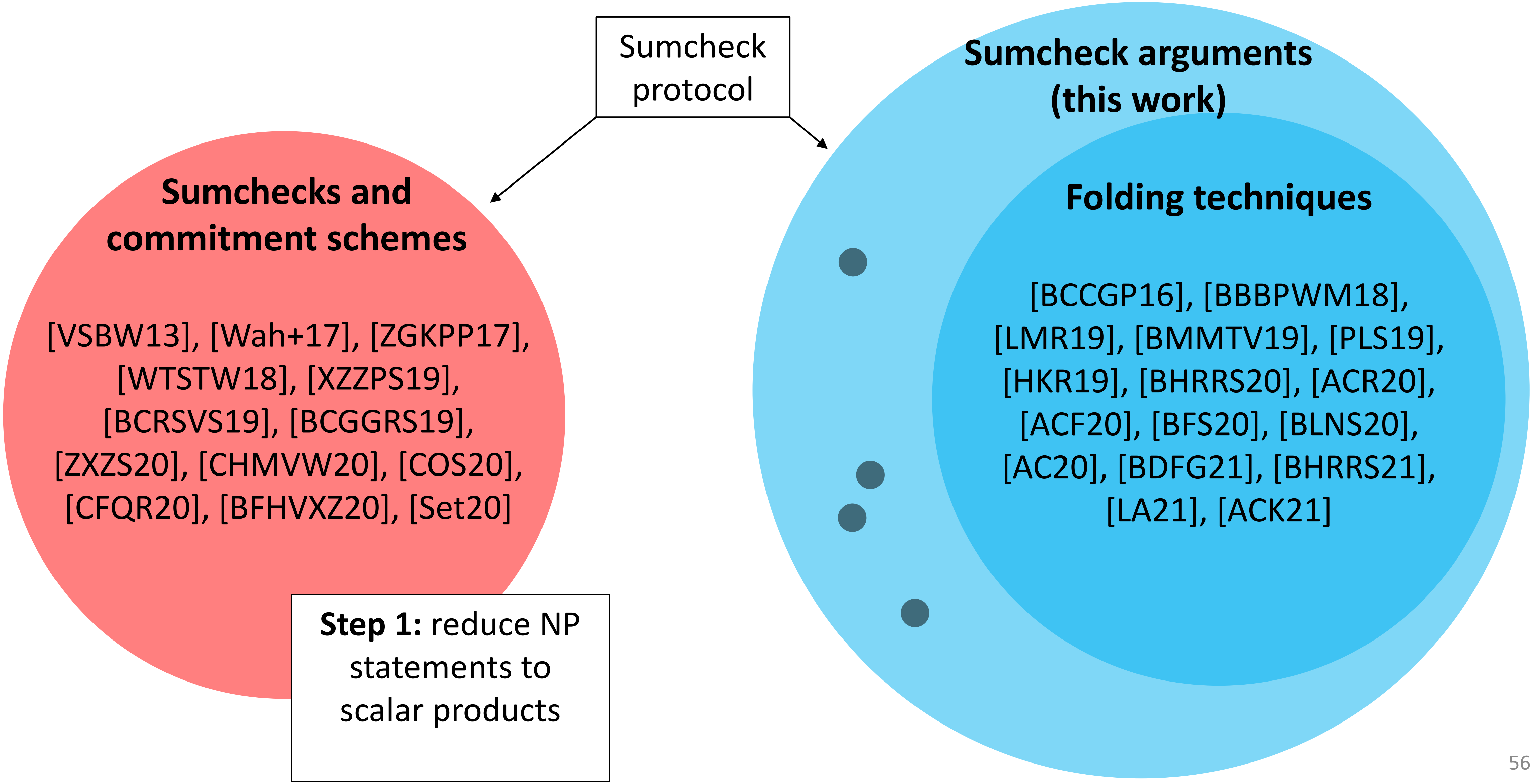
$$p(X_1, \dots, X_\ell) = f\big(p_m(X_1, \dots, X_\ell), p_{ck}(X_1, \dots, X_\ell)\big) \in \mathbb{C}[X_1, \dots, X_\ell]$$

(with one extra verifier check) is a succinct argument of knowledge with communication $\ell \cdot \deg(p)$

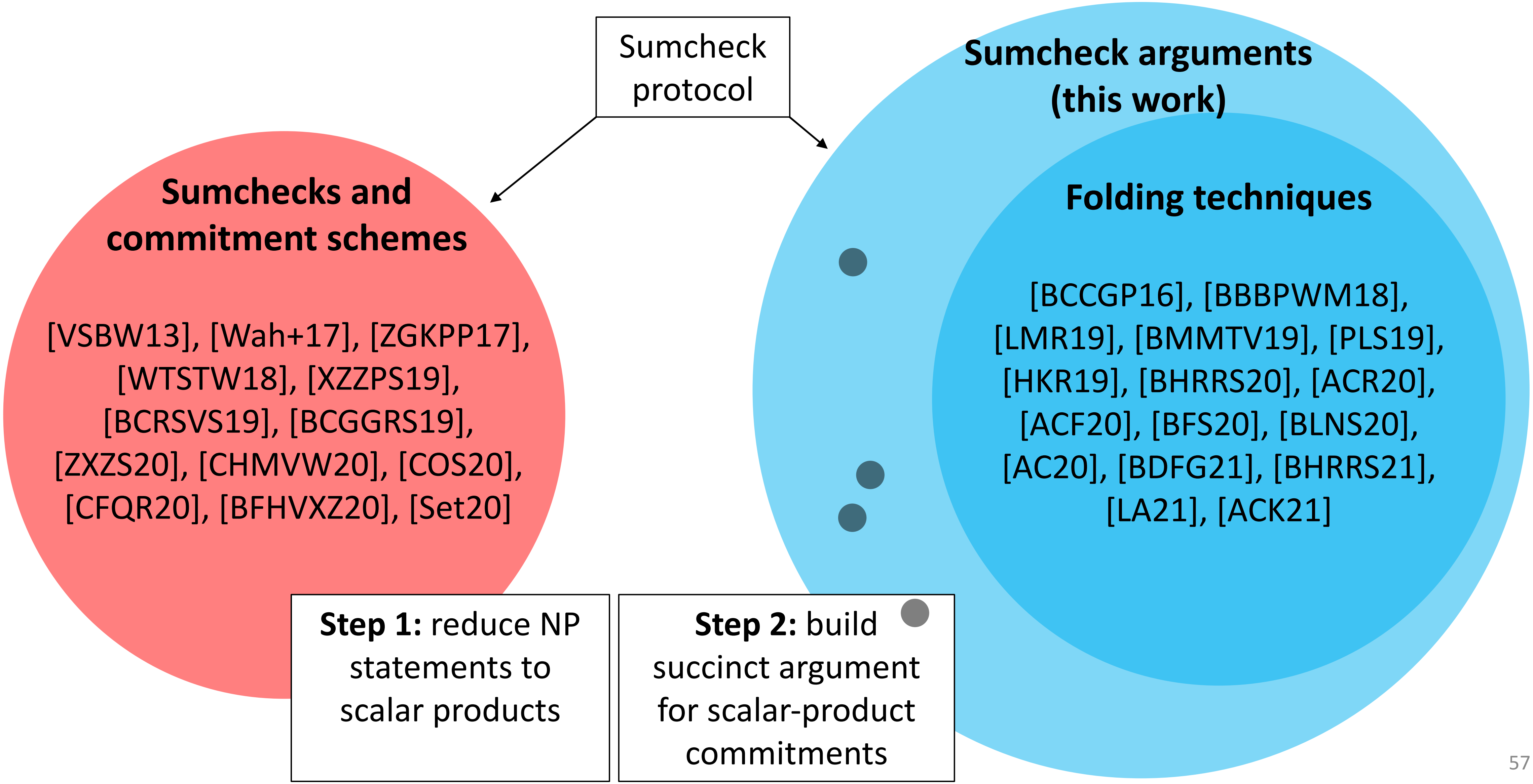
Application: succinct arguments for NP



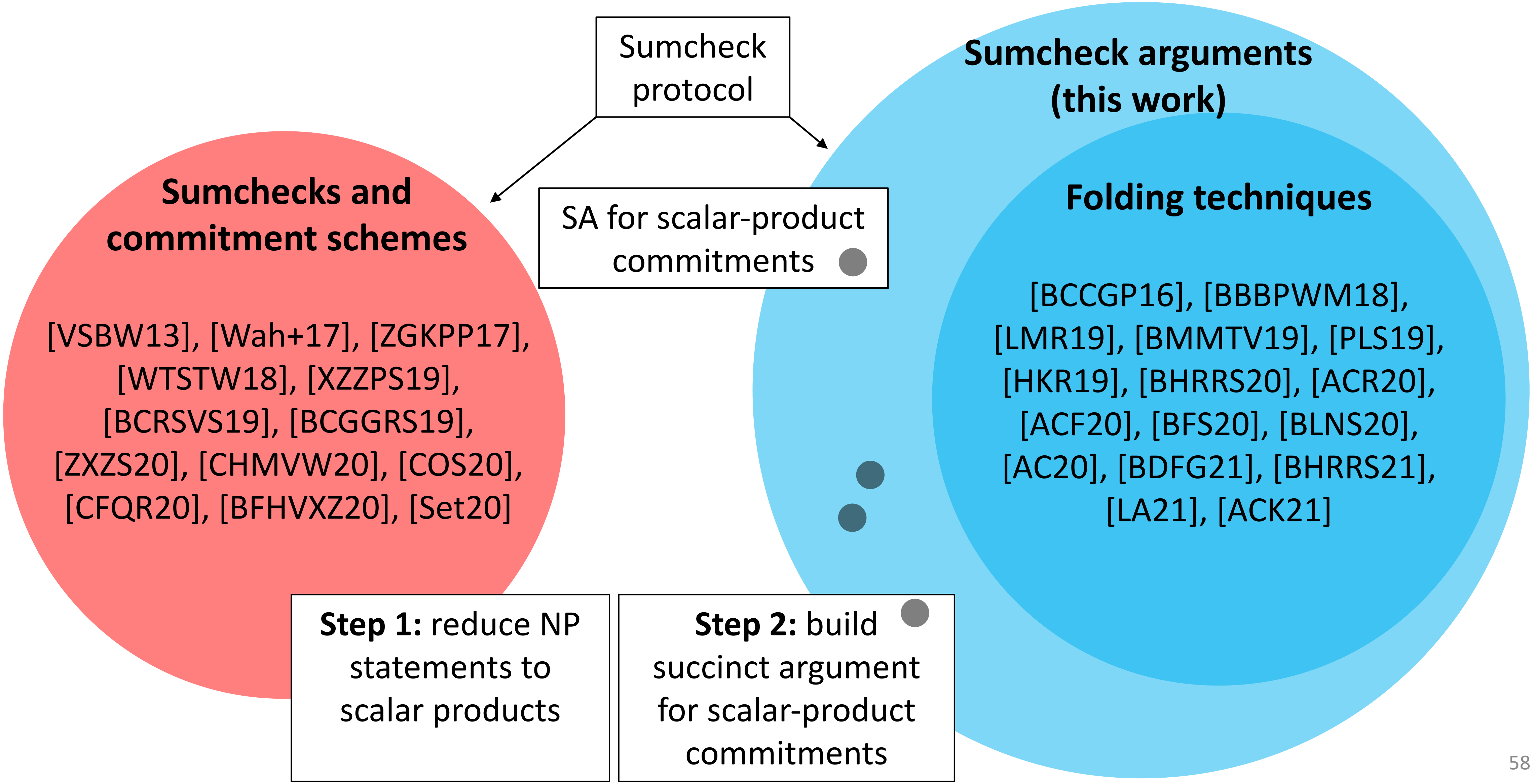
Application: succinct arguments for NP



Application: succinct arguments for NP



Application: succinct arguments for NP



Lattice-based succinct arguments for NP

[Bootle Chiesa **Sotiraki** '21]

Corollary: Assuming SIS is hard over $R_q \coloneqq \mathbb{Z}_q[X]/\langle X^d + 1\rangle$ and $p \ll q$ primes, there is a *zero-knowledge* succinct argument of knowledge for NP with

R1CS Ring	Prover time	Verifier time	Proof size
R_p	$O(n)$ ops in R_p, R_q	$O(n)$ ops in R_p, R_q	$O(\log n)$ elems of R_q

Lattice-based succinct arguments for NP

[Bootle Chiesa **Sotiraki** ‘21]

Corollary: Assuming SIS is hard over $R_q := \mathbb{Z}_q[X]/\langle X^d + 1\rangle$ and $p \ll q$ primes, there is a *zero-knowledge* succinct argument of knowledge for NP with

R1CS Ring	Prover time	Verifier time	Proof size
R_p	$O(n)$ ops in R_p, R_q	$O(n)$ ops in R_p, R_q	$O(\log n)$ elems of R_q

Concurrent work:

- [LA21] gives impossibility results and improvements for lattice POKs
- [ACK21] gives lattice-based succinct arguments for NP

Lattice-based succinct arguments for NP

[Bootle Chiesa **Sotiraki** '21]

Corollary: Assuming SIS is hard over $R_q := \mathbb{Z}_q[X]/\langle X^d + 1\rangle$ and $p \ll q$ primes, there is a *zero-knowledge* succinct argument of knowledge for NP with

R1CS Ring	Prover time	Verifier time	Proof size
R_p	$O(n)$ ops in R_p, R_q	$O(n)$ ops in R_p, R_q	$O(\log n)$ elems of R_q

Concurrent work:

- [LA21] gives impossibility results and improvements for lattice POKs
- [ACK21] gives lattice-based succinct arguments for NP

Lattice-based succinct arguments for NP

[Bootle Chiesa **Sotiraki** '23]

Corollary: Assuming SIS is hard over $R_q \coloneqq \mathbb{Z}_q[X]/\langle X^d + 1\rangle$ and $p \ll q$ primes, there is a *zero-knowledge* succinct argument of knowledge for NP with preprocessing such that

R1CS Ring	Prover time	Verifier time	Proof size
R_p	$O(n)$ ops in R_p, R_q	$\text{polylog}(n)$ops in R_p, R_q	$\text{polylog}(n)$ elems of R_q

Concurrent work:

- [LA21] gives impossibility results and improvements for lattice POKs
- [ACK21] gives lattice-based succinct arguments for NP

Techniques

Sumcheck argument for Pedersen

Common input:

- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Sumcheck argument for Pedersen

Common input:
- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Opening:
$\underline{a} \in \mathbb{F}^n$

P

V

Sumcheck argument for Pedersen

Common input:
- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Opening:
$\underline{a} \in \mathbb{F}^n$

P

sumcheck protocol for

$$\sum_{\underline{\omega} \in \{-1,1\}^{\log(n)}} p_{\underline{a}}(\underline{\omega}) p_{\underline{G}}(\underline{\omega}) = n\, C$$

V

Sumcheck argument for Pedersen

Common input:

- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Opening:
$\underline{a} \in \mathbb{F}^n$

P

sumcheck protocol for

$$\sum_{\underline{\omega} \in \{-1,1\}^{\log(n)}} p_{\underline{a}}(\underline{\omega}) p_{\underline{G}}(\underline{\omega}) = n\, C$$

V

$p_{\underline{a}}(\underline{r})$

Sumcheck argument for Pedersen

Common input:

- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Opening:
$\underline{a} \in \mathbb{F}^n$

P

sumcheck protocol for

$$\sum_{\underline{\omega} \in \{-1,1\}^{\log(n)}} p_{\underline{a}}(\underline{\omega}) p_{\underline{G}}(\underline{\omega}) = n\, C$$

$p_{\underline{a}}(\underline{r})$

V

Consistency check:
$p_{\underline{a}}(\underline{r}) p_{\underline{G}}(\underline{r}) = q_{\log(n)}(r_{\log(n)})$?

Sumcheck argument for Pedersen

Common input:
- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Opening:
$\underline{a} \in \mathbb{F}^n$

P

sumcheck protocol for

$$\sum_{\underline{\omega} \in \{-1,1\}^{\log(n)}} p_{\underline{a}}(\underline{\omega}) p_{\underline{G}}(\underline{\omega}) = n\, C$$

$p_{\underline{a}}(\underline{r})$

V

Consistency check:
$p_{\underline{a}}(\underline{r}) p_{\underline{G}}(\underline{r}) = q_{\log(n)}(r_{\log(n)})$?

Communication: $O(\log(n))$
Verifier computation: $O(n)$

Sumcheck argument for Pedersen

Common input:

- commitment $C \in \mathbb{G}$
- key $\underline{G} \in \mathbb{G}^n$

Claim: $\exists \underline{a} \in \mathbb{F}^n$ s.t. $C = \langle \underline{a}, \underline{G} \rangle$

Opening:
$\underline{a} \in \mathbb{F}^n$

P

sumcheck protocol for

$$\sum_{\underline{\omega} \in \{-1,1\}^{\log(n)}} p_{\underline{a}}(\underline{\omega}) p_{\underline{G}}(\underline{\omega}) = n\, C$$

$p_{\underline{a}}(\underline{r})$

V

Consistency check:
$p_{\underline{a}}(\underline{r}) p_{\underline{G}}(\underline{r}) = q_{\log(n)}(r_{\log(n)})$?

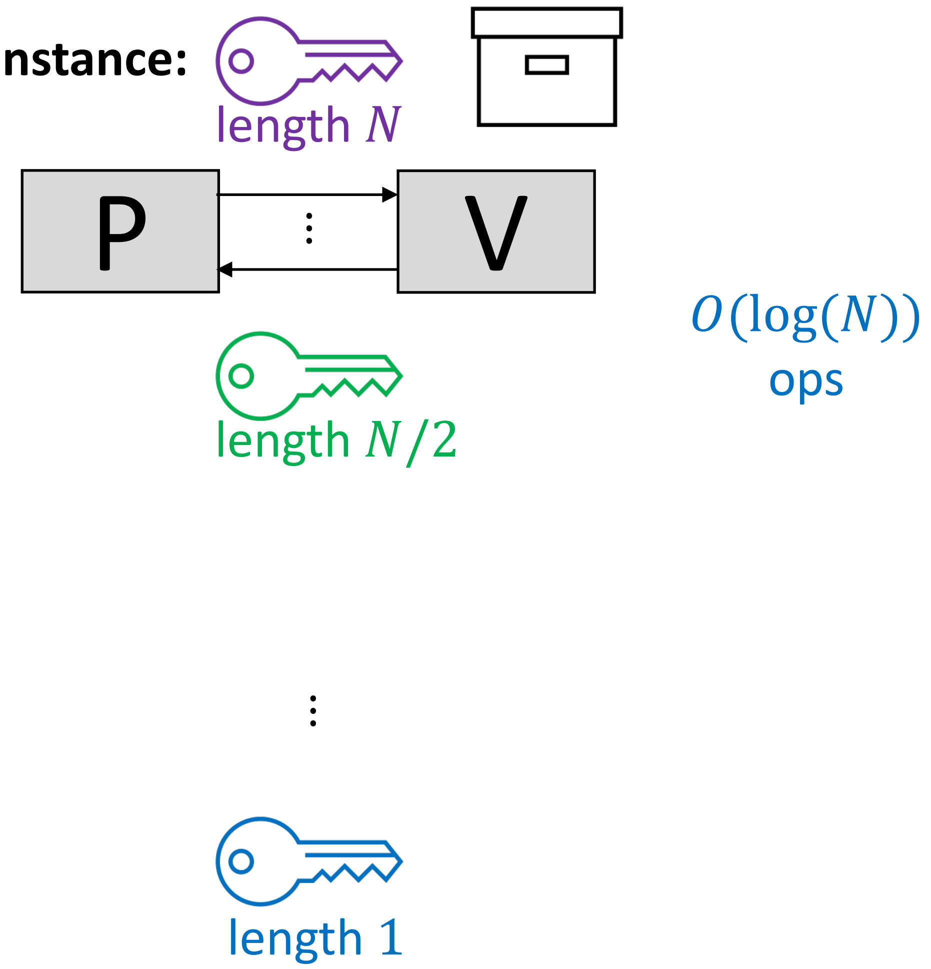
Communication: $O(\log(n))$
Verifier computation: $O(n)$

Succinct verification via delegation [Bootle Chiesa **Sotiraki** '23]

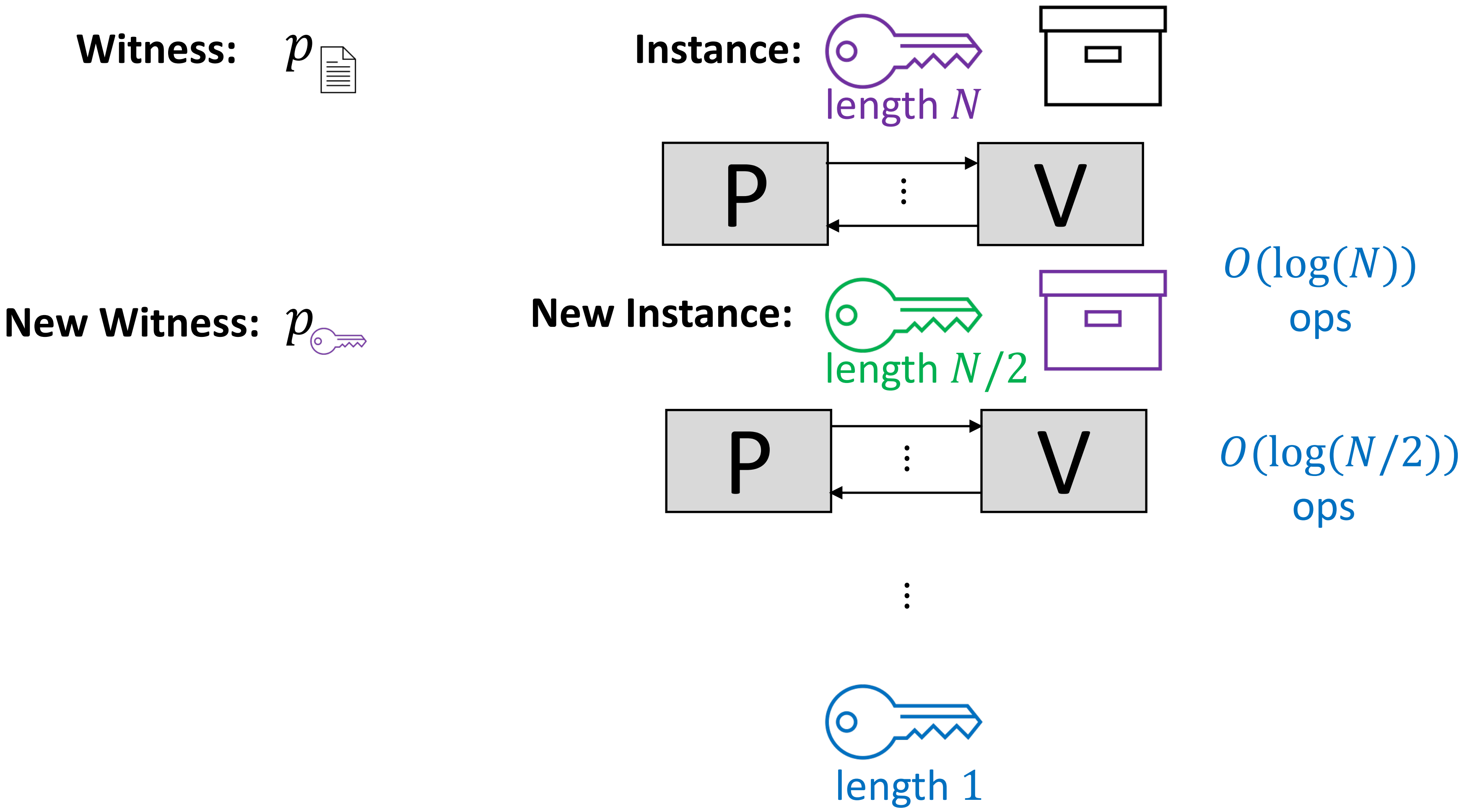
Succinct verification via delegation [Bootle Chiesa **Sotiraki** '23]

Witness: p

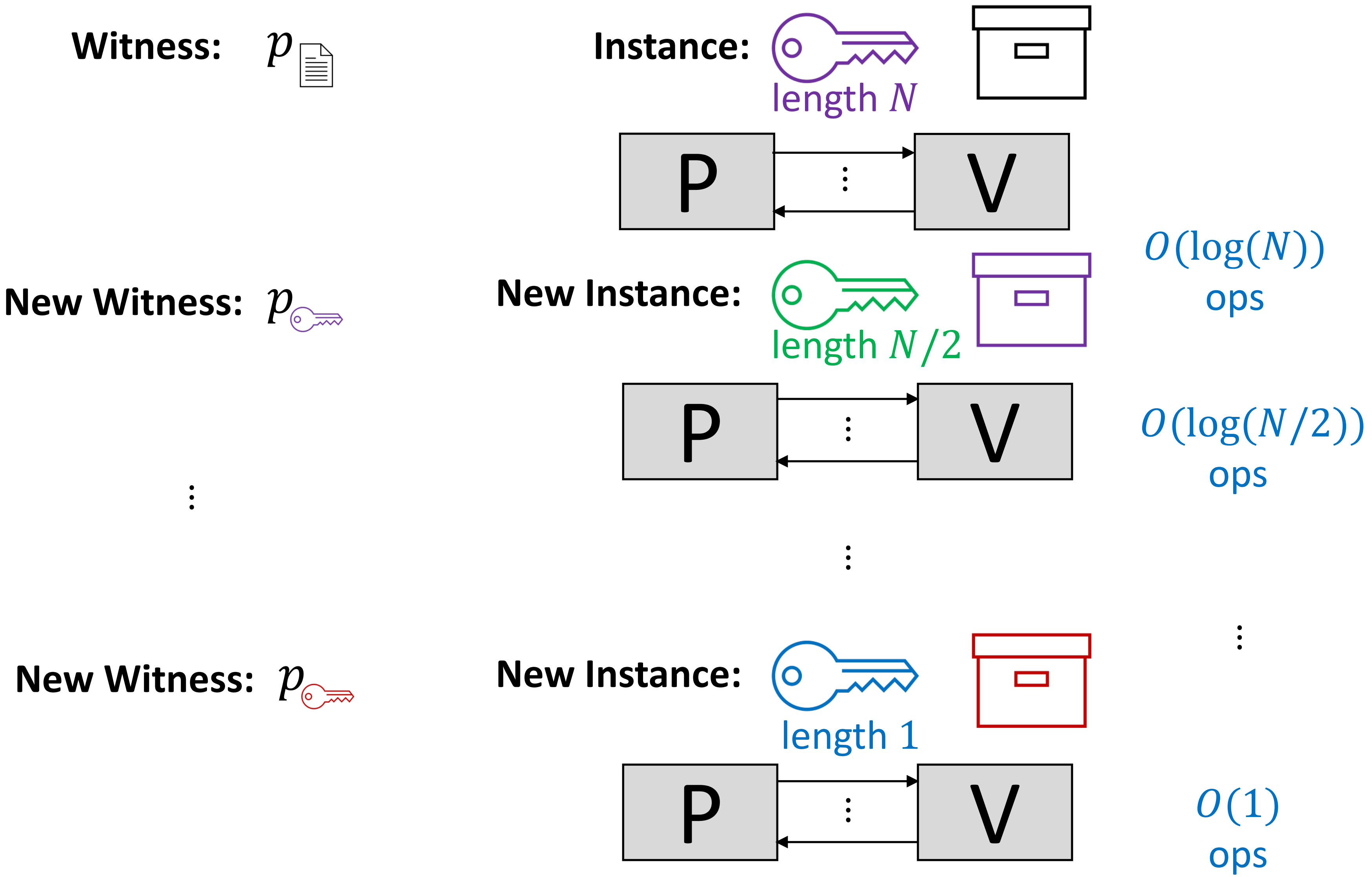
Instance:



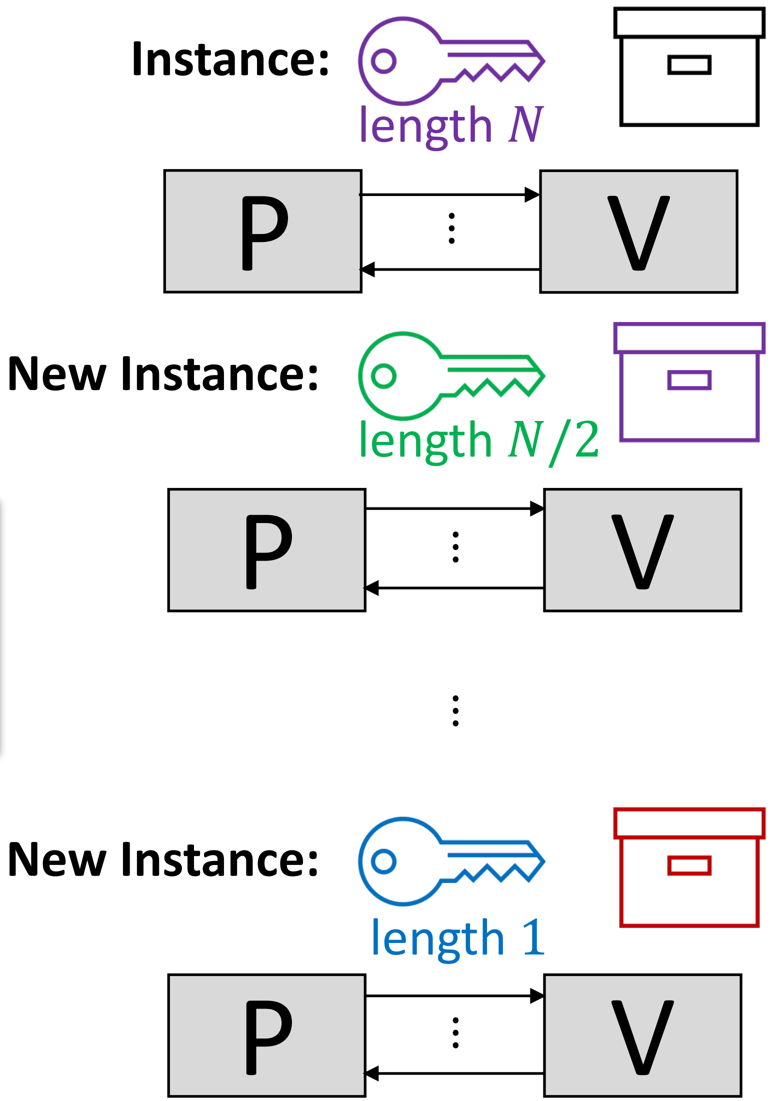
Succinct verification via delegation [Bootle Chiesa **Sotiraki** '23]



Succinct verification via delegation [Bootle Chiesa **Sotiraki** ‘23]



Succinct verification via delegation [Bootle Chiesa **Sotiraki** '23]



Generalises approach from [Lee21], [Thaler] beyond pairings

Soundness

What kind of soundness? **Knowledge soundness**

Soundness

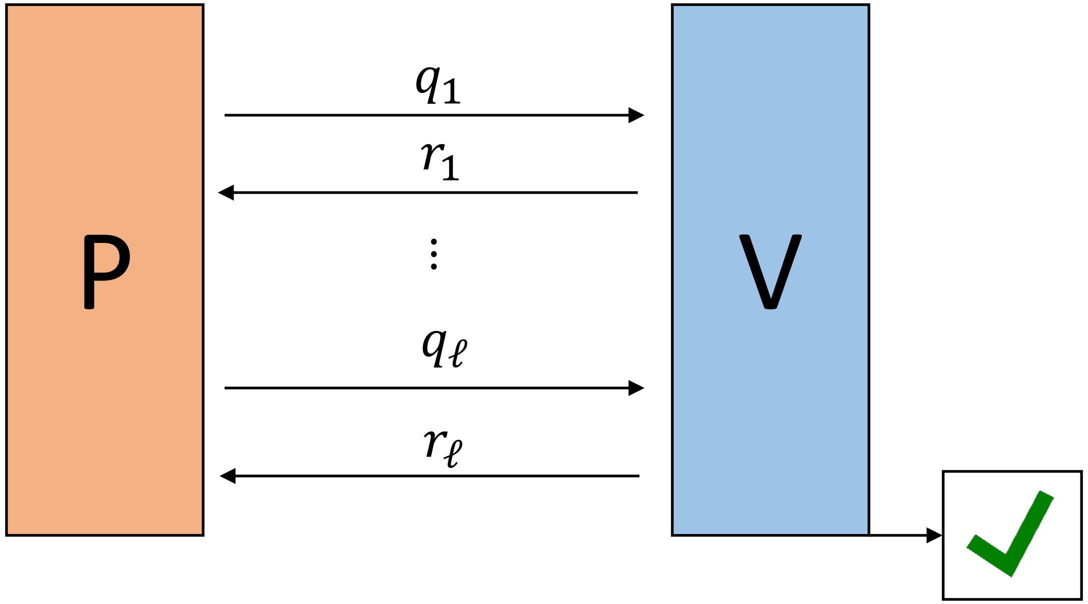
What kind of soundness? **Knowledge soundness**

There exists an extractor that given a suitable tree of *accepting transcripts* for a commitment key ck and commitment C, finds an opening m such that $C = \mathrm{Com}(ck, m)$.

Soundness

What kind of soundness? **Knowledge soundness**

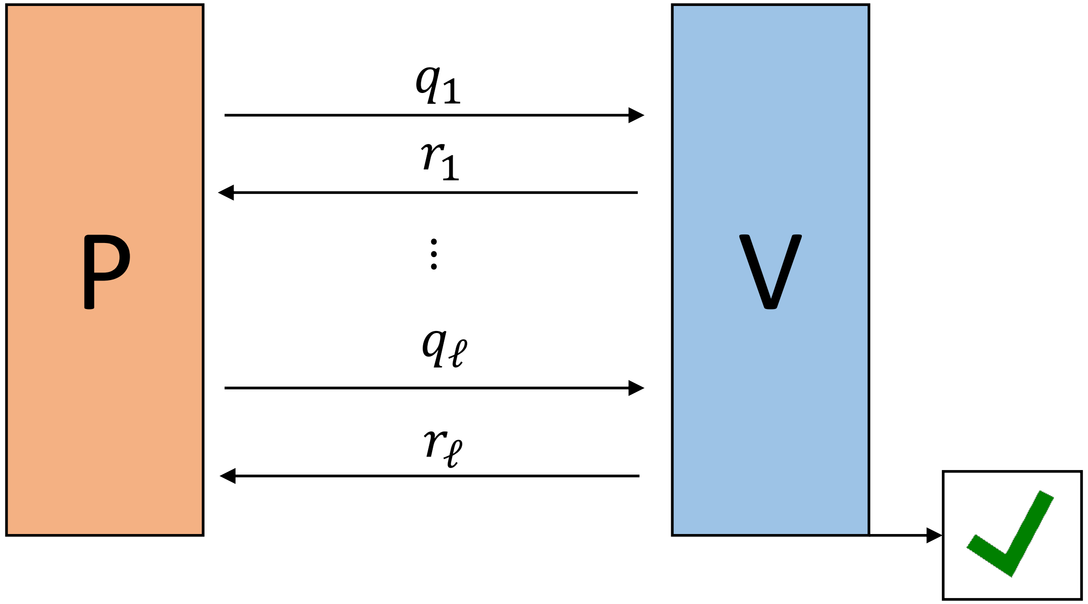
There exists an extractor that given a suitable tree of *accepting transcripts* for a commitment key ck and commitment C, finds an opening m such that $C = \mathrm{Com}(ck, m)$.

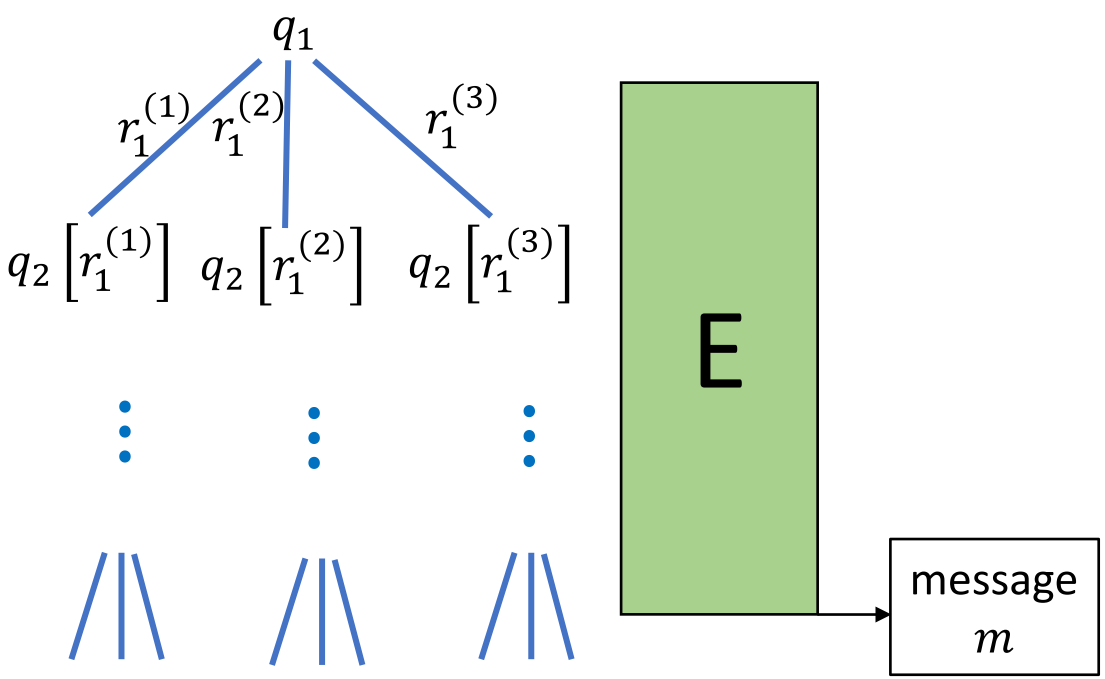


Soundness

What kind of soundness? **Knowledge soundness**

There exists an extractor that given a suitable tree of *accepting transcripts* for a commitment key ck and commitment C, finds an opening m such that $C = \mathrm{Com}(ck, m)$.





From groups to rings

From groups to rings

Everything so far extends to general $\mathbb{F}$-vector spaces, e.g., bilinear groups [BMMTV19].

From groups to rings

Everything so far extends to general $\mathbb{F}$-vector spaces, e.g., bilinear groups [BMMTV19].

Pedersen commitments for bilinear groups: $\langle \underbrace{\underline{\boldsymbol{a}}}_{\mathbb{G}_1}, \underbrace{\underline{\boldsymbol{G}_1}}_{\mathbb{G}_2} \rangle \in \mathbb{G}_{\boldsymbol{T}}$

From groups to rings

Everything so far extends to general $\mathbb{F}$-vector spaces, e.g., bilinear groups [BMMTV19].

Pedersen commitments for bilinear groups: $\langle \underline{\boldsymbol{a}}, \underline{\boldsymbol{G_1}} \rangle \in \mathbb{G}_{\boldsymbol{T}}$

$\underline{\boldsymbol{a}}$ — $\mathbb{G}_1$, $\underline{\boldsymbol{G_1}}$ — $\mathbb{G}_2$

Lattices and groups of unknown order?

From groups to rings

Everything so far extends to general $\mathbb{F}$-vector spaces, e.g., bilinear groups [BMMTV19].

Pedersen commitments for bilinear groups: $\langle \underset{\mathbb{G}_1}{\underline{\boldsymbol{a}}}, \underset{\mathbb{G}_2}{\underline{\boldsymbol{G_1}}} \rangle \in \mathbb{G}_{\boldsymbol{T}}$

Lattices and groups of unknown order?

Solution: an abstraction for mathematical structures where folding techniques can work

From groups to rings:
bilinear modules

From groups to rings:
bilinear modules

R-module M: generalization of vector space over rings

From groups to rings: bilinear modules

R-module M: generalization of vector space over rings

Assumption	Messages	Keys	Commitments	Ideal
BRA	small M_L	M_R	M_T	I
DLOG	$\mathbb{F}_p$	$\mathbb{G}$	$\mathbb{G}$	$\{0\}$
DPAIR[AFGHO10]	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	$\{0\}$
UO [BFS20]	small $\mathbb{Z}$	$\mathbb{G}$	$\mathbb{G}$	$n\mathbb{Z}$ for suitable small n
RSIS [Ajtai94]	small R_q	R_q^d	R_q^d	$n\mathbb{Z}$ for suitable small n

Takeaways

- There are lattice-based transparent, succinct arguments
- Many commitment schemes are sumcheck friendly
- We can recast many different cryptographic settings as bilinear modules

Takeaways

- There are lattice-based transparent, succinct arguments
- Many commitment schemes are sumcheck friendly
- We can recast many different cryptographic settings as bilinear modules

Thanks!