Zero Knowledge Succinct Arguments:
an Introduction

Alessandro Chiesa
UC Berkeley

Motivation

cryptography is a powerful tool
for building secure systems

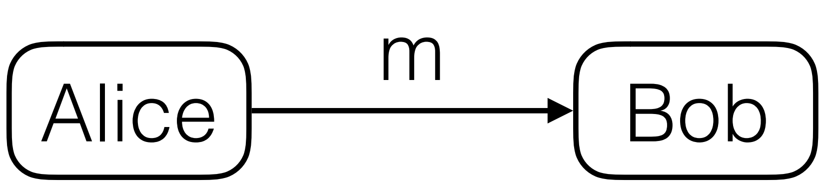
cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

confidentiality



cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

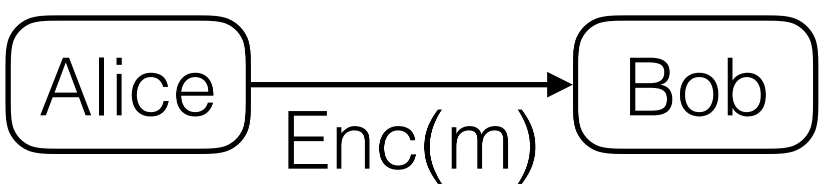
confidentiality

$$\boxed{\text{Alice}} \xrightarrow[\text{Enc(m)}]{} \boxed{\text{Bob}}$$

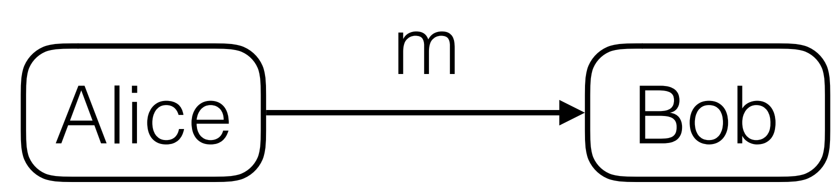
cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

confidentiality



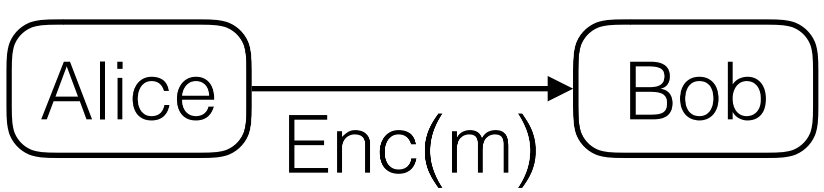
authenticity



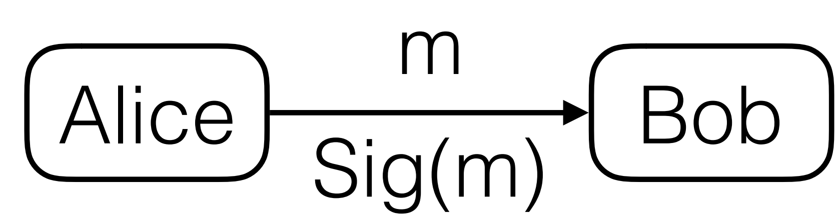
cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

confidentiality



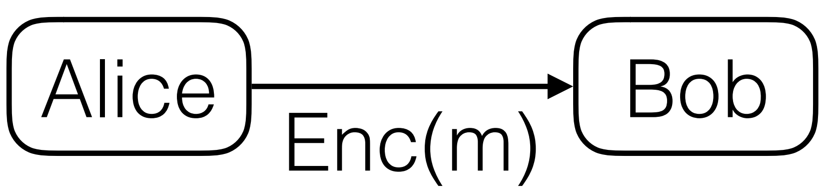
authenticity



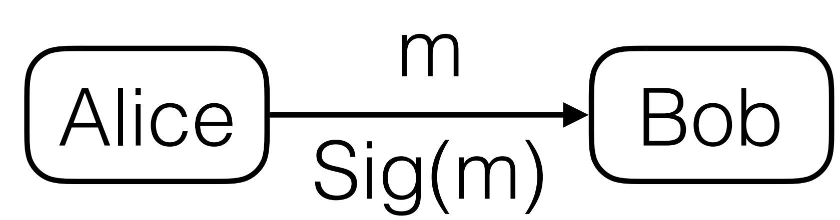
cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

confidentiality



authenticity



what about security properties for **computation**?

cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

confidentiality

$\boxed{\text{Alice}} \xrightarrow[\text{Enc(m)}]{} \boxed{\text{Bob}}$

authenticity

$\boxed{\text{Alice}} \xrightarrow[\text{Sig(m)}]{\text{m}} \boxed{\text{Bob}}$

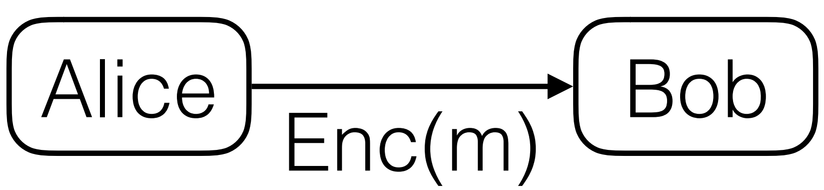
what about security properties for **computation**?

cryptographic proofs offer
privacy-preserving integrity for computation

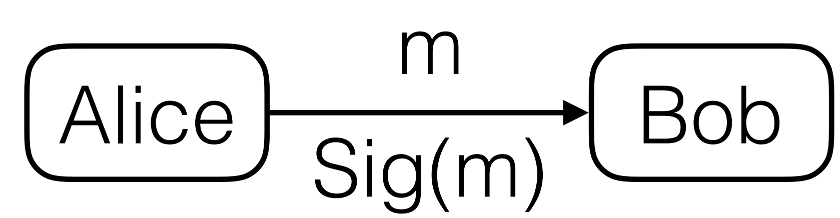
cryptography is a powerful tool
for building secure systems

much of the cryptography used today
offers security properties for **data**

confidentiality



authenticity



what about security properties for **computation**?

cryptographic proofs offer
privacy-preserving integrity for computation

one of the exciting crypto deployment frontiers today

Cryptographic Proofs

Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

Cryptographic Proofs

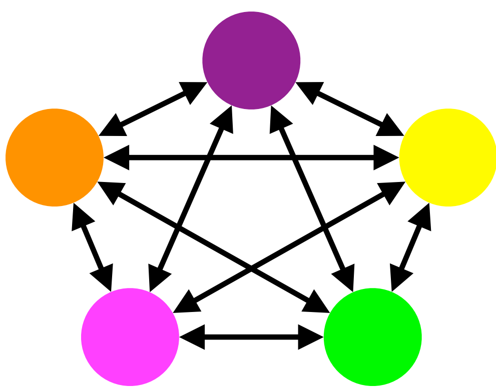
a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y=F(x_1,\ldots,x_n)$
via a multi-party protocol

Cryptographic Proofs

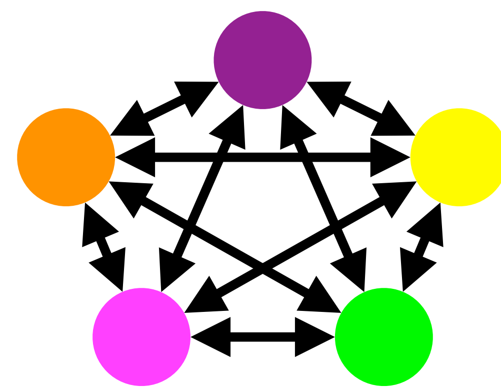
a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $\boldsymbol{y}=\boldsymbol{F}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$
via a multi-party protocol



passive security

[GMW87]

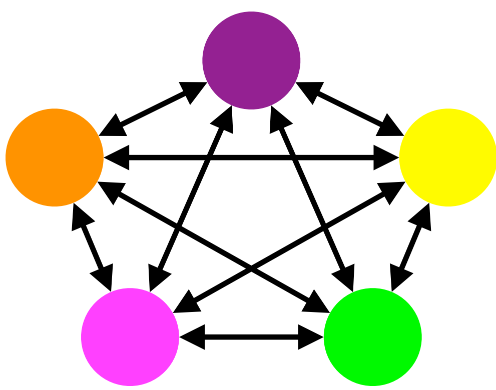


active security

Cryptographic Proofs

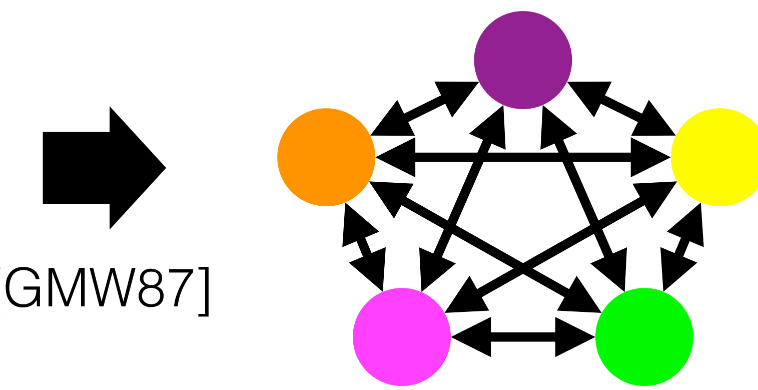
a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y=F(x_1,...,x_n)$
via a multi-party protocol



passive security

[GMW87]



active security

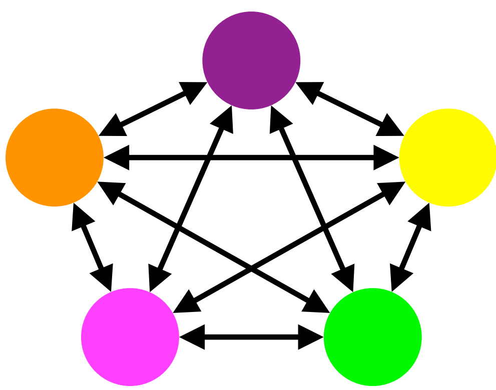
Key properties

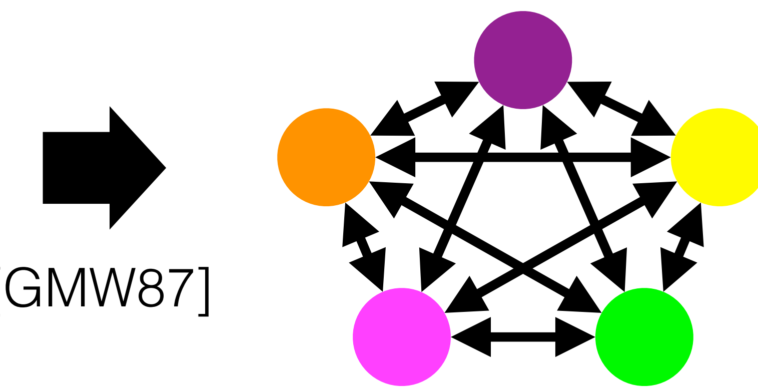
- zero knowledge
- proof of knowledge

Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$
via a multi-party protocol


passive security



active security

Key properties

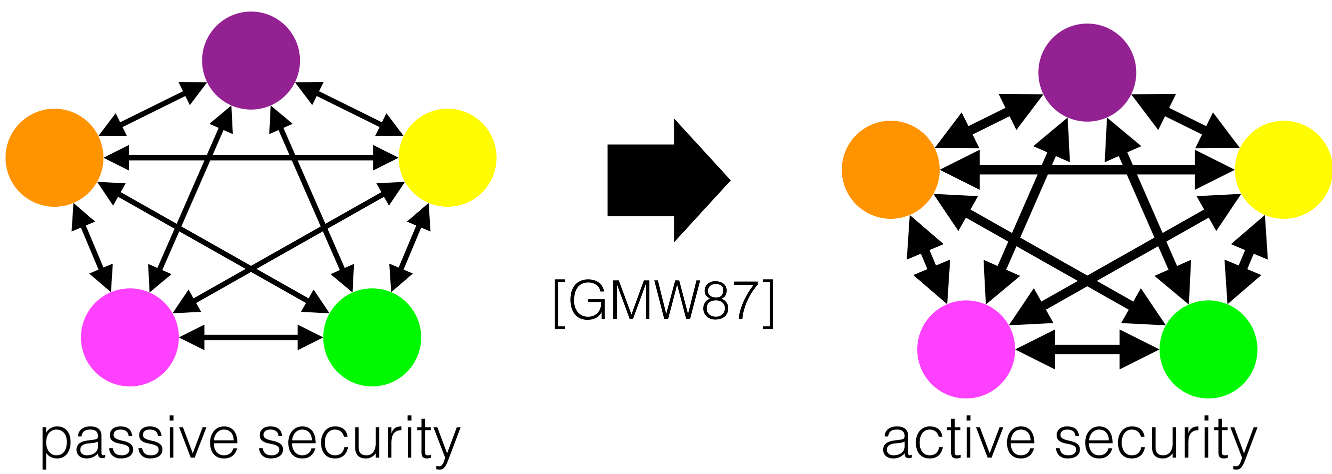
- zero knowledge
- proof of knowledge

2010s blockchain technology

Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

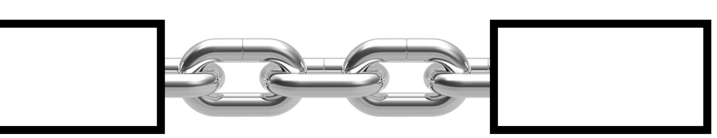
1980s securely compute $\boldsymbol{y}=\boldsymbol{F}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$
via a multi-party protocol

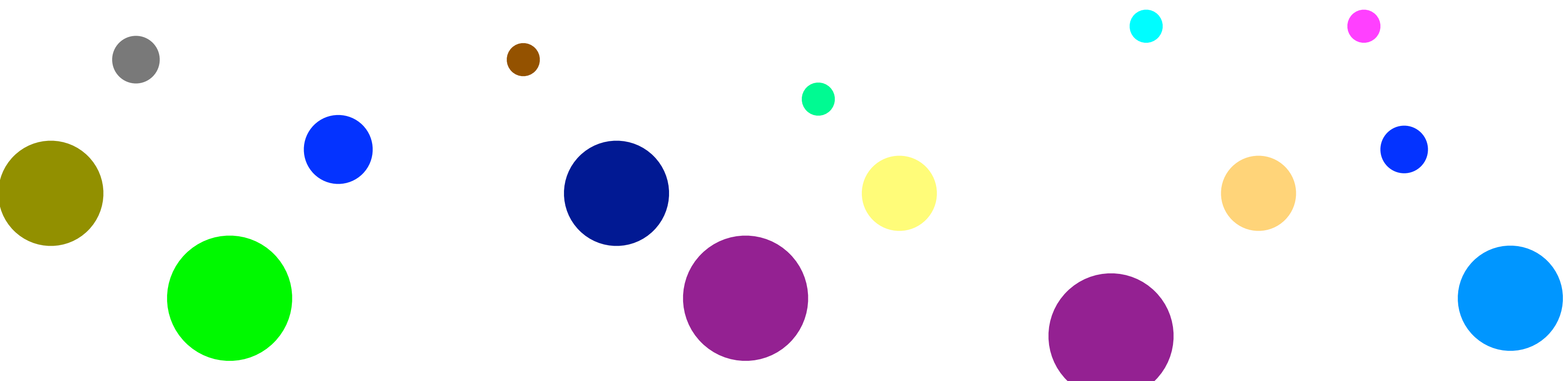


Key properties

- zero knowledge
- proof of knowledge

2010s blockchain technology

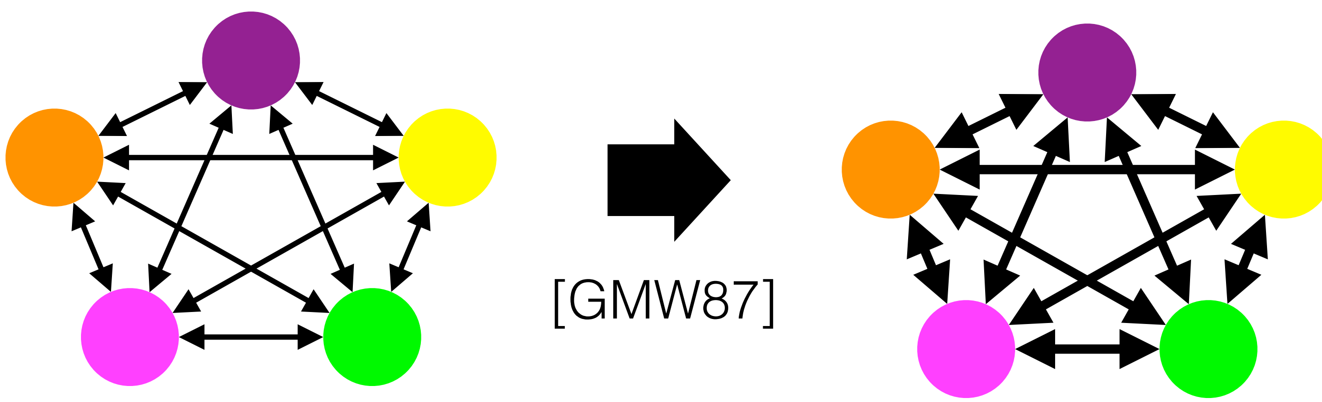




Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y=F(x_1,\ldots,x_n)$
via a multi-party protocol

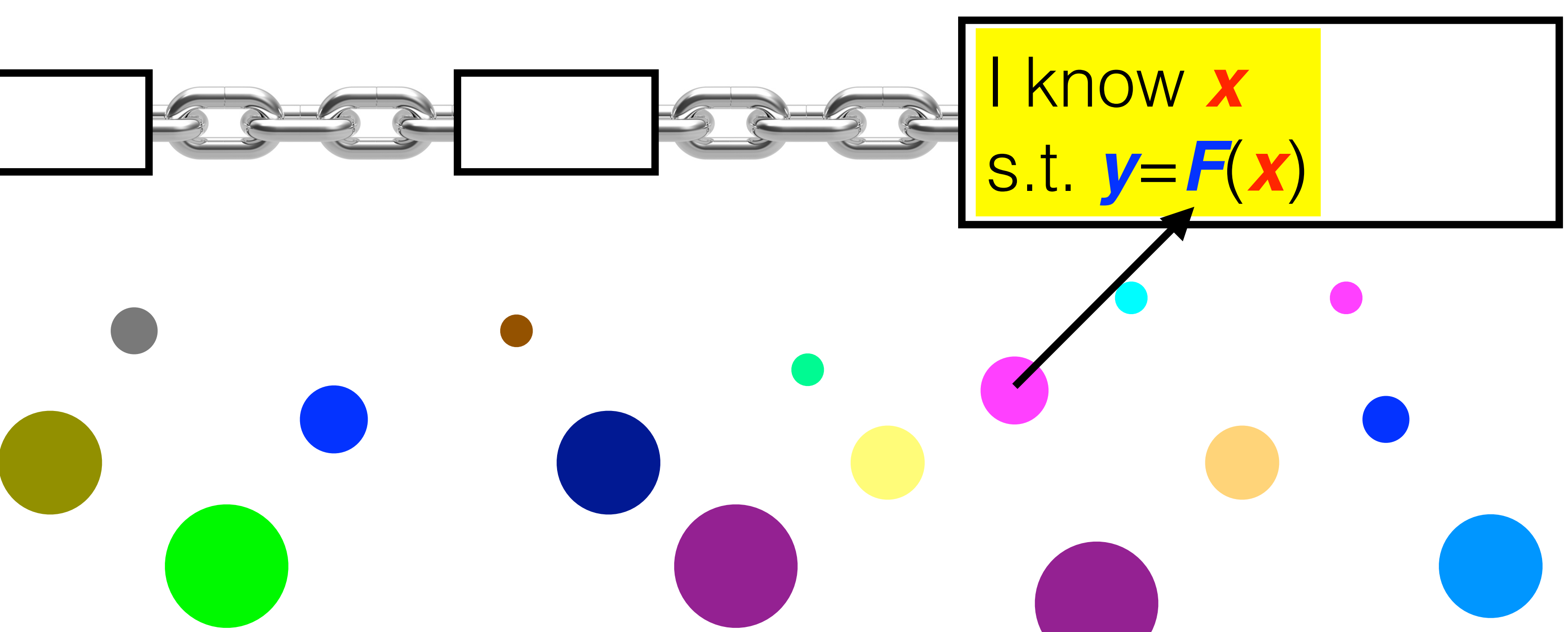


passive security active security

Key properties

- zero knowledge
- proof of knowledge

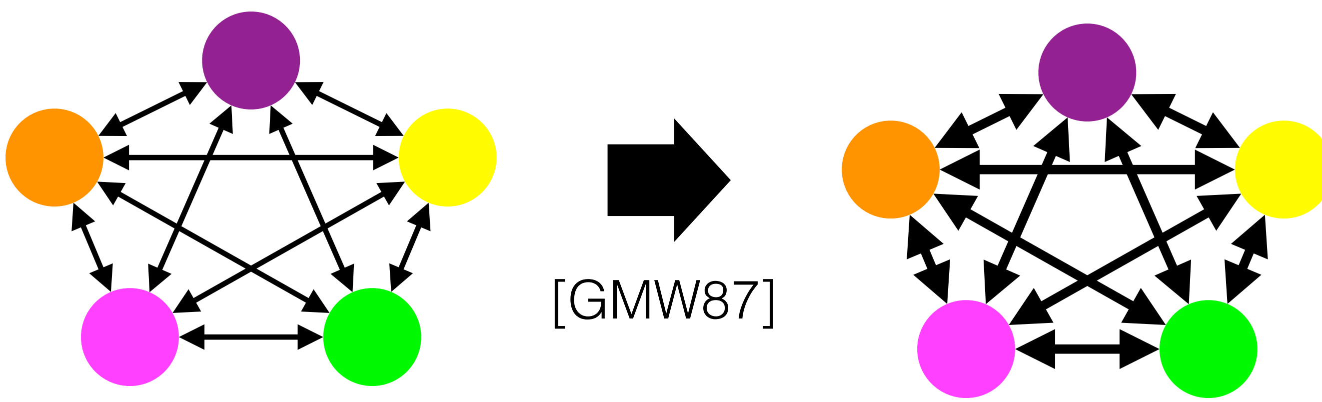
2010s blockchain technology



Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y=F(x_1,\ldots,x_n)$
via a multi-party protocol

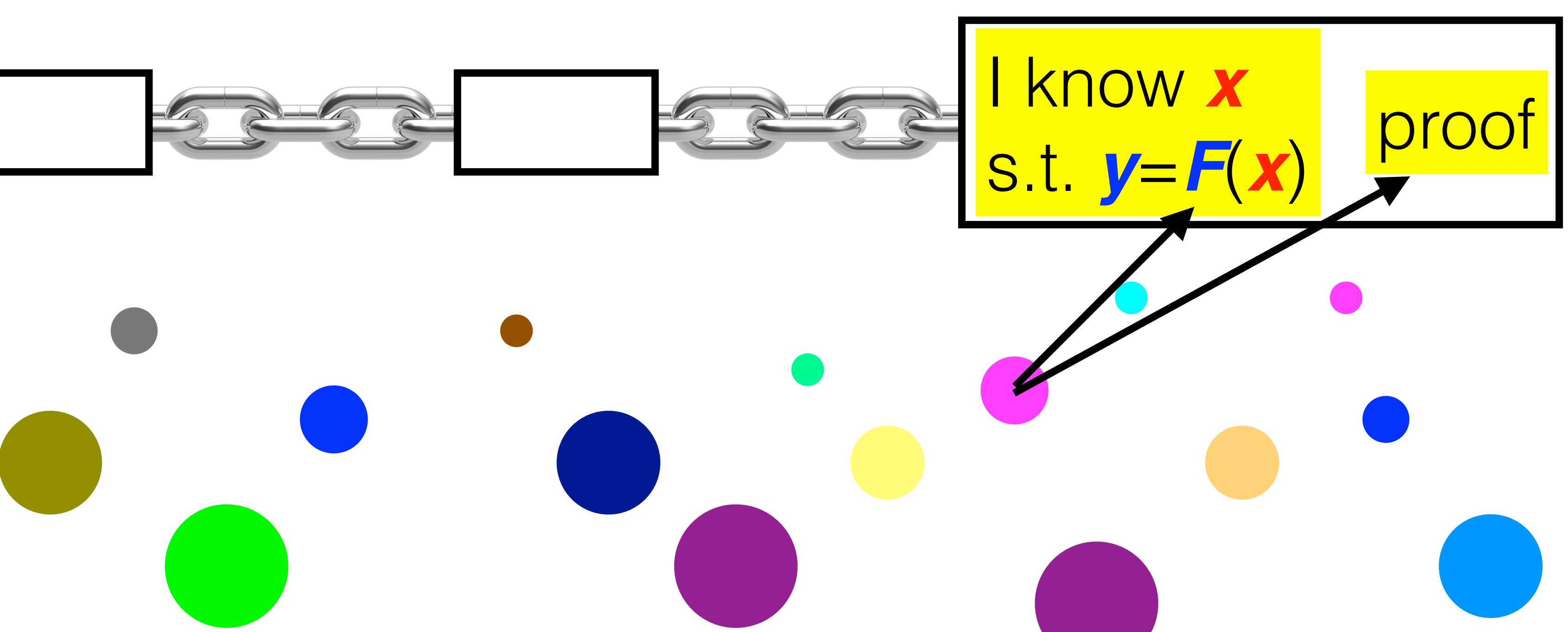


passive security   active security

Key properties

- zero knowledge
- proof of knowledge

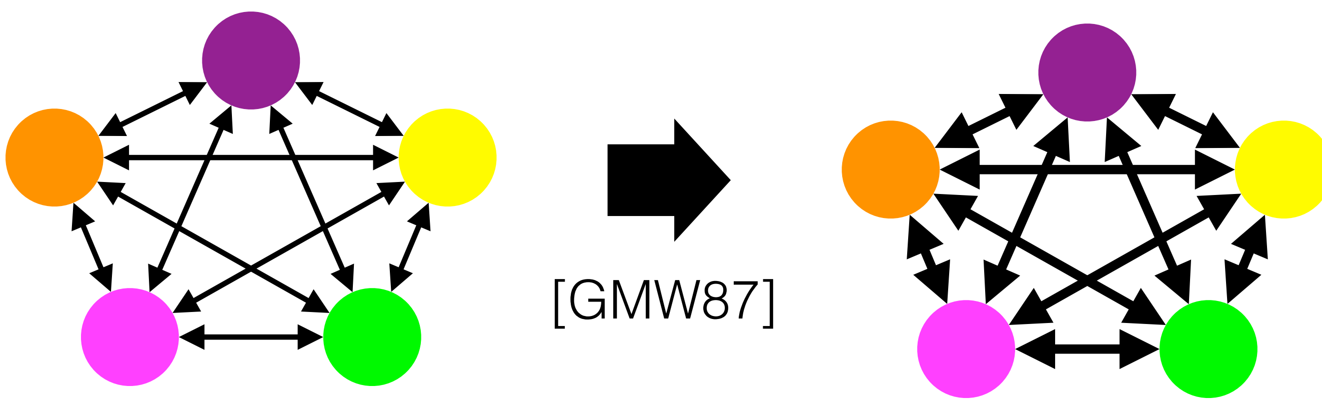
2010s blockchain technology



Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y=F(x_1,\ldots,x_n)$
via a multi-party protocol

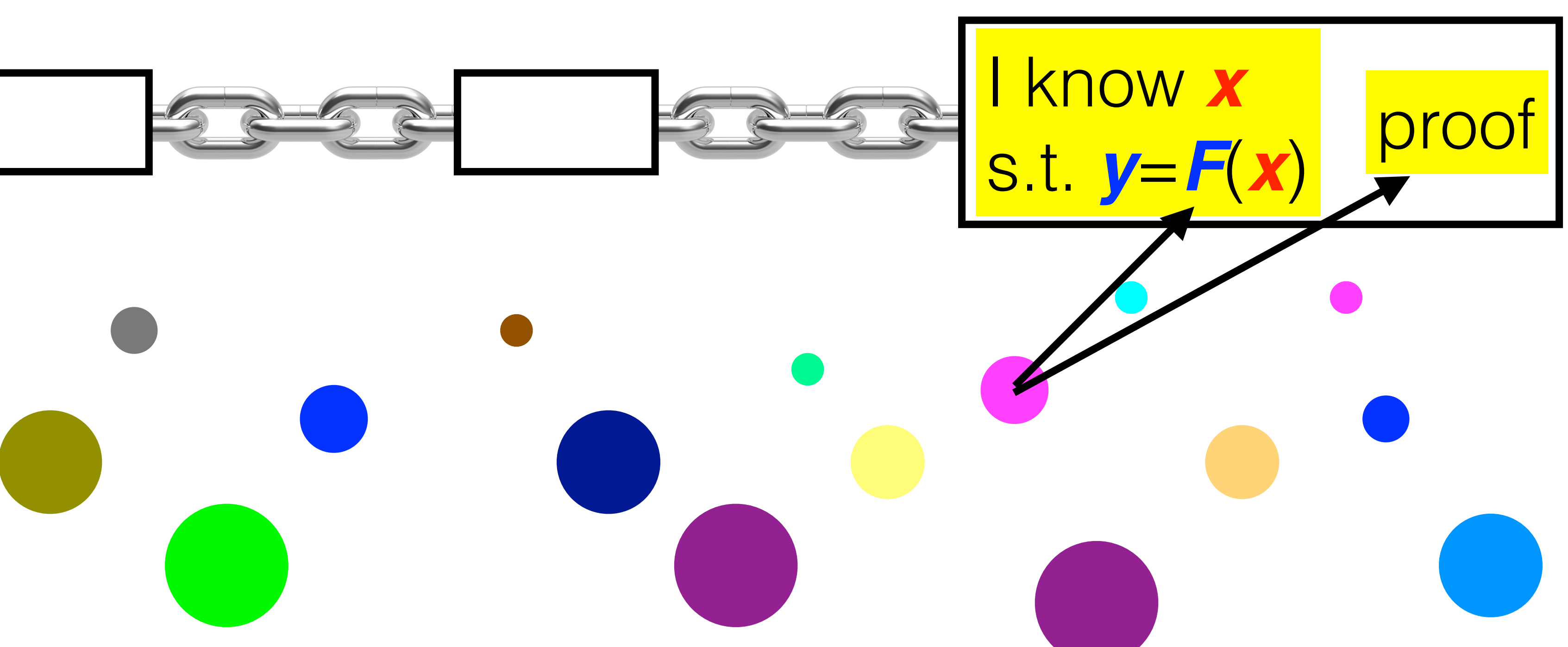


passive security active security

Key properties

- zero knowledge
- proof of knowledge

2010s blockchain technology



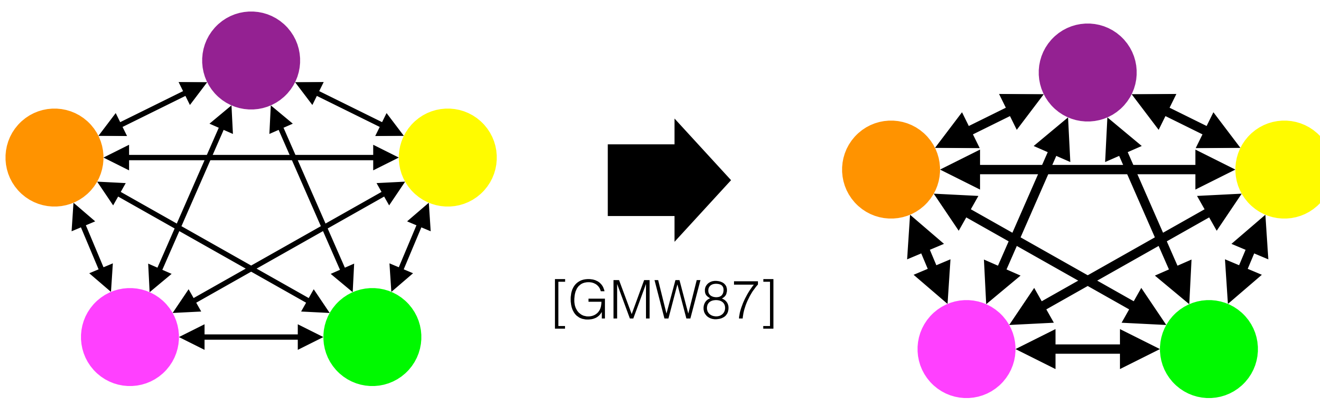
Additional key properties

- non-interactive
- publicly verifiable
- succinct

Cryptographic Proofs

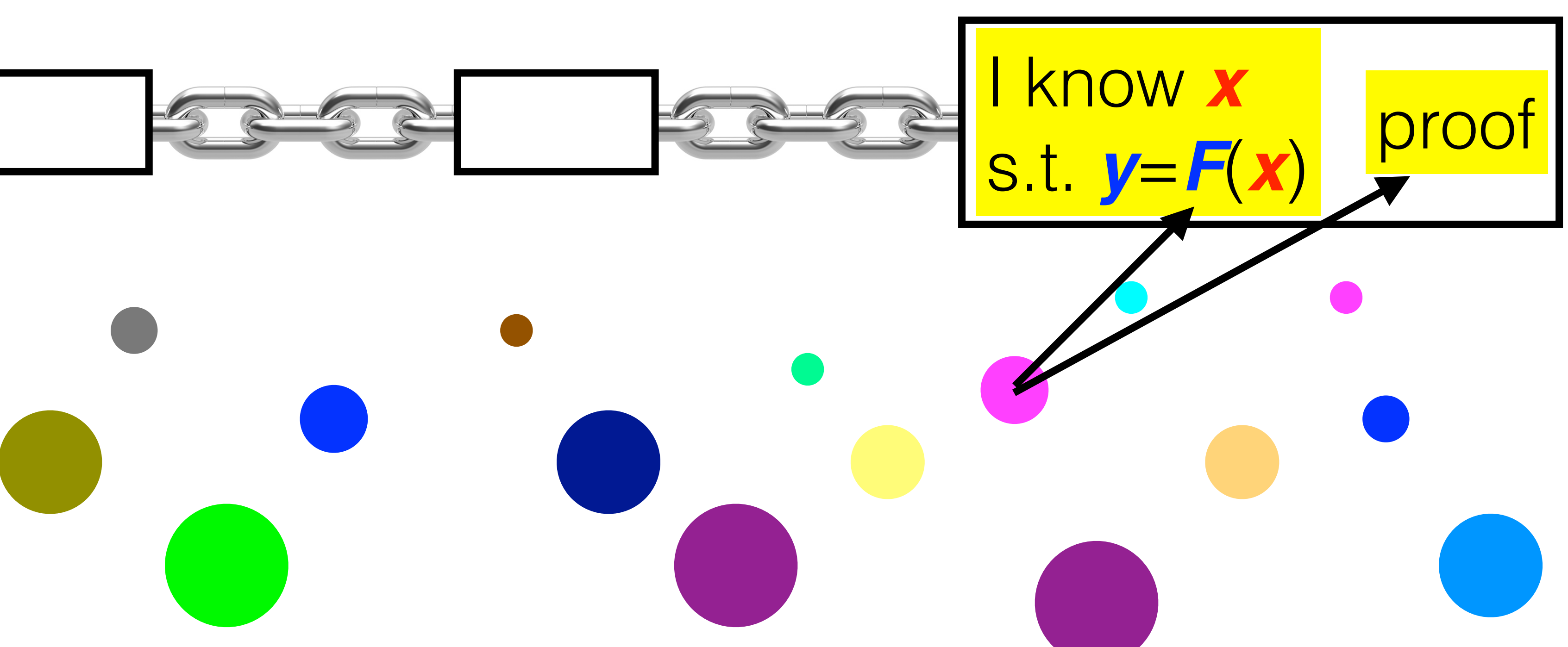
a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y=F(x_1,\ldots,x_n)$
via a multi-party protocol



passive security active security

2010s blockchain technology



zk-SNARK

Key properties

- zero knowledge
- proof of knowledge

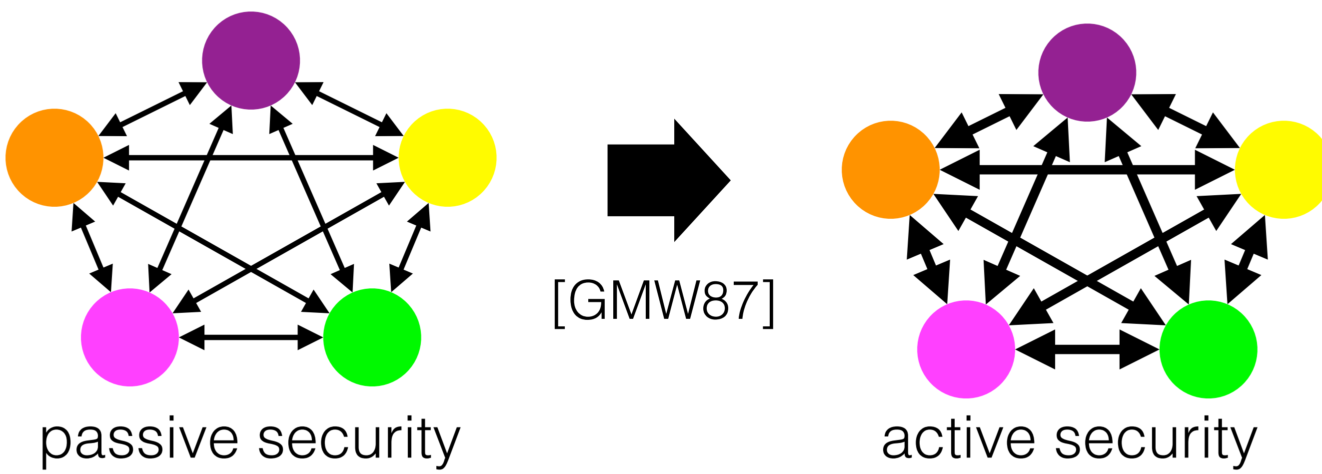
Additional key properties

- non-interactive
- publicly verifiable
- succinct

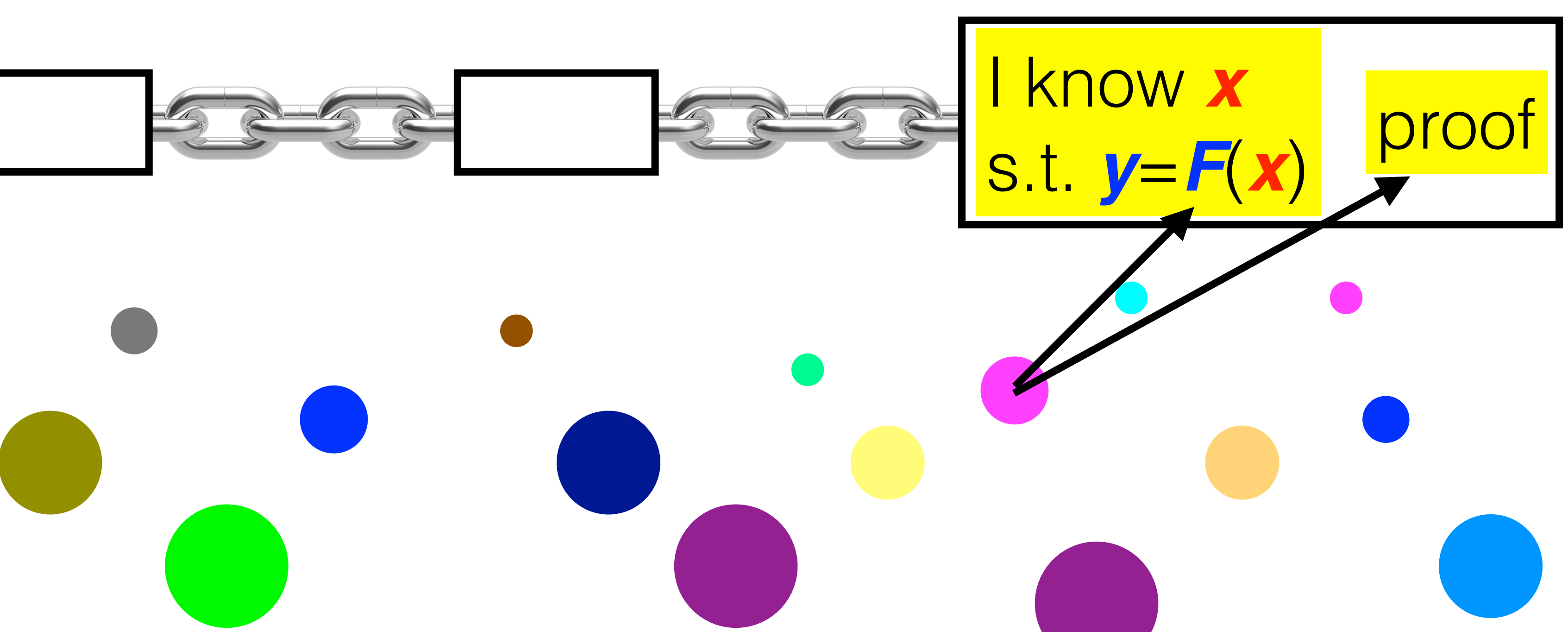
Cryptographic Proofs

a powerful defense against malicious behavior
especially in **distributed protocols**

1980s securely compute $y = F(x_1, \ldots, x_n)$
via a multi-party protocol



2010s blockchain technology



zk-SNARK

Key properties

- zero knowledge
- ~~proof of knowledge~~

Additional key properties

- non-interactive
- publicly verifiable
- succinct

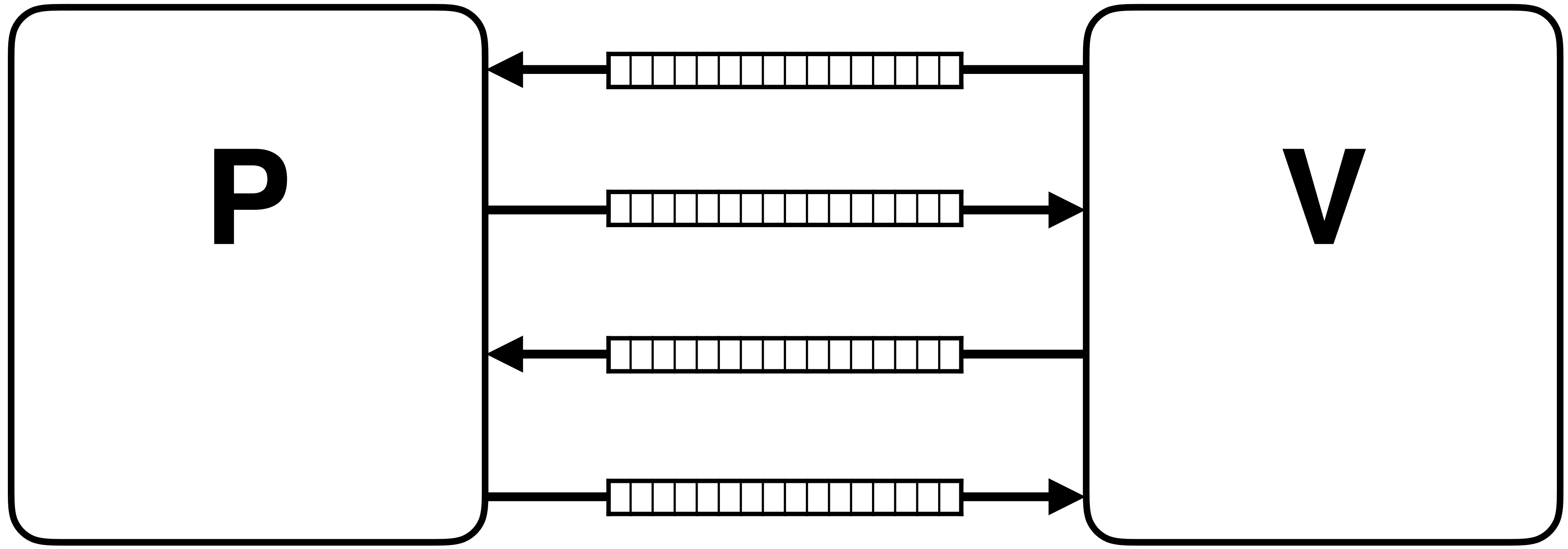
Origins

Zero Knowledge Proofs

[GMR85]

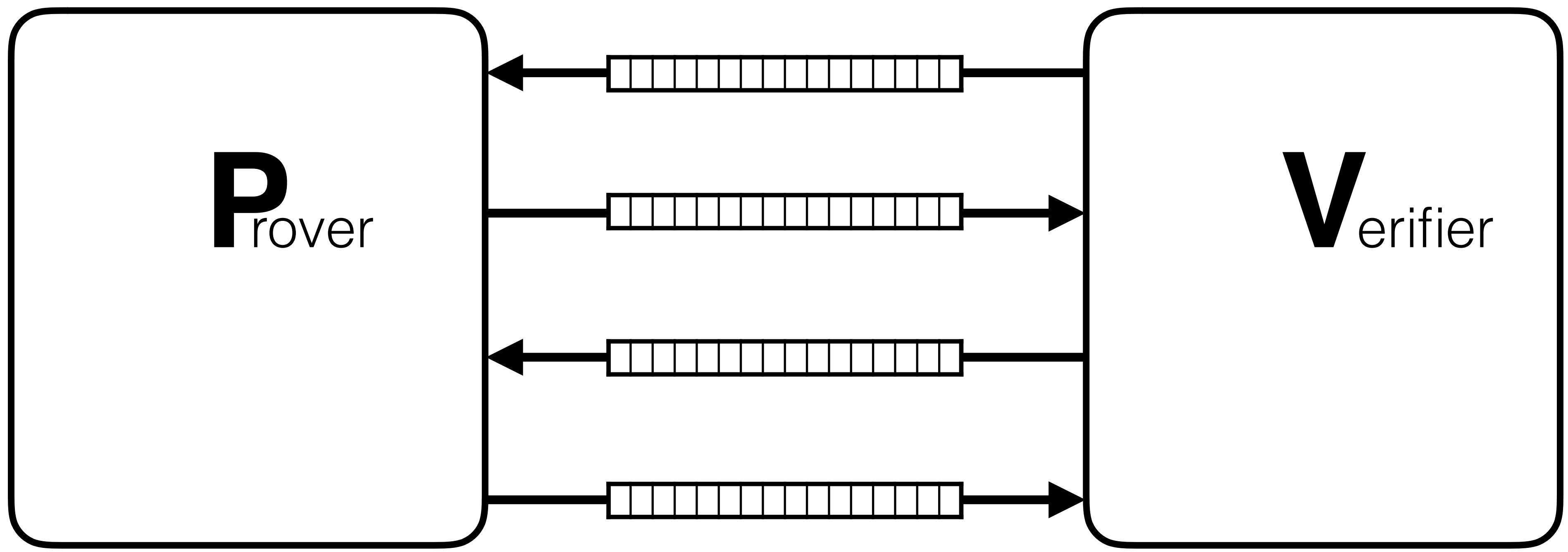
Zero Knowledge Proofs

[GMR85]



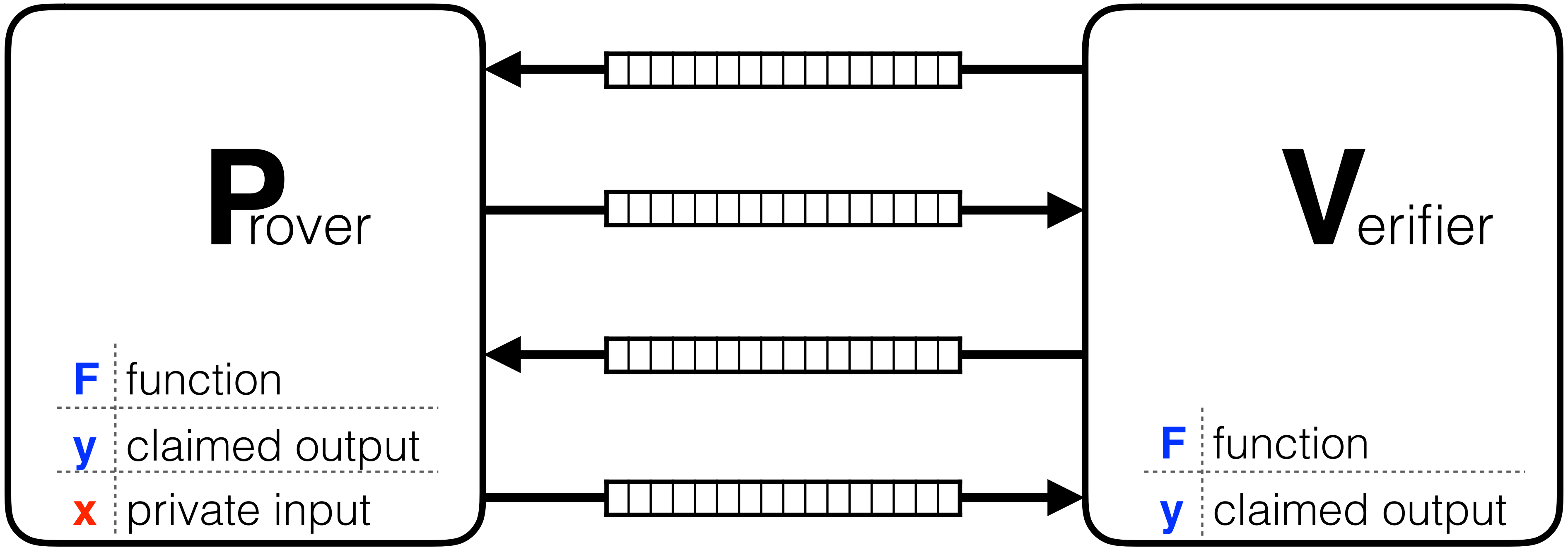
Zero Knowledge Proofs

[GMR85]



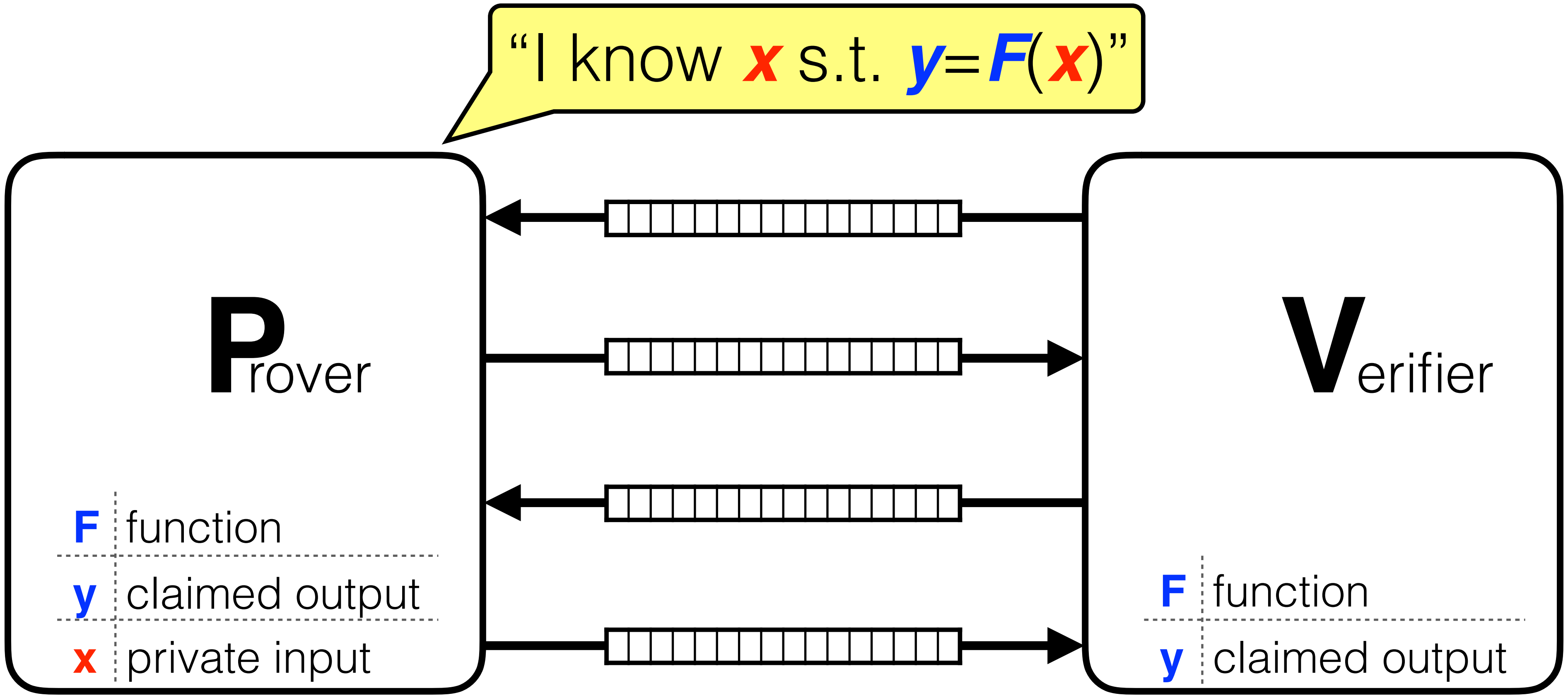
Zero Knowledge Proofs

[GMR85]



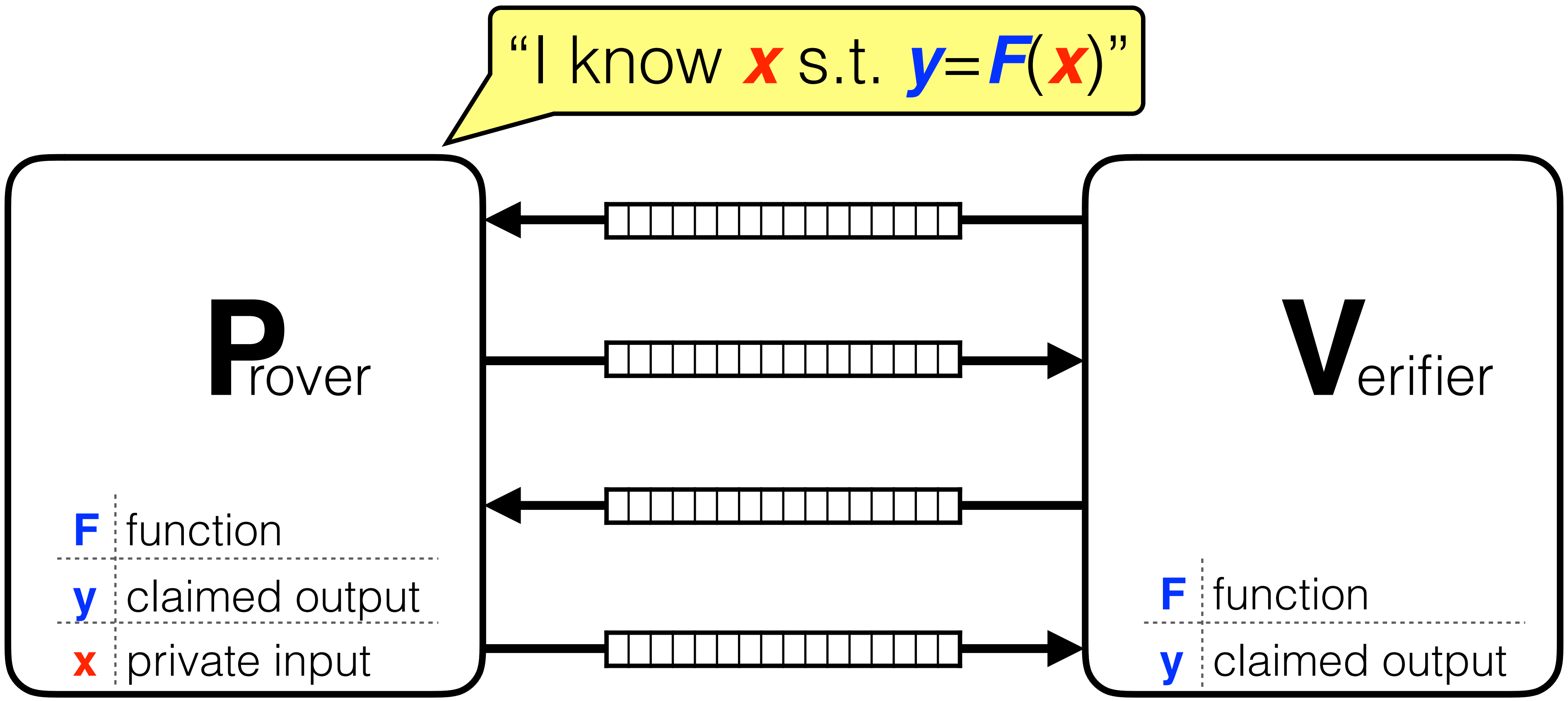
Zero Knowledge Proofs

[GMR85]



Zero Knowledge Proofs

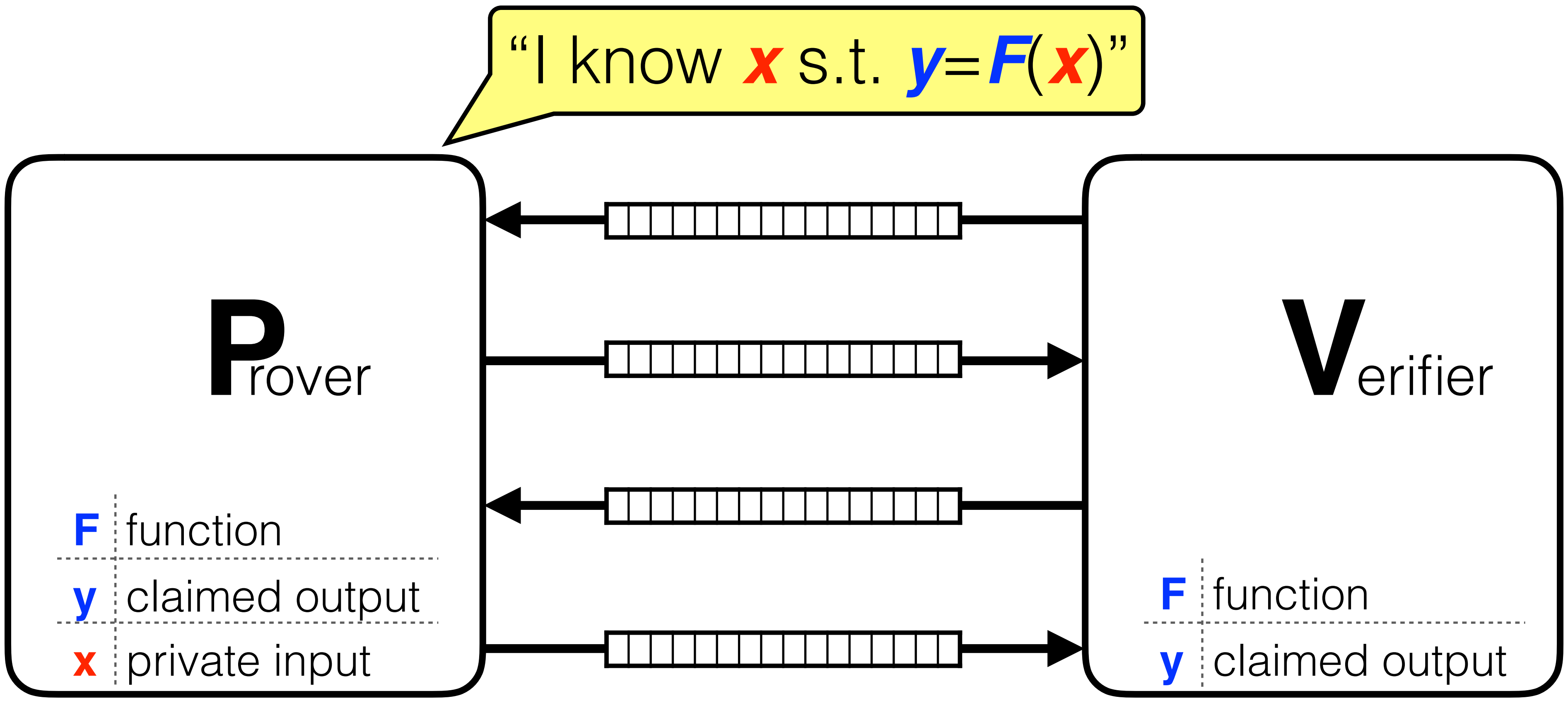
[GMR85]



completeness $\exists\, x\colon y = F(x) \rightarrow \Pr[\mathbf{P}(F,y,x) \text{ convinces } \mathbf{V}(F,y)] = 1$

Zero Knowledge Proofs

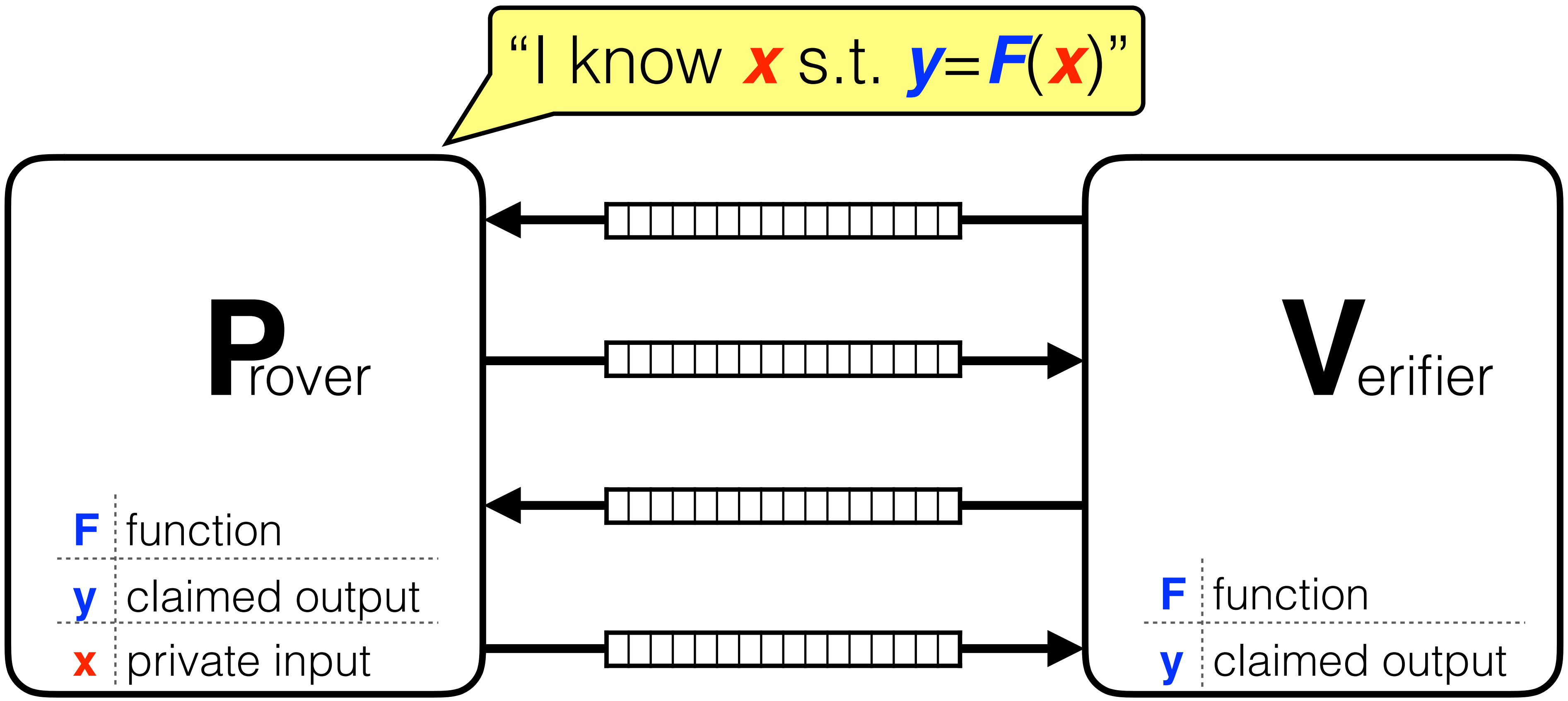
[GMR85]



completeness	$\exists\, x$: $y=F(x) \rightarrow \Pr[\mathbf{P}(F,y,x)$ convinces $\mathbf{V}(F,y)]=1$
soundness	$\nexists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{P'}$, $\Pr[\mathbf{P'}$ convinces $\mathbf{V}(F,y)]\approx 0$

Zero Knowledge Proofs

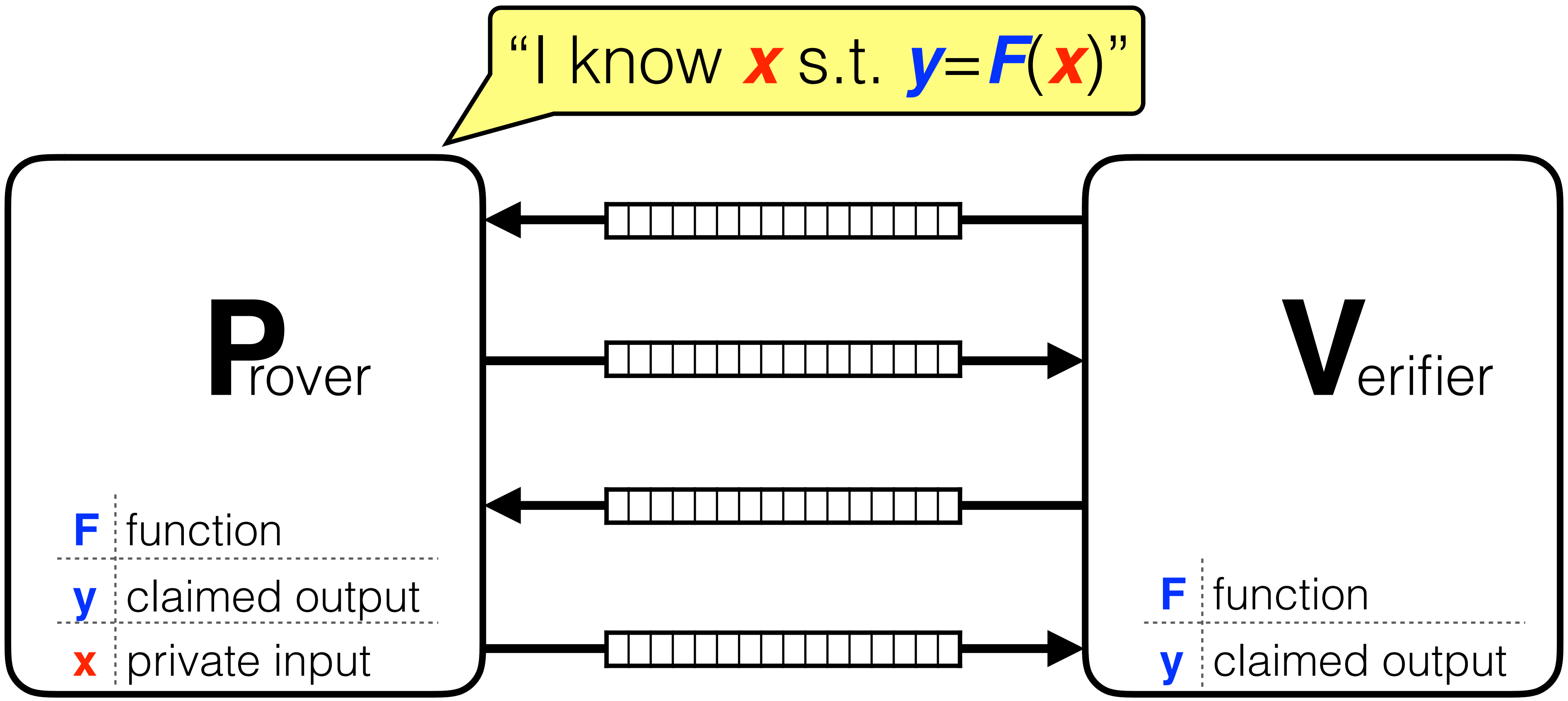
[GMR85]



completeness	$\exists\, x$: $y=F(x) \rightarrow \Pr[\mathbf{P}(F,y,x)$ convinces $\mathbf{V}(F,y)]=1$
soundness	$\nexists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{P}'$, $\Pr[\mathbf{P}'$ convinces $\mathbf{V}(F,y)]\simeq 0$
zero knowledge	$\exists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{V}'$, $\mathbf{S}(\mathbf{V}',F,y) \simeq$ view of $\mathbf{V}'$ with $\mathbf{P}(F,y,x)$

Zero Knowledge Proofs

[GMR85]

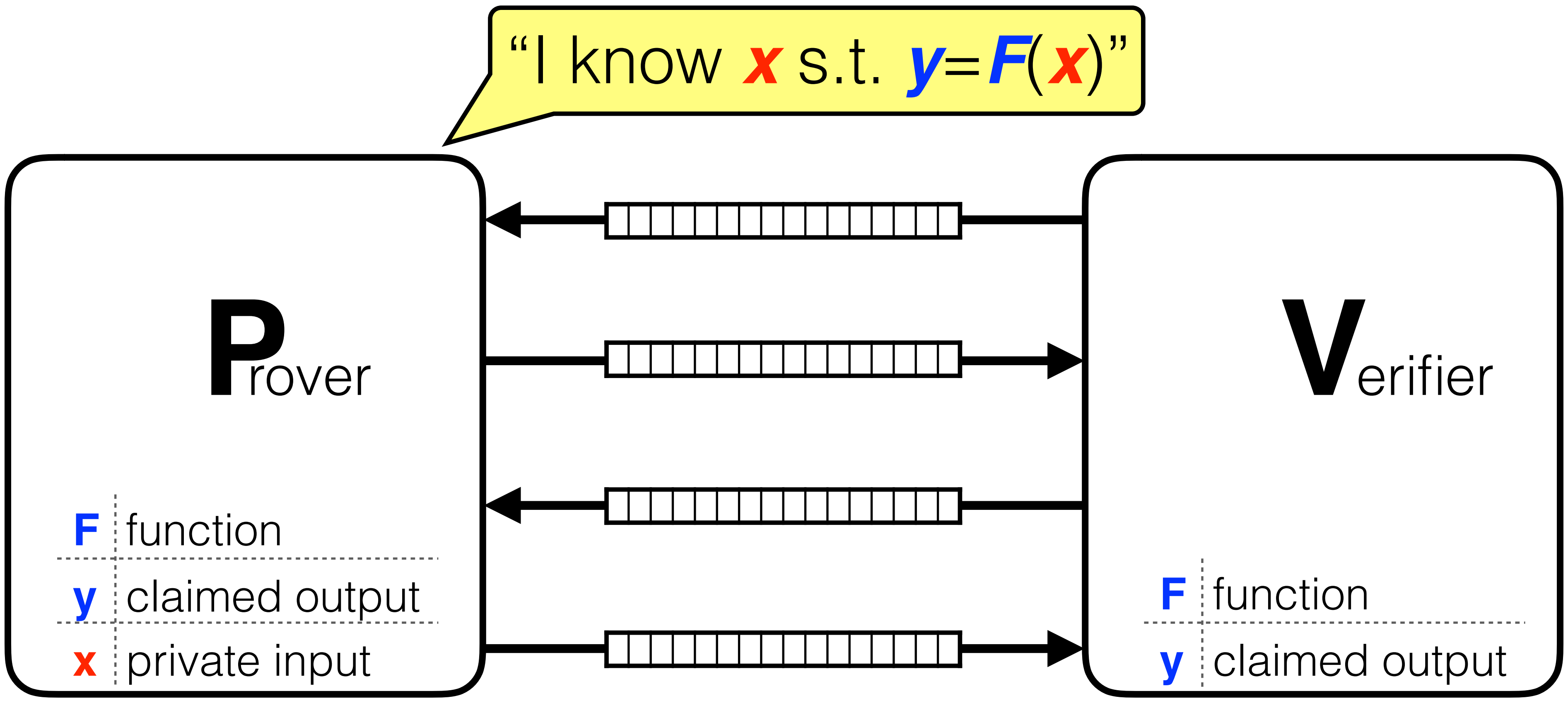


completeness	$\exists\, x$: $y=F(x) \rightarrow \Pr[\mathbf{P}(F,y,x)$ convinces $\mathbf{V}(F,y)]=1$
soundness	$\nexists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{P'}, \Pr[\mathbf{P'}$ convinces $\mathbf{V}(F,y)]\approx 0$
zero knowledge	$\exists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{V'}, \mathbf{S}(\mathbf{V'},F,y) \simeq$ view of $\mathbf{V'}$ with $\mathbf{P}(F,y,x)$

simulator

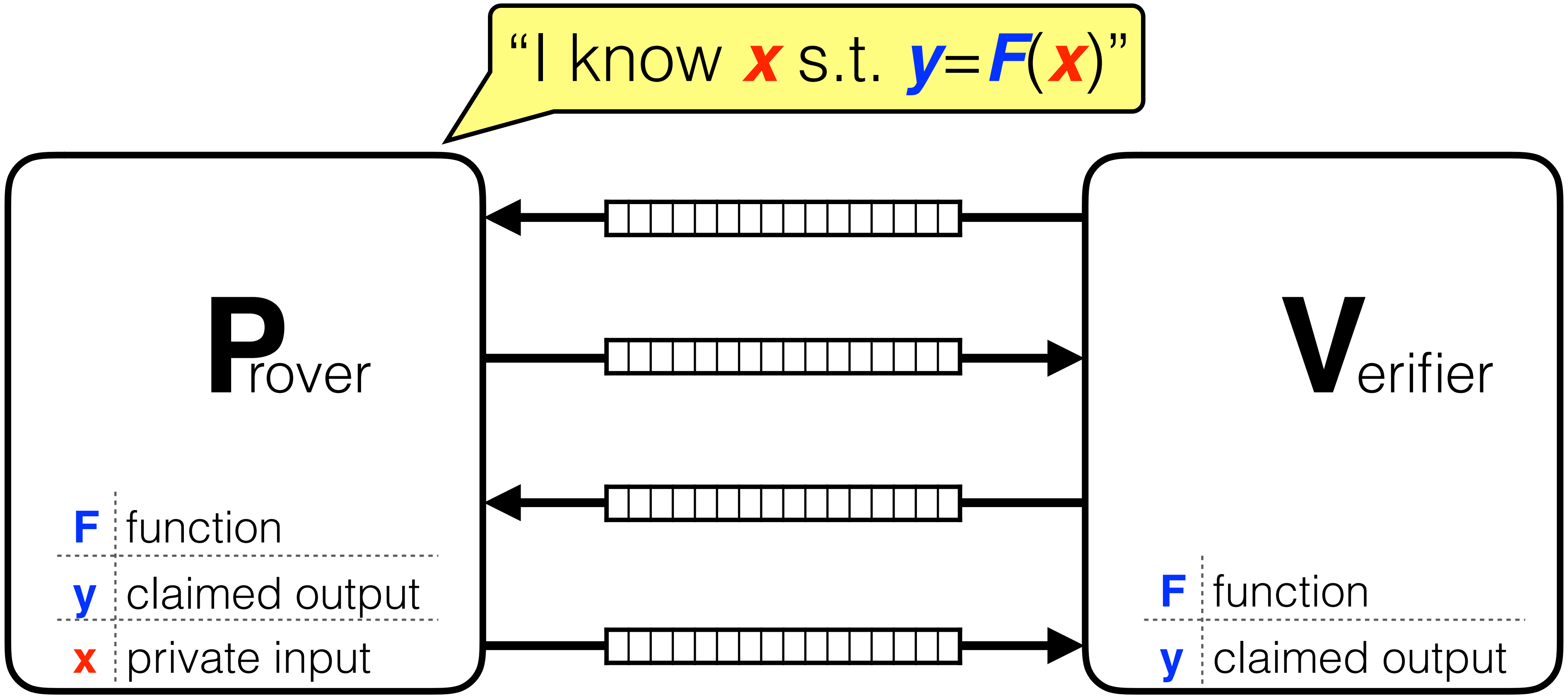
Zero Knowledge Proofs

[GMR85]



Zero Knowledge Proofs

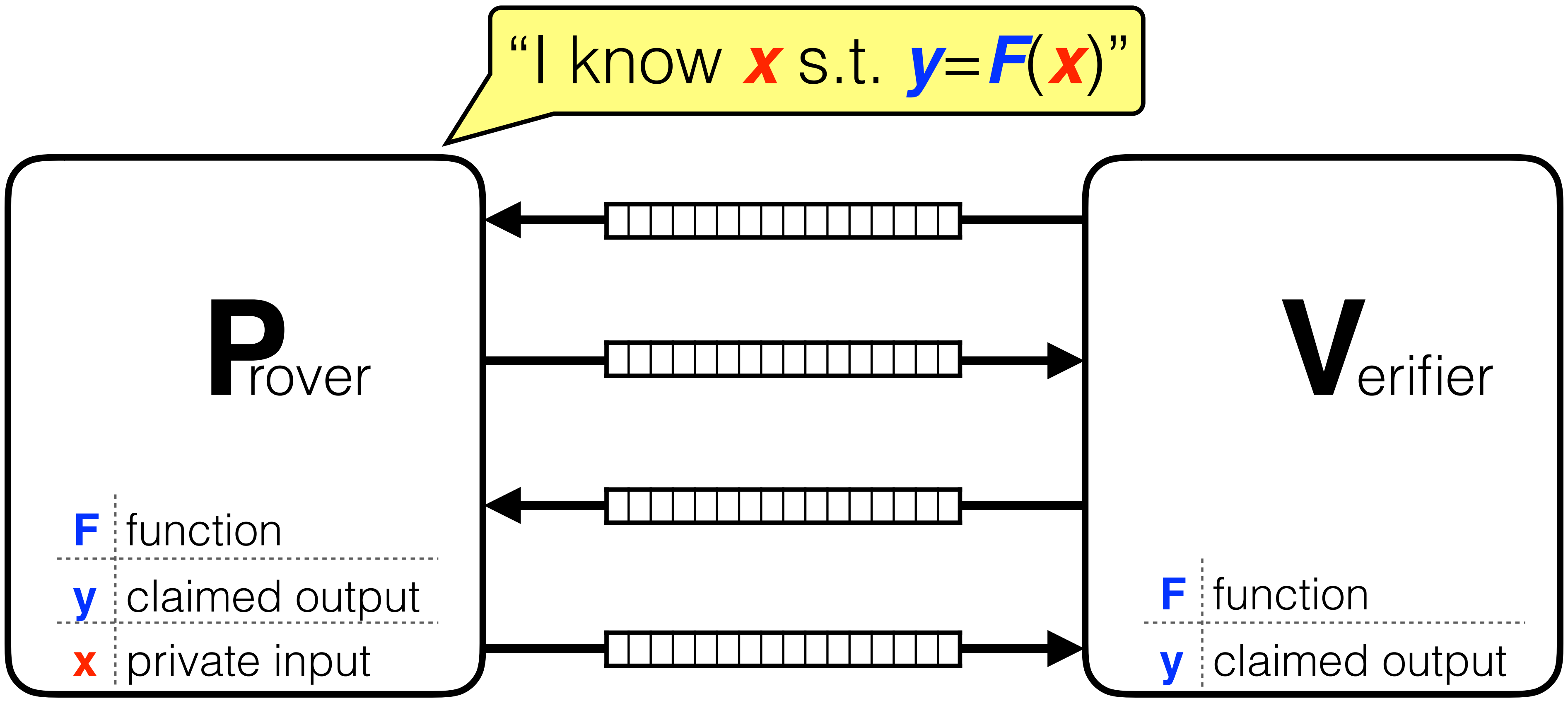
[GMR85]



[GMR85]: ZKPs for certain number-theoretic problems (QR,QNR)

Zero Knowledge Proofs

[GMR85]

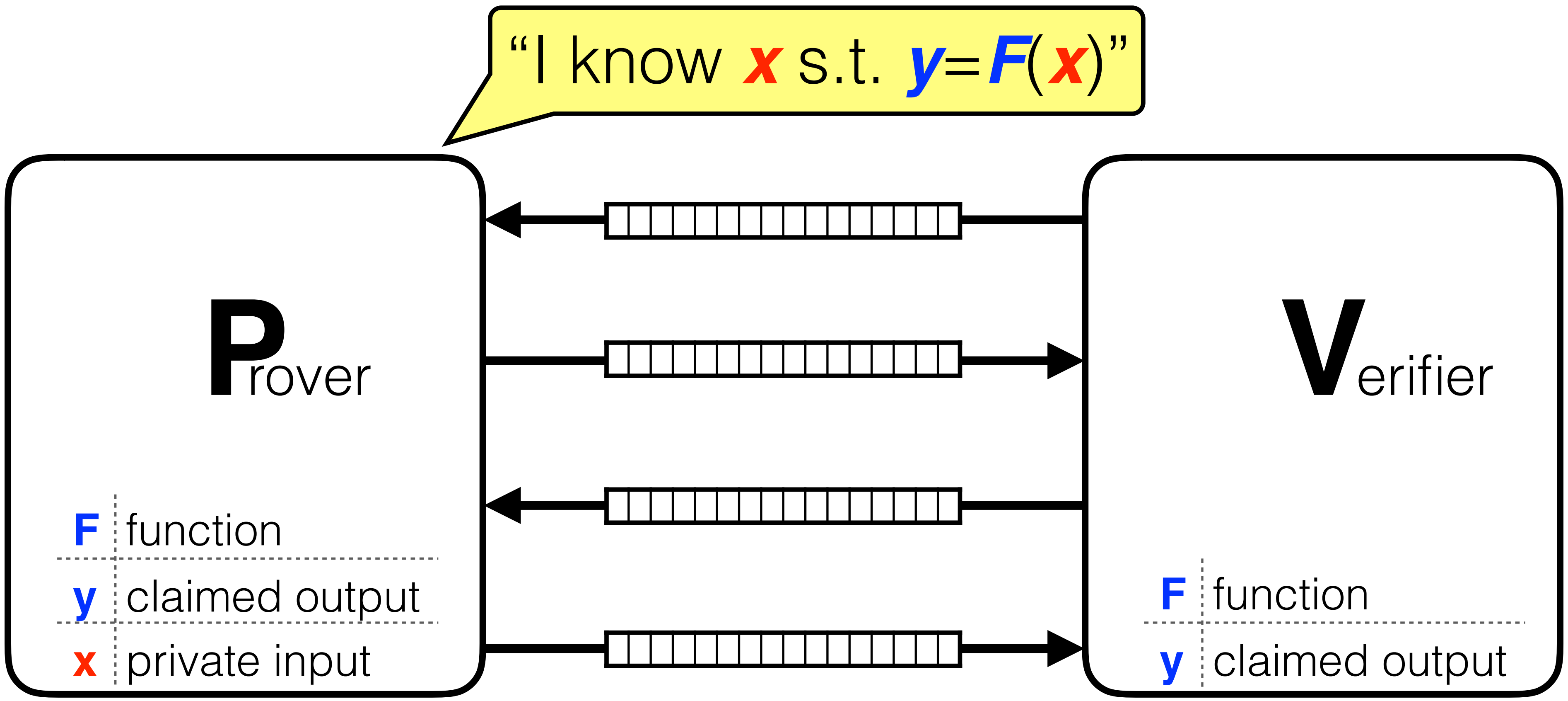


[GMR85]: ZKPs for certain number-theoretic problems (QR,QNR)

If one-way functions exist:

Zero Knowledge Proofs

[GMR85]



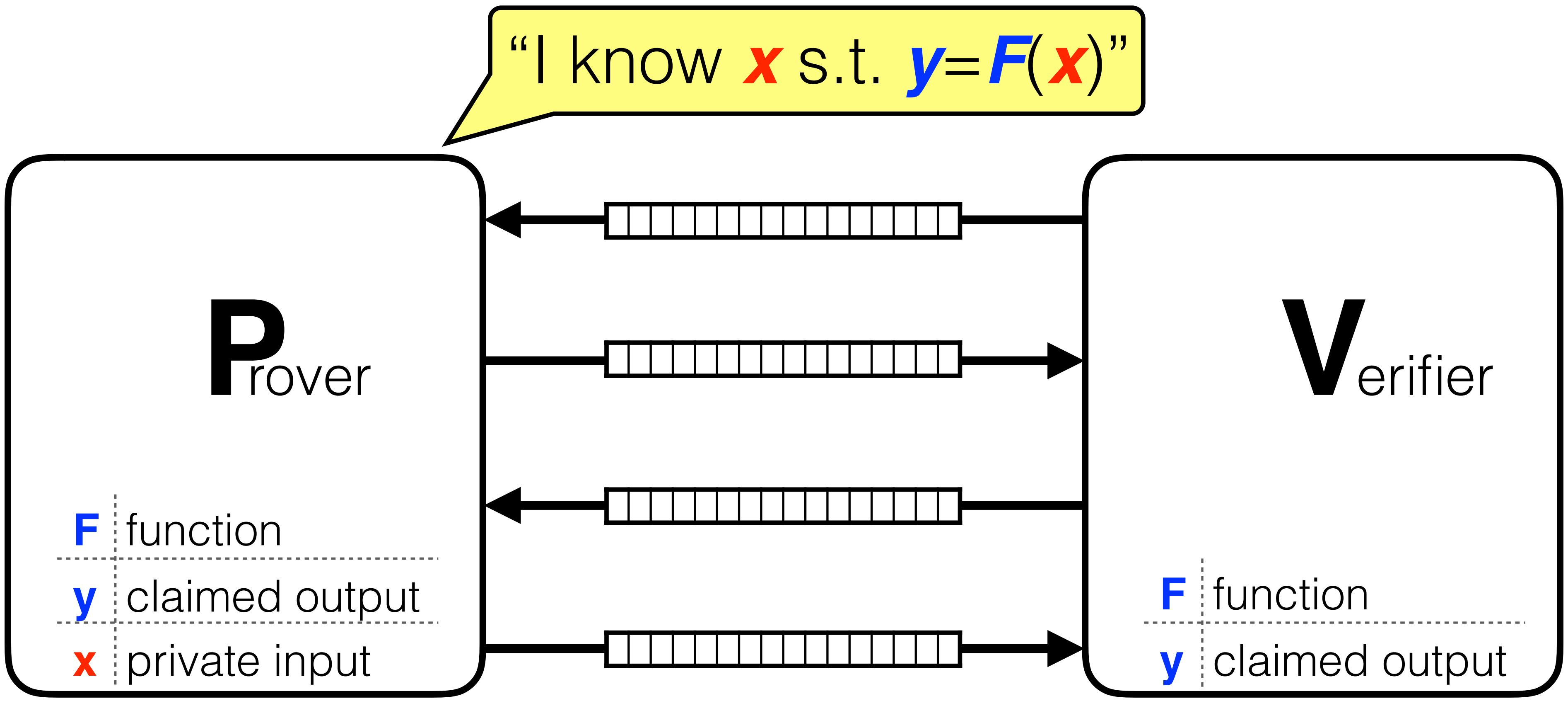
[GMR85]: ZKPs for certain number-theoretic problems (QR,QNR)

If one-way functions exist:

[GMW86]: ZKPs for all poly-**time** computable functions **F**

Zero Knowledge Proofs

[GMR85]



[GMR85]: ZKPs for certain number-theoretic problems (QR,QNR)

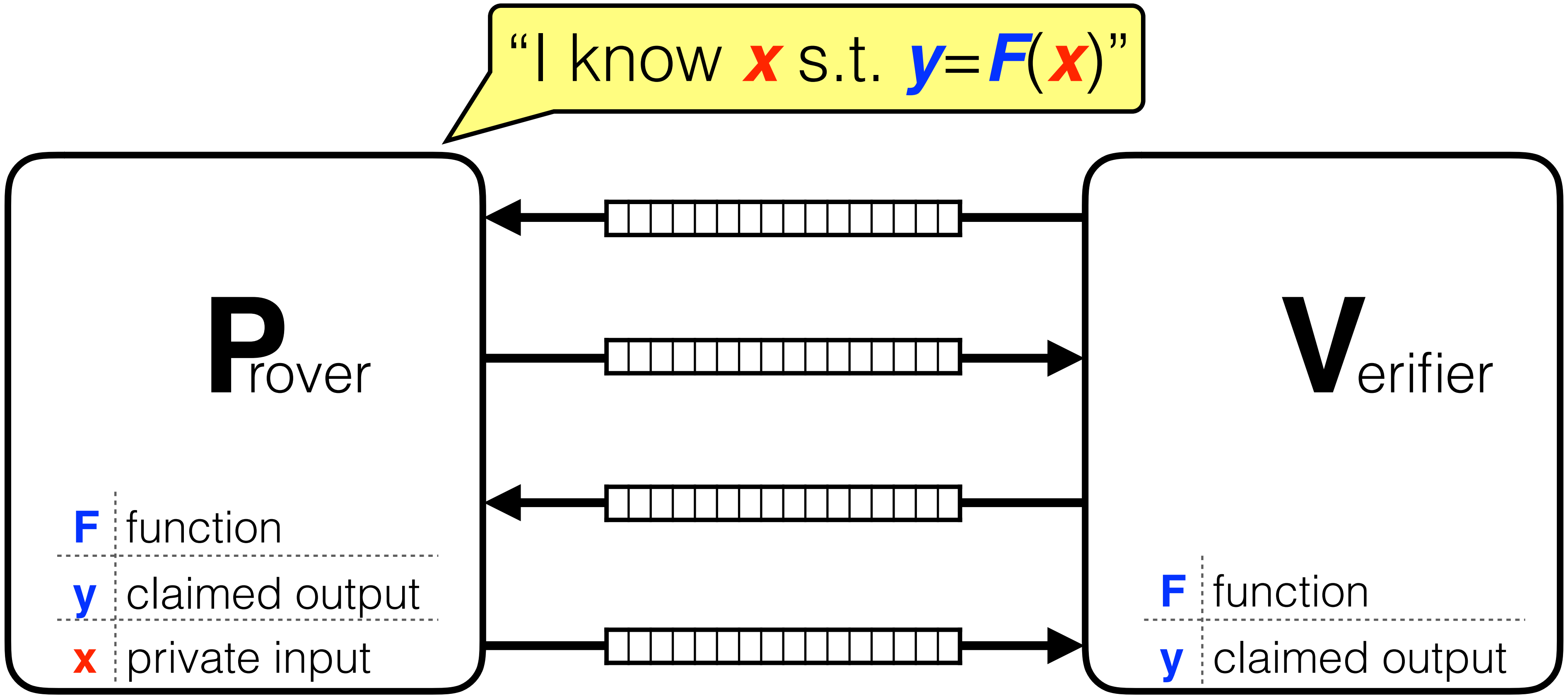
If one-way functions exist:

[GMW86]: ZKPs for all poly-**time** computable functions **F**

[BGGHKMR88]: ZKPs for all poly-**space** computable functions **F**

Zero Knowledge Proofs

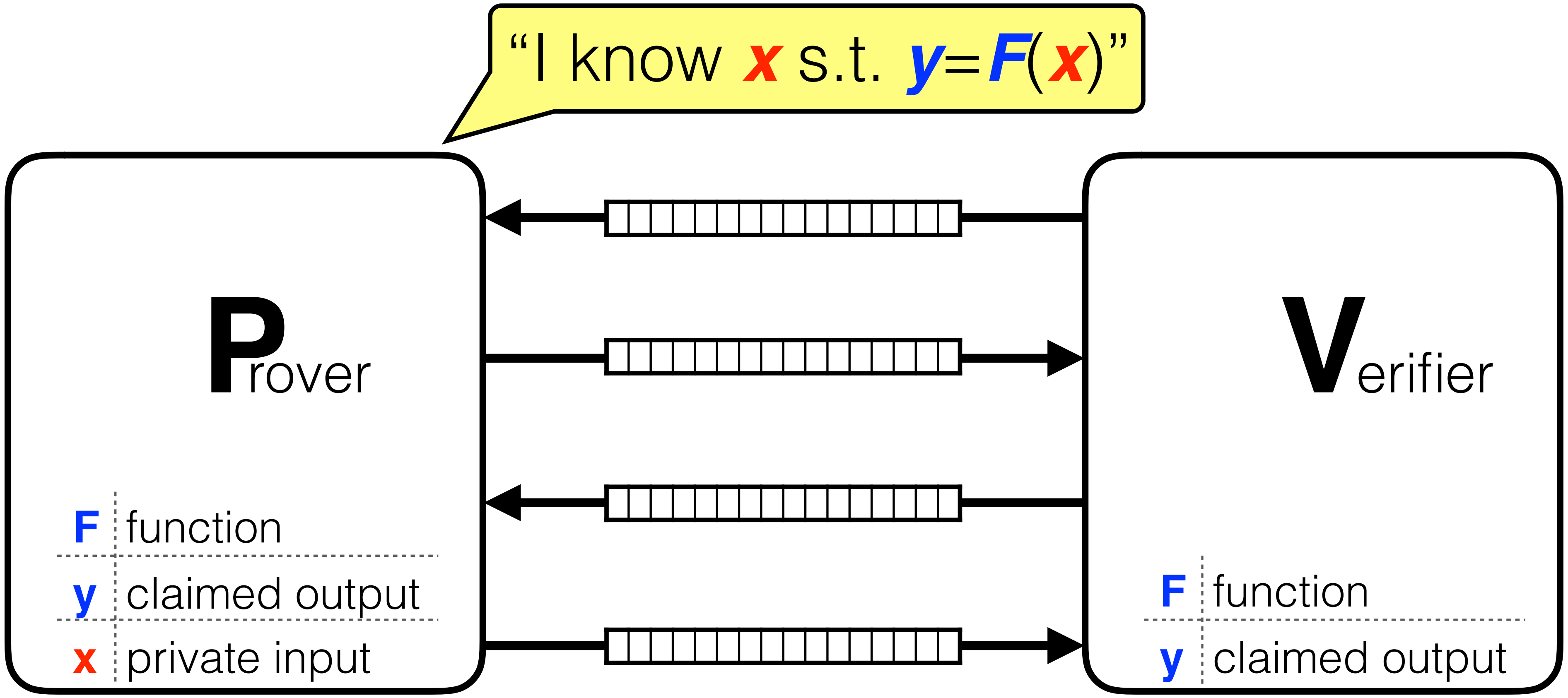
[GMR85]



Powerful cryptographic primitive.

Zero Knowledge Proofs

[GMR85]

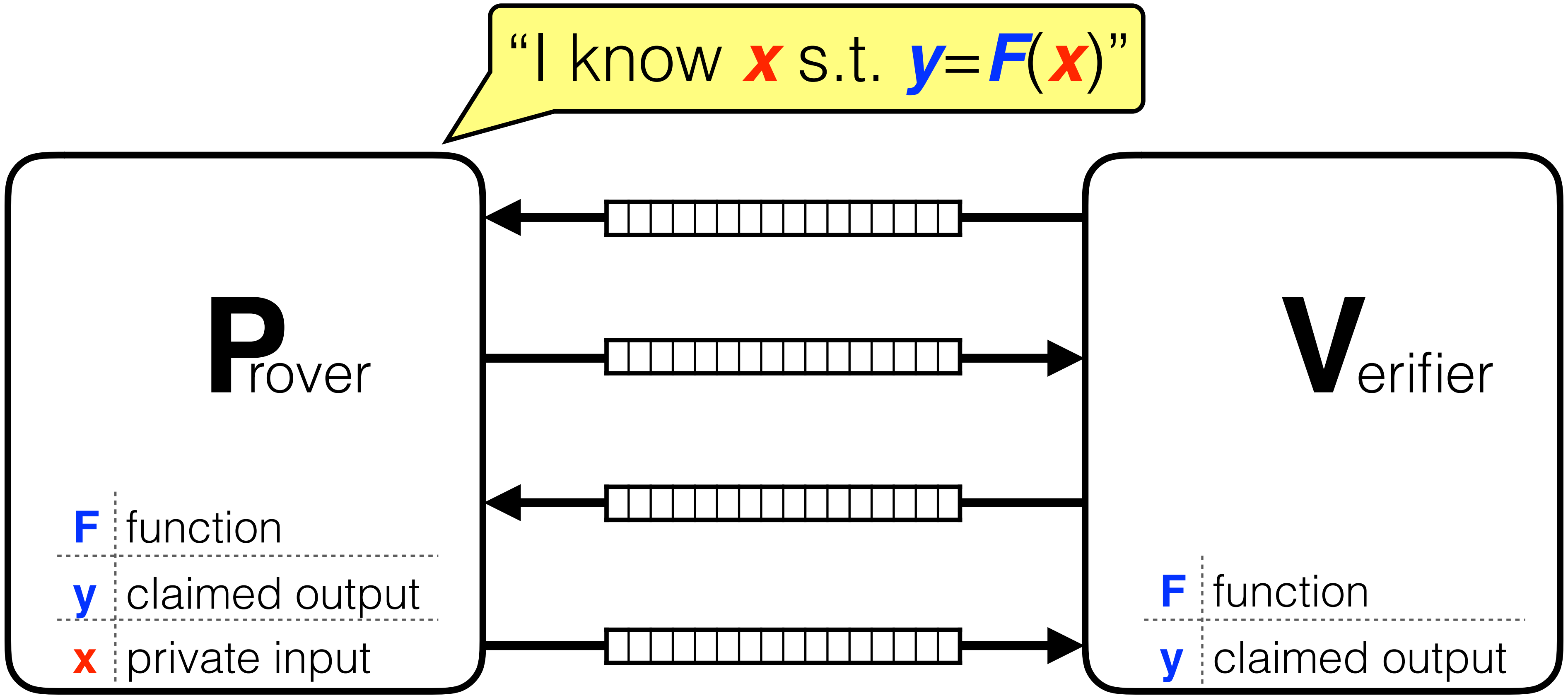


Powerful cryptographic primitive.

BUT

Zero Knowledge Proofs

[GMR85]



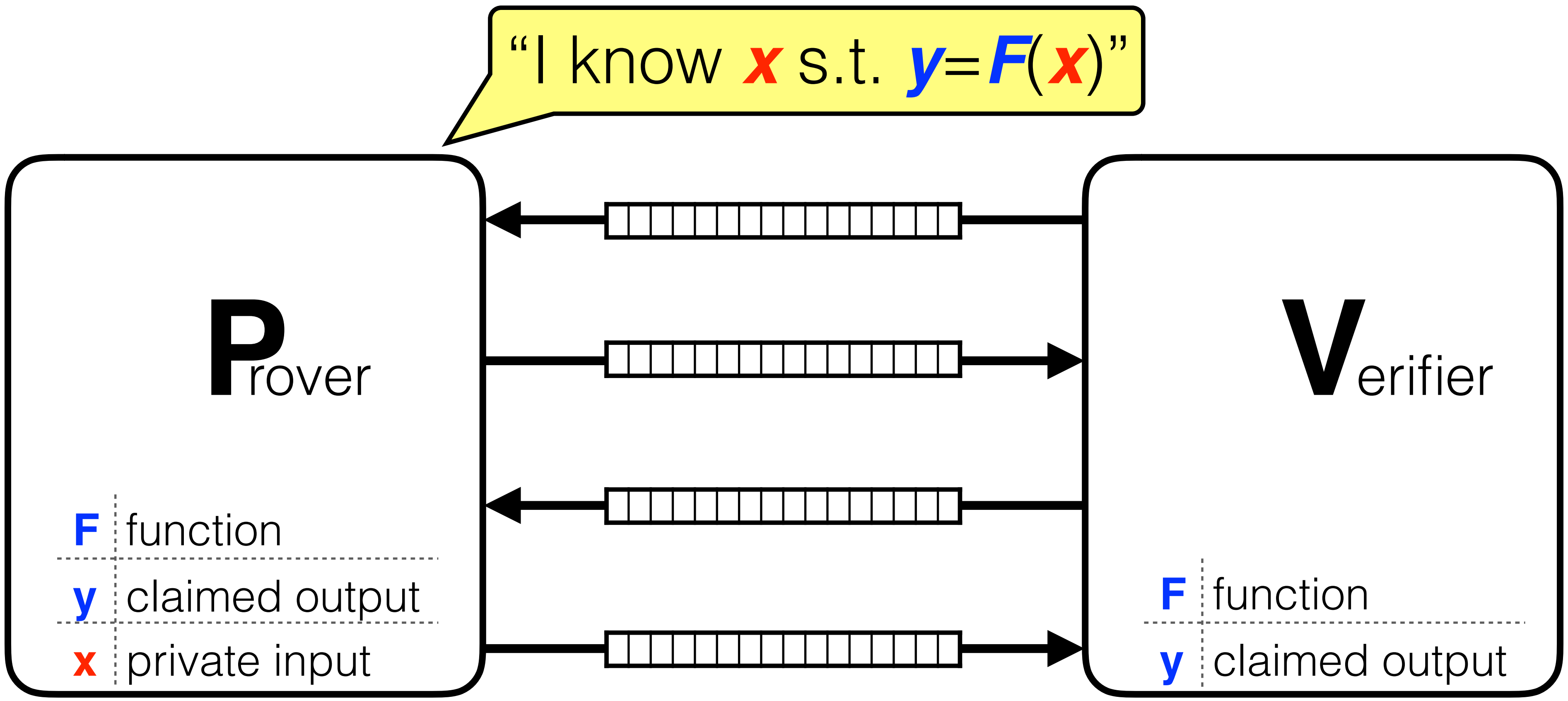
Powerful cryptographic primitive.

BUT

interactive

Zero Knowledge Proofs

[GMR85]



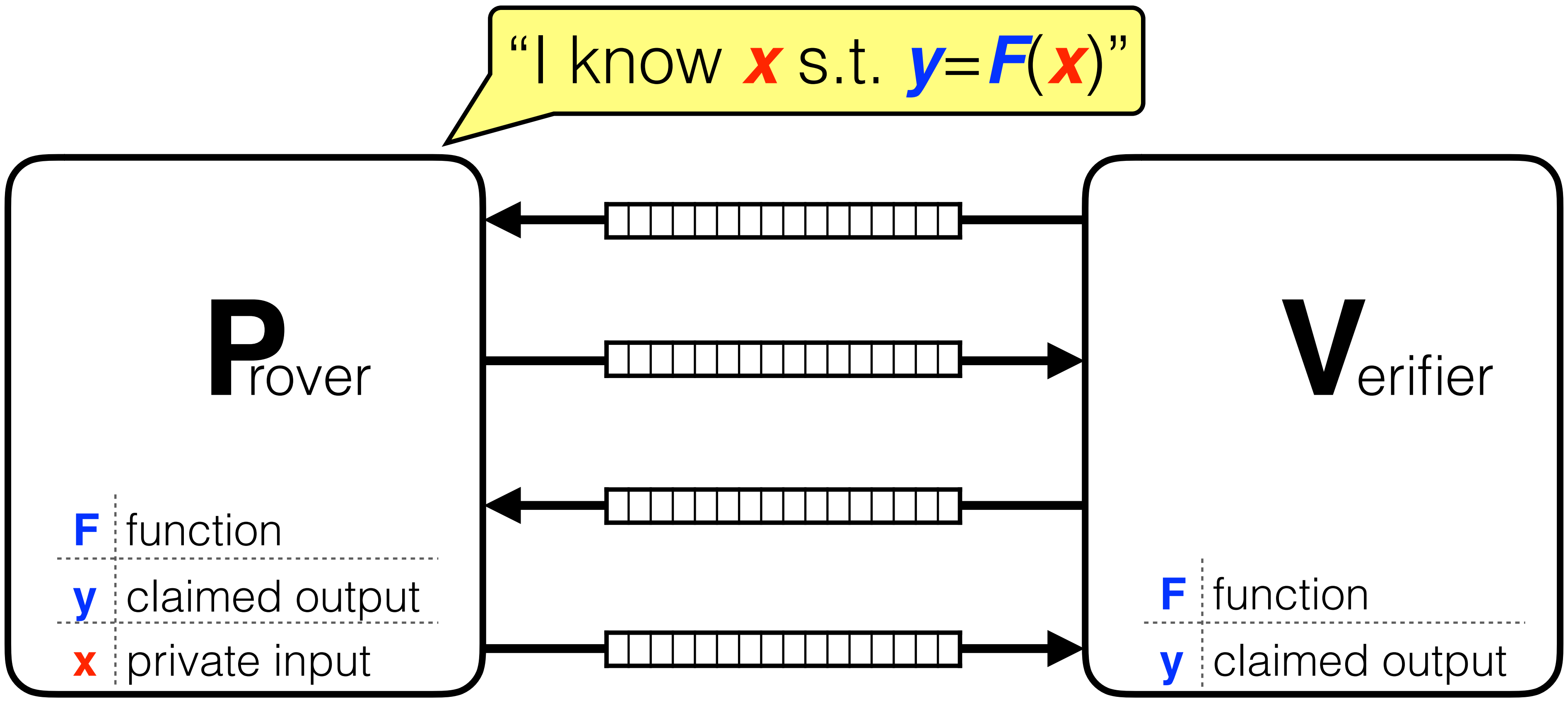
Powerful cryptographic primitive.

BUT

interactive **not succinct**

Zero Knowledge Proofs

[GMR85]



Powerful cryptographic primitive.

BUT

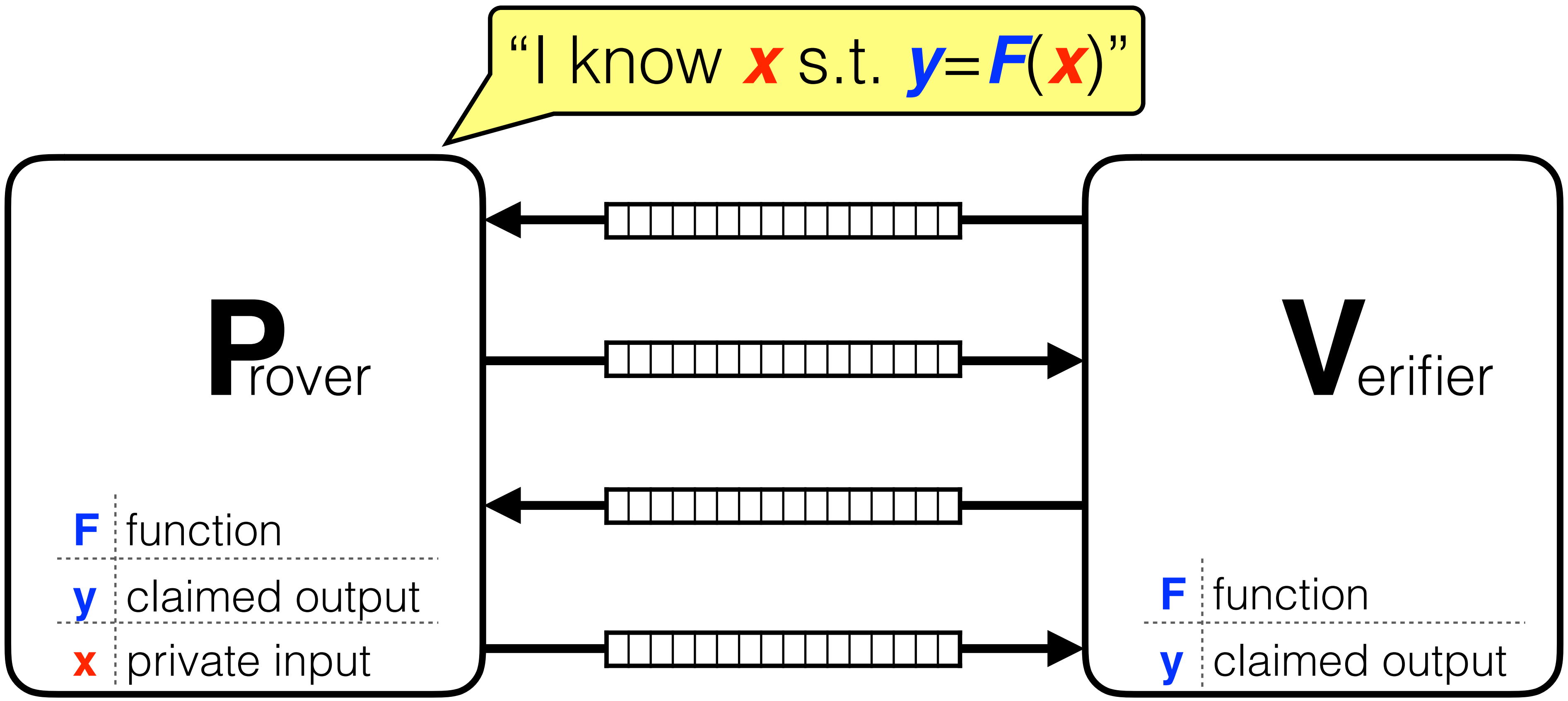
interactive

not succinct

communication complexity & verification complexity are proportional to time(**F**)

Zero Knowledge Proofs

[GMR85]



Powerful cryptographic primitive.

BUT

interactive

not succinct

communication complexity
& verification complexity
are proportional to time(**F**)

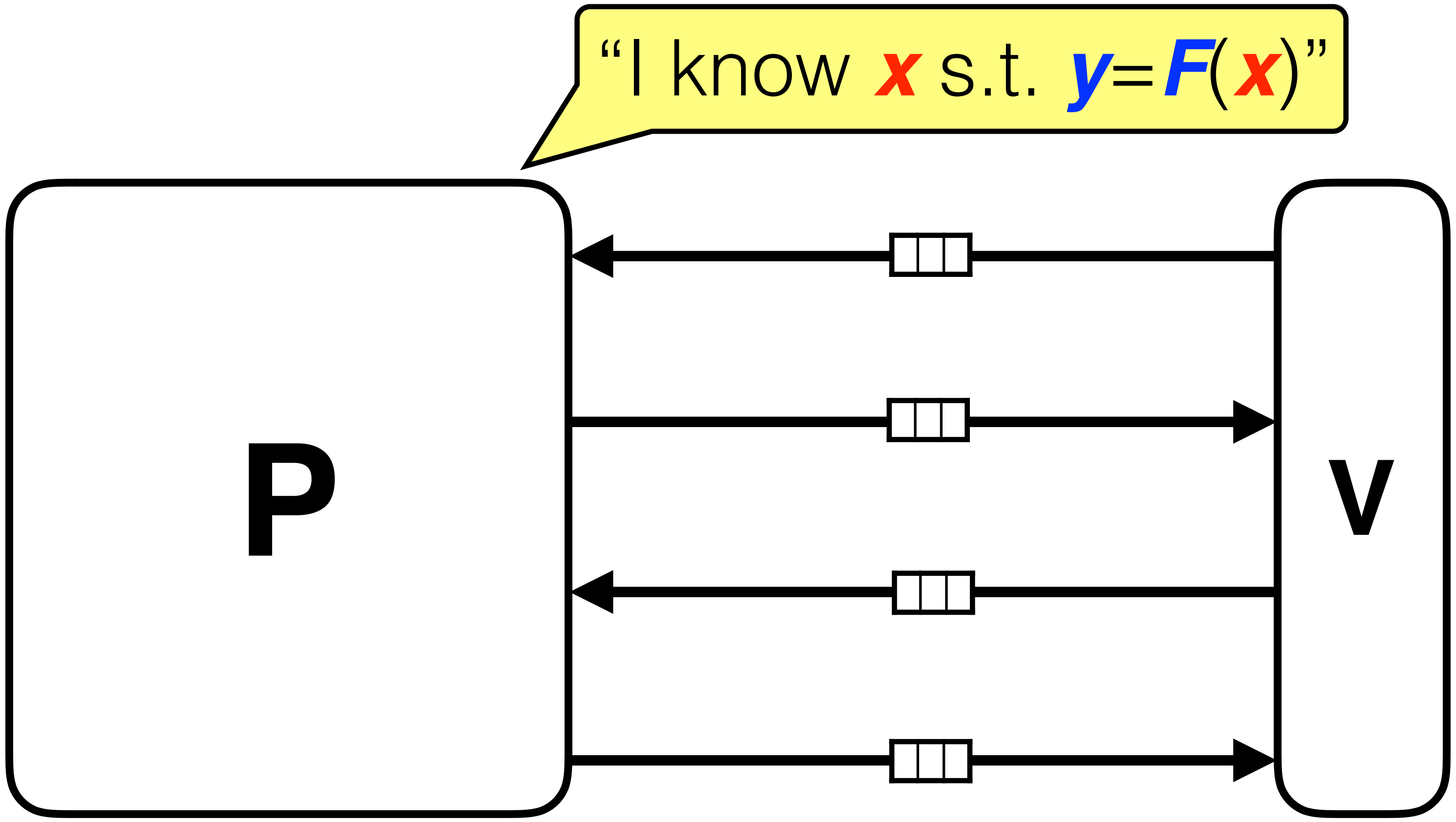
for typical **F**
size(**F**) ≪ time(**F**)

Zero Knowledge **Succinct** Proofs

[Kilian92][Micali94]

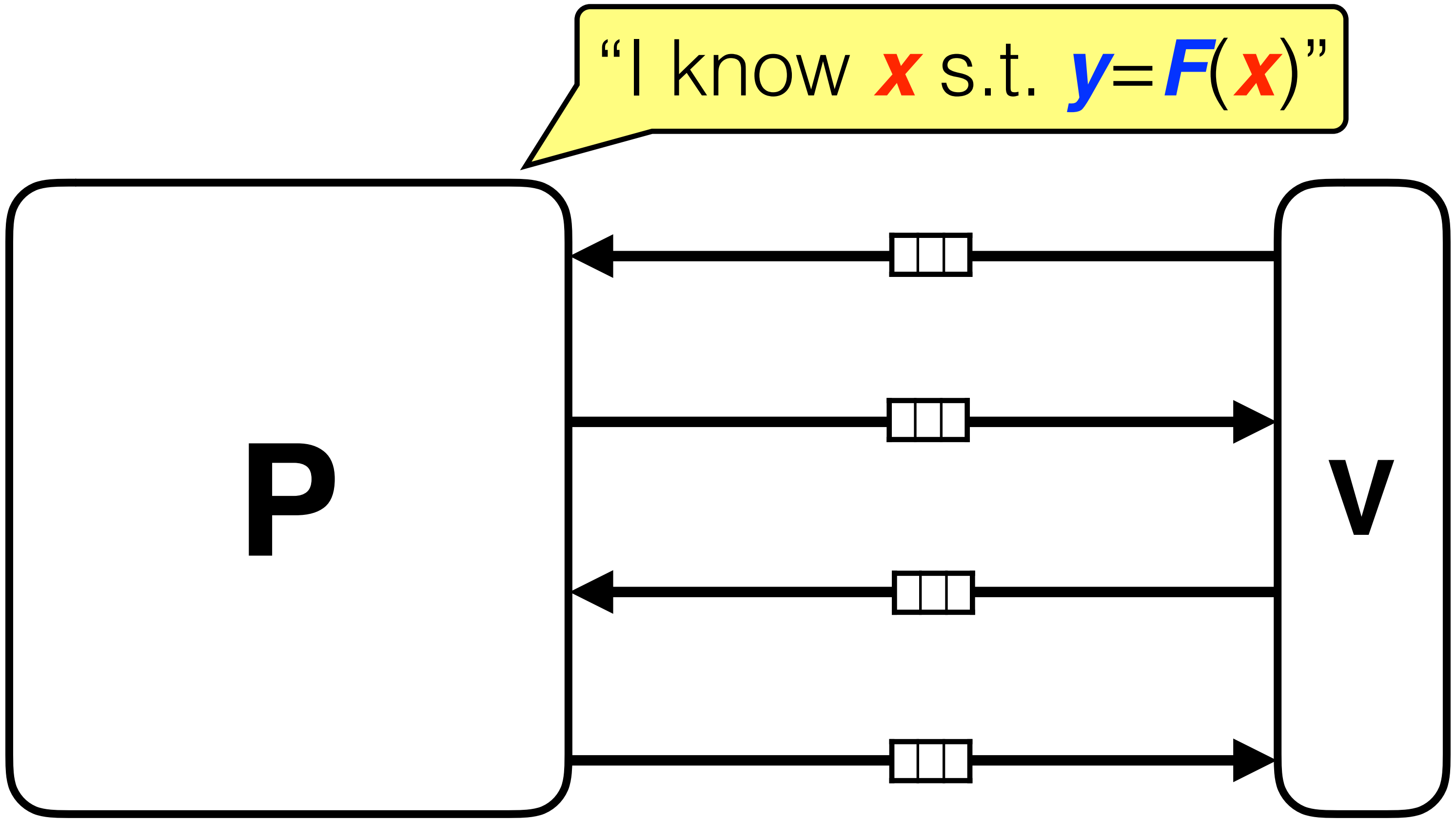
Zero Knowledge **Succinct** Proofs

[Kilian92][Micali94]



Zero Knowledge **Succinct** Proofs

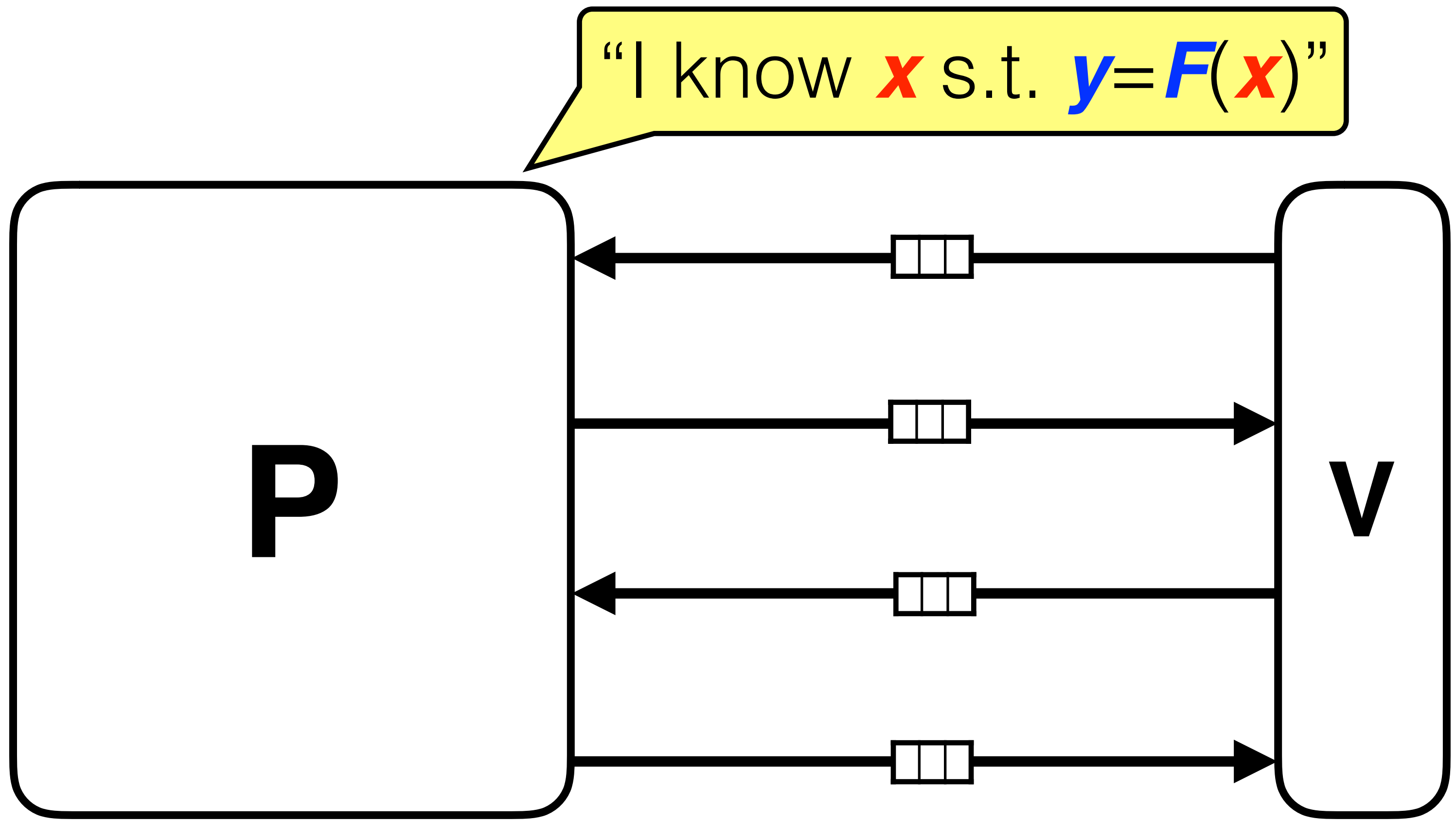
[Kilian92][Micali94]



completeness	$\exists\, x$: $y=F(x) \rightarrow \Pr[\mathbf{P}(F,y,x)$ convinces $\mathbf{V}(F,y)]=1$
soundness	$\nexists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{P'}\ \Pr[\mathbf{P'}$ convinces $\mathbf{V}(F,y)]\approx 0$
zero knowledge	$\exists\, x$: $y=F(x) \rightarrow \forall\, \mathbf{V'}$, $\mathbf{S}(\mathbf{V'},F,y) \simeq$ view of $\mathbf{V'}$ with $\mathbf{P}(F,y,x)$
succinctness	$\mathbf{V}(F,y)$ runs in time proportional to $\lvert F\rvert+\lvert y\rvert$ (not time$(F)+\lvert y\rvert$)

Zero Knowledge **Succinct** Proofs

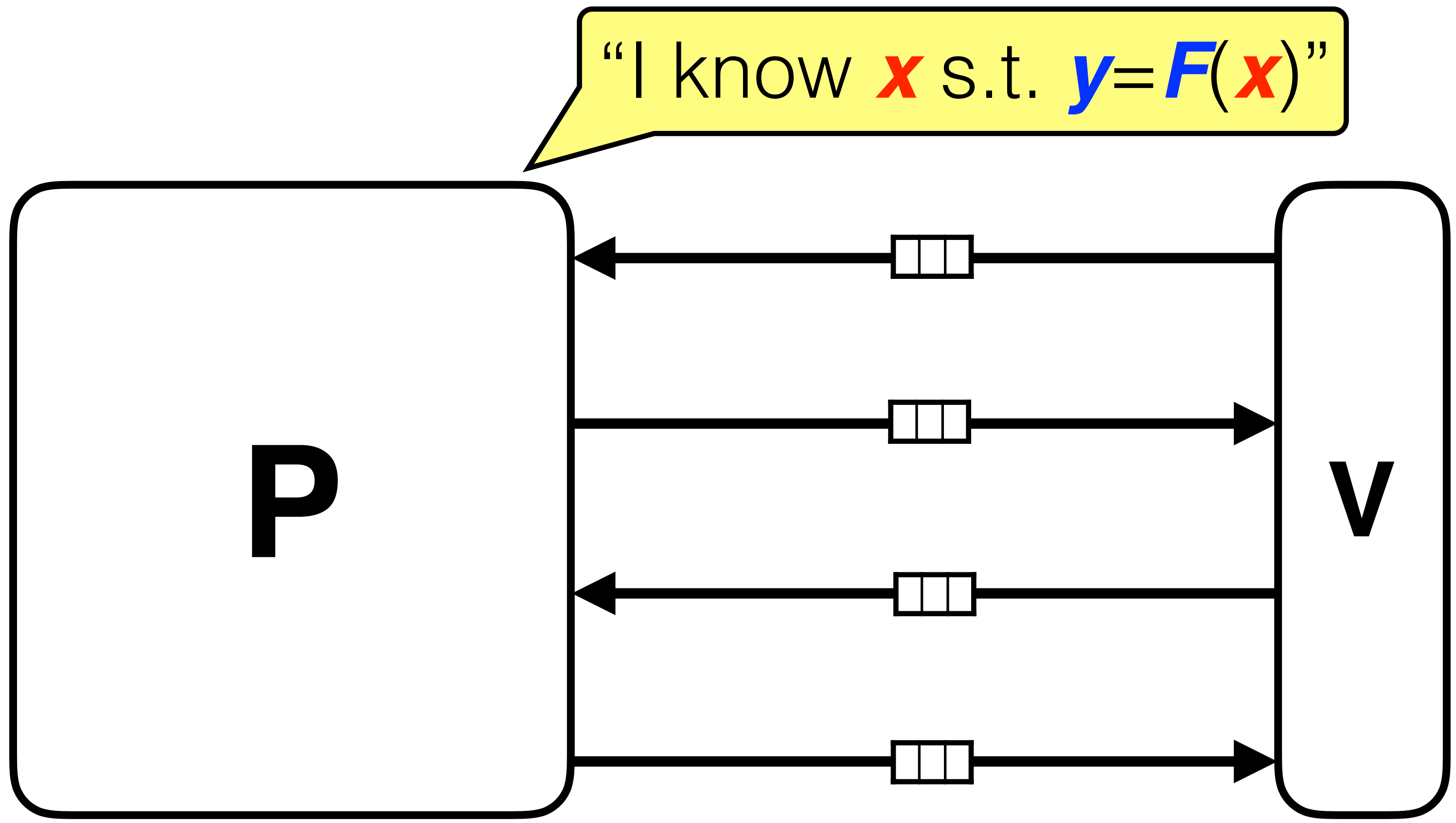
[Kilian92][Micali94]



completeness	$\exists\ x\colon\ y=F(x) \rightarrow \Pr[\mathbf{P}(F,y,x)\text{ convinces }\mathbf{V}(F,y)]=1$
soundness*	$\nexists\ x\colon\ y=F(x) \rightarrow \forall\ \mathbf{P'}\ \Pr[\mathbf{P'}\text{ convinces }\mathbf{V}(F,y)]\approx 0$
zero knowledge	$\exists\ x\colon\ y=F(x) \rightarrow \forall\ \mathbf{V'},\ \mathbf{S}(\mathbf{V'},F,y) \simeq$ view of $\mathbf{V'}$ with $\mathbf{P}(F,y,x)$
succinctness	$\mathbf{V}(F,y)$ runs in time proportional to $\lvert F\rvert+\lvert y\rvert$ (not time$(F)+\lvert y\rvert$)

Zero Knowledge **Succinct** Proofs

[Kilian92][Micali94]

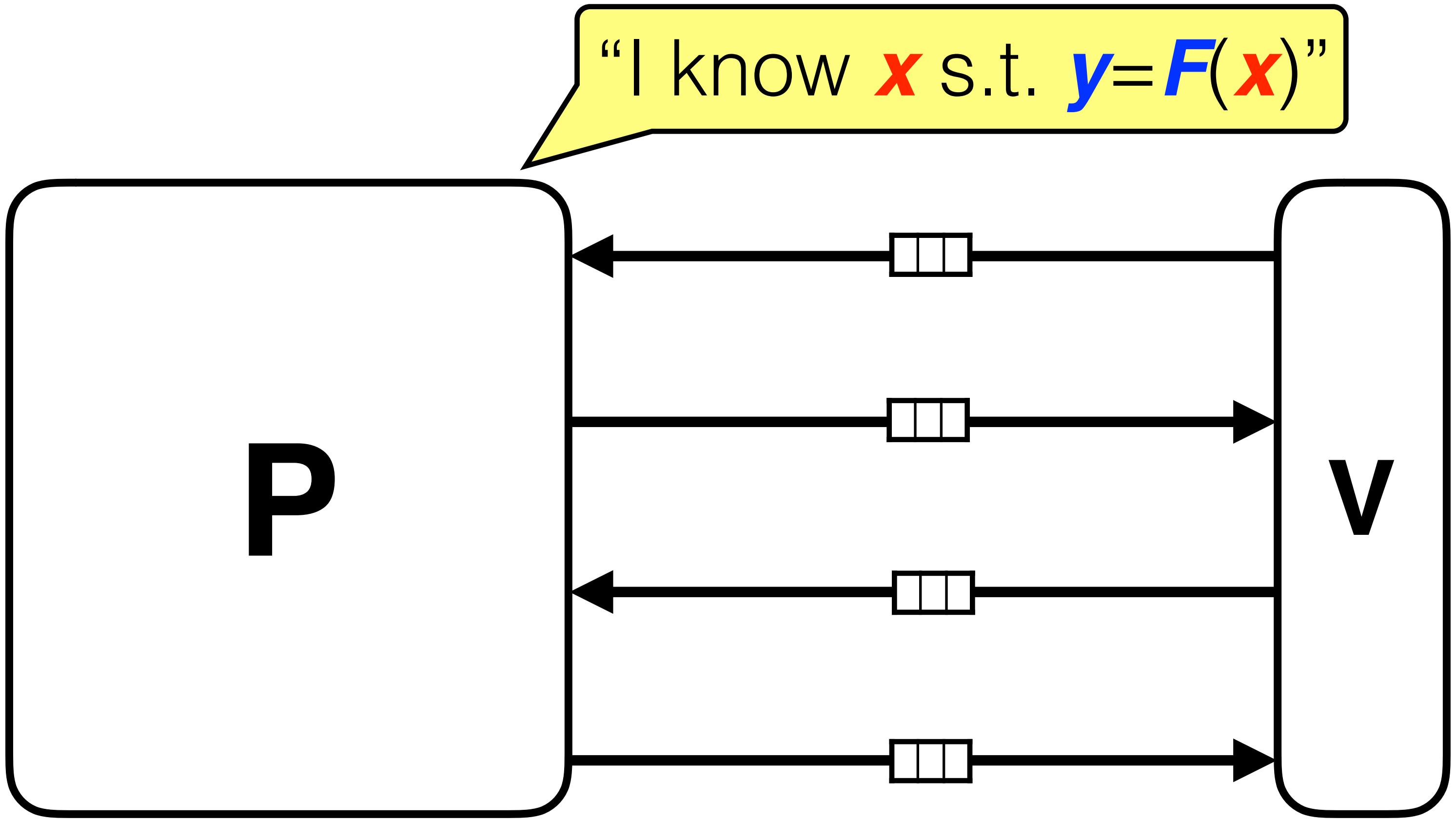


completeness	$\exists\, x$: $y=F(x)$ → Pr[**P**(F,y,x) convinces **V**(F,y)]=1
soundness*	$\nexists\, x$: $y=F(x)$ → $\forall$ **P'** Pr[**P'** convinces **V**(F,y)]≈0
zero knowledge	$\exists\, x$: $y=F(x)$ → $\forall$ **V'**, **S**(**V'**,F,y) ≃ view of **V'** with **P**(F,y,x)
succinctness	**V**(F,y) runs in time proportional to $\lvert F\rvert+\lvert y\rvert$ (not time(F)+$\lvert y\rvert$)

* must relax to *computational* soundness: $\forall$ PPT **P'** … [GH98]

Zero Knowledge **Succinct** ~~Proofs~~ Arguments

[Kilian92][Micali94]



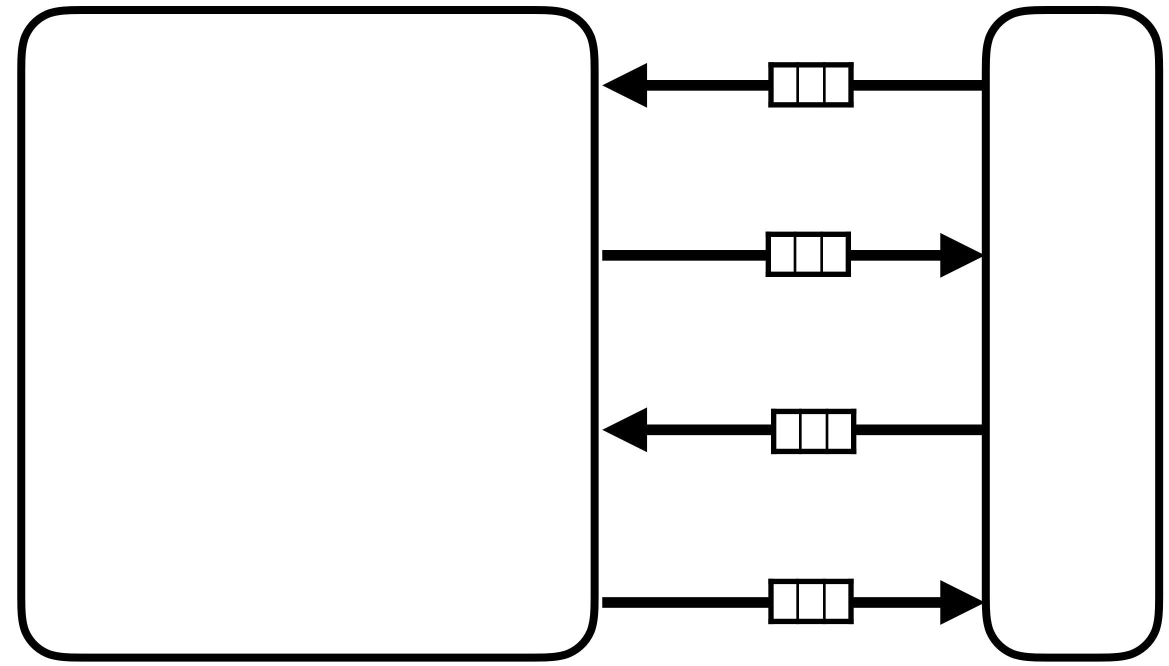
completeness	$\exists\, x\!: y=F(x) \rightarrow \Pr[\mathbf{P}(F,y,x)$ convinces $\mathbf{V}(F,y)]=1$
soundness*	$\nexists\, x\!: y=F(x) \rightarrow \forall\, \mathbf{P'}\ \Pr[\mathbf{P'}$ convinces $\mathbf{V}(F,y)]\approx 0$
zero knowledge	$\exists\, x\!: y=F(x) \rightarrow \forall\, \mathbf{V'},\ \mathbf{S}(\mathbf{V'},F,y) \simeq$ view of $\mathbf{V'}$ with $\mathbf{P}(F,y,x)$
succinctness	$\mathbf{V}(F,y)$ runs in time proportional to $\lvert F\rvert+\lvert y\rvert$ (not time$(F)+\lvert y\rvert$)

* must relax to *computational* soundness: ∀ PPT **P'** … [GH98]

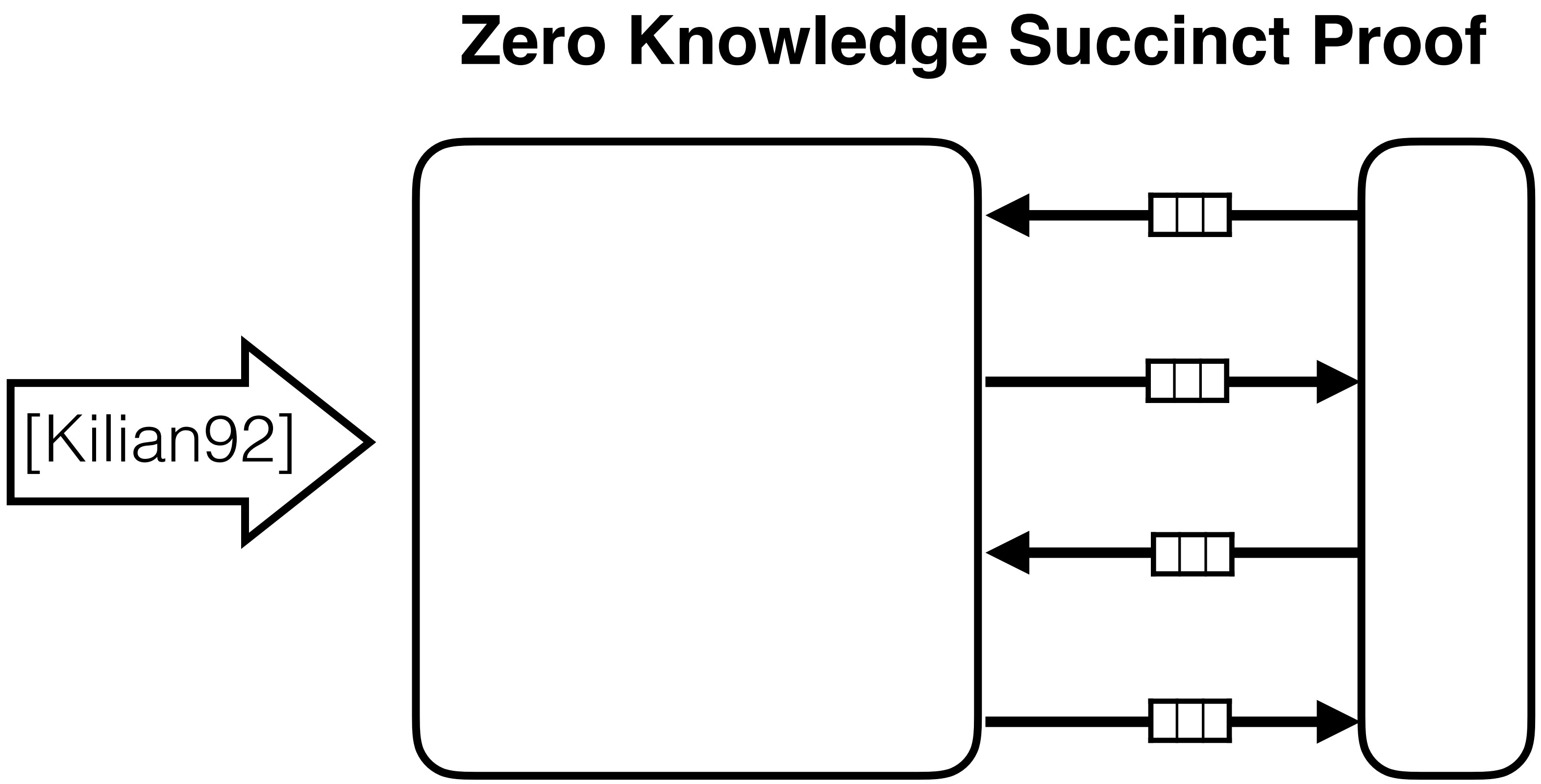
Achieving Succinctness

Achieving Succinctness

Zero Knowledge Succinct Proof



Achieving Succinctness

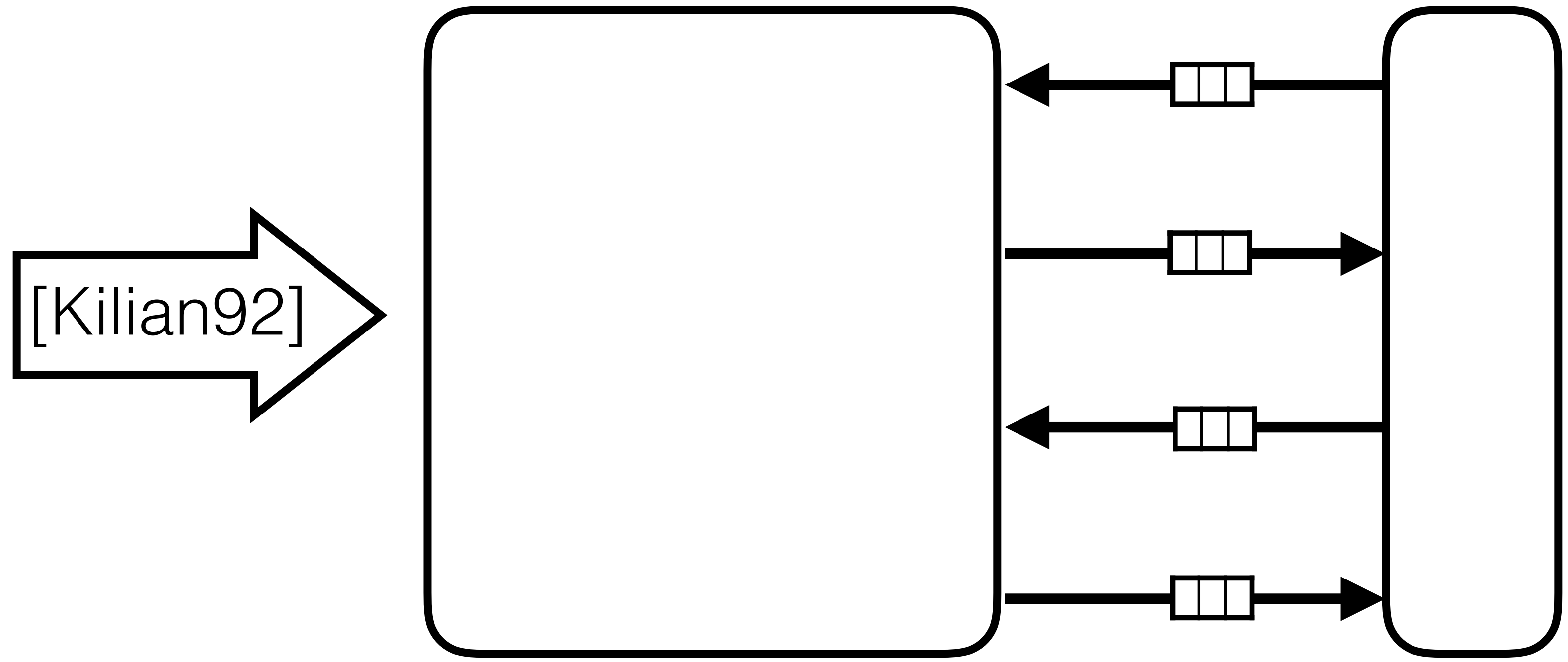


Achieving Succinctness

Probabilistically Checkable Proof

[BFLS91][FGLSS96][AS92][ALMSS92]

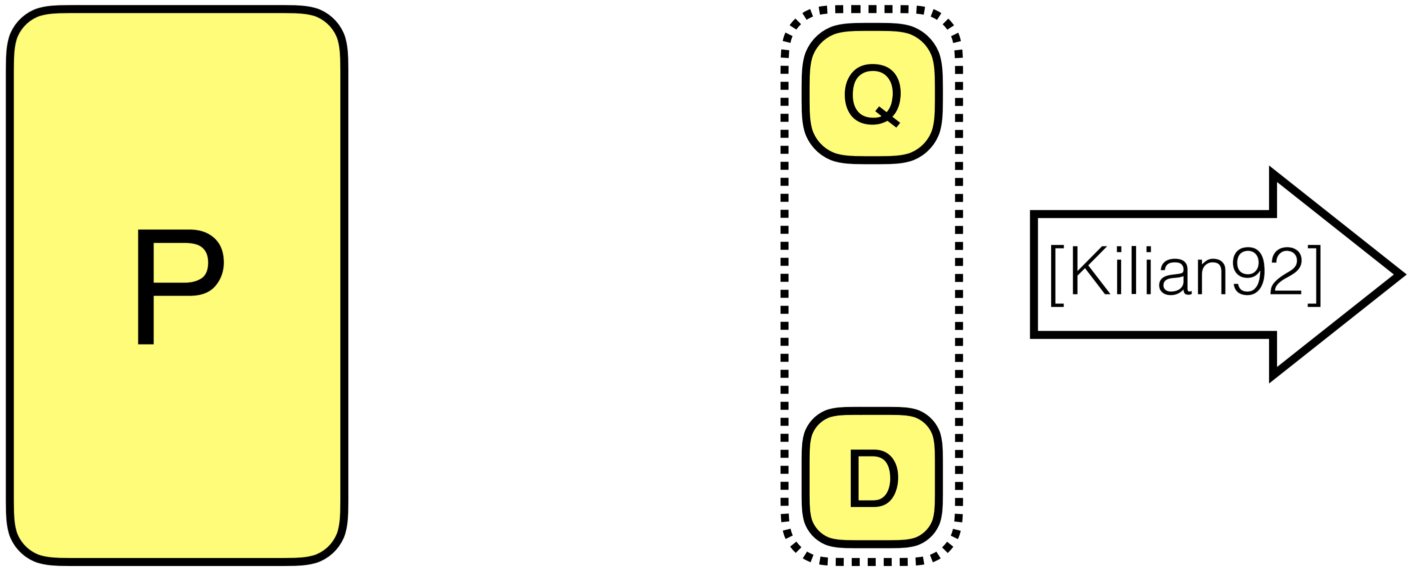
Zero Knowledge Succinct Proof



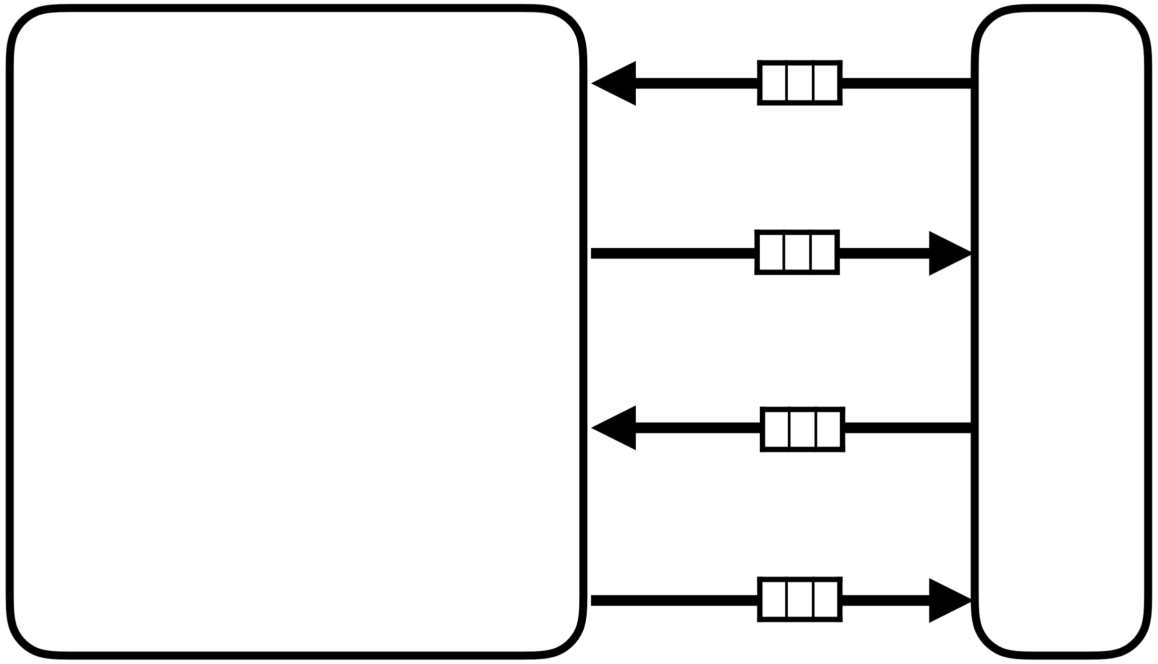
Achieving Succinctness

Probabilistically Checkable Proof

[BFLS91][FGLSS96][AS92][ALMSS92]



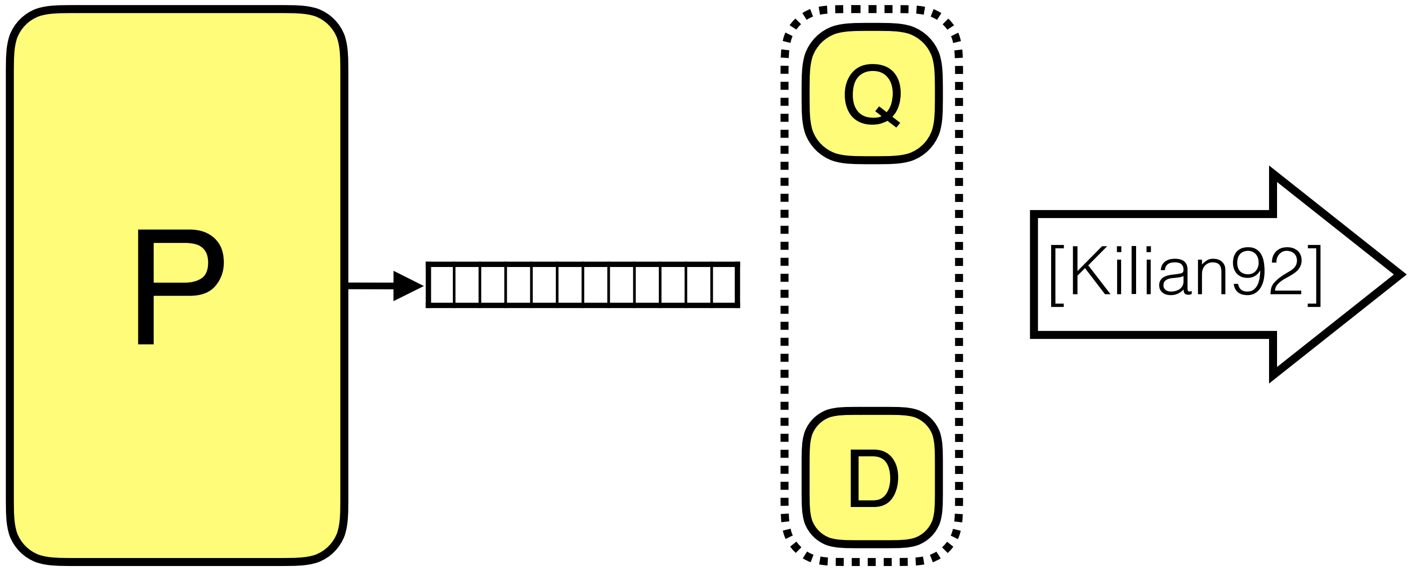
Zero Knowledge Succinct Proof



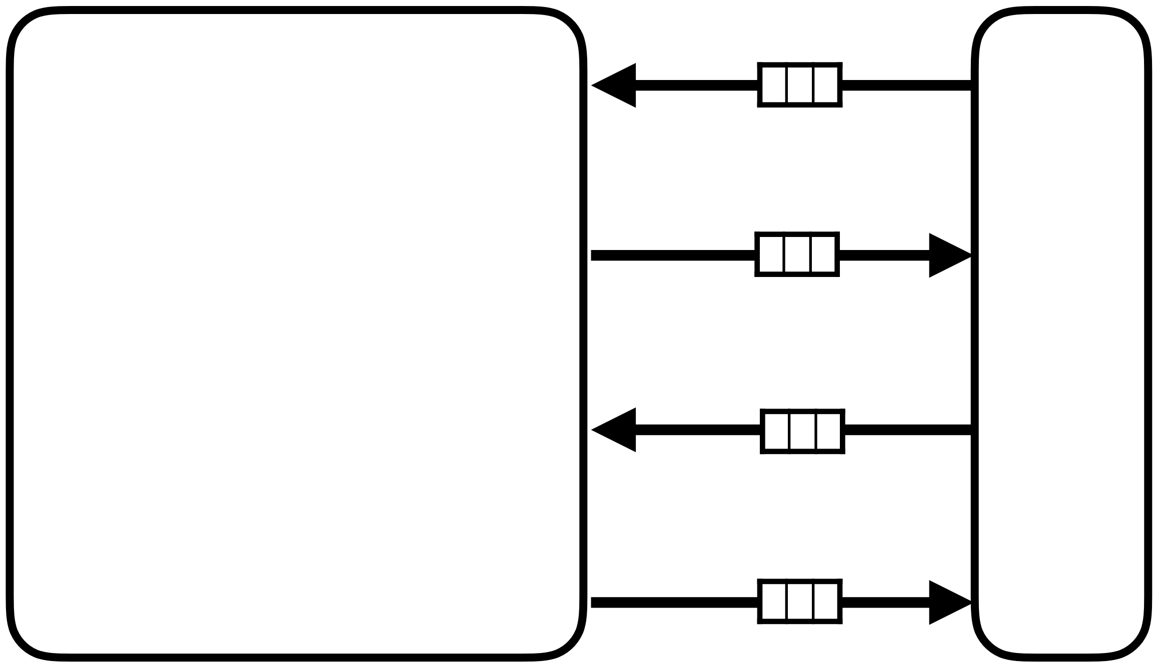
Achieving Succinctness

Probabilistically Checkable Proof

[BFLS91][FGLSS96][AS92][ALMSS92]



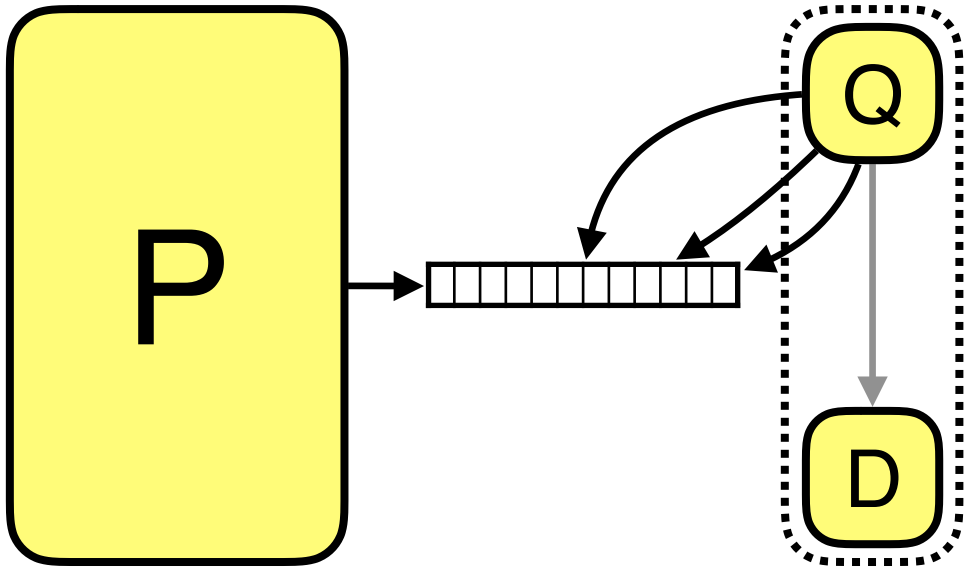
Zero Knowledge Succinct Proof



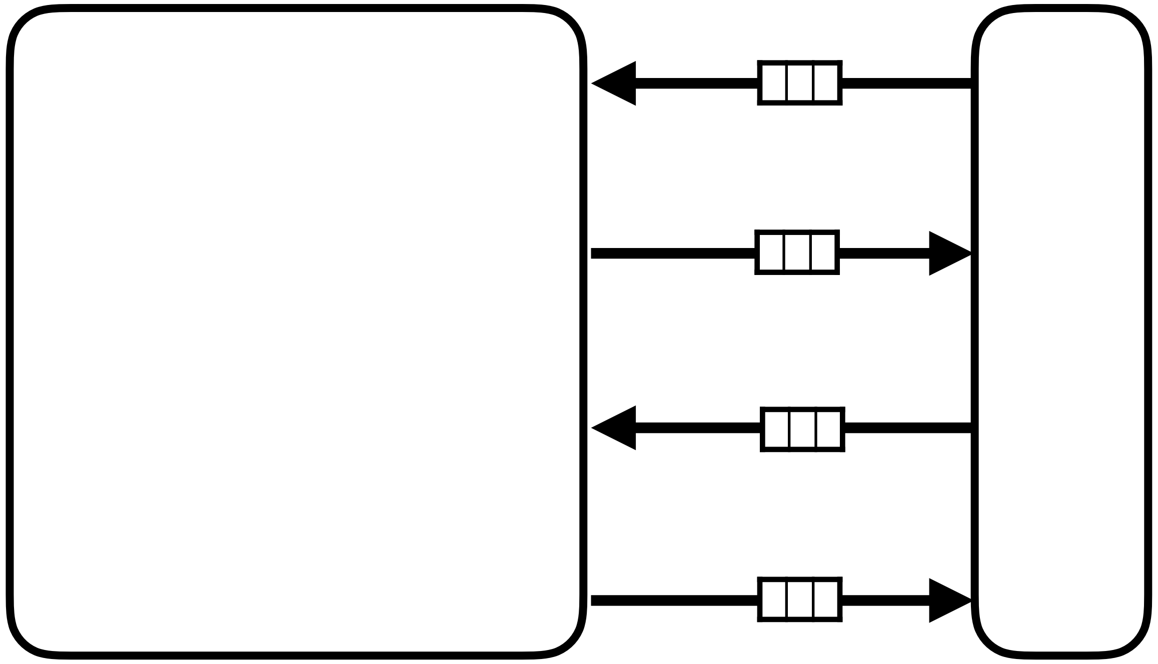
Achieving Succinctness

Probabilistically Checkable Proof

[BFLS91][FGLSS96][AS92][ALMSS92]



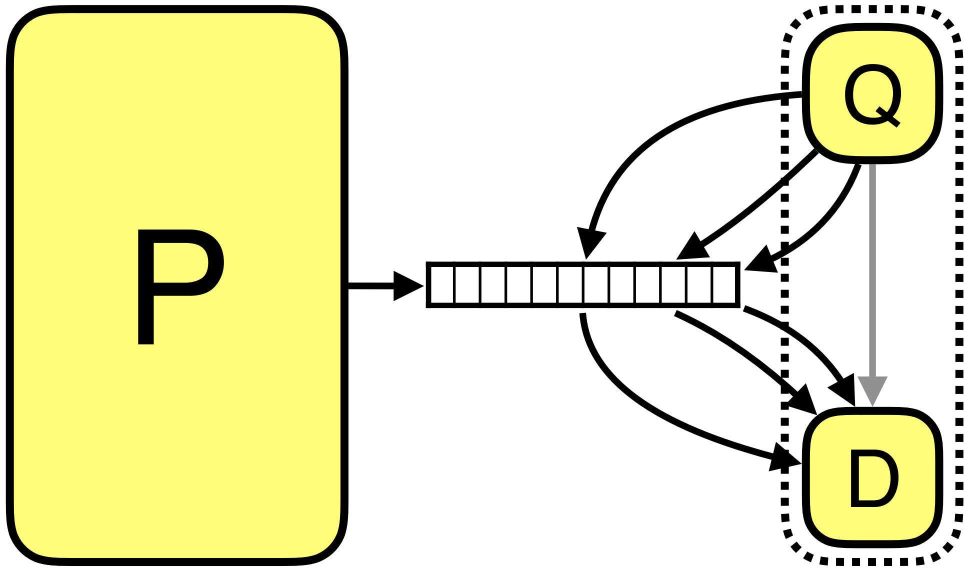
Zero Knowledge Succinct Proof



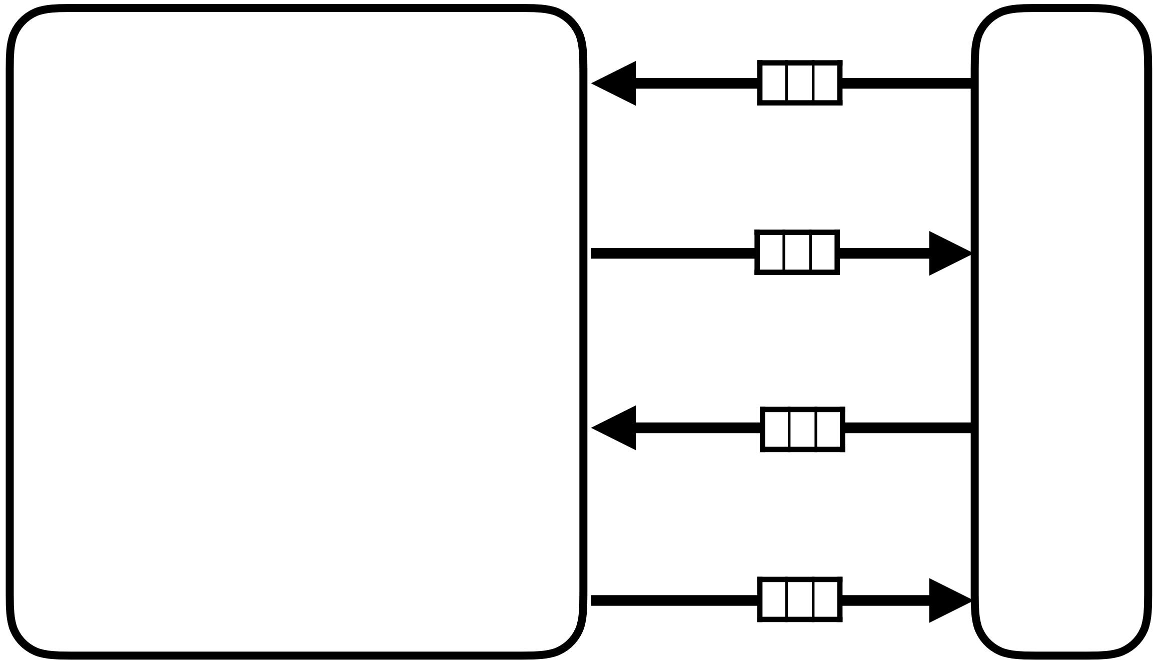
Achieving Succinctness

Probabilistically Checkable Proof

[BFLS91][FGLSS96][AS92][ALMSS92]

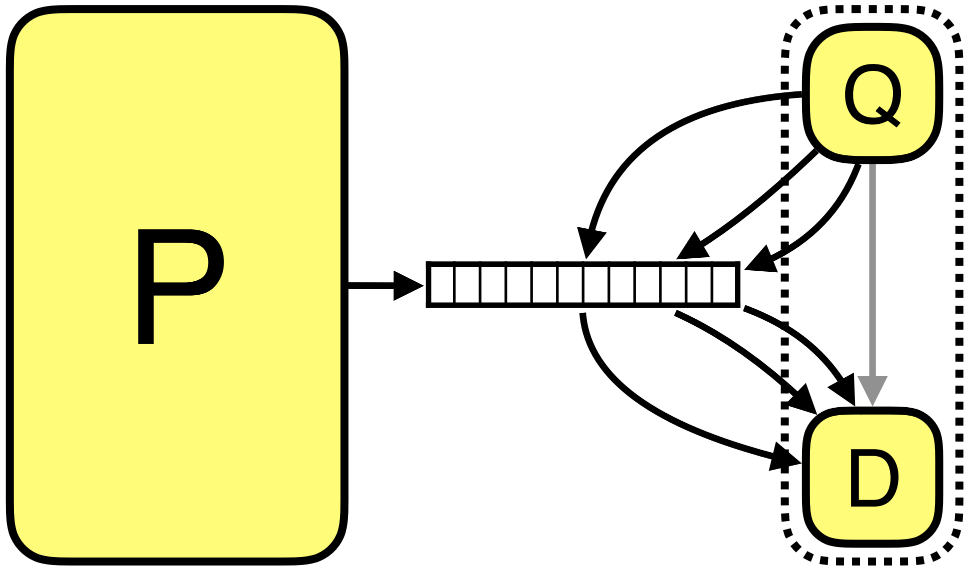


Zero Knowledge Succinct Proof

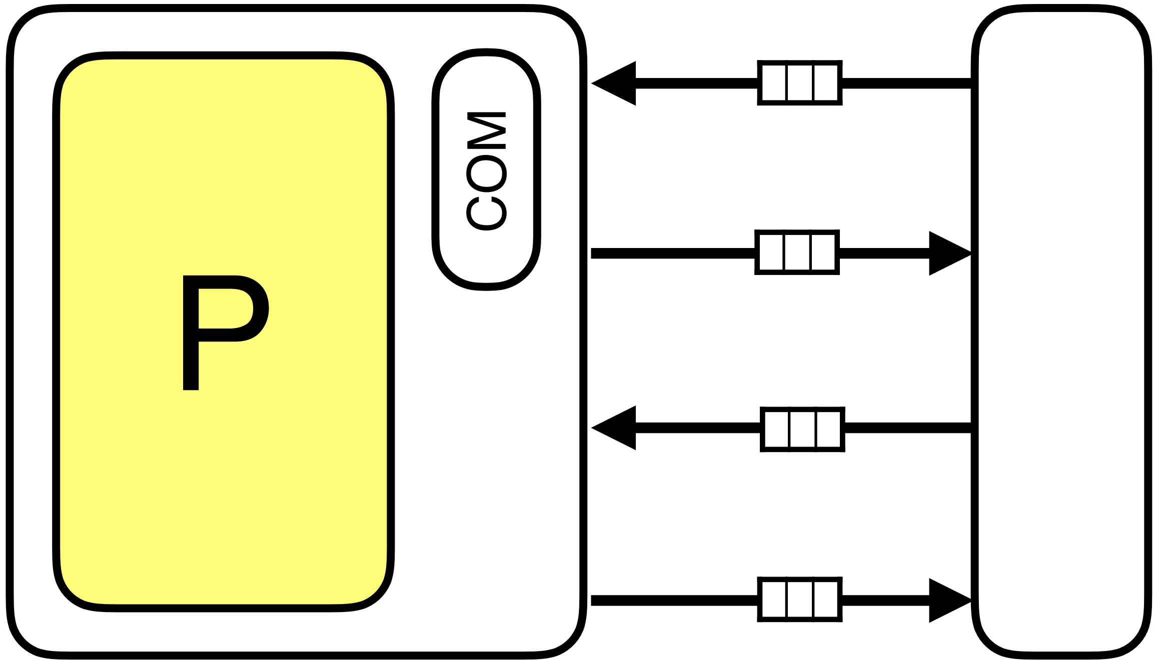


Achieving Succinctness

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]



Zero Knowledge Succinct Proof

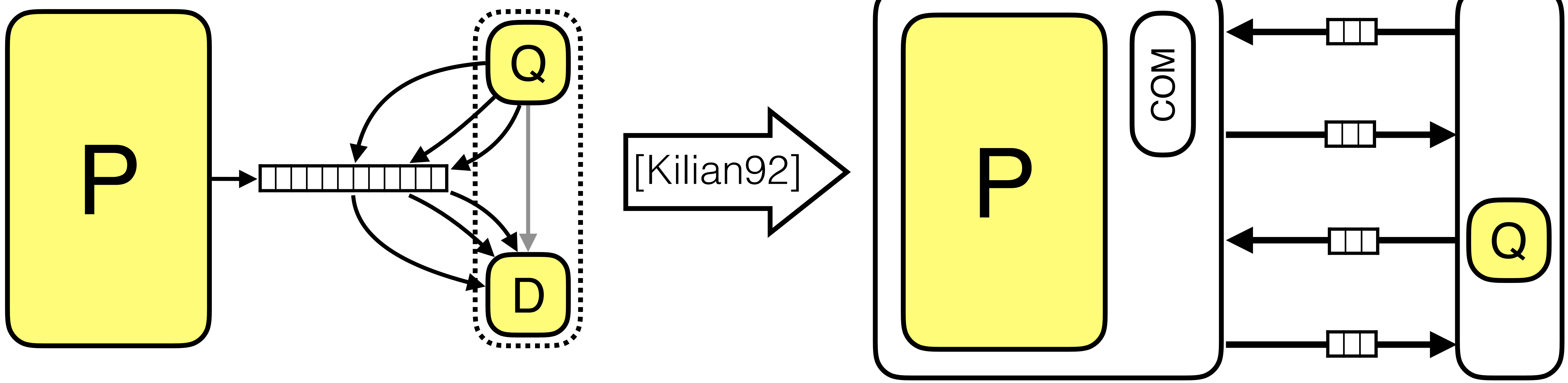


Achieving Succinctness

Probabilistically Checkable Proof

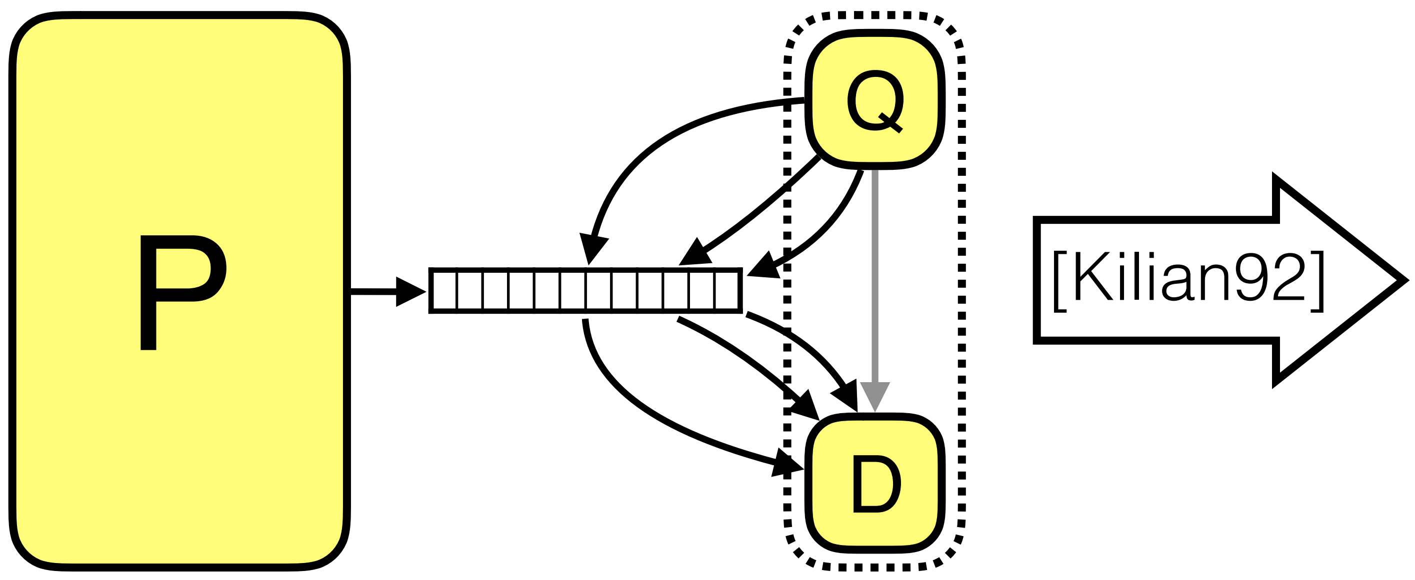
[BFLS91][FGLSS96][AS92][ALMSS92]

Zero Knowledge Succinct Proof

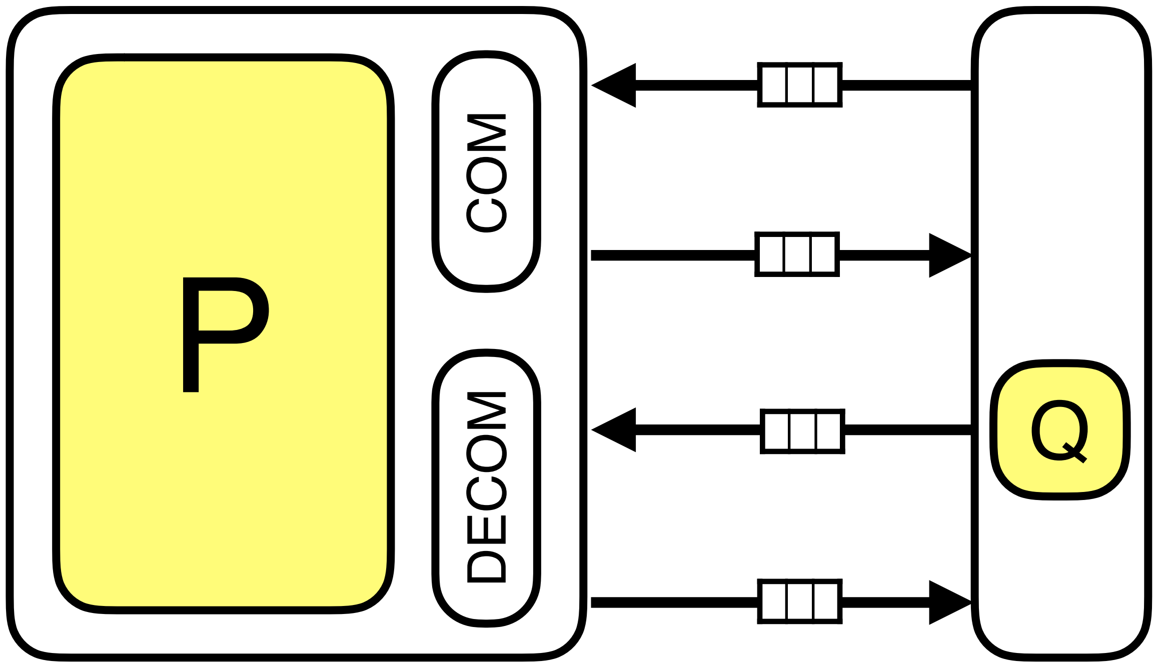


Achieving Succinctness

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]



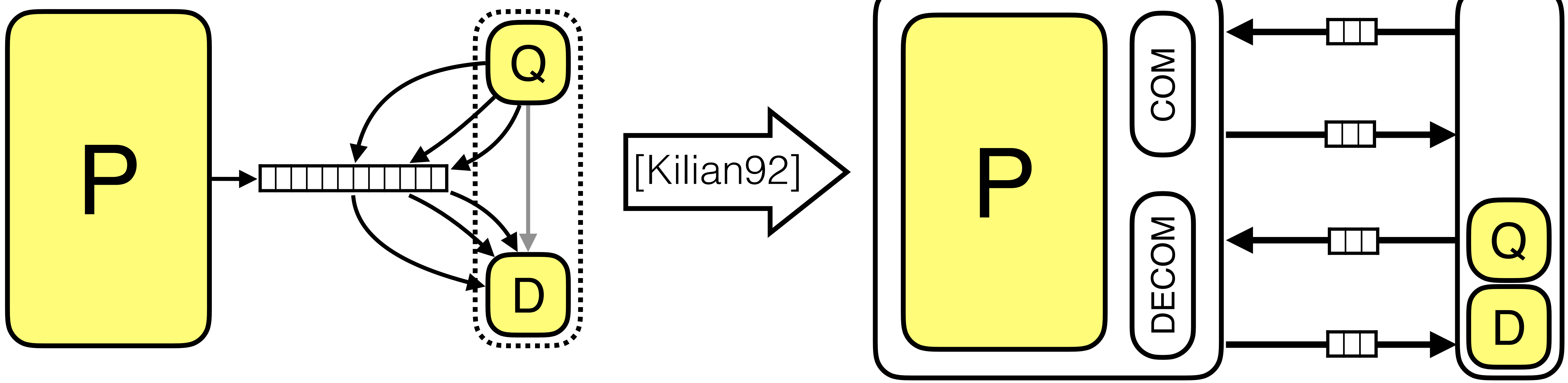
Zero Knowledge Succinct Proof



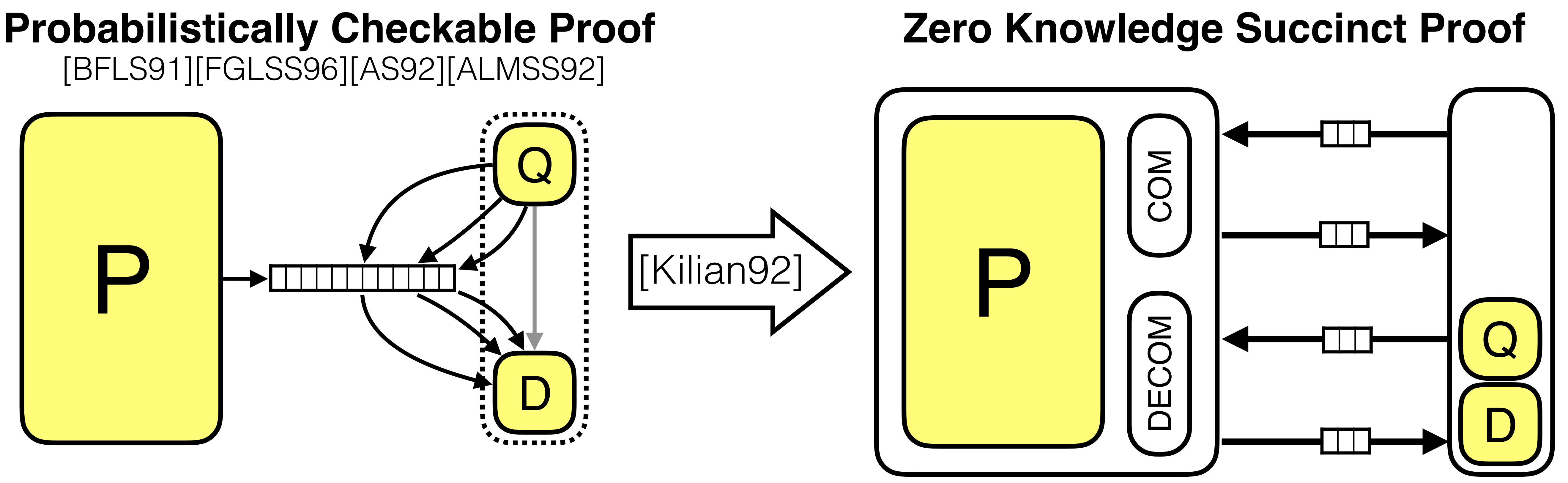
Achieving Succinctness

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

Zero Knowledge Succinct Proof



Achieving Succinctness



TOFIX

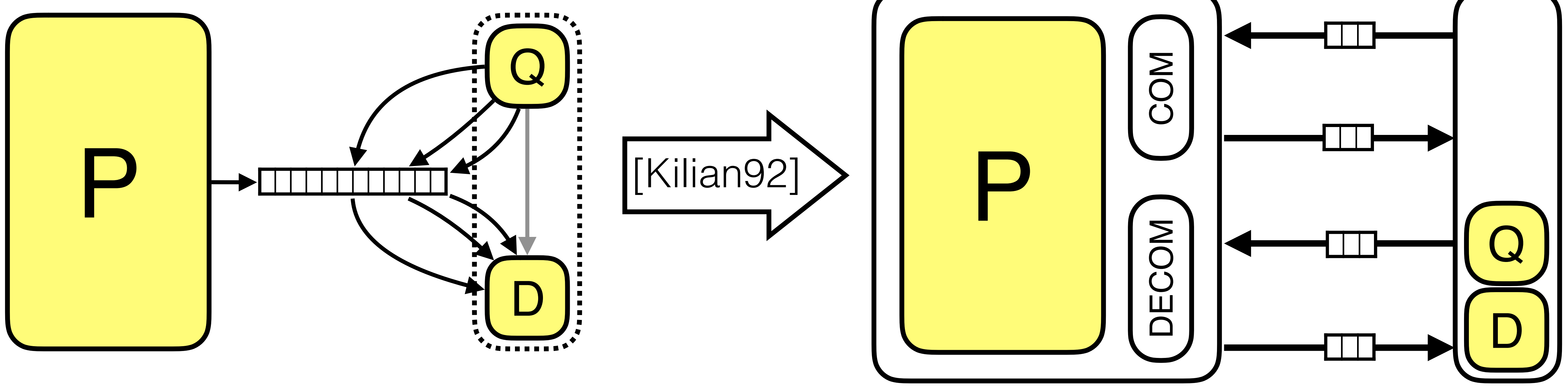
interactive

not succinct

Achieving Succinctness

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

Zero Knowledge Succinct Proof



TOFIX

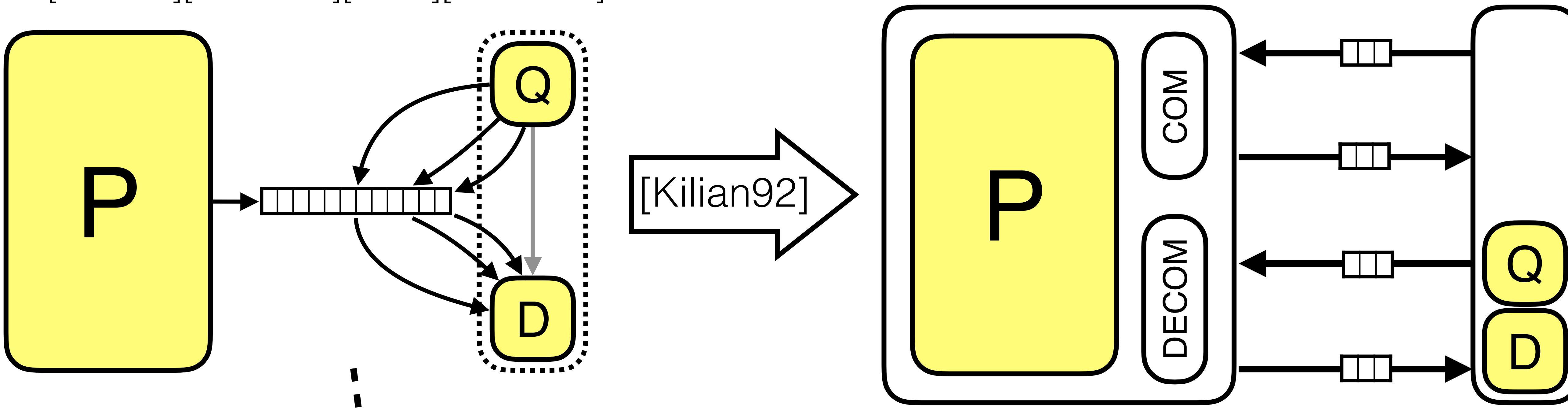
interactive

~~**not succinct**~~

Achieving Succinctness

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

Zero Knowledge Succinct Proof



TOFIX

interactive

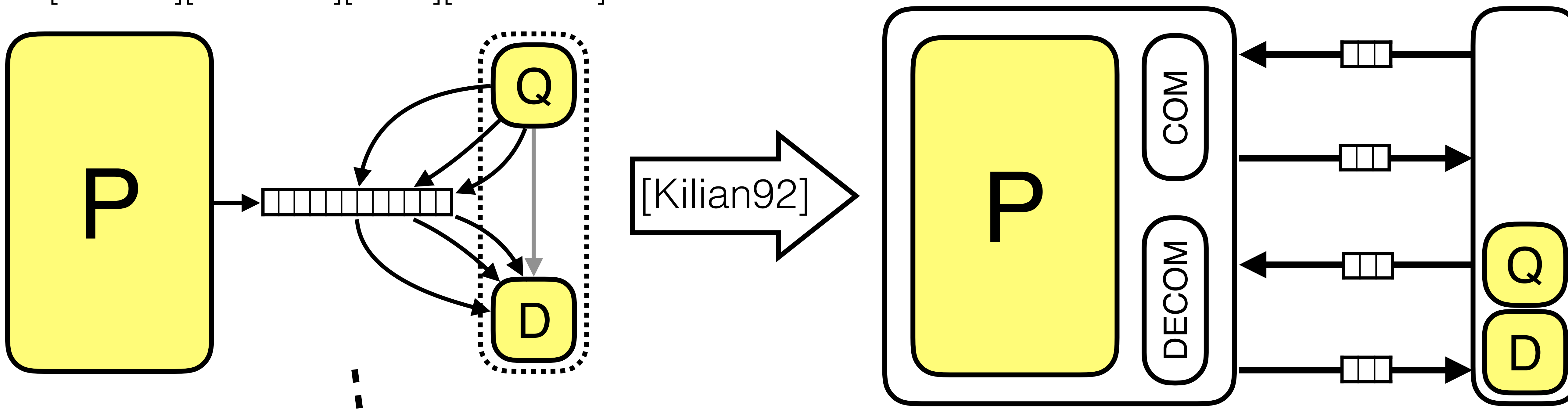
~~**not succinct**~~

bad concrete efficiency

Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

Zero Knowledge Succinct Proof



TOFIX

interactive

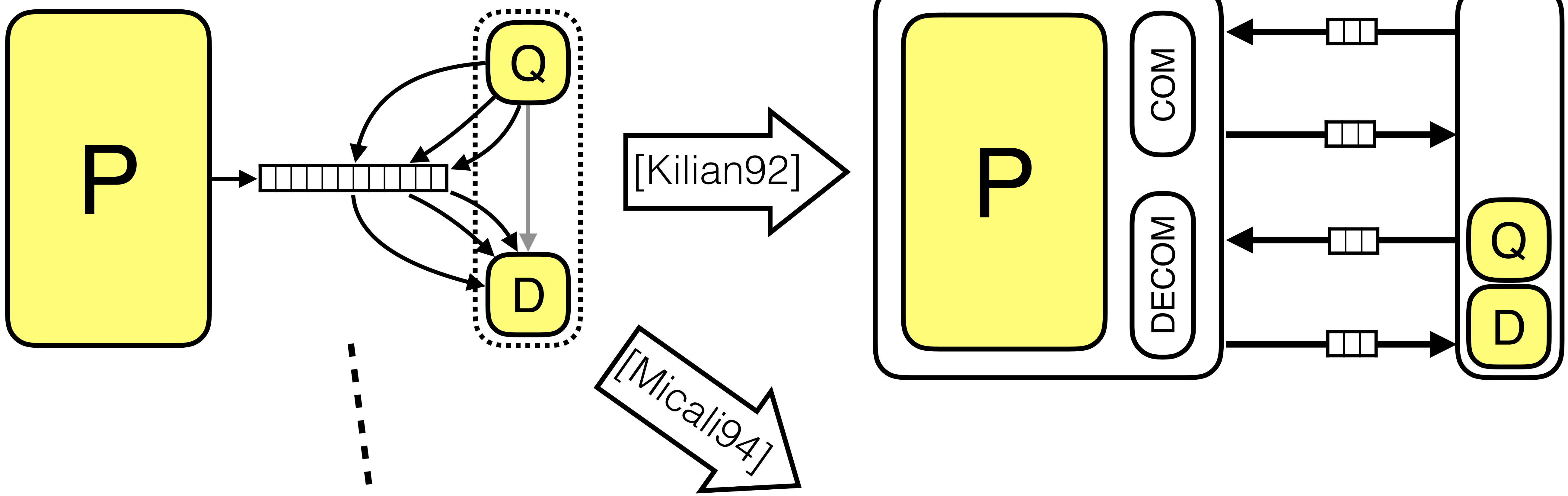
~~not succinct~~

bad concrete efficiency

Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

Zero Knowledge Succinct Proof



TOFIX

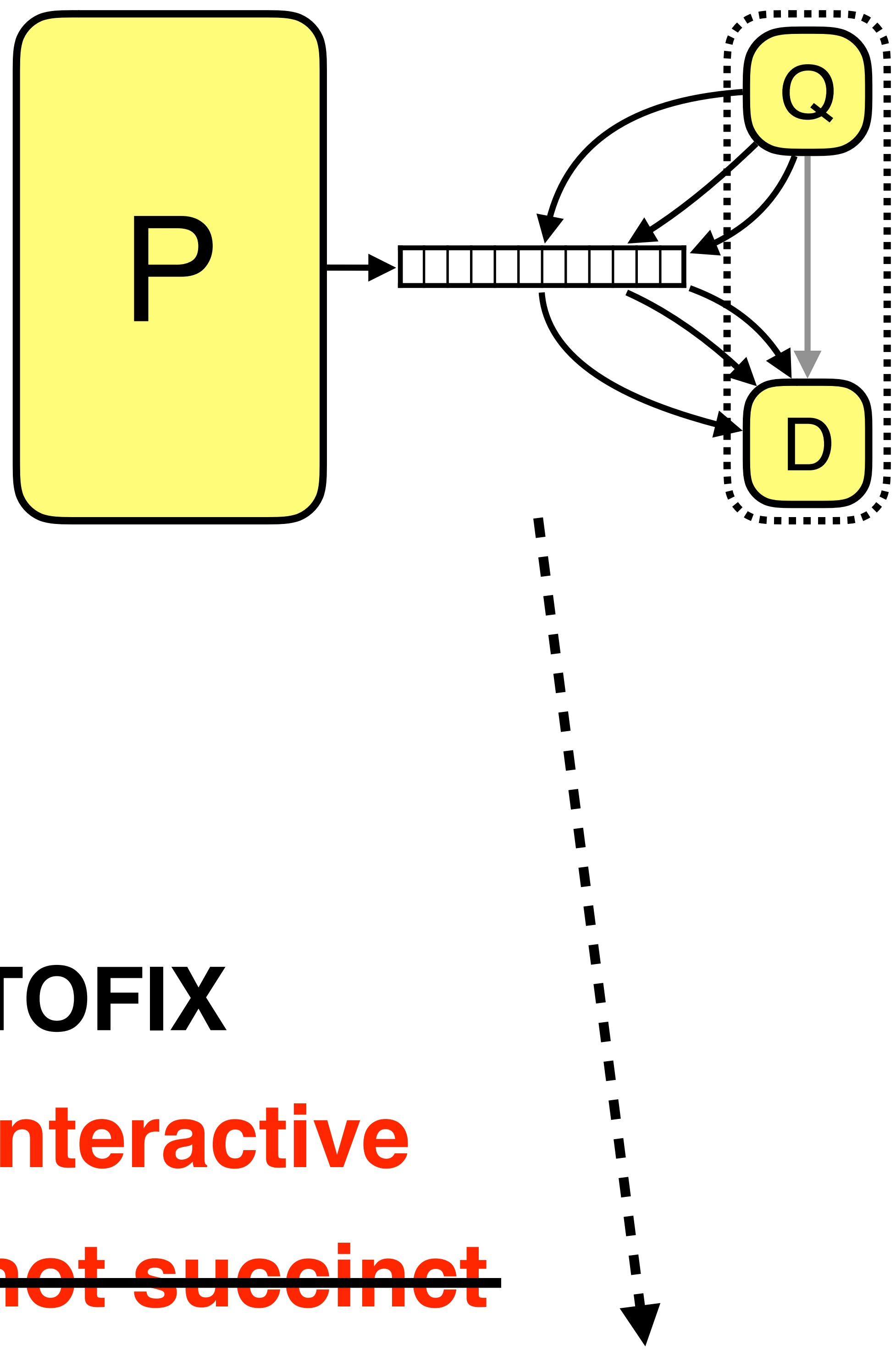
interactive

~~**not succinct**~~

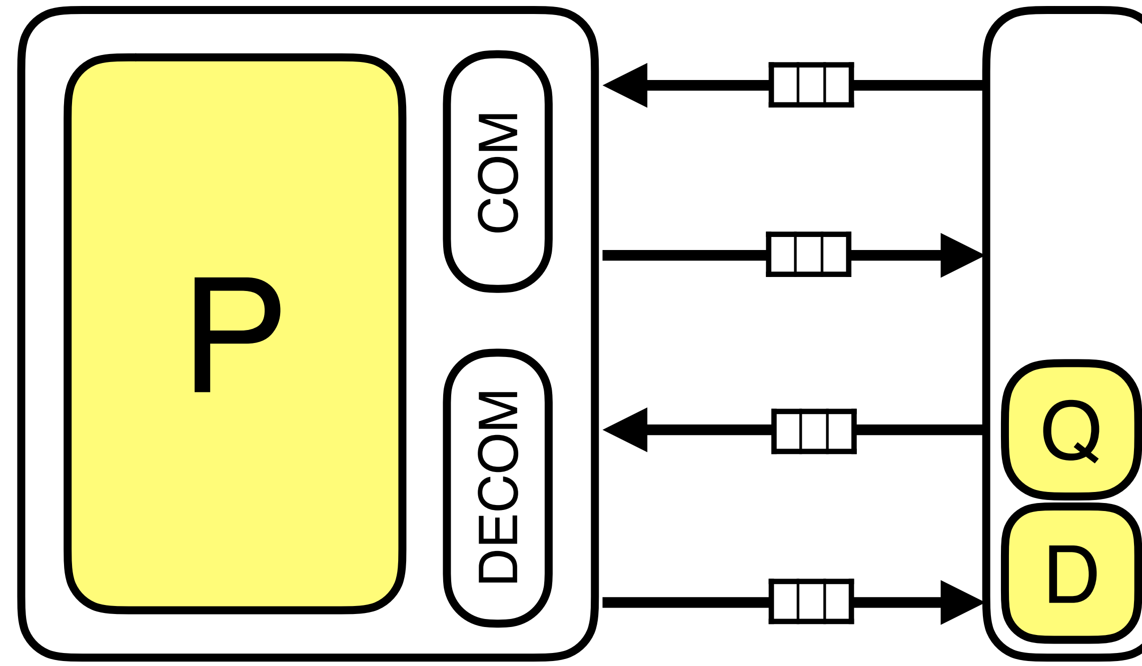
bad concrete efficiency

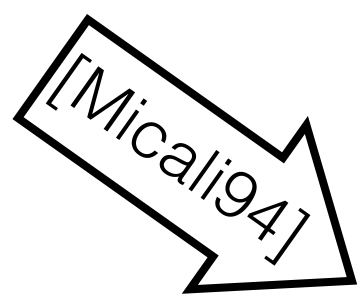
Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

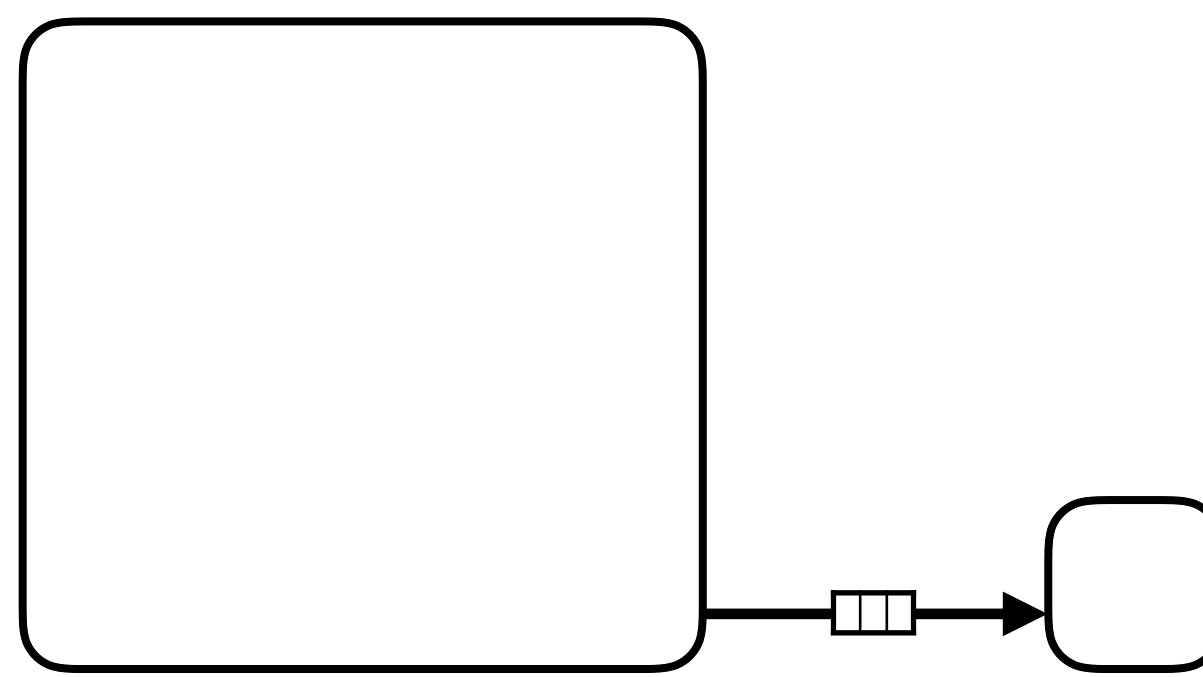


Zero Knowledge Succinct Proof





(the first)
Zero Knowledge SNARK



TOFIX

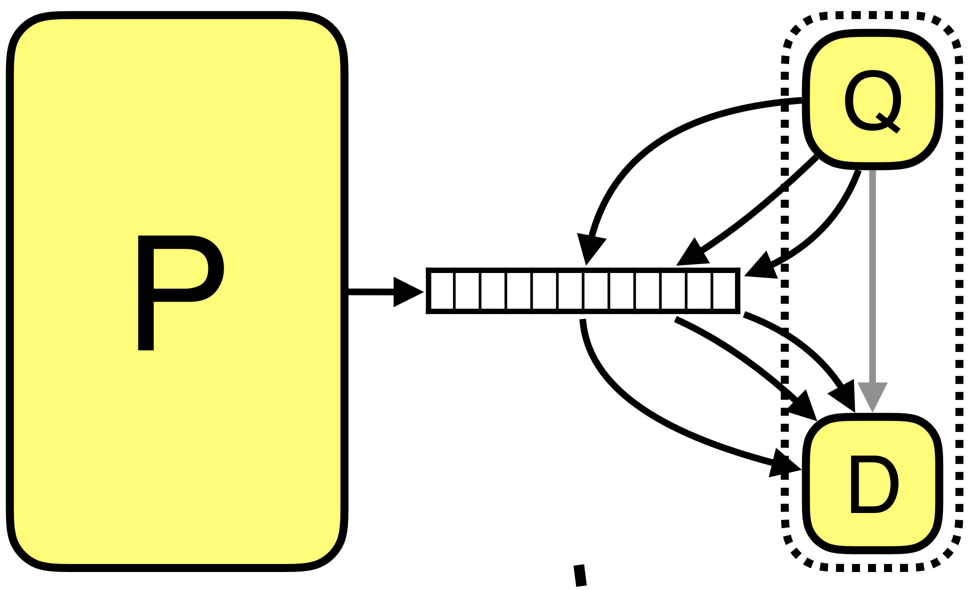
interactive

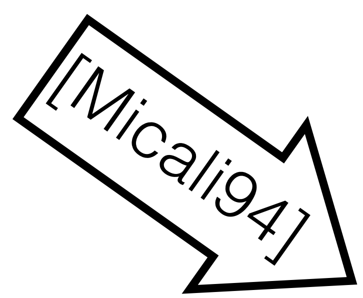
~~**not succinct**~~

bad concrete efficiency

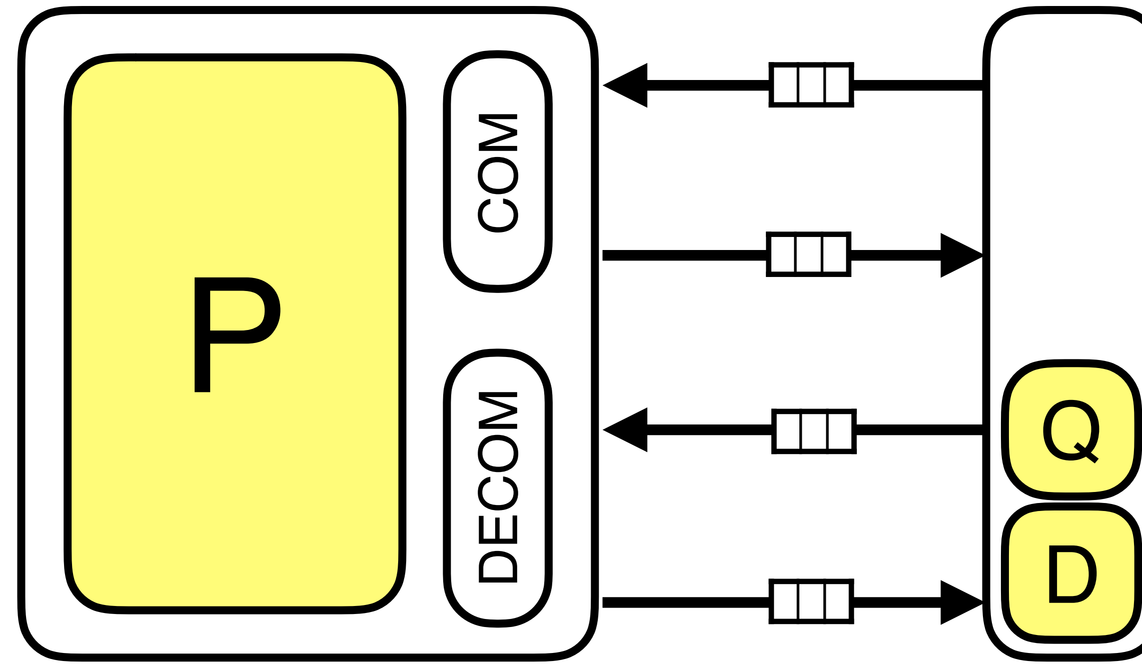
Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

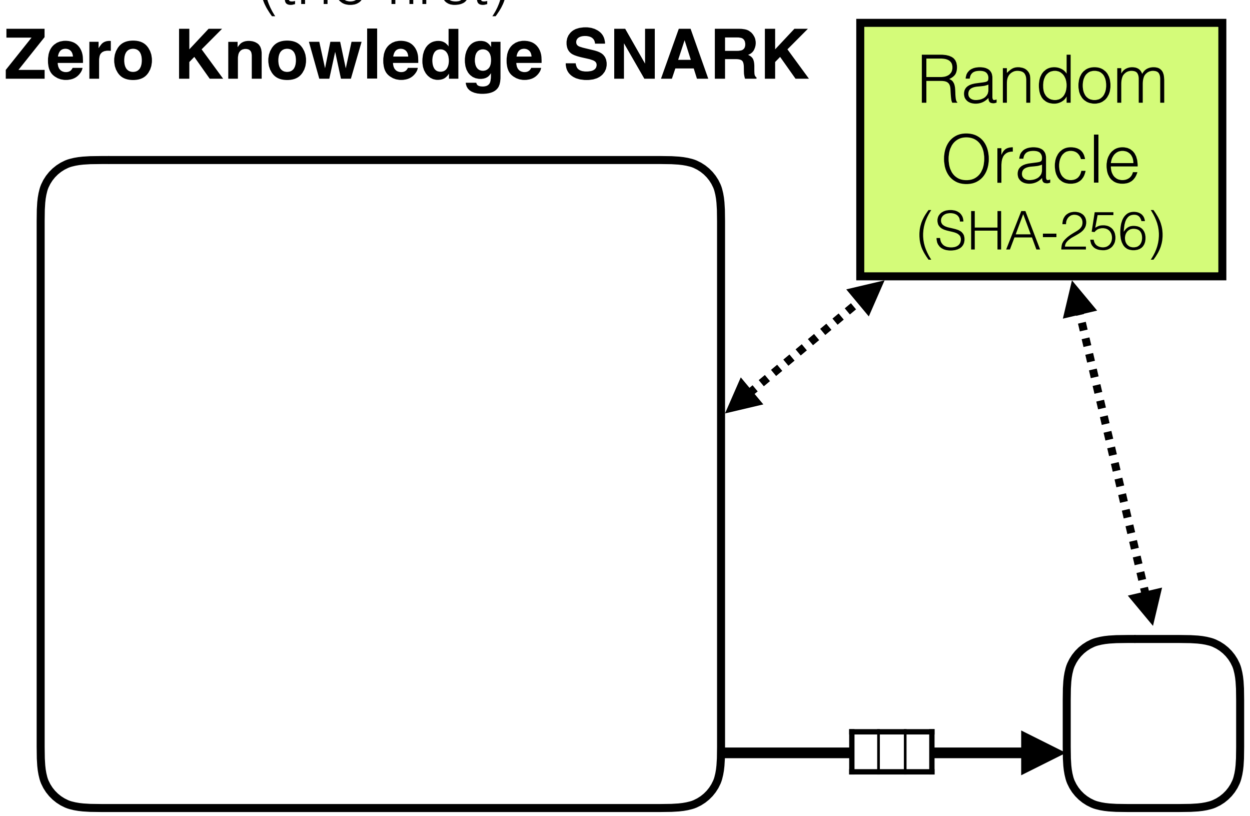




Zero Knowledge Succinct Proof



(the first)
Zero Knowledge SNARK



TOFIX

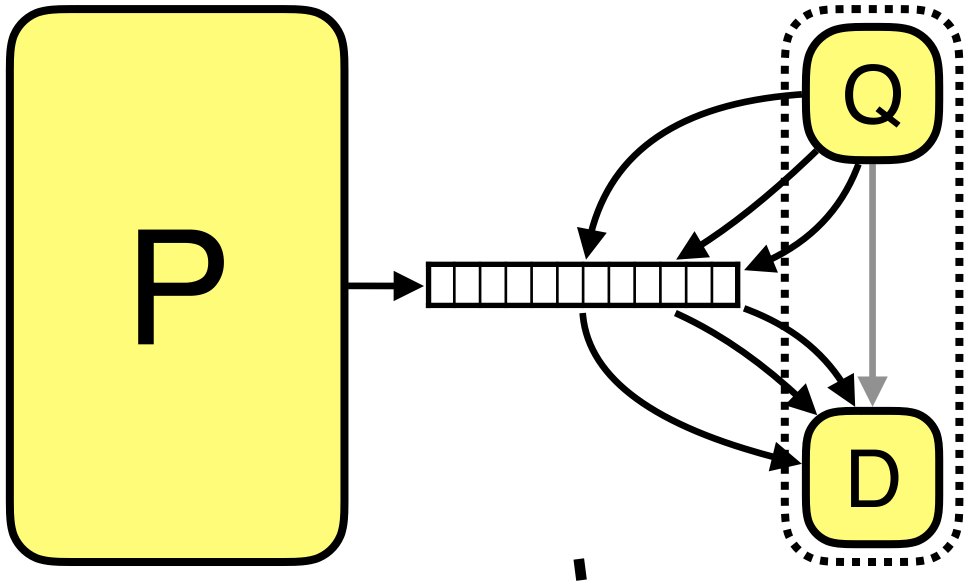
interactive

~~**not succinct**~~

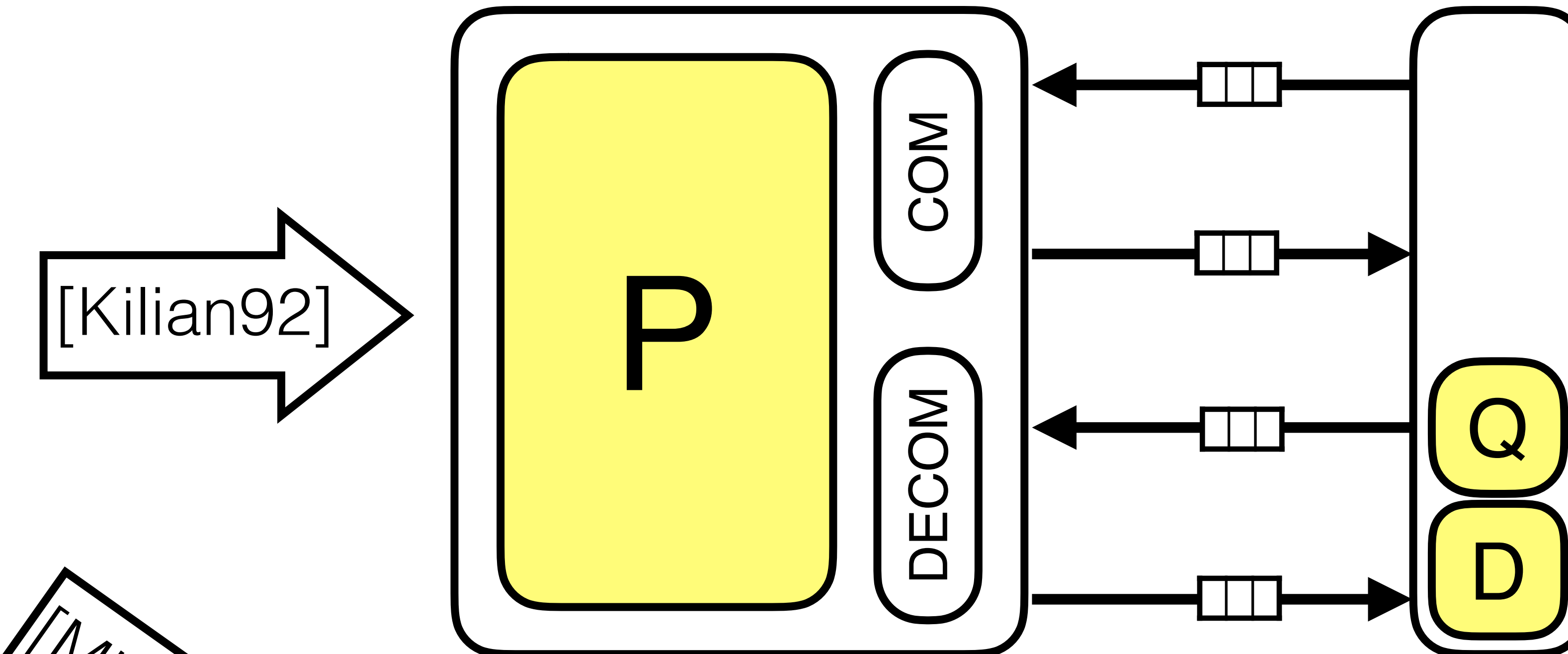
bad concrete efficiency

Achieving Non-Interactivity

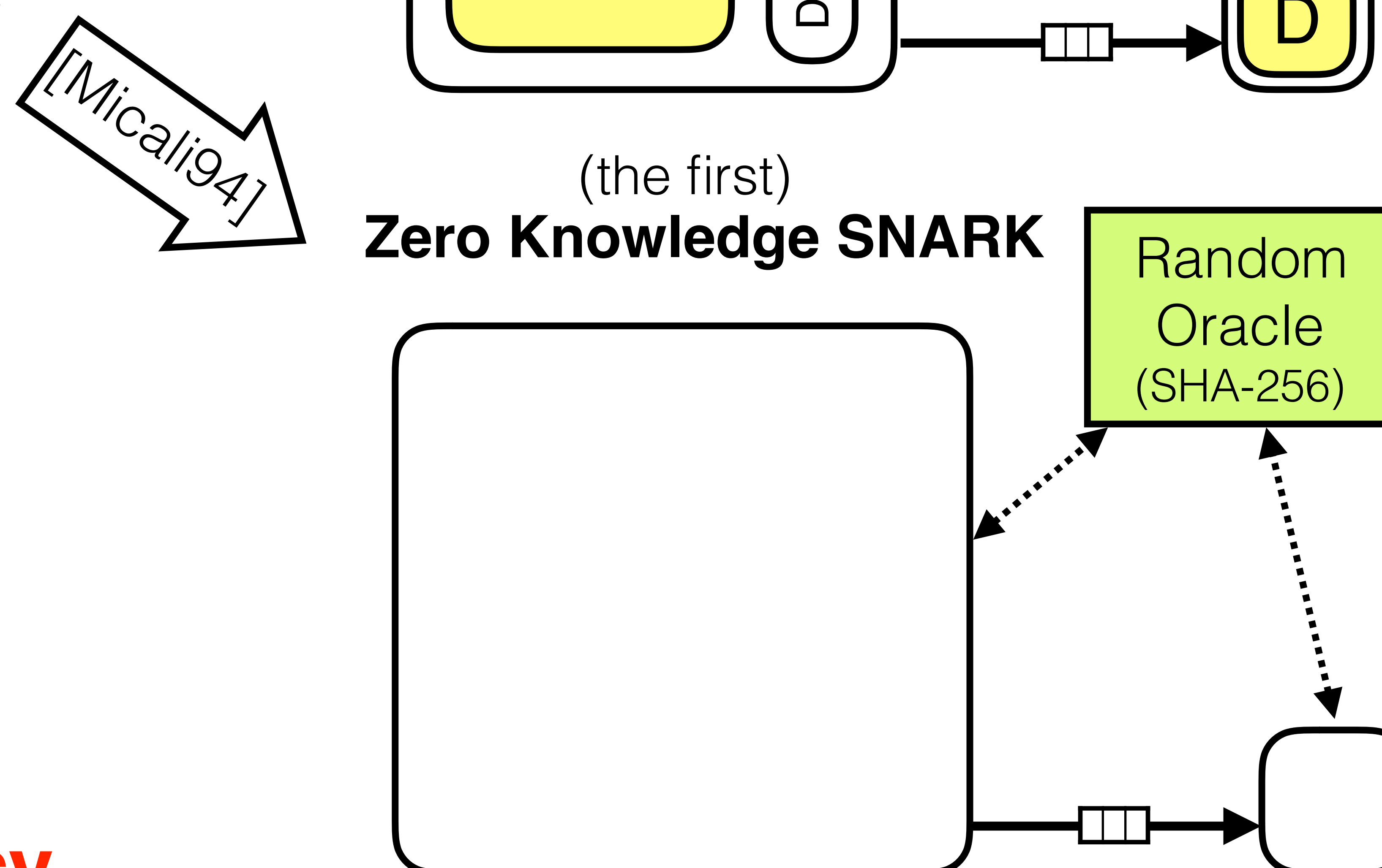
Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]



Zero Knowledge Succinct Proof



(the first)
Zero Knowledge SNARK



TOFIX

~~**interactive**~~

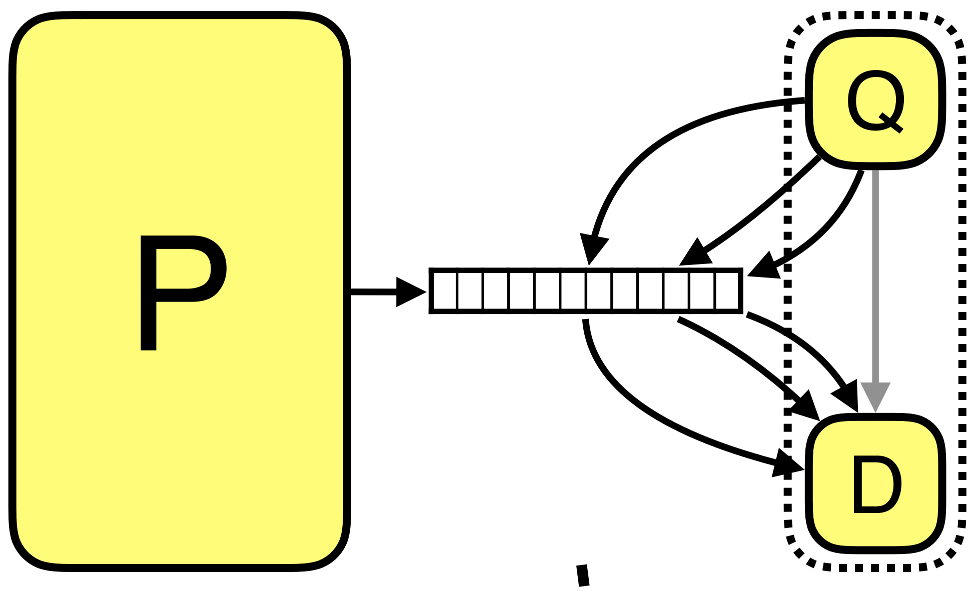
~~**not succinct**~~

bad concrete efficiency

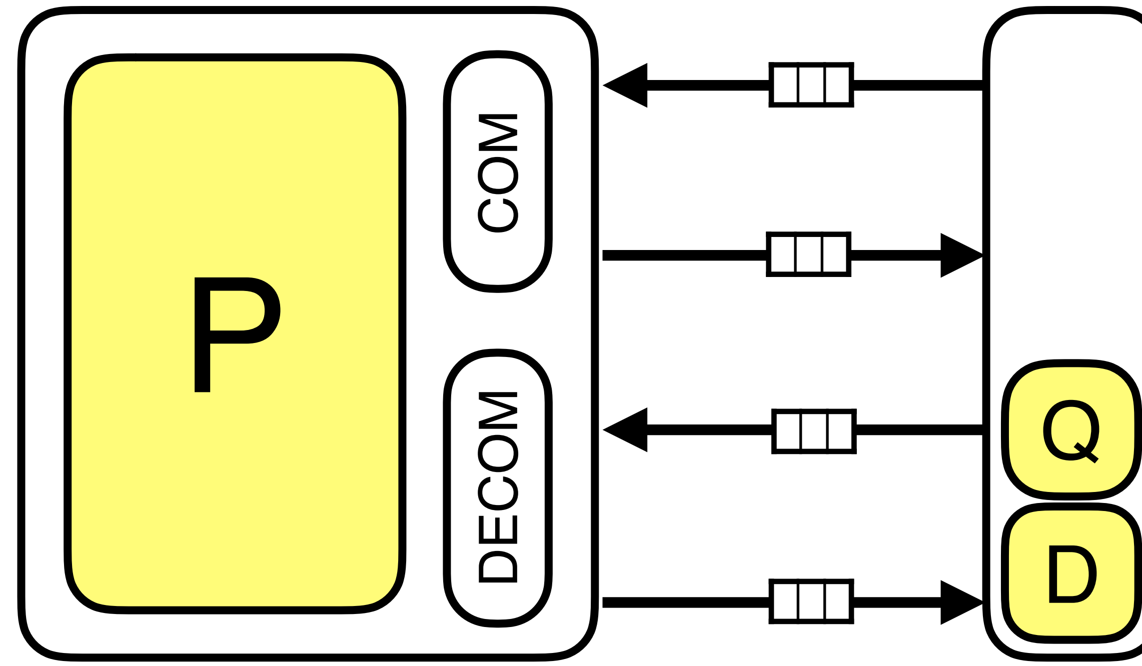
Achieving Non-Interactivity

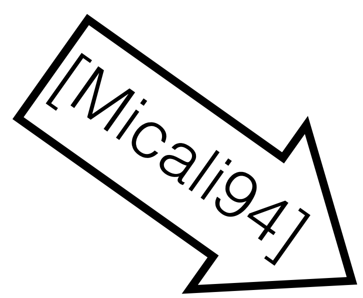
Probabilistically Checkable Proof

[BFLS91][FGLSS96][AS92][ALMSS92]



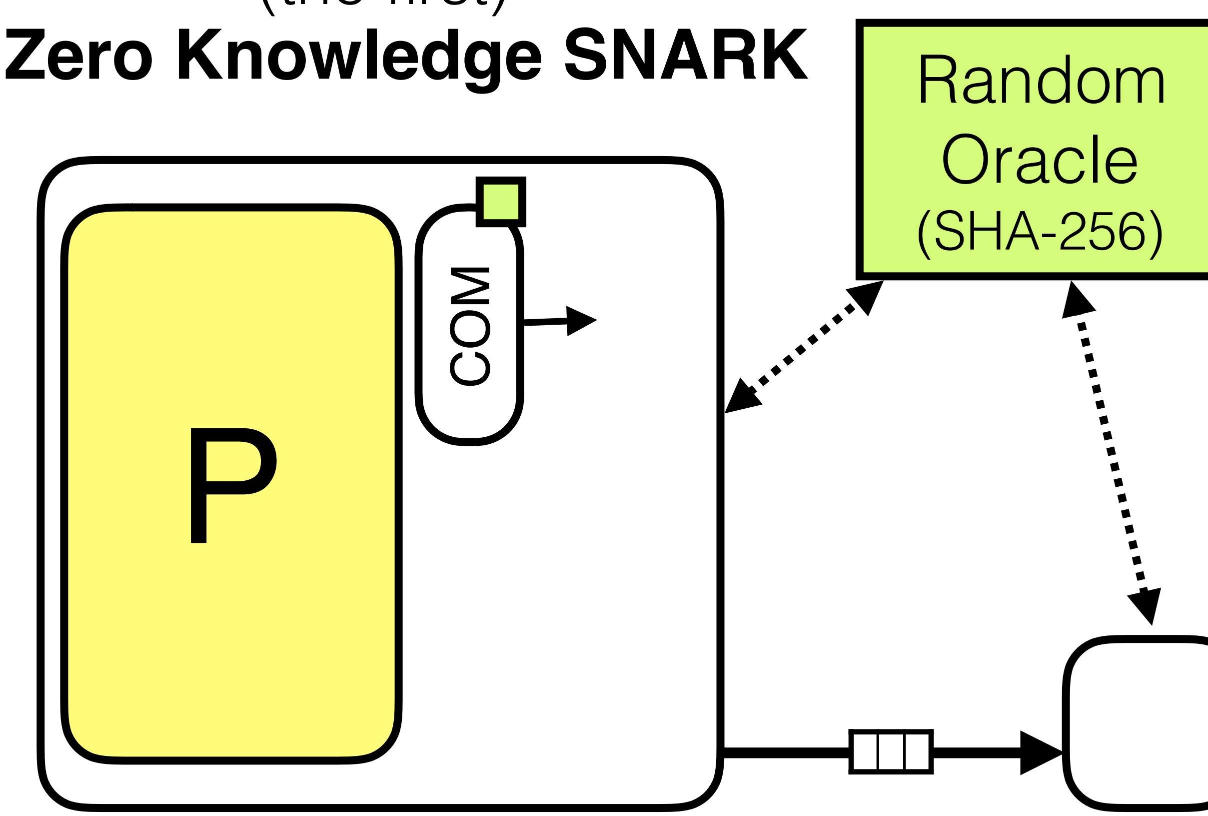
Zero Knowledge Succinct Proof





(the first)

Zero Knowledge SNARK



TOFIX

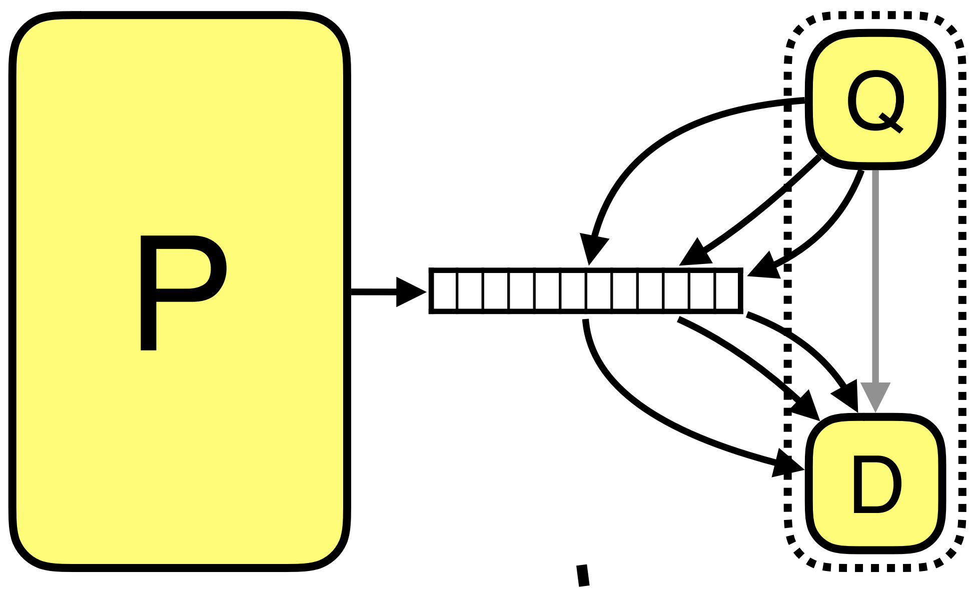
~~**interactive**~~

~~**not succinct**~~

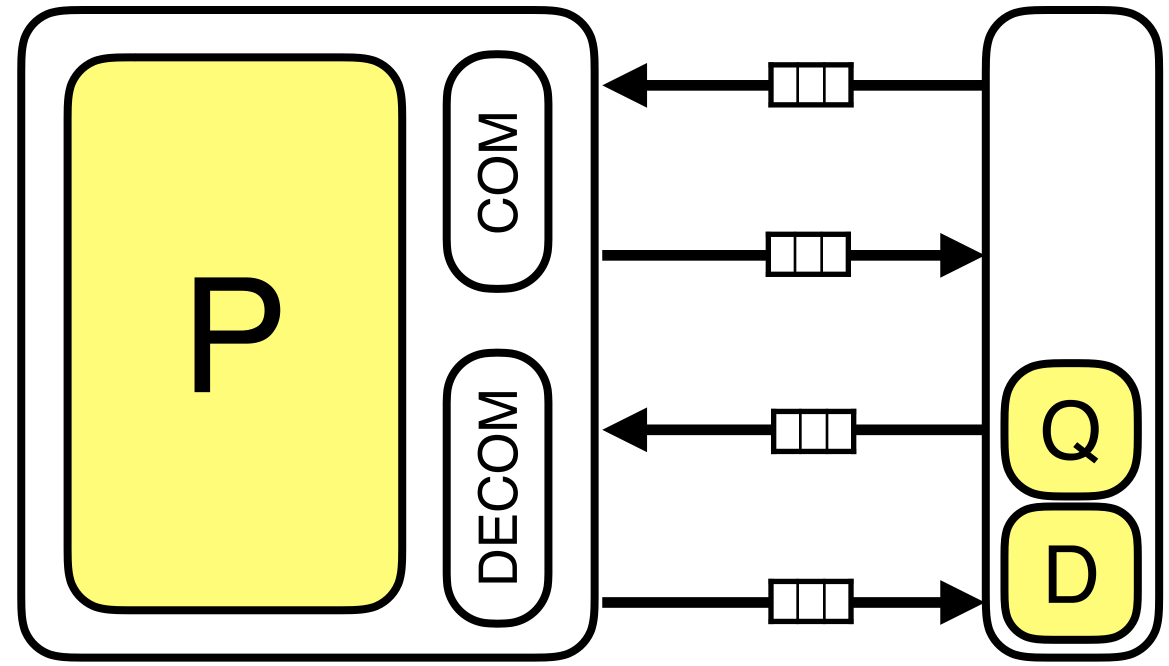
bad concrete efficiency

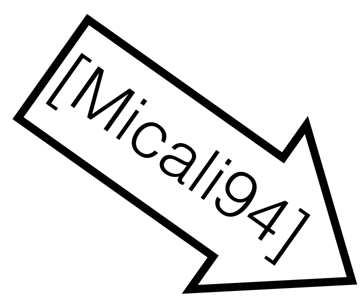
Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

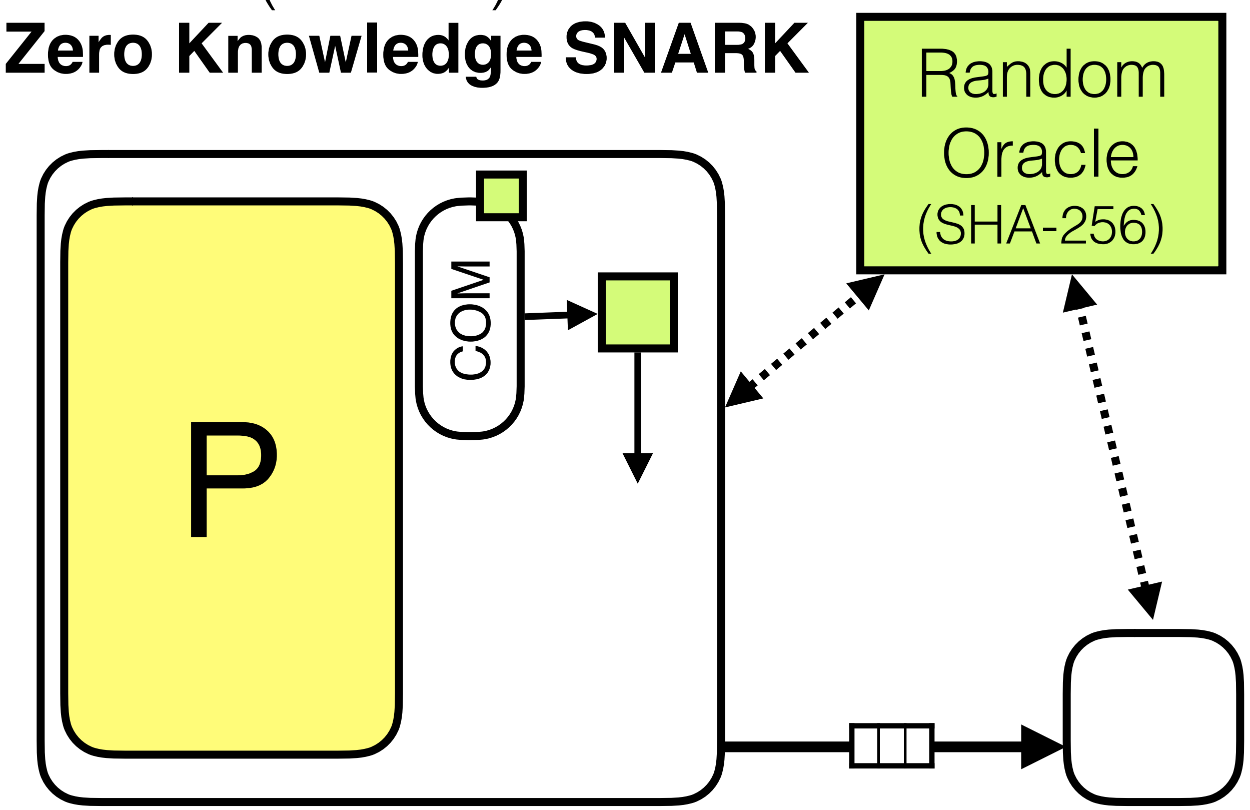


Zero Knowledge Succinct Proof





(the first)
Zero Knowledge SNARK



TOFIX

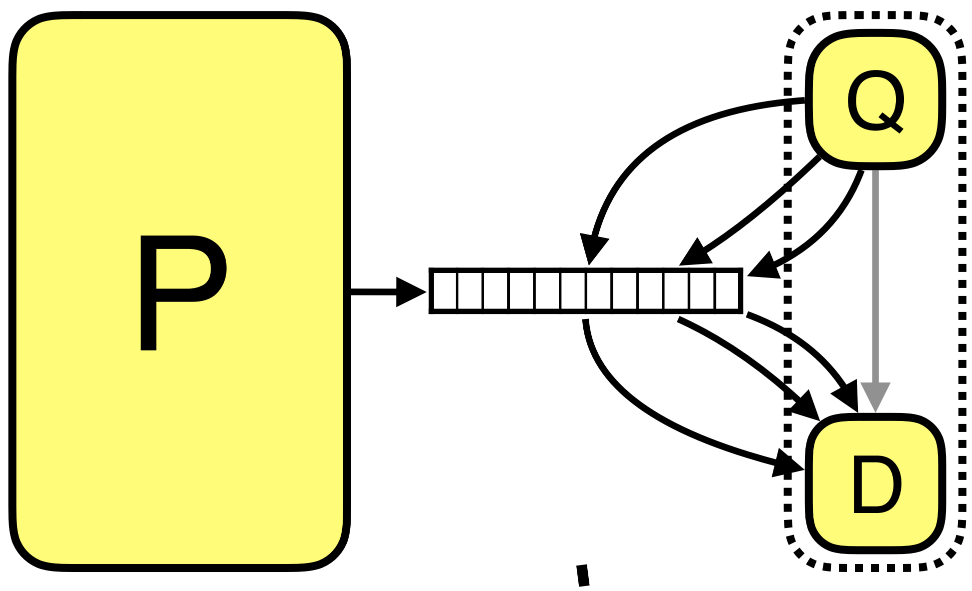
~~interactive~~

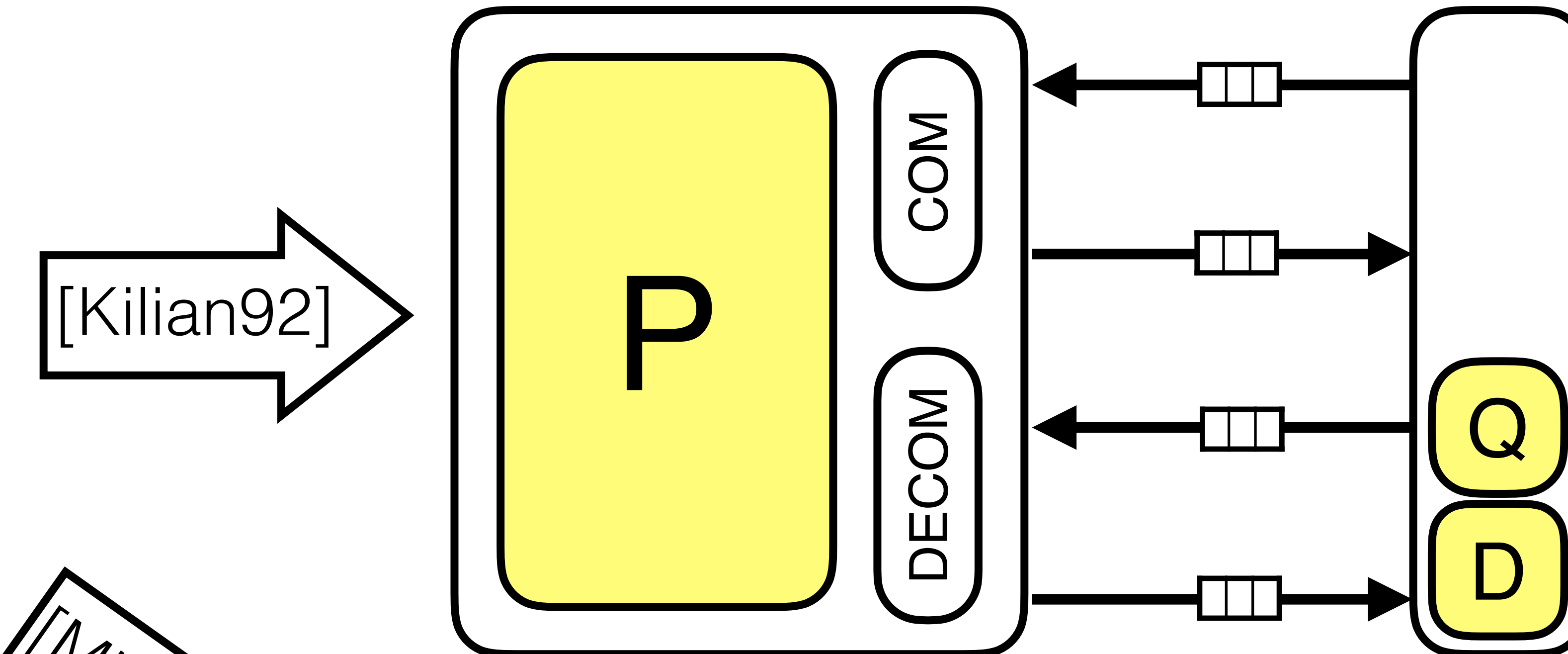
~~not succinct~~

bad concrete efficiency

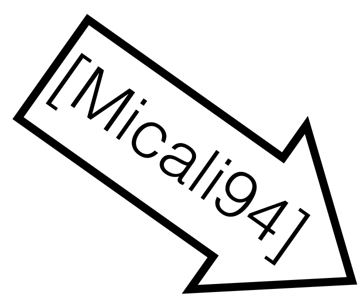
Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

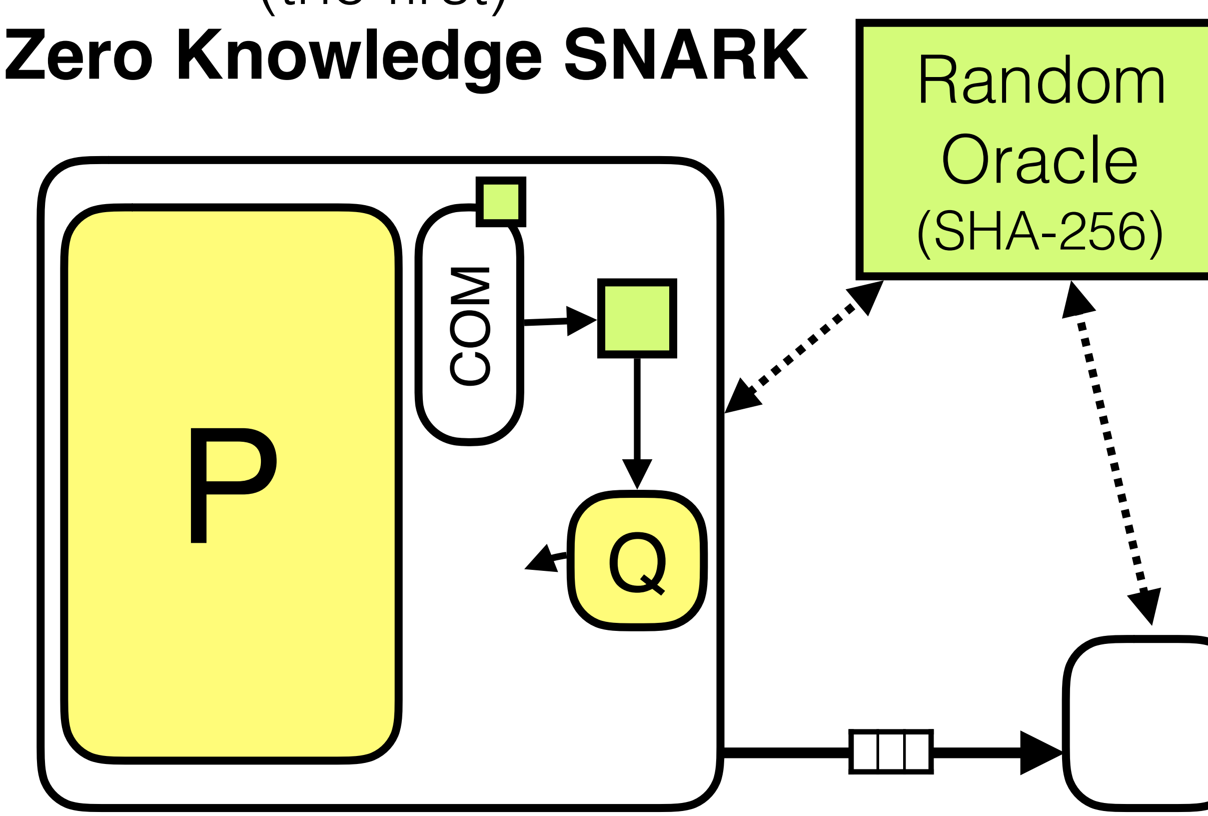




Zero Knowledge Succinct Proof



(the first)
Zero Knowledge SNARK



TOFIX

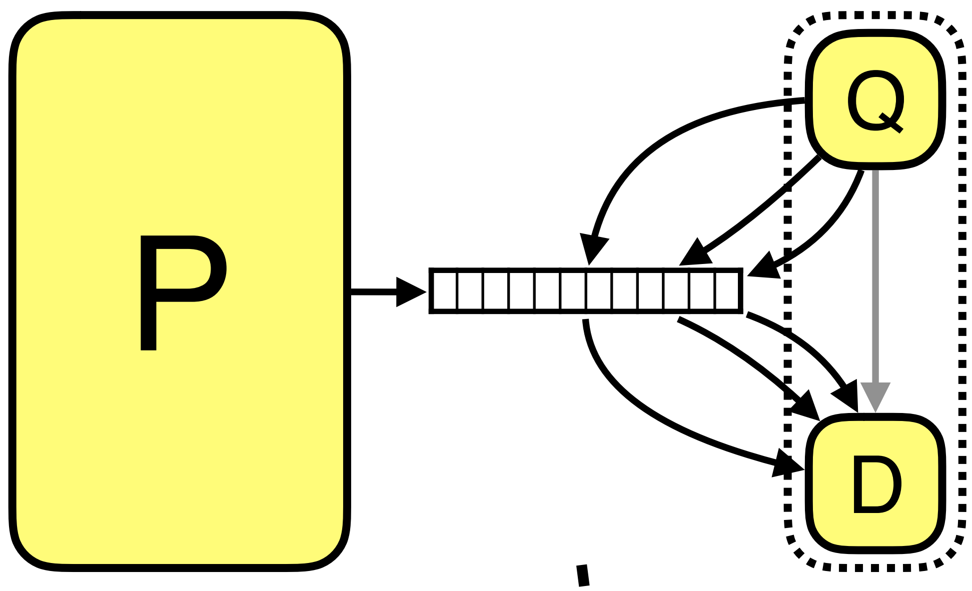
~~interactive~~

~~not succinct~~

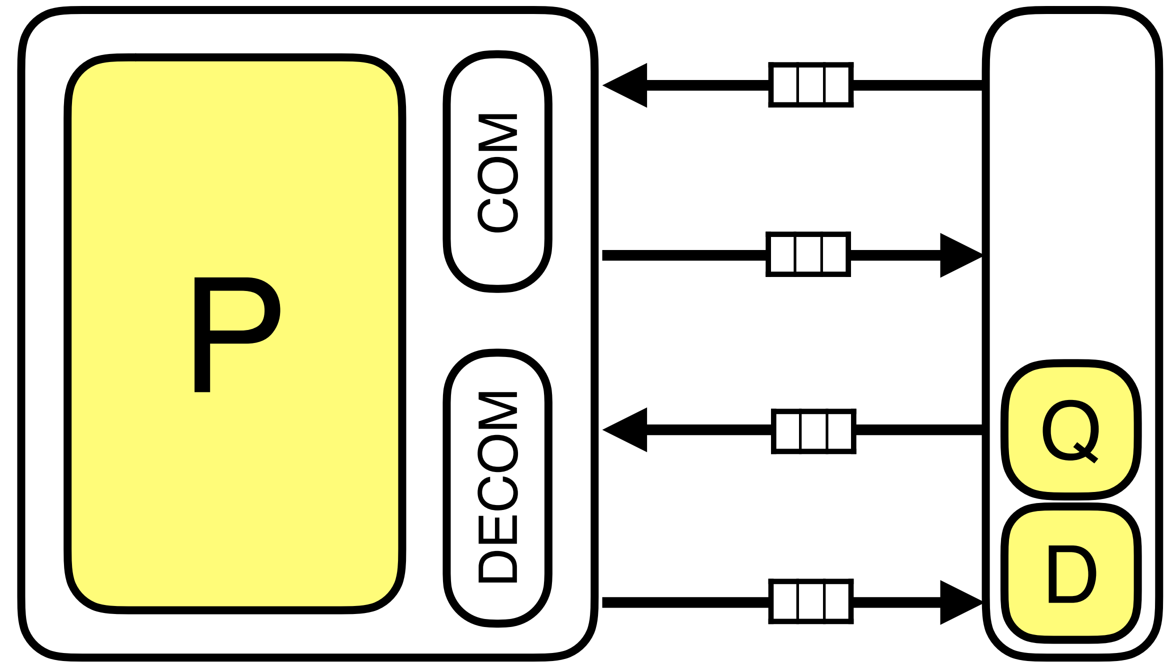
bad concrete efficiency

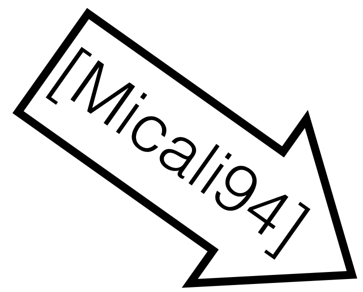
Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

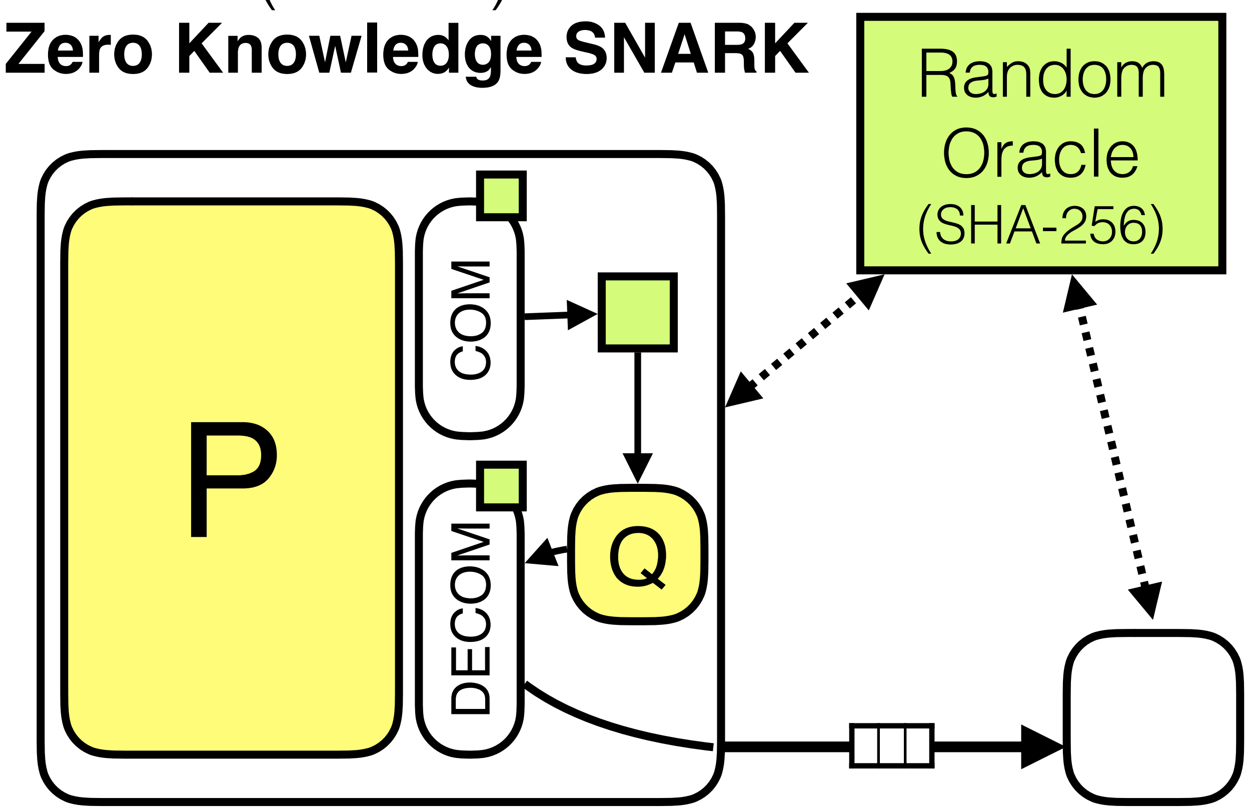


Zero Knowledge Succinct Proof





(the first)
Zero Knowledge SNARK



TOFIX

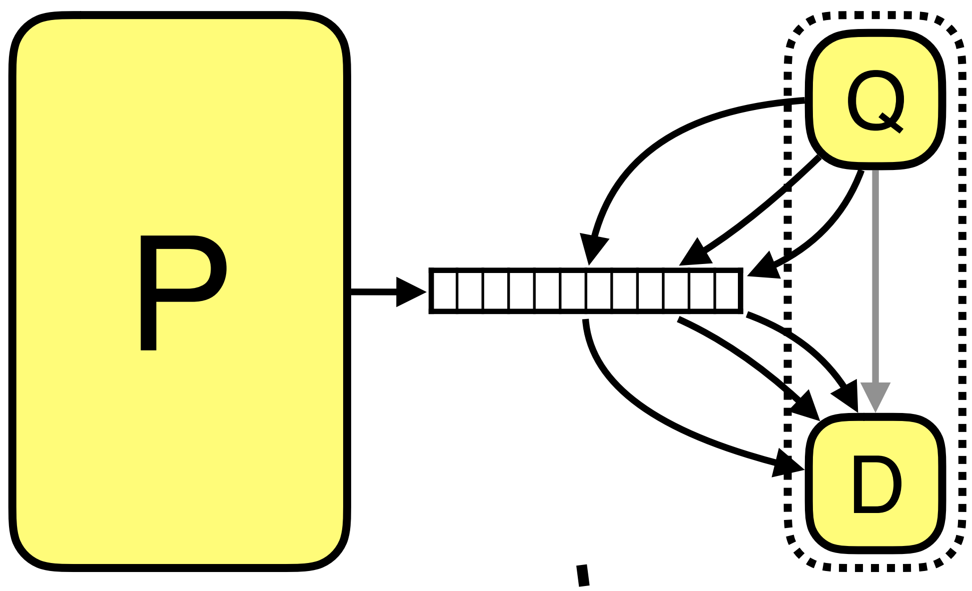
~~interactive~~

~~not succinct~~

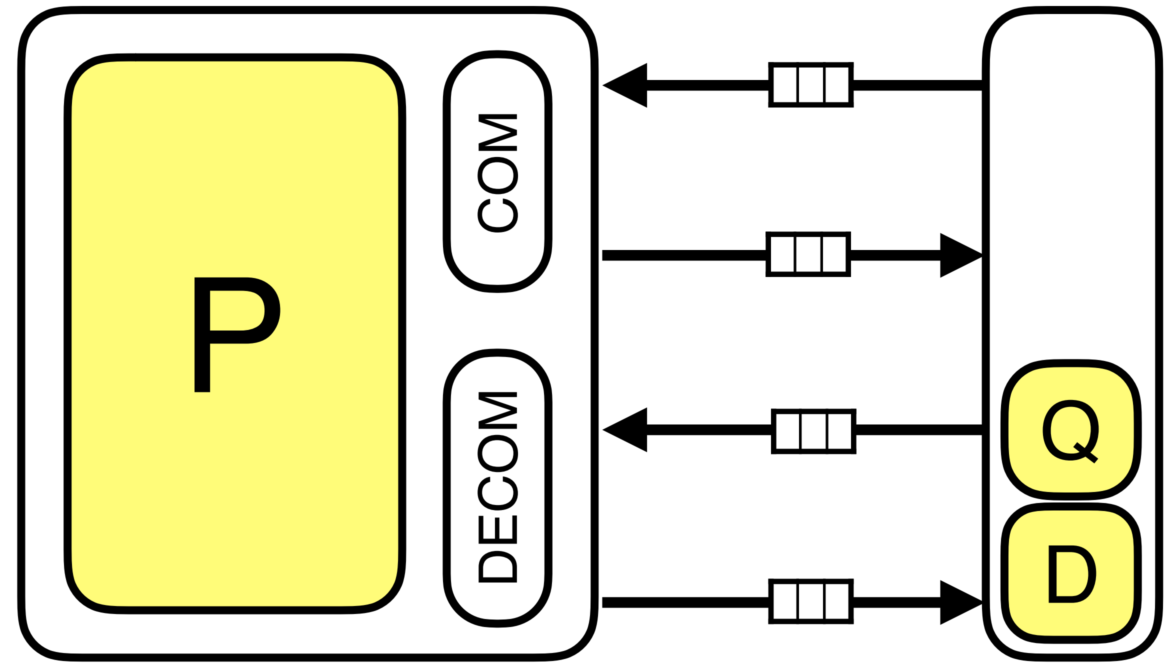
bad concrete efficiency

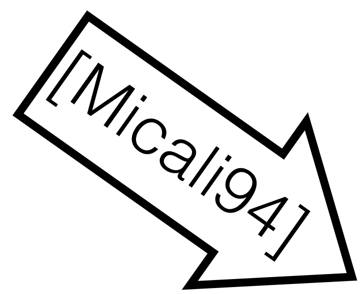
Achieving Non-Interactivity

Probabilistically Checkable Proof
[BFLS91][FGLSS96][AS92][ALMSS92]

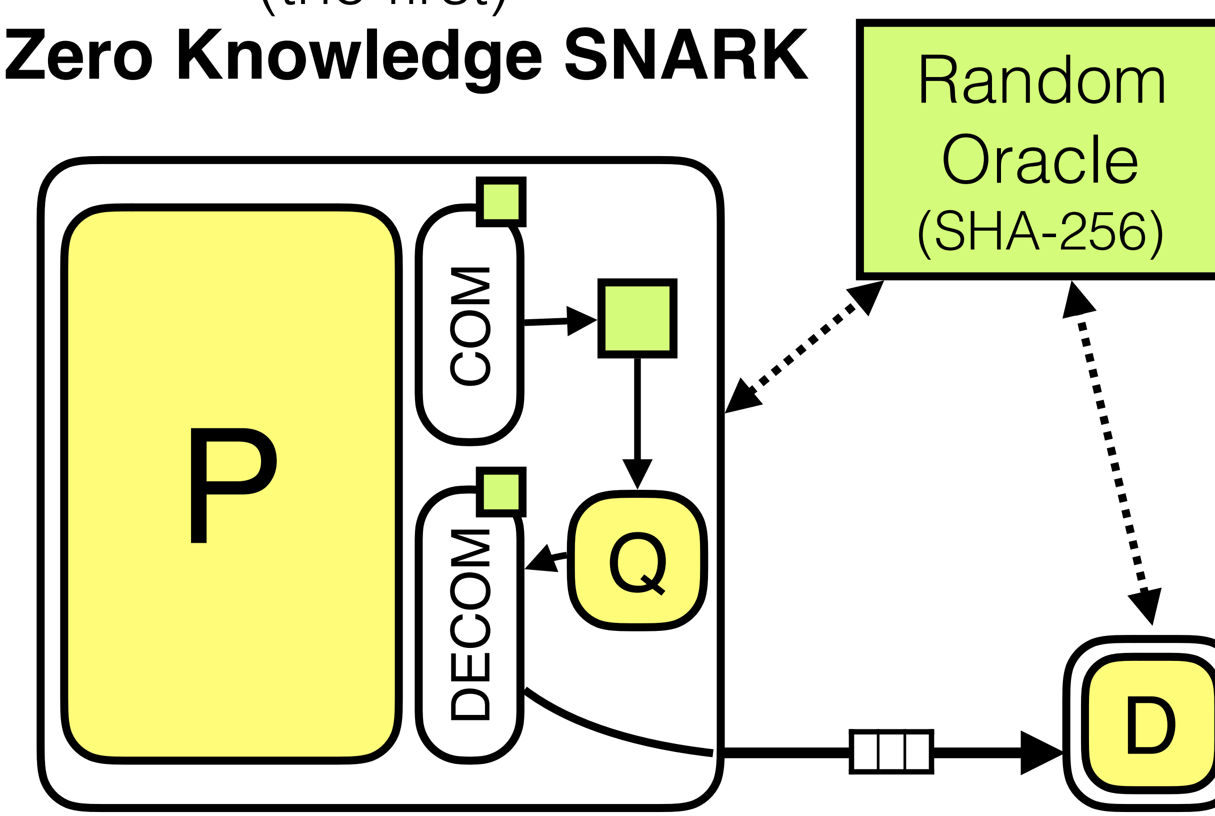


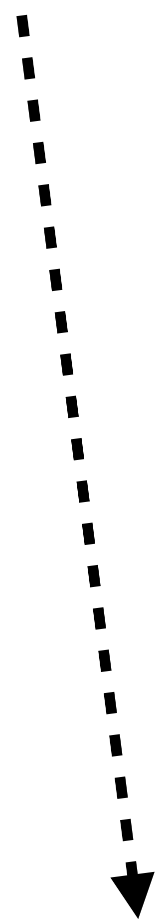
Zero Knowledge Succinct Proof





(the first)
Zero Knowledge SNARK





TOFIX

~~**interactive**~~

~~**not succinct**~~

bad concrete efficiency

Modern Era

The Quest for ZK-SNARKs without Random Oracles

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent
[D 92]

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent
[D 92]

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent
[D 92]

Extractable Hash Functions

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent
[D 92]

Extractable Hash Functions

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent
[D 92]

Extractable Hash Functions

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

based on **PCPs**

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent [D 92]

Extractable Hash Functions

Linear-Only Encryption/Encoding

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

based on **PCPs**

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Knowledge of Exponent [D 92]

Extractable Hash Functions

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

based on **PCPs**

Linear-Only Encryption/Encoding

[G 10]
[L 12]
[B**C**IOP 13]
[GGPR 13]
[PGHR 13]
[B**C**GTV 13]
[B**C**TV 14a]
[DFGK 14]
[WSRBW 15]
[BBFR 15]
[B**C**TV 14b]
[**C**TV 15]
[CFHKKNPZ 15]
[BISW 17]
⋮

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Generic Group Model
[G 16]

Knowledge of Exponent
[D 92]

Extractable Hash Functions

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

based on **PCPs**

Linear-Only Encryption/Encoding

[G 10]
[L 12]
[B**C**IOP 13]
[GGPR 13]
[PGHR 13]
[B**C**GTV 13]
[B**C**TV 14a]
[DFGK 14]
[WSRBW 15]
[BBFR 15]
[B**C**TV 14b]
[**C**TV 15]
[CFHKKNPZ 15]
[BISW 17]
⋮

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Generic Group Model
[G 16]

Knowledge of Exponent
[D 92]

Extractable Hash Functions

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

based on **PCPs**

Linear-Only Encryption/Encoding

[G 10]
[L 12]
[B**C**IOP 13]
[GGPR 13]
[PGHR 13]
[B**C**GTV 13]
[B**C**TV 14a]
[DFGK 14]
[WSRBW 15]
[BBFR 15]
[B**C**TV 14b]
[**C**TV 15]
[CFHKKNPZ 15]
[BISW 17]
⋮

based on **linear PCPs**

The Quest for ZK-SNARKs without Random Oracles

Negative result: constructing them "requires strong assumptions" [GW11]

Positive results (under strong assumptions):

Generic Group Model [G 16]

Knowledge of Exponent [D 92]

Extractable Hash Functions

[BC**C**T 12]
[DFH 12]
[GLR 12]
[B**C** 12]
[BC**C**T 13]
[BC**C**GLRT 16]

based on **PCPs**

Linear-Only Encryption/Encoding

[G 10]
[L 12]
[B**C**IOP 13]
[GGPR 13]
[PGHR 13]
[B**C**GTV 13]
[B**C**TV 14a]
[DFGK 14]
[WSRBW 15]
[BBFR 15]
[B**C**TV 14b]
[**C**TV 15]
[CFHKKNPZ 15]
[BISW 17]
⋮

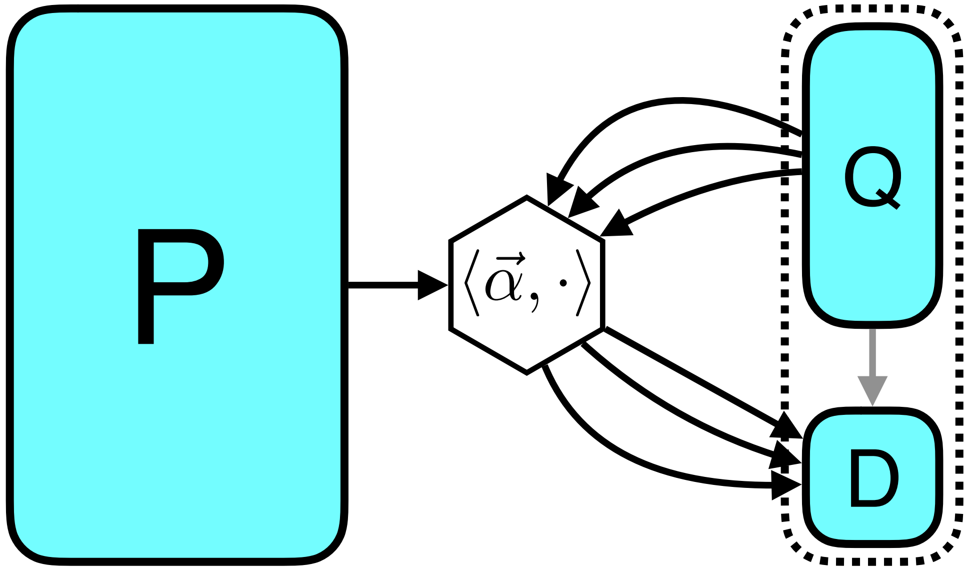
based on **linear PCPs**

strike a different tradeoff…

ZK-SNARKs from Linear PCPs

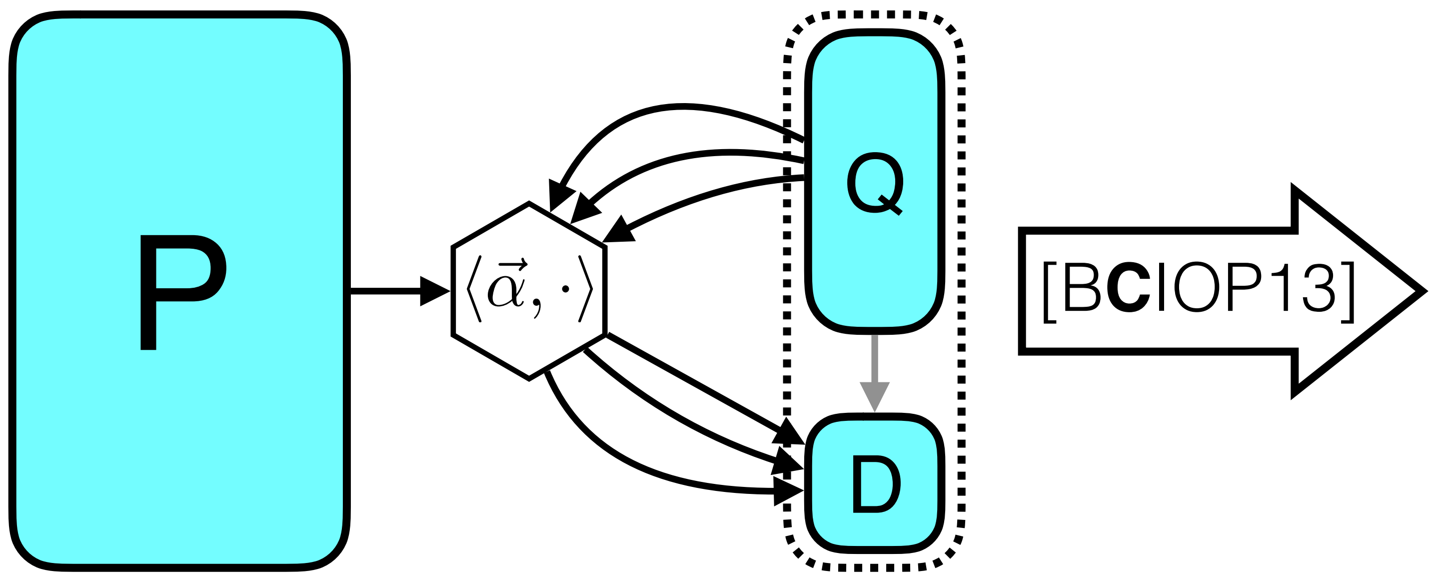
ZK-SNARKs from Linear PCPs

Linear PCP
[IKO07][B**C**IOP13]



ZK-SNARKs from Linear PCPs

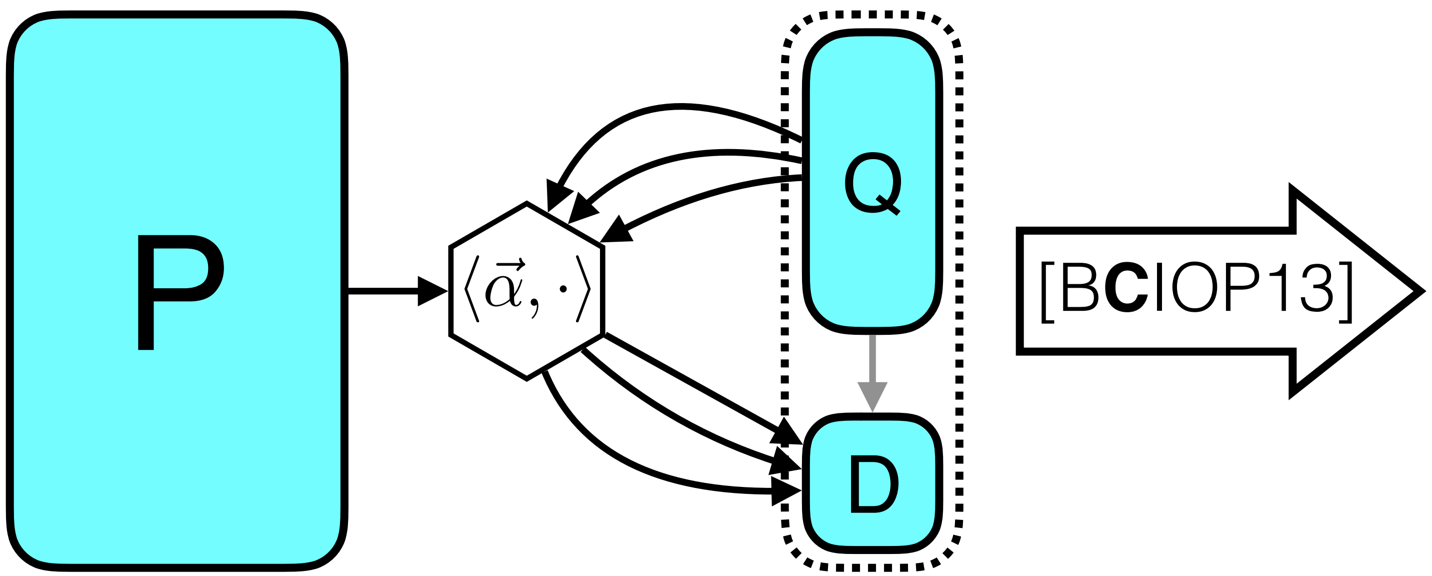
Linear PCP
[IKO07][B**C**IOP13]



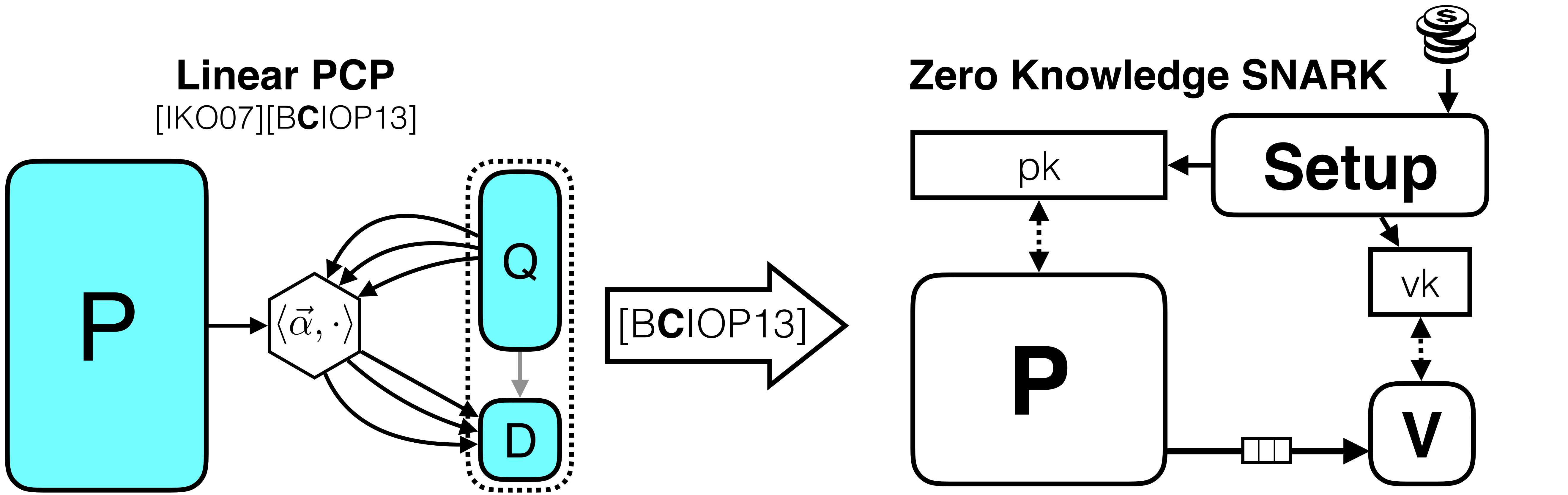
ZK-SNARKs from Linear PCPs

Linear PCP
[IKO07][B**C**IOP13]

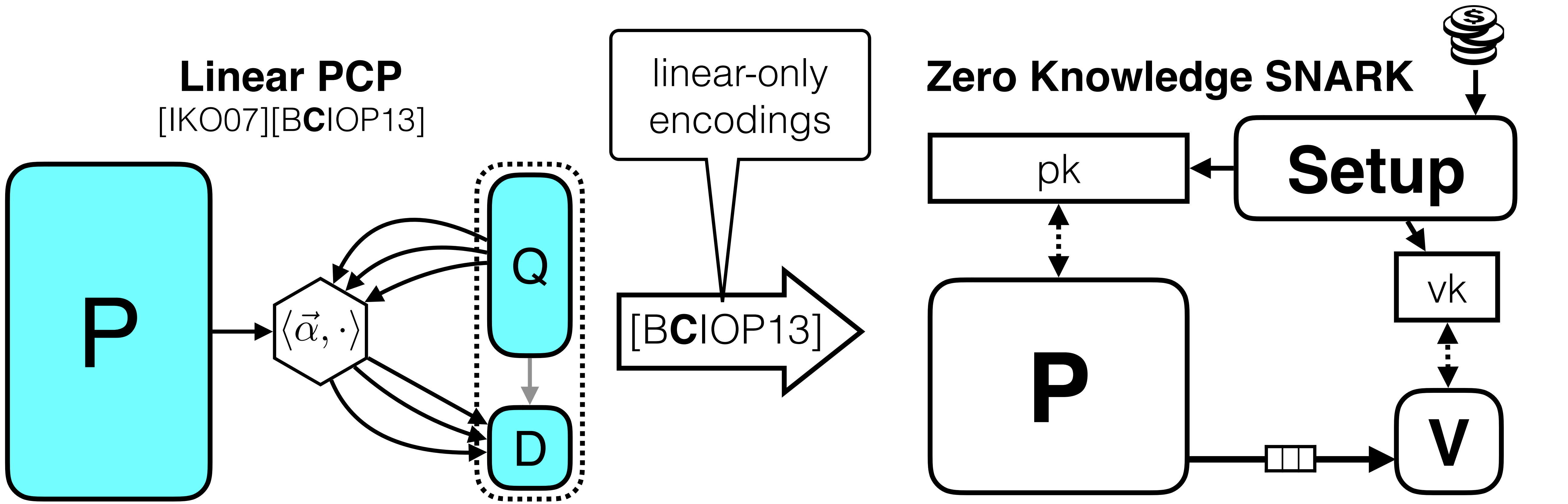
Zero Knowledge SNARK



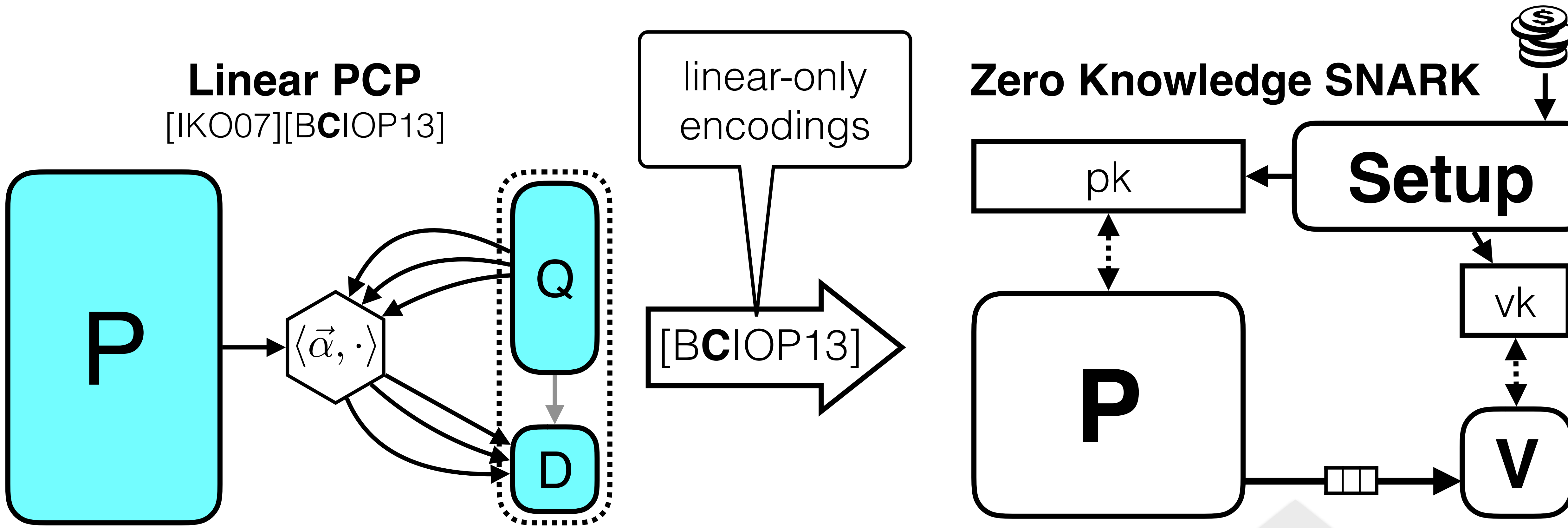
ZK-SNARKs from Linear PCPs



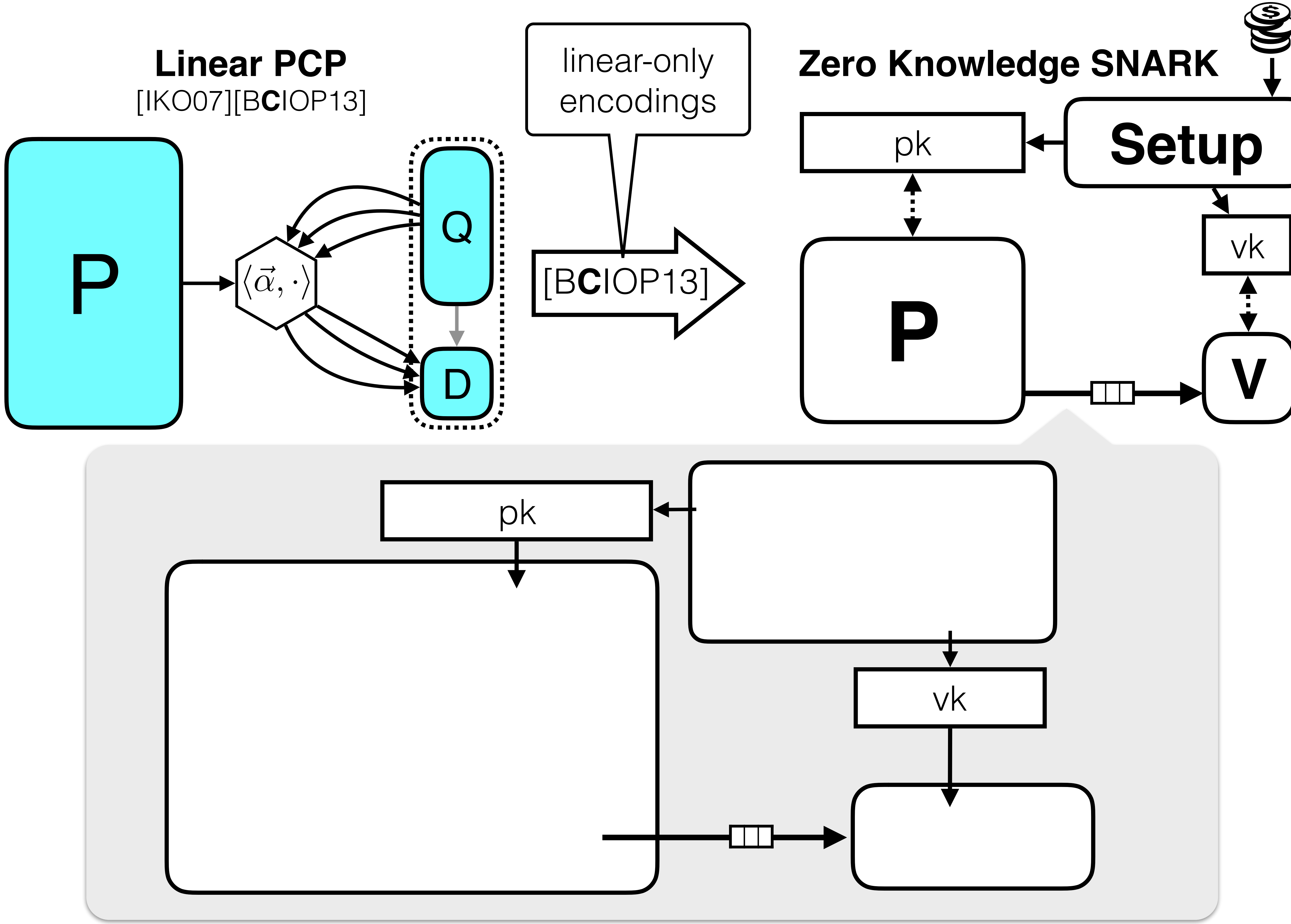
ZK-SNARKs from Linear PCPs



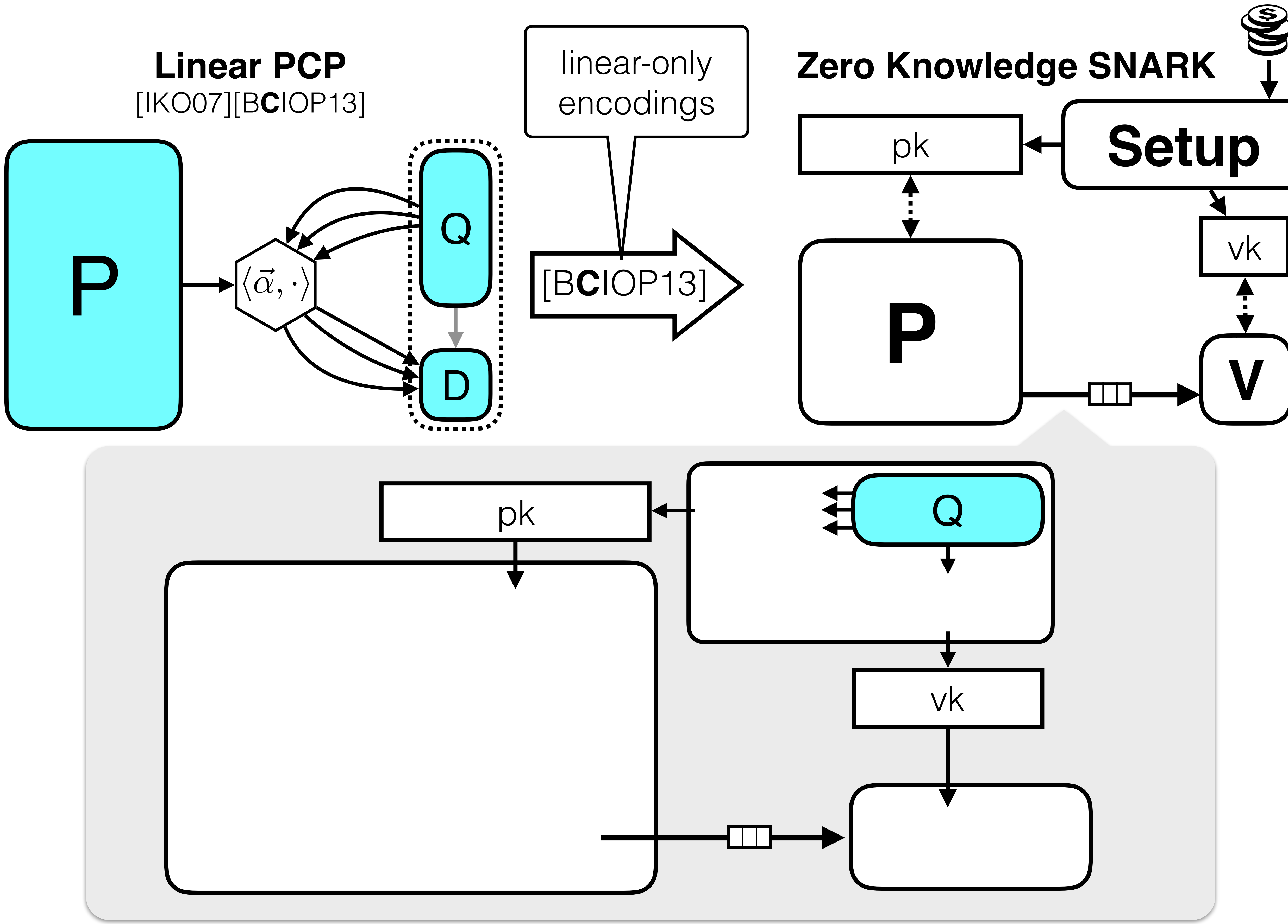
ZK-SNARKs from Linear PCPs



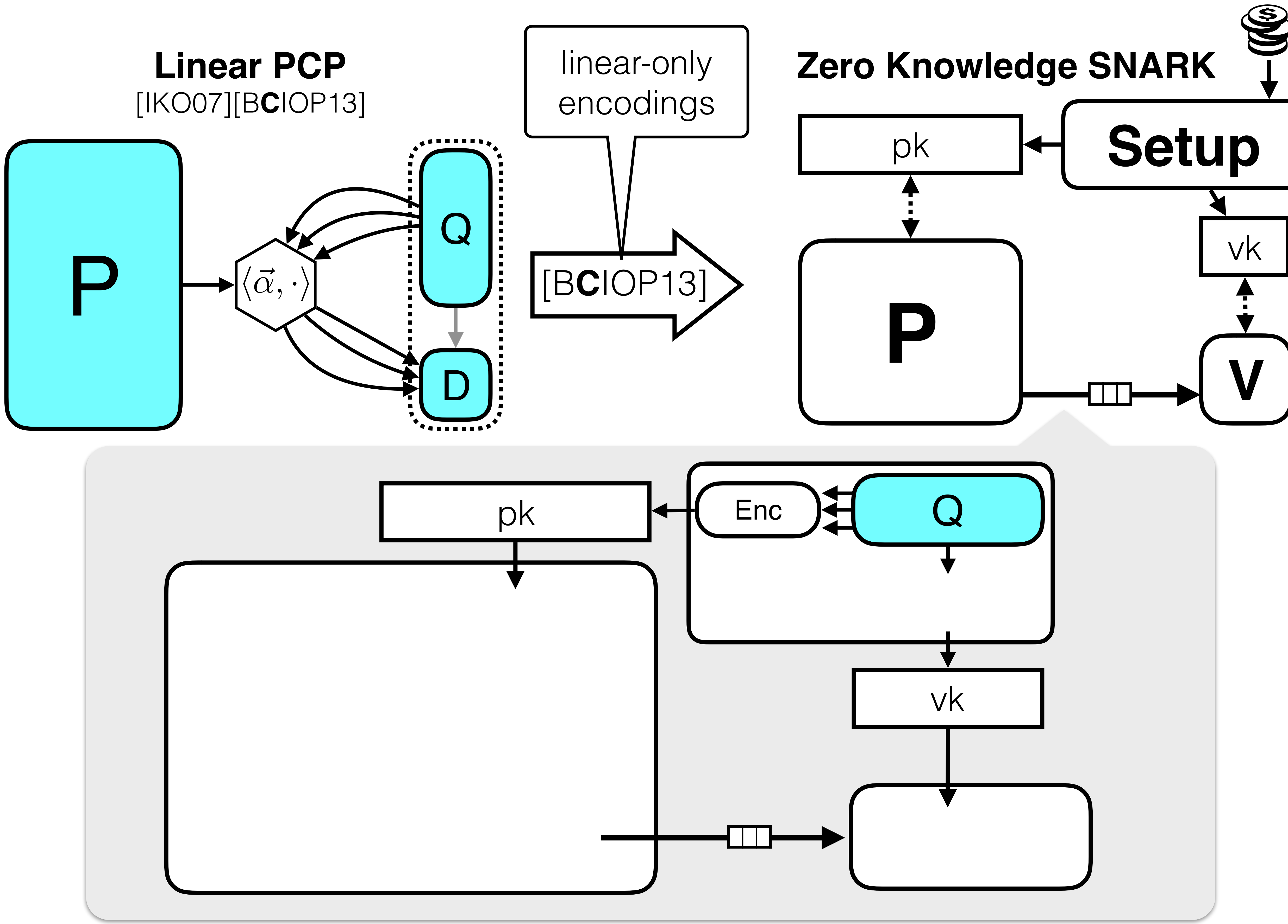
ZK-SNARKs from Linear PCPs



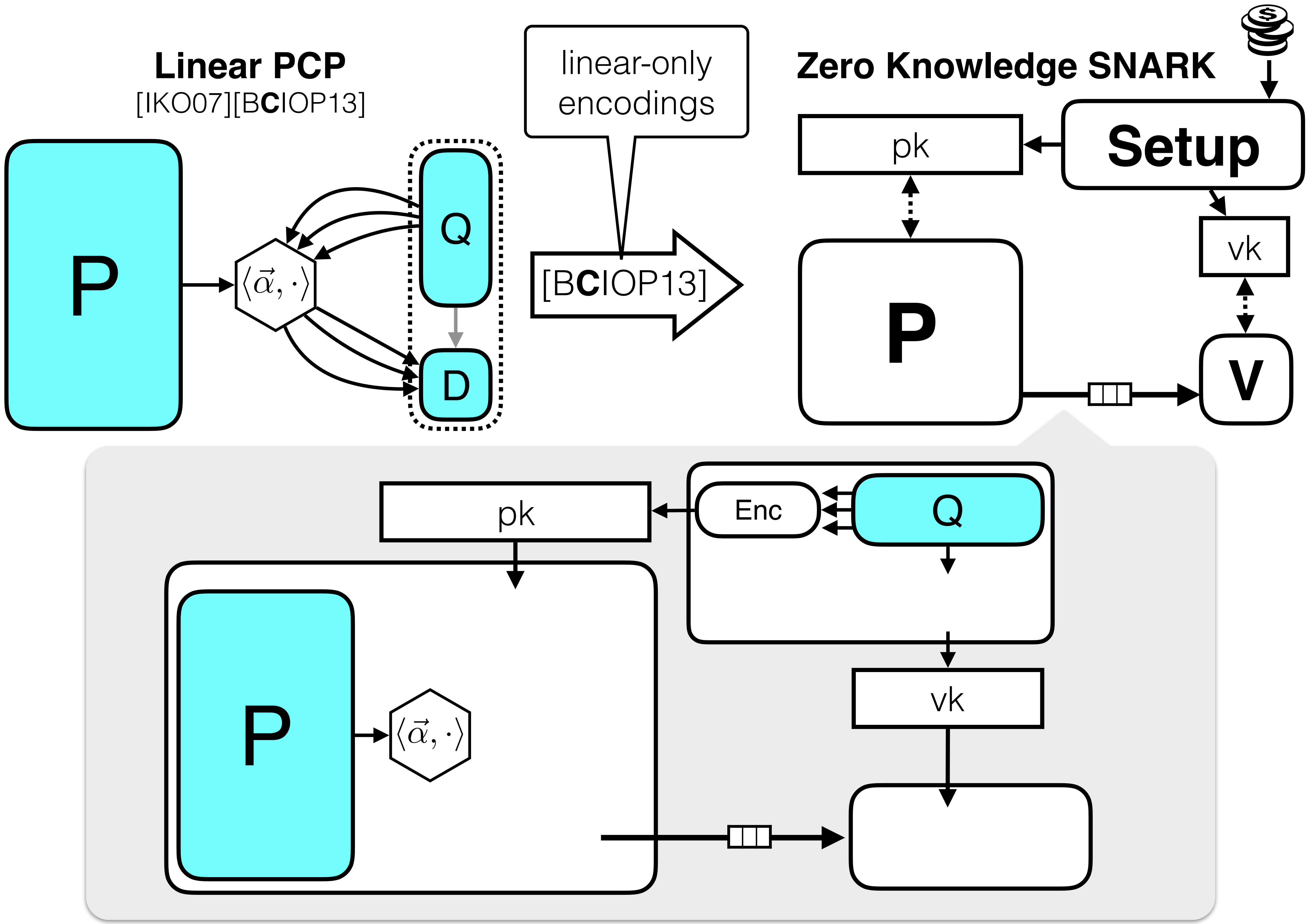
ZK-SNARKs from Linear PCPs



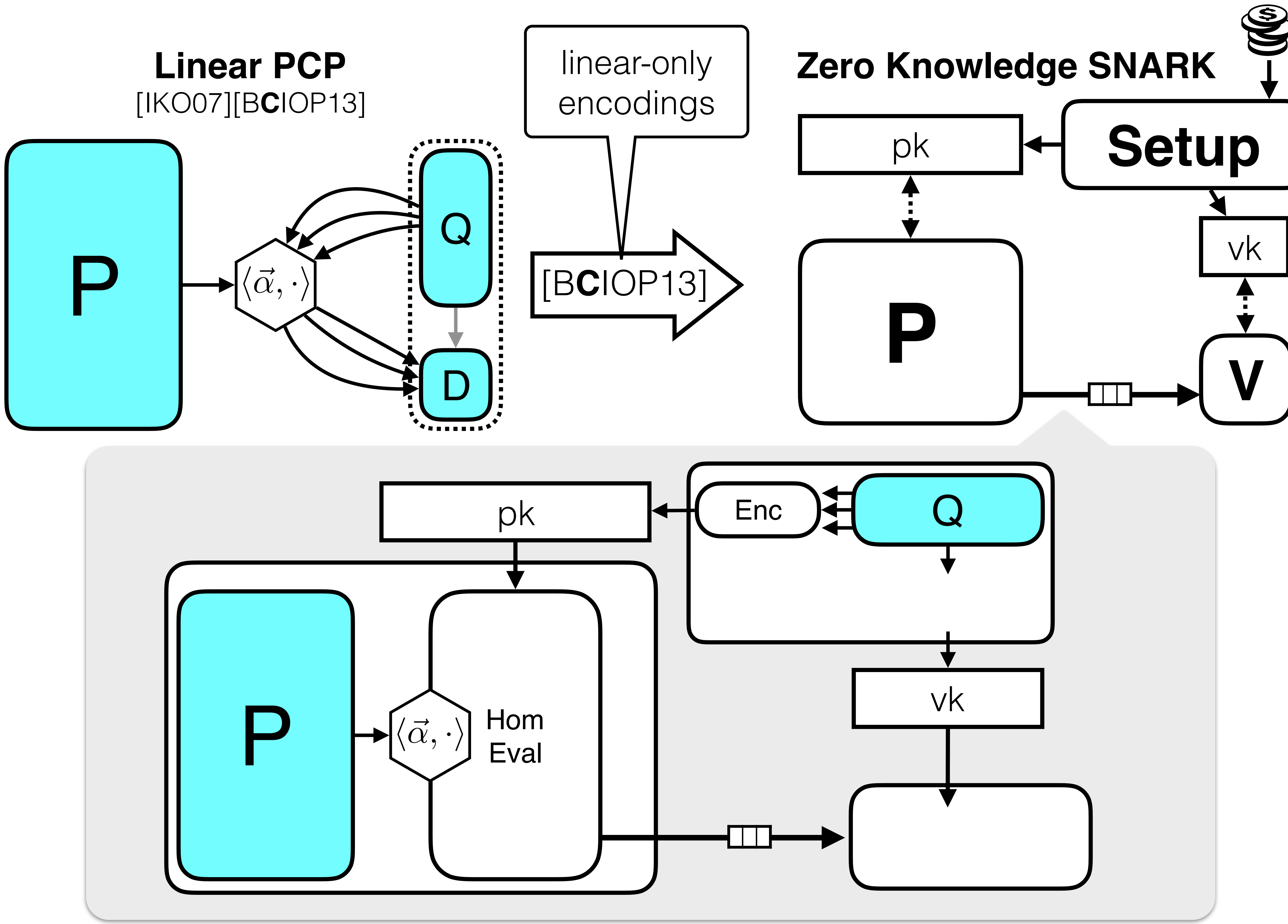
ZK-SNARKs from Linear PCPs



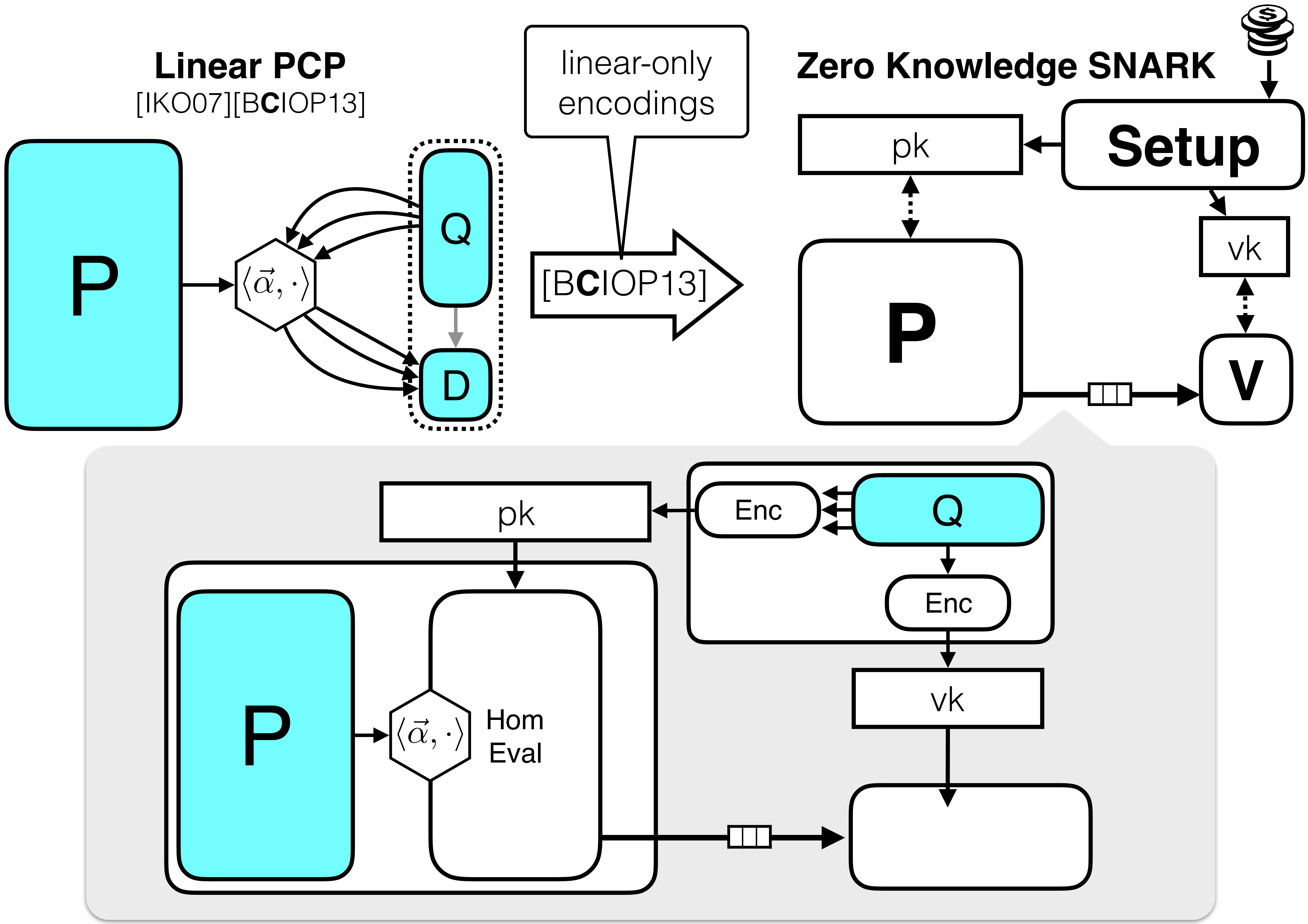
ZK-SNARKs from Linear PCPs



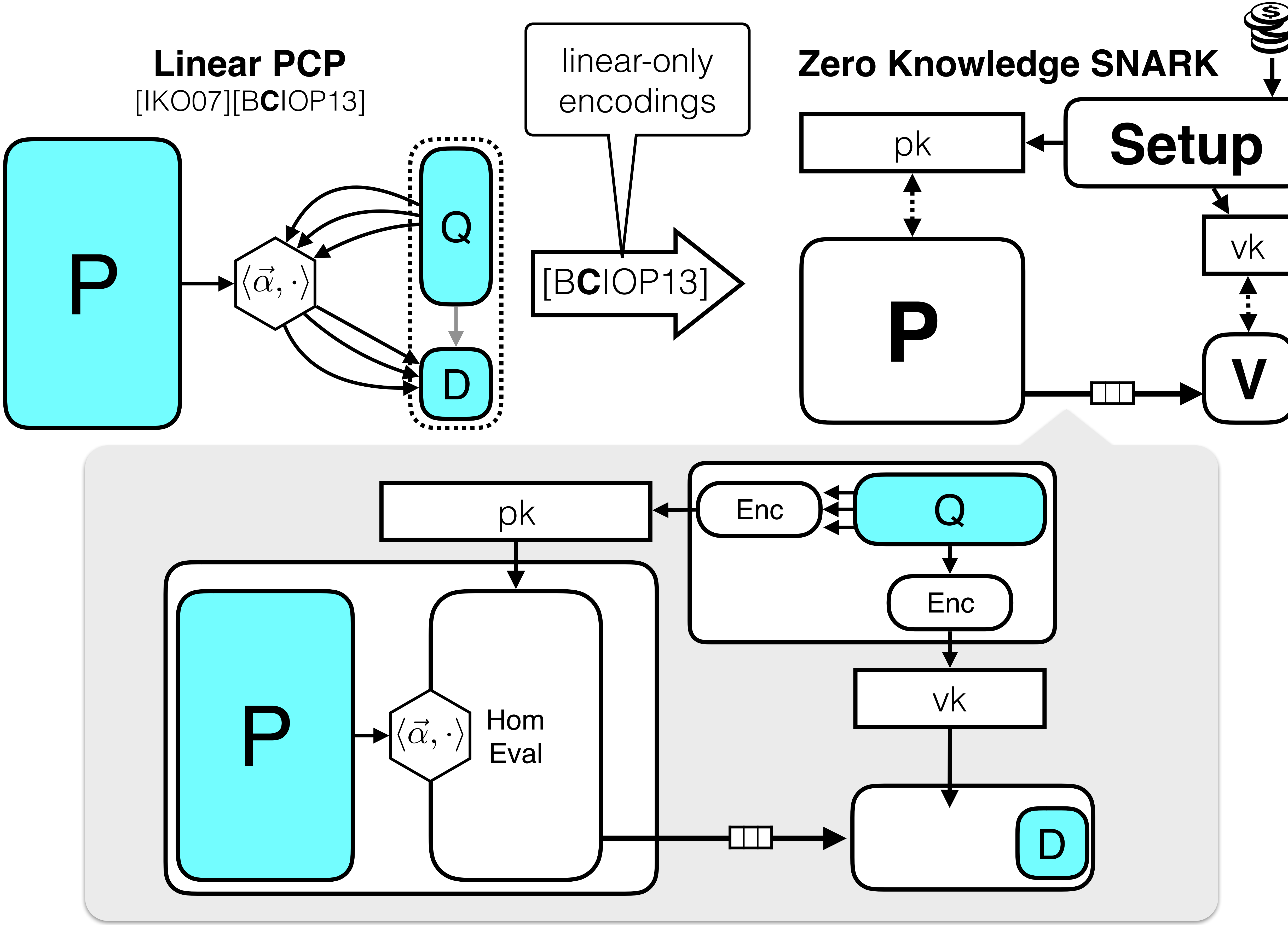
ZK-SNARKs from Linear PCPs



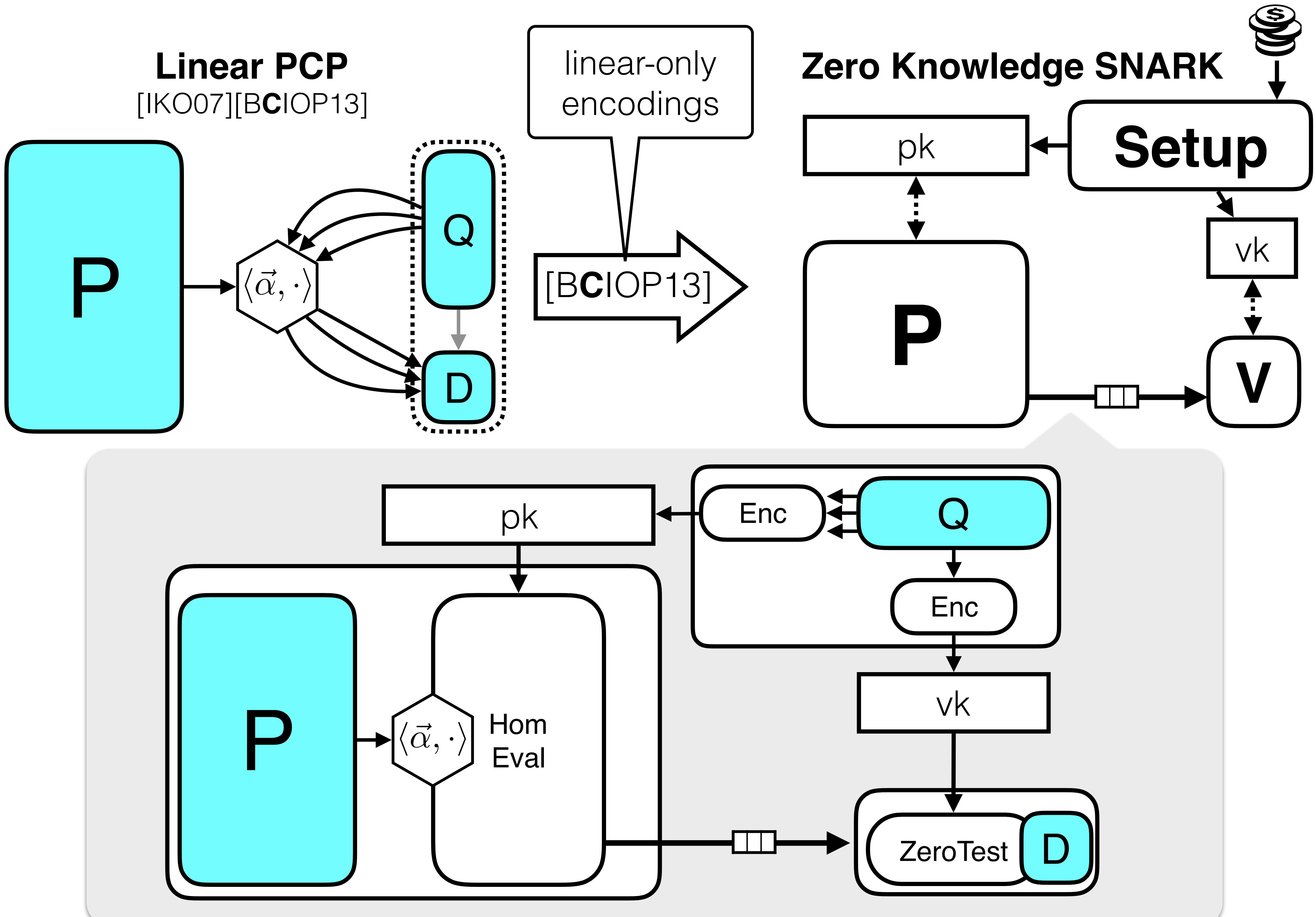
ZK-SNARKs from Linear PCPs



ZK-SNARKs from Linear PCPs



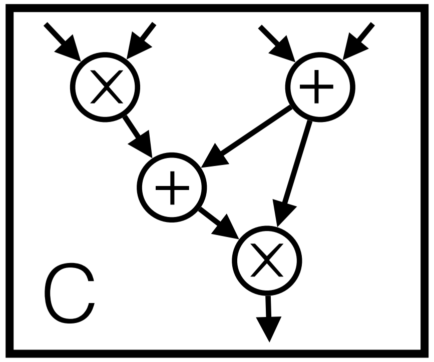
ZK-SNARKs from Linear PCPs



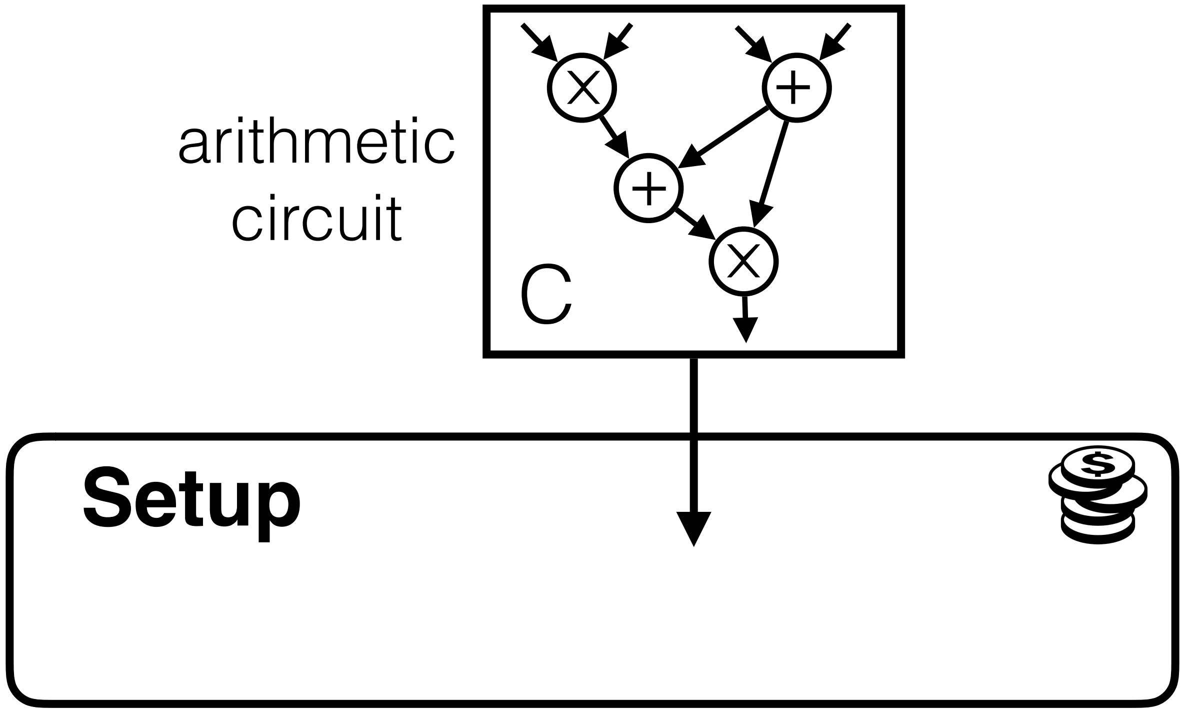
ZK-SNARKs from Linear PCPs

ZK-SNARKs from Linear PCPs

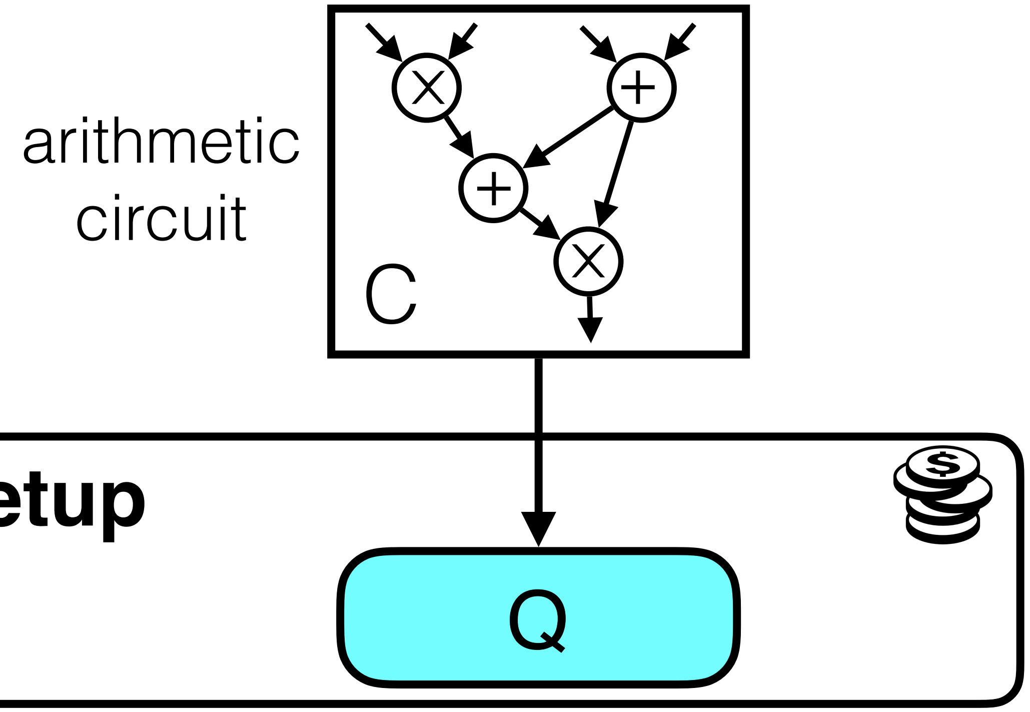
arithmetic circuit



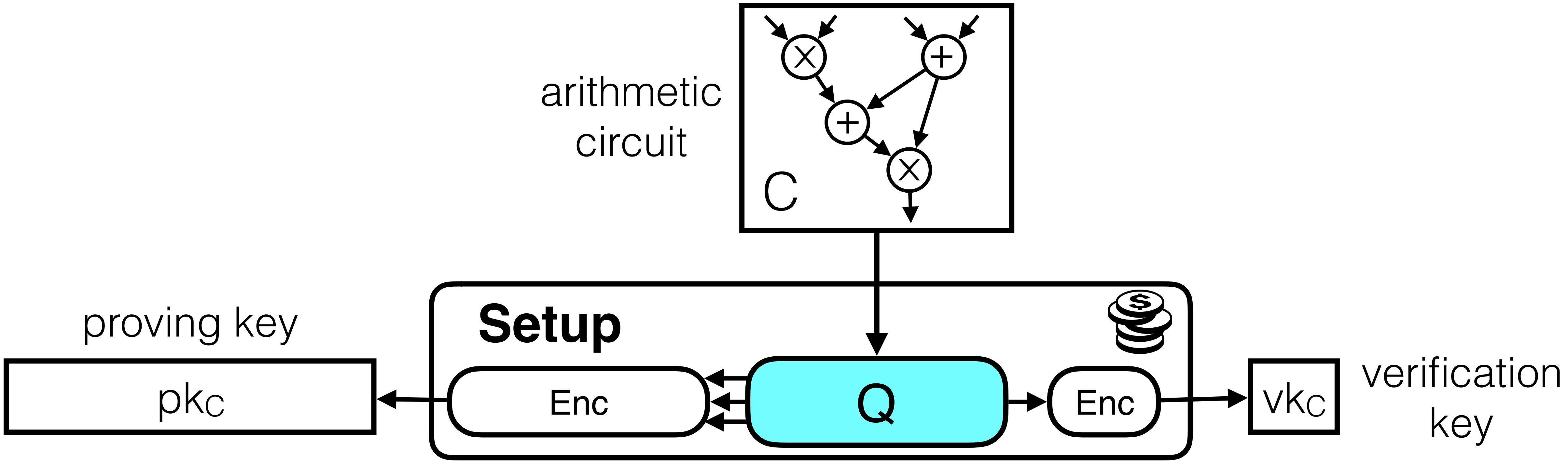
ZK-SNARKs from Linear PCPs



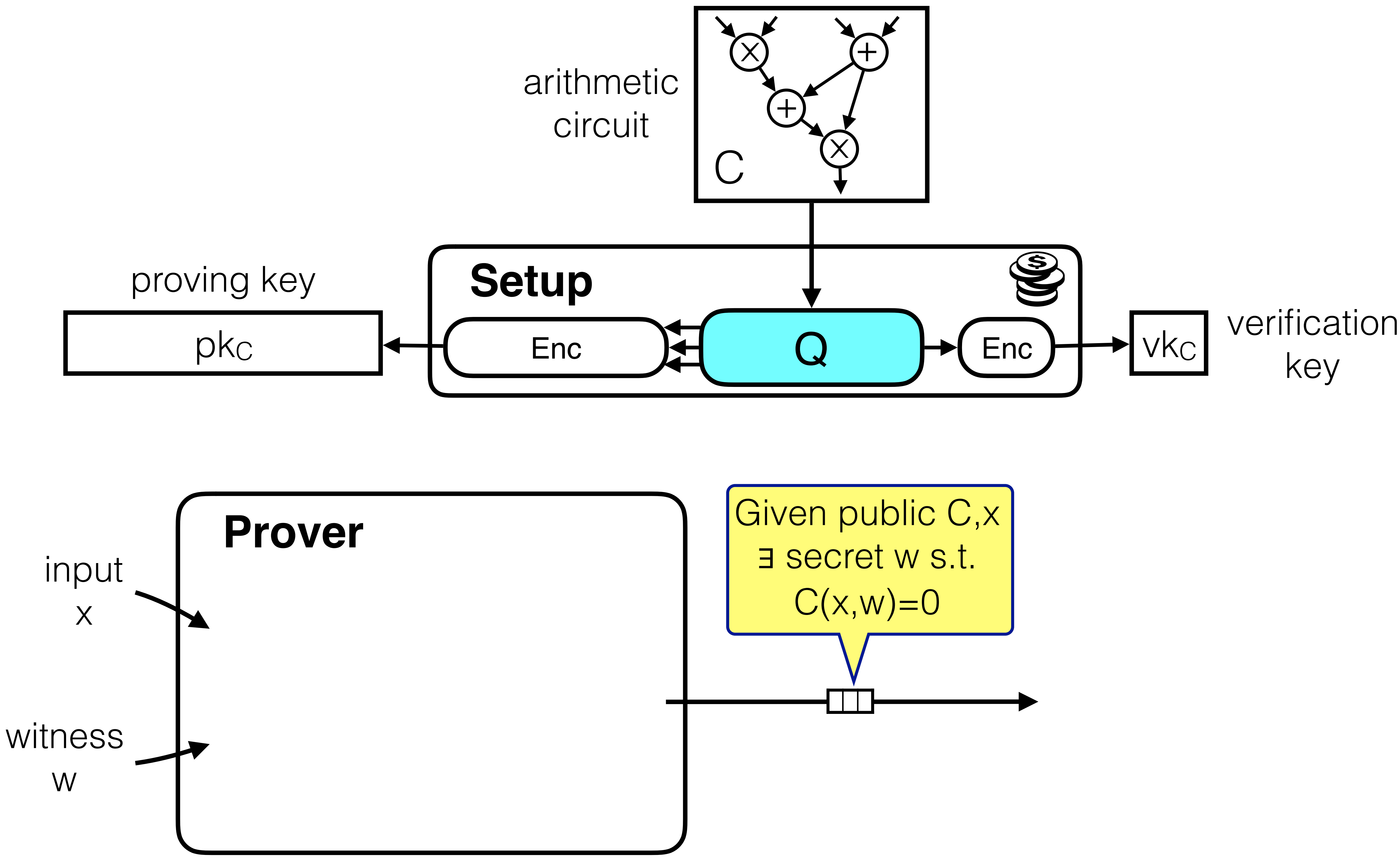
ZK-SNARKs from Linear PCPs



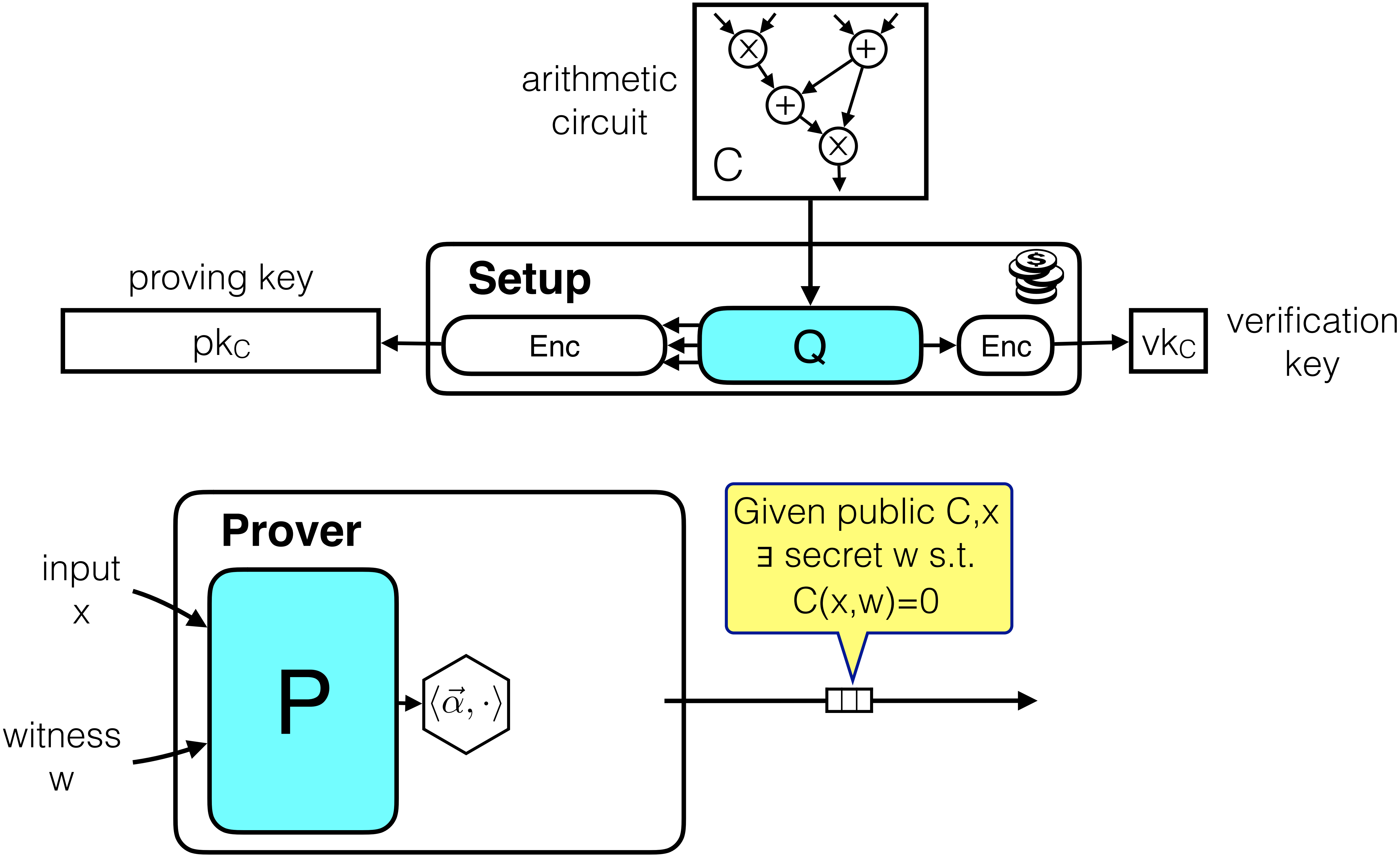
ZK-SNARKs from Linear PCPs



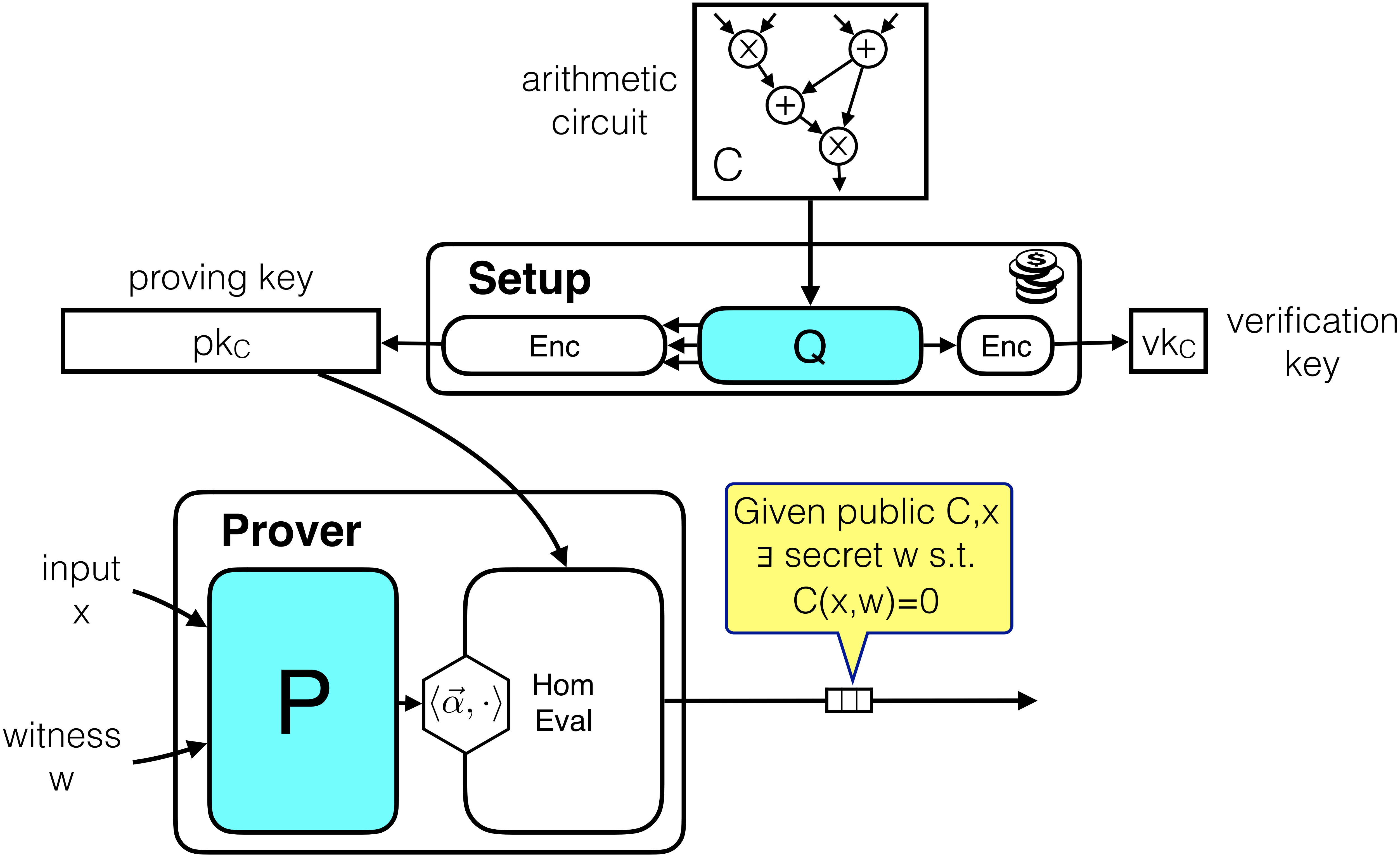
ZK-SNARKs from Linear PCPs



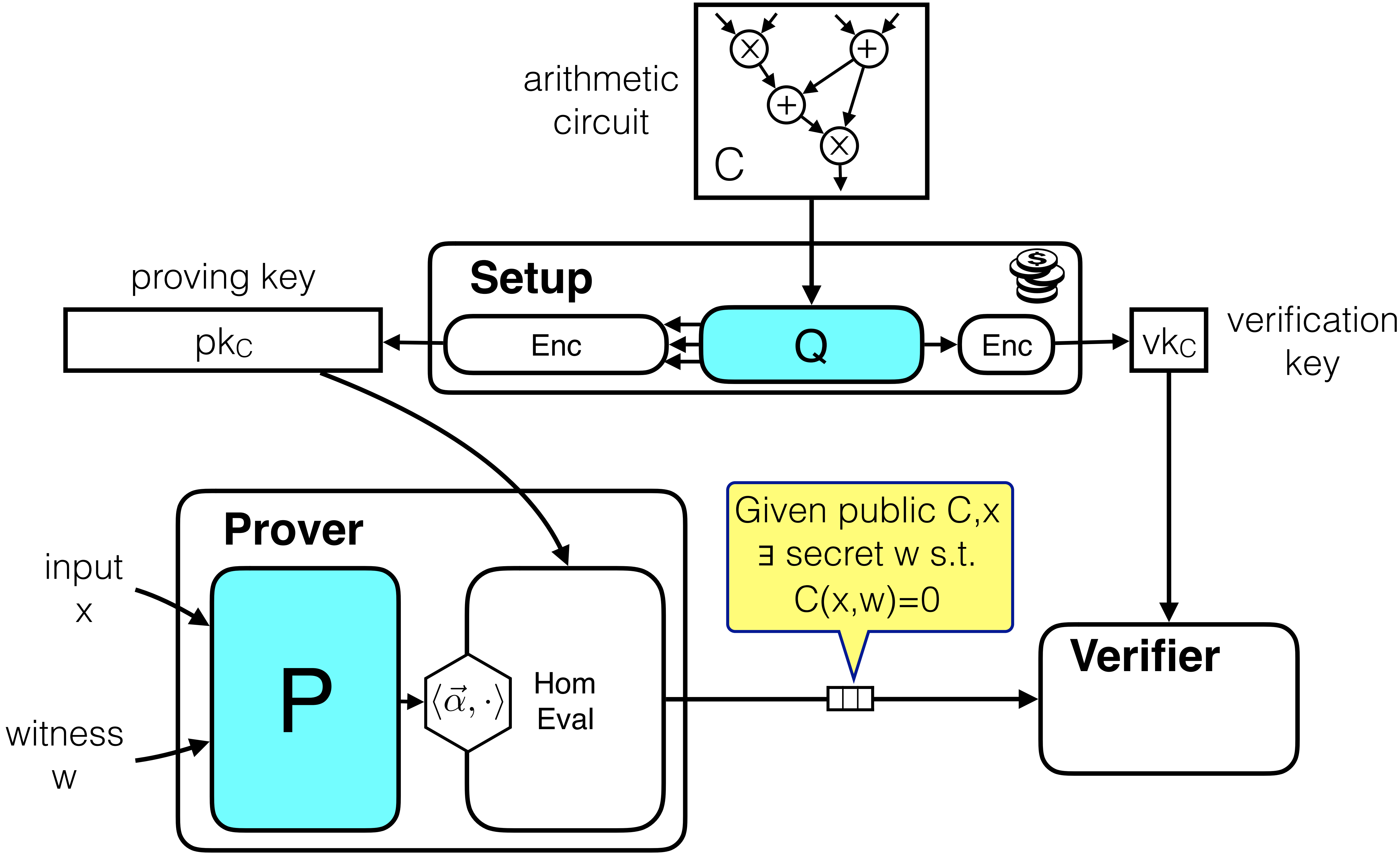
ZK-SNARKs from Linear PCPs



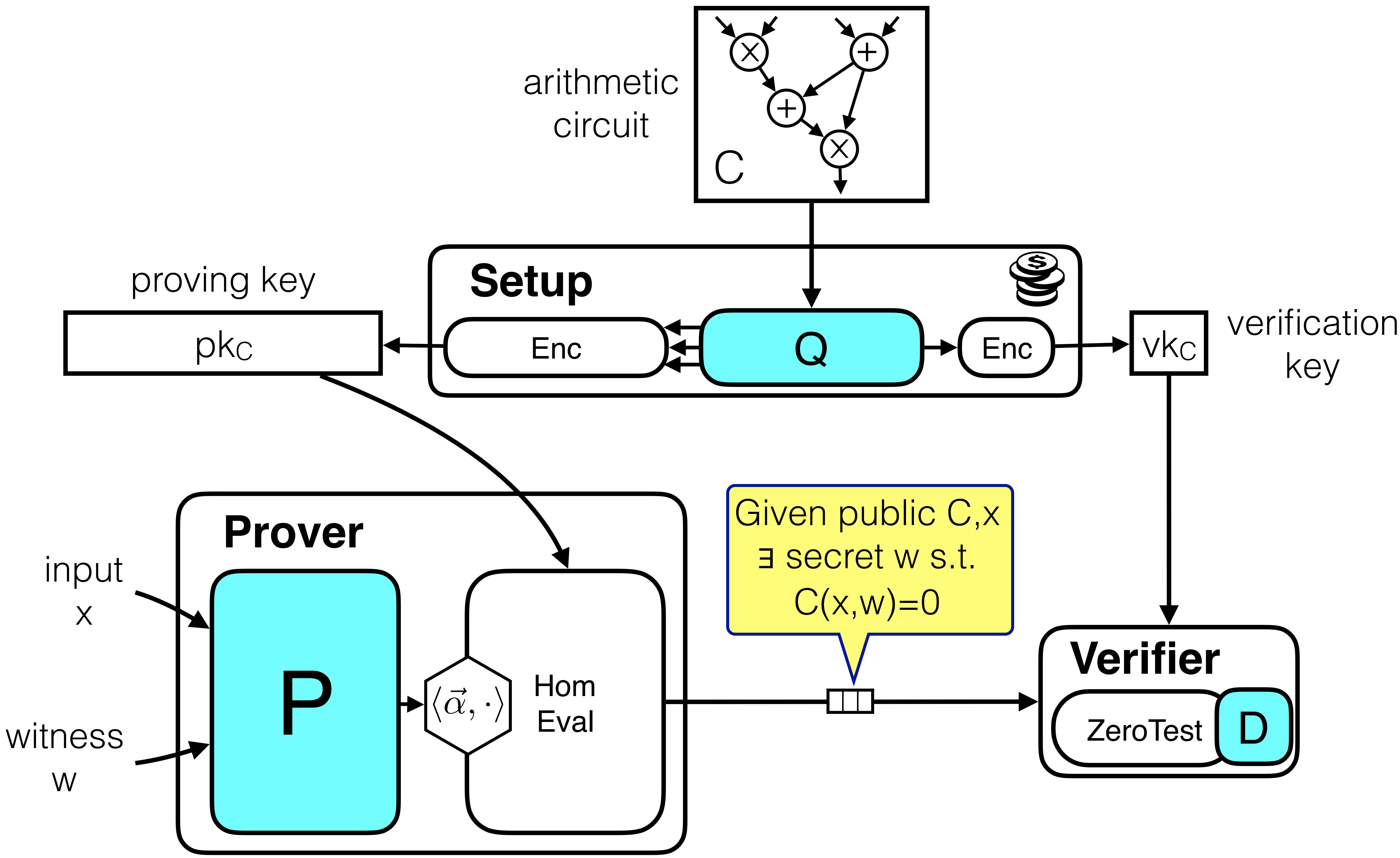
ZK-SNARKs from Linear PCPs



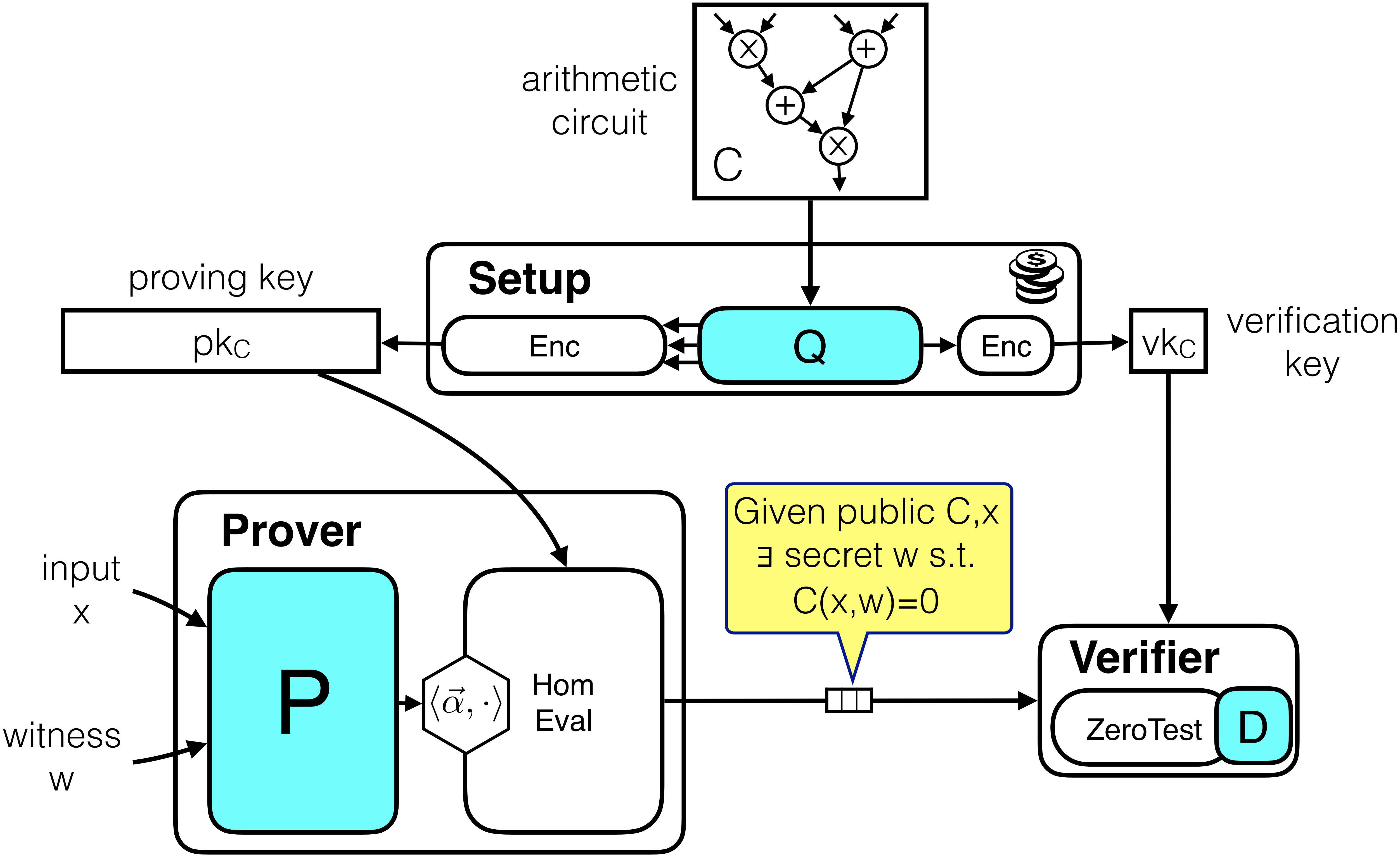
ZK-SNARKs from Linear PCPs



ZK-SNARKs from Linear PCPs

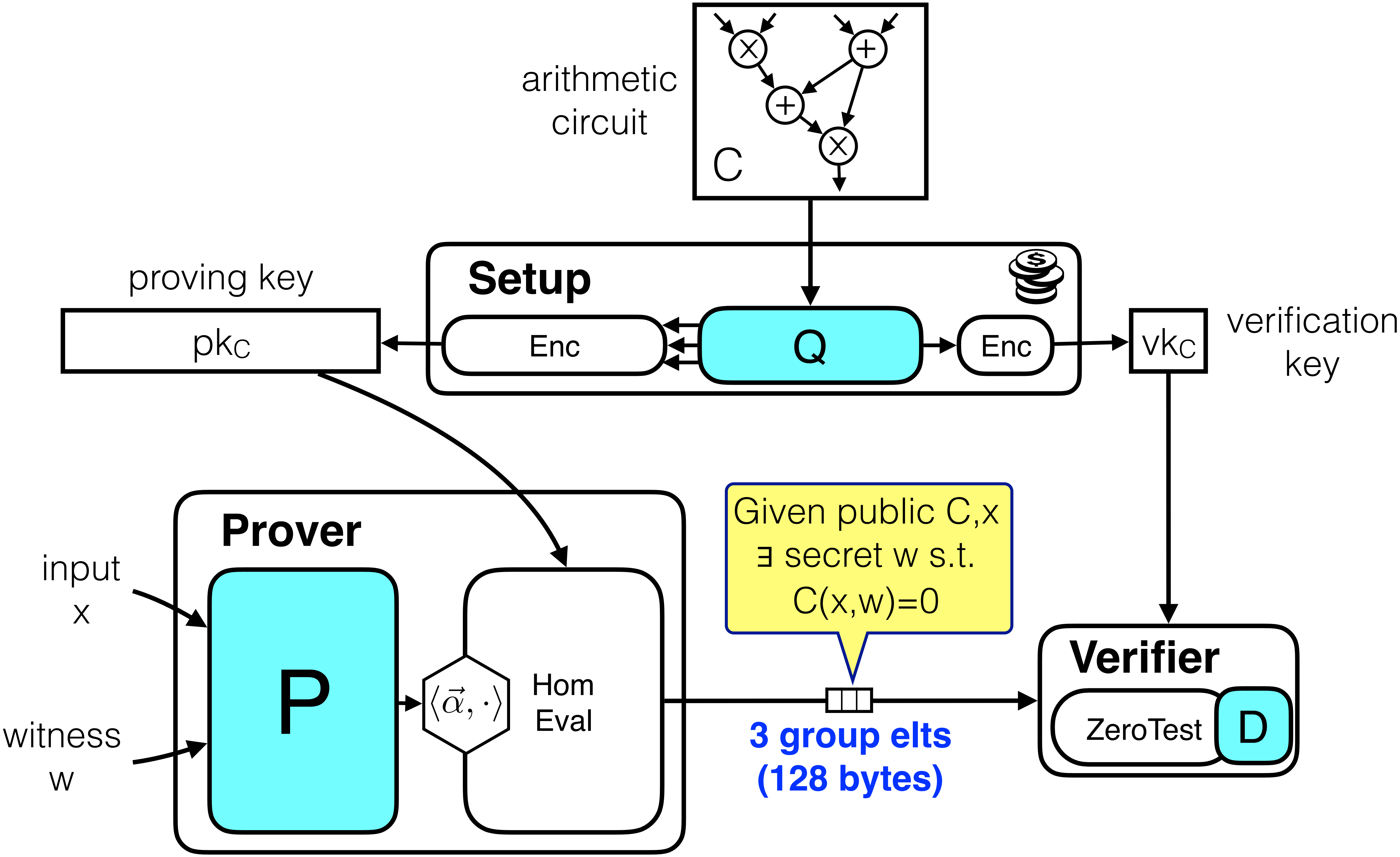


ZK-SNARKs from Linear PCPs



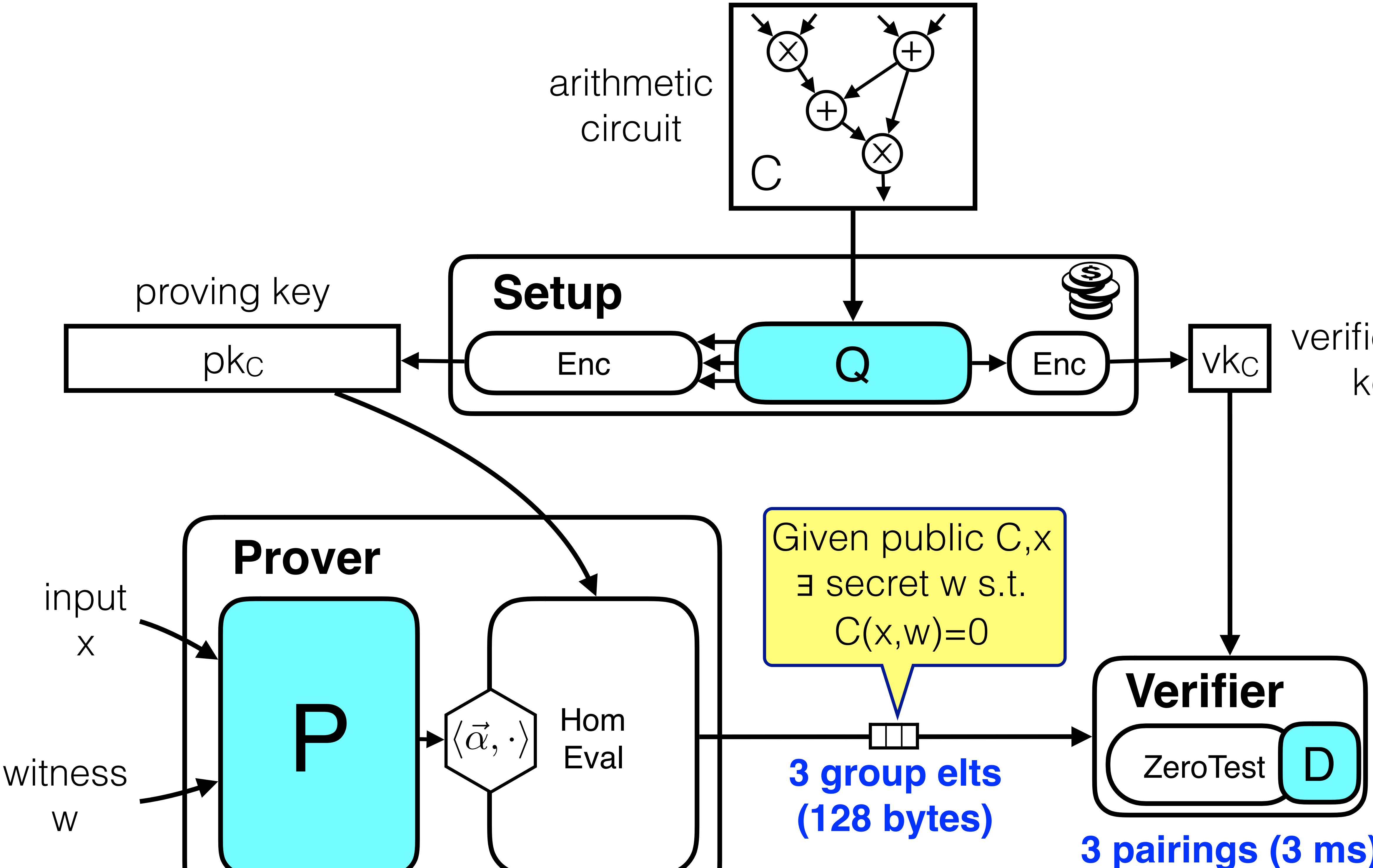
libsnark's implementation of [Groth EUROCRYPT '16]

ZK-SNARKs from Linear PCPs



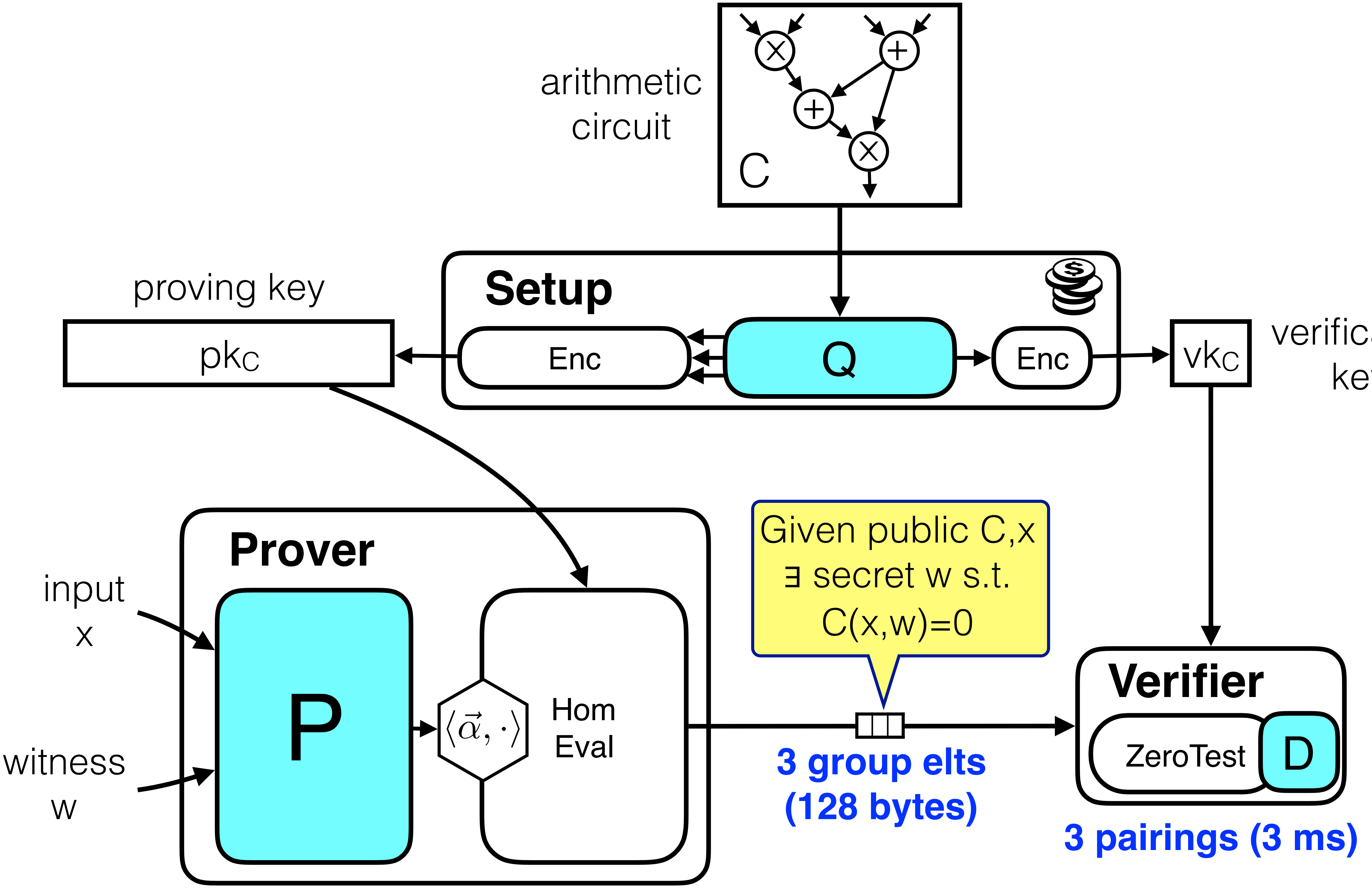
libsnark's implementation of [Groth EUROCRYPT '16]

ZK-SNARKs from Linear PCPs



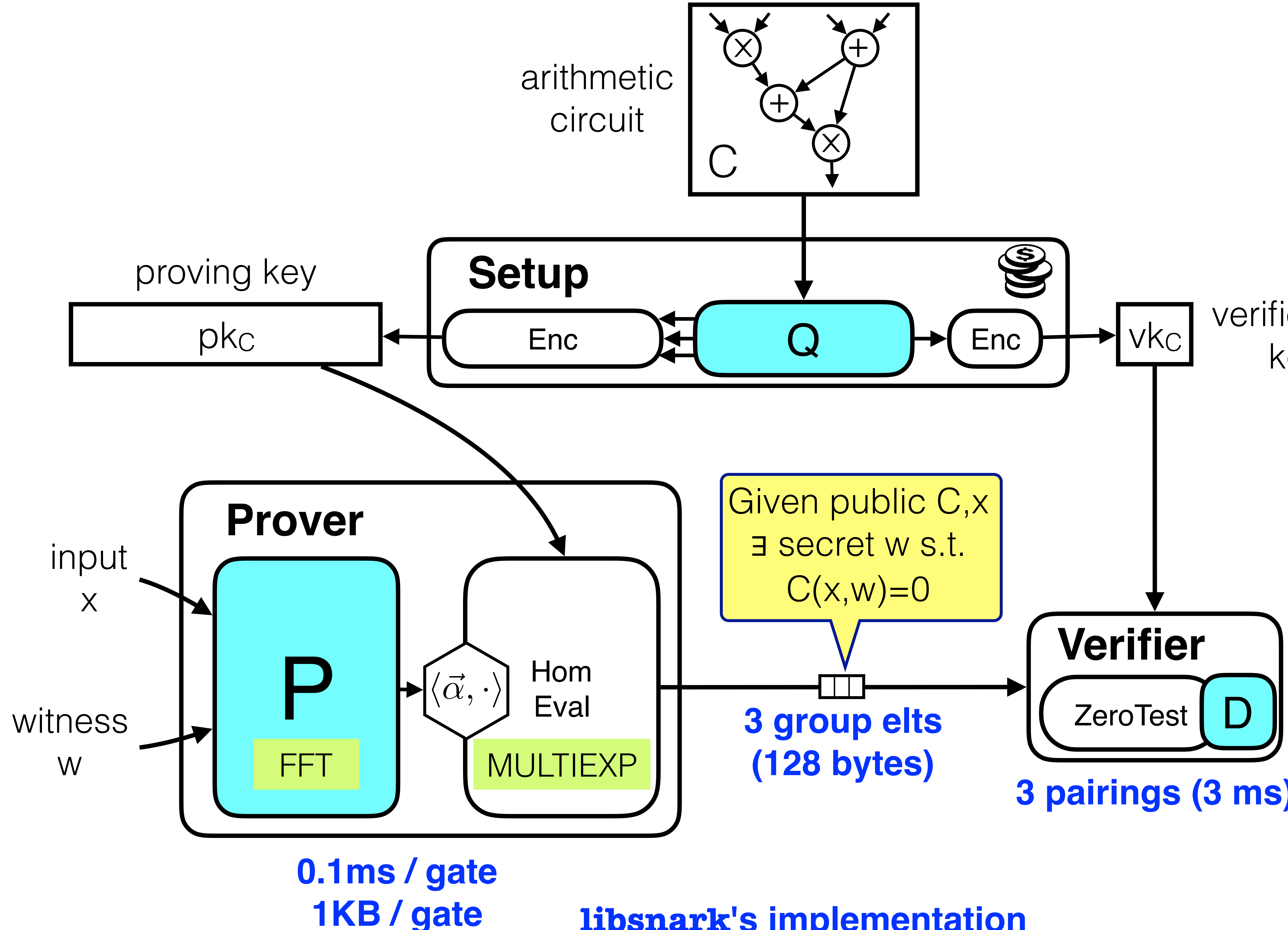
libsnark's implementation
of [Groth EUROCRYPT '16]

ZK-SNARKs from Linear PCPs



libsnark's implementation of [Groth EUROCRYPT '16]

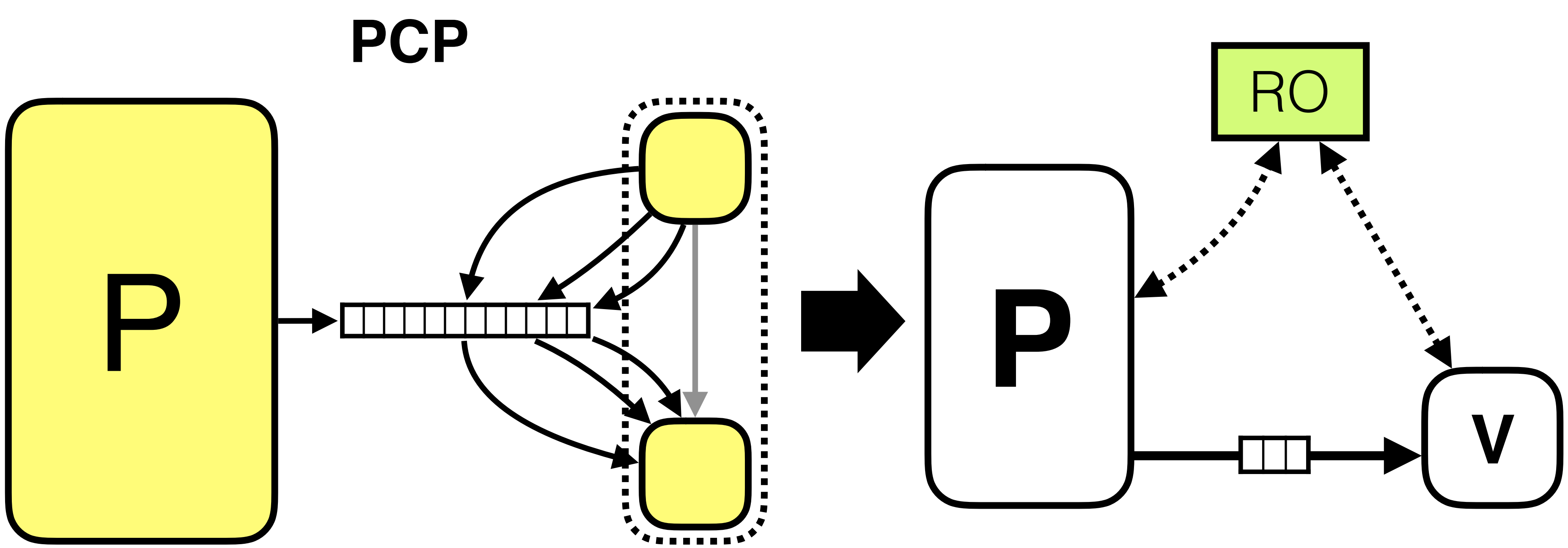
ZK-SNARKs from Linear PCPs

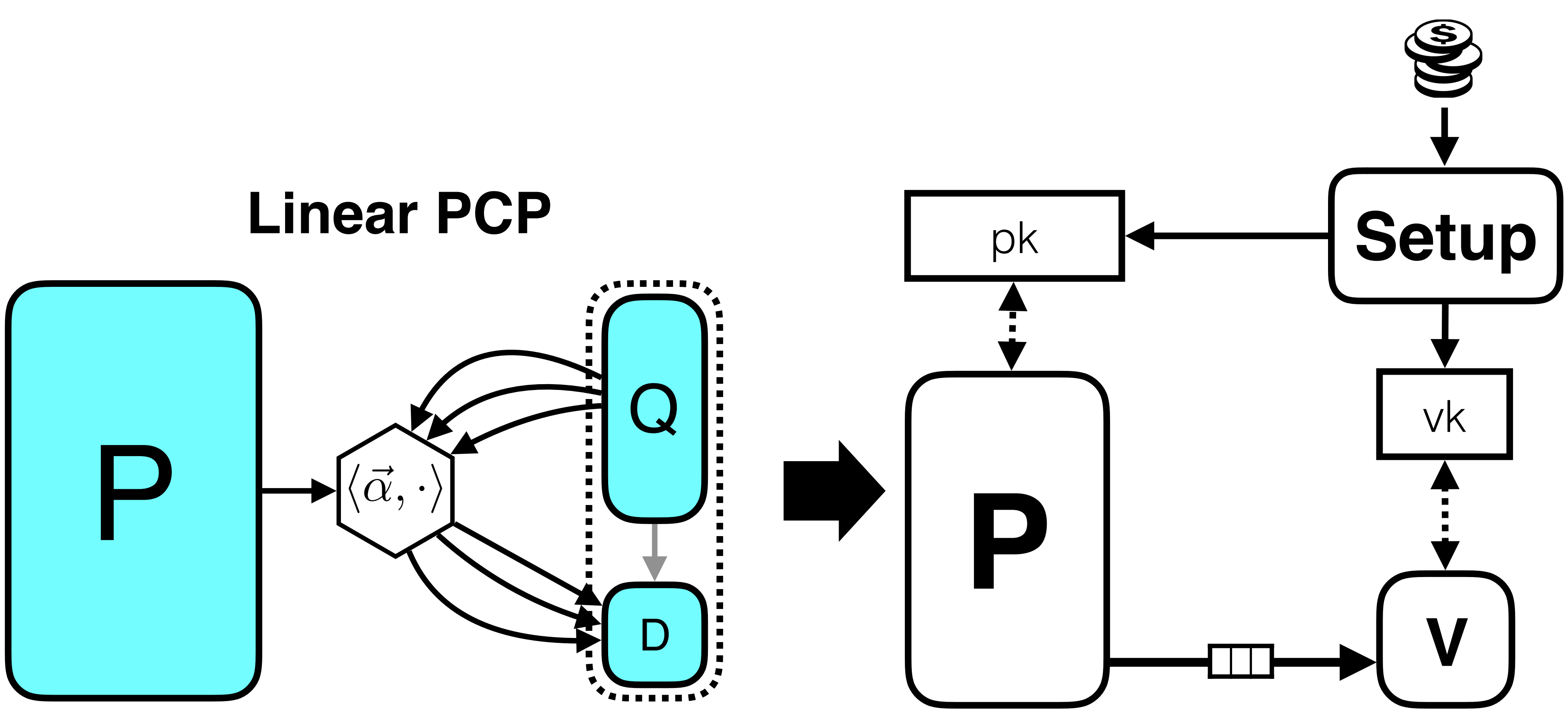


libsnark's implementation of [Groth EUROCRYPT '16]

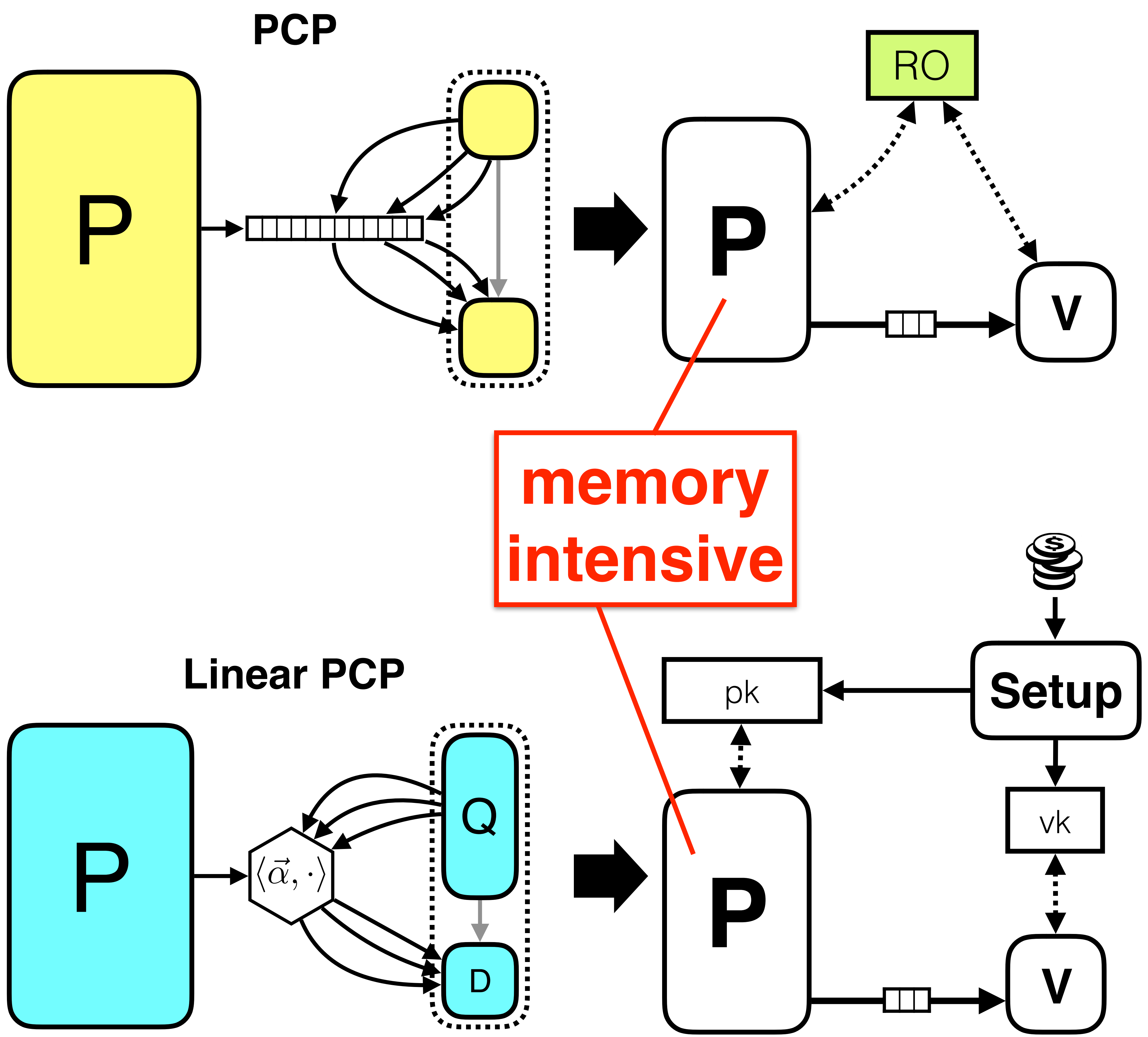
Which approach is better?

Which approach is better?

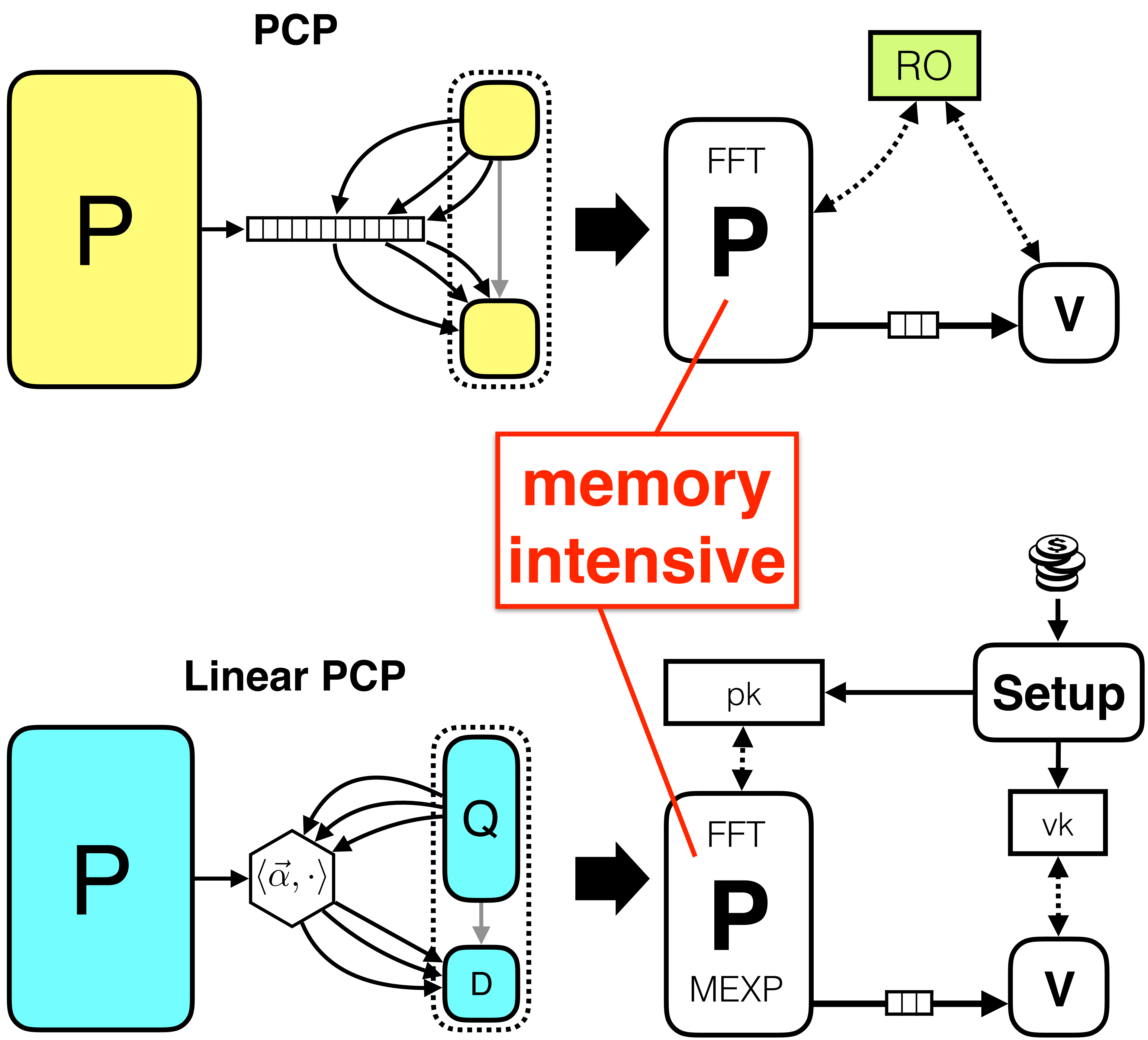




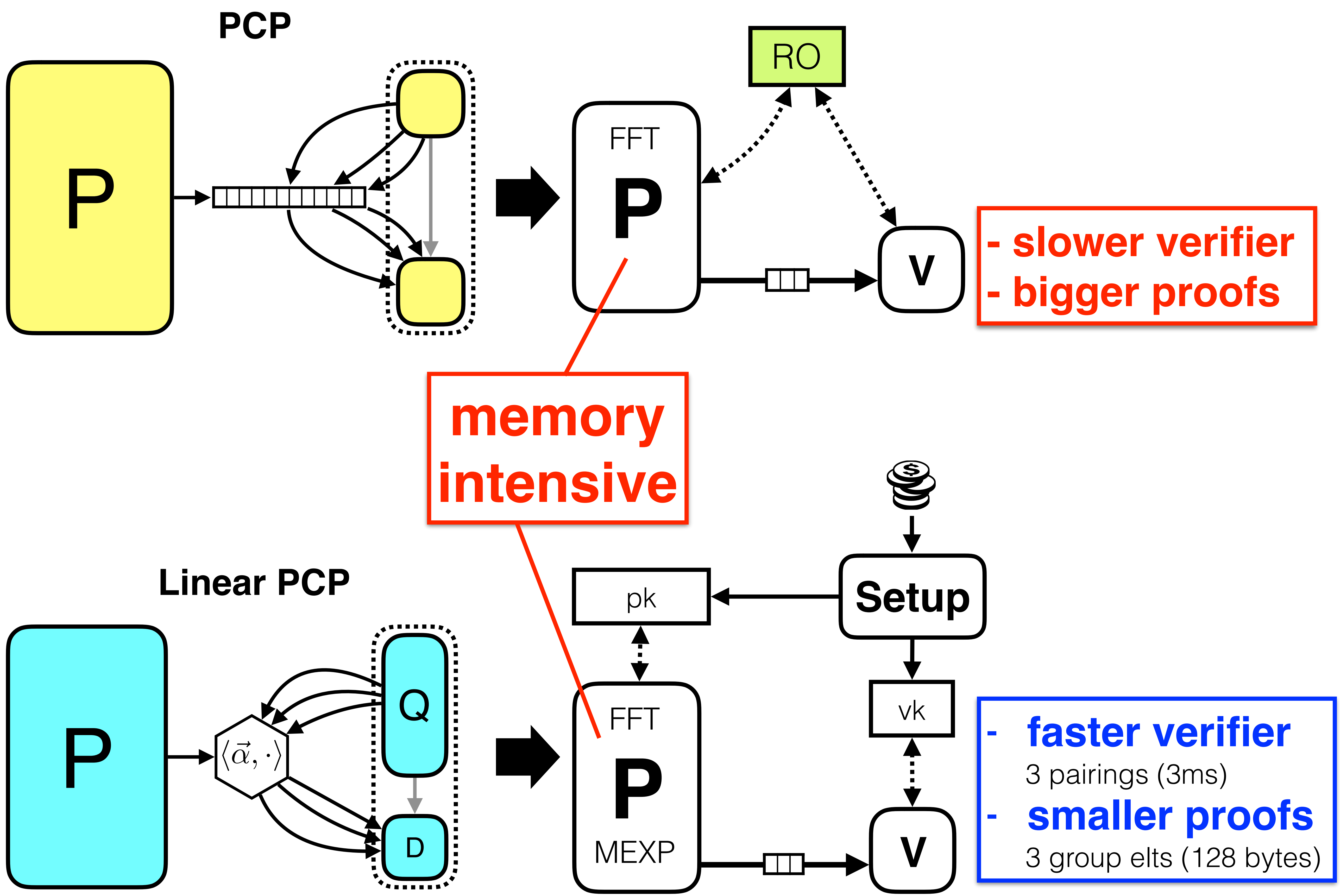
Which approach is better?



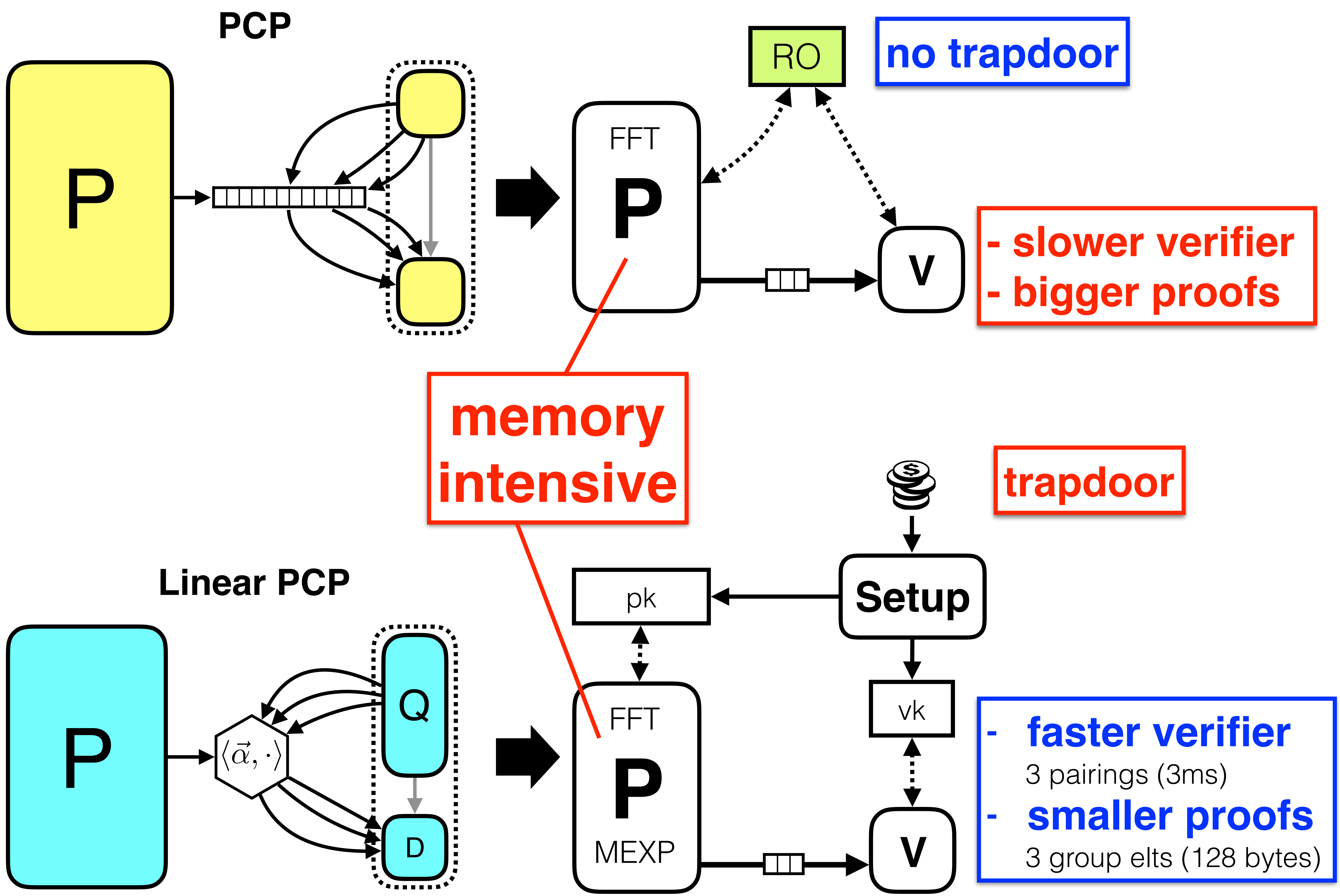
Which approach is better?



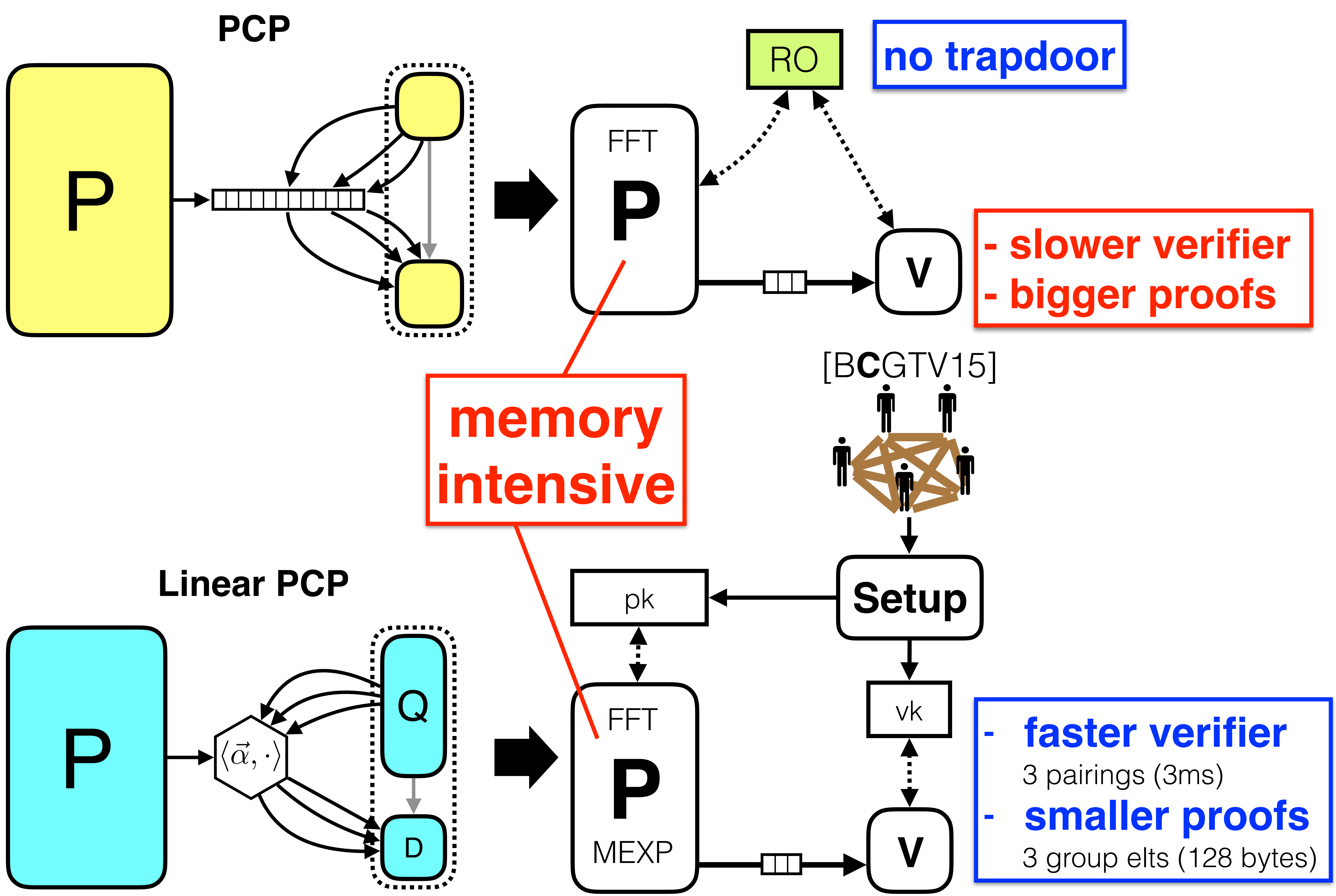
Which approach is better?



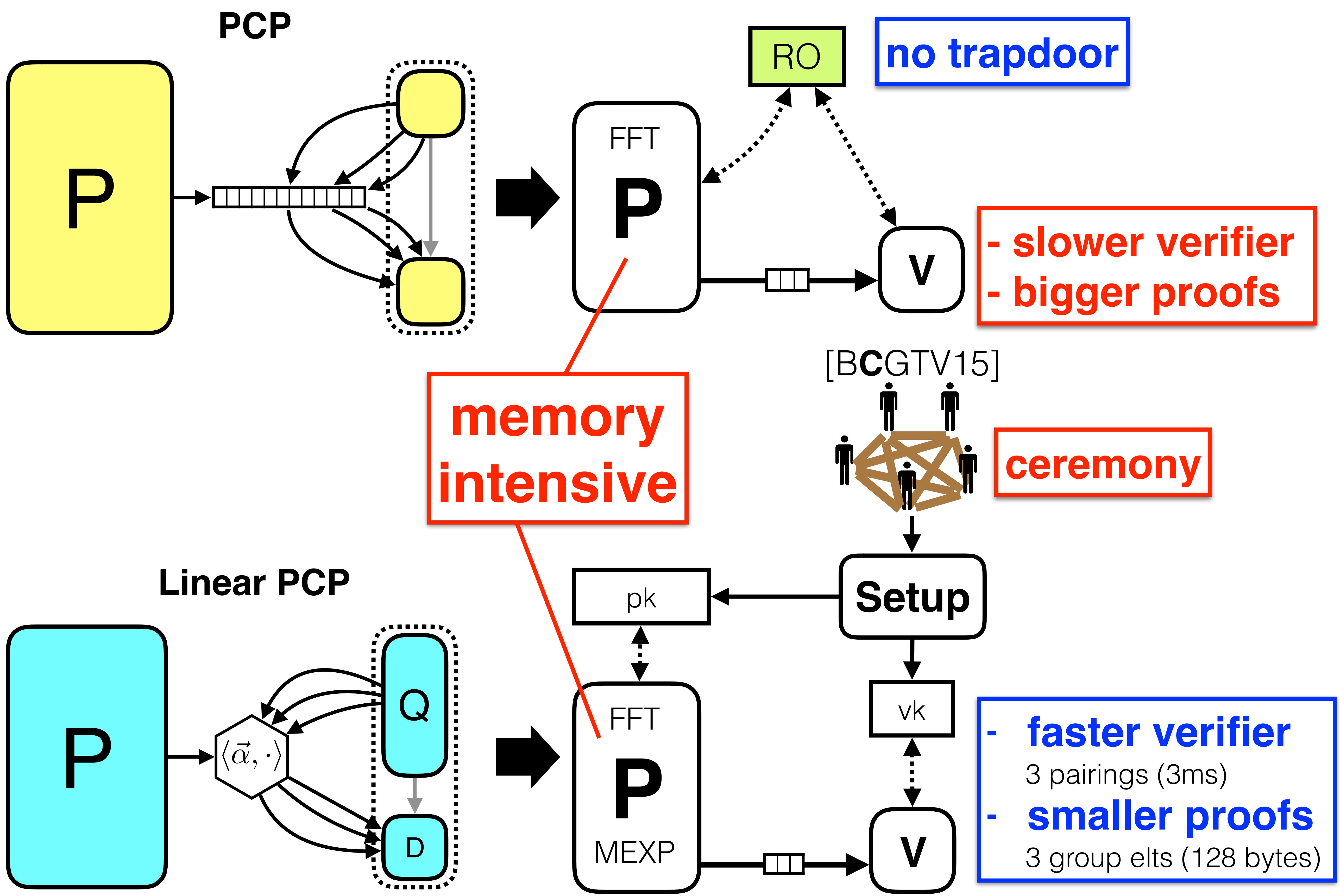
Which approach is better?



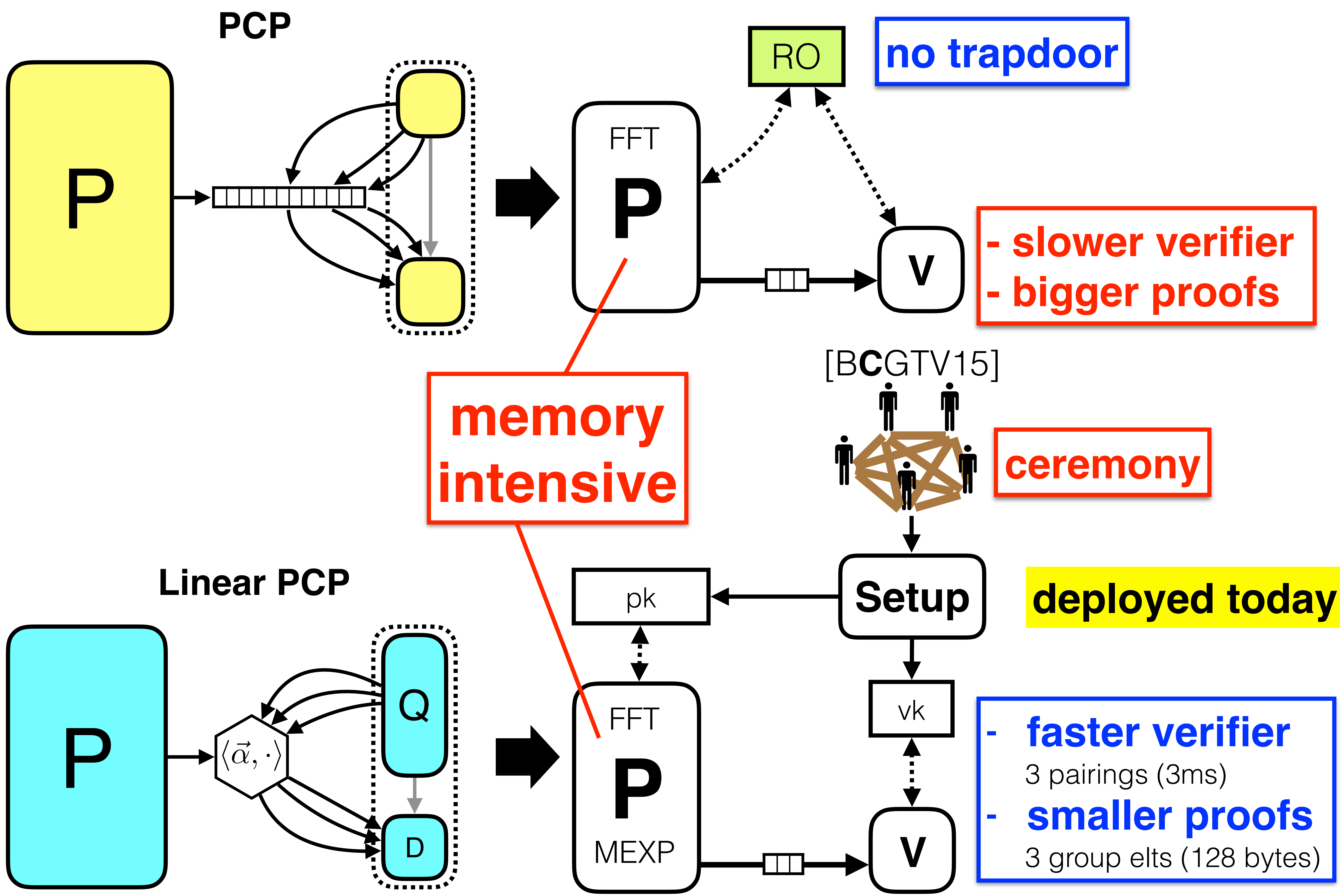
Which approach is better?



Which approach is better?



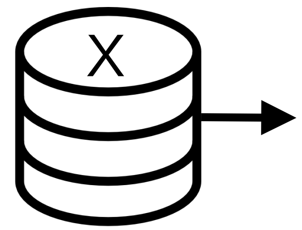
Which approach is better?



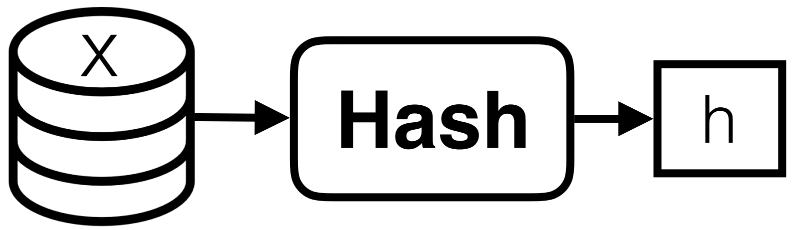
Frontiers

Authenticated Inputs

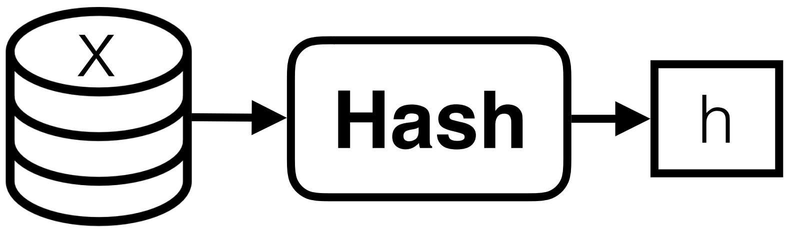
Authenticated Inputs

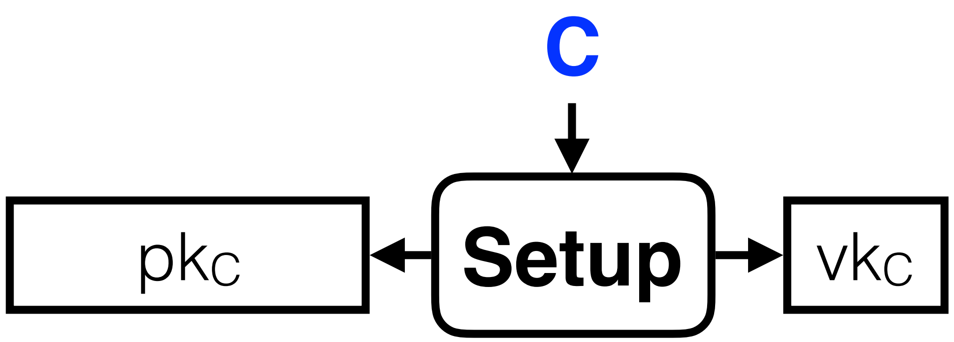


Authenticated Inputs

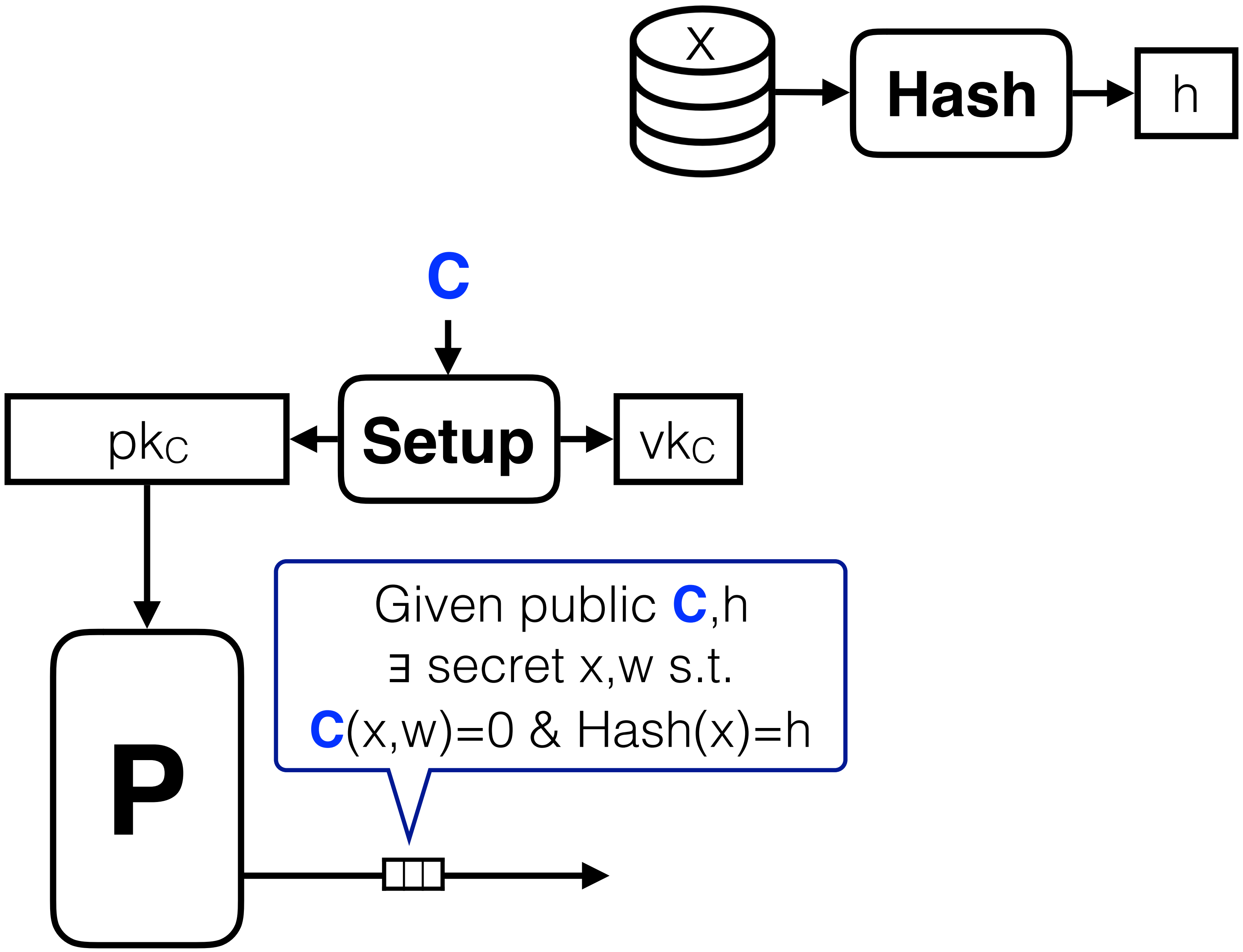


Authenticated Inputs

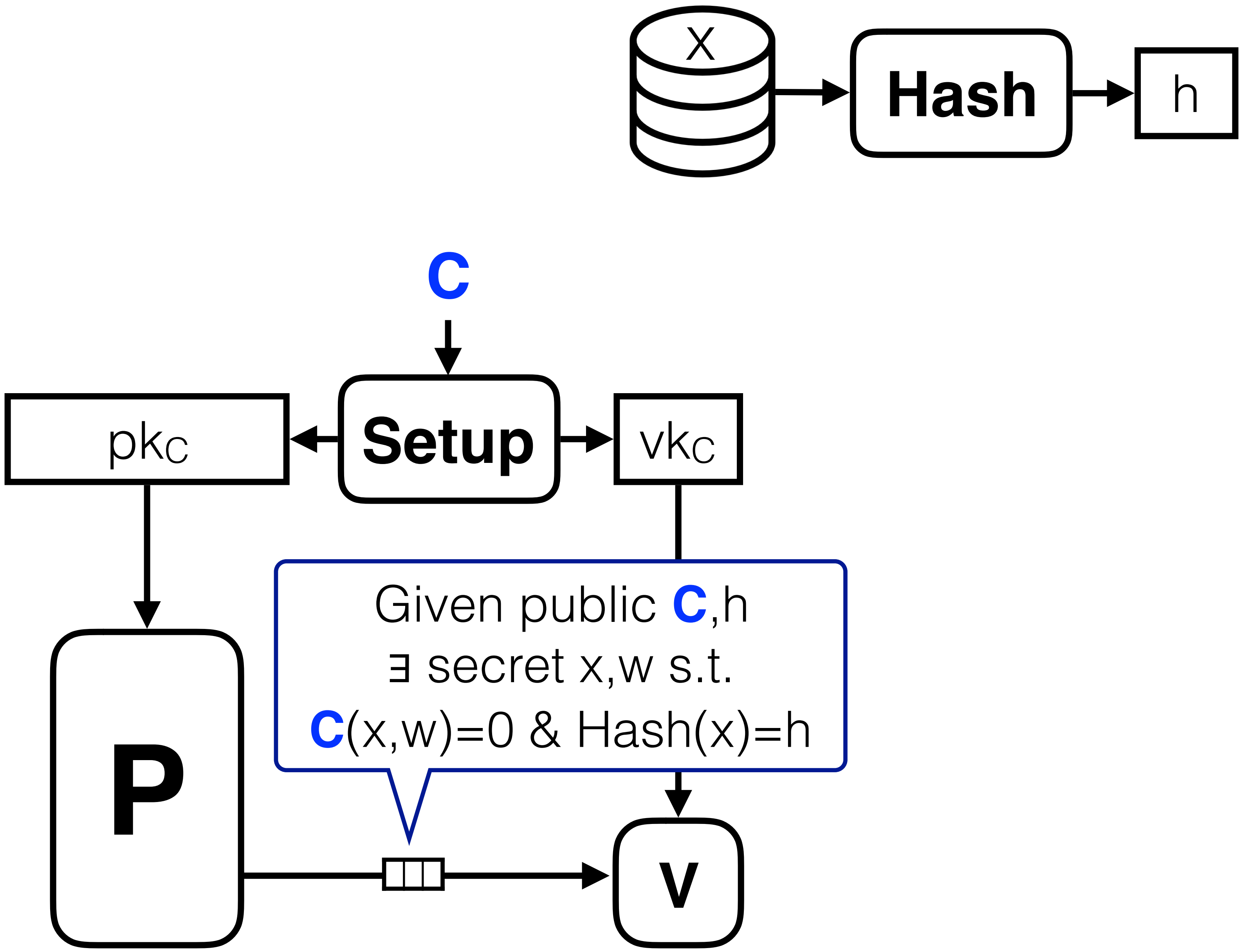




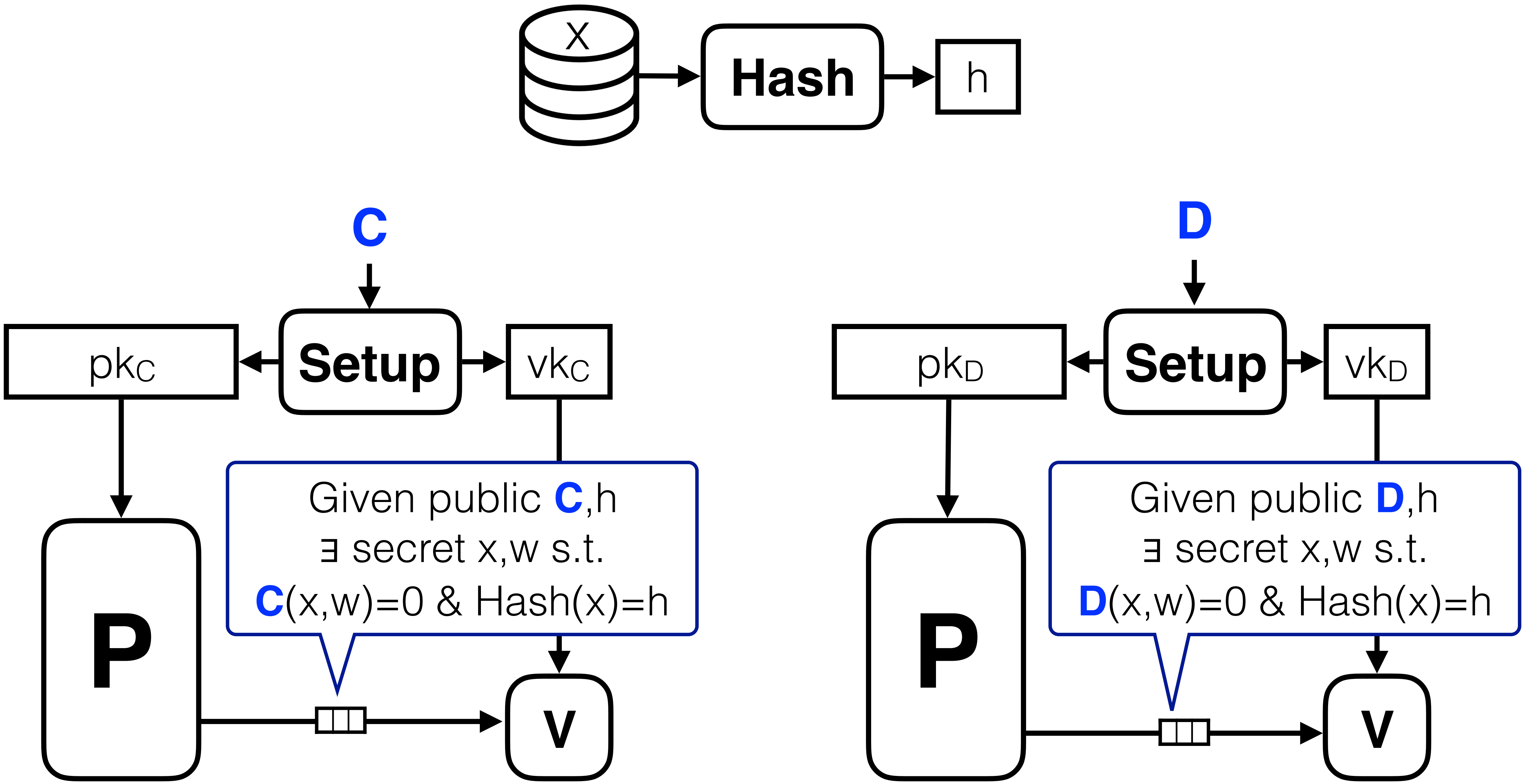
Authenticated Inputs



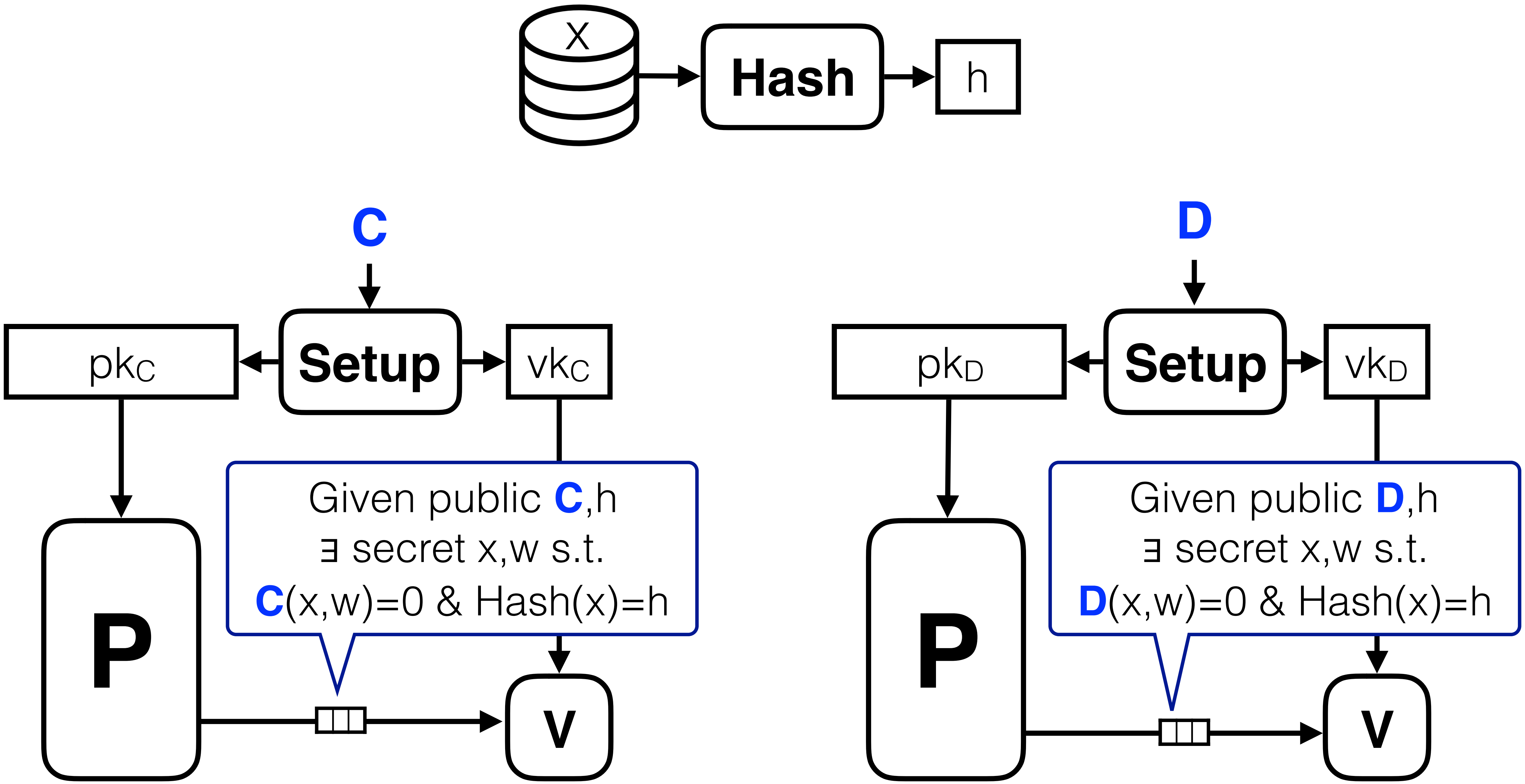
Authenticated Inputs



Authenticated Inputs

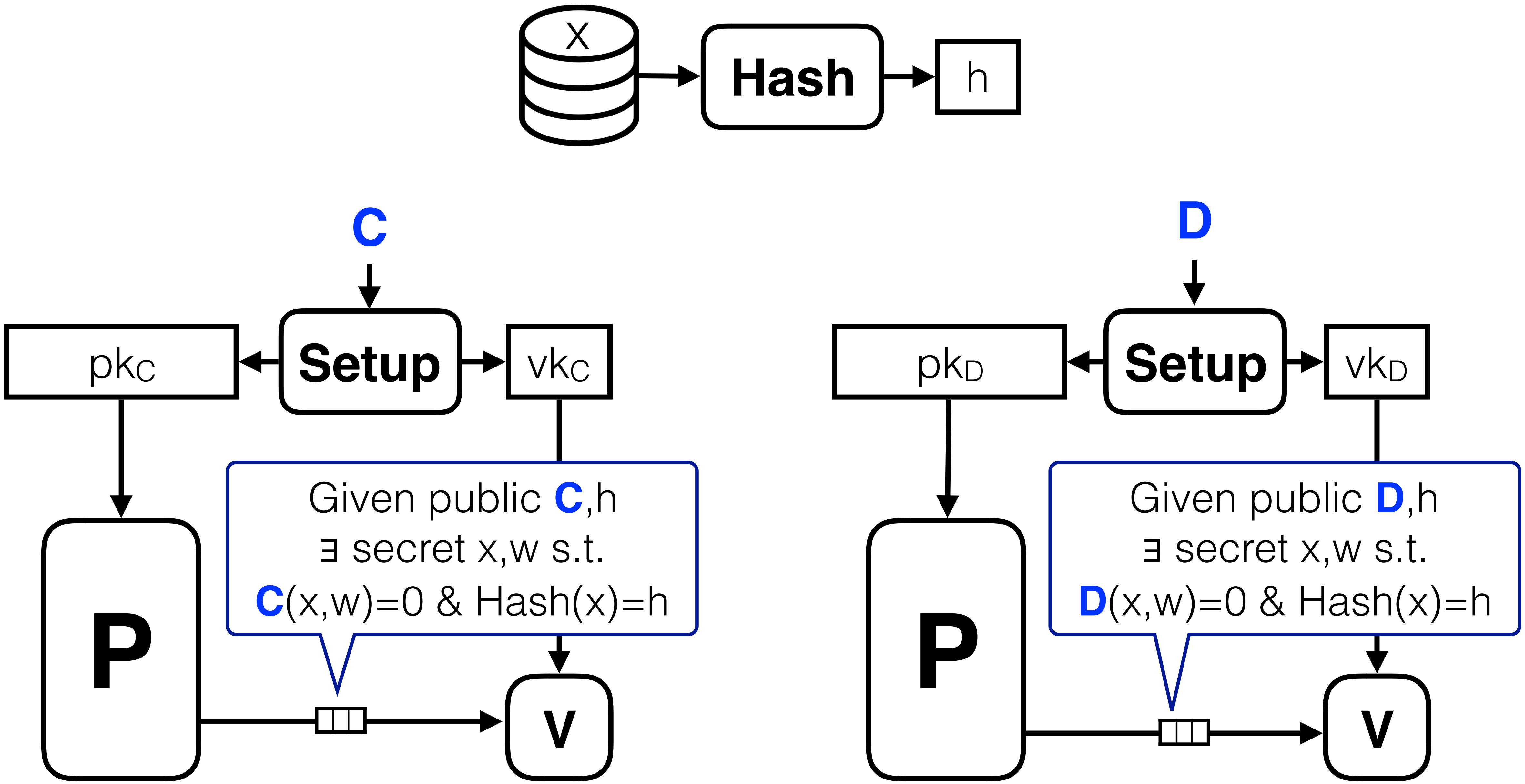


Authenticated Inputs



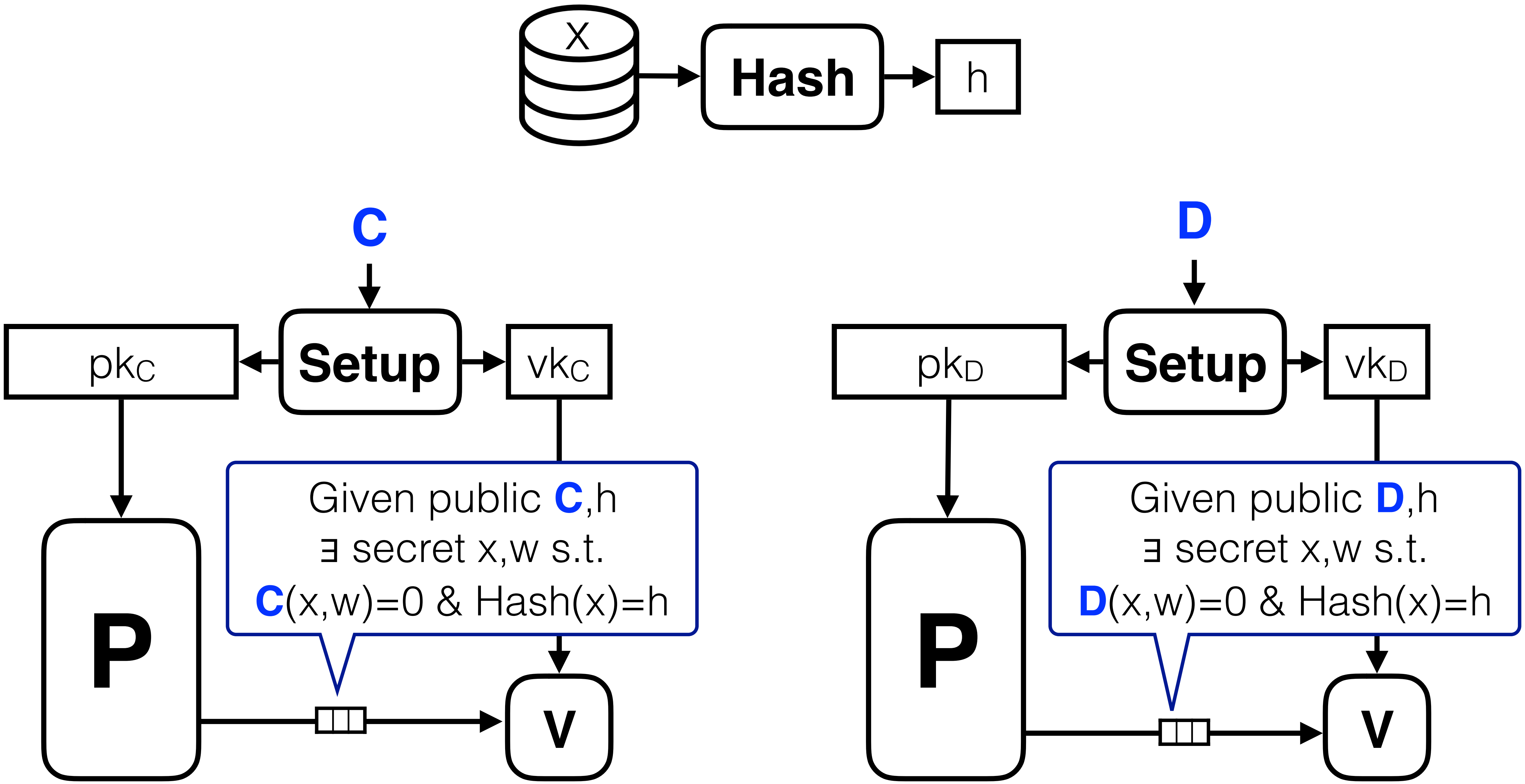
- generic uses of ZK-SNARKs are expensive

Authenticated Inputs



- generic uses of ZK-SNARKs are expensive
- **better:** co-design Hash and SNARK [FFGKOP CCS '16]

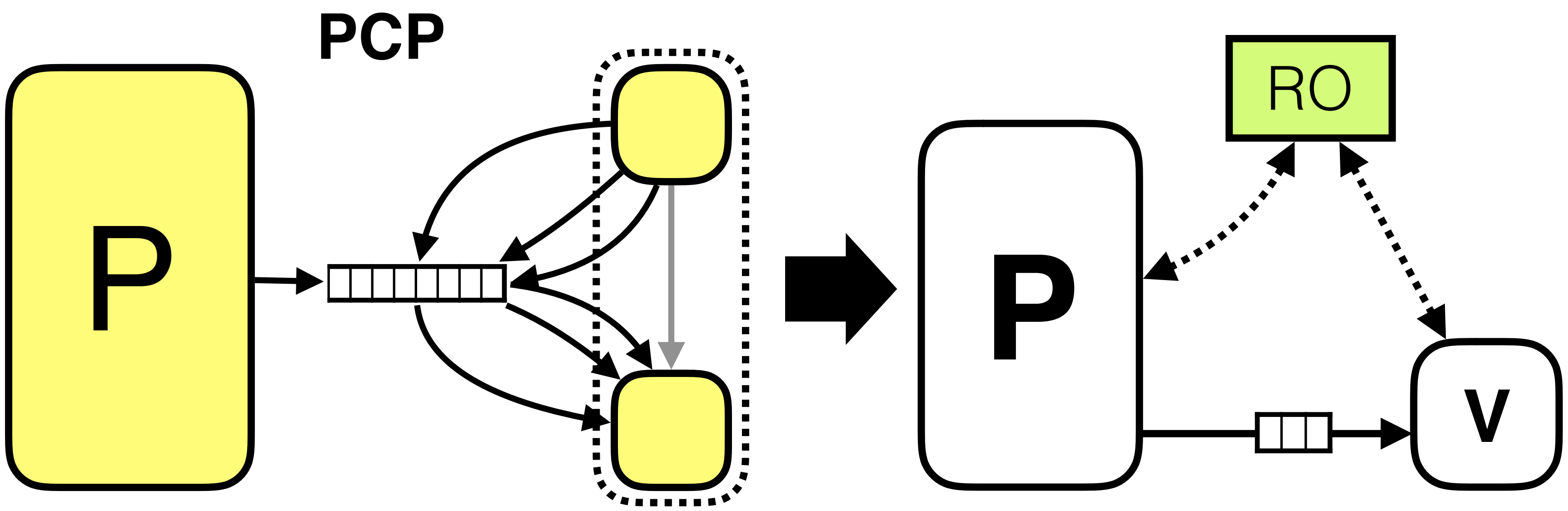
Authenticated Inputs



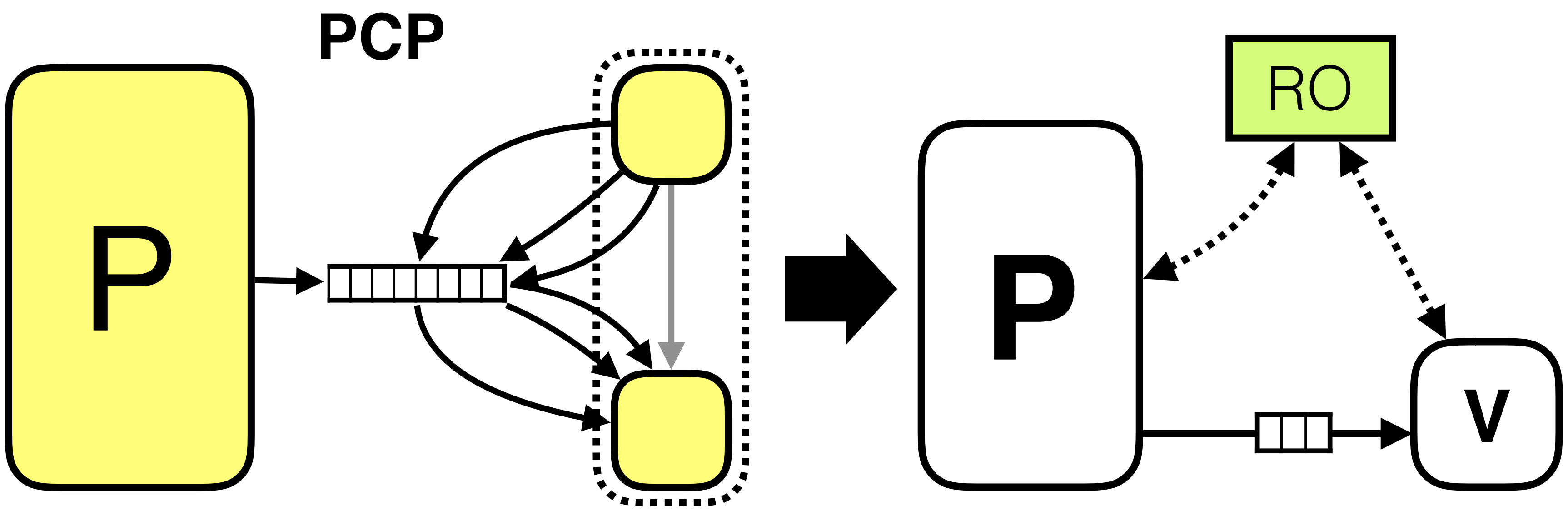
- generic uses of ZK-SNARKs are expensive
- **better:** co-design Hash and SNARK [FFGKOP CCS '16]
- **open:** preserve ZK?

Post-Quantum Security

Post-Quantum Security

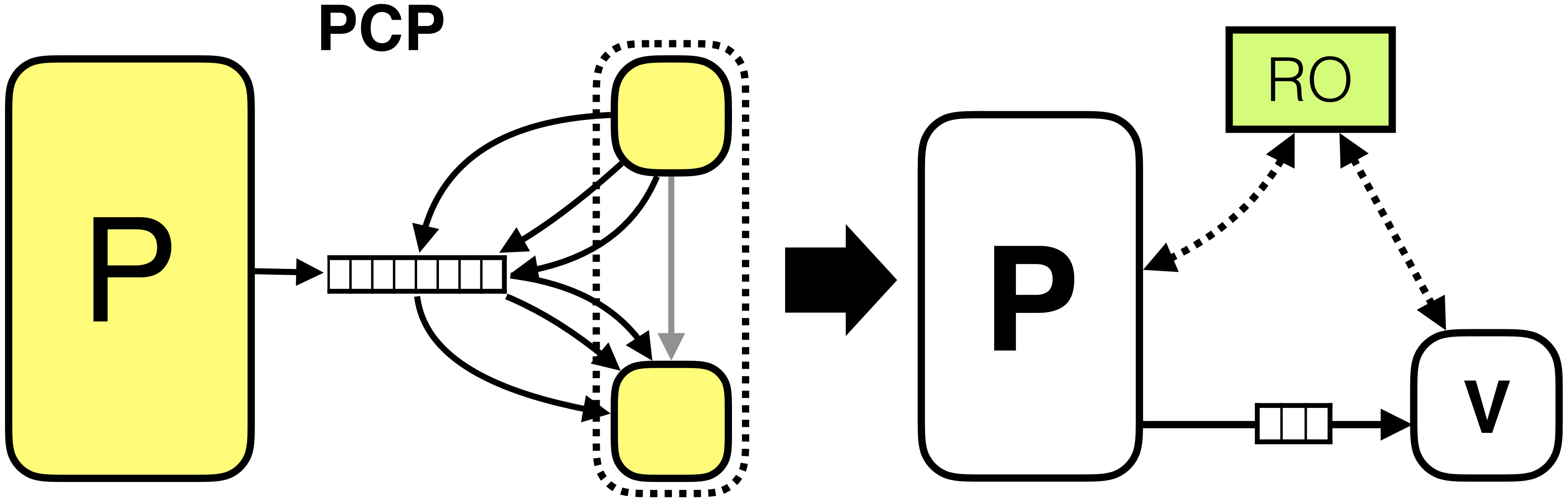


Post-Quantum Security

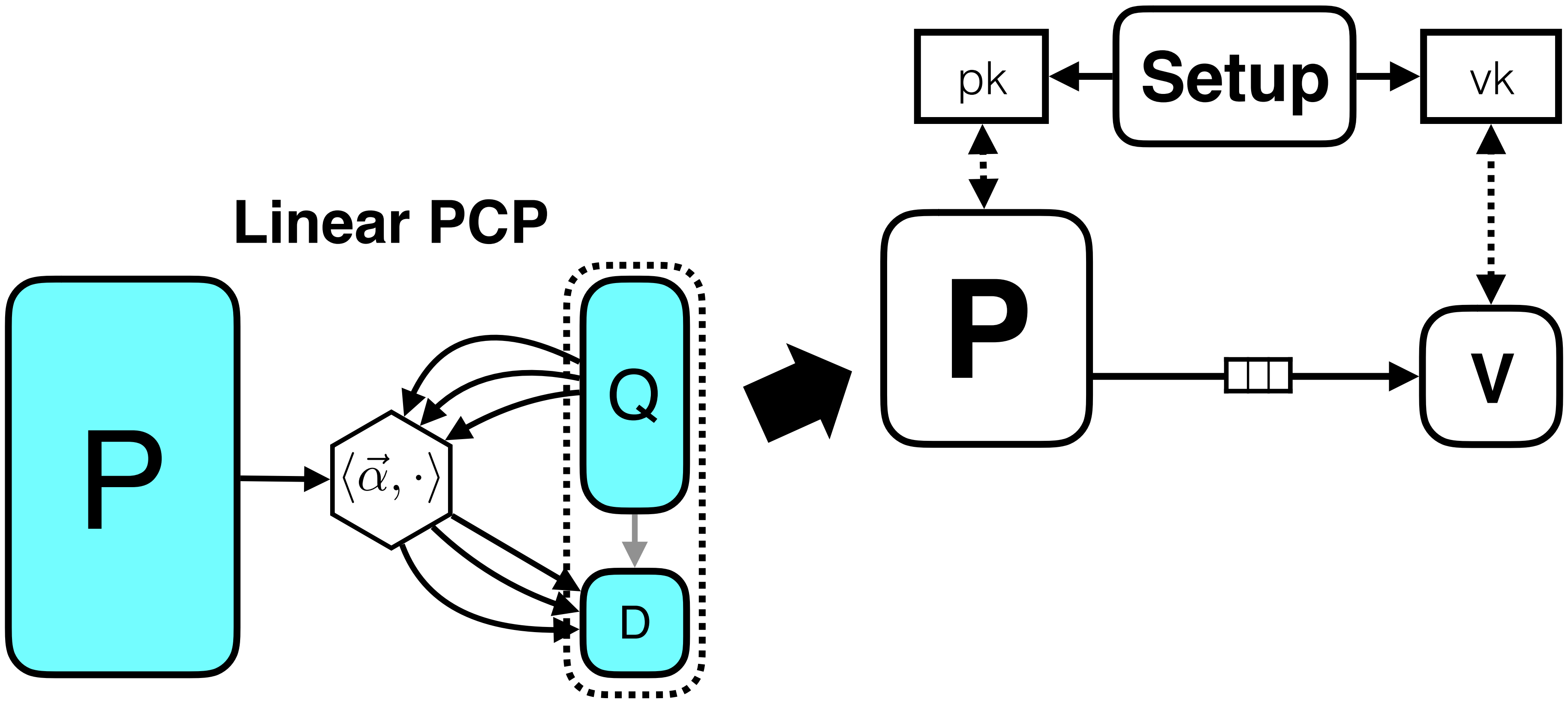


Looks solid.

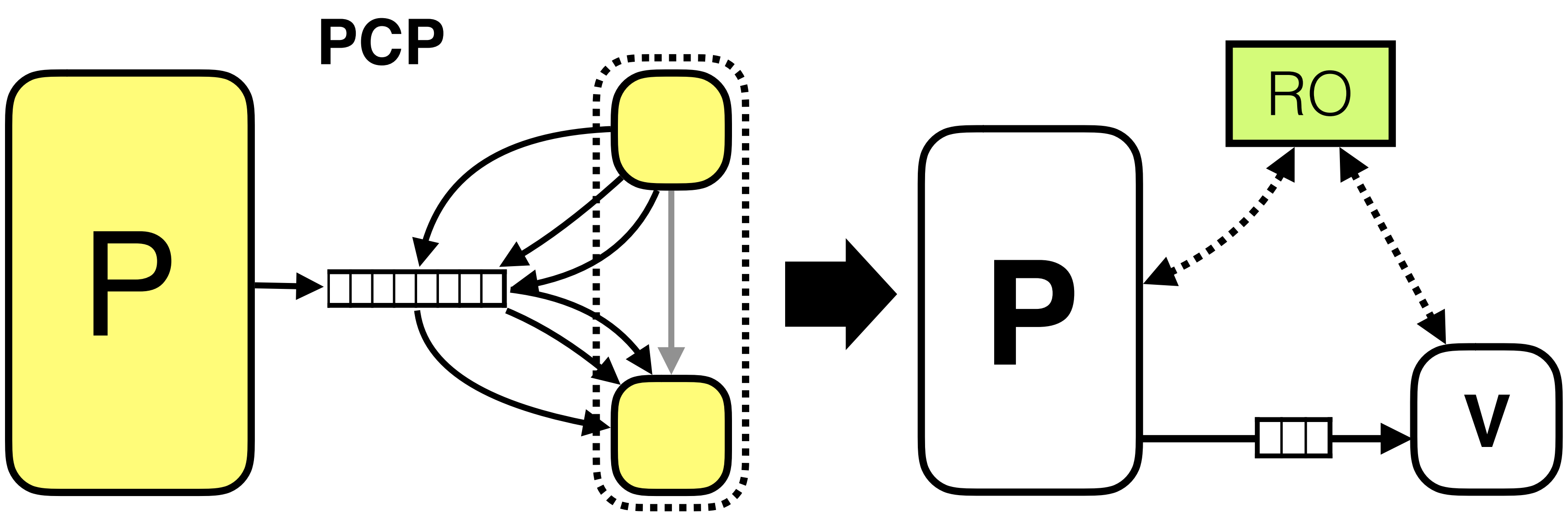
Post-Quantum Security



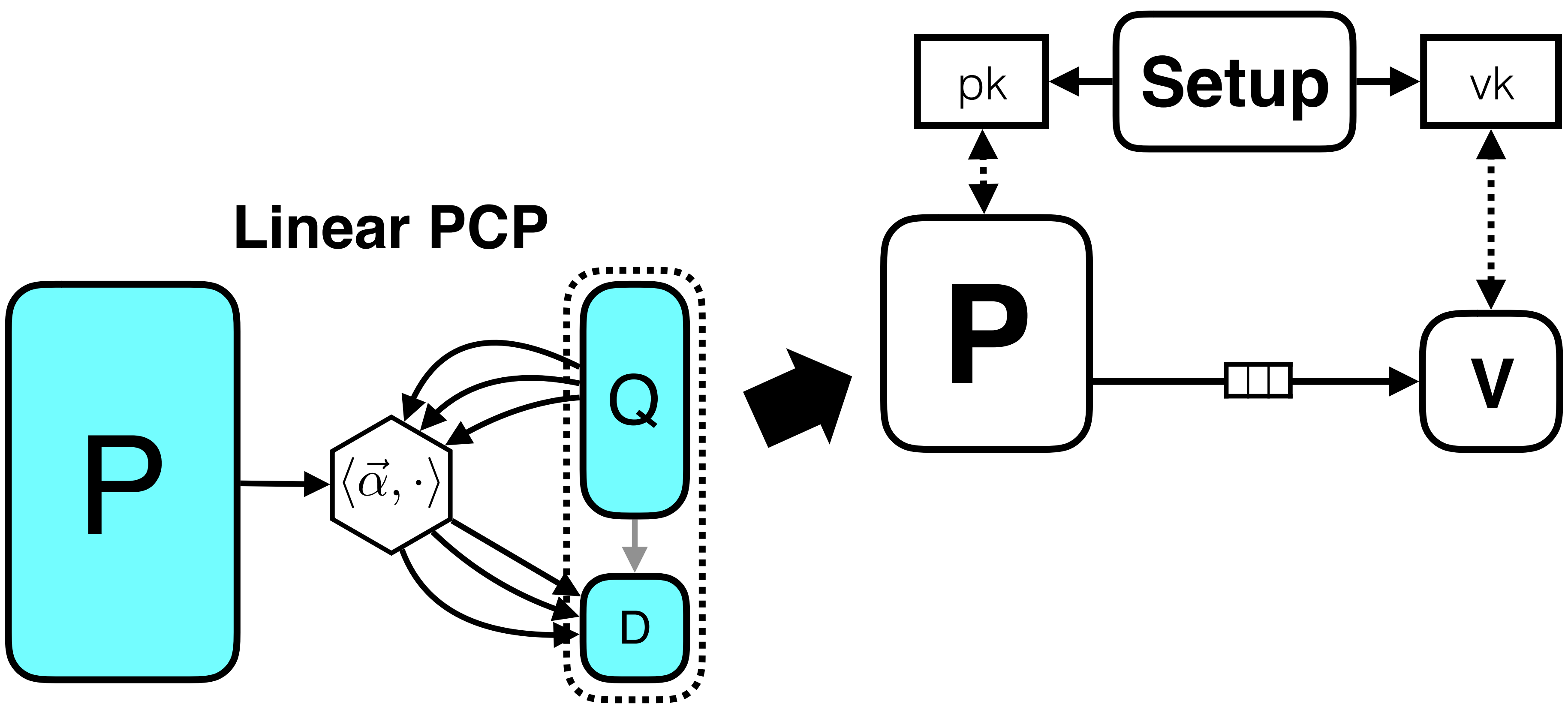
Looks solid.



Post-Quantum Security

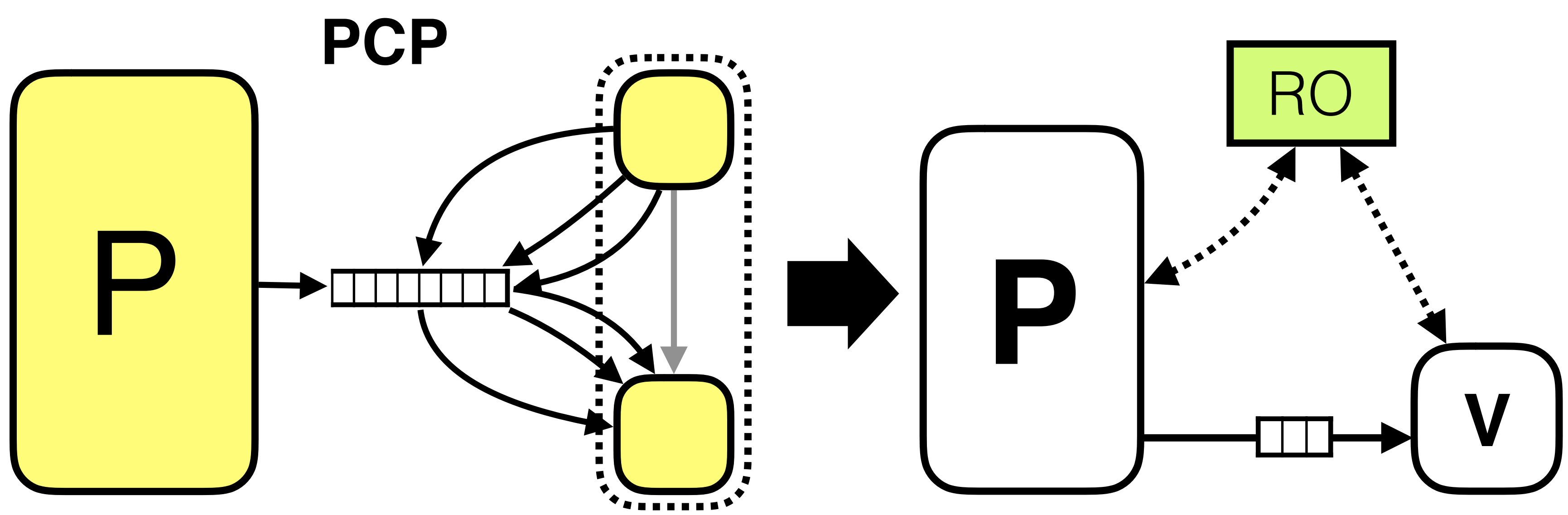


Looks solid.

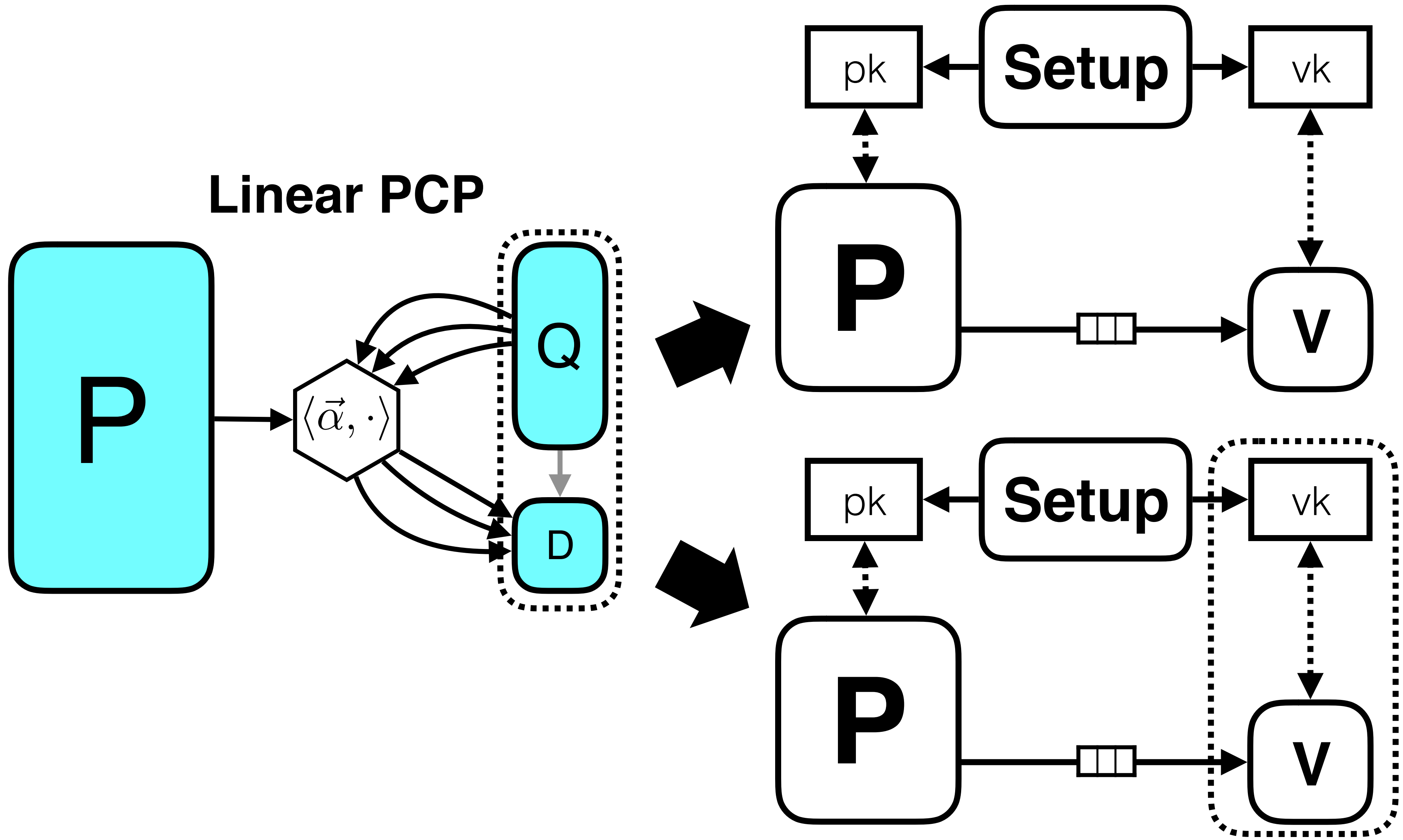


Based on hardness of DLOG in EC groups. ☹️

Post-Quantum Security



Looks solid.

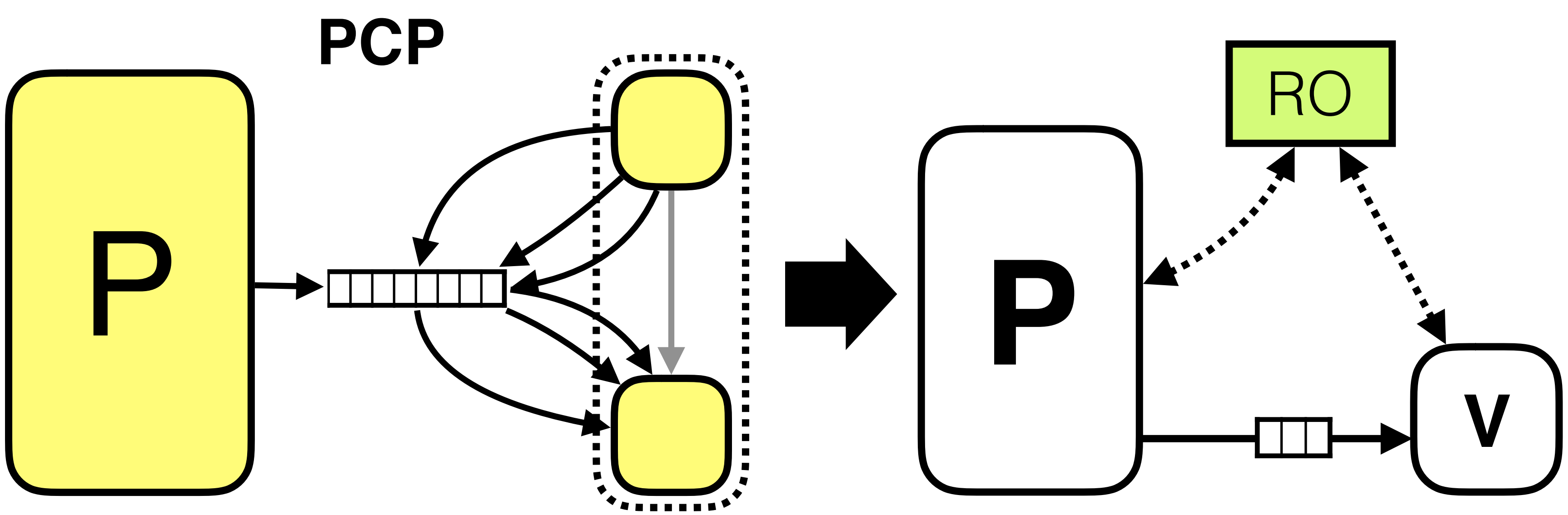


Based on hardness of DLOG in EC groups.

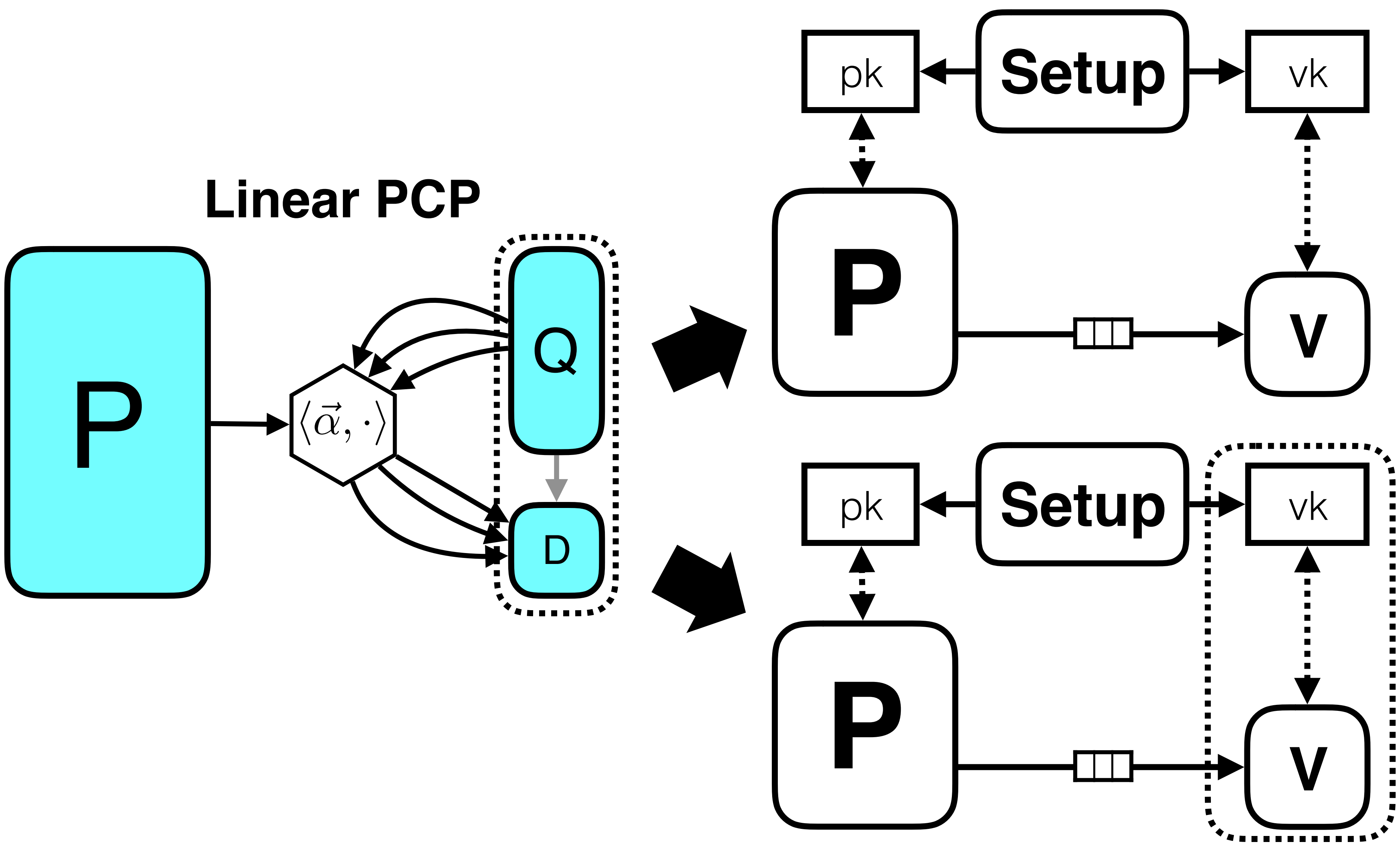
☹️

[BISW EUROCRYPT '17]

Post-Quantum Security



Looks solid.

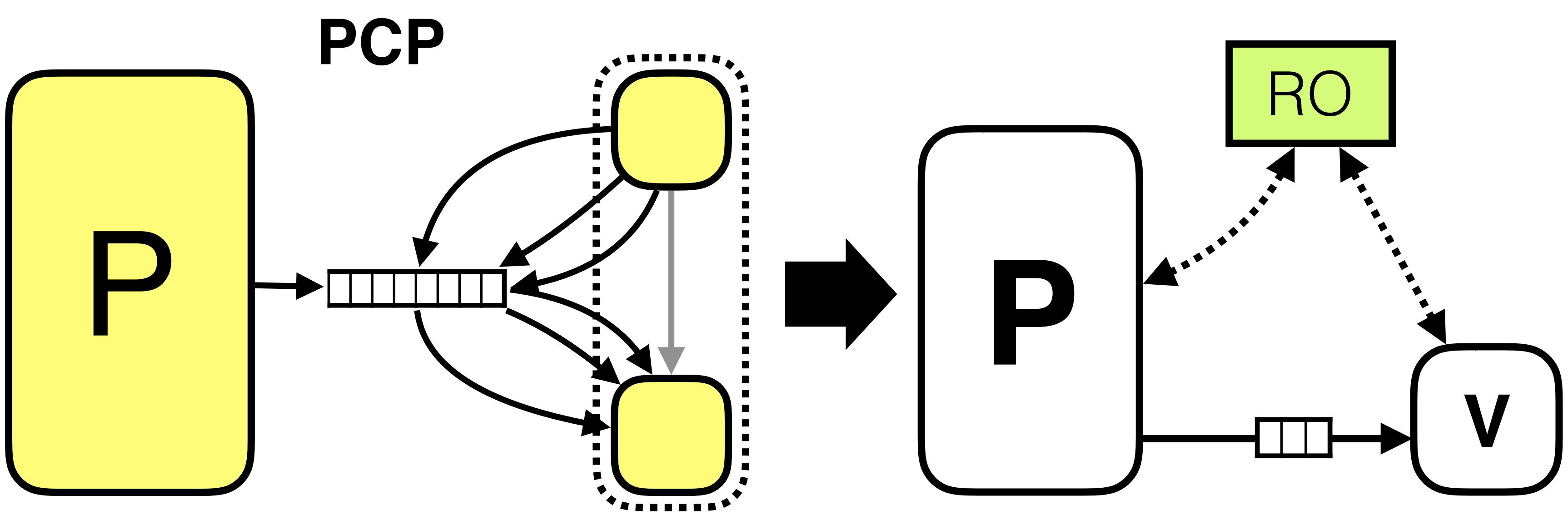


Based on hardness of DLOG in EC groups. ☹️

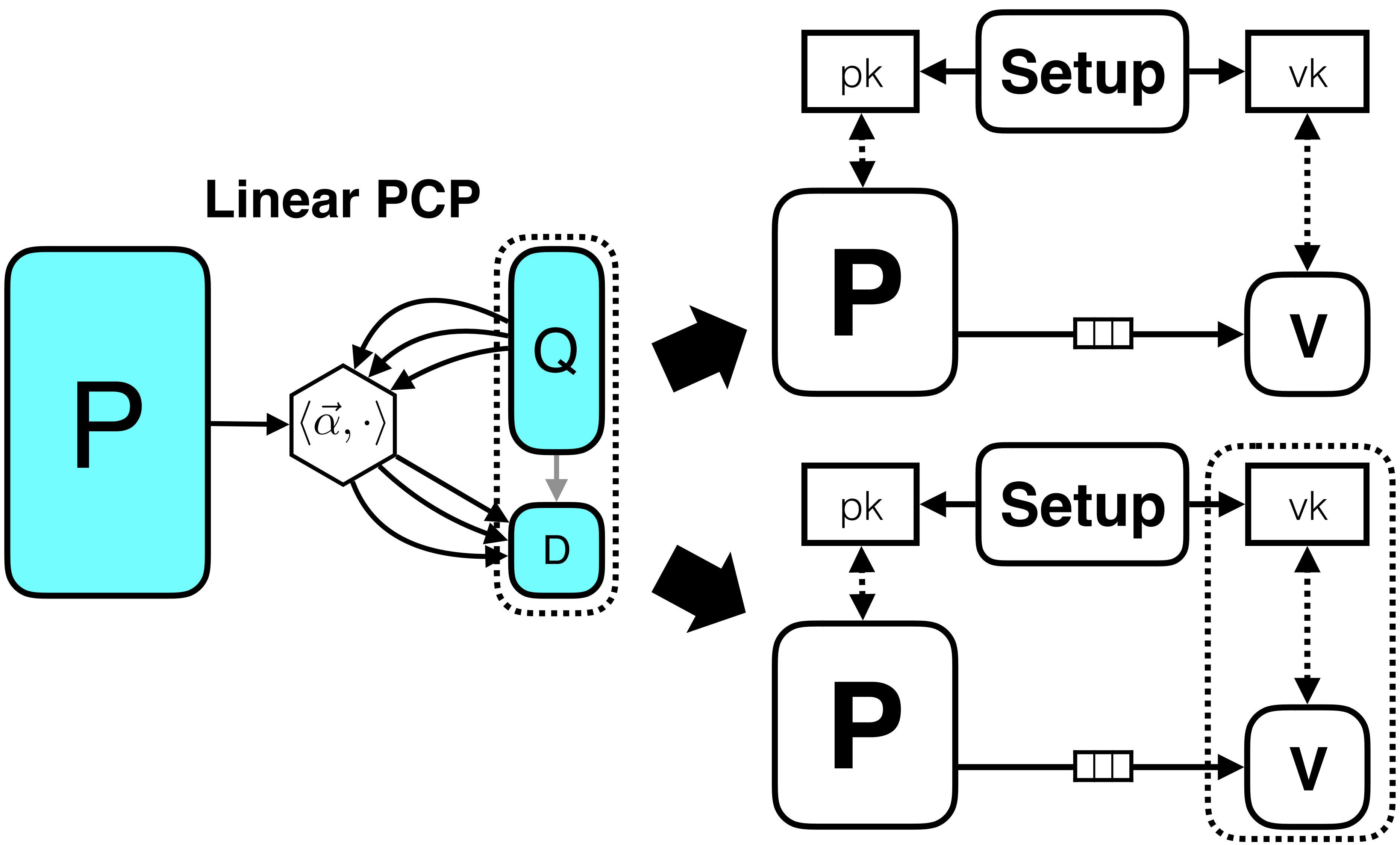
[BISW EUROCRYPT '17]

Lattice-based privately-verifiable ZK-SNARK

Post-Quantum Security



Looks solid.



Based on hardness of DLOG in EC groups. ☹️

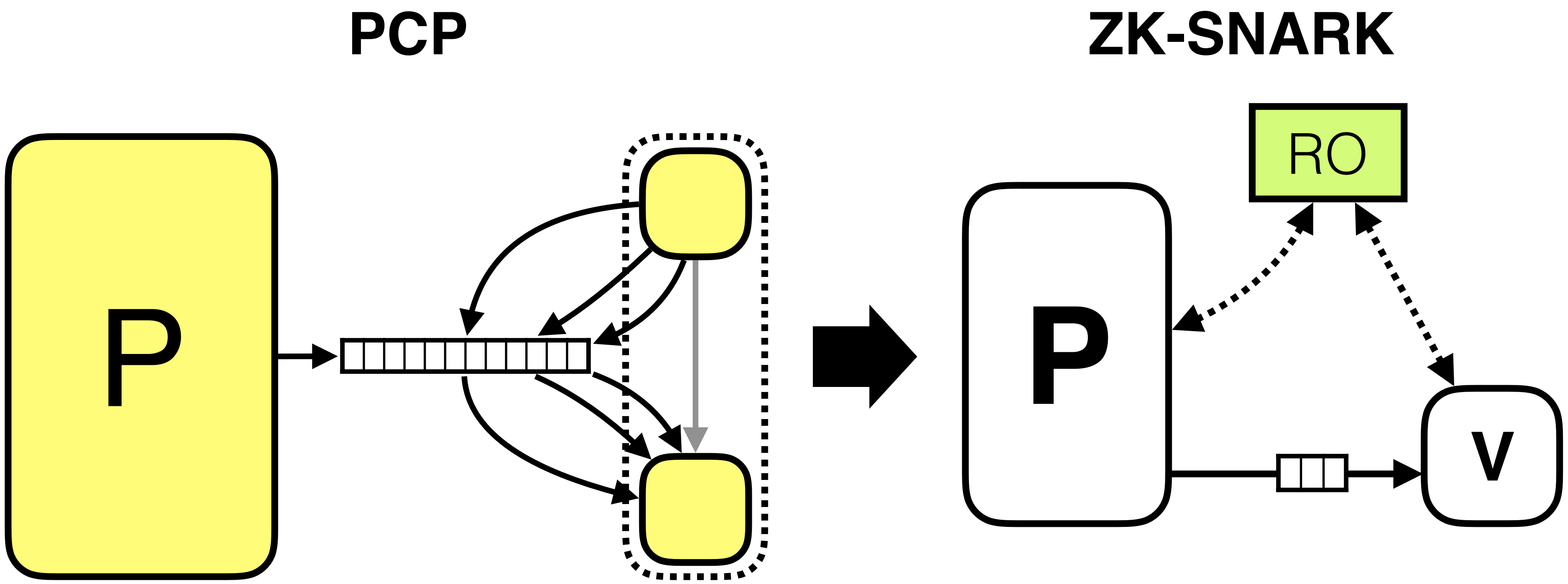
[BISW EUROCRYPT '17]

Lattice-based privately-verifiable ZK-SNARK

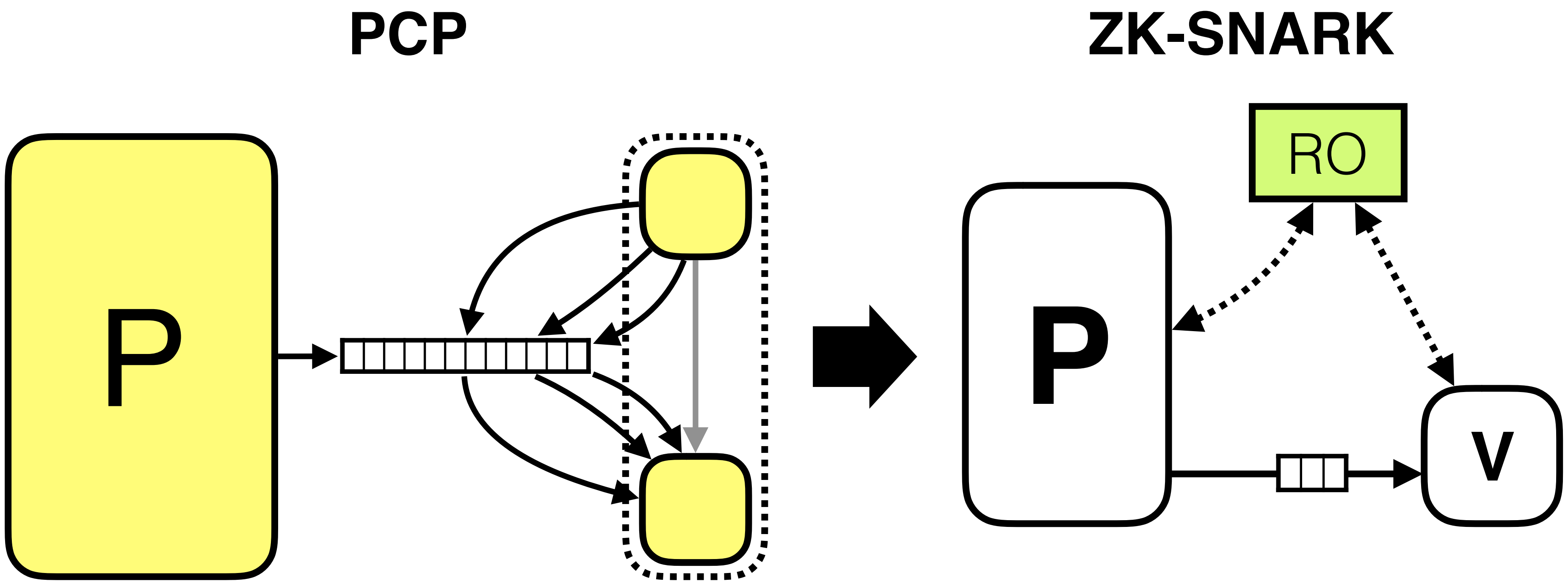
OPEN: public?

PCP-Based ZK-SNARKs

PCP-Based ZK-SNARKs

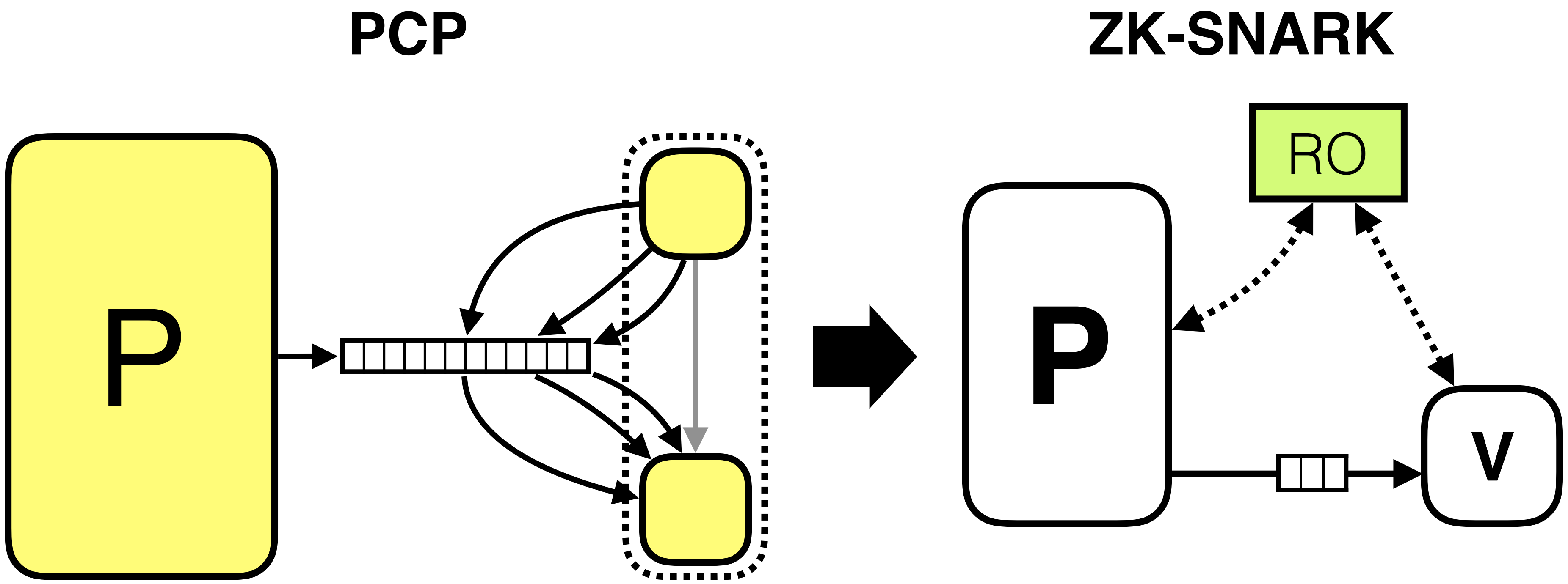


PCP-Based ZK-SNARKs



✓ succinct

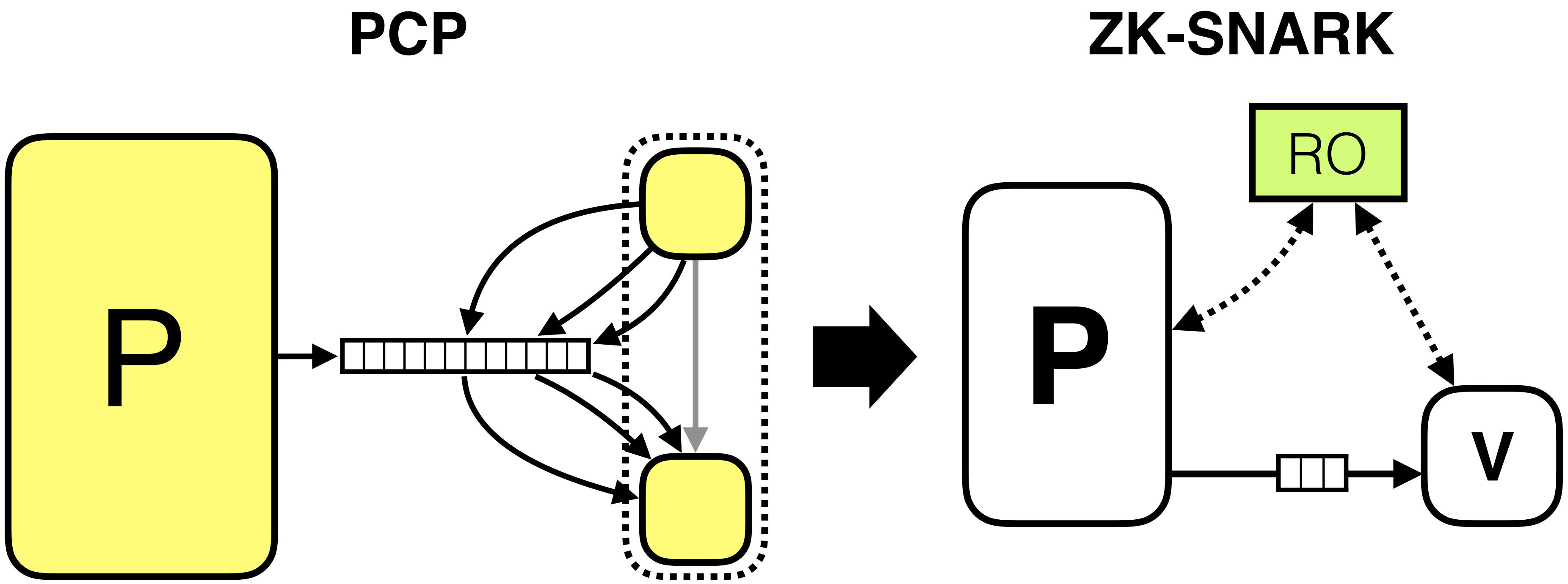
PCP-Based ZK-SNARKs



✓ succinct

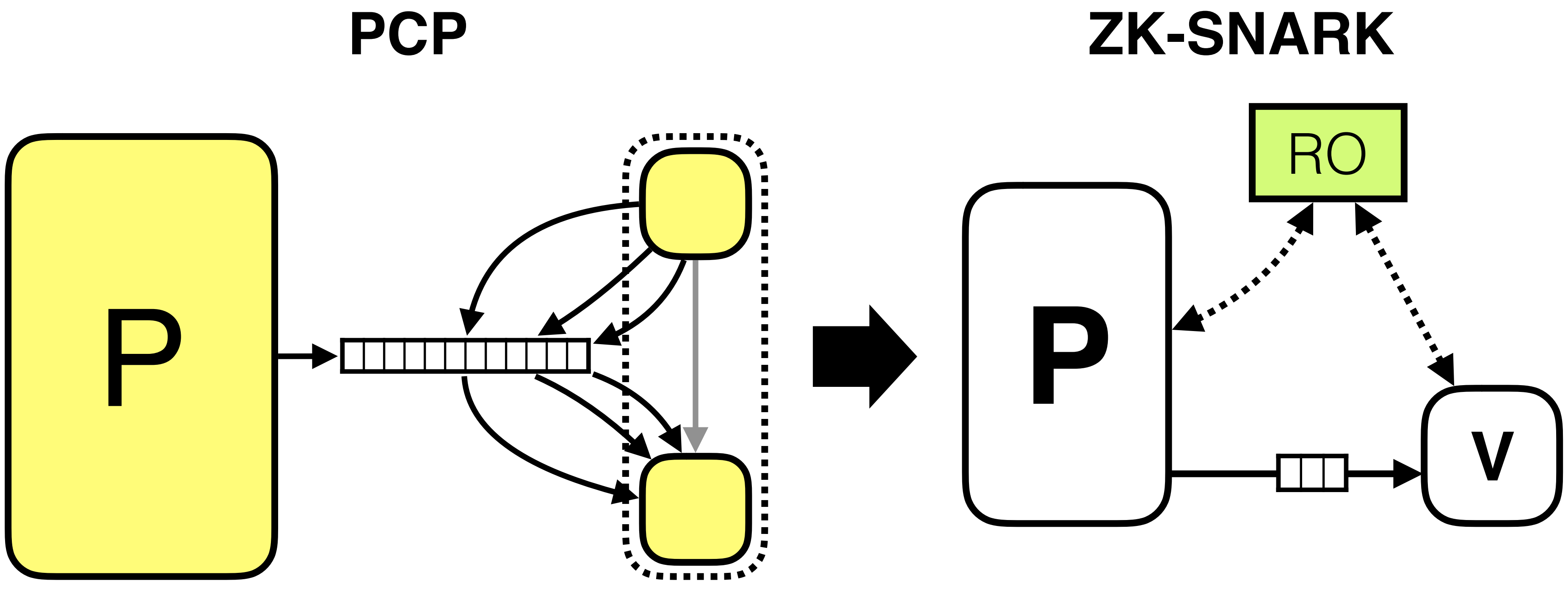
✓ non-interactive

PCP-Based ZK-SNARKs



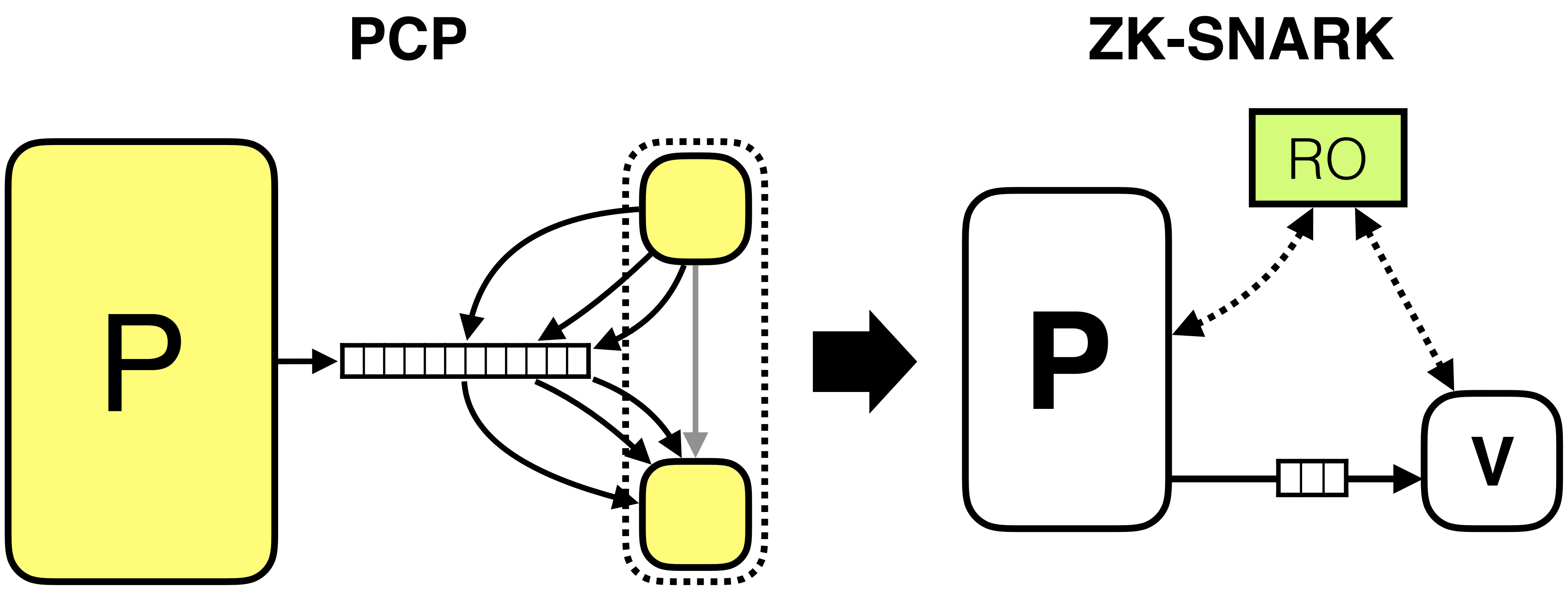
- ✓ succinct
- ✓ non-interactive
- ✓ no setup

PCP-Based ZK-SNARKs



- ✓ succinct
- ✓ non-interactive
- ✓ no setup
- ✓ post-quantum secure

PCP-Based ZK-SNARKs



✓ succinct

✓ non-interactive

✓ no setup

✓ post-quantum secure

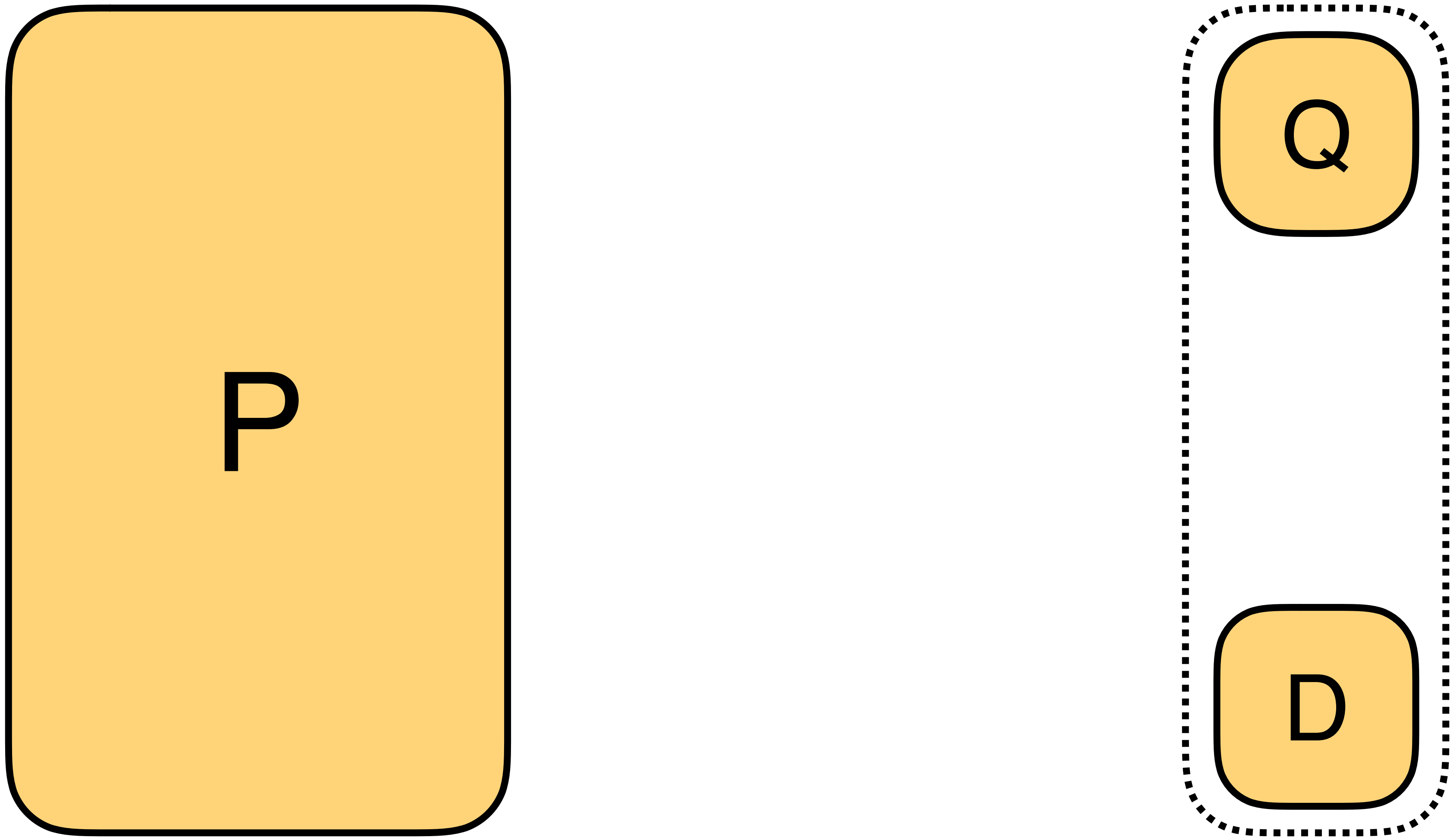
😭 terrible concrete efficiency

Interactive Oracle Proofs

[B**C**S16][RRR16]

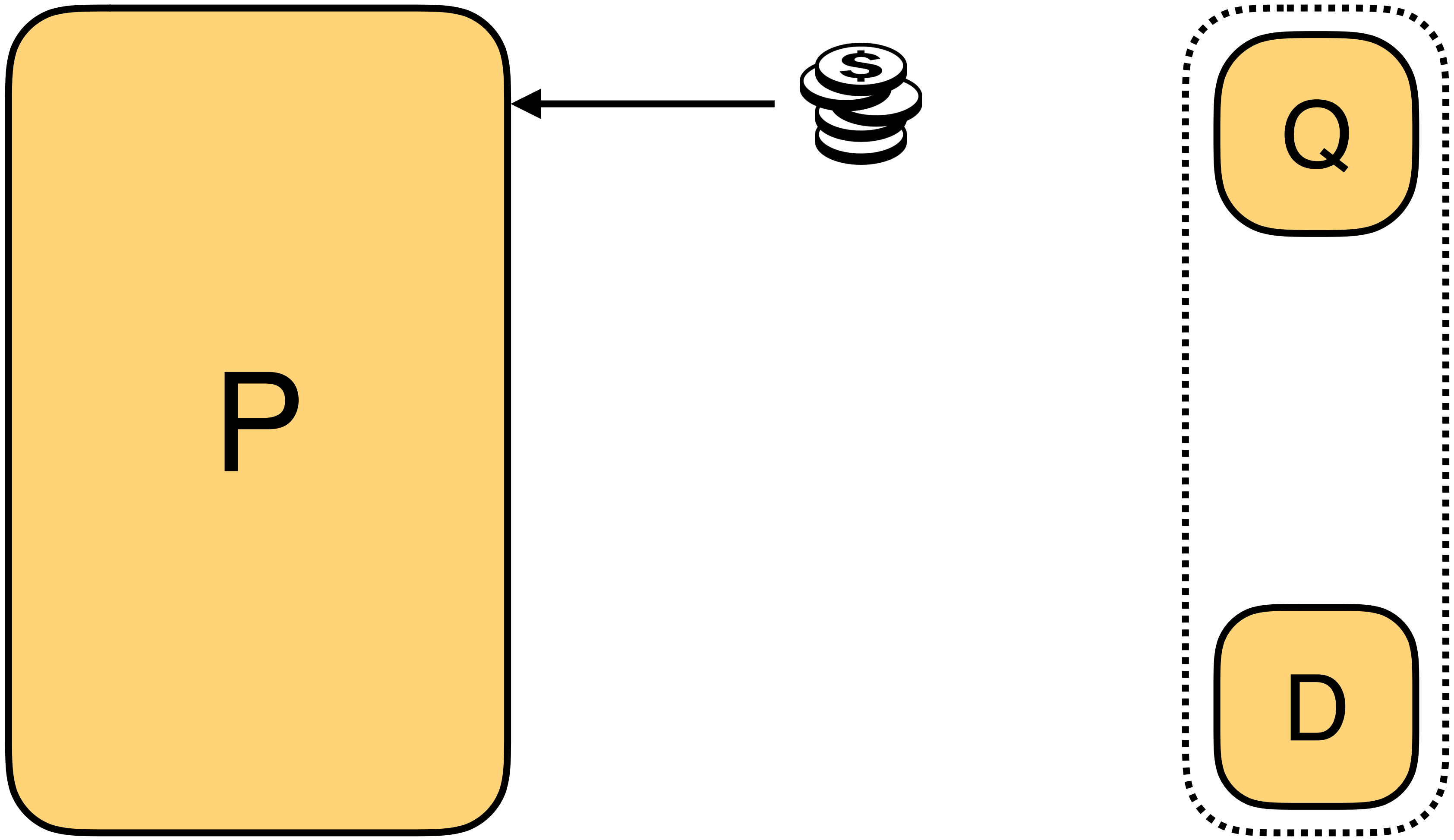
Interactive Oracle Proofs

[B**C**S16][RRR16]



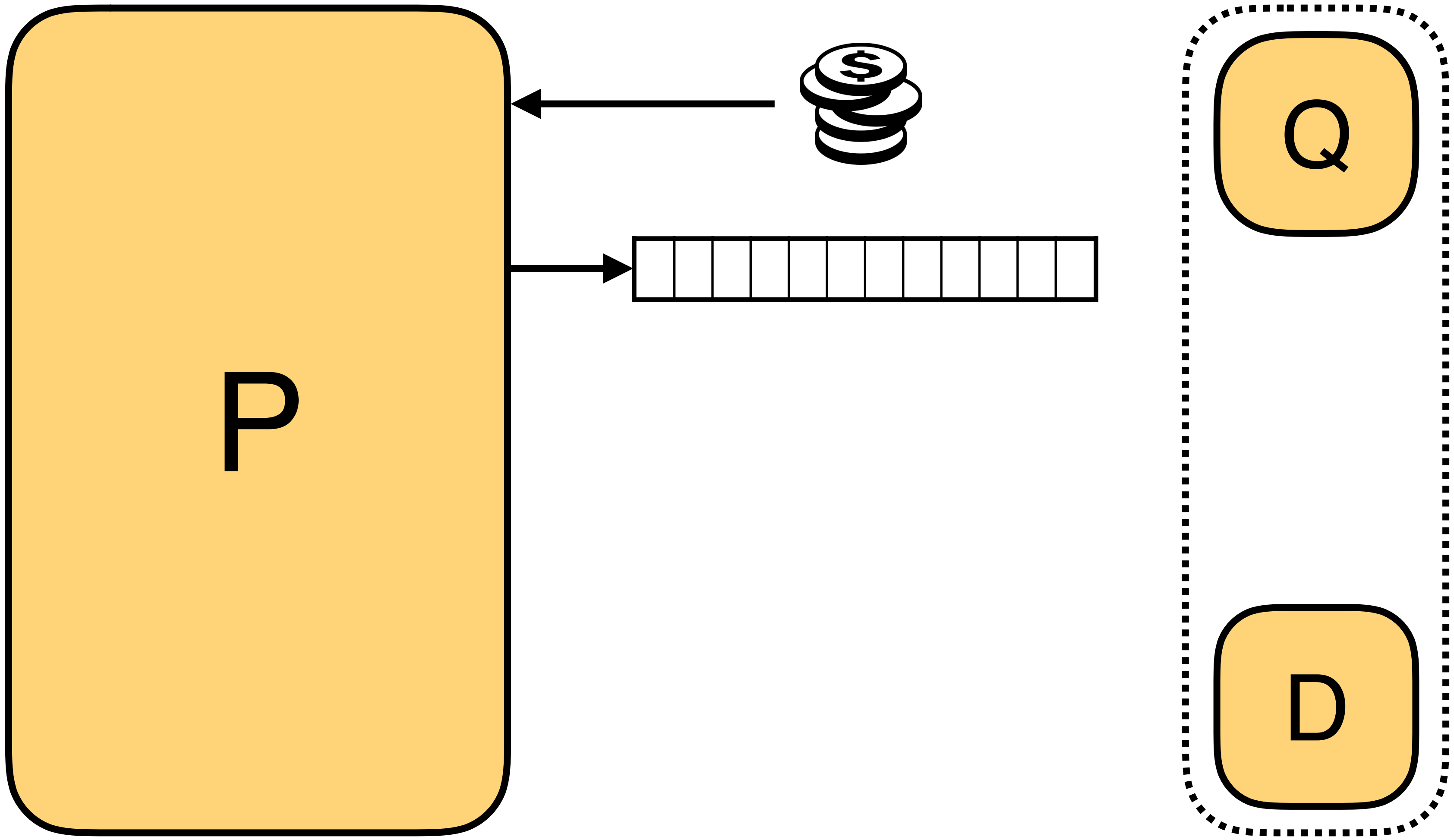
Interactive Oracle Proofs

[BCS16][RRR16]



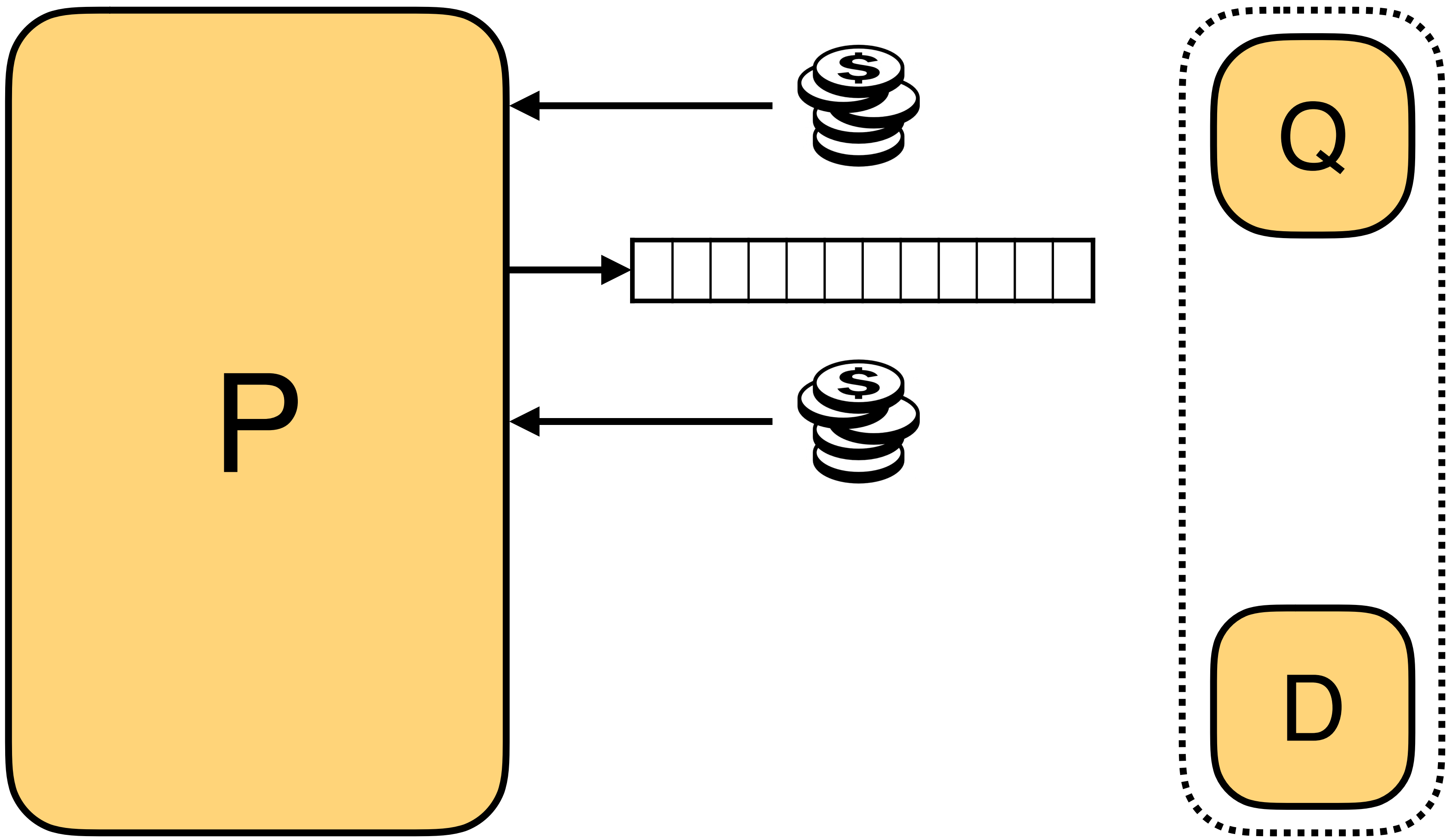
Interactive Oracle Proofs

[B**C**S16][RRR16]



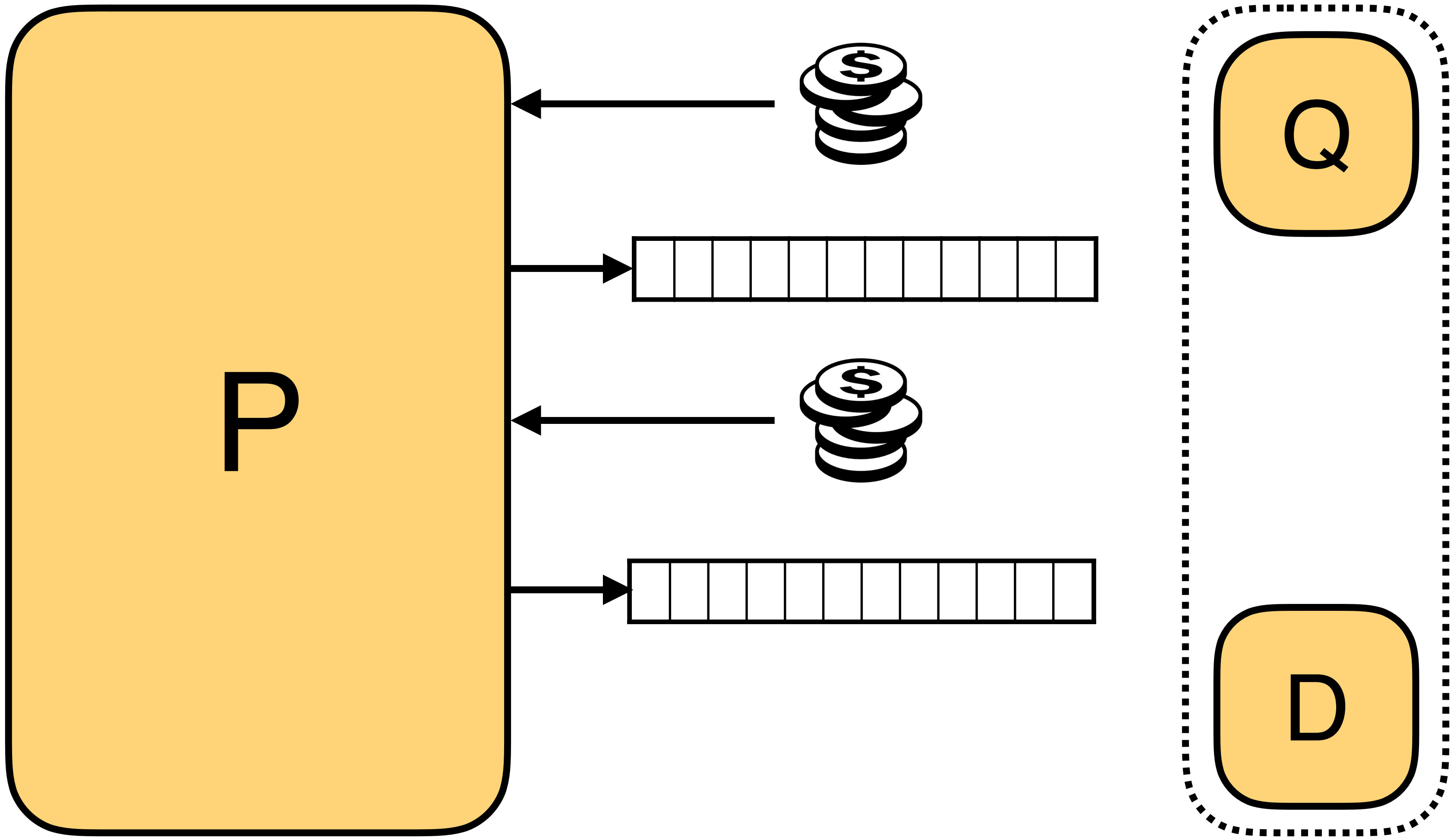
Interactive Oracle Proofs

[B**C**S16][RRR16]



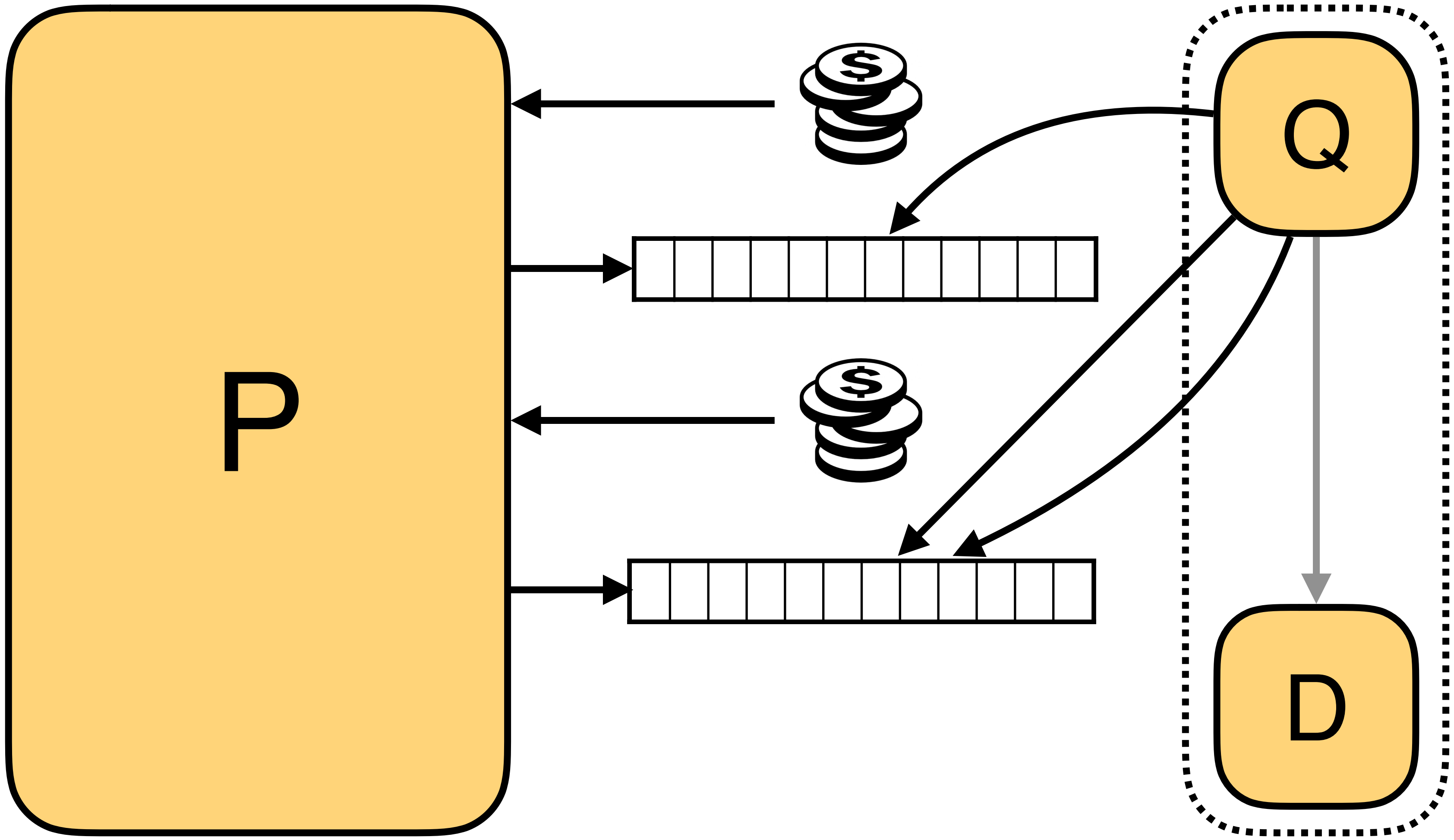
Interactive Oracle Proofs

[B**C**S16][RRR16]



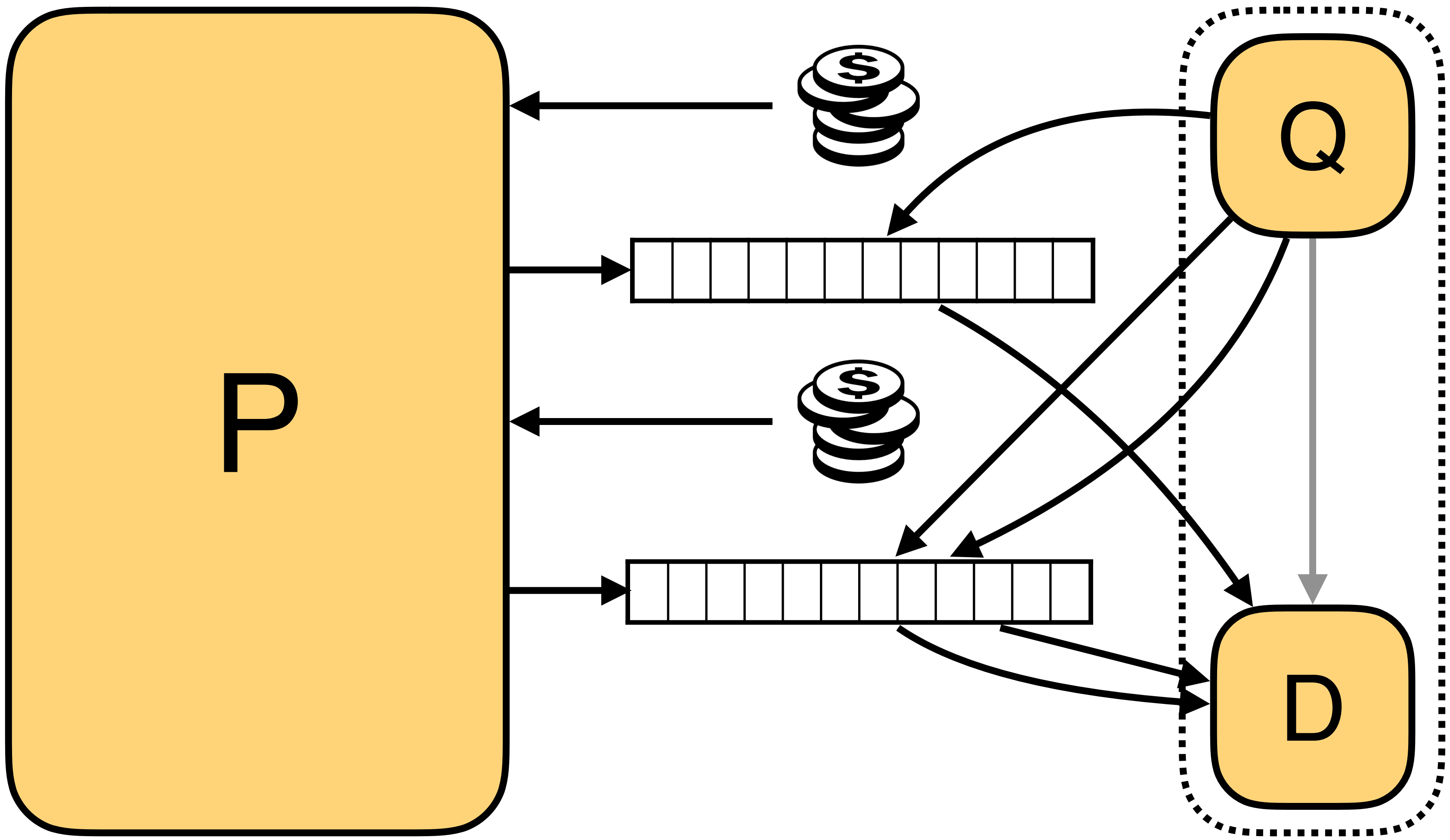
Interactive Oracle Proofs

[B**C**S16][RRR16]



Interactive Oracle Proofs

[B**C**S16][RRR16]



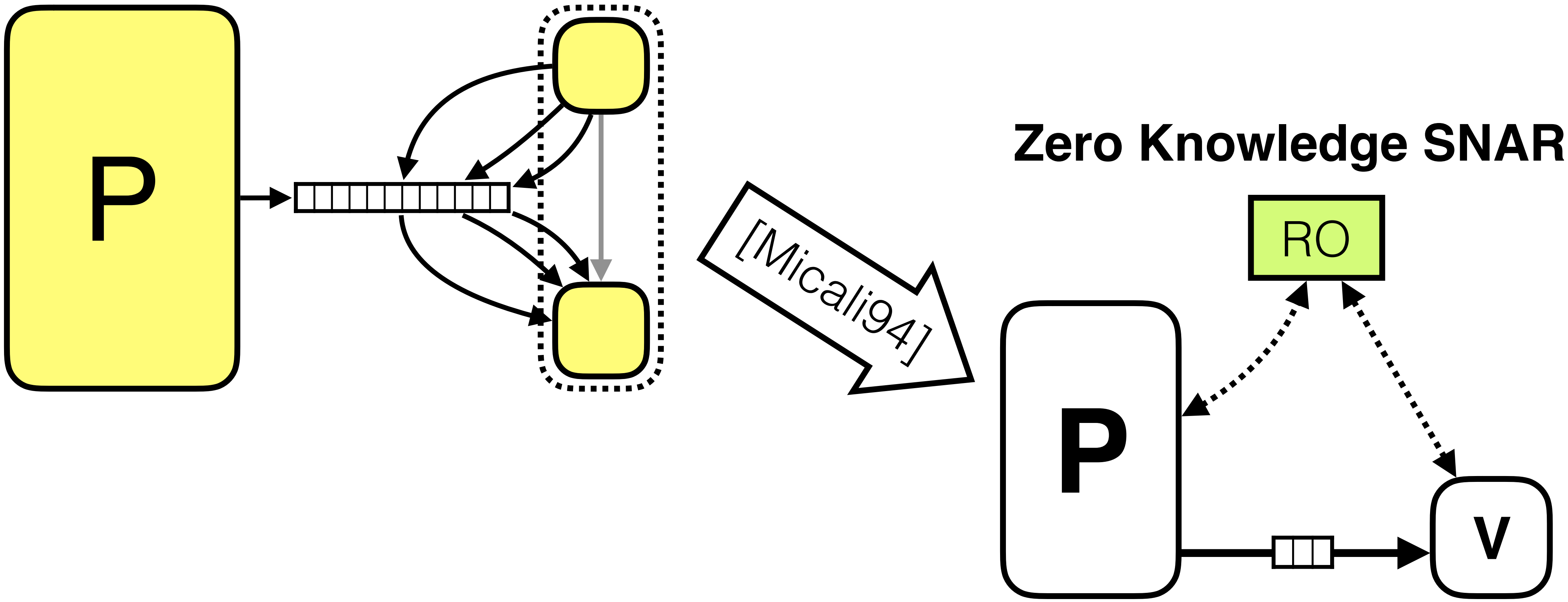
The verifier can simultaneously leverage randomness, interaction, and probabilistic checking.

ZK-SNARKs From IOPs

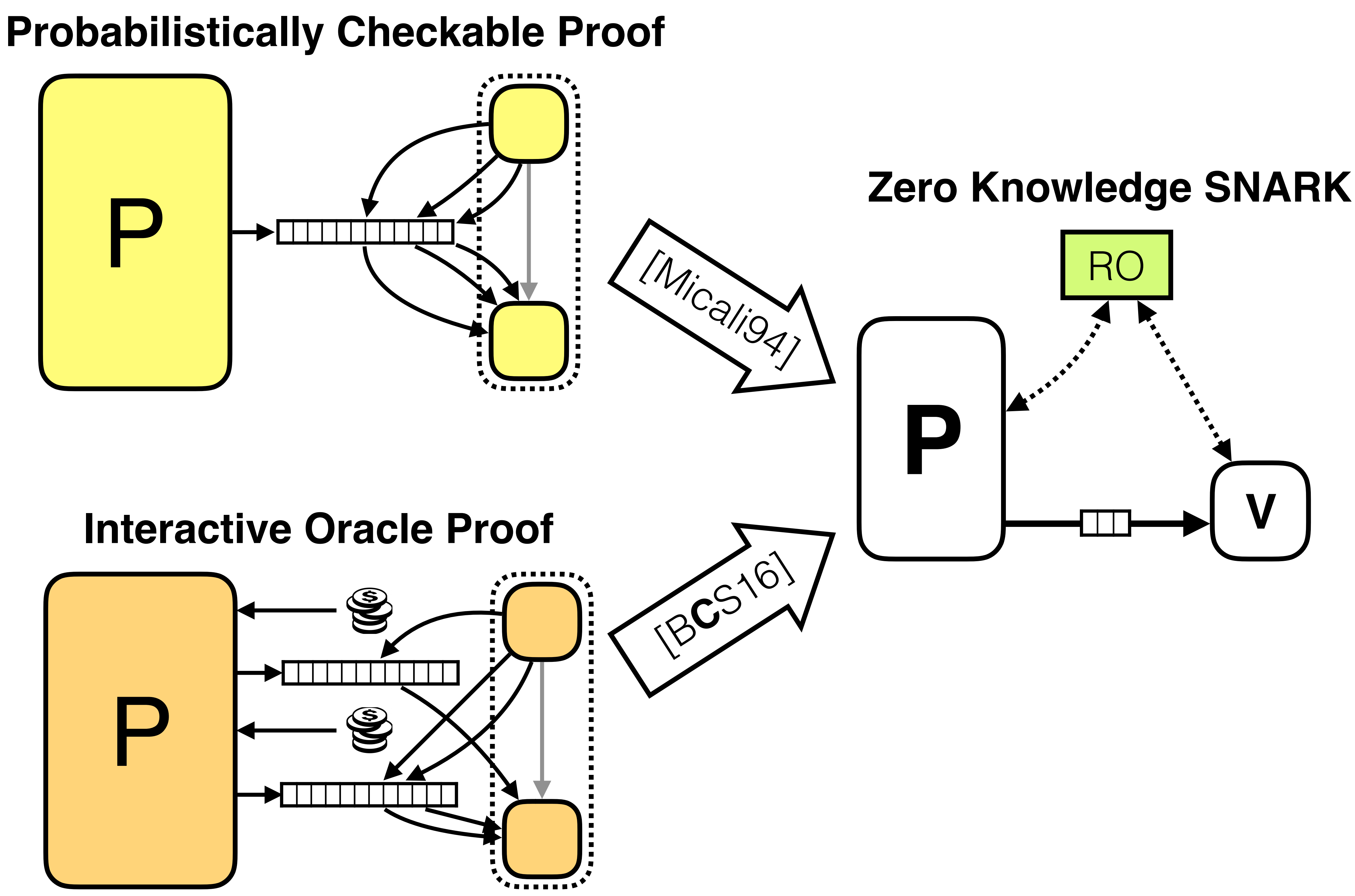
ZK-SNARKs From IOPs

Probabilistically Checkable Proof

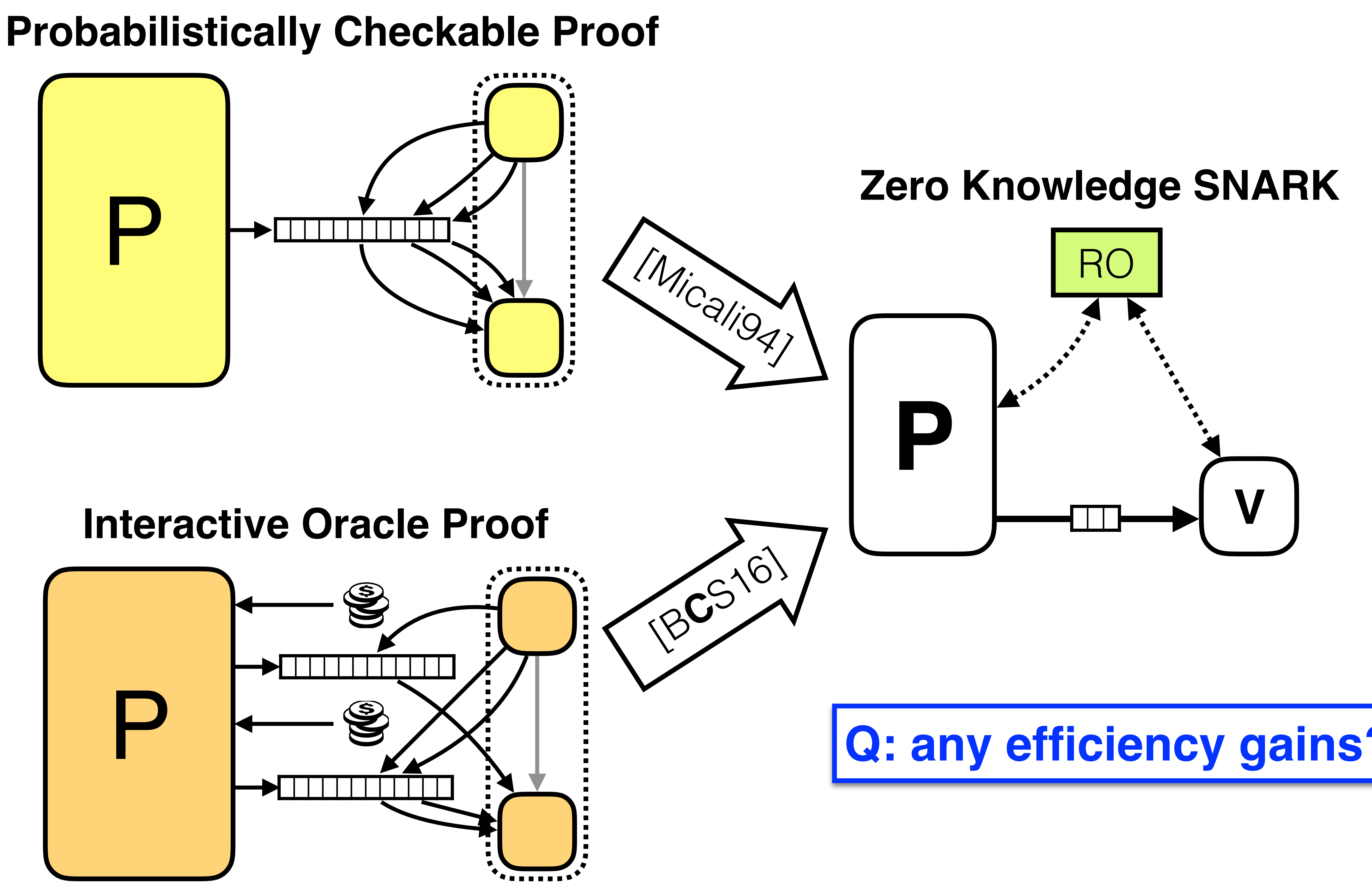
Zero Knowledge SNARK



ZK-SNARKs From IOPs



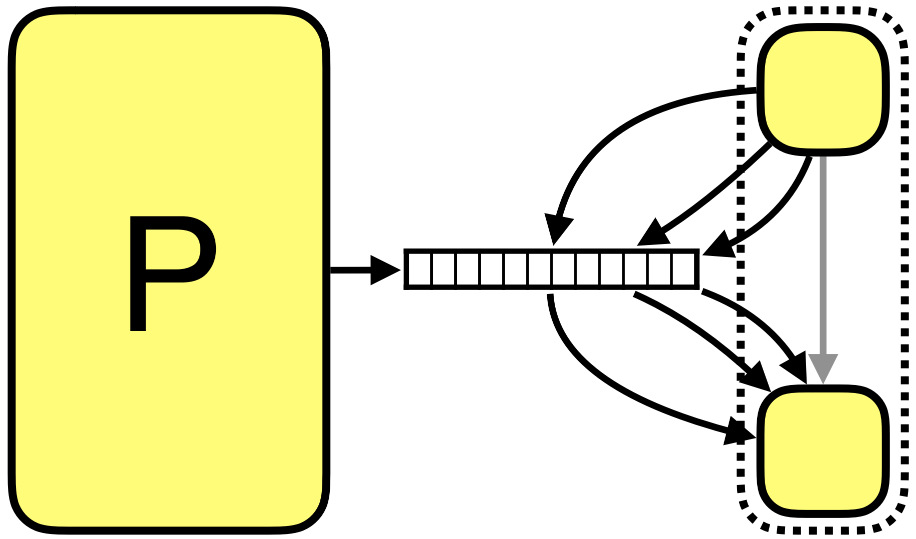
ZK-SNARKs From IOPs

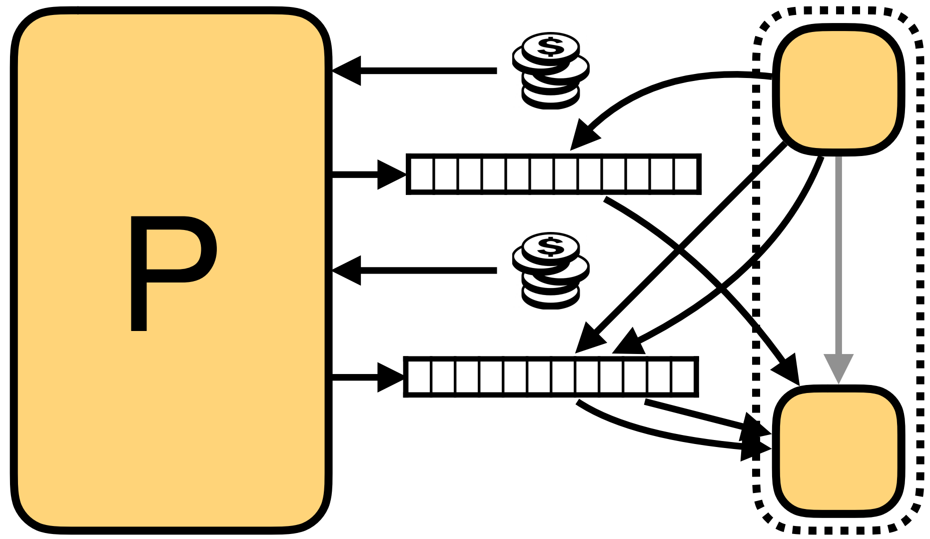


Q: any efficiency gains?

IOPs are more efficient than PCPs

IOPs are more efficient than PCPs

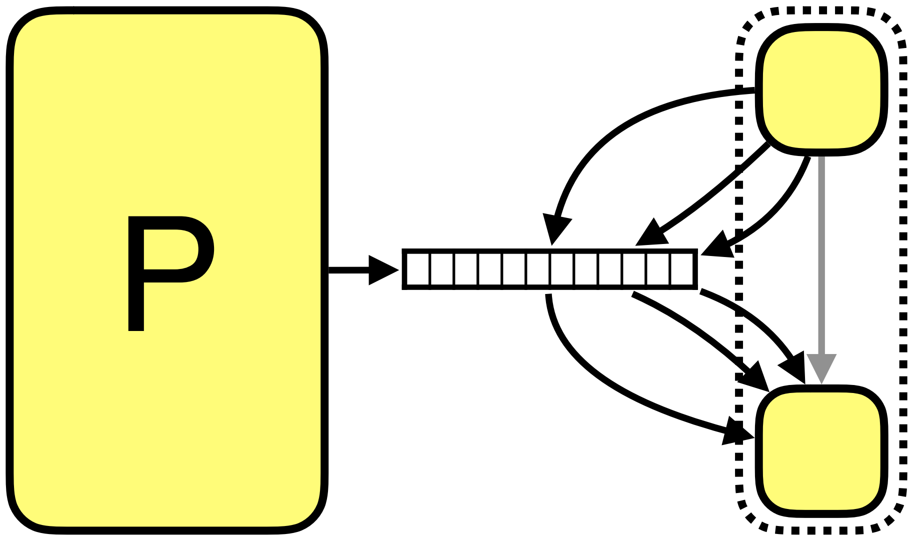


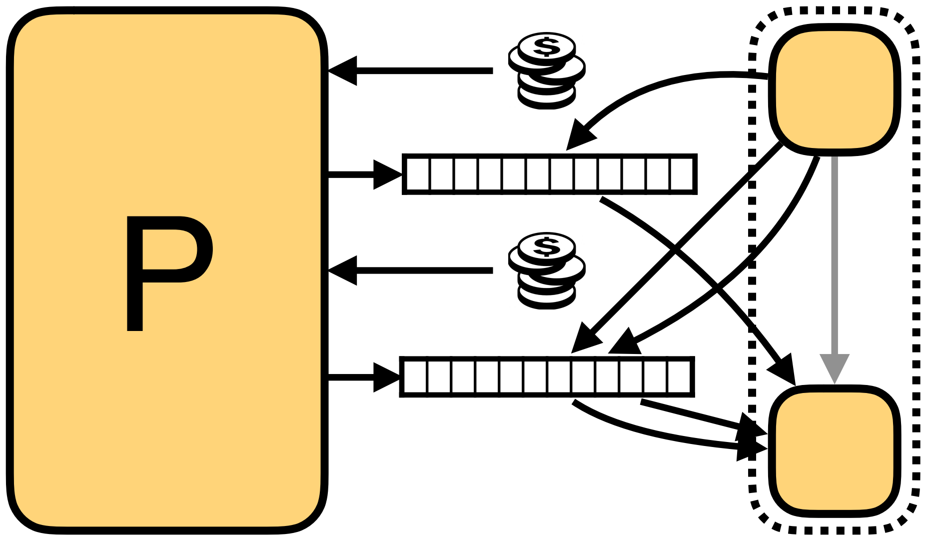


IOPs are more efficient than PCPs

best proof length
without ZK

IOPs are more efficient than PCPs

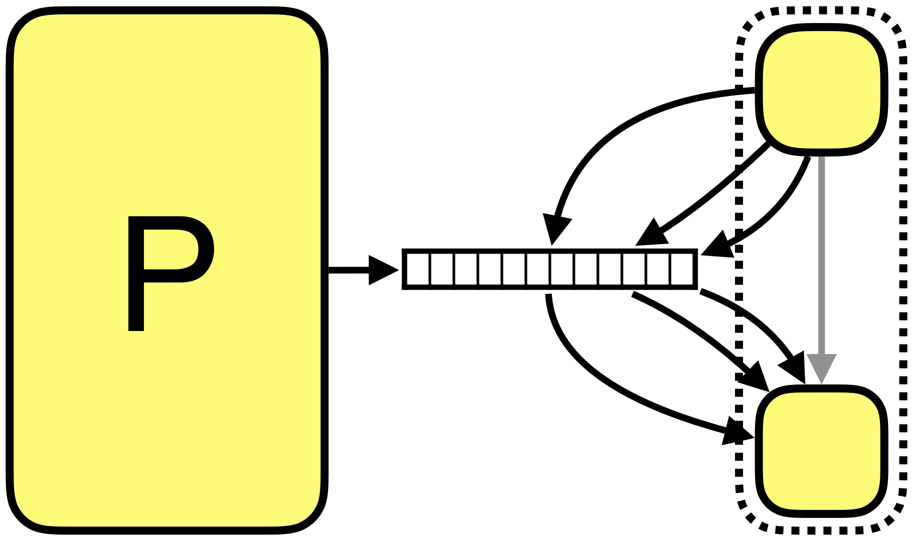


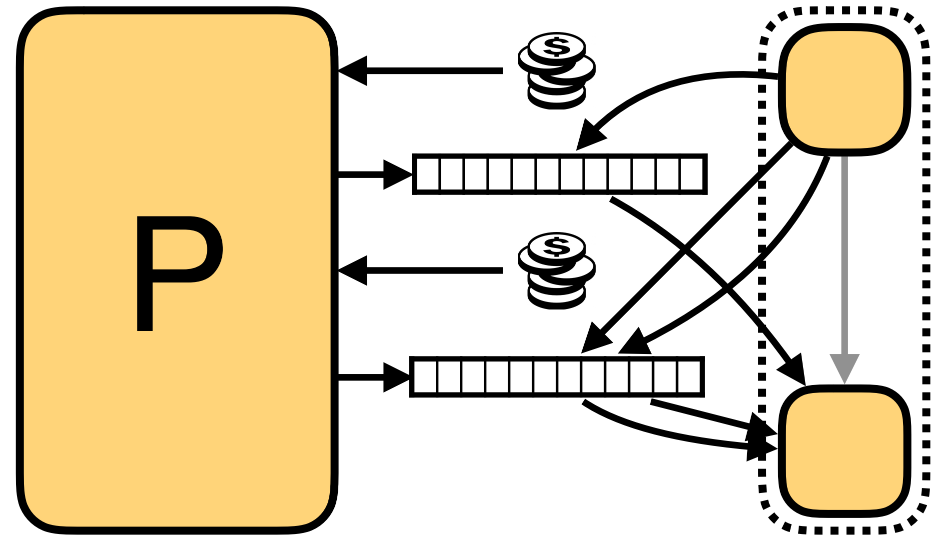


quasilinear
[BS08][Din07]

best proof length
without ZK

IOPs are more efficient than PCPs



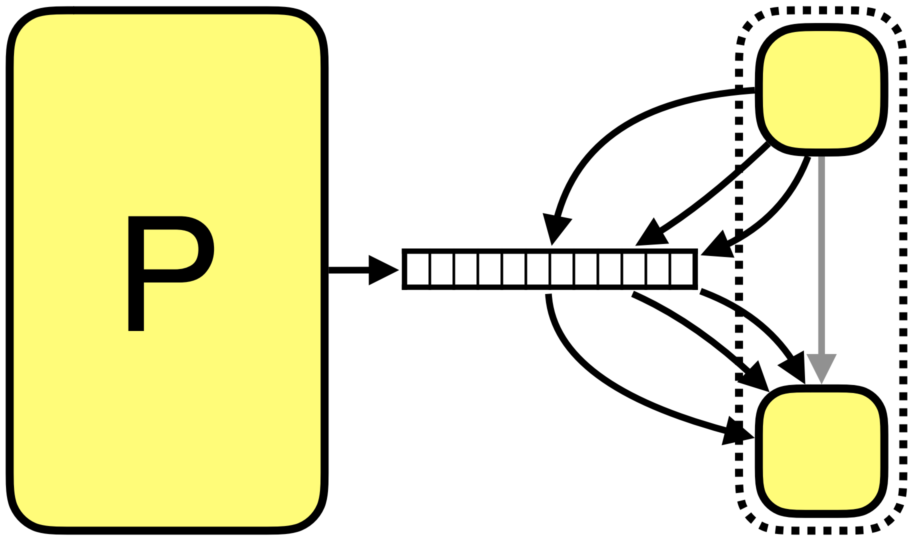


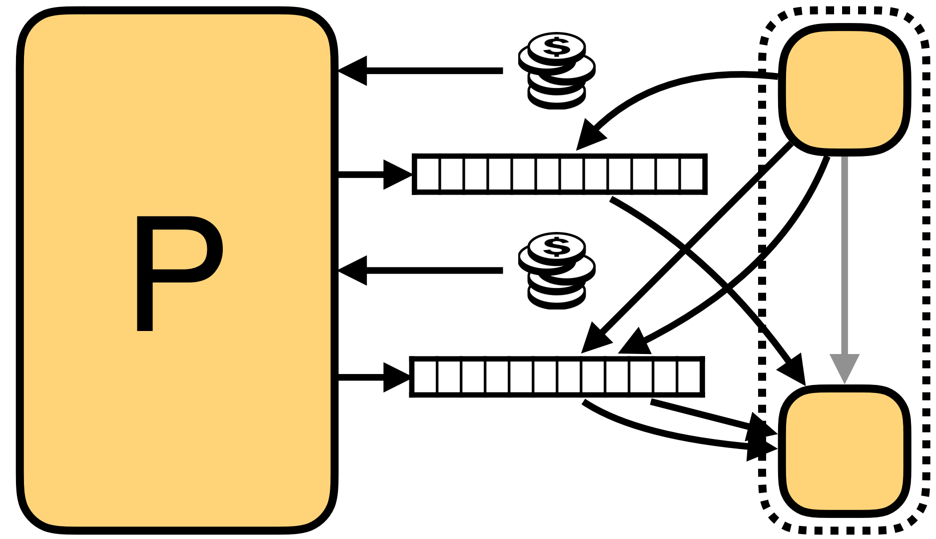
quasilinear
[BS08][Din07]

best proof length
without ZK

linear
[B**C**GRS16]

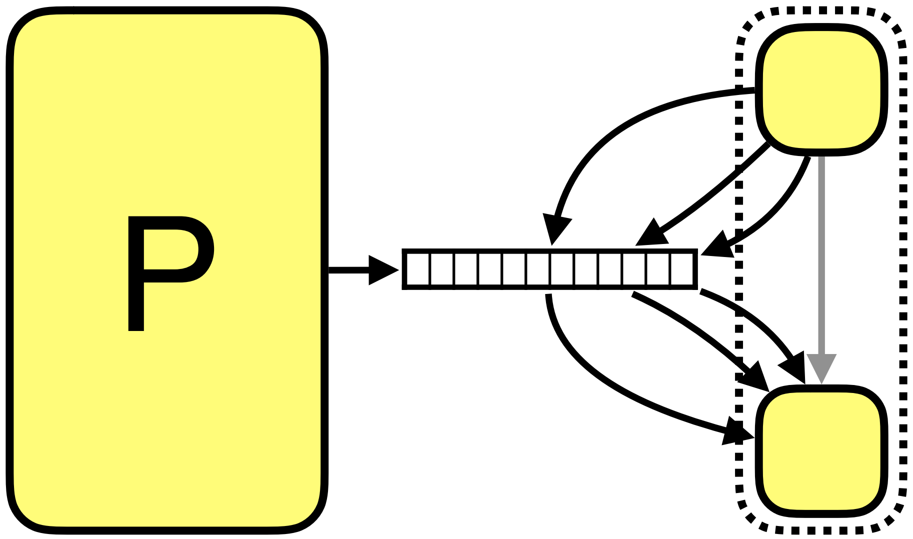
IOPs are more efficient than PCPs

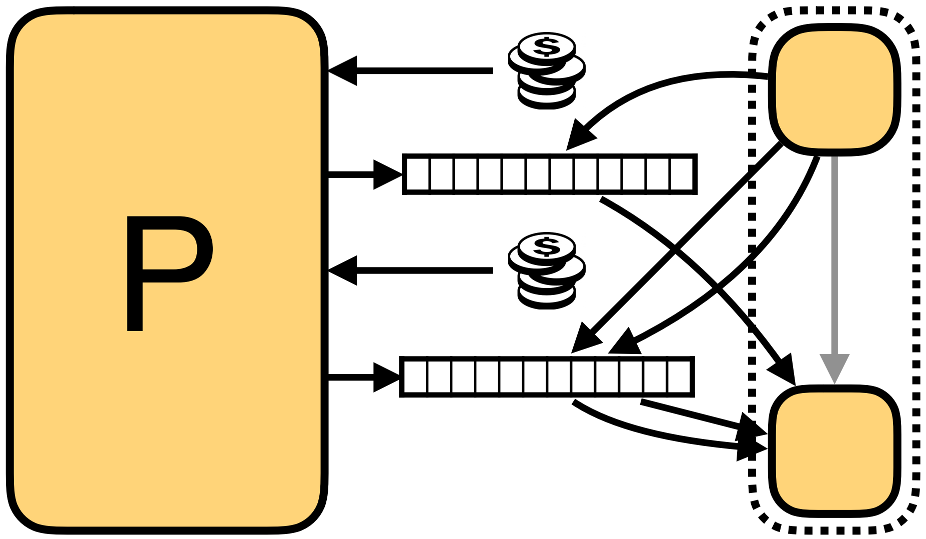




quasilinear [BS08][Din07]	best proof length without ZK	**linear** [B**C**GRS16]
	best proof length with ZK	

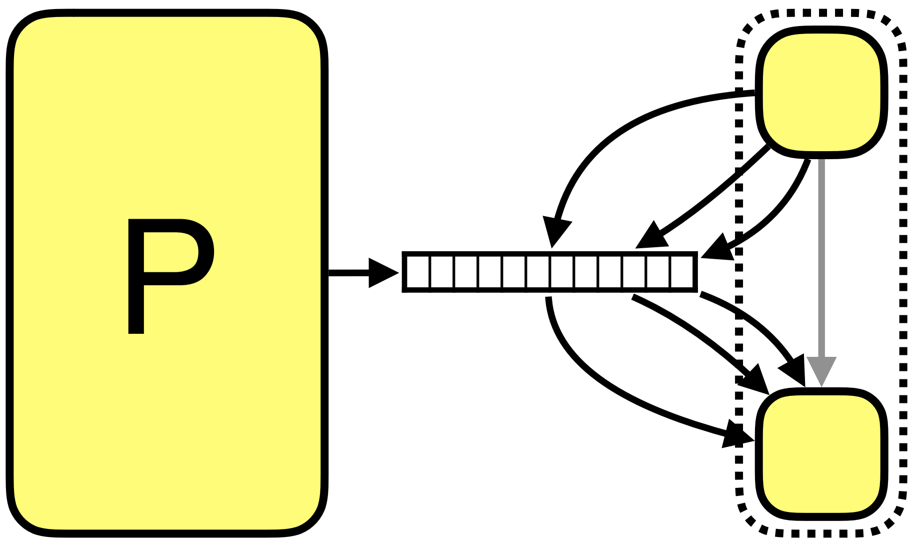
IOPs are more efficient than PCPs

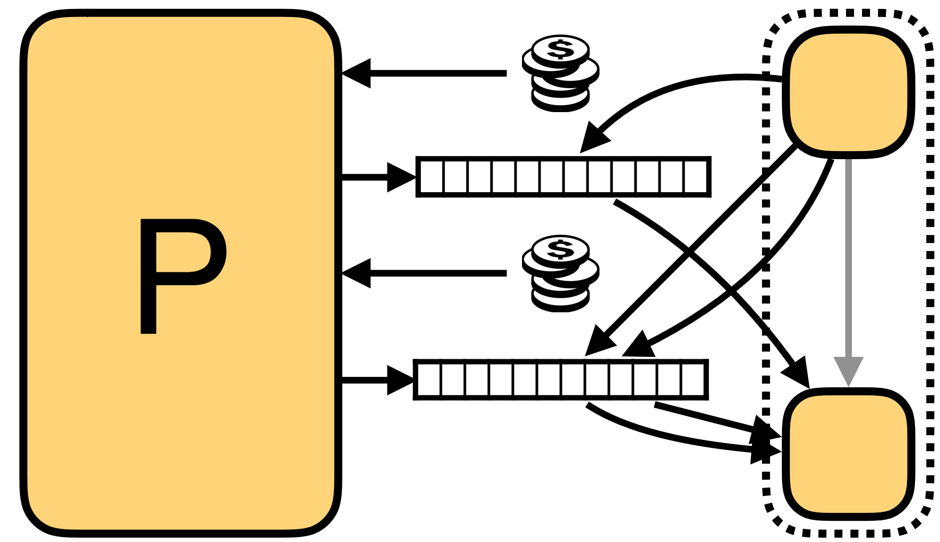




quasilinear [BS08][Din07]	best proof length without ZK	**linear** [B**C**GRS16]
polynomial [KPT97]	best proof length with ZK	

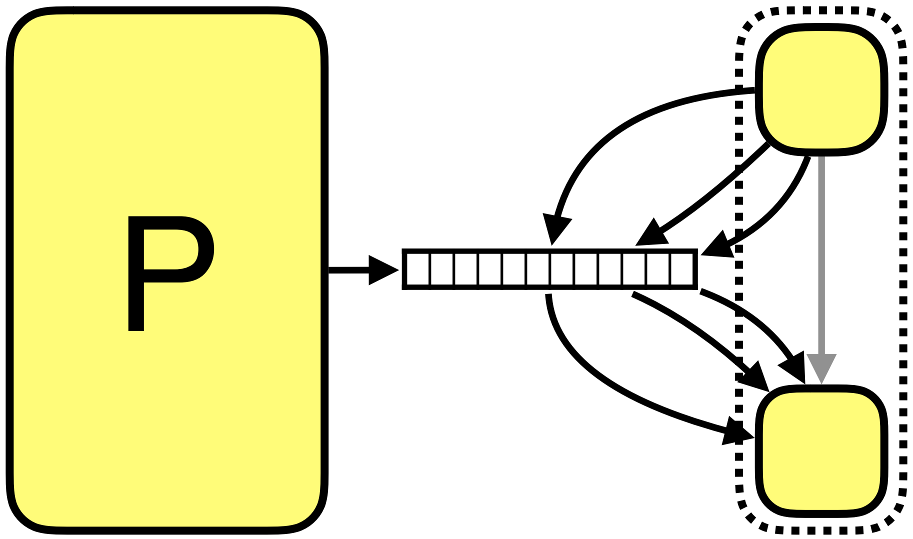
IOPs are more efficient than PCPs

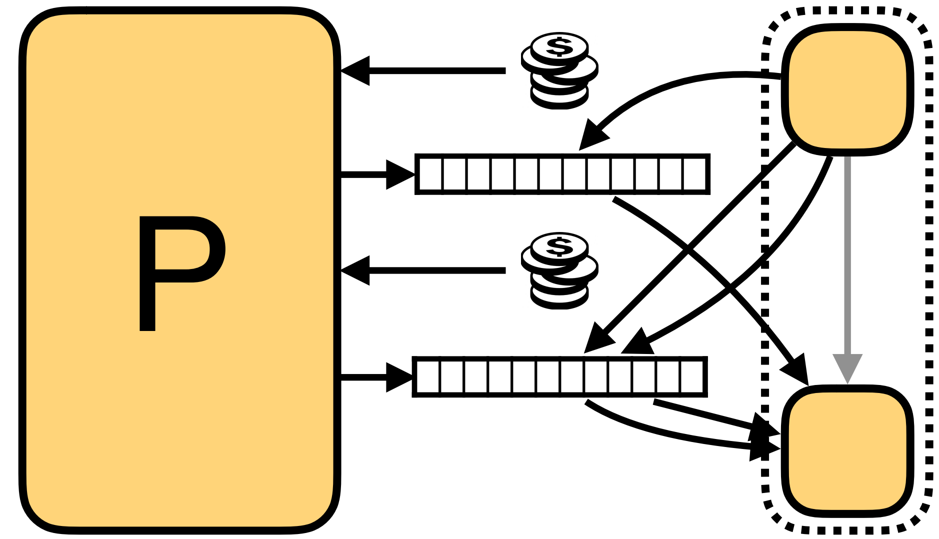




quasilinear [BS08][Din07]	best proof length without ZK	**linear** [B**C**GRS16]
polynomial [KPT97]	best proof length with ZK	**quasilinear** [B**C**GV16]

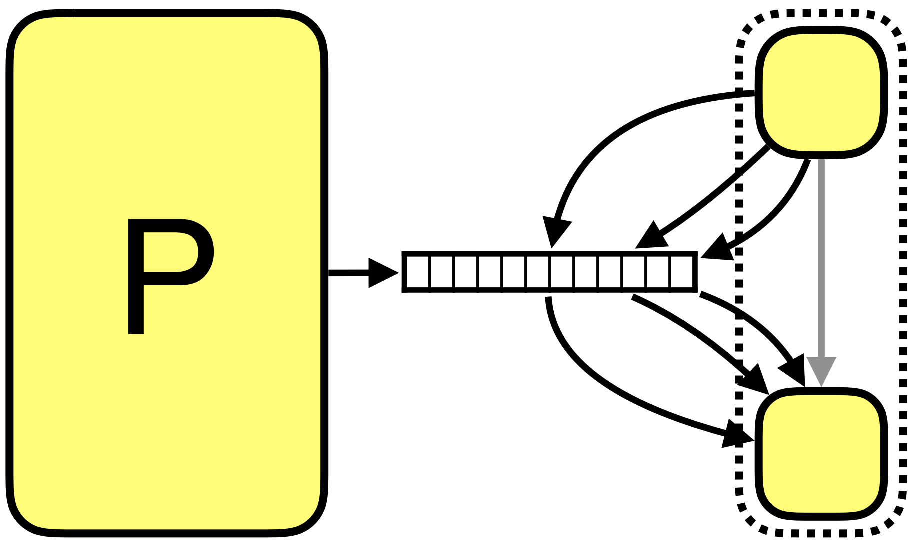
IOPs are more efficient than PCPs

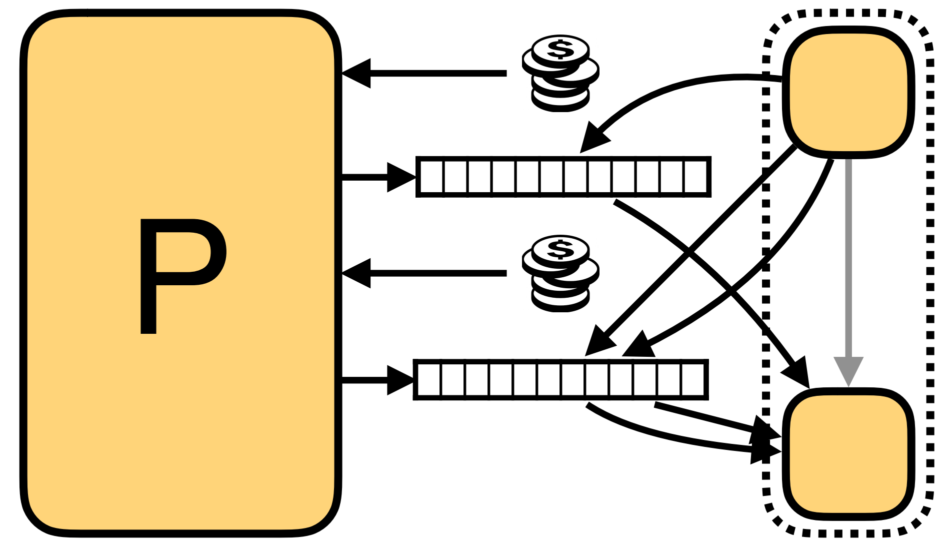




quasilinear [BS08][Din07]	best proof length without ZK	**linear** [B**C**GRS16]
polynomial [KPT97]	best proof length with ZK	**quasilinear** [B**C**GV16]
		cheaper ZK… [B**C**FGRS16][B**C**FS17]

IOPs are more efficient than PCPs

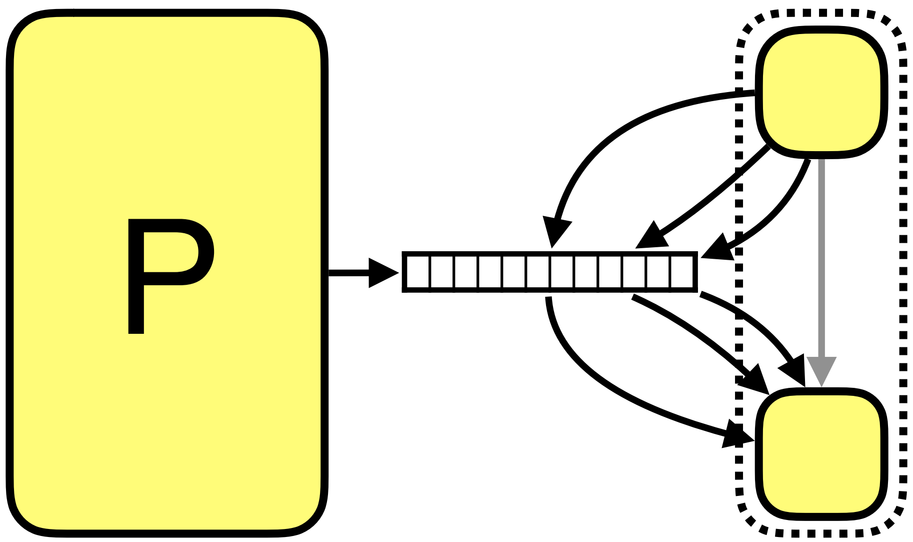


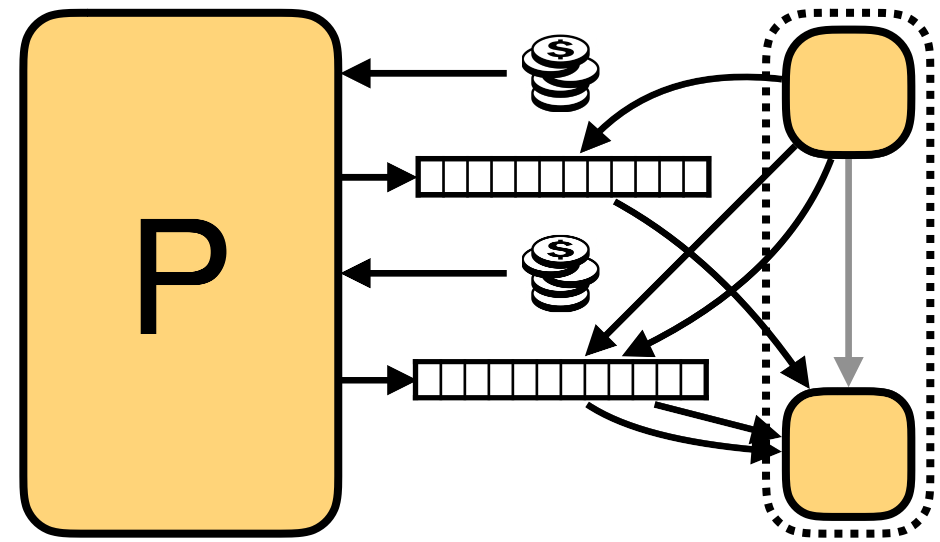


quasilinear [BS08][Din07]	best proof length without ZK	**linear** [B**C**GRS16]
polynomial [KPT97]	best proof length with ZK	**quasilinear** [B**C**GV16]
		cheaper ZK… [B**C**FGRS16][B**C**FS17]

Encouraging progress, and it already improved working prototypes.

IOPs are more efficient than PCPs



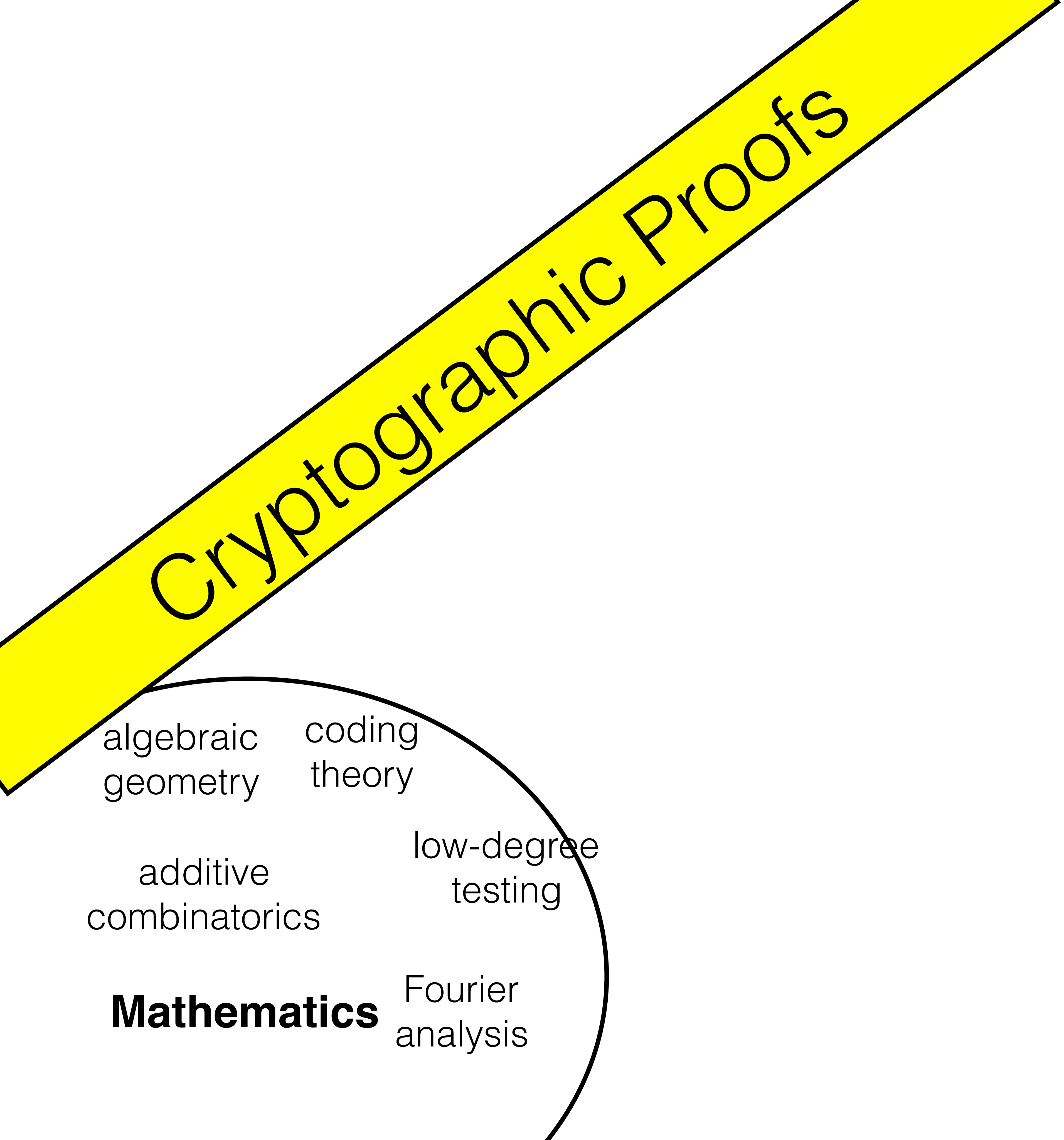


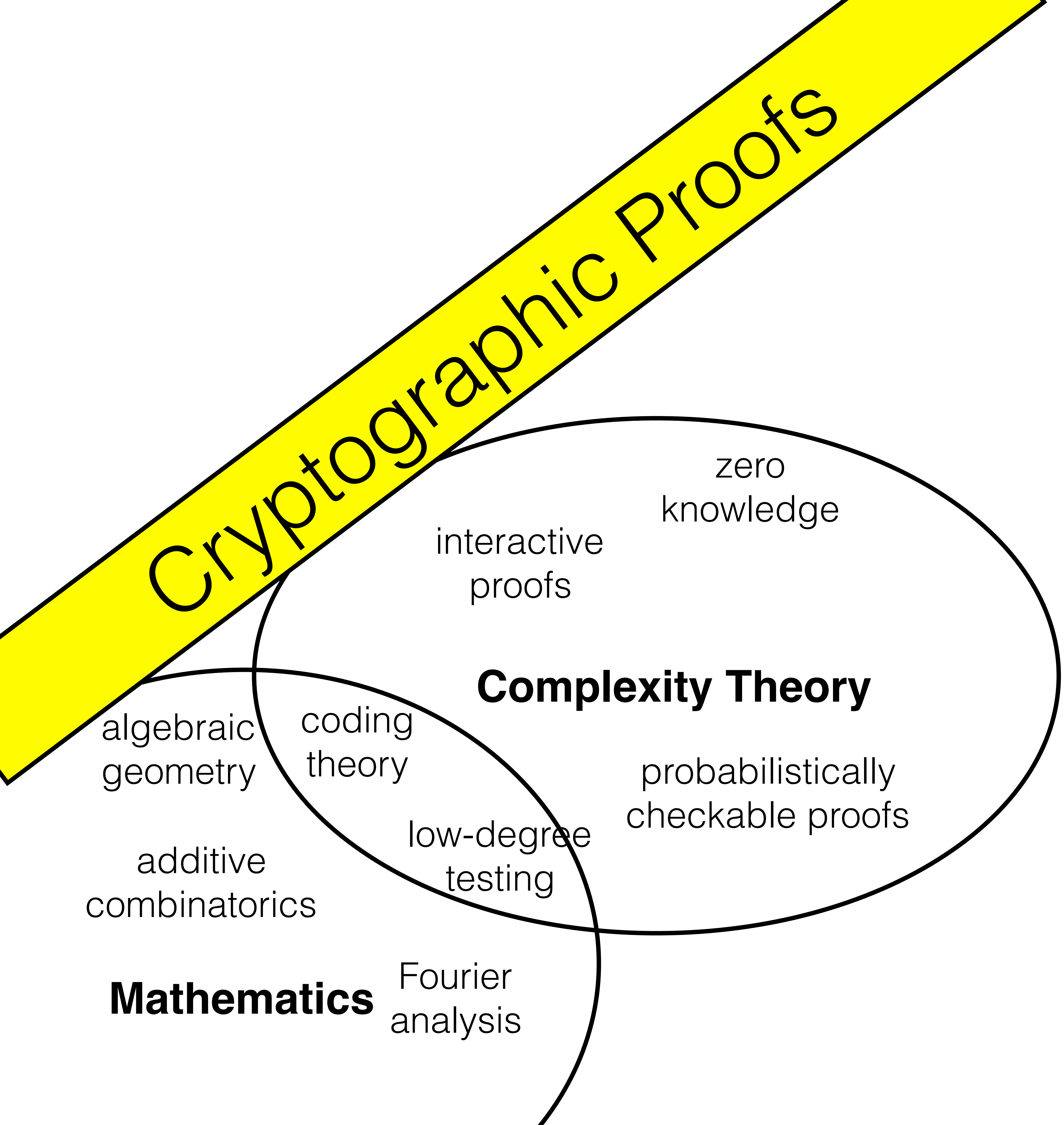
quasilinear [BS08][Din07]	best proof length without ZK	**linear** [B**C**GRS16]
polynomial [KPT97]	best proof length with ZK	**quasilinear** [B**C**GV16]
		cheaper ZK… [B**C**FGRS16][B**C**FS17]

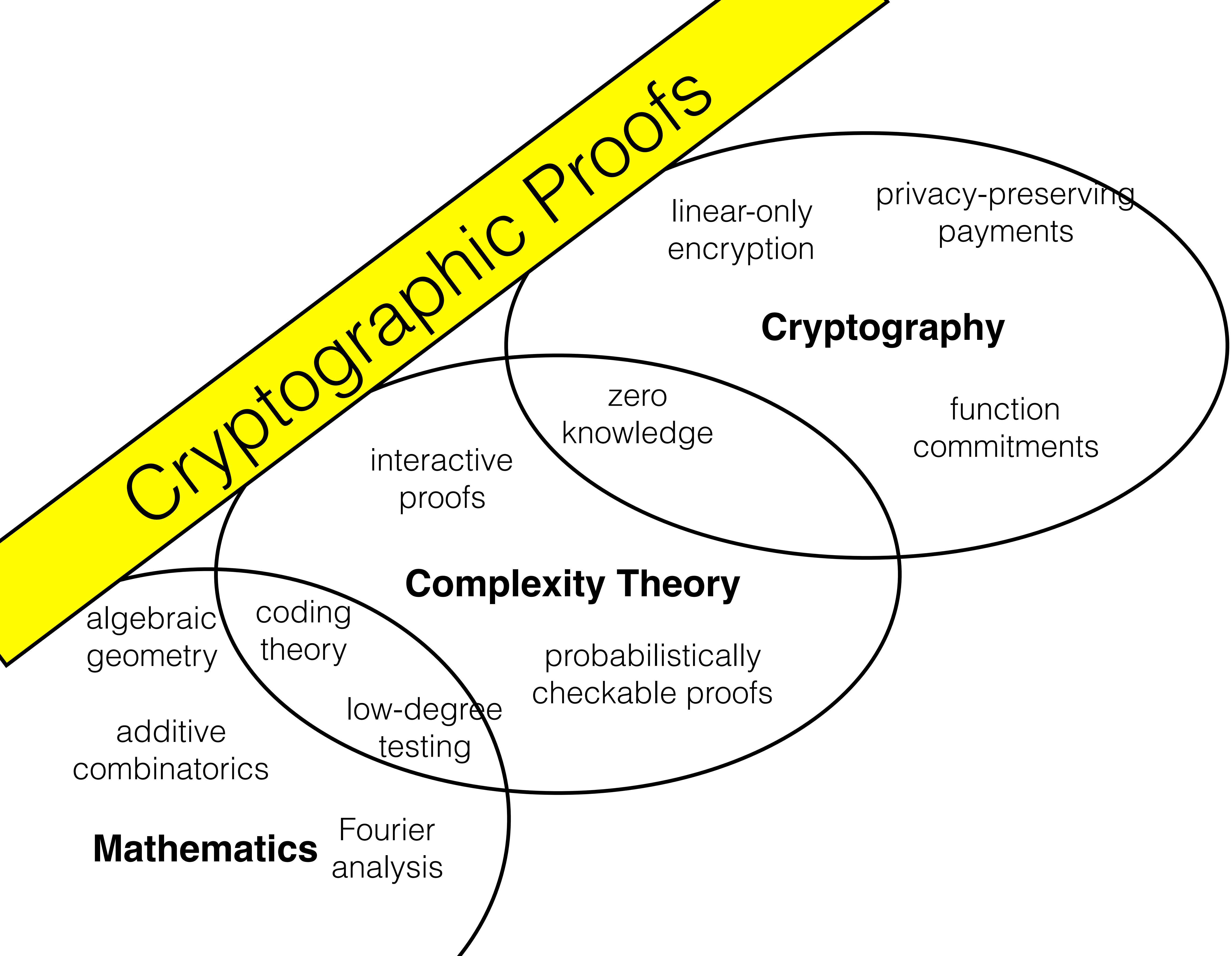
Encouraging progress, and it already improved working prototypes.

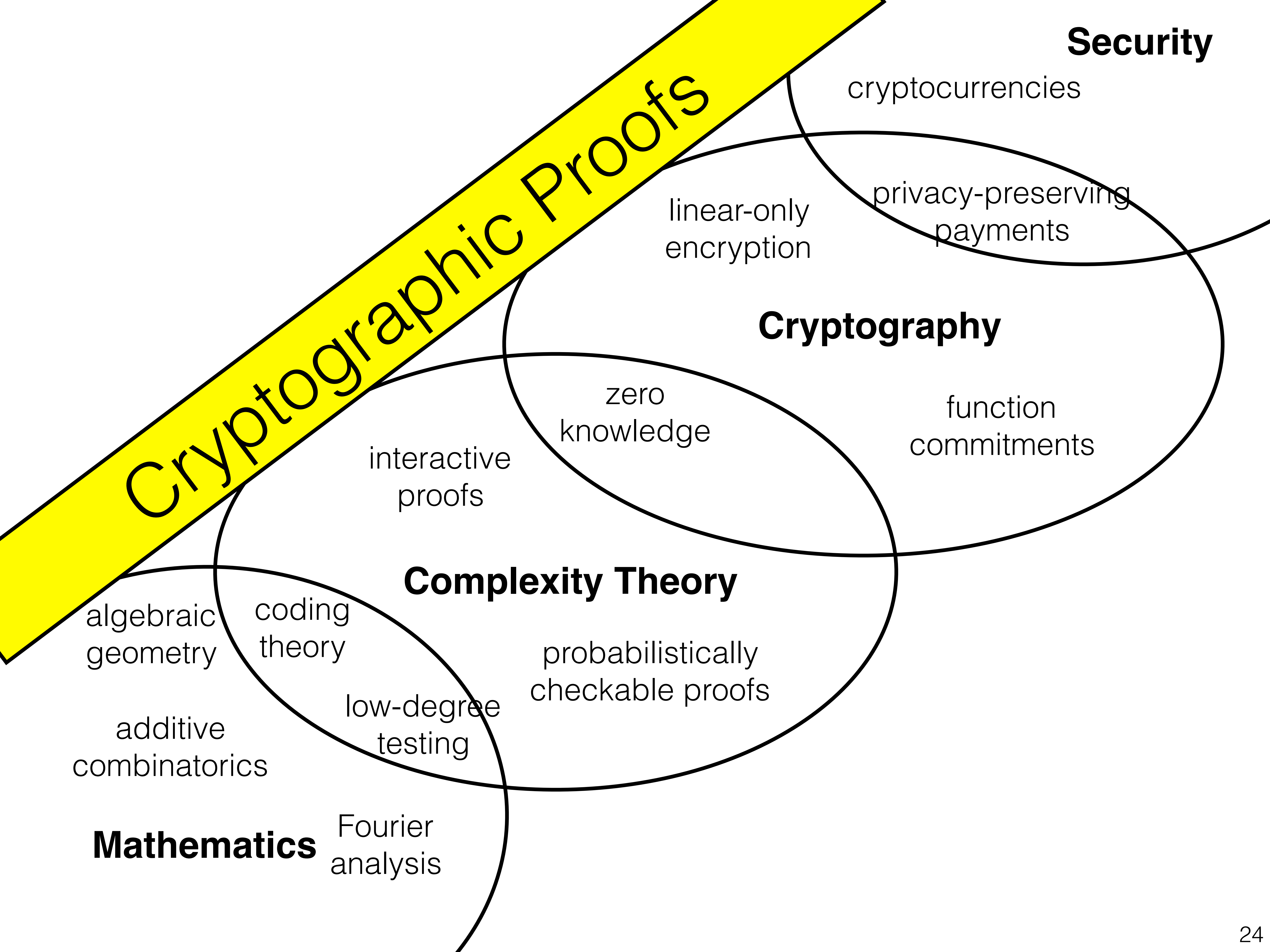
Still more research is needed for practical deployment.

Cryptographic Proofs

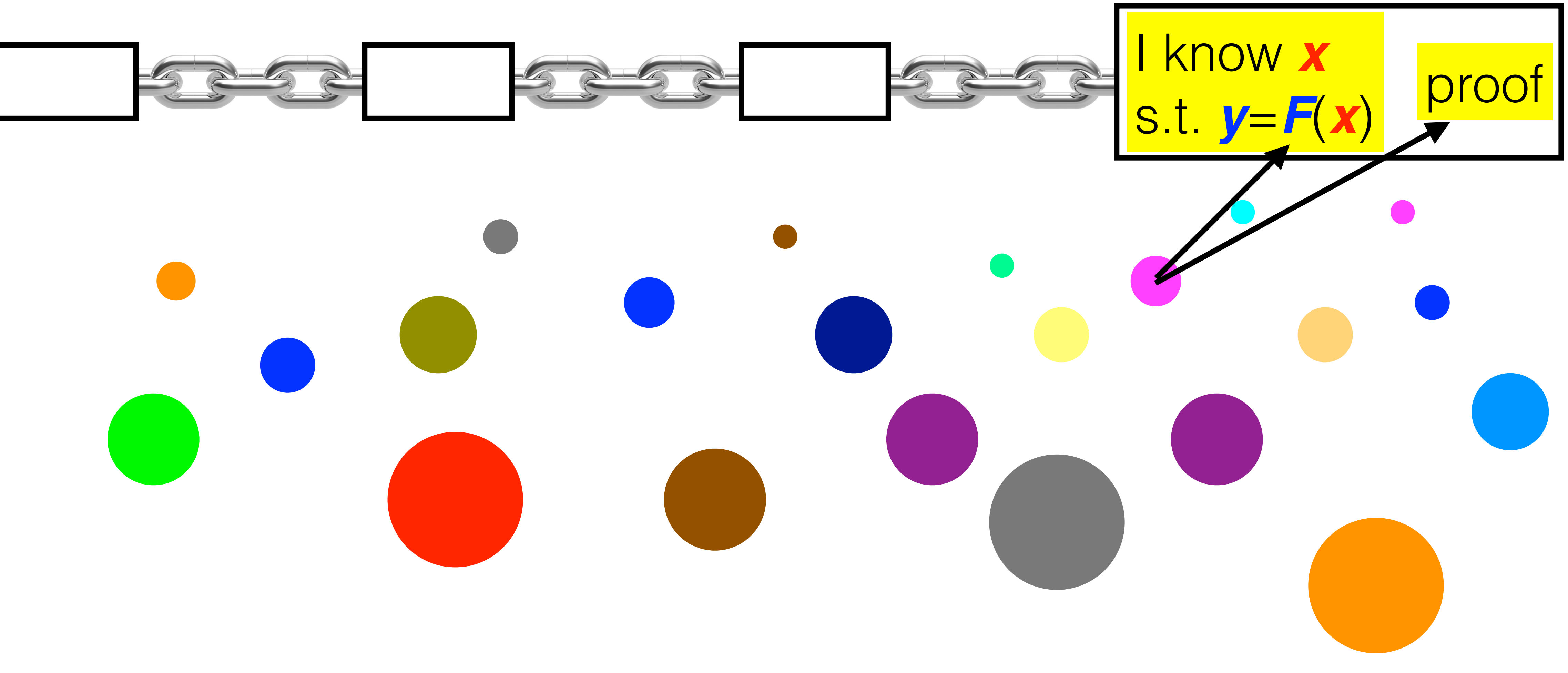

Cryptographic Proofs
algebraic geometry
coding theory
low-degree testing
additive combinatorics
Mathematics
Fourier analysis


Cryptographic Proofs
zero knowledge
interactive proofs
Complexity Theory
coding theory
algebraic geometry
probabilistically checkable proofs
low-degree testing
additive combinatorics
Mathematics
Fourier analysis

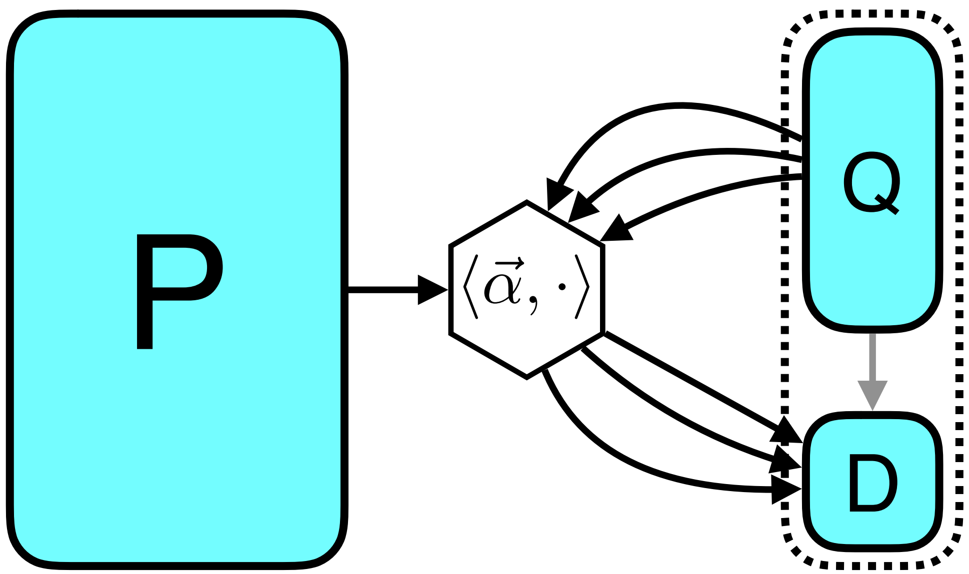

Cryptographic Proofs
linear-only encryption
privacy-preserving payments
Cryptography
zero knowledge
function commitments
interactive proofs
Complexity Theory
algebraic geometry
coding theory
probabilistically checkable proofs
low-degree testing
additive combinatorics
Mathematics
Fourier analysis


Cryptographic Proofs
Security
cryptocurrencies
privacy-preserving payments
linear-only encryption
Cryptography
zero knowledge
function commitments
interactive proofs
Complexity Theory
algebraic geometry
coding theory
probabilistically checkable proofs
low-degree testing
additive combinatorics
Mathematics
Fourier analysis

Thanks!



A Simple Linear PCP



Def: The language $\mathcal{L}$ consists of tuples

$$(p_1,\ldots,p_m)$$

where each p_i is a quadratic polynomial over **F** in n variables such that there is an assignment $\mathbf{w}=(w_1,\ldots,w_n)$ such that

$$p_1(\mathbf{w}) = \ldots = p_m(\mathbf{w}) = 0.$$

Theorem:

The language $\mathcal{L}$ has a linear PCP with

- proof length $(n+1)^2$,
- query complexity 3,
- soundness error $2/|\mathbf{F}|$.

A Simple Linear PCP

A Simple Linear PCP

Bundling. Let $r_1, \ldots, r_m \in \mathbb{F}$ be random and $\omega = (\omega_1, \ldots, \omega_n) \in \mathbb{F}^n$.
If $p_1(\omega) = \cdots = p_m(\omega) = 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability 1.
If $\exists j \in [m]$ s.t. $p_j(\omega) \neq 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability $\leq 1/|\mathbb{F}|$.

A Simple Linear PCP

Bundling. Let $r_1, \dots, r_m \in \mathbb{F}$ be random and $\omega = (\omega_1, \dots, \omega_n) \in \mathbb{F}^n$.
If $p_1(\omega) = \dots = p_m(\omega) = 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability 1.
If $\exists j \in [m]$ s.t. $p_j(\omega) \neq 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability $\leq 1/|\mathbb{F}|$.

Prover. Given an assignment $\omega = (\omega_1, \dots, \omega_n) \in \mathbb{F}^n$, the prover writes the linear function:

$$\vec{\alpha} := \begin{pmatrix} 1 & \omega_1 & \omega_2 & \dots & \omega_n \\ \omega_1 & \omega_1^2 & \omega_1\omega_2 & \dots & \omega_1\omega_n \\ \omega_2 & \omega_2\omega_1 & \omega_2^2 & \dots & \omega_2\omega_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_n & \omega_n\omega_1 & \omega_n\omega_2 & \dots & \omega_n^2 \end{pmatrix} .$$

A Simple Linear PCP

Bundling. Let $r_1, \ldots, r_m \in \mathbb{F}$ be random and $\omega = (\omega_1, \ldots, \omega_n) \in \mathbb{F}^n$.
If $p_1(\omega) = \cdots = p_m(\omega) = 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability 1.
If $\exists j \in [m]$ s.t. $p_j(\omega) \neq 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability $\leq 1/|\mathbb{F}|$.

Prover. Given an assignment $\omega = (\omega_1, \ldots, \omega_n) \in \mathbb{F}^n$, the prover writes the linear function:

$$\vec{\alpha} := \begin{pmatrix} 1 & \omega_1 & \omega_2 & \ldots & \omega_n \\ \omega_1 & \omega_1^2 & \omega_1\omega_2 & \ldots & \omega_1\omega_n \\ \omega_2 & \omega_2\omega_1 & \omega_2^2 & \ldots & \omega_2\omega_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_n & \omega_n\omega_1 & \omega_n\omega_2 & \ldots & \omega_n^2 \end{pmatrix} .$$

Verifier. The verifier has oracle access to some linear function

$$\vec{\alpha} = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} & \cdots & \alpha_{0,n} \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,n} \\ \alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,0} & \alpha_{n,1} & \alpha_{n,2} & \cdots & \alpha_{n,n} \end{pmatrix} .$$

and thinks of each quadratic polynomial p_i as

$$p_i = \begin{pmatrix} p_{i,0,0} & p_{i,0,1} & p_{i,0,2} & \cdots & p_{i,0,n} \\ 0 & p_{i,1,1} & p_{i,1,2} & \cdots & p_{i,1,n} \\ 0 & 0 & p_{i,2,2} & \cdots & p_{i,2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p_{i,n,n} \end{pmatrix} .$$

A Simple Linear PCP

Bundling. Let $r_1, \dots, r_m \in \mathbb{F}$ be random and $\omega = (\omega_1, \dots, \omega_n) \in \mathbb{F}^n$.
If $p_1(\omega) = \cdots = p_m(\omega) = 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability 1.
If $\exists j \in [m]$ s.t. $p_j(\omega) \neq 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability $\leq 1/|\mathbb{F}|$.

Prover. Given an assignment $\omega = (\omega_1, \dots, \omega_n) \in \mathbb{F}^n$, the prover writes the linear function:

$$\vec{\alpha} := \begin{pmatrix} 1 & \omega_1 & \omega_2 & \dots & \omega_n \\ \omega_1 & \omega_1^2 & \omega_1\omega_2 & \dots & \omega_1\omega_n \\ \omega_2 & \omega_2\omega_1 & \omega_2^2 & \dots & \omega_2\omega_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_n & \omega_n\omega_1 & \omega_n\omega_2 & \dots & \omega_n^2 \end{pmatrix} .$$

Verifier. The verifier has oracle access to some linear function

$$\vec{\alpha} = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} & \dots & \alpha_{0,n} \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,n} \\ \alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,0} & \alpha_{n,1} & \alpha_{n,2} & \dots & \alpha_{n,n} \end{pmatrix} .$$

and thinks of each quadratic polynomial p_i as

$$p_i = \begin{pmatrix} p_{i,0,0} & p_{i,0,1} & p_{i,0,2} & \dots & p_{i,0,n} \\ 0 & p_{i,1,1} & p_{i,1,2} & \dots & p_{i,1,n} \\ 0 & 0 & p_{i,2,2} & \dots & p_{i,2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & p_{i,n,n} \end{pmatrix} .$$

Verifier (cont'd). The verifier samples $r_1, \dots, r_m \in \mathbb{F}$ and $s_0, \dots, s_n \in \mathbb{F}$ at random and then generates three queries:

- $\vec{q}_1 := \sum_{i=1}^m r_i p_i$;
- $\vec{q}_2 := (s_0, s_1, \dots, s_n) \otimes (1, 0, \dots, 0)$;
- $\vec{q}_3 := (s_0, s_1, \dots, s_n) \otimes (s_0, s_1, \dots, s_n)$.

Upon receiving answers $a_1 := \langle \vec{\alpha}, \vec{q}_1 \rangle$, $a_2 := \langle \vec{\alpha}, \vec{q}_2 \rangle$, $a_3 := \langle \vec{\alpha}, \vec{q}_3 \rangle$, check that

$$a_1 = 0 \quad \text{and } a_2^2 = a_3 \ .$$

A Simple Linear PCP

Bundling. Let $r_1, \ldots, r_m \in \mathbb{F}$ be random and $\omega = (\omega_1, \ldots, \omega_n) \in \mathbb{F}^n$.
If $p_1(\omega) = \cdots = p_m(\omega) = 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability 1.
If $\exists j \in [m]$ s.t. $p_j(\omega) \neq 0$ then $\sum_{i=1}^m r_i p_i(\omega) = 0$ with probability $\leq 1/|\mathbb{F}|$.

Prover. Given an assignment $\omega = (\omega_1, \ldots, \omega_n) \in \mathbb{F}^n$, the prover writes the linear function:

$$\vec{\alpha} := \begin{pmatrix} 1 & \omega_1 & \omega_2 & \cdots & \omega_n \\ \omega_1 & \omega_1^2 & \omega_1\omega_2 & \cdots & \omega_1\omega_n \\ \omega_2 & \omega_2\omega_1 & \omega_2^2 & \cdots & \omega_2\omega_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_n & \omega_n\omega_1 & \omega_n\omega_2 & \cdots & \omega_n^2 \end{pmatrix} .$$

Verifier. The verifier has oracle access to some linear function

$$\vec{\alpha} = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} & \cdots & \alpha_{0,n} \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,n} \\ \alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,0} & \alpha_{n,1} & \alpha_{n,2} & \cdots & \alpha_{n,n} \end{pmatrix} .$$

and thinks of each quadratic polynomial p_i as

$$p_i = \begin{pmatrix} p_{i,0,0} & p_{i,0,1} & p_{i,0,2} & \cdots & p_{i,0,n} \\ 0 & p_{i,1,1} & p_{i,1,2} & \cdots & p_{i,1,n} \\ 0 & 0 & p_{i,2,2} & \cdots & p_{i,2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p_{i,n,n} \end{pmatrix} .$$

Verifier (cont'd). The verifier samples $r_1, \ldots, r_m \in \mathbb{F}$ and $s_0, \ldots, s_n \in \mathbb{F}$ at random and then generates three queries:

- $\vec{q}_1 := \sum_{i=1}^m r_i p_i$;
- $\vec{q}_2 := (s_0, s_1, \ldots, s_n) \otimes (1, 0, \ldots, 0)$;
- $\vec{q}_3 := (s_0, s_1, \ldots, s_n) \otimes (s_0, s_1, \ldots, s_n)$.

Upon receiving answers $a_1 := \langle \vec{\alpha}, \vec{q}_1 \rangle$, $a_2 := \langle \vec{\alpha}, \vec{q}_2 \rangle$, $a_3 := \langle \vec{\alpha}, \vec{q}_3 \rangle$, check that

$$a_1 = 0 \quad \text{and } a_2^2 = a_3 \ .$$

Proof. Observe that:

- $a_1 = \langle \vec{\alpha}, \vec{q}_1 \rangle = \langle \vec{\alpha}, \sum_{i=1}^m r_i \cdot p_i \rangle = \sum_{i=1}^m r_i \cdot \langle \vec{\alpha}, p_i \rangle$;
- $a_2 = \langle \vec{\alpha}, \vec{q}_2 \rangle = \sum_{i=0}^n \alpha_{0,i} \cdot s_i$, so that $a_2^2 = \sum_{i,j=0}^n \alpha_{0,i}\alpha_{0,j} \cdot s_i s_j$;
- $a_3 = \langle \vec{\alpha}, \vec{q}_3 \rangle = \sum_{i,j=0}^n \alpha_{i,j} \cdot s_i s_j$.

Distinguish between two cases:

1. There are i, j such that $\alpha_{i,j} \neq \alpha_i \alpha_j$.
   Then $\Pr_{s_0,\ldots,s_n}[a_2^2 = a_3] \leq 2/|\mathbb{F}|$.
2. For all i, j it holds that $\alpha_{i,j} = \alpha_i \alpha_j$.
   Thus there is $(\omega_0, \ldots, \omega_n)$ such that $\vec{\alpha} = (\omega_0, \ldots, \omega_n) \otimes (\omega_0, \ldots, \omega_n)$.
   WLOG can assume that $\omega_0 = 1$.
   Then $\langle \vec{\alpha}, p_i \rangle = p_i(\omega)$ for every i, so that $a_1 = \sum_{i=1}^m r_i p_i(\omega)$.