Using Zero-Knowledge Proofs to Fight Disinformation

Trisha Datta and Dan Boneh

Stanford University

Ukraine conflict: Many misleading images have been shared online

By Alistair Coleman & Shayan Sardarizadeh
BBC Monitoring

24 February 2022

Ukraine conflict: Many misleading images have been shared online

By Alistair Cole
BBC Monitorin

24 February 202

Fact-checking videos and pictures from Ukraine

Since Russia's attacks on Ukraine began, we have seen several videos and pictures go viral that are actually fake posts.

Ukraine conflict: Many misleading images have been shared online

By Alistair Cole
BBC Monitorin

24 February 202

Fact-checking videos and pictures from Ukraine

Since Rus
and pictu

False social media posts are hindering earthquake relief efforts in Turkey. You can help stop that

Ukraine conflict: Many misleading images have been shared online

By Alistair Cole
BBC Monitorin

24 February 202

Fact-checking videos and pictures from Ukraine

Since Rus
and pictu

False social media posts are hindering earthquake relief efforts in Turkey. You can help sto

THE NEWS / UKRAINE

BBC Breakfast uses old footage of Russian parade rehearsal to show invasion of Ukraine

C2PA: A Content Provenance Standard

Sony Unlocks In-Camera Forgery-Proof Technology

04 Aug, 2022

embedded certified signing key sk_1

Adobe arm BBC intel Microsoft SONY Truepic

C2PA: A Content Provenance Standard

Sony Unlocks In-Camera Forgery-Proof Technology

04 Aug, 2022

embedded certified signing key sk_1

Adobe arm BBC intel Microsoft SONY Truepic

C2PA: A Content Provenance Standard

Sony Unlocks In-Camera Forgery-Proof Technology

04 Aug, 2022

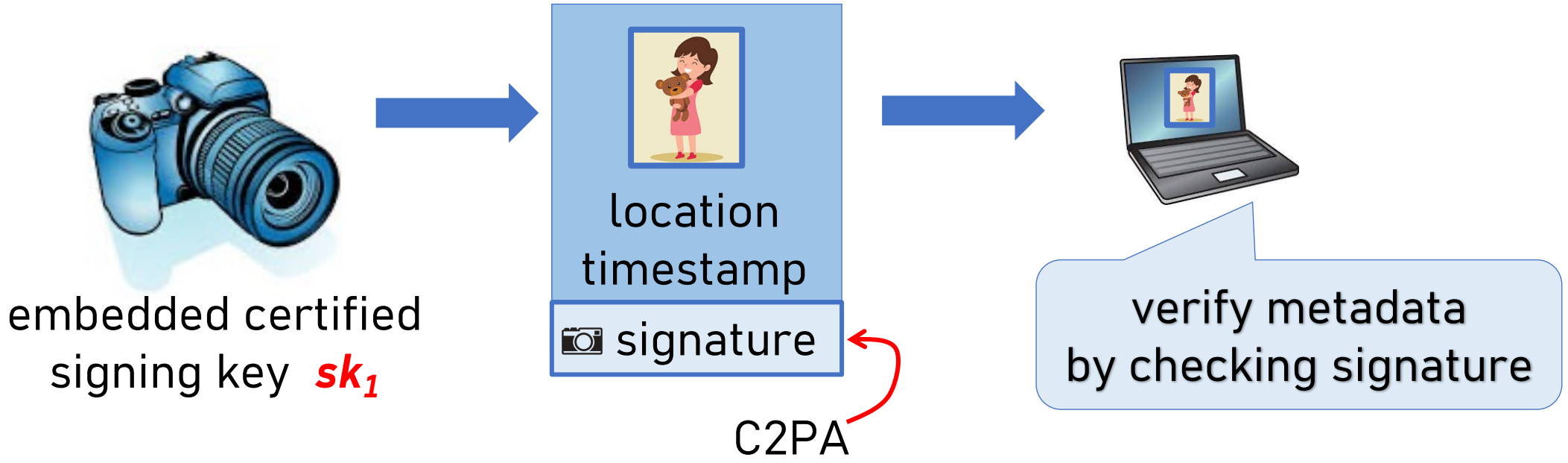
embedded certified signing key sk_1

Adobe arm BBC intel Microsoft SONY Truepic

C2PA: A Content Provenance Standard

Sony Unlocks In-Camera Forgery-Proof Technology

04 Aug, 2022



Adobe arm BBC intel Microsoft SONY Truepic

C2PA: A Content Provenance Standard

Sony Unlocks In-Camera Forgery-Proof Technology

04 Aug, 2022

See Rivadeneira, "Harms Modelling in the C2PA," 2022 for security analysis of C2PA

embedded certified signing key sk_1

verify metadata by checking signature

Adobe arm BBC intel Microsoft SONY Truepic

A Problem: Post-Processing

- Newspapers often process photos before publication
  - At minimum, need to resize (90 MB → 8 MB)
  - Allowable operations from the *Associated Press:*

    cropping, grayscale, …

A Problem: Post-Processing

- Newspapers often process photos before publication
  - At minimum, need to resize (90 MB → 8 MB)
  - Allowable operations from the *Associated Press:*

    cropping, grayscale, …

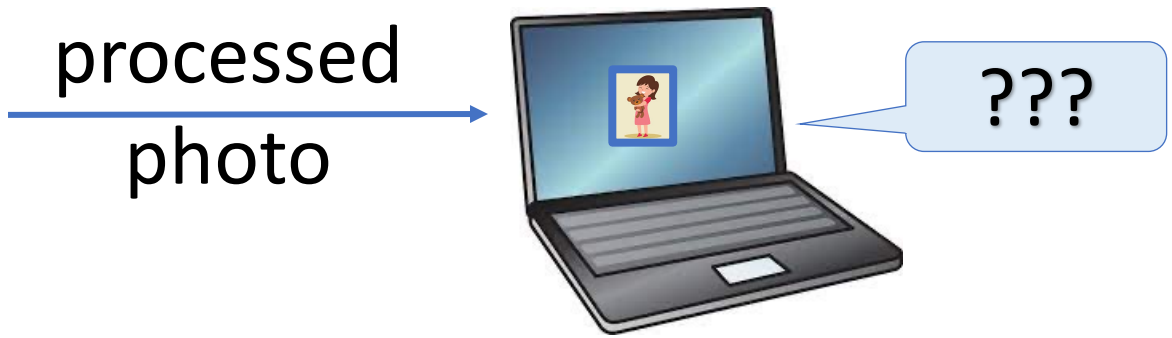
Problem: browser cannot verify the C2PA signature of a processed photo

A Problem: Post-Processing

- Newspapers often process photos before publication
  - At minimum, need to resize (90 MB → 8 MB)
  - Allowable operations from the *Associated Press:*

    cropping, grayscale, …

Problem: browser cannot verify the C2PA signature of a processed photo

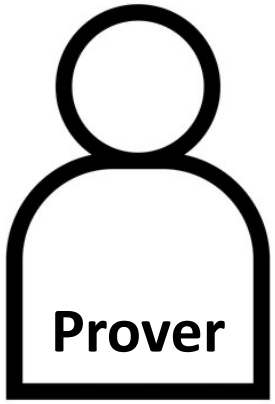


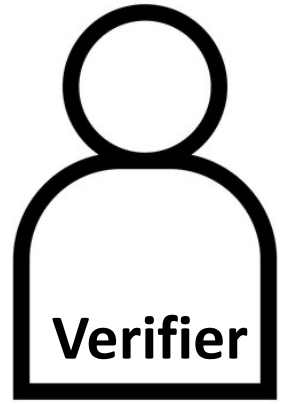
Proposed Solution: ZK-SNARKs!

- ZK-SNARK: efficiently verifiable statement about a secret witness

Proposed Solution: ZK-SNARKs!

- ZK-SNARK: efficiently verifiable statement about a secret witness



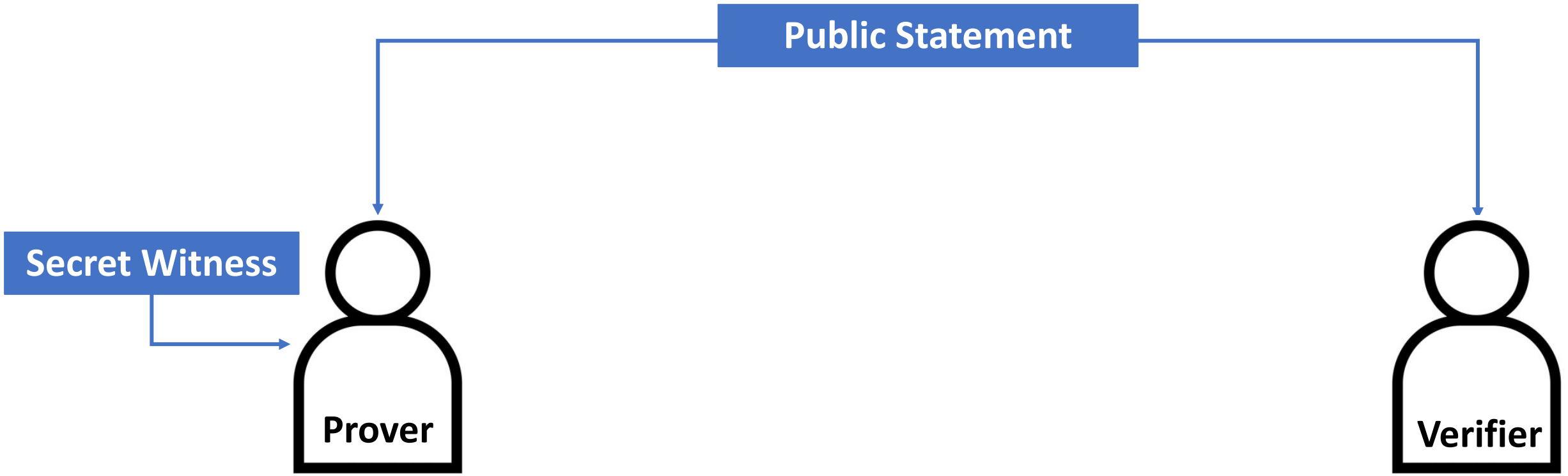


Proposed Solution: ZK-SNARKs!

- ZK-SNARK: efficiently verifiable statement about a secret witness

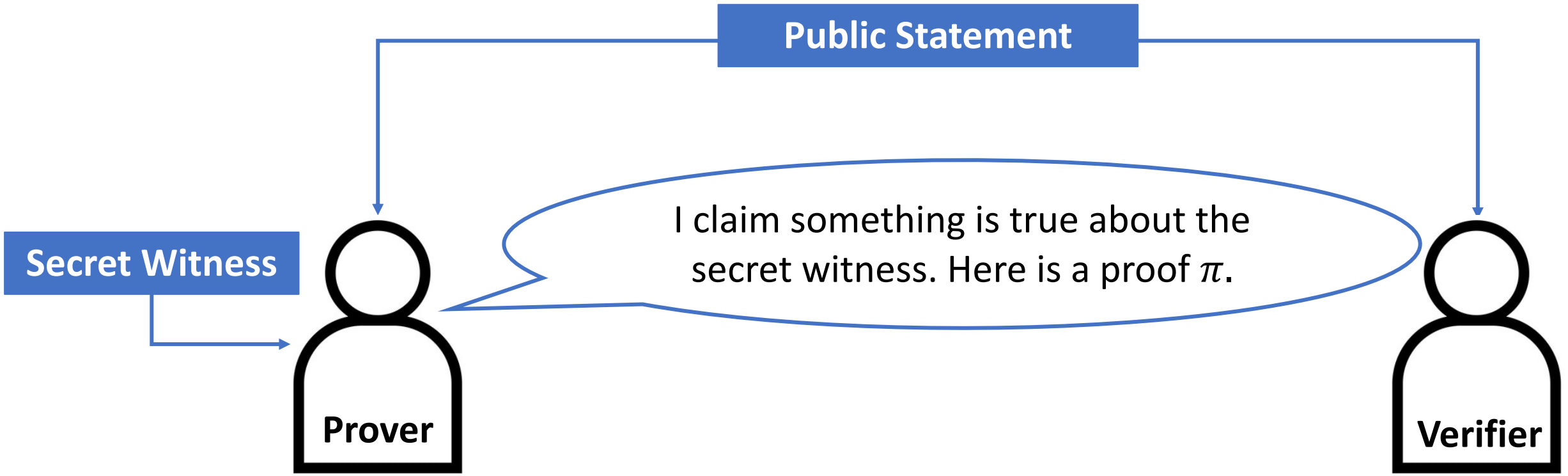
Proposed Solution: ZK-SNARKs!

- ZK-SNARK: efficiently verifiable statement about a secret witness



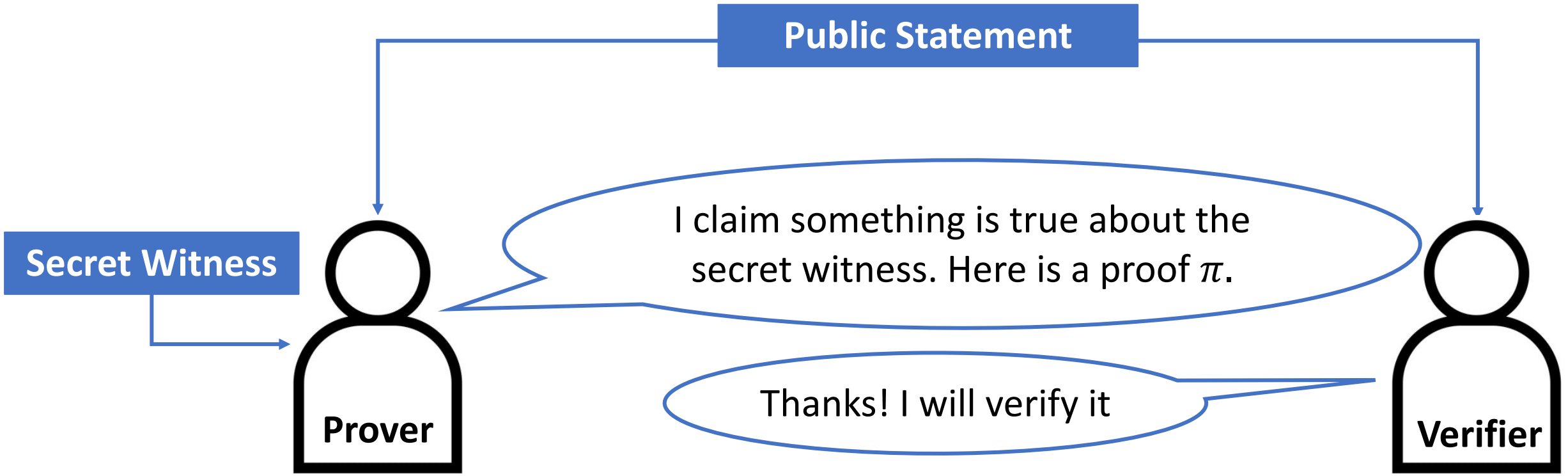
Proposed Solution: ZK-SNARKs!

- ZK-SNARK: efficiently verifiable statement about a secret witness



Proposed Solution: ZK-SNARKs!

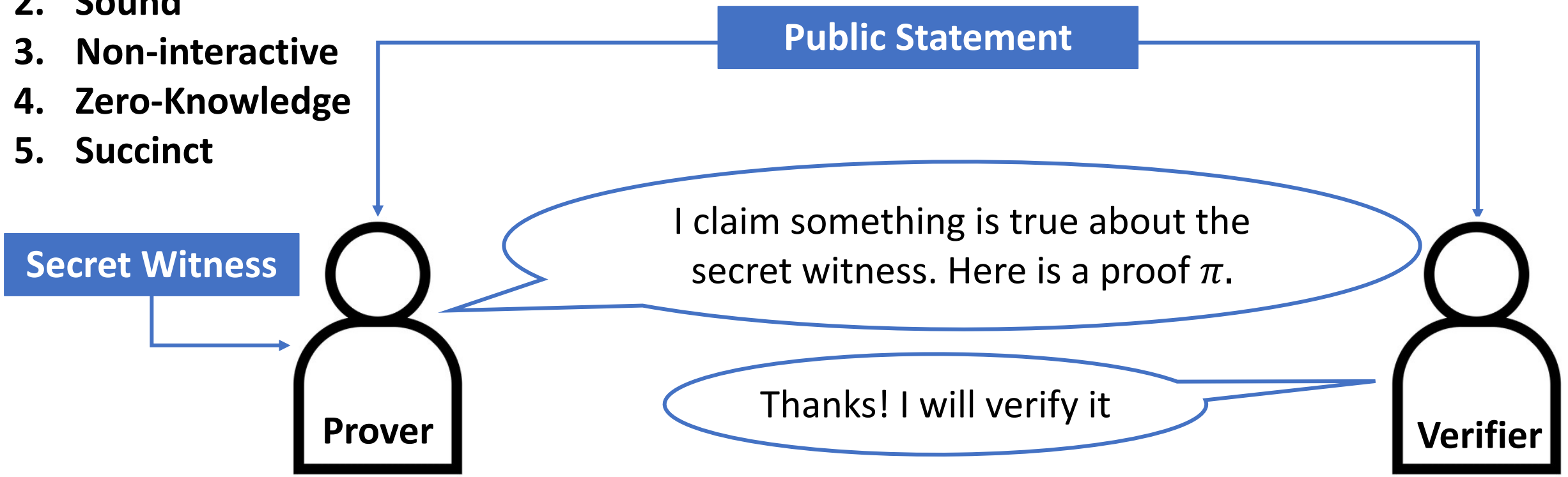
- ZK-SNARK: efficiently verifiable statement about a secret witness



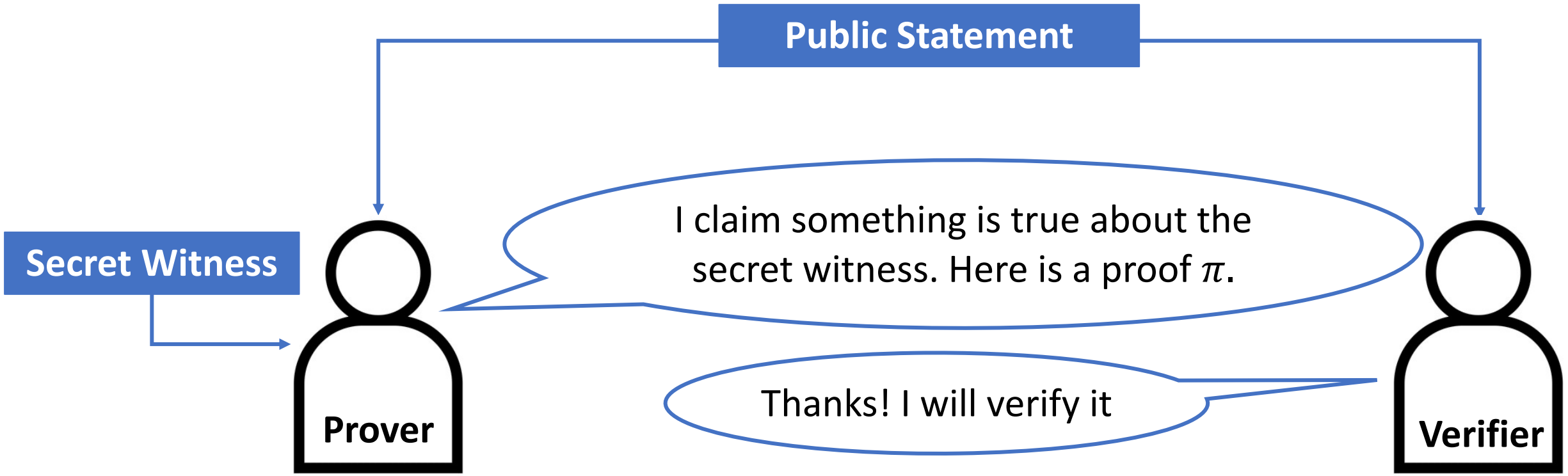
Proposed Solution: ZK-SNARKs!

- ZK-SNARK: efficiently verifiable statement about a secret witness

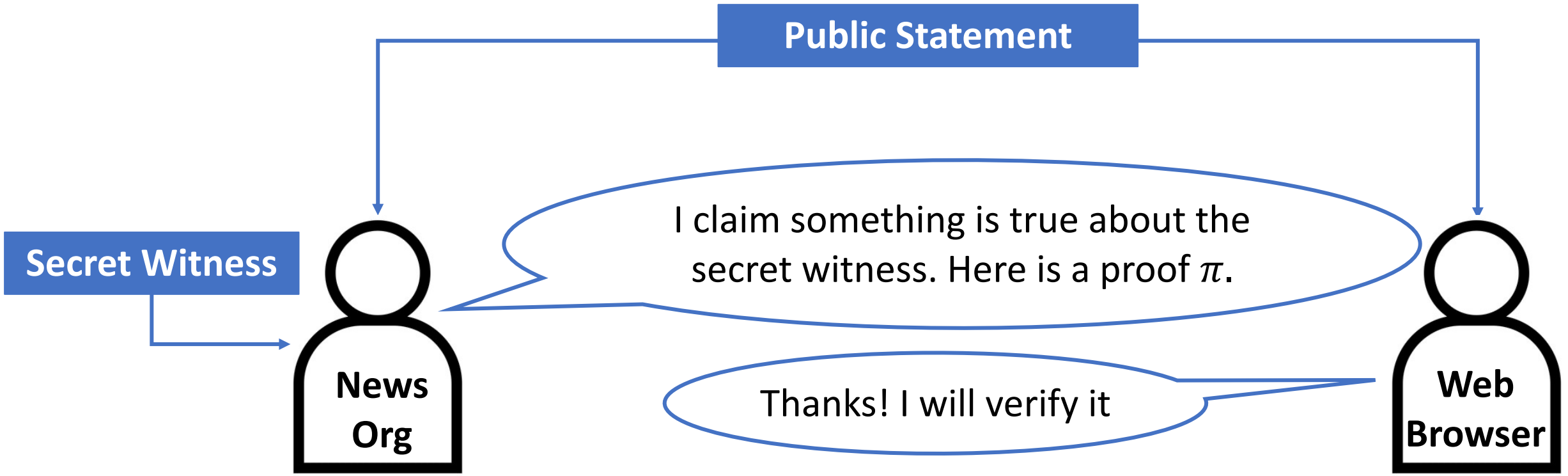
1. **Complete**
2. **Sound**
3. **Non-interactive**
4. **Zero-Knowledge**
5. **Succinct**



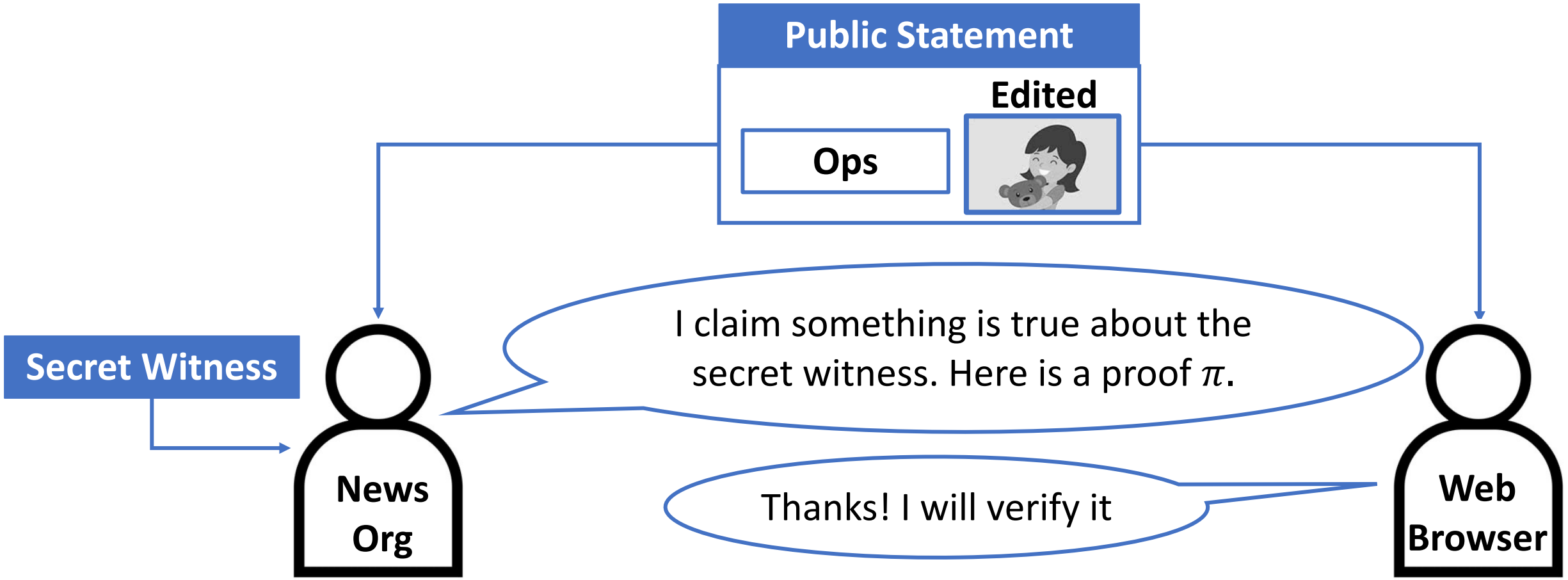
Proposed Solution: ZK-SNARKs!



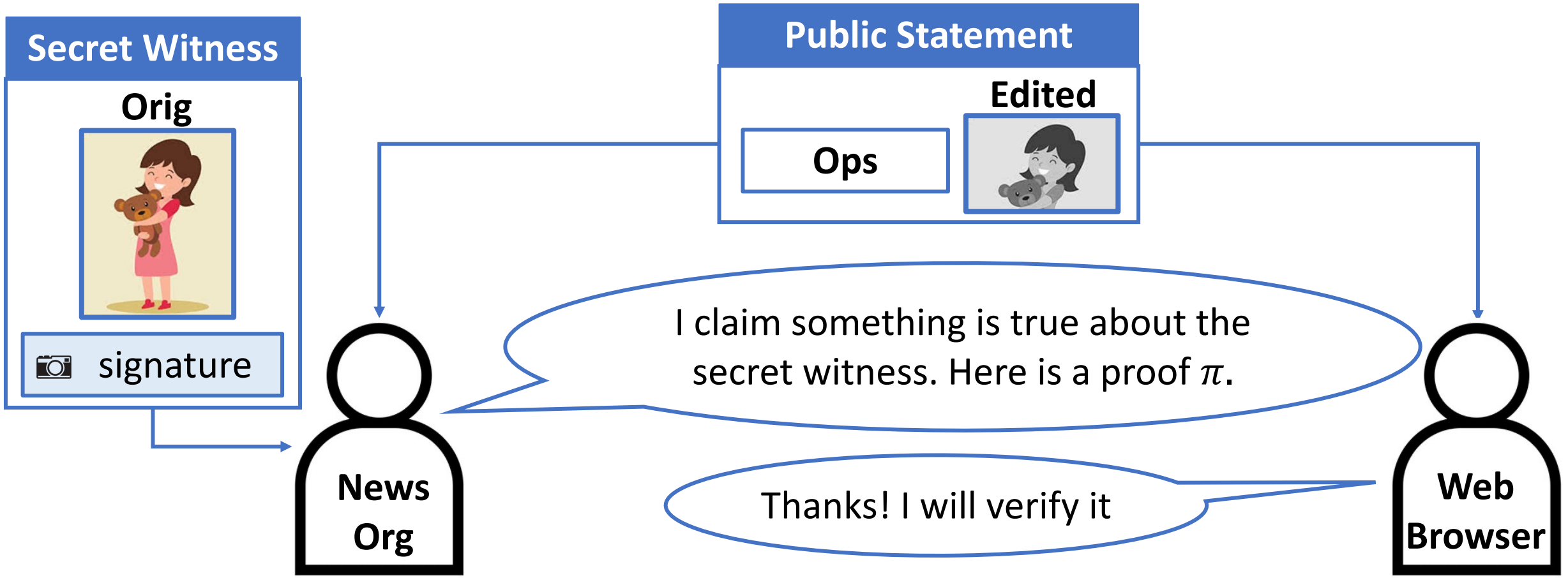
Proposed Solution: ZK-SNARKs!



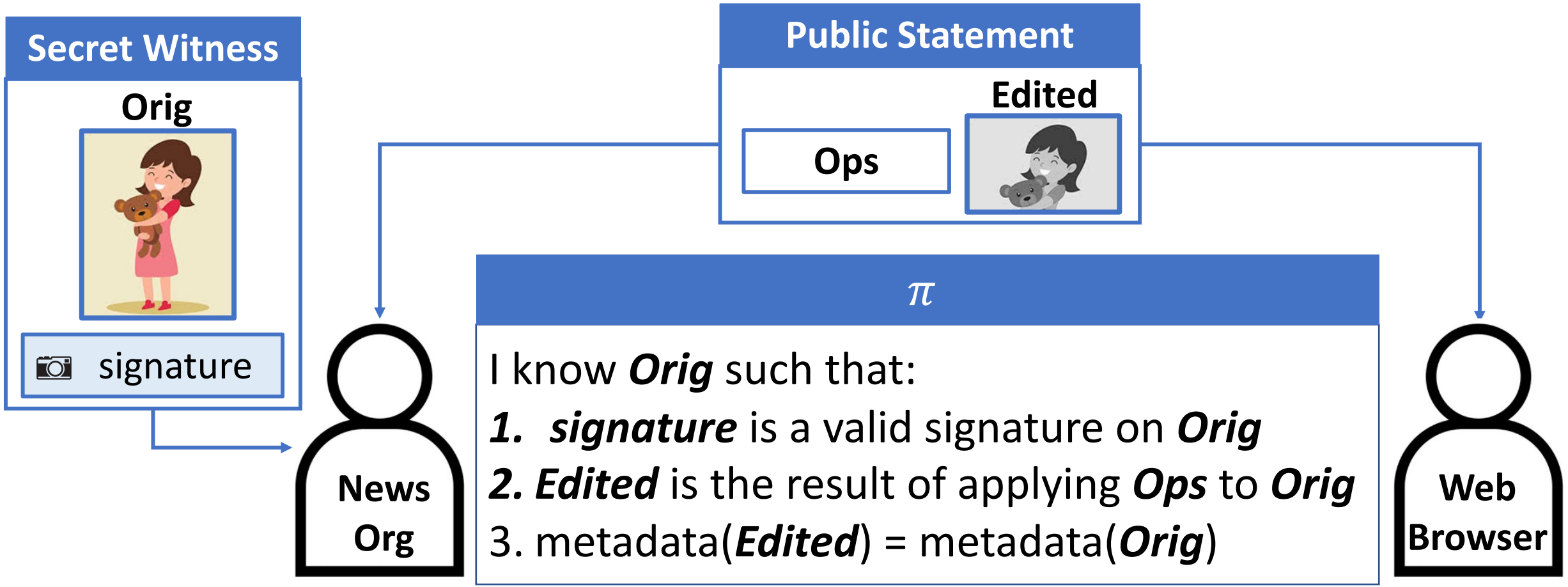
Proposed Solution: ZK-SNARKs!



Proposed Solution: ZK-SNARKs!



Proposed Solution: ZK-SNARKs!



Proposed Solution: ZK-SNARKs!

We rely on all properties of ZK-SNARKs!

Secret Witness

Orig

signature

Public Statement

Edited

Ops

News Org

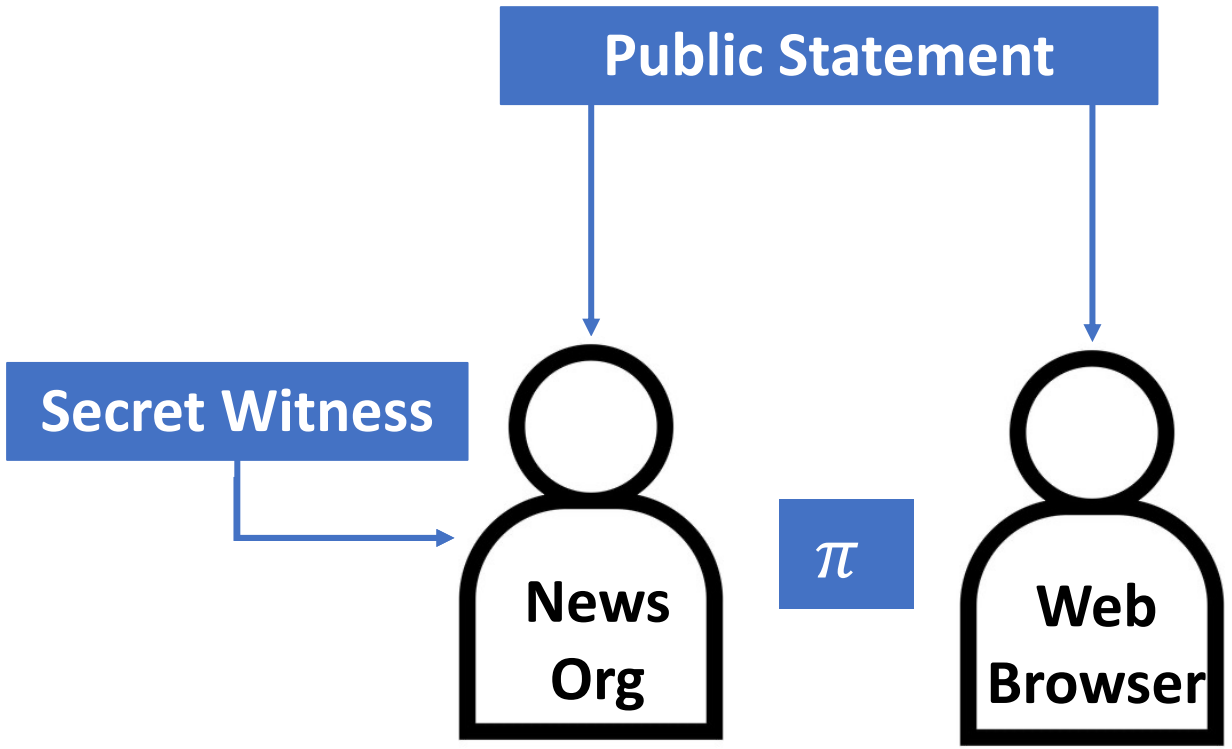
π

I know ***Orig*** such that:

1. ***signature*** is a valid signature on ***Orig***
2. ***Edited*** is the result of applying ***Ops*** to ***Orig***
3. metadata(***Edited***) = metadata(***Orig***)

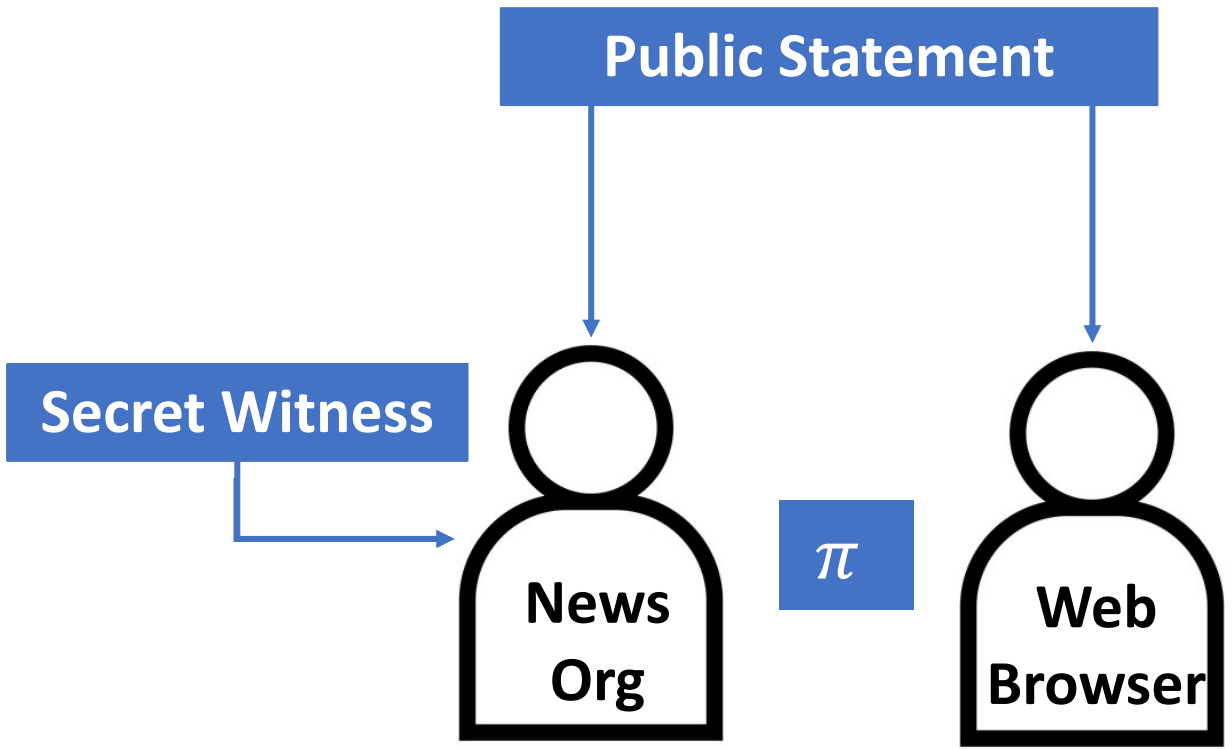
Web Browser

Proposed Solution: ZK-SNARKs!



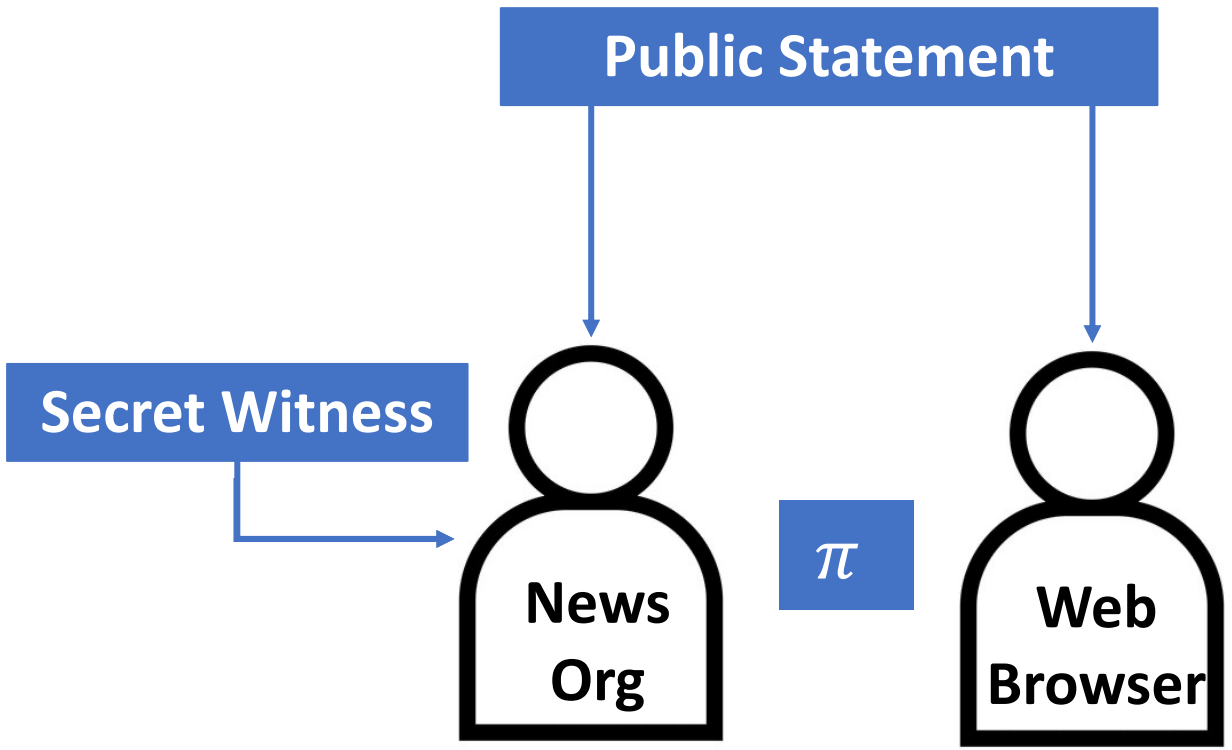
1. **Complete/Sound:** verifier doesn't need to trust prover

Proposed Solution: ZK-SNARKs!



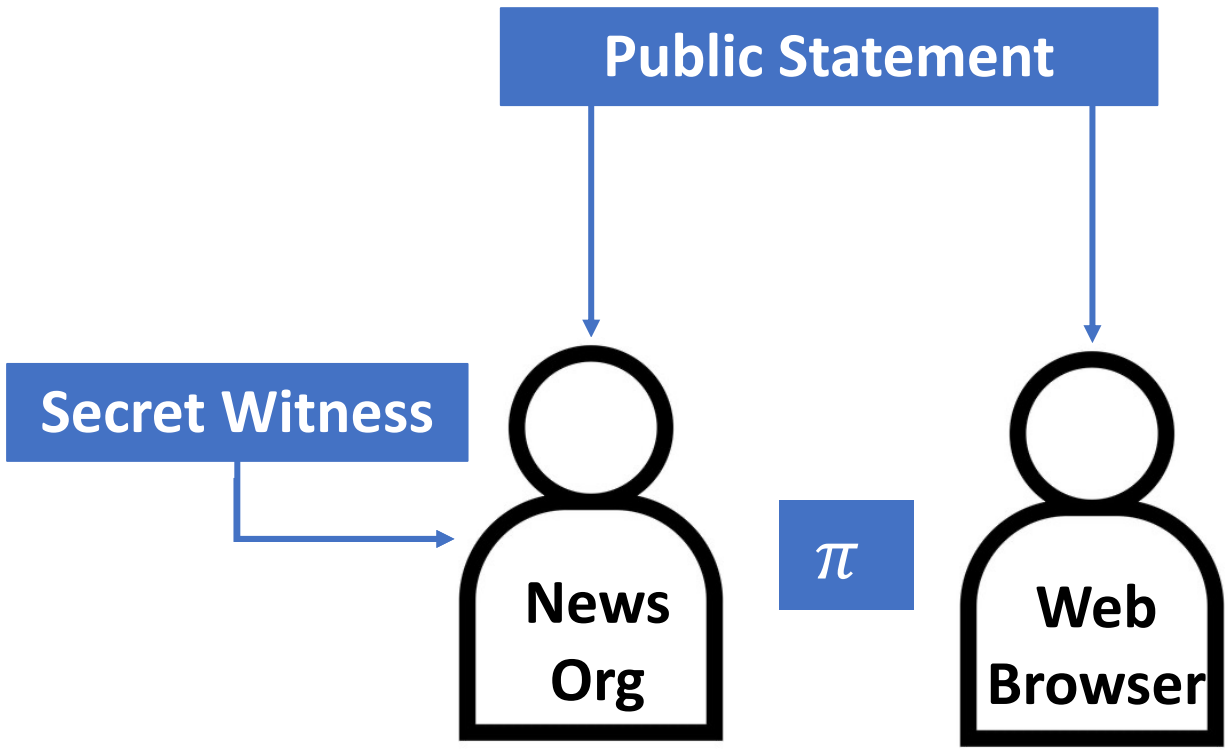
1. **Complete/Sound:** verifier doesn't need to trust prover
2. **Non-interactive:** interactivity would entail unique proofs for each browser

Proposed Solution: ZK-SNARKs!



1. **Complete/Sound:** verifier doesn't need to trust prover
2. **Non-interactive:** interactivity would entail unique proofs for each browser
3. **Zero-Knowledge:** useful for ops such as cropping

Proposed Solution: ZK-SNARKs!



1. **Complete/Sound:** verifier doesn't need to trust prover
2. **Non-interactive:** interactivity would entail unique proofs for each browser
3. **Zero-Knowledge:** useful for ops such as cropping
4. **Succinct:** web browser can efficiently verify proof

Proofs for Post-Processing Ops

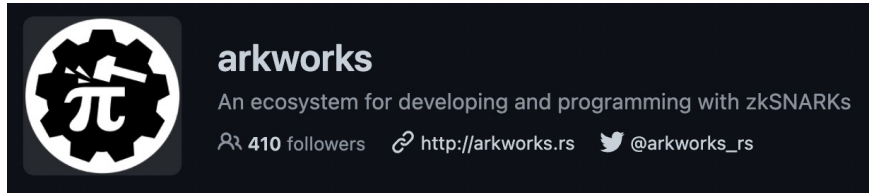
- PhotoProof (Naveh and Tromer, 2016): a few minutes to generate photo editing proofs for 128 x 128 pixel image

Proofs for Post-Processing Ops

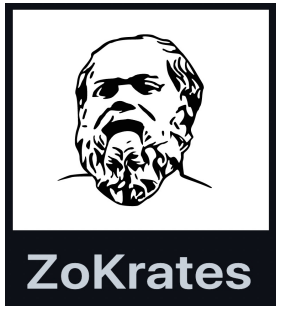
- PhotoProof (Naveh and Tromer, 2016): a few minutes to generate photo editing proofs for 128 x 128 pixel image
- New tools enable faster development!

==> circom
CIRCUIT COMPILER

halo2 crates.io v0.1.0-beta.2



gnark
A fast zk-SNARK library that offers a high-level API to design circuits



The Noir Programming Language

Performance for Post-Processing Ops Proofs

For resizing, cropping, grayscale ops on images of about 6000 x 4000 pixels (~30MP) using Circom:

- Proof generation time: <1 second
- Witness generation time: <4 minutes

by newspaper once per image

- Verification time: 2 ms
- Proof size: <1 KB

by browser

Verifying Signatures in a SNARK Prover

π
I know ***Orig*** such that: ***1. signature*** is a valid signature on ***Orig*** ***2. Edited*** is the result of applying ***Ops*** to ***Orig*** 3. metadata(***Edited***) = metadata(***Orig***)

Verifying Signatures in a SNARK Prover

π
I know ***Orig*** such that: ***1. signature*** is a valid signature on ***Orig*** ***2. Edited*** is the result of applying ***Ops*** to ***Orig*** 3. metadata(***Edited***) = metadata(***Orig***)

Verifying Signatures in a SNARK Prover

π
I know ***Orig*** such that: 1. ***signature*** is a valid signature on ***Orig*** ? 2. ***Edited*** is the result of applying ***Ops*** to ***Orig*** ✓ 3. metadata(***Edited***) = metadata(***Orig***)

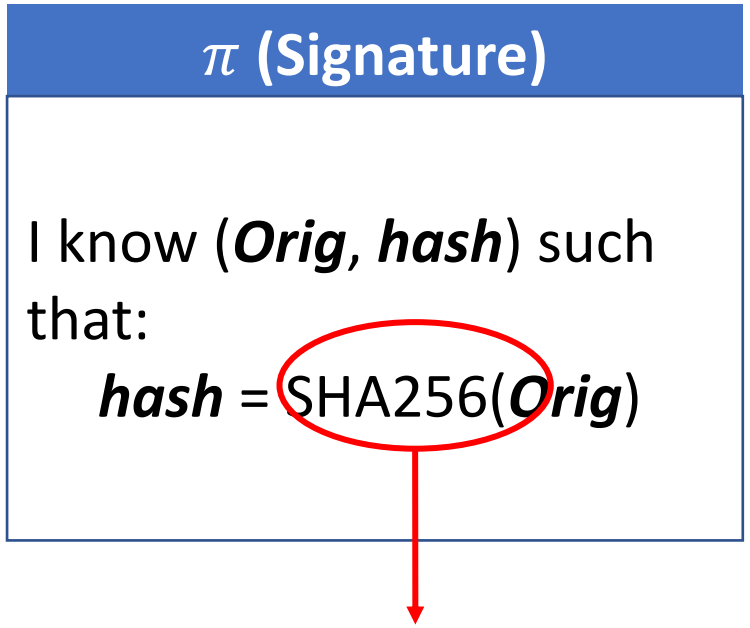
Verifying Signatures in a SNARK Prover

Attempt 1

π **(Signature)**
I know (***Orig***, ***hash***) such that: ***hash*** = SHA256(***Orig***)

Verifying Signatures in a SNARK Prover

Attempt 1



Verifying Signatures in a SNARK Prover

Attempt 1

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = SHA256(***Orig***)

Too slow for 30 MP!

Attempt 2

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(***Orig***)

Verifying Signatures in a SNARK Prover

Attempt 1

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = SHA256(***Orig***)

Too slow for 30 MP!

Attempt 2

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(***Orig***)

SNARK-friendly hash

Verifying Signatures in a SNARK Prover

Attempt 1

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = SHA256(***Orig***)

Too slow for 30 MP!

Attempt 2

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(***Orig***)

SNARK-friendly hash …but still too slow for 30MP!

Verifying Signatures in a SNARK Prover

Attempt 1

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = SHA256(***Orig***)

Too slow for 30 MP!

Attempt 2

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(***Orig***)

SNARK-friendly hash …but still too slow for 30MP!

Attempt 3

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(LatticeHash(***Orig***))

Verifying Signatures in a SNARK Prover

Attempt 1

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = SHA256(***Orig***)

Too slow for 30 MP!

Attempt 2

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(***Orig***)

SNARK-friendly hash …but still too slow for 30MP!

Attempt 3

π (Signature)

I know (***Orig***, ***hash***) such that:

hash = Poseidon(LatticeHash(***Orig***))

Verifying Signatures in a SNARK Prover

Poseidon hash of lattice hash [GGH'96 , SCMPGLW'08]

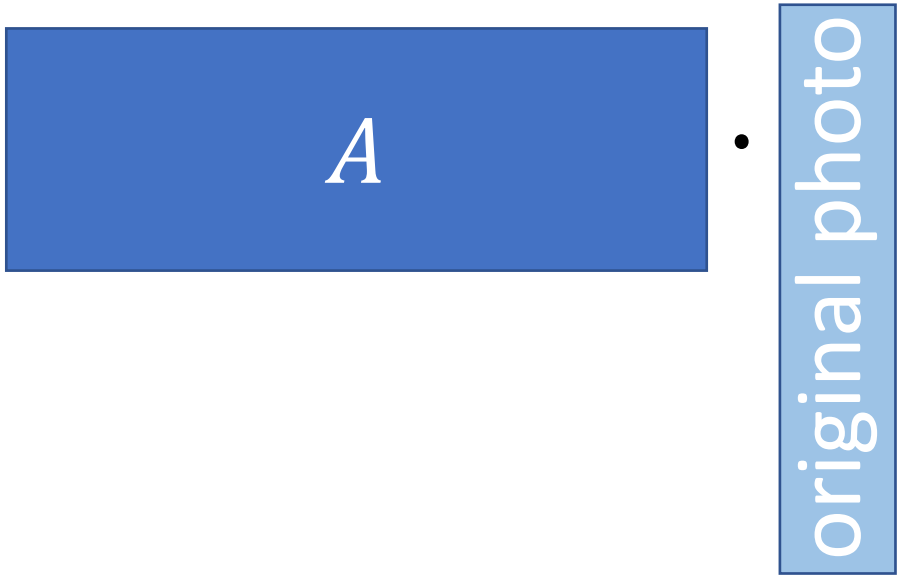
Verifying Signatures in a SNARK Prover

Poseidon hash of lattice hash [GGH'96 , SCMPGLW'08]

- A is random matrix from finite field $\mathbb{F}_q$

Verifying Signatures in a SNARK Prover

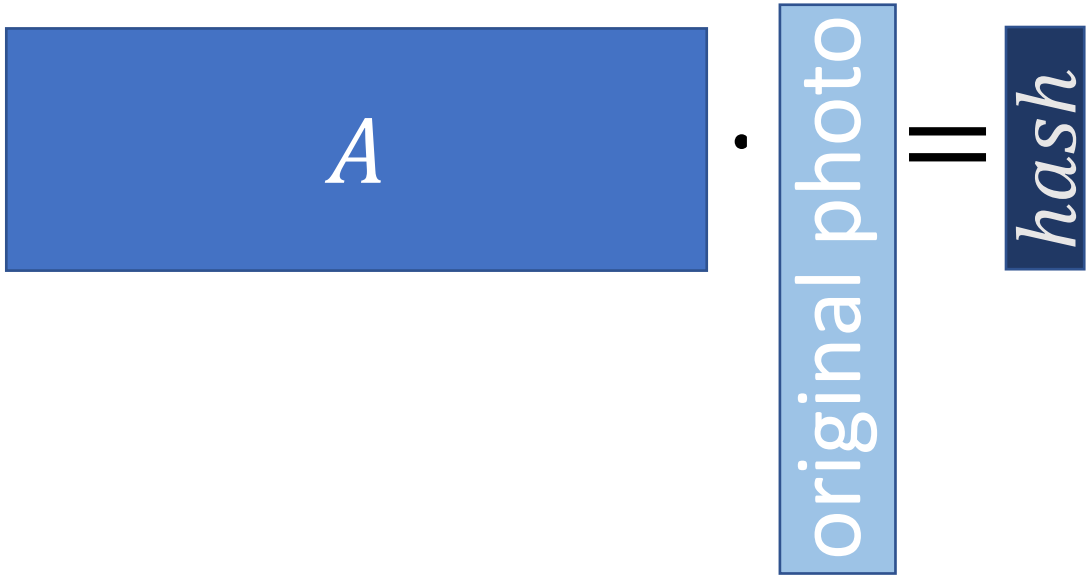
Poseidon hash of lattice hash [GGH'96 , SCMPGLW'08]



- A is random matrix from finite field $\mathbb{F}_q$
- $\vec{x}$ is low norm vector in $\mathbb{F}_q$ representing the input

Verifying Signatures in a SNARK Prover

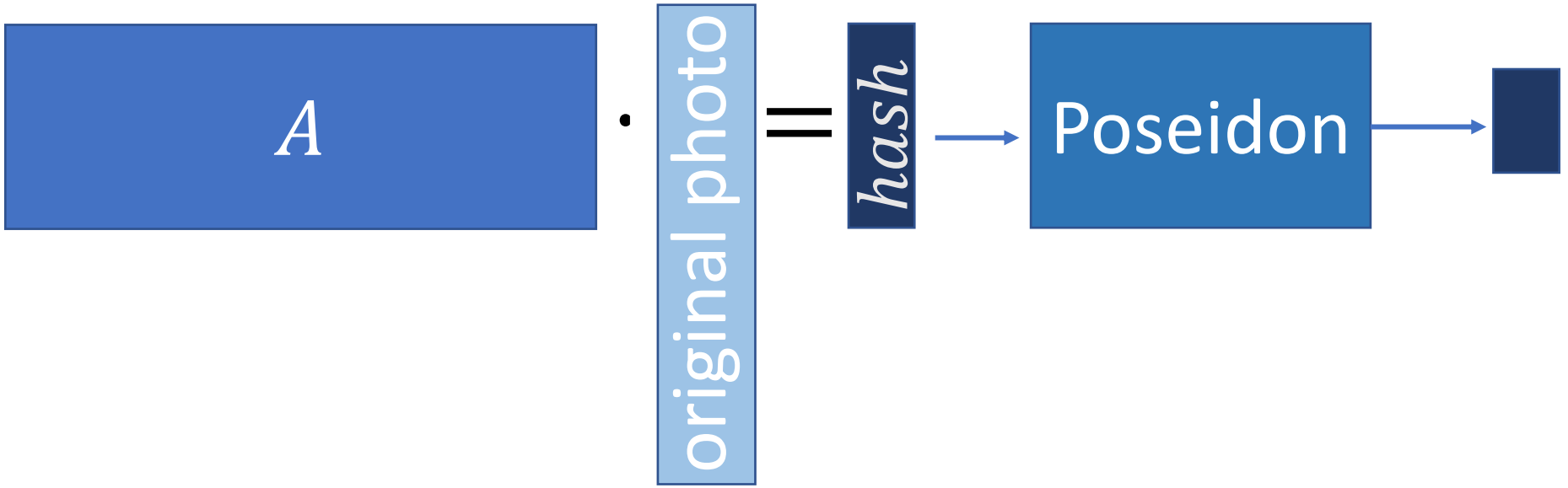
Poseidon hash of lattice hash [GGH'96 , SCMPGLW'08]



- A is random matrix from finite field $\mathbb{F}_q$
- $\vec{x}$ is low norm vector in $\mathbb{F}_q$ representing the input

Verifying Signatures in a SNARK Prover

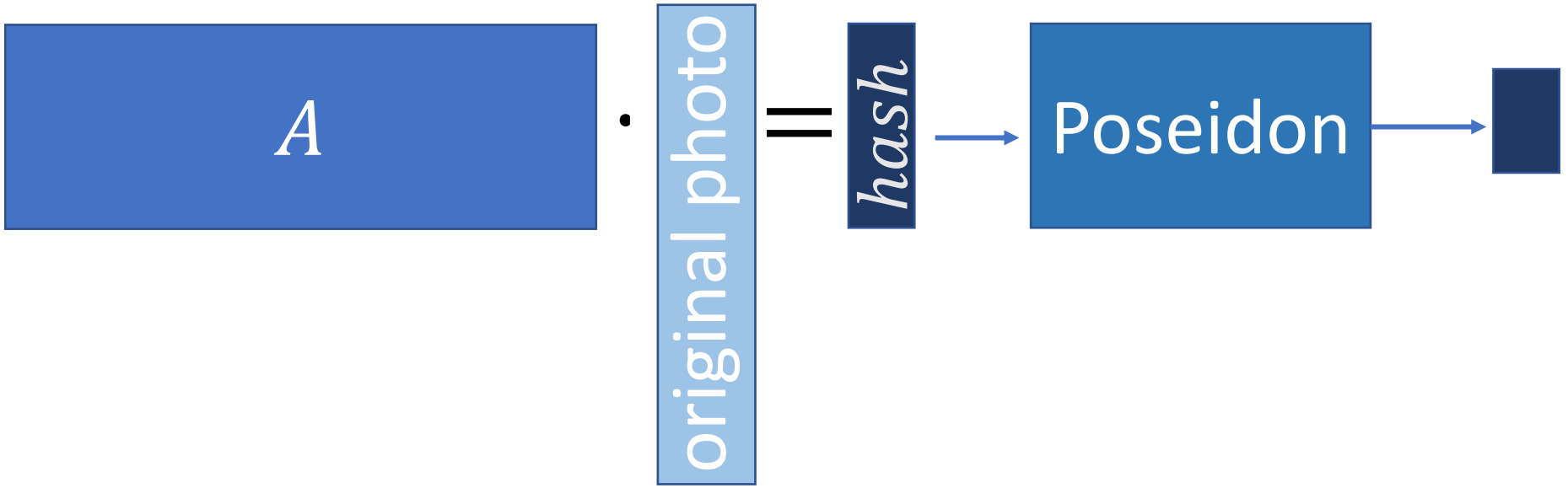
Poseidon hash of lattice hash [GGH'96 , SCMPGLW'08]



- A is random matrix from finite field $\mathbb{F}_q$
- $\vec{x}$ is low norm vector in $\mathbb{F}_q$ representing the input

Verifying Signatures in a SNARK Prover

Poseidon hash of lattice hash [GGH'96 , SCMPGLW'08]



- Collision-resistant assuming SIS → prover must prove original photo representation is low norm

Verifying Signatures in a SNARK Prover

To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \ldots, R-1\}^n$:

***[BCGJM'18, GWC'19, GW'20]

Verifying Signatures in a SNARK Prover

To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \ldots, R-1\}^n$:

$R = 3$

$\vec{x}$ | 2 | 0 | 2 |

***[BCGJM'18, GWC'19, GW'20]

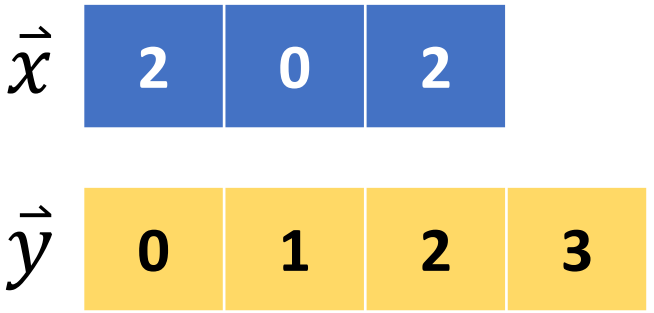
Verifying Signatures in a SNARK Prover

To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \ldots, R-1\}^n$:

- $\vec{y} := [0, 1, \ldots, R-1]$

$R = 3$



***[BCGJM'18, GWC'19, GW'20]

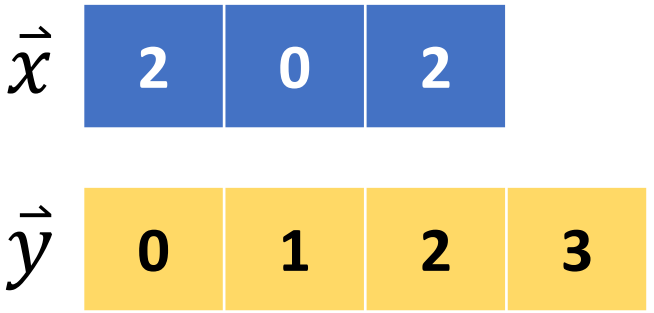
Verifying Signatures in a SNARK Prover

To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \dots, R-1\}^n$:

- $\vec{y} := [0, 1, \dots, R-1]$
- $\vec{z} := sort(\vec{x} \,||\vec{y})$

$R = 3$



***[BCGJM'18, GWC'19, GW'20]

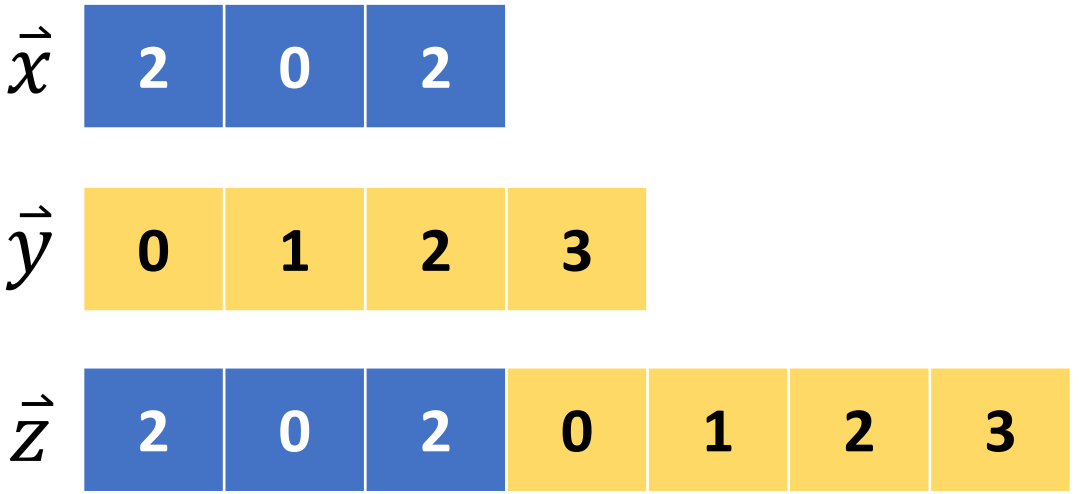
Verifying Signatures in a SNARK Prover

To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \ldots, R-1\}^n$:

- $\vec{y} := [0, 1, \ldots, R-1]$
- $\vec{z} := sort(\vec{x} \,||\vec{y})$

$R = 3$



***[BCGJM'18, GWC'19, GW'20]

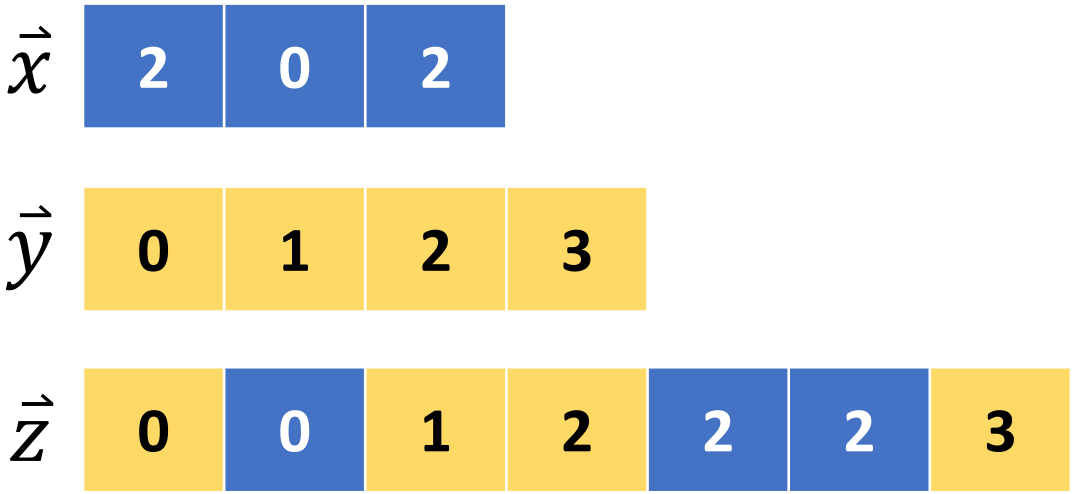
Verifying Signatures in a SNARK Prover

To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \dots, R-1\}^n$:

- $\vec{y} := [0, 1, \dots, R-1]$
- $\vec{z} := sort(\vec{x} \,||\vec{y})$

$R = 3$



***[BCGJM'18, GWC'19, GW'20]

Verifying Signatures in a SNARK Prover

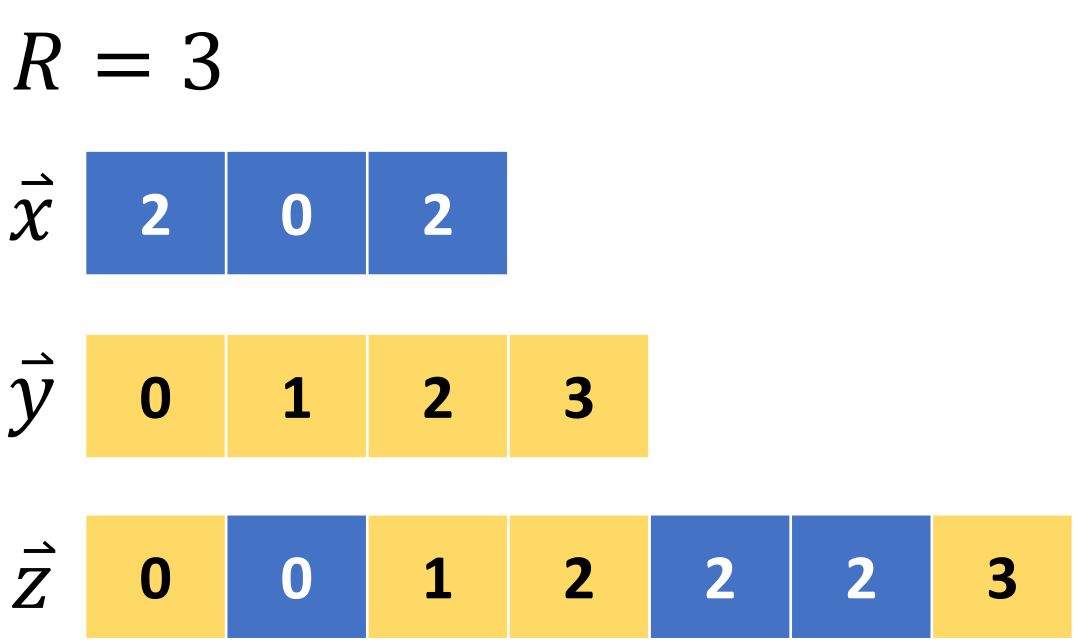
To prove $\vec{x}$ is low norm,

i.e., $\vec{x} \in \{0, 1, \ldots, R-1\}^n$:

- $\vec{y} := [0, 1, \ldots, R-1]$
- $\vec{z} := sort(\vec{x} \,||\vec{y})$

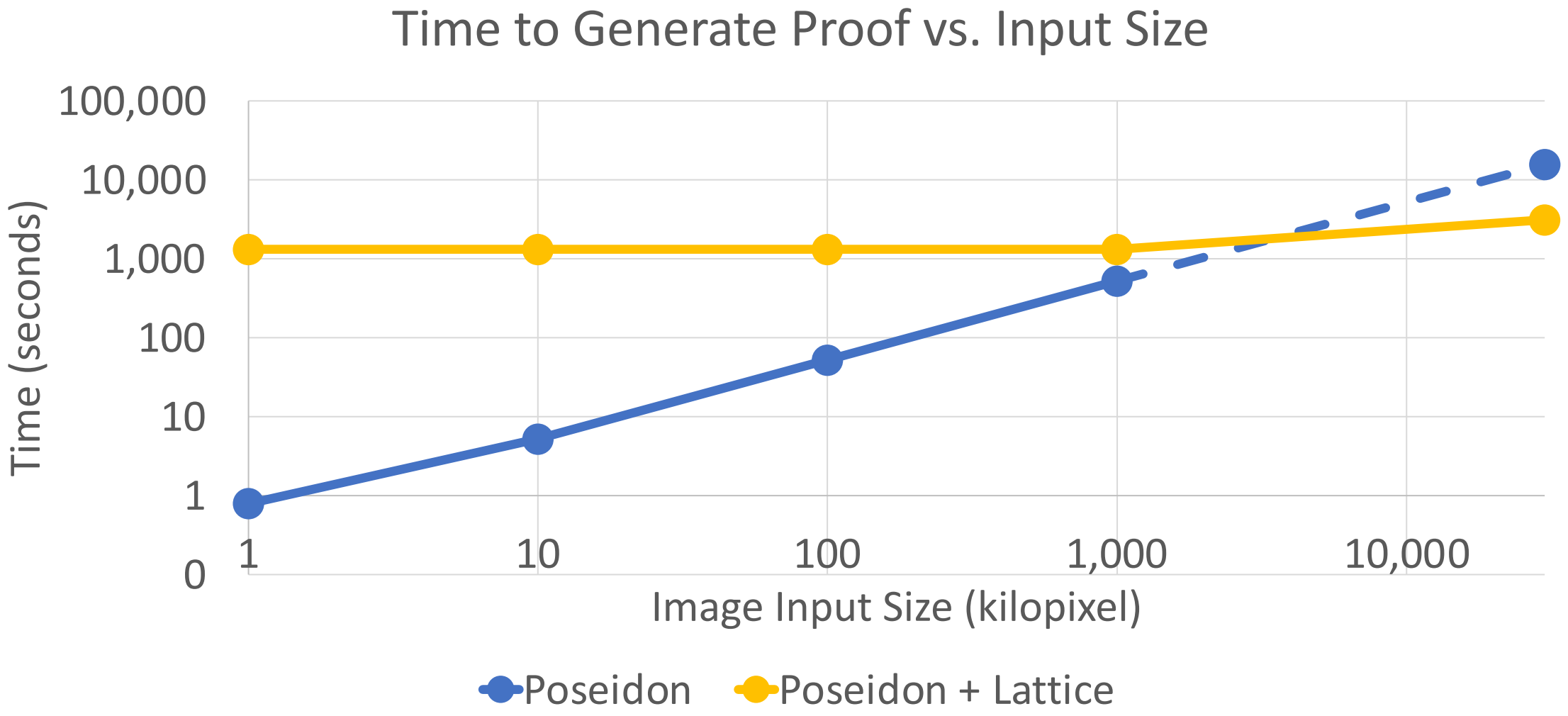
Must show:

- $\vec{z}$ is permutation of $\vec{x}$ and $\vec{y}$
- $\vec{z}[i+1] - \vec{z}[i] \in \{0, 1\}$

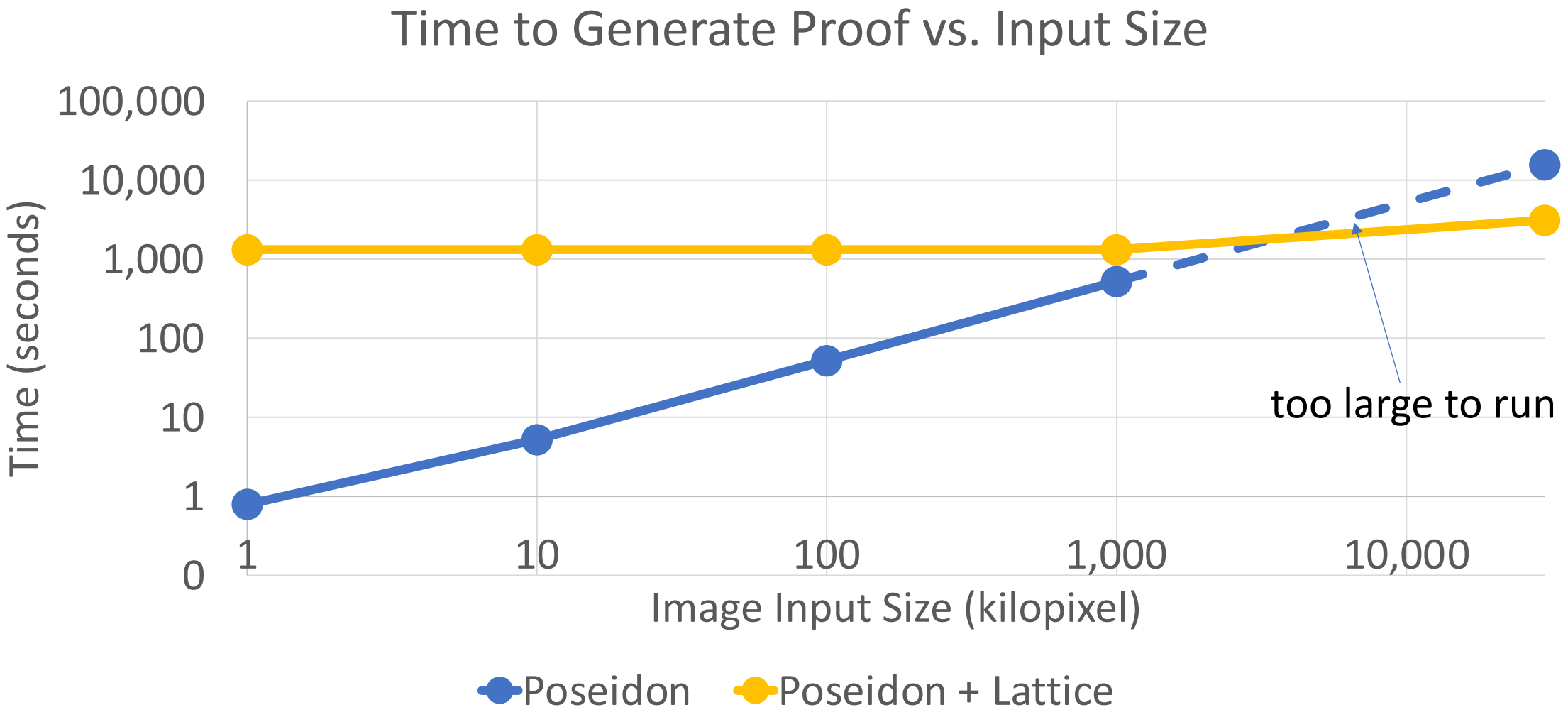


***[BCGJM'18, GWC'19, GW'20]

Performance Results for Proving Signatures



Performance Results for Proving Signatures



Conclusions

- Proof systems have greatly improved due to their need in blockchains $\Rightarrow$ non-blockchain applications benefit
- Proofs about large images (4000×6000) can be done in reasonable time
- Applicable to C2PA for image authenticity
- Open problem: ZK proofs for videos?