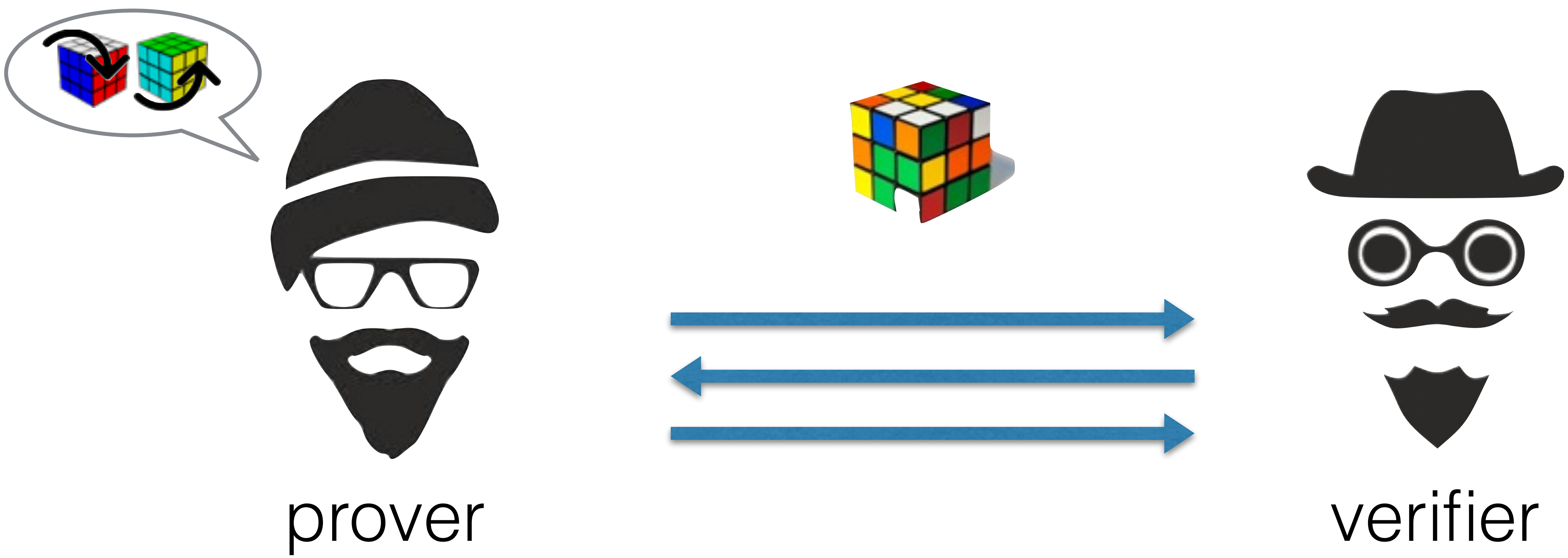
Efficient Designated-Verifier Non-Interactive Zero-Knowledge Proofs of Knowledge

Pyrros Chaidos, *Geoffroy Couteau*

HELLENIC REPUBLIC
National and Kapodistrian University of Athens

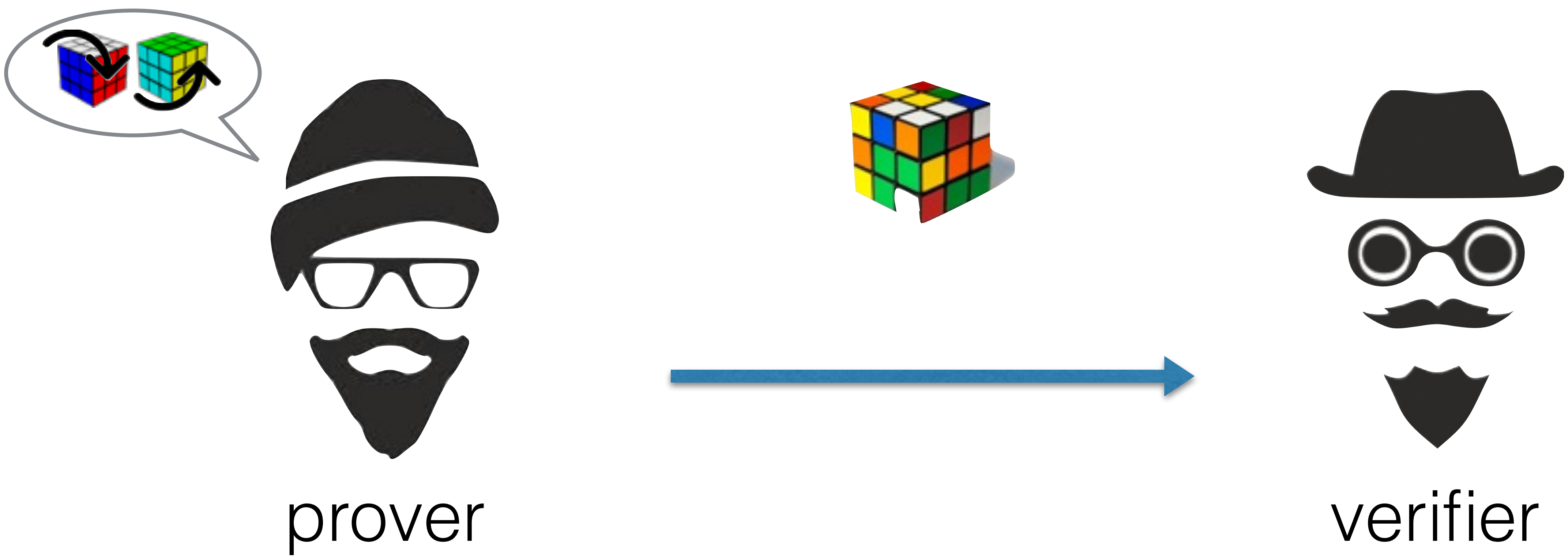
KIT
Karlsruher Institut für Technologie

Zero-Knowledge Proof



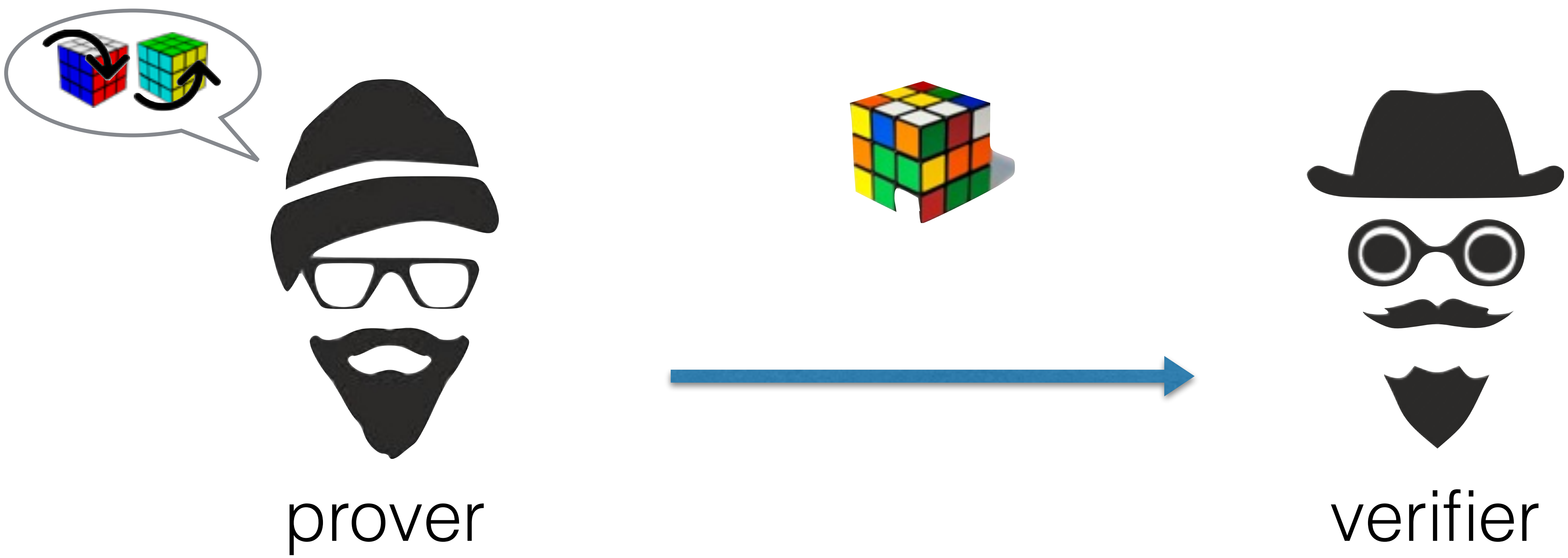
- Complete: if P knows a solution, V accepts
- Sound: if there is no solution, P cannot convince V
- Zero-Knowledge: V does not learn the solution

Non-Interactive Zero-Knowledge Proof



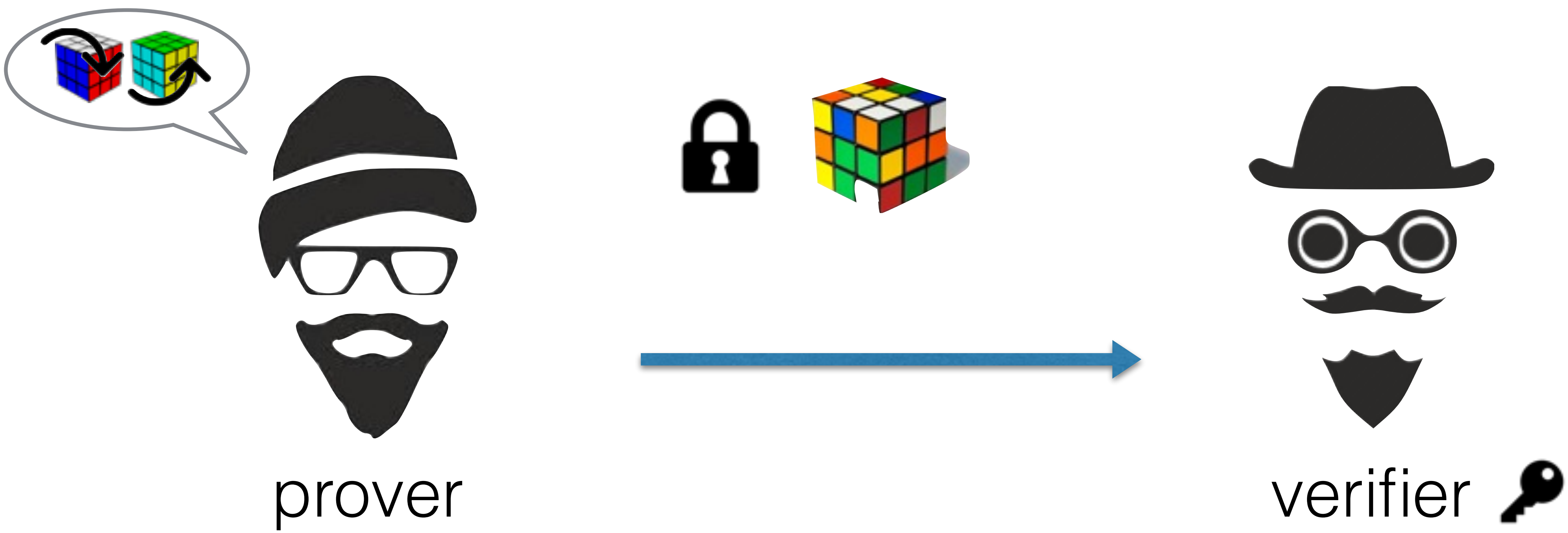
- Complete: if P knows a solution, V accepts
- Sound: if there is no solution, P cannot convince V
- Zero-Knowledge: V does not learn the solution

Non-Interactive Zero-Knowledge Proof of Knowledge



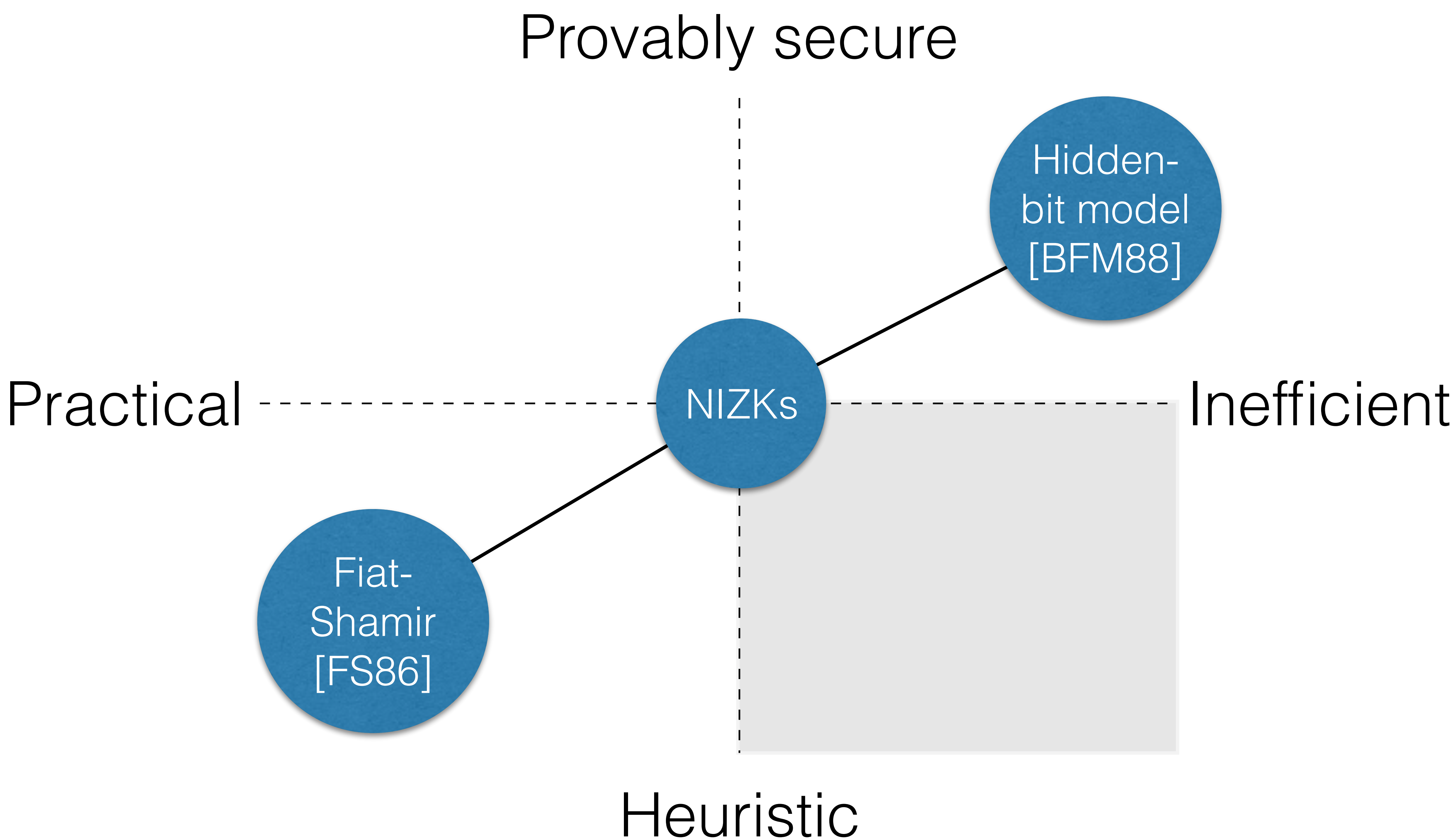
- Complete: if P knows a solution, V accepts
- Sound: P must know a solution to convince V
- Zero-Knowledge: V does not learn the solution

Designated-Verifier NIZKPoK

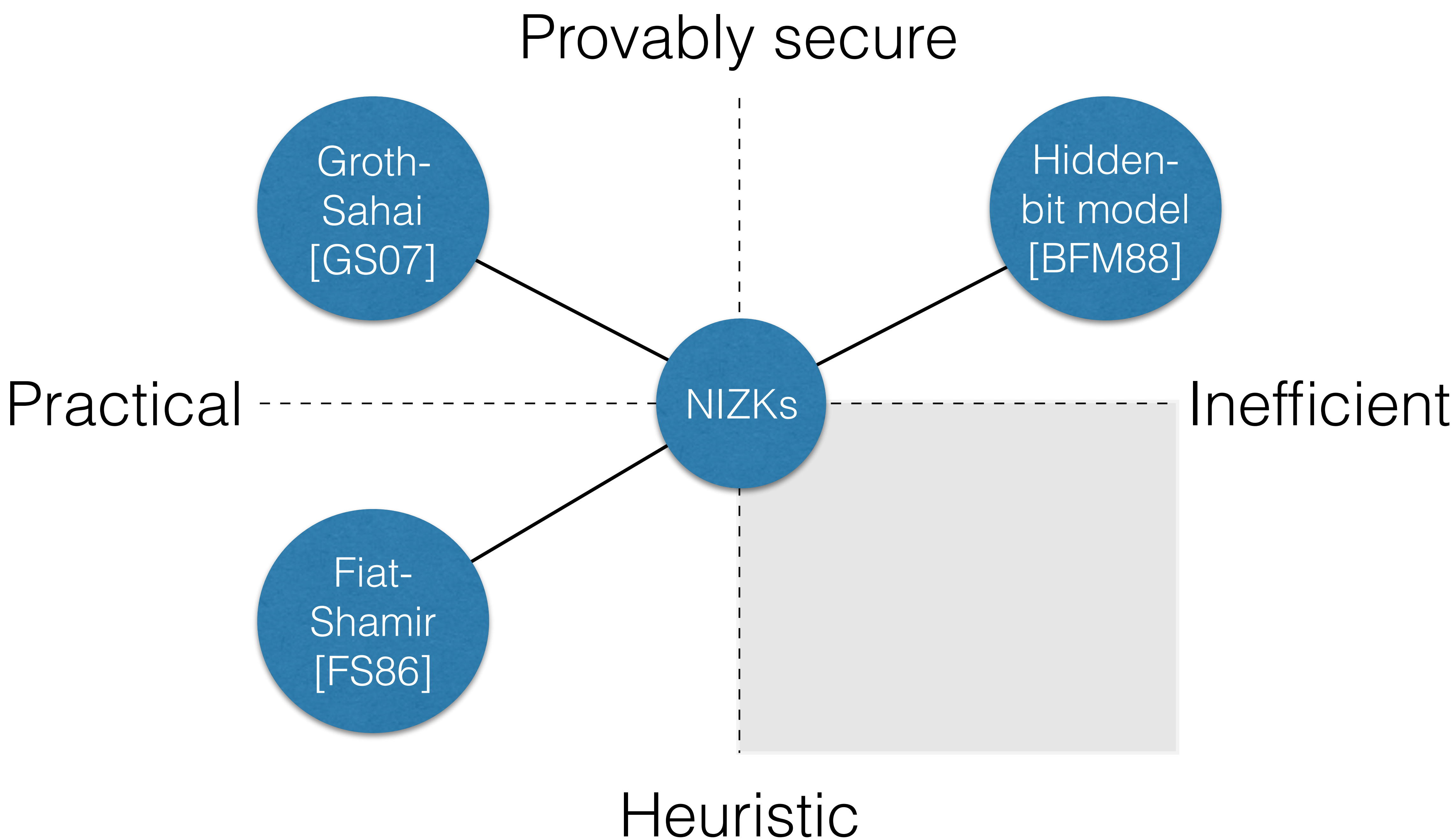


- Complete: if P knows a solution, V accepts
- Sound: P must know a solution to convince V
- Zero-Knowledge: V does not learn the solution

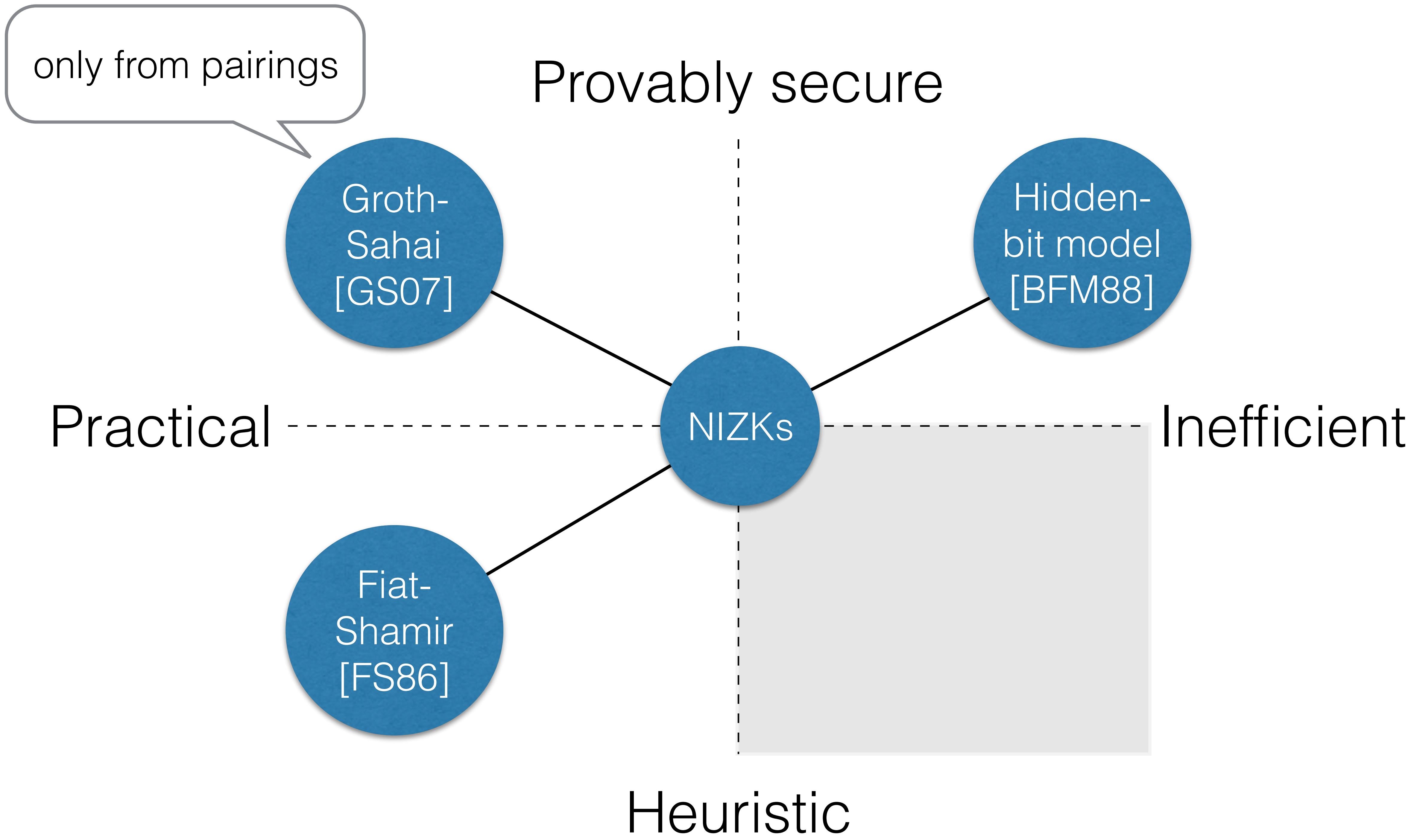
The NIZK Landscape



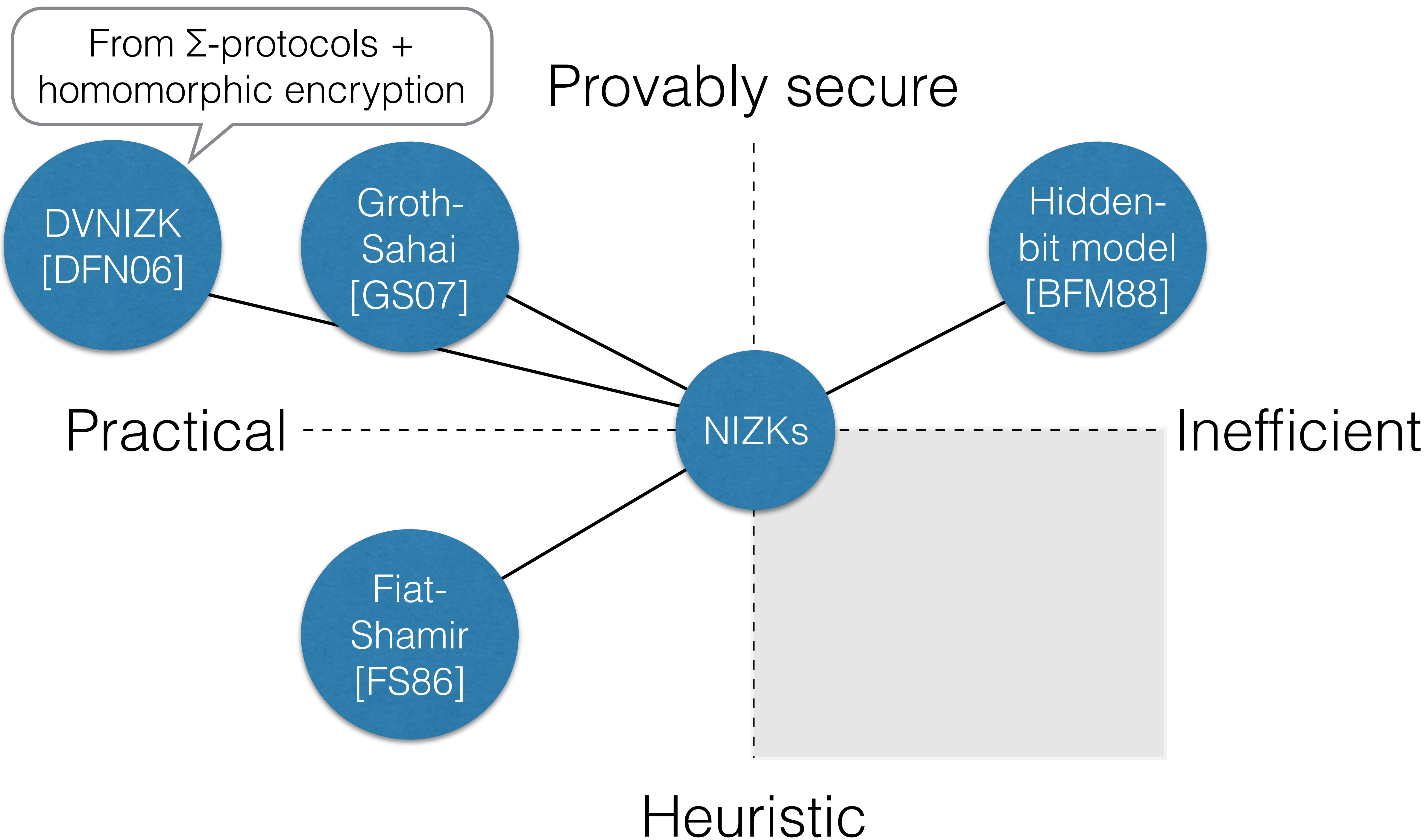
The NIZK Landscape



The NIZK Landscape

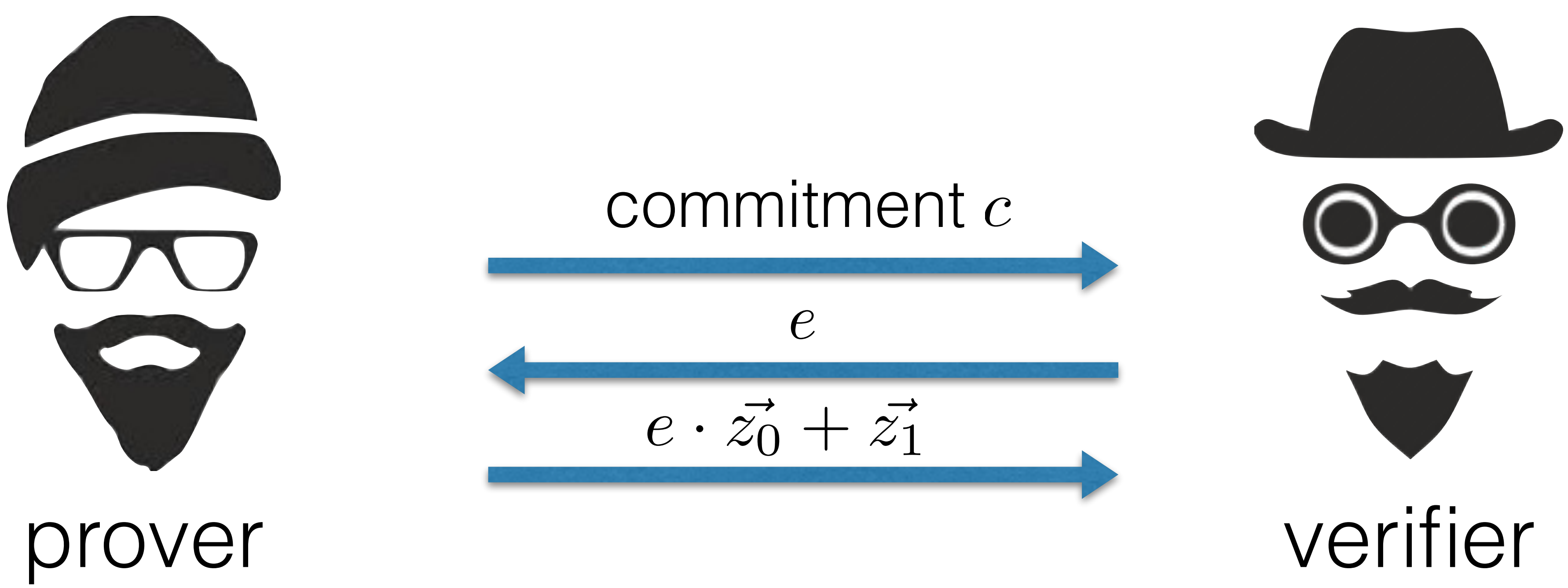


The NIZK Landscape



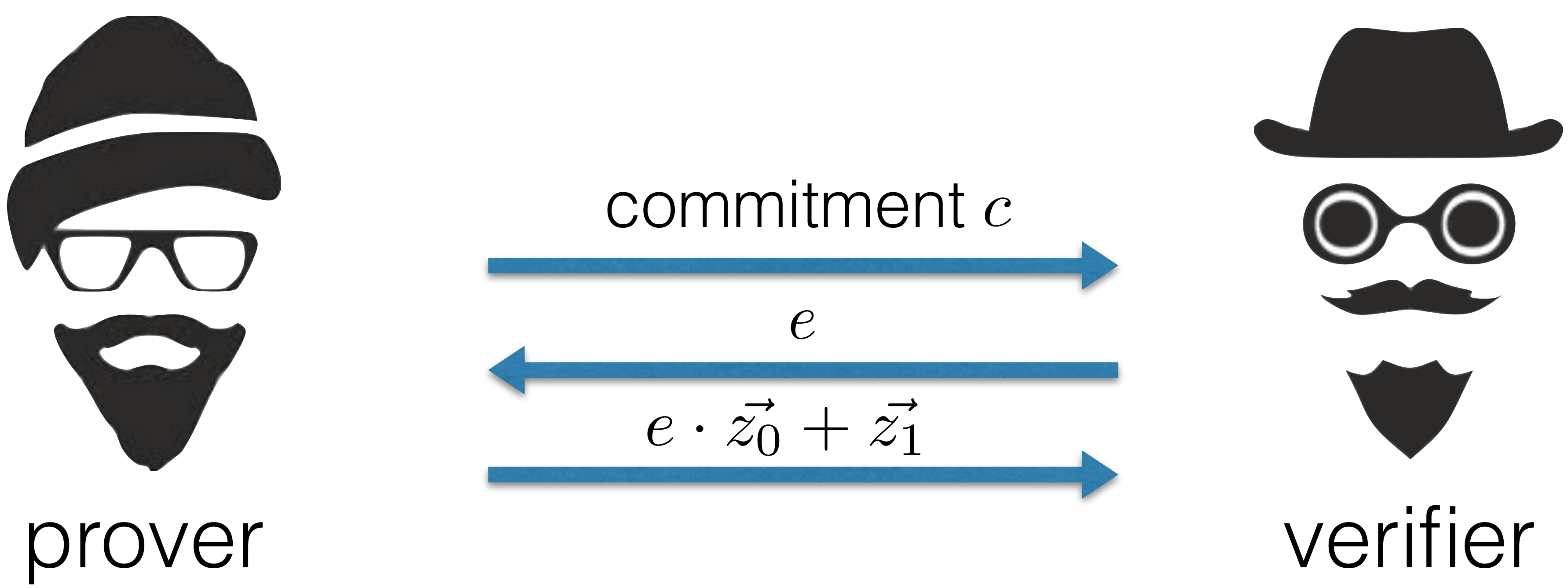
The DFN Compiler

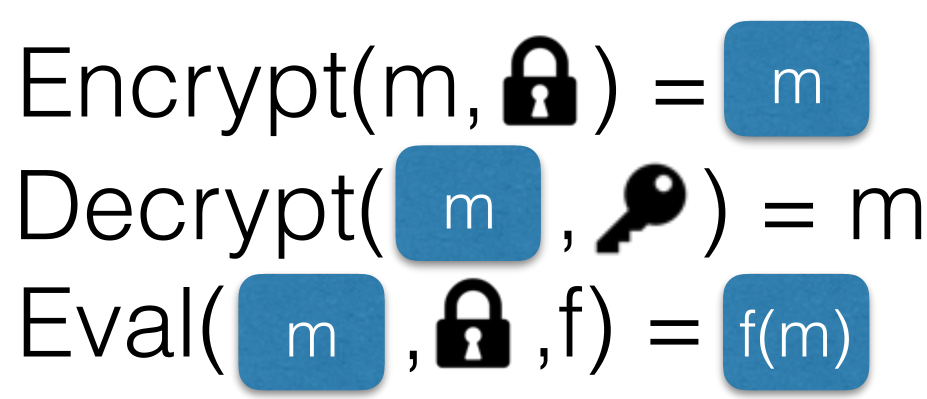
Σ-protocol



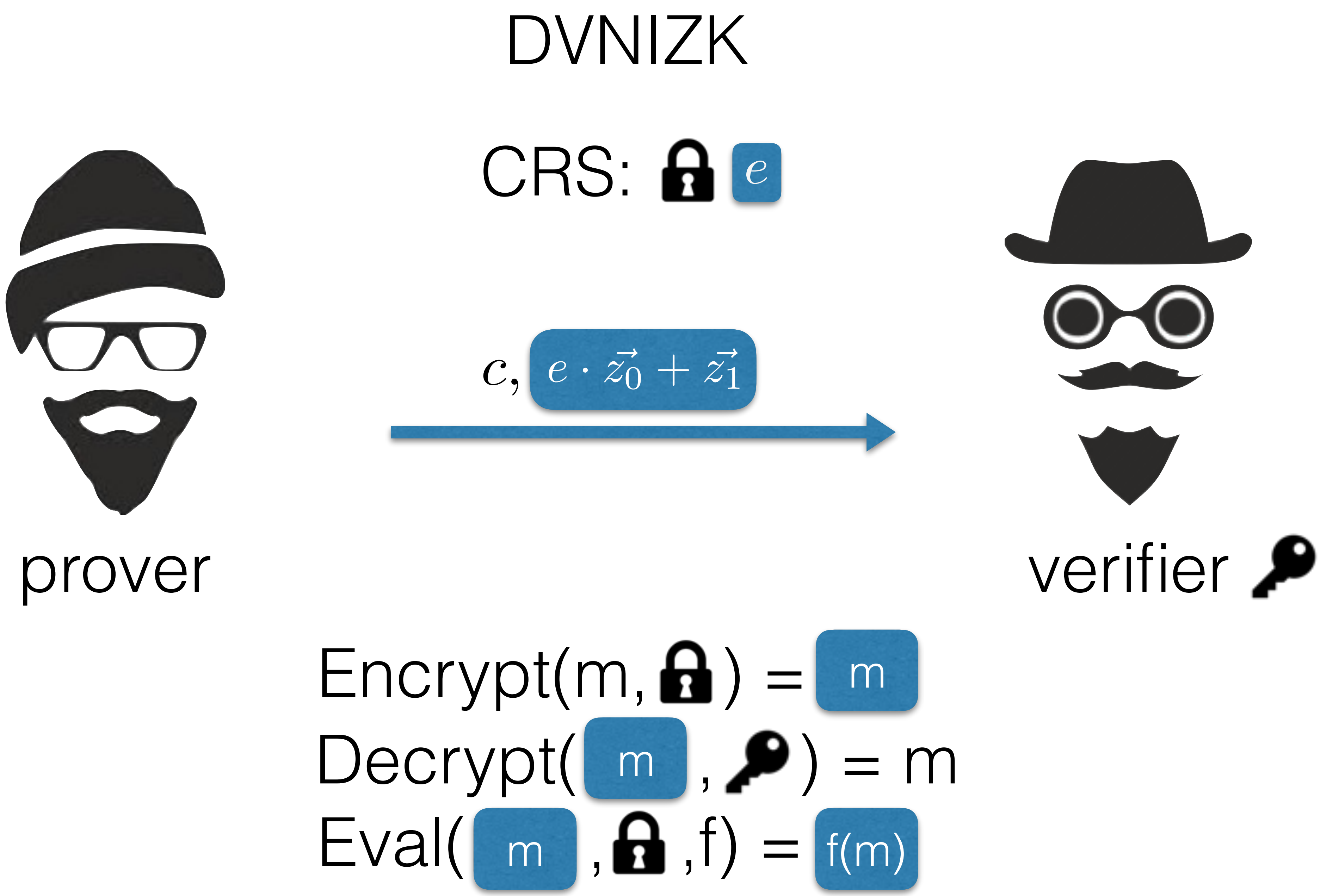
The DFN Compiler

Σ-protocol





The DFN Compiler



The DFN Compiler

- ✘ Non-standard assumption
- ✘ Not a proof of knowledge
- ✘ Soundness is bounded

The DFN Compiler

- ✗ Non-standard assumption
  ✗ Not a proof of knowledge
  ✗ Soundness is bounded

The DFN Compiler

- ✗ Non-standard assumption
- ▶ ✗ Not a proof of knowledge
- ✗ Soundness is bounded

crucial in anonymous authentication

The DFN Compiler

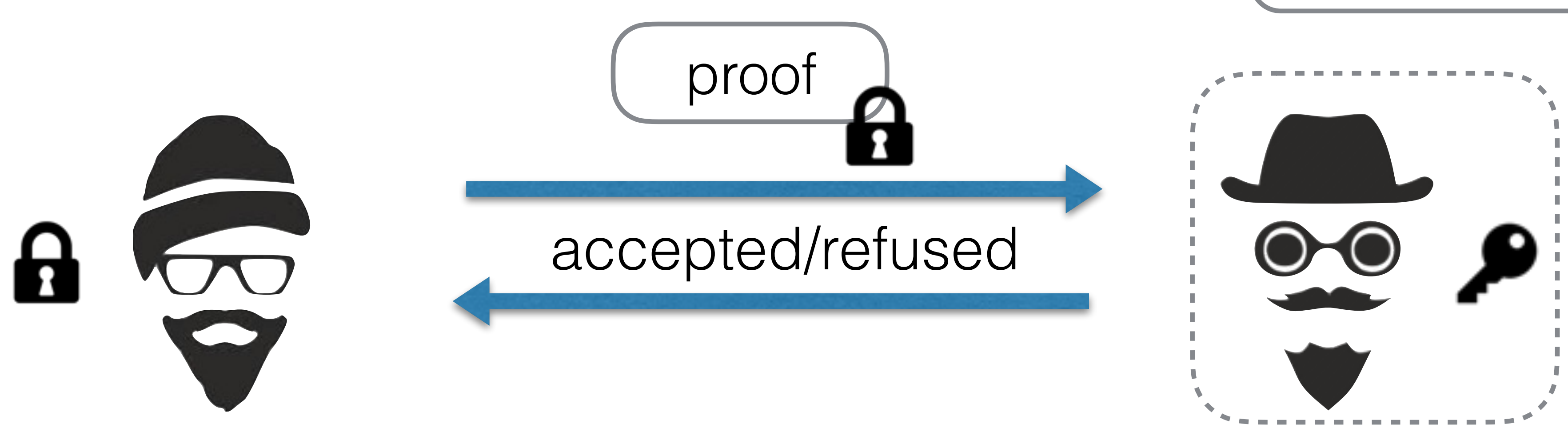
- ✗ Non-standard assumption
- ✗ Not a proof of knowledge
- ▶ ✗ Soundness is bounded

crucial in anonymous authentication

The DFN Compiler

- ✗ Non-standard assumption
- ✗ Not a proof of knowledge
- ▶ ✗ Soundness is bounded

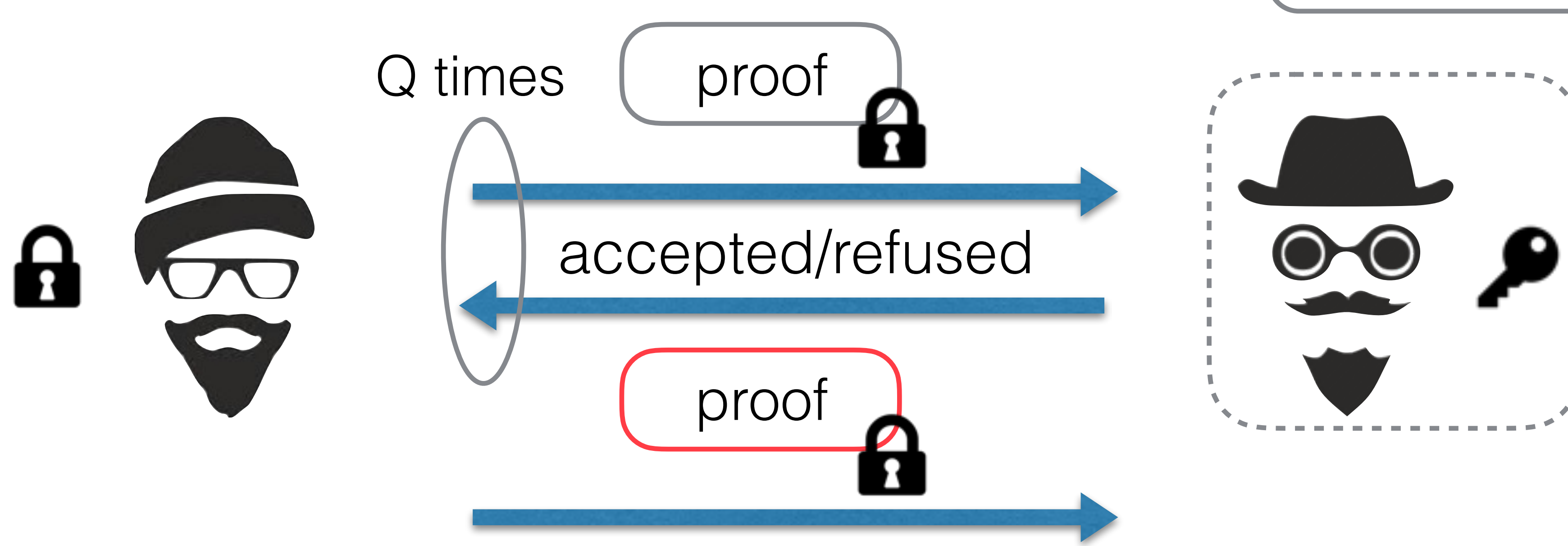
crucial in anonymous authentication



The DFN Compiler

- ✗ Non-standard assumption
- ✗ Not a proof of knowledge
- ▶ ✗ Soundness is bounded

crucial in anonymous authentication

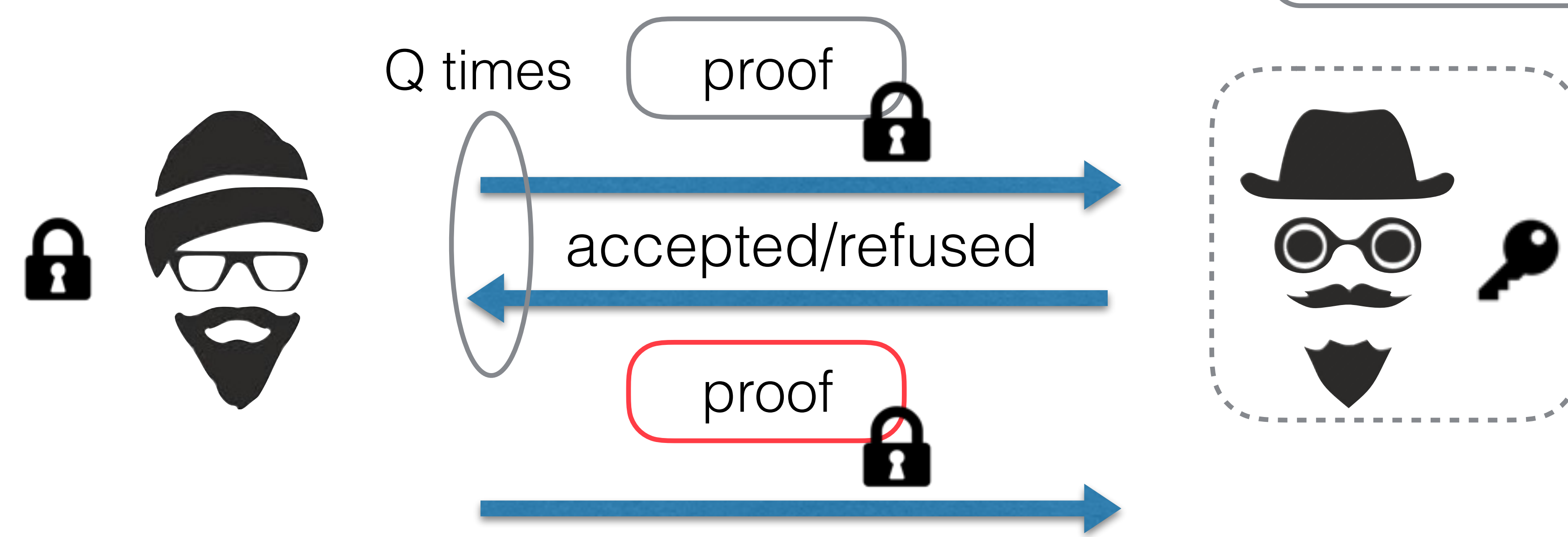


The DFN Compiler

- ✗ Non-standard assumption
- ✗ Not a proof of knowledge
- ✗ Soundness is bounded

there is an attack

crucial in anonymous authentication



The [CG15] Compiler

- ✓ Standard assumption
- ✗ Not a proof of knowledge
- ✗ Soundness is bounded

there is an attack

crucial in anonymous authentication

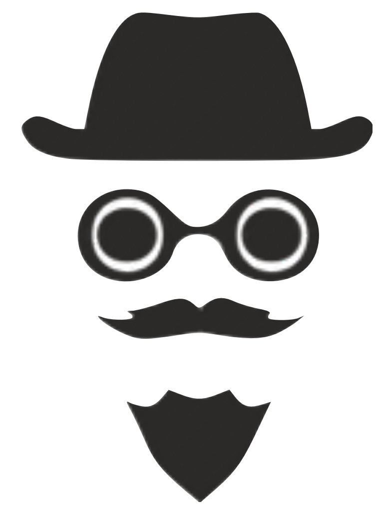
This work

- ✓ Statistical soundness
- ✓ Proof of knowledge
- ✓ Unbounded soundness

Our Technique

CRS: $e; r_e$

c, $e \cdot \vec{z_0} + \vec{z_1}; r$ →



prover

verifier

Eval($e; r_e$, , $x \mapsto x \cdot \vec{z_0} + \vec{z_1}$)

Decrypt($e \cdot \vec{z_0} + \vec{z_1}; r$,)

$= e \cdot \vec{z_0} + \vec{z_1}$

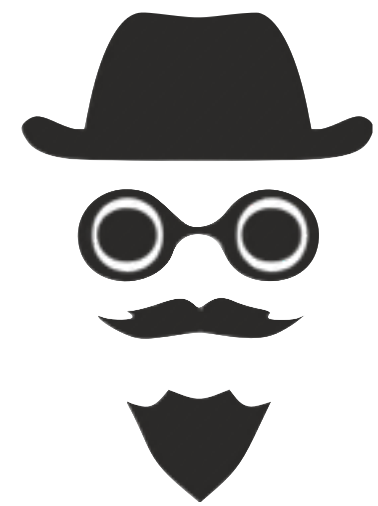
Our Technique

Idea: embed e as a random coin

CRS: $0; e$

c, $\vec{z}_1; -e\vec{r}$ $\vec{z}_0; \vec{r}$

prover



verifier

f: $(x_0, x_1) \mapsto ex_0 + x_1$

Eval($0; 1$, , $x \mapsto \vec{r} + \vec{z}_0$) = $\vec{z}_0; \vec{r}$

Eval($0; e$, , $x \mapsto -x\vec{r} + \vec{z}_1$) = $\vec{z}_1; -e\vec{r}$

Eval($\vec{z}_1; -e\vec{r}$ $\vec{z}_0; \vec{r}$, ,f)

= $e \cdot \vec{z}_0 + \vec{z}_1; 0$

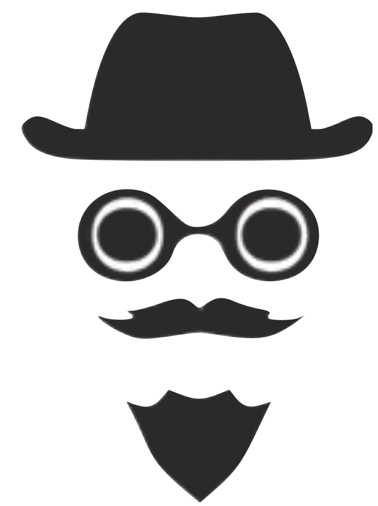
need a *decodable* scheme

Our Technique

use as extraction trapdoor

CRS: $0; e$

$c,$ $\vec{z}_1; -e\vec{r}$ $\vec{z}_0; \vec{r}$



prover

verifier ✗ e

f: $(x_0, x_1) \mapsto ex_0 + x_1$

Eval($0; 1$, , $x \mapsto \vec{r} + \vec{z}_0$) = $\vec{z}_0; \vec{r}$

Eval($0; e$, , $x \mapsto -x\vec{r} + \vec{z}_1$) = $\vec{z}_1; -e\vec{r}$

Eval($\vec{z}_1; -e\vec{r}$ $\vec{z}_0; \vec{r}$, ,f)

= $e \cdot \vec{z}_0 + \vec{z}_1; 0$

secret key not needed anymore!

Why Extraction Works?

- Ext decrypts $\vec{z_0}$ and $\vec{z_1}$
- Knowing 🔑 does not allow P to cheat:
  - Let (M,R) be the plaintext/random coin orders
  - Pick a scheme with gcd(M,R) = 1
  - $0; e$ leaks only e mod R (statistically)
  - Ext can extract e mod M from an incorrect proof

What's in the Paper?

- We build a general, Groth-Sahai-like framework of unbounded DVNIZK proofs of knowledge, for statements over an abelian group.
- A dual variant of our framework handles statements over additively homomorphic cryptosystems.
- A third variant gives up on properties (unbounded, PoK) for efficiency, beating even Fiat-Shamir.

Thank you for your attention

Questions?