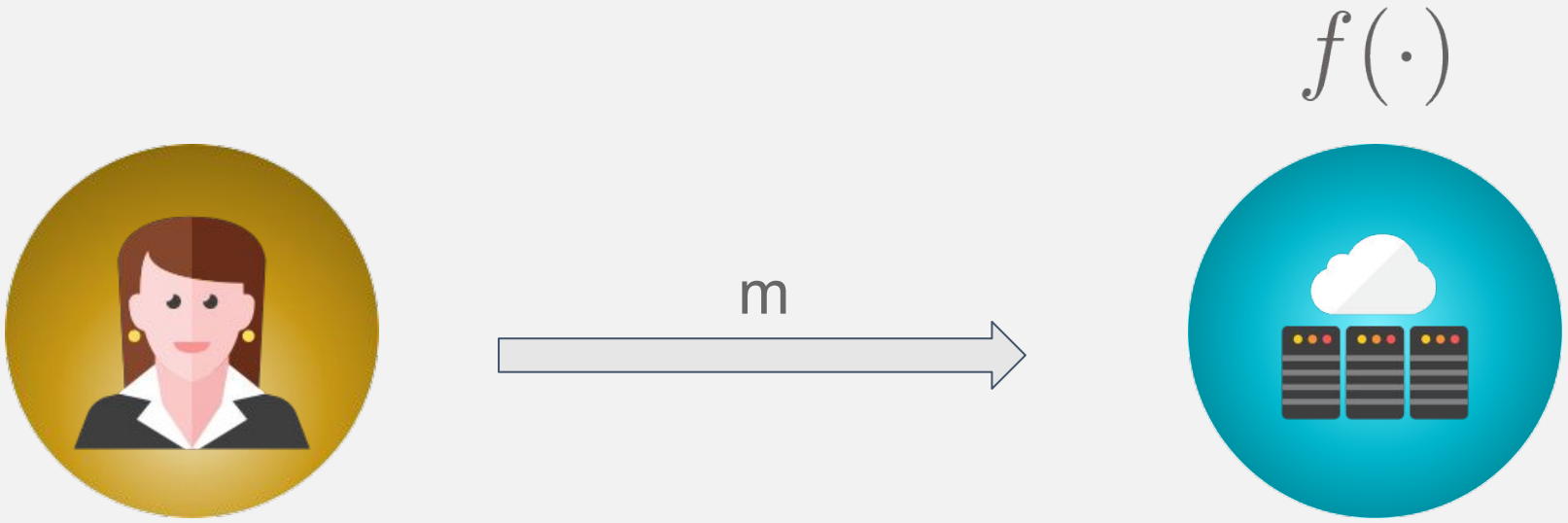
On the (In)security of SNARKs in the Presence of Oracles

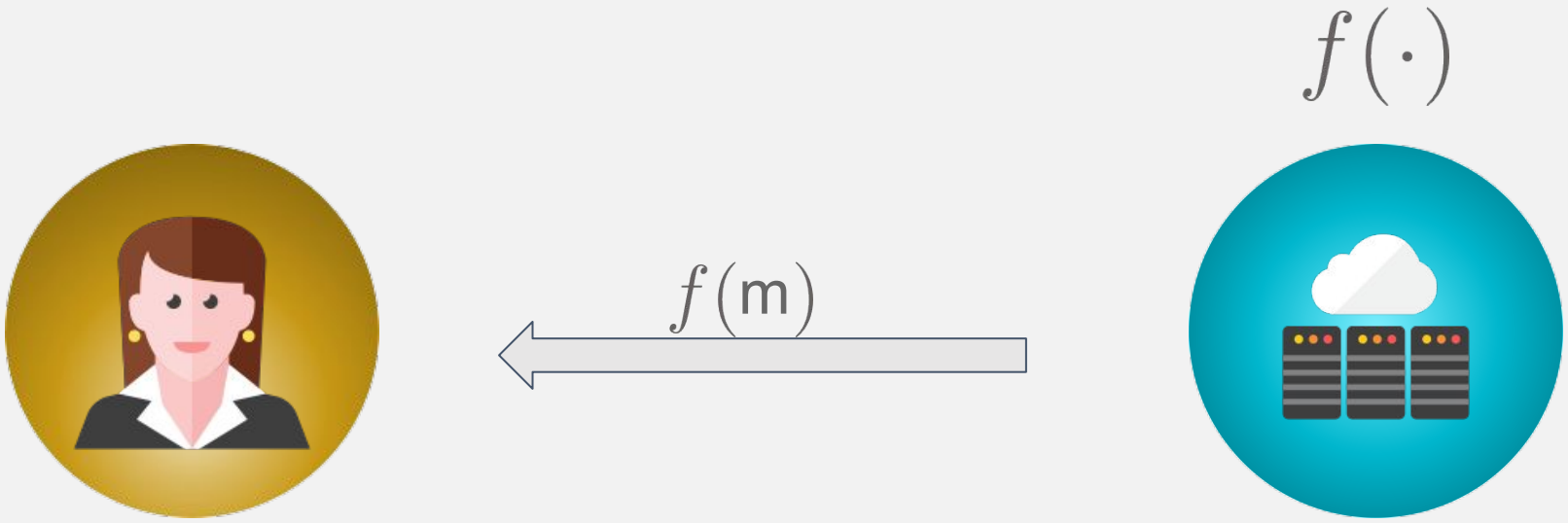
Anca Nitulescu, Dario Fiore

École Normale Supérieure, CNRS and INRIA, Paris, France

Delegate NP computation



Delegate NP computation



Proof ?

Arguments of knowledge

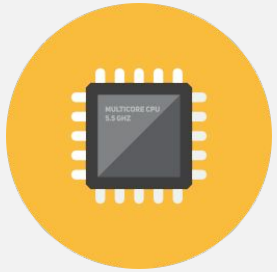
$\mathsf{Gen}(1^\lambda, T) \to \mathsf{crs}$

$\mathsf{Prove}(\mathsf{crs}, y, w) \to \pi : (y, w) \in R$

$\mathsf{Ver}(\mathsf{crs}, y, \pi) \to 0/1$

Succinct Non-Interactive Arguments of Knowledge

Efficiency



Succinctness

Non - Interactivity

SECURITY: PROOF OF KNOWLEDGE

$\forall \mathcal{A}$

crs, aux

$\exists\ \varepsilon_{\mathcal{A}}$

crs, aux

$$\Pr\left[\begin{array}{c}\mathsf{Ver}(\mathsf{crs}, y, \pi) = 1 \\ \wedge \\ (y, w) \notin R\end{array}\;\middle|\;\begin{array}{c}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda) \\ (y, \pi) \leftarrow \mathcal{A}(\mathsf{crs}, \mathsf{aux}) \\ w \leftarrow \varepsilon_{\mathcal{A}}(\mathsf{crs}, \mathsf{aux})\end{array}\right] \approx 0$$

Non-Black-Box Extraction

$\mathcal{A}$

crs, aux

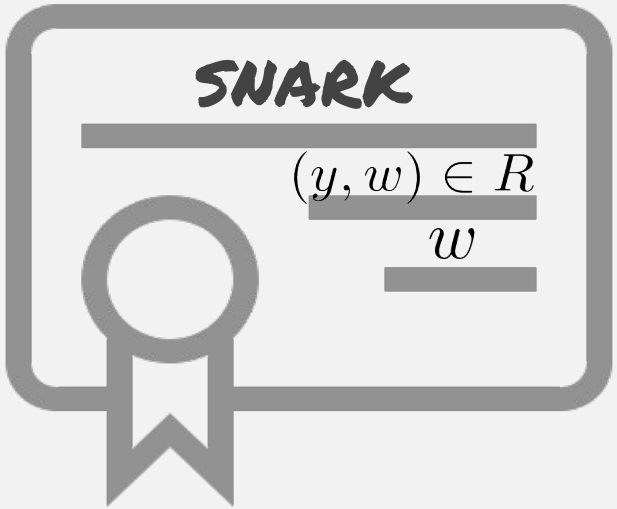
Adversary

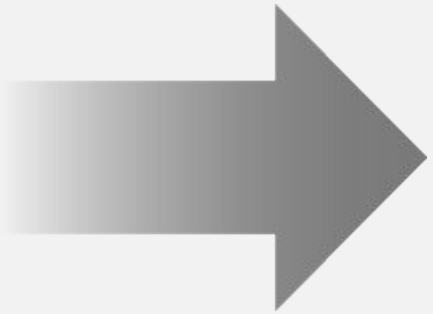
$\mathcal{E}$

crs, aux

Extractor

π





Overview

Starting Point

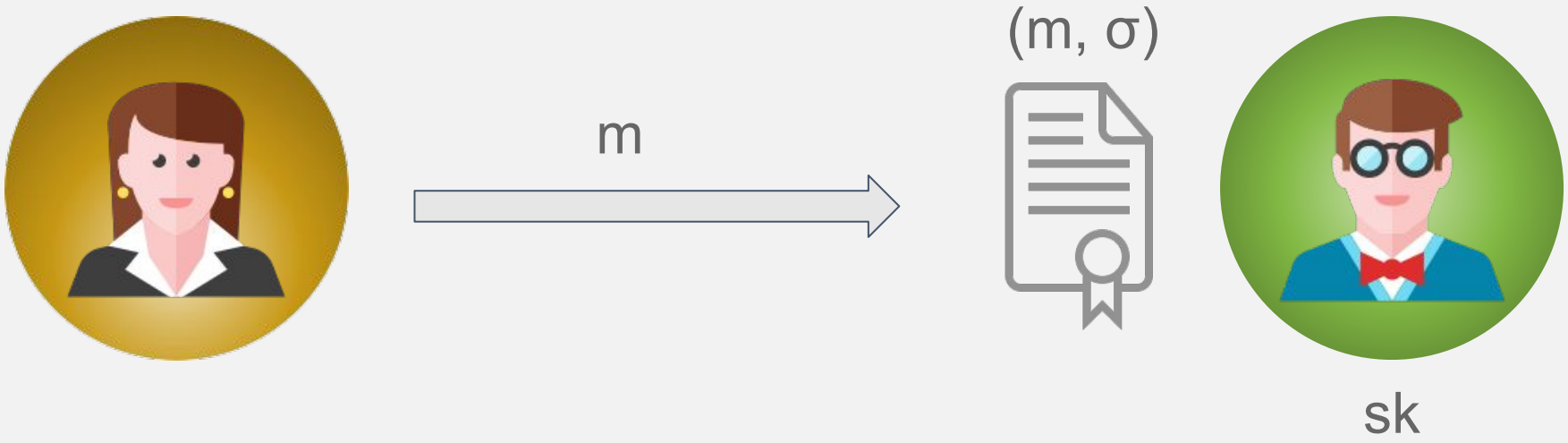
- Protocols with SNARKs
- Security proofs: settings where extraction is problematic
- Need of new security notion: O-SNARK

Study of O-SNARKs

- Impossibility result
- Some “restrictive” instantiations from classical SNARKs
- Applications where O-SNARK is useful

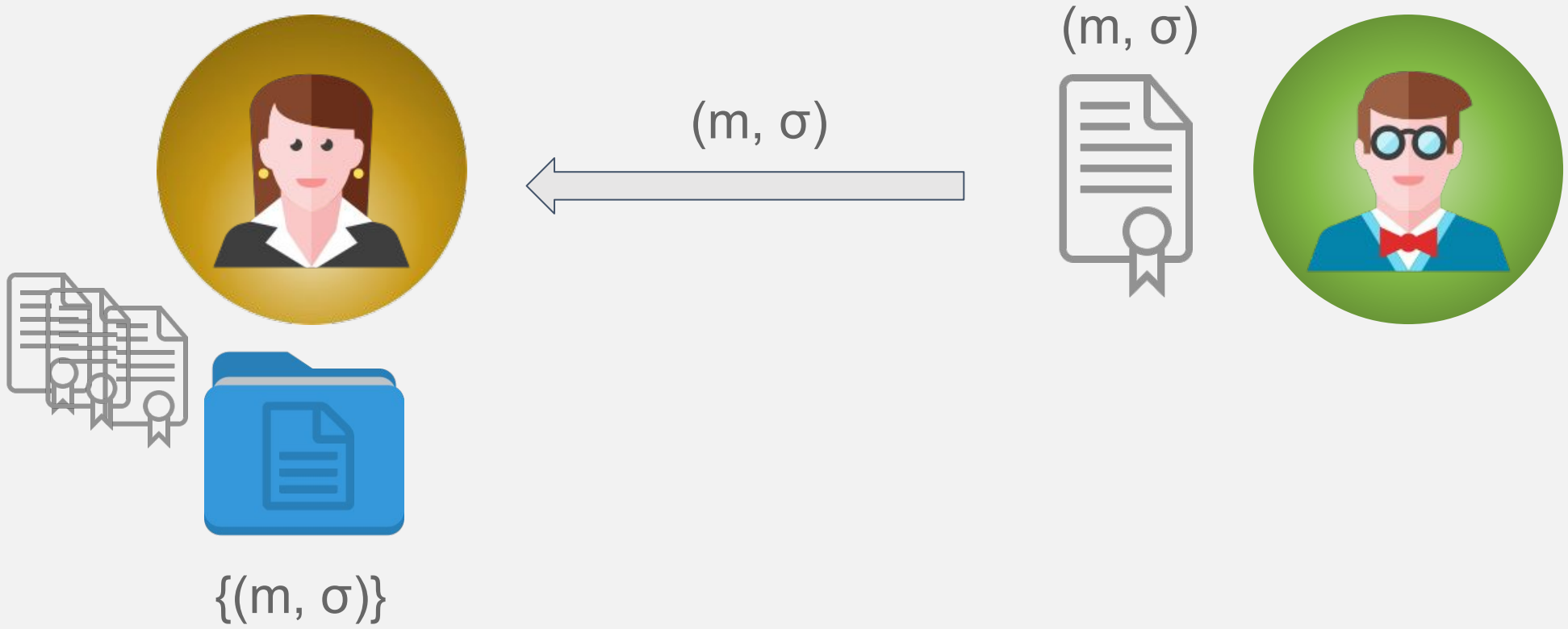
Case Study Application

Proving Knowledge of Signatures



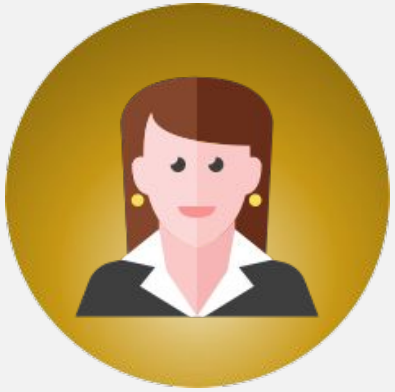
Case Study Application

Proving Knowledge of Signatures



Case Study Application

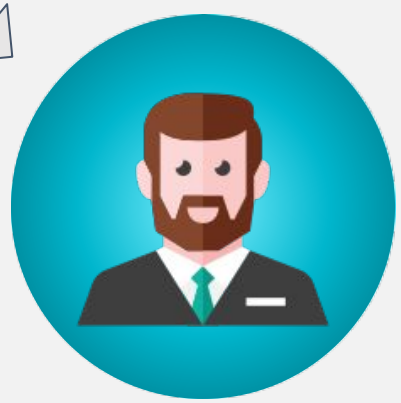
Proving Knowledge of Signatures



SNARK

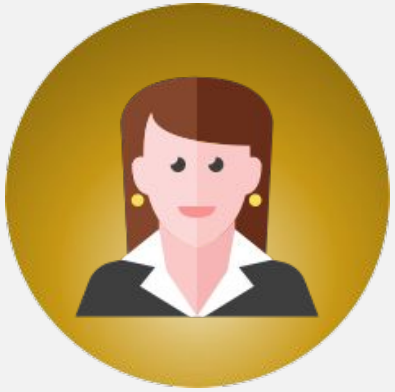
$\exists\ (m, \sigma)$
$Vfy(m,\sigma)=1$
$P(m)=1$

SNARK



Case Study Application

Proving Knowledge of Signatures



SNARK

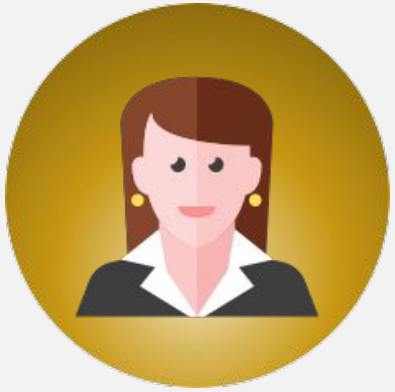
(m, σ)

P(m)=1

Ver(m, σ)=1

Case Study Application

Proving Knowledge of Signatures

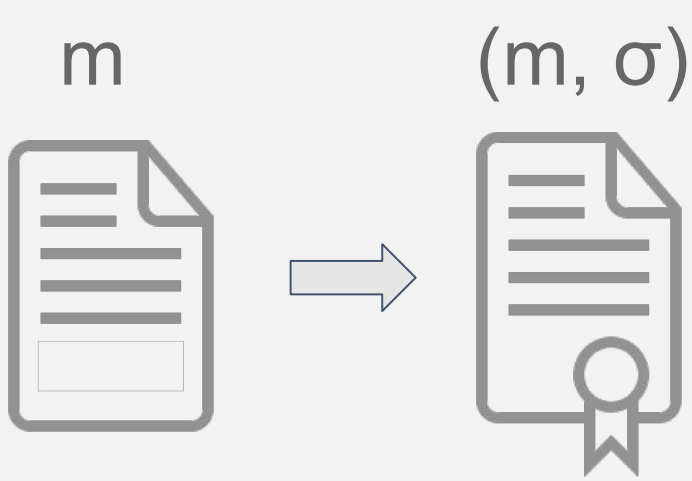


Security Proof ?

SNARK

Security Proof

Unforgeability of Σ

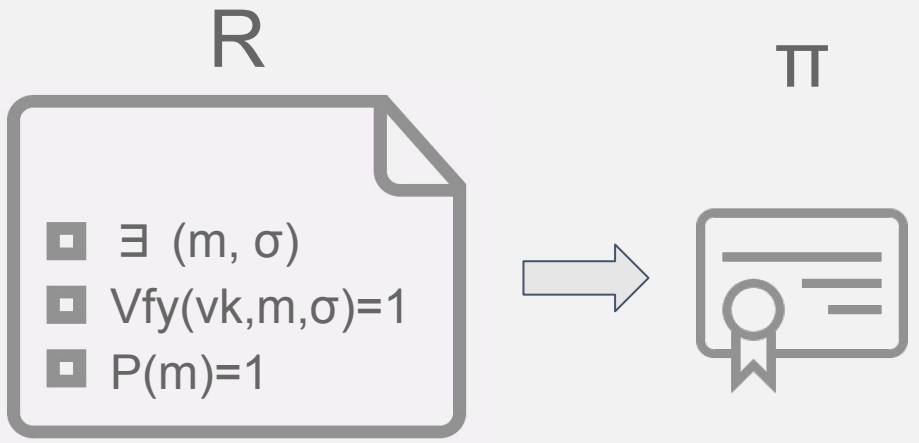


KeyGen(λ): (sk, vk)

Sign(sk, m): σ

Vfy(vk, m, σ): 0 / 1

Proof of knowledge of Π

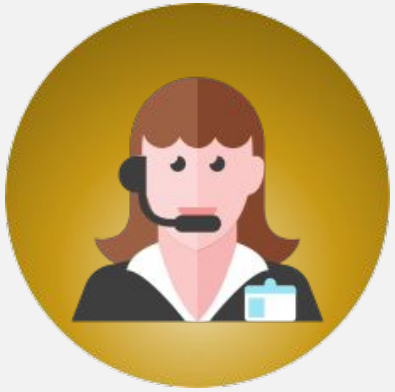


Gen(λ): crs

Prove(crs, y, (m, σ)): π

Ver(vk, y, π): 0 / 1

Security Proof



(m*, σ*)

Breaks proof of knowledge of Π

Type I

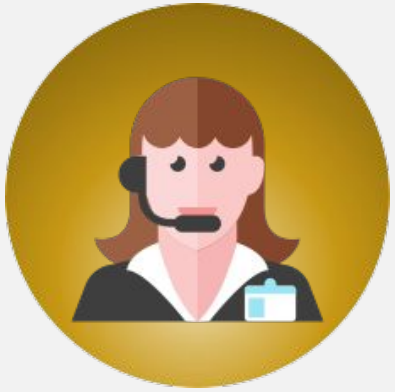
π

SNARK

(m*, σ*)

P(m*)=0 /
Vfy(m*, σ*)=0

Security Proof



(m*, σ*)

Forgery on Σ!

Type II

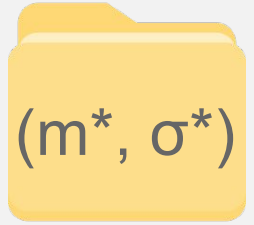
π

SNARK

(m*, σ*)

P(m*)=1

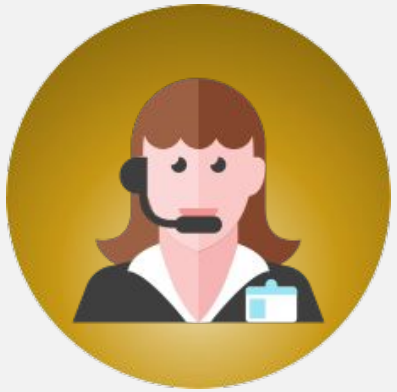
Vfy(m*, σ*)=1



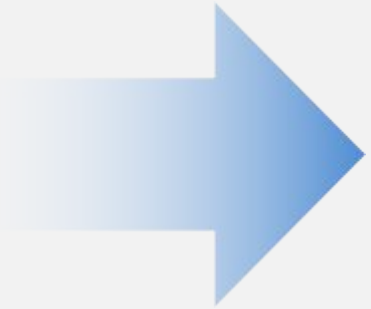
(m*, σ*)

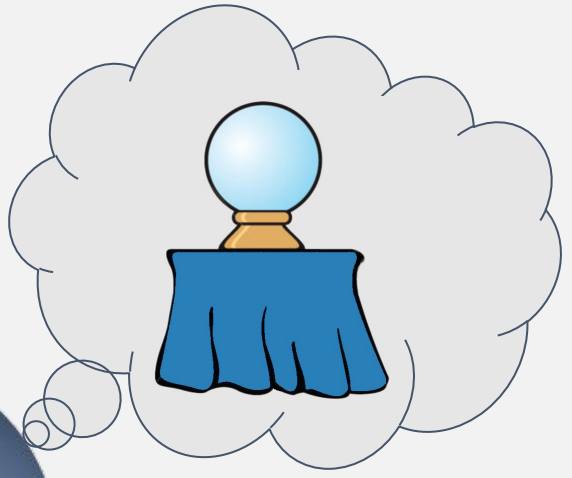
Security Proof

SNARK



CHEATING PROVER



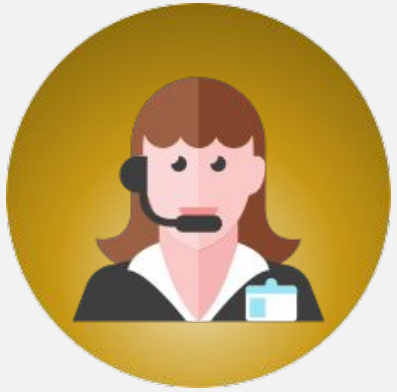


ORACLE

Σ- FORGER

Security Proof

Π



crs, aux

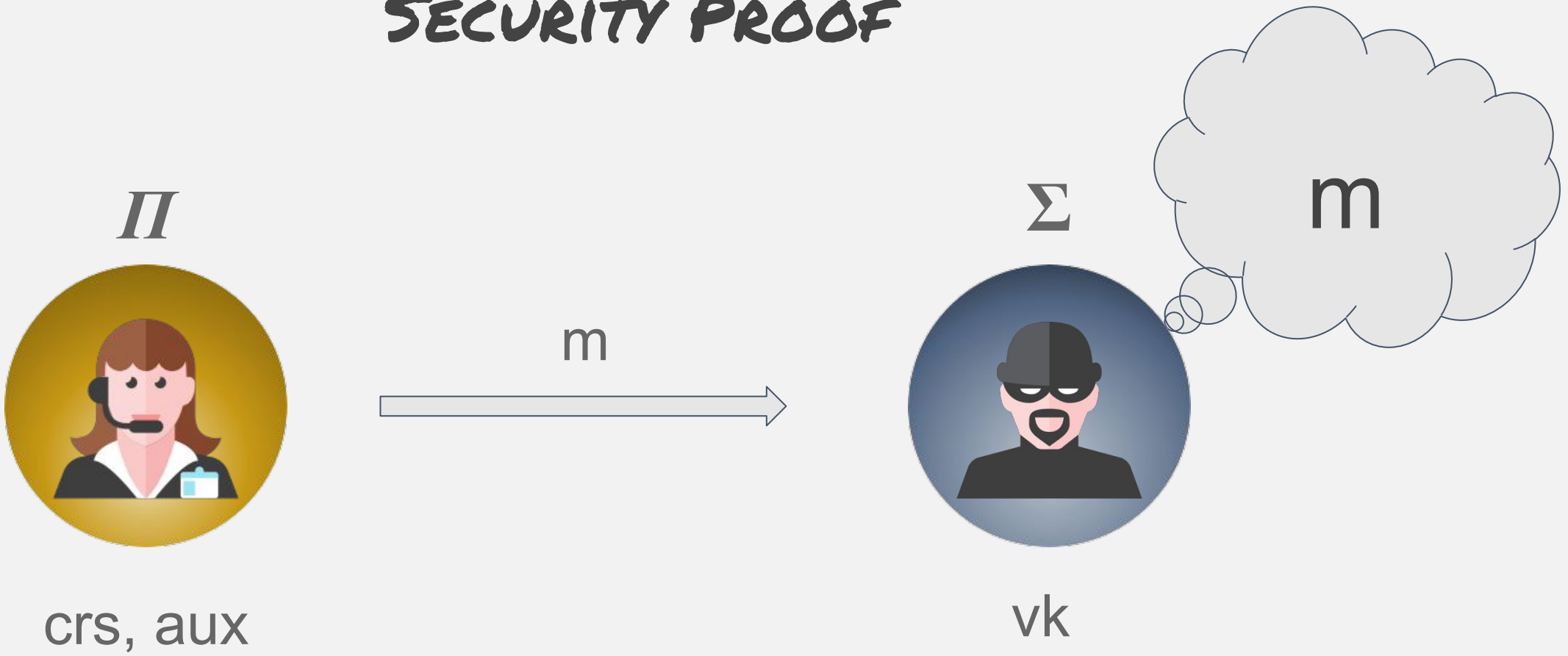
crs, vk

Σ

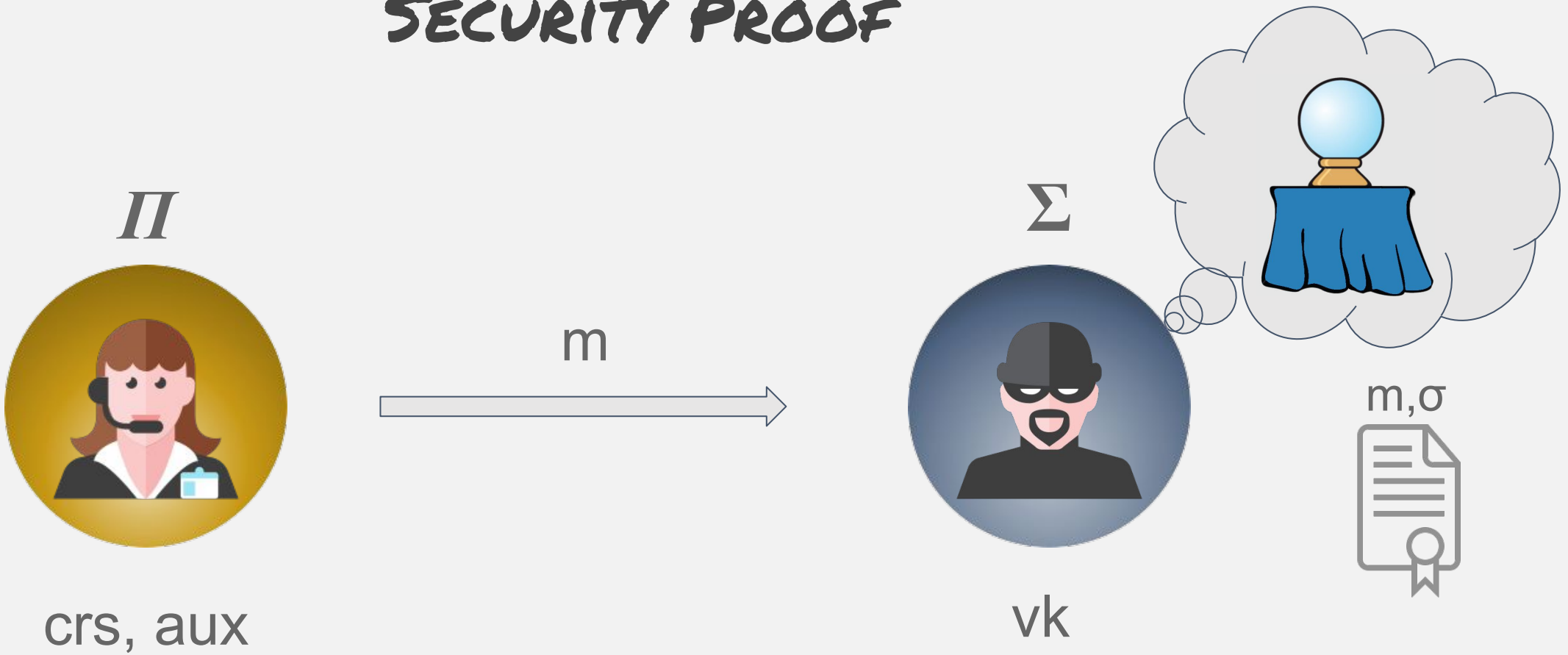
vk

crs←$

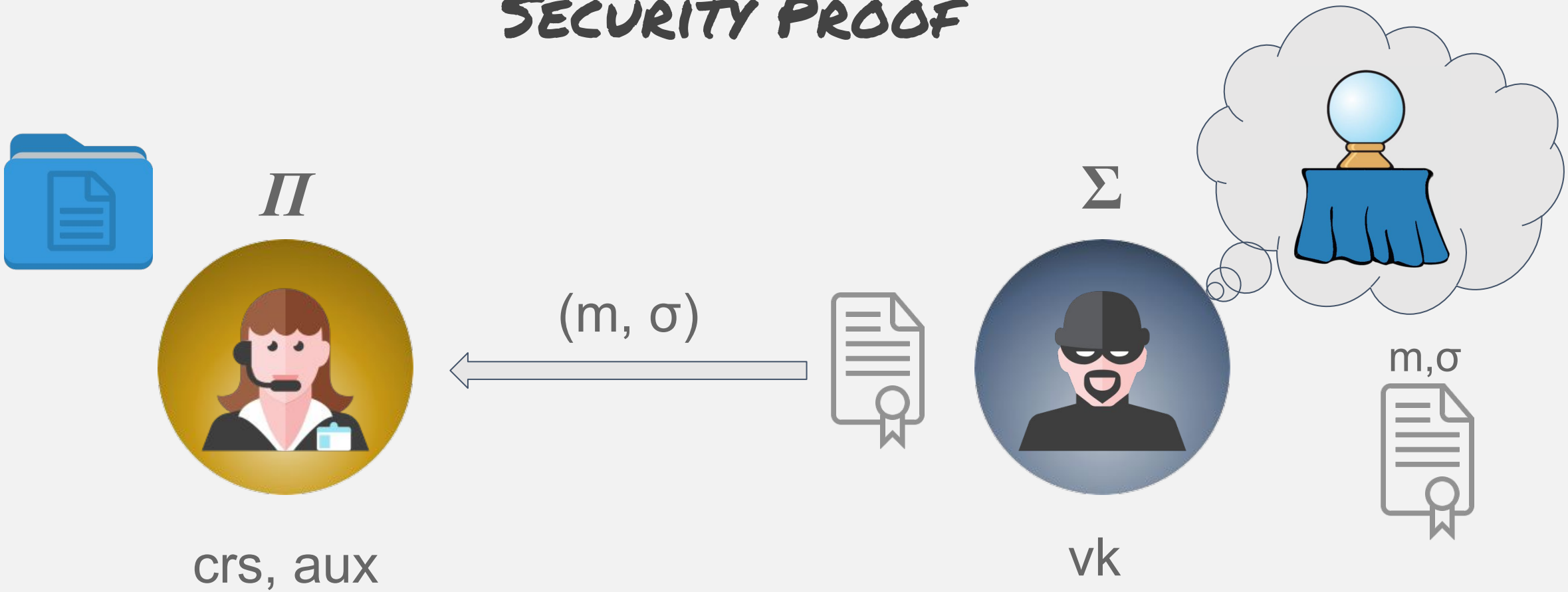
Security Proof



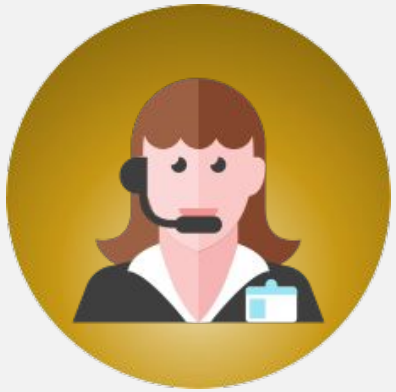
Security Proof

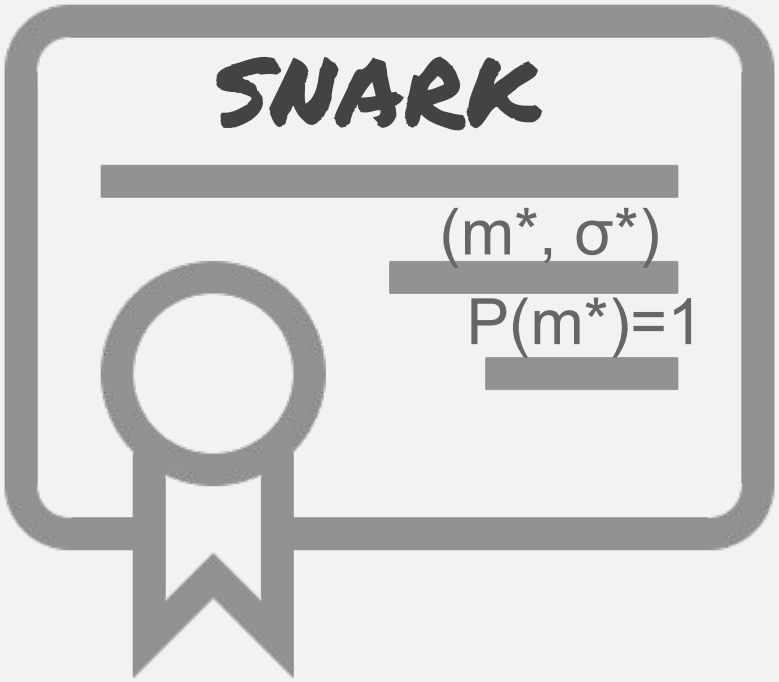


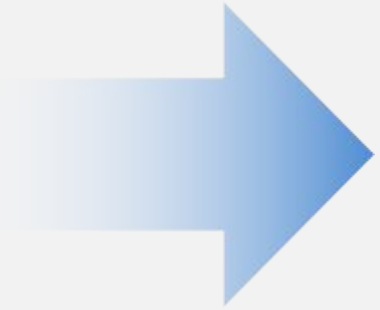
Security Proof



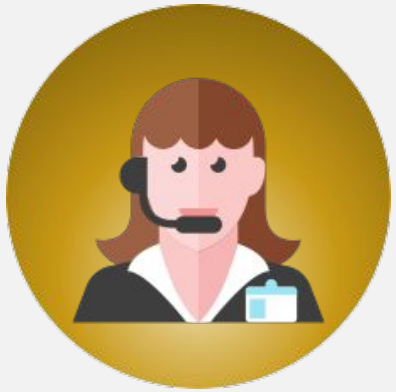
Security Proof



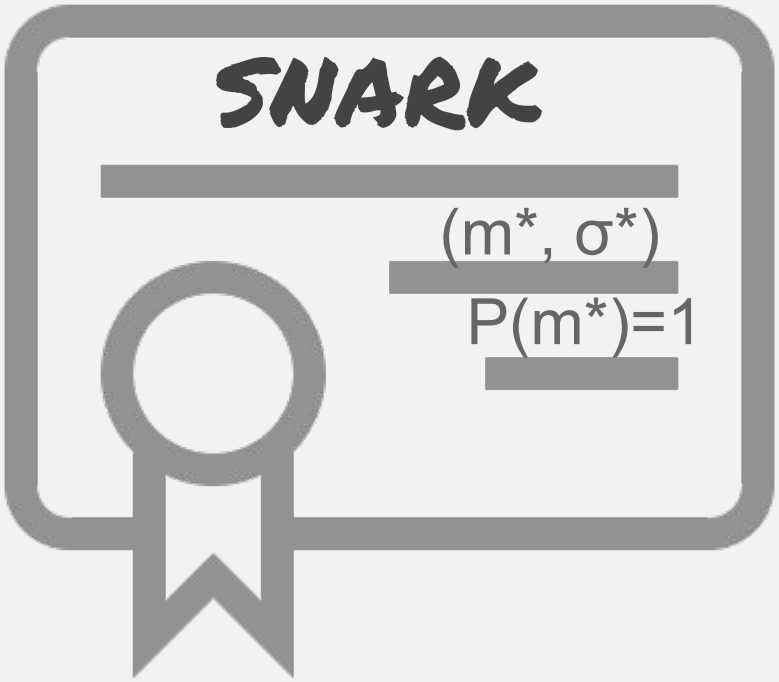


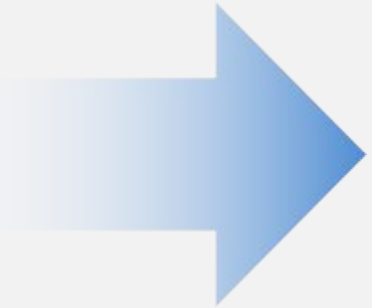


Security Proof



EXTRACTOR





Forgery !

Extraction with Oracles ?

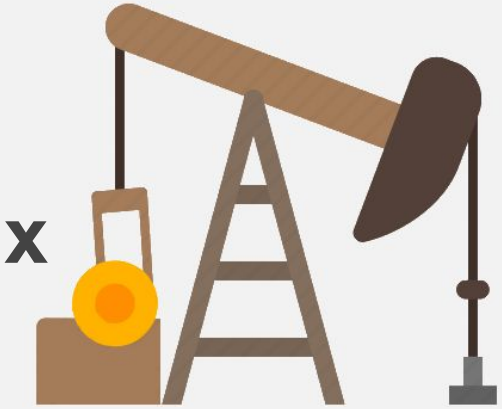
Standard Proof of Knowledge

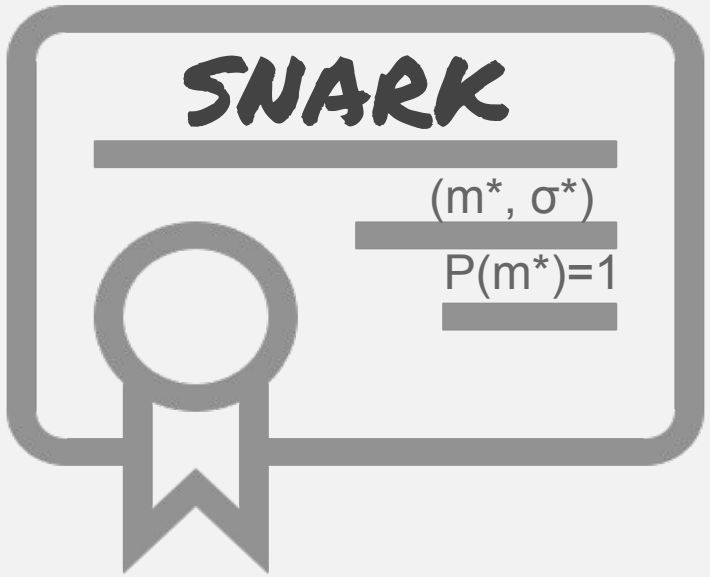
$\mathcal{A}$

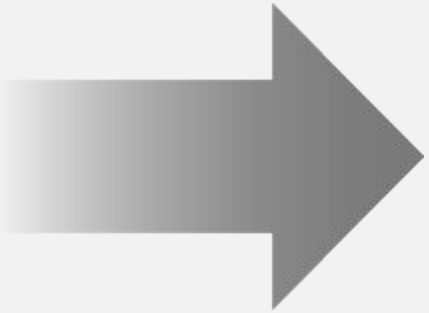
crs, aux

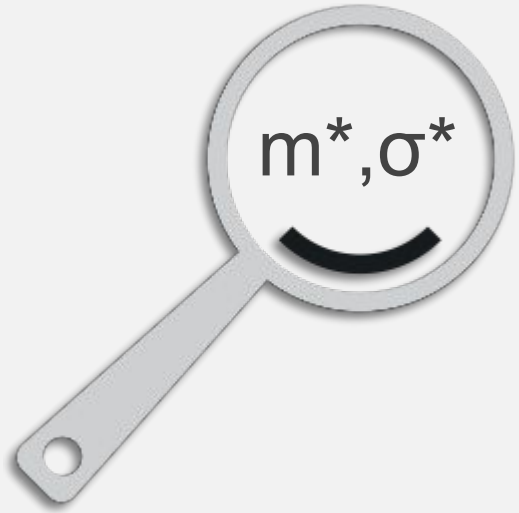
$\mathcal{E}$

crs, aux





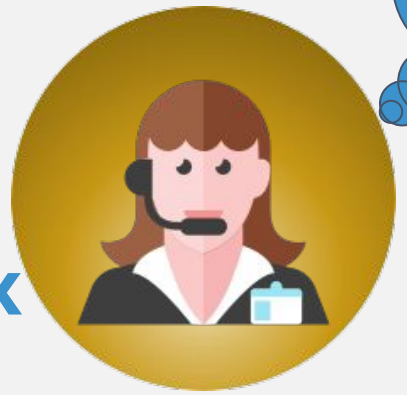




Extraction?

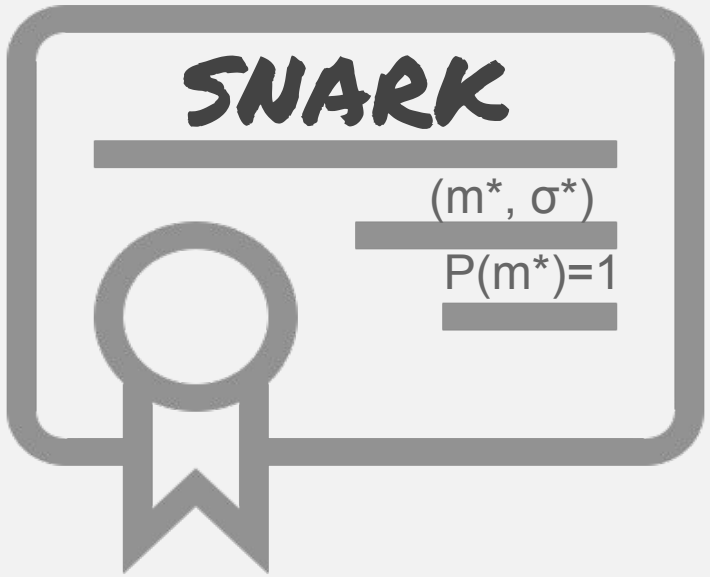
$\mathcal{A}$

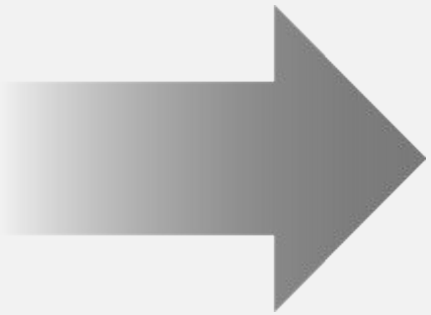
crs, aux



$\mathcal{E}$

crs, aux

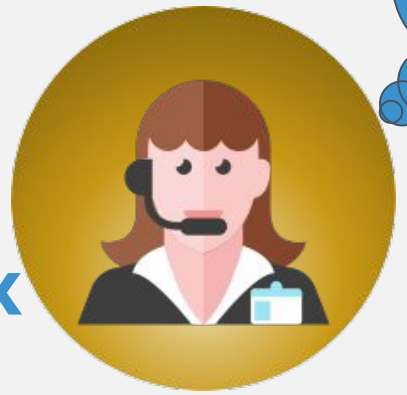




Extraction?

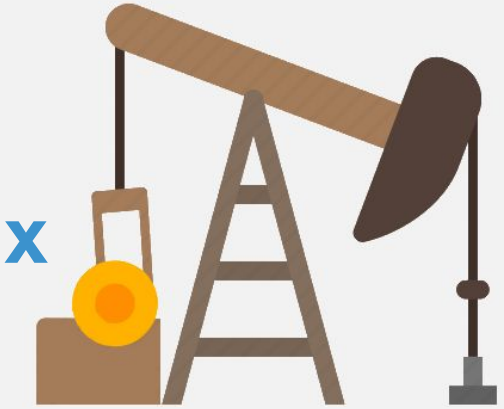
$\mathcal{A}$

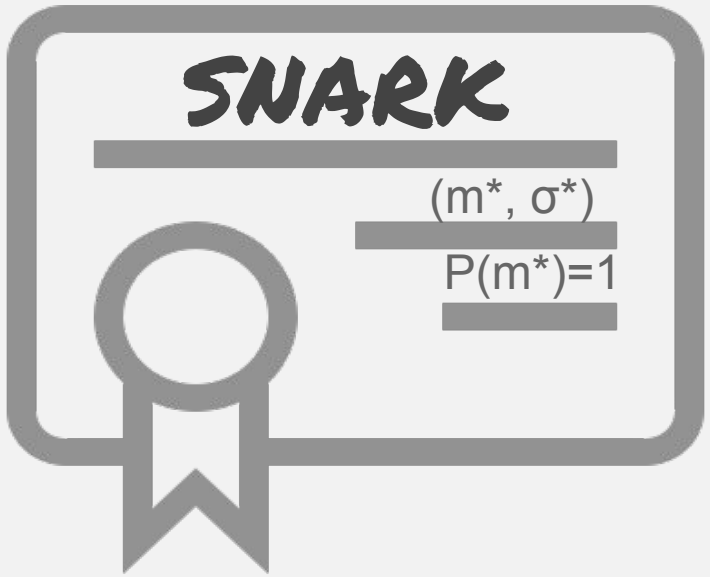
crs, aux

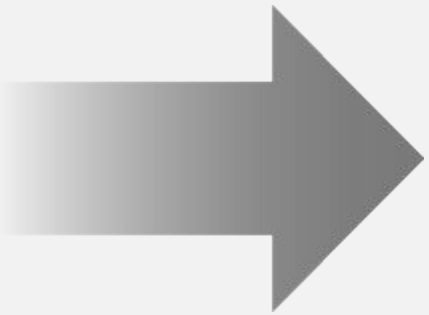


$\mathcal{E}$

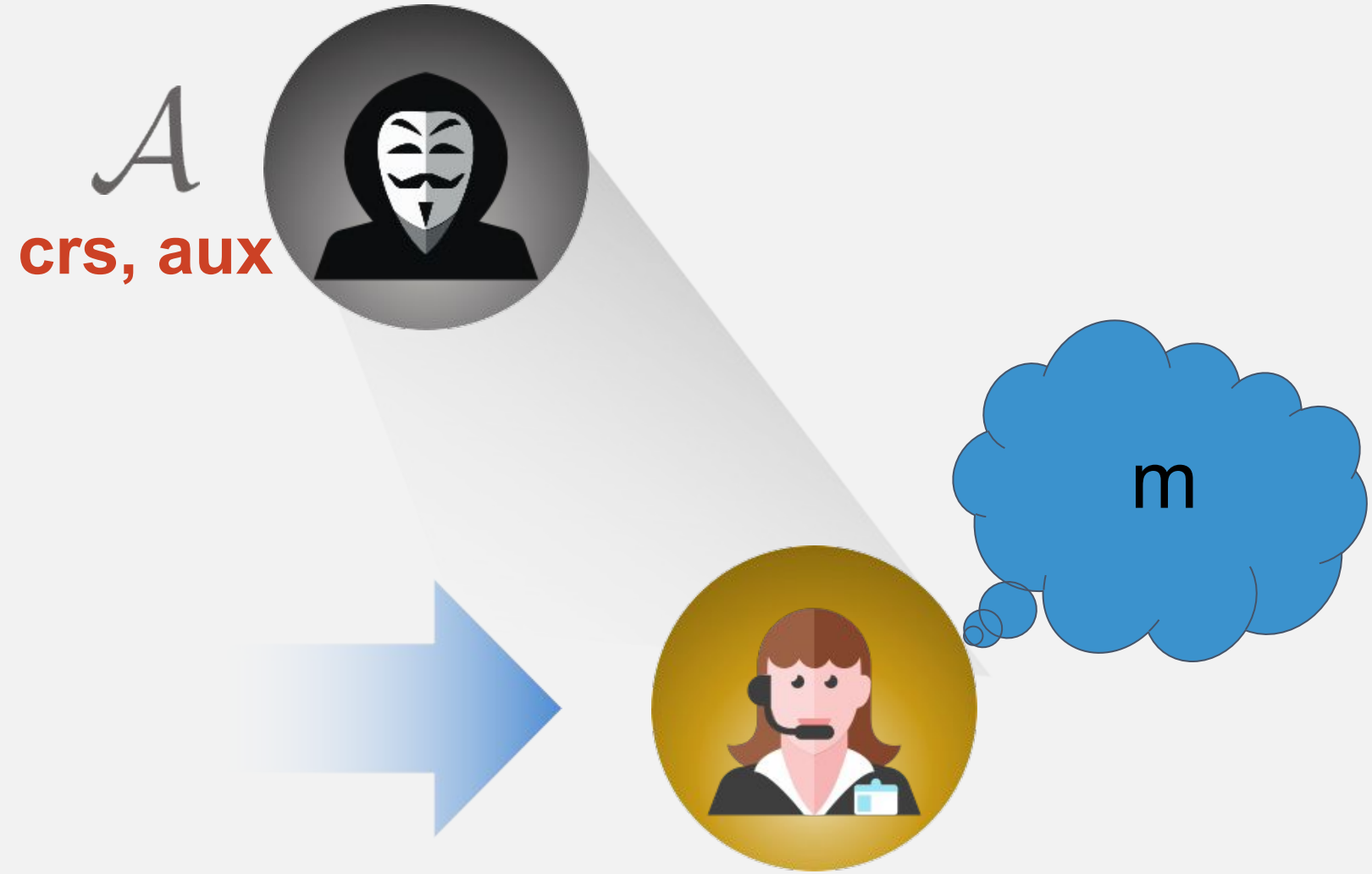
crs, aux



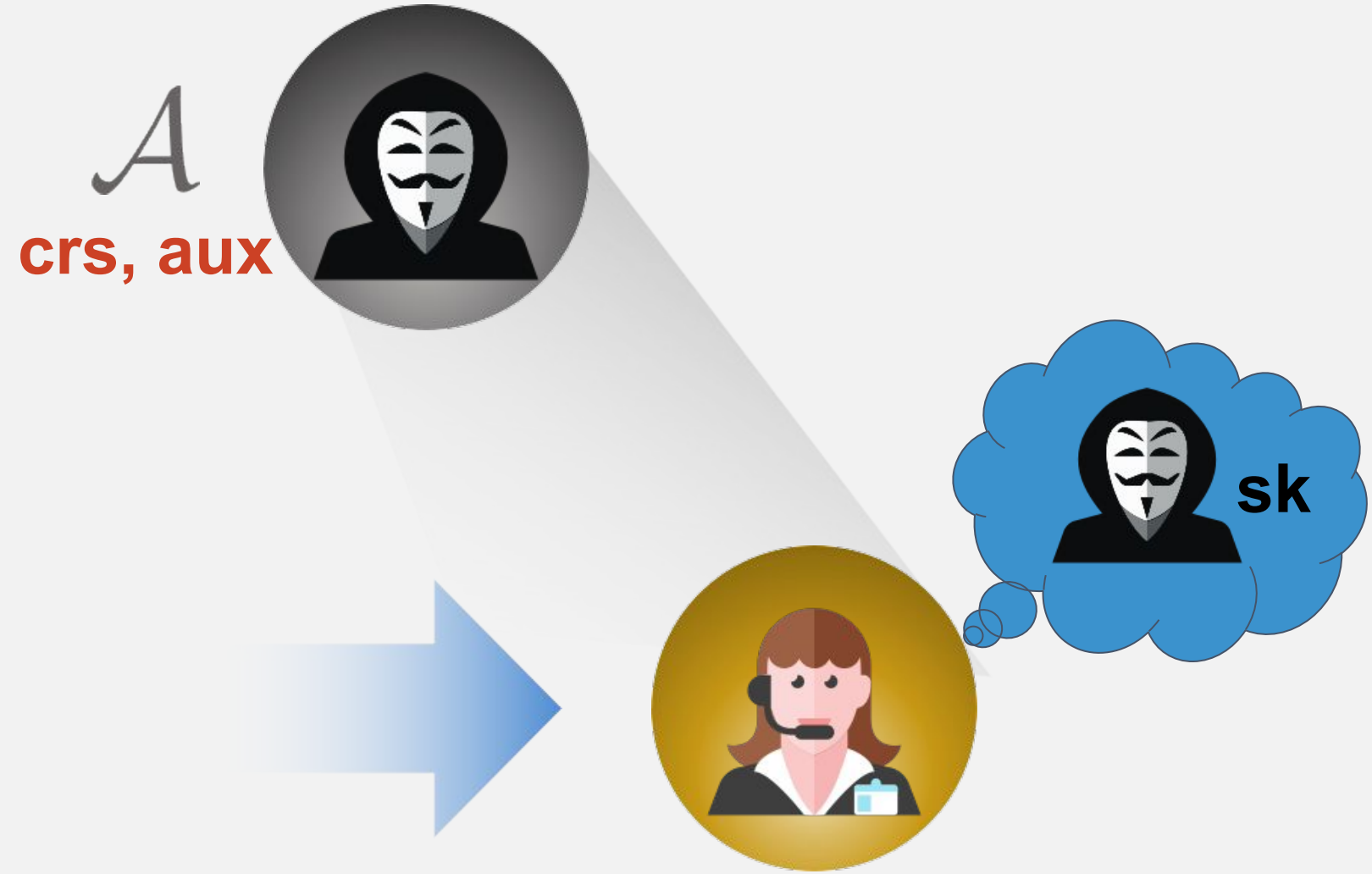




Non-Black-Box Extraction



Non-Black-Box Extraction



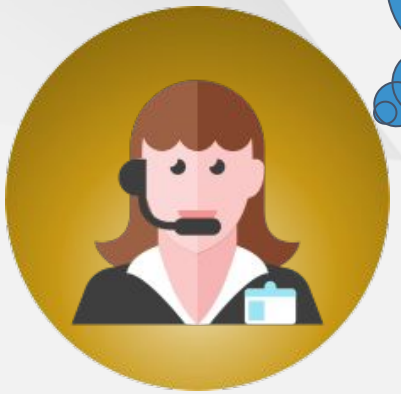
Non-Black-Box Extraction

$\mathcal{A}$

crs, aux
sk

sk ?

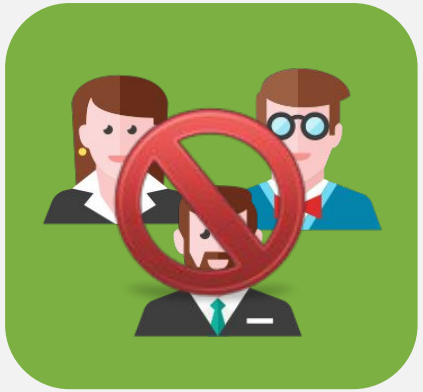
Σ- Forger



$\mathcal{E}$

crs, aux
sk

Our Contributions

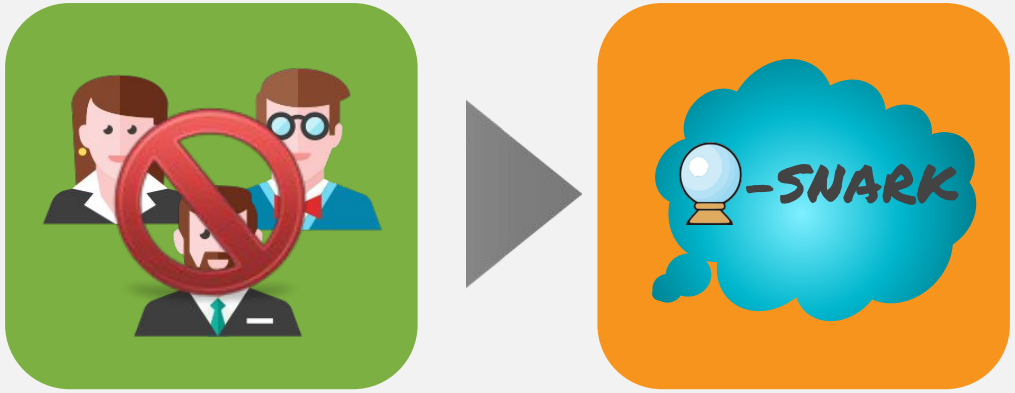


PROTOCOLS:

Tool to prove knowledge -> **SNARK**

Main problem -> Security proofs with ORACLES

Our Contributions



SOLUTION:

New security notion -> **O-SNARK**

Proof of knowledge -> Extraction with ORACLEs

Our Contributions

STUDY:

Impossibility -> Extraction is **NOT** feasible for all ORACLEs

Our Contributions

Good news:

O-SNARKs exist!

-> constructions

-> applications

O-SNARK Definition

$\forall \mathcal{A}^{\mathcal{O}}$

crs, aux

$\exists\ \varepsilon_{\mathcal{A}}$

crs, aux

qt

$$\Pr\left[\begin{array}{c} \mathsf{Ver}(\mathsf{crs}, y, \pi) = 1 \\ \wedge \\ (y, w) \notin R \end{array} \;\middle|\; \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Gen}(1^{\lambda}) \\ (y, \pi) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{crs}, \mathsf{aux}) \\ w \leftarrow \varepsilon_{\mathcal{A}}(\mathsf{crs}, \mathsf{aux}, \mathsf{qt}) \end{array}\right] \approx 0$$

Impossibility Theorem

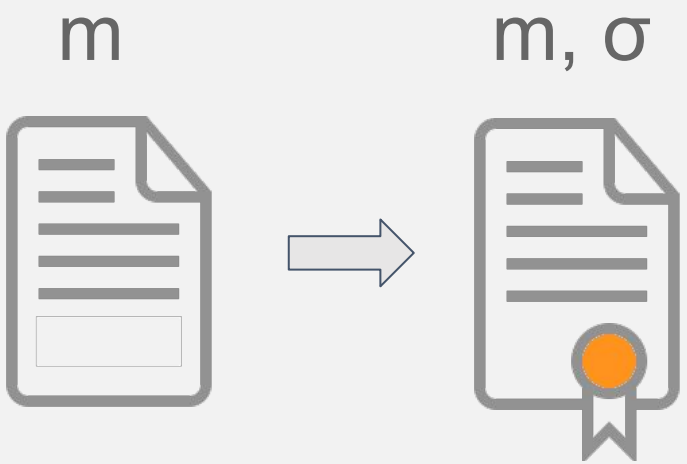
$\forall p(\cdot)\ \exists \Sigma_p$ secure signature scheme

$\forall$ **O-SNARK** Π for NP, such that $|\pi| \leq p(\cdot)$,

does not satisfy proof of knowledge for oracle O_{Σ_p}.

Assumption: One-Way Functions (UOWHF)

Signature Scheme Σ

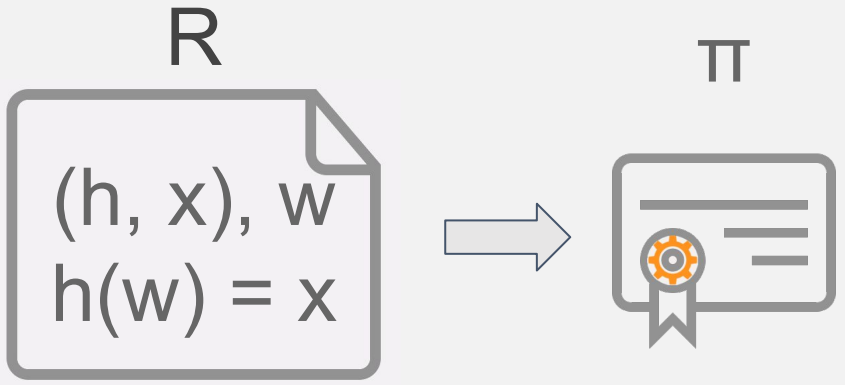


KeyGen(λ): (sk, vk)

Sign(sk, m): σ =**Σ**(m)

Vfy(vk, m, σ): 0 / 1

SNARK Π for NP



Gen(λ): crs

Prove(crs, (h, x), w): π

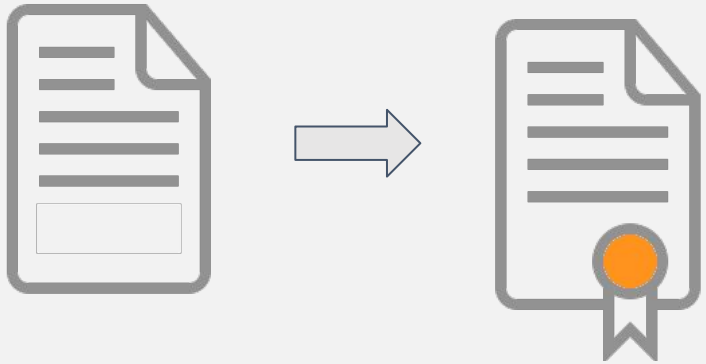
Ver(vk, (h, x), π): 0 / 1

Σ_p

COUNTEREXAMPLE SIGNATURE SCHEME

regular signing

I



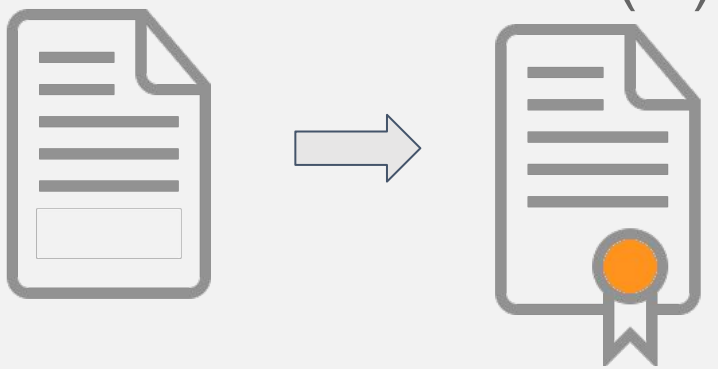
m

$\sigma = \Sigma(m)$

Σ_p

COUNTEREXAMPLE SIGNATURE SCHEME

I



sampling hash preimage

II

$h \leftarrow \mathcal{H}$

$w \leftarrow \{0,1\}^*$

$x = h(w)$

Σ_p COUNTEREXAMPLE SIGNATURE SCHEME

I

m

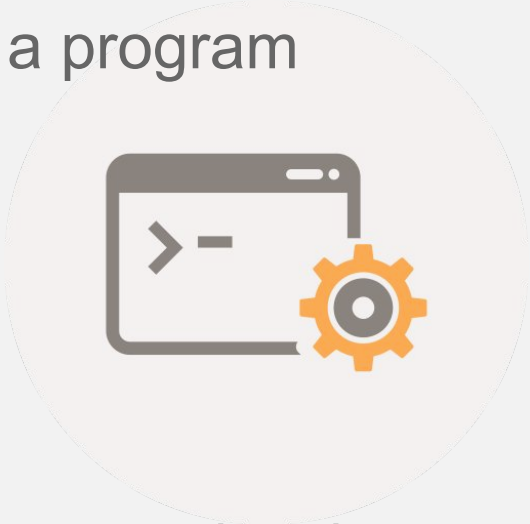
σ =**Σ**(m)

II

h ← $\mathcal{H}$
w ← {0,1}*
x = h(w)

III

interpreting m as a program

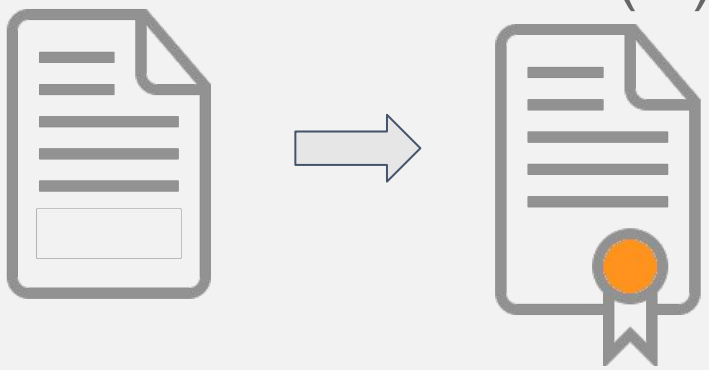


m

P(· , ·)

Σ_p COUNTEREXAMPLE SIGNATURE SCHEME

I



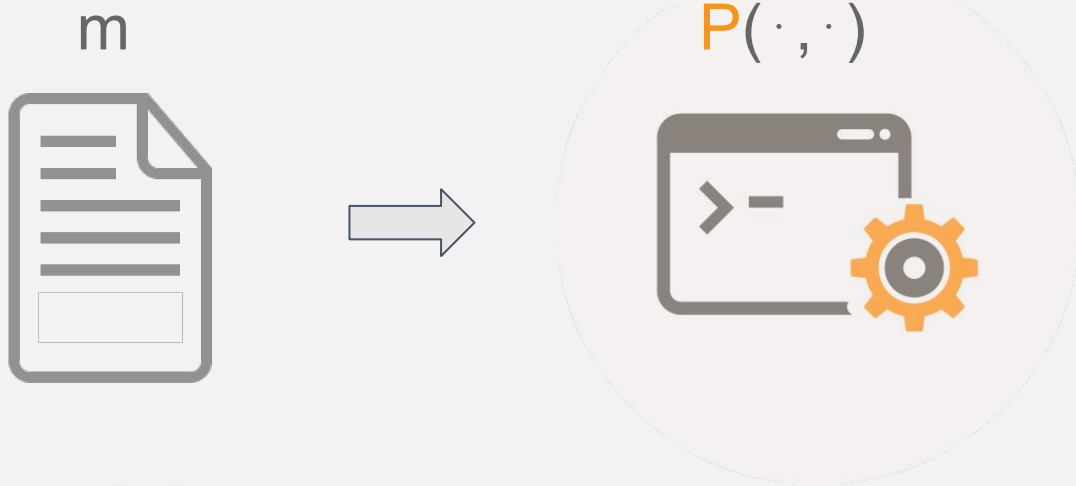
II

$h \leftarrow \mathcal{H}$

$w \leftarrow \{0,1\}^*$

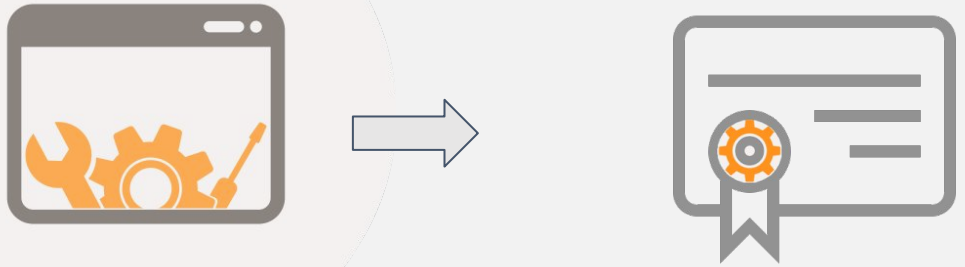
$x = h(w)$

III



proving knowledge of preimage

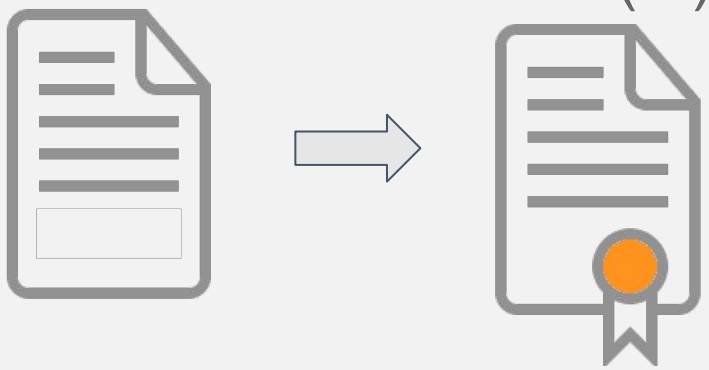
IV



$|\pi| < p(\lambda)$

Σ_p COUNTEREXAMPLE SIGNATURE SCHEME

I



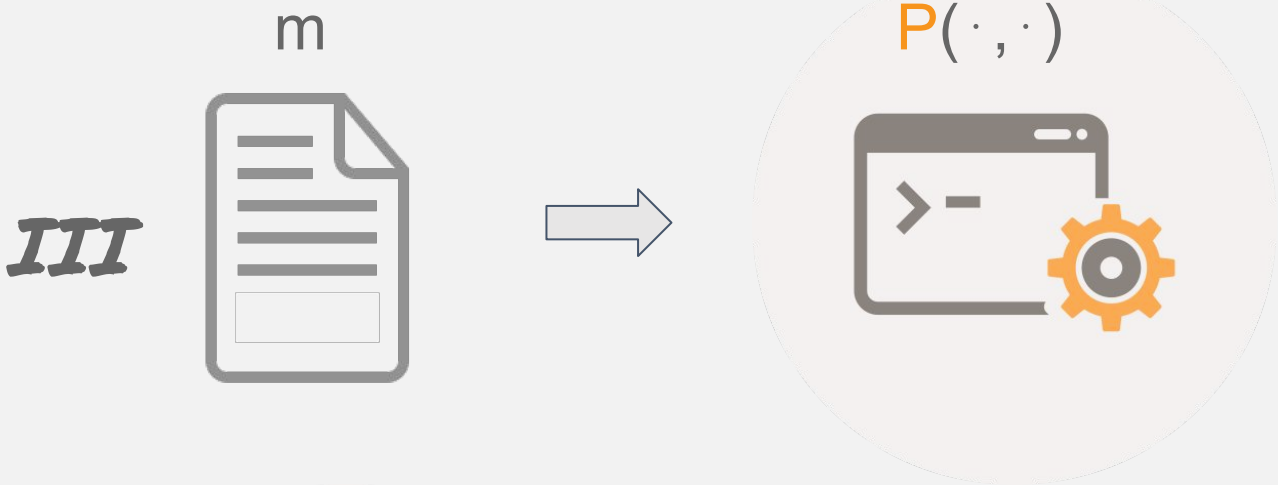
II

$h \leftarrow \mathcal{H}$

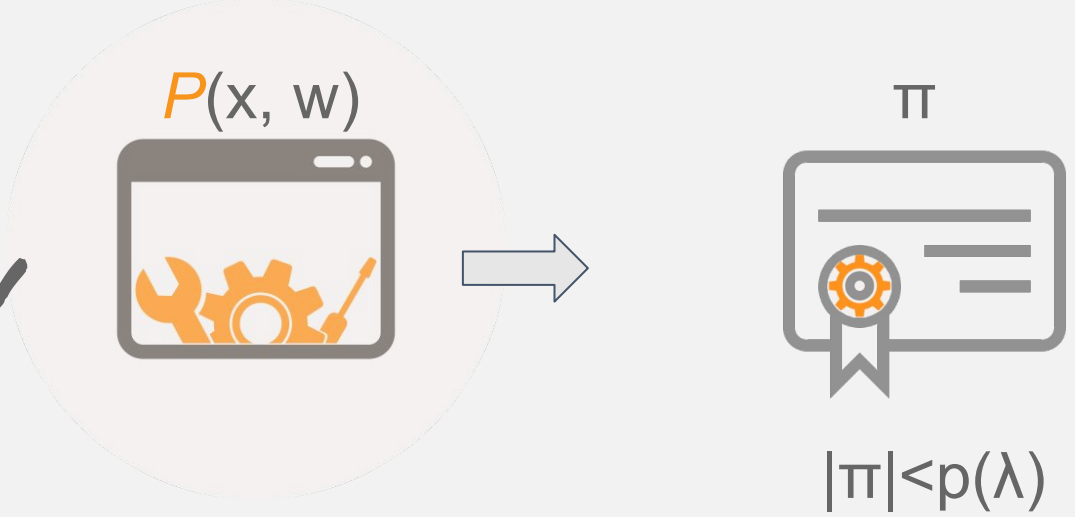
$w \leftarrow \{0,1\}^*$

$x = h(w)$

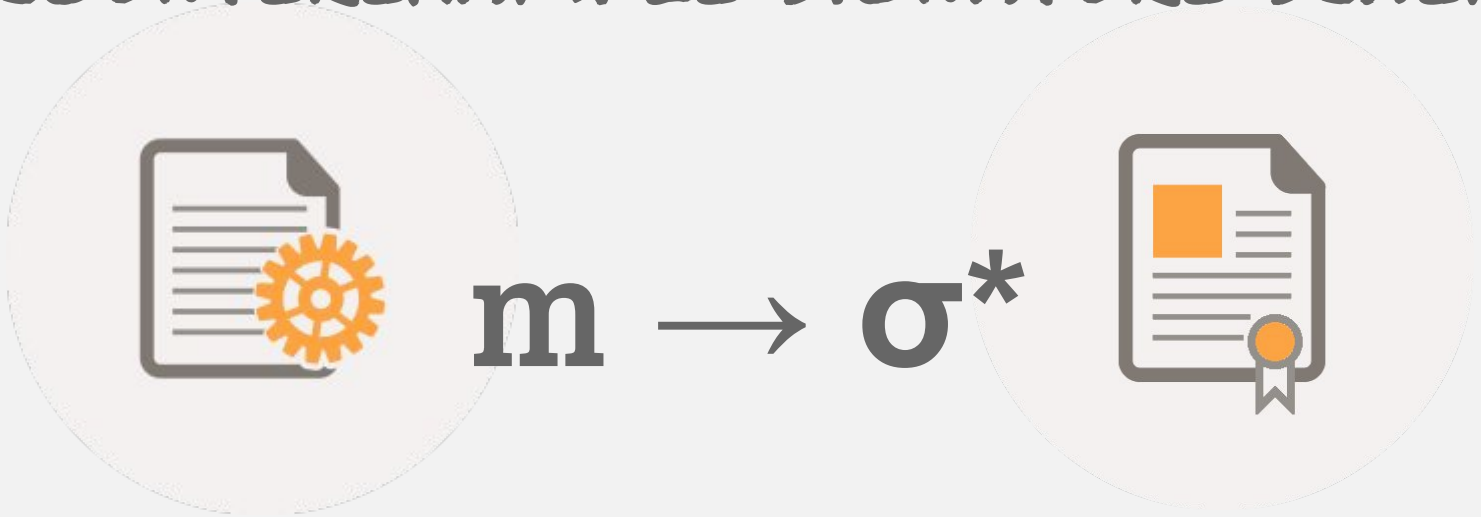
III



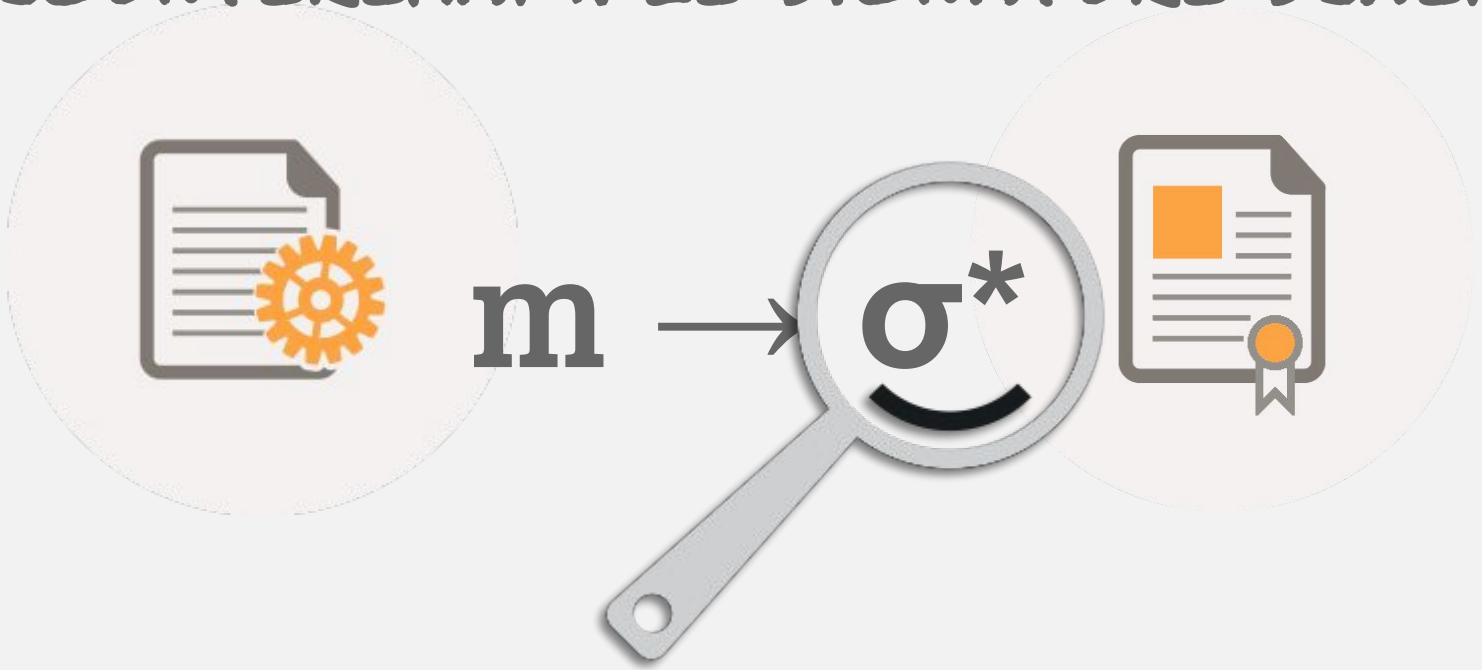
IV



COUNTEREXAMPLE SIGNATURE SCHEME



Σ_p COUNTEREXAMPLE SIGNATURE SCHEME

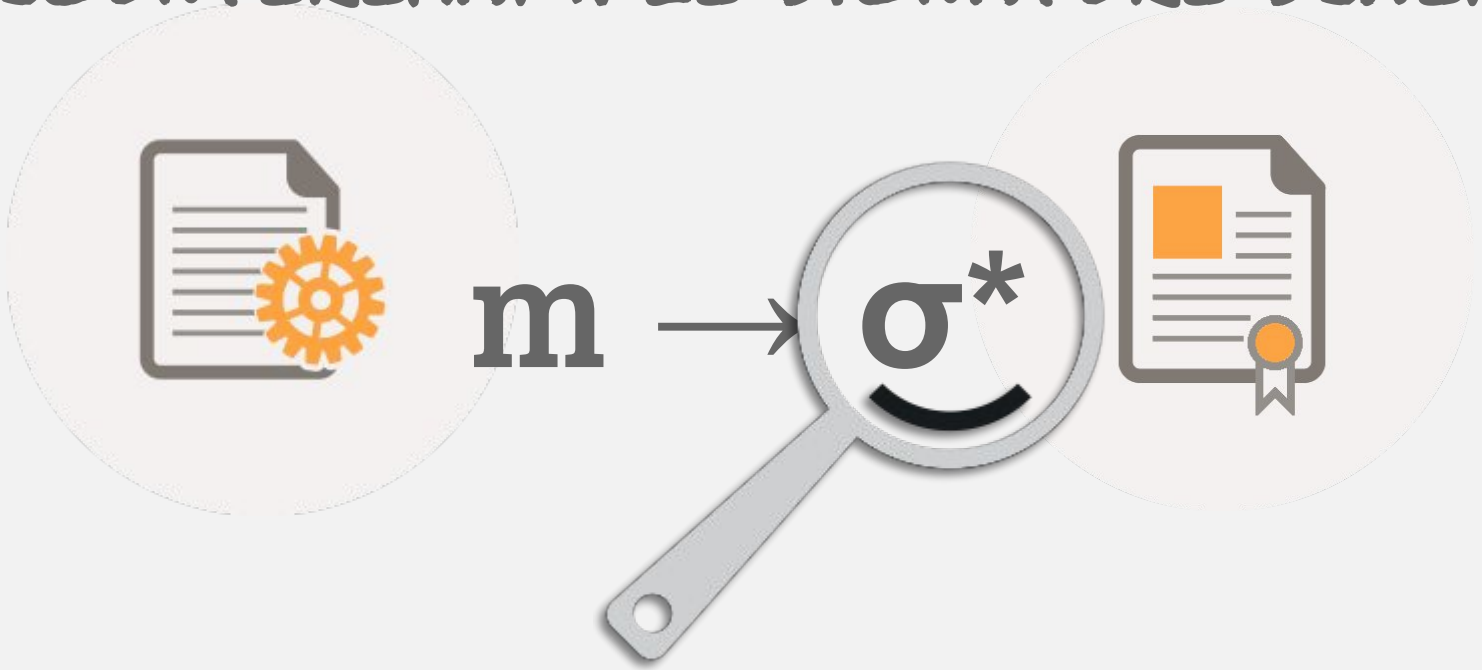


$\sigma \leftarrow \Sigma(m)$

COUNTEREXAMPLE SIGNATURE SCHEME

$h \leftarrow \mathcal{H}$

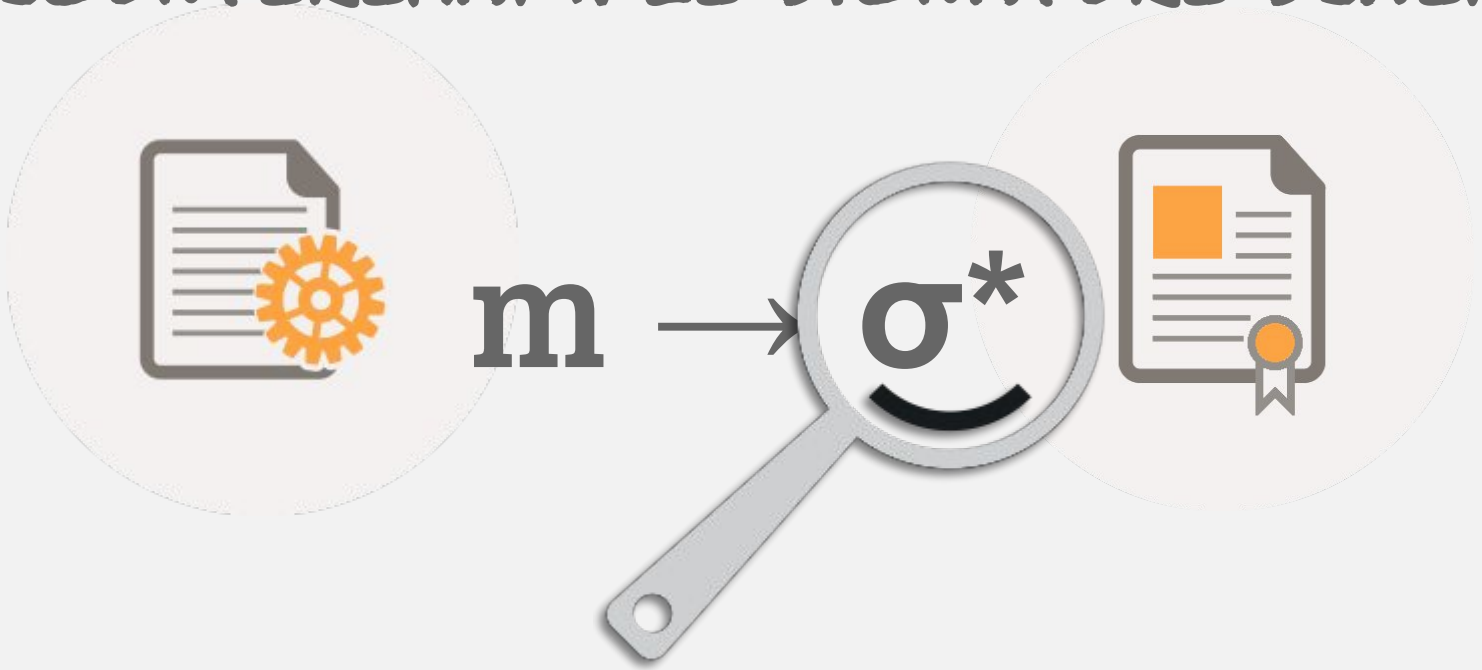
Σ_p COUNTEREXAMPLE SIGNATURE SCHEME

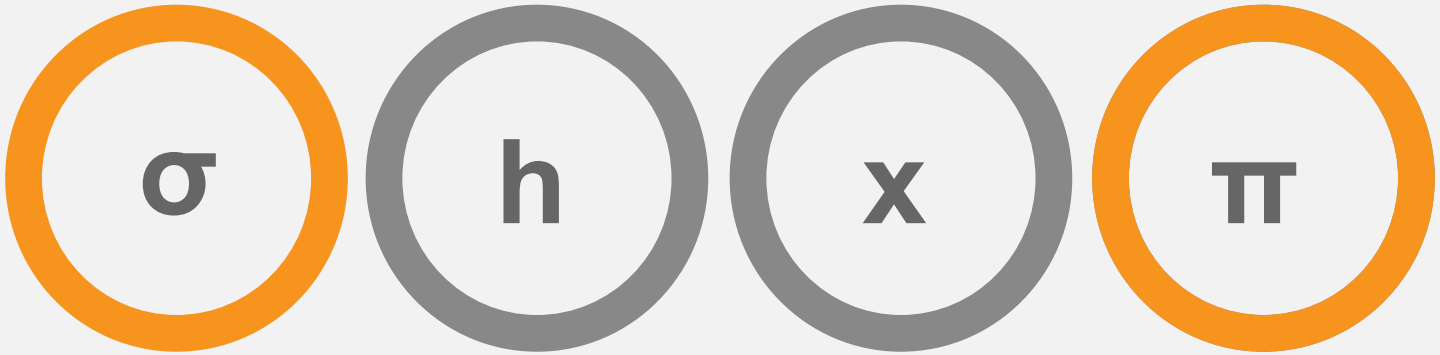


$w \leftarrow \{0,1\}^*$

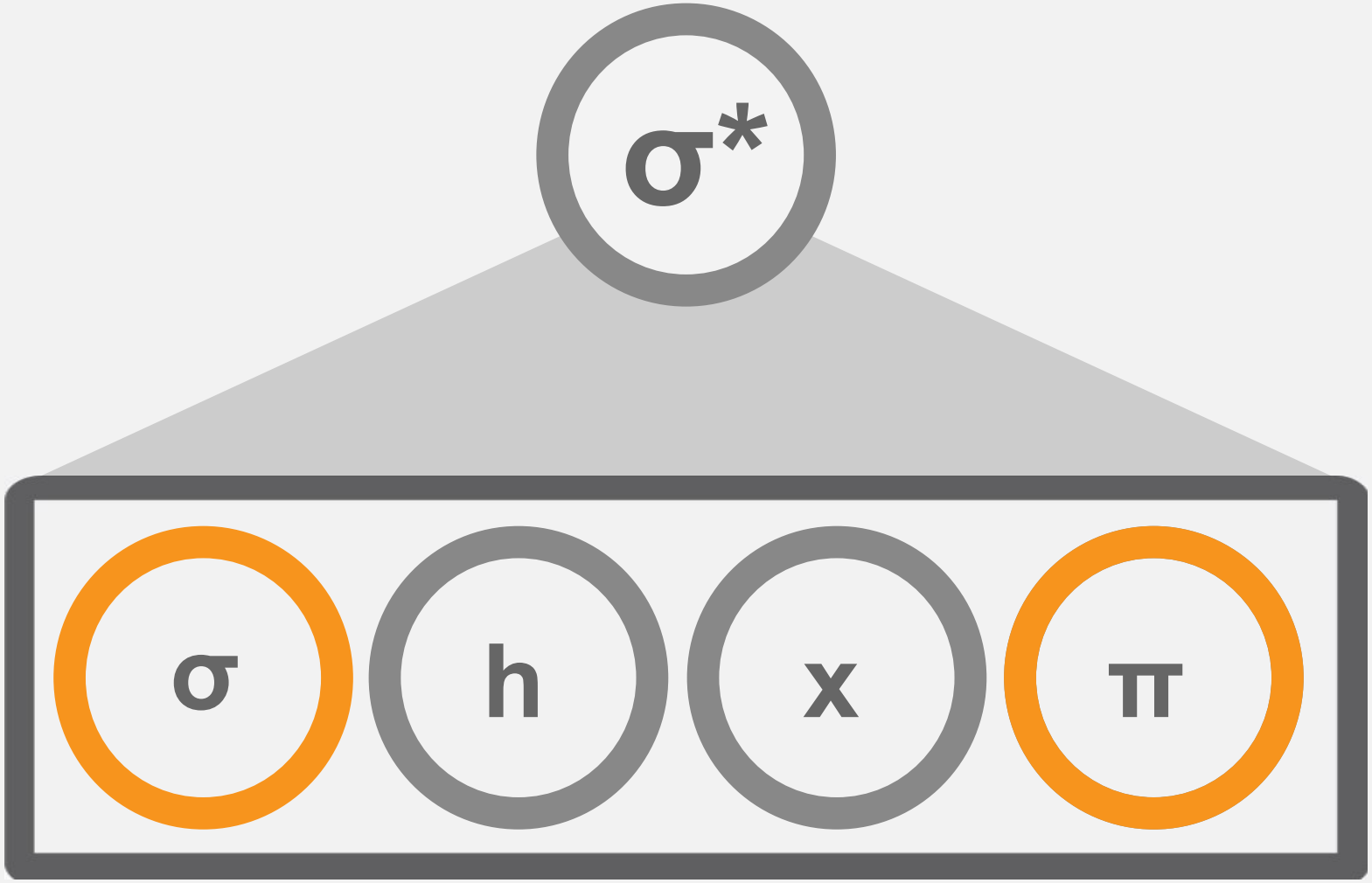
$x = h(w)$

COUNTEREXAMPLE SIGNATURE SCHEME





COUNTEREXAMPLE SIGNATURE SCHEME



Non-Existence of Extractors for O^{Σ_p}

QUERY

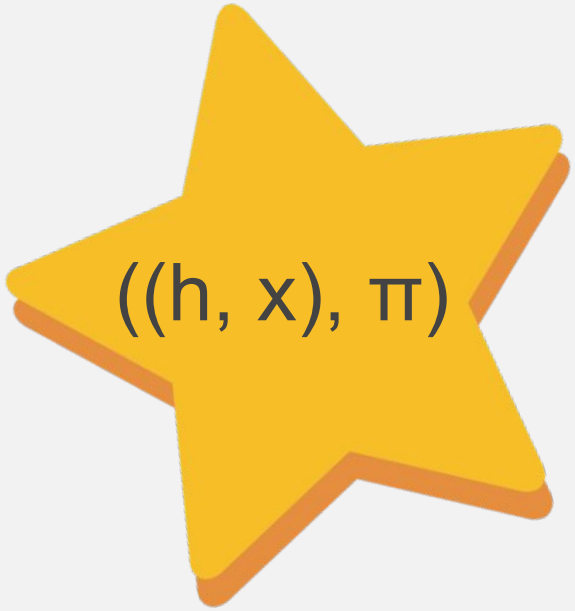
m = Prove(crs, · , ·)

ANSWER

σ* = (σ, h, x, π)

π = Prove(crs, (h, x), w)

O-SNARK Adversary



QUERY

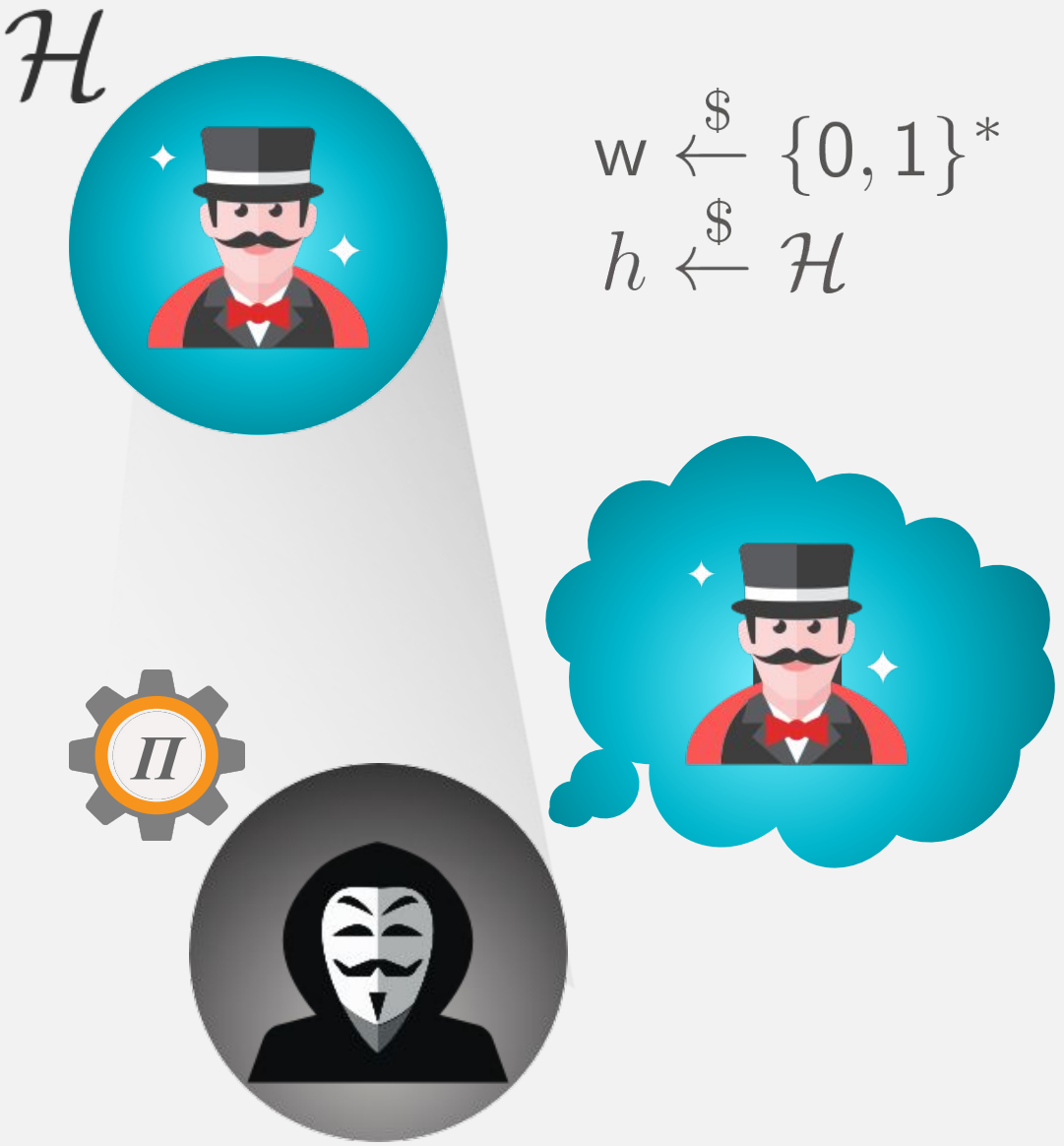
m = Prove(crs, · , ·)

ANSWER

σ* = (σ, h, x, π)

π = Prove(crs, (h, x), w)

Target Collision Resistance Adversary



$\mathsf{w} \xleftarrow{\$} \{0,1\}^*$
$h \xleftarrow{\$} \mathcal{H}$

$\mathrm{crs} \leftarrow \mathsf{Gen}(1^\lambda)$
$(\mathrm{sk}, \mathrm{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$

QUERY $\mathsf{m} = \mathsf{Prove}(\mathrm{crs}, \cdot, \cdot)$

ANSWER

$\mathrm{x} = \mathrm{h}(\mathrm{w}), \quad \sigma = \Sigma(m)$
$\sigma^* = (\sigma, h, x, \pi)$
$\pi = \mathsf{Prove}(\mathrm{crs}, (h, x), \mathsf{w})$

Target Collision Resistance Adversary

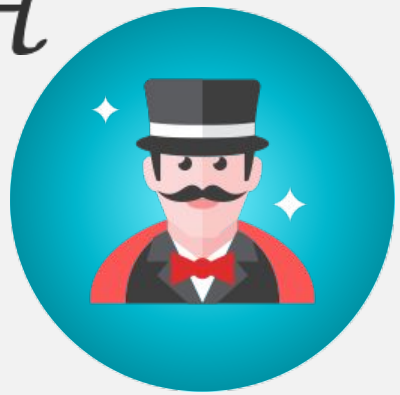
$\mathcal{H}$

((h, x), π)

π = Prove(crs, (h, x), w)

Non-Existence of Extractors for O^{Σ_p}

$\mathcal{H}$



$$|\pi| \leq p(\lambda)$$

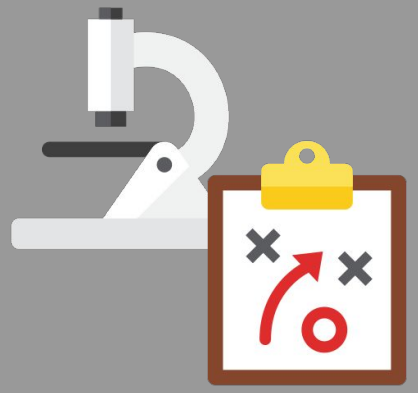
$$\mathsf{w} \neq \mathsf{w}^*$$

$$h(\mathsf{w}) = h(\mathsf{w}^*) = x$$

Target Collision on h

Existence of O-SNARK

- O-SNARKs do not exist for all Oracles
- Overcome the impossibility?
- "Break" the adaptive power of the adversary



Existence of O-SNARK

Random Oracle Model

- Micali's CS proofs are O-SNARKs in ROM
- Hash & Sign Oracles allow O-SNARKs

Standard Model

- Signing oracles with polynomial message space
- Non-Adaptive O-SNARKs: queries declared in advance

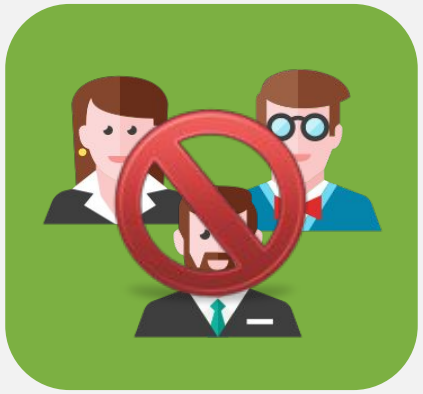
Applications of O-SNARK

- Succinct Functional Signatures [BGI14]
- Homomorphic Signatures [BF11]
- SNARKs on Authenticated Data [BBFR15]

Open Questions

- Artificial counterexamples: Find *“more natural”* ones?
- For what classes of signature oracles O-SNARKs exist?
- Find other “*benign*” Oracles that allow O-SNARKs?

Summary



Starting Point

- Protocols with SNARKs
- Security proofs: settings where **NO** extraction
- New security notion: **O-SNARK**

Study of O-SNARKs

- Impossibility result for Σ_p
- Some “restrictive” instantiations from SNARKs
- Applications where O-SNARK is useful

Thank you