On Attacking Hash functions in Cryptographic schemes

Workshop "Quantum cryptanalysis of post-quantum cryptography"
Simons institute for the Theory of Computing

Christian Majenz

CWI
Centrum Wiskunde & Informatica

QuSoft
Research Center for Quantum Software

Hash functions…

Hash functions…

…are everywhere in cryptography.

Hash functions…

…are everywhere in cryptography.

Hash functions…

…are everywhere in cryptography.

Hash functions…

…are everywhere in cryptography.

Hash functions…

…are everywhere in cryptography.

- Commitments
- Noninteractive zero knowledge
- …

Hash functions…

…are everywhere in cryptography.

- Commitments
- Noninteractive zero knowledge
- …

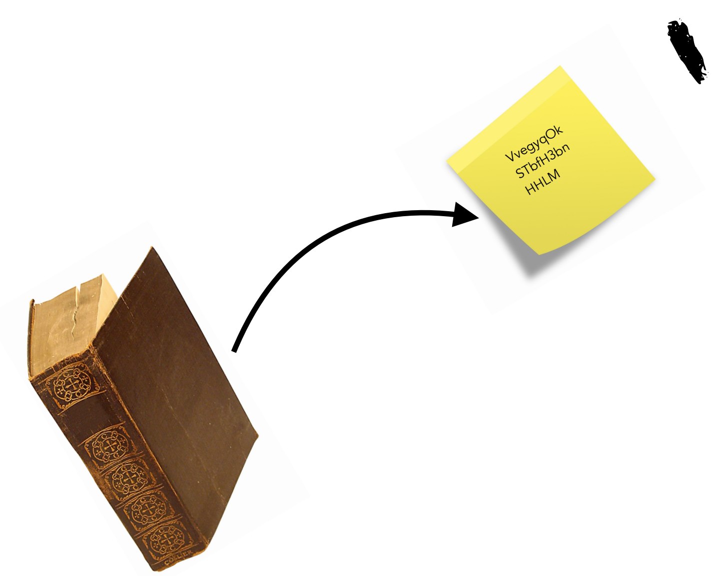
Outline

1. Intro: Hash functions
    i. Basics, security
    ii. The (quantum) random oracle model
    iii. Domain extension
2. Points of attack
3. Hash-function-based generic transformations: Fiat-Shamir and Fujisaki-Okamoto
4. Attacks and attack approaches against Fiat-Shamir and Fujisaki-Okamoto

Intro: Hash functions

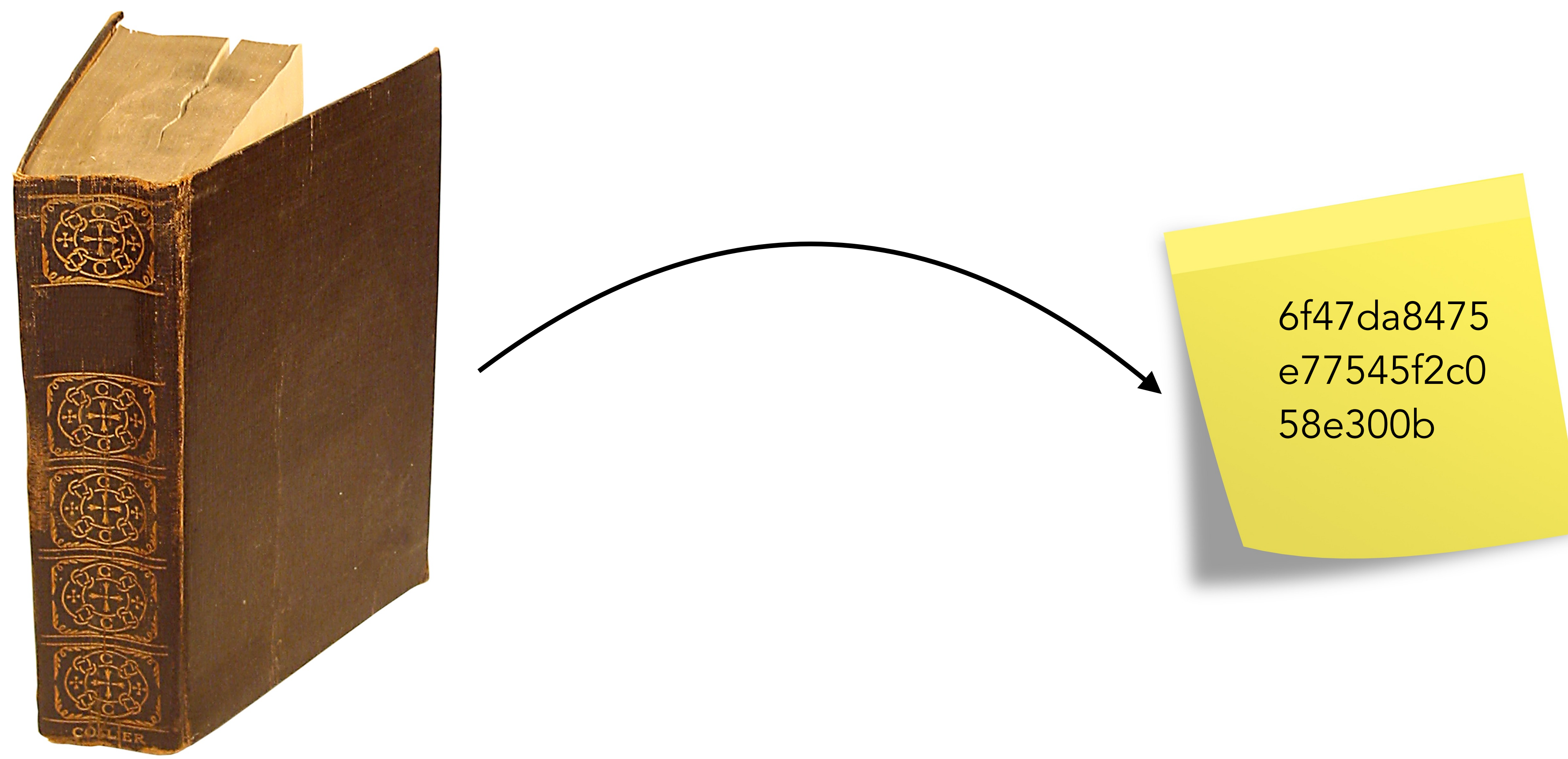
#cryptoisawesome?

What is a hash function?



Hash functions

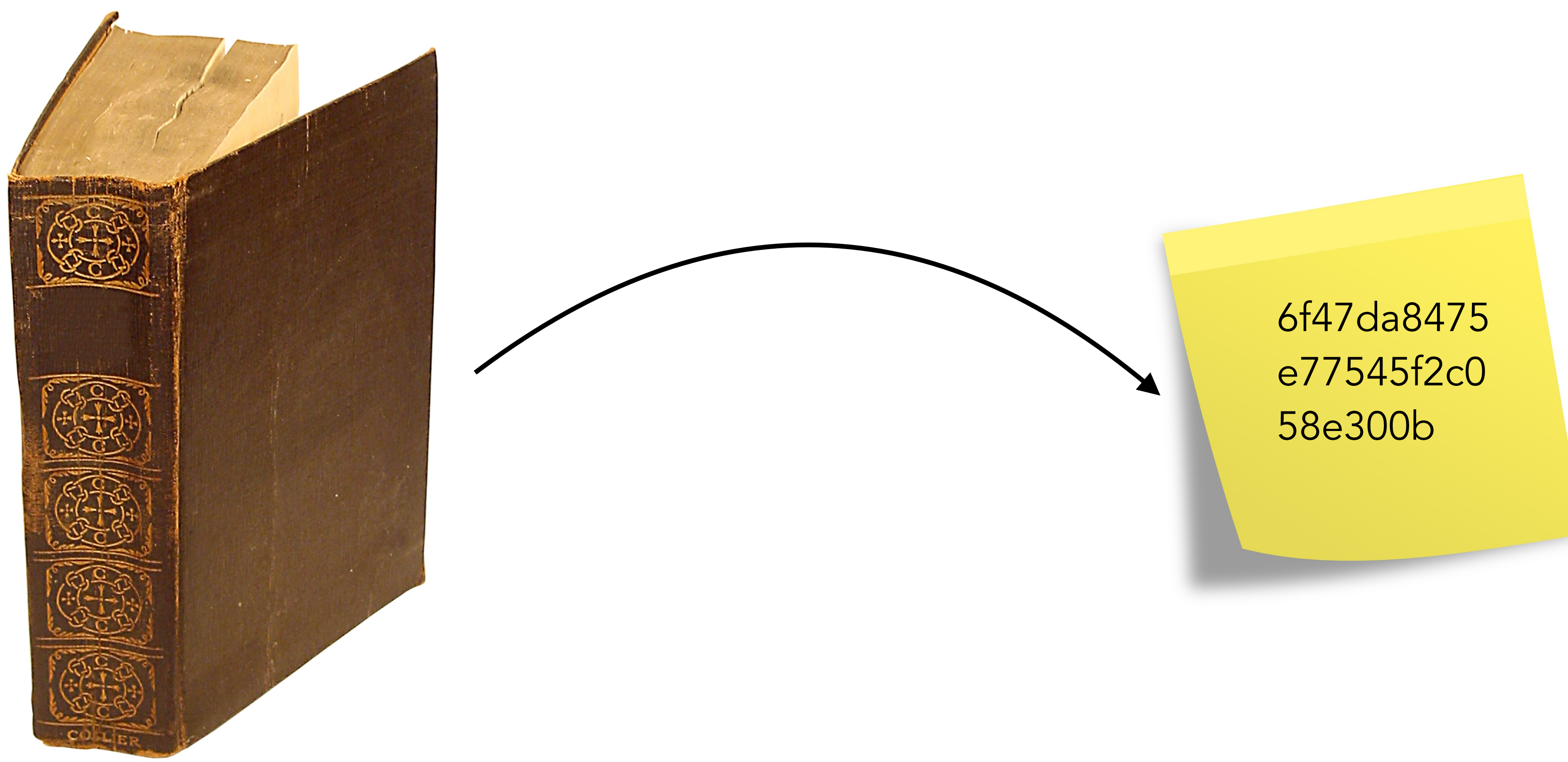
Definition: Function (family) $H : \{0,1\}^* \rightarrow \{0,1\}^n$



Hash functions

Definition: Function (family) $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Defining intuition: Hash functions "look random"



Hash functions

Definition: Function (family) $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Defining intuition: Hash functions "look random"

Security properties informed by this intuition:

Hash functions

Definition: Function (family) $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Defining intuition: Hash functions "look random"

Security properties informed by this intuition:

- One-wayness
- Collision resistance
- Collapsingness
- Correlation intractability
- Bernoulli preservingness
- …

Hash functions

Definition: Function (family) $H : \{0,1\}^* \to \{0,1\}^n$

Defining intuition: Hash functions "look random"

Security properties informed by this intuition:

- One-wayness
- Collision resistance
- Collapsingness
- Correlation intractability
- Bernoulli preservingness
- …

Random function has all of these properties

Hash functions

Definition: Function (family) $H : \{0,1\}^* \to \{0,1\}^n$

Defining intuition: Hash functions "look random"

Security properties informed by this intuition:

- One-wayness
- Collision resistance
- Collapsingness
- Correlation intractability
- Bernoulli preservingness
- …

Random function has all of these properties

$\Longrightarrow$(Quantum) Random Oracle Model

Example application: Hash-and-sign

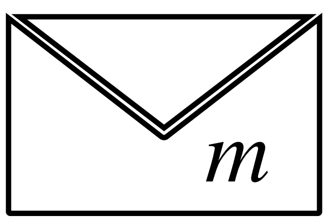
Example application: Hash-and-sign

Alice

Bob

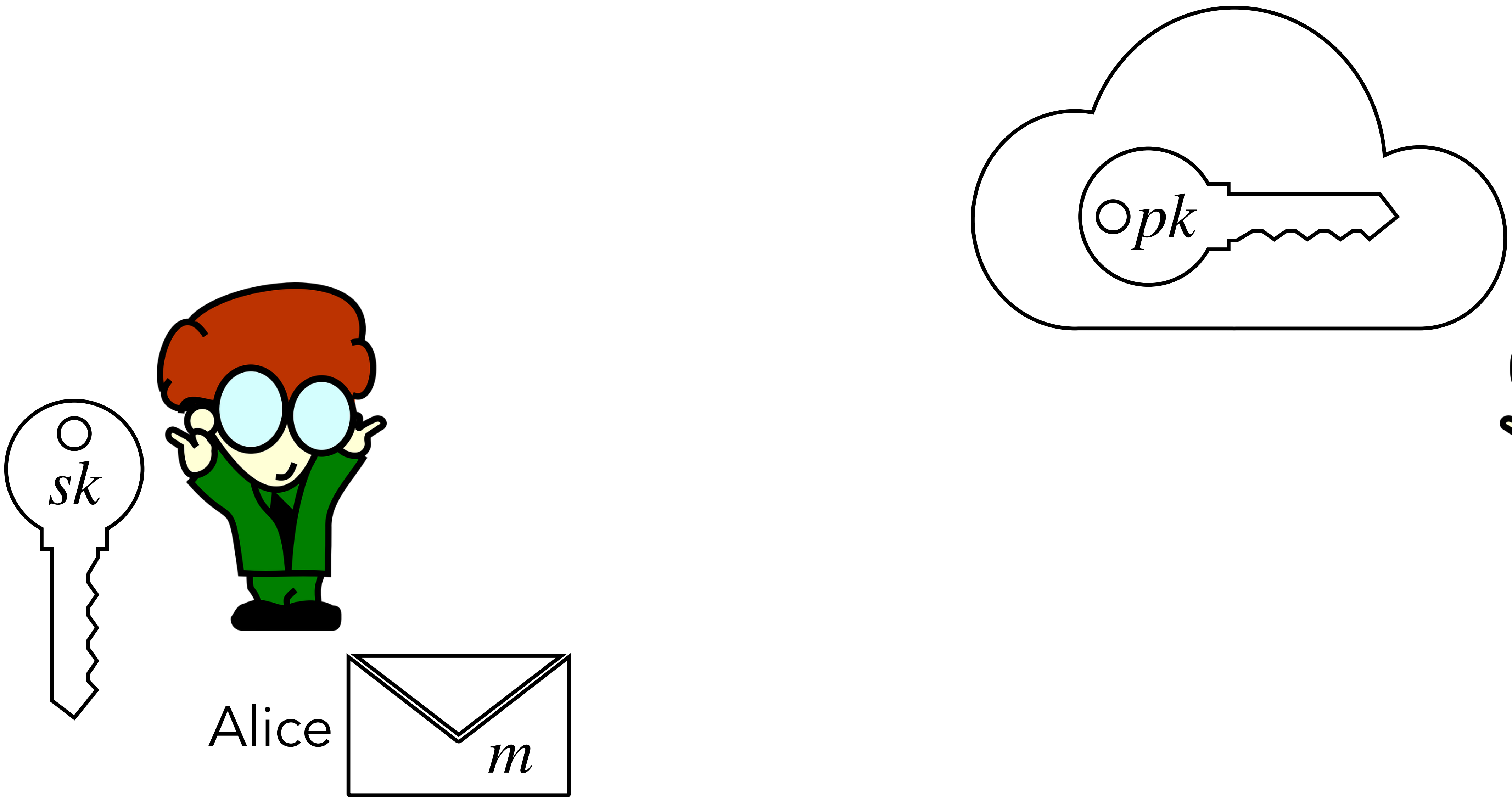
Example application: Hash-and-sign

Alice

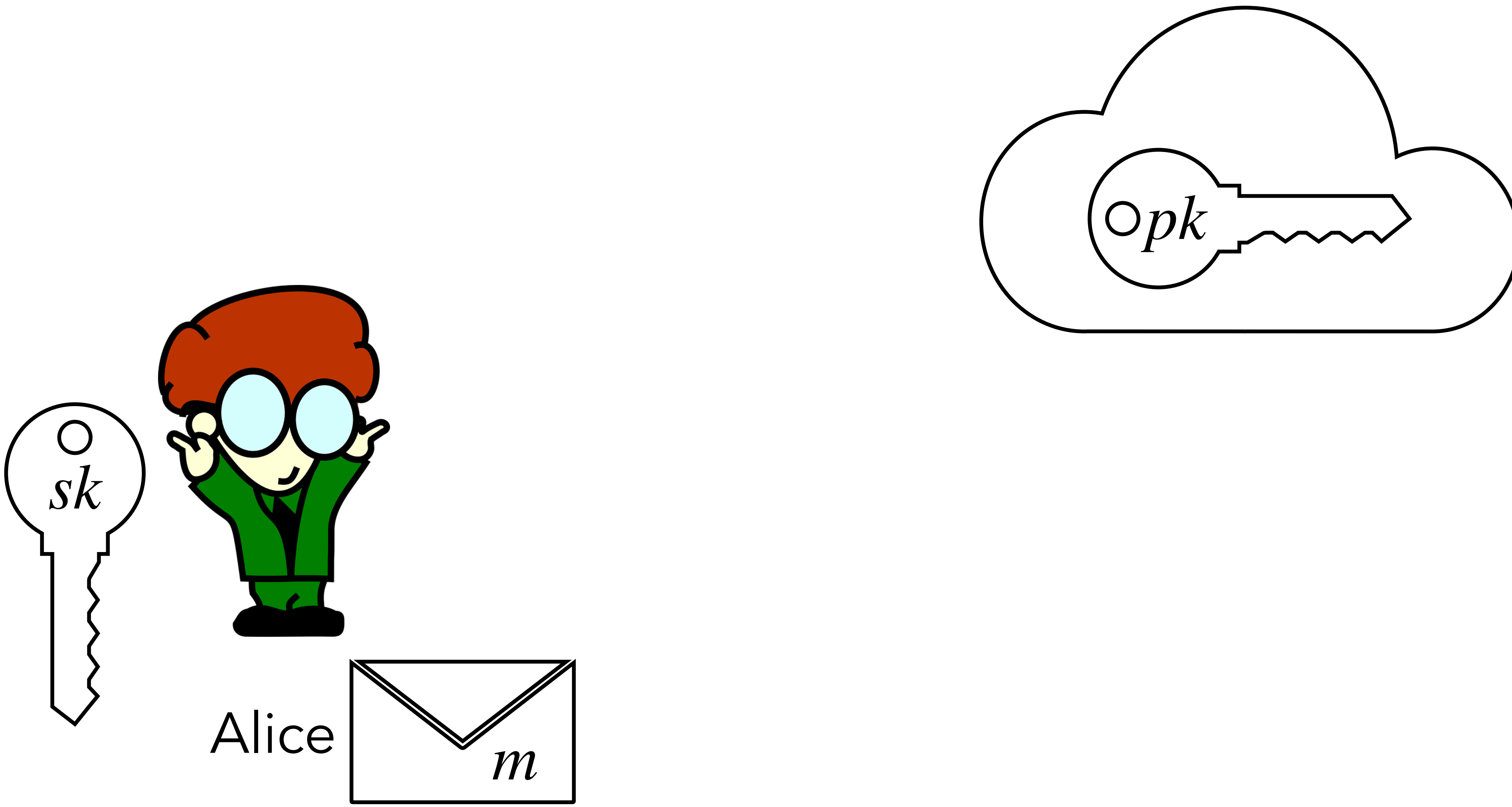


Bob

Example application: Hash-and-sign

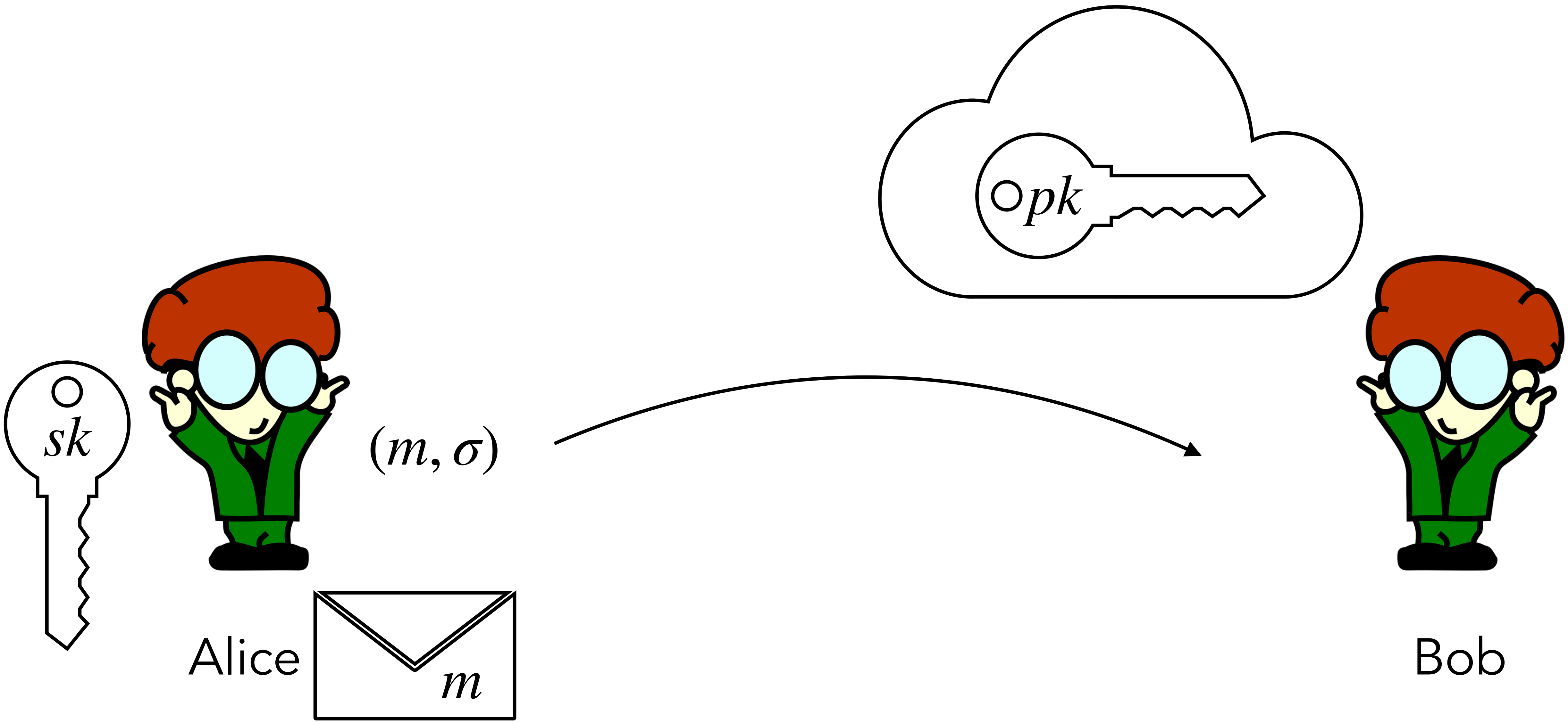


Example application: Hash-and-sign



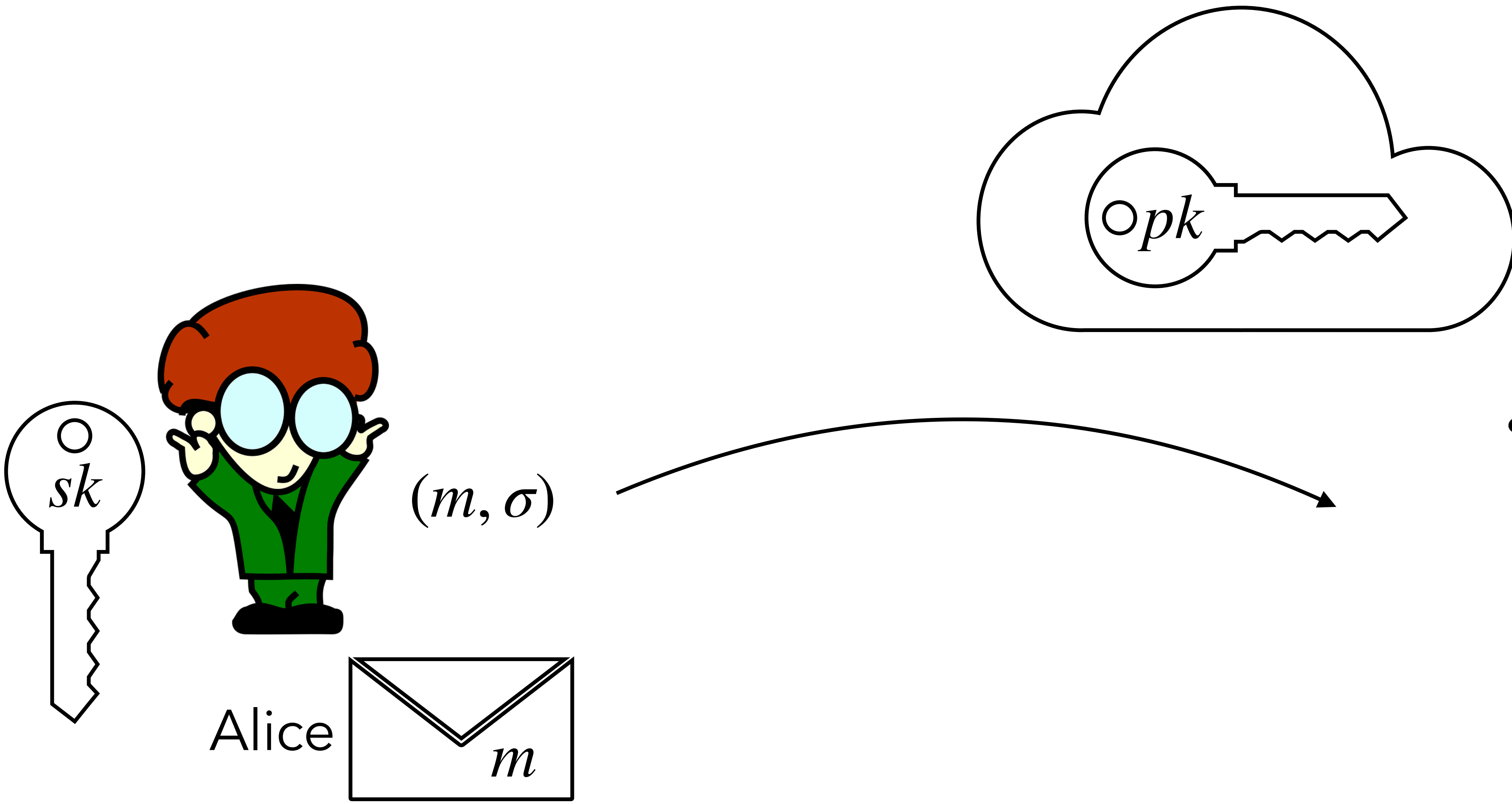
$$\sigma = \mathrm{Sign}_{sk}(m)$$

Example application: Hash-and-sign



$$\sigma = \mathrm{Sign}_{sk}(m)$$

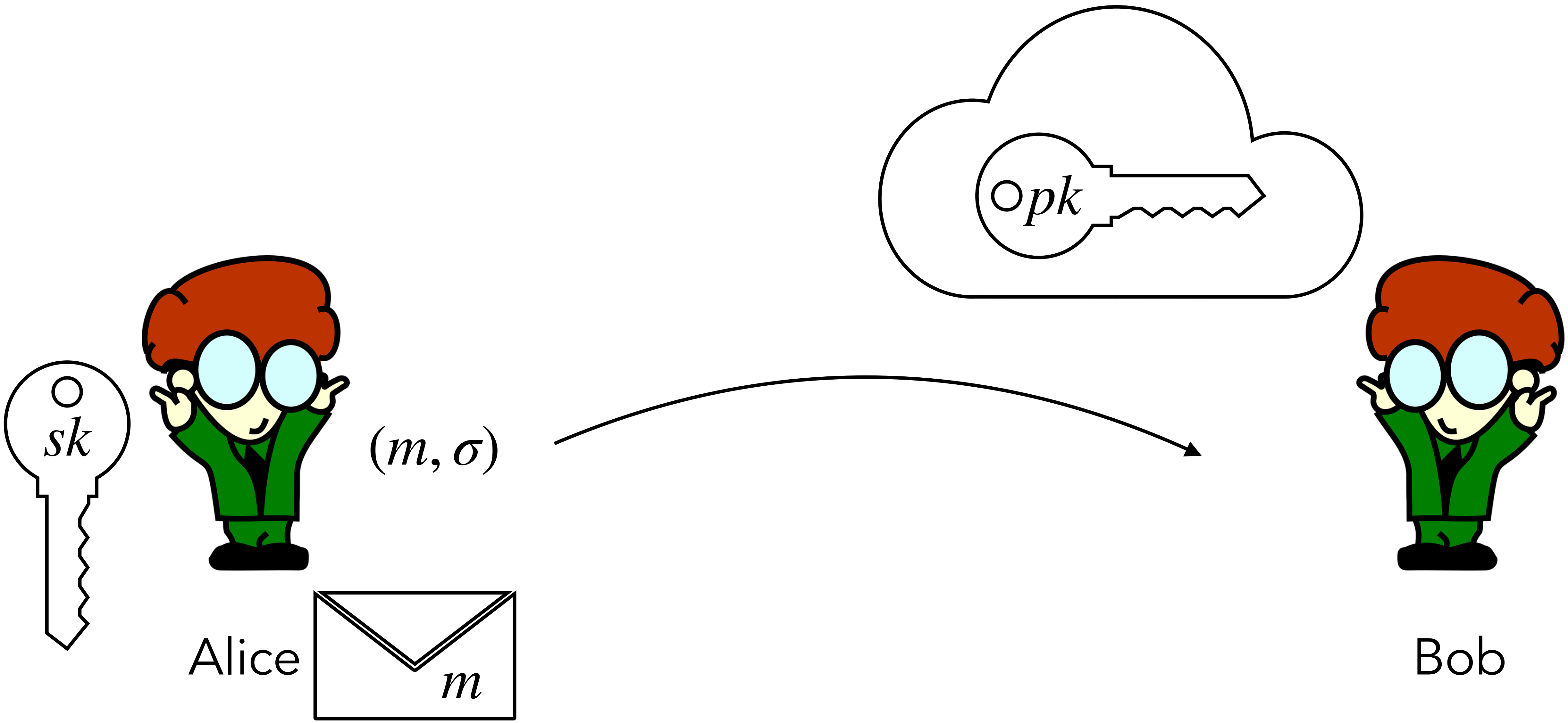
Example application: Hash-and-sign



$\sigma = \mathrm{Sign}_{sk}(m)$

- $\mathrm{Ver}_{pk}(m, \sigma) = \mathrm{accept}$

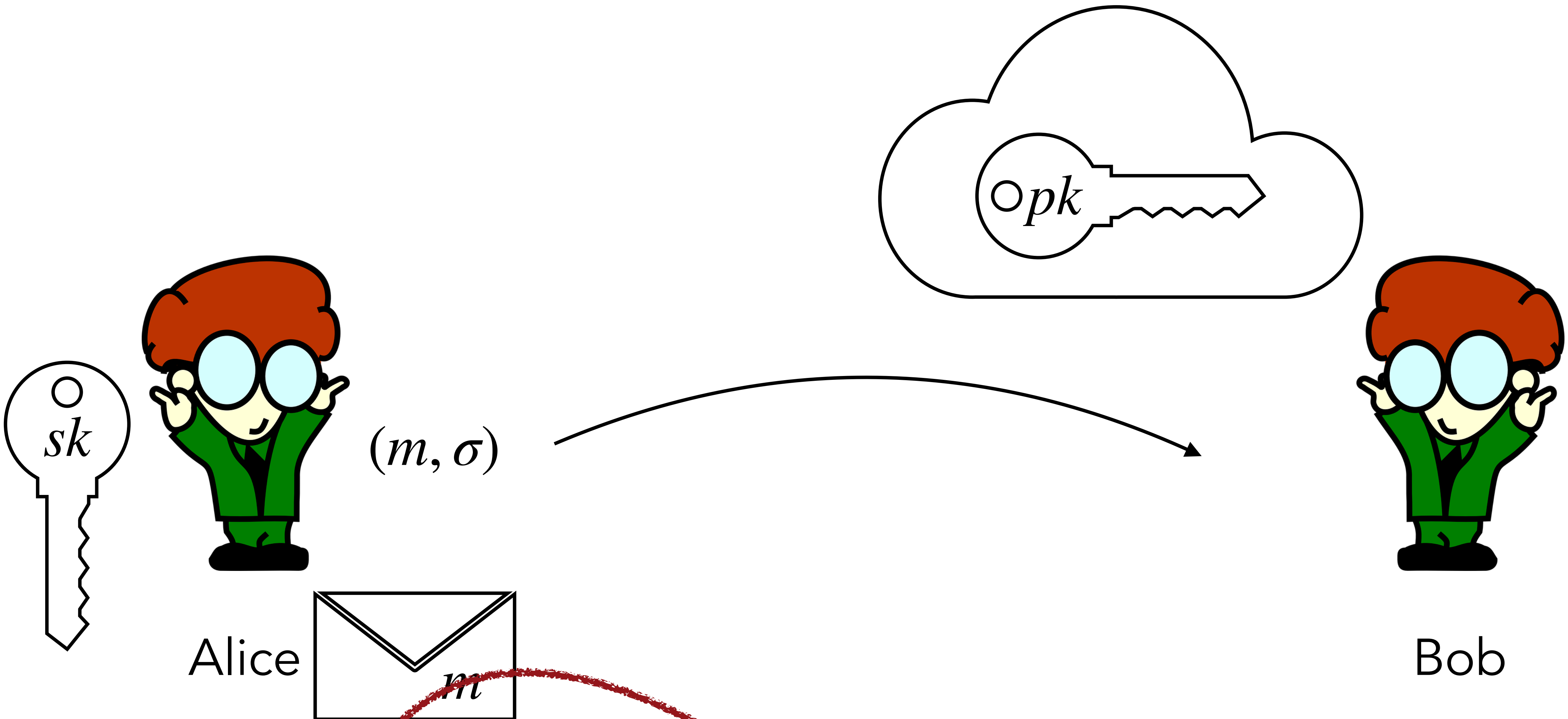
Example application: Hash-and-sign



$\sigma = \mathrm{Sign}_{sk}(H(m))$

- $\mathrm{Ver}_{pk}(m, \sigma) = \text{accept}$

Example application: Hash-and-sign



$\sigma = \mathrm{Sign}_{sk}(H(m))$

collision resistance suffices

- $\mathrm{Ver}_{pk}(m, \sigma) = \mathrm{accept}$

(Quantum) random oracle model

Model hash function as random function $H : \{0,1\}^* \to \{0,1\}^n$

(Quantum) random oracle model

Model hash function as random function $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Public oracle acess to H

(Quantum) random oracle model

Model hash function as random function $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Public oracle acess to H

Post-quantum security: need to allow quantum oracle access

(Quantum) random oracle model

Model hash function as random function $H : \{0,1\}^* \to \{0,1\}^n$

Public oracle acess to H

Post-quantum security: need to allow quantum oracle access

$\Longrightarrow$ Quantum random oracle model (QROM), Boneh et al.

(Quantum) random oracle model

Model hash function as random function $H: \{0,1\}^* \to \{0,1\}^n$

Public oracle acess to H

Post-quantum security: need to allow quantum oracle access

$\Longrightarrow$Quantum random oracle model (QROM), Boneh et al.

Allows public oracle access to $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$

(Quantum) random oracle model

Model hash function as random function $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Public oracle acess to H

Post-quantum security: need to allow quantum oracle access

$\Longrightarrow$ Quantum random oracle model (QROM), Boneh et al.

Allows public oracle access to $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$

+ Has enabled security proofs for more efficient cryptographic schemes
- It's not the real world!

Domain extension

We want: $H : \{0,1\}^* \to \{0,1\}^n$

Domain extension

We want: $H: \{0,1\}^* \to \{0,1\}^n$

Infinite set!

Domain extension

We want: $H: \{0,1\}^* \to \{0,1\}^n$

Infinite set!

Easier: $f: \{0,1\}^k \to \{0,1\}^\ell$

But with the same security properties

Domain extension

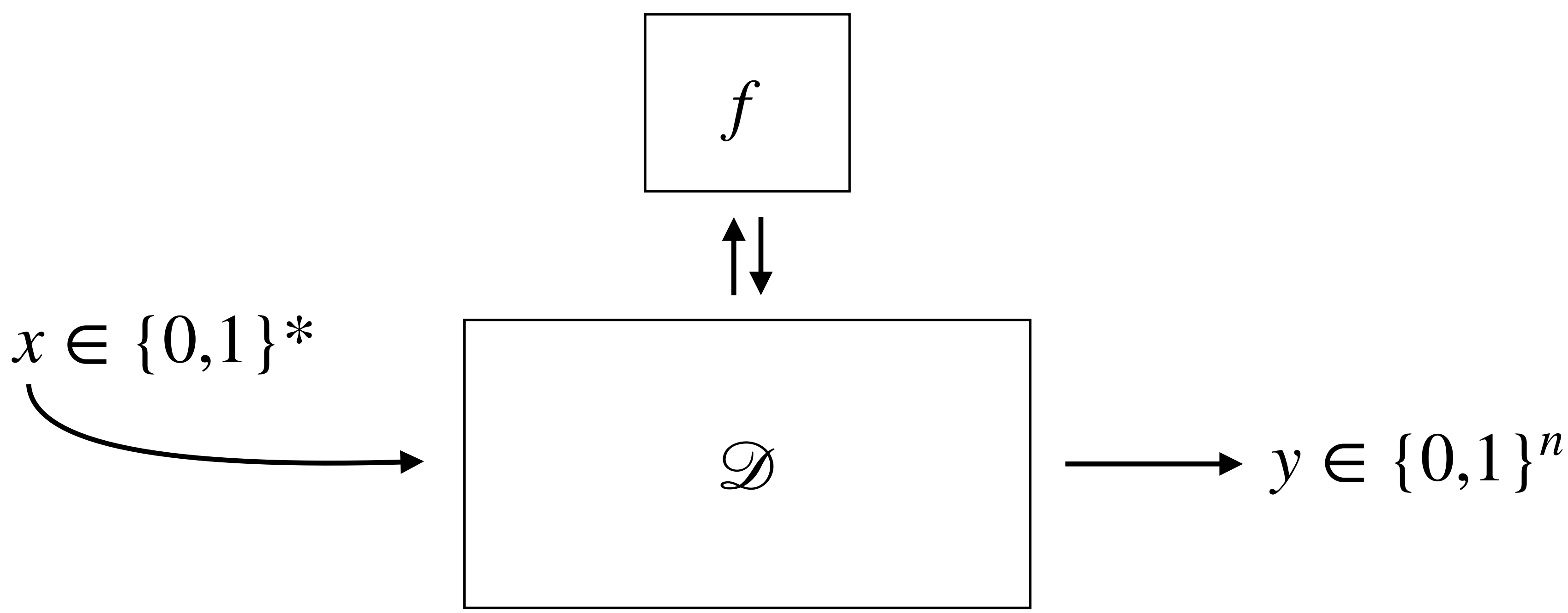
We want: $H: \{0,1\}^* \to \{0,1\}^n$

Infinite set!

Easier: $f: \{0,1\}^k \to \{0,1\}^\ell$

But with the same security properties

Domain extension scheme $\mathscr{D}$: compute $y = H(x)$ by



Such that H inherits the security properties of f

Domain extension

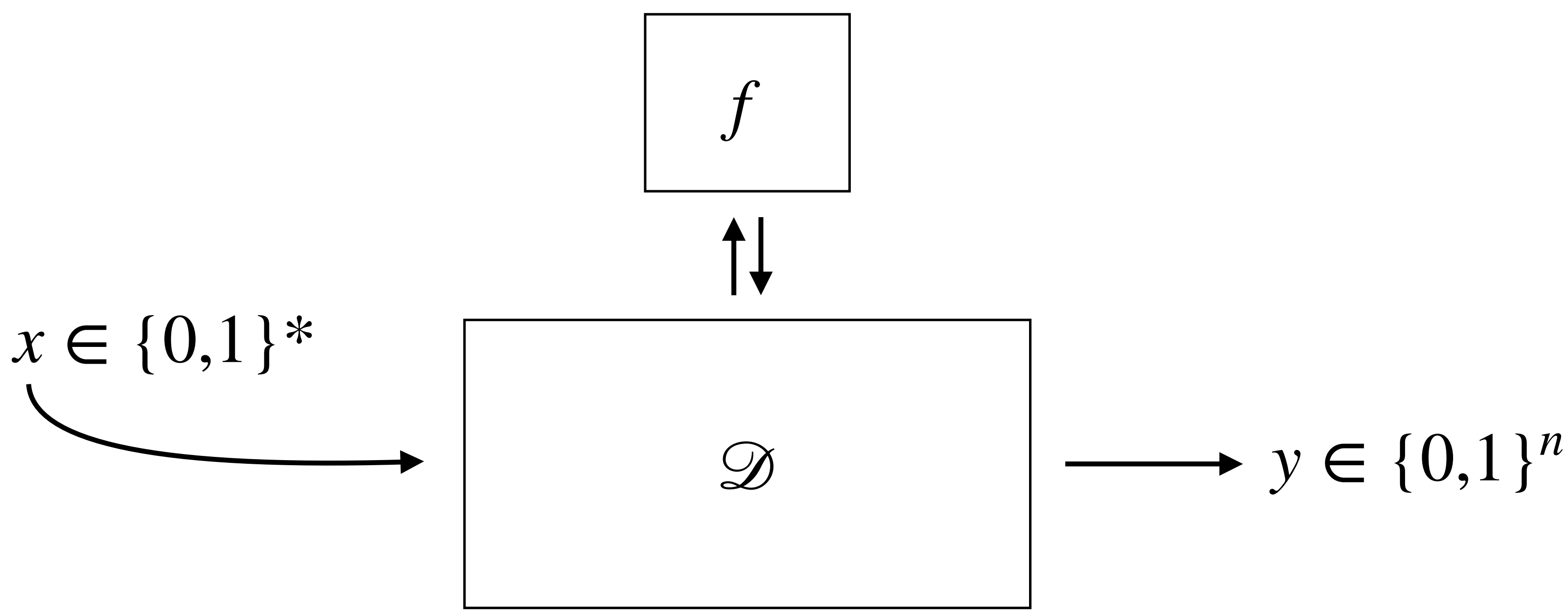
We want: $H : \{0,1\}^* \rightarrow \{0,1\}^n$

Infinite set!

Easier: $f : \{0,1\}^k \rightarrow \{0,1\}^\ell$

But with the same security properties

Domain extension scheme $\mathscr{D}$: compute $y = H(x)$ by



Such that H inherits the security properties of f

SHA-1 SHA-2, SHA-3 work like this.

Example: the sponge construction

A particular domain extension scheme used e.g. in SHA-3

Example: the sponge construction

A particular domain extension scheme used e.g. in SHA-3

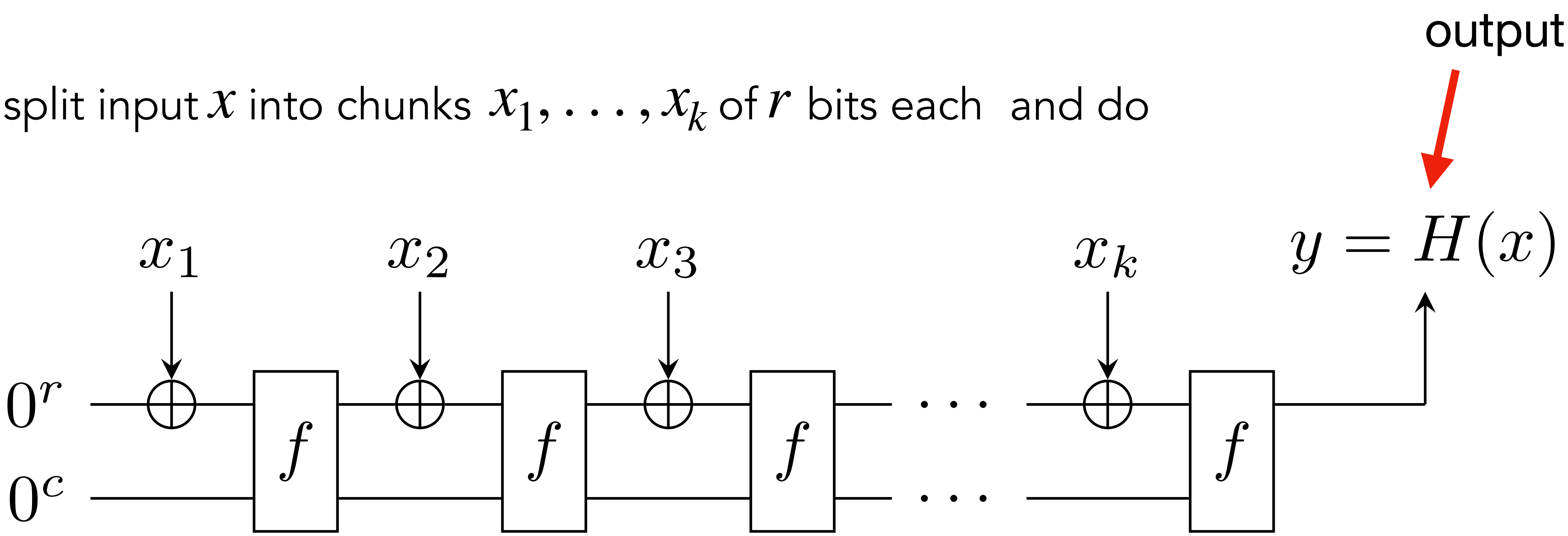
H: split input x into chunks $x_1, \ldots, x_k$ of r bits each

In SHA3-512:
$r = 576$

Example: the sponge construction

A particular domain extension scheme used e.g. in SHA-3

H: split input x into chunks $x_1, \ldots, x_k$ of r bits each and do



In SHA3-512:

$r = 576$

$c = 1024$

Hash functions in the NIST competition

Hash functions in the NIST competition

Digital signature schemes:

Hash functions in the NIST competition

Digital signature schemes:

- All: “hash and sign”.

Hash functions in the NIST competition

Digital signature schemes:

- All: “hash and sign”.
- Some: Fiat-Shamir Transformation

Hash functions in the NIST competition

Digital signature schemes:

- All: “hash and sign”.
- Some: Fiat-Shamir Transformation
- Some: hash-based

Hash functions in the NIST competition

Digital signature schemes:

- All: “hash and sign”.
- Some: Fiat-Shamir Transformation
- Some: hash-based

Key encapsulation schemes

Hash functions in the NIST competition

Digital signature schemes:

- All: “hash and sign”.
- Some: Fiat-Shamir Transformation
- Some: hash-based

Key encapsulation schemes

- All: hash-based key derivation

Hash functions in the NIST competition

Digital signature schemes:

- All: “hash and sign”.
- Some: Fiat-Shamir Transformation
- Some: hash-based

Key encapsulation schemes

- All: hash-based key derivation
- Some: Fujisaki-Okamoto Transformation

Points of attack

How to attack hash functions?

Fixed-length hash function

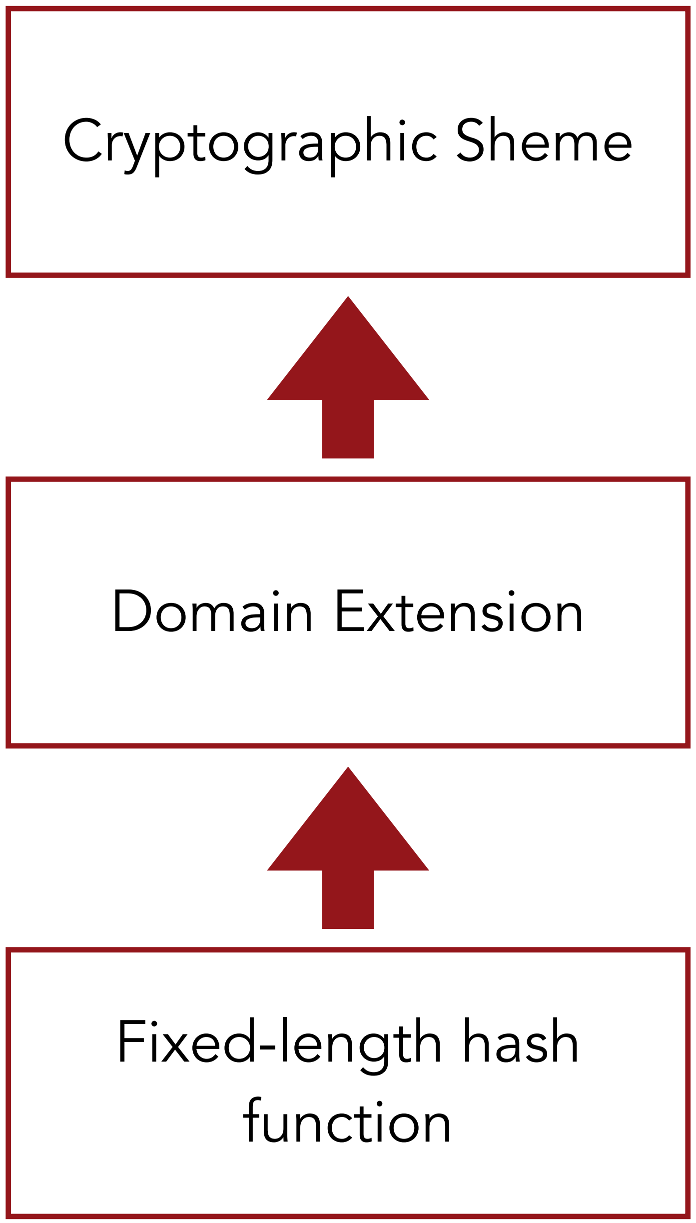
How to attack hash functions?

Domain Extension

↑

Fixed-length hash function

How to attack hash functions?



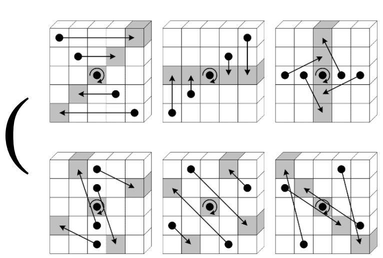
How to attack hash functions?

Cryptographic Sheme

Domain Extension

Fixed-length hash function

Attack using the structure of the fixed length hash function

“$H(x) = \mathscr{D}($  $)$“

How to attack hash functions?

Cryptographic Sheme

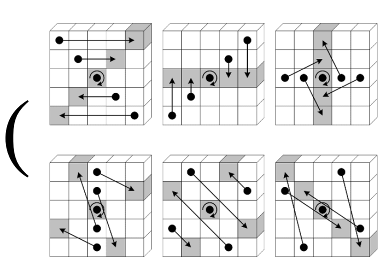
Domain Extension

Attack the domain extension scheme

“$H = \mathscr{D}(f)$“

Fixed-length hash function

Attack using the structure of the fixed length hash function

“$H(x) = \mathscr{D}($  $)$“

How to attack hash functions?

Cryptographic Sheme

Attack cryptographic scheme via its use of H

“$H = H$“

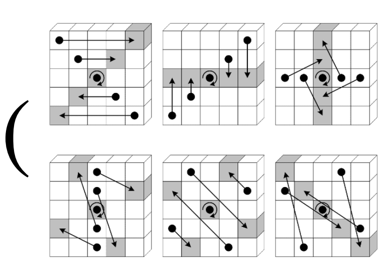
Domain Extension

Attack the domain extension scheme

“$H = \mathscr{D}(f)$“

Fixed-length hash function

Attack using the structure of the fixed length hash function

“$H(x) = \mathscr{D}($  $)$“

How to attack hash functions?

Attack using the structure of the fixed length hash function

How to attack hash functions?

Attack using the structure of the fixed length hash function

Finding Hash Collisions with Quantum Computers by Using Differential Trails with Smaller Probability than Birthday Bound

Akinori Hosoyamada[1,2] and Yu Sasaki[1]

[1] NTT Secure Platform Laboratories, Tokyo, Japan,
{akinori.hosoyamada.bh,yu.sasaki.sk}@hco.ntt.co.jp
[2] Nagoya University, Nagoya, Japan, hosoyamada.akinori@nagoya-u.jp

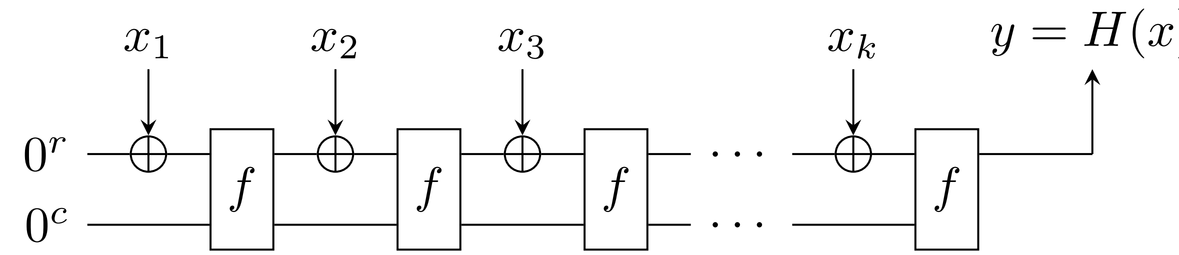
Hosoyamada, A. And Sasaki, Y. “Finding Hash Collisions with Quantum Computers by Using Differential Trails with Smaller Probability than Birthday Bound”, EUROCRYPT 2020

How to attack hash functions?

Attack the domain extension scheme

How to attack hash functions?

Attack the domain extension scheme

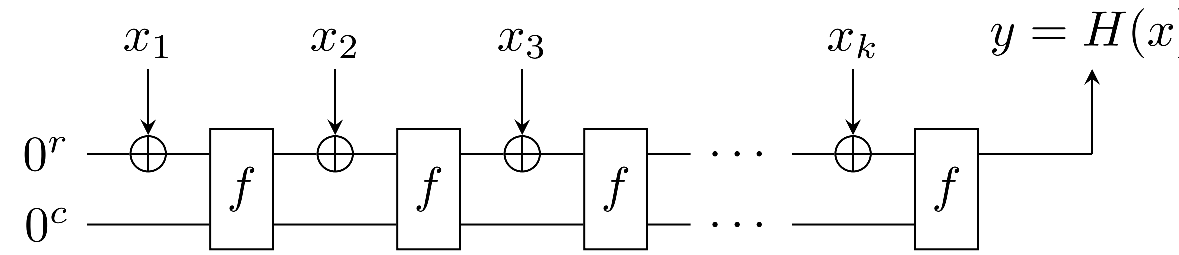


Theorem 16. *Let $\mathbf{S}_{c,r,\mathbf{f},pad,n}(m)$ be a sponge construction with arbitrary block function $\mathbf{f}$. There exists a quantum algorithm* COLL-RO *making at most $q_{\mathbf{f}}$ quantum queries to $\mathbf{f}$ and $q_{\mathcal{H}}$ quantum queries to a random oracle $\mathcal{H}$.* COLL-RO *outputs colliding messages $m \neq \hat{m}$ such that $\mathbf{S}_{c,r,\mathbf{f},pad,n}(m) = \mathbf{S}_{c,r,\mathbf{f},pad,n}(\hat{m})$ with probability at least $1/8$, where $q_{\mathbf{f}} := 2k_{\mathrm{Amb}} \cdot \min\{\frac{c+6+2r}{r} 2^{c/3}, \frac{2n+6+3r}{r} 2^{n/3}\}$, and $q_{\mathcal{H}} := 2k_{\mathrm{Amb}} \cdot \min\{2^{c/3}, 2^{n/3}\} + 2$, where k_{Amb} is the constant from Theorem 14 and pad is any padding function which appends at most $2r$ bits.*

Czajkowski, J., Bruinderink, L. G., Hülsing, A., Schaffner, C., & Unruh, D. “Post-quantum security of the sponge construction”, PQCrypto 2018

How to attack hash functions?

Attack the domain extension scheme



Theorem 16. *Let* $\mathbf{S}_{c,r,\mathbf{f},pad,n}(m)$ *be a sponge construction with arbitrary block function* $\mathbf{f}$*. There exists a quantum algorithm* COLL-RO *making at most* $q_{\mathbf{f}}$ *quantum queries to* $\mathbf{f}$ *and* $q_{\mathcal{H}}$ *quantum queries to a random oracle* $\mathcal{H}$*.* COLL-RO *outputs colliding messages* $m \neq \hat{m}$ *such that* $\mathbf{S}_{c,r,\mathbf{f},pad,n}(m) = \mathbf{S}_{c,r,\mathbf{f},pad,n}(\hat{m})$ *with probability at least* $1/8$*, where* $q_{\mathbf{f}} := 2k_{\mathrm{Amb}} \cdot \min\{\frac{c+6+2r}{r} 2^{c/3}, \frac{2n+6+3r}{r} 2^{n/3}\}$*, and* $q_{\mathcal{H}} := 2k_{\mathrm{Amb}} \cdot \min\{2^{c/3}, 2^{n/3}\} + 2$*, where* k_{Amb} *is the constant from Theorem 14 and pad is any padding function which appends at most* $2r$ *bits.*

Finds collision for sponge
by finding collision of f

Czajkowski, J., Bruinderink, L. G., Hülsing, A., Schaffner, C., & Unruh, D. "Post-quantum security of the sponge construction", PQCrypto 2018

How to attack hash functions?

Attack cryptographic scheme
via its use of H

How to attack hash functions?

Attack cryptographic scheme
via its use of H

Remainder of this talk:
2 Examples

Fiat-Shamir and Fujisaki-Okamoto

Sigma-protocols

Sigma-protocols

Prover

Sigma-protocols

Prover

Verifier

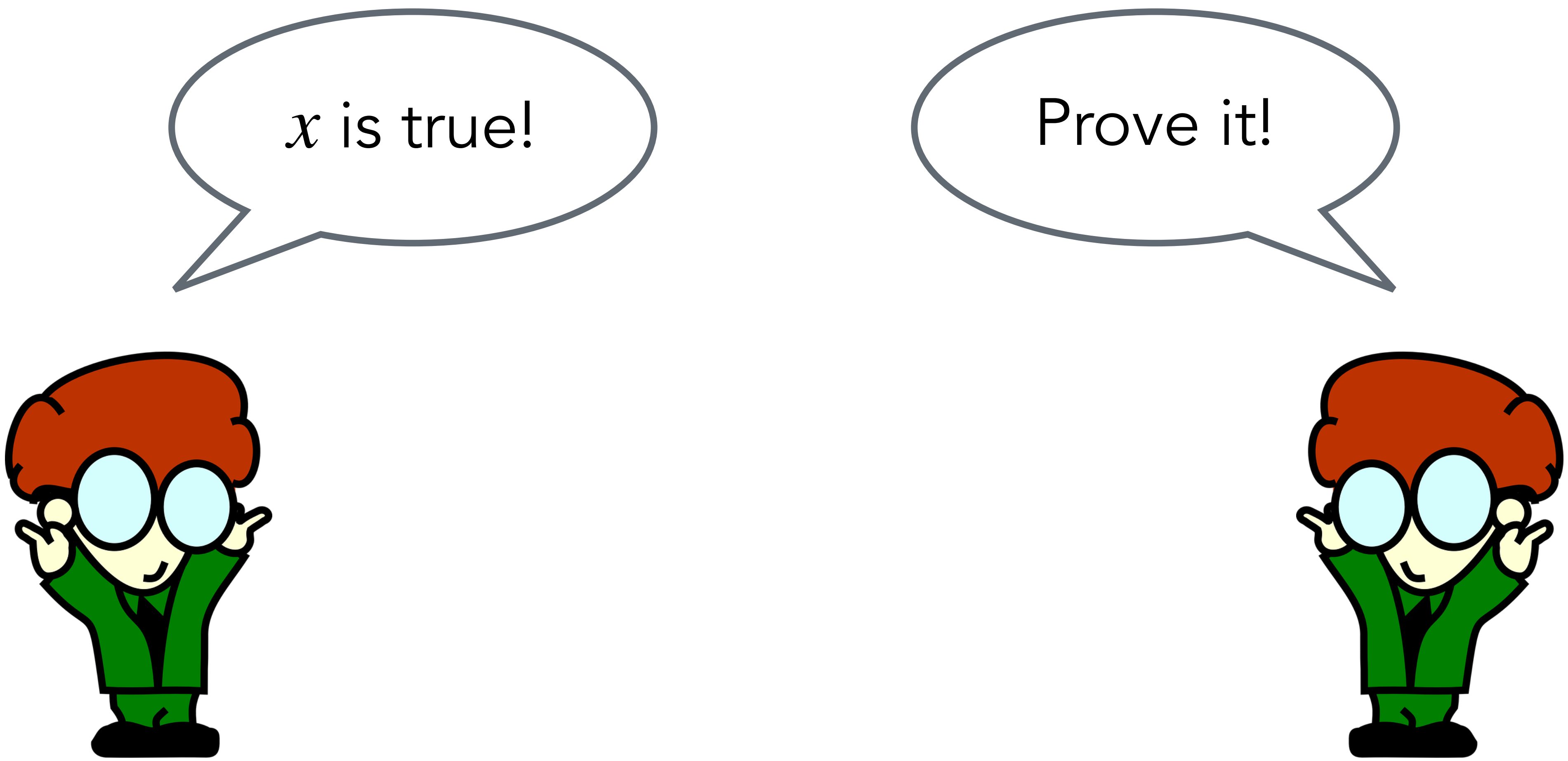
Sigma-protocols

x is true!

Prover

Verifier

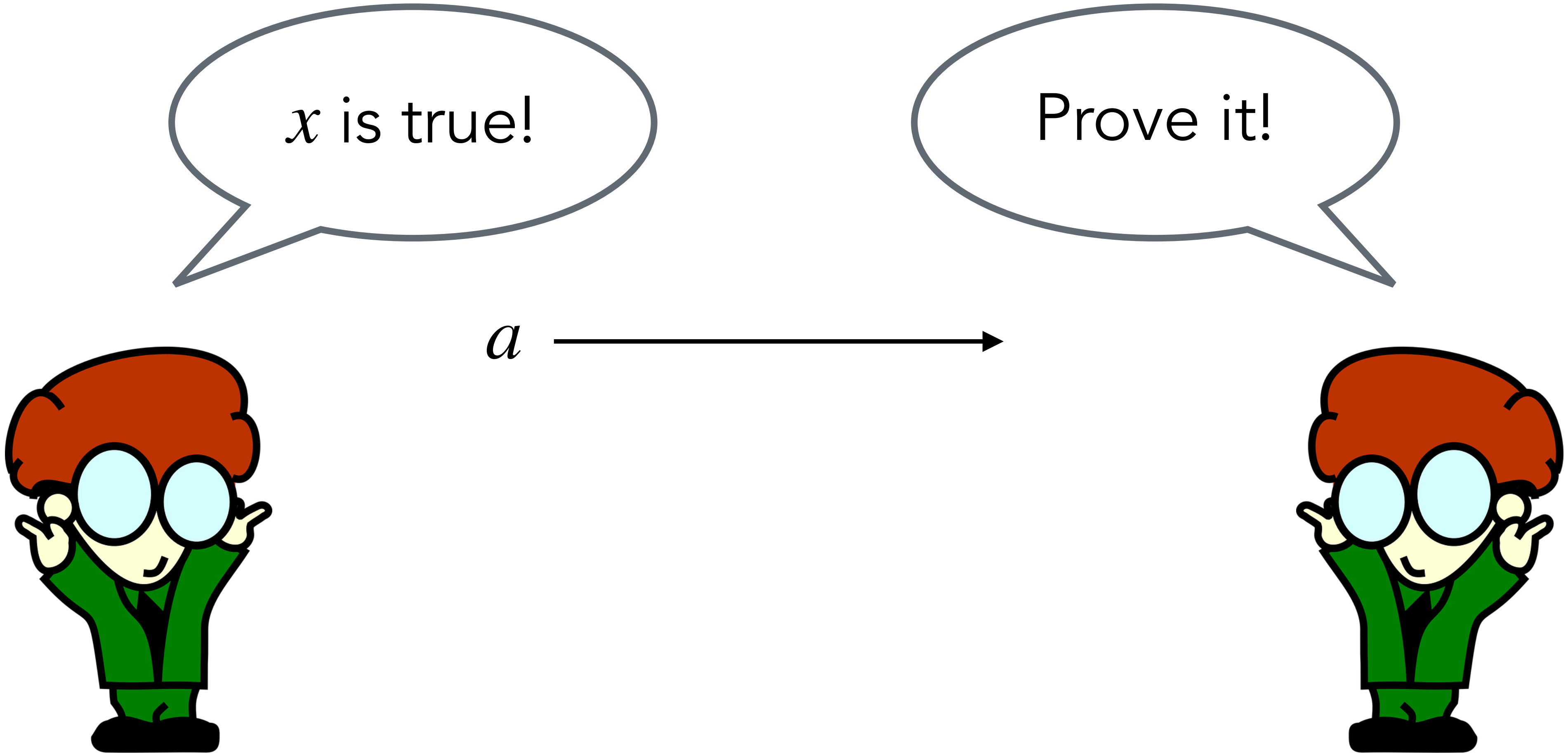
Sigma-protocols



Prover

Verifier

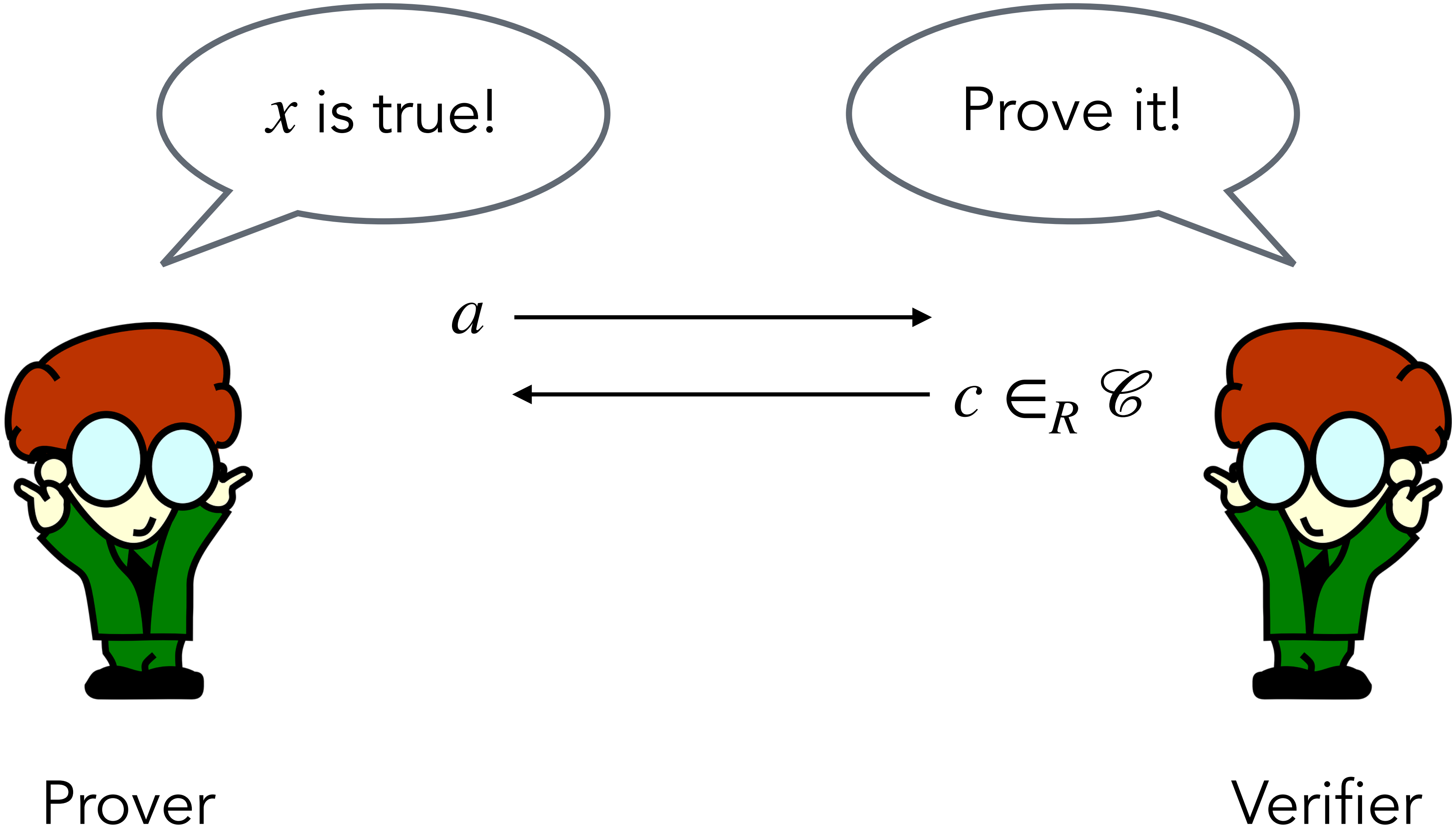
Sigma-protocols



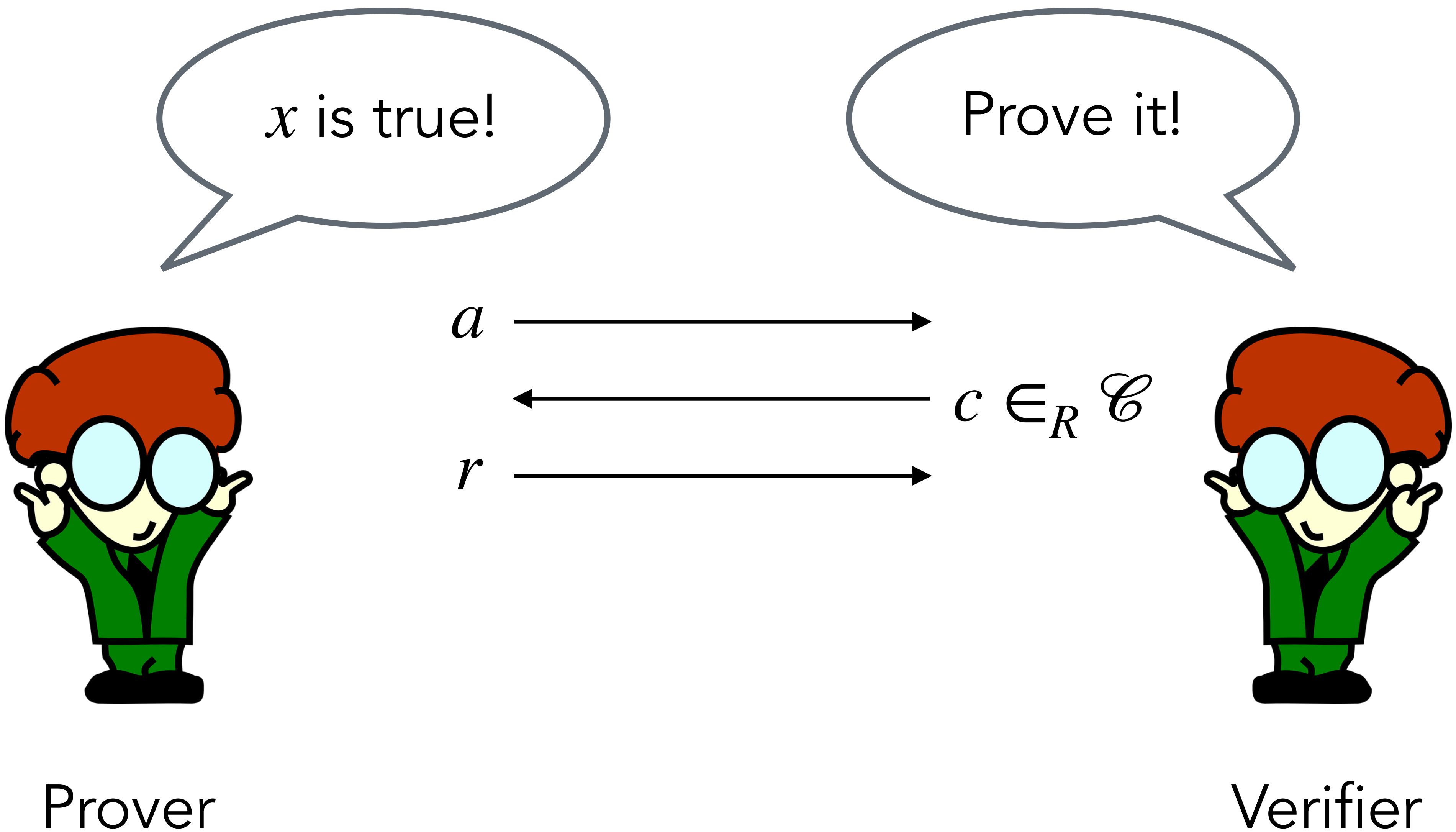
Prover

Verifier

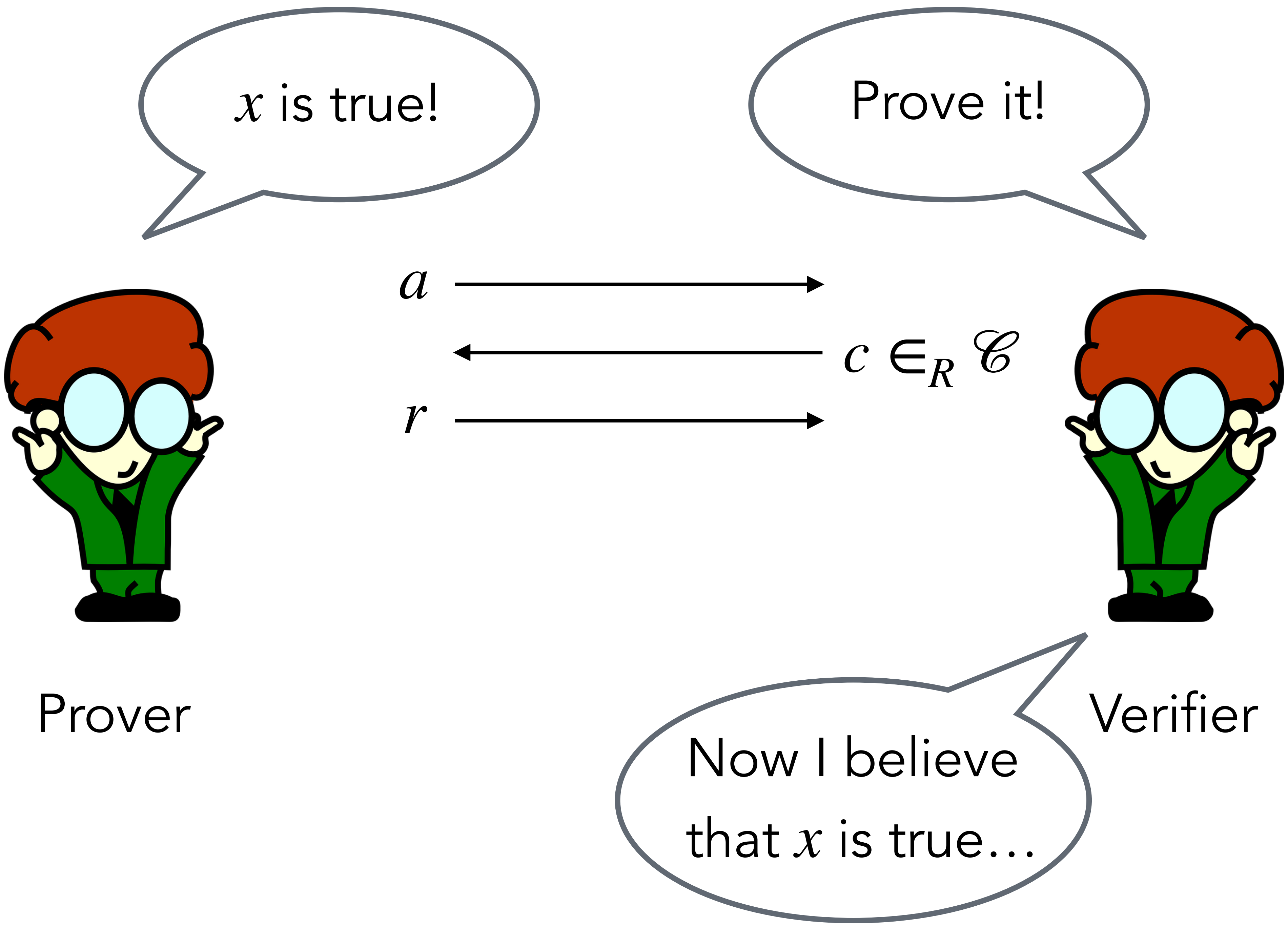
Sigma-protocols



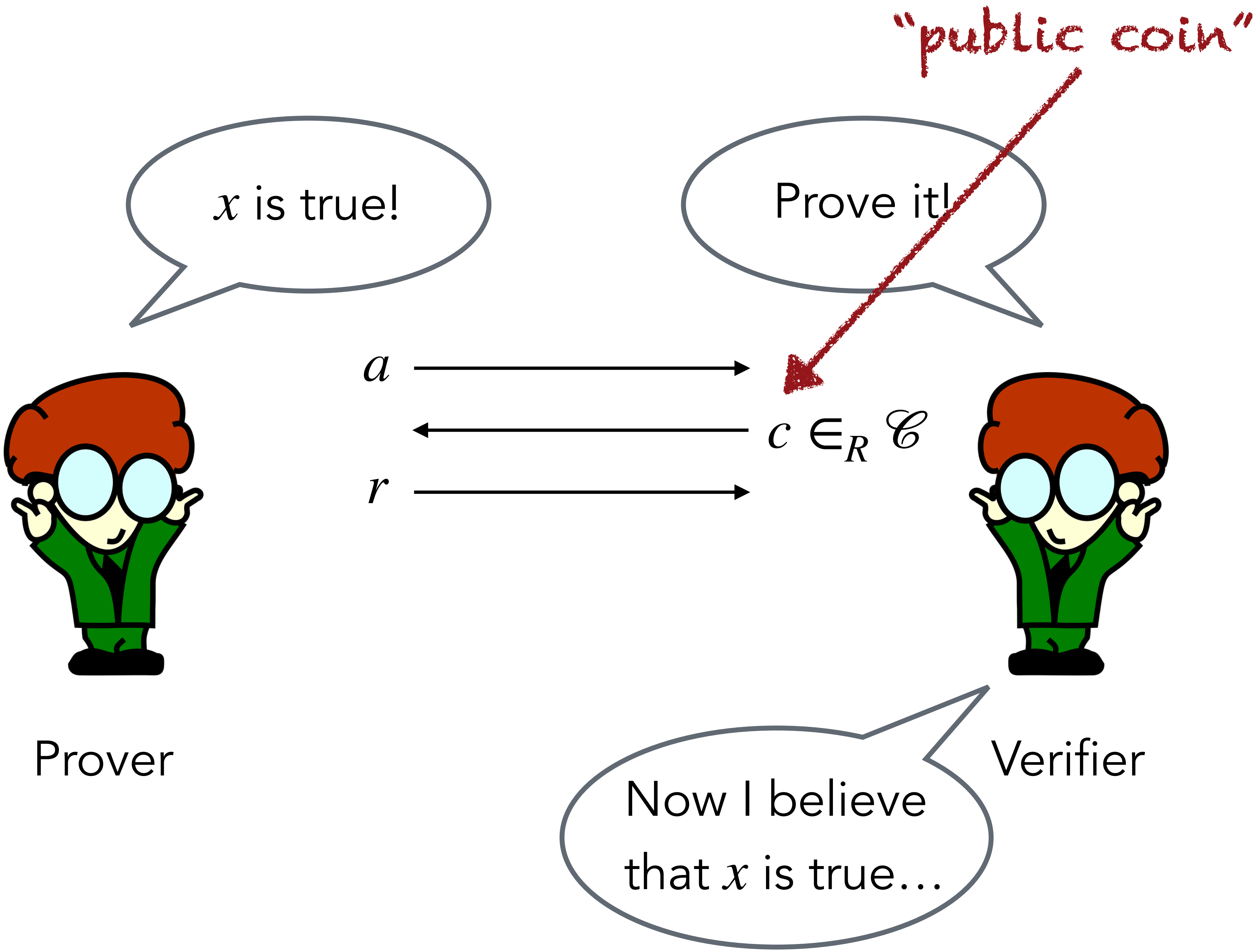
Sigma-protocols



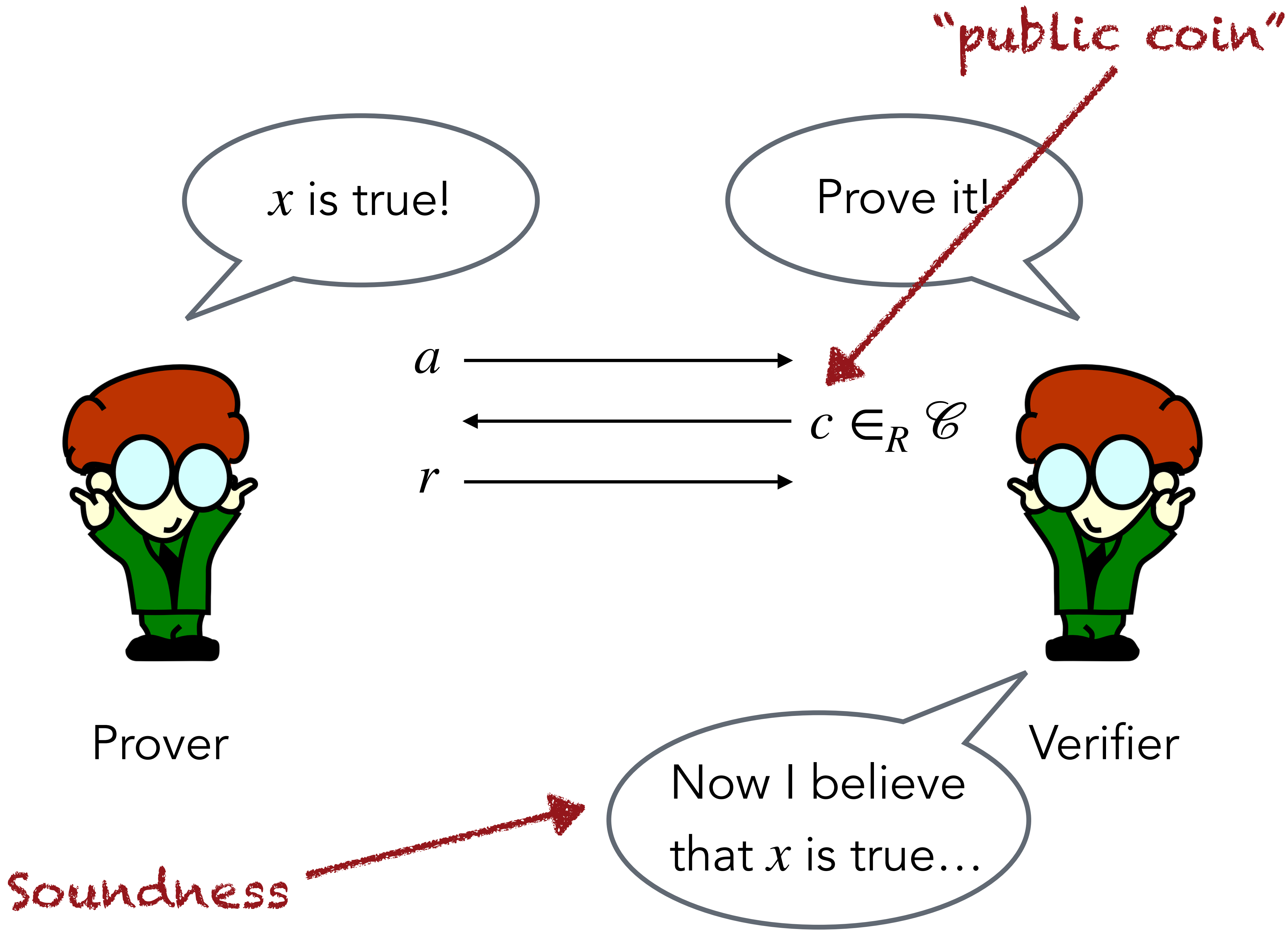
Sigma-protocols



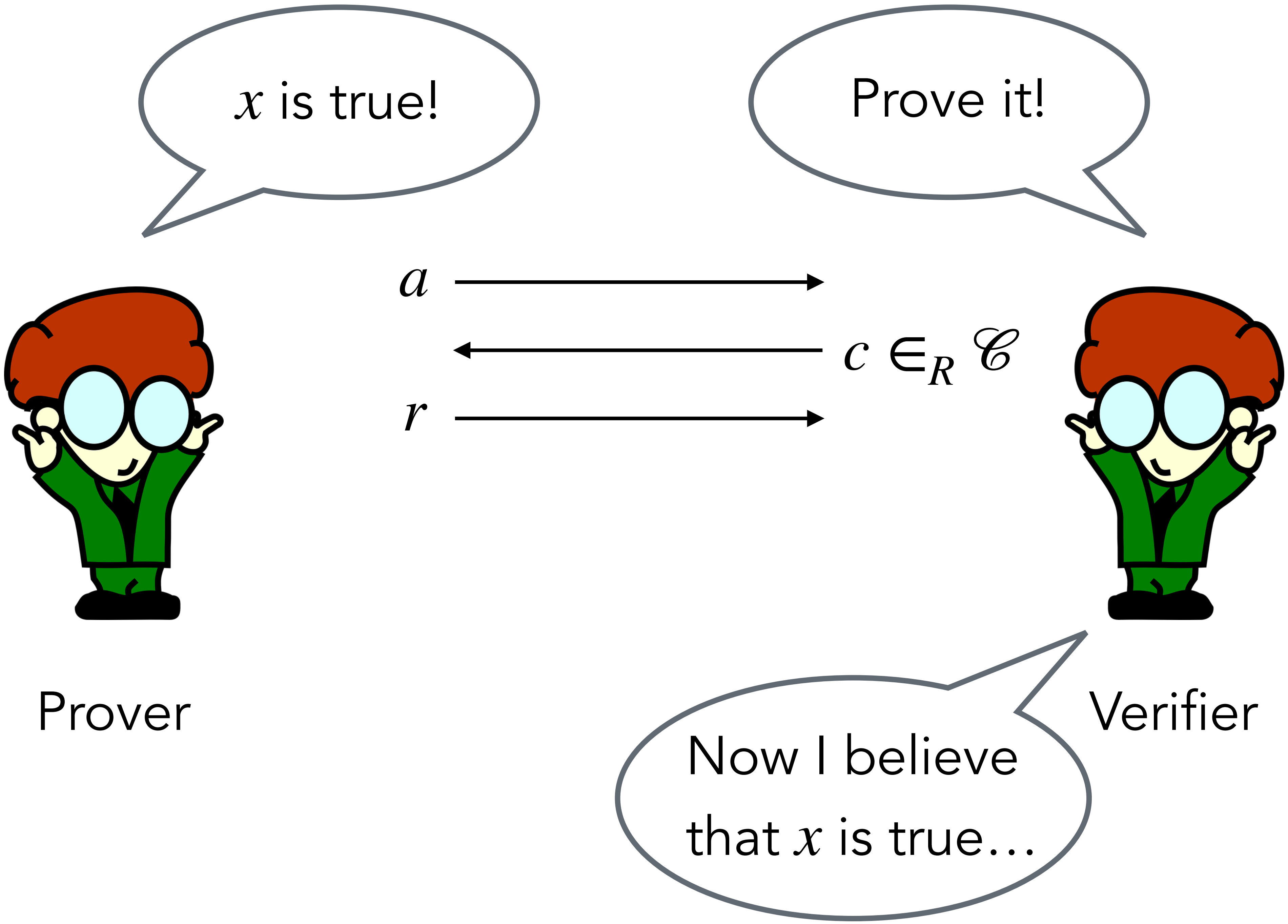
Sigma-protocols



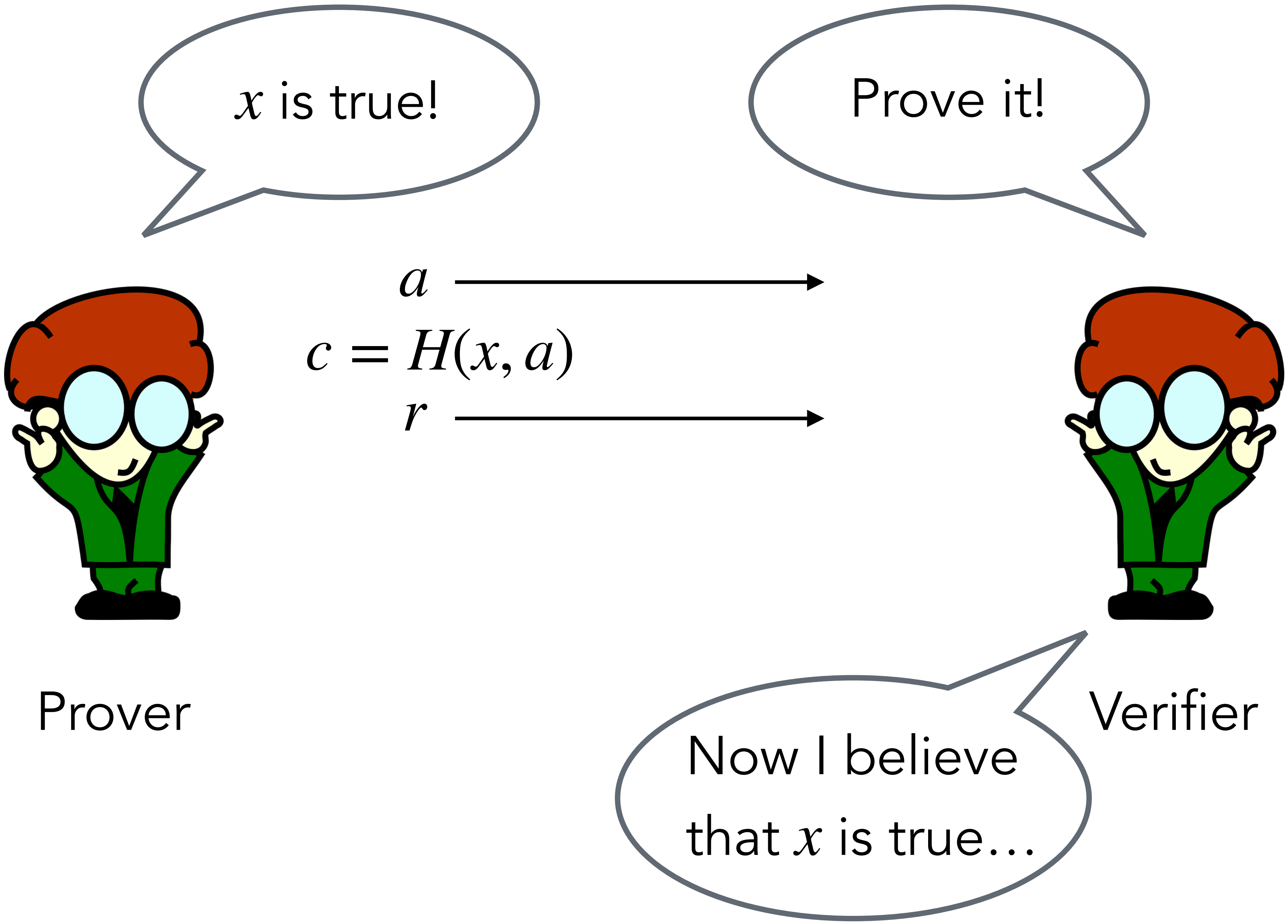
Sigma-protocols



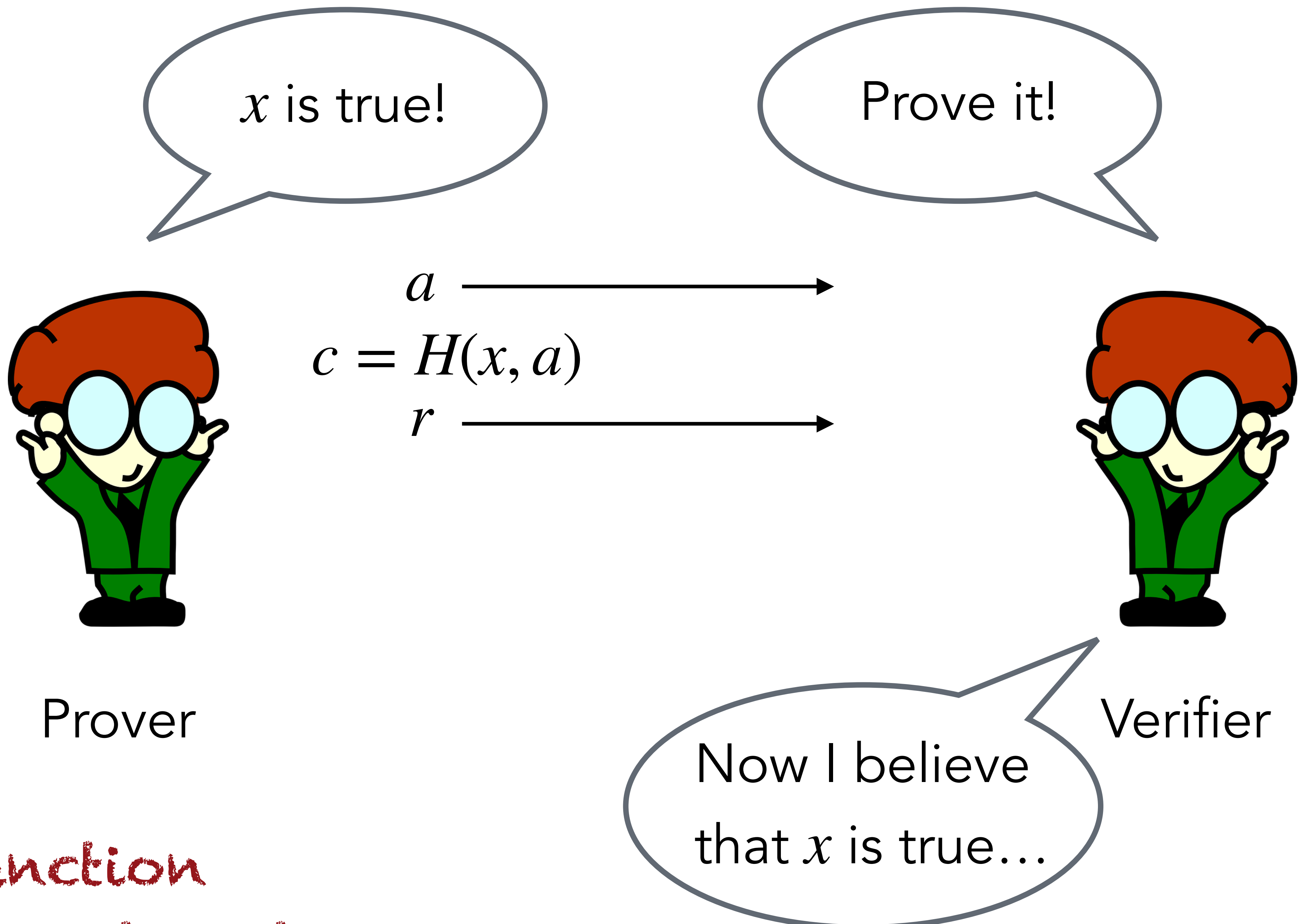
Fiat-Shamir transformation



Fiat-Shamir transformation

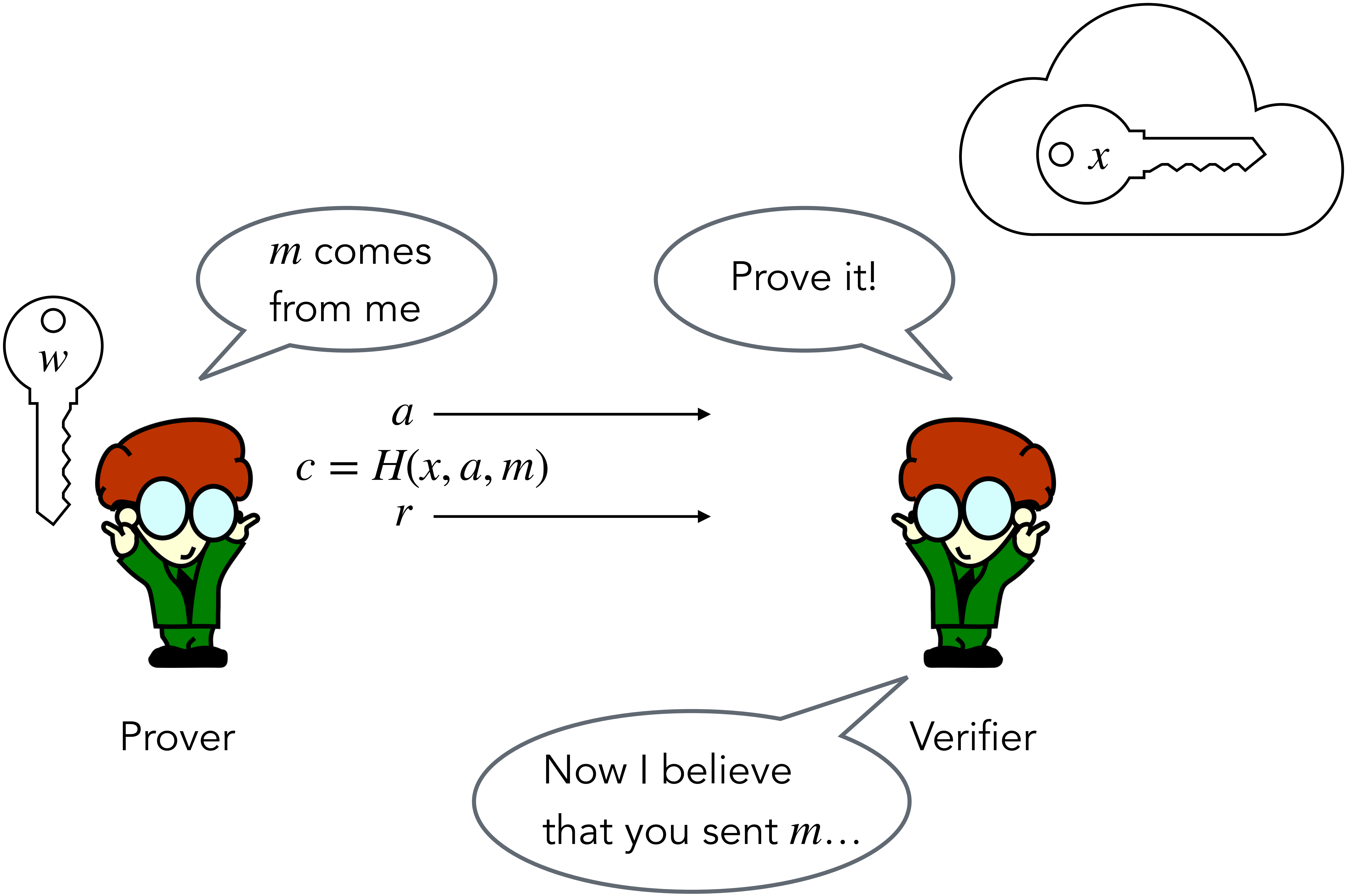


Fiat-Shamir transformation



Hash function
replaces interaction

Fiat-Shamir signature scheme



Fujisaki-Okamoto transformation

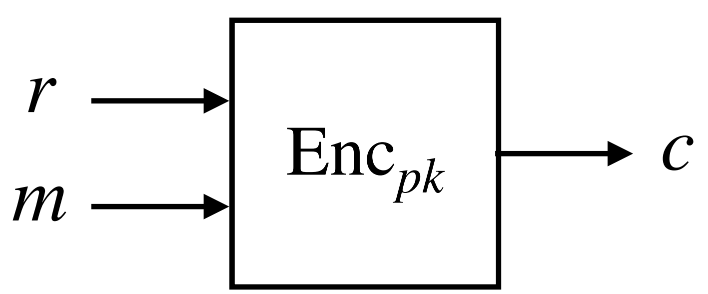
Upgrades weak security to chosen-ciphertext security for key encapsulation

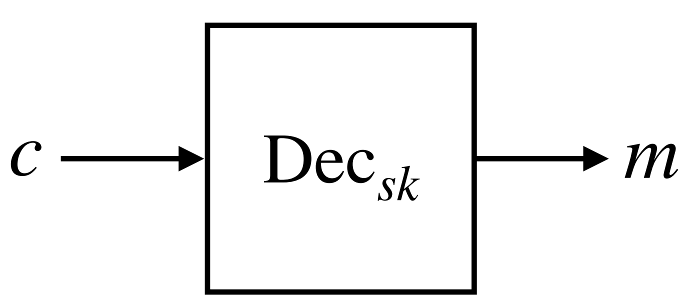
“Derandomize, Hash&reincrypt”

Fujisaki-Okamoto transformation

Upgrades weak security to chosen-ciphertext security for key encapsulation

"Derandomize, Hash&reincrypt"



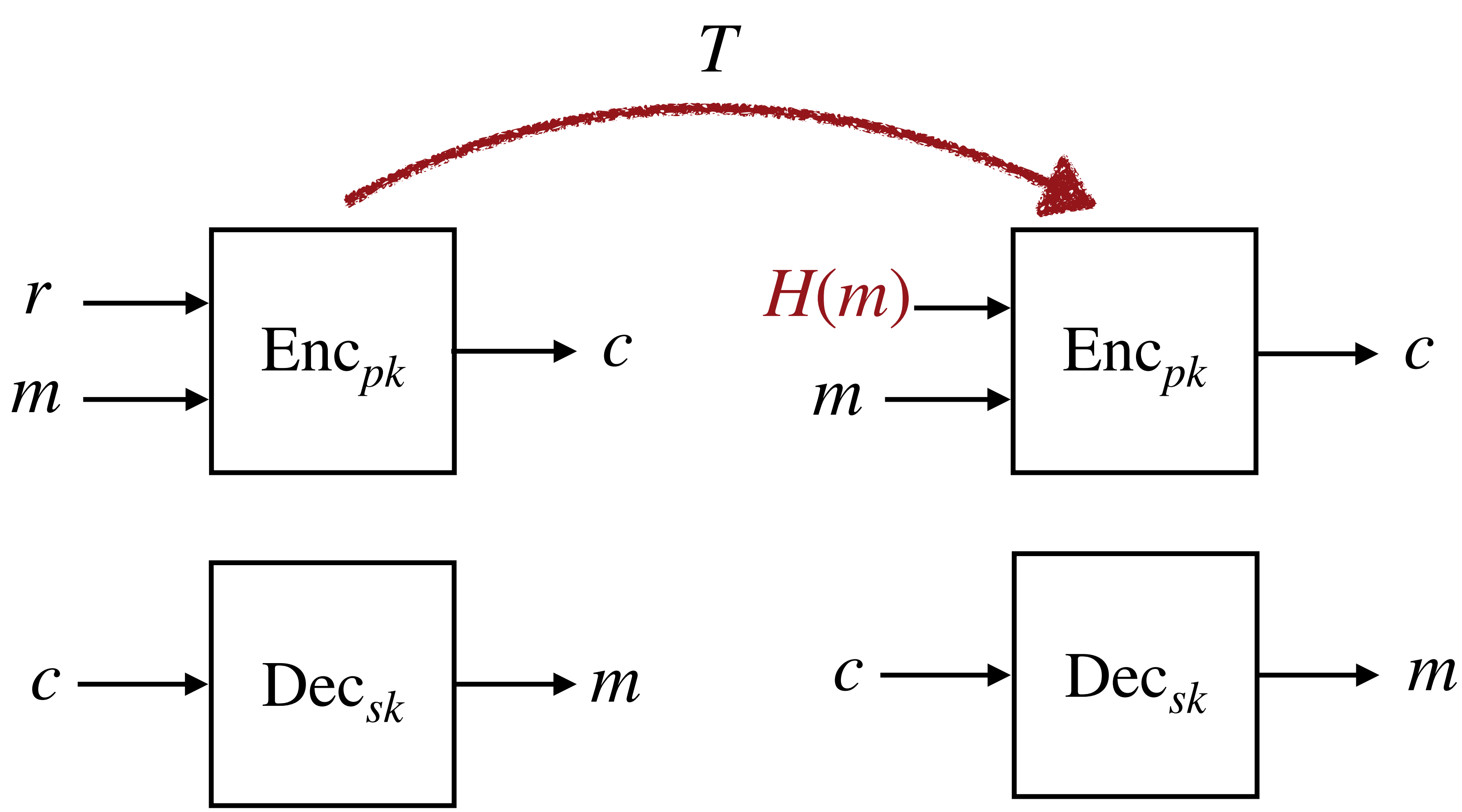


Fujisaki-Okamoto transformation

Upgrades weak security to chosen-ciphertext security for key encapsulation

"Derandomize, Hash&reincrypt"

"Derandomize"

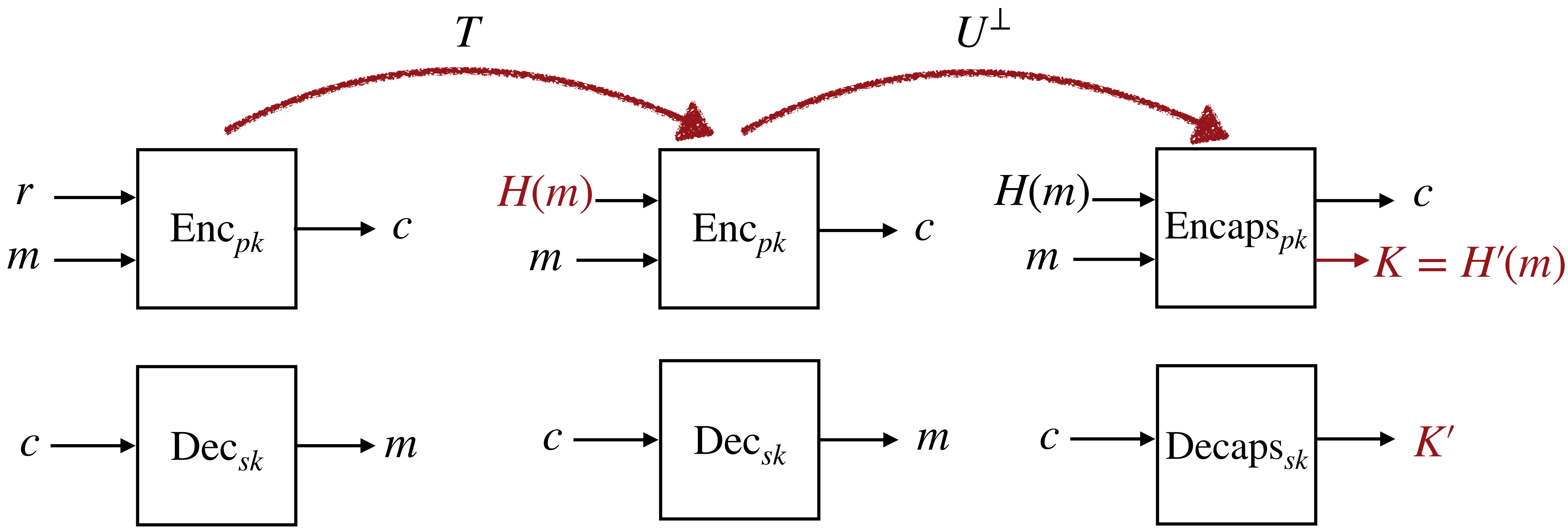


Fujisaki-Okamoto transformation

Upgrades weak security to chosen-ciphertext security for key encapsulation

"Derandomize, Hash&reincrypt"

"Derandomize" "Hash&reincrypt"



$$K' = \begin{cases} H'(m) & c = \mathrm{Enc}_{pk}(m, H(m)) \\ \perp & \text{else} \end{cases}$$

Attacks and attack approaches

Fiat-Shamir transformation in the QROM

Theorem (Don, Fehr, M, Schaffner '19):

An dishonest prover making q quantum queries to the random oracle can prove a wrong statement in the Fiat-Shamir Transformation $\mathsf{FS}(\Sigma)$ of a sigma protocol Σ with probability at most

$$\varepsilon_{\mathsf{FS}(\Sigma)}(q) \leq (2q+1)^2 \varepsilon_\Sigma,$$

Where ε_Σ is the soundness error of Σ.

Don, J., Fehr, S., Majenz, C., & Schaffner, C., "Security of the Fiat-Shamir transformation in the quantum random-oracle model", Crypto 2019

Fiat-Shamir transformation in the QROM

Theorem (Don, Fehr, M, Schaffner ’19):

An dishonest prover making q quantum queries to the random oracle can prove a wrong statement in the Fiat-Shamir Transformation FS(Σ) of a sigma protocol Σ with probability at most

$$\varepsilon_{\mathsf{FS}(\Sigma)}(q) \leq (2q+1)^2 \varepsilon_\Sigma,$$

Where ε_Σ is the soundness error of Σ.

(Independent work: Liu&Zhandry, Crypto 2019, less tight…)

Don, J., Fehr, S., Majenz, C., & Schaffner, C., “Security of the Fiat-Shamir transformation in the quantum random-oracle model”, Crypto 2019

Liu, Q. and Zhandry, M., “Revisiting Post-Quantum Fiat-Shamir”, Crypto 2019

Fiat-Shamir transformation in the QROM

Theorem (Don, Fehr, M, Schaffner '19):

An dishonest prover making q quantum queries to the random oracle can prove a wrong statement in the Fiat-Shamir Transformation FS(Σ) of a sigma protocol Σ with probability at most

$$\varepsilon_{\mathsf{FS}(\Sigma)}(q) \leq (2q+1)^2 \varepsilon_\Sigma,$$

Where ε_Σ is the soundness error of Σ.

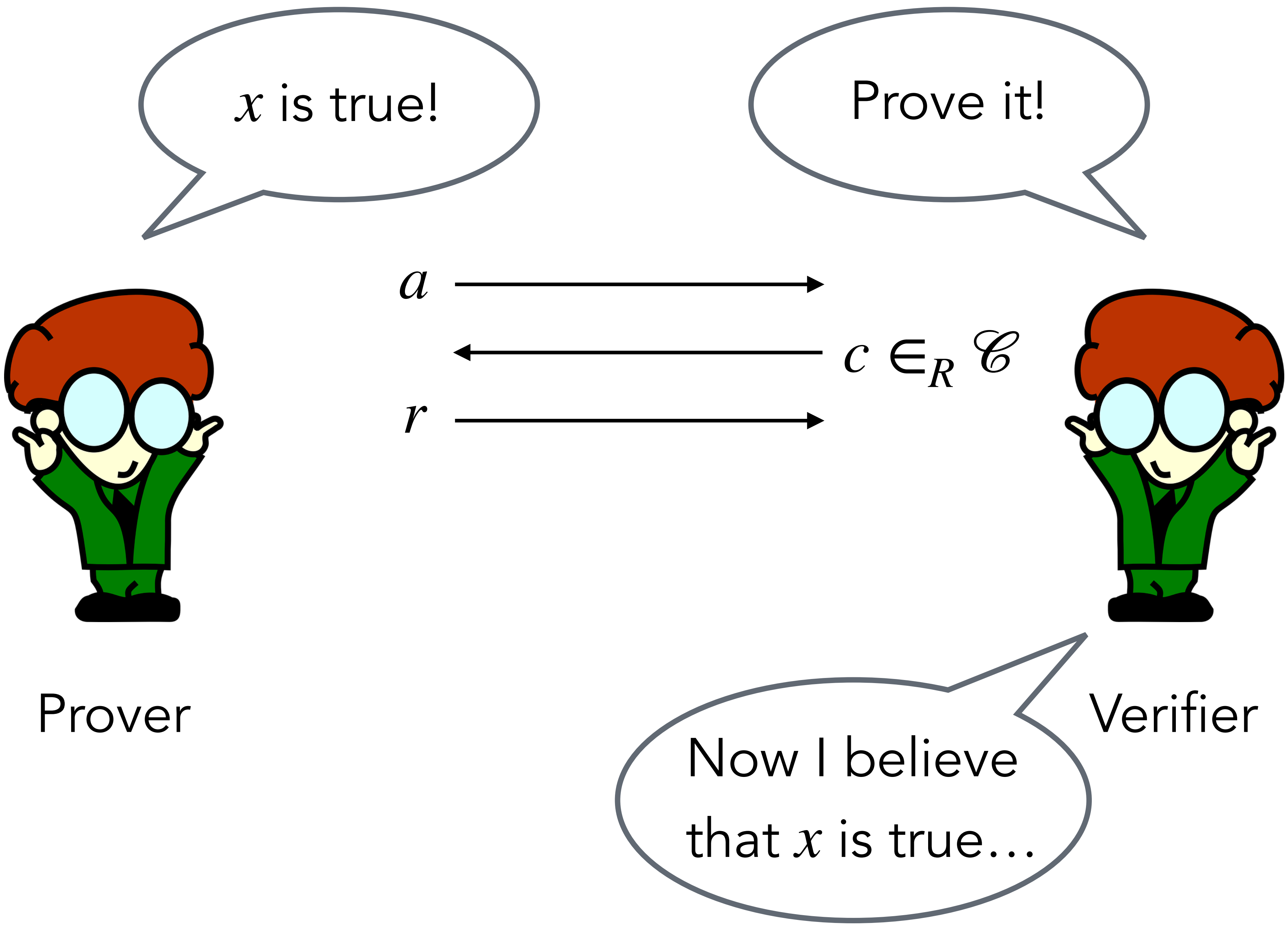
Can we find a matching attack?

(Independent work: Liu&Zhandry, Crypto 2019, less tight…)

Don, J., Fehr, S., Majenz, C., & Schaffner, C., "Security of the Fiat-Shamir transformation in the quantum random-oracle model", Crypto 2019

Liu, Q. and Zhandry, M., "Revisiting Post-Quantum Fiat-Shamir", Crypto 2019

Zero knowledge



Zero knowledge

Verifier learns something from (a, c, r)

Zero knowledge

Verifier learns something from (a, c, r)

At least "x is true"!

Zero knowledge

Verifier learns something from (a, c, r)

At least “x is true”!

Zero knowledge: Verifier learns nothing else.

Zero knowledge

Verifier learns something from (a, c, r)

At least "x is true"!

Zero knowledge: Verifier learns nothing else.

Formally: (a, c, r) can be *simulated*

Zero knowledge

Verifier learns something from (a, c, r)

At least "x is true"!

Zero knowledge: Verifier learns nothing else.

Formally: (a, c, r) can be *simulated*

Definition (Honest-verifier zero knowledge, informal):
A sigma protocol Σ is honest-verifier zero knowledge (HVZK) if there exists a simulator $\mathcal{S}$ such that for all true statements x, $(a, c, r) \leftarrow \mathcal{S}(x)$ is indistinguishable from a transcript from the protocol.

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

$(a, c, r) \leftarrow \mathcal{S}(x)$ such that $H(x, a) = c$ has small $p > 0$

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

$(a, c, r) \leftarrow \mathcal{S}(x)$ such that $H(x, a) = c$ has small $p > 0$

Idea: Grover-search for such a transcript!

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

$(a, c, r) \leftarrow \mathcal{S}(x)$ such that $H(x, a) = c$ has small $p > 0$

Idea: Grover-search for such a transcript!

$\mathcal{S}$ is a public, randomized algorithm, $\mathcal{S}(x; \rho) = (a, c, r)$

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

$(a, c, r) \leftarrow \mathcal{S}(x)$ such that $H(x, a) = c$ has small $p > 0$

Idea: Grover-search for such a transcript!

$\mathcal{S}$ is a public, randomized algorithm, $\mathcal{S}(x; \rho) = (a, c, r)$

$$f_x^H(\rho) = \begin{cases} 1 & \text{if } \mathcal{S}(x;\rho) = (a, c, r) \text{ such that } H(x, a) = c \\ 0 & \text{else} \end{cases}$$

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

$(a, c, r) \leftarrow \mathcal{S}(x)$ such that $H(x, a) = c$ has small $p > 0$

Uses one query to H

Idea: Grover-search for such a transcript!

$\mathcal{S}$ is a public, randomized algorithm, $\mathcal{S}(x; \rho) = (a, c, r)$

$$f_x^H(\rho) = \begin{cases} 1 & \text{if } \mathcal{S}(x;\rho) = (a,c,r) \text{ such that } H(x,a) = c \\ 0 & \text{else} \end{cases}$$

Attack

How can $\mathcal{S}$ even exist for Σ with soundness?

$\mathcal{S}(x)$ can choose (a, c, r) in any order!

$(a, c, r) \leftarrow \mathcal{S}(x)$ such that $H(x, a) = c$ has small $p > 0$

Uses one query to H

Idea: Grover-search for such a transcript!

$\mathcal{S}$ is a public, randomized algorithm, $\mathcal{S}(x; \rho) = (a, c, r)$

$$f_x^H(\rho) = \begin{cases} 1 & \text{if } \mathcal{S}(x;\rho) = (a,c,r) \text{ such that } H(x,a) = c \\ 0 & \text{else} \end{cases}$$

Theorem (informal; Don, Fehr, M '20):

Let Σ be a sigma protocol that is perfectly HVZK and has special soundness + some mild additional properties. Then there exists a quantum polynomial-time attacker making q queries to H that succeeds with probability $\varepsilon_{\mathsf{FS}(\Sigma)}(q) \geq q^2 \varepsilon_\Sigma$.

Attack

Theorem (informal, Don, Fehr, M '20):

Let Σ be a sigma protocol that is perfectly HVZK and has special soundness + some mild additional properties. Then there exists a quantum polynomial-time attacker making q queries to H that succeeds with probability $\varepsilon_{\mathsf{FS}(\Sigma)}(q) \geq q^2 \varepsilon_{\Sigma}$.

How relevant is the attack?

Attack

Theorem (informal, Don, Fehr, M '20):

Let Σ be a sigma protocol that is perfectly HVZK and has special soundness + some mild additional properties. Then there exists a quantum polynomial-time attacker making q queries to H that succeeds with probability $\varepsilon_{\mathsf{FS}(\Sigma)}(q) \geq q^2 \varepsilon_{\Sigma}$.

How relevant is the attack?

Sigma protocols for Fiat-Shamir signatures

- are HVZK
- Have special soundness or similar

The QROM is uninstantiable

Did we figure out Fiat-Shamir?

The QROM is uninstantiable

Did we figure out Fiat-Shamir?

In the QROM: yes.

The QROM is uninstantiable

Did we figure out Fiat-Shamir?

In the QROM: yes.

Theorem (Canetti, Goldreich, Halevi '98):

There exists a digital signature scheme Π^H using a hash function H, such that

i) Π^H is secure in the ROM

ii) Π^H is insecure for any efficient H

Canetti, R., Goldreich, O., Halevi, S., "The random oracle methodology, revisited", STOC 1998

The QROM is uninstantiable

Did we figure out Fiat-Shamir?

In the QROM: yes.

Theorem (Canetti, Goldreich, Halevi '98):

There exists a digital signature scheme Π^H using a hash function H, such that

i) Π^H is secure in the ROM

ii) Π^H is insecure for any efficient H

Theorem (Eaton and Song '19):

The digital signature scheme Π^H from above is secure in the QROM.

Canetti, R., Goldreich, O., Halevi, S., "The random oracle methodology, revisited", STOC 1998

Eaton, E. and Song, F., "A Note on the Instantiability of the Quantum Random Oracle", eprint 2019

The QROM is uninstantiable

Did we figure out Fiat-Shamir?

In the QROM: yes.

Theorem (Canetti, Goldreich, Halevi '98):
There exists a digital signature scheme Π^H using a hash function H, such that

i) Π^H is secure in the ROM

ii) Π^H is insecure for any efficient H

Theorem (Eaton and Song '19):
The digital signature scheme Π^H from above is secure in the QROM.

Better attacks possible, but likely using structure of H.

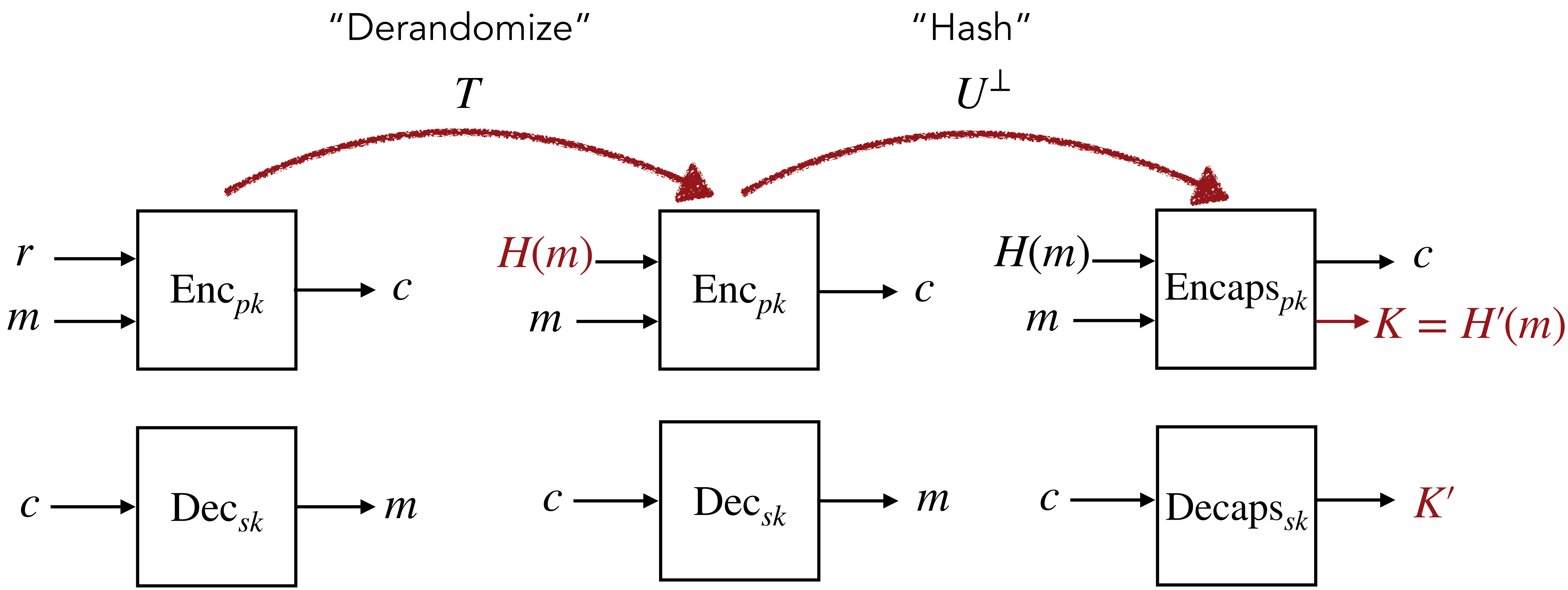
Canetti, R., Goldreich, O., Halevi, S., "The random oracle methodology, revisited", STOC 1998

Eaton, E. and Song, F., "A Note on the Instantiability of the Quantum Random Oracle", eprint 2019

Fujisaki-Okamoto transformation

Upgrades weak security to chosen-ciphertext security for key encapsulation

“Derandomize, then Hash”

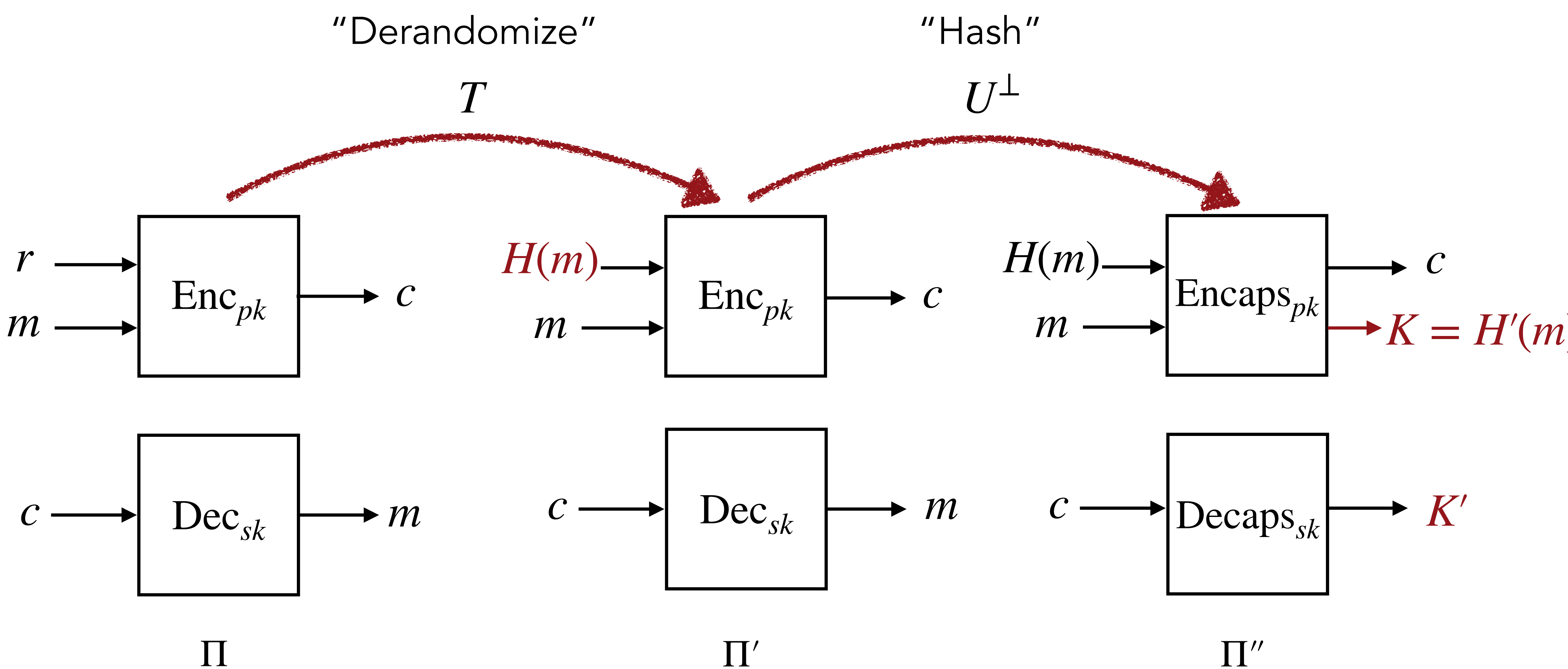


$$K' = \begin{cases} m & K = \mathrm{Enc}_{pk}(m, H(m)) \\ \perp & \text{else} \end{cases}$$

Fujisaki-Okamoto transformation

Upgrades weak security to chosen-ciphertext security for key encapsulation

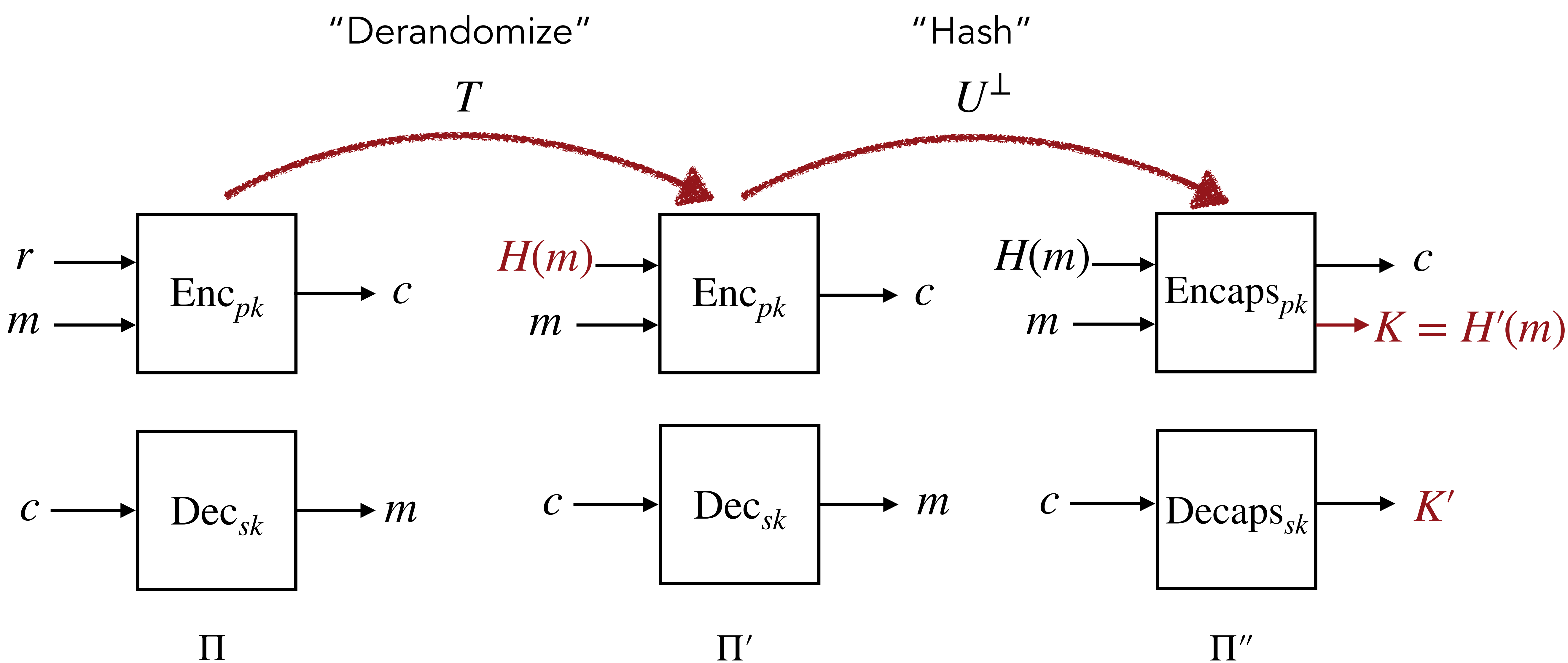
"Derandomize, then Hash"



Fujisaki-Okamoto transformation

Upgrades weak security to chosen-ciphertext security for key encapsulation

"Derandomize, then Hash"



For proving post-quantum security, model H, H' as random oracles (QROM)

Fujisaki-Okamoto transformation in the QROM

Π IND-CPA Π' OW-CPA Π'' IND-CCA

Fujisaki-Okamoto transformation in the QROM

Π IND-CPA

Π' OW-CPA $\xrightarrow[\text{Tight reduction}]{U^{\not\perp}}$ Π'' IND-CCA

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. “Tighter proofs of CCA security in the quantum random oracle model” TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\Pi \text{ IND-CPA} \xrightarrow[\text{Lossy reduction}]{T} \Pi' \text{ OW-CPA} \xrightarrow[\text{Tight reduction}]{U^{\not\perp}} \Pi'' \text{ IND-CCA}$$

Multiplicative loss q

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. “Tighter proofs of CCA security in the quantum random oracle model” TCC 2019

Fujisaki-Okamoto transformation in the QROM

Π IND-CPA $\xrightarrow{T}$ Π' OW-CPA $\xrightarrow{U^{\not\perp}}$ Π'' IND-CCA

Lossy reduction

Multiplicative loss q

Tight reduction

Tight in the classical ROM!

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\Pi \text{ IND-CPA} \xrightarrow[\text{Lossy reduction}]{T} \Pi' \text{ OW-CPA} \xrightarrow[\text{Tight reduction}]{U^{\not\perp}} \Pi'' \text{ IND-CCA}$$

Multiplicative loss q

No attack known that exploits this gap

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\begin{tikzcd} \Pi \text{ IND-CPA} \arrow[r, dashed, "T", "\text{Lossy reduction}"'] & \Pi' \text{ OW-CPA} \arrow[r, "U^{\not\perp}", "\text{Tight reduction}"'] & \Pi'' \text{ IND-CCA} \end{tikzcd}$$

Multiplicative loss q

No attack known that exploits this gap

Vanilla approach (Grover)?

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. “Tighter proofs of CCA security in the quantum random oracle model” TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\Pi\ \text{IND-CPA} \overset{T}{\dashrightarrow} \Pi'\ \text{OW-CPA} \xrightarrow{U^{\not\perp}} \Pi''\ \text{IND-CCA}$$

Lossy reduction

Multiplicative loss q

Tight reduction

No attack known that exploits this gap

Vanilla approach (Grover)? Probably not…

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\Pi \text{ IND-CPA} \xrightarrow[\text{Lossy reduction}]{T} \Pi' \text{ OW-CPA} \xrightarrow[\text{Tight reduction}]{U^{\not\perp}} \Pi'' \text{ IND-CCA}$$

Multiplicative loss q

No attack known that exploits this gap

Vanilla approach (Grover)? Probably not…

Other algorithms?

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

Π IND-CPA $\xrightarrow{T}$ Π' OW-CPA $\xrightarrow{U^{\not\perp}}$ Π'' IND-CCA

T: Lossy reduction

Multiplicative loss q

$U^{\not\perp}$: Tight reduction

No attack known that exploits this gap

Vanilla approach (Grover)? Probably not…

Other algorithms?

(This is the question from Dan's email)

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\Pi \text{ IND-CPA} \xrightarrow{T} \Pi' \text{ OW-CPA} \xrightarrow{U^{\not\perp}} \Pi'' \text{ IND-CCA}$$

Lossy reduction

Multiplicative loss q

Tight reduction

Reductions in the QROM

No attack known that exploits this gap

Vanilla approach (Grover)? Probably not…

Other algorithms?

(This is the question from Dan's email)

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

$$\Pi \text{ IND-CPA} \xrightarrow{T} \Pi' \text{ OW-CPA} \xrightarrow{U^{\not\perp}} \Pi'' \text{ IND-CCA}$$

Lossy reduction

Multiplicative loss q

Tight reduction

Reductions in the QROM
⇒ same insufficiency as for FS

No attack known that exploits this gap

Vanilla approach (Grover)? Probably not…

Other algorithms?

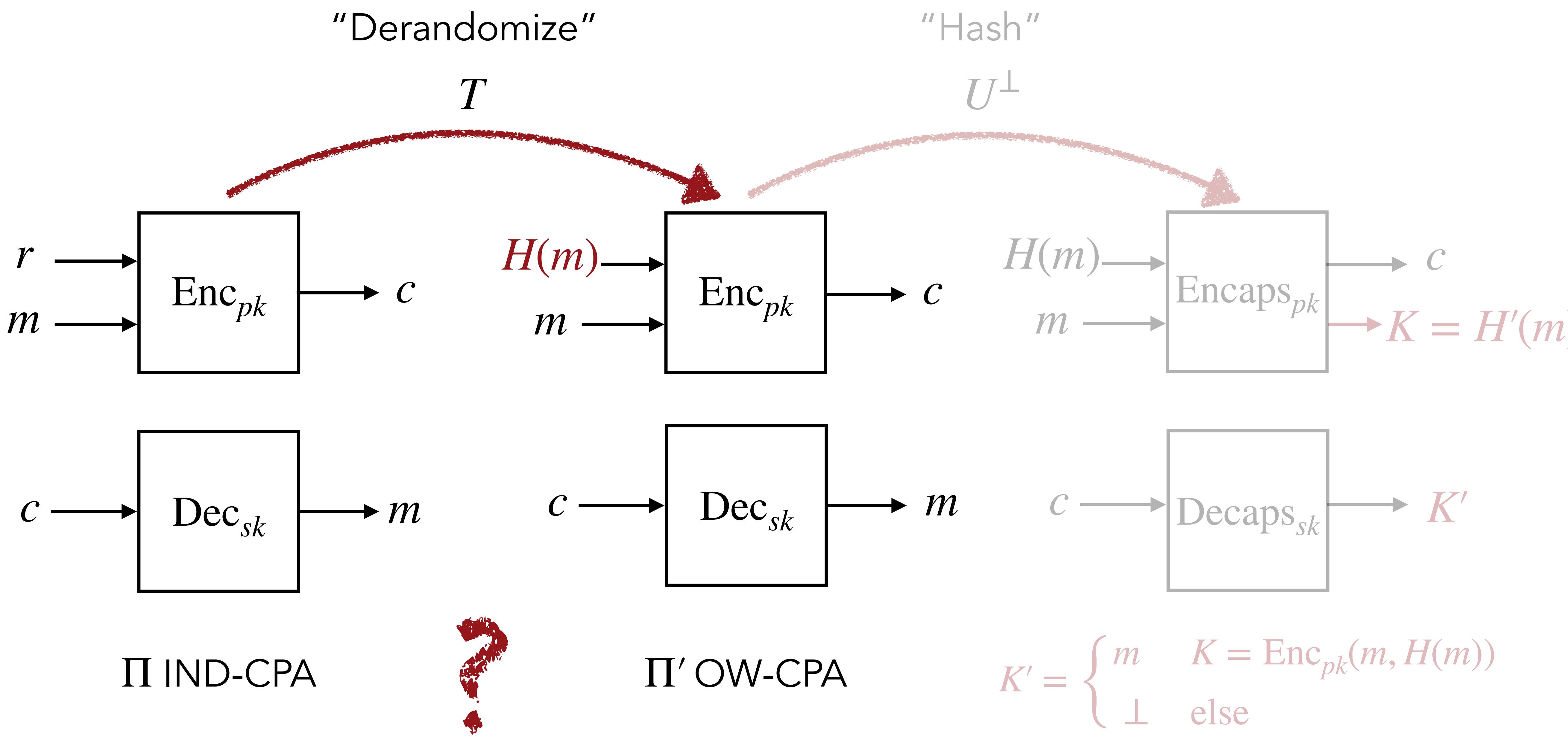
(This is the question from Dan's email)

Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., & Persichetti, E. "Tighter proofs of CCA security in the quantum random oracle model" TCC 2019

Fujisaki-Okamoto transformation in the QROM

Upgrades weak security to chosen-ciphertext security for key encapsulation

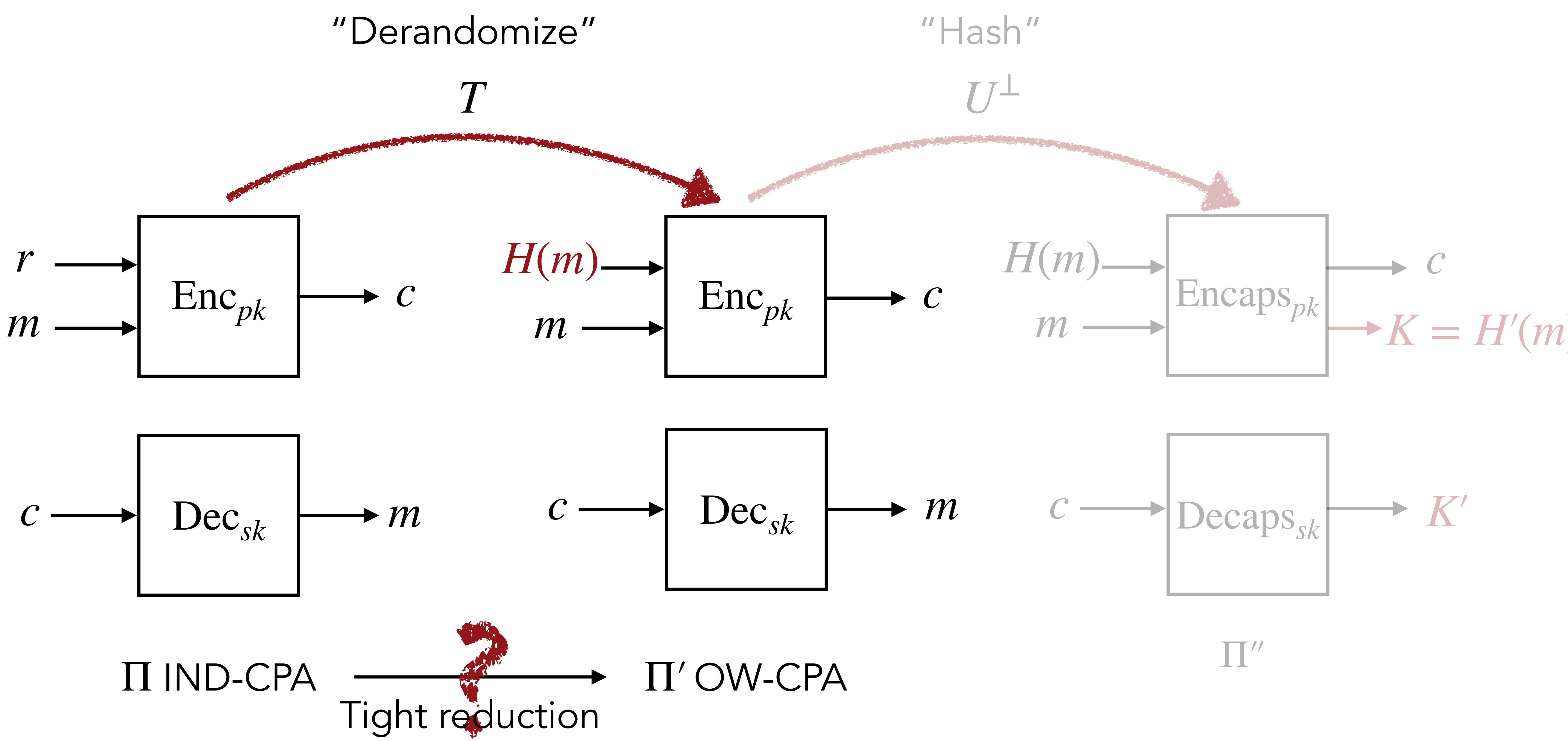
“Derandomize, then Hash”



Fujisaki-Okamoto transformation in the QROM

Upgrades weak security to chosen-ciphertext security for key encapsulation

"Derandomize, then Hash"



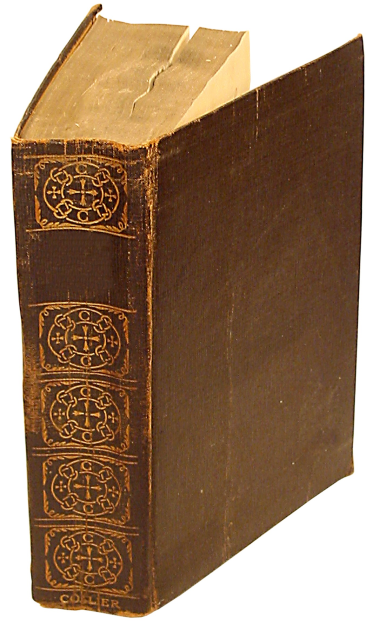
Summary

Hash functions are used everywhere. ⇒We need to subject them to quantum cryptanalysis!

Attacks possible at different levels

Hash function application in schemes: some open questions regarding attacks

Polynomial improvements over trivial, but: important for parameter choice



Thanks!