Diverse Modules and Zero-Knowledge

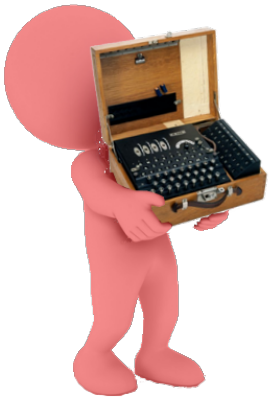
Fabrice Ben Hamouda--Guichoux

Thèse effectuée à l'ENS
sous la direction de
Michel Abdalla et David Pointcheval

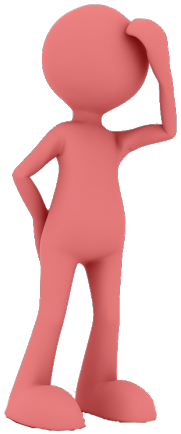
IBM Research

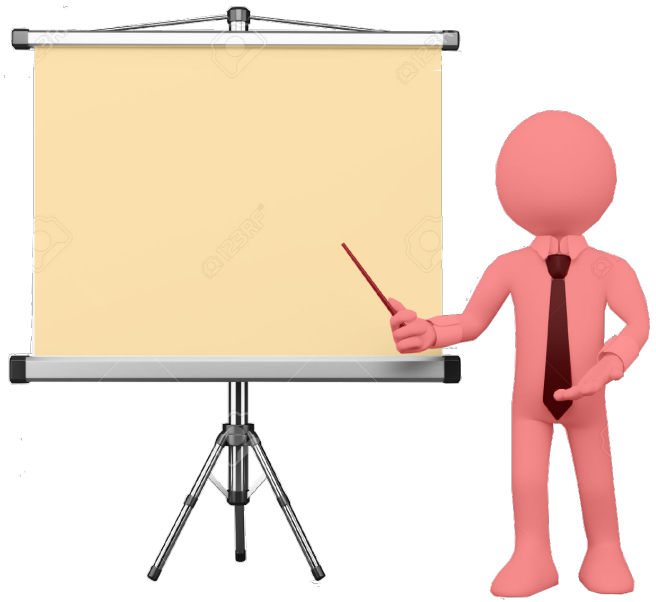
Congrès SIF 2017
Vendredi 3 Février 2017

Introduction



Introduction

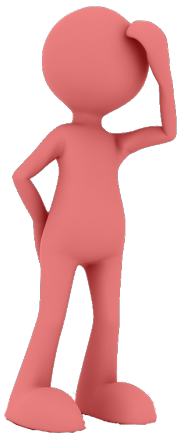


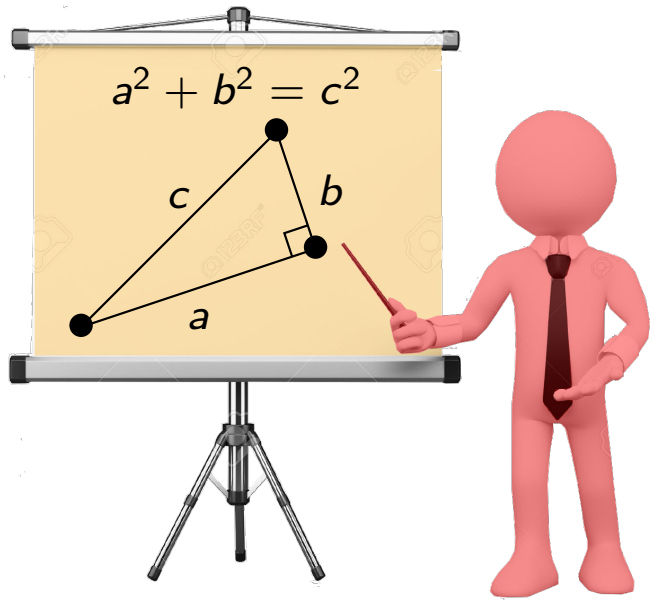


Introduction

- Classical Mathematical proofs

  statement — e.g., Pythagorean theorem

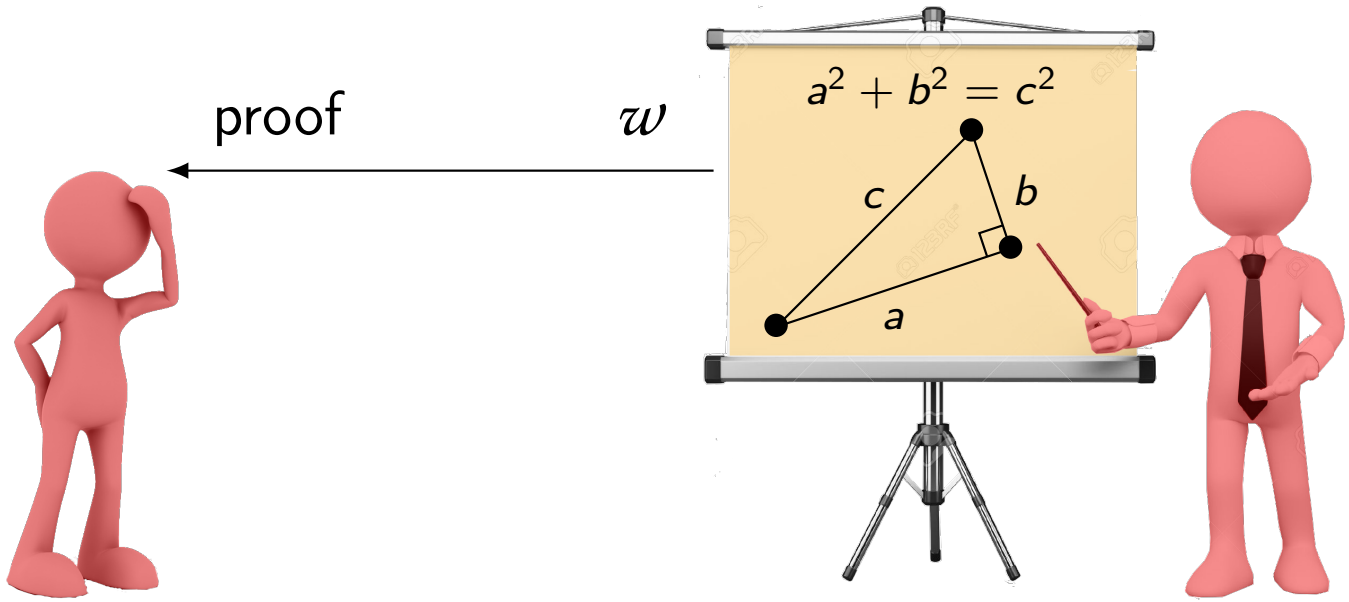




Introduction

- Classical Mathematical proofs

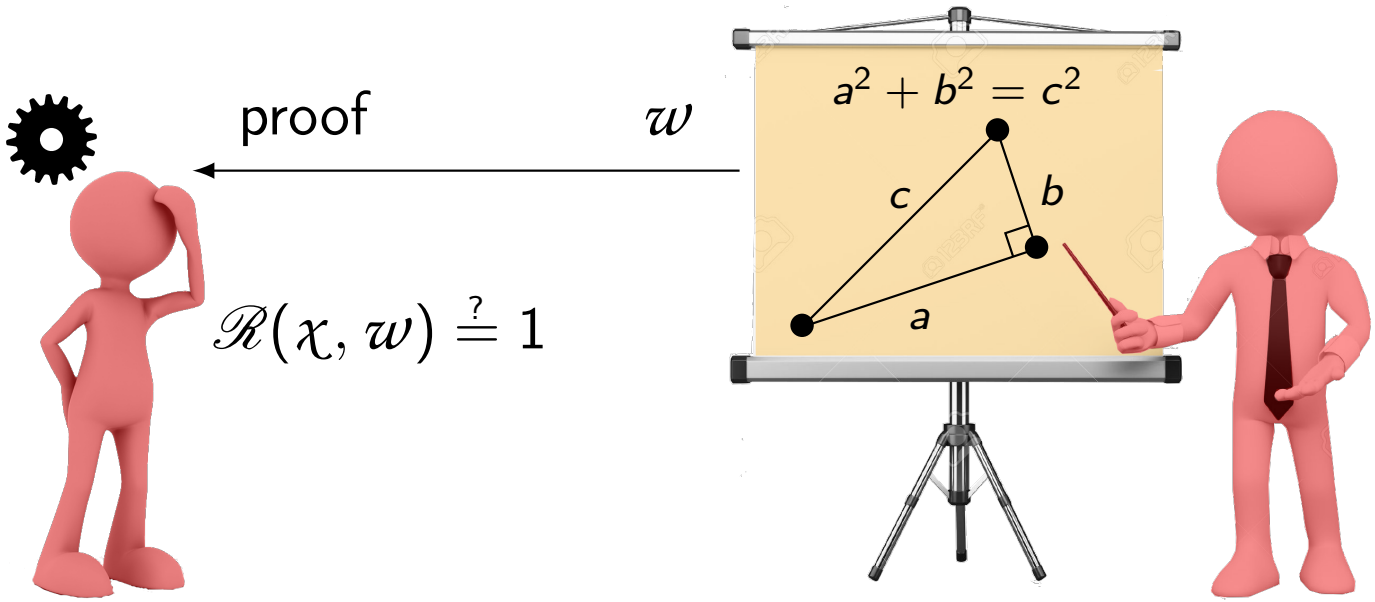
  statement e.g., Pythagorean theorem



Introduction

- Classical Mathematical proofs

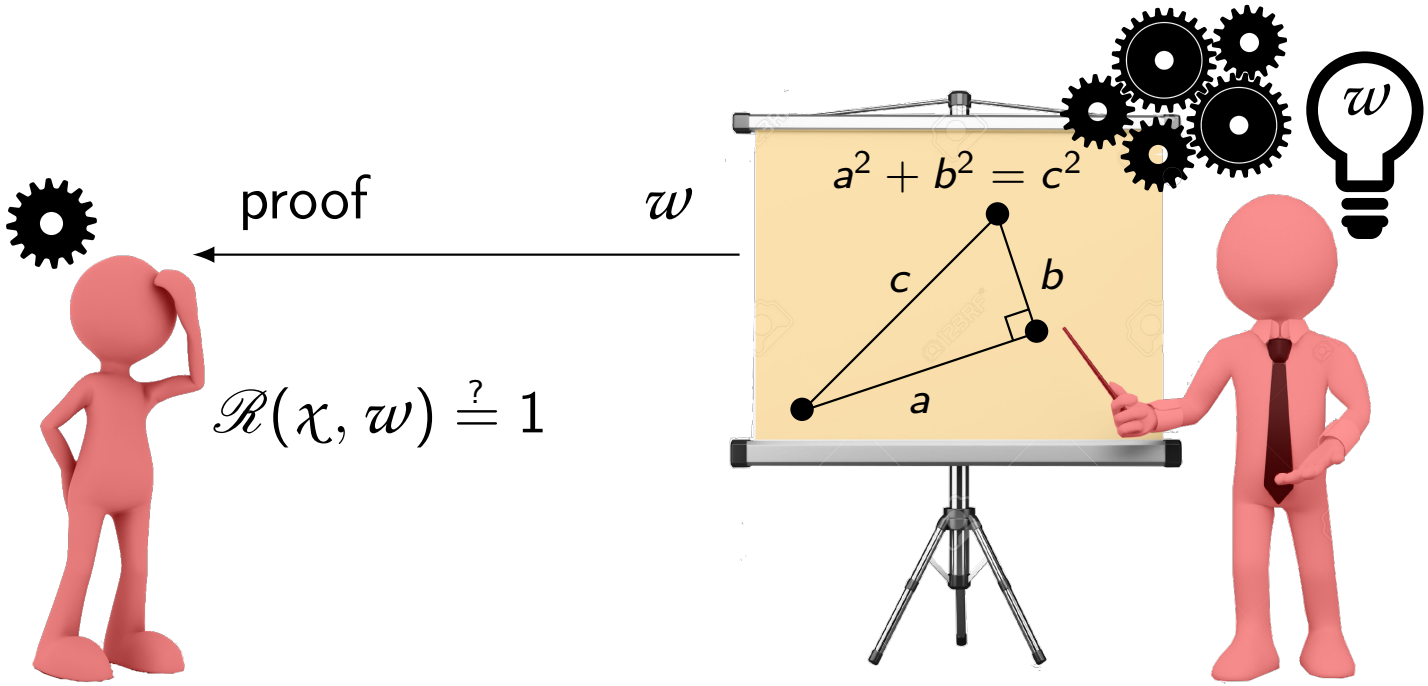
  statement e.g., Pythagorean theorem



Introduction

- Classical Mathematical proofs

  statement e.g., Pythagorean theorem

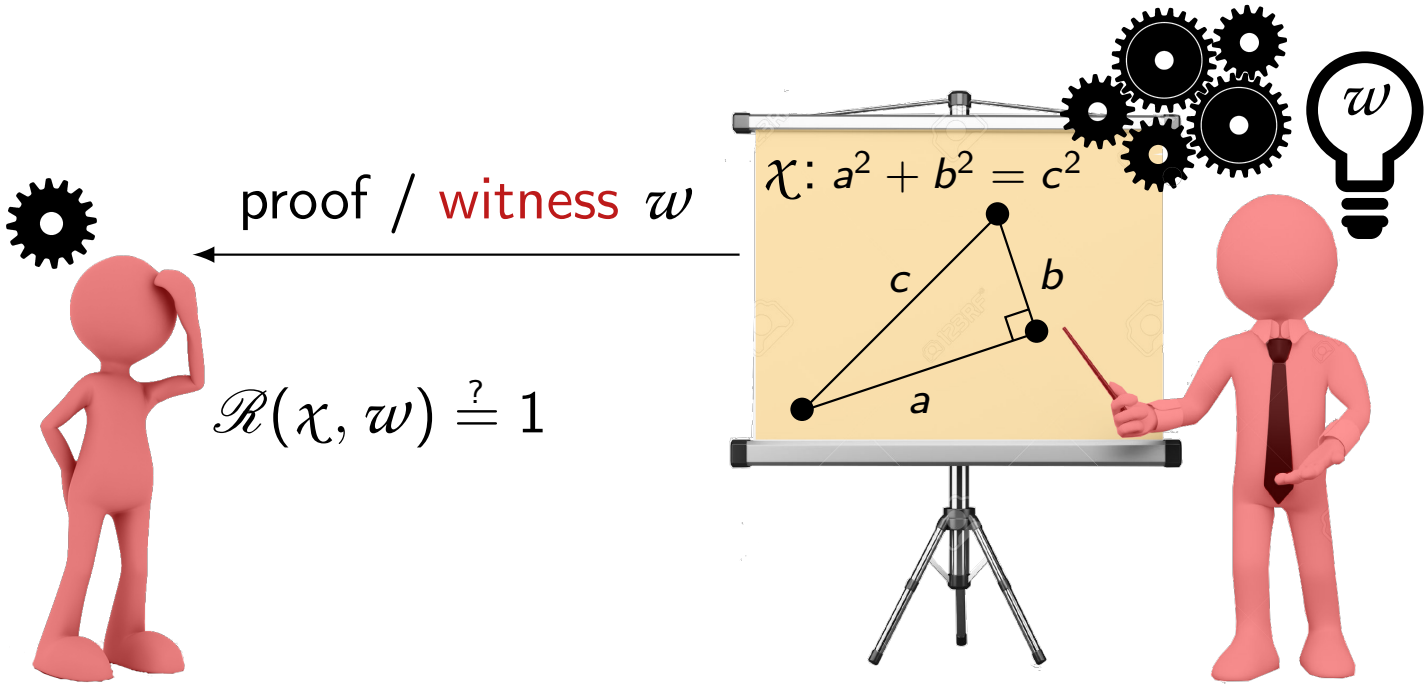


Introduction

- Classical Mathematical proofs / NP language

  statement / word x e.g., Pythagorean theorem

$$x \in \mathscr{L} \subseteq \mathscr{X} \iff \exists w,\ \mathscr{R}(x, w) = 1;$$

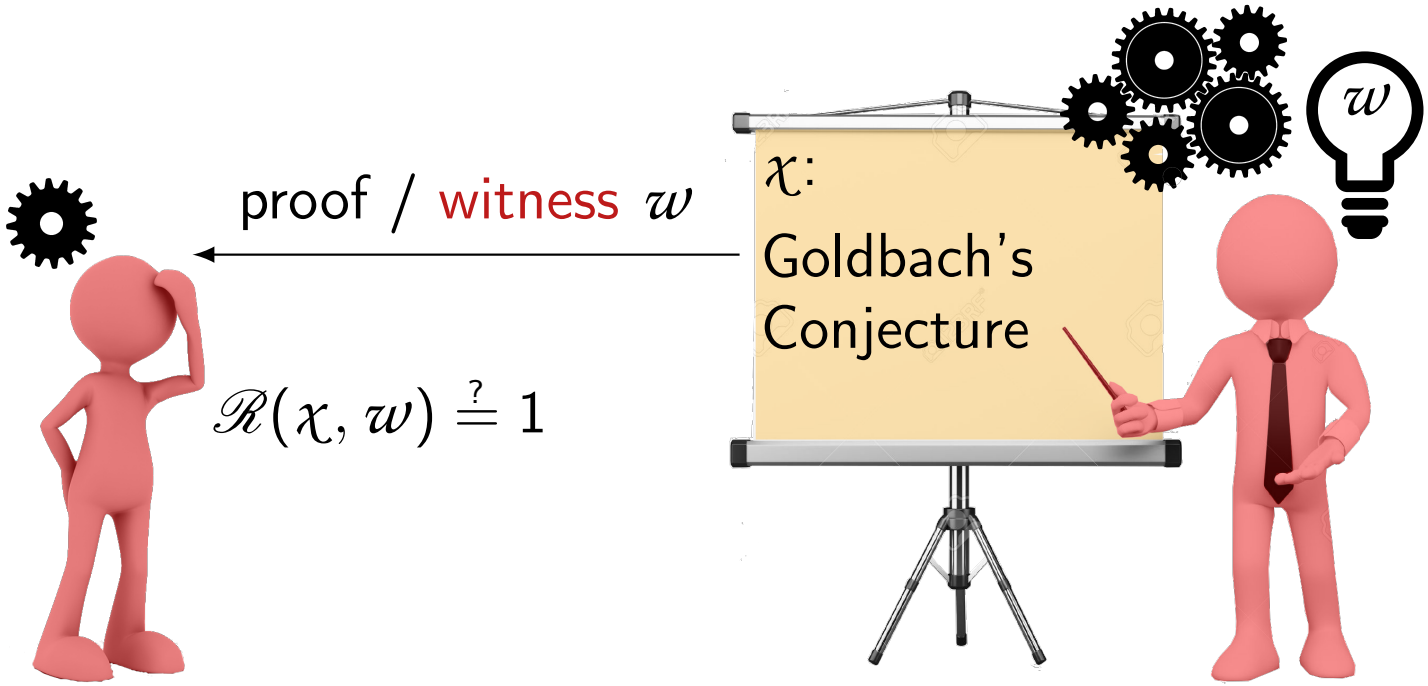


Introduction

- Classical Mathematical proofs / NP language

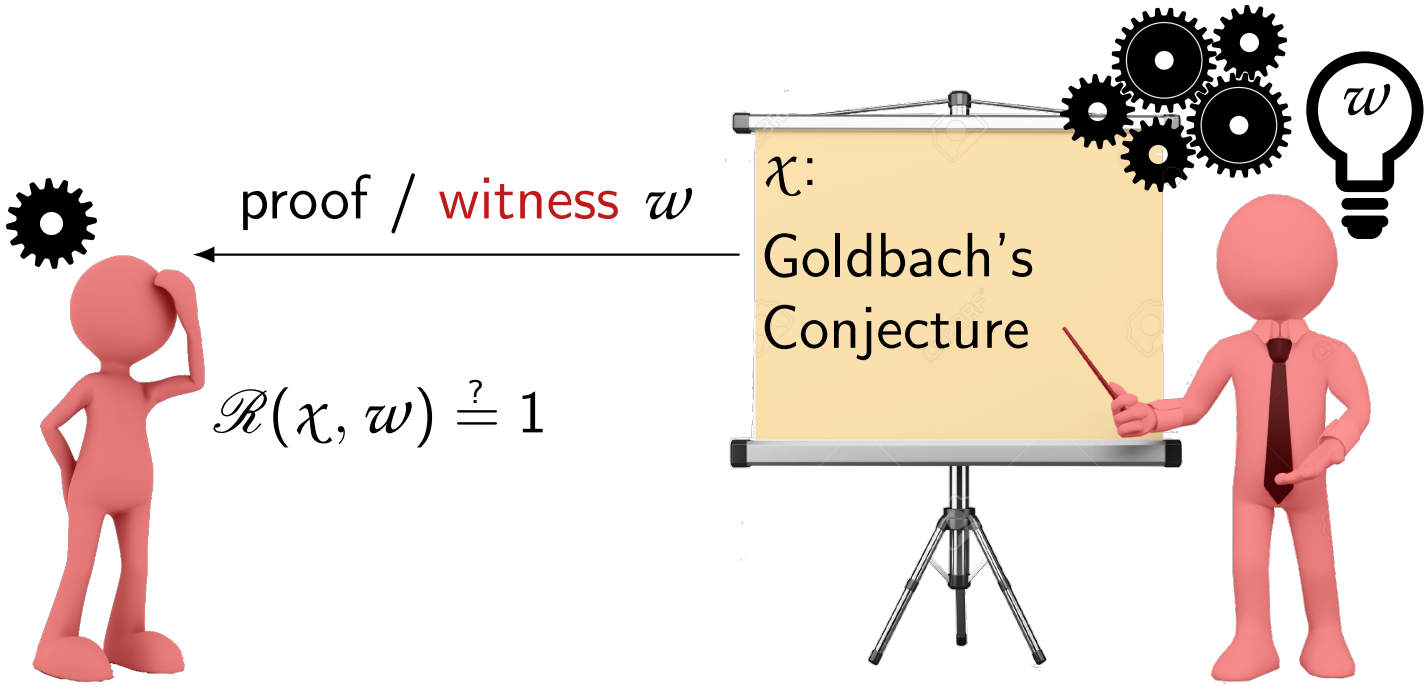
  statement / word x e.g., Pythagorean theorem

$$x \in \mathscr{L} \subseteq \mathscr{X} \iff \exists w,\ \mathscr{R}(x, w) = 1;$$



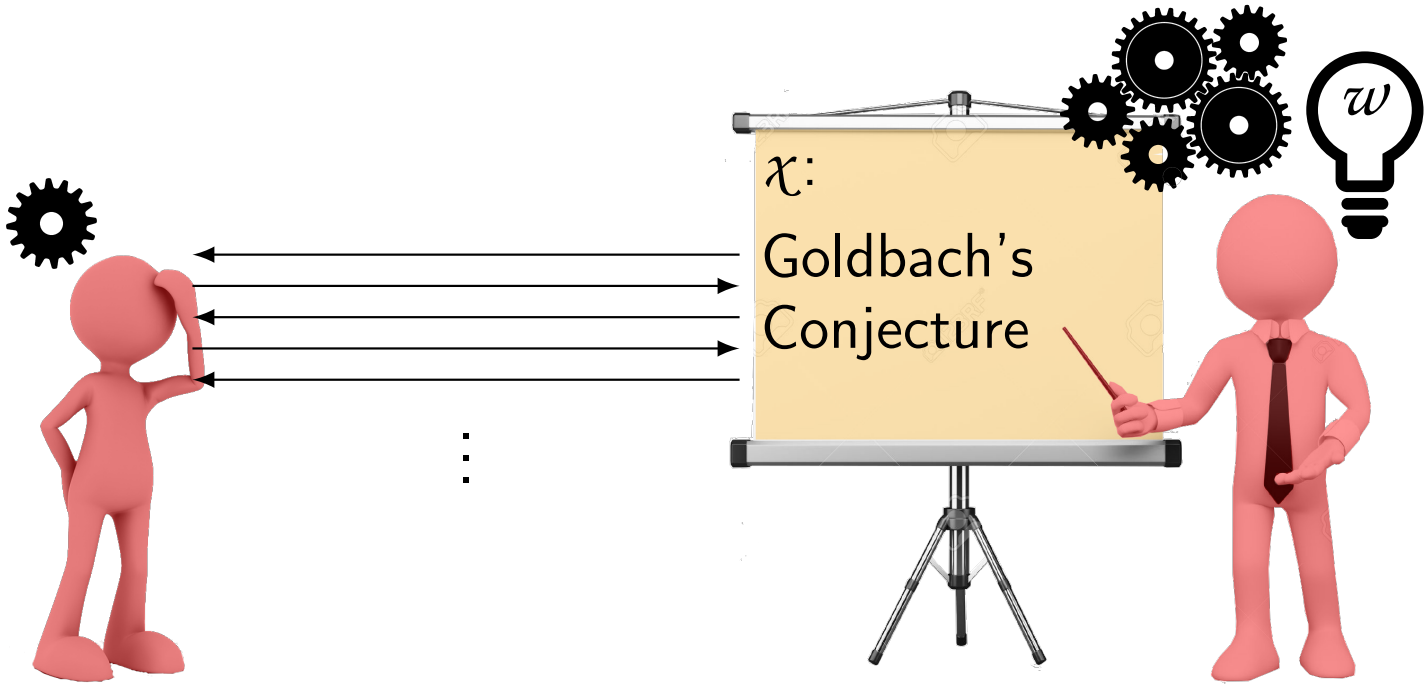
Introduction

- Zero-knowledge proofs
  - Goldwasser-Micali-Rackoff 1985
  - Verifier learns nothing except "$x \in \mathscr{L}$"



Introduction

- Zero-knowledge proofs
  - Goldwasser-Micali-Rackoff 1985
  - Verifier learns nothing except "$x \in \mathscr{L}$"



Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]
- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)


- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]
- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]
- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]

- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - $\rightarrow$ IND-CCA encryption scheme [CS98]

- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]

- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]

Simple languages

- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof Systems / Smooth Projective Hash Functions (SPHFs)

- Introduced by Cramer and Shoup [CS02]
  - → IND-CCA encryption scheme [CS98]

Simple languages

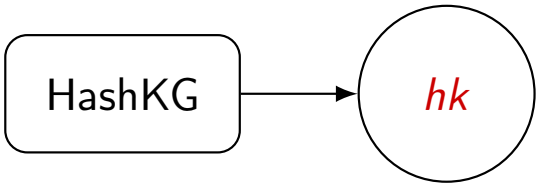
- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11]
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

More Complex Languages

Smooth Projective Hash Functions (SPHFs)

Definition

NP language $\mathscr{L}$: $x \in \mathscr{L} \subseteq \mathscr{X} \iff \exists w,\ \mathscr{R}(x, w) = 1$

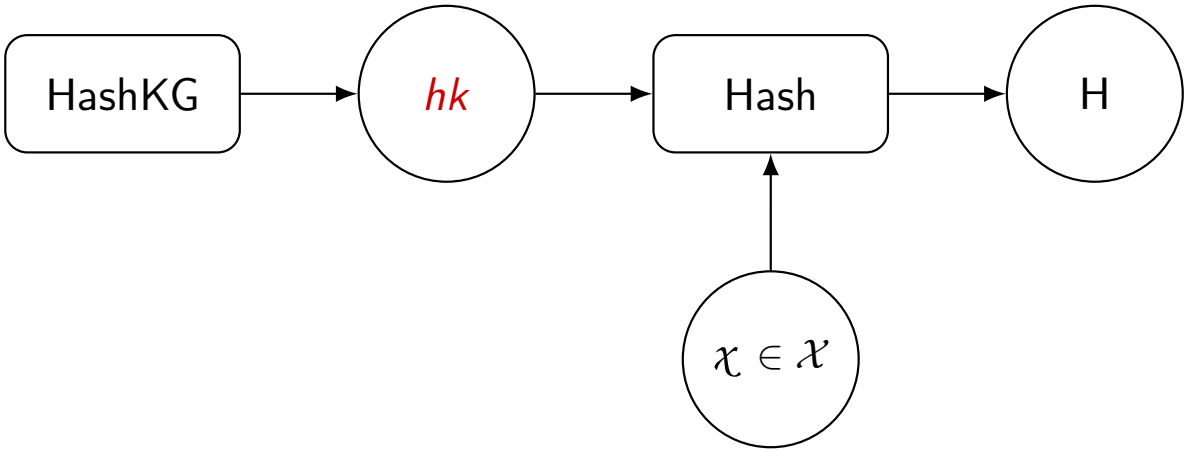


$hk \xleftarrow{\$}$ HashKG()

Smooth Projective Hash Functions (SPHFs)

Definition

NP language $\mathscr{L}$: $x \in \mathscr{L} \subseteq \mathscr{X} \iff \exists w,\ \mathscr{R}(x, w) = 1$



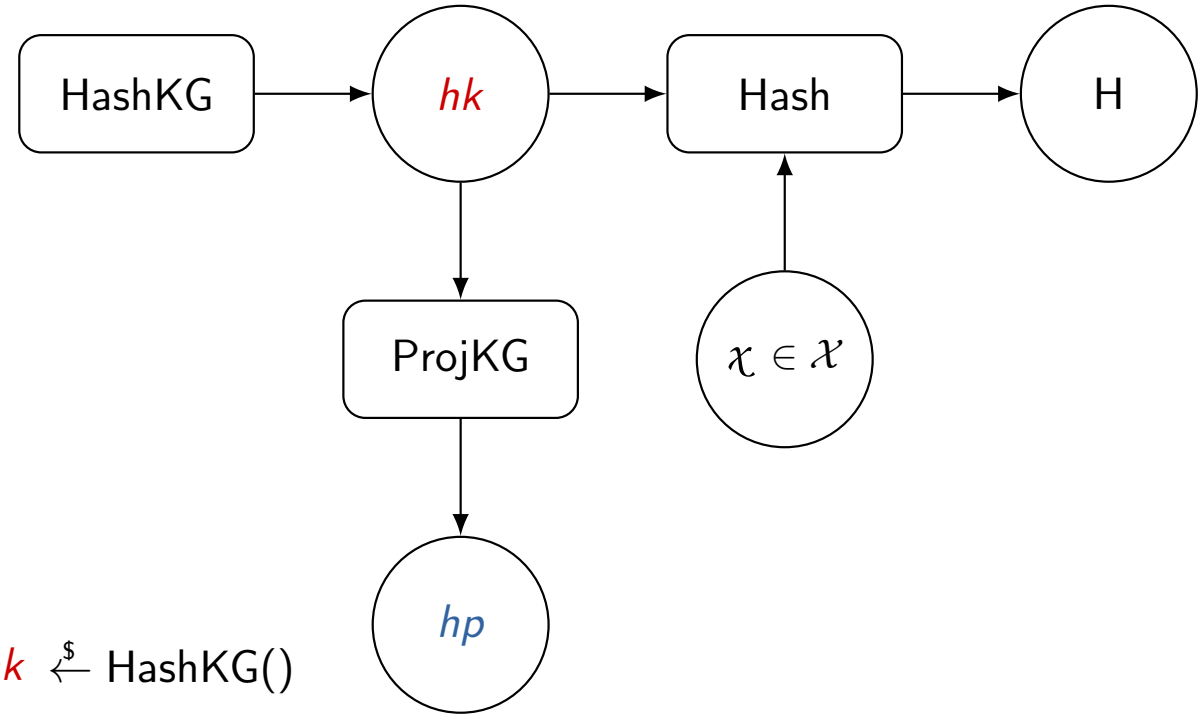
$hk \xleftarrow{\$} \mathsf{HashKG}()$

$\mathsf{H} \leftarrow \mathsf{Hash}(hk, x)$

Smooth Projective Hash Functions (SPHFs)

Definition

NP language $\mathscr{L}$: $x \in \mathscr{L} \subseteq \mathscr{X} \iff \exists w,\ \mathscr{R}(x, w) = 1$



$hk \xleftarrow{\$} \text{HashKG}()$

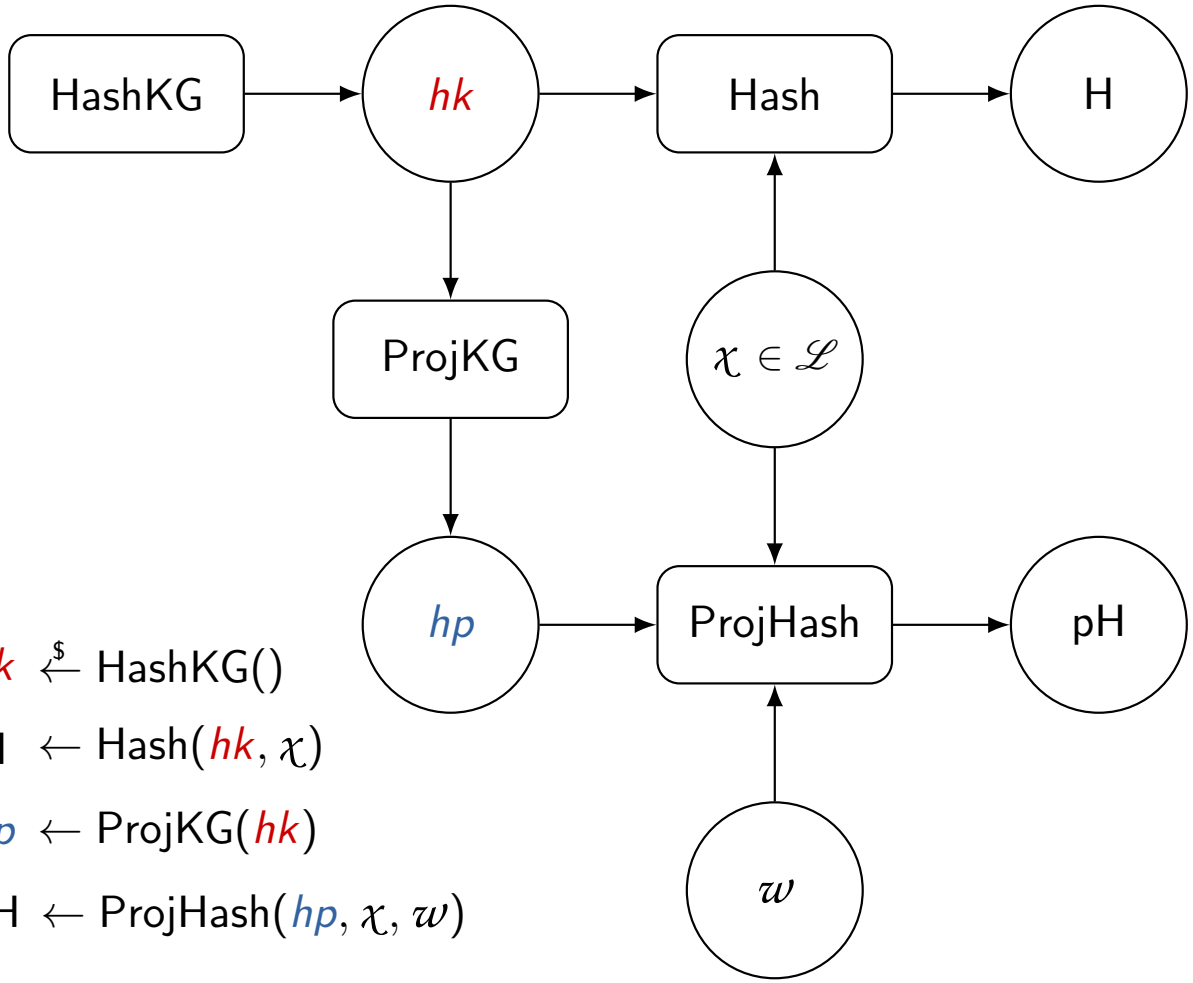
$\text{H} \leftarrow \text{Hash}(hk, x)$

$hp \leftarrow \text{ProjKG}(hk)$

Smooth Projective Hash Functions (SPHFs)

Definition

NP language $\mathscr{L}$: $x \in \mathscr{L} \subseteq \mathcal{X} \iff \exists w,\ \mathscr{R}(x, w) = 1$



$hk \xleftarrow{\$} \mathsf{HashKG}()$

$\mathsf{H} \leftarrow \mathsf{Hash}(hk, x)$

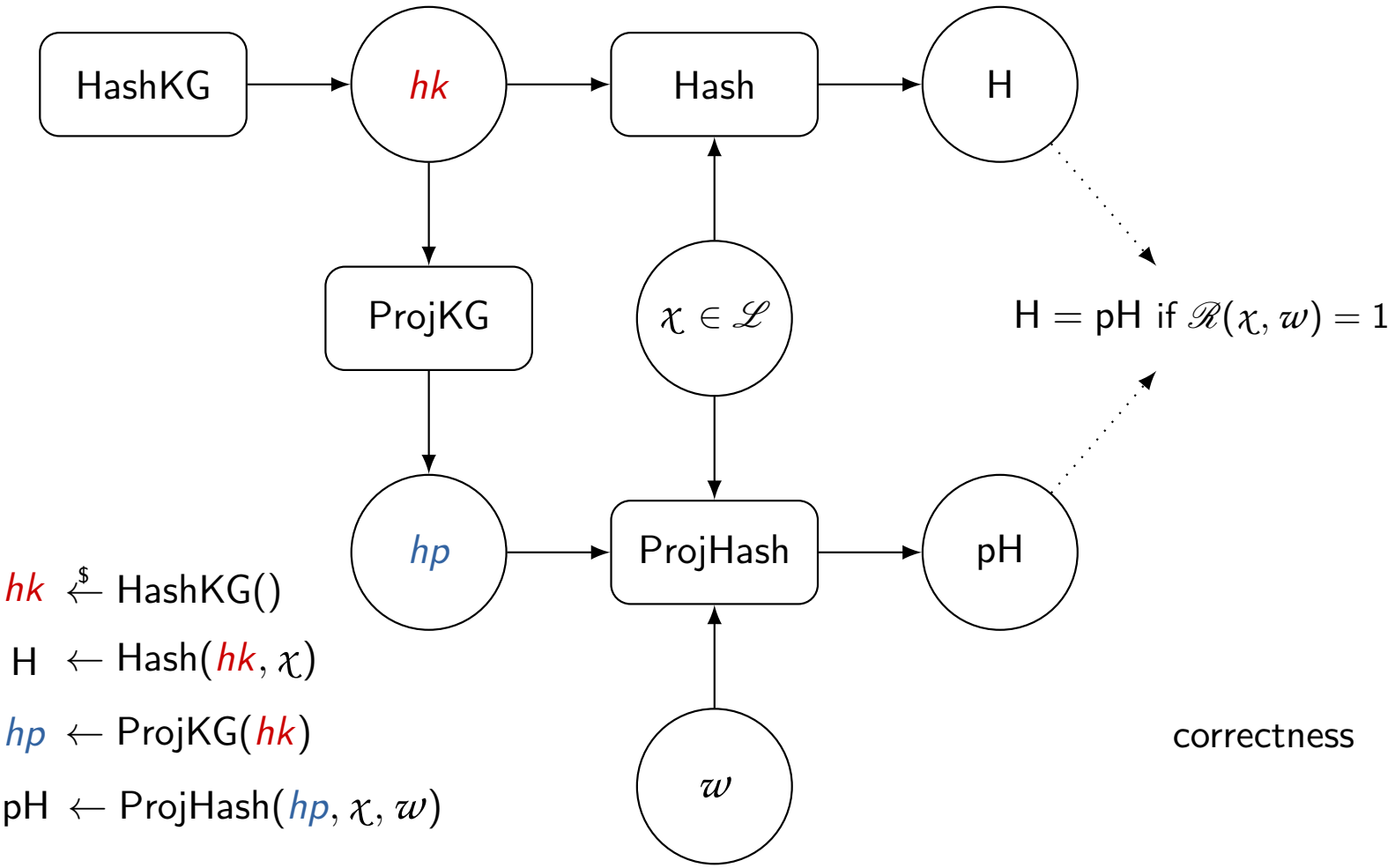
$hp \leftarrow \mathsf{ProjKG}(hk)$

$\mathsf{pH} \leftarrow \mathsf{ProjHash}(hp, x, w)$

Smooth Projective Hash Functions (SPHFs)

Definition

NP language $\mathscr{L}$: $x \in \mathscr{L} \subseteq \mathcal{X} \iff \exists w,\ \mathscr{R}(x, w) = 1$



$hk \xleftarrow{\$}$ HashKG()

H $\leftarrow$ Hash(hk, x)

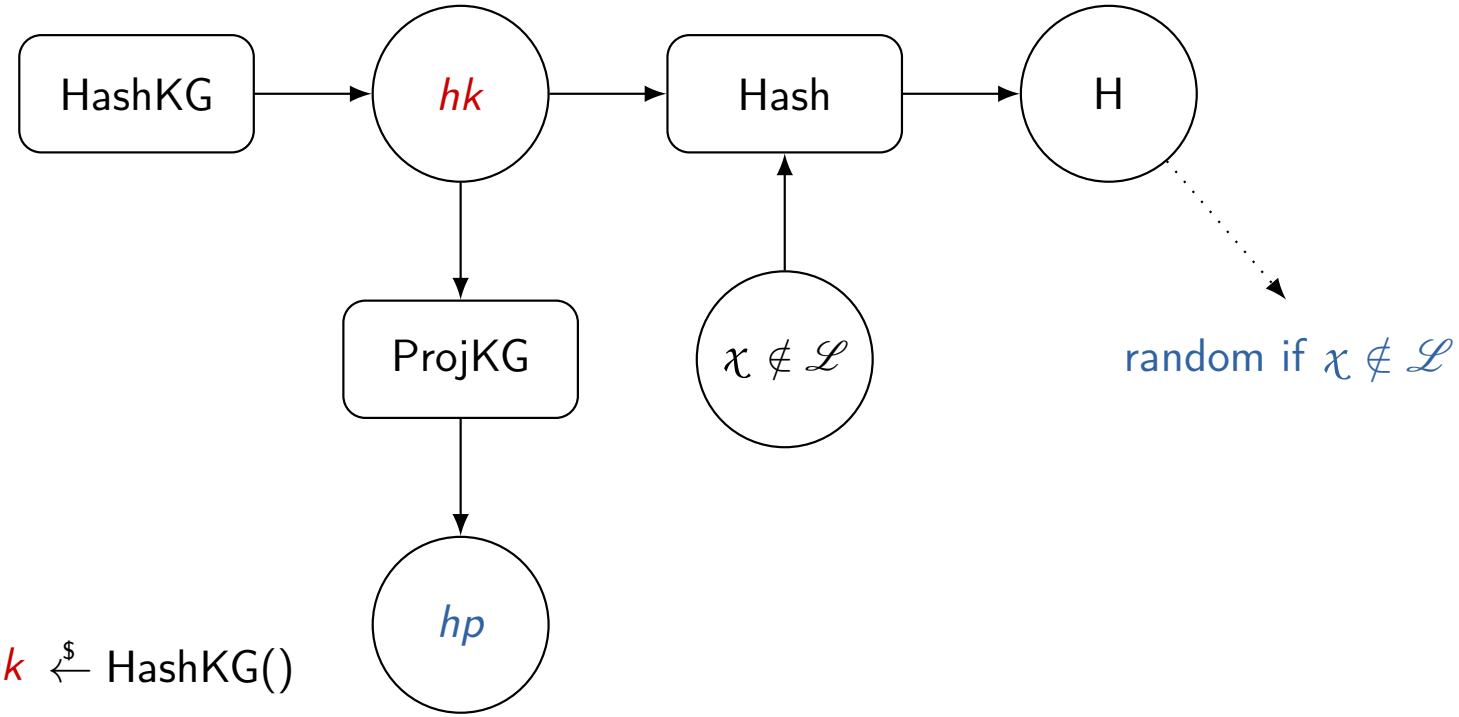
$hp \leftarrow$ ProjKG(hk)

pH $\leftarrow$ ProjHash(hp, x, w)

Smooth Projective Hash Functions (SPHFs)

Definition

NP language $\mathscr{L}$: $x \in \mathscr{L} \subseteq \mathcal{X} \iff \exists w,\ \mathscr{R}(x, w) = 1$



$hk \xleftarrow{\$}$ HashKG()

H $\leftarrow$ Hash(hk, x)

$hp \leftarrow$ ProjKG(hk)

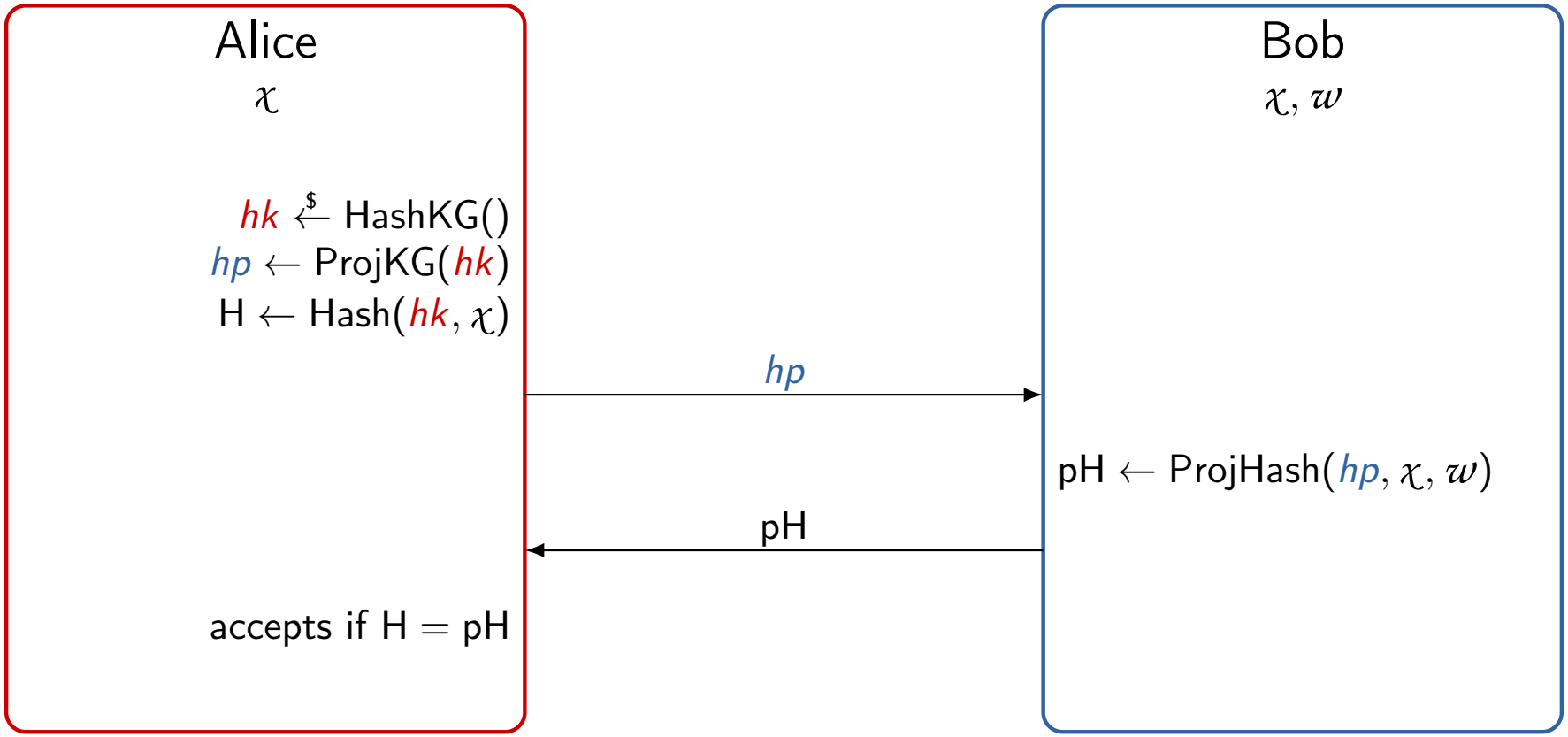
pH $\leftarrow$ ProjHash(hp, x, w)

smoothness

Direct Applications of SPHFs

Honest-Verifier Zero-Knowledge Proofs

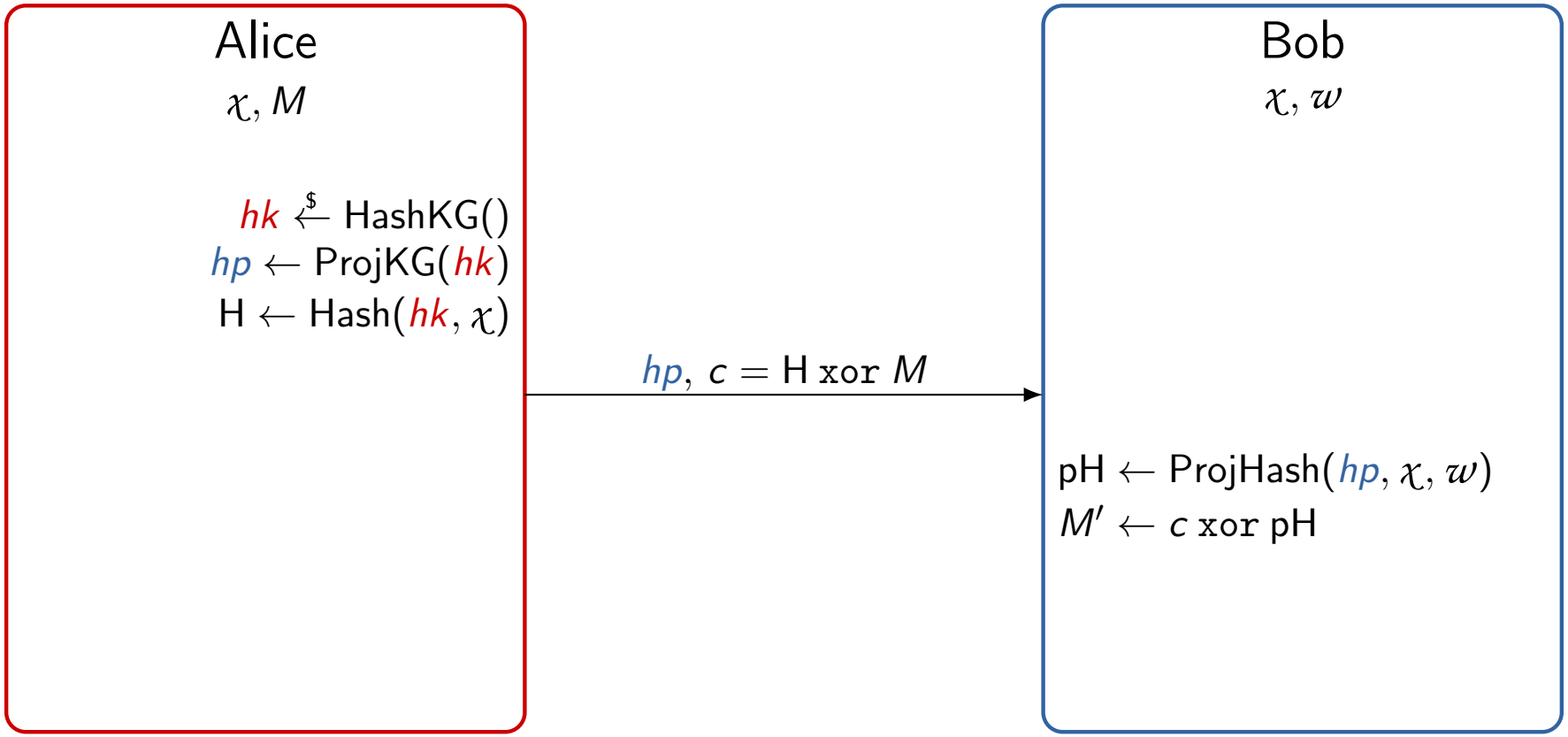
Bob wants to prove to Alice that $x \in \mathcal{L}$.



Direct Applications of SPHFs

Implicit Arguments

Alice wants to send M to Bob if $\chi \in \mathcal{L}$.



Agenda

1. Diverse Modules
   - Languages Handled by SPHFs?
   - ElGamal
   - Diverse Modules
   - Conjunctions and Disjunctions

2. (Implicit) Zero-Knowledge Arguments and Applications
   - (Implicit) Zero-Knowledge Arguments
   - Application to Two-Party Secure Computation

Languages Handled by SPHFs?

- Any NP language?

Languages Handled by SPHFs?

- Any NP language?
  - $\Rightarrow$ polynomial hierarchy collapses [GGSW13]

Languages Handled by SPHFs?

- Any NP language?
  - $\Rightarrow$ polynomial hierarchy collapses [GGSW13]
- This thesis: "algebraic languages"

Languages Handled by SPHFs?

- Any NP language?
  - $\Rightarrow$ polynomial hierarchy collapses [GGSW13]

- This thesis: "algebraic languages"
  - Diverse vector spaces: $\mathscr{L}$ subspace of vector space $\mathscr{X}$ algebraic languages over prime-order cyclic groups

Languages Handled by SPHFs?

- Any NP language?
  - $\Rightarrow$ polynomial hierarchy collapses [GGSW13]

- This thesis: "algebraic languages"
  - Diverse vector spaces: $\mathscr{L}$ subspace of vector space $\mathscr{X}$
    algebraic languages over prime-order cyclic groups
  - Diverse modules: $\mathscr{L}$ submodule of module $\mathscr{X}$
    module = vector space over a ring

Cyclic Groups and Hard Problems

$(\mathbb{G}, +)$ cyclic group of prime order p, generator $\boxed{g}$:

- "Encoding" of $x \in \mathbb{Z}_p$:

$$x \in \mathbb{Z}_p \qquad \mapsto \qquad x \cdot \boxed{g} \in \mathbb{G}$$

Cyclic Groups and Hard Problems

$(\mathbb{G}, +)$ cyclic group of prime order p, generator $\boxed{g}$:

- "Encoding" of $x \in \mathbb{Z}_p$:

$$x \in \mathbb{Z}_p \qquad \mapsto \qquad x \cdot \boxed{g} \in \mathbb{G}$$

- We assume DDH:

$$(\boxed{g},\ x \cdot \boxed{g},\ y \cdot \boxed{g},\ xy \cdot \boxed{g}) \quad \approx_c \quad (\boxed{g},\ x \cdot \boxed{g},\ y \cdot \boxed{g},\ z \cdot \boxed{g})$$

with $x, y, z \xleftarrow{\$} \mathbb{Z}_p$

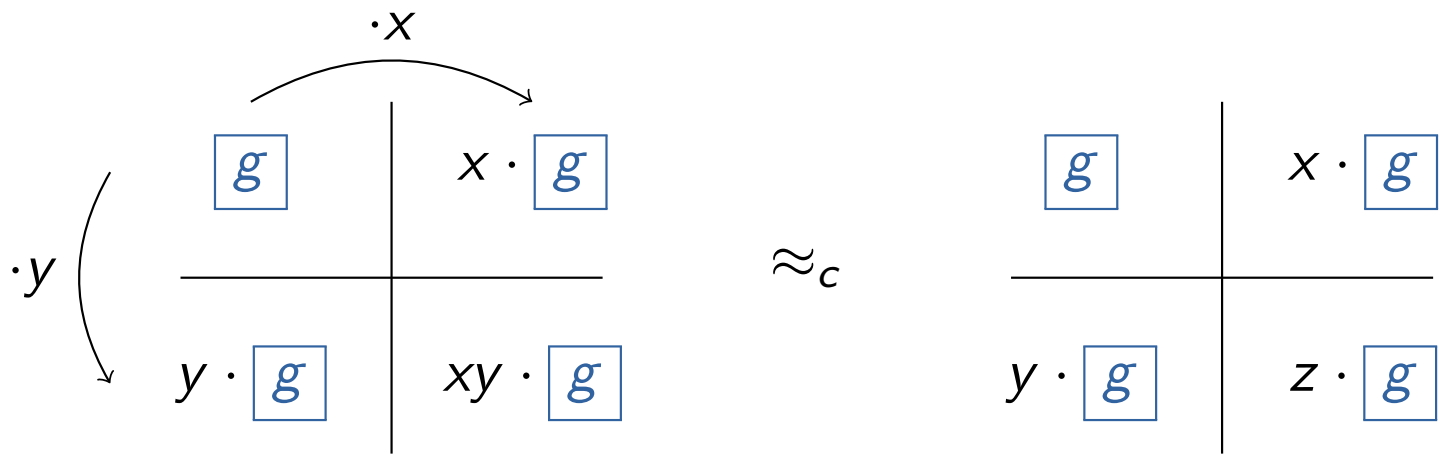
Cyclic Groups and Hard Problems

$(\mathbb{G}, +)$ cyclic group of prime order p, generator $\boxed{g}$:

- "Encoding" of $x \in \mathbb{Z}_p$:

$$x \in \mathbb{Z}_p \qquad \mapsto \qquad x \cdot \boxed{g} \in \mathbb{G}$$

- We assume DDH:



with $x, y, z \xleftarrow{\$} \mathbb{Z}_p$

Warm-up: ElGamal Encryption Scheme

Receiver: $s \xleftarrow{\$} \mathbb{Z}_p$ (secret)

$\boxed{g}$ $\qquad$ $\boxed{h} = s \cdot \boxed{g}$

Sender: message $\boxed{M}$, randomness $r \xleftarrow{\$} \mathbb{Z}_p$

$\boxed{u} = r \cdot \boxed{g}$ $\qquad$ $\boxed{v} = r \cdot \boxed{h} + \boxed{M}$

Warm-up: ElGamal Encryption Scheme

Receiver: $s \xleftarrow{\$} \mathbb{Z}_p$ (secret)

$\boxed{g}$ $\qquad$ $\boxed{h} = s \cdot \boxed{g}$

Sender: message $\boxed{M}$, randomness $r \xleftarrow{\$} \mathbb{Z}_p$

$\boxed{u} = r \cdot \boxed{g}$ $\qquad$ $\boxed{v} = r \cdot \boxed{h} + \boxed{M}$

Decryption: $\boxed{M} = \boxed{v} - s \cdot \boxed{u}$

Warm-up: ElGamal Encryption Scheme

Receiver: $s \xleftarrow{\$} \mathbb{Z}_p$ (secret)

$\boxed{g}$ $\qquad$ $\boxed{h} = s \cdot \boxed{g}$

Sender: message $\boxed{M}$, randomness $r \xleftarrow{\$} \mathbb{Z}_p$

$\boxed{u} = r \cdot \boxed{g}$ $\qquad$ $\boxed{v} = r \cdot \boxed{h} + \boxed{M}$

Decryption: $\boxed{M} = \boxed{v} - s \cdot \boxed{u}$

Security under DDH:

$$(\boxed{g}, \boxed{h}, r \cdot \boxed{g}, r \cdot \boxed{h}) \quad \approx_c \quad (\boxed{g}, \boxed{h}, r \cdot \boxed{g}, z \cdot \boxed{g})$$

Warm-up: ElGamal Encryption Scheme

SPHF for ciphertexts of $\boxed{M} = 0$

$\mathscr{L} = \{(\boxed{u}, \boxed{v}) \mid \exists r \in \mathbb{Z}_p,\ \boxed{u} = r \cdot \boxed{g} \text{ and } \boxed{v} = r \cdot \boxed{h}\}$

Receiver: $s \xleftarrow{\$} \mathbb{Z}_p$ (secret)

$\boxed{g}$ $\qquad$ $\boxed{h} = s \cdot \boxed{g}$

Sender: message $\boxed{M}$, randomness $r \xleftarrow{\$} \mathbb{Z}_p$

$\boxed{u} = r \cdot \boxed{g}$ $\qquad$ $\boxed{v} = r \cdot \boxed{h} + \boxed{M}$

Warm-up: ElGamal Encryption Scheme

SPHF for ciphertexts of $\boxed{M} = 0$

$\mathscr{L} = \{(\boxed{u}, \boxed{v}) \mid \exists r \in \mathbb{Z}_p,\ \boxed{u} = r \cdot \boxed{g} \text{ and } \boxed{v} = r \cdot \boxed{h}\}$

Receiver: $s \xleftarrow{\$} \mathbb{Z}_p$ (secret)

$\boxed{g}$ $\qquad$ $\boxed{h} = s \cdot \boxed{g}$

Sender: message $\boxed{M}$, randomness $r \xleftarrow{\$} \mathbb{Z}_p$

$\boxed{u} = r \cdot \boxed{g}$ $\qquad$ $\boxed{v} = r \cdot \boxed{h} + \boxed{M}$

SPHF: $hk = (\alpha, \beta) \xleftarrow{\$} \mathbb{Z}_p^2$

$\boxed{hp} = \alpha \cdot \boxed{g} + \beta \cdot \boxed{h}$

$\boxed{\mathrm{H}} = \alpha \cdot \boxed{u} + \beta \cdot \boxed{v}$

Warm-up: ElGamal Encryption Scheme

SPHF for ciphertexts of $\boxed{M} = 0$

$\mathscr{L} = \{(\boxed{u}, \boxed{v}) \mid \exists r \in \mathbb{Z}_p,\ \boxed{u} = r \cdot \boxed{g} \text{ and } \boxed{v} = r \cdot \boxed{h}\}$

Receiver: $s \xleftarrow{\$} \mathbb{Z}_p$ (secret)

$\boxed{g}$ $\qquad$ $\boxed{h} = s \cdot \boxed{g}$

$\cdot r$

Sender: message $\boxed{M}$, randomness $r \xleftarrow{\$} \mathbb{Z}_p$

$\boxed{u} = r \cdot \boxed{g}$ $\qquad$ $\boxed{v} = r \cdot \boxed{h} + \boxed{M}$

SPHF: $hk = (\alpha, \beta) \xleftarrow{\$} \mathbb{Z}_p^2$

$\boxed{hp} = \alpha \cdot \boxed{g} + \beta \cdot \boxed{h}$

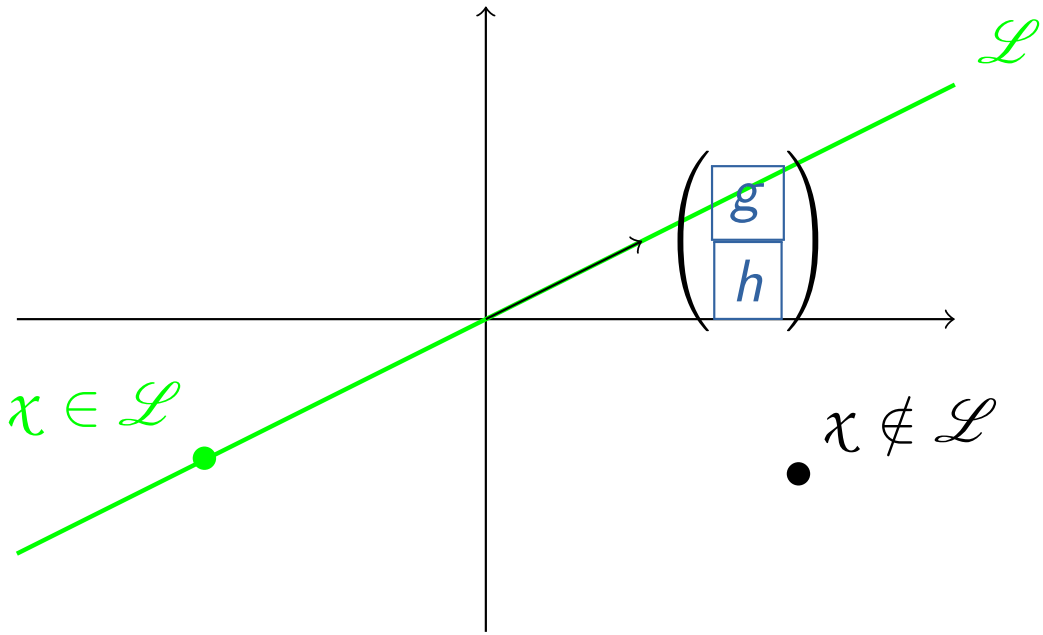
$\boxed{\mathrm{H}} = \alpha \cdot \boxed{u} + \beta \cdot \boxed{v}$

$\boxed{\mathrm{pH}} = r \cdot \boxed{hp}$

Diverse Vector Spaces [CS02, BBCPV Crypto'13]

Basically, languages of the form:

$\mathscr{L}$ is a subspace of a vector space $\mathcal{X} = \mathbb{G}^n \approx \mathbb{Z}_p^n$

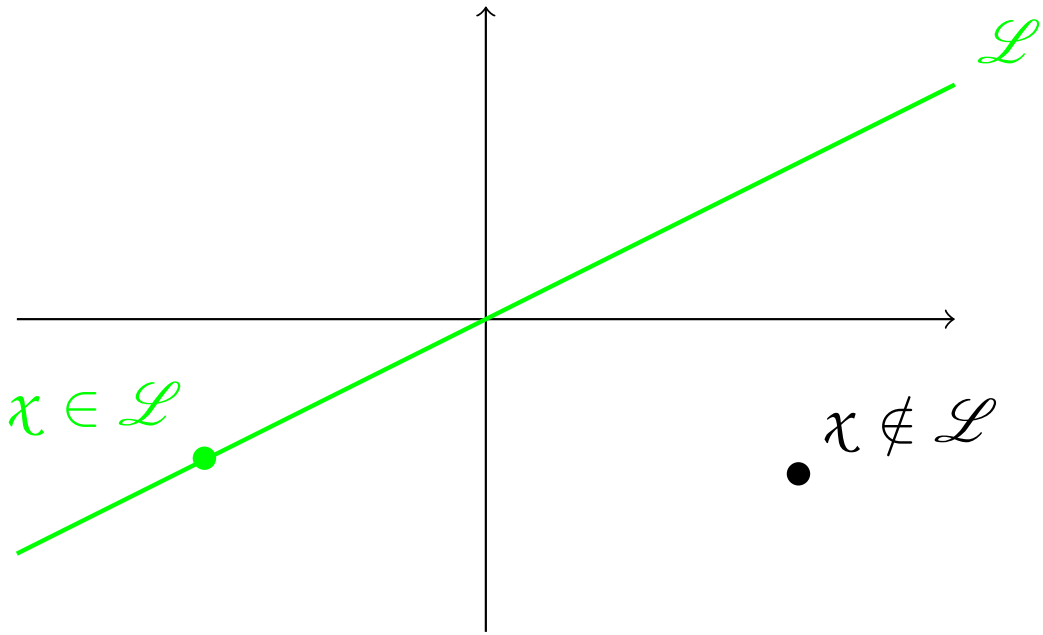


$$\mathscr{L} = \{(\boxed{u}, \boxed{v}) \mid \exists r \in \mathbb{Z}_p, \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} \cdot r\} \quad \subseteq \quad \mathbb{G}^2$$

Diverse Modules

Basically, languages of the form:

$\mathscr{L}$ is a submodule of a module $\mathcal{X} = \mathbb{G}^n \approx \mathbb{Z}_M^n$



Why Diverse Modules?

- Diverse vector spaces / diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$

Why Diverse Modules?

- Diverse vector spaces / diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$

- Diverse modules $\longrightarrow$ SPHF:
  - almost all SPHFs over cyclic groups constructed this way

Why Diverse Modules?

- Diverse vector spaces / diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$
- Diverse modules $\longrightarrow$ SPHF:
  - almost all SPHFs over cyclic groups constructed this way
- Combinations / enhancements
  - conjunction
  - disjunction
  - ...

Why Diverse Modules?

- Diverse vector spaces / diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$
- Diverse modules $\longrightarrow$ SPHF:
  - almost all SPHFs over cyclic groups constructed this way
- Combinations / enhancements
  - conjunction
  - disjunction
  - ...

Conjunctions and Disjunctions [ABP Eurocrypt'15]

$\mathscr{L}$ is a subspace of a vector space $\mathcal{X} \approx \mathbb{Z}_p^n$

- Conjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := \mathscr{L}_1 \times \mathscr{L}_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

Conjunctions and Disjunctions [ABP Eurocrypt'15]

$\mathscr{L}$ is a subspace of a vector space $\mathcal{X} \approx \mathbb{Z}_p^n$

- Conjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := \mathscr{L}_1 \times \mathscr{L}_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

- Disjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := (\mathscr{L}_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times \mathscr{L}_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

Conjunctions and Disjunctions [ABP Eurocrypt'15]

$\mathscr{L}$ is a subspace of a vector space $\mathcal{X} \approx \mathbb{Z}_p^n$

- Conjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := \mathscr{L}_1 \times \mathscr{L}_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

- Disjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := (\mathscr{L}_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times \mathscr{L}_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

But wait!

$\mathscr{L}$ is not a vector space and $\langle \mathscr{L} \rangle = \mathcal{X}$!

Conjunctions and Disjunctions [ABP Eurocrypt'15]

$\mathscr{L}$ is a subspace of a vector space $\mathcal{X} \approx \mathbb{Z}_p^n$

- Conjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := \mathscr{L}_1 \times \mathscr{L}_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

- Disjunction of $\mathscr{L}_1 \subseteq \mathcal{X}_1$ and $\mathscr{L}_2 \subseteq \mathcal{X}_2$:

$$\mathscr{L} := (\mathscr{L}_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times \mathscr{L}_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}$$

But wait!

$\mathscr{L}$ is not a vector space and $\langle \mathscr{L} \rangle = \mathcal{X}$!

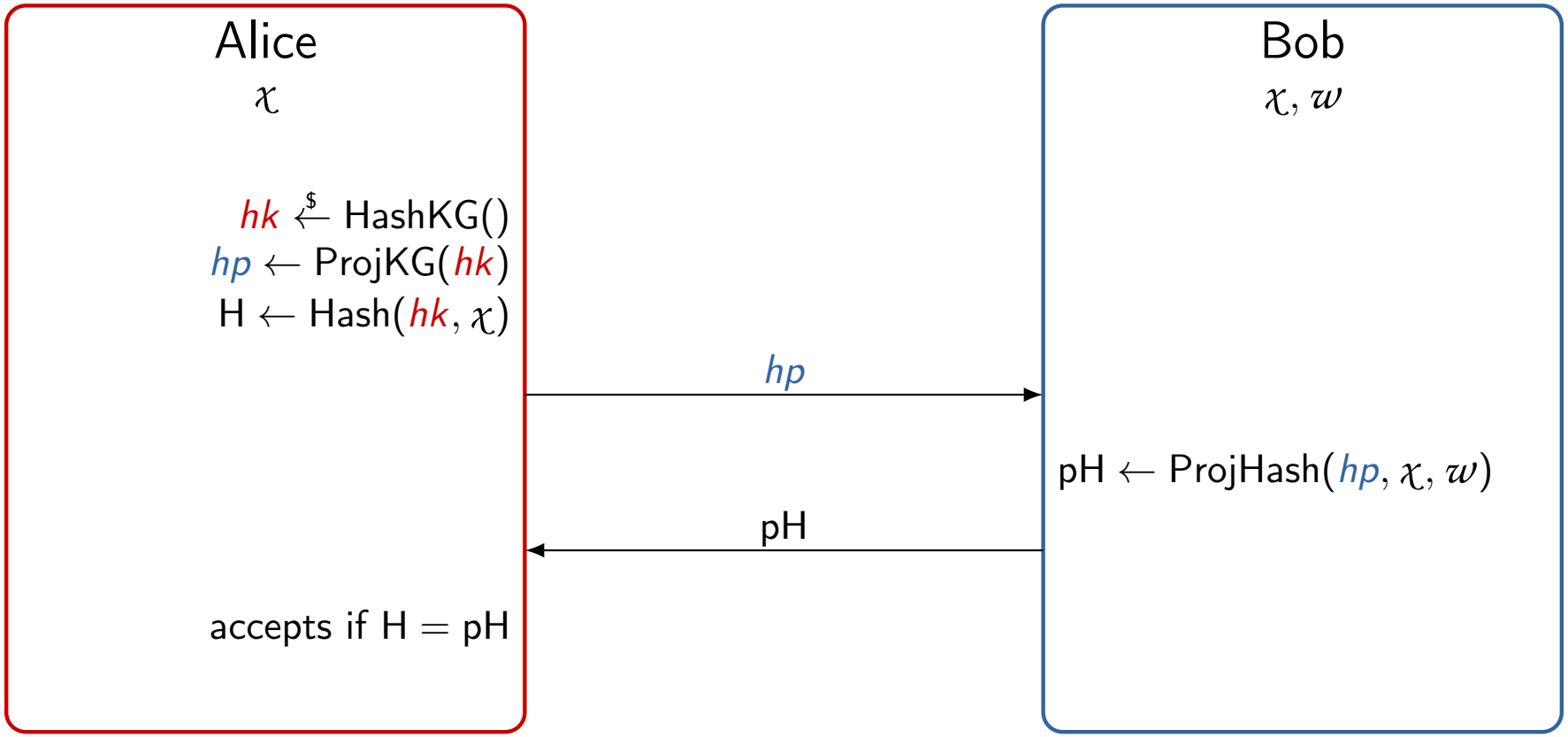
Idea: Tensor Product

$$\mathscr{L} := \langle (\mathscr{L}_1 \otimes \mathcal{X}_2) \cup (\mathcal{X}_1 \otimes \mathscr{L}_2) \rangle \quad \subseteq \quad \mathcal{X}_1 \otimes \mathcal{X}_2 =: \mathcal{X}$$

Direct Applications of SPHFs

Honest-Verifier Zero-Knowledge Proofs

Bob wants to prove to Alice that $x \in \mathcal{L}$.

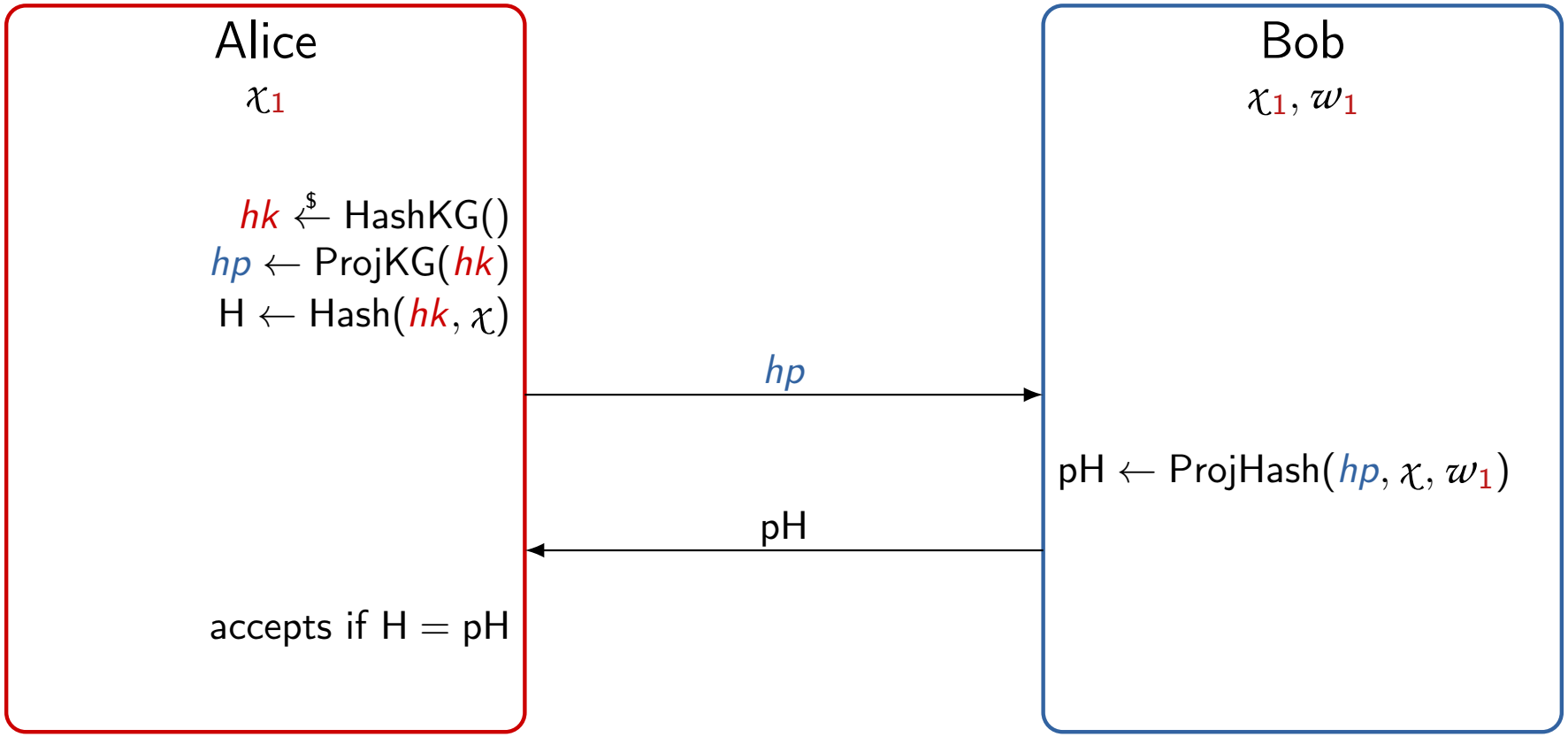


Applications of SPHFs for Disjunction of Languages

Zero-Knowledge Arguments [BBCPV Crypto'13]

Bob wants to prove to Alice that $x_1 \in \mathcal{L}_1$ in zero-knowledge.

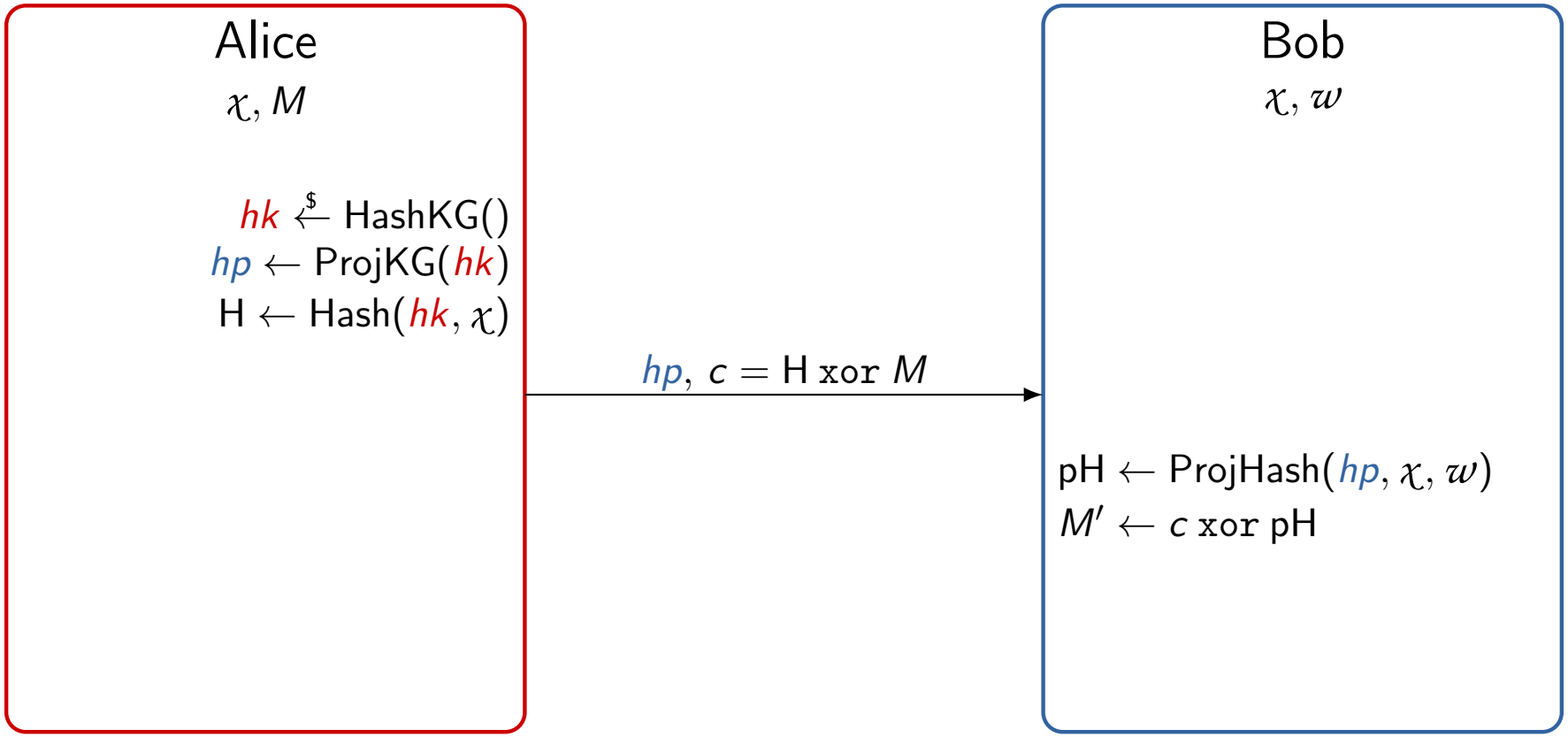
$x = (x_1, x_2)$ — CRS: x_2



Direct Applications of SPHFs

Implicit Arguments

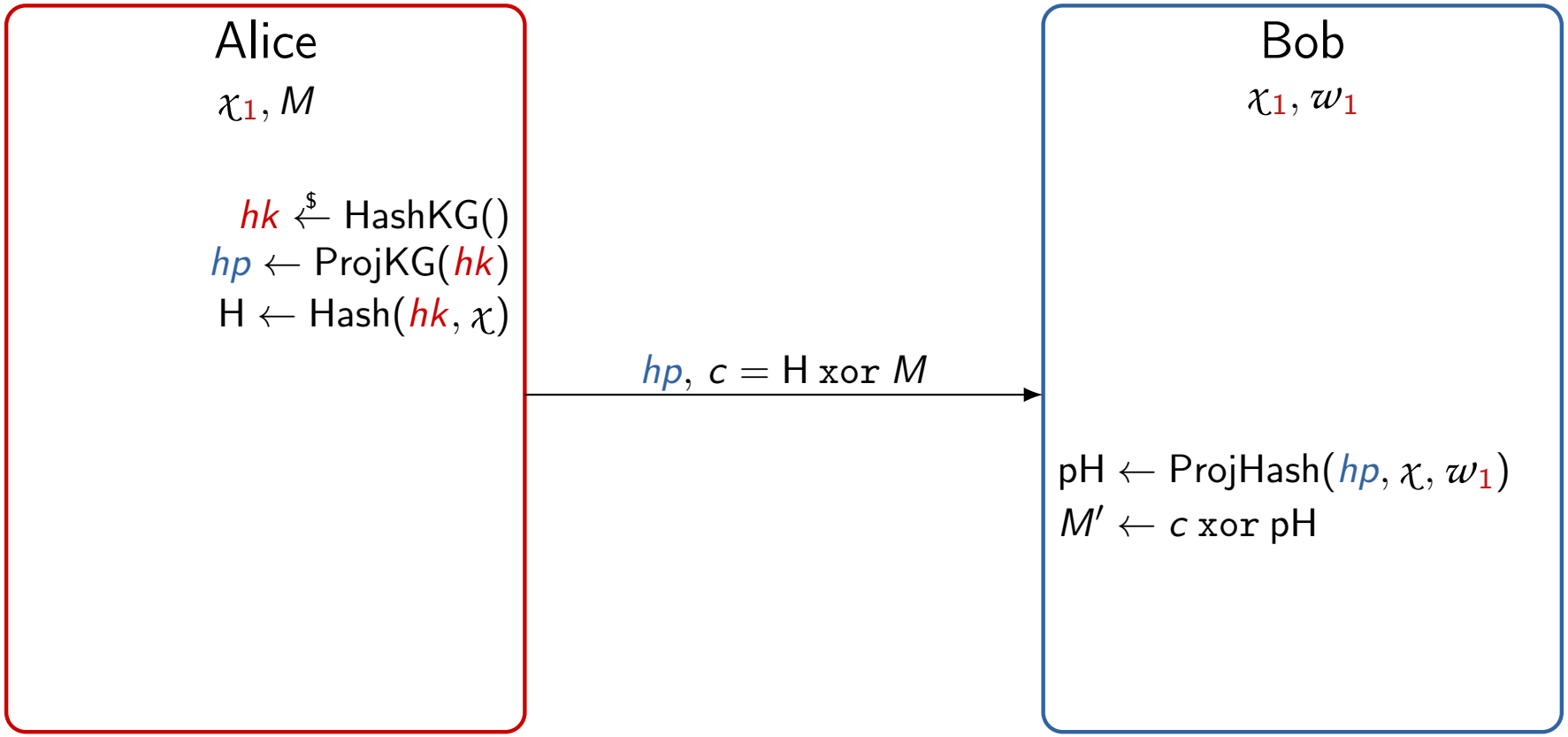
Alice wants to send M to Bob if $x \in \mathcal{L}$.



Applications of SPHFs for Disjunction of Languages

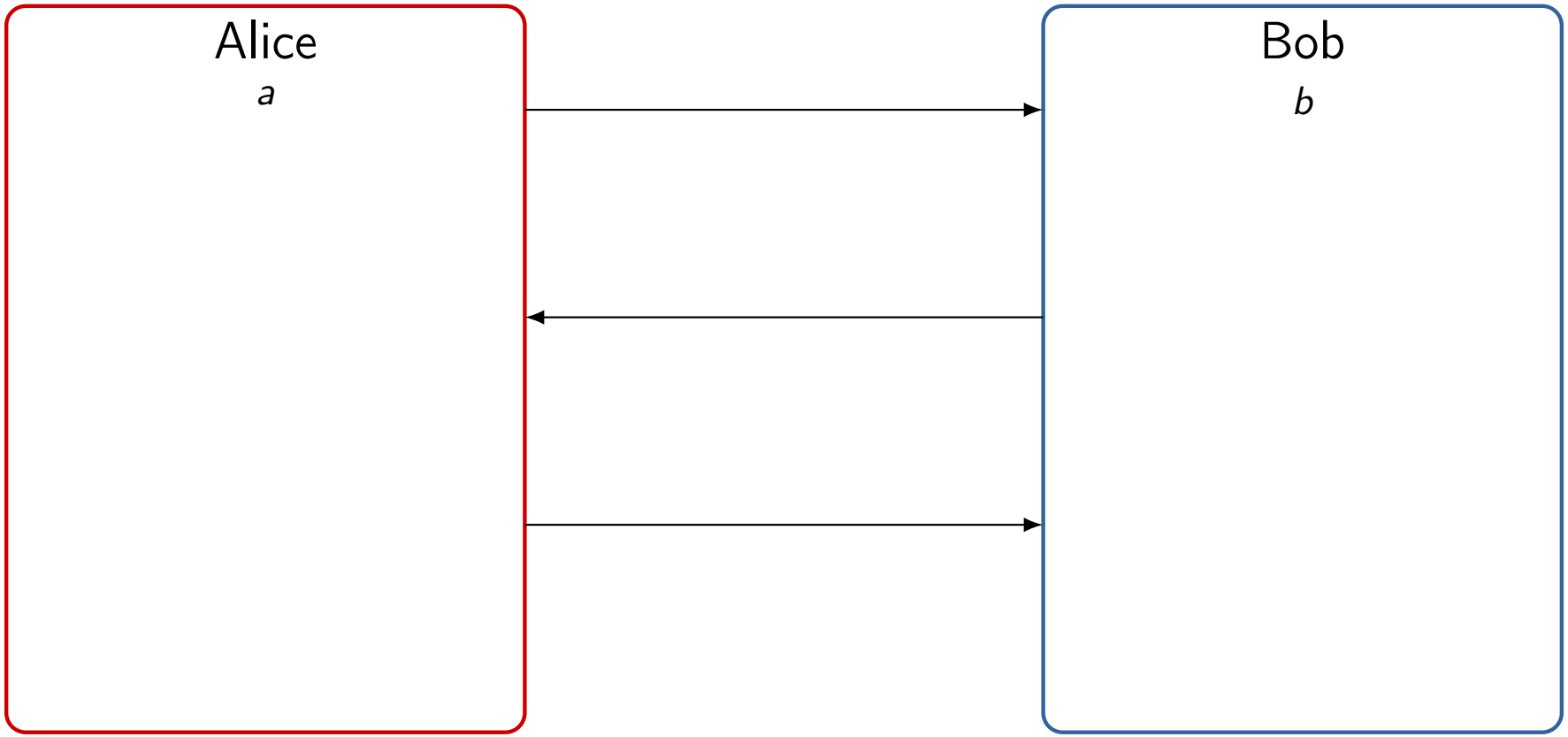
Implicit Zero-Knowledge Arguments (iZK) [BCPW, Crypto'15]

Alice wants to send M to Bob if $x_1 \in \mathcal{L}_1$
in zero-knowledge. $x = (x_1, x_2)$ — CRS: x_2



Application to Two-Party Secure Computation

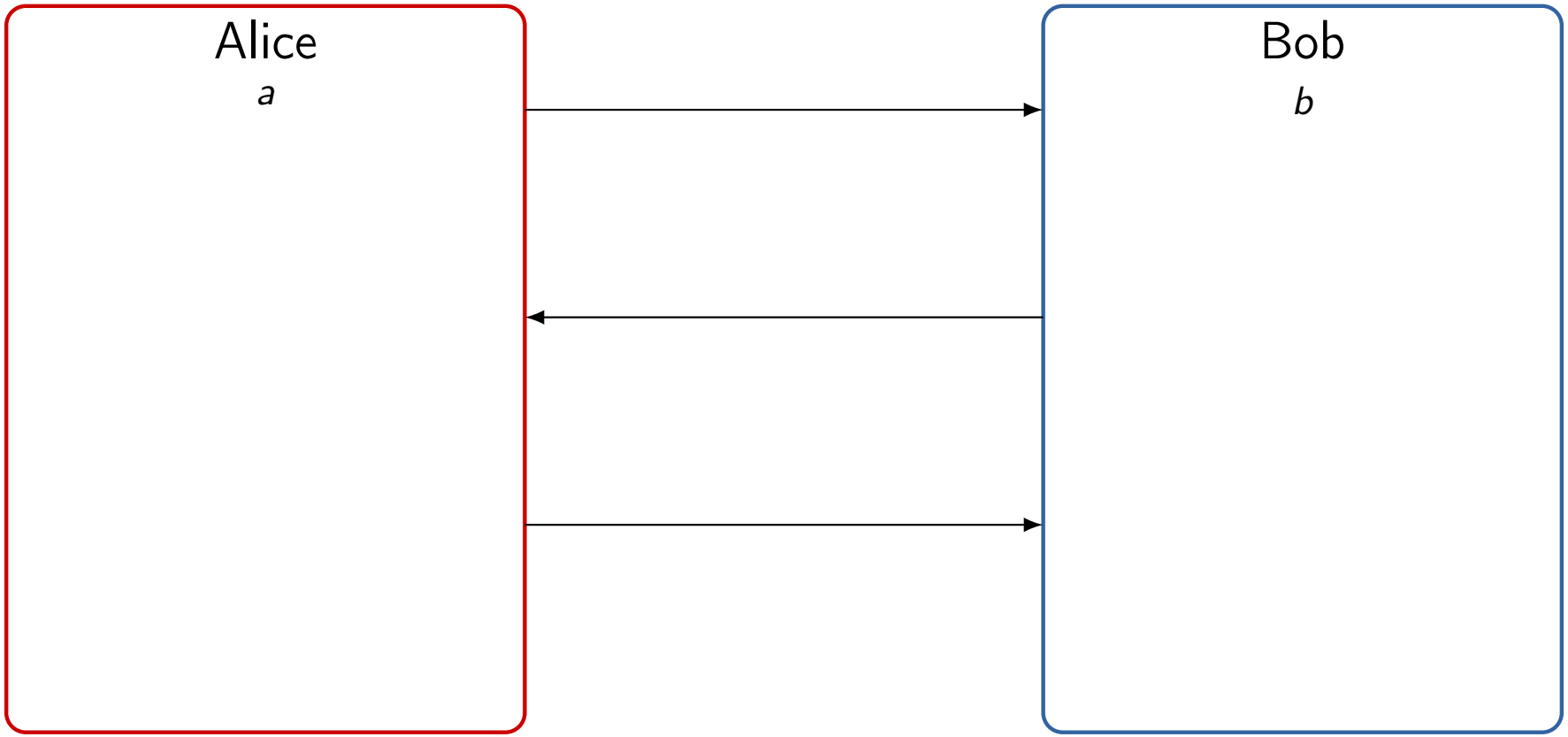
Alice has input a, Bob has input b, Bob wants to get $f(a, b)$



Application to Two-Party Secure Computation

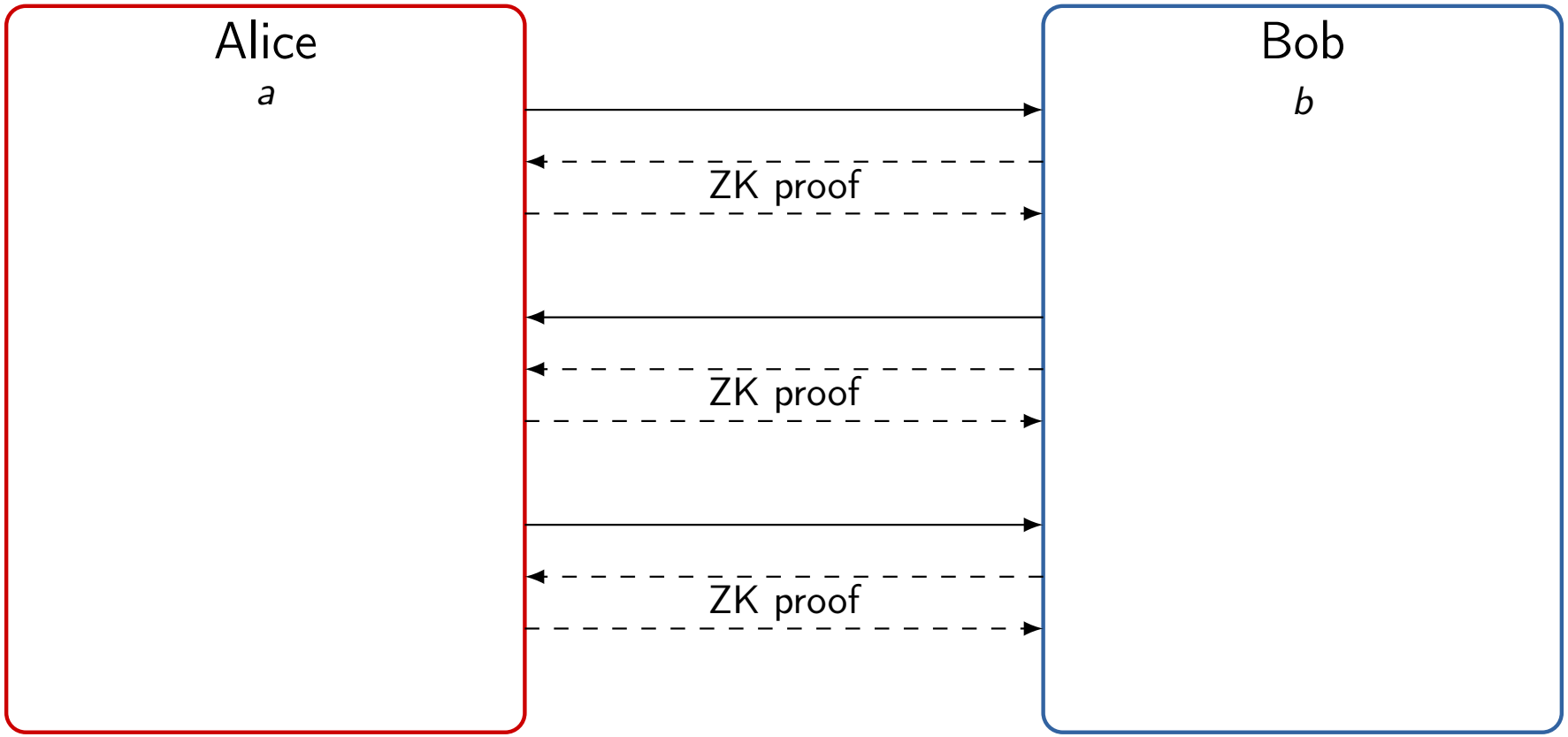
Alice has input a, Bob has input b, Bob wants to get $f(a, b)$

Honest but curious $\longrightarrow$ malicious



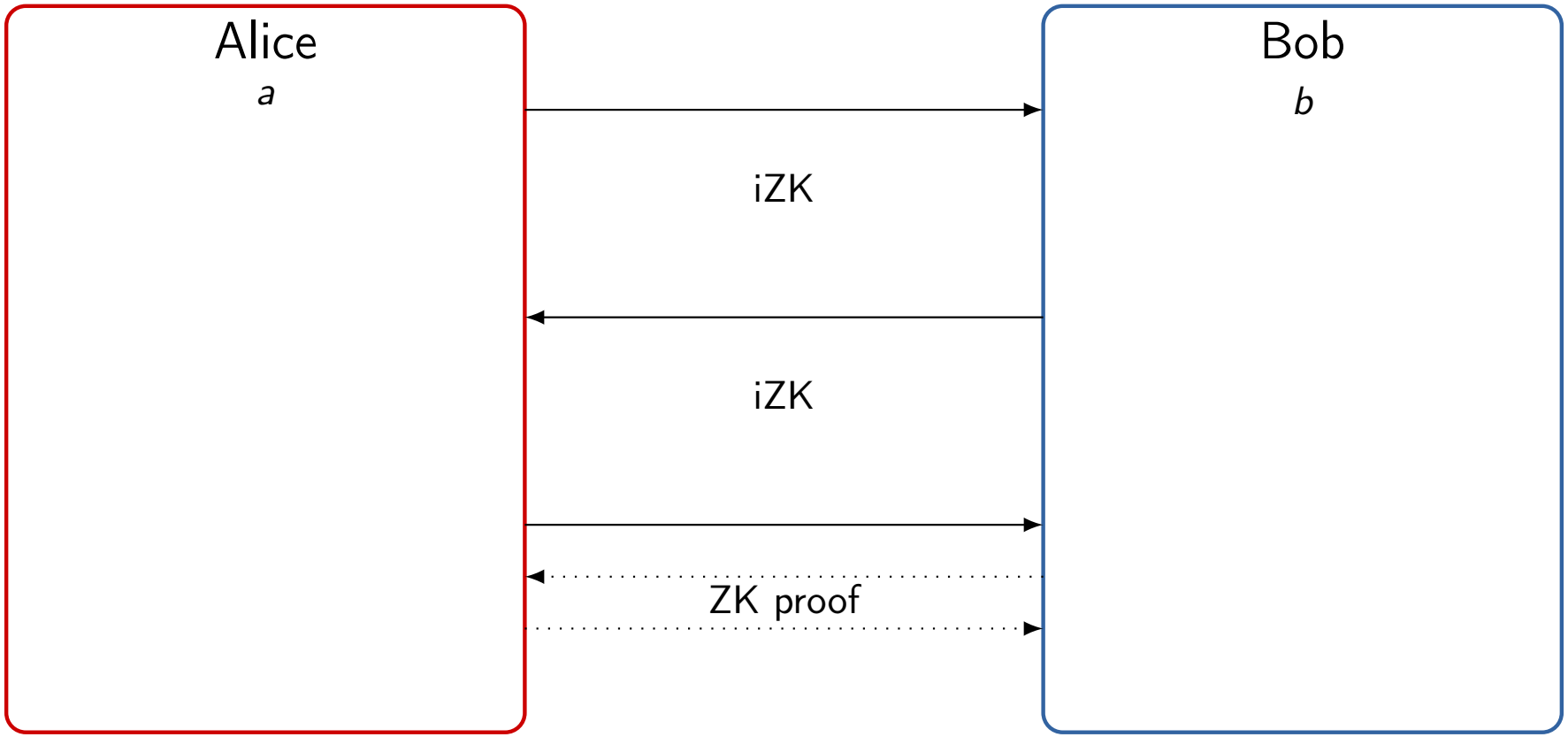
Application to Two-Party Secure Computation

Alice has input a, Bob has input b, Bob wants to get $f(a, b)$
Honest but curious $\longrightarrow$ malicious



Application to Two-Party Secure Computation

Alice has input a, Bob has input b, Bob wants to get $f(a, b)$
Honest but curious $\longrightarrow$ malicious



Conclusion

- Diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$

Conclusion

- Diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$

- Combinations / enhancements:
  - conjunction, disjunction, "t-smoothness"

Conclusion

- Diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$
- Combinations / enhancements:
  - conjunction, disjunction, "t-smoothness"
- Applications:
  - iZK / TSPHFs = zero-knowledge versions of SPHFs
    $\longrightarrow$ lightweight alternative to zero-knowledge arguments
    e.g., two-party secure computation
  - SPHFs $\longrightarrow$ PAKE, OT, HVZK, WE, ...
  - constant-size QA-NIZK

Thank you for your attention! Questions?

- Diverse modules:
  - algebraic representations of languages
  - language $\mathscr{L}$ submodule of $\mathscr{X}$
- Combinations / enhancements:
  - conjunction, disjunction, "t-smoothness"
- Applications:
  - iZK / TSPHFs = zero-knowledge versions of SPHFs
    $\longrightarrow$ lightweight alternative to zero-knowledge arguments
    e.g., two-party secure computation
  - SPHFs $\longrightarrow$ PAKE, OT, HVZK, WE, ...
  - constant-size QA-NIZK

References I

Michel Abdalla, Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, and David Pointcheval.

SPHF-friendly non-interactive commitments.

In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 214–234. Springer, Heidelberg, December 2013.

Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud.

New techniques for SPHFs and efficient one-round PAKE protocols.

In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 449–475. Springer, Heidelberg, August 2013.

References II

Ronald Cramer and Victor Shoup.

A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack.

In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998.

Ronald Cramer and Victor Shoup.

Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption.

In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002.

Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters.

Witness encryption and its applications.

In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

References III

Rosario Gennaro and Yehuda Lindell.

A framework for password-based authenticated key exchange.

In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 524–543. Springer, Heidelberg, May 2003.

`http://eprint.iacr.org/2003/032.ps.gz`.

Charanjit S. Jutla and Arnab Roy.

Relatively-sound NIZKs and password-based key-exchange.

In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 485–503. Springer, Heidelberg, May 2012.

References IV

Charanjit S. Jutla and Arnab Roy.

Dual-system simulation-soundness with applications to UC-PAKE and more.

Cryptology ePrint Archive, Report 2014/805, 2014.

http://eprint.iacr.org/2014/805.

Yael Tauman Kalai.

Smooth projective hashing and two-message oblivious transfer.

In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 78–95. Springer, Heidelberg, May 2005.

Jonathan Katz, Rafail Ostrovsky, and Moti Yung.

Efficient password-authenticated key exchange using human-memorable passwords.

In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 475–494. Springer, Heidelberg, May 2001.

References V

Jonathan Katz and Vinod Vaikuntanathan.

Round-optimal password-based authenticated key exchange.

In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 293–310. Springer, Heidelberg, March 2011.