Disjunctions for Hash Proof Systems: New Constructions and Applications

Michel Abdalla, *Fabrice Benhamouda*, and David Pointcheval

École Normale Supérieure, CNRS, INRIA, PSL, Paris, France

European Research Council

Established by the European Commission

Eurocrypt 2015, Sofia, Bulgaria
Monday, April 27

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → 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, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → IND-CCA encryption scheme [CS98],

Simple languages
DDH, Paillier/DCR, QR, ...

- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11],
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

Introduction

Hash Proof System / Smooth Projective Hash Function (SPHF)

- Introduced by Cramer and Shoup [CS02]
  - → Implicit designated-verifier proofs,
  - → IND-CCA encryption scheme [CS98],

Simple languages
DDH, Paillier/DCR, QR, ...

- Applications:
  - Password-Authenticated Key Exchange (PAKE) [KOY01, GL03, KV11],
  - Oblivious Transfer [Kal05, ABB+13]
  - Relatively-Sound / Dual-System NIZK [JR12, JR14a]
  - Zero-Knowledge Arguments [BBC+13]
  - Witness Encryption [GGSW13]

More Complex Languages

Introduction

Languages

Over a cyclic group $\mathbb{G}$ of prime order p (g, h: generators):

[CS02] DDH:

$$\{(u, e) \mid \exists r \in \mathbb{Z}_p,\ u = g^r \text{ and } e = h^r\};$$

[ACP09] Conjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ and } C_2 \in L_2\} = L_1 \times L_2;$$

[BBC+13] ElGamal/Cramer-Shoup-like ciphertexts of $M_1, \ldots, M_n \in \mathbb{G}$ satisfying:

- a system of multi-exponentiation equations;
- a system of (quadratic) pairing equations
  more expressive than Groth-Sahai NIZK [GS08];

Introduction

Languages

Over a cyclic group $\mathbb{G}$ of prime order p (g, h: generators):

[CS02] ElGamal ciphertexts of $M \in \mathbb{G}$:

$$\{(u, e) \mid \exists r \in \mathbb{Z}_p,\ u = g^r \text{ and } e = h^r \cdot M\};$$

[ACP09] Conjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ and } C_2 \in L_2\} = L_1 \times L_2;$$

[BBC+13] ElGamal/Cramer-Shoup-like ciphertexts of $M_1, \ldots, M_n \in \mathbb{G}$ satisfying:

- a system of multi-exponentiation equations;
- a system of (quadratic) pairing equations
  more expressive than Groth-Sahai NIZK [GS08];

Introduction
Languages

Over a cyclic group $\mathbb{G}$ of prime order p (g, h: generators):

[CS02] ElGamal ciphertexts of $M \in \mathbb{G}$:

$$\{(u, e) \mid \exists r \in \mathbb{Z}_p,\ u = g^r \text{ and } e = h^r\};$$

[ACP09] Conjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ and } C_2 \in L_2\} = L_1 \times L_2;$$

[BBC+13] ElGamal/Cramer-Shoup-like ciphertexts of $M_1, \ldots, M_n \in \mathbb{G}$ satisfying:

- a system of multi-exponentiation equations;
- a system of (quadratic) pairing equations more expressive than Groth-Sahai NIZK [GS08];

Introduction

Languages

Over a cyclic group $\mathbb{G}$ of prime order p (g, h: generators):

[CS02] ElGamal ciphertexts of $M \in \mathbb{G}$:

$$\{(u, e) \mid \exists r \in \mathbb{Z}_p,\ u = g^r \text{ and } e = h^r\};$$

[ACP09] Conjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ and } C_2 \in L_2\} = L_1 \times L_2;$$

[BBC+13] ElGamal/Cramer-Shoup-like ciphertexts of $M_1, \ldots, M_n \in \mathbb{G}$ satisfying:

- a system of multi-exponentiation equations;
- a system of (quadratic) pairing equations more expressive than Groth-Sahai NIZK [GS08];

Contributions

- SPHF for disjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ or } C_2 \in L_2\};$$

(under some condition and with bilinear groups)

- + other tools for SPHF: 2-smoothness, PrPHF, ...
- Applications:
  - Quasi-Adaptive NIZK [JR13]
    - 1 group element (under DDH), as [JR14b]
    - even for one-time simulation-soundness
    - application: threshold Cramer-Shoup-like encryption scheme
  - First one-round PAKE for $k \geq 3$ players;
  - Link with homomorphic signatures [LPJY13];
  - New construction of Trapdoor SPHF (TSPHF) [BBC+13] (zero-knowledge variant of SPHF).

Contributions

- SPHF for disjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ or } C_2 \in L_2\};$$

(under some condition and with bilinear groups)

- + other tools for SPHF: 2-smoothness, PrPHF, ...
- Applications:
  - Quasi-Adaptive NIZK [JR13]
    - 1 group element (under DDH), as [JR14b]
    - even for one-time simulation-soundness
    - application: threshold Cramer-Shoup-like encryption scheme
  - First one-round PAKE for $k \geq 3$ players;
  - Link with homomorphic signatures [LPJY13];
  - New construction of Trapdoor SPHF (TSPHF) [BBC+13] (zero-knowledge variant of SPHF).

Contributions

- SPHF for disjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ or } C_2 \in L_2\};$$

(under some condition and with bilinear groups)

- + other tools for SPHF: 2-smoothness, PrPHF, ...
- Applications:
  - Quasi-Adaptive NIZK [JR13]
    - 1 group element (under DDH), as [JR14b]
    - even for one-time simulation-soundness
    - application: threshold Cramer-Shoup-like encryption scheme
  - First one-round PAKE for $k \geq 3$ players;
  - Link with homomorphic signatures [LPJY13];
  - New construction of Trapdoor SPHF (TSPHF) [BBC+13] (zero-knowledge variant of SPHF).

Contributions

- SPHF for disjunction of two languages L_1 and L_2:

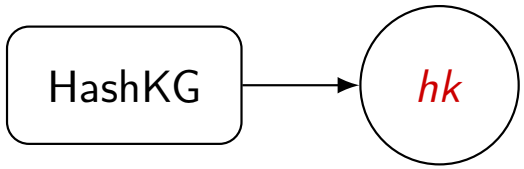
$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ or } C_2 \in L_2\};$$

(under some condition and with bilinear groups)

- + other tools for SPHF: 2-smoothness, PrPHF, ...
- Applications:
  - Quasi-Adaptive NIZK [JR13]
    - 1 group element (under DDH), as [JR14b]
    - even for one-time simulation-soundness
    - application: threshold Cramer-Shoup-like encryption scheme
  - First one-round PAKE for $k \geq 3$ players;
  - Link with homomorphic signatures [LPJY13];
  - New construction of Trapdoor SPHF (TSPHF) [BBC+13] (zero-knowledge variant of SPHF).

SPHF
Definition

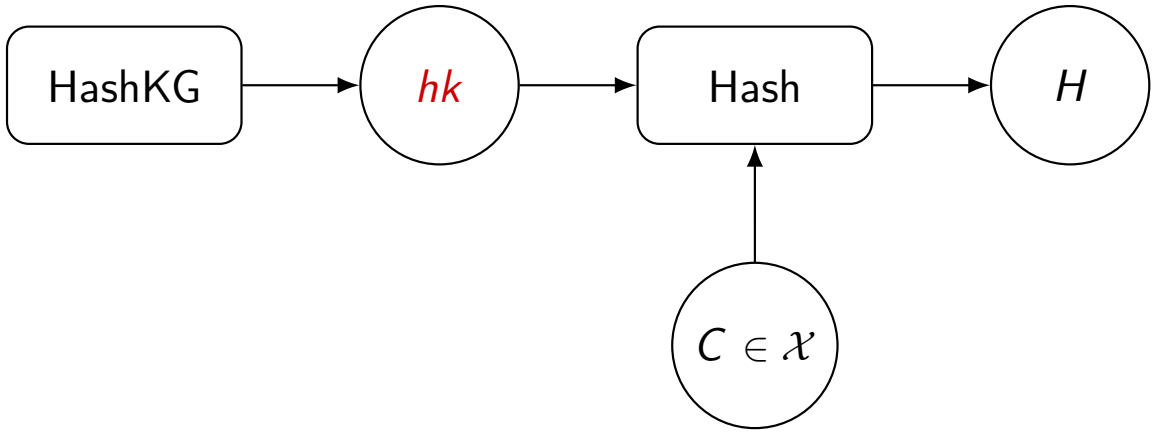
L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1.$



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

SPHF
Definition

L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1$.

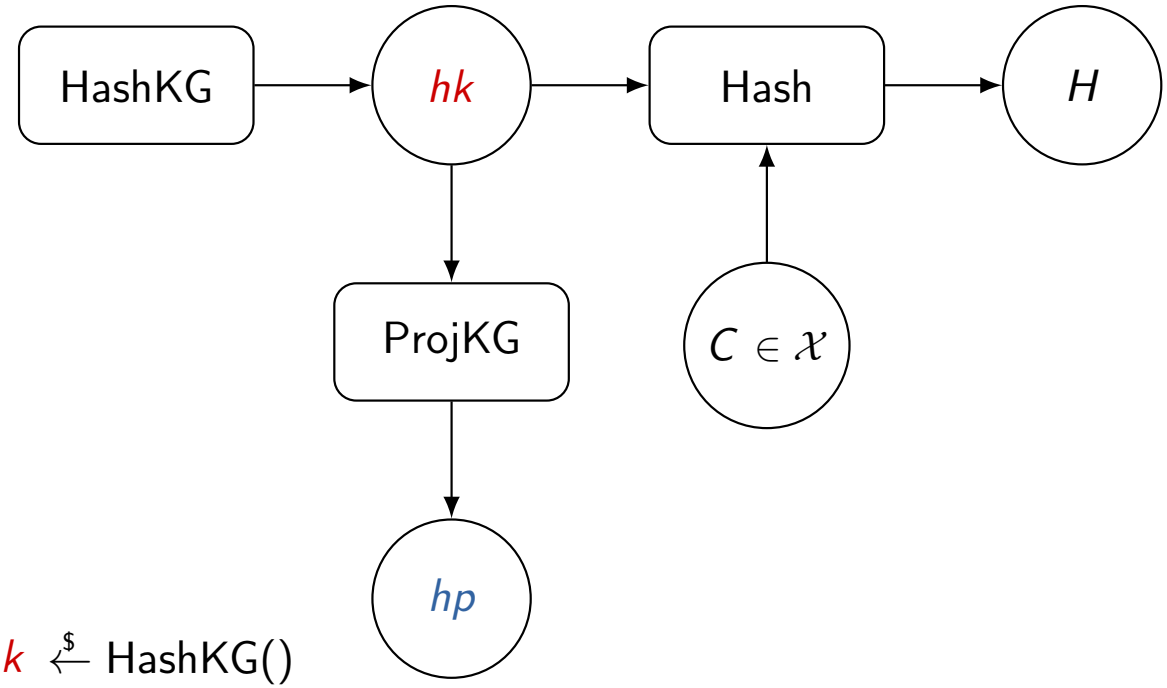


$hk \xleftarrow{\$} \mathsf{HashKG}()$

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

SPHF
Definition

L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1.$



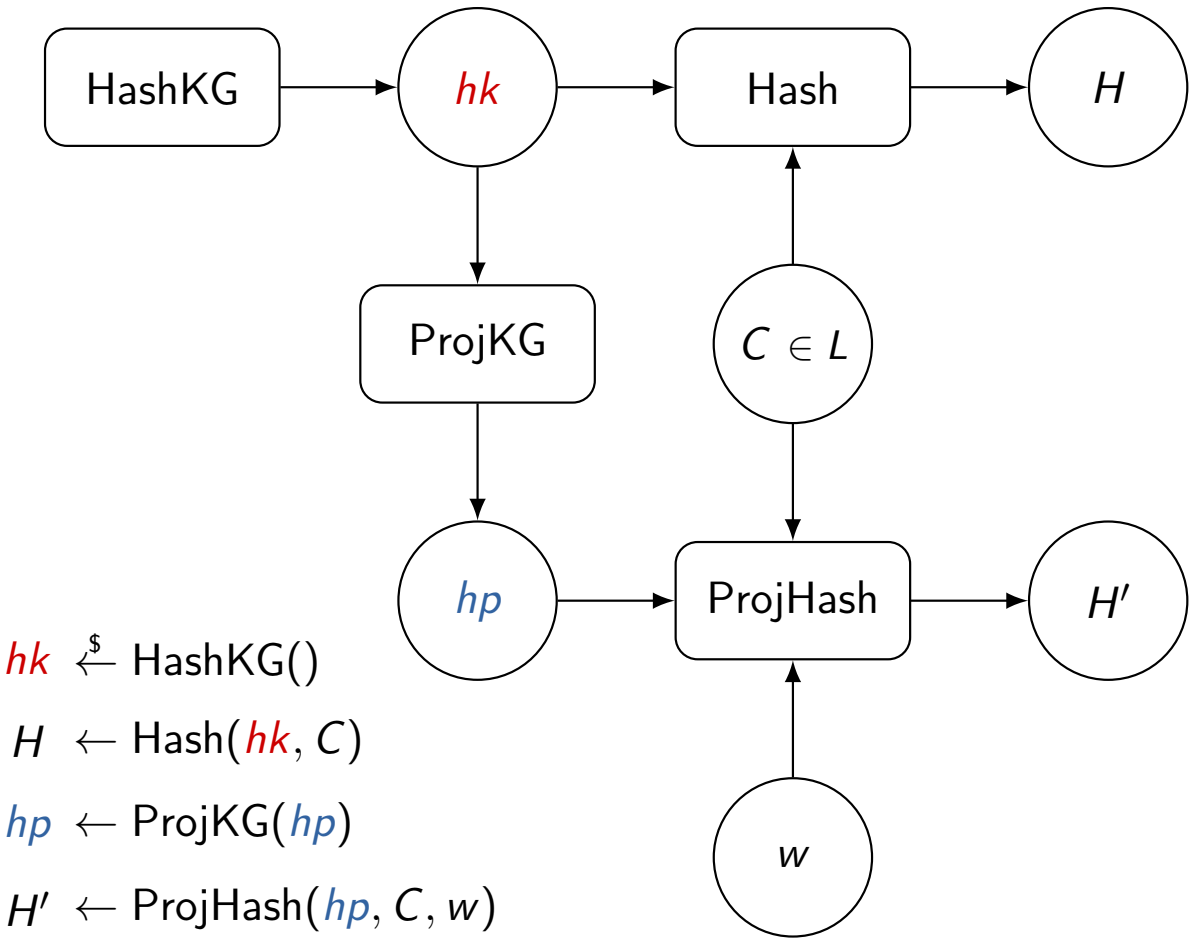
$hk \xleftarrow{\$} \text{HashKG}()$

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

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

SPHF
Definition

L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1.$



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

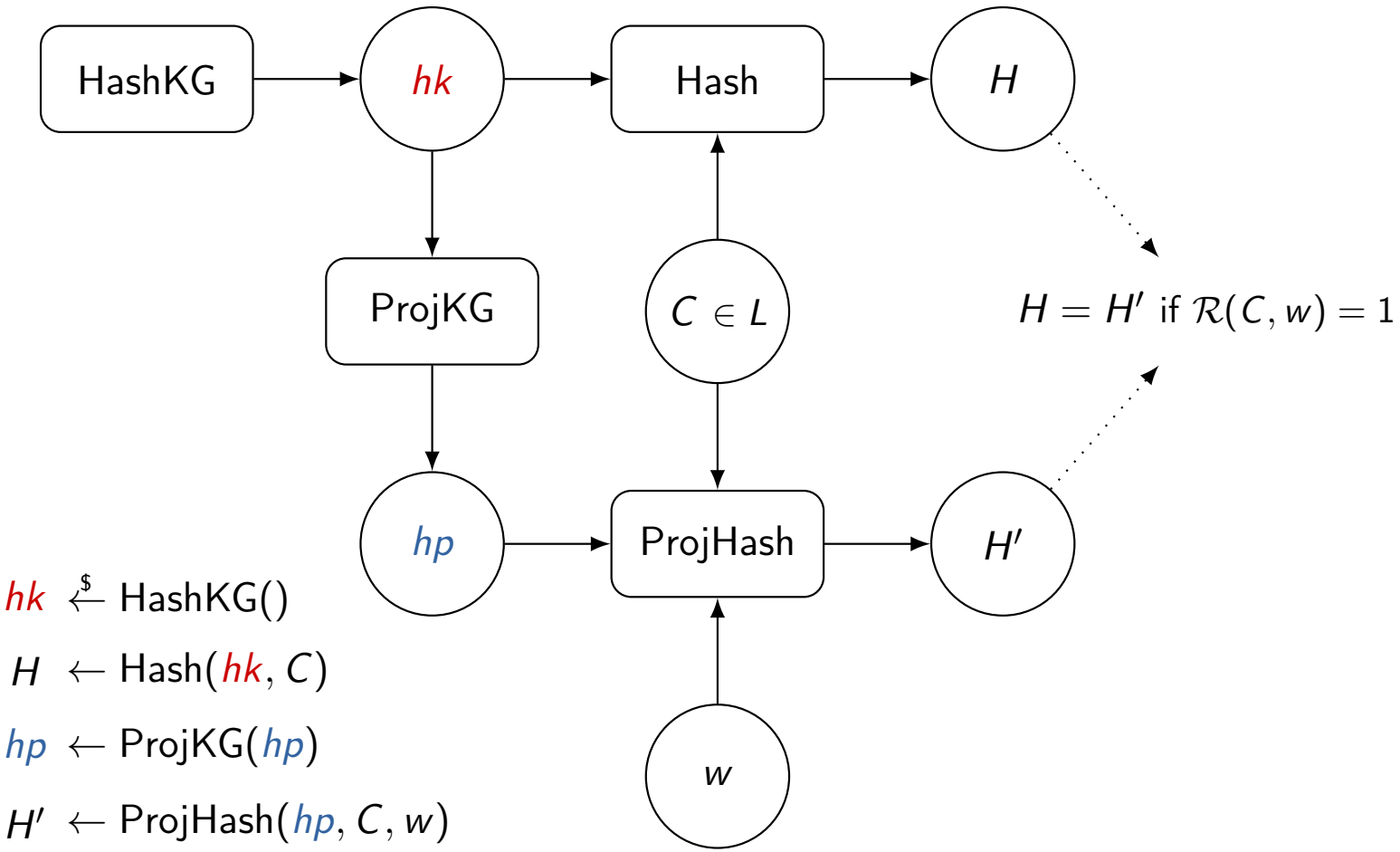
$H \leftarrow \text{Hash}(hk, C)$

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

$H' \leftarrow \text{ProjHash}(hp, C, w)$

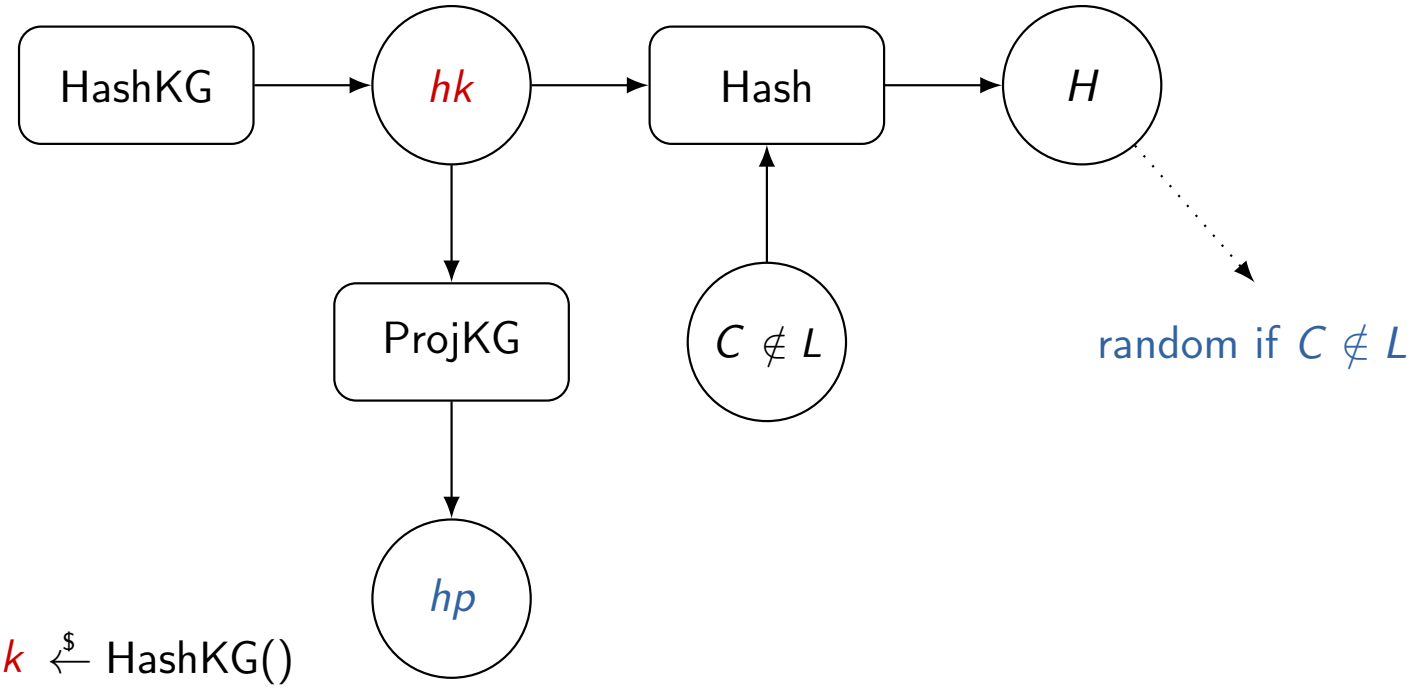
SPHF
Definition

L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1.$



SPHF
Definition

L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1.$



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

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

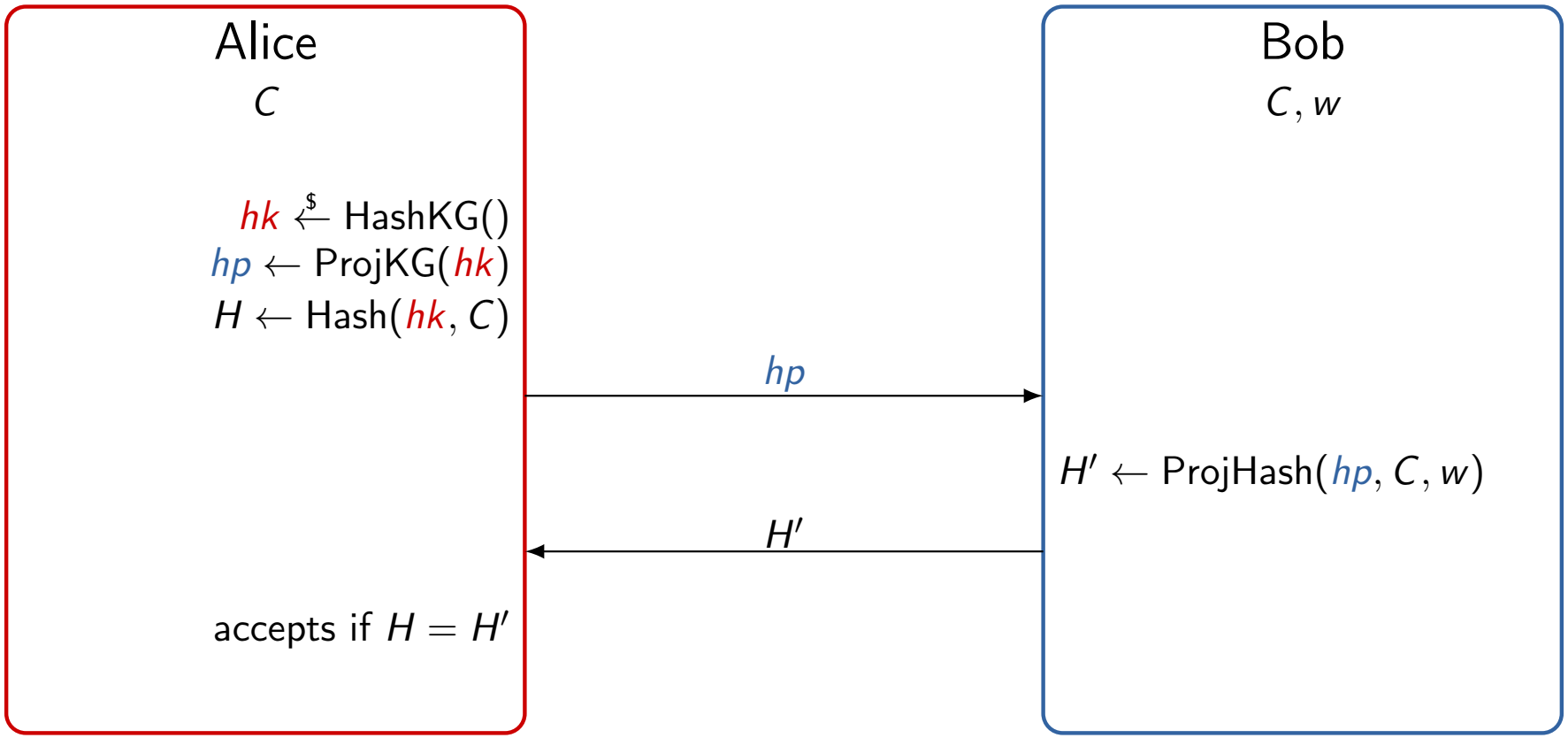
$hp \leftarrow \text{ProjKG}(hp)$

$H' \leftarrow \text{ProjHash}(hp, C, w)$

Direct Applications of SPHF

Honest-Verifier Zero-Knowledge Proof

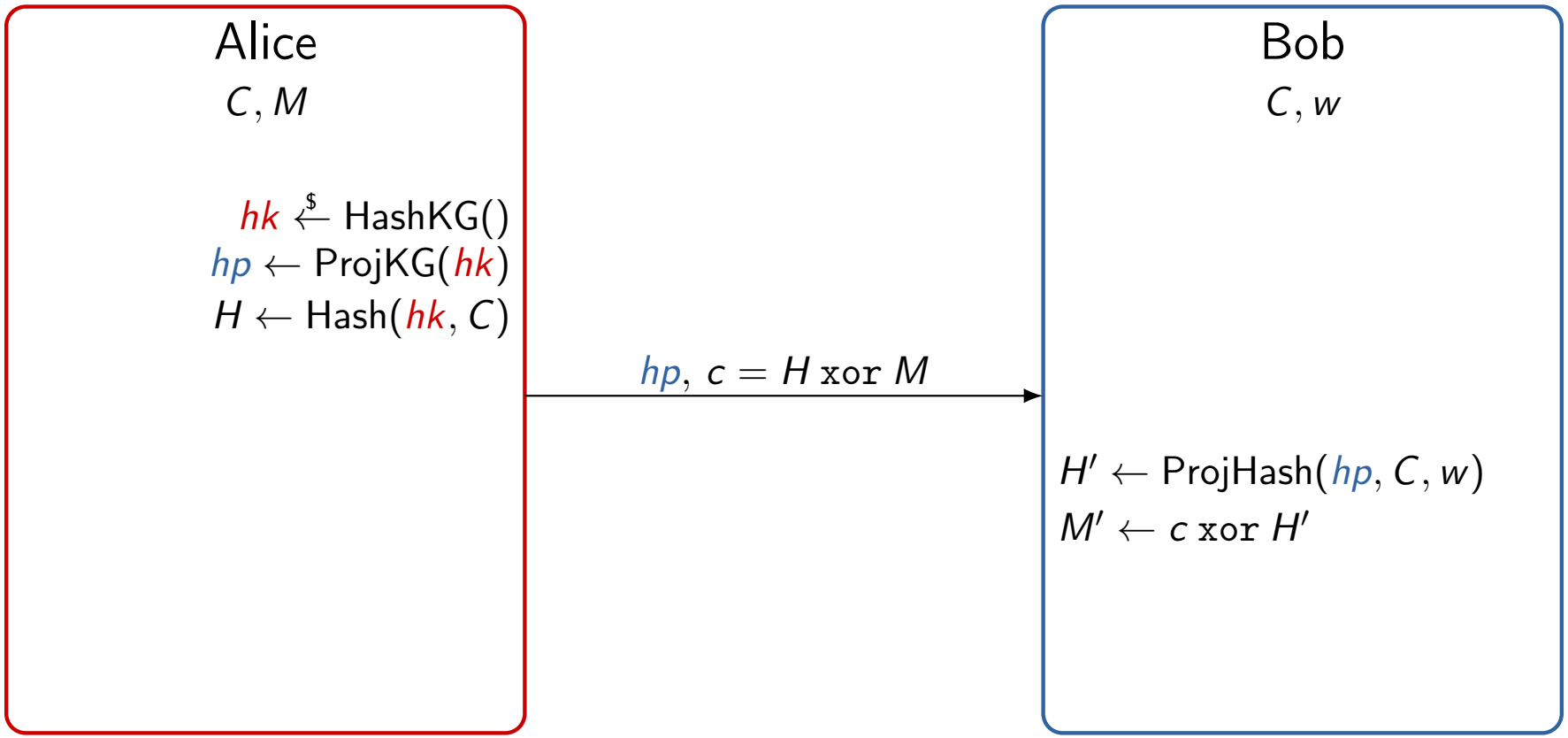
Bob wants to prove to Alice that $C \in L$.



Direct Applications of SPHF

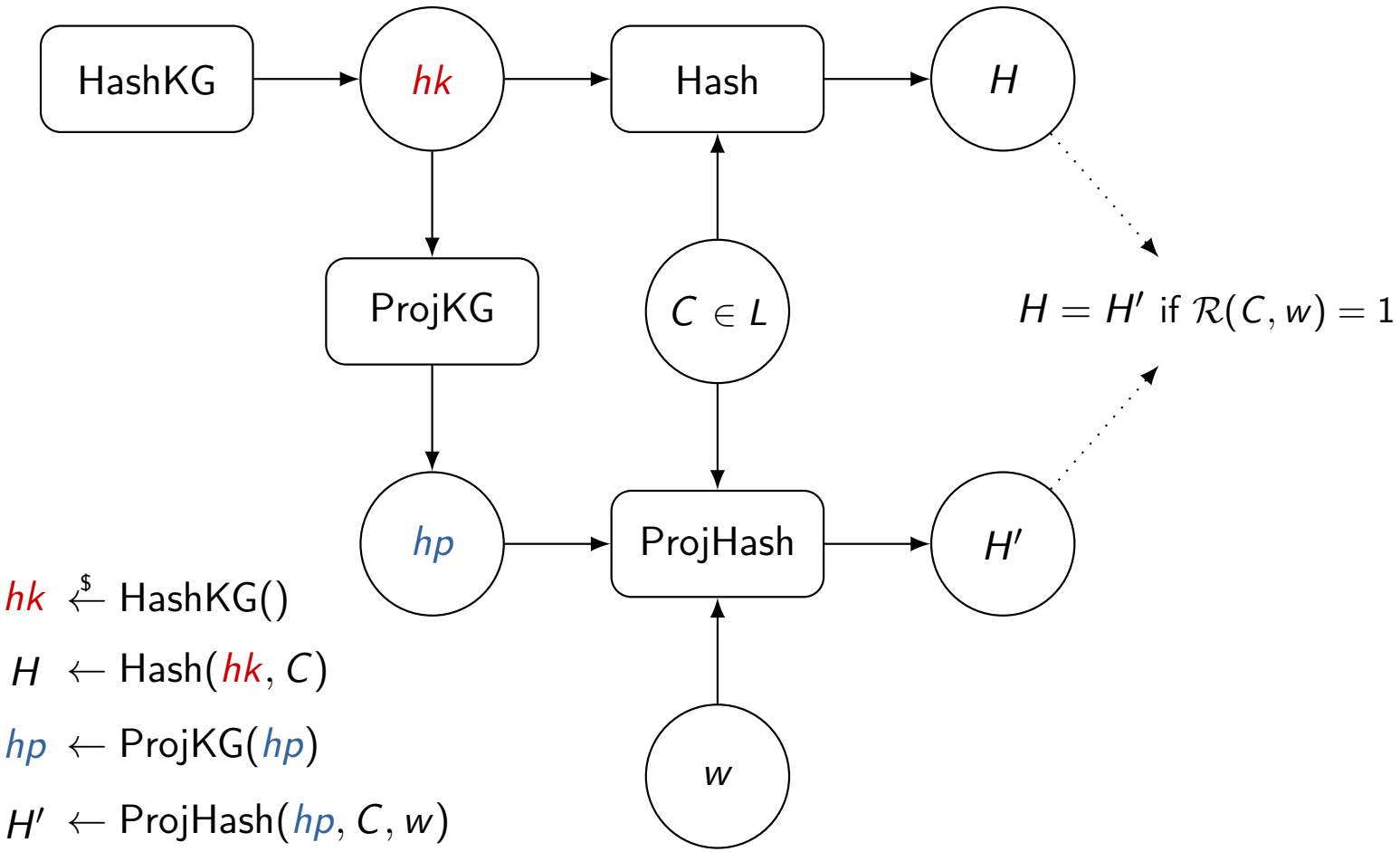
Implicit Argument / Witness Encryption

Alice wants to send M to Bob if $C \in L$.



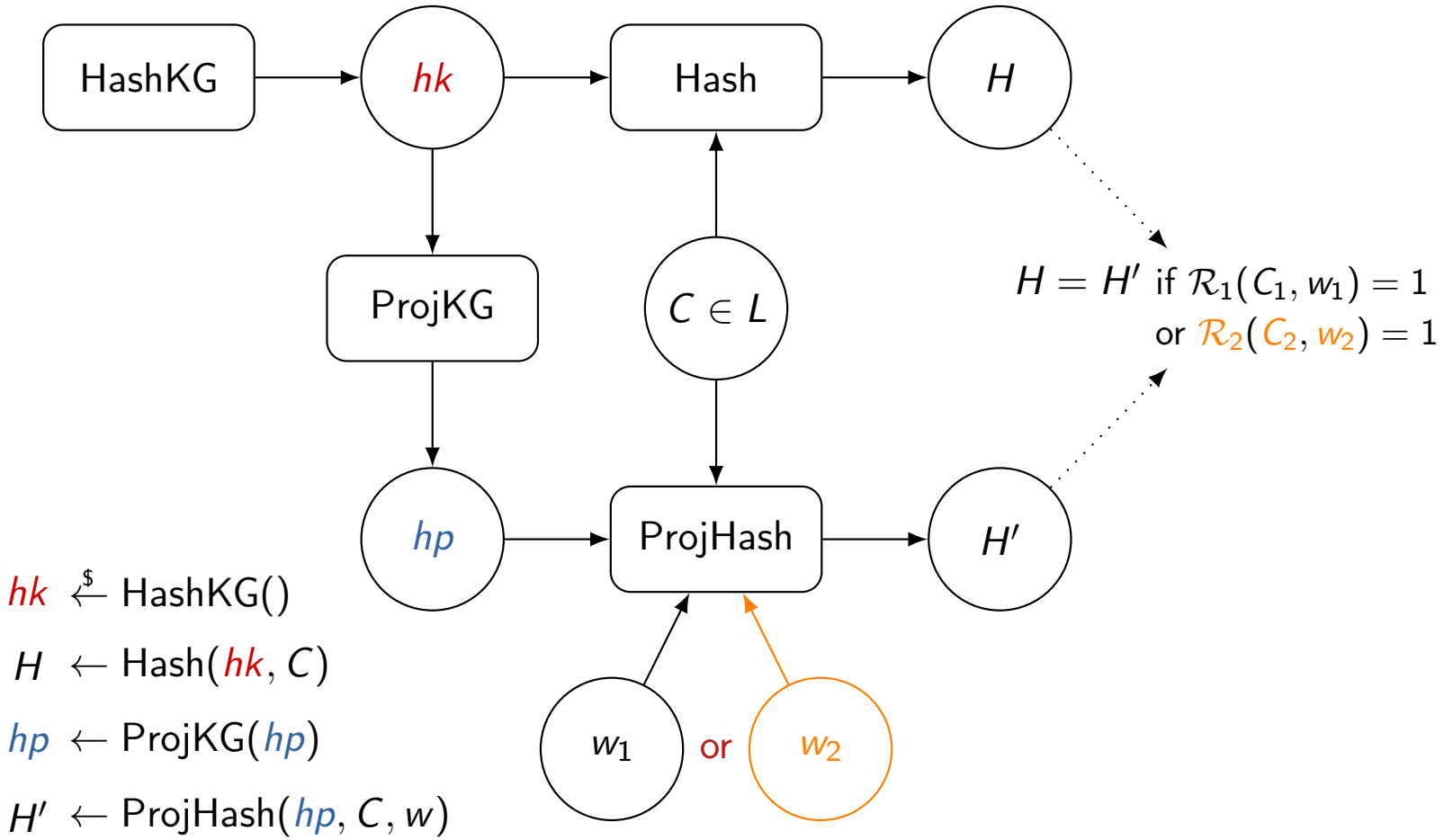
SPHF
Definition

L: NP language, i.e.: $C \in L \subseteq \mathcal{X} \iff \exists w, \mathcal{R}(C, w) = 1.$



Disjunction

$$C = (C_1, C_2) \in L \iff C_1 \in L_1 \text{ or } C_2 \in L_2$$
$$\iff \exists w_1, \mathcal{R}_1(C_1, w_1) = 1 \text{ or } \exists w_2, \mathcal{R}_2(C_2, w_2) = 1.$$



How to use that? Why is it useful?

In all applications except PAKE:

- L_1:
  - original language;
  - word C_1 for the proof;
- L_2:
  - used to add functionnality;
  - word C_2 in the CRS;
  - DDH for example

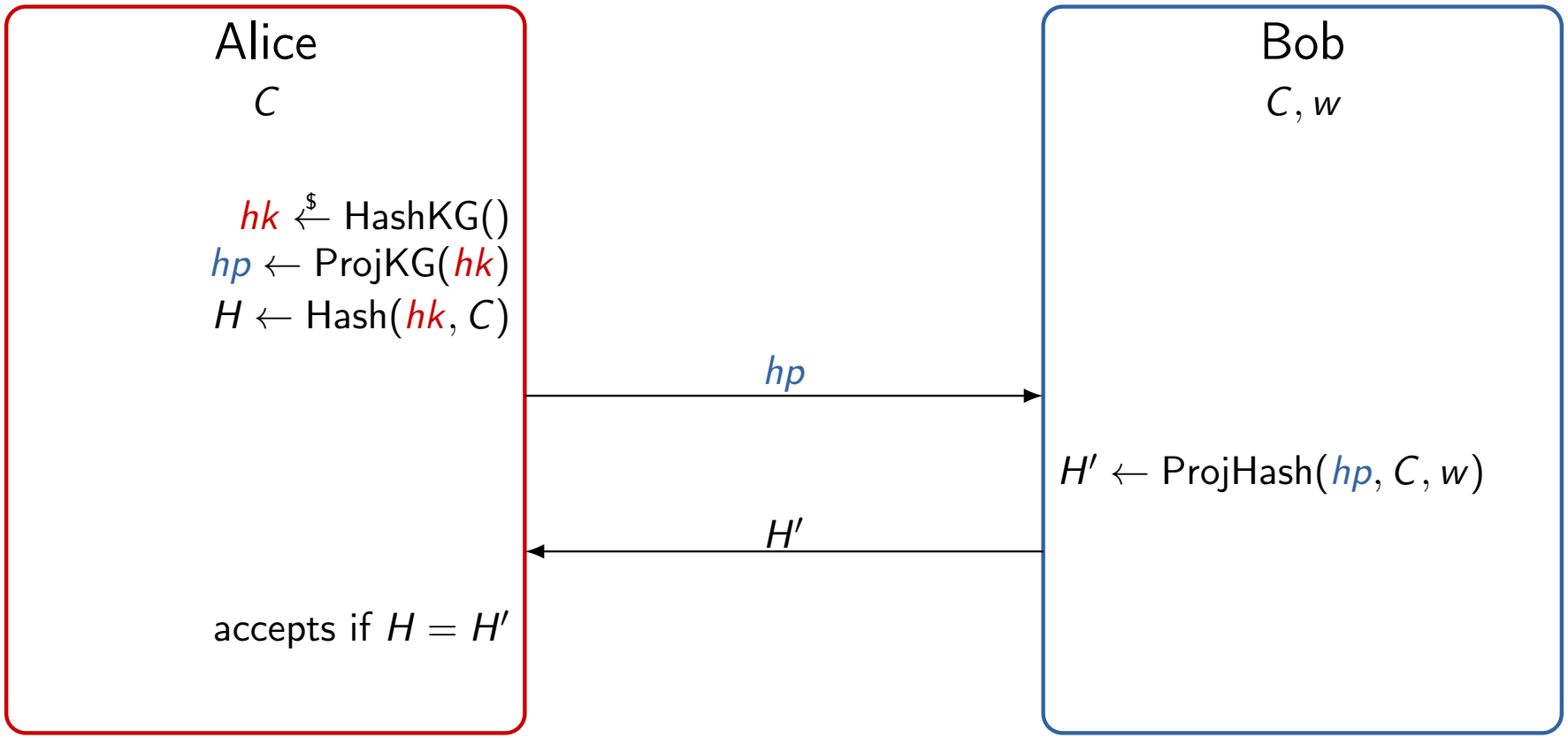
$$\{C_2 = (u, e) \mid \exists r \in \mathbb{Z}_p,\ u = g^r \text{ and } e = h^r\};$$

  or k-Lin or any MDDH assumption;
- 3 ways to compute H:
  - knowing hk;
  - knowing hp and witness w_1 for C_1;
  - knowing hp and witness w_2 for C_2.

Direct Applications of SPHF

Honest-Verifier Zero-Knowledge Proof

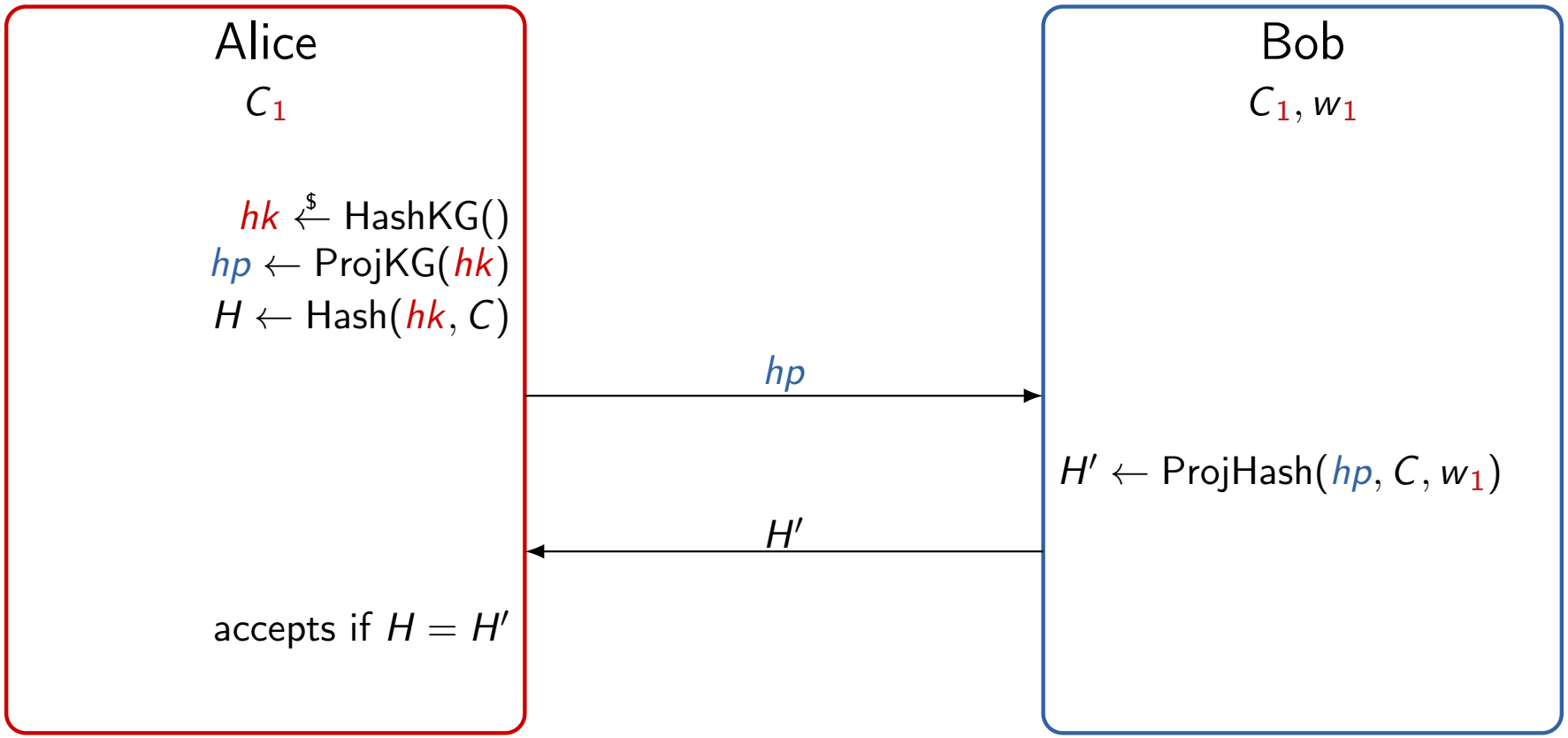
Bob wants to prove to Alice that $C \in L$.



Direct Applications of SPHF
Zero-Knowledge Argument

Bob wants to prove to Alice that $C_1 \in L_1$ in zero-knowledge.

$C = (C_1, C_2)$ — CRS: $C_2 \rightarrow \begin{cases} \text{stat. zero-knowledge} & \text{if } C_2 \in L_2 \\ \text{stat. soundness} & \text{if } C_2 \notin L_2 \end{cases}$

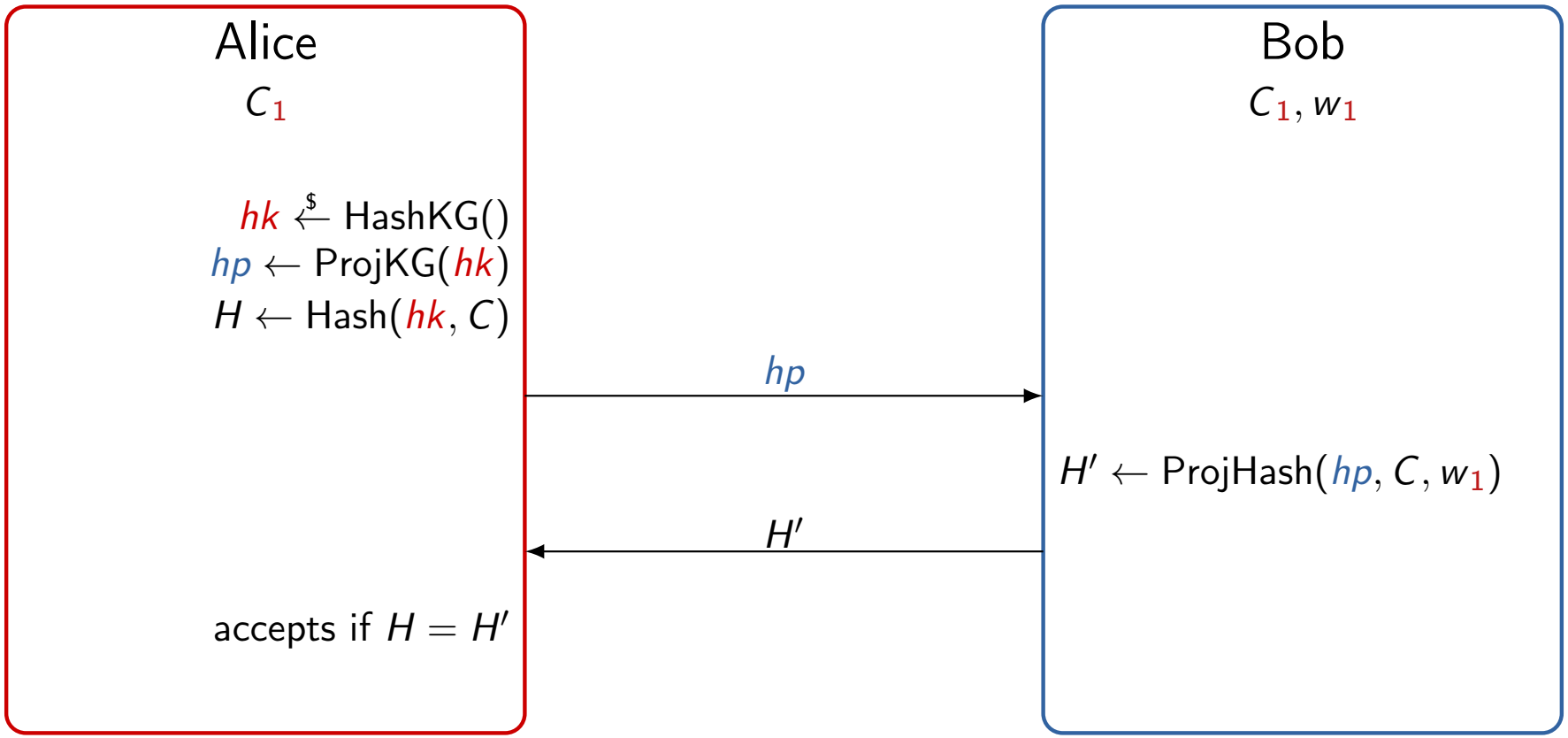


Direct Applications of SPHF

Zero-Knowledge Argument

Bob wants to prove to Alice that $C_1 \in L_1$ in zero-knowledge.

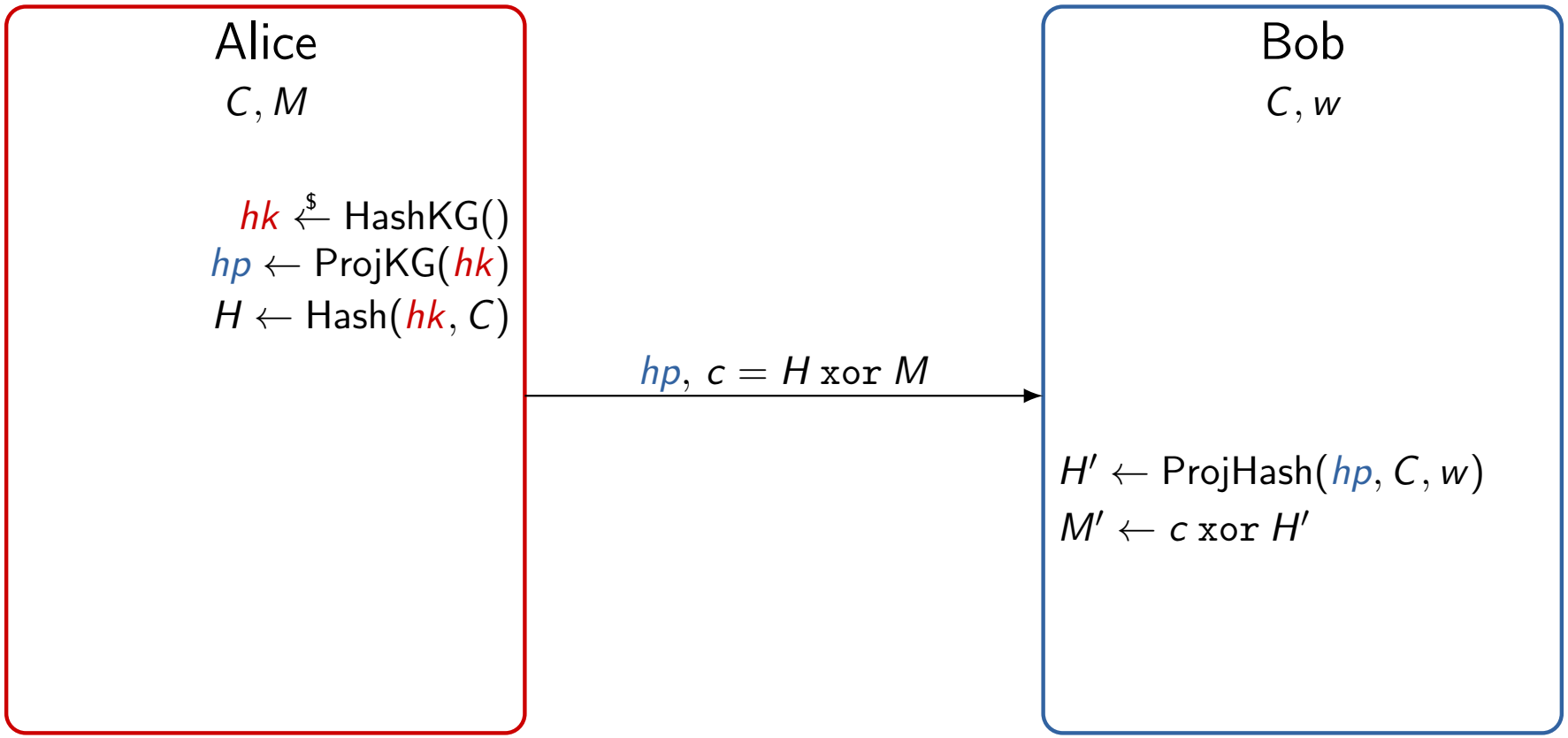
$C = (C_1, C_2)$ — CRS: $C_2 \rightarrow \begin{cases} \text{stat. zero-knowledge} & \text{if } C_2 \in L_2 \\ \text{stat. soundness} & \text{if } C_2 \notin L_2 \end{cases}$



Direct Applications of SPHF

Implicit Argument / Witness Encryption

Alice wants to send M to Bob if $C \in L$.

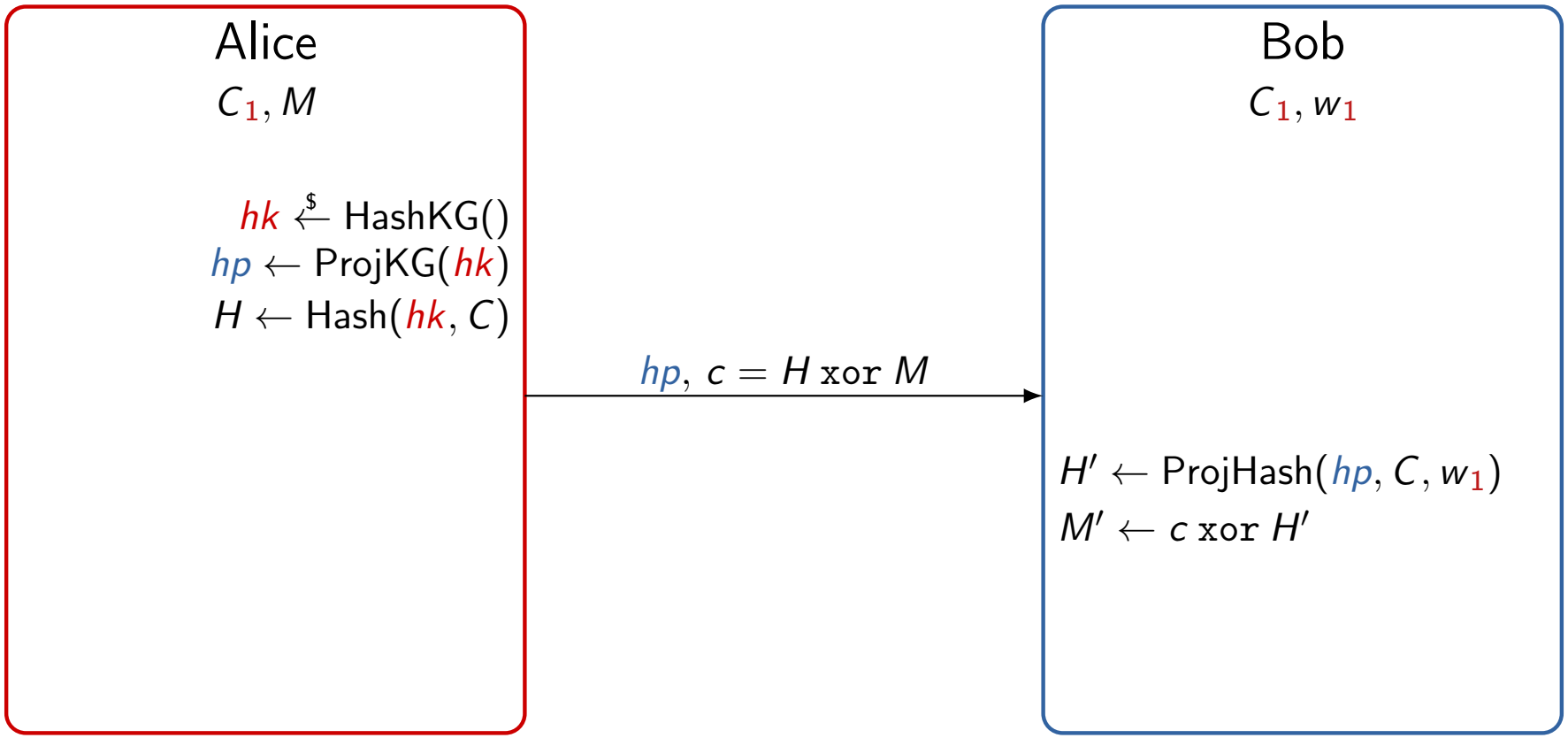


Direct Applications of SPHF

Implicit Zero-Knowledge Argument / Zero-Knowledge Witness Encryption

Alice wants to send M to Bob if $C_1 \in L_1$ in zero-knowledge.

$C = (C_1, C_2)$ — CRS: C_2



TSPHF

Previous ideas captured by

$$\text{Trapdoor SPHF} = \text{TSPHF}$$

- introduced in [BBC$^+$13];
- here: clean explanation.

Disjunction of two SPHFs?

- Ideally fully generic

  $$\text{SPHF for } L_1 + \text{SPHF for } L_2 \longrightarrow \text{SPHF for } L$$

  L: disjunction of L_1 and L_2

  $\longrightarrow$ seems hard

- Here:
  - Specific languages L_1 and L_2 over cyclic group $\mathbb{G}_1$ and $\mathbb{G}_2$
  - Use bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$

Disjunction of two SPHFs?

- Ideally fully generic

  $$\text{SPHF for } L_1 + \text{SPHF for } L_2 \longrightarrow \text{SPHF for } L$$

  L: disjunction of L_1 and L_2

  $\longrightarrow$ seems hard

- Here:
  - Specific languages L_1 and L_2 over cyclic group $\mathbb{G}_1$ and $\mathbb{G}_2$
  - Use bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$

Disjunction of two SPHFs?

- Ideally fully generic

  $$\text{SPHF for } L_1 + \text{SPHF for } L_2 \longrightarrow \text{SPHF for } L$$

  L: disjunction of L_1 and L_2

  $\longrightarrow$ seems hard

- Here:
  - Specific languages L_1 and L_2 over cyclic group $\mathbb{G}_1$ and $\mathbb{G}_2$
  - Use bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$

Which languages?

Framework [BBC+13]: SPHF when

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

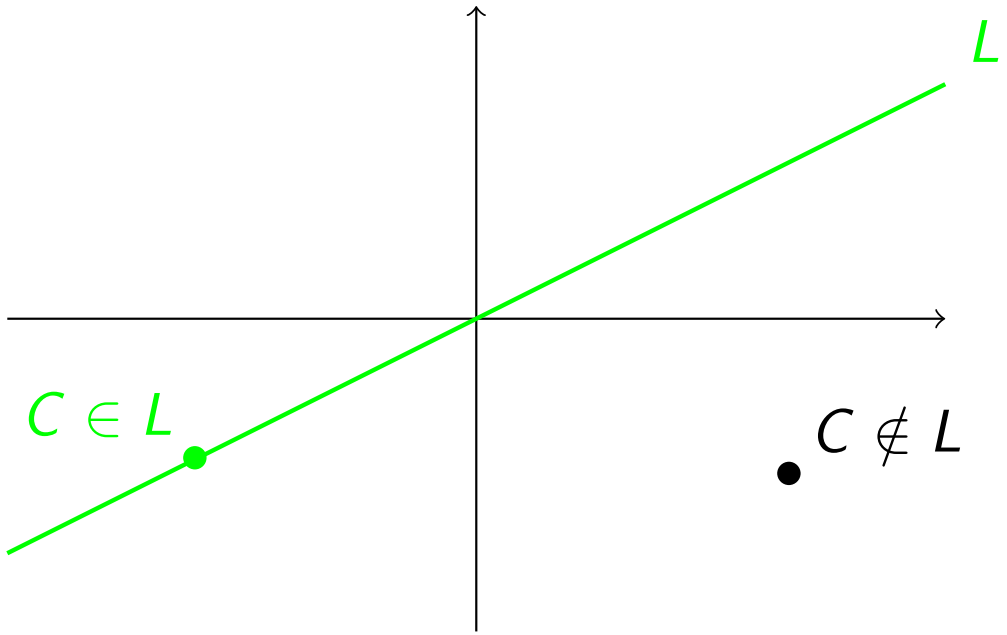
Diverse Vector Space

encompasses most SPHF over cyclic groups.

Which languages?

Framework [BBC+13]: SPHF when

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



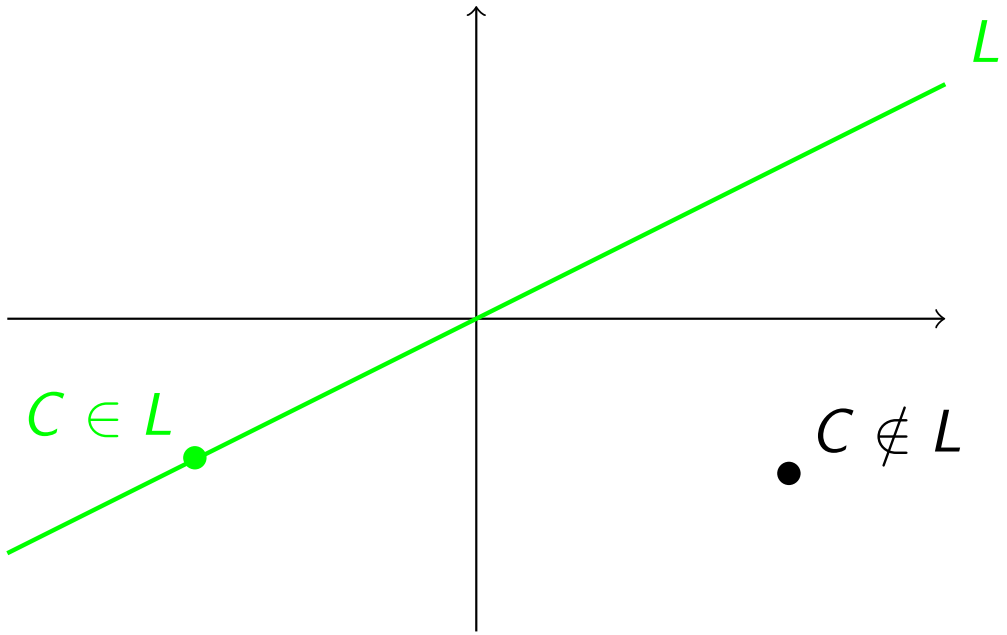
Diverse Vector Space

encompasses most SPHF over cyclic groups.

Which languages?

Framework [BBC+13]: SPHF when

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



$$L = \{(u, e) \mid \exists r \in \mathbb{Z}_p,\ u = g^r \text{ and } e = h^r\}$$

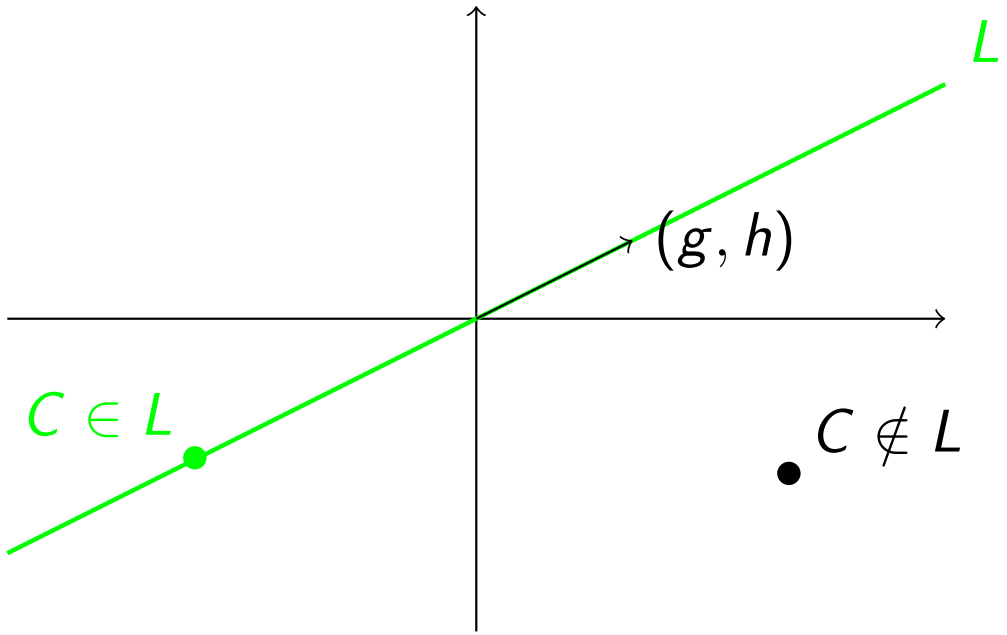
Diverse Vector Space

encompasses most SPHF over cyclic groups.

Which languages?

Framework [BBC+13]: SPHF when

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



$$L = \{(u, e) \mid \exists r \in \mathbb{Z}_p,\ (u, e) = r \bullet (g, h)\}$$

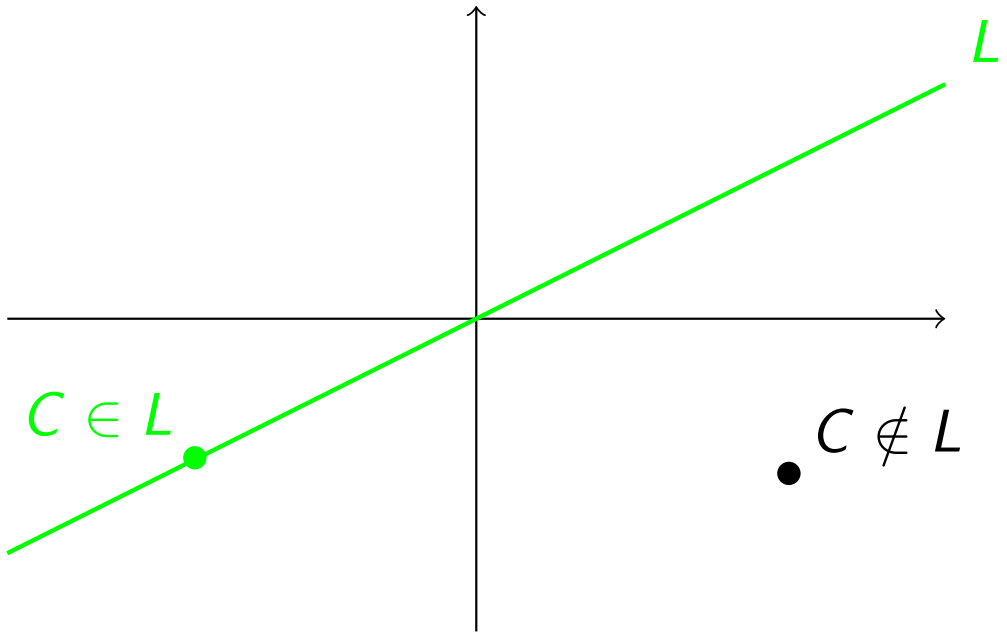
Diverse Vector Space

encompasses most SPHF over cyclic groups.

Which languages?

Framework [BBC+13]: SPHF when

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



$$L = \{(u, e) \mid \exists r \in \mathbb{Z}_p,\ (u, e) = r \bullet (g, h)\}$$

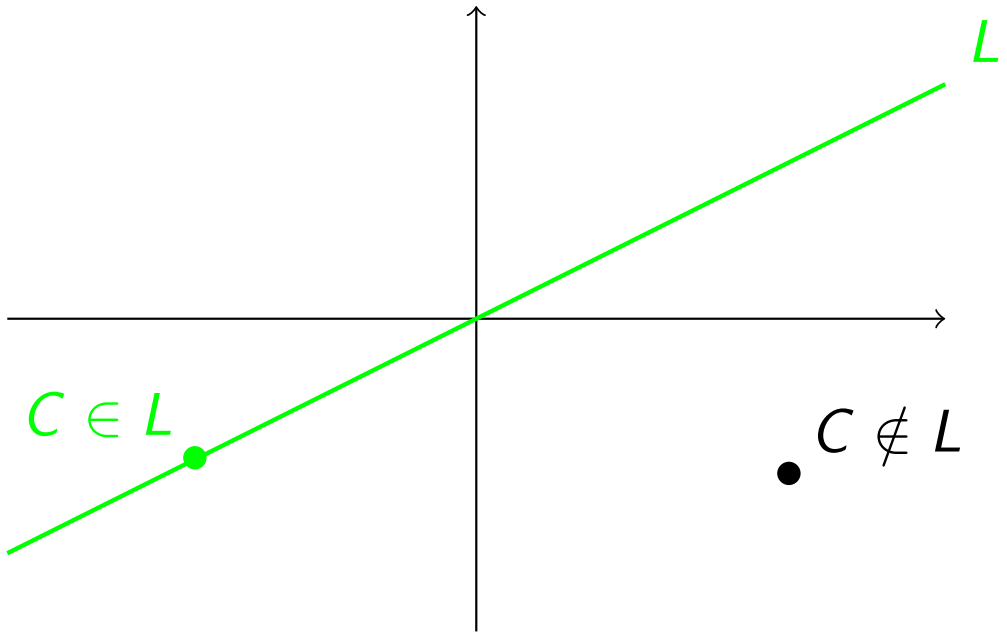
Diverse Vector Space

encompasses most SPHF over cyclic groups.

Which languages?

Framework [BBC+13]: SPHF when

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



$$L = \{(u, e) \mid \exists r \in \mathbb{Z}_p,\ (u, e) = r \bullet (g, h)\}$$

Diverse Vector Space

encompasses most SPHF over cyclic groups.

Construction of an SPHF for a Diverse Vector Space

- hk: random linear map: $\mathcal{X} \to \mathbb{G}$
- hp: hk restricted to L
- Hash value of C:

$$H := hk(C),$$

can be computed from hp, if $C \in L$.

Smoothness

No information on $hk(C)$ for $C \notin L$ (knowing only hp).

Construction of an SPHF for a Diverse Vector Space

- hk: random linear map: $\mathcal{X} \to \mathbb{G}$
- hp: hk restricted to L
- Hash value of C:

$$H := hk(C),$$

can be computed from hp, if $C \in L$.

Smoothness

No information on $hk(C)$ for $C \notin L$ (knowing only hp).

Conjunction and Disjunction

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

- Conjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := L_1 \times L_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X};$$

- Disjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := (L_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times L_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}.$$

But wait!

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

Idea: Tensor Product

$$L := \langle (L_1 \otimes \mathcal{X}_2) \cup (\mathcal{X}_1 \otimes L_2) \rangle \quad \subseteq \quad \mathcal{X}_1 \otimes \mathcal{X}_2 =: \mathcal{X}.$$

Conjunction and Disjunction

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

- Conjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := L_1 \times L_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X};$$

- Disjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := (L_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times L_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}.$$

But wait!

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

Idea: Tensor Product

$$L := \langle (L_1 \otimes \mathcal{X}_2) \cup (\mathcal{X}_1 \otimes L_2) \rangle \quad \subseteq \quad \mathcal{X}_1 \otimes \mathcal{X}_2 =: \mathcal{X}.$$

Conjunction and Disjunction

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

- Conjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := L_1 \times L_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X};$$

- Disjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := (L_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times L_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}.$$

But wait!

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

Idea: Tensor Product

$$L := \langle (L_1 \otimes \mathcal{X}_2) \cup (\mathcal{X}_1 \otimes L_2) \rangle \quad \subseteq \quad \mathcal{X}_1 \otimes \mathcal{X}_2 =: \mathcal{X}.$$

Conjunction and Disjunction

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

- Conjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := L_1 \times L_2 \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X};$$

- Disjunction of $L_1 \subseteq \mathcal{X}_1$ and $L_2 \subseteq \mathcal{X}_2$: SPHF for

$$L := (L_1 \times \mathcal{X}_2) \cup (\mathcal{X}_1 \times L_2) \quad \subseteq \quad \mathcal{X}_1 \times \mathcal{X}_2 =: \mathcal{X}.$$

But wait!

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

Idea: Tensor Product

$$L := \langle (L_1 \otimes \mathcal{X}_2) \cup (\mathcal{X}_1 \otimes L_2) \rangle \quad \subseteq \quad \mathcal{X}_1 \otimes \mathcal{X}_2 =: \mathcal{X}.$$

Quasi-Adaptive Non-Interactive Zero-Knowledge Arguments (QA-NIZK)

- CRS σ (depending on $L = \mathsf{QA}$) + trapdoor $\mathcal{T}$
- Proof of $C_1 \in L_1$ with witness w_1:

$$\pi \xleftarrow{\$} \mathsf{Prove}(\sigma, C_1, w_1);$$

- Verification

$$\mathsf{Ver}(\sigma, C_1, \pi);$$

- Simulation

$$\pi \xleftarrow{\$} \mathsf{Sim}(\sigma, \mathcal{T}, C_1).$$

Construction of QA-NIZK from Disjunction

- $L_2 \subseteq \mathcal{X}_2$: DDH, k-Lin, MDDH, ...
- CRS $\sigma = hp$ + trapdoor $\mathcal{T} = hk$
- Proof π of C_1: hash values of (C_1, e_j)
  e_j basis of $\mathcal{X}_2$
- Verification using hp
- Simulation using hk.

Soundness?

Valid π for $C_1 \longrightarrow$ compute hash value H'' of any (C_1, C_2)
If $C_1 \notin L_1$ and $C_2 \notin L_2$:

$$H'' = \mathsf{Hash}(hk, (C_1, C_2)) \quad \text{unpredictable!}$$

Construction of QA-NIZK from Disjunction

- $L_2 \subseteq \mathcal{X}_2$: DDH, k-Lin, MDDH, ...
- CRS $\sigma = hp$ + trapdoor $\mathcal{T} = hk$
- Proof π of C_1: hash values of (C_1, e_j)
  e_j basis of $\mathcal{X}_2$
- Verification using hp
- Simulation using hk.

Soundness?

Valid π for $C_1 \longrightarrow$ compute hash value H'' of any (C_1, C_2)
If $C_1 \notin L_1$ and $C_2 \notin L_2$:

$$H'' = \mathrm{Hash}(hk, (C_1, C_2)) \quad \text{unpredictable!}$$

Construction of QA-NIZK from Disjunction

- $L_2 \subseteq \mathcal{X}_2$: DDH, k-Lin, MDDH, ...
- CRS $\sigma = hp$ + trapdoor $\mathcal{T} = hk$
- Proof π of C_1: hash values of (C_1, e_j)
  e_j basis of $\mathcal{X}_2$
- Verification using hp
- Simulation using hk.

Soundness?

Valid π for $C_1 \longrightarrow$ compute hash value H'' of any (C_1, C_2)
If $C_1 \notin L_1$ and $C_2 \notin L_2$:

$$H'' = \mathsf{Hash}(hk, (C_1, C_2)) \quad \text{unpredictable!}$$

Comparison: QA-NIZK for Linear Subspaces

- When L_2 = DDH:
  - Same resulting construction as [JR14b];
  - Proof = 1 group element;
  - \+ Even for one-time simulation-soundness!
- When L_2 = MDDH (general case):
  - Same construction as [KW15];
  - \+ Different proof and prior work;
  - − Stronger assumption: MDDH instead of Ker-MDDH.

Contributions

- SPHF for disjunction of two languages L_1 and L_2:

$$L = \{(C_1, C_2) \mid C_1 \in L_1 \text{ or } C_2 \in L_2\};$$

(under some condition and with bilinear groups)

- + other tools for SPHF: 2-smoothness, PrPHF, ...
- Applications:
  - Quasi-Adaptive NIZK [JR13]
    - 1 group element (under DDH), as [JR14b]
    - even for one-time simulation-soundness
    - application: threshold Cramer-Shoup-like encryption scheme
  - First one-round PAKE for $k \geq 3$ players;
  - Link with homomorphic signatures [LPJY13];
  - New construction of Trapdoor SPHF (TSPHF) [BBC+13] (zero-knowledge variant of SPHF).

But beyond that...

Hash proof systems are cool!

But beyond that...

Hash proof systems are cool!

But beyond that...

Hash proof systems are cool!

- Lightweight alternative to NIZK or Zero-Knowledge proofs;
- Even more applications: PAKE;
- Large family of "algebraic" languages.

To sum up!

Next time you need a proof
think SPHF!

But beyond that...

Hash proof systems are cool!

- Lightweight alternative to NIZK or Zero-Knowledge proofs;
- Even more applications: PAKE;
- Large family of "algebraic" languages.

To sum up!

Next time you need a proof
think SPHF!

Thank you for your attention!

Questions?

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, December 2013.

Michel Abdalla, Céline Chevalier, and David Pointcheval.

Smooth projective hashing for conditionally extractable commitments.

In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 671–689. Springer, August 2009.

References II

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, August 2013.

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, August 1998.

References III

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, 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.

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, May 2003.

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

References IV

Jens Groth and Amit Sahai.

Efficient non-interactive proof systems for bilinear groups.

In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, April 2008.

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, May 2012.

Charanjit S. Jutla and Arnab Roy.

Shorter quasi-adaptive NIZK proofs for linear subspaces.

In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, December 2013.

References V

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.

Charanjit S. Jutla and Arnab Roy.

Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces.

In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, August 2014.

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, May 2005.

References VI

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, May 2001.

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, March 2011.

Eike Kiltz and Hoeteck Wee.

Quasi-adaptive NIZK for linear subspaces revisited.

In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, April 2015.

References VII

Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung.

Linearly homomorphic structure-preserving signatures and their applications.

In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, August 2013.