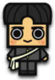
Non-Interactive Zero-Knowledge Proofs of Non-Membership

O. Blazy, C. Chevalier, D. Vergnaud

XLim / Université Paris II / ENS

Proof of Knowledge

Alice

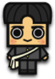


Bob

- Interactive method for one party to prove to another the knowledge of a secret $\mathcal{S}$.

Classical Instantiations : Schnorr proofs, Sigma Protocols ...

Proving that a statement is not satisfied



Alice

Bob

- Interactive method for one party to prove to another the knowledge of a secret $\mathcal{S}$ that does not belong to a language $\mathcal{L}$.

Applications

- Credentials
- Enhanced Authenticated Key Exchange

Additional properties

- Non-Interactive
- Zero-Knowledge
- Implicit

Applications

- Credentials
- Enhanced Authenticated Key Exchange

Additional properties

- Non-Interactive
- Zero-Knowledge
- Implicit

Applications

- Credentials
- Enhanced Authenticated Key Exchange

Additional properties

- Non-Interactive
- Zero-Knowledge
- Implicit

Applications

- Credentials
- Enhanced Authenticated Key Exchange

Additional properties

- Non-Interactive
- Zero-Knowledge
- Implicit

Applications

- Credentials
- Enhanced Authenticated Key Exchange

Additional properties

- Non-Interactive
- Zero-Knowledge
- Implicit

A
A
B
C
C

Zero-Knowledge Proof Systems

- Introduced in 1985 by Goldwasser, Micali and Rackoff.

$\rightsquigarrow$ Reveal nothing other than the validity of assertion being proven

- Used in many cryptographic protocols
  - Anonymous credentials
  - Anonymous signatures
  - Online voting
  - ...

Zero-Knowledge Proof Systems

- Introduced in 1985 by Goldwasser, Micali and Rackoff.

  $\rightsquigarrow$ Reveal nothing other than the validity of assertion being proven

- Used in many cryptographic protocols
  - Anonymous credentials
  - Anonymous signatures
  - Online voting
  - ...

Zero-Knowledge Proof Systems

- Introduced in 1985 by Goldwasser, Micali and Rackoff.

  $\rightsquigarrow$ Reveal nothing other than the validity of assertion being proven

- Used in many cryptographic protocols
  - **Anonymous credentials**
  - **Anonymous signatures**
  - **Online voting**
  - ...

Zero-Knowledge Interactive Proof

Alice

Bob

- interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** if $\mathcal{S}$ is true, the honest verifier will be convinced of this fact
2. **Soundness:** if $\mathcal{S}$ is false, no cheating prover can convince the honest verifier that it is true
3. **Zero-knowledge:** if $\mathcal{S}$ is true, no cheating verifier learns anything other than this fact.

Zero-Knowledge Interactive Proof

Alice

Bob

- interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** if $\mathcal{S}$ is true, the honest verifier will be convinced of this fact
2. **Soundness:** if $\mathcal{S}$ is false, no cheating prover can convince the honest verifier that it is true
3. **Zero-knowledge:** if $\mathcal{S}$ is true, no cheating verifier learns anything other than this fact.

Non-Interactive Zero-Knowledge Proof

Alice

Bob

- non-interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** $\mathcal{S}$ is true $\rightsquigarrow$ verifier will be convinced of this fact
2. **Soundness:** $\mathcal{S}$ is false $\rightsquigarrow$ no cheating prover can convince the verifier that $\mathcal{S}$ is true
3. **Zero-knowledge:** $\mathcal{S}$ is true $\rightsquigarrow$ no cheating verifier learns anything other than this fact.

Certification of Public Keys: SPHF [ACP09]

A user can ask for the certification of pk, but if he knows the associated sk only:

With a Smooth Projective Hash Function

$\mathcal{L}$: pk and $C = \mathcal{C}(\mathsf{sk}; r)$ are associated to the same sk

- U sends his pk, and an encryption C of sk;
- A generates the certificate Cert for pk, and sends it,
  masked by $\mathsf{Hash} = \mathsf{Hash}(\mathsf{hk}; (\mathsf{pk}, C))$;
- U computes $\mathsf{Hash} = \mathsf{ProjHash}(\mathsf{hp}; (\mathsf{pk}, C), r))$, and gets Cert.

A user can ask for the certification of pk, but if he knows the associated sk only:

With a Smooth Projective Hash Function

$\mathcal{L}$: pk and $C = \mathcal{C}(\mathsf{sk}; r)$ are associated to the same sk

- U sends his pk, and an encryption C of sk;
- A generates the certificate Cert for pk, and sends it, masked by $\mathsf{Hash} = \mathsf{Hash}(\mathsf{hk}; (\mathsf{pk}, C))$;
- U computes $\mathsf{Hash} = \mathsf{ProjHash}(\mathsf{hp}; (\mathsf{pk}, C), r))$, and gets Cert.

Implicit proof of knowledge of sk

Smooth Projective Hash Functions [CS02]

Definition [CS02,GL03]

Let $\{H\}$ be a family of functions:

- X, domain of these functions
- L, subset (a language) of this domain

such that, for any point x in L, $H(x)$ can be computed by using

- either a *secret* hashing key hk: $H(x) = \mathsf{Hash}_L(\mathsf{hk}; x)$;
- or a *public* projected key hp: $H'(x) = \mathsf{ProjHash}_L(\mathsf{hp}; x, w)$

Public mapping $\mathsf{hk} \mapsto \mathsf{hp} = \mathsf{ProjKG}_L(\mathsf{hk}, x)$

SPHF Properties

For any $x \in X$, $H(x) = \mathsf{Hash}_L(\mathsf{hk}; x)$
For any $x \in L$, $H(x) = \mathsf{ProjHash}_L(\mathsf{hp}; x, w)$
w witness that $x \in L, \mathsf{hp} = \mathsf{ProjKG}_L(\mathsf{hk}, x)$

Smoothness

For any $x \notin L$, $H(x)$ and hp are independent

Pseudo-Randomness

For any $x \in L$, $H(x)$ is pseudo-random, without a witness w

The latter property requires L to be a hard-partitioned subset of X.

SPHF Properties

For any $x \in X$, $H(x) = \mathsf{Hash}_L(\mathsf{hk}; x)$
For any $x \in L$, $H(x) = \mathsf{ProjHash}_L(\mathsf{hp}; x, w)$
w witness that $x \in L, \mathsf{hp} = \mathsf{ProjKG}_L(\mathsf{hk}, x)$

Smoothness

For any $x \notin L$, $H(x)$ and hp are independent

Pseudo-Randomness

For any $x \in L$, $H(x)$ is pseudo-random, without a witness w

The latter property requires L to be a hard-partitioned subset of X.

SPHF Properties

For any $x \in X$, $H(x) = \mathsf{Hash}_L(\mathsf{hk}; x)$
For any $x \in L$, $H(x) = \mathsf{ProjHash}_L(\mathsf{hp}; x, w)$
w witness that $x \in L, \mathsf{hp} = \mathsf{ProjKG}_L(\mathsf{hk}, x)$

Smoothness

For any $x \notin L$, $H(x)$ and hp are independent

Pseudo-Randomness

For any $x \in L$, $H(x)$ is pseudo-random, without a witness w

The latter property requires L to be a hard-partitioned subset of X.

SPHF Properties

For any $x \in X$, $H(x) = \mathsf{Hash}_L(\mathsf{hk}; x)$
For any $x \in L$, $H(x) = \mathsf{ProjHash}_L(\mathsf{hp}; x, w)$
w witness that $x \in L, \mathsf{hp} = \mathsf{ProjKG}_L(\mathsf{hk}, x)$

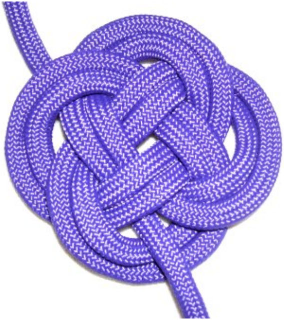
Smoothness

For any $x \notin L$, $H(x)$ and hp are independent

Pseudo-Randomness

For any $x \in L$, $H(x)$ is pseudo-random, without a witness w

The latter property requires L to be a hard-partitioned subset of X.



Global Idea

- π: Proof that $W \in \mathcal{L}$
- π: Randomizable, Indistinguishability of Proof
- π': Proof that π was computed honestly

Global Idea

- π: Proof that $W \in \mathcal{L}$
- π: Randomizable, Indistinguishability of Proof
- π': Proof that π was computed honestly

Global Idea

- π: Proof that $W \in \mathcal{L}$
- π: Randomizable, Indistinguishability of Proof
- π': Proof that π was computed honestly

Global Idea

- π: Proof that $W \in \mathcal{L}$
- π: Randomizable, Indistinguishability of Proof
- π': Proof that π was computed honestly

To prove that $W \notin \mathcal{L}$

- Try to prove that $W \in \mathcal{L}$ which will output a π
- π will not be valid
- Compute π' stating that π was computed honestly

Global Idea

- π: Proof that $W \in \mathcal{L}$
- π: Randomizable, Indistinguishability of Proof
- π': Proof that π was computed honestly

To prove that $W \notin \mathcal{L}$

- Try to prove that $W \in \mathcal{L}$ which will output a π
- π will not be valid
- Compute π' stating that π was computed honestly

Global Idea

- π: Proof that $W \in \mathcal{L}$
- π: Randomizable, Indistinguishability of Proof
- π': Proof that π was computed honestly

To prove that $W \notin \mathcal{L}$

- Try to prove that $W \in \mathcal{L}$ which will output a π
- π will not be valid
- Compute π' stating that π was computed honestly

From a very high level

- If an adversary forges a proof, this means that both π and π' are valid
- Either π was not computed honestly, and under the Soundness of π' this should not happen
- Or π was computed honestly but lead to an invalid proof, and under the Completeness of π this should not happen

From a very high level

- If an adversary forges a proof, this means that both π and π' are valid
- Either π was not computed honestly, and under the Soundness of π' this should not happen
- Or π was computed honestly but lead to an invalid proof, and under the Completeness of π this should not happen

From a very high level

- If an adversary forges a proof, this means that both π and π' are valid
- Either π was not computed honestly, and under the Soundness of π' this should not happen
- Or π was computed honestly but lead to an invalid proof, and under the Completeness of π this should not happen

Possible Instantiations

Proof π	Proof π'	Interactive	Properties
Groth Sahai	Groth Sahai	No	Zero-Knowledge
SPHF	SPHF	Yes	Implicit
Groth Sahai	SPHF	Depends	ZK, Implicit

Anonymous Credentials

Allows user to authenticate while protecting their privacy.

- Recent work, build non-interactive credentials for NAND
- By combining with ours, it leads to *efficient* Non-Interactive Credentials
- No accumulators are needed

Anonymous Credentials

Allows user to authenticate while protecting their privacy.

- Recent work, build non-interactive credentials for NAND
- By combining with ours, it leads to *efficient* Non-Interactive Credentials
- No accumulators are needed

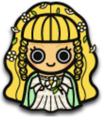
Anonymous Credentials

Allows user to authenticate while protecting their privacy.

- Recent work, build non-interactive credentials for NAND
- By combining with ours, it leads to *efficient* Non-Interactive Credentials
- No accumulators are needed

Language Authenticated Key Exchange

Alice

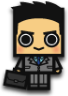


$\rightarrow \mathcal{C}(M_B)$

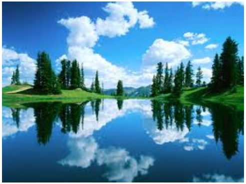
$\mathcal{C}(M_A), \mathsf{hp}_B \leftarrow$

$\rightarrow \mathsf{hp}_A$

Bob



$H_B \cdot H'_A$



$H'_B \cdot H_A$

Same value iff languages are as expected, and users know witnesses.

Summing up

- Proposed a generic framework to prove negative statement *
- Gives several instantiation of this framework, allowing some modularity
- Works outside pairing environment

Open Problems

- Be compatible with post-quantum cryptography
- Weaken the requirements, on the building blocks

Summing up

- Proposed a generic framework to prove negative statement *
- Gives several instantiation of this framework, allowing some modularity
- Works outside pairing environment

Open Problems

- Be compatible with post-quantum cryptography
- Weaken the requirements, on the building blocks