

I – Zero-Knowledge Proofs and Applications

David Pointcheval

Ecole normale supérieure, CNRS & INRIA



ENS – Paris – 2016/2017

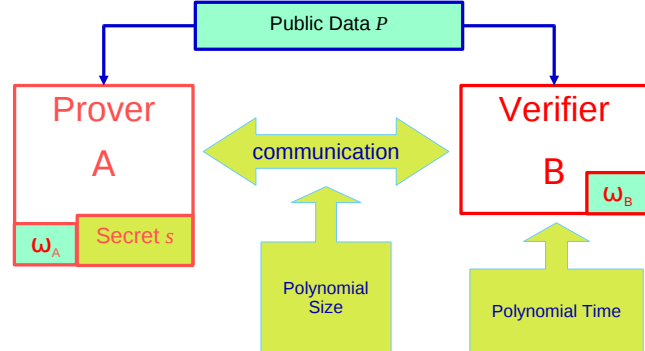
- 1 Zero-Knowledge Proofs of Knowledge
 - Introduction
 - 3-Coloring
 - Examples
- 2 Signatures
 - From Identification to Signature
 - Forking Lemma
- 3 Zero-Knowledge Proofs of Membership
 - Introduction
 - Example: DH

Outline

Proof of Knowledge

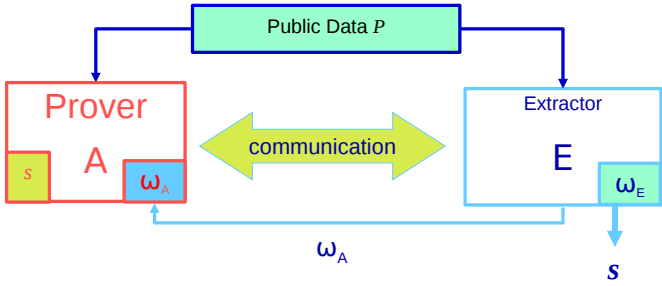
- 1 Zero-Knowledge Proofs of Knowledge
 - Introduction
 - 3-Coloring
 - Examples
- 2 Signatures
- 3 Zero-Knowledge Proofs of Membership

How do I prove that I know a solution s to a problem P ?



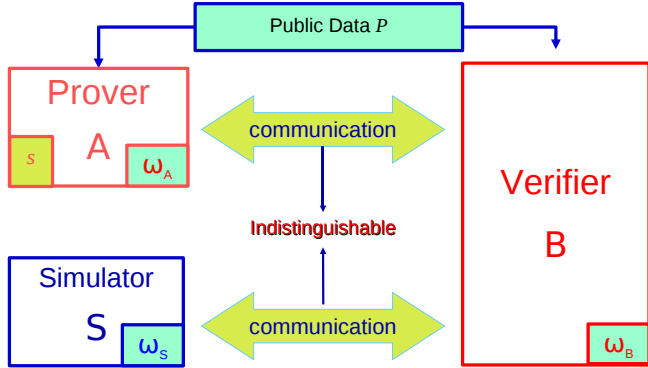
Proof of Knowledge: Soundness

\mathcal{A} knows something... What does it mean?
 the information can be extracted: **extractor**



Proof of Knowledge: Zero-Knowledge

How do I prove that I know a solution s to a problem P ?
 I reveal the solution...
 How can I do it without revealing any information?
 Zero-knowledge: **simulation** and **indistinguishability**

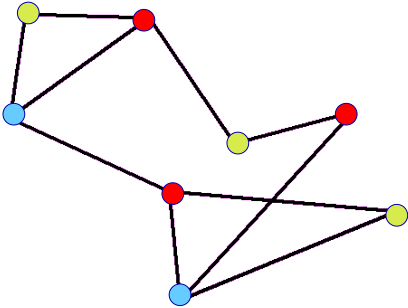


Outline

- 1 Zero-Knowledge Proofs of Knowledge**
 - Introduction
 - 3-Coloring**
 - Examples
- 2 Signatures**
- 3 Zero-Knowledge Proofs of Membership**

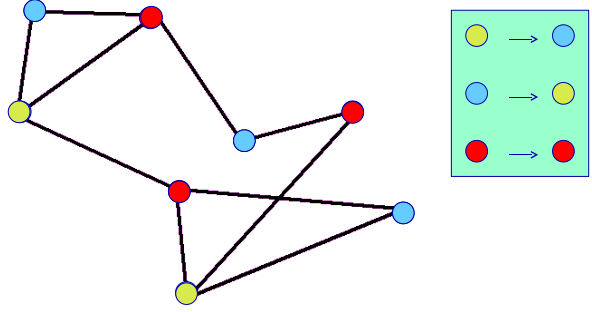
Proof of Knowledge

How do I prove that I know a 3-color covering,
 without revealing any information?



Proof of Knowledge

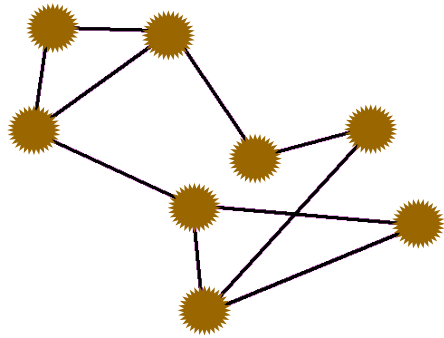
How do I prove that I know a 3-color covering, without revealing any information?



I choose a random permutation on the colors and I apply it to the vertices

Proof of Knowledge

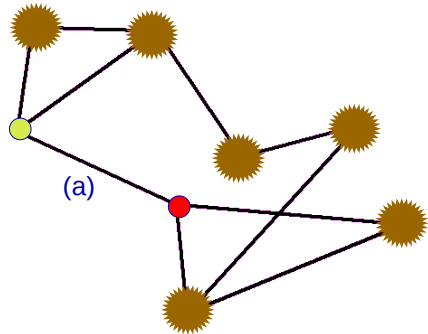
How do I prove that I know a 3-color covering, without revealing any information?



I mask the vertices and send it to the verifier

Proof of Knowledge

How do I prove that I know a 3-color covering, without revealing any information?

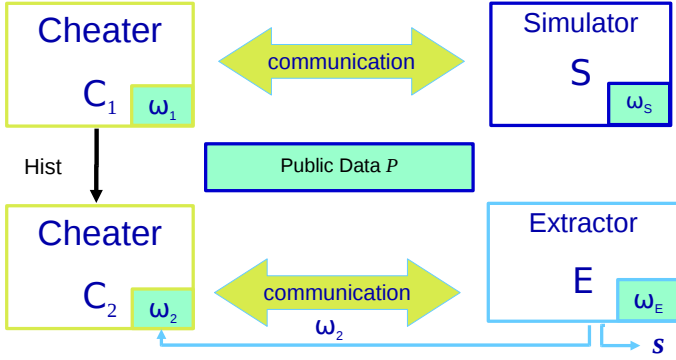


The verifier chooses an edge I open it

The verifier checks the validity: 2 different colors

Secure Multiple Proofs of Knowledge: Authentication

If there exists an efficient adversary, then one can solve the underlying problem:



1 Zero-Knowledge Proofs of Knowledge

- Introduction
- 3-Coloring
- Examples

2 Signatures

3 Zero-Knowledge Proofs of Membership

Generic Proof

- Proof of knowledge of x , such that $\mathcal{R}(x, y)$
- \mathcal{P} builds a commitment r and sends it to \mathcal{V}
- \mathcal{V} chooses a challenge $h \xleftarrow{R} \{0, 1\}^k$ for \mathcal{P}
- \mathcal{P} computes and sends the answer s
- \mathcal{V} checks (r, h, s)

Σ -Protocol

- Proof of knowledge of x
- \mathcal{P} sends a commitment r
- \mathcal{V} sends a challenge h
- \mathcal{P} sends the answer s
- \mathcal{V} checks (r, h, s)

Special soundness

If one can answer to two different challenges $h \neq h'$:
 $\implies s$ and s' for a unique x
 \implies one can extract x

SQRT Fiat-Shamir Proof

[Fiat-Shamir – Crypto '86]

Fiat-Shamir Proof: Extraction

- Setting: $n = pq$
 \mathcal{P} knows x , such that $X = x^2 \pmod n$ and wants to prove it to \mathcal{V}
- \mathcal{P} chooses $r \xleftarrow{R} \mathbb{Z}_n^*$, sets and sends $R = r^2 \pmod n$
- \mathcal{V} chooses $b \xleftarrow{R} \{0, 1\}$ and sends it to \mathcal{P}
- \mathcal{P} computes and sends $s = x^b \times r \pmod n$
- \mathcal{V} checks whether $s^2 \stackrel{?}{=} X^b R \pmod n$

One then reiterates t times

For a fixed R , two valid answers s and s' satisfy

$$s^2/X = R = (s')^2 \pmod n \implies X = (s/s')^2 \pmod n$$

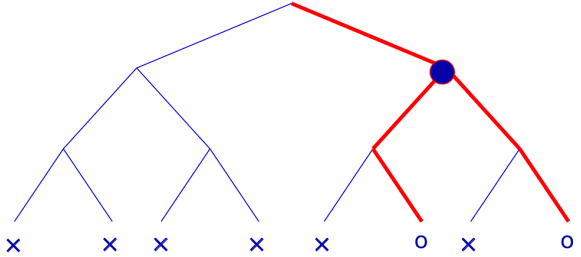
And thus $x = s/s' \pmod n \implies$ **Special Soundness**

More precisely: the execution of t repetitions depends on

- (b_1, \dots, b_t) from the verifier \mathcal{V}
- ω that (together with the previous b_i ($i < k$)) determines R_k from the prover \mathcal{P}

If $\Pr_{\omega, (b_i)}[\mathcal{V} \text{ accepts } \mathcal{P}] > 1/2^t + \epsilon$,
 there is a good fraction of ω (more than $\epsilon/2$)
 such that $\Pr_{(b_i)}[\mathcal{V} \text{ accepts } \mathcal{S}] \geq 1/2^t + \epsilon/2$.

For such a good ω : a good node along the successful path



Honest Verifier

Simulation of a triplet: $(R = r^2 \bmod n, b, s = x^b \times r \bmod n)$
 for $r \xleftarrow{R} \mathbb{Z}_n^*$ and $b \xleftarrow{R} \{0, 1\}$
 Similar to: $(R = s^2/X^b \bmod n, b, s)$
 for $s \xleftarrow{R} \mathbb{Z}_n^*$ and $b \xleftarrow{R} \{0, 1\}$
 Simulation: random s and b , and set $(R = s^2/X^b \bmod n, b, s)$

Any Verifier

Simulation of a triplet: $(R = r^2 \bmod n, b = \mathcal{V}(\text{view}), s = x^b \times r \bmod n)$
 for $r \xleftarrow{R} \mathbb{Z}_n^*$ only!
 Similar to: $(R = s^2/X^b \bmod n, b = \mathcal{V}(\text{view}), s)$ for $s \xleftarrow{R} \mathbb{Z}_n^*$
 Simulation: random s and β , and set $R = s^2/X^\beta \bmod n$
 upon reception of b : if $b = \beta$, output s , else rewind
 b and β independent: rewind once over 2 \implies linear time

- Setting: $n = pq$ and an exponent e
 \mathcal{P} knows x , such that $X = x^e \bmod n$ and wants to prove it to \mathcal{V}
- \mathcal{P} chooses $r \xleftarrow{R} \mathbb{Z}_n^*$, sets and sends $R = r^e \bmod n$
- \mathcal{V} chooses $b \xleftarrow{R} \{0, 1\}^t$ and sends it to \mathcal{P}
- \mathcal{P} computes and sends $s = x^e \times r \bmod n$
- \mathcal{V} checks whether $s^e \stackrel{?}{=} X^b R \bmod n$

For a fixed R , two valid answers s and s' satisfy
 $s^e/X^b = R = (s')^e/X^b \bmod n \implies X^{b-b} = (s'/s)^e \bmod n$

If e prime and bigger than 2^t , then e and $b' - b$ are relatively prime:
 Bezout: $ue + v(b' - b) = 1 \implies X^{v(b' - b)} = (s'/s)^{ve} = X^{1-ue} \bmod n$
 As a consequence: $X = ((s'/s)^v X^u)^e \implies$ **Special Soundness**

DL Schnorr Proof

[Schnorr – Eurocrypt '89 - Crypto '89]

Outline

- Setting: $\mathbb{G} = \langle g \rangle$ of order q
 \mathcal{P} knows x , such that $y = g^{-x}$ and wants to prove it to \mathcal{V}
- \mathcal{P} chooses $k \xleftarrow{R} \mathbb{Z}_q^*$, sets and sends $r = g^k$
- \mathcal{V} chooses $h \xleftarrow{R} \{0, 1\}^t$ and sends it to \mathcal{P}
- \mathcal{P} computes and sends $s = k + xh \bmod q$
- \mathcal{V} checks whether $r \stackrel{?}{=} g^s y^h$

For a fixed r , two valid answers s and s' satisfy

$$g^s y^h = r = g^{s'} y^h \implies y^{h-h} = g^{s-s}$$

And thus $x = (s - s')(h' - h)^{-1} \bmod q \implies$ **Special Soundness**

- 1 Zero-Knowledge Proofs of Knowledge
- 2 Signatures
 - From Identification to Signature
 - Forking Lemma
- 3 Zero-Knowledge Proofs of Membership

Zero-Knowledge Proof

- Proof of knowledge of x , such that $\mathcal{R}(x, y)$
- \mathcal{P} builds a commitment r and sends it to \mathcal{V}
- \mathcal{V} chooses a challenge $h \xleftarrow{R} \{0, 1\}^k$ for \mathcal{P}
- \mathcal{P} computes and sends the answer s
- \mathcal{V} checks (r, h, s)

Signature

- \mathcal{H} viewed as a random oracle
- Key Generation $\rightarrow (y, x)$
private: x public: y
 - Signature of $m \rightarrow (r, h, s)$
Commitment r
Challenge $h = \mathcal{H}(m, r)$
Answer s
 - Verification of (m, r, s)
compute $h = \mathcal{H}(m, r)$
and check (r, h, s)

Zero-Knowledge Proof

- Proof of knowledge of x
- \mathcal{P} sends a commitment r
- \mathcal{V} sends a challenge h
- \mathcal{P} sends the answer s
- \mathcal{V} checks (r, h, s)

Signature

- Key Generation $\rightarrow (y, x)$
- Signature of $m \rightarrow (r, h, s)$
Commitment r
Challenge $h = \mathcal{H}(m, r)$
Answer s
- Verification of (m, r, s)
compute $h = \mathcal{H}(m, r)$
and check (r, h, s)

Special soundness

If one can answer to two different challenges $h \neq h'$: s and s' for a unique commitment r , one can extract x

Outline

- 1 Zero-Knowledge Proofs of Knowledge
- 2 Signatures
 - From Identification to Signature
 - Forking Lemma
- 3 Zero-Knowledge Proofs of Membership

Forking Lemma

[Pointcheval-Stern – Eurocrypt '96]

The **Forking Lemma** shows an efficient reduction between the signature scheme and the identification scheme, but basically, if an adversary \mathcal{A} produces, with probability $\varepsilon \geq 2/2^k$, a valid signature (m, r, h, s) , then within $T' = 2T$, one gets two valid signatures (m, r, h, s) and (m, r, h', s') , with $h \neq h'$ with probability $\varepsilon' \geq \varepsilon^2/32q_H^3$.

The **special soundness** provides the secret key.

1 Zero-Knowledge Proofs of Knowledge

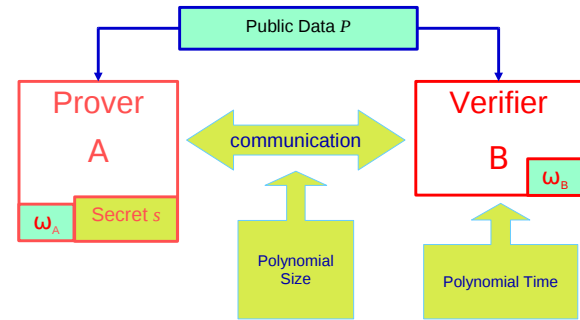
2 Signatures

3 Zero-Knowledge Proofs of Membership

- Introduction
- Example: DH

How do I prove that a word w lies in a language \mathcal{L} : $P = (w, \mathcal{L})$?

- if $\mathcal{L} \in \mathcal{BPP}$: anybody can publicly check it
- if $\mathcal{L} \in \mathcal{NP} \setminus \mathcal{BPP}$: a witness s can help prove that $w \in \mathcal{L}$



If $w \notin \mathcal{L}$:

- Proof (perfect soundness): a powerful \mathcal{A} cannot cheat
- Argument (computational soundness): a limited \mathcal{A} cannot cheat

Proof of Membership

Outline

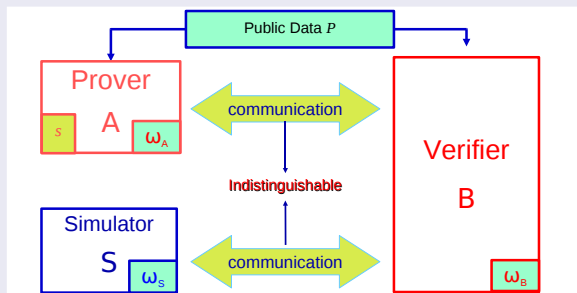
Soundness

$w \in \mathcal{L}$... what does it mean?
a witness **exists**, different from knowing it: no need of extractor

Zero-Knowledge

How do I prove there exists a witness s ? I reveal it...
How can I do it without revealing any information?

Zero-knowledge:
simulation
and **indistinguishability**



1 Zero-Knowledge Proofs of Knowledge

2 Signatures

3 Zero-Knowledge Proofs of Membership

- Introduction
- Example: DH

Diffie-Hellman Language

In a group $\mathbb{G} = \langle g \rangle$ of prime order q ,

the **DDH**(g, h) assumption states it is hard to distinguish

$$\mathcal{L} = (u = g^x, v = h^x) \text{ from } \mathbb{G}^2 = (u = g^x, v = h^y)$$

- \mathcal{P} knows x , such that $(u = g^x, v = h^x)$ and wants to prove it to \mathcal{V}
- \mathcal{P} chooses $k \xleftarrow{R} \mathbb{Z}_q^*$, sets and sends $U = g^k$ and $V = h^k$
- \mathcal{V} chooses $h \xleftarrow{R} \{0, 1\}^t$ and sends it to \mathcal{P}
- \mathcal{P} computes and sends $s = k + xh \bmod q$
- \mathcal{V} checks whether $U \stackrel{?}{=} g^s u^h$ and $V \stackrel{?}{=} h^s v^h$

For a fixed (U, V) , two valid answers s and s' satisfy

$$g^s u^h = U = g^{s'} u^h \quad h^s v^h = V = h^{s'} v^h$$

- if one sets $y = (s - s')(h' - h)^{-1} \bmod q \implies u = g^y$ and $v = h^y$
- there exists a witness: **Perfect Soundness**