Proof of Knowledge

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

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
Proof of Knowledge: Soundness

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

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:
3-Pass Zero-Knowledge Proofs

Generic Proof
- Proof of knowledge of $x$, such that $R(x, y)$
- $\mathcal{P}$ builds a commitment $r$ and sends it to $\mathcal{V}$
- $\mathcal{V}$ chooses a challenge $h \overset{\$}{\leftarrow} \{0, 1\}^k$ 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 $r$
$\implies$ one can extract $x$

Fiat-Shamir Proof: Extraction

More precisely: the execution of $t$ repetitions depends on
- $(b_1, \ldots, b_t)$ from the verifier $\mathcal{V}$
- $\omega$ that (together with the previous $b_i$ $(i < k)$) determines $R_k$ from the prover $\mathcal{P}$
If $\Pr_{\omega, (b_1)}[\mathcal{V} \text{ accepts } \mathcal{P}] > 1/2^t + \varepsilon$,
there is a good fraction of $\omega$ (more than $\varepsilon/2$)
such that $\Pr_{(b_3)}[\mathcal{V} \text{ accepts } S] \geq 1/2^t + \varepsilon/2$.
For such a good $\omega$: a good node along the successful path
Fiat-Shamir Proof: Simulation

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

Any Verifier
Simulation of a triplet: \((R = r^2 \mod n, b = V(\text{view}), s = x^b \times r \mod n)\)
for \(r \in \mathbb{Z}_n^*\) only!
Similar to: \((R = s^2 / X^b \mod n, b = V(\text{view}), s)\) for \(s \in \mathbb{Z}_n^*\)
Simulation: random \(s\) and \(B\), and set \(R = s^2 / X^B \mod n\)
upon reception of \(b\): if \(b = B\), output \(s\), else rewind \(b\) and \(B\) independent: rewind once over 2 \(\Rightarrow\) linear time

RSA GQ Proof

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

For a fixed \(R\), two valid answers \(s\) and \(s'\) satisfy
\[s^e / X^b = R = (s')^e / X^{b'} \mod n \Rightarrow X^{b' - b} = (s'/s)^e \mod n\]
If \(e\) prime and bigger than \(2^t\), then \(e\) and \(b' - b\) are relatively prime:
Bezout: \(ue + v(b' - b) = 1 \Rightarrow X^{uv(b' - b)} = (s'/s)^v X^{1 - ue} \mod n\)
As a consequence: \(X = ((s'/s)^v X^u)^e \Rightarrow \text{Special Soundness}\)

DL Schnorr Proof

- Setting: \(G = \langle g \rangle\) of order \(q\)
  \(P\) knows \(x\), such that \(y = g^{-x}\) and wants to prove it to \(V\)
- \(P\) chooses \(k \in \mathbb{Z}_q^*\), sets and sends \(r = g^k\)
- \(V\) chooses \(h \in \{0, 1\}^t\) and sends it to \(P\)
- \(P\) computes and sends \(s = k + xh \mod q\)
- \(V\) checks whether \(r \equiv g^s y^h \mod q\)

For a fixed \(r\), two valid answers \(s\) and \(s'\) satisfy
\[g^s y^h = r = g^{s'} y^{h'} \Rightarrow y^{h' - h} = g^{s - s'}\]
And thus \(x = (s - s')(h' - h)^{-1} \mod q \Rightarrow \text{Special Soundness}\)

Outline

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 \(R(x, y)\)
- \(P\) builds a commitment \(r\) and sends it to \(V\)
- \(V\) chooses a challenge \(h\) \(R \leftarrow \{0, 1\}\)
- \(P\) computes and sends the answer \(s\)
- \(V\) checks \((r, h, s)\)

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

The **Forking Lemma** shows an efficient reduction between the signature scheme and the identification scheme, but basically, if an adversary \(A\) 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.
Proof of Membership

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

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

If $w \not\in L$:
- Proof (perfect soundess): a powerful $A$ cannot cheat
- Argument (computational soundness): a limited $A$ cannot cheat

Soundness

$w \in L$... 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
In a group $\mathbb{G} = \langle g \rangle$ of prime order $q$, the DDH($g, h$) assumption states it is hard to distinguish $L = (u = g^x, v = h^x)$ from $G^2 = (u = g^k, v = h^y)$

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

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