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?

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

I mask the vertices and send it to the verifier

Secure Multiple Proofs of Knowledge: Authentication

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

The verifier chooses an edge
I open it
The verifier checks the validity: 2 different colors
**Outline**

1. Zero-Knowledge Proofs of Knowledge
   - Introduction
   - 3-Coloring
   - Examples

2. Signatures

3. Zero-Knowledge Proofs of Membership

---

### 3-Pass Zero-Knowledge Proofs

**Generic Proof**

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

**\( \Sigma \)-Protocol**

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

### Special soundness

If one can answer to two different challenges \( h \neq h' \):

\[ s \text{ and } s' \text{ for a unique } r \implies \text{one can extract } x \]

---

### SQRT Fiat-Shamir Proof

**[Fiat-Shamir – Crypto ’86]**

- Setting: \( n = pq \)
  - \( P \) knows \( x \), such that \( X = x^2 \mod n \) and wants to prove it to \( V \)
  - \( P \) chooses \( r \sim \mathbb{Z}_n^* \), sets and sends \( R = r^2 \mod n \)
  - \( V \) chooses \( b \sim \{0, 1\} \) and sends it to \( P \)
  - \( P \) computes and sends \( s = x^b \times r \mod n \)
  - \( V \) checks whether \( s^2 \equiv X^b \mod n \)

One then reiterates \( t \) times

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

\[ s^2 / X = R = (s')^2 \mod n \implies X = (s/s')^2 \mod n \]

And thus \( x = s/s' \mod n \implies \text{Special Soundness} \)
Fiat-Shamir Proof: Simulation

Honest Verifier

Simulation of a triplet: \((R = r^2 \mod n, b, s = x^b \times r \mod n)\)

for \(r \overset{\$}{\leftarrow} \mathbb{Z}_n^*\) and \(b \overset{\$}{\leftarrow} \{0, 1\}\)

Similar to: \((R = s^2 / X^b \mod n, b, s)\)

for \(s \overset{\$}{\leftarrow} \mathbb{Z}_n^*\) and \(b \overset{\$}{\leftarrow} \{0, 1\}\)

Simulation: random \(s\) and \(b\), and set \((R = s^2 / X^b \mod n, b, s)\)

Any Verifier

Simulation of a triplet: \((R = r^2 \mod n, b = \mathcal{V}(\text{view}), s = x^b \times r \mod n)\)

for \(r \overset{\$}{\leftarrow} \mathbb{Z}_n^*\) only!

Similar to: \((R = s^2 / X^b \mod n, b = \mathcal{V}(\text{view}), s)\) for \(s \overset{\$}{\leftarrow} \mathbb{Z}_n^*\)

Simulation: random \(s\) and \(\beta\), and set \(R = s^2 / X^2 \mod n\)

upon reception of \(b\): if \(b = \beta\), output \(s\), else rewind \(b\) and \(\beta\) independent: rewind once over 2 \(\implies\) linear time

RSA GQ Proof

[Guillou-Quisquater – Crypto ’87 – Eurocrypt ’88]

Setting: \(n = pq\) and an exponent \(e\)

\(\mathcal{P}\) knows \(x\), such that \(X = x^e \mod n\) and wants to prove it to \(\mathcal{V}\)

\(\mathcal{P}\) chooses \(r \overset{\$}{\leftarrow} \mathbb{Z}_n^*\), sets and sends \(R = r^e \mod n\)

\(\mathcal{V}\) chooses \(b \overset{\$}{\leftarrow} \{0, 1\}^t\) and sends it to \(\mathcal{P}\)

\(\mathcal{P}\) computes and sends \(s = x^e \times r \mod n\)

\(\mathcal{V}\) checks whether \(s^e \overset{?}{=} X^b R \mod n\)

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

\(s^e / X^b = R = (s')^e / X^{b'} \mod n \implies X^{b' - b} = (s'/s)^e \mod n\)

If \(e\) prime and bigger than \(2^t\), then \(e\) and \(b' - b\) are relatively prime:

Bezout: \(ue + v(b' - b) = 1 \implies X^{v(b' - b)} = (s'/s)^e = X^{1 - ue} \mod n\)

As a consequence: \(X = ((s'/s)^v X^u)^e \implies \text{Special Soundness}\)

Outline

1 Zero-Knowledge Proofs of Knowledge

2 Signatures

   - From Identification to Signature
   - Forking Lemma

3 Zero-Knowledge Proofs of Membership

DL Schnorr Proof

[Schnorr – Eurocrypt ’89 - Crypto ’89]

Setting: \(G = \langle g \rangle\) of order \(q\)

\(\mathcal{P}\) knows \(x\), such that \(y = g^{-x}\) and wants to prove it to \(\mathcal{V}\)

\(\mathcal{P}\) chooses \(k \overset{\$}{\leftarrow} \mathbb{Z}_q^*\), sets and sends \(r = g^k\)

\(\mathcal{V}\) chooses \(h \overset{\$}{\leftarrow} \{0, 1\}^t\) and sends it to \(\mathcal{P}\)

\(\mathcal{P}\) computes and sends \(s = k + xh \mod q\)

\(\mathcal{V}\) checks whether \(r \overset{\$}{=} g^s y^h\)

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 \(x = (s - s')(h' - h)^{-1} \mod q \implies \text{Special Soundness}\)
**Generic Zero-Knowledge Proofs**

**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 \( h \)
- \( h \leftarrow \{0, 1\}^k \) for \( P \)
- \( P \) 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 \((r, h, s)\)

**Zero-Knowledge Proof**
- Proof of knowledge of \( x \)
- \( P \) sends a commitment \( r \)
- \( V \) sends a challenge \( h \)
- \( h = H(m, r) \)
- \( P \) sends the answer \( s \)
- \( V \) checks \((r, h, s)\)

**Signature**
- Key Generation \( \rightarrow (y, x) \)
- 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 soundness): 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
Diffie-Hellman Language

In a group \( G = \langle g \rangle \) of prime order \( q \),
the DDH\((g, h)\) assumption states it is hard to distinguish
\( \mathcal{L} = (u = g^x, v = h^x) \) 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 \( k \overset{R}{\leftarrow} \mathbb{Z}_q^* \), sets and sends \( U = g^k \) and \( V = h^k \)
- \( \mathcal{V} \) chooses \( h \overset{R}{\leftarrow} \{0, 1\}^t \) and sends it to \( \mathcal{P} \)
- \( \mathcal{P} \) computes and sends \( s = k + xh \mod q \)
- \( \mathcal{V} \) 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 \( y = (s - s')(h' - h)^{-1} \mod q \Rightarrow u = g^y \) and \( v = h^y \)
- there exists a witness: Perfect Soundness