Zero-Knowledge Proofs of Knowledge

Saravanan Vijayakumaran
sarva@ee.iitb.ac.in

Department of Electrical Engineering
Indian Institute of Technology Bombay

October 14, 2019

Proofs of Knowledge

- Proofs in which prover asserts knowledge of a secret
- Example
  - Let L_{iso} be the encoding of pairs of graphs which are isomorphic

$$L_{\text{iso}} = \{(G_1, G_2) \mid \exists \phi \text{ such that } \phi : G_1 \to G_2 \text{ is an isomorphism}\}$$

  - Prover claims to know a ϕ, instead of just claiming that $(G_1, G_2) \in L_{\text{iso}}$
- Zero-knowledge proofs are not necessarily proofs of knowledge
- How to capture the notion of a machine knowing something?
- The "something" can be captured by a binary relation
  - Let $R \subset \{0,1\}^* \times \{0,1\}^*$ be a binary relation
  - The language L_R is given by

$$L_R = \{x \mid \exists w \text{ such that } (x, w) \in R\}$$

  - Any w such that $(x, w) \in R$ is called a **witness** for the membership of x in L_R

Proof of Knowledge Definition

- **Main idea:** If a prover P^* claims to know a witness, then this witness should be extractable from P^*
- **Definition:** Let $\kappa : \{0,1\}^* \rightarrow [0,1]$ be a function. A protocol (P, V) is a proof of knowledge for the relation R with knowledge error κ if
  - **Completeness:** If P and V follow the protocol on input x and private input w to P where $(x, w) \in R$, then V always accepts
  - **Knowledge soundness:** There exists a constant $c > 0$ and a **PPT** machine K, called the knowledge extractor, such that for every interactive prover P^* and every $x \in L_R$, the machine K satisfies the following condition:
    Let $\epsilon(x)$ be the probability that V accepts on input x after interacting with P^*. If $\epsilon(x) > \kappa(x)$, then upon input x and oracle access to P^*, the machine K outputs a string w such that $(x, w) \in R$ with probability $\epsilon(x) - \kappa(x)$.
- The knowledge error is the probability of being able to convince a verifier without knowing w

Proof of Knowledge Alternative Definition

- **Main idea:** If a prover P^* claims to know a witness, then this witness should be extractable from P^*
- **Definition:** Let $\kappa : \{0,1\}^* \to [0,1]$ be a function. A protocol (P, V) is a proof of knowledge for the relation R with knowledge error κ if
  - **Completeness:** If P and V follow the protocol on input x and private input w to P where $(x, w) \in R$, then V always accepts
  - **Knowledge soundness:** There exists a constant $c > 0$ and a **probabilistic** machine K, called the knowledge extractor, such that for every interactive prover P^* and every $x \in L_R$, the machine K satisfies the following condition:
    Let $\epsilon(x)$ be the probability that V accepts on input x after interacting with P^*. If $\epsilon(x) > \kappa(x)$, then upon input x and oracle access to P^*, the machine K outputs a string w such that $(x, w) \in R$ within an **expected** number of steps bounded by

$$\frac{|x|^c}{\epsilon(x) - \kappa(x)}.$$

Schnorr Identification Scheme

- Let G be a cyclic group of order q with generator g
- Identity corresponds to knowledge of private key x where $h = g^x$
- A prover wants to prove that she knows x to a verifier without revealing it
  1. Prover picks $k \leftarrow \mathbb{Z}_q$ and sends initial message $I = g^k$
  2. Verifier sends a challenge $r \leftarrow \mathbb{Z}_q$
  3. Prover sends $s = rx + k \bmod q$
  4. Verifier checks $g^s \cdot h^{-r} \stackrel{?}{=} I$
- The knowledge extractor K does the following
  1. After the initial message I from prover, K sends a challenge $r \in \mathbb{Z}_q$
  2. K receives the response s from prover
  3. K rewinds the protocol to the step when I was received
  4. K sends a challenge $r' \neq r$ and receives s' from the prover
  5. K extracts x from the pairs (r, s) and (r', s')
- This protocol is a PoK but not ZK!
- It is however HVZK

Zero-Knowledge Proof of Knowledge

- An interactive proof system is a ZKPoK if it satisfies:
  - **Completeness:** Honest prover convinces honest verifier
  - **Zero-Knowledge:** Malicious verifiers learn nothing more than statement validity
  - **Knowledge soundness:** Ensures prover knows witness

ZKPoK for Quadratic Residuosity

- Interactive protocol for QR of $x = w^2$ modulo $N = pq$
  - P picks $r \xleftarrow{\$} \mathbb{Z}_N^*$ and sends $y = r^2$ to V
  - V picks a bit $b \xleftarrow{\$} \{0, 1\}$ and sends b to P
  - If $b = 0$, P sends $z = r$. If $b = 1$, P sends $z = wr$
  - If $b = 0$, V checks $z^2 = y$. If $b = 1$, V checks $z^2 = xy$
- We already proved completeness and zero-knowledge. Only need to show knowledge soundness
- The knowledge extractor K does the following
  1. After the initial message y from prover, K sends the challenge bit $b = 0$
  2. K receives the response z_0 from prover
  3. K rewinds the protocol to the step when y was received
  4. K sends challenge bit $b = 1$ and receives z_1 from the prover
  5. K extracts w as $\frac{z_1}{z_0}$

ZKPoK for Graph Isomorphism

- An isomorphism ϕ between graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ exists
- Prover and verifier execute the following protocol
  - Prover picks a random permutation π from the set of permutations of V_2
  - Prover calculates $F = \{(\pi(u), \pi(v) \mid (u, v) \in E_2\}$ and sends the graph $G' = (V_2, F)$ to verifier
  - Verifier picks $\sigma \in \{1, 2\}$ randomly and sends it to prover
  - If $\sigma = 2$, then prover sends π to the verifier. Otherwise, it sends $\pi \circ \phi$ to the verifier where $(\pi \circ \phi)(v)$ is defined as $\pi(\phi(v))$
  - If the received mapping is an isomorphism between G_σ and G', the verifier accepts. Otherwise, it rejects
- The knowledge extractor K does the following
  1. After the initial message G' from prover, K sends the challenge $\sigma = 1$
  2. K receives the response ψ_1 from prover
  3. K rewinds the protocol to the step when G' was received
  4. K sends challenge $\sigma = 2$ and receives ψ_2 from the prover
  5. K extracts an isomorphism as $\psi_2^{-1} \circ \psi_1$

References

- *On Σ-protocols*, Ivan Damgård, `http://www.cs.au.dk/~ivan/Sigma.pdf`
- Section 4.7 of *Foundations of Cryptography, Volume I* by Oded Goldreich
- Yehuda Lindell's lecture in the 9th BIU Winter School on Cryptography
  - `https://cyber.biu.ac.il/event/the-9th-biu-winter-school-on-cryptography/`
  - ZKPoKs `http://cyber.biu.ac.il/wp-content/uploads/2018/08/WS-19-3-ZKPOK_D1-5.pdf`
  - ZKPoKs `https://www.youtube.com/watch?v=RvGsjnoYRRg`