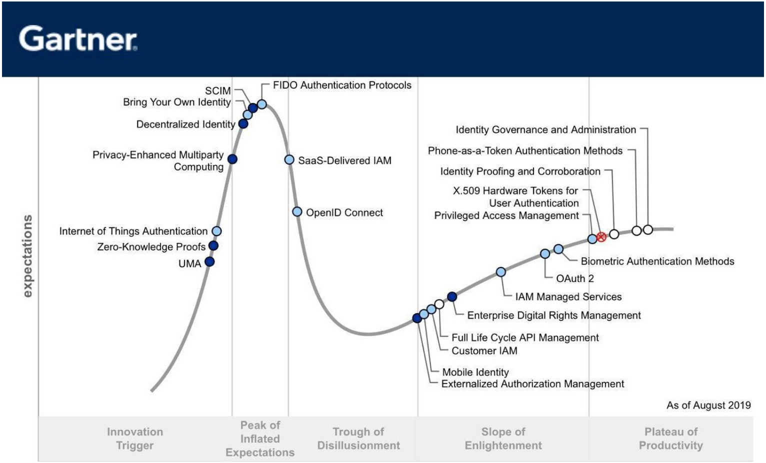
Zero Knowledge Proofs

Saravanan Vijayakumaran
sarva@ee.iitb.ac.in

Department of Electrical Engineering
Indian Institute of Technology Bombay

September 30, 2019

Gartner Hype Cycle for Identity



Source: `https://twitter.com/IdentityMonk/status/1158564314577612800`

Zero Knowledge Proofs

- Proofs that yield nothing beyond the validity of an assertion
- Examples of assertions
  - I know the discrete log of a group element wrt a generator
  - I know an isomorphism between two graphs G_1, G_2
- Proofs are a sequence of statements each of which is an axiom or follows from axioms via derivation rules
  - Traditional proofs do not have explicit provers and verifiers
- ZKPs involve explicit interaction between prover and verifier
- Prover and verifier will be modeled as algorithms or machines
  - Verifier is assumed to be probabilistic polynomial-time (PPT)
  - Prover may or may not be PPT

Examples of Interactive Proofs

- Proving that two chalks have different colours to a colour-blind verifier
- Proof of Quadratic Residuosity
  - For a positive integer N, x is called a quadratic residue modulo N if

$$x = w^2 \bmod N \text{ for some } w$$

  - Suppose $N = pq$ for distinct primes p and q with $|p| = |q| = n$.
  - Without knowing the factorization of N, the best algorithms for checking $x \in QR_N$ run in $\exp\left(\mathcal{O}(n^{\frac{1}{3}})\right)$ steps
  - Using the factorization of N, $x \in QR_N$ can be checked in time which is polynomial in n
- Proof of Quadratic Non-Residuosity
  - Exhaustive checking is not feasible
  - Use an idea similar to the chalks example
- More details on the last two examples `http://cyber.biu.ac.il/wp-content/uploads/2018/08/WS-19-1-ZK-intro.pdf`

Knowledge vs Information

- In information theory, entropy is used to quantify information
- Entropy of a discrete random variable X defined over an alphabet $\mathcal{X}$ is

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

- Knowledge is related to computational difficulty, whereas information is not
  - Suppose Alice and Bob know Alice's public key
  - Alice sends her private key to Bob
  - Bob has not gained new information (in the information-theoretic sense)
  - But Bob now knows a quantity he could not have calculated by himself
- Knowledge is related to publicly known objects, whereas information relates to private objects
  - Suppose Alice tosses a fair coin and sends the outcome to Bob
  - Bob gains one bit of information (in the information-theoretic sense)
  - We say Bob has not gained any knowledge as he could have tossed a coin himself

Modeling Assertions and Proofs

- The complexity class $\mathcal{NP}$ captures the asymmetry between proof generation and verification
- A language is a subset of $\{0,1\}^*$
- Each language $L \in \mathcal{NP}$ has a polynomial-time verification procedure for proofs of statements "$x \in L$"
  - Example: L is the encoding of pairs of finite isomorphic graphs
- Let $R \subset \{0,1\}^* \times \{0,1\}^*$ be a relation
- R is said to be polynomial-time-recognizable if the assertion "$(x, y) \in R$" can be checked in time poly$(|x|, |y|)$
- Each $L \in \mathcal{NP}$ is given by a PTR relation R_L such that

  $$L = \{x \mid \exists y \text{ such that } (x, y) \in R_L\}$$

  and $(x, y) \in R_L$ only if $|y| \leq \text{poly}(|x|)$
- Any y for which $(x, y) \in R_L$ is a proof of the assertion "$x \in L$"

Interactive Proof Systems

- Let $\langle A, B\rangle(x)$ denote the output of B when interacting with A on common input x
- Output 1 is interpreted as "accept" and 0 is interpreted as "reject"

Definition

A pair of interactive machines (P, V) is called an **interactive proof system for a language** L if machine V is polynomial-time and the following conditions hold:

- **Completeness**: For every $x \in L$,

$$\Pr[\langle P, V\rangle(x) = 1] \geq \frac{2}{3}$$

- **Soundness**: For every $x \notin L$ and every interactive machine B,

$$\Pr[\langle B, V\rangle(x) = 1] \leq \frac{1}{3}$$

- Remarks
  - Soundness condition refers to any possible prover while completeness condition refers only to the prescribed prover
  - Prescribed prover is allowed to fail with probability $\frac{1}{3}$
  - Arbitrary provers are allowed to succeed with probability $\frac{1}{3}$
  - These probabilities can be made arbitrarily small by repeating the interaction

Generalized Interactive Proof Systems

Definition

Let $c, s : \mathbb{N} \to \mathbb{R}$ be functions satisfying $c(n) > s(n) + \frac{1}{p(n)}$ for some polynomial $p(\cdot)$. A pair of interactive machines (P, V) is called a **generalized** interactive proof system for a language L with **completeness bound** $c(\cdot)$ and **soundness bound** $s(\cdot)$ if machine V is polynomial-time and the following conditions hold:

- **Completeness**: For every $x \in L$,

$$\Pr\left[\langle P, V\rangle(x) = 1\right] \geq c(|x|)$$

- **Soundness**: For every $x \notin L$ and every interactive machine B,

$$\Pr\left[\langle B, V\rangle(x) = 1\right] \leq s(|x|)$$

The following three conditions are equivalent

- There exists an interactive proof system for L with completeness bound $\frac{2}{3}$ and soundness bound $\frac{1}{3}$
- For every polynomial $q(\cdot)$, there exists an interactive proof system for L with error probabilistic $\max\left(1 - c(|x|), s(|x|)\right)$ bounded above by $2^{-q(|x|)}$
- There exists a polynomial $q(\cdot)$ and a generalized interactive proof system for the language L, with acceptance gap $c(|x|) - s(|x|)$ bounded below by $\frac{1}{q(|x|)}$.

Graph Isomorphism

- Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection $\pi : V_1 \mapsto V_2$ such that $(u, v) \in E_1 \iff (\pi(u), \pi(v)) \in E_2$

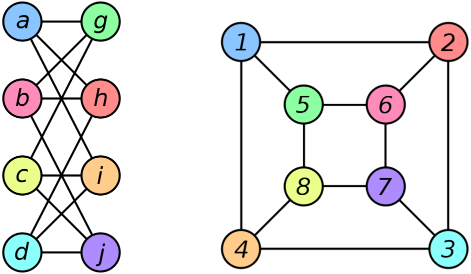


Image source: `https://en.wikipedia.org/wiki/Graph_isomorphism`

$$\pi(a) = 1, \pi(b) = 6, \pi(c) = 8, \pi(d) = 3,$$
$$\pi(g) = 5, \pi(h) = 2, \pi(i) = 4, \pi(j) = 7$$

Interactive Proof for Graph Non-Isomorphism

- Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection $\pi : V_1 \mapsto V_2$ such that $(u, v) \in E_1 \iff (\pi(u), \pi(v)) \in E_2$
- Graphs G_1 and G_2 are non-isomorphic if no such bijection exists
- Prover and verifier execute the following protocol
    - Verifier picks $\sigma \in \{1, 2\}$ randomly and a random permutation π from the set of all permutations over V_σ
    - Verifier calculates $F = \{(\pi(u), \pi(v) \mid (u, v) \in E_\sigma\}$ and sends the graph $G' = (V_\sigma, F)$ to prover
    - Prover finds $\tau \in \{1, 2\}$ such that G' is isomorphic to G_τ and sends τ to verifier
    - If $\tau = \sigma$, verifier accepts claim. Otherwise, it rejects.
- Remarks
    - Verifier is a PPT machine but no known PPT implementation for prover
    - If G_1 and G_2 are not isomorphic, then verifier always accepts
    - If G_1 and G_2 are isomorphic, then verifier rejects with probability at least $\frac{1}{2}$
    - Acceptance gap is bounded from below by $\frac{1}{2}$

Zero Knowledge Interactive Proofs

- Consider an interactive proof system (P, V) for a language L
  - In an interactive proof, we need to guard against a malicious prover
  - To guarantee zero knowledge, we need to guard against a malicious verifier
- Recall that knowledge is related to computational difficulty
- Informal definition
  - An interactive proof system is **zero knowledge** if whatever can be efficiently computed **after interaction** with P on input x can also be efficiently computed from x (**without interaction**)
- Formal definition (ideal)
  - We say (P, V) is **perfect zero knowledge** if for every PPT interactive machine V^* there exists a PPT algorithm M^* such that for every $x \in L$ the random variables $\langle P, V^* \rangle(x)$ and $M^*(x)$ are **identically distributed**
  - M^* is called a **simulator** for the interaction of V^* with P
- Unfortunately, the above definition is too strict
- A relaxed definition is used instead

Perfect Zero Knowledge

Definition

Let (P, V) be an interactive proof system for a language L. We say that (P, V) is **perfect zero knowledge** if for every PPT interactive machine V^* there exists a PPT algorithm M^* such that for every $x \in L$ the following two conditions hold:

1. With probability at most $\frac{1}{2}$, machine M^* outputs a special symbol $\perp$
2. Let $m^*(x)$ be the random variable describing the distribution of $M^*(x)$ conditioned on $M^*(x) \neq \perp$. Then the random variables $\langle P, V^* \rangle(x)$ and $m^*(x)$ are **identically distributed**

- Remarks
  - M^* is called a **perfect simulator** for the interaction of V^* with P
  - By repeated interactions, the probability that the simulator fails to generate the identical distribution can be made negligible
- **Alternative formulation**: Replace $\langle P, V^* \rangle(x)$ with $\text{view}_{V^*}^{P}(x)$
  - A verifier's view consists of messages it receives and any randomness it generates
  - Simulator M^* has to change accordingly

ZK Proof 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
- Remarks
  - Verifier is a PPT machine. If ϕ is known to prover, it is a PPT machine
  - If G_1 and G_2 are isomorphic, then verifier always accepts
  - If G_1 and G_2 are not isomorphic, then verifier rejects with probability $\frac{1}{2}$
  - The prover is perfect zero knowledge (to be argued)

Simulator for Graph Isomorphism Transcript

- For an arbitrary PPT verifier V^*, $\text{view}_{V^*}^{P}(x) = \langle G', \sigma, \psi \rangle$ where ψ is an isomorphism between G_σ and G'
- The simulator M^* uses V^* as a subroutine
- On input (G_1, G_2), simulator randomly picks $\tau \in \{1, 2\}$ and generates a random isomorphic copy G'' of G_τ
    - Note that G'' is identically distributed to G'
- Simulator gives G'' to V^* and receives $\sigma \in \{1, 2\}$ from it
    - V^* is asking for an isomorphism from G_σ to G''
- If $\sigma = \tau$, then the simulator can provide the isomorphism $\pi : G_\tau \mapsto G''$
- If $\sigma \neq \tau$, then the simulator outputs $\perp$
- If the simulator does not output $\perp$, then $\langle G'', \tau, \pi \rangle$ is identically distributed to $\langle G', \sigma, \psi \rangle$

ZK Proof 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 $z^2 = xy$
- If $x \in QR_N$, then V always accepts
- We want to prove that if $x \notin QR_N$, then for any P^*

$$\Pr[\langle P^*, V\rangle(x) = 1] \leq \frac{1}{3}$$

- Using the fact that QR_N is a group, we can argue that

$$\Pr[\langle P^*, V\rangle(x) = 1] \geq \frac{2}{3} \implies x \in QR_N$$

- For an arbitrary PPT verifier V^*, $\text{view}_{V^*}^P(x) = \langle y, b, z\rangle$ where $z^2 = x^b y$
  - To show the protocol is ZK, consider a simulator M^* which does the following
  - M^* picks $z \xleftarrow{\$} \mathbb{Z}_N^*$ and $b \xleftarrow{\$} \{0, 1\}$
  - M^* sets $y = \frac{z^2}{x^b}$
  - If $V^*(y) = b$, then M^* outputs $\langle y, b, z\rangle$. Otherwise, M^* outputs $\perp$

ZK Proof for Quadratic Non-Residuosity

- Interactive protocol for QNR of x modulo $N = pq$
  - V picks $y \xleftarrow{\$} \mathbb{Z}_N^*$ and a bit $b \xleftarrow{\$} \{0, 1\}$
  - If $b = 0$, V sends $z = y^2$. If $b = 1$, V sends $z = xy^2$
  - If $z \in QR_N$, P sends $b' = 0$. If $z \in \overline{QR}_N$, P sends $b' = 1$
  - V accepts if $b' = b$
- If $x \notin QR_N$, then V always accepts. Otherwise, it rejects with probability $\frac{1}{2}$
- The above protocol is HVZK but **not** ZK!
- Consider a PPT verifier V^* which wants to find out if some $u \in \mathbb{Z}_N^*$ is in QR_N
  - By replacing x in the above protocol with u, verifier V^* can get information about u
  - If the protocol was ZK, then there exists a PPT M^* which can get the same information without interacting with P
  - This contradicts the non-existence of PPT algorithms for checking membership in QR_N
- **Solution:** V has to prove that it either knows the square root of z or zx^{-1} to P
- The number of interaction rounds increases from 2 to 4

ZK Proof for Quadratic Non-Residuosity

- ZK Interactive protocol for QNR of x modulo $N = pq$
  - V picks $y \xleftarrow{\$} \mathbb{Z}_N^*$ and a bit $b \xleftarrow{\$} \{0,1\}$
  - If $b = 0$, V sends $z = y^2$. If $b = 1$, V sends $z = xy^2$
  - For $1 \leq j \leq m$,
    - V picks $r_{j,1}, r_{j,2} \xleftarrow{\$} \mathbb{Z}_N^*$ and $\text{bit}_j \xleftarrow{\$} \{0,1\}$
    - V computes $\alpha_j = r_{j,1}^2$ and $\beta_j = xr_{j,2}^2$.
    - If $\text{bit}_j = 1$, V sends $\text{pair}_j = (\alpha_j, \beta_j)$. If $\text{bit}_j = 0$, V sends $\text{pair}_j = (\beta_j, \alpha_j)$.
  - P sends V a bit string $[i_1, i_2, \ldots, i_m] \in \{0,1\}^m$
  - V sends P the sequence $v_1, v_2, \ldots, v_m$
    - If $i_j = 0$, then $v_j = (r_{j,1}, r_{j,2})$.
    - If $i_j = 1$, then $v_j = yr_{j,1}$ if $b = 0$. So V sends a square root of $z\alpha_j$
    - If $i_j = 1$, then $v_j = xyr_{j,2}$ if $b = 1$. So V sends a square root of $z\beta_j$
  - P checks the following:
    - If $i_j = 0$, P checks if $(r_{j,1}^2, r_{j,2}^2 x)$ equals pair_j, possibly with elements in the pair interchanged.
    - If $i_j = 1$, P checks if $v_j^2 z^{-1}$ is a member of pair_j.
  - If all checks pass and $z \in QR_N$, P sends $b' = 0$. If $z \in \overline{QR}_N$, P sends $b' = 1$
  - V accepts if $b' = b$

ZK Proofs for $\mathcal{NP}$

- **Goal:** To construct ZK proofs for every language in $\mathcal{NP}$
- Possible if we assume the existence of perfectly binding and computationally hiding commitment schemes
  - **Example:** El Gamal commitment scheme
  - Let G, H be generators of a group $\mathcal{G}$ of order p
  - Discrete logarithms are assumed to be hard to compute in $\mathcal{G}$
  - G and H have unknown discrete logarithms wrt each other
  - El Gamal commitment to a message $m \in \mathbb{Z}_p$ is given by

$$\text{Com}_{G,H}(m, r) = (rG, rH + mG)$$

- In contrast, Pedersen commitments are perfectly hiding and computationally binding
- $\mathcal{NP}$-complete languages = "Hardest" $\mathcal{NP}$ languages
  - **Examples:** SAT, Graph 3-coloring
- Proof strategy
  - Give a ZK proof of graph 3-coloring
  - Since every $\mathcal{NP}$ language can be reduced to graph 3-coloring, we are done

Graph 3-Coloring

- Assigning one of three colors to each vertex such that no two adjacent vertices have the same color

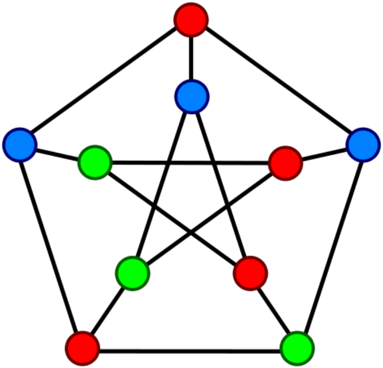


Image source: `https://en.wikipedia.org/wiki/Graph_coloring`

ZK Proof for Graph 3-Coloring

- Common input: A simple 3-colorable graph $G = (V, E)$ where $|V| = n$ and $V = \{1, 2, \ldots, n\}$
- Prover has a 3-coloring of G given by $\psi : V \to \{1, 2, 3\}$ such that $\psi(u) \neq \psi(v)$ for all $(u, v) \in E$
- Interactive proof
  1. Prover selects a random permutation $\pi : \{1, 2, 3\} \to \{1, 2, 3\}$ and sets $\phi(v) = \pi(\psi(v))$
  2. Prover computes commitments $c_v = \text{com}(\phi(v))$ for all $v \in V$ and sends $c_1, c_2, \ldots, c_n$ to verifier
  3. Verifier selects an edge $(u, v) \in E$ and sends it to prover
  4. Prover opens the commitments of the colors $\phi(u)$ and $\phi(v)$
  5. Verifier checks commitment openings and if $\phi(u) \neq \phi(v)$
- **Completeness**: If G is 3-colorable, verifier accepts with probability 1
- **Soundness**: If G is not 3-colorable, there exists at least one edge with $\phi(u) = \phi(v)$ and the verifier rejects with probability at least $\frac{1}{|E|}$
- The acceptance gap is $\frac{1}{|E|}$ which is bounded below by $1/\binom{n}{2}$
- **Zero knowledge**: For any V^*, consider a simulator M^* which independently selects n values $e_1, e_2, \ldots, e_n$ from $\{1, 2, 3\}$ and creates commitments to each of them. If the query edge from V^* is (u, v), then $e_u \neq e_v$ with probability $\frac{2}{3}$ and M^* opens the commitments. If $e_u = e_v$, M^* outputs $\perp$.

References

- Sections 4.1, 4.2, 4.3, 4.4.2 of *Foundations of Cryptography, Volume I* by Oded Goldreich
- Alon Rosen's lecture in the 9th BIU Winter School on Cryptography
  - `https://cyber.biu.ac.il/event/the-9th-biu-winter-school-on-cryptography/`
  - ZKP intro `https://www.youtube.com/watch?v=6uGimDYZPMw`
  - ZKP intro `http://cyber.biu.ac.il/wp-content/uploads/2018/08/WS-19-1-ZK-intro.pdf`
  - ZK for NP `https://www.youtube.com/watch?v=cQ-BI1WWzjU`
  - ZK for NP `http://cyber.biu.ac.il/wp-content/uploads/2018/08/WS-19-2-ZK-for-NP.pdf`
  - Widgerson interview is in the first 8 minutes of `https://www.youtube.com/watch?v=cAI7Iw_bkZs`
- *The Knowledge Complexity of Interactive Proof Systems*, S. Goldwasser, S. Micali, C. Rackoff, 1989. `https://doi.org/10.1137/0218012`