Succinct NIZKs

Zhewei Xu

In CS 476, we have presented two methods to create NIZKs for the class NP. However, the construction has several issues. ROM is built upon an unconstructable, idealized oracle machine. CRS requires a string of extreme length, i.e. $O(n^9)$ for HAMCYCLE.[1] In 2013, Bitansky et al. provides a succinct method to construct and verify NIZKs through Linear PCP.[2]

PCP/ Linear PCP

Unfortunately, we will treat PCP as a black box, but only use some of its properties. Informally, in PCP, a prover $\mathcal{P}_{PCP}$ takes a problem x, a witness w, and produces a proof $\pi \in \{0,1\}^{poly}$. A verifier $\mathcal{V}_{PCP}$ queries 3 random bits i_1, i_2, i_3, receives $\pi[i_1], \pi[i_2], \pi[i_3]$, and outputs T or F. PCP is complete and sound.

A linear PCP (LPCP) is similar to PCP. The prover $\mathcal{P}_{LPCP}$ produces $\mathbb{F}^u$, a degree u polynomial over a field $\mathbb{F}$, represented as a vector. The verifier $\mathcal{V}_{LPCP}$ queries a series of vectors $q \in \mathbb{F}^u$, and receives the inner product, denoted as $\langle \pi, q_1 \rangle, \langle \pi, q_2 \rangle$...

From PCP to SNARGs

NP can be reduced to CIRCUITSAT. The proof is a circuit C of size $\Omega(|C|)$. The verifier $\mathcal{V}$ checks an assignment of C in $\Omega(|x| + |C|)$.

An NIZK is succinct if the proof is of size $O(poly(|x|, log|C|))$, and the runtime is in $O(poly(|x|, \lambda, log|C|))$, where λ is a statistical security parameter.

[1] Katz. 2004. UMD, CMSC 858K: Advanced topics in cryptography. Lecture 13.

[2] Bitansky, Chiesa, Ishai, Ostrovsky, Paneth. 2013. Succinct Non-Interactive Arguments via Linear Interactive Proofs.

Using the CRS[3] model, we outline the procedure to create a SNARG.

1. $\mathcal{V}$ generates queries, $q_1, ..., q_k$, and the state of the LPCP, τ. The queries are embedded into $crs = (q_1, ..., q_k)$, which is shared by both parties. τ is kept secret by $\mathcal{V}$.
2. $\mathcal{P}$ computes $(x, w) \to \pi \in \mathbb{F}^u$. $\mathcal{P}$ then computes $\langle \pi, q_1 \rangle, \langle \pi, q_2 \rangle, ...$ and sends them to $\mathcal{V}$.
3. $\mathcal{V}$ uses LPCP to verify the results.

The protocol is complete by LPCP.

The protocol is succinct. The proof exchanged between $\mathcal{P}$ and $\mathcal{V}$ are a series of dot products w.r.t to C, $O(log|C|)$. The total number of such dot products depends on the number of runs required to satisfy λ, $O(\lambda)$. $\mathcal{V}$ is able to verify the result in $O(|x| \cdot |\mathbb{F}|) = O(|x| \cdot \lambda)$ by LPCP.

The protocol is *not* sound.

1. In PCP and LPCP, $\mathcal{V}$ generates q after $\mathcal{P}$ has produced π. However. for non-interaction, q must be generated beforehand. Thus, a malicious $\mathcal{P}$ may tailor his proof to q. This is overcome by a homomorphic encryption of q. Let q_{ij} := i-th query and j-th degree. $\mathcal{V}$ generates $\rho_{ij} := ENC(PubK, q_{ij})$ to hide the positioning of his queries. $\mathcal{P}$ then calculates ρ'_{ij},

   $$\rho'_{ij} = ENC(PubK, \langle \pi, q_i \rangle) = \sum_j \pi_j ENC(PubK, q_{ij}) = \sum_j \pi_j \rho_{ij}$$

   $\mathcal{V}$ decrypts the result using his secret key.

[3] I apologize for the inconsistencies regarding the abbreviation CRS. CRS can both stand for “common reference string” or “common random string”. In our CS 476 report, it is meant to be “common random string”. Here, we use “common reference string”. That being said, it is possible to convert one to the other, e.g. the $O(n^9)$ construction in HAMCYCLE.

2. A malicious $\mathcal{P}$ can cheat by producing ρ'_{ij} through a non-linear combination of $\langle \pi, q \rangle$. This is enforced by forcing a linear-only encryption scheme.

3. A malicious $\mathcal{P}$ can calculate $\langle \pi_i, q_i \rangle$ using different π's, creating different proofs for different queries. A consistency check is added, in which $\mathcal{V}$ generates $\{\alpha_1, ..., \alpha_k\} \in \mathbb{F}$ secretly, and adds a query $q_{k+1} = \sum_i \alpha_i q_i$. $\mathcal{V}$ checks that $\sum_i \alpha_i \langle \pi_i, q_i \rangle = \langle \pi_{k+1}, q_{k+1} \rangle$.

   - If $\mathcal{P}$ is honest, that is, $\forall i, \pi_i = \pi_{k+1} = \Pi$,

   $$\sum_i \alpha_i \langle \Pi, q_i \rangle = \langle \Pi, \sum_i \alpha_i q_i \rangle = \langle \Pi, q_{k+1} \rangle = \langle \pi_{k+1}, q_{k+1} \rangle$$

   - If $\mathcal{P}$ is dishonest, that is, $\exists i, j, \pi_i \neq \pi_j$,

   $$\begin{aligned} & \sum_i \alpha_i \langle \pi_i, q_i \rangle - \langle \pi_{k+1}, q_{k+1} \rangle \\ &= \sum_i \alpha_i \langle \pi_i, q_i \rangle - \langle \pi_{k+1}, \sum_i \alpha_i q_i \rangle \\ &= \sum_i \alpha_i \langle \pi_i, q_i \rangle - \sum_i \alpha_i \langle \pi_{k+1}, q_i \rangle \\ &= \sum_i \alpha_i \langle \pi_i - \pi_{k+1}, q_i \rangle \\ &\neq 0 \text{ (This check fails w.p. } \frac{1}{\mathbb{F}}) \end{aligned}$$

Conclusion

Again, it is important to note that PCPs and LPCPs on their own are purely computational models, and do not exist. Nevertheless, many constructions are proposed, notably Arora et al.'s construction through Walsh-Hadamard code,[4] and Gennaro et al.'s construction through QSP,[5] which is currently the most commonly used method.

The area is still relatively new and under heavy research, perhaps for faster runtimes, and stronger security.

Main sources

- Florian. 2020. Stanford, CS 355: Topics in Cryptography. Lecture 11.
- Dima. 2019. Stanford, CS 355: Topics in Cryptography. Lecture 17: Succinct Non-interactive Arguments.

[4] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and the Hardness of Approximation Problems. FOCS'92, JACM, 1998.

[5] Gennaro, Gentry, Parno, Raykova. 2013. Quadratic Span Programs and Succinct NIZKs without PCPs.