# [name]: Efficient zero-knowledge proof with optimal prover computation

November 1, 2018

## 1 Preliminary

In this section, we will introduce some useful results and definitions.

### 1.1 Interactive Proof

Traditional proof involves two static objects: a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. The prover $\mathcal{P}$ takes a statement $x$ as input and generate a string $\pi$ as a proof, then the verifier $\mathcal{V}$ checks if the statement $x$ and proof $\pi$ are correct. A interactive proof is a stronger notion of proof, it allows a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ of the validity of some statement. The interactive proof runs in several rounds, allows the verifier to ask questions in each round based on prover's answers of previous rounds. We phrase this in term of $\mathcal{P}$ trying to convince $\mathcal{V}$ that $f(x) = 1$. The proof system is interesting iff the running time of $\mathcal{V}$ is less than the time of directly computing the function $f$.

We formalize the "interactive proof" in the following:

**Definition 1.** Let f be a boolean function. A pair of interactive machines $\langle \mathcal{P}, \mathcal{V} \rangle$ is an interactive proof for f with soundness $\epsilon$ if the following holds:

- **Completeness.** For every $x$ such that $f(x) = 1$ it holds that $\Pr[\langle \mathcal{P}, \mathcal{V} \rangle(x) = \mathit{accept}] = 1$.
- **$\epsilon$-Soundness.** For any $x$ with $f(x) \neq 1$ and any $\mathcal{P}^*$ it holds that $\Pr[\langle \mathcal{P}^*, \mathcal{V} \rangle = \mathit{accept}] \leq \epsilon$

### 1.2 Sum Check Protocol

The sum check problem is a fundamental problem that serves as a building block for varies applications. Informally the problem requires us to sum on a binary hypercube $(b_1, b_2, ..., b_l)$ for a given polynomial $g(x_1, x_2, ..., x_l)$. Directly compute the function requires exponential computation, Lund et al. [3] proposed a interactive proof protocol such that a computational unbounded prover $\mathcal{P}$ can convince a computational bounded verifier $\mathcal{V}$ that

$$H = \sum_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} ... \sum_{b_l \in \{0,1\}} g(b_1, b_2, ..., b_l)$$

Using this protocol, even a polynomial bounded verifier can verify the statement above. Now we formally define the problem and provide a description of the protocol.

**Definition 2.** Let $g$ be a $l$-variate polynomial $g(b_1, b_2, ..., b_l)$ over a field $\mathbb{F}$; the prover's goal is to convince that

$$H = \sum_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} ... \sum_{b_l \in \{0,1\}} g(b_1, b_2, ..., b_l)$$

**Protocol 1** (**Sum Check**). *The protocol proceeds in $l$ rounds.*

- *In the first round, the prover sends a univariate polynomial*

$$g_1(x_1) \stackrel{def}{=} \sum_{b_2 \in \{0,1\}} ... \sum_{b_l \in \{0,1\}} g(x_1, b_2, b_3, ..., b_l)$$

  *, the verifier checks $H = g_1(0) + g_1(1)$. Then the verifier sends a random number $r_1$ to prover, and sets $G_1 \stackrel{def}{=} g_1(r_1)$.*

- *In $i$-th round, where $2 \le i \le l-1$, the prover sends*

$$g_i(x_i) \stackrel{def}{=} \sum_{b_{i+1} \in \{0,1\}} \sum_{b_{i+2} \in \{0,1\}} ... \sum_{b_l \in \{0,1\}} g(r_1, r_2, ..., r_{i-1}, x_i, b_{i+1}, b_{i+2}, ..., b_l)$$

  *. Then the verifier checks $G_{i-1} = g_i(0) + g_i(1)$, and then sends a random number $r_i$ to prover. The verifier sets $G_i \stackrel{def}{=} g_i(r_i)$.*

- *In $l$-th round, the prover sends*

$$g_l(x_l) \stackrel{def}{=} g(r_1, r_2, ..., r_{l-1}, x_l)$$

  *, the verifier checks $G_{l-1} = g_l(0) + g_l(1)$. Then verifier generate a random number $r_l$ and sets $G_l \stackrel{def}{=} g_l(r_l)$. The verifier also compute Answer $\stackrel{def}{=} g(r_1, r_2, ..., r_l)$ locally. Verifier will accept iff $G_l =$ Answer.*

**Definition 3** (**Multi-linear Extension**). Let $V : \{0,1\}^l \to \mathbb{F}$ be a function. A *multi-linear extension* is a unique polynomial $\tilde{V} : \mathbb{F}^l \to \mathbb{F}$ defined as:

$$\tilde{V}(x_1, x_2, ..., x_l) \stackrel{def}{=} \sum_{b \in \{0,1\}^l} \prod_{i=1}^{l} [((1-x_i)(1-b_i) + x_i b_i) \times V(b)]$$

where $b_i$ is $i$-th bit of b.

### 1.3 CMT Protocol

CMT Protocol (Cormode et al.) [1] is based on the work of Goldwasser et al. [2], gives us a efficient implementation of GKR protocol. We will futhur improve CMT protocol to optimize prover time to optimal and make it zero knowledge. In this section, we will briefly introduce the old protocol.

## References


[1] G. Cormode, M. Mitzenmacher, and J. Thaler, *Practical verified computation with streaming interactive proofs*, in Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12, New York, NY, USA, 2012, ACM, pp. 90–112.

[2] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, *Delegating computation: Interactive proofs for muggles*, J. ACM, 62 (2015), pp. 27:1–27:64.

[3] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, *Algebraic methods for interactive proof systems*, J. ACM, 39 (1992), pp. 859–868.