Lightweight Sublinear Arguments Without a Trusted Setup

Cauchy Huang

The National University of Singapore

Cauchy_0326xz@outlook.com

2024/04/12

Overview

1. Contributions
2. MPC in ZKP
   - MPC Model
   - MPC combined with ZKP - IKOS Protocol
3. Self-Contained construction of zkIPCP
   - Error Collecting Codes based ZKP
   - Testing Interleaved Linear Codes
   - Testing Linear Constraints over Interleaved RS Codes
   - Testing Quadratic Constraints over Interleaved RS Codes
   - IPCP for Arithmetic Circuits
   - Making it ZK

Contributions

- A zkSNARG with following properties:
  - sublinear - $\sqrt{|C|}$ in communication complexity
  - only employs symmetric-key primitives - transparent setup and light-weight implementation
  - improved amortized communication complexity with sublinear verification time

MPC Model

Def. MPC Model

Parties: A sender client S, n servers $P_1, \ldots, P_n$ and a receiver client R.
Input: S, R and P_i has as public input a statement x of some NP relation L, and S has additionally a secret witness w.
Interactive: P_i obtains random shares from S, and outputs $L(x, w)$ to R.

Definition 3.1 (Correctness). *We say that* Π *realizes a deterministic* $n+1$*-party functionality* $(x, r_1, \ldots, r_n)$ *with perfect (resp., statistical) correctness if for all inputs* $(x, r_1, \ldots, r_n)$*, the probability that the output of some player is different from the output of* f *is* 0 *(resp., negligible in* κ*), where the probability is over the independent choices of the random inputs* $r_1, \ldots, r_n$*.*

Definition 3.2 (t_p-Privacy). *Let* $1 \leq t_p < n$*. We say that* Π *realizes* f *with perfect* t_p*-privacy if there is a* PPT *simulator* $\mathcal{S}$ *such that for any inputs* $(x, r_1, \ldots, r_n)$ *and every set of corrupted players* $T \subset [n]$*, where* $|T| \leq t_p$*, the joint view* $\mathbf{View}_T(x, r_1, \ldots, r_n)$ *of players in* T *is distributed identically to* $\mathcal{S}(T, x, \{r_i\}_{i \in T}, f_T(x, r_1, \ldots, r_n))$*.*

Definition 3.3 (Statistical t_r-Robustness). *We say that* Π *realizes* f *with statistical* t_r*-robustness if it is perfectly correct in the presence of a honest-but-curious adversary as in Definition 3.1, and furthermore for any (unbounded) active adversary that* adaptively *corrupts a set* T *of at most* t_r *players, and for any inputs* $(x, r_1, \ldots, r_n)$*, the following robustness property holds. If there is no* $(r_1, \ldots, r_n)$ *such that* $f(x, r_1, \ldots, r_n) = 1$*, then the probability that* R *outputs* 1 *in an execution of* Π *in which the inputs of the honest players are consistent with* $(x, r_1, \ldots, r_n)$ *is negligible in* κ *where* κ *is a statistical parameter that the protocol* Π *receives as input.*

IKOS Protocol

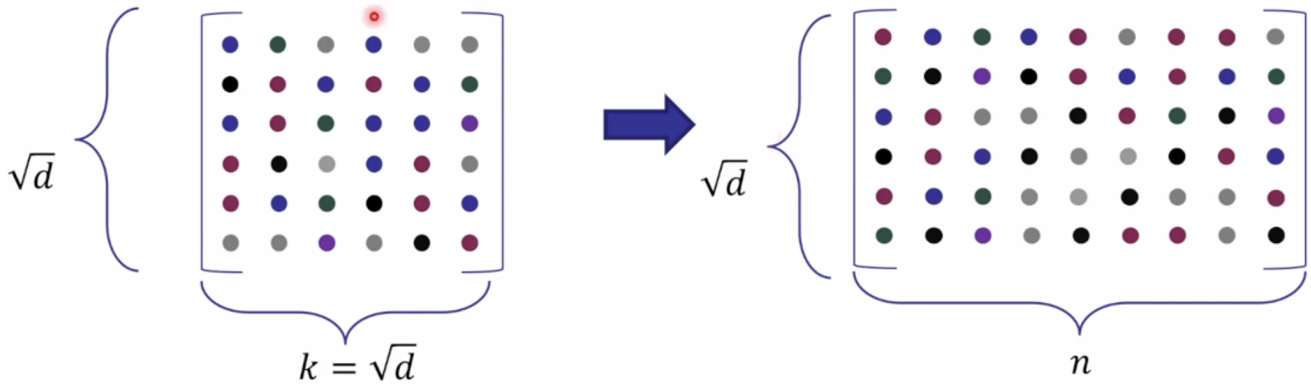
- Reference: https://dl.acm.org/doi/abs/10.1145/1250790.1250794

Protocol Π_{ZKIPCP}.

- **Input:** The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ share a common input statement x and a circuit description C that realizes $\mathcal{R}$. $\mathcal{P}$ additionally has input w such that $\mathcal{R}(x, w) = 1$.
- **Oracle** π**:** The prover runs the MPC protocol Π "in-its-head" as follows. It picks a random input r_S and invokes S on $(x, w; r_S)$ and a random input r_i for every server P_i. The prover computes the views of the servers up to the end of Phase 1 in Π, denoted by $(V_1, \ldots, V_n)$, and sets the oracle as the n symbols $(V_1, \ldots, V_n)$.
- **The interactive protocol.**
  (1) $\mathcal{V}$ picks a random challenge r of length l and sends it to the sender.
  (2) Upon receiving the challenge r, prover $\mathcal{P}$ sends the view V of R.[2]
  (3) $\mathcal{V}$ computes the output of R from the view and checks if R does not abort. It then picks a random subset Q of $[n]$ of size t_p uniformly at random (with repetitions) from $[n]$, and queries the oracle on Q.
  (4) $\mathcal{V}$ obtains from the oracle the views of the servers in Q.
  (5) $\mathcal{V}$ aborts if the views of the servers are *inconsistent* with the view of R. Otherwise, it accepts and halts.

Error Collecting Codes based ZKP

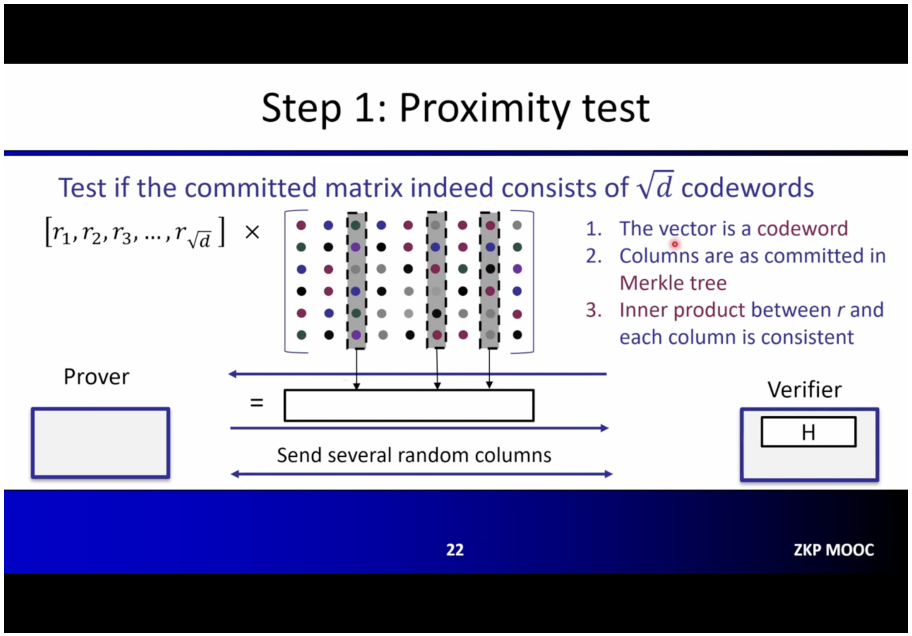
Encoding the polynomial

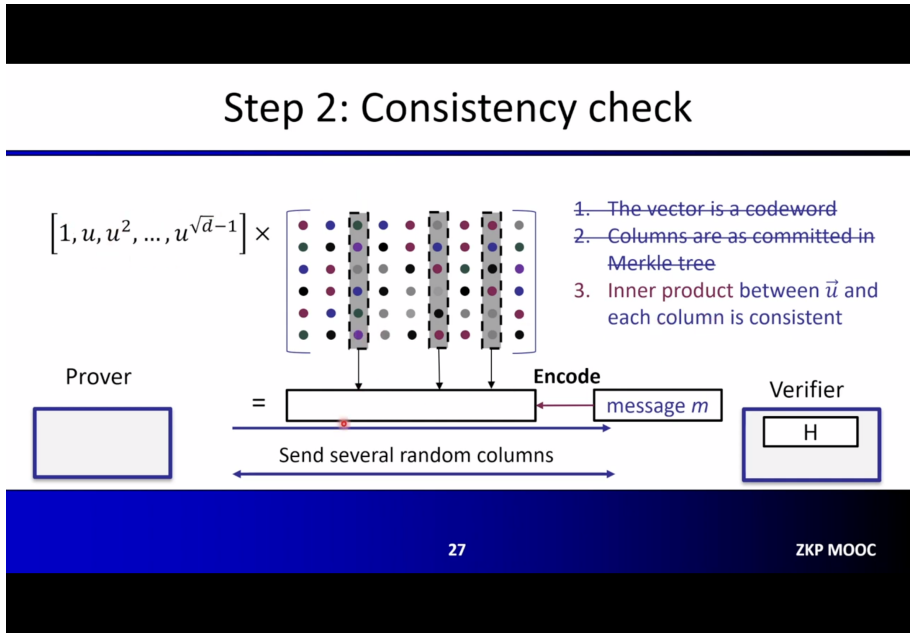


Encode each row with a linear code

18 ZKP MOOC

Error Collecting Codes based ZKP





Testing Interleaved Linear Codes

- **Oracle:** A purported L^m-codeword U. Depending on the context, we may view U either as a matrix in $\mathbb{F}^{m \times n}$ in which each row U_i is a purported L-codeword, or as a sequence of n symbols $(U[1], \ldots, U[n])$, $U[j] \in \mathbb{F}^m$.
- **Interactive testing:**
  (1) $\mathcal{V}$ picks a random linear combinations $r \in \mathbb{F}^m$ and sends r to $\mathcal{P}$.
  (2) $\mathcal{P}$ responds with $w = r^T U \in \mathbb{F}^n$.
  (3) $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols $U[j]$, $j \in Q$.
  (4) $\mathcal{V}$ accepts iff $w \in L$ and w is consistent with U_Q and r. That is, for every $j \in Q$ we have $\sum_{i=1}^{m} r_j \cdot U_{i,j} = w_j$.

Encoded Messages

Definition 4.5 (Encoded message). *Let* $L = \mathrm{RS}_{\mathbb{F},n,k,\eta}$ *be an RS code and* $\zeta = (\zeta_1, \ldots, \zeta_\ell)$ *be a sequence of distinct elements of* $\mathbb{F}$ *for* $\ell \leq k$. *For* $u \in L$ *we define the message* $\mathrm{Dec}_\zeta(u)$ *to be* $(p_u(\zeta_1), \ldots, p_u(\zeta_\ell))$, *where* p_u *is the polynomial (of degree* $< k$*) corresponding to* u. *For* $U \in L^m$ *with rows* $u^1, \ldots, u^m \in L$, *we let* $\mathrm{Dec}_\zeta(U)$ *be the length-*$m\ell$ *vector* $x = (x_{11}, \ldots, x_{1\ell}, \ldots, x_{m1}, \ldots, x_{m\ell})$ *such that* $(x_{i1}, \ldots, x_{i\ell}) = \mathrm{Dec}_\zeta(u^i)$ *for* $i \in [m]$. *Finally, when* ζ *is clear from the context, we say that* U encodes x *if* $x = \mathrm{Dec}_\zeta(U)$.

Testing Linear Constraints over Interleaved RS Codes

Test-Linear-Constraints-IRS$(\mathbb{F}, L = \mathrm{RS}_{\mathbb{F},n,k,\eta}, m, t, \zeta, A, b; U)$

- **Oracle:** A purported L^m-codeword U that should encode a message $x \in \mathbb{F}^{m\ell}$ satisfying $Ax = b$.
- **Interactive testing:**
  (1) $\mathcal{V}$ picks a random vector $r \in \mathbb{F}^{m\ell}$ and sends r to $\mathcal{P}$.
  (2) $\mathcal{V}$ and $\mathcal{P}$ compute

$$r^T A = (r_{11}, \ldots, r_{1\ell}, \ldots, r_{m1}, \ldots, r_{m\ell})$$

  and, for $i \in [m]$, let $r_i(\cdot)$ be the unique polynomial of degree $< \ell$ such that $r_i(\zeta_c) = r_{ic}$ for every $c \in [\ell]$.

  (3) $\mathcal{P}$ sends the $k + \ell - 1$ coefficients of the polynomial defined by $q(\bullet) = \sum_{i=1}^{m} r_i(\bullet) \cdot p_i(\bullet)$, where p_i is the polynomial of degree $< k$ corresponding to row i of U.
  (4) $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols $U[j]$, $j \in Q$.
  (5) $\mathcal{V}$ accepts if the following conditions hold:
    (a) $\sum_{c\in[\ell]} q(\zeta_c) = \sum_{i\in[m],c\in[\ell]} r_{ic} b_{ic}$.
    (b) For every $j \in Q$, $\sum_{i=1}^{m} r_i(\eta_j) \cdot U_{i,j} = q(\eta_j)$.

Testing Quadratic Constraints over Interleaved RS Codes

Test-Quadratic-Constraints-IRS$(\mathbb{F}, L = \mathrm{RS}_{\mathbb{F},n,k,\eta}, m, t, \zeta, a, b; U^x, U^y, U^z)$

- **Oracle:** Purported L^m-codewords U^x, U^y, U^z that should encode messages $x, y, z \in \mathbb{F}^{m\ell}$ satisfying $x \odot y + a \odot z = b$.
- **Interactive testing:**
  (1) Let $U^a = \mathrm{Enc}_\zeta(a)$ and $U^b = \mathrm{Enc}_\zeta(b)$.
  (2) $\mathcal{V}$ picks a random linear combinations $r \in \mathbb{F}^m$ and sends r to $\mathcal{P}$.
  (3) $\mathcal{P}$ sends the $2k-1$ coefficients of the polynomial p_0 defined by $p_0(\bullet) = \sum_{i=1}^m r_i \cdot p_i(\bullet)$, where $p_i(\bullet) = p_i^x(\bullet) \cdot p_i^y(\bullet) + p_i^a(\bullet) \cdot p_i^z(\bullet) - p_i^b(\bullet)$, and where p_i^x, p_i^y, p_i^z are the polynomials of degree $< k$ corresponding to row i of U^x, U^y, U^z, and p_i^a, p_i^b are the polynomials of degree $< \ell$ corresponding to row i of U^a, U^b.
  (4) $\mathcal{V}$ picks a random index set $Q \subset [n]$ of size t, and queries $U^x[j], U^y[j], U^z[j], j \in Q$.
  (5) $\mathcal{V}$ accepts if the following conditions hold:
    (a) $p_0(\zeta_c) = 0$ for every $c \in [\ell]$.
    (b) For every $j \in Q$, it holds that

$$\sum_{i=1}^m r_i \cdot \left[U_{i,j}^x \cdot U_{i,j}^y + U_{i,j}^a \cdot U_{i,j}^z - U_{i,j}^b \right] = p_0(\eta_j).$$

Final IPCP Protocol for Arithmetic Circuits

- **The interactive protocol:**
  $\mathcal{V}$ and $\mathcal{P}$ run the following tests.
  (1) // Test if U is e-close to a code in L^{4m}
  Test-Interleaved$(\mathbb{F}, L, 4m, t; U)$

(2) // Test if addition gates are correct.
Test-Linear-Constraints-IRS
$(\mathbb{F}, L, m, t, \zeta, P_{\text{add}}, \mathbf{0}^{m\ell}; U^w)$

(3) // Test if multiplication gates are correct.
- **Test-Linear-Constraints-IRS**
$\left(\mathbb{F}, L, 2m, t, \zeta, \left[I_{m\ell} | -P_x\right], \mathbf{0}^{2m\ell}; \begin{bmatrix} U^x \\ U^w \end{bmatrix}\right)$
- **Test-Linear-Constraints-IRS**
$\left(\mathbb{F}, L, 2m, t, \zeta, \left[I_{m\ell} | -P_y\right], \mathbf{0}^{2m\ell}; \begin{bmatrix} U^y \\ U^w \end{bmatrix}\right)$
- **Test-Linear-Constraints-IRS**
$\left(\mathbb{F}, L, 2m, t, \zeta, \left[I_{m\ell} | -P_z\right], \mathbf{0}^{2m\ell}; \begin{bmatrix} U^a \\ U^w \end{bmatrix}\right)$
- **Test-Quadratic-Constraints-IRS**
$(\mathbb{F}, L, m, t, \zeta, (-1)^{m\ell}, \mathbf{0}^{m\ell}; U^x, U^y, U^z)$

Affine

4.6.1 ZK Testing of Interleaved Linear Codes. Recall that in the verification algorithm **Test-Interleaved** from Section 4.1, $\mathcal{V}$ obtains a linear combination of the form $w = r^T U$, where $U \in \mathbb{F}^{m \times n}$ is a matrix whose rows should be codewords in L. A natural approach for making this linear combination hide U is by allowing the prover to add to the rows of U an additional random codeword u' that is used for blinding.

A simple implementation of this idea that provides a slightly inferior soundness guarantee is the following. Apply the algorithm **Test-Interleaved** to L^{m+1}, with an extended oracle U' whose first m rows contain U and whose last row is u'. Letting $w' = r^T U + r' u'$ be the random linear combination obtained by $\mathcal{V}$, the test fails to be zero-knowledge when $r' = 0$, which occurs with $1/|\mathbb{F}|$ probability. Alternatively, settling for a slightly worse soundness guarantee (where $e/|\mathbb{F}|$ is replaced by $e/(|\mathbb{F}| - 1)$), one could just let r' be a random *nonzero* field element, and get perfect zero-knowledge.

It turns out, however, that one could fix r' to 1 and still get the same soundness guarantee about U as in Lemma 4.2 since we can apply the same the decomposition argument. This "affine" variant of **Test-Interleaved** is described and analyzed in Appendix C.

The End