TL;DR

Building off of existing theory and practical, we propose a novel protocol for efficient and robust decentralized coded computing using binary field SNARK constructions and coding-theoretic techniques. It leverages a tower of binary field extensions to natively capture various data types and utilizes block-level polynomial commitments and PLONKish arithmetization for efficient verification of the computations.

Background

Coded computing has emerged as a promising approach for injecting redundancy into decentralized computations for robustness against faults and stragglers. However, existing solutions using zero-knowledge proofs (ZKPs) and fully homomorphic encryption (FHE) face challenges in terms of efficiency, flexibility and scalability.

Proposal

Our protocol synergistically combines state-of-the-art techniques from binary field SNARKs and coding theory:

- Use a tower of binary field extensions $\mathbb{F}_2 \subseteq \mathbb{F}_{2^2} \subseteq \mathbb{F}_{2^4} \subseteq \mathbb{F}_{2^8} \subseteq \cdots \subseteq \mathbb{F}_{2^{128}}$

to efficiently work with various data types

- Apply a block-level polynomial commitment scheme to commit coded boolean data with optimal rate and polylogarithmic proof size
- Adapt PLONKish techniques like product and permutation arguments over the binary fields to support expressive computations
- Introduce a shifting virtual polynomial for efficient rotations of coded data chunks
- Reconcile the different components via an interactive proof system with tower field arithmetic

Illustration

Consider a multilinear polynomial $f \in \mathbb{F}_2[X_1, \ldots, X_d]$

of degree $\leq d$

. We encode the coefficients block-wise into a vector $\vec{f} \in \mathbb{F}_{2^{\lceil \log{d} \rceil}}^{2^d/d}$

as follows:

1. Partition the coefficients into $2^d/d$

blocks $\{\vec{c}_i \in \mathbb{F}_2^d\}_{i=1}^{2^d/d}$

1. For each block i

, evaluate $g_i(X) = \sum_{j=0}^{d-1} c_{i,j} X^j$

at a fixed element $\alpha_i \in \mathbb{F}_{2^{\lceil \log{d} \rceil}}$

1. Define $\vec{f} = (g_1(\alpha_1), g_2(\alpha_2), \ldots, g_{2^d/d}(\alpha_{2^d/d}))$

To avoid embedding overhead when committing, we directly work with the block-encoded vector $\vec{f}$

which has length $O(2^d/d)$

over the extension field instead of the full coefficient vector over $\mathbb{F}_2$

of length 2^d

.

Advantages

Our protocol achieves several advantages over prior works:

- Efficient prover times of $\widetilde{O}(mN^2)$

field operations and verifier times of $\widetilde{O}(N^2/\rho)$

field operations for N

constraints and rate-ρ

encoding, where m

is the number of variables per constraint

- Proof sizes of $O(N^2 \log k / \rho)$

bits that compares favorably to FRI-based systems like STARKs and RedShift

- Flexibility to work with multiple binary extension fields and compute natively over $\mathbb{F}_2$

, enabling more compact constraint systems in some cases compared to R1CS

- Short structured reference strings that can be generated transparently without extra setup assumptions

Applications

This protocol can be applied to efficiently verify computations expressed as boolean or arithmetic circuits in a decentralized setting with redundancy against faults. Potential use cases include:

- Privacy-preserving outsourcing of computations to a network of untrusted workers
- Scalable and robust multi-party computation (MPC) with low online communication
- Transparent and succinct proof systems for general computations with purely algebraic security assumptions

Conclusion

We presented a high-performance cryptographic protocol for decentralized coded computing by combining binary field SNARKs and coding theory in a novel way. Our construction overcomes challenges in prior works and achieves asymptotic and concrete efficiency for a wide class of computations. This work expands the capabilities of zero-knowledge proof systems and enables exciting applications in privacy-enhancing technologies.