Accelerating Post-quantum Secure zkSNARKs by Optimizing Additive FFT

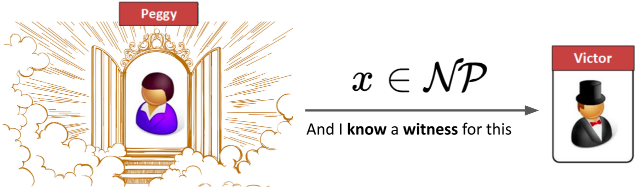
Mohammadtaghi Badakhshan, Susanta Samanta, and Guang Gong
{mbadakhshan, ssamanta, ggong}@uwaterloo.ca

Selected Areas in Cryptography (SAC) 2025
Toronto, Canada
(13 – 15 August 2025)

Table of Contents

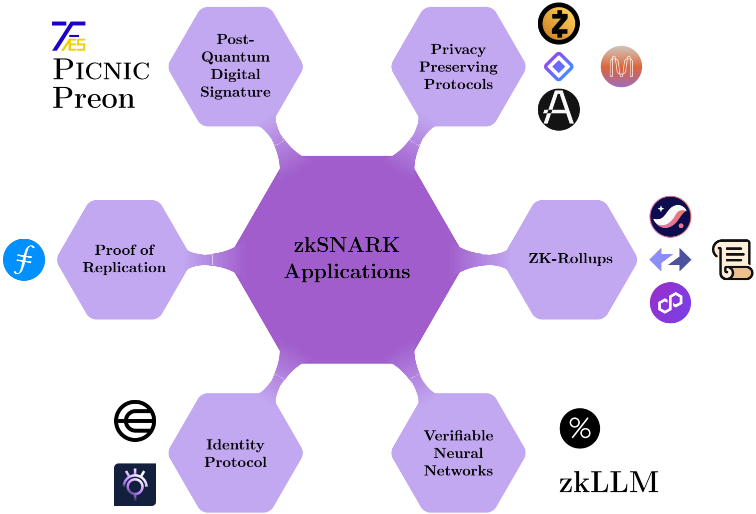
Table of Contents

zkSNARK: Zero-Knowledge Succinct Non-interactive Argument of Knowledge

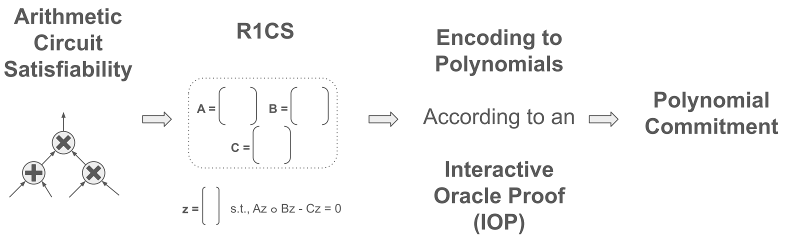


- Peggy (Prover) creates a **non-interactive (publicly verifiable)** argument about $x \in \mathcal{NP}$ and she **knows** a witness w for this.
- The argument convinces Victor (Verifier) that $x \in \mathcal{NP}$ but does not reveal any knowledge about w.

Applications of zkSNARKs

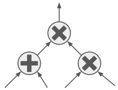


A Brief Background on zkSNARKs' Construction



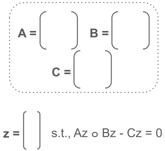
A Brief Background on zkSNARKs' Construction

Arithmetic Circuit Satisfiability



⇒

R1CS



⇒

Encoding to Polynomials

According to an

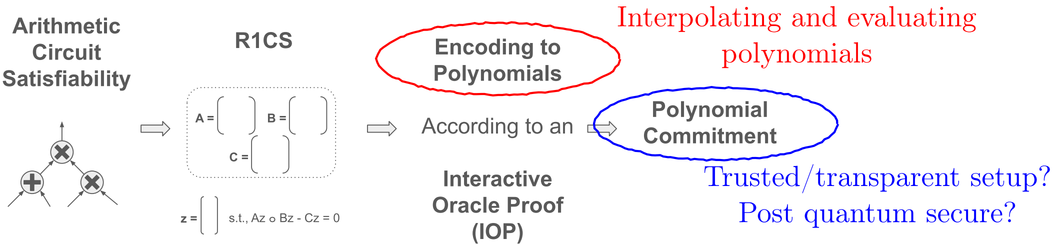
Interactive Oracle Proof (IOP)

⇒

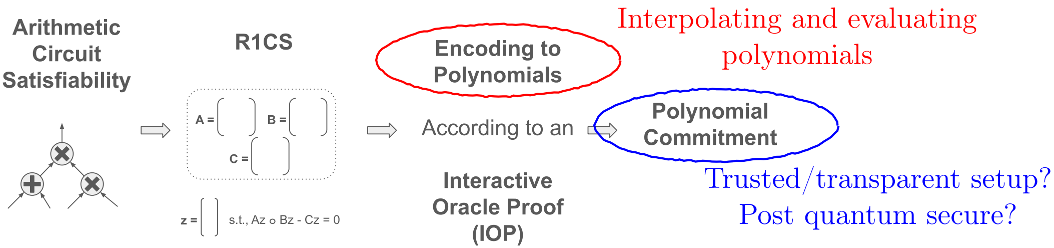
Polynomial Commitment

Interpolating and evaluating polynomials

A Brief Background on zkSNARKs' Construction



A Brief Background on zkSNARKs' Construction



Example:
Fast Reed-Solomon IOP (FRI) + Merkle Hash Tree (MHT) Commitment
(Ben-Sasson et al., 2018)

is a core component of many transparent setup post-quantum secure zkSNARKs.

Core Assumptions and Domains in Modern zkSNARKs

zkSNARK	Cryptographic Assumption	Algebraic Domain
Groth16	KoE	elliptic-curve group
PLONK	KoE	elliptic-curve group
HALO	ECDLP	elliptic-curve group
Aurora	CRH	any algebraic domain
STARK	CRH	any algebraic domain
Fractal	CRH	any algebraic domain
Ligero	CRH	any algebraic domain

Core Assumptions and Domains in Modern zkSNARKs

	zkSNARK	Cryptographic Assumption	Algebraic Domain
	Groth16	KoE	elliptic-curve group
	PLONK	KoE	elliptic-curve group
	HALO	ECDLP	elliptic-curve group
FRI-based	Aurora	CRH	any algebraic domain
FRI-based	STARK	CRH	any algebraic domain
FRI-based	Fractal	CRH	any algebraic domain
	Ligero	CRH	any algebraic domain

Challenges and Limitations in FRI-based zkSNARKs and Ligero

- High prover / verifier time complexity (addressed in this paper)

Polygon ZK-EVM	Generates a proof for 27 transactions on 128 vCPUs and 1024 GB RAM (Chaliasos et al., 2024).	**311 seconds**
Preon (Post-Quantum Signature)	Creates one signature (256-bit security) on a single-core AMD EPYC 73F3 @ 3.5 GHz (NIST PQC Round 1: Additional Signatures).	**417 seconds**

- Large argument (proof) size
- etc.

Main Contributions

1. **Acceleration of Aurora (Ben-Sasson et al., 2019) zkSNARK**: Optimized polynomial evaluation and interpolation via additive FFT improvements.

Main Contributions

1. **Acceleration of Aurora (Ben-Sasson et al., 2019) zkSNARK**: Optimized polynomial evaluation and interpolation via additive FFT improvements.
2. **Cantor additive FFT:**
   - Exact finite field operation counts.
   - Precomputation techniques and memory–runtime trade-offs, with extra savings for structured subspaces.
   - Applied in Aurora to measure acceleration gains.

Main Contributions

1. **Acceleration of Aurora (Ben-Sasson et al., 2019) zkSNARK**: Optimized polynomial evaluation and interpolation via additive FFT improvements.
2. **Cantor additive FFT:**
   - Exact finite field operation counts.
   - Precomputation techniques and memory–runtime trade-offs, with extra savings for structured subspaces.
   - Applied in Aurora to measure acceleration gains.
3. **Gao–Mateer additive FFT:**
   - Reduced finite field operation counts using Cantor special basis.
   - Precomputation techniques and memory–runtime trade-offs.

Main Contributions

1. **Acceleration of Aurora (Ben-Sasson et al., 2019) zkSNARK**: Optimized polynomial evaluation and interpolation via additive FFT improvements.
2. **Cantor additive FFT:**
   - Exact finite field operation counts.
   - Precomputation techniques and memory–runtime trade-offs, with extra savings for structured subspaces.
   - Applied in Aurora to measure acceleration gains.
3. **Gao–Mateer additive FFT:**
   - Reduced finite field operation counts using Cantor special basis.
   - Precomputation techniques and memory–runtime trade-offs.
4. **LCH additive FFT:**
   - Compared with other FFTs when LCH uses the Cantor special basis.
   - Measured acceleration gains when integrated into Aurora.

Table of Contents

Discrete Fourier Transform (DFT)

Let $W = \{\eta_0, \eta_1, \ldots, \eta_{n-1}\}$ be the evaluation set.

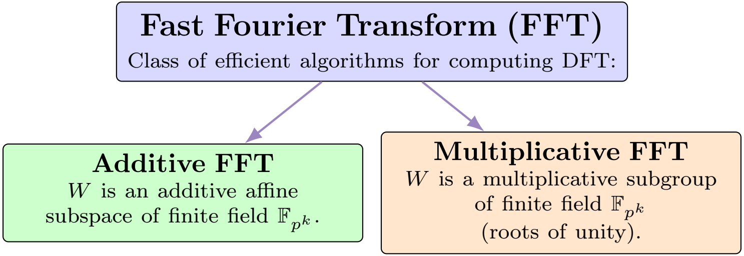
$$\textbf{DFT: } f(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1} \xrightarrow{\text{evaluation}} \begin{bmatrix} f(\eta_0) \\ f(\eta_1) \\ \vdots \\ f(\eta_{n-1}) \end{bmatrix}$$

$$\textbf{IDFT: } \begin{bmatrix} (\eta_0, f(\eta_0)) \\ (\eta_1, f(\eta_1)) \\ \vdots \\ (\eta_{n-1}, f(\eta_{n-1})) \end{bmatrix} \xrightarrow{\text{interpolation}} f(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$$

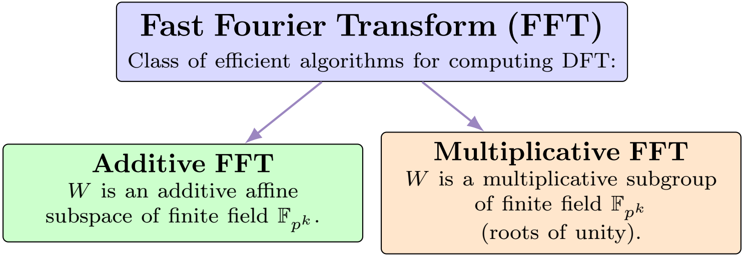
The prover algorithm in **Aurora** performs $\boxed{7\lambda_i + \lambda_i' + 10}$ polynomial evaluations and interpolations, where λ_i and λ_i' are repetition parameters [1].

[1] The number of times of calling lincheck and FRI-LDT subprotocols.

Fast Fourier Transform (FFT)

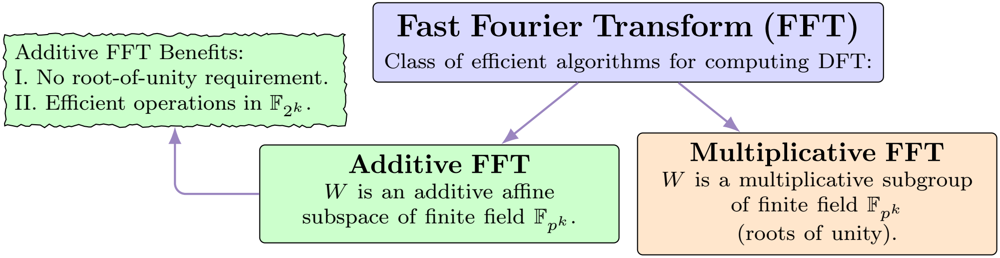


Fast Fourier Transform (FFT)



- The structure of W is dictated by the *polynomial commitment scheme* in zkSNARKs.
- **FRI+MHT** supports both additive affine subspaces (of binary fields) and multiplicative subgroups (of prime fields).

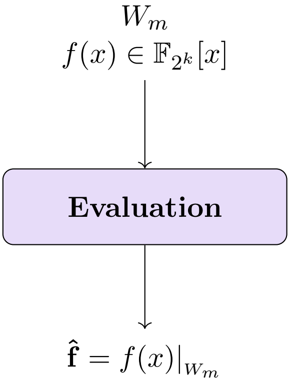
Fast Fourier Transform (FFT)



- The structure of W is dictated by the *polynomial commitment scheme* in zkSNARKs.
- **FRI+MHT** supports both additive affine subspaces (of binary fields) and multiplicative subgroups (of prime fields).

Additive FFT Algorithms

- Cantor (1989)
- von zur Gathen and Gerhard (1996)



- Gao and Mateer (2010)
- Lin, Chung, and Han (2014)

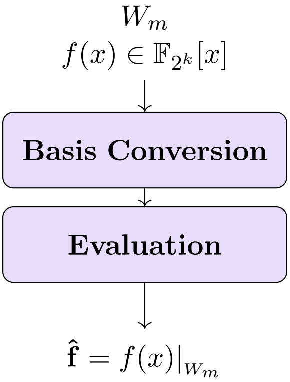


Table of Contents

Cantor Additive FFT Algorithm (Cantor, 1989)

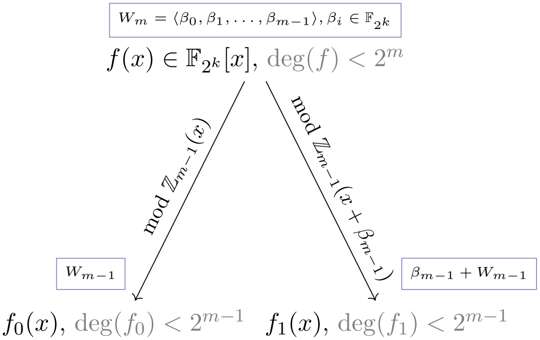
- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial $\rightarrow$ $\mathbb{Z}_{W_i}(x + \theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$

$$W_m = \langle \beta_0, \beta_1, \ldots, \beta_{m-1} \rangle, \beta_i \in \mathbb{F}_{2^k}$$

$$f(x) \in \mathbb{F}_{2^k}[x], \deg(f) < 2^m$$

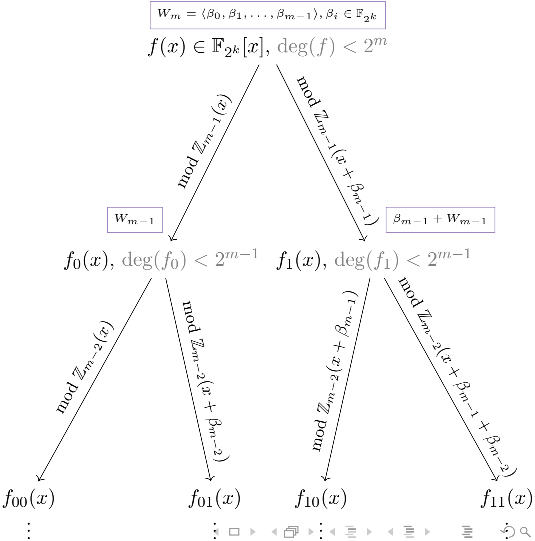
Cantor Additive FFT Algorithm (Cantor, 1989)

- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial $\rightarrow$ $\mathbb{Z}_{W_i}(x+\theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$



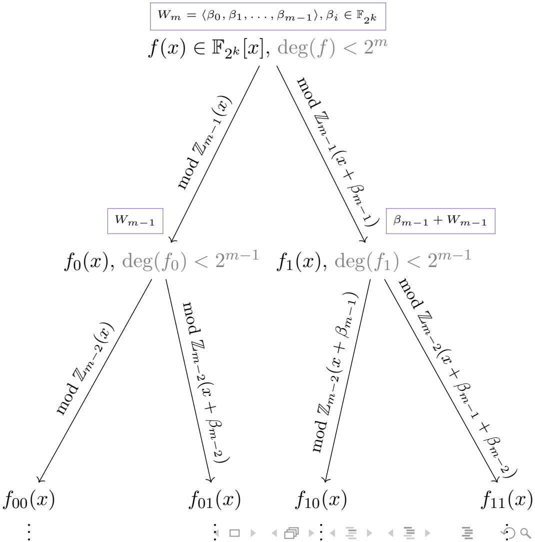
Cantor Additive FFT Algorithm (Cantor, 1989)

- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial $\rightarrow$ $\mathbb{Z}_{W_i}(x+\theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$



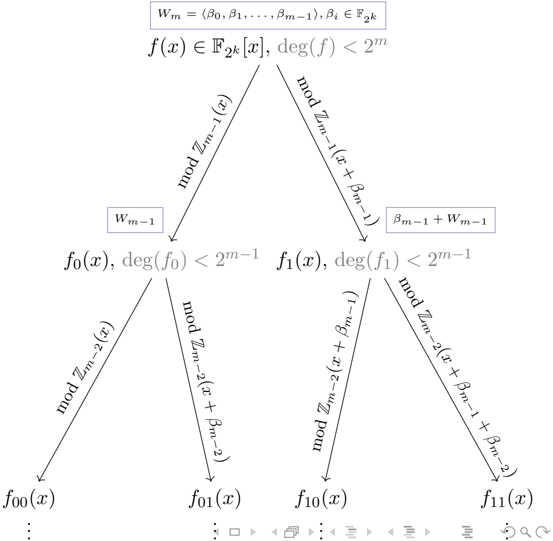
Cantor Additive FFT Algorithm (Cantor, 1989)

- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial → $\mathbb{Z}_{W_i}(x+\theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$
- For $f(x) \in \mathbb{F}_{2^k}[x]$, with $k = t \cdot 2^\ell$ $(2^\ell \geq m)$ $\{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ denotes Cantor special basis if $\boxed{\beta_{i-1} = \beta_i^2 + \beta_i \text{ and } \beta_0 = 1}$.



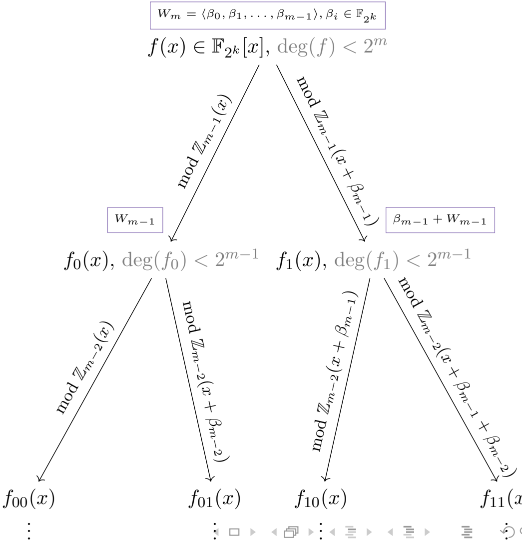
Cantor Additive FFT Algorithm (Cantor, 1989)

- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial $\rightarrow$ $\mathbb{Z}_{W_i}(x+\theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$
- For $f(x) \in \mathbb{F}_{2^k}[x]$, with $k = t \cdot 2^\ell$ $(2^\ell \geq m)$ $\{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ denotes Cantor special basis if $\boxed{\beta_{i-1} = \beta_i^2 + \beta_i \text{ and } \beta_0 = 1}$.
- If W_i is spanned by the Cantor special basis:



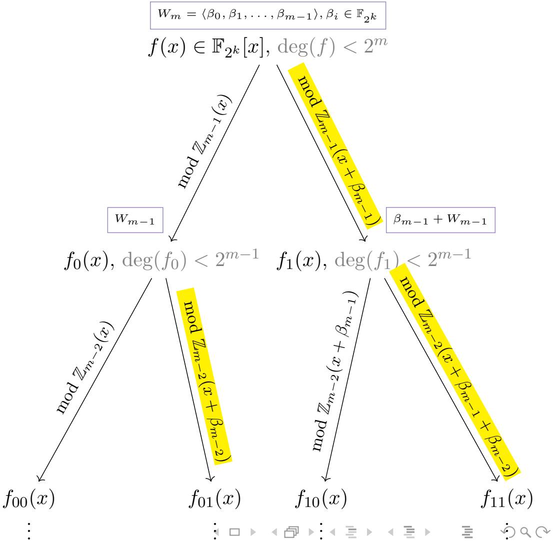
Cantor Additive FFT Algorithm (Cantor, 1989)

- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial $\rightarrow$ $\mathbb{Z}_{W_i}(x+\theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$
- For $f(x) \in \mathbb{F}_{2^k}[x]$, with $k = t \cdot 2^\ell$ $(2^\ell \geq m)$ $\{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ denotes Cantor special basis if $\boxed{\beta_{i-1} = \beta_i^2 + \beta_i \text{ and } \beta_0 = 1}$.
- If W_i is spanned by the Cantor special basis:
    - $\mathbb{Z}_{W_i}(\beta_{i+\ell}) = \beta_\ell \rightarrow \mathbb{Z}_{W_i}(\beta_i) = 1$
    - $\mathbb{Z}_{W_i}(x) \in \mathbb{F}_2[x]$

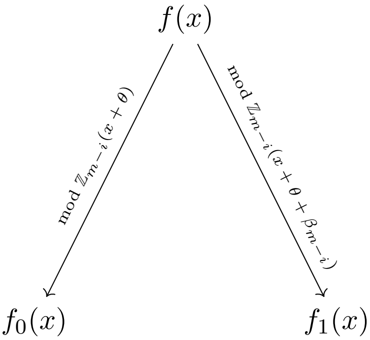


Cantor Additive FFT Algorithm (Cantor, 1989)

- $\mathbb{Z}_{W_i}(x)$ denotes a vanishing polynomial over $W_i := \langle \beta_0, \beta_1, \ldots, \beta_i \rangle$.
- $\mathbb{Z}_{W_i}(x)$ is a linearized polynomial $\rightarrow$ $\mathbb{Z}_{W_i}(x+\theta) = \mathbb{Z}_{W_i}(x) + \mathbb{Z}_{W_i}(\theta)$
- For $f(x) \in \mathbb{F}_{2^k}[x]$, with $k = t \cdot 2^\ell$ $(2^\ell \geq m)$ $\{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ denotes Cantor special basis if $\boxed{\beta_{i-1} = \beta_i^2 + \beta_i \text{ and } \beta_0 = 1}$.
- If W_i is spanned by the Cantor special basis:
    - $\mathbb{Z}_{W_i}(\beta_{i+\ell}) = \beta_\ell \rightarrow \mathbb{Z}_{W_i}(\beta_i) = 1$
    - $\mathbb{Z}_{W_i}(x) \in \mathbb{F}_2[x]$

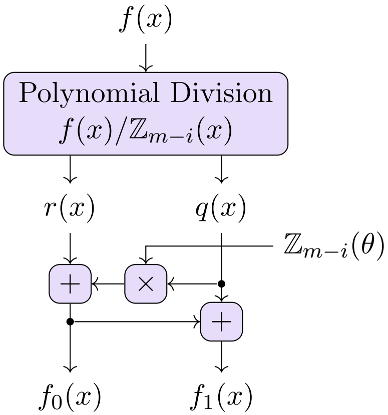


Cantor Additive FFT: Polynomial Division Algorithm



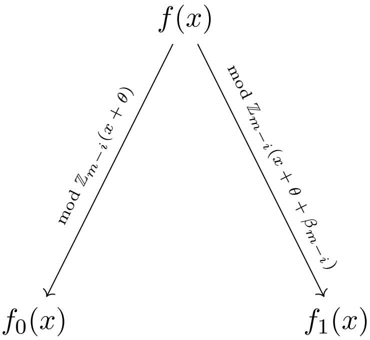
$$f(x) \in \mathbb{F}_{2^k}[x],$$
$$\deg(f) < 2^{m-i+1}$$

$$\equiv$$



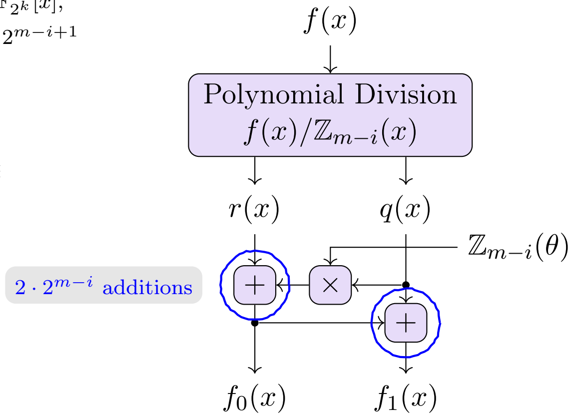
$$f_0(x) = r(x) + \mathbb{Z}_{W_{m-i}}(\theta)\, q(x),$$
$$f_1(x) = f_0(x) + q(x)$$

Cantor Additive FFT: Polynomial Division Algorithm



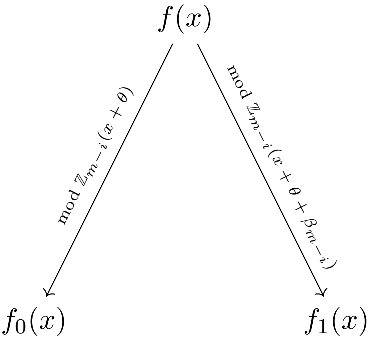
$$f(x) \in \mathbb{F}_{2^k}[x],$$
$$\deg(f) < 2^{m-i+1}$$

$$\equiv$$



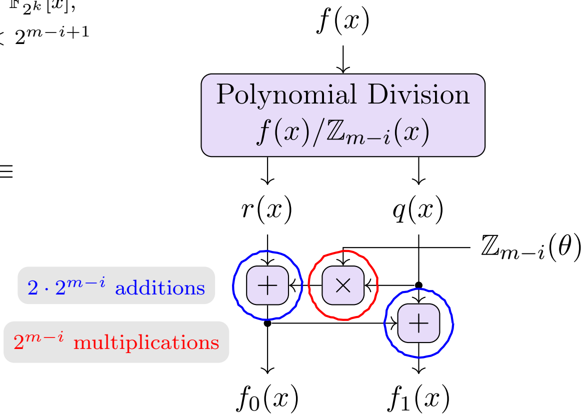
$$f_0(x) = r(x) + \mathbb{Z}_{W_{m-i}}(\theta)\, q(x),$$
$$f_1(x) = f_0(x) + q(x)$$

Cantor Additive FFT: Polynomial Division Algorithm



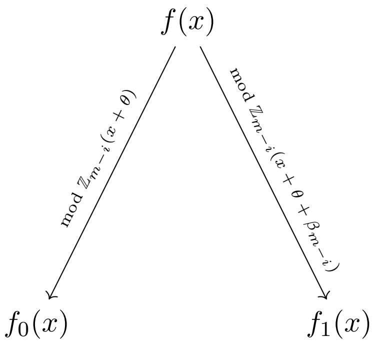
$f(x) \in \mathbb{F}_{2^k}[x]$,
$\deg(f) < 2^{m-i+1}$

$\equiv$



$$f_0(x) = r(x) + \mathbb{Z}_{W_{m-i}}(\theta)\, q(x),$$
$$f_1(x) = f_0(x) + q(x)$$

Cantor Additive FFT: Polynomial Division Algorithm



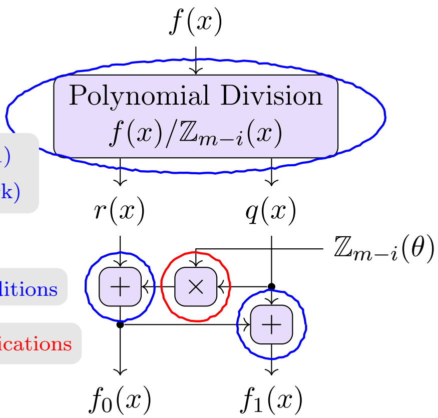
$$f_0(x) = r(x) + \mathbb{Z}_{W_{m-i}}(\theta)\, q(x),$$
$$f_1(x) = f_0(x) + q(x)$$

$f(x) \in \mathbb{F}_{2^k}[x]$,
$\deg(f) < 2^{m-i+1}$

$2^{m-i}(2^{\mathrm{wt}(m-i)} - 1)$ additions (This work)

$2 \cdot 2^{m-i}$ additions

2^{m-i} multiplications

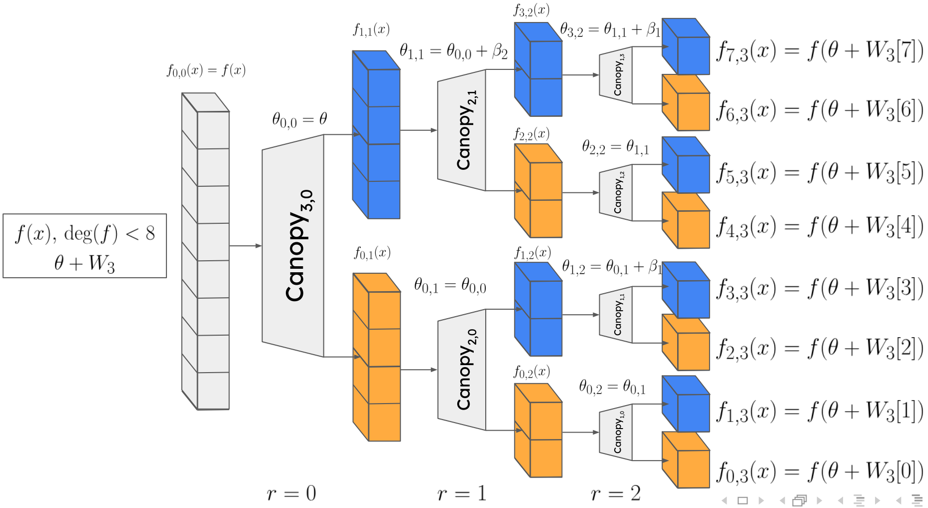


Cantor Additive FFT: Finite Field Operations

The exact number of finite field additions and multiplications in the Cantor additive FFT

- Additions: $\frac{1}{2} n \log_2 n + \frac{1}{2} n \sum_{r=0}^{\log_2(n)-1} 2^{\mathrm{wt}(r)}$
- Multiplications: $\frac{1}{2} n \log_2 n$

Cantor Additive FFT: Final Structure



Cantor Additive FFT: Precomputation

- $\mathbb{Z}_{W_{i,r}}(\theta_{i,r})$ can be precomputted for any predetermined $\theta + W_m$
- This requires $2^m - 1$ elements in $\mathbb{F}_{2^k}$.

Special Case $\theta \in \{\beta_0, \beta_1, \ldots, \beta_{b-1}\}$:

- It is only required to compute $\langle \beta_0, \beta_1, \ldots, \beta_{b-1} \rangle$ for any FFT of length $2^m < 2^b$.
- Multiple lookup tables:

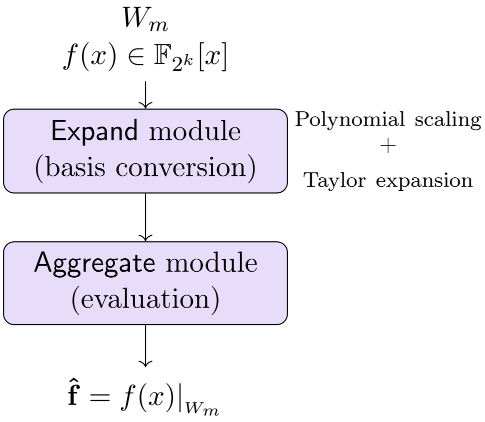
$$\langle \beta_0, \ldots, \beta_{\frac{b}{\ell}-1} \rangle, \langle \beta_{\frac{b}{\ell}}, \ldots, \beta_{2\frac{b}{\ell}-1} \rangle, \ldots, \langle \beta_{(\ell-1)\frac{b}{\ell}}, \ldots, \beta_{\ell\frac{b}{\ell}-1} \rangle$$

which requires $\ell \cdot 2^{\frac{b}{\ell}}$ elements in $\mathbb{F}_{2^k}$.

Table of Contents

Gao–Mateer Additive FFT (Gao and Mateer, 2010)

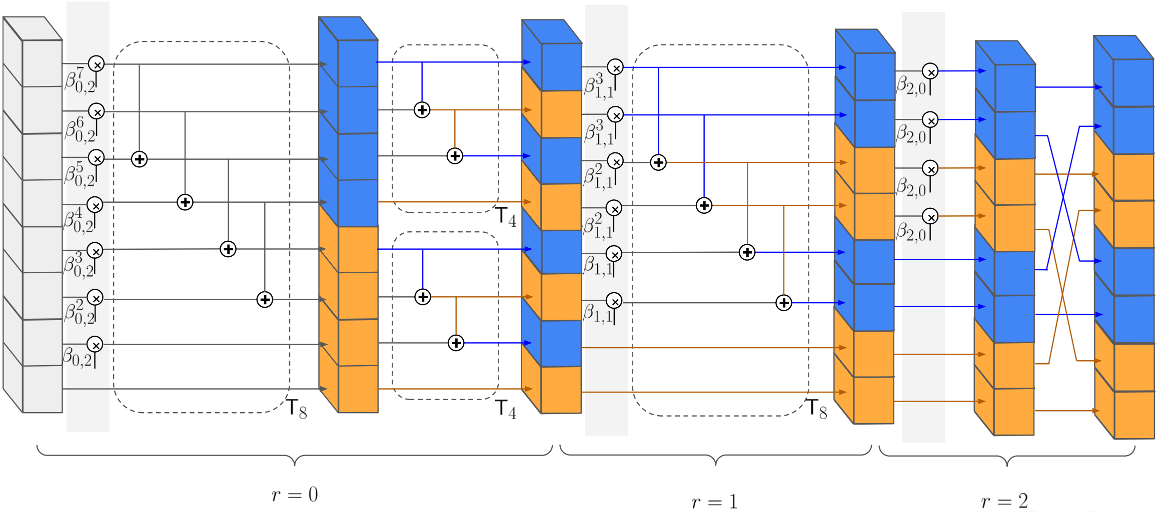
- Applicable to $f(x) \in \mathbb{F}_{2^k}[x]$ for any arbitrary k.



Optimizations:

- Precomputation:
  - Level 1: Basis elements for each round of the Expand and Aggregate module.
  - Level 2: All multiplication factors in the both modules.
- Using the Cantor Special Basis:
  - Saves $n \log_2 n - n + 1$ multiplications in the Expand module.

Gao–Mateer Additve FFT Optimization: Expand Module



Gao–Mateer Additve FFT: Cost of Each Component

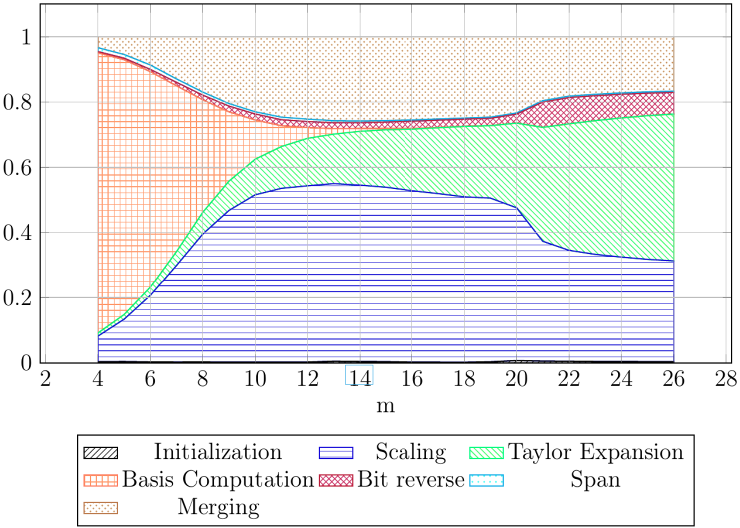
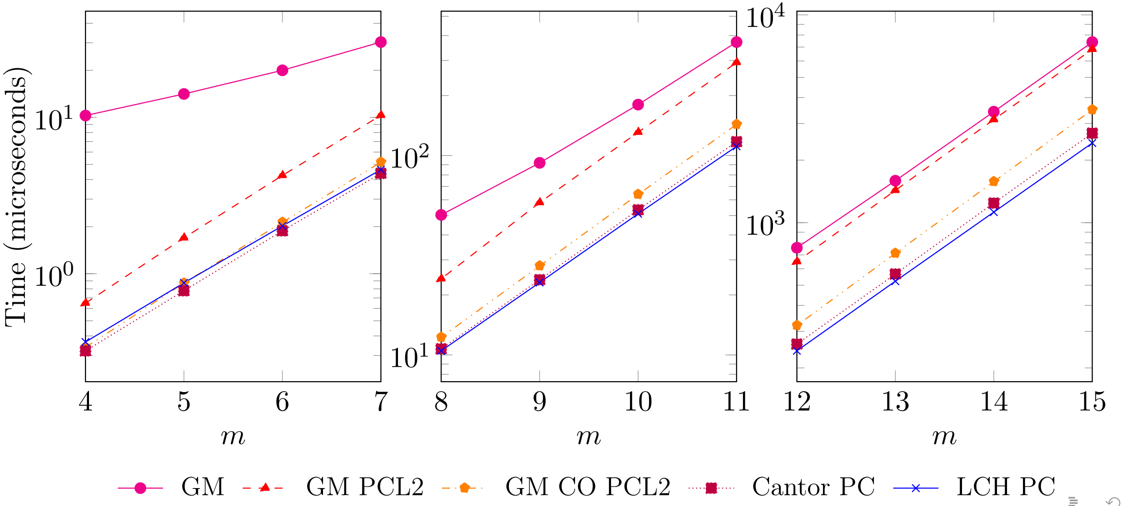


Table of Contents

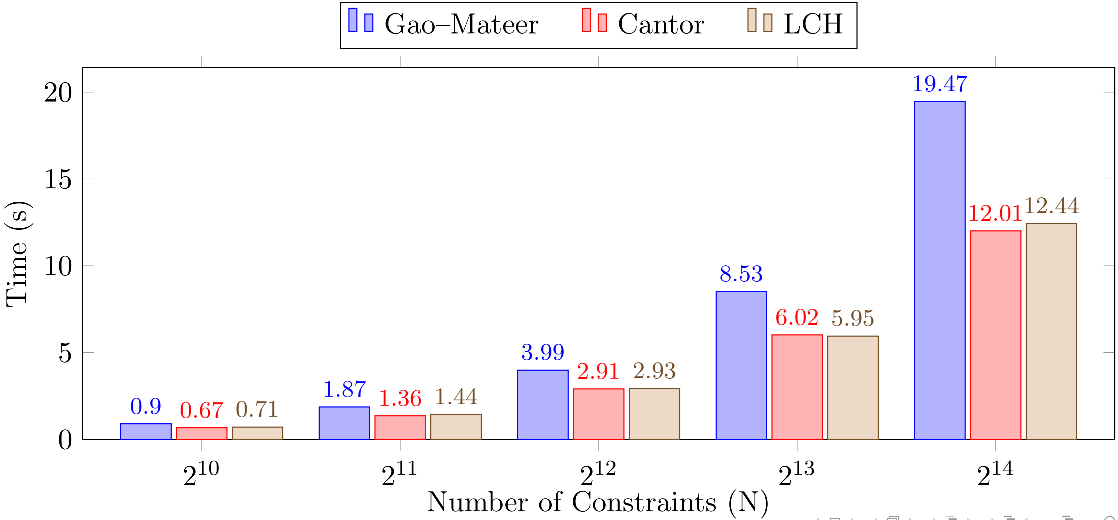
Finitie Field Operation Comparison

FFT	Basis	Basis Conversion	Evaluation
GM	General	#A: $\frac{1}{4}n(\log_2 n)^2 - \frac{1}{4}n\log_2 n$ #M: $n\log_2 n - n + 1$	#A: $n\log_2 n$ #M: $\frac{1}{2}n\log_2 n$
	Cantor	#A: $\frac{1}{4}n(\log_2 n)^2 - \frac{1}{4}n\log_2 n$ #M: 0	#A: $n\log_2 n$ #M: $\frac{1}{2}n\log_2 n$
LCH	General (Lin et al., 2016)	#A: $O(n(\log_2 n)^2)$ #M: $O(n\log_2 n)$	#A: $n\log_2 n$ #M: $\frac{1}{2}n\log_2 n$
	Cantor (Lin et al., 2016)	#A: $O(n\log_2 n\log_2\log_2 n)$ #M: 0	#A: $n\log_2 n$ #M: $\frac{1}{2}n\log_2 n$
Cantor	Cantor	N/A	#A: $\frac{1}{2}n\log_2 n + \frac{1}{2}n\sum_{r=0}^{\log_2(n)-1} 2^{\mathrm{wt}(r)}$ #M: $\frac{1}{2}n\log_2 n$

Additive FFT Benchmark (input length: $n = 2^m$)



Aurora Prover Timing Comparison Across FFT Implementations



Aurora Prover Timing Comparison Across FFT Implementations

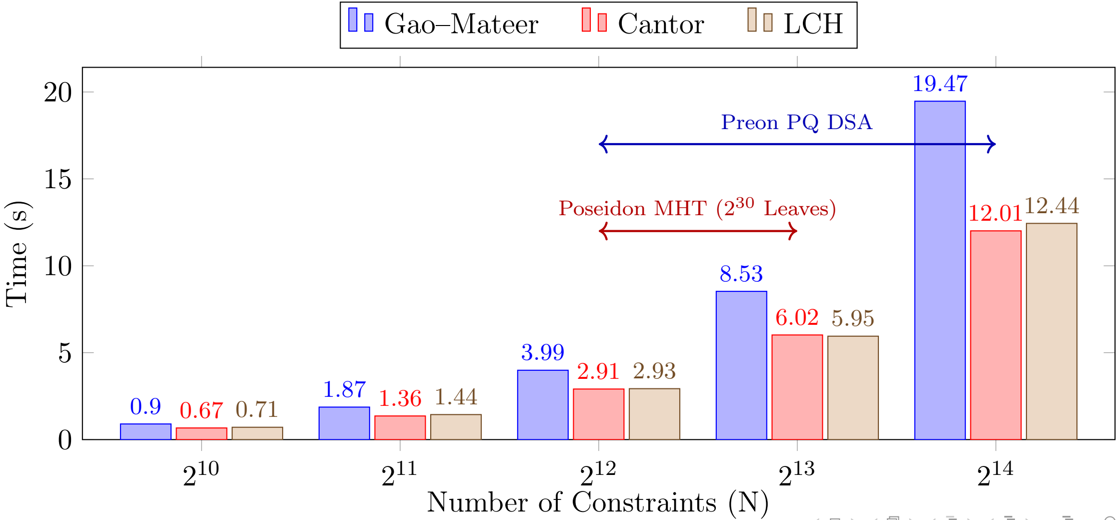


Table of Contents

Conclusions

- Leveraging the Cantor special basis enables the integration of Cantor and LCH additive FFTs into post-quantum secure zk-SNARKs, e,g., Aurora.

Conclusions

- Leveraging the Cantor special basis enables the integration of Cantor and LCH additive FFTs into post-quantum secure zk-SNARKs, e,g., Aurora.
- Replacing the Gao–Mateer FFT with Cantor and LCH additive FFTs significantly reduces computation time.

Conclusions

- Leveraging the Cantor special basis enables the integration of Cantor and LCH additive FFTs into post-quantum secure zk-SNARKs, e,g., Aurora.
- Replacing the Gao–Mateer FFT with Cantor and LCH additive FFTs significantly reduces computation time.
- The results are justified by a detailed cost analysis (finite field additions and multiplications) of additive FFTs and the complexity evaluation of FFT calls in Aurora.

Conclusions

- Leveraging the Cantor special basis enables the integration of Cantor and LCH additive FFTs into post-quantum secure zk-SNARKs, e,g., Aurora.
- Replacing the Gao–Mateer FFT with Cantor and LCH additive FFTs significantly reduces computation time.
- The results are justified by a detailed cost analysis (finite field additions and multiplications) of additive FFTs and the complexity evaluation of FFT calls in Aurora.
- We proposed precomputation techniques that reduce overhead for both Cantor and Gao–Mateer FFTs when the affine subspace basis is fixed.

Future Works

- Extend FFT optimizations such as applying tower field constructions to accelerate field multiplications.

Future Works

- Extend FFT optimizations such as applying tower field constructions to accelerate field multiplications.
- Improve the Cantor additive FFT throughput through parallelization as it has pure divide-and-conquer (radix-4 or over processing-in-memory).

Future Works

- Extend FFT optimizations such as applying tower field constructions to accelerate field multiplications.
- Improve the Cantor additive FFT throughput through parallelization as it has pure divide-and-conquer (radix-4 or over processing-in-memory).
- Extend the optimizations to other post-quantum secure zkSNARKs over binary extension fields, such as STARK, Fractal, Ligero, etc.

Future Works

- Extend FFT optimizations such as applying tower field constructions to accelerate field multiplications.
- Improve the Cantor additive FFT throughput through parallelization as it has pure divide-and-conquer (radix-4 or over processing-in-memory).
- Extend the optimizations to other post-quantum secure zkSNARKs over binary extension fields, such as STARK, Fractal, Ligero, etc.
- Side-channel analysis of additive FFT implementations.

References I

Ben-Sasson, E., I. Bentov, Y. Horesh, and M. Riabzev (2018). Fast Reed-Solomon interactive oracle proofs of proximity. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella (Eds.), *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, Volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Dagstuhl, Germany, pp. 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Ben-Sasson, E., A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward (2019). Aurora: Transparent succinct arguments for R1CS. In Y. Ishai and V. Rijmen (Eds.), *Advances in Cryptology – EUROCRYPT 2019*, Cham, pp. 103–128. Springer International Publishing.

Cantor, D. G. (1989). On arithmetical algorithms over finite fields. *Journal of Combinatorial Theory, Series A 50*(2), 285–300.

Chaliasos, S., I. Reif, A. Torralba-Agell, J. Ernstberger, A. Kattis, and B. Livshits (2024). Analyzing and benchmarking zk-rollups. In R. Böhme and L. Kiffer (Eds.), *6th Conference on Advances in Financial Technologies (AFT 2024)*, Volume 316 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Dagstuhl, Germany, pp. 6:1–6:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Gao, S. and T. Mateer (2010). Additive fast Fourier transforms over finite fields. *IEEE Transactions on Information Theory 56*(12), 6265–6272.

References II

Lin, S.-J., T. Y. Al-Naffouri, Y. S. Han, and W.-H. Chung (2016). Novel Polynomial Basis With Fast Fourier Transform and Its Application to Reed–Solomon Erasure codes. *IEEE Transactions on Information Theory 62*(11), 6284–6299.

Lin, S.-J., W.-H. Chung, and Y. S. Han (2014). Novel polynomial basis and its application to reed-solomon erasure codes. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 316–325.

von zur Gathen, J. and J. Gerhard (1996). Arithmetic and factorization of polynomial over $\mathbb{F}_2$ (extended abstract). In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, ISSAC '96, New York, NY, USA, pp. 1–9. Association for Computing Machinery.

Thank You!

Mohammadtaghi Badakhshan, Susanta Samanta, Guang Gong

Communications Security (ComSec) Lab
University of Waterloo

Any questions?