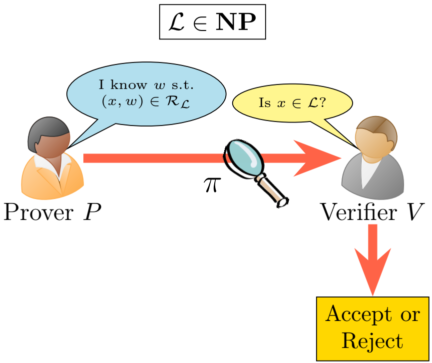
Field-Agnostic SNARKs from Expand-Accumulate Codes

CRYPTO 2024

Alexander R. Block[1] Zhiyong Fang[2] Jonathan Katz[3]
Justin Thaler[4] Hendrik Waldner[5] Yupeng Zhang[6]

[1]Georgetown University and University of Maryland
[2]Texas A&M University
[3]Google and University of Maryland
[4]a16z crypto research and Georgetown University
[5]University of Maryland
[6]University of Illinois Urbana-Champaign

Succinct **N**on-interactive **AR**guments of **K**nowledge



Completeness: $\forall (x, w) \in \mathcal{R}_\mathcal{L}$:

$$\Pr[V(x, \pi) = 1 \mid \pi \leftarrow P(x, w)] = 1$$

ε**-Soundness:** $\forall x \notin \mathcal{L}$, $\forall$ PPT P^*:

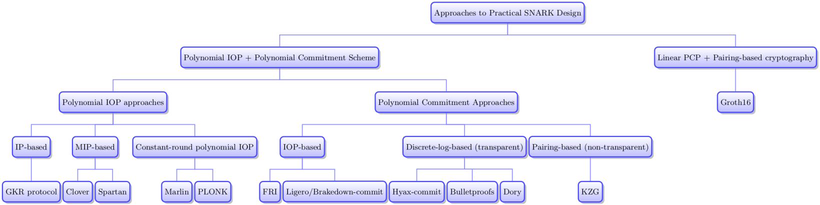
$$\Pr[V(x, \pi^*) = 1 \mid \pi^* \xleftarrow{\$} P^*(x)] \leqslant \varepsilon(x, \lambda)$$

ε**-Knowledge Soundness:** $\exists$ PPT extractor $\mathcal{E}$ such that $\forall x$ and $\forall$ PPT P^*:

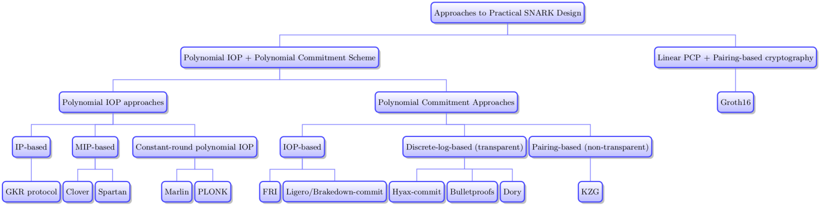
$$\Pr[(x, \mathcal{E}^{P^*}(x)) \in \mathcal{R}_\mathcal{L}] + \varepsilon(x, \lambda) \geqslant \Pr[V(x, \pi^*) = 1 \mid \pi^* \xleftarrow{\$} P^*(x)]$$

Succinctness: $|\pi| = o_\lambda(|w|)$; ideally $O_\lambda(\text{polylog}(|w|))$

Building Concretely Efficient SNARKs [Tha22]



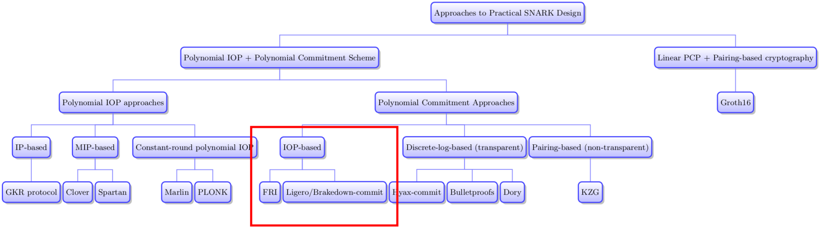
Building Concretely Efficient SNARKs [Tha22]



This Work

Build SNARKs from Error-correcting Codes

Building Concretely Efficient SNARKs [Tha22]



This Work

Build SNARKs from Error-correcting Codes

Code(-based) SNARKs

- Why Code-based SNARKs?

Code(-based) SNARKs

- Why Code-based SNARKs?
    - Transparent setup

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security
  - Concretely efficient *if the underlying code is concretely efficient*

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security
  - Concretely efficient *if the underlying code is concretely efficient*

More on this later

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security
  - Concretely efficient *if the underlying code is* <u>*concretely efficient*</u>

More on this later

- SNARKs from Codes are built upon the PIOP + PCS paradigm

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security
  - Concretely efficient *if the underlying code is concretely efficient*

More on this later

- SNARKs from Codes are built upon the PIOP + PCS paradigm

$\boxed{\text{Code SNARK}}$ =

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security
  - Concretely efficient *if the underlying code is <u>concretely efficient</u>*

More on this later

- SNARKs from Codes are built upon the PIOP + PCS paradigm

$$\boxed{\text{Code SNARK}} = \boxed{\text{PIOP}} + \boxed{\text{Code-based PCS}}$$

Code(-based) SNARKs

- Why Code-based SNARKs?
  - Transparent setup
  - Plausible PQ security
  - Concretely efficient *if the underlying code is <u>concretely efficient</u>*

More on this later

- SNARKs from Codes are built upon the PIOP + PCS paradigm

$$\boxed{\text{Code SNARK}} = \boxed{\text{PIOP}} + \boxed{\text{Code-based PCS}}$$

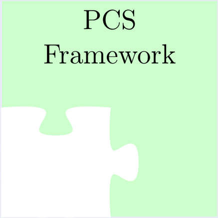
Goal: design Code-based *Polynomial Commitment Scheme*

PCS from any Linear Code

Ligero/Brakedown-based PCS

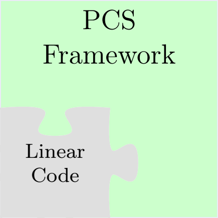
PCS from any Linear Code

Ligero/Brakedown-based PCS



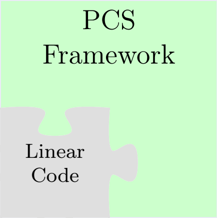
PCS from any Linear Code

Ligero/Brakedown-based PCS



PCS from any Linear Code

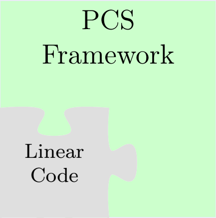
Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

PCS from any Linear Code

Ligero/Brakedown-based PCS

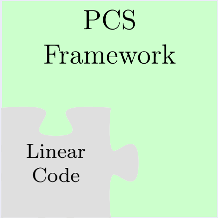


PCS from any linear code (derived from [BCG20])

Pros

PCS from any Linear Code

Ligero/Brakedown-based PCS



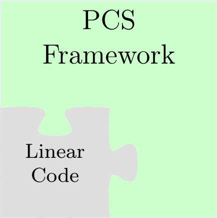
PCS from any linear code (derived from [BCG20])

Pros

Plug-and-play with any linear code

PCS from any Linear Code

Ligero/Brakedown-based PCS



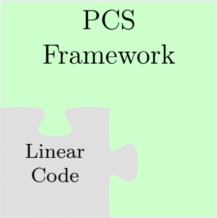
PCS from any linear code (derived from [BCG20])

Pros

- Plug-and-play with any linear code
- SNARK properties directly linked to code properties

PCS from any Linear Code

Ligero/Brakedown-based PCS

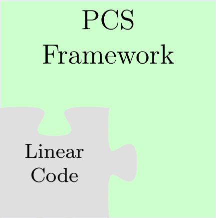


PCS from any linear code (derived from [BCG20])

Pros

- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

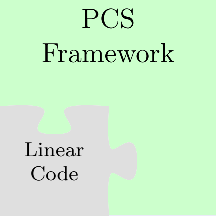
Pros

- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

Cons

PCS from any Linear Code

Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

Pros

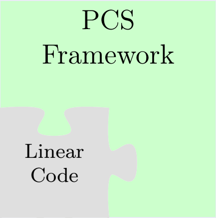
- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

Cons

- sqrt proof sizes

PCS from any Linear Code

Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

Pros

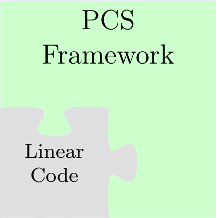
- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

Cons

- sqrt proof sizes
- Prior work:
  - concretely large proofs [Brakedown]
  - quasi-linear time, not field-agnostic [Ligero]

PCS from any Linear Code

Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

Pros

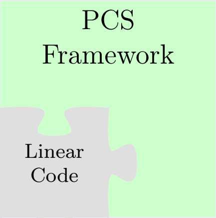
- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

Cons

- sqrt proof sizes
- Prior work:
  - concretely large proofs [Brakedown]
  - quasi-linear time, not field-agnostic [Ligero]
- Design + analysis of new codes is difficult

PCS from any Linear Code

Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

Pros

- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

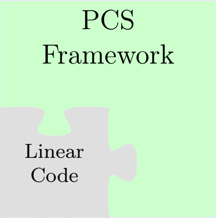
Cons

- sqrt proof sizes
- Prior work:
  - concretely large proofs [Brakedown]
  - quasi-linear time, not field-agnostic [Ligero]
- Design + analysis of new codes is difficult

Why field-agnostic?

PCS from any Linear Code

Ligero/Brakedown-based PCS



PCS from any linear code (derived from [BCG20])

Pros

- Plug-and-play with any linear code
- SNARK properties directly linked to code properties
- prover time = encoding time
- field-agnostic
- good code distance = small proofs

Cons

- sqrt proof sizes
- Prior work:
  - concretely large proofs [Brakedown]
  - quasi-linear time, not field-agnostic [Ligero]
- Design + analysis of new codes is difficult

Why field-agnostic?

Prover can experience $\approx 25\times$ slow down if the SNARK doesn't support field of the underlying computation!

Our Results: Bird's Eye View

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes
via the Brakedown PCS Framework

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

Provide an alternate distance analysis + proof via classic coding techniques

Better concrete parameters than [BCG+22]

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

Provide an alternate distance analysis + proof via classic coding techniques

Better concrete parameters than [BCG+22]

2

Generalize EA codes to arbitrary finite fields

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

Provide an alternate distance analysis + proof via classic coding techniques

Better concrete parameters than [BCG+22]

2

Generalize EA codes to arbitrary finite fields

Extend binary analysis to FFs

Answers open problem of [BCG+22]

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

- Provide an alternate distance analysis + proof via classic coding techniques
- Better concrete parameters than [BCG+22]

2

Generalize EA codes to arbitrary finite fields

- Extend binary analysis to FFs
- Answers open problem of [BCG+22]
- Quasi-linear encoding, concretely large min distance

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

- Provide an alternate distance analysis + proof via classic coding techniques
- Better concrete parameters than [BCG+22]

2

Generalize EA codes to arbitrary finite fields

- Extend binary analysis to FFs
- Answers open problem of [BCG+22]
- Quasi-linear encoding, concretly large min distance

3

Concretely efficient SNARKs from EA codes

Our Results: Bird's Eye View

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1

Improved distance analysis of binary EA codes

- Provide an alternate distance analysis + proof via classic coding techniques
- Better concrete parameters than [BCG+22]

2

Generalize EA codes to arbitrary finite fields

- Extend binary analysis to FFs
- Answers open problem of [BCG+22]
- Quasi-linear encoding, concretly large min distance

3

Concretely efficient SNARKs from EA codes

- Field-agnostic
- Proof size + verifier time comparable with [Brakedown], only $\approx 1.2\times$ prover overhead

Our Results: Comparison

Table 1: Performance of field-agnostic SNARKs based on linear codes for a statement modeled as an arithmetic circuit of size M and depth d.

	Prover Time	Proof Size	Verifier Time
[Brakedown]	$O(M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
[BaseFold]	$O(M \log M)$	$O(\log^2 M)$	$O(\log^2 M)$
This Work	$O(M \log M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
Proof of ECDSA verification			
[Brakedown]	0.17s	2.2MB	0.062s
[BaseFold]	0.273s	5.5MB	0.021s
Ours (provable)	0.23s	1.1MB	0.068s
Ours (conjectured)	0.23s	0.78MB	0.067s

Our Results: Comparison

Table 1: Performance of field-agnostic SNARKs based on linear codes for a statement modeled as an arithmetic circuit of size M and depth d.

	Prover Time	Proof Size	Verifier Time
[Brakedown]	$O(M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
[BaseFold]	$O(M \log M)$	$O(\log^2 M)$	$O(\log^2 M)$
This Work	$O(M \log M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
Proof of ECDSA verification			
[Brakedown]	0.17s	2.2MB	0.062s
[BaseFold]	0.273s	5.5MB	0.021s
Ours (provable)	0.23s	1.1MB	0.068s
Ours (conjectured)	0.23s	0.78MB	0.067s

Faster than Basefold

Our Results: Comparison

Table 1: Performance of field-agnostic SNARKs based on linear codes for a statement modeled as an arithmetic circuit of size M and depth d.

	Prover Time	Proof Size	Verifier Time
[Brakedown]	$O(M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
[BaseFold]	$O(M \log M)$	$O(\log^2 M)$	$O(\log^2 M)$
This Work	$O(M \log M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
Proof of ECDSA verification			
[Brakedown]	0.17s	2.2MB	0.062s
[BaseFold]	0.273s	5.5MB	0.021s
Ours (provable)	0.23s	1.1MB	0.068s
Ours (conjectured)	0.23s	0.78MB	0.067s

Faster than Basefold

Concretely smaller proofs

Our Results: Comparison

Table 1: Performance of field-agnostic SNARKs based on linear codes for a statement modeled as an arithmetic circuit of size M and depth d.

	Prover Time	Proof Size	Verifier Time
[Brakedown]	$O(M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
[BaseFold]	$O(M \log M)$	$O(\log^2 M)$	$O(\log^2 M)$
This Work	$O(M \log M)$	$O(\sqrt{M})$	$O(\sqrt{M})$
Proof of ECDSA verification			
[Brakedown]	0.17s	2.2MB	0.062s
[BaseFold]	0.273s	5.5MB	0.021s
Ours (provable)	0.23s	1.1MB	0.068s
Ours (conjectured)	0.23s	0.78MB	0.067s

Faster than Basefold

Concretely smaller proofs

Comparable to Brakedown

Remainder of the Talk

- Error-correcting codes overview
- EA Codes overview
- IOWE technique for distance analysis
- EA Code over any finite field analysis
- Experimental results

Definition 1 (Linear Codes)

Linear Error-Correcting Codes

Definition 1 (Linear Codes)

Let $\mathbb{F}$ be a finite field. A $[N, n, d]$ linear error correcting code C for $n \leqslant N$ is a n-dimensional subspace of $\mathbb{F}^N$ such that the minimum distance of C, denoted as $\Delta(C)$, is d, where $\Delta(C) := \min_{y \in C \setminus \{0^N\}} \{\mathsf{wt}(y)\}$.

Linear Error-Correcting Codes

Definition 1 (Linear Codes)

Let $\mathbb{F}$ be a finite field. A $[N, n, d]$ linear error correcting code C for $n \leqslant N$ is a n-dimensional subspace of $\mathbb{F}^N$ such that the minimum distance of C, denoted as $\Delta(C)$, is d, where $\Delta(C) := \min_{y \in C \setminus \{0^N\}} \{\mathsf{wt}(y)\}$.

Equivalently

$C \colon \mathbb{F}^n \to \mathbb{F}^N$ such that $C(x) := x\mathbf{G}$ for rank-n $\mathbf{G} \in \mathbb{F}^{n \times N}$ and $x \in \mathbb{F}^n$.

Definition 1 (Linear Codes)

Let $\mathbb{F}$ be a finite field. A $[N, n, d]$ linear error correcting code C for $n \leqslant N$ is a n-dimensional subspace of $\mathbb{F}^N$ such that the minimum distance of C, denoted as $\Delta(C)$, is d, where $\Delta(C) := \min_{y \in C \setminus \{0^N\}} \{\mathsf{wt}(y)\}$.

Equivalently

$C\colon \mathbb{F}^n \to \mathbb{F}^N$ such that $C(x) := x\mathbf{G}$ for rank-n $\mathbf{G} \in \mathbb{F}^{n \times N}$ and $x \in \mathbb{F}^n$.

Parameters of Interest

Linear Error-Correcting Codes

Definition 1 (Linear Codes)

Let $\mathbb{F}$ be a finite field. A $[N, n, d]$ linear error correcting code C for $n \leqslant N$ is a n-dimensional subspace of $\mathbb{F}^N$ such that the minimum distance of C, denoted as $\Delta(C)$, is d, where $\Delta(C) := \min_{y \in C \setminus \{0^N\}} \{\mathsf{wt}(y)\}$.

Equivalently

$C \colon \mathbb{F}^n \to \mathbb{F}^N$ such that $C(x) := x\mathbf{G}$ for rank-n $\mathbf{G} \in \mathbb{F}^{n \times N}$ and $x \in \mathbb{F}^n$.

Parameters of Interest

- **Rate:** $R = n/N$

Linear Error-Correcting Codes

Definition 1 (Linear Codes)

Let $\mathbb{F}$ be a finite field. A $[N, n, d]$ linear error correcting code C for $n \leqslant N$ is a n-dimensional subspace of $\mathbb{F}^N$ such that the minimum distance of C, denoted as $\Delta(C)$, is d, where $\Delta(C) := \min_{y \in C \setminus \{0^N\}} \{\mathsf{wt}(y)\}$.

Equivalently

$C\colon \mathbb{F}^n \to \mathbb{F}^N$ such that $C(x) := x\mathbf{G}$ for rank-n $\mathbf{G} \in \mathbb{F}^{n \times N}$ and $x \in \mathbb{F}^n$.

Parameters of Interest

- **Rate:** $R = n/N$
- **Encoding Time:** Time to compute $x \cdot \mathbf{G}$

Definition 1 (Linear Codes)

Let $\mathbb{F}$ be a finite field. A $[N, n, d]$ linear error correcting code C for $n \leqslant N$ is a n-dimensional subspace of $\mathbb{F}^N$ such that the minimum distance of C, denoted as $\Delta(C)$, is d, where $\Delta(C) := \min_{y \in C \setminus \{0^N\}} \{\mathsf{wt}(y)\}$.

Equivalently

$C \colon \mathbb{F}^n \to \mathbb{F}^N$ such that $C(x) := x\mathbf{G}$ for rank-n $\mathbf{G} \in \mathbb{F}^{n \times N}$ and $x \in \mathbb{F}^n$.

Parameters of Interest

- **Rate:** $R = n/N$
- **Encoding Time:** Time to compute $x \cdot \mathbf{G}$
- **Relative Distance:** $\delta(C) := \Delta(C)/N$

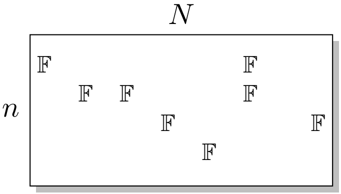
Expand-Accumulate Codes [BCG+22]

Expand-Accumulate Codes [BCG+22]

$$
\begin{array}{cc}
 & N \\
n & \begin{bmatrix}
\mathbb{F} & & & & & \mathbb{F} & & \\
 & \mathbb{F} & \mathbb{F} & & & \mathbb{F} & & \\
 & & & \mathbb{F} & & & & \mathbb{F} \\
 & & & & \mathbb{F} & & &
\end{bmatrix}
\end{array}
$$

Random sparse $\mathbf{E}$

Expand-Accumulate Codes [BCG+22]



Random sparse $\mathbf{E}$

$\mathbf{E}$ increases $\mathsf{wt}(x)$ w.h.p. (*expander*)

Expand-Accumulate Codes [BCG+22]

$$
\begin{array}{cc}
 & N \\
n & \begin{bmatrix}
\mathbb{F} & & & & & \mathbb{F} & & \\
 & \mathbb{F} & \mathbb{F} & & & \mathbb{F} & & \\
 & & & \mathbb{F} & & & & \mathbb{F} \\
 & & & & \mathbb{F} & & &
\end{bmatrix}
\end{array}
\qquad
\begin{array}{cc}
N & \\
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & 1 & 1 & 1 & 1 & 1 \\
 & & & & 1 & 1 & 1 & 1 \\
 & & & & & 1 & 1 & 1 \\
 & & & & & & 1 & 1 \\
 & & & & & & & 1
\end{bmatrix} & N
\end{array}
$$

Random sparse $\mathbf{E}$

$\mathbf{E}$ increases $\mathsf{wt}(x)$ w.h.p. (*expander*)

Accumulator $\mathbf{A}$

Expand-Accumulate Codes [BCG+22]

$$C(x) := x\mathbf{E}\mathbf{A}$$

$$
\overbrace{
\underbrace{n\left\{\begin{array}{cccccccc}
\mathbb{F} & & & & & \mathbb{F} & & \\
& \mathbb{F} & \mathbb{F} & & & \mathbb{F} & & \\
& & & \mathbb{F} & & & & \mathbb{F} \\
& & & & \mathbb{F} & & &
\end{array}\right.}_{N}
\quad
\underbrace{\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
& 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
& & 1 & 1 & 1 & 1 & 1 & 1 \\
& & & 1 & 1 & 1 & 1 & 1 \\
& & & & 1 & 1 & 1 & 1 \\
& & & & & 1 & 1 & 1 \\
& & & & & & 1 & 1 \\
& & & & & & & 1
\end{array}}_{N}\Bigg\}N
}
$$

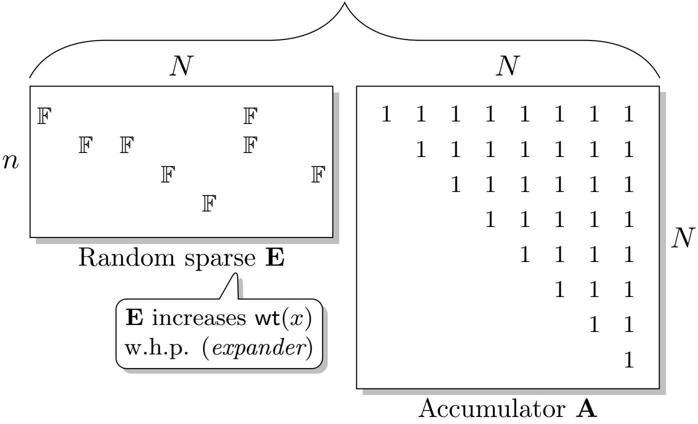
Random sparse **E**

E increases $\mathsf{wt}(x)$ w.h.p. (*expander*)

Accumulator **A**

Expand-Accumulate Codes [BCG+22]

$$C(x) := x\mathbf{E}\mathbf{A}$$



How to sample $\mathbf{E}$?

Sampling Expander Matrix

Sampling Expander Matrix

(Generalized) Bernoulli

$$\mathbf{E}_{i,j} \leftarrow \mathsf{Ber}_p(\mathbb{F}), \forall i, j$$

$$\mathsf{Ber}_p(\mathbb{F}) := \begin{cases} x \xleftarrow{\$} \mathbb{F} \setminus \{0\} & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$$

Sampling Expander Matrix

(Generalized) Bernoulli

$$\mathbf{E}_{i,j} \leftarrow \mathsf{Ber}_p(\mathbb{F}), \forall i, j$$

$$\mathsf{Ber}_p(\mathbb{F}) := \begin{cases} x \xleftarrow{\$} \mathbb{F} \setminus \{0\} & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$$

[BCG+22] prove $\delta(\mathbf{EA}) = \Theta(1)$ for $\mathbb{F}_2$ and $p = \Theta(\log(N)/N)$, conjecture same for $\mathbb{F}_{>2}$

Sampling Expander Matrix

(Generalized) Bernoulli

$$\mathbf{E}_{i,j} \leftarrow \mathsf{Ber}_p(\mathbb{F}), \forall i, j$$

$$\mathsf{Ber}_p(\mathbb{F}) := \begin{cases} x \xleftarrow{\$} \mathbb{F} \setminus \{0\} & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$$

[BCG+22] prove $\delta(\mathbf{EA}) = \Theta(1)$ for $\mathbb{F}_2$ and $p = \Theta(\log(N)/N)$, conjecture same for $\mathbb{F}_{>2}$

Fixed Row Weights

$$\mathbf{E}_i \leftarrow \mathsf{Fixed}(N, t, \mathbb{F}), \forall i \in [n]$$

$$\mathsf{Fixed}(N, t, \mathbb{F}) := \mathbb{U}(\{y \in \mathbb{F}^N : \mathsf{wt}(y) = t\})$$

Sampling Expander Matrix

(Generalized) Bernoulli

$$\mathbf{E}_{i,j} \leftarrow \mathsf{Ber}_p(\mathbb{F}), \forall i, j$$

$$\mathsf{Ber}_p(\mathbb{F}) := \begin{cases} x \xleftarrow{\$} \mathbb{F} \setminus \{0\} & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$$

[BCG+22] prove $\delta(\mathbf{EA}) = \Theta(1)$ for $\mathbb{F}_2$ and $p = \Theta(\log(N)/N)$, conjecture same for $\mathbb{F}_{>2}$

Fixed Row Weights

$$\mathbf{E}_i \leftarrow \mathsf{Fixed}(N, t, \mathbb{F}), \forall i \in [n]$$

$$\mathsf{Fixed}(N, t, \mathbb{F}) := \mathbb{U}(\{y \in \mathbb{F}^N : \mathsf{wt}(y) = t\})$$

[BCG+22] conjecture $\delta(\mathbf{EA}) = \Theta(1)$ for $\mathbb{F}$ and $t = \Theta(\log(N))$

EA Codes: Our Approach

Juxtaposed EA Code

$$C[\mathbf{E}_1, \mathbf{E}_2](x) := (x\mathbf{E}_1\mathbf{A}) \| (x\mathbf{E}_2\mathbf{A})$$

$$\mathbf{E}_1 \leftarrow \mathsf{Fixed}(n, N, t, \mathbb{F}) \text{ and } \mathbf{E}_2 \leftarrow \mathsf{Ber}_p^{n\times N}(\mathbb{F})$$

$$R = n/N = \Theta(1),\ t = \Theta(\log(N)),\ p = t/N$$

Juxtaposed EA Codes: Our Results

Theorem 1

Over any $\mathbb{F}$*, for* $R = n/N$ *constant, there exist constants* $\delta \in (0, 1/2)$ *and* $c^* > 5$ *such that for* $t = \Theta(\log(N))$ *and* $p = t/N$*, the juxtaposed EA code* $C[\mathbf{E}_1, \mathbf{E}_2]$ *over* $\mathbb{F}$ *has constant relative distance* δ *with at least* $1 - 1/\text{poly}(N^{5-c^*})$ *probability.*

If $\mathbb{F} = \mathbb{F}_2$*, then the above holds for* $c^* > 4$ *with probability at least* $1 - 1/\text{poly}(N^{4-c^*})$

Juxtaposed EA Codes: Our Results

Theorem 1

Over any $\mathbb{F}$, *for* $R = n/N$ *constant, there exist constants* $\delta \in (0, 1/2)$ *and* $c^* > 5$ *such that for* $t = \Theta(\log(N))$ *and* $p = t/N$, *the juxtaposed EA code* $C[\mathbf{E}_1, \mathbf{E}_2]$ *over* $\mathbb{F}$ *has constant relative distance* δ *with at least* $1 - 1/\mathrm{poly}(N^{5-c^*})$ *probability.*

If $\mathbb{F} = \mathbb{F}_2$, *then the above holds for* $c^* > 4$ *with probability at least* $1 - 1/\mathrm{poly}(N^{4-c^*})$

Notes

- We consider juxtaposed EA codes due to limitations in our analysis of $\mathbf{E}_1$.

Juxtaposed EA Codes: Our Results

Theorem 1

Over any $\mathbb{F}$, for $R = n/N$ constant, there exist constants $\delta \in (0, 1/2)$ and $c^* > 5$ such that for $t = \Theta(\log(N))$ and $p = t/N$, the juxtaposed EA code $C[\mathbf{E}_1, \mathbf{E}_2]$ over $\mathbb{F}$ has constant relative distance δ with at least $1 - 1/\mathrm{poly}(N^{5-c^*})$ probability.

If $\mathbb{F} = \mathbb{F}_2$, then the above holds for $c^* > 4$ with probability at least $1 - 1/\mathrm{poly}(N^{4-c^*})$

Notes

- We consider juxtaposed EA codes due to limitations in our analysis of $\mathbf{E}_1$.
- We conjecture both $\mathbf{E}_1\mathbf{A}$ and $\mathbf{E}_2\mathbf{A}$ are good codes.

Juxtaposed EA Codes: Our Results

Theorem 1

Over any $\mathbb{F}$, for $R = n/N$ constant, there exist constants $\delta \in (0, 1/2)$ and $c^* > 5$ such that for $t = \Theta(\log(N))$ and $p = t/N$, the juxtaposed EA code $C[\mathbf{E}_1, \mathbf{E}_2]$ over $\mathbb{F}$ has constant relative distance δ with at least $1 - 1/\mathrm{poly}(N^{5-c^*})$ probability.

If $\mathbb{F} = \mathbb{F}_2$, then the above holds for $c^* > 4$ with probability at least $1 - 1/\mathrm{poly}(N^{4-c^*})$

Notes

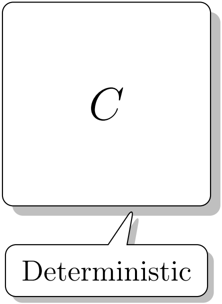
- We consider juxtaposed EA codes due to limitations in our analysis of $\mathbf{E}_1$.
- We conjecture both $\mathbf{E}_1\mathbf{A}$ and $\mathbf{E}_2\mathbf{A}$ are good codes.
- Parameters in above theorem are nowhere near tight; can be tightened up with better Stirling approximations.

IOWE Technique

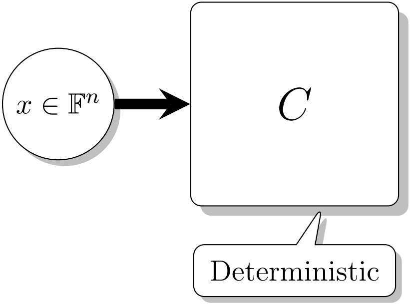
Input-Output Weight Enumerator

IOWE Technique

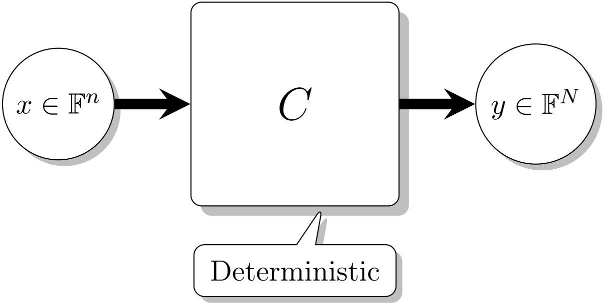
Input-Output Weight Enumerator



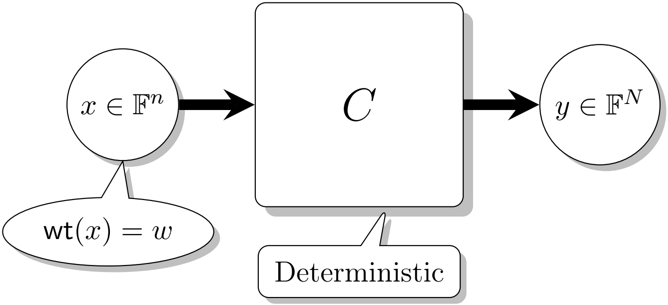
Input-Output Weight Enumerator



Input-Output Weight Enumerator

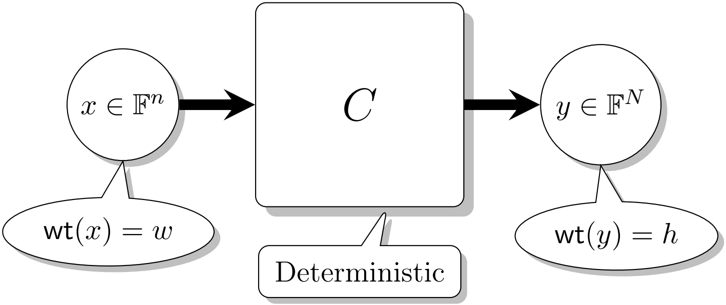


Input-Output Weight Enumerator

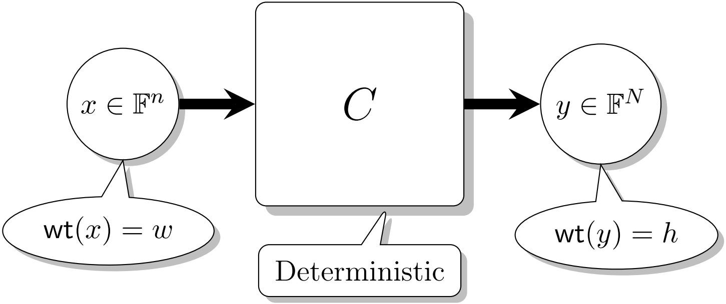


IOWE Technique

Input-Output Weight Enumerator



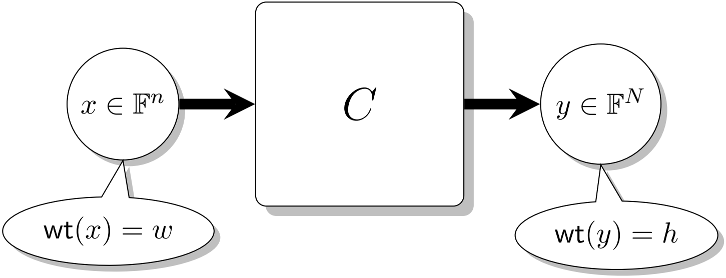
Input-Output Weight Enumerator

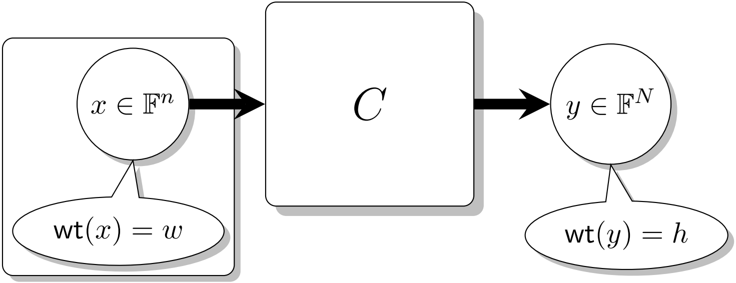


Definition 2 (IOWE)

$$C^N_{w,h} := \left|\{x \in \mathbb{F}^n : \mathsf{wt}(x) = w \ \wedge \ \mathsf{wt}(C(x)) = h\}\right|$$

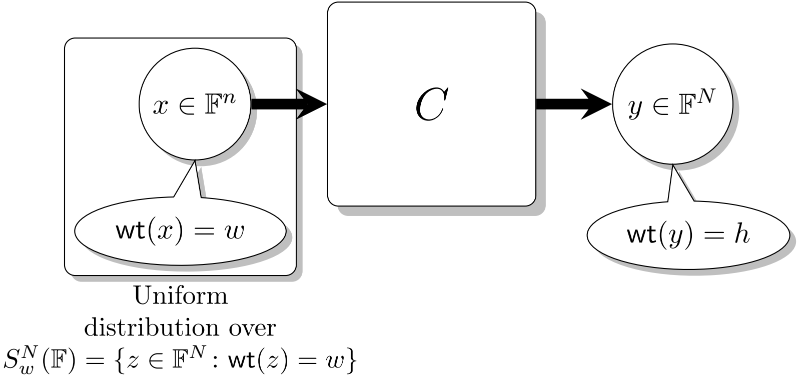
IOWE Technique





Uniform
distribution over
$S_w^N(\mathbb{F}) = \{z \in \mathbb{F}^N : \mathsf{wt}(z) = w\}$

IOWE Technique



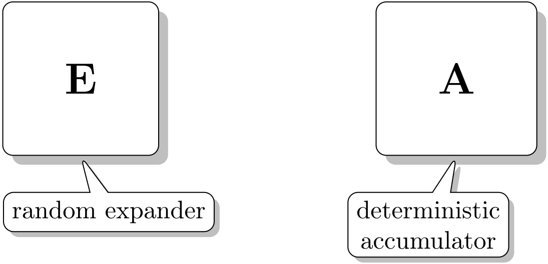
$$\Pr[\mathsf{wt}(C(x)) = h \colon x \xleftarrow{\$} S_w^N(\mathbb{F})] = \frac{C_{w,h}^N}{\binom{n}{w}}$$

IOWE Technique for EA Codes

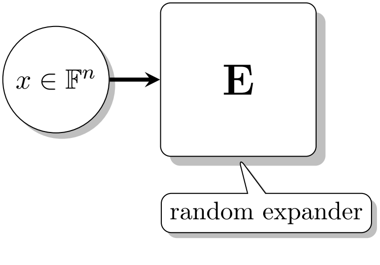
E

A

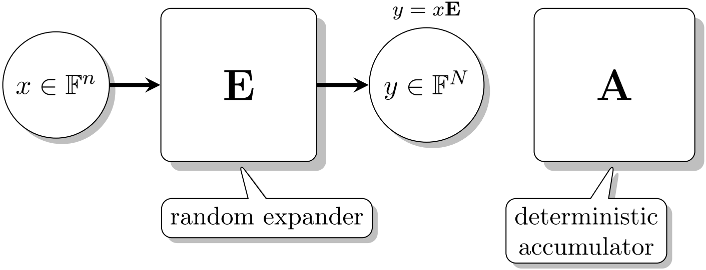
IOWE Technique for EA Codes



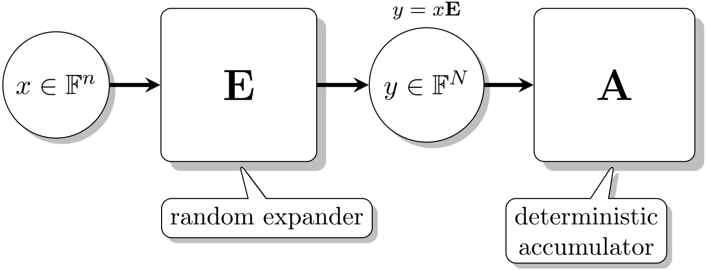
IOWE Technique for EA Codes



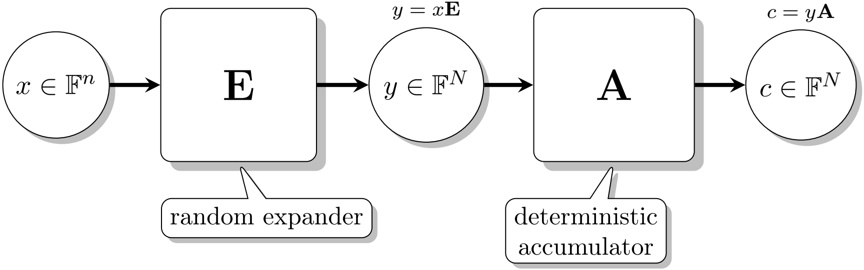
IOWE Technique for EA Codes



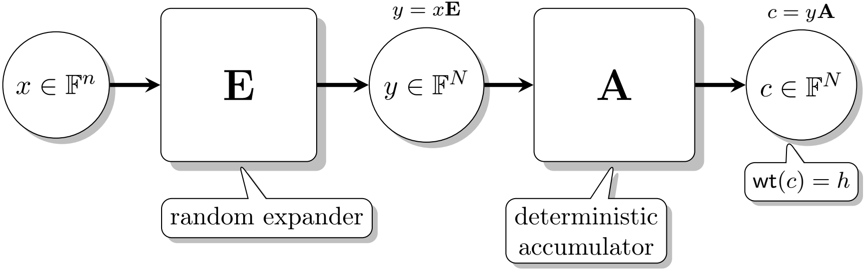
IOWE Technique for EA Codes



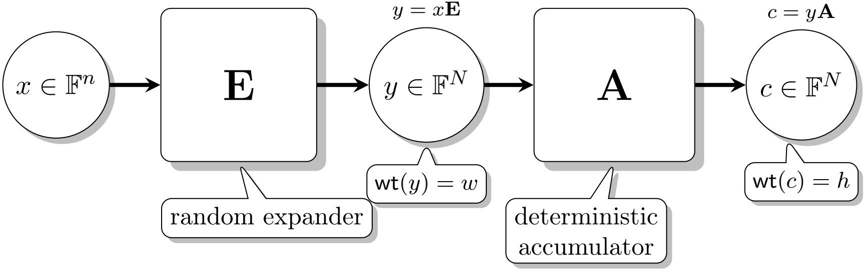
IOWE Technique for EA Codes



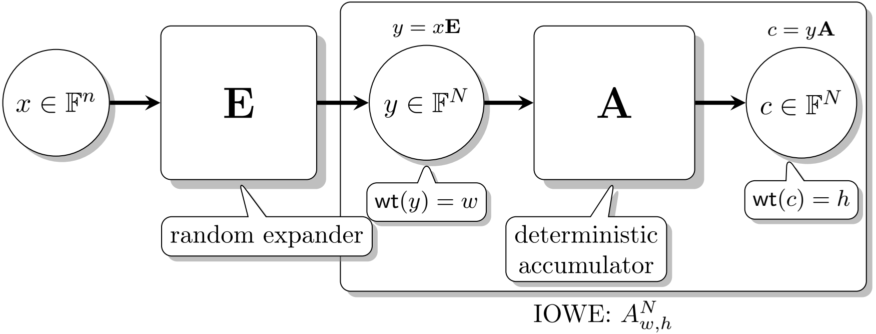
IOWE Technique for EA Codes



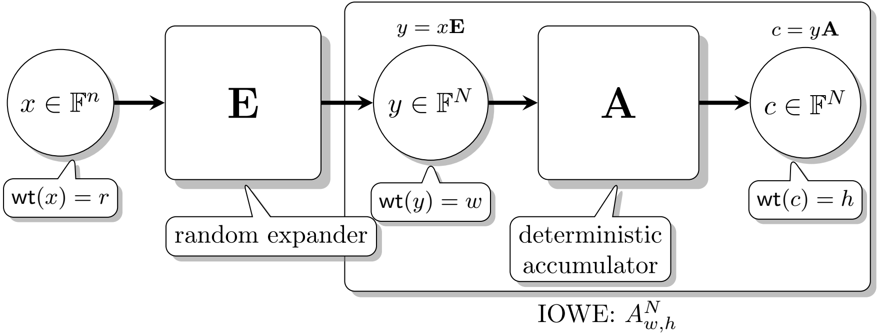
IOWE Technique for EA Codes



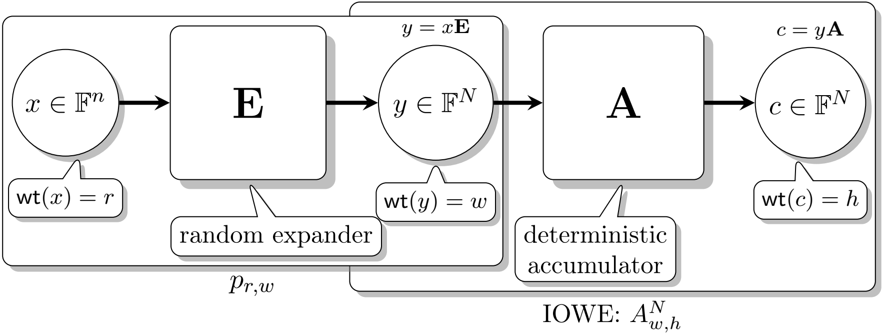
IOWE Technique for EA Codes



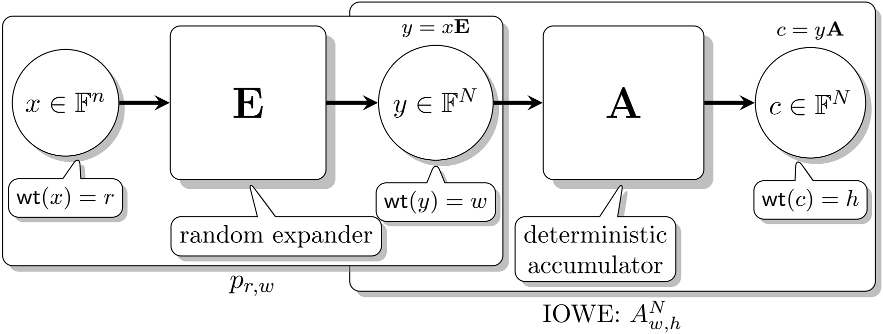
IOWE Technique for EA Codes



IOWE Technique for EA Codes

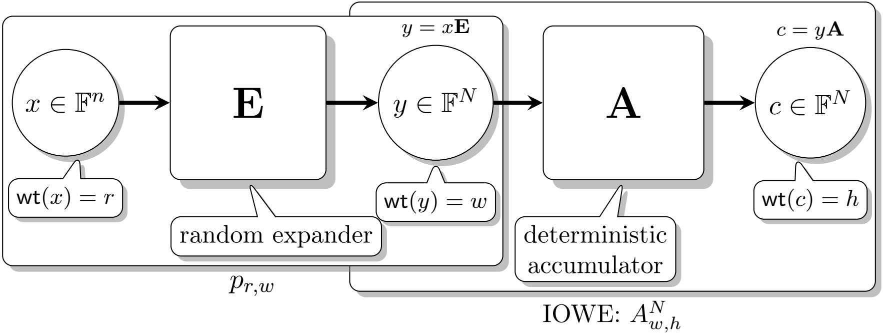


IOWE Technique for EA Codes



Properties of $\mathbf{E}$

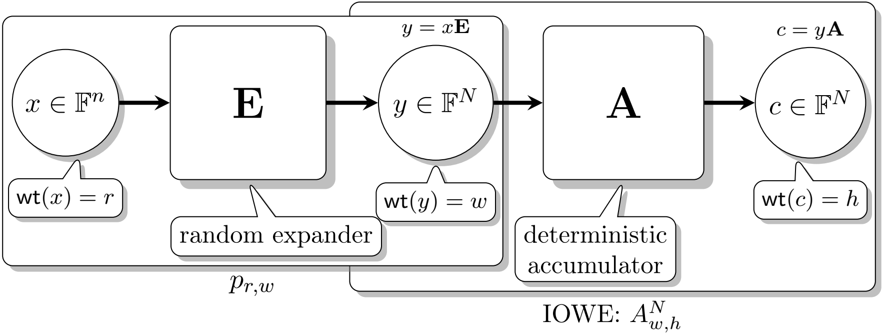
IOWE Technique for EA Codes



Properties of $\mathbf{E}$

- $p_{r,w} = \Pr_{\mathbf{E}}[\mathsf{wt}(x\mathbf{E}) = w \mid \mathsf{wt}(x) = r]$

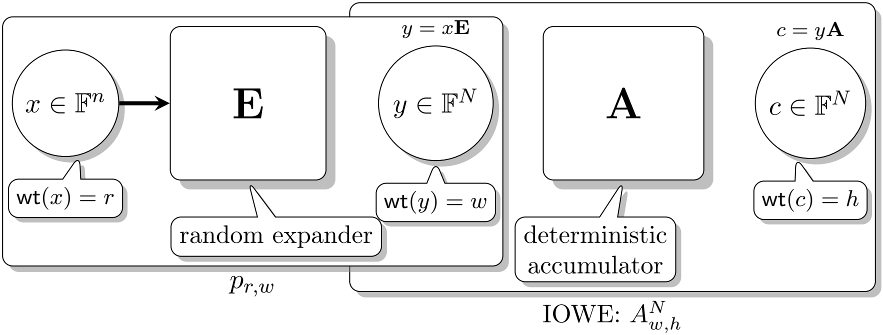
IOWE Technique for EA Codes



Properties of $\mathbf{E}$

- $p_{r,w} = \Pr_{\mathbf{E}}[\mathsf{wt}(x\mathbf{E}) = w \mid \mathsf{wt}(x) = r]$
- If $\mathsf{wt}(y) = w$, then $y \sim \mathbb{U}(\{z \in \mathbb{F}^N : \mathsf{wt}(z) = w\})$

IOWE Technique for EA Codes



Distance Analysis (Binary Case)

$$\Pr_{\mathbf{E}}\left[\exists x \in \mathbb{F}_2^n \setminus \{0^n\} \colon \mathsf{wt}(x\mathbf{EA}) \leqslant \delta N\right] \leqslant$$

$$\sum_{r=1}^{n} \binom{n}{r} \cdot \sum_{w=1}^{N} p_{r,w} \cdot \sum_{h=1}^{\delta N} A^N_{w,h} \Big/ \binom{N}{w}$$

Accumulator IOWE

Binary IOWE Accumulator [DJM98]

$$A^{N,2}_{w,h} = \binom{h-1}{\lceil w/2 \rceil - 1}\binom{N-h}{\lfloor w/2 \rfloor}$$

Accumulator IOWE

Binary IOWE Accumulator [DJM98]

$$A_{w,h}^{N,2} = \binom{h-1}{\lceil w/2 \rceil - 1}\binom{N-h}{\lfloor w/2 \rfloor}$$

Theorem 2

For finite field $\mathbb{F}_q$, $N \in \mathbb{N}$, and $w, h \in [N]$, the IOWE of the $N \times N$ accumulator matrix over $\mathbb{F}_q$ is

$$A_{w,h}^{N,q} = \sum_{i=0}^{w-1} \binom{h-1}{\left\lceil \frac{w-i}{2} \right\rceil - 1}\binom{N-h}{\left\lfloor \frac{w-i}{2} \right\rfloor}\binom{h - \left\lceil \frac{w-i}{2} \right\rceil}{i}(q-1)^{\left\lceil \frac{w-i}{2} \right\rceil}(q-2)^i.$$

Distance of EA Codes over Finite Fields

- Given IOWE $A_{w,h}^{N,q}$, we can directly bound the distance of the EA code over any $\mathbb{F}$

Distance of EA Codes over Finite Fields

- Given IOWE $A^{N,q}_{w,h}$, we can directly bound the distance of the EA code over any $\mathbb{F}$, right?

Distance of EA Codes over Finite Fields

- Given IOWE $A_{w,h}^{N,q}$, we can directly bound the distance of the EA code over any $\mathbb{F}$, right?

Distance Analysis ($\mathbb{F}_q$ Case)

$$\sum_{r=1}^{n} \binom{n}{r} (q-1)^r \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N} A_{w,h}^{N,q} / \binom{N}{w} (q-1)^w =$$

Distance of EA Codes over Finite Fields

- Given IOWE $A^{N,q}_{w,h}$, we can directly bound the distance of the EA code over any $\mathbb{F}$, right?

Distance Analysis ($\mathbb{F}_q$ Case)

$$\sum_{r=1}^{n} \binom{n}{r} (q-1)^r \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N} A^{N,q}_{w,h} / \binom{N}{w} (q-1)^w =$$

$$\sum_{r=1}^{n} \binom{n}{r} (q-1)^r \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N}$$

$$\frac{\sum_{i=0}^{w-1} \binom{h-1}{\lceil \frac{w-i}{2} \rceil - 1} \binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h - \lceil \frac{w-i}{2} \rceil}{i} (q-1)^{\lceil \frac{w-i}{2} \rceil} \cdot (q-2)^i}{\binom{N}{w} (q-1)^w}$$

Distance of EA Codes over Finite Fields

- Given IOWE $A^{N,q}_{w,h}$, we can directly bound the distance of the EA code over any $\mathbb{F}$, right?

Distance Analysis ($\mathbb{F}_q$ Case)

$$\sum_{r=1}^{n} \binom{n}{r} (q-1)^r \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N} A^{N,q}_{w,h} \Big/ \binom{N}{w} (q-1)^w =$$

$$\sum_{r=1}^{n} \binom{n}{r} (q-1)^r \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N}$$

$$\frac{\sum_{i=0}^{w-1} \binom{h-1}{\lceil \frac{w-i}{2} \rceil - 1} \binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h - \lceil \frac{w-i}{2} \rceil}{i} (q-1)^{\lceil \frac{w-i}{2} \rceil} \cdot (q-2)^i}{\binom{N}{w} (q-1)^w}$$

We were unable to bound this for $q > 4$!

Overcoming the Union Bound

Main Observation

Naively applying the Union Bound does not work!

Overcoming the Union Bound

Main Observation

Naively applying the Union Bound does not work!

Overcoming the Union Bound

- We can carefully refine the Union Bound step-by-step rather than applying it in one-shot.

Overcoming the Union Bound

Main Observation

Naively applying the Union Bound does not work!

Overcoming the Union Bound

- We can carefully refine the Union Bound step-by-step rather than applying it in one-shot.
- Intuition: carefully partition unions of events until we can apply a truncated Union Bound on events that depend on the field size q.

Overcoming the Union Bound

Main Observation

Naively applying the Union Bound does not work!

Overcoming the Union Bound

- We can carefully refine the Union Bound step-by-step rather than applying it in one-shot.
- Intuition: carefully partition unions of events until we can apply a truncated Union Bound on events that depend on the field size q.
- Final distance bound we analyze is:

Overcoming the Union Bound

Main Observation

Naively applying the Union Bound does not work!

Overcoming the Union Bound

- We can carefully refine the Union Bound step-by-step rather than applying it in one-shot.
- Intuition: carefully partition unions of events until we can apply a truncated Union Bound on events that depend on the field size q.
- Final distance bound we analyze is:

$$\sum_{r=1}^{n} \binom{n}{r} \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N} \frac{\sum_{i=0}^{w-1} \binom{h-1}{\lceil \frac{w-i}{2} \rceil - 1} \binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h - \lceil \frac{w-i}{2} \rceil}{i}}{\binom{N}{w}}$$

Overcoming the Union Bound

Main Observation

Naively applying the Union Bound does not work!

Overcoming the Union Bound

- We can carefully refine the Union Bound step-by-step rather than applying it in one-shot.
- Intuition: carefully partition unions of events until we can apply a truncated Union Bound on events that depend on the field size q.
- Final distance bound we analyze is:

$$\sum_{r=1}^{n} \binom{n}{r} \sum_{w=1}^{N} p_{r,w} \sum_{h=1}^{\delta N} \frac{\sum_{i=0}^{w-1} \binom{h-1}{\lceil \frac{w-i}{2} \rceil - 1} \binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h - \lceil \frac{w-i}{2} \rceil}{i}}{\binom{N}{w}}$$

Looks like binary case; able to bound this!

Experiments

- Implementation of PCS + SNARK in Rust
- SNARK relies on Spartan PIOP [Set20]
- Artifact available: https://artifacts.iacr.org/crypto/2024/a10/

Parameters

- Distance $\delta = 1/10$ with probability 2^{-100}, calculated numerically
- Rate $R = 1/2$, $n \in 2^{\{10,11,12\}}$, $N = n/R$
- Sparsity $t \geqslant 18 \log(N)$
- $\mathbb{F}$ is the scalar field of the BN254 curve unless otherwise stated.
- "Brakedown-improved" refers to using the improved Brakedown parameters due to [Hab23]

Experiments

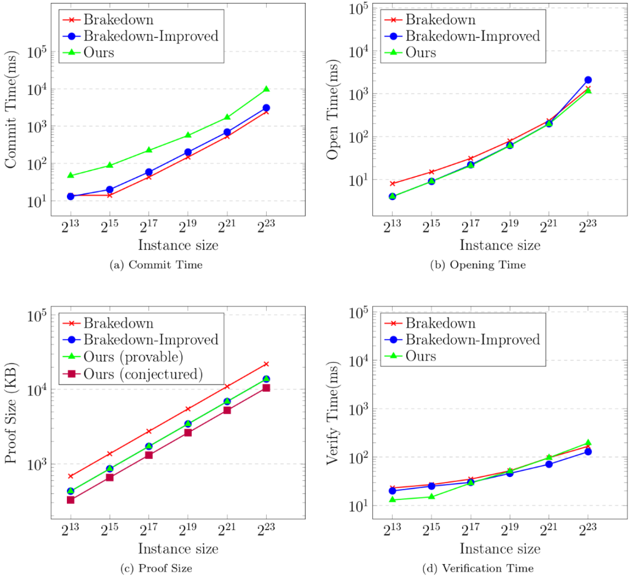


Figure 1: Performance of polynomial commitment schemes.

EXPERIMENTS

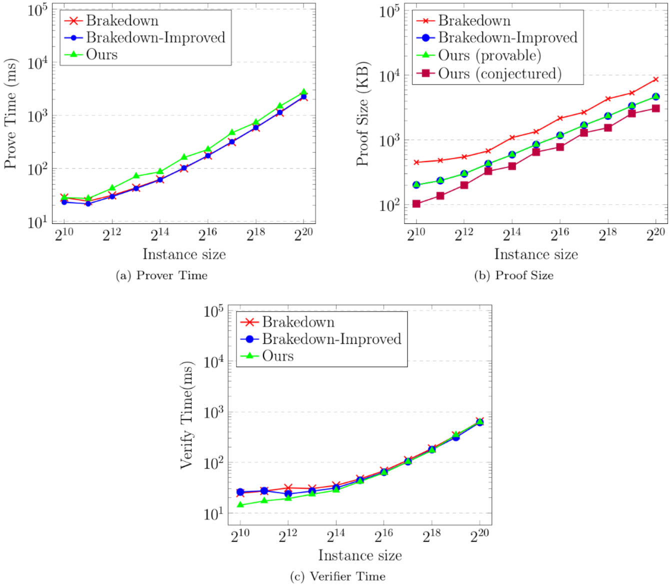


Figure 2: Performance of SNARKs on random R1CS instances.

Table 2: Performance of different SNARKs for ECDSA verification. Native field is the scalar field of the secp256k1 curve.

R1CS Size	Scheme	Prover time	Proof size	Verifier time
2^{21} (non-native)	Ligero	103s	20 MB	57 s
2^{21} (non-native)	Aurora	534s	148 KB	15.2 s
2^{21} (non-native)	Groth16	149s	128 B	2 ms
2^{16} (native)	Brakedown	0.17s	2.2 MB	62 ms
2^{16} (native)	Brakedown-Improved	0.17s	1.1 MB	64 ms
2^{16} (native)	Ours (provable)	0.23s	1.1 MB	68 ms
2^{16} (native)	Ours (conjectured)	0.23s	778 KB	67 ms

Table 2: Performance of different SNARKs for ECDSA verification. Native field is the scalar field of the secp256k1 curve.

R1CS Size	Scheme	Prover time	Proof size	Verifier time
2^{21} (non-native)	Ligero	103s	20 MB	57 s
2^{21} (non-native)	Aurora	534s	148 KB	15.2 s
2^{21} (non-native)	Groth16	149s	128 B	2 ms
2^{16} (native)	Brakedown	0.17s	2.2 MB	62 ms
2^{16} (native)	Brakedown-Improved	0.17s	1.1 MB	64 ms
2^{16} (native)	Ours (provable)	0.23s	1.1 MB	68 ms
2^{16} (native)	Ours (conjectured)	0.23s	778 KB	67 ms

Comparable to Brakedown

Experiments

Table 2: Performance of different SNARKs for ECDSA verification. Native field is the scalar field of the secp256k1 curve.

R1CS Size	Scheme	Prover time	Proof size	Verifier time
2^{21} (non-native)	Ligero	103s	20 MB	57 s
2^{21} (non-native)	Aurora	534s	148 KB	15.2 s
2^{21} (non-native)	Groth16	149s	128 B	2 ms
2^{16} (native)	Brakedown	0.17s	2.2 MB	62 ms
2^{16} (native)	Brakedown-Improved	0.17s	1.1 MB	64 ms
2^{16} (native)	Ours (provable)	0.23s	1.1 MB	68 ms
2^{16} (native)	Ours (conjectured)	0.23s	778 KB	67 ms

Concretely small proofs

Comparable to Brakedown

Experiments

Table 2: Performance of different SNARKs for ECDSA verification. Native field is the scalar field of the secp256k1 curve.

R1CS Size	Scheme	Prover time	Proof size	Verifier time
2^{21} (non-native)	Ligero	103s	20 MB	57 s
2^{21} (non-native)	Aurora	534s	148 KB	15.2 s
2^{21} (non-native)	Groth16	149s	128 B	2 ms
2^{16} (native)	Brakedown	0.17s	2.2 MB	62 ms
2^{16} (native)	Brakedown-Improved	0.17s	1.1 MB	64 ms
2^{16} (native)	Ours (provable)	0.23s	1.1 MB	68 ms
2^{16} (native)	Ours (conjectured)	0.23s	778 KB	67 ms

Slightly slower than Brakedown

Concretely small proofs

Comparable to Brakedown

Summary

New Code-PCS from Expand-Accumulate Codes via the Brakedown PCS Framework

1. Improved distance analysis of binary EA codes

- Provide an alternate distance analysis + proof via classic coding techniques
- Better concrete parameters than [BCG+22]

2. Generalize EA codes to arbitrary finite fields

- Extend binary analysis to FFs
- Answers open problem of [BCG+22]
- Quasi-linear encoding, concretly large min distance

3. Concretely efficient SNARKs from EA codes

- Field-agnostic
- Proof size + verifier time comparable with [GLS+23], only $\approx 1.2\times$ prover overhead

Thank you!

References I

[BCG+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 603–633. Springer, Cham, August 2022. DOI: 10.1007/978-3-031-15979-4_21.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In *Theory of Cryptography Conference*, pages 19–46. Springer, 2020.

[DJM98] Dariush Divsalar, Hui Jin, and Robert J McEliece. Coding theorems for "turbo-like" codes. In *Proceedings of the annual Allerton Conference on Communication control and Computing*, volume 36, pages 201–210, 1998.

[GLS+23] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: linear-time and field-agnostic SNARKs for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 193–226. Springer, Cham, August 2023. DOI: 10.1007/978-3-031-38545-2_7.

[Hab23] Ulrich Haböck. Brakedown's expander code. *Cryptology ePrint Archive*, 2023.

References II

[Set20] Srinath Setty. Spartan: efficient and general-purpose zkSNARKs without trusted setup. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 704–737. Springer, Cham, August 2020. DOI: 10.1007/978-3-030-56877-1_25.

[Tha22] Justin Thaler. Proofs, arguments, and zero-knowledge. *Found. Trends Priv. Secur.*, 4(2-4):117–660, 2022. URL: https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.html.