Field-Agnostic SNARKs from Expand-Accumulate Codes

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

[1]University of Illinois at Chicago. `alexander.r.block@gmail.com`
[2]Texas A&M University. `zhiyong.fang.1997@gmail.com`
[3]Google and University of Maryland. `jkatz2@gmail.com`
[4]a16z crypto research and Georgetown University. `justin.r.thaler@gmail.com`
[5]Nethermind Research. `hendrik.waldner@nethermind.io`
[6]University of Illinois Urbana-Champaign. `zhangyp@illinois.edu`

November 15, 2024

Abstract

Efficient realizations of succinct non-interactive arguments of knowledge (SNARKs) have gained popularity due to their practical applications in various domains. Among existing schemes, those based on error-correcting codes are of particular interest because of their good concrete efficiency, transparent setup, and plausible post-quantum security. However, many existing code-based SNARKs suffer from the disadvantage that they only work over specific finite fields.

In this work, we construct a code-based SNARK that does not rely on any specific underlying field; i.e., it is *field-agnostic*. Our construction follows the framework of Brakedown (CRYPTO '23) and builds a polynomial commitment scheme (and hence a SNARK) based on recently introduced *expand-accumulate codes*. Our work generalizes these codes to arbitrary finite fields; our main technical contribution is showing that, with high probability, these codes have constant rate and constant relative distance (crucial properties for building efficient SNARKs), solving an open problem from prior work.

As a result of our work we obtain a SNARK where, for a statement of size M, the prover time is $O(M \log M)$ and the proof size is $O(\sqrt{M})$. We demonstrate the concrete efficiency of our scheme empirically via experiments. Proving ECDSA verification on the secp256k1 curve requires only 0.23s for proof generation, 2 orders of magnitude faster than SNARKs that are not field-agnostic. Compared to the original Brakedown result (which is also field-agnostic), we obtain proofs that are 1.9–2.8$\times$ smaller due to the good concrete distance of our underlying error-correcting code, while introducing only a small overhead of 1.2$\times$ in the prover time.

1 Introduction

Succinct non-interactive arguments of knowledge (SNARKs) allow a prover to convince a verifier of the truth of some statement via a (non-interactive) proof of sublinear size. SNARKs have been the subject of extensive interest in the past few years, leading to efficient implementations and real-world applications, most notably in the blockchain domain. Among candidate constructions, SNARKs based on error-correcting codes [AHIV17, BBHR19, BCR+19, ZXZS20, BCG+17, BCG20, BCL22, GLS+23, XZS22] have recently received attention because of their concrete efficiency, transparent setup, and plausible post-quantum security. However, existing schemes based on error-correcting codes suffer from either or both of the following disadvantages:

*Work done while at Georgetown University and the University of Maryland.
†Work done while at the University of Maryland.

Reliance on specific fields: Schemes that are based on Reed-Solomon codes (e.g., Ligero [AHIV17], Stark [BBHR19], Aurora [BCR+19], and Virgo [ZXZS20]) rely on the Fast Fourier Transform (FFT) to obtain quasi-linear prover time, making such schemes best suited for "FFT-friendly" fields (i.e., fields containing a multiplicative subgroup of order 2^ϕ for large ϕ). This limitation can lead to significant overhead when proving computations in a different field. For example, the verification circuit for an ECDSA signature on the secp256k1 curve (important for zkRollups [But] and zkBridges [XZC+22]) is $\approx 25\times$ larger when written over an FFT-friendly field rather than the native underlying field.

Large proofs: SNARKs based on any linear error-correcting code [BCG+17, BCG20, BCL22], including those with implementations such as Brakedown [GLS+23], are field-agnostic and offer linear prover time if the code has linear-time encoding [Spi96, DI14]. However, these schemes suffer from larger proof sizes in practice due to the poor concrete distance of the underlying code.

In this paper, we present a new code-based SNARK that is field-agnostic and has small concrete proof size. It is known from prior work that SNARKs can be built from a polynomial interactive oracle proof (PIOP) and a polynomial commitment scheme (PCS). We follow the framework of Brakedown [GLS+23] by building a polynomial commitment scheme (and thus a SNARK) from a linear code, but instead of the linear codes used in prior work [Spi96, DI14, GLS+23], we utilize recently proposed expand-accumulate (EA) codes [BCG+22]. EA codes have a simple and highly efficient encoding algorithm (see Section 2.1 for the formal description). However, their distance was only proven to be asymptotically good over the binary field [BCG+22], and an analysis over general fields was left open. A constant relative distance of the code is necessary to prove the soundness of the PCS and the SNARK, and better distance leads to smaller proof size (see Section 5 for the formula). Moreover, the encoding time determines prover time of the PCS and the SNARK. We address this as follows.

Improved distance analysis of the binary EA code. First, we propose an alternative approach for proving that the EA code over binary fields is asymptotically good (i.e., that the relative minimum distance of the code is a constant). This is crucial for their application to code-based SNARKs [BCG+17, AHIV17, GLS+23] whose proof size is inversely proportional to the relative distance of the code used. For message size $n > 2^{10}$ and rate $\frac{1}{2}$, we achieve a provable relative distance of $\delta = 0.1$ except with probability at most 2^{-100}. This is concretely better than prior provable bounds [BCG+22], which give a distance of $\delta = 0.023$ with the same error probability.

Generalization to large finite fields. More importantly, we generalize the above approach to the case of EA codes over arbitrary finite fields and thus formally prove that those codes are asymptotically good as well. For parameters as above, we can achieve the same provable distance $\delta = 0.1$ over any finite field. This is better than the code used in Brakedown, which achieves $\delta = 0.04$ for rate 0.65 [GLS+23, Figure 2].

The same technique can be applied to analyze other linear codes similar to the EA code, such as the repeat-accumulate code [DJM98, GM08] and the expand-convolute code [RRT23]. It may therefore be of independent interest for analyzing primitives beyond SNARKs, such as pseudorandom correlation generators [BCG+22, RRT23].

Field-agnostic SNARKs from EA codes. We use the EA code to construct a field-agnostic SNARK with prover time $O(M \log M)$ and proof size $O(\sqrt{M})$, where M is the size of the statement being proved (when represented using R1CS). We provide a comparison to relevant prior work in Table 1. Although the prover time is asymptotically quasilinear, we show that it is concretely comparable to existing schemes with linear prover time due to the simple encoding algorithm for the EA code. For example, with $M = 2^{20}$ constraints our prover time is 2.7s, which is only $1.2\times$ slower than Brakedown. At the same time, our proof size is 1.9–2.8$\times$ smaller than Brakedown (depending on whether provable or conjectured distance bounds are used). Moreover, the fact that the SNARK is field-agnostic makes it very flexible; for example, we are able to generate proofs of ECDSA signature verification on the secp256k1 curve in only 0.23s, with a proof of size 0.78–1.1MB. This performance is significantly better than the performance of non-field-agnostic SNARKs.

Table 1: Performance of SNARKs based on linear codes for a statement modeled as an arithmetic circuit of size M and depth d. The setting of the experiments is provided in Section 6. The reported verifier asymptotics assume that certain polynomials capturing the constraint system or circuit can be evaluated at any desired point in, say, polylogarithmic time. If this is not the case, the polynomials can be committed by any honest party in a linear-time pre-processing phase to allow for sublinear verification in the online phase.

	Prover Time	Proof Size	Verifier Time	Field Agnostic?
Ligero [AHIV17]	$O(M \log M)$	$O(\sqrt{M})$	$O(\sqrt{M})$	✗
Aurora [BCR+19]	$O(M \log M)$	$O(\log^2 M)$	$O(M)$	✗
Stark [BBHR19]	$O(M \log M)$	$O(\log^2 M)$	$O(\log^2 M)$	✗
Virgo [ZXZS20]	$O(M \log M)$	$O(d \log M)$	$O(d \log M)$	✗
Orion [XZS22]	$O(M)$	$O(\log^2 M)$	$O(\log^2 M)$	✗
Brakedown [GLS+23]	$O(M)$	$O(\sqrt{M})$	$O(\sqrt{M})$	✓
BaseFold [ZCF23]	$O(M \log M)$	$O(\log^2 M)$	$O(\log^2 M)$	✓
This Work	$O(M \log M)$	$O(\sqrt{M})$	$O(\sqrt{M})$	✓
Proof of ECDSA verification				
Ligero [AHIV17]	103s	20MB	57s	✗
Aurora [BCR+19]	534s	0.15MB	15.2s	✗
Brakedown [GLS+23]	0.17s	2.2MB	0.062s	✓
BaseFold [ZCF23]	0.273s	5.5MB	0.021s	✓
Ours (provable)	0.23s	1.1MB	0.068s	✓
Ours (conjectured)	0.23s	0.78MB	0.067s	✓

1.1 Related Work

In recent years, there has been a significant progress on building concretely efficient SNARKs [Tha22]. Constructions can be based on various cryptographic techniques, such as bilinear pairings [PHGR13, Gro16, GWC19], discrete logarithms [BCC+16, BBB+18, BHR+20], groups of unknown order [BFS20, BHR+21], interactive proofs [WTs+18, XZZ+19], interactive oracle proofs (IOPs) [AHIV17, BBHR19, BCR+19, ZXZS20], and lattices [BBC+18, BLNS20, BCS23]. Our work focuses on SNARKs based on linear error-correcting codes.

The idea of building SNARKs using error-correcting codes was proposed by Bootle et al. [BCG+17] and later refined using the technique of tensor IOPs [BCG20]. Subsequent work [BCL22] further achieved zero-knowledge and succinct verification time. On the practical side, Golovnev et al. [GLS+23] proposed building a polynomial commitment using tensor IOPs; this led to Brakedown, which has been implemented. Xie et al. [XZS22] improved the proof size of Brakedown from square-root to polylogarithmic using code-switching, but the resulting scheme is not field-agnostic. All these schemes use the error-correcting code of Spielman [Spi96] and its generalizations and/or refinements [DI14, GLS+23] to achieve a linear-time prover. However, the proof size is concretely large because the minimum distance of the code is low. Recently, Zeilberger et al. [ZCF23] construct BaseFold, which is a field-agnostic SNARK based on a different approach. As shown in Table 1, the prover time in our scheme is a little better than BaseFold, and our proof size is significantly smaller. (Performance of BaseFold was estimated based on [ZCF23, Section 6].)

Another line of work [BCKL23, BCKL22] introduces new techniques to allow field-specific SNARKs (in particular, those based on DEEP-ALI [BGKS20], related to FRI [BBHR18] and Stark [BBHR19]) to operate over arbitrary finite fields. Intuitively, if a given computation is over $\mathbb{F}_p$ for some prime power p, these works give a way to embed the computation into a suitably large extension field $\mathbb{F}_q$ such that $\mathbb{F}_q$ has the required structure for the field-specific SNARK (e.g., for Reed-Solomon codes, $\mathbb{F}_q$ has a large enough cyclic subgroup of order 2^{ϕ}). However, for an instance of size M, it is required that $q = \Omega(2^{2M})$, leading to much larger fields (and thus proof sizes, in terms of bits) to obtain field-agnosticism. This also says nothing of the concrete overheads needed to find a suitable field, and the fact that the underlying SNARK would still rely on expensive FFTs (in the case of those using Reed-Solomon codes).

EA Codes. EA codes were proposed by Boyle et al. [BCG+22], where they were used for generating correlated pseudorandomness for secure computation. A variant called expand-convolute codes [RRT23] has also been proposed. Both works only analyze the minimum distance of the codes over the binary field.

1.2 Organization of the Paper

A SNARK can be constructed from any linear error-correcting code that is asymptotically good, i.e., has constant relative distance [BCG+17, BCG20, BCL22, XZS22, GLS+23]; we summarize the framework we rely on [GLS+23] in Section 5. Thus, in the remainder of the paper, we focus on the analysis of the EA code itself. Following the preliminaries in Section 2, we present an improved distance analysis of the EA code over the binary field in Section 3. Then, in Section 4, we show how to generalize our techniques to any finite field. We discuss the concrete performance of our SNARK built from the EA code, and compare it with existing SNARKs based on linear codes, in Section 6.

2 Preliminaries

$\mathbb{F}_q$ denotes the finite field of size q (we do not require q to be prime), and $\mathbb{F}$ is a generic finite field. We write λ for the security parameter, and $\mathsf{negl}(\lambda)$ for a negligible function in λ. For positive integer N, we define $[N] := \{1, \dots, N\}$. We use boldface lowercase letters for vectors and boldface uppercase letters for matrices; vectors are row vectors by default. For a vector $\mathbf{v}$, we write $\mathsf{wt}(\mathbf{v})$ for the Hamming weight (i.e., number of nonzero entries) of $\mathbf{v}$. For $\mathbf{u}, \mathbf{v}$ of the same dimension we let $\Delta(\mathbf{u}, \mathbf{v}) = \mathsf{wt}(\mathbf{u} - \mathbf{v})$ be their Hamming distance.

Given $\mathbf{v} \in \mathbb{F}^n$, the *support of* $\mathbf{v}$ is the set $\mathsf{supp}(\mathbf{v}) := \{i \in [n] \colon \mathbf{v}_i \neq 0\}$. For $r \leq n$, we define $\mathsf{supps}(n, r) := \{\mathbf{v} \in \mathbb{F}^n \colon \mathsf{wt}(\mathbf{v}) = r\}$. For set $S \subseteq [n]$ such that $|S| = r$, we then define $\mathsf{supps}(n, r, S) := \{\mathbf{v} \in \mathbb{F}^n \colon \mathsf{supps}(\mathbf{v}) = S\}$ to be the set of all vectors $\mathbf{v}$ with support S. Note that by definition, we have: (1) $\mathsf{supps}(n, r, S) \subset \mathsf{supps}(n, r)$ for any $S \subseteq [n]$ of size r; (2) for distinct $S_1, S_2 \subseteq [n]$ of size r, $\mathsf{supps}(n, r, S_1) \cap \mathsf{supps}(n, r, S_2) = \emptyset$; and (3) $\mathsf{supps}(n, r) = \cup_S \mathsf{supps}(n, r, S)$, where the union is over all $S \subseteq [n]$ such that $|S| = r$.

2.1 Error-correcting Codes

A *linear error-correcting code* of block length N and dimension $n \leq N$ over a field $\mathbb{F}$ is an n-dimensional linear subspace $C \subseteq \mathbb{F}^N$. Such a code is defined by a rank-n encoding matrix $\mathbf{G} \in \mathbb{F}^{n \times N}$; the encoding of a message $\mathbf{x} \in \mathbb{F}^n$ is given by $\mathbf{x} \cdot \mathbf{G} \in \mathbb{F}^N$. The *rate* of the code is $R = n/N$. The *distance* of the code C is $\min_{\mathbf{u}, \mathbf{v} \in C,\, \mathbf{u} \neq \mathbf{v}} \Delta(\mathbf{u}, \mathbf{v})$, i.e., the minimum Hamming distance between any two codewords in C. When C is a linear code the minimum distance is equal to the minimum weight of the code $\min_{\mathbf{u} \in C, \mathbf{u} \neq \mathbf{0}} \mathsf{wt}(\mathbf{u})$. We let $\mathsf{d}(C) := \min_{\mathbf{u} \in C, \mathbf{u} \neq \mathbf{0}} \mathsf{wt}(\mathbf{u})/N$ denote the *relative distance* of C.

Expand-Accumulate Codes. Expand-Accumulate (EA) codes are a simple family of linear error-correcting codes with fast encoding times [BCG+22]. Throughout this paper, we let $\mathbf{A} \in \mathbb{F}^{N \times N}$ be the so-called "accumulator matrix" [DJM98], which is the upper-triangular matrix with all nonzero entries equal to 1. Thus, for $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{F}^N$ and $\mathbf{y} = (y_1, \dots, y_N) = \mathbf{x}\mathbf{A}$ we have $y_i = \sum_{j \leq i} x_j$. It is not hard to see that the following holds (a proof is provided in Appendix B.1):

Lemma 2.1. *For* $\mathbf{x} \in \mathbb{F}^N$ *with* $\mathsf{wt}(\mathbf{x}) = w$*, it holds that* $\mathsf{wt}(\mathbf{x}\mathbf{A}) \geq \lceil w/2 \rceil$.

The encoding matrix for an EA code is $\mathbf{E} \cdot \mathbf{A}$ where $\mathbf{E} \in \mathbb{F}^{n \times N}$ is an "expanding matrix" that is chosen to be sparse for fast encoding. There are various ways to choose $\mathbf{E}$; below, we highlight two variants:

- One way to choose $\mathbf{E}$ is by sampling each entry of $\mathbf{E}$ independently from the *generalized Bernoulli distribution* $\mathsf{Ber}_p(\mathbb{F})$ that samples a uniform non-zero element of $\mathbb{F}$ with probability p, and samples 0 with probability $1 - p$. Boyle et al. [BCG+22] introduced and analyzed the minimum distance of this variant of the EA code over the binary field.

- Another possibility is to sample each row of $\mathbf{E}$ uniformly from the set $\{\mathbf{x} \in \mathbb{F}^N : \mathsf{wt}(\mathbf{x}) = t\}$ for some parameter t; we denote this process by $\mathbf{E} \leftarrow \mathsf{BP}(n, N, t, \mathbb{F})$. (Note that if $t = p \cdot N$ then the sparsity matches the expected sparsity of $\mathbf{E}$ using the Bernoulli approach.) Boyle et al. [BCG+22] also considered this code over $\mathbb{F}_2$ and used it in their implementation; however, they left analyzing the minimum distance of this code as an open problem.

We analyze a version of the EA code that combines the above approaches. Specifically, we consider what we call the *juxtaposed EA code* where codewords are the concatenation of codewords from the EA codes above. That is, we consider the code $C[\mathbf{E}_1, \mathbf{E}_2] : \mathbb{F}^n \to \mathbb{F}^{2N}$ defined as $C[\mathbf{E}_1, \mathbf{E}_2](\mathbf{x}) \mapsto (\mathbf{x}\mathbf{E}_1\mathbf{A}) \| (\mathbf{x}\mathbf{E}_2\mathbf{A})$, where $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F})$ and $\mathbf{E}_2 \leftarrow \mathsf{Ber}_p(\mathbb{F})^{n \times N}$. Intuitively, we do this because of limitations of our analysis. In particular, we are only able to derive good bounds on the minimum distance of the first EA code for "low-weight" inputs (i.e., inputs $\mathbf{x}$ with $\mathsf{wt}(\mathbf{x}) = O(n/t)$); to bound the minimum distance of the juxtaposed code for all inputs, we handle "high-weight" inputs using the second EA code. However, we conjecture that $\mathbf{x}\mathbf{E}_1$ alone is an asymptotically good code (i.e., has constant rate and relative distance), and leave proving this as an open question.

3 The Juxtaposed EA Code Over Binary Fields

In this section, we bound the relative distance of our juxtaposed EA code over the binary field. Boyle et al. [BCG+22] previously analyzed a different version of the EA code, but our analysis differs from theirs and is intended primarily as a warm-up for the next section where we extend our analysis to general finite fields.

Boyle et al. [BCG+22] show that the EA code with $\mathbf{E} \leftarrow \mathsf{Ber}_p^{n \times N}(\mathbb{F}_2)$ has constant relative distance with high probability; their analysis uses a random walk on an expander graph. We take a different approach to analyzing the distance of our juxtaposed EA code. Our approach adopts the *input-output weight enumerator* (IOWE) technique that has been used to compute the distance of the so-called *Repeat-Accumulate* (RA) and *Repeat-Accumulate-Accumulate* (RAA) codes [DJM98, GM08]. To apply the IOWE technique to a linear code with rank-N encoding matrix $\mathbf{G}$, we calculate $G_{w,h}$—the number of input vectors of weight w mapped to codewords of weight h—for all w, h. For a uniform input vector $\mathbf{y}$ of weight w, we then have $\Pr_{\mathbf{y}}[\mathsf{wt}(\mathbf{y}\mathbf{G}) = h] = G_{w,h}/\binom{N}{w}$. By taking a Union Bound over all $h \leq \delta N$, we can bound the probability that $\mathsf{wt}(\mathbf{y} \cdot \mathbf{G})$ is less than or equal to δN—i.e., that it is a "bad codeword",—by $\sum_{h=1}^{\delta N} G_{w,h}/\binom{N}{w}$.

Taking the accumulation matrix $\mathbf{A}$ as the matrix $\mathbf{G}$ above, analyzing the minimum distance of any expand-accumulate code $\mathbf{EA}$ for relative distance parameter δ amounts to analyzing the following bound:

$$\begin{aligned}
&\Pr_{\mathbf{E}}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} : \mathsf{wt}(\mathbf{x}\mathbf{E}\mathbf{A}) \leq \delta N] \\
&\leq \sum_{r=1}^{n} \binom{n}{r} \sum_{w=1}^{N} \Pr_{\mathbf{E}}[\mathsf{wt}(\mathbf{x}\mathbf{E}) = w \mid \mathsf{wt}(\mathbf{x}) = r] \cdot \sum_{h=1}^{\delta N} \frac{A_{w,h}}{\binom{N}{w}},
\end{aligned}$$

where the distribution $\mathbf{x}\mathbf{E}$, when conditioned on $\mathsf{wt}(\mathbf{x}\mathbf{E}) = w$, is identical to the uniform distribution over vectors of weight w. This gives a way to analyze the distance of any EA code, if $\mathbf{E}$ is appropriately distributed.

Analyzing the distance of $C[\mathbf{E}_1, \mathbf{E}_2]$ amounts to applying the IOWE technique to analyze the minimum distance of each encoding matrix $\mathbf{E}_1\mathbf{A}$ and $\mathbf{E}_2\mathbf{A}$ individually. As noted previously, we are only able to prove good distance bounds for $\mathbf{E}_1\mathbf{A}$ for inputs of weight $O(n/t)$. Thus, to handle all possible input vectors, our distance analysis is broken into two cases. For nonzero inputs $\mathbf{x}$ of weight at most $O(n/t)$, we show that $\mathbf{x}\mathbf{E}_1\mathbf{A}$ has minimum weight δ with probability at least $1 - 1/\operatorname{poly}(N)$. For higher-weight inputs, we show that $\mathbf{x}\mathbf{E}_2\mathbf{A}$ has minimum weight δ with probability at least $1 - 1/\operatorname{poly}(N)$. Together, this implies that the code $C[\mathbf{E}_1, \mathbf{E}_2]$ has constant relative distance $\delta/2$ with probability at least $1 - 2/\operatorname{poly}(N)$. We summarize this result in the following theorem.

Theorem 3.1. *Let $R < 1$ be a constant. There exist constants $\gamma \geq 1$, $\delta \leq 1/(16e)^2$, and $c^* > 4$ such that for n sufficiently large, taking $N = n/R$, $t = \gamma \log N$, and $p = t/N$ we have*

$$\Pr_{\mathbf{E}_1, \mathbf{E}_2}[\mathsf{d}(C[\mathbf{E}_1, \mathbf{E}_2]) \leq \delta/2] \leq O(N^{4-c^*}),$$

where $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_2)$ *and* $\mathbf{E}_2 \leftarrow \mathsf{Ber}_p^{n\times N}(\mathbb{F}_2)$.

Proof Overview. In Section 3.1, we calculate the IOWE for the accumulator matrix $\mathbf{A}$. We then prove Theorem 3.1 by upper bounding the quantity

$$\Pr_{\mathbf{E}_1,\mathbf{E}_2}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(C(\mathbf{x})) \leq \delta N],$$

where $C(\mathbf{x}) \mapsto (\mathbf{x}\mathbf{E}_1\mathbf{A})\|(\mathbf{x}\mathbf{E}_2\mathbf{A})$. (Recall that codewords of C have length $2N$, so the above equation corresponds to relative distance $\delta/2$.) To bound the above, we separately bound

$$\Pr_{\mathbf{E}_1\leftarrow\mathsf{BP}(n,N,t,\mathbb{F}_2)}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N \wedge \mathsf{wt}(\mathbf{x}) \leq \beta n/t]$$

in Section 3.2 and

$$\Pr_{\mathbf{E}_2\leftarrow\mathsf{Ber}_p^{n\times N}(\mathbb{F}_2)}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_2\mathbf{A}) \leq \delta N \wedge \mathsf{wt}(\mathbf{x}) > \beta n/t]$$

in Section 3.3, for some constant β.

3.1 The IOWE of the Binary Accumulate Code

We calculate the IOWE for the binary accumulator matrix $\mathbf{A}$. Define

$$A_{w,h}^{N,q} := \left|\{\mathbf{x} \in \mathbb{F}_q^N : \mathsf{wt}(\mathbf{x}) = w \wedge \mathsf{wt}(\mathbf{x}\mathbf{A}) = h\}\right|.$$

For the binary field, Divsalar et al. [DJM98] provide the value of $A_{w,h}^{N,2}$ without proof; here, we provide a proof for completeness.

Theorem 3.2 ([DJM98])**.** *For any $N \in \mathbb{N}$ and $w, h \in [N]$, it holds that*

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

Proof. We start with a simple and well-known result.

Lemma 3.3 (Stars and Bars)**.** *The number of ways to place k indistinguishable items into b bins is $\binom{k+b-1}{b-1}$.*

Proof (Sketch). This can be seen by considering $k+b+1$ "slots", where the first and last slots contain a mark and each remaining slot either contains a mark or is blank. Placing $b-1$ marks in the $k+b-1$ intermediate slots corresponds to a placement of k items into b bins: the number of items in the ith bin ($1 \leq i \leq b$) is the number of (empty) slots between the ith and the $(i+1)$th marks. The lemma follows. □

Returning to the main theorem, let $\mathbf{x} = (x_1, \ldots, x_N)$ be a vector of weight w and let $\mathbf{y} = \mathbf{x}\mathbf{A}$. Assume w is even; we discuss how to adjust the calculation for odd w at the end. For $i = 1, \ldots, w/2$, define the ith *run* of $\mathbf{x}$ to be the indices from the $(2i-1)$th and $2i$th occurrence of a 1 in $\mathbf{x}$, and call each set of consecutive indices of $\mathbf{x}$ not contained in a run (including at the beginning and end of $\mathbf{x}$) a *non-run*; i.e., if $\mathbf{x} = (1, 0, 0, 0, 1, 0, 0, 1, 0, 1, \ldots)$ then

$$(\underbrace{\quad}_{\text{1st non-run}} \underbrace{1, 0, 0, 0, 1}_{\text{1st run}}, \underbrace{0, 0}_{\text{2nd non-run}}, \underbrace{1, 0, 1}_{\text{2nd run}}, \ldots).$$

(The 1st non-run is empty in this example.) Note that $\mathbf{x}$ has $w/2+1$ non-runs. The ones in $\mathbf{y}$ correspond exactly to the runs of $\mathbf{x}$; specifically, if there is a run in $\mathbf{x}$ from position i to position j, then there are ones in $\mathbf{y}$ from position i to position $j-1$. Thus, the number of ones in $\mathbf{y}$ corresponding to the ith run of $\mathbf{x}$ is one more than the number of indices inside (i.e., not including the indices that are 1 at the beginning and end of the run) the ith run of $\mathbf{x}$, and the total number of ones in $\mathbf{y}$ is equal to the total number of indices inside the runs of $\mathbf{x}$ plus the number of runs (which is $w/2$). In other words, $\mathbf{y}$ has Hamming weight h iff the total number of indices inside all the runs of $\mathbf{x}$ is exactly $h - w/2$.

Let $r = w/2$ be the number of runs in $\mathbf{x}$, and $r' = r + 1$ be the number of non-runs. Summarizing the above discussion, the number of vectors $\mathbf{x}$ of Hamming weight w that map to a vector $\mathbf{y}$ of Hamming weight h is the number of ways to construct a vector $\mathbf{x}$ of Hamming weight w having r runs, such that the total number of indices inside those runs is $h - r$, or in other words the number of vectors $\mathbf{x}$ with r runs such that there are $h - r$ indices inside the runs and $N - w - (h - r)$ indices in the non-runs. Applying Lemma 3.3, we see that the number of ways to place $h - r$ indices in the r runs is $\binom{h-1}{r-1}$. Using Lemma 3.3 again, the number of ways of distributing $N - w - h + r$ indices in the r' non-runs is $\binom{N-w-h+2r}{r'-1}$. Since $r = w/2$ and $r' = r + 1 = w/2 + 1$, this gives a total of

$$\binom{h-1}{r-1} \cdot \binom{N-w-h+2r}{r'-1} = \binom{h-1}{\frac{w}{2}-1} \cdot \binom{N-h}{\frac{w}{2}}$$

ways of choosing such an $\mathbf{x}$, matching the theorem when w is even.

When w is odd, there is one additional run (from the last 1 in $\mathbf{x}$ until the end of $\mathbf{x}$) and there is no longer a non-run at the end of $\mathbf{x}$. There are thus $r = \lceil w/2 \rceil$ runs and the same number $r' = r$ of non-runs, and so a total of

$$\binom{h-1}{r-1}\binom{N-w-h+2r}{r'-1} = \binom{h-1}{\lceil\frac{w}{2}\rceil-1}\binom{N-h}{\lceil\frac{w}{2}\rceil-1} = \binom{h-1}{\lceil\frac{w}{2}\rceil-1}\binom{N-h}{\lfloor\frac{w}{2}\rfloor}$$

ways of choosing $\mathbf{x}$ in that case. □

3.2 Distance Analysis for Low-weight Messages

Here, we bound

$$\Pr_{\mathbf{E}_1}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \le \delta N \wedge \mathsf{wt}(\mathbf{x}) \le \beta n/t]$$

for some constant β (which is a parameter of the matrix $\mathbf{E}_1$, discussed shortly). We first introduce the notion of an *expander*. This will be crucial to our analysis, as the matrix $\mathbf{E}_1$ is sampled such that it is a good expander with high probability.

Definition 3.4 (Expander)**.** *Let $G = (L, R, E)$ be a bipartite graph, with all edges in E being between a vertex in the left vertex set L and a vertex in the right vertex set R. Let $\Gamma(S)$ be the neighbor set of vertex set S. We say G is an $(n; t, \alpha, \beta, \epsilon)$-**expander** if $|L| = n, |R| = N = \alpha n$, and the following hold:*

1. *Degree: The degree of every vertex in L is t.*
2. *Expansion: $|\Gamma(S)| \ge (1-\epsilon)t|S|$ for every $S \subseteq L$ with $|S| \le \frac{\beta|L|}{t}$.*

As the degree of each left vertex is t, a set of $|S|$ left vertices can have at most $t \cdot |S|$ neighbors; the second condition requires that every small set of left vertices has almost that many neighbors. Since the right vertex set has $|R| = \alpha n$ vertices, $|\Gamma(S)| \ge (1-\epsilon)t|S|$ is not possible if $|S| > \frac{\alpha n}{(1-\epsilon)t}$; thus, expansion is only required for small subsets of L.

A random bipartite graph is an expander with high probability [HLW06, XZS22]. A proof of the following is analogous to the proof of [XZS22, Lemma 2] with appropriate adjustment of the parameters.

Lemma 3.5 ([HLW06, XZS22])**.** *There exist constants α, β, ϵ and $t = \Theta(\log \alpha n)$, such that a random t-left-regular bipartite graph with n left vertices and $N = \alpha n$ right vertices is an $(n; t, \alpha, \beta, \epsilon)$-expander except with probability at most* $\mathsf{negl}(N)$.

Proof. Let $G = (L, R, E)$ be a bipartite graph with n vertices on the left and $N = \alpha n$ vertices on the right, where each left vertex connects to a randomly chosen set of t vertices on the right. Let $s = |S|$ be the cardinality of a left subset of vertices $S \subseteq L$ such that $s \le \frac{\beta n}{t}$, and let $u = |U|$ be the cardinality of a right subset of vertices $U \subseteq R$ such that $u \leq (1-\epsilon)ts$. Let $X_{S,U}$ be an indicator random variable for the event that all the edges from S connect to U. Then, for a particular S, if $\sum_{U\in R} X_{S,U} = 0$, the number of neighboring vertices of S must be larger than $(1-\epsilon)ts$. Otherwise, if there exists a $U \in R$ such that $X_{S,U} = 1$, i.e., all edges from S connect to U, then the graph is not an expander. As the edges are sampled randomly, the probability of this non-expanding event is $(\frac{u}{N})^{st}$. Therefore, summing over all S and by the Union Bound, the probability of a non-expanding graph is:

$$\begin{aligned}
\Pr[\sum_{S,U} X_{S,U} > 0] &\leq \sum_{S,U} \Pr[X_{S,U} = 1] = \sum_{S,U} \left(\frac{u}{N}\right)^{st} \\
&\leq \sum_{s=2}^{\frac{\beta n}{t}} \binom{n}{s}\binom{N}{u}\left(\frac{u}{N}\right)^{st} \\
&\leq \sum_{s=2}^{\frac{\beta n}{t}} \binom{n}{s}\binom{N}{(1-\epsilon)ts}\left(\frac{(1-\epsilon)ts}{N}\right)^{st}
\end{aligned}$$

Using the inequality $\binom{n}{s} \leq \left(\frac{ne}{s}\right)^s$, the probability above is:

$$\leq \sum_{s=2}^{\frac{\beta n}{t}} \left(\frac{ne}{s}\right)^s \left(\frac{Ne}{(1-\epsilon)ts}\right)^{(1-\epsilon)ts} \left(\frac{(1-\epsilon)ts}{N}\right)^{st} = \sum_{s=2}^{\frac{\beta n}{t}} e^{(1-\epsilon)ts+s}\left(\frac{n}{s}\right)^s \left(\frac{(1-\epsilon)ts}{N}\right)^{\epsilon st}.$$

Recall that α, β, ϵ are constants and that $t = \Theta(\log(N))$. For simplicity, assume that $t = \gamma\ln(N)$ (this only changes some constants in the analysis). Then we have:

$$\begin{aligned}
e^{(1-\epsilon)ts+s}\left(\frac{n}{s}\right)^s\left(\frac{(1-\epsilon)ts}{N}\right)^{\epsilon st} &= \left(\frac{e^{(1-\epsilon)t+1}N}{\alpha s}\left(\frac{s}{N}\right)^{\epsilon t}((1-\epsilon)t)^{\epsilon t}\right)^s \\
\overset{2\leq s\leq \frac{\beta n}{t}}{\leq} \left(\frac{e^{(1-\epsilon)t+1}N}{2\alpha}\left(\frac{\frac{\beta n}{t}}{N}\right)^{\epsilon t}((1-\epsilon)t)^{\epsilon t}\right)^s &= \left(\frac{e^{(1-\epsilon)t+1}N}{2\alpha}\left(\frac{\beta}{\alpha t}\right)^{\epsilon t}((1-\epsilon)t)^{\epsilon t}\right)^s
\end{aligned}$$

$$\begin{aligned}
&\overset{t=\gamma\log(N)}{=} \left(\frac{e^{(1-\epsilon)\gamma\log(N)+1}N}{2\alpha}\left(\frac{\beta}{\alpha\gamma\log(N)}\right)^{\epsilon\gamma\log(N)}((1-\epsilon)\gamma\log(N))^{\epsilon\gamma\log(N)}\right)^s \\
&= \left(\frac{e}{2\alpha}\cdot N^{1+\gamma+1-\epsilon\gamma(\log(\log(N^{\frac{\alpha\gamma}{\beta}})-\log(\log(N^{(1-\epsilon)\gamma})))}\right)^s
\end{aligned}$$

Since ϵ and γ are constants, it remains to show that:

$$\begin{aligned}
&0 < \log(\log(N^{\frac{\alpha\gamma}{\beta}}) - \log(\log(N^{(1-\epsilon)\gamma})) \\
\iff &0 < \log\left(\frac{\frac{\alpha\gamma}{\beta}\log(N)}{(1-\epsilon)\gamma\log(N)}\right) \\
\iff &1 < \frac{\alpha}{(1-\epsilon)\beta} \iff (1-\epsilon) < \frac{\alpha}{\beta}.
\end{aligned}$$

The last inequality holds since G is an expander graph (Definition 3.4). Concluding this analysis, we obtain an overall bound for the described event to occur of $\mathsf{negl}(N)$, which implies the probability $1-\mathsf{negl}(N)$ stated in the theorem. □

Note that sampling $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_2)$ is equivalent to sampling (the adjacency matrix of) a random t-left-regular bipartite graph, and thus defines an expander with high probability. Referring to $\mathbf{E}$ as an expander when the bipartite graph it defines is an expander, we have the following lemma.

Lemma 3.6. *Let $\alpha, \beta, \epsilon, \gamma$ be constants. If $\mathbf{E}$ is a $(n; t, \alpha, \beta, \epsilon)$-expander, then for $\mathbf{x} \in \{0,1\}^n$ with $r := \mathsf{wt}(\mathbf{x}) \leq \frac{\beta n}{t}$, we have $\mathsf{wt}(\mathbf{xE}) \geq (1-2\epsilon)t \cdot r$. Specifically, if $t = \gamma \log N$ then $\mathsf{wt}(\mathbf{xE}) = \Omega(w \log N)$.*

Proof. For set $S \subseteq L$, define the *unique neighbors* of S as

$$U(S) = \{r \in \Gamma(S) : |\Gamma(\{r\}) \cap S| = 1)\}.$$

That is, the unique neighbors of $S \subseteq L$ are those vertices that have an edge with only one vertex in S. Fix $S \subseteq L$ with $|S| \leq \frac{\beta|L|}{t}$. Since $\Gamma(S) \geq (1-\epsilon)t|S|$ and $t|S| \geq U(S) + 2(\Gamma(S) - U(S))$, it follows that $U(S) \geq (1-2\epsilon)t|S|$, which gives us $U(S) \geq (1-2\epsilon)t\frac{\beta|L|}{t}$. Taking into account the above observation that sampling $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_2)$ is equivalent to sampling the adjacency matrix of a random t-left-regular bipartite graph, it follows that $\mathsf{wt}(\mathbf{xE}) \geq (1-2\epsilon)t \cdot \frac{\beta|L|}{t} \geq (1-2\epsilon)t \cdot \frac{\beta n}{t} = (1-2\epsilon)t \cdot r$. If $t = \gamma \log(N)$, then this implies $\mathsf{wt}(\mathbf{xE}) = \Omega(w \log N)$ which concludes the proof of the lemma. $\square$

Let $\eta > 2^{\frac{4}{(1-2\epsilon)\gamma}}$ be a constant and set

$$\delta := 1/(\eta e)^2 < 1/(2^{\frac{4}{(1-2\epsilon)\gamma}} e)^2. \tag{1}$$

We need to bound

$$\begin{aligned}
&\Pr_{\mathbf{E}_1}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N \wedge \mathsf{wt}(\mathbf{x}) \leq \beta n/t] \\
&\leq \Pr_{\mathbf{E}_1}[\mathbf{E}_1 \text{ is not an expander}] \\
&+ \sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=1}^{N} \Pr_{\mathbf{E}_1}[\mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w \mid \mathsf{wt}(\mathbf{x}) = r \wedge \mathbf{E}_1 \text{ is an expander}] \sum_{h=1}^{\delta N} \frac{A_{w,h}^{N,2}}{\binom{N}{w}},
\end{aligned}$$

where $A_{w,h}^{N,2}$ is the IOWE of $\mathbf{A}$. By Lemma 3.5, the first term is negligible, so we focus on the second term. Let $c = (1-2\epsilon)$, since $r \leq \beta n/t$ and $t = \gamma \log N$, Lemma 3.6 implies

$$\begin{aligned}
\sum_{w=1}^{N} \Pr_{\mathbf{E}_1}[\mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w \mid \mathsf{wt}(\mathbf{x}) = r \wedge \mathbf{E}_1 \text{ is an expander}] \sum_{h=1}^{\delta N} \frac{A_{w,h}^{N,2}}{\binom{N}{w}} & \\
\leq \sum_{w=ct\cdot r}^{N} \sum_{h=1}^{\delta N} \frac{A_{w,h}^{N,2}}{\binom{N}{w}} & \\
= \sum_{w=ct\cdot r}^{2\delta N} \sum_{h=\lceil w/2 \rceil}^{\delta N} \frac{A_{w,h}^{N,2}}{\binom{N}{w}} &
\end{aligned}$$

using Lemma 2.1 to truncate the summations in the final equality. Applying Theorem 3.2, this means we need to bound

$$\sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=ct\cdot r}^{2\delta N} \sum_{h=\lceil w/2 \rceil}^{\delta N} \frac{\binom{N-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{N}{w}}.$$

Suppose $w = 2k$ is even, Stirling's approximation $(n/k)^k \leq \binom{n}{k} \leq (en/k)^k$ gives

$$\sum_{h=\lceil w/2 \rceil}^{\delta N} \frac{\binom{N-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{N}{w}} \leq \left(\frac{2k}{N}\right)^{2k} \frac{e^{2k}}{k^k(k-1)^k} \sum_{h=k}^{\delta N} ((N-h)(h-1))^k.$$

For $\delta < 1/2$ (which is the case in Theorem 3.1), the term $(N-h)(h-1)$ is maximized at $h = \delta N$. To show that this is true we can show that $(N-h)(h-1) < (N-(h+1))((h+1)-1)$ for all $h < \frac{1}{2}N$. The inequality can be rewritten as $h(N-h+1)-N < h(N-h-1) \iff N-h+1-N/h < N-h-1 \iff 0 < N-h$ which holds for $h < \frac{1}{2}N$. Thus,

$$\left(\frac{2k}{N}\right)^{2k} \frac{e^{2k}}{k^k(k-1)^k} \sum_{h=k}^{\delta N} ((N-h)(h-1))^k \leq (4e\delta(1-\delta))^k 4\delta N.$$

For $w = 2k+1$ odd, a similar derivation gives the upper bound $(4e^2\delta(1-\delta))^k \cdot e^2\delta N$. Together, this means

$$\sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=c\cdot r}^{2\delta N} \sum_{h=\lceil w/2 \rceil}^{\delta N} \frac{\binom{N-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{N}{w}}$$
$$\leq \sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=c\cdot r}^{2\delta N} (4e^2\delta(1-\delta))^{\lfloor w/2 \rfloor} \cdot e^2\delta N.$$

Our choice of δ ensures that $4e^2\delta(1-\delta) < 1$, so the above summand is a decreasing function in w. Thus,

$$\sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=cr}^{2\delta N} (4e^2\delta(1-\delta))^{\lfloor w/2 \rfloor} \cdot e^2\delta N$$
$$\leq \sum_{r=1}^{\beta n/t} \left(\frac{en}{r}\right)^r \cdot (2\delta N) \cdot (4e^2\delta(1-\delta))^{cr/2} \cdot e^2\delta N$$
$$= \sum_{r=1}^{\beta n/t} \left(\frac{eRN}{r} \cdot (4e^2\delta(1-\delta))^{c/2}\right)^r \cdot 2e^2(\delta N)^2.$$

For ease of presentation, let $g = 2\sqrt{\delta(1-\delta)}$. Then the above summand becomes

$$\left(\frac{eRN}{r} \cdot (4e^2\delta(1-\delta))^{c/2}\right)^r \cdot 2e^2(\delta N)^2 = \left(\frac{eRN}{r} \cdot (ge)^{(1-2\epsilon)\cdot t}\right)^r \cdot 2e^2(\delta N)^2.$$

Recall that $t = \gamma \log N$, therefore,

$$(ge)^{(1-2\epsilon)t} = (ge)^{(1-2\epsilon)\gamma \log N} = \left(2^{\log(ge)}\right)^{(1-2\epsilon)\gamma \log N} = N^{(1-2\epsilon)\gamma \log(ge)}$$

and

$$\left(\frac{eRN}{r} \cdot (ge)^c\right)^r \cdot 2e^2(\delta N)^2 = \left(\frac{eRN^{1+(1-2\epsilon)\gamma \log(ge)}}{r}\right)^r \cdot 2e^2(\delta N)^2.$$

For sufficiently large N, our choice of δ ensures the above decreases in r, and the maximum is obtained at $r = 1$. This results in

$$\sum_{r=1}^{\beta n/t} \left(\frac{eRN(4e^2\delta(1-\delta))^{c/2}}{r}\right)^r 2e^2(\delta N)^2$$
$$\leq [(2\beta R^2 e^3)/(\gamma \log(N))] \cdot \delta^2 N^{4+(1-2\epsilon)\gamma \log(ge)}.$$

Our choice of δ again implies

$$(2\beta R^2 e^3)/(\gamma \log(N)) \cdot \delta^2 N^{4+(1-2\epsilon)\gamma \log(ge)}$$
$$\leq \frac{2\beta R^2}{\eta^4 e \gamma \log(N)} \cdot N^{4-(1-2\epsilon)\gamma \log(\eta^2 e/2)} = o(N^{4-c^*}),$$

where $c^* = (1-2\epsilon)\gamma\log(\eta^2 e/2)$ is a constant. Summarizing:

$$\begin{aligned}\Pr_{\mathbf{E}_1}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \le \delta N \wedge \mathsf{wt}(\mathbf{x}) \le & \beta n/t] \\ & \le o(N^{4-c^*}) + \mathsf{negl}(N) \\ & = o(N^{4-c^*}).\end{aligned}$$

This concludes the proof for the case of low-weight messages.

3.3 Distance Analysis for High-weight Messages

In this section, we turn towards upper bounding

$$\Pr_{\mathbf{E}_2 \leftarrow \mathsf{Ber}_p^{n\times N}(\mathbb{F}_2)}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_2\mathbf{A}) \le \delta N \wedge \mathsf{wt}(\mathbf{x}) > \beta n/t].$$

To leverage our analysis from the preceding section, we show that $\mathbf{E}_2$ has good expansion for inputs of weight $r > \beta n/t$. Specifically, we have the following lemma:

Lemma 3.7. *Let $\epsilon \in (0,1/2)$, $R < 1$, $\beta \le (1/R)(1-\epsilon)$, and $\gamma > 1/(1-2\epsilon)$ be constants. Let $n \in \mathbb{N}$ and $N = n/R$. For any $\mathbf{x} \in \mathbb{F}_q^n$ such that $r = \mathsf{wt}(\mathbf{x}) \ge \beta n/t$, we have*

$$\Pr_{\mathbf{E}_2 \leftarrow \mathsf{Ber}_p^{n\times N}(\mathbb{F}_q)}[\mathsf{wt}(\mathbf{x}\mathbf{E}_2) < r(1-2\epsilon)\gamma\log N] \le \mathsf{negl}(N).$$

Note that the above holds for any finite field $\mathbb{F}_q$. The proof is deferred to Appendix B.4.

Focusing on the binary case and letting δ, c, c^* be as in the previous section, Lemma 3.7 implies that

$$\begin{aligned}&\Pr_{\mathbf{E}_2}[\exists \mathbf{x} \in \{0,1\}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_2\mathbf{A}) \le \delta N \wedge \mathsf{wt}(\mathbf{x}) \ge \beta n/t] \\ &\le \mathsf{negl}(N) + \sum_{r=\beta n/t}^{n} \binom{n}{r} \sum_{w=ctr}^{2\delta N} \sum_{h=\lceil w/2\rceil}^{\delta N} \frac{\binom{N-h}{\lfloor \frac{w}{2}\rfloor}\binom{h-1}{\lceil \frac{w}{2}\rceil - 1}}{\binom{N}{w}}.\end{aligned}$$

The above is nearly identical to the sum analyzed in Section 3.2, except that we now sum over $r \ge \beta n/t$. Following our prior analysis, we obtain

$$\begin{aligned}&\sum_{r=\beta n/t}^{n} \binom{n}{r} \sum_{w=ctr}^{2\delta N} \sum_{h=\lceil w/2\rceil}^{\delta N} \frac{\binom{N-h}{\lfloor \frac{w}{2}\rfloor}\binom{h-1}{\lceil \frac{w}{2}\rceil - 1}}{\binom{N}{w}} \\ &\le \sum_{r=\beta n/t}^{n} \left(\frac{eRN^{1+(1-2\epsilon)\gamma\log(ge)}}{r}\right)^r 2e^2(\delta N)^2,\end{aligned}$$

where $g = 2\sqrt{\delta(1-\delta)}$ as in the previous section. As before, for our choice of δ the summand above is a decreasing function in r. Taking $r = 1$, we obtain the nearly identical bound:

$$\begin{aligned}&\sum_{r=\beta n/t}^{n} \left(\frac{eRN^{1+(1-2\epsilon)\gamma\log(ge)}}{r}\right)^r 2e^2(\delta N)^2 \\ &\le (n - (\beta n/t))eRN^{1+(1-2\epsilon)\gamma\log(ge)}2e^2(\delta N)^2 \\ &= O(N^{4-c^*}).\end{aligned}$$

The results of this section and the previous one complete the proof of Theorem 3.1.

4 The Juxtaposed EA Code Over Arbitrary Finite Fields

We now turn to our main contribution: generalizing the EA code to an arbitrary finite field $\mathbb{F} = \mathbb{F}_q$. As with the binary case, we prove a bound on the (relative) minimum distance of our EA code using the IOWE technique. As a first step, we compute $A_{w,h}^{N,q}$, the IOWE of $\mathbf{A}$ over an arbitrary finite field of size q (see Theorem 4.2); to the best of our knowledge, this has not been done before. We then follow the IOWE technique (Section 3) to bound the minimum distance of the juxtaposed EA code over $\mathbb{F}_q$, giving us the following theorem.

Theorem 4.1. *Let $R = n/N$ be a constant, then there exist constants $\gamma \geq 1$, $\delta \leq 1/(5 \cdot 2^5 \cdot e)^2$, and $c^* > 5$ such that for $t = \gamma \log N$, $p = t/N$, we have*

$$\Pr_{\mathbf{E}_1, \mathbf{E}_2}[\mathsf{d}(C[\mathbf{E}_1, \mathbf{E}_2]) \geq \delta/2] \geq 1 - O(N^{5-c^*}),$$

where $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_q)$ and $\mathbf{E}_2 \leftarrow \mathsf{Ber}_p^{n \times N}(\mathbb{F}_q)$.

In Section 4.1, we state and prove the IOWE of the accumulator matrix over $\mathbb{F}_q$. In Section 4.2, we give a warm-up discussion towards proving Theorem 4.1 by describing a natural, but failed, attempt at proving our theorem, along with how we overcome the issues that arise in this natural attempt. We then prove Theorem 4.1 in two cases, as with the binary case: in Section 4.3, we analyze low-weight input messages; in Section 4.4, we analyze high-weight input messages.

4.1 IOWE of the Accumulate Code for Arbitrary Finite Fields

We will begin by proving an analogous theorem about the IOWE for the accumulate code $\mathbf{A}$ when we assume $\mathbf{A} \in \{0,1\}^{N \times N} \subseteq \mathbb{F}_q^{N \times N}$. To the best of our knowledge, this result has not been shown before and may be of independent interest (e.g., extending binary codes that use $\mathbf{A}$ to large finite fields).

Theorem 4.2. *For a finite field $\mathbb{F}_q$, $N \in \mathbb{N}$, and $w, h \in [N]$, we have*

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

For $q = 2$, this exactly captures Theorem 3.2 with the convention that $0^0 = 1$.

Proof. For a vector $\mathbf{x} = (x_1, \ldots, x_N)$, let $\mathbf{y} = \mathbf{x} \cdot \mathbf{A}$; i.e., $\mathbf{y} = (y_1, \ldots, y_N)$ with $y_i = \sum_{j=1}^{i} x_j$. A *run* in $\mathbf{x}$ is now defined to be a consecutive sequence of indices $i, i+1, \ldots, j$ such that $y_{i-1} = 0$, $y_i, \ldots, y_{j-1} \neq 0$, and $y_j = 0$ (where we set $y_0 = y_{N+1} = 0$); note that this agrees with how a run was defined in the proof of Theorem 3.2 for the case $q = 2$. A non-run is a consecutive sequence of indices that are not part of a run. For a run $i, \ldots, j$ with $j \leq N$, we say x_i *begins* the run and x_j *ends* it. (Note that a run that starts at position i and continues until position N does not have an entry that ends it.) Observe that only a non-zero entry of $\mathbf{x}$ can start or end a run, and if x_j ends the run $i, \ldots, j$ then $x_j = -(x_i + \cdots + x_{j-1})$.

For a vector $\mathbf{x}$ of weight w, let i be the number of non-zero entries of $\mathbf{x}$ that do *not* start or end a run. The number of runs in $\mathbf{x}$ is then $r = \lceil \frac{w-i}{2} \rceil$, and the number of non-runs is $r' = \lfloor \frac{w-i}{2} \rfloor + 1$. As in Theorem 3.2, the number of ways of arranging the locations of the runs is

$$\binom{h-1}{r-1} \cdot \binom{N-h}{r'-1}.$$

Now, however, there are additional possibilities for $\mathbf{x}$:

- Each of the r runs can start with any of the $q-1$ non-zero elements of $\mathbb{F}$.
- As in Theorem 3.2, the total number of entries in the middle of all the runs is exactly $h - r$; thus, there are $\binom{h-r}{i}$ positions in which the i non-zero elements of $\mathbf{x}$ that are in the middle of a run can be placed.

- Each of the i non-zero elements of $\mathbf{x}$ in the middle of a run can take one of $q-2$ possible values—it cannot be 0, nor can it be a non-zero value that would end the run.

- The value of each nonzero element that ends a run is equal to the sum of the previous elements in that run (and thus is fixed).

Overall, the number of vectors $\mathbf{x}$ of weight w that map to a vector of weight h and have exactly i non-zero entries in the middle of a run is

$$\begin{aligned}&\binom{h-1}{r-1}\cdot\binom{N-h}{r'-1}\cdot(q-1)^r\cdot\binom{h-r}{i}\cdot(q-2)^i\\ =\ &\binom{h-1}{\lceil\frac{w-i}{2}\rceil-1}\binom{N-h}{\lfloor\frac{w-i}{2}\rfloor}\binom{h-\lceil\frac{w-i}{2}\rceil}{i}(q-1)^{\lceil\frac{w-i}{2}\rceil}(q-2)^i.\end{aligned}$$

Since $h>0$, there must be at least one run, which must be started by some nonzero entry of $\mathbf{x}$; thus, $i<w$. Taking the sum of the above equation over $0\le i<w$ gives the theorem. $\square$

4.2 Warm-up: A (Failed) Approach for Proving Theorem 4.1

Give the IOWE of the accumulator in Theorem 4.2, we turn towards a natural, but failed, application of the Union Bound to prove Theorem 4.1. Let $\mathbb{F}:=\mathbb{F}_q$ be the finite field we are considering, let $\beta,\gamma,R,\epsilon,\delta$ be appropriately defined constants, and let $n\in\mathbb{N}$, $N:=n/R$, and $t=\gamma\log(N)$. Also let $\mathbf{E}\in\mathbb{F}^{n\times N}$ be a randomly sampled expansion matrix (either $\mathbf{E}_1$ or $\mathbf{E}_2$), and $\mathbf{A}\in\{0,1\}^{N\times N}\subseteq\mathbb{F}^{N\times N}$ be the accumulator matrix. To analyze the distance of the code, we are interested in upper bounding the following probability:

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

In particular, analyzing the above bound with respect to both $\mathbf{E}_1$ and $\mathbf{E}_2$ is sufficient to prove Theorem 4.1; i.e., if the above probability is at most $O(N^{5-c^*})$ for both $\mathbf{xE}_1\mathbf{A}$ and $\mathbf{xE}_2\mathbf{A}$, then our theorem follows.

To analyze the above probability one may try to apply the Union Bound and obtain

$$\begin{aligned}&\Pr_{\mathbf{E}}[\exists x\in\mathbb{F}^n\setminus\{0^n\}\colon \mathsf{wt}(x\mathbf{EA})\le\delta N]\\ &\le\sum_{r=1}^{n}\binom{n}{r}(q-1)^r\cdot\Bigg(\sum_{w=1}^{N}\Pr_{\mathbf{E}}[\mathsf{wt}(\mathbf{xE})=w\mid\mathsf{wt}(\mathbf{x})=r]\\ &\qquad\qquad\times\sum_{h=1}^{\delta N}\Pr_{\mathbf{E}}[\mathsf{wt}(\mathbf{xEA})=h\mid\mathsf{wt}(\mathbf{x})=r\wedge\mathsf{wt}(\mathbf{xE})=w]\Bigg)\\ &=\sum_{r=1}^{n}\binom{n}{r}(q-1)^r\cdot\Bigg(\sum_{w=1}^{N}\Pr_{\mathbf{E}}[\mathsf{wt}(\mathbf{xE})=w\mid\mathsf{wt}(\mathbf{x})=r]\cdot\sum_{h=1}^{\delta N}\frac{A^{N,q}_{w,h}}{\binom{N}{w}(q-1)^w}\Bigg).\end{aligned}$$

Pushing everything into the innermost summation, this summand becomes

$$\frac{\binom{n}{r}}{\binom{N}{w}}\frac{A^{N,q}_{w,h}}{(q-1)^{w-r}}=\frac{\binom{n}{r}}{\binom{N}{w}}\frac{\sum_{i=0}^{w-1}\binom{N-h}{\lfloor\frac{w-i}{2}\rfloor}\binom{h-1}{\lceil\frac{w-i}{2}\rceil-1}\binom{h-\lceil\frac{w-i}{2}\rceil}{i}(q-1)^{\lceil\frac{w-i}{2}\rceil}(q-2)^i}{(q-1)^{w-r}}.$$

Unfortunately, the term $(q-1)^{\lceil\frac{w-i}{2}\rceil}(q-2)^i/(q-1)^{w-r}$ becomes problematic for $q\ge 4$; in many cases it becomes exponential in q (rather than $1/q$) for certain settings of i and r. It is also not clear how the other binomial terms can counteract this exponential growth; moreover the probability outside this term $(\Pr[\mathsf{wt}(\mathbf{xE})=w|\mathsf{wt}(\mathbf{x})=r])$ may or may not be able to handle this growth in q. We were not able to show that it can handle this growth of q. Therefore, we cannot naively apply the Union Bound directly.

Step-by-step Union Bound Refinement. Our technique essentially boils down to appropriately defining unions of probability events that are equivalent to the probability we are interested in analyzing, applying the Union Bound in stages, then finally applying a truncated Union Bound ($\Pr[\cup_i E_i] \leq \max\{1, \sum_i \Pr[E_i]\}$) to avoid the previously described problems.

For $\mathbf{x} \in \mathbb{F}^n$, let $\mathcal{E}_{\mathbf{x}}$ denote the event that $\mathsf{wt}(\mathbf{xEA}) \leq \delta N$. Then our desired probability is given exactly by

$$\Pr_{\mathbf{E}}\left[\exists \mathbf{x} \in \mathbb{F}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{xEA}) \leq \delta N\right] = \Pr_{\mathbf{E}}\left[\cup_{\mathbf{x} \in \mathbb{F}^n \setminus \{0^n\}} \mathcal{E}_{\mathbf{x}}\right]. \tag{3}$$

We now partition $\cup_{\mathbf{x}} \mathcal{E}_{\mathbf{x}}$. Recall from Section 2 that $\mathsf{supps}(n, r)$ is the set of r-weight vectors of $\mathbb{F}^n$. Now let $\mathcal{S}_r^{(n)}$ denote the event "$\exists \mathbf{x} \in \mathsf{supps}(n, r) \colon \mathsf{wt}(\mathbf{xEA}) \leq \delta N$". Clearly, by definition we have $\cup_{r \in [n]} \mathcal{S}_r^{(n)} = \cup_{\mathbf{x} \in \mathbb{F}^n \setminus \{0^n\}} \mathcal{E}_{\mathbf{x}}$, and thus

$$\Pr_{\mathbf{E}}\left[\exists \mathbf{x} \in \mathbb{F}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{xEA}) \leq \delta N\right] = \Pr_{\mathbf{E}}\left[\cup_{r \in [n]} \mathcal{S}_r^{(n)}\right].$$

Next, we further partition each event $\mathcal{S}_r^{(n)}$. Fix $r \in [n]$ and let $T \subseteq [n]$ be a set such that $|T| = r$. Again from Section 2, recall that $\mathsf{supps}(n, r, T)$ is the set of all vectors $\mathbf{v} \in \mathbb{F}$ with support T. With this, let $\mathcal{C}(n, r, T)$ denote the event "$\exists \mathbf{x} \in \mathsf{supps}(n, r, T) \colon \mathsf{wt}(\mathbf{xEA}) \leq \delta N$". By definition of $\mathcal{C}(n, r, T)$, we have $\mathcal{S}_r^{(n)} = \cup_T \mathcal{C}(n, r, T)$ for any $r \in [n]$ and all $T \subseteq [n]$ such that $|T| = r$. This gives us the following equalities:

$$\cup_{\mathbf{x} \in \mathbb{F}^n \setminus \{0^n\}} \mathcal{E}_{\mathbf{x}} = \cup_{r \in [n]} \mathcal{S}_r^{(n)} = \cup_{r \in [n]} \cup_{T \subseteq [n], |T| = r} \mathcal{C}(n, r, T).$$

By the above equalities and the Union Bound, our original probability becomes

$$\begin{aligned}
\Pr_{\mathbf{E}}[\cup_{\mathbf{x}} \mathcal{E}_{\mathbf{x}}] = \Pr_{\mathbf{E}}[\cup_r \cup_T \mathcal{C}(n, r, T)] \leq \sum_{r=1}^{n} \Pr_{\mathbf{E}}\left[\cup_T \mathcal{C}(n, r, T)\right] &\leq \sum_{r=1}^{n} \sum_{T \subseteq [n], |T| = r} \Pr_{\mathbf{E}}[\mathcal{C}(n, r, T)] \\
&\leq \sum_{r=1}^{n} \binom{n}{r} \Pr_{\mathbf{E}}[\mathcal{C}(n, r, T^*)],
\end{aligned} \tag{4}$$

where $T^* \subseteq [n]$ (of size r) satisfies $\Pr_{\mathbf{E}}[\mathcal{C}(n, r, T)] \leq \Pr_{\mathbf{E}}[\mathcal{C}(n, r, T^*)]$ for all $T \subseteq [n]$ (of size r).

4.3 Distance Analysis for Low-weight Messages

We now proceed to analyze Eq. (4) in two parts. In this section, we analyze the case of low-weight input messages. This corresponds to analyzing the matrix $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_q)$. Note that, in this case, sampling $\mathbf{E}_1$ uniformly from this set is identical to sampling a t-left regular bipartite graph with edge weights sampled uniformly from $\mathbb{F}_q^{\times}$. In particular, Lemmas 3.5 and 3.6 also hold for $\mathbf{E}_1$ sampled in this way. For this case, we analyze the following truncated summation of Eq. (4):

$$\sum_{r=1}^{\beta n/t} \binom{n}{r} \Pr_{\mathbf{E}_1}[\mathcal{C}(n, r, T^*)] = \sum_{r=1}^{\beta n/t} \binom{n}{r} \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x} \in \mathsf{supps}(n, r, T^*)} \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N\right].$$

By Lemma 3.6, any $\mathbf{x} \in \mathbb{F}^n$ of weight at most $\beta n/t$ will expand to a vector of weight at least $r \cdot ct$, where $c = (1 - 2\epsilon)$ is a constant and ϵ is a parameter of the expander $\mathbf{E}_1$. Using the Union Bound with intersection, we can rewrite the right hand side above as

$$\begin{aligned}
&\Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x} \in \mathsf{supps}(n, r, T^*)} \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N\right] \\
&= \Pr_{\mathbf{E}_1}[\mathbf{E}_1 \text{ is not an expander}] && (5) \\
&+ \sum_{w=1}^{N} \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}} (\mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w \wedge \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N) \wedge \mathbf{E}_1 \text{ is an expander}\right]. && (6)
\end{aligned}$$

By Lemma 3.5, the matrix $\mathbf{E}_1$ is not an expander with $\mathsf{negl}(N)$ probability, so Eq. (5) is upper bounded by $\mathsf{negl}(N)$. Therefore, we can focus on bounding Eq. (6). By conditioning on the event that $\mathbf{E}_1$ is an expander and bounding the probability $\mathbf{E}_1$ is an expander by 1, we can upper bound Eq. (6) as

$$\begin{aligned}\text{Eq. (6)} &\leq \sum_{w=rct}^{N} \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(\mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w)\right] \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}\mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w\right] \\ &\leq \sum_{w=rct}^{2\delta N} \sum_{h=\lceil w/2 \rceil}^{\delta N} \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(\mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) = h \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w)\right].\end{aligned}$$

To proceed with the analysis, consider how a vector $\mathbf{y} \in \mathsf{supps}(N, w)$ can map to a vector $\mathbf{z} = \mathbf{y}\mathbf{A}$ of weight h. Intuitively, this consists of two steps: (1) $\mathbf{y}$ has a "bad" setting of non-zeros (i.e, the non-zero values are "bad"); and (2) the non-zeros of $\mathbf{y}$ are in a "bad" support. In particular, (1) states that the non-zeros of $\mathbf{y}$ can be permuted into another vector $\mathbf{y}'$ such that $\mathsf{wt}(\mathbf{y}'\mathbf{A}) = h$; and (2) states that the permutation from (1) is the identity (i.e., $\mathbf{y}$ already has a bad support and bad values). Let E_1 denote the first event and E_2 denote the second event just described. Then we can re-write the final probability above as

$$\begin{aligned}&\Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(\mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) = h \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w)\right] = \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(E_1 \wedge E_2 \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w)\right] \\ &= \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(E_2 \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w \wedge E_1)\right] \cdot \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(E_1 \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w)\right] \\ &\leq \Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(E_2 \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w \wedge E_1)\right],\end{aligned}$$

where the final inequality follows from upper bounding the second union by 1.

By our choice of $\mathbf{E}_1$ (as well as $\mathbf{E}_2$), we know that if $\mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w$, then $\mathbf{x}\mathbf{E}_1$ has non-zeros in a uniformly random support of w non-zeros in a vector of length N; i.e., the support of $\mathbf{x}\mathbf{E}_1$ is a uniformly sampled subset $T \subseteq [N]$ of size w. Finally, we need to count the number of "bad" supports to analyze the probability of event E_2; i.e., the number of weight w supports $T \subseteq [N]$ that map to a vector of weight h under the accumulator matrix $\mathbf{A}$. By Theorem 4.2, we know that this number is upper bounded by the number $A_{w,h}^{N,q}$ *without* the $(q-1)$ and $(q-2)$ terms;[1] i.e., this number is at most

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

Thus we have

$$\Pr_{\mathbf{E}_1}\left[\cup_{\mathbf{x}}(E_2 \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1) = w \wedge E_1)\right] \leq \frac{\sum_{i=0}^{w-1} \binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h-1}{\lceil \frac{w-i}{2} \rceil - 1} \binom{h - \lceil \frac{w-i}{2} \rceil}{i}}{\binom{N}{w}}.$$

Putting everything together, we have

$$\begin{aligned}&\Pr_{\mathbf{E}_1}\left[\exists \mathbf{x} \in \mathbb{F}^n \setminus \{0^n\} \colon \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N \wedge \mathsf{wt}(\mathbf{x}) \leq \beta n/t\right] \\ &\leq \sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=rct}^{2\delta N} \sum_{h=\lceil w/2 \rceil}^{\delta N} \sum_{i=0}^{w-1} \frac{\binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h-1}{\lceil \frac{w-i}{2} \rceil - 1} \binom{h - \lceil \frac{w-i}{2} \rceil}{i}}{\binom{N}{w}}.\end{aligned} \tag{7}$$

We now upper bound Eq. (7) to obtain an expression for the distance δ. First, we will make use of the following inequalities: $(x/y)^y \leq \binom{x}{y} \leq (ex/y)^y$; $\binom{x}{\lfloor y/2 \rfloor} \leq (4ex/y)^{y/2}$; and $\binom{x-1}{\lceil y/2 \rceil - 1} \leq \binom{x}{\lceil y/2 \rceil - 1} \leq (4ex/y)^{y/2}$.

[1] A simple numerical calculation shows that it is an upper bound and not equality; see Appendix B.2.

Now let ST denote the summand of Eq. (7). Using these inequalities, we have

$$\begin{aligned}\mathsf{ST} &\leq \left(\frac{w}{N}\right)^{w}\left(\frac{4e(N-h)}{w-i}\right)^{\frac{w-i}{2}}\left(\frac{4eh}{w-i}\right)^{\frac{w-i}{2}}\left(\frac{e(h-\lceil\frac{w-i}{2}\rceil)}{i}\right)^{i}\\ &= \left(\frac{4ew}{N}\right)^{w}\frac{((N-h)h)^{\frac{w-i}{2}}\cdot(h-\lceil\frac{w-i}{2}\rceil)^{i}}{(w-i)^{w-i}\cdot(4i)^{i}}.\end{aligned}$$

Over the choice of h, the numerator is maximized at $h=\delta N$. Plugging this in, we upper bound the above expression as:

$$\begin{aligned}&\left(\frac{4ew}{N}\right)^{w}\frac{((N-h)h)^{\frac{w-i}{2}}\cdot(h-\lceil\frac{w-i}{2}\rceil)^{i}}{(w-i)^{w-i}\cdot(4i)^{i}}\\ &\leq \left(\frac{4ew}{N}\right)^{w}\frac{N^{w-i}(1-\delta)^{\frac{w-i}{2}}(\delta)^{\frac{w-i}{2}}(\delta N-\lceil\frac{w-i}{2}\rceil)^{i}}{(w-i)^{w-i}(4i)^{i}}\\ &= (4ew\sqrt{\delta(1-\delta)})^{w}\left(\frac{\delta}{\sqrt{\delta(1-\delta)}}\right)^{i}\frac{1}{(w-i)^{w-i}(4i)^{i}}\cdot\left(1-\frac{1}{\delta N}\left\lceil\frac{w-i}{2}\right\rceil\right)^{i}.\end{aligned}$$

Since $0\leq i\leq w-1$ and $crt\leq w\leq 2\delta N$, the term $1-\frac{1}{\delta N}\left\lceil\frac{w-i}{2}\right\rceil$ is at most 1. Thus we can upper bound the above as

$$\leq\left(4ew\sqrt{\delta(1-\delta)}\right)^{w}\left(\frac{\delta}{\sqrt{\delta(1-\delta)}}\right)^{i}\frac{1}{(w-i)^{w-i}(4i)^{i}}.$$

Next, rationalizing $\sqrt{\delta}$ in the denominator, we have $\delta/\sqrt{\delta(1-\delta)}=\sqrt{\frac{\delta}{1-\delta}}$. We assume $\delta<1/2$, which implies that $\delta<1-\delta$ and $\delta/(1-\delta)<1$; thus $\sqrt{\frac{\delta}{1-\delta}}<1$. Over the choice of i, this term is maximized at $i=0$. On the other hand, the term $(w-i)^{w-i}(4i)^{i}$ is minimized at $i=w/5$,[2] and thus this term becomes

$$(w-i)^{w-i}(4i)^{i}\geq(4w/5)^{\frac{4w}{5}}(4w/5)^{\frac{w}{5}}=(4w/5)^{w}.$$

All together, we have

$$\begin{aligned}(4ew\sqrt{\delta(1-\delta)})^{w}\left(\frac{\delta}{\sqrt{\delta(1-\delta)}}\right)^{i}\frac{1}{(w-i)^{w-i}(4i)^{i}} &\leq(4ew\sqrt{\delta(1-\delta)})^{w}\left(\frac{5}{4w}\right)^{w}\\ &=(5e\sqrt{\delta(1-\delta)})^{w}.\end{aligned}$$

Plugging in this upper bound into Eq. (7), we obtain

$$\begin{aligned}\text{Eq. (7)} &\leq\sum_{r=1}^{\beta n/t}\binom{n}{r}\sum_{w=crt}^{2\delta N}\sum_{h=\lceil w/2\rceil}^{\delta N}\sum_{i=0}^{w-1}(5e\sqrt{\delta(1-\delta)})^{w}\\ &=\sum_{r=1}^{\beta n/t}\binom{n}{r}\sum_{w=crt}^{2\delta N}(ge)^{w}\cdot w\cdot(\delta N-\lceil w/2\rceil),\end{aligned}$$

where we set $g:=5\sqrt{\delta(1-\delta)}$ for ease of notation. For our purposes, we will choose $\delta<1/2$ such that $ge<1$. So assuming $ge<1$, notice that $(ge)^{w}$ is decreasing in w. Moreover, the term $(\delta N-\lceil w/2\rceil)$ is also decreasing

[2] This can be seen by taking the derivative of the function; see Appendix B.3.

in w, and the term w is increasing in w. Thus the summand above is a decreasing function in w in the interval $[crt, 2\delta N]$ and is maximized at $w = crt$. This gives us

$$\begin{aligned}
&\sum_{r=1}^{\beta n/t} \binom{n}{r} \sum_{w=crt}^{2\delta N} (ge)^w \cdot w \cdot (\delta N - \lceil w/2 \rceil) \\
&\leq \sum_{r=1}^{\beta n/t} \left(\frac{eRN}{r}\right)^r (ge)^{crt} \cdot (crt) \cdot (\delta N - \lceil crt/2 \rceil)(2\delta N - crt),
\end{aligned}$$

where we recall that $n = RN$. The above summand is decreasing for $1 \leq r \leq \beta n/t$ so long as $c\gamma \log(1/ge) > 1$ (which will be satisfied by our parameter choices),so the summand is maximized at $r = 1$. This gives us

$$\begin{aligned}
&\sum_{r=1}^{\beta n/t} \left(\frac{eRN}{r}\right)^r (ge)^{crt} \cdot (crt) \cdot (\delta N - \lceil crt/2 \rceil)(2\delta N - crt) \\
&\leq \frac{\beta RN}{t} \cdot (eRN) \cdot (ge)^{ct} \cdot (ct) \cdot (\delta N - \lceil ct/2 \rceil)(2\delta N - ct) \\
&\leq 2\beta ce(RN)^2 (ge)^{ct} (\delta N)^2 = 2\beta ce(R\delta)^2 N^4 (ge)^{ct}.
\end{aligned}$$

Recall that $t = \gamma \log(N)$ for expander parameter γ. Then we can rewrite the above expression as

$$\begin{aligned}
2\beta ce(R\delta)^2 N^4 (ge)^{ct} &= 2\beta ce(R\delta)^2 N^4 (ge)^{c\gamma \log(N)} \\
&= 2\beta ce(R\delta)^2 N^{4 + c\gamma \log(ge)}.
\end{aligned}$$

In order for the term $N^{4+c\gamma \log(ge))}$ to be inverse polynomial, we require

$$4 + c\gamma \log(ge) < 0 \implies g < 2^{-\frac{4}{c\gamma}}/e \implies \delta(1-\delta) < 1/(5 \cdot 2^{\frac{4}{c\gamma}} e)^2.$$

Since $\delta(1-\delta) < \delta$ for $\delta < 1/2$, choosing δ such that $\delta < 1/(5 \cdot 2^{4/(c\gamma)} e)^2$ will satisfy our requirements. Let $\eta > 2^{\frac{4}{c\gamma}}$ be a constant. Then we define $\delta := 1/(5\eta e)^2$. Now substituting this δ into our final bound, we obtain

$$\begin{aligned}
2\beta ce(R\delta)^2 N^{4 - c\gamma \log(1/(ge))} &= 2\beta ce \frac{R^2}{(5\eta e)^4} N^{4 - c\gamma \log(1/(ge))} && (8) \\
&\leq \frac{2\beta c R^2}{(5\eta)^4 e^3} N^{4 - c\gamma \log(e(5\eta)^2)} && (9) \\
&= O(N^{4-c^*}), && (10)
\end{aligned}$$

where $c^* := c^*(\epsilon, \gamma, \eta) = (1 - 2\epsilon)\gamma \log(e(5\eta)^2)$ is a constant. Thus, we have shown

$$\Pr_{\mathbf{E}_1}[\exists \mathbf{x} \in \mathbb{F}^n \setminus \{0^n\} \mid \mathsf{wt}(\mathbf{x}\mathbf{E}_1\mathbf{A}) \leq \delta N] \leq O(N^{4-c^*}) + \mathsf{negl}(N) = O(N^{4-c^*}),$$

completing the low-weight input vector case, so long as $c\gamma \log(1/ge) > 1$. Recalling our choice of δ above, we have that

$$4 + c\gamma \log(ge) < 0 \implies 4 < -c\gamma \log(ge) \implies 4 < c\gamma \log(1/ge),$$

so clearly $c\gamma \log(1/ge) > 1$ holds.

4.4 Distance Analysis for High-weight Messages

Now, we turn to the case when input vectors $\mathbf{x}$ have weight $> \beta n/t$. In this case, we perform the same analysis as before but now with respect to the random Bernoulli expander matrix $\mathbf{E}_2$. Recall that we defined $\mathbf{E}_2 \in \mathbb{F}^{n\times N}$ as taking i.i.d. entries from the distribution $\mathsf{Ber}_p(\mathbb{F})$, which with probability p samples a uniformly random element from the set $\mathbb{F}\setminus\{0\}$ and with probability $1-p$ samples $0 \in \mathbb{F}$. Here, $p = t/N$ for $t = \gamma\log(N)$.

Ideally, for this case we would like to leverage the prior analysis we have already performed for the matrix $\mathbf{E}_1$. In particular, if we can prove that $\mathbf{E}_2$ also has good expansion for inputs of weight $r \geq \beta n/t$, then our analysis can proceed identically as in the low-weight case. Note that Lemma 3.7, which we presented in Section 3.3, gives us exactly this property. In particular, with Lemma 3.7, we have that with overwhelming probability (i.e., $1-\mathsf{negl}(N)$) $\mathsf{wt}(\mathbf{x}\mathbf{E}_2) \geq rct$ whenever $r \geq \beta n/t$. This gives us the following new probability to analyze for this case:

$$\begin{aligned}
&\Pr_{\mathbf{E}_2}\left[\exists x \in \mathbb{F}^n \setminus \{0^n\} \colon \mathsf{wt}(x\mathbf{E}_2\mathbf{A}) \leq \delta N \wedge \mathsf{wt}(x) \geq \beta n/t\right] \\
&\leq \mathsf{negl}(N) + \sum_{r=\beta n/t}^{n} \binom{n}{r} \sum_{w=rct}^{2\delta N} \sum_{h=\lceil w/2\rceil}^{\delta N} \sum_{i=0}^{w-1} \frac{\binom{N-h}{\lfloor \frac{w-i}{2}\rfloor}\binom{h-1}{\lceil\frac{w-i}{2}\rceil-1}\binom{h-\lceil\frac{w-i}{2}\rceil}{i}}{\binom{N}{w}}. \qquad (11)
\end{aligned}$$

Here, the $\mathsf{negl}(N)$ term comes from Lemma 3.7.

With Eq. (11), we have shown that for the high weight case, when using the Bernoulli matrix $\mathbf{E}_2$, we obtain nearly the same probability as in the weight $r \leq \beta n/t$ case. In fact, we can simply begin the analysis of Eq. (11) using the upper bound obtained on the summations over w, h, and i before analyzing the summation over r. This gives us

$$\begin{aligned}
\text{Eq. (11)} \leq\ &\mathsf{negl}(N) \\
&+ \sum_{r=\beta n/t}^{n} \left(\frac{eRN}{r}\right)^r (ge)^{crt} \cdot (crt) \cdot \left(\delta N - \left\lceil \frac{crt}{2}\right\rceil\right)(2\delta N - crt).
\end{aligned}$$

By the same analysis as before, the summation term on the second line is decreasing in r by our choice of parameters, so it is maximized at $r = \beta n/t$. However, for simplicity, we maximize this term at $r = 1$ to obtain the same bound as before. All together with the prior analysis for the weight $r \leq \beta n/t$ case, we have

$$\begin{aligned}
\text{Eq. (11)} &\leq \mathsf{negl}(N) + \left(RN - \frac{\beta RN}{t}\right)\left(2\beta ce(R\delta)^2 N^4 (ge)^{ct}\right) \\
&= \mathsf{negl}(N) + 2\beta ce R^3\delta^2 N^5 (ge)^{ct} \\
&= \mathsf{negl}(N) + 2\beta ce R^3\delta^2 N^{5+c\gamma\log(ge)}.
\end{aligned}$$

By our prior analysis, for parameter $\eta > 2^{\frac{5}{c\gamma}}$, setting $\delta := 1/(5\eta e)^2$ gives us

$$\begin{aligned}
\mathsf{negl}(N) + 2\beta ce R^3\delta^2 N^{5+c\gamma\log(ge)} &\leq \mathsf{negl}(N) + \frac{2\beta cR^3}{(5\eta)^4 e^3} N^{5-c\gamma\log(e(5\eta)^2)} \qquad (12) \\
&= O(N^{5-c^*}) + \mathsf{negl}(N) = O(N^{5-c^*}) \qquad (13)
\end{aligned}$$

for constant $c^* = (1-2\epsilon)\gamma\log(e(5\eta)^2)$.

Finally, we argue that $c^* > 5$. First, note that, in the low-weight case, we defined $\eta > 2^{4/(c\gamma)}$, and in this case we choose $\eta > 2^{5/(c\gamma)}$. Thus, c^* for the high-weight case needs to be larger than in the low-weight case; since $c^* > 5$ is what we require in this case, this choice will ensure that the probability in the low-weight case is also inverse polynomial. Expanding, we have

$$\begin{aligned}
c^* &= (1-2\epsilon)\gamma\log(e(5\eta)^2) = (1-2\epsilon)\gamma(\log(e) + 2\log(5) + 2\log(\eta)) \\
&> (1-2\epsilon)\gamma(\log(e) + 2\log(5) + (10/((1-2\epsilon)\gamma))\log(2)) \\
&= (1-2\epsilon)\gamma(\log(e) + 2\log(5)) + 10 \\
&> 5,
\end{aligned}$$

where the last inequality follows since $\gamma > 0$ and $\epsilon \in (0, 1/2)$. This completes the proof of Theorem 4.1.

4.5 Conjectures

Here, we briefly discuss some conjectures related to the EA codes we consider in this paper and utilize in the next section (Section 6). First, with respect to $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F})$, we have shown that the code $\mathbf{E}_1\mathbf{A}$ is asymptotically good so long as the input vectors $\mathbf{x}$ have weight at most $O(n/t)$. This partially resolves an open question of [BCG+22]. We believe that this limitation is due to a limitation in our analysis and not inherent to the code $\mathbf{E}_1\mathbf{A}$ itself. In particular, we conjecture that $\mathbf{E}_1\mathbf{A}$ is asymptotically good.

Conjecture 4.3. *Let $R = n/N$ be a constant. There exist constants $\gamma \geq 1$ and $\delta < 1/2$ such that for $t = \gamma \log N$, we have $\Pr_{\mathbf{E}_1}[\mathsf{d}(\mathbf{E}_1\mathbf{A}) \geq \delta] \geq 1 - 1/\operatorname{poly}(N)$, where $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_q)$.*

Pseudo-distance of EA Codes Over General Finite Fields. Boyle et al. [BCG+22] formally defined the notion of *pseudo-distance*, which appeared in prior work as the *binary shortest vector problem* [AHI+17]. At a high-level, a code C has pseudo-distance δ if no efficient adversary can find a non-zero vector $\mathbf{x}$ such that $\mathsf{wt}(C(\mathbf{x})) \leq \delta N$, except with negligible probability. Note that their original definition was for binary linear codes, but naturally extends to linear codes over arbitrary finite fields. [BCG+22] states various conjectures regarding the pseudo-distance of their various EA codes over the binary field. Here, we make one similar conjecture about the pseudo-distance of our EA codes over finite fields.

Conjecture 4.4. *Let $R = n/N$ be a constant. There exist constants $\delta < (1 - R)/2$ and γ such that the code $\mathbf{E}_1\mathbf{A}$ (for $\mathbf{E}_1 \leftarrow \mathsf{BP}(n, N, t, \mathbb{F}_q)$) has pseudo-distance δ with high probability if $t = \gamma \log N$.*

5 Polynomial Commitments and SNARKs from Codes

We build a polynomial commitment scheme (PCS) following the approach proposed in [GLS+23]. It is constructed from any linear error-correcting code and the security relies on the constant relative distance of the code. Therefore, the protocol only uses a generic encoding algorithm of the code, instead of any specific linear code. Due to space limitations, we refer the readers to prior work such as [GLS+23] for the formal definitions of R1CS, PCS, and SNARKs.

The PCS construction also uses Merkle trees [Mer90], which are a data structure that allows one to commit to a vector of messages by a single hash value h, such that any message m_i can be later revealed and verified with the hash values on the path from m_i to the root. We denote the algorithm as follows:

- $h \leftarrow \mathsf{Merkle.Commit}(\mathbf{m})$.
- $(m_i, \pi_i) \leftarrow \mathsf{Merkle.Open}(\mathbf{m}, \mathbf{i})$.
- $\{0, 1\} \leftarrow \mathsf{Merkle.Verify}(\pi_{\mathsf{i}}, m_i, h)$.

We present the scheme in Protocol 1 for completeness. A polynomial evaluation can be expressed as a tensor product. Here we give an example of multilinear polynomials, which can be used to construct SNARKs based on the approaches in [ZGK+17, WTs+18, XZZ+19, ZXZS20, Set20], but the protocol works for any polynomial. Given a multilinear polynomial ϕ, its evaluation on input vector $x_0, x_1, ..., x_{\log N-1}$ is:

$$\phi(x_0, x_1, ..., x_{\log N-1}) = \sum_{i_0=0}^{1} \sum_{i_1=0}^{1} ... \sum_{i_{\log N-1}=0}^{1} c_{i_0 i_1 ... i_{\log N-1}} x_0^{i_0} x_1^{i_1} ... x_{\log N-1}^{i_{\log N-1}}.$$

The degree of each variable is either 0 or 1 by the definition of a multilinear polynomial, and thus there are N monomials and coefficients with $\log N$ variables. Let $i_0 i_1 ... i_{\log N-1}$ be the binary representation of number i, i.e., $i = \sum_{j=0}^{\log N-1} 2^j i_j$. We use c to denote the coefficients where $c[i] = c_{i_0 i_1 ... i_{\log N-1}}$. Similarly, we define $X_i = x_0^{i_0} x_1^{i_1} ... x_{\log N-1}^{i_{\log N-1}}$. For $k = \sqrt{N}$, define vectors $\mathbf{x}_0, \mathbf{x}_1$ as $\mathbf{x_0} = \{\mathbf{X_0}, \mathbf{X_1}, ..., \mathbf{X_{k-1}}\}, \mathbf{x_1} =$

Protocol 1 Polynomial Commitment Scheme from any Linear Error-correcting Code [GLS+23]

Public input: The evaluation point $\mathbf{x}$, parsed as a tensor product $\mathbf{x} = \mathbf{x_0} \otimes \mathbf{x_1}$;
Private input: The polynomial ϕ, the coefficients of ϕ are denoted by c.
Let C be the $[n, k, d]$-linear code, $E_C : \mathbb{F}^k \to \mathbb{F}^n$ be the encoding function, $N = k \times k$. If N is not a perfect square, we can pad it to the next perfect square.
We use a python style notation to select the i-th column of a matrix $\mathsf{mat}[:, i]$.

function PC.Commit(ϕ)
  Parse c as a $k \times k$ matrix C. The prover computes the encoding C_1 locally. In particular, C_1 is a $k \times n$ matrix by encoding each row of the matrix C using E_C.
  for $i \in [n]$ **do**
    Compute the Merkle tree root $\mathsf{Root}_i = \mathsf{Merkle.Commit}(\mathsf{C}_1[:, \mathsf{i}])$.
  Compute a Merkle tree root $\mathsf{Root} = \mathsf{Merkle.Commit}([\mathsf{Root}_0, ..., \mathsf{Root}_{n-1}])$ and output Root as the commitment.
function PC.open($\phi, \mathbf{x}$)
  $\mathcal{P}$ receives a random vector $r \in \mathbb{F}^k$ from $\mathcal{V}$, and computes $y_r = \sum_{i=0}^{k-1} r[i]\mathsf{C}[i]$, $y_1 = \sum_{i=0}^{k-1} \mathbf{x_0}[\mathbf{i}]\mathsf{C}[\mathbf{i}]$.
  $\mathcal{P}$ sends y_r, y_1 to $\mathcal{V}$.
  $\mathcal{V}$ randomly samples $t \in [n]$ indices as an array I and sends it to $\mathcal{P}$.
  for $i \in I$ **do**
    $\mathcal{P}$ sends $\mathsf{C}_1[:, i]$ and $(\mathsf{Root}_i, \pi_i) \leftarrow \mathsf{Merkle.Open}([\mathsf{Root}_0, ..., \mathsf{Root}_{n-1}], i)$ to $\mathcal{V}$.
function PC.Verify($\pi_{\mathbf{x}}, \mathbf{x}, \mathbf{y} = \phi(\mathbf{x}), \mathsf{Root}$)
  Compute $c_r = E_C(y_r)$ and $c_1 = E_C(y_1)$.
  for $\forall i \in I$ **do**
    Check if $c_r[i] == \langle r, \mathsf{C}_1[:, i]\rangle$.
    Check if $c_1[i] == \langle \mathbf{x_0}, \mathsf{C}_\mathbf{1}[:, \mathbf{i}]\rangle$.
    Compute $\mathsf{Root}_i = \mathsf{Merkle.Commit}\mathsf{C}_1[:, i])$; run $\mathsf{Merkle.Verify}(\pi_i, \mathsf{Root}_i, \mathsf{Root})$.
  Check if $y == \langle \mathbf{x_1}, \mathbf{y_1}\rangle$.
  Output 1 if all if the above conditions hold, otherwise, output 0.

$\{\mathbf{X_{0\times k}}, \mathbf{X_{1\times k}}, \mathbf{X_{2\times k}}, ..., \mathbf{X_{(k-1)\times k}}\}$. Then we have $X = \mathbf{x_0} \otimes \mathbf{x_1}$. The polynomial evaluation is reduced to a tensor product $\phi(x_0, x_1, ..., x_{\log N-1}) = \langle c, \mathbf{x_0} \otimes \mathbf{x_1}\rangle$.

Theorem 5.1 ([GLS+23])**.** *Protocol 1 is a polynomial commitment scheme with soundness error $\frac{N}{\mathbb{F}} + (1 - \frac{\delta}{3})^t + (1 - \frac{2\delta}{3})^t$, where δ is the relative distance of the code, and t is the number of opened columns in Protocol 1.*

Note that the above theorem directly tells us that for fixed soundness error $2^{-\lambda}$, larger minimum distance δ implies that a smaller number of column openings t are needed to achieve the desired soundness error. This directly implies that larger minimum distance gives us smaller proof sizes.

With the constant relative distance of the EA code, by setting $t = \lambda \cdot \frac{1}{-\log_2(1-\delta/3)}$, the soundness error can be made $2^{-\lambda}$. The proof size is $O(tk) = O(t\sqrt{N})$. By optimizing the dimensions of the coefficient matrix, the proof size can be improved to $O(\sqrt{tN})$ (see [GLS+23] for details).

Finally, we build a SNARK by combining the multivariate polynomial commitment with the sum-check protocol as in [GLS+23], which is based on the SNARK for R1CS in Spartan [Set20]. We refer the readers to [Set20, GLS+23] for the construction and the proof. The scheme consists of a single multilinear polynomial commitment and opening, and a sum-check protocol, thus the prover time and the proof size are both dominated by those of the PCS. The scheme can be made non-interactive via the Fiat-Shamir [FS87] heuristic, and has plausible post-quantum security. The scheme can be made zero-knowledge using known transformations in [BCG+17, BCL22, XZS22]. By instantiating the linear code with the EA code, we have the following theorem.

Theorem 5.2. *In the framework of [GLS+23], Protocol 1 instantiated with the EA code yields a SNARK with $O(M \log M)$ prover time and $O(\sqrt{M})$ proof size for R1CS relations of size M.*

6 Experiments

We implement (in Rust) a framework corresponding to the polynomial commitment scheme and the SNARK in [GLS+23], and instantiate it using our juxtaposed EA code. We then compare the resulting constructions to Ligero [AHIV17], Aurora [BCR+19], Brakedown [GLS+23], and Groth16 [Gro16]. For Ligero and Aurora, we use the open-source implementations from libiop [sl23] with a 256-bit prime field. We provide our own implementation of Brakedown by incorporating the linear code from that work into our generic framework. For Groth16, we use the open-source implementation available at [BSCTV14]. We conduct all out experiments on an AWS c5a.16xlarge Ubuntu 22.04 machine with 64 cores and 124G memory. Reported numbers are averages over 5 runs. Our code used for these experiments can be found at `https://artifacts.iacr.org/crypto/2024/a10/`.

Among other SNARKs, Plonk [GWC19] does not use R1CS and, in general, it has worse performance than Groth16 (though with the advantage of universal setup and allowing for customized gates). Stark [BBHR19] uses the Algebraic Intermediate Representation (AIR) instead of circuits or R1CS, and we were therefore not able to directly compare it to our work. Orion [XZS22] uses recursive proofs to reduce the proof size. We could also use the same technique, but then our scheme would no longer be field-agnostic.

Parameters. To obtain the concrete parameters, we numerically compute the failure probability based on Equations (7) and (11) to achieve provable distance $\delta = 0.1$ with failure probability 2^{-100}. We use the following parameters for estimation: $n = 1024, N = 2048, \beta = 2, \epsilon = 0.1$, and $\gamma \geq 18$. We conjectured the failure probability will only be smaller with the increase of code length since the analysis shows it is inverse-polynomial in N. This is also evident by the numerical computation for $n \in \{1024, 2048, 4096\}$. We also use the same parameters for our code with conjectured distance $\delta = 1/4$ based on Conjecture 4.4.

For Brakedown, we report the performance with two sets of parameters. One from the original Brakedown paper [GLS+23], and the other from a recently improved analysis of the code [Hab23]. In the first case, we use the same set of parameters as Brakedown does in their experiments, i.e. distance=0.04, rate=0.65, $\alpha = 0.178, c_n = 7, d_n = 22$. In the second case, we managed to derive a set of parameters based on the analysis in that paper, targeting the same rate and distance as we used for our code. Specifically, we use rate=0.5, distance=0.1, $\alpha = 0.3, c_n = 11, d_n = 22$.

We conduct all the experiments in the scalar field of the BN254 curve.

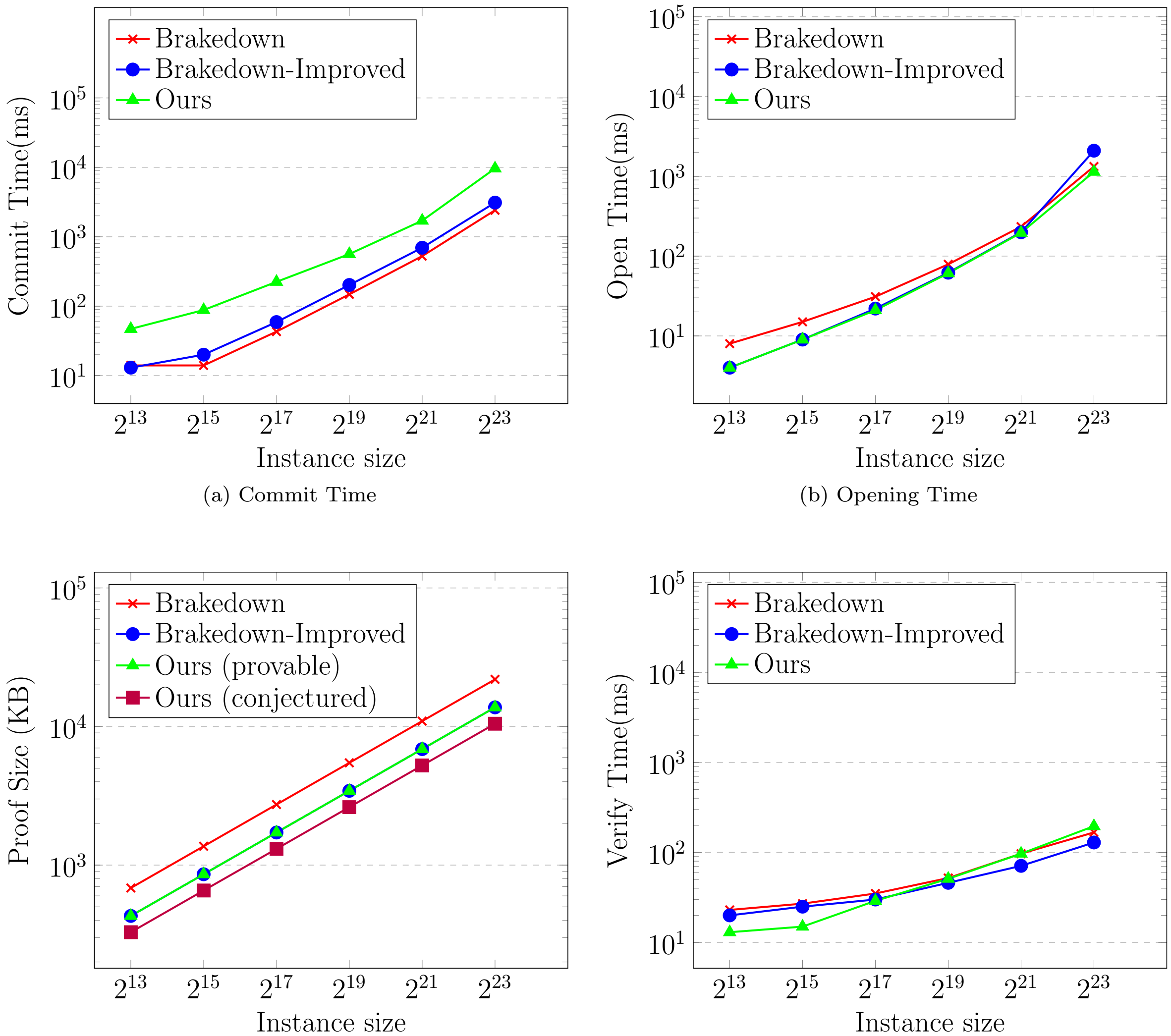
Additional optimizations. We introduce the following two optimizations in our implementation:

- *Matrix dimensions.* As shown in Protocol 1, constructing a polynomial commitment involves arranging the coefficients of a polynomial of degree M in an $O(\sqrt{M}) \times O(\sqrt{M})$ matrix. Since multiple columns and one row are opened in the proof, we set the column size to be smaller than the row size.

- *Parallelization.* We parallelize the commit and verification phases of the polynomial commitment scheme. In the commit phase, we parallelize the encoding of different rows of the coefficient matrix. In the verification phase, however, we parallelize the sparse matrix multiplication within the encoding since there are only two messages to encode.

  Our juxtaposed EA code benefits more from parallelization than the code used by Brakedown, which has a recursive encoding algorithm; we view this as an advantage of our construction. Ligero can theoretically be parallelized as well, but we were not able to add parallelization to the existing library.

Remark 6.1. There is one additional optimization due to Diamond and Posen [DP24], which applies to Protocol 1 but we *do not implement* it in our work. Essentially, this work shows how one can skip the testing phase of the Brakedown polynomial commitment scheme. This results in roughly a $\sqrt{2}\times$ improvement in proof size, and roughy a $2\times$ improvement in prover and verifier time. As this optimization applies to the PCS derived from both Brakedown's code and our EA code, this improvement holds in both cases.



(a) Commit Time

(b) Opening Time

(c) Proof Size

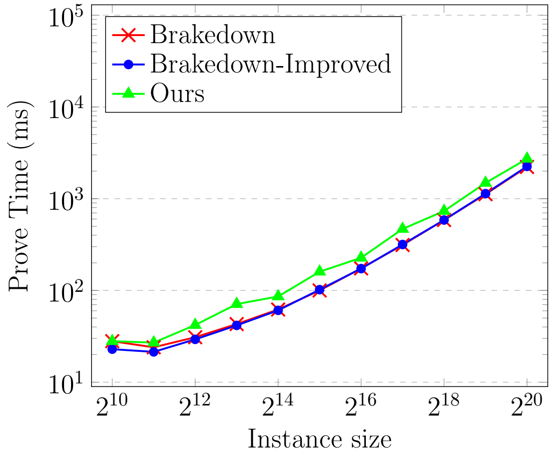
(d) Verification Time

Figure 1: Performance of polynomial commitment schemes.

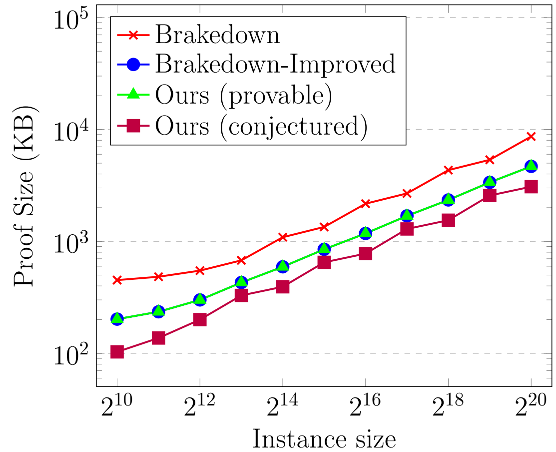
6.1 Polynomial Commitment Scheme

In this section, we report on the performance of the polynomial commitment scheme based on our EA code, and compare it with Brakedown. The results are shown in Fig. 1. The time to compute a commitment, which depends largely on the encoding time of the underlying code, is larger in our scheme than in Brakedown. However, although the asymptotic encoding time for the EA code is $O(N \log N)$ and the encoding time of the code used in Brakedown is $O(N)$, the observed difference in the commitment times is only 3–4$\times$ thanks to the concretely simple encoding algorithm of the EA code. The time to open a polynomial at an evaluation point is similar in both schemes, and is significantly faster than the time to compute a commitment.

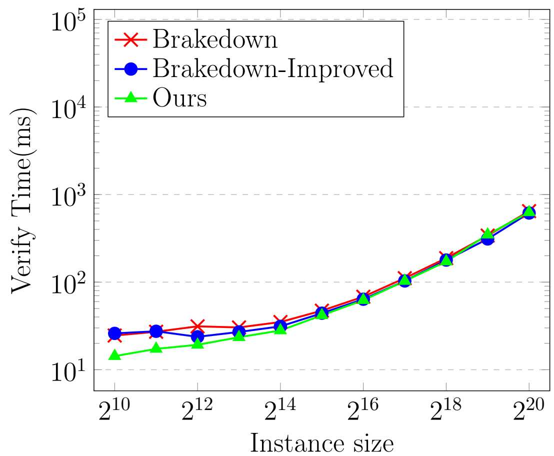
The main advantage of our scheme over Brakedown is in the proof size. As shown in Fig. 1, our proof size is 430KB for instance size 2^{13} when using the provable distance of our code, and only 328KB using the conjectured distance. These are 1.6–2.1$\times$ smaller than the proof size of Brakedown. The verifier times of both schemes are similar.



(a) Prover Time



(b) Proof Size



(c) Verifier Time

Figure 2: Performance of SNARKs on random R1CS instances.

6.2 SNARKs

Next, we benchmark the performance of the SNARK built from our EA code and compare it to both the original Brakedown paper and the follow-up improvements. (We only benchmark the variants of these schemes with verifier time linear in the number of constraints. In the holographic setting, the verifier time can be made sublinear in both schemes using the same technique [Set20], and thus we expect that the comparison would remain roughly the same.) Fig. 2 shows the prover time, proof size, and verifier time of our scheme and Brakedown. For 2^{20} constraints, our prover time is 2.7s, which is only 1.2× slower than the original Brakedown result. The proof size of our SNARK is 4.6MB using the provable distance of our EA code, and can be further reduced to 3.0MB using the conjectured distance. These are 1.9–2.8× smaller than the proof size of the original Brakedown. The verifier times of both schemes are again similar. When compared to the improved Brakedown, our proof sizes for provable distance are close to the improved Brakedown proof sizes, while our conjectured distance still gives us smaller proof sizes. The prover and verifier times of the improved Brakedown are similar to the original Brakedown.

Table 2: Performance of different SNARKs for ECDSA verification.

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

6.3 ECDSA Verification

Some SNARK applications are faster when a native field can be used. For example, ECDSA signature verification (a common computation used in zkRollups and zkBridges) is, in practice, typically defined on the secp256k1 curve, which is in turn based on a particular field. To use a SNARK to prove signature verification, we can either express the verification algorithm in the scalar field of secp256k1 directly, or in a different field. However, the latter is very expensive since implementing field additions and multiplications in another field introduces a large overhead. For example, the open-source implementation of the ECDSA signature verification on a non-native field—i.e., the scalar field of the bn254 curve—in the circom library [Sun24] involves 2^{21} R1CS constraints, whereas we were able to implement ECDSA verification in the native field with only 2^{16} constraints. This is why field-agnostic SNARKs can have significant advantages.

We report on the performance of several SNARKs for ECDSA verification in Table 2. For Ligero and Groth16 (which are not field-agnostic), we report their performance on an R1CS of size 2^{21}. For Brakedown and our scheme, we report the performance on our R1CS implementation in the native field. As shown in the table, the prover time in our scheme is 448×, 2321×, and 648× better compared to Ligero, Aurora, and Groth16, respectively. One part of the speedup comes from the smaller instance size, and another comes from the more-efficient encoding of the EA code. The proof size of our scheme is also 18–30× smaller than for Ligero, although it is still 4.5× larger than Aurora, and significantly larger than Groth16. The comparison to Brakedown is similar to the case of general SNARKs discussed in the previous section; namely, our scheme has slower prover time but smaller proof size.

Remark 6.2. Note that curve-based SNARKs (e.g., those based on discrete-log or pairings) can be used over any large enough field by finding a curve with a scalar field equal to the target field. For the secp256k1 curve, the work of [SSS+22] does this with a new curve called secq. This implies that for secp256k1 signature verification, one can use curve-based SNARKs instantiated with the secq curve. However, this approach is not truly field agnostic, as it needs a secure elliptic curve to be defined over (an extension field of) the field of interest, and it can be difficult to find new curves for every target field of interest. Moreover, such an approach is not post-quantum secure, and there are questions about the security of curves over extension fields [GHS02, Fre98, Gau09, PQ12].

Acknowledgments

Work supported in part by DARPA under Contract Nos. HR00112020025 and HR001120C0087 and by NSF awards #2401481 and #2154705. This work has been done while Hendrik Waldner was at the University of Maryland, supported by a MC2 postdoctoral fellowship. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as reflecting the position or policy of the the Department of Defense or the U.S. Government, and no official endorsement should be inferred.

References

[AHI+17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In Christos H. Papadimitriou, editor, *ITCS 2017*, volume 4266, pages 7:1–7:31, 67, January 2017. LIPIcs. doi:10.4230/LIPIcs.ITCS.2017.7.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, October / November 2017. doi:10.1145/3133956.3134104.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018. doi:10.1109/SP.2018.00020.

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In Hovav Shacham and Alexandra Boldyreva, editors, *Crypto 2018, Part II*, volume 10992 of *LNCS*, pages 669–699. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96881-0_23.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl, July 2018. doi:10.4230/LIPIcs.ICALP.2018.14.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *Crypto 2019, Part III*, volume 11694 of *LNCS*, pages 701–732. Springer, Heidelberg, August 2019. doi:10.1007/978-3-030-26954-8_23.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Eurocrypt 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_12.

[BCG+17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Asiacrypt 2017, Part III*, volume 10626 of *LNCS*, pages 336–365. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70700-6_12.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 19–46. Springer, Heidelberg, November 2020. doi:10.1007/978-3-030-64378-2_2.

[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, Heidelberg, August 2022. doi:10.1007/978-3-031-15979-4_21.

[BCKL22] Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, and David Levit. Scalable and transparent proofs over all large fields, via elliptic curves - (ECFFT part II). In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 467–496. Springer, Heidelberg, November 2022. doi:10.1007/978-3-031-22318-1_17.

[BCKL23] Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, and David Levit. Elliptic curve fast fourier transform (ECFFT) part I: low-degree extension in time $O(n \log n)$ over all finite fields. In *SODA*, pages 700–737. SIAM, 2023.

[BCL22] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge IOPs with linear-time prover and polylogarithmic-time verifier. In Orr Dunkelman and Stefan Dziembowski, editors, *Eurocrypt 2022, Part II*, volume 13276 of *LNCS*, pages 275–304. Springer, Heidelberg, May / June 2022. doi:10.1007/978-3-031-07085-3_10.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *Eurocrypt 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Heidelberg, May 2019. doi:10.1007/978-3-030-17653-2_4.

[BCS23] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Lattice-based succinct arguments for NP with polylogarithmic-time verification. In Helena Handschuh and Anna Lysyanskaya, editors, *Crypto 2023, Part II*, volume 14082 of *LNCS*, pages 227–251. Springer, Heidelberg, August 2023. doi:10.1007/978-3-031-38545-2_8.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent SNARKs from DARK compilers. In Anne Canteaut and Yuval Ishai, editors, *Eurocrypt 2020, Part I*, volume 12105 of *LNCS*, pages 677–706. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45721-1_24.

[BGKS20] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling outside the box improves soundness. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 5:1–5:32. LIPIcs, January 2020. doi:10.4230/LIPIcs.ITCS.2020.5.

[BHR+20] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Public-coin zero-knowledge arguments with (almost) minimal time and space overheads. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 168–197. Springer, Heidelberg, November 2020. doi:10.1007/978-3-030-64378-2_7.

[BHR+21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Time- and space-efficient arguments from groups of unknown order. In Tal Malkin and Chris Peikert, editors, *Crypto 2021, Part IV*, volume 12828 of *LNCS*, pages 123–152, Virtual Event, August 2021. Springer, Heidelberg. doi:10.1007/978-3-030-84259-8_5.

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, *Crypto 2020, Part II*, volume 12171 of *LNCS*, pages 441–469. Springer, Heidelberg, August 2020. doi:10.1007/978-3-030-56880-1_16.

[BSCTV14] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von Neumann architecture. In *Proceedings of the USENIX Security Symposium*, 2014.

[But] Vitalik Buterin. An Incomplete Guide to Rollups. https://vitalik.eth.limo/general/2021/01/05/rollup.html.

[DI14] Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the Gilbert-Varshamov bound and their cryptographic applications. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, page 169–182. ACM, 2014. doi:10.1145/2554797.2554815.

[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.

[DP24] Benjamin E. Diamond and Jim Posen. Proximity testing with logarithmic randomness. *IACR Communications in Cryptology*, 1(1), 2024. doi:10.62056/aksdkp10.

[Fre98] Gerhard Frey. How to disguise an elliptic curve (weil descent), 1998. URL: http://www.cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Crypto'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.

[Gau09] Pierrick Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *Journal of Symbolic Computation*, 44(12):1690–1702, 2009. Gröbner Bases in Cryptography, Coding Theory, and Algebraic Combinatorics. URL: https://www.sciencedirect.com/science/article/pii/S074771710800182X, doi:10.1016/j.jsc.2008.08.005.

[GHS02] Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and destructive facets of weil descent on elliptic curves. *J. Cryptology*, 15:19–46, 2002. doi:10.1007/s00145-001-0011-x.

[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, Heidelberg, August 2023. doi:10.1007/978-3-031-38545-2_7.

[GM08] Venkatesan Guruswami and Widad Machmouchi. Explicit interleavers for a repeat accumulate accumulate (RAA) code construction. In *ISIT*, pages 1968–1972. IEEE, 2008.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Eurocrypt 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_11.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. https://eprint.iacr.org/2019/953.

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

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[Mer90] Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *Crypto'89*, volume 435 of *LNCS*, pages 218–238. Springer, Heidelberg, August 1990. doi:10.1007/0-387-34805-0_21.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. doi:10.1109/SP.2013.47.

[PQ12] Christophe Petit and Jean-Jacques Quisquater. On polynomial systems arising from a Weil descent. In Xiaoyun Wang and Kazue Sako, editors, *Asiacrypt 2012*, volume 7658 of *LNCS*, pages 451–466. Springer, Heidelberg, December 2012. doi:10.1007/978-3-642-34961-4_28.

[RRT23] Srinivasan Raghuraman, Peter Rindal, and Titouan Tanguy. Expand-convolute codes for pseudorandom correlation generators from LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *Crypto 2023, Part IV*, volume 14084 of *LNCS*, pages 602–632. Springer, Heidelberg, August 2023. doi:10.1007/978-3-031-38551-3_19.

[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, Heidelberg, August 2020. doi:10.1007/978-3-030-56877-1_25.

[sl23] scipr lab. libiop, 2023. URL: https://github.com/scipr-lab/libiop.

[Spi96] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[SSS+22] Huachuang Sun, Haifeng Sun, Kevin Singh, Akhil Sai Peddireddy, Harshad Patil, Jianwei Liu, and Weikeng Chen. The inspection model for zero-knowledge proofs and efficient zerocash with secp256k1 keys. Cryptology ePrint Archive, Report 2022/1079, 2022. https://eprint.iacr.org/2022/1079.

[Sun24] Yi Sun. circom-ecdsa, 2024. URL: https://github.com/0xPARC/circom-ecdsa.

[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.

[WTs+18] Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. Doubly-efficient zkSNARKs without trusted setup. In *2018 IEEE Symposium on Security and Privacy*, pages 926–943. IEEE Computer Society Press, May 2018. doi:10.1109/SP.2018.00060.

[XZC+22] Tiancheng Xie, Jiaheng Zhang, Zerui Cheng, Fan Zhang, Yupeng Zhang, Yongzheng Jia, Dan Boneh, and Dawn Song. zkBridge: Trustless cross-chain bridges made practical. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 3003–3017. ACM Press, November 2022. doi:10.1145/3548606.3560652.

[XZS22] Tiancheng Xie, Yupeng Zhang, and Dawn Song. Orion: Zero knowledge proof with linear prover time. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Crypto 2022, Part IV*, volume 13510 of *LNCS*, pages 299–328. Springer, Heidelberg, August 2022. doi:10.1007/978-3-031-15985-5_11.

[XZZ+19] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In Alexandra Boldyreva and Daniele Micciancio, editors, *Crypto 2019, Part III*, volume 11694 of *LNCS*, pages 733–764. Springer, Heidelberg, August 2019. doi:10.1007/978-3-030-26954-8_24.

[ZCF23] Hadas Zeilberger, Binyi Chen, and Ben Fisch. Basefold: Efficient field-agnostic polynomial commitment schemes from foldable codes. *Cryptology ePrint Archive*, 2023.

[ZGK+17] Yupeng Zhang, Daniel Genkin, Jonathan Katz, Dimitrios Papadopoulos, and Charalampos Papamanthou. A zero-knowledge version of vSQL. Cryptology ePrint Archive, Report 2017/1146, 2017. https://eprint.iacr.org/2017/1146.

[ZXZS20] Jiaheng Zhang, Tiancheng Xie, Yupeng Zhang, and Dawn Song. Transparent polynomial delegation and its applications to zero knowledge proof. In *2020 IEEE Symposium on Security and Privacy*, pages 859–876. IEEE Computer Society Press, May 2020. doi:10.1109/SP40000.2020.00052.

A Definitions of Polynomial Commitments and SNARKs

Definition A.1 (Rank-1 Constraint System). *A Rank-1 Constraint System (R1CS) is defined by a tuple $(\mathbb{F}, A, B, C, M, N, io)$. Here $A, B, C \in \mathbb{F}^{N\times M}$ are sparse in the sense that the number of non-zero entries $N = O(M)$.* io *is the public input vector satisfying $|io| < M$.*

An R1CS relation $\mathcal{R}_{R1CS}$ *can be denoted as* $\{\langle(\mathbb{F}, A, B, C, M, N, \mathsf{io}), w\rangle \mid z := (1, \mathsf{io}, w) \in \mathbb{F}^M; Az \circ Bz = Cz\}$*, where* $\circ$ *denotes element-wise product.*

Definition A.2 (Succinct non-interactive argument of knowledge (SNARK))**.** *Consider a relation* $\mathcal{R}$ *with public instance* u *and private witnesses* w*, a SNARK for* $\mathcal{R}$ *consists of PPT algorithms* ($\mathcal{G}$, $\mathcal{P}$, $\mathcal{V}$) *with the following properties:*

- *Completeness. For any instance* $(u, w) \in \mathcal{R}$

$$\Pr\left[\ \mathcal{V}(u, vk, \pi) = 1 \ \middle|\ \begin{array}{c} (pk, vk) \leftarrow \mathcal{G}(1^\lambda), \\ \pi \leftarrow \mathcal{P}(u, w, pk), \end{array}\right] = 1$$

- *Knowledge Soundness: for any PPT adversary* $\mathbf{A}$*, there exists an expected PPT knowledge extractor* $\mathbf{E}$ *such that for any instance* u*,*

$$\Pr\left[\ \begin{array}{c} w \leftarrow \mathbf{E}^{\mathbf{A}}(u, \pi^*, pk), \\ \\ (u, w) \notin R \end{array} \ \middle|\ \begin{array}{c} (pk, vk) \leftarrow \mathcal{G}(1^\lambda), \\ \pi^* \leftarrow \mathbf{A}(u, pk), \\ \mathcal{V}(u, vk, \pi^*) = 1 \end{array}\right] \leq \mathsf{negl}(\lambda)$$

  where $\mathbf{E}^{\mathbf{A}}$ *denotes that* $\mathbf{E}$ *has oracle access to (the next-message functions of)* $\mathbf{A}$*.*

- *Succinctness. A SNARK is said to be succinct if the proof size is sublinear in the size of the instance and witness.*

Definition A.3 (Polynomial Commitment)**.** *A polynomial commitment scheme allows one party to commit to a polynomial. Subsequently, it enables them to convince another party that the polynomial, when evaluated at a particular point, results in a specific value. It consists of the following algorithms:*

- $\mathsf{pp} \leftarrow \mathsf{PC.setup}(1^\lambda, \mathcal{F})$*: Given the security parameter and a family of polynomials, output the public parameters* pp.
- $\mathsf{com} \leftarrow \mathsf{PC.commit}(\phi, \mathsf{pp})$*: On input a polynomial* ϕ *and the public parameter pp, output the commitment* $\mathcal{C}$.
- $(y, \pi) \leftarrow \mathsf{PC.open}(\phi, \mathsf{pp}, x)$*: On input the polynomial, the public parameters, and the point to be evaluated at, output* $y = \phi(x)$*, and the proof.*
- $\{0, 1\} \leftarrow \mathsf{PC.verify}(\mathsf{pp}, \mathsf{com}, x, y, \pi)$*: Verify the proof against the claim* $\phi(x) = y$*, output 1 if verification passes, otherwise 0.*

A polynomial commitment scheme satisfies the following properties:

- *Completeness. For any polynomial* $\phi \in \mathcal{F}$*, and any evaluation point* x*. The following probability holds:*

$$\Pr\left[\ \mathsf{PC.verify}(\mathsf{pp}, \mathsf{com}, x, y\pi) = 1 \ \middle|\ \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{PC.setup}(1^\lambda, \mathcal{F}), \\ \mathsf{com} \leftarrow \mathsf{PC.commit}(\phi, \mathsf{pp}), \\ (y, \pi) \leftarrow \mathsf{PC.open}(\phi, \mathsf{pp}, x), \end{array}\right] = 1$$

- *Knowledge soundness. For any PPT adversary* $\mathbf{A}$*, there exists an expected PPT extractor* $\mathbf{E}$ *that on given oracle access to* $\mathbf{A}$*, outputs the polynomial such that*

$$Pr\left[\ \begin{array}{c} \phi' \leftarrow \mathbf{E}^{\mathbf{A}}(\mathsf{pp}, \mathsf{com}, x, y, \pi), \\ \mathsf{PC.commit}(\phi') = \mathsf{com}, \\ \phi'(x) \neq y \end{array} \ \middle|\ \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{PC.setup}(1^\lambda, \mathcal{F}), \\ (\mathsf{com}, y, x, \pi) \leftarrow \mathbf{A}(\mathsf{pp}), \\ \mathsf{PC.verify}(\mathsf{pp}, \mathsf{com}, x, y, \pi) = 1 \end{array}\right] \leq \mathsf{negl}(\lambda)$$

B Deferred Proofs

B.1 Proof of Lemma 2.1

Intuitively, the lemma follows since for any vector $\mathbf{x} \in \mathbb{F}^n$ of weight w, in the worst case, every entry of $\mathbf{x}$ cancels with the next entry when applying the accumulation matrix $\mathbf{A}$ (i.e., the worst case is the "binary" case). More formally, fix any $w \geq 1$. First, consider w to be even, let $a \in \mathbb{F} \setminus \{0\}$ be any element and let $\mathbf{a}^* = (a, -a, \ldots, a, -a) \in \mathbb{F}^w$. Now, for any $0 \leq k \leq n - w$, define $\mathbf{x} = (\mathbf{0}^k, \mathbf{a}^*, \mathbf{0}^{n-w-k}) \in \mathbb{F}^n$, then $\mathsf{wt}(\mathbf{x}) = w$ and $\mathsf{wt}(\mathbf{xA}) = w/2$, as $\mathbf{y} = \mathbf{xA}$ is defined as $\mathbf{y}_i = \sum_{j=1}^{i} \mathbf{x}_j$. For odd w, let $w' = w - 1$ and set $\mathbf{a}^* = (a, -a, \ldots, a, -a) \in \mathbb{F}^{w'}$ as before for any non-zero a. Moreover, let $b \in \mathbb{F} \setminus \{0\}$ also be arbitrary. Then, for any $0 \leq k \leq n - w - 1$, define $\mathbf{x} = (\mathbf{0}^k, \mathbf{a}^*, \mathbf{0}^{n-w-k-1}, b) \in \mathbb{F}^n$. Again, we clearly have $\mathsf{wt}(\mathbf{x}) = w$ and $\mathsf{wt}(\mathbf{xA}) = w'/2 + 1 = \lceil w/2 \rceil$.

B.2 IOWE for Combinations

Recall that Theorem 4.2 gives us exactly the number of vectors $\mathbb{F}^N$ of weight w that map to a vector of weight h under the accumulation matrix. However, for our analysis we need to count the number weight w supports $T \subseteq [N]$ that map to a vector of weight h under the accumulation matrix. More formally, this corresponds to the following quantity:

$$\left|\left\{T \subseteq [N], |T| = w \colon \exists \mathbf{x} \in \mathbb{F}^N \text{ s.t. } \mathbf{x} \in \mathsf{supps}(N, w, T) \wedge \mathsf{wt}(\mathbf{xA}) = h\right\}\right|, \tag{14}$$

where $\mathbf{x} \in \mathsf{supps}(N, w, T)$ denotes that $\mathbf{x}$ has weight w and support $T \subseteq [N]$. Clearly, the quantity

$$\sum_{i=0}^{w-1} \binom{N-h}{\lfloor \frac{w-i}{2} \rfloor} \binom{h-1}{\lceil \frac{w-i}{2} \rceil} \binom{h - \lceil \frac{w-i}{2} \rceil}{i} \tag{15}$$

is an upper bound on the size of this set. However, it is a strict upper bound and is, in fact, over counting. This can be seen via a simple numerical calculation (e.g., using Sagemath) along with the following theorem.

Theorem B.1. *Let $N \in \mathbb{N}$ and $\mathbb{F}_q$ be a finite field.*

1. *For any $w \in [N]$, we have*

$$\sum_{h=1}^{N} A_{w,h}^{N,q} = \binom{N}{w}(q-1)^w.$$

2. *For any $h \in [N]$, we have*

$$\sum_{w=1}^{N} A_{w,h}^{N,q} = \binom{N}{h}(q-1)^h.$$

3. *We have*

$$\sum_{h=1}^{N} \sum_{w=1}^{N} A_{w,h}^{N,q} = q^N - 1.$$

Proof. Note that the accumulation matrix is a square rank N matrix. In particular, it is a bijection from $\mathbb{F}^N$ to $\mathbb{F}^N$. This readily implies the third summation. For the first summation, notice that for any fixed w there are exactly $\binom{N}{w}(q-1)^w$ vectors of $\mathbb{F}^N$ with Hamming weight w. Summing over all possible mappings of weight h vectors to weight w vectors in $\mathbb{F}^N$, because the accumulator matrix is a bijection, we must map to all vectors of weight w. The same argument holds for the second summation as well, completing the proof. $\square$

Using the above theorem, one can numerically compute $\sum_{h=1}^{N}$ Eq. (15) and compare it with $\binom{N}{w}$ for various values of w and find that Eq. (15) exceeds $\binom{N}{w}$. However, for our purposes, since we sum only for $h \leq \delta N$, this over counting does not stop us from obtaining our results. Note that any direct improvement to counting Eq. (14) will directly improve our upper bound.

B.3 Derivatives and Lower Bounds

In this section, we justify various bounds in our proof of Theorem 4.1.

Minimum of $(w-i)^{w-i}(4i)^i$. Taking the derivative of this function with respect to i, we have

$$\frac{d}{di}\left((w-i)^{w-i}(4i)^i\right) = (4i)^i \cdot (w-i)^{w-i} \cdot \ln\left(\frac{4i}{w-i}\right).$$

The above derivative has a single root at

$$\ln\left(\frac{4i}{w-i}\right) = 0 \implies \frac{4i}{w-i} = 1 \implies i = w/5.$$

Now evaluate the derivative first at $i = w/6$:

$$(5w/6)^{5w/6}(2w/3)^{w/6}\ln((4w/6)/(5w/6)) < 0$$

where the inequality follows since $(4w/6)/(5w/6) < 1$ and the other terms are positive. Similarly, evaluating the derivative at $i = w/4$, we have

$$(3w/4)^{3w/4}(w)^{w/4}\ln((w)/(3w/4)) > 0$$

where the inequality follows since $w/(3w/4) > 1$. Thus the minimum of our function is at $i = w/5$.

Decreasing summand. During one step of our analysis, we claim that the summand of

$$\sum_{r=1}^{\beta n/t}\left(\frac{eRN}{r}\right)^r (ge)^{crt} \cdot (crt) \cdot (\delta N - \lceil crt/2 \rceil)\,(2\delta N - crt)$$

is decreasing for $1 \leq r \leq \beta n/t$, $ge < 1$, and $c\gamma > 1$. Here, γ, β, c, and g are constants. To justify this claim, we need to show that

$$\left(\frac{eRN}{r}\right)^r (ge)^{crt}$$

is a decreasing function in r. Recalling that $t = \gamma \log(N)$, we can write

$$\begin{aligned}(ge)^{crt} &= (ge)^{rc\gamma\log(N)} = ((ge)^{\log(N^{\gamma c})})^r = ((2^{\log(ge)})^{\log(N^{\gamma c})})^r \\ &= (2^{\log(N^{c\gamma\log(ge)})})^r = (N^{c\gamma\log(ge)})^r.\end{aligned}$$

This gives us

$$\begin{aligned}\left(\frac{eRN}{r}\right)^r (ge)^{crt} &= \left(\frac{eRN}{r}\right)^r (N^{c\gamma\log(ge)}))^r \\ &= \left(\frac{eRN}{r} \cdot N^{c\gamma\log(ge)}\right)^r \\ &= \left(\frac{eRN^{1+c\gamma\log(ge)}}{r}\right)^r \\ &= \left(\frac{eR}{rN^{c\gamma\log(1/ge)-1}}\right)^r\end{aligned}$$

Thus this is decreasing in r as long as $c\gamma\log(1/ge) > 1$.

B.4 Proof of Lemma 3.7

To aid in the proof, we use the following lemma.

Lemma B.2 (Extended Piling Up Lemma)**.** *Let $\mathbb{F}_q$ be a finite field and let $p \in (0,1)$. Define $\alpha = 1/(q-1)$ Then for any $k \in \mathbb{N}$ and i.i.d. $X_1, \dots, X_k \leftarrow \mathsf{Ber}_p(\mathbb{F}_q)$ and $X := \sum_i X_i$, we have*

$$\Pr[X = 0] = \left(\alpha + (1 - p\cdot(1+\alpha))^k\right)/(1+\alpha). \tag{16}$$

Moreover, for any $a \in \mathbb{F}_q \setminus \{0\}$, we have

$$\Pr[X = a] = \left(1 - (1 - p\cdot(1+\alpha))^k\right)\cdot(\alpha/(1+\alpha)). \tag{17}$$

Proof. We proceed by induction on k. For $k = 1$, we have $X_1 \leftarrow \mathsf{Ber}_p(\mathbb{F})$. We know that $\Pr[X_1 = 0] = 1 - p$ with probability p, we have $X_1 \leftarrow U_{\mathbb{F}\setminus\{0\}}$; i.e., X_1 is sampled from the uniform distribution over non-zero elements of $\mathbb{F}$ with probability p. Thus, for any $a \neq 0$, we have $\Pr[X_1 = a] = \alpha p$. For the right hand side of Eq. (16), we have

$$\begin{aligned}\frac{1}{1+\alpha}\cdot(\alpha + (1 - p\cdot(1+\alpha))) &= \frac{\alpha + 1 - p\cdot(1+\alpha)}{1+\alpha}\\ &= \frac{\alpha+1}{1+\alpha} - \frac{p\cdot(1+\alpha)}{1+\alpha}\\ &= 1-p.\end{aligned}$$

For the right hand side of Eq. (17), we have

$$\begin{aligned}\frac{\alpha}{1+\alpha}\cdot(1 - (1 - p\cdot(1+\alpha))) &= \frac{\alpha - \alpha + \alpha\cdot p\cdot(1+\alpha)}{1+\alpha}\\ &= \alpha\cdot p.\end{aligned}$$

For the induction step, assume that Eqs. (16) to (17) hold for $k-1$ random variables

$$X_1, \dots, X_{k-1} \leftarrow \mathsf{Ber}_p(\mathbb{F}).$$

Then for k random variables $X_1, \dots, X_{k-1}, X_k \leftarrow \mathsf{Ber}_p(\mathbb{F})$, it suffices to consider the probability that $\sum_i X_i = 0$.

$$\Pr\left[\sum_{i\in[k]} X_i = 0\right] \tag{18}$$

$$\begin{aligned}&= \Pr\left[\sum_{i\in[k-1]} X_i + X_k = 0\right]\\ &= \Pr\left[\sum_{i\in[k-1]} X_i = 0 \wedge X_k = 0\right] + \Pr\left[\sum_{i\in[k-1]} X_i = a \neq 0 \wedge X_k = -a\right]\\ &= \Pr\left[\sum_{i\in[k-1]} X_i = 0\right]\cdot\Pr[X_k = 0] + \Pr\left[\sum_{i\in[k-1]} X_i = a\right]\cdot\Pr[X_k = -a]\end{aligned} \tag{19}$$

where the second equality holds since the random variables are taken over a finite field $\mathbb{F}$ and because the events are mutually exclusive, and the third equality holds because the random variables are i.i.d. sampled.

Let $\phi = (1 - p \cdot (1+\alpha))$ for ease of presentation. By our induction hypothesis, the first product becomes:

$$\Pr\left[\sum_{i\in[k-1]} X_i = 0\right] \cdot \Pr\left[X_k = 0\right] = \frac{1}{1+\alpha} \cdot \left(\alpha + \phi^{k-1}\right) \cdot (1-p),$$

while the second product becomes:

$$\begin{aligned}\Pr\left[\sum_{i\in[k-1]} X_i = a\right] \cdot \Pr\left[X_k = -a\right] &= \left(1 - \Pr\left[\sum_{i\in[k-1]} X_i = 0\right]\right) \cdot \alpha p \\ &= \left(1 - \frac{\alpha + \phi^{k-1}}{1+\alpha}\right) \cdot \alpha p.\end{aligned}$$

Going back to Eq. (19) and setting $B = (\alpha + \phi^{k-1})/(1+\alpha)$, we have

$$\begin{aligned}\text{Eq. (19)} &= B \cdot (1-p) + (1-B) \cdot \alpha p \\ &= B - Bp + \alpha p - B\alpha p \\ &= B + \alpha p - Bp(1+\alpha) \\ &= \alpha p + B(1 - p \cdot (1+\alpha)) \\ &= \alpha p + B \cdot \phi \\ &= \alpha p + \left(\frac{\alpha + \phi^{k-1}}{1+\alpha}\right) \cdot \phi \\ &= \alpha p + \frac{\alpha \cdot \phi}{1+\alpha} + \frac{\phi^k}{1+\alpha} \\ &= \frac{\alpha p(1+\alpha)}{1+\alpha} + \frac{\alpha - \alpha p(1+\alpha)}{1+\alpha} + \frac{\phi^k}{1+\alpha} \\ &= \frac{\alpha + \phi^k}{1+\alpha} \\ &= \frac{1}{1+\alpha} \cdot \left(\alpha + (1 - p \cdot (1+\alpha))^k\right).\end{aligned}$$

This completes the proof for both cases as $\Pr[\sum_i X_i \neq 0] = 1 - \Pr[\sum_i X_i = 0]$. ☐

Lemma B.2 immediately yields the following corollary.

Corollary B.3. *Let $\mathbb{F}_q$ be a finite field and let $p \in (0, (q-1)/q)$. Then for any $k \in \mathbb{N}$, i.i.d. $X_1, \ldots, X_k \leftarrow \mathsf{Ber}_p(\mathbb{F}_q)$, and $X := \sum_i X_i$, we have $X \equiv \mathsf{Ber}_{\rho(k)}(\mathbb{F}_q)$, where*

$$\rho(k) := \frac{1}{1+\alpha}\left(1 - \left[1 - p \cdot (1+\alpha)\right]^r\right)$$

and $\alpha = 1/(q-1)$.

Proof. By definition, $\mathsf{Ber}_{\rho(k)}(\mathbb{F})$ samples an element $a \xleftarrow{\$} \mathbb{F} \setminus \{0\}$ with probability $\rho(k)$ and outputs 0 with probability $1 - \rho(k)$. Thus, for $Y \leftarrow \mathsf{Ber}_{\rho(k)}(\mathbb{F})$, $Y = a$ for $a \neq 0$ with probability $\alpha \cdot \rho(k)$ and $Y = 0$ with probability $1 - \rho(k)$. By definition of $\rho(k)$, we see that

$$\begin{aligned}\alpha \cdot \rho(k) &= \frac{\alpha}{1+\alpha} \cdot (1 - (1 - p \cdot (1+\alpha))^k) = \text{Eq. (17)}; \\ (1 - \rho(k)) &= 1 - \frac{1}{1+\alpha} \cdot \left(1 - (1 - p \cdot (1+\alpha))^k\right) \\ &= \frac{1}{1+\alpha} \cdot \left(1 + \alpha - 1 + (1 - p \cdot (1+\alpha))^k\right) \\ &= \frac{1}{1+\alpha} \cdot (\alpha + (1 - p \cdot (1+\alpha))^n) = \text{Eq. (16)}.\end{aligned}$$

This completes the proof. □

Proof of Lemma 3.7. Lemma B.2 and Corollary B.3 imply that for any $\mathbf{x} \in \mathbb{F}_q^n$ of weight r, the vector $\mathbf{y} = \mathbf{x}\mathbf{E}_2$ for Bernoulli matrix $\mathbf{E}_2$ has the following property: every $\mathbf{y}_i$ is a sum of r i.i.d. random variables sampled from $\mathsf{Ber}_p(\mathbb{F}_q)$ (exactly characterized by Lemma B.2), and that each $\mathbf{y}_i$ is distributed as another Bernoulli random variable (Corollary B.3). In particular, let $\alpha = 1/(q-1)$ and define $\rho(r)$ as

$$\rho(r) := \left(1 - \left[1 - p \cdot (1+\alpha)\right]^r\right)/(1+\alpha).$$

Then every $\mathbf{y}_i$ is distributed as $\mathsf{Ber}_{\rho(r)}(\mathbb{F}_q)$. Together, this implies that $\mathbf{x}\mathbf{E}_2$ is distributed as a binomial distribution; in particular, we have

$$\Pr_{\mathbf{E}_2}[\mathsf{wt}(\mathbf{x}\mathbf{E}_2) = w \mid \mathsf{wt}(\mathbf{x}) = r] = \binom{N}{w}\rho(r)^w \cdot (1-\rho(r))^{N-w}.$$

To proceed with the analysis, we wish to argue that except with negligible probability over $\mathbf{E}_2$, any vector of weight $r \geq \beta n/t$ maps to a vector of weight at least $crt = rc\gamma\log(N)$. In particular, we want to show that

$$\Pr_{\mathbf{E}_2}[\mathsf{wt}(\mathbf{x}\mathbf{E}_2) < rc\gamma\log(N) \mid \mathsf{wt}(\mathbf{x}) \geq \beta n/t] \leq \mathsf{negl}(N).$$

For ease of notation, let $\alpha = 1/(1+q)$ and let $\phi = 1 - p(q/(q-1)) = 1 - p(1+\alpha)$. Then, with this notation, we have $\rho(r) = \frac{1}{1+\alpha}(1-\phi^r)$. Since $\mathbf{x}\mathbf{E}_2$ follows the binomial distribution, we know that $\mathbb{E}[\mathsf{wt}(\mathbf{x}\mathbf{E}_2) \mid \mathsf{wt}(\mathbf{x}) = r] = \rho(r)N$. Let $T = \rho(r)N - rct$ and $\mu = \rho(r)N$. Then, by Chernoff-Hoeffding, we have

$$\begin{aligned}\Pr_{\mathbf{E}_2}[\mathsf{wt}(\mathbf{x}\mathbf{E}_2) < rc\gamma\log(N) \mid \mathsf{wt}(\mathbf{x}) = r] &= \Pr_{\mathbf{E}_2}[\mathsf{wt}(\mathbf{x}\mathbf{E}_2) < \mu - T \mid \mathsf{wt}(\mathbf{x}) = r] \\ &\leq \exp\left(-2T^2/N\right).\end{aligned}$$

To upper bound this expression, we lower bound the parameter $T = \rho(r)N - rct$. We begin by lower bounding the term $\rho(r) = (1/(1+\alpha))(1-\phi^r)$. We assume that $p < (q-1)/q$ so that $\phi < 1$. With this in mind, $\phi < 1$ implies that ϕ^r is a decreasing function in r, so $1 - \phi^r$ is increasing with r. Thus the minimum of $\rho(r)$ is achieved at the minimum of r; in this case, $\beta n/t \leq r \leq n$ and thus

$$\rho(r) \geq \rho(\beta n/t) = \frac{1}{1+\alpha}\left(1 - \phi^{\beta n/t}\right).$$

To continue to lower bound this term, we upper bound the term $\phi^{\beta n/t}$. Recall that $p = t/N$ and $n = RN$ for $R < 1$, then we have

$$\phi^{\beta n/t} = (1-p(1+\alpha))^{\frac{\beta RN}{t}} = (1-p(1+\alpha))^{\frac{\beta R}{p}} = (1-p(1+\alpha))^{\frac{\beta R(1+\alpha)}{p(1+\alpha)}}.$$

Now, we use the inequality $(1-\frac{1}{x})^x \leq 1/e$ for all $x \geq 1$. Set $x := \frac{1}{p(1+\alpha)}$, then

$$\begin{aligned}(1-p(1+\alpha))^{\frac{\beta R(1+\alpha)}{p(1+\alpha)}} &= \left(1 - \frac{1}{x}\right)^{\beta R(1+\alpha)\cdot x} = \left(\left(1-\frac{1}{x}\right)^x\right)^{\beta R(1+\alpha)} \\ &\leq \left(\frac{1}{e}\right)^{\beta R(1+\alpha)} = \exp(-\beta R(1+\alpha)).\end{aligned}$$

Finally, note that $\alpha = 1/(q-1)$ and is a decreasing function in the field size q. This implies that $1 \leq 1 + \alpha < 2$, which means $\exp(-\beta R(1+\alpha)) \leq \exp(-\beta R)$ and $1/(1+\alpha) > 1/2$. Together, this implies that $\rho(r) \geq (1 - \exp(-\beta R))/2$.

By our assumptions, R and β are constants, so $\varphi := (1 - \exp(-\beta R))/2$ is a constant. Moving back to the parameter T, we have shown that

$$\begin{aligned} T = \rho(r)N - crt &> \frac{1 - \exp(-\beta R)}{2} \cdot N - crt \\ &\geq \varphi N - cRNt \\ &= N\left(\varphi - \frac{cR\gamma \log(N)}{N}\right). \end{aligned}$$

Going back to our original bound, we now have

$$\begin{aligned} \exp\left(-\frac{2T^2}{N}\right) &\leq \exp\left(-\frac{2}{N}\left(N\left(\varphi - \frac{cR\gamma\log(N)}{N}\right)\right)^2\right) \\ &= \exp\left(-2N\left(\varphi - \frac{cR\gamma\log(N)}{N}\right)^2\right). \end{aligned}$$

Notice that $\log(N)/N$ is a decreasing function of N and thus $\varphi - cR\gamma\log(N)/N$ is an increasing function in N. Thus $\varphi - cR\gamma\log(N)/N \geq \varphi$, resulting in

$$\begin{aligned} \exp\left(-2N\left(\varphi - \frac{cR\gamma\log(N)}{N}\right)^2\right) &\leq \exp\left(-2N(\varphi)^2\right)\right) \\ &= \exp\left(-N\frac{(1-\exp(-\beta R))^2}{2}\right) \end{aligned}$$

which is negligible in N for constants β and R. $\square$