HammR: A ZKP Protocol for Fixed Hamming-Weight Restricted-Entry Vectors

Felice Manganiello and Freeman Slaughter
{manganm, fslaugh} @ clemson.edu

Clemson University

Abstract. In this paper, we introduce HammR, a generic Zero-Knowledge Proof (ZKP) protocol demonstrating knowledge of a secret vector that has a fixed Hamming weight with entries taken from a shifted multiplicative group. As special cases, we are able to directly apply this protocol to restricted vectors and to rank-1 vectors, which are vectors with entries that lie in a dimension one subspace of $\mathbb{F}_q$. We show that these proofs can be batched with low computational overhead, and further prove that this general framework is complete, sound, and zero-knowledge, thus truly a genuine ZKP. Finally, we present applications of HammR to various Syndrome Decoding Problems, including the Regular and Restricted SDPs, as well as other implementations such as lookup instances, proof of proximity, and electronic voting protocols.

1 Introduction

The history of ZKPs is rich and complex, full of profound and compelling results. The first zero-knowledge identification scheme (ZKID) was introduced in [56], demonstrating knowledge of quadratic residuosity without revealing any other information about the value. Later, [22] defined the concept of a *non-interactive* zero-knowledge protocol in the common reference string model [29], which has since become a standard framework. Similarly, ZKID's can be turned into non-interactive signature schemes via the Fiat-Shamir heuristic [48,68,42], which has also become standard practice. [69] proposed the first sumcheck protocol, which permitted users to reduce a sum of multivariate polynomial evaluations into just a single evaluation at a randomly selected point. The GKR protocol [55] leverages the sumcheck protocol so that a user can prove knowledge of the output of an arithmetic circuit. In 2010, the KZG scheme [64] was introduced to commit to polynomials using bilinear pairings and formed a building block for many important protocols like Pinocchio [79], Groth16 [57], and Plonk [49]. However, a drawback of the KZG scheme was that it required a trusted setup - this was addressed in Bulletproofs [27], in which a user proves knowledge of an opening to a Pedersen commitment that satisfies a certain inner product relation. Other ways around trusted setup were introduced in Marlin [33] and Sonic [70]. As one can see, there has been shocking growth in the number of zero-knowledge proof protocols, with Ben-Sasson even going so far as to describe it as a Cambrian explosion [85].

The first code-based zero-knowledge scheme was Stern's work [87], which was soon after improved [86], and introduced the idea of a ZKID from a code-based cryptographic framework. Many code-based zero-knowledge schemes, which enjoy protection against a quantum adversary using Shor's algorithm [84,83] or Grover's algorithm [59], inherent this protection from McEliece [72], which was the earliest code-based cryptosystem. McEliece has been improved upon in various ways through the decades [31,51,89], including being the root of many modern cryptographic proposals. For instance, Classic McEliece [1] was a NIST post-quantum standardization [76] candidate, BIKE [2] and HQC [73] are both modern examples of code-based key encapsulation methods, and CROSS [9] is a contemporary code-based digital signature scheme.

Two other code-based signature schemes based on the Hamming metric include Wave [41] and McEliece [38], though Hamming is not the only suitable choice. Signature schemes from other metrics include proposals like [50,3] which use the rank metric, and also [10,9] under the restricted metric. There has been a flurry of recent works introducing and improving signature schemes, with some notable examples that include more novel problems, such as [8,34,82]. For a more in-depth history of code-based cryptography, we reference [78,43,65].

Our motivation for this paper is to explore the productive intersection between the strict privacy of zero-knowledge protocols and the errors in coding theory, specifically focusing on vectors with a certain Hamming weight and bounded entries.

The paper is organized as follows: Section 3 opens this paper by introducing some notation and presenting the Hamming and rank weights, the two most common choices of metrics in code-based cryptography, which we employ in the HammR protocol. Subsection 3.1 defines sets of bounded vectors and then shows that rank-1 vectors and restricted vectors are special cases of these sets. Subsection 3.2 introduces probabilistic arguments to characterize when an inner product is 0 for random input vectors, which will be used in the HammR protocol to argue probabilistically that a vector is in one of the bounded sets defined earlier.

Section 4 presents the HammR protocol, which is a zero-knowledge proof protocol demonstrating that a vector is contained in one of the previous bounded sets. Subsection 4.1 outlines the necessary pieces of the protocol, showcasing how each one is relevant to the protocol as a whole. Subsection 4.2 then displays how these pieces are arranged together to give a single round of HammR, as well as introduces subprotocols for precomputation and commitments. Subsection 4.3 shows how to batch multiple instances of HammR into a single round, resulting in increased efficiency and lower computational overhead. Subsection 4.4 proves that the aggregated protocol is a ZKP: that it is complete, sound, and zero-knowledge. This then shows that the single round HammR is itself a ZKP. Finally, Subsection 4.5 applies the Fiat-Shamir heuristic to HammR to obtain a non-interactive ZKP, where we are careful to avoid subtle pitfalls such as the weak Fiat-Shamir transform.

Section 5 presents important applications of the HammR ZKP protocol. Subsection 5.1 introduces some well-known syndrome decoding problems which we use

to illustrate HammR, such as the Restricted and Regular, as well as two novel NP-Complete problems: the Generic-Error Regular Syndrome Decoding Problem and the Rank-1 Regular Syndrome Decoding Problem. We demonstrate how HammR can be used to zero-knowledge prove that a vector satisfies the Hamming weight and entry conditions of these problems. Subsection 5.2 highlights other implementations in which HammR may be utilized, such as lookup instances, proof of proximity, and electronic voting protocols. Finally, Section 6 includes concluding remarks.

2 Technical Overview

We introduce HammR, a sigma protocol zero-knowledge proof for demonstrating that a committed vector $\mathbf{v} \in \mathbb{F}_q^n$ has Hamming weight t and entries in a bounded subset $\mathcal{B}$. We leverage a Bulletproofs-style verification technique with inner product arguments to demonstrate a proof of the relation

$$K = h^{\gamma} g^{t} \text{ and } \mathbf{v} \in \mathcal{B},$$

where $\gamma \in \mathbb{F}_q^*$ is the blinding factor for commitment K. Our proof takes advantage of the fact that there exists a projection $\pi : \mathcal{B} \to \{0,1\}^n$, which will preserve the Hamming weight of $\mathbf{v}$, which is essential for our proof. As an example, to demonstrate in a zero-knowledge manner that $\mathrm{wt}(\mathbf{v}) = t$, it is enough to show that

$$\langle \mathbb{1}, \pi(\mathbf{v}) \rangle = t,$$

reducing the weight condition to an inner product proof without revealing the entries of $\mathbf{v}$. We prove that this scheme can be efficiently batched, providing the simultaneous proof of multiple instances, then that the batched protocol is a ZKP - that it is complete, sound, and zero-knowledge. Furthermore, this protocol can be applied to a number of syndrome decoding problems, exhibiting knowledge of a valid vector adhering to the weight and entry conditions while remaining independent of the specific syndrome equations.

3 Sets of Bounded Vectors and Their Characterization

In this section, we recall multiple standard weights used in code-based cryptography and computationally characterize sets of bounded vectors, which will be the starting point of our protocols. We begin by introducing some notation and well-known weights.

Definition 1. *Let $\mathbb{F}_p$ denote the field of p elements, where p is prime. We also denote $\mathbb{F}_q$ to be the field of $q = p^{\ell}$ elements, where q is a prime power. $\mathbb{F}_q^n$ denotes the set of all n-length vectors with entries in $\mathbb{F}_q$. We also write $\mathbb{F}_q^*$ to denote $\mathbb{F}_q \setminus \{0\}$.*

Definition 2. *Let $\mathbb{1}$ denote the vector of all 1's. Similarly, we let $\mathbf{0}$ denote the vector of all 0's. The length should be clear, but when needed, we write $\mathbb{1}^n$.*

Though the HammR protocol relies on some the existence of a group of order p^ℓ, where p is prime. We note here the existence, then defer the rest of the group theory to Appendix B.

Definition 3. *Let $\mathbb{G}_p$ denote a cyclic group with prime p order, and let $\mathbb{G}$ be a cyclic group of prime power order p^ℓ. If $\mathbb{G}_1$ is a subgroup of $\mathbb{G}_2$, we will denote this with $\mathbb{G}_1 \leq \mathbb{G}_2$.*

Let us introduce is two weights: the Hamming weight and the rank weight.

Definition 4 (Hamming weight). *Given $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$, the Hamming weight of $\mathbf{x}$ is*

$$\mathrm{wt}(\mathbf{x}) = |\{i = 1, \ldots, n \mid x_i \neq 0\}|.$$

We now introduce the rank weight over a subfield of $\mathbb{F}_q$. Let $\mathbb{F}_{p^i}$ be a subfield of $\mathbb{F}_q$, $r = \frac{\ell}{i}$, and $\{y_1, \ldots, y_r\} \subseteq \mathbb{F}_q$ be a basis of $\mathbb{F}_q$ over $\mathbb{F}_{p^i}$, then there exists an isomorphism between $\mathbb{F}_q$ and $\mathbb{F}_{p^r}$ that lifts an $x \in \mathbb{F}_q$ to $(x_1, \ldots, x_r)^\top \in \mathbb{F}_{p^r}$. This isomorphism can be generalized to elements of $\mathbb{F}_q^n$ to obtain a matrix over $\mathbb{F}_{p^i}$, meaning that $\mathbf{x} \in \mathbb{F}_q^n$ lifts to the matrix

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{r,1} & x_{r,2} & \cdots & x_{r,n} \end{pmatrix} \in \mathbb{F}_{p^i}^{r \times n}.$$

Definition 5 (Rank weight). *Given a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$, the rank weight of $\mathbf{x}$ over $\mathbb{F}_{p^i}$ is*

$$\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x}) = \mathrm{rank} \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{r,1} & x_{r,2} & \cdots & x_{r,n} \end{pmatrix},$$

where we say that a vector $\mathbf{x} \in \mathbb{F}_q^n$ is rank-1 if $\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x}) = 1$.

The definition of $\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x})$ uses the matrix representation of $\mathbf{x}$, which requires the choice of a basis for $\mathbb{F}_q$. Nonetheless, the $\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x})$ is an invariant independent of the choice of the basis.

3.1 Sets of Bounded Vectors

We are now ready to define and computationally characterize the sets of bounded and restricted vectors of interest in our research. The general set of vectors we consider are vectors with fixed Hamming weight and restricted entries that belong to a translated subgroup of $\mathbb{F}_q^*$.

Definition 6. *Let $\mathcal{G}$ be a subgroup of $\mathbb{F}_q^*$ of order ω, and $\lambda \in \mathbb{F}_q^*$, we define the set of fixed Hamming weight t and restricted entries as*

$$\mathcal{B}_{\lambda,\omega}^t = \left\{\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n \mid \mathrm{wt}(\mathbf{x}) = t,\ \lambda x_i \in \mathcal{G} \cup \{0\} \text{ for } i \in \{1, \ldots, n\}\right\}.$$

This definition is generic and allows us to represent vectors used as errors in different Syndrome Decoding Problems as elements of $\mathcal{B}_{\lambda,\omega}^t$ for a specific choice of parameters λ and ω, which uniquely define $\mathcal{G}$.

Rank-1 Vectors. It is possible to show the relation with rank weights, meaning that all sets of rank-1 vectors are $\mathcal{B}_{\lambda,\omega}^t$ sets.

Lemma 1. *Let $\mathbb{F}_{p^i}$ be a subfield of $\mathbb{F}_q$ and $\mathbf{x} \in \mathbb{F}_q^n$, Then*

$$\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x}) = 1 \text{ if and only if } \mathbf{x} \in \mathcal{B}_{\lambda,p^i-1}^t \text{ for some } \lambda \in \mathbb{F}_q^*.$$

Proof. From [81, Lemma 10], we have that a vector $\mathbf{x} \in \mathbb{F}_q^n$ with $\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x}) = t$ has a decomposition into $\mathbf{x} = \boldsymbol{\eta}\mathbf{y}$, where $\boldsymbol{\eta} \in (\mathbb{F}_q^*)^t$ and $\mathbf{y} \in \mathbb{F}_{p^i}^{t\times n}$. Since by hypothesis $\mathbf{x}$ is a rank-1 vector over $\mathbb{F}_{p^i}$, then $t = 1$, and we have that $\eta \in \mathbb{F}_q^*$ and $\mathbf{y} \in \mathbb{F}_{p^i}^n$. Then $\mathbf{x} \in \mathcal{B}_{\lambda,p^i-1}^t$, with $\lambda = \eta^{-1}$, using $\mathcal{G} = \mathbb{F}_{p^i}^*$.
Conversely, if $\mathbf{x} \in \mathcal{B}_{\lambda,p^i-1}^t$, then for λ being the inverse of the first non-zero entry of $\mathbf{x}$, we have that $\lambda\mathbf{x} \in (\mathcal{G} \cup \{0\})^n$ where $\mathcal{G}$ has order $p^i - 1$. As a consequence, $\mathcal{G}$ corresponds to $\mathbb{F}_{p^i}^*$, meaning that $\mathrm{rank}_{\mathbb{F}_{p^i}}(\mathbf{x}) = 1$ by [81, Lemma 10].

Restricted Vectors. Restricted vectors were defined in [10], and a modification of these vectors was implemented in CROSS [9] as error vectors.

Lemma 2. *Let $\mathcal{G}$ be a subgroup of $\mathbb{F}_q^*$ with order ω and $\mathbf{v} \in \mathbb{F}_q^n$ with $\mathrm{wt}(\mathbf{v}) = t$. Then*

$$\mathbf{v} \in (\mathcal{G} \cup \{0\})^n \text{ if and only if } \mathbf{v} \in \mathcal{B}_{\lambda,\omega}^t \text{ with } \lambda \in \mathcal{G}.$$

This lemma is a direct consequence of the fact that if $\lambda \in \mathcal{G}$, then all the operations are restricted to the subgroup.

Remark 1. It is important to note that restricted vectors and rank-1 vectors do not coincide; indeed, it can be proven that rank-1 vectors and restricted vectors are equivalent if and only if $\mathcal{G} = \mathbb{F}_{p^i}^*$ for a subfield $\mathbb{F}_{p^i}$ in $\mathbb{F}_q$ and $\lambda \in \mathcal{G}$.

We are ready to computationally characterize the sets $\mathcal{B}_{\lambda,\omega}^t$. Before proceeding, we need to introduce some notation: we denote by $\star$ the Schur (or Hadamard) product over $\mathbb{F}_q^n$. Precisely, we mean that the product is taken entrywise, so for $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$,

$$\mathbf{x} \star \mathbf{y} = (x_1y_1, x_2y_2, \ldots, x_ny_n) \in \mathbb{F}_q^n.$$

To take the Schur product of $\mathbf{x}$ with itself s times, we write $\mathbf{x}^s = (x_1^s, \ldots, x_n^s)$.

Definition 7. *We denote the (weight-preserving) projection of $\mathcal{B}^t_{\lambda,\omega}$ on $\{0,1\}$ with the following map*

$$\begin{aligned}\pi : \mathcal{B}^t_{\lambda,\omega} &\to \{0,1\}^n \\ \mathbf{v} &\mapsto (\lambda\mathbf{v})^\omega.\end{aligned}$$

Theorem 1. *Given a vector $\mathbf{v} \in \mathbb{F}_q^n$, let $\mathbf{w} = (\lambda\mathbf{v})^\omega$. Then*

$$\mathbf{v} \in \mathcal{B}^t_{\lambda,\omega} \text{ if and only if } \mathbf{w} \in \{0,1\}^n.$$

Furthermore, $\mathrm{wt}(\mathbf{w}) = \mathrm{wt}(\mathbf{v})$.

Proof. The theorem follows from the fact that

$$v \in \mathcal{G} \text{ if and only if } v^{|\mathcal{G}|} = v^\omega = 1.$$

Consequently, the order ω uniquely defines the subgroup $\mathcal{G}$ of $\mathbb{F}_q^*$.

In the next section we focus on providing probabilistic arguments that will allow us to create the protocols.

3.2 Probabilistic Arguments for Sets of Bounded Vectors

Theorem 1 characterizes the set of bounded vectors, but it does not permit us to build a ZKP just yet. In this section we introduce some probabilistic arguments that allow us to motivate the challenges of the protocols. Traditionally for inner product arguments, one applies the Schwartz-Zippel lemma to obtain a probabilistic bound, then works backwards to determine how many test vectors $\mathbf{y}$ are sufficient to be convinced of an inner product relation below some probability threshold. In our case, we can in fact speak with exactitude using the following probabilistic argument.

We start with some preliminaries. We denote by $x \xleftarrow{\$} X$ when an element $x \in X$ is selected uniformly at random from the set X.

Definition 8. *A function $f : \mathbb{N} \to \mathcal{R}$ is a negligible function if for every $n \in \mathbb{N}$ there exists an $N \in \mathbb{N}$ such that for all $x > N$ we have*

$$|f(x)| < x^{-n}.$$

Equivalently, $|f(x)|$ is bounded by the reciprocal of any polynomial in x, for x large enough.

Definition 9. *Given a random variable $\mathcal{E}$ and events $E, (E_s)_{s\in\mathbb{N}} \subseteq \mathcal{E}$ such that $\lim_{s\to\infty} E_s = E$, we say the event E occurs with high probability (w.h.p) when $\lim_{s\to\infty} Pr[E_s] = 1$.*

Let $X_{\mathbf{y}}$ denote $\langle \mathbf{u}, \mathbf{y} \rangle$ for $\mathbf{u}, \mathbf{y} \in \mathbb{F}_q^n$. We aim to characterize when the event $E = \{\mathbf{u} = \mathbf{0}\}$ occurs for any given $\mathbf{u}$ from an analysis of the events $E_s = \{X_{\mathbf{y}_s} = 0\}$ where $\mathbf{y}_s \in \mathbb{F}_q^n$ is chosen uniformly at random for $s \in \mathbb{N}$. Using the law of total probability with no restrictions on $\mathbf{u}$, we have for a randomly selected $\mathbf{y}$ that

$$\begin{aligned}\Pr[X_{\mathbf{y}} = 0] &= \Pr[X_{\mathbf{y}} = 0 \mid \mathbf{u} = \mathbf{0}] \cdot \Pr[\mathbf{u} = \mathbf{0}] + \Pr[X_{\mathbf{y}} = 0 \mid \mathbf{u} \neq \mathbf{0}] \cdot \Pr[\mathbf{u} \neq \mathbf{0}] \\ &= \frac{1}{q^n} + \frac{1}{q}\left(1 - \frac{1}{q^n}\right) \approx \frac{1}{q}.\end{aligned}$$

Hence, we apply Bayes theorem to see that

$$\begin{aligned}\Pr[\mathbf{u} = \mathbf{0} \mid X_{\mathbf{y}} = 0] &= \frac{\Pr[X_{\mathbf{y}} = 0 \mid \mathbf{u} = \mathbf{0}] \cdot \Pr[\mathbf{u} = \mathbf{0}]}{\Pr[X_{\mathbf{y}} = 0]} \\ &= \frac{1}{q^{n-1} + \frac{q-1}{q}} \approx \frac{1}{q^{n-1}}.\end{aligned}$$

This tells us that for an arbitrary $\mathbf{u} \in \mathbb{F}_q^n$, if $\langle \mathbf{u}, \mathbf{y} \rangle = 0$ for a randomly chosen $\mathbf{y}$, then we can say that $\mathbf{u}$ is actually the zero vector $\mathbf{0}$ with only a small probability. If this holds for a number of different $\mathbf{y}_i$ vectors which are selected uniformly at random, then one can argue that the probability $\mathbf{u} = \mathbf{0}$ should increase exponentially. The following lemmas make more rigorous this informal heuristic.

Lemma 3. *Let $\mathbf{u}, \mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{F}_q^n$, then $\mathbf{u} = \mathbf{0}$ if and only if $\mathbf{y}_1, \ldots, \mathbf{y}_n$ are $\mathbb{F}_q$-linearly independent and $\langle \mathbf{u}, \mathbf{y}_i \rangle = 0$ for $1 \leq i \leq n$.*

Lemma 4. *Let $\mathbf{u}, \mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{F}_q^n$ and let $k = \dim_{\mathbb{F}_q} \langle \mathbf{y}_1, \ldots, \mathbf{y}_n \rangle$ then the*

$$\Pr[\mathbf{u} = \mathbf{0} \mid X_{\mathbf{y}_1} = 0, \ldots, X_{\mathbf{y}_n} = 0] \approx \frac{1}{q^{n-k}}.$$

Proof. If $X_{\mathbf{y}_1} = 0, \ldots, X_{\mathbf{y}_n} = 0$, then $\mathbf{u}$ belongs to a subspace of dimension

$$n - \dim_{\mathbb{F}_q} \langle \mathbf{y}_1, \ldots, \mathbf{y}_n \rangle = n - k$$

over $\mathbb{F}_q$, so then the probability follows from the argument above and simple counting.

Let $(\mathbf{y}_i)_{i \in \mathbb{N}} \subseteq \mathbb{F}_q^n$ be a sequence of of vectors chosen uniformly at random. Then, by Lemma 3, one would be certain that $\mathbf{u} = \mathbf{0}$ as soon as $\langle \mathbf{u}, \mathbf{y}_i \rangle = 0$ for a sufficiently large amount of vectors $\mathbf{y}_i$. Worse-case scenario, one needs $q^{n-1} + 1$ this amount will ensure the existence of n $\mathbb{F}_q$-linearly independent vectors. From a probabilistic perspective though, for q large enough and $s \leq n$, if $y_1, \ldots, y_s \xleftarrow{\$} \mathbb{F}_q^n$ then, with high probability, the vectors form a $\mathbb{F}_q$-linearly independent set.
Since we want to ensure the use of linearly independent vectors, we are not choosing the vectors uniformly at random over the whole space $\mathbb{F}_q^n$, but we will restrict the choice to vectors of the form

$$\mathbf{y} = (y, y^2, \ldots, y^n) \in \mathbb{F}_q^n \tag{1}$$

for a given $y \in \mathbb{F}_q^*$. At this point, the choice of linearly independent vectors reduces to the choice of distinct elements. The following result follows from standard results of the Vandermonde matrix.

Lemma 5. *Let $s \leq n$ and $y_1, \ldots, y_s \in \mathbb{F}_q^*$, then the vectors $\mathbf{y}_i = (y_i, y_i^2, \ldots, y_i^n)$ for $1 \leq i \leq s$ are $\mathbb{F}_q$-linear independent if and only if $y_1, \ldots, y_s$ are distinct.*

Remark 2. For the remainder of the manuscript, when we write $\mathbf{y}$, we intend the vector defined as in (1).

4 The HammR Protocol

We now present our generic framework that permits us to prove a number of important characteristics about a vector over $\mathbb{F}_q$. We present this protocol as a ZKP interactive protocol, where a prover wishes to demonstrate that a secret vector $\mathbf{v}$ is in $\mathcal{B}^t_{\lambda,\omega}$, which will be done using $\mathbf{w} = (\lambda\mathbf{v})^\omega$ from Theorem 1. The verifier is allowed to pick challenges $y, x \in \mathbb{F}_q^*$, which are incorporated into the prover's calculations in order to demonstrate that they are being done honestly.

4.1 Key Components of the HammR Protocol

Now we want to characterize probabilistically whether a vector $\mathbf{w} \in \mathbb{F}_q^n$ has entries only in $\{0, 1\}$ using the arguments from Section 3.2. This will permit us to verify that $\mathbf{w}$ is indeed a projection.

Lemma 6. *Let $y_1, \ldots, y_s \in \mathbb{F}_q^*$ be distinct, and let $\mathbf{y}_1, \ldots, \mathbf{y}_s \in \mathbb{F}_q^n$ be the vectors defined by (1). Given $\mathbf{w} \in \mathbb{F}_q^n$, if*

$$\langle \mathbf{w} \star (\mathbb{1} - \mathbf{w}), \mathbf{y}_i \rangle = \mathbf{0}$$

for $1 \leq i \leq s$, then with high probability $\mathbf{w} \in \{0,1\}^n$. Once $s = n$, then $\mathbf{w} \in \{0,1\}^n$ with certainty.

This is a consequence of the probabilistic argument from Section 3.2.
Thus, we introduce the following three equations which will serve as the building blocks for HammR.

$$\langle \mathbb{1} - \mathbf{w}, \mathbb{1} \rangle = n - t \tag{2}$$

$$\langle (\mathbb{1} - \mathbf{w}) + (\lambda\mathbf{v})^\omega, \mathbf{y} \rangle = \langle \mathbb{1}, \mathbf{y} \rangle \tag{3}$$

$$\langle (\mathbb{1} - \mathbf{w}) \star (\lambda\mathbf{v})^\omega, \mathbf{y} \rangle = 0 \tag{4}$$

Theorem 2. *For λ and ω from Theorem 1, if a vector $\mathbf{v} \in \mathcal{B}^t_{\lambda,\omega}$, then it satisfies the above conditions (2), (3), and (4) for any choice of $\mathbf{y}$. Conversely, if $\mathbf{v}$ satisfies these conditions for enough randomly selected $\mathbf{y}$, then w.h.p. $\mathbf{v} \in \mathcal{B}^t_{\lambda,\omega}$.*

Proof. The forward direction is immediately verifiable. For the other direction, by Theorem 1, then (2) is equivalent to $\mathrm{wt}(\mathbf{v}) = t$. If (3) holds, then w.h.p. $\mathbf{w} = (\lambda\mathbf{v})^\omega$, demonstrating that the projection was taken honestly. With this in mind, finally (4) shows that w.h.p. $\mathbf{w} \in \{0,1\}^n$, so the verifier can be confident that $\mathbf{w}$ is actually a projection.

We remark that the above theorem holds for $\mathbf{y}$ taken uniformly at random, even though for our protocol we specify $\mathbf{y} = (y, y^2, \ldots, y^n) \in \mathbb{F}_q^n$ using $y \in \mathbb{F}_q^*$. Additionally, the w.h.p. can be improved by taking n test vectors, by appealing to the probabilistic argument from Section 3.2.
Instead of proving that conditions (2), (3), and (4) hold separately to demonstrate that $\mathbf{v} \in \mathcal{B}^t_{\lambda,\omega}$, they can be packaged into a single inner product condition. Using the indeterminate z, where z is selected by the verifier to ensure that the calculations are done honestly, we form

$$z^2\langle \mathbb{1} - \mathbf{w}, \mathbb{1}\rangle + z\langle (\mathbb{1} - \mathbf{w}) + (\lambda\mathbf{v})^\omega, \mathbf{y}\rangle + \langle (\mathbb{1} - \mathbf{w}) \star (\lambda\mathbf{v})^\omega, \mathbf{y}\rangle = z^2(n-t) + z\langle \mathbb{1}, \mathbf{y}\rangle$$

which we can then package. Akin to Bulletproofs [27], we add $z^2\langle \mathbb{1}, \mathbf{y}\rangle + z^3\langle \mathbb{1}, \mathbb{1}\rangle$ to both sides, then conglomerate the conditions to obtain

$$\langle (\mathbb{1} - \mathbf{w}) + z\mathbb{1}, y \star ((\lambda\mathbf{v})^\omega + z\mathbb{1}) + z^2\mathbb{1}\rangle = d_0,$$

where $d_0 = (z^2+z^3)n - z^2t + (z+z^2)\langle \mathbb{1}, \mathbf{y}\rangle$. Define $\varepsilon := (z^2+z^3)n + (z+z^2)\langle \mathbb{1}, \mathbf{y}\rangle$ so that $d_0 = \varepsilon - z^2t$. These values will be used in the protocol to assist the verifier in confirming the veracity of committed values using verifier-chosen information, in specific: $y, z \in \mathbb{F}_q^*$ in the precomputation step.
To turn this single inner product into the HammR protocol, the prover selects random obfuscating vectors $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{F}_q^n$ and fixes the following two polynomials in the indeterminate x:

$$\mathbf{p}_1(x) = (\mathbb{1} - \mathbf{w}) + z\mathbb{1} + \mathbf{s}_1 x \tag{5}$$

$$\mathbf{p}_2(x) = \mathbf{y} \star ((\lambda\mathbf{v})^\omega + z\mathbb{1} + \mathbf{s}_2 x) + z^2\mathbb{1}. \tag{6}$$

These polynomials serve to exhibit conditions (2), (3), and (4) from earlier, while ensuring that no information about $\mathbf{v}$ is leaked. Later, after these polynomials have been committed to, the verifier will communicate a bit $x_0 \in \mathbb{F}_q^*$ in the main portion of the protocol, and the prover will evaluate $\mathbf{p}_1(x)$ and $\mathbf{p}_2(x)$ at x_0, which will permit the verifier's calculations on the commitments to work out as purported.

4.2 Assembling HammR

We now define the following commitment subprotocols which will build up to HammR.
The scalar commit protocol in Algorithm 1 is simply the standard Pedersen commitment. The vector commit protocol is the vector version of the Pedersen commitment, the security of which we discuss in Appendix B. As both are

Scalar commit, **vector** commit, **and** $\mathsf{commit}_{\mathbf{y}^{-1}}$ **protocols**
Public: $g, h \in \mathbb{G},\ \mathbf{g}, \mathbf{h} \in \mathbb{G}^n$ **Private:** $v \in \mathbb{F}_q,\ \mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$
Commit to v by sampling $\gamma \xleftarrow{\$} \mathbb{F}_q$, then we define $\mathsf{commit}(\gamma, v) = h^\gamma g^v$.
Commit to $\mathbf{a}$ and $\mathbf{b}$ by sampling $\gamma \xleftarrow{\$} \mathbb{F}_q$, then we define $\mathsf{commit}(\gamma, \mathbf{a}, \mathbf{b}) = h^\gamma \mathbf{g}^{\mathbf{a}} \mathbf{h}^{\mathbf{b}}$.
For $\mathbf{y}^{-1} \in \mathbb{F}_q^n$, we set $\mathsf{commit}_{\mathbf{y}^{-1}}(\gamma, \mathbf{a}, \mathbf{b}) = h^\gamma \mathbf{g}^{\mathbf{a}} \mathbf{h}^{\mathbf{y}^{-1} \star \mathbf{b}}$.

Algorithm 1: The commitment protocols

Pedersen commitments, but it will be clear from the input which one is being referenced, we use the same name with no ambiguity. The instance $\mathsf{commit}_{\mathbf{y}^{-1}}$ is equivalent to commit, and will be used solely to improve readability; while not strictly necessary, it compresses the protocol. We note that in the $\mathsf{commit}_{\mathbf{y}^{-1}}$ protocol, $\mathbf{y}^{-1} = (y^{-1}, y^{-2}, y^{-3}, \ldots, y^{-n}) \in \mathbb{F}_q^n$.

HammR setup **Protocol**		
Public: $g, h \in \mathbb{G},\ \ \mathbf{g}, \mathbf{h} \in \mathbb{G}^n,\ \ t \in \mathbb{N}$ **Private:** $\mathbf{w}, \mathbf{v} \in \mathbb{F}_q^n,\ \ \lambda \in \mathbb{F}_q^*,\ \ \omega \in \mathbb{N}$		
PROVER		**VERIFIER**
Sample $\alpha, \rho, \gamma \xleftarrow{\$} \mathbb{F}_q$ Sample $\mathbf{s}_1, \mathbf{s}_2 \xleftarrow{\$} \mathbb{F}_q^n$ $C_1 = \mathsf{commit}(\alpha, \mathbb{1} - \mathbf{w}, (\lambda \mathbf{v})^\omega)$ $C_2 = \mathsf{commit}(\rho, \mathbf{s}_1, \mathbf{s}_2)$ $K = \mathsf{commit}(\gamma, t)$	$\xrightarrow{C_1,\ C_2,\ K}$	
	$\xleftarrow{y,\ z}$	Sample $y, z \xleftarrow{\$} \mathbb{F}_q^\star$ Set $\varepsilon = (z^2 + z^3)n + (z + z^2)\langle \mathbb{1}, \mathbf{y} \rangle$

Algorithm 2: the setup algorithm for HammR

Algorithm 2 is the HammR setup protocol, and serves to precompute the values that will be used in the initial steps of Algorithm 3, such as commitments to the secret vectors.
We now present in Algorithm 3 the HammR protocol for bounded vectors, but defer the security proofs until after introducing the batching protocol, as proving the security of the aggregation is sufficient.

4.3 Aggregating Proofs

We prove here that batching m proofs can be done more efficiently than simply repeating the protocol m times. We begin by batching these instances, then in

The HammR ZKP Protocol		
Public: $q, n \in \mathbb{N}, \quad g, h \in \mathbb{G}, \quad \mathbf{g}, \mathbf{h} \in \mathbb{G}^n$ **Private:** $\mathbf{v}, \mathbf{w}, \mathbf{s}_1, \mathbf{s}_2 \in \mathbb{F}_q^n, \quad \lambda \in \mathbb{F}_q^*, \quad \omega \in \mathbb{N}$		
PROVER		**VERIFIER**
$\mathbf{p}_1(x) = (\mathbb{1} - \mathbf{w}) + z\mathbb{1} + \mathbf{s}_1 x$ $\hat{\mathbf{v}} = (\lambda \mathbf{v})^\omega + z\mathbb{1} + \mathbf{s}_2 x$ $\mathbf{p}_2(x) = \mathbf{y} \star \hat{\mathbf{v}} + z^2\mathbb{1}$ $d(x) = \langle \mathbf{p}_1(x), \mathbf{p}_2(x) \rangle$ $d(x) := d_2 x^2 + d_1 x + d_0$ Sample $\delta_1,\ \delta_2 \xleftarrow{\$} \mathbb{F}_q$		
$D_i = \mathsf{commit}(\delta_i, d_i)$	$\xrightarrow{D_i}$	
	$\xleftarrow{x_0}$	Sample $x_0 \xleftarrow{\$} \mathbb{F}_q^\star$
$\hat{\mathbf{p}}_i = \mathbf{p}_i(x_0)$ $\hat{d} = \langle \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2 \rangle$ $\delta_{x_0} = \delta_2 x_0^2 + \delta_1 x_0 - z^2 \gamma$		
$\mu = \rho x_0 + \alpha$	$\xrightarrow{\hat{d},\ \delta_{x_0},\ \hat{\mathbf{p}}_i,\ \mu}$	$\mathsf{commit}(\delta_{x_0}, \hat{d}) \stackrel{?}{=} D_2^{x_0^2} \cdot D_1^{x_0} \cdot K^{-z^2} \cdot g^{\varepsilon}$ $P := \mathsf{commit}_{\mathbf{y}^{-1}}(\mu, \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2)$ $P \stackrel{?}{=} C_2^{x_0} \cdot C_1 \cdot \mathbf{g}^{z\mathbb{1}} \cdot \mathbf{h}^{z\mathbb{1} + z^2 \mathbf{y}^{-1}}$ $\hat{d} \stackrel{?}{=} \langle \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2 \rangle$

Algorithm 3: the main portion of the HammR protocol

the next section we demonstrate that the aggregate protocol is a ZKP. This is sufficient to demonstrate that the protocol as presented is a ZKP, by taking $m = 1$.

Suppose we have m vectors $\mathbf{v}_i \in \mathbb{F}_q^n$, and we aim to prove that $\mathbf{v}_i \in \mathcal{B}_{\lambda,\omega}^t$ for $i = 1, \ldots, m$, where $\mathcal{B}_{\lambda,\omega}^t$ is taken from Theorem 1.

We form the weight distribution vector $\mathbf{t} \in \mathbb{N}^m$ from the individual t_i's, and the binding vector $\boldsymbol{\gamma} \in \mathbb{F}_q^m$ from the individual binding factors γ_i. Similarly, let $\mathbf{g}$ and $\mathbf{h}$ denote the vectors formed from the group elements g and h respectively. Let $\mathbf{K} \in \mathbb{G}^m$ be the vector formed from the individual commitments, so that $K_i = h^{\gamma_i} g^{t_i}$ for randomly selected blinding factors $\gamma_i \xleftarrow{\$} \mathbb{F}_q^*$. We also let $\mathbf{v} = (\mathbf{v}_1 \,||\, \ldots \,||\, \mathbf{v}_m) \in \mathbb{F}_q^{nm}$ denote the concatenation of the vectors for each individual proof.

So concretely, this framework admits a general proof protocol for the relation

$$\mathbf{K} = \mathbf{h}^{\boldsymbol{\gamma}} \mathbf{g}^{\mathbf{t}} \text{ and } \mathbf{v}_i \in \mathcal{B}_{\lambda,\omega}^t \text{ for } i = 1, \ldots, m \tag{7}$$

Similarly, the vectors $\mathbf{p}_1(x)$, $\mathbf{p}_2(x) \in \mathbb{F}_q^{nm}[x]$ will also be changed:

$$\mathbf{p}_1(x) = (\mathbb{1}^{nm} - \mathbf{w}) + z\mathbb{1}^{nm} + \mathbf{s}_1 x$$

$$\mathbf{p}_2(x) = \mathbf{y}^{nm} \star ((\lambda \mathbf{v})^\omega + z\mathbb{1}^{nm} + \mathbf{s}_2 x) + \sum_{i=1}^{m} z^{i+1} \left(\mathbf{0}^{n(i-1)} \,||\, \mathbb{1}^n \,||\, \mathbf{0}^{n(m-i)} \right).$$

We must also adjust the values for δ_{x_0} and ε to include the cross terms

$$\delta_{x_0} = \delta_2 x_0^2 + \delta_1 x_0 - \sum_{i=1}^{m} z^{i+1}\gamma_i$$
$$\varepsilon = (z+z^2)\langle \mathbb{1}^{nm}, \mathbf{y}^{nm}\rangle + (1+z)n\sum_{i=1}^{m} z^{i+1}.$$

Thus,

$$d_0 = \varepsilon - \sum_{i=1}^{m} z^{i+1}t_i = (z+z^2)\langle \mathbb{1}^{nm}, \mathbf{y}^{nm}\rangle + \sum_{i=1}^{m} z^{i+1}((1+z)n - t_i),$$

where the $m = 1$ case collapses to the values we saw previously in the single round HammR protocol: $d_0 = (z^2+z^3)n - z^2 t + (z+z^2)\langle \mathbb{1}, \mathbf{y}\rangle$.
Then the verifier's checks take the form

$$\mathsf{commit}(\delta_{x_0}, \hat{d}) \stackrel{?}{=} D_2^{x_0^2} \cdot D_1^{x_0} \cdot \mathbf{K}^{-z\mathbf{z}^m} \cdot g^{\varepsilon},$$
$$\mathsf{commit}_{\mathbf{y}^{-1}}(\mu, \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2) \stackrel{?}{=} C_2^{x_0} \cdot C_1 \cdot (\mathbf{g}\cdot\mathbf{h})^z \prod_{i=1}^{m} \mathbf{h}_{[n(i-1)+1,\, ni]}^{z^{i+1}\mathbf{y}^{-1}}$$

where $\mathbf{a}_{[i,\,j]}$ represents taking a slice of the vector $\mathbf{a}$ from indices i to j inclusive, so $\mathbf{a}_{[i,\,j]} = (a_i, a_{i+1}, \ldots, a_{j-1}, a_j)$.

4.4 Proof that Batched HammR is a ZKP

Here, we present a security proof of the batched protocol, which implies the same properties for the unbatched version. In order to be a full-fledged ZK scheme, Algorithm 3 is required that satisfy the following informal properties:

- Completeness: if a prover knows a valid vector, then they will always pass the verifier's checks.
- Soundness: if a prover does *not* know a valid vector, then they can only pass the verifier's checks with small probability.
- Zero-Knowledge: no party learns any pertinent information about the vector except that it is valid.

To see a more formal treatment of the definitions for these zero-knowledge terms we use here, we refer to Appendix C.
We begin by showing completeness:

Theorem 3. *Algorithm 3 is complete.*

Proof. Completeness follows from Theorem 2, since if an honest prover knows a valid vector $\mathbf{v}$, then they will successfully pass each of the checks (2), (3), and (4) in the protocol.

We now prove that Algorithm 3 enjoys witness extended emulation, which is a more robust notion of the traditional soundness. See Appendix C for a more complete treatment of these terms.

Theorem 4. *Algorithm 3 enjoys witness extended emulation.*

Proof. To demonstrate that this protocol achieves witness extended emulation, we construct an extractor algorithm $\mathcal{E}_{\mathsf{HammR}}$ that accepts mn distinct challenges for y, $m+2$ values for challenge z, then 3 values for x. Additionally this algorithm is able to call an efficient inner product extractor algorithm $\mathcal{E}_{\mathsf{ip}}$, which runs in time $\mathsf{poly}(n)$, and is required by Lemma 7, from Appendix C. At the end, it then produces either a valid witness or an instance that violates the discrete log assumption (Assumption 1 in Appendix A).
To start, $\mathcal{E}_{\mathsf{HammR}}$ will call $\mathcal{E}_{\mathsf{ip}}$, which accepts input $P = h^{\mu}\mathbf{g}^{\mathbf{p}_1}\mathbf{h}^{\mathbf{p}_2}$ and $\langle \mathbf{p}_1, \mathbf{p}_2 \rangle = d$, then returns witnesses $\mathbf{p}_1$ and $\mathbf{p}_2$. Using these witnesses, along with two valid transcripts obtained from the x challenges, we can compute linear combinations of the check for P and calculate α, $\mathbb{1} - \mathbf{w}$, and $(\lambda\mathbf{v})^{\omega}$ such that $C_1 = h^{\alpha}\mathbf{g}^{\mathbb{1}-\mathbf{w}}\mathbf{h}^{(\lambda\mathbf{v})^{\omega}}$, then also ρ, $\mathbf{s}_1$, and $\mathbf{s}_2$ such that $C_2 = h^{\rho}\mathbf{g}^{\mathbf{s}_1}\mathbf{h}^{\mathbf{s}_2}$. If the extractor ends up with a different set of presentations for C_1 or C_2, then they can violate the discrete logarithm assumption, so the values seen here must be the ones obtained under Assumption 1.
From this presentation, one can see that for all challenges (x, y, z), the following equality is true:

$$\mathbf{p}_1(x) = (\mathbb{1}^{nm} - \mathbf{w}) + z\mathbb{1}^{nm} + \mathbf{s}_1 x$$
$$\mathbf{p}_2(x) = \mathbf{y}^{nm} \star ((\lambda\mathbf{v})^{\omega} + z\mathbb{1}^{nm} + \mathbf{s}_2 x) + \sum_{i=1}^{m} z^{i+1}\mathbf{b}_{i,m,n}$$

where we introduce the notation $\mathbf{b}_{i,m,n} = (\mathbf{0}^{n(i-1)} \mid\mid \mathbb{1}^n \mid\mid \mathbf{0}^{n(m-i)})$ for brevity. Additionally, from now on, we write $\mathbf{y}$ to mean $\mathbf{y}^{nm}$ and $\mathbb{1}$ instead of $\mathbb{1}^{nm}$. If there exists a challenge set where these do not hold, then a violation of the discrete logarithm assumption has been found.
For a fixed pair of y and z, we take three distinct values of x and use linear algebra to determine d_1 and δ_1 such that $D_1 = h^{\delta_1}g^{d_1}$, then also d_2 and δ_2 such that $D_2 = h^{\delta_2}g^{d_2}$. This can be done via linear algebra in the exponents.
We can also, from the m distinct values of the z challenges, find t_i and γ_i such that $h^{\gamma_i}g^{t_i} = K_i$ for $i = 1, \ldots, m$. If any transcript doesn't have equality between $\langle \mathbf{p}_1(x_j), \mathbf{p}_2(x_j) \rangle$ and $d_2x_j^2 + d_1x_j + d_0$ for any of $j = 1, 2, 3$, then the binding property of Pedersen commitments (see Appendix B) has been violated, contrary to our assumptions.
Assuming all the previously mentioned equalities hold, then for all challenges y and z and three distinct challenges x_j, we have that $\alpha(x_j) - \beta(x_j) = 0$, where $\alpha(x_j) = \langle \mathbf{p}_1(x_j), \mathbf{p}_2(x_j) \rangle = \alpha_2x_j^2 + \alpha_1x_j + \alpha_0$ and $\beta(x_j) = d_2x_j^2 + d_1x_j + d_0$, for $j = 1, 2, 3$. Since the difference of these polynomials has degree 2, but three x_j roots, we get that it must be the 0 polynomial - hence, each of the terms are equal. In particular, the constant terms for both polynomials must be equal, hence α_0

and d_0 are equal. Since α_0 is the constant term of $\langle \mathbf{p}_1(x_j), \mathbf{p}_2(x_j)\rangle$, we can expand this inner product and subtract d_0 to get 0. Recalling that $d_0 = \varepsilon - \sum_{i=1}^{m} z^{i+1} t_i$ and $\varepsilon = (z+z^2)\langle \mathbb{1}, \mathbf{y}\rangle + (1+z)n\sum_{i=1}^{m} z^{i+1}$, we find that

$$
\begin{aligned}
0 = \alpha_0 - d_0 &= \alpha_0 - \left(\varepsilon - \sum_{i=1}^{m} z^{i+1} t_i\right) \\
&= \langle (\mathbb{1} - \mathbf{w}) + z\mathbb{1}, \mathbf{y} \star ((\lambda\mathbf{v})^{\omega} + z\mathbb{1}) + \sum_{i=1}^{m} z^{i+1}\mathbf{b}_{i,m,n}\rangle \\
&\quad - \left((z+z^2)\langle \mathbb{1}, \mathbf{y}\rangle + (1+z)n\sum_{i=1}^{m} z^{i+1} - \sum_{i=1}^{m} z^{i+1} t_i\right) \\
&= \langle (\mathbb{1} - \mathbf{w}) \star (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle - z\langle \mathbf{w} - (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle \\
&\quad + \sum_{i=1}^{m} z^{i+1}\langle \mathbb{1} - \mathbf{w}, \mathbf{b}_{i,m,n}\rangle + \sum_{i=1}^{m} z^{i+2}\langle \mathbb{1}, \mathbf{b}_{i,m,n}\rangle \\
&\quad - \left(\sum_{i=1}^{m} z^{i+1} n(1-t_i) + \sum_{i=1}^{m} z^{i+2} n\right) \\
&= \langle (\mathbb{1} - \mathbf{w}) \star (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle - z\langle \mathbf{w} - (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle \\
&\quad + \sum_{i=1}^{m} z^{i+1}(t_i - \langle \mathbf{w}, \mathbf{b}_{i,m,n}\rangle) + \sum_{i=1}^{m} z^{i+2}(\langle \mathbb{1}, \mathbf{b}_{i,m,n}\rangle - n) \\
&= \langle (\mathbb{1} - \mathbf{w}) \star (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle - z\langle \mathbf{w} - (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle + \sum_{i=1}^{m} z^{i+1}(t_i - \langle \mathbf{w}_i, \mathbb{1}^n\rangle)
\end{aligned}
$$

which follows since, recalling that $\mathbf{b}_{i,m,n} = (\mathbf{0}^{n(i-1)} \,||\, \mathbb{1}^n \,||\, \mathbf{0}^{n(m-i)})$, we have $\langle \mathbf{w}, \mathbf{b}_{i,m,n}\rangle = \langle \mathbf{w}_i, \mathbb{1}^n\rangle$.
Since this holds for $m+2$ challenges for z, but the above is a polynomial of degree $m+1$ in z, it must be identically the 0 polynomial, so each term is individually equal to 0. We can split up the above equality into three different equalities:

$$
\begin{aligned}
0 &= \langle (\mathbb{1} - \mathbf{w}) \star (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle && \text{(constant term)} && (8)\\
0 &= \langle \mathbf{w} - (\lambda\mathbf{v})^{\omega}, \mathbf{y}\rangle && \text{(linear term)} && (9)\\
t_i &= \langle \mathbf{w}_i, \mathbb{1}^n\rangle \quad \text{for } i = 1, \ldots, m, && \text{(higher terms)} && (10)
\end{aligned}
$$

where we point out that these are exactly the batching of the conditions from Equations (2), (3), and (4). Therefore $\mathbf{v}_i \in \mathcal{B}^t_{\lambda,\omega}$ for $i = 1, \ldots, m$, as claimed in the relation (7).
Thus, since $K_i = h^{\gamma_i} g^{t_i}$ for $i = 1, \ldots, m$, we have that $\mathbf{K} = \mathbf{h}^{\boldsymbol{\gamma}}\mathbf{g}^{\mathbf{t}}$, and the algorithm $\mathcal{E}_{\mathsf{HammR}}$ has extracted a list of valid witnesses $(\boldsymbol{\gamma}, \mathbf{t})$ for Relation (7). This extractor algorithm has runtime $3mn(m+2)\mathsf{poly}(n)$ times, which is polynomial in m and n. It will either output a valid witness or it will violate the discrete logarithm assumption, which we assume happens with negligible probability. Thus

we can apply the general forking lemma, Lemma 7 from Appendix C, and thusly Algorithm 3 obtains witness extended emulation.

Finally, we finish the proof by demonstrating zero-knowledge.

Theorem 5. *Algorithm 3 achieves the zero-knowledge property.*

Proof. To demonstrate zero-knowledge, we show that there exists a simulator algorithm $\mathcal{S}$ that can generate proofs which are indistinguishable from a genuine interaction. This simulator $\mathcal{S}$ computes the two values $\hat{D}_1$ and $\hat{C}_2$ as defined below:

$$\hat{D}_1 = \left(\mathbf{K}^{-z\mathbf{z}^m} \cdot g^{\varepsilon - \hat{d}} \cdot h^{-\delta_{x_0}} \cdot D_2^{x_0^2}\right)^{-x_0^{-1}},$$

$$\hat{C}_2 = \left(C_1 \cdot h^{-\mu} \cdot \mathbf{g}^{-\hat{\mathbf{p}}_1 + z\mathbb{1}^{nm}} \cdot \mathbf{h}^{-\hat{\mathbf{p}}_2 + z\mathbb{1}^{nm}} \prod_{i=1}^{m} \mathbf{h}_{[n(i-1)+1,\, ni]}^{z^{i+1}\mathbf{y}^{-nm}}\right)^{-x_0^{-1}},$$

but then selects all other values as uniformly at random.
To highlight that this algorithm will appear to honestly produce a valid transcript, we show that these two values are sufficient to pass the verifier's checks.

$$\mathsf{commit}(\hat{d}, \delta_{x_0}) \stackrel{?}{=} D_2^{x_0^2} \cdot \hat{D}_1^{x_0} \cdot \mathbf{K}^{-z\mathbf{z}^m} \cdot g^{\varepsilon} = g^{\hat{d}} \cdot h^{\delta_{x_0}}$$

and

$$\mathsf{commit}(\mu, \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2) \stackrel{?}{=} \hat{C}_2^{x_0} \cdot C_1 \cdot (\mathbf{g} \cdot \mathbf{h})^z \prod_{i=1}^{m} \mathbf{h}_{[n(i-1)+1,\, ni]}^{z^{i+1}\mathbf{y}^{-1}} = h^{\mu} \cdot \mathbf{g}^{\hat{\mathbf{p}}_1} \cdot \mathbf{h}^{\hat{\mathbf{p}}_2}$$

thus $\mathcal{S}$ can appear to successfully pass a verifier's challenges by selecting random values for $\hat{d}$, δ_{x_0}, and μ, and picking random vectors $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$. This means that a third party who's privy to the interaction of a prover and verifier can never be certain that what they're listening to isn't just a simulator algorithm $\mathcal{S}$. Hence Algorithm 3 obtains the zero-knowledge property, as any eavesdropping third party can never be certain that the prover in the interaction truly knows a valid vector, since they could always be simply emulating $\mathcal{S}$.

4.5 Turning HammR Non-Interactive

In order to turn this protocol non-interactive, we hash transcripts and apply the Fiat-Shamir heuristic [48,6], so we define the following values:

$$y := \text{Hash}\,(q, n, g, h, \mathbf{g}, \mathbf{h}, C_1, C_2, K) \text{ and } z := \text{Hash}\,(q, n, g, h, \mathbf{g}, \mathbf{h}, y, C_1, C_2, K).$$

There are certain issues with Fiat-Shamir that we aim to avoid [54], such as the so-called *Frozen Heart* vulnerability; this comes from *FoRging Of ZEro kNowledge proofs* along with Fiat-Shamir, which is the heart of turning a sigma protocol non-interactive [62]. For more details on the practicality of this vulnerability, we refer to [37,75,20,40,74].

We strive to avoid this vulnerability which stems from the weak Fiat-Shamir heuristic [17], so to this end, we hash public data, the generators, and all three commitments into the challenge y, then include y into the hash digest of z.

5 Applications of HammR

In this section we introduce a number of problems based on the standard syndrome decoding problem (SDP) that forms the basis for much of code-based cryptography. The purpose for these problems will become apparent in the next section, where we apply HammR to these problems.

5.1 Application to Syndrome Decoding Problems

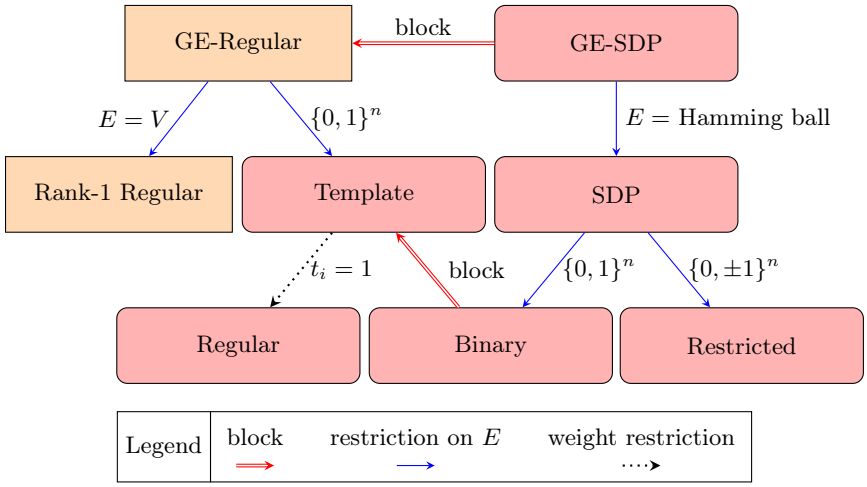

Fig. 1: Dependency diagram for SDPs. The squared orange boxes are problems introduced in this paper.

Problem 1 (Syndrome Decoding Problem). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and $t \in \mathbb{N}$, to recover the error vector $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$ and $\mathrm{wt}(\mathbf{e}) = t$.

A few things to note: there is an inferred *if such a solution exists* at the end of the above definition. We suppress this addendum for readability, but the above and all subsequent problems should be read as still containing it. We also point out that many authors prefer the $\mathrm{wt}(\mathbf{e}) \leq t$ variant of the SDP - clearly, the equality condition is stronger. Finally, we present these problems as search problems, but will abuse notation and refer to them as NP-complete, even though this term is

reserved for decisional problems. This is not an issue, as there exists a canonical search-to-decision reduction.
Problem 1 was shown to be NP-complete over a binary alphabet in [16], then later over any finite field by [11].
In [71], the authors introduce a generic version of the SDP, which is presented here in the language of this paper.

Problem 2 (Generic Error Syndrome Decoding Problem (GE-SDP)). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a set $E \subseteq \mathbb{F}_q^n$, to recover the error vector $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$ and $\mathbf{e} \in E$.

Taking E to be a Hamming ball recovers the traditional SDP, hence this problem is NP-Complete. One could also set E to be a rank ball [12], a Lee ball [90], or really any set one pleases. In particular, selecting a multiplicative group gives the following problem.

Problem 3 (Restricted Syndrome Decoding Problem, [10]). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a multiplicative group $\mathcal{G} \leq \mathbb{F}_q^*$, to recover the error vector $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$ and $\mathbf{e} \in (\mathcal{G} \cup \{0\})^n$.

This problem was shown to be NP-Complete in [10], and has since become the motivation for the CROSS signature scheme [9].

Problem 4 (Regular Syndrome Decoding Problem [7]). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ and syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, to recover the error vector $\mathbf{e} = (\mathbf{e}_1||\ldots||\mathbf{e}_r) \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$ and $\mathrm{wt}(\mathbf{e}_i) = 1$, with each non-zero entry equal to 1 for $i = 1, \ldots, r$.

Problem 4, though originally over $\mathbb{F}_2$, was introduced in [7] for the purpose of creating provably secure cryptographic hash functions, where they subsequently prove that the Regular SDP is NP-complete. Additionally, this primitive has been used with Multi Party Computation from [61] to design a novel signature scheme in [26]; for more background, see the citations in [26, Section 2.2]. In [47,30], the authors show that pairing the Regular SDP with *the in the head* paradigm leads to increased efficiency. Note that in [47], the Regular SDP is referred to as the *d-split syndrome decoding problem* - this title can be found in other references too, such as [53]. This problem has also found use in oblivious linear-function evaluation, in the field of functional secret sharing [24]. Mixing Regular SDP with the *vector* oblivious linear evaluation or VOLE framework from [14] has lead to further efficiency gains [18,77]. The practical difficulty of solving Problem 4 is discussed in [39,25], and different parameter regimes for the Regular SDP have been studied in [44,60,67], supporting the notion that the choice of parameters are of critical importance regarding the problem's difficulty. There exists a generalization of this problem where each $\mathbf{e}_i$ that forms the vector $\mathbf{e} = (\mathbf{e}_1||\ldots||\mathbf{e}_r)$ is required to have $\mathrm{wt}(\mathbf{e}_i) = t_i$ such that $\mathrm{wt}(\mathbf{e}) = \sum_{i=1}^{r} t_i$

instead of $\mathrm{wt}(\mathbf{e}_i) = 1$, but the entries still in $\{0, 1\}$. It can also be viewed as a blocking of the Binary SDP, which is simply Problem 1 over $\mathbb{F}_2$. This has been called the *Template Syndrome Decoding Problem* in [19], though the authors focus on side-channel attacks and make no claims about the hardness. This problem clearly is at least as difficult as the SDP, because setting $r = n$ collapses to Problem 1, with each $t_i \in \{0, 1\}$. We present it here over $\mathbb{F}_q$, instead of the usual $\mathbb{F}_2$.

Problem 5 (Template Syndrome Decoding Problem). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and $t \in \mathbb{N}$, to recover the error vector $\mathbf{e} = (\mathbf{e}_1||\ldots||\mathbf{e}_r) \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$, where $\mathrm{wt}(\mathbf{e}_i) = t_i$ with $t = \sum_{i=1}^{r} t_i$, such that each non-zero entry of $\mathbf{e}_i$ is equal to 1.

Another clear direction for generalizing the Regular SDP and Template SDP is permitting the entries of $\mathbf{e}$ to be taken from a more arbitrary space, instead of $\{0, 1\}$. This can also be viewed as a block version of the GE-SDP, where $\mathbf{e} = (\mathbf{e}_1||\ldots||\mathbf{e}_r)$ and each $\mathbf{e}_i$ is an instance of Problem 2. To the best of our knowledge, this problem remains unnamed, so we identify it here:

Problem 6 (Generic-Error Regular Syndrome Decoding Problem (GE-Regular SDP)). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, weight distribution vector $(t_1, \ldots, t_r)$, and error sets $E_i \subseteq \mathbb{F}_q$, to recover the error vector $\mathbf{e} = (\mathbf{e}_1||\ldots||\mathbf{e}_r) \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$, where the non-zero entries of $\mathbf{e}_i$ are taken from E_i and $\mathrm{wt}(\mathbf{e}_i) = t_i$ for $i = 1, \ldots, r$.

Corollary 1. *The GE-Regular SDP is NP-Complete.*

This can be readily seen, as specifying $E_i = \{0, 1\}$ and $t_i = 1$ for $i = 1, \ldots, r$ in the GE-SDP returns exactly the Regular SDP.
Below, we formalize a novel special case of the GE-Regular SDP. This represents a generalization of the Regular SDP, where the blocks are permitted to take entries in some subspace V satisfying $\dim_{\mathbb{F}_p^i}(V) = 1$.

Problem 7 (Rank-1 Regular Syndrome Decoding Problem). This problem asks upon input of full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and weight distribution vector $(t_1, \ldots, t_r)$, to recover the error vector $\mathbf{e} = (\mathbf{e}_1||\ldots||\mathbf{e}_r) \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{He}$ and $\mathbf{e}_i$ is rank-1 satisfying $\mathrm{wt}(\mathbf{e}_i) = t_i$ for $i = 1, \ldots, r$.

This problem is clearly at least as hard as Problem 4, because it contains the instance when $\mathbf{e} \in \{0, 1\}^n$, which devolves into the Regular SDP, hence:

Corollary 2. *The Rank-1 Regular SDP is NP-Complete.*

We now shift towards applying HammR to these problems by demonstrating the validity of the Hamming weight and entry restrictions, zero-knowledge proving that a vector represents a valid instance of its corresponding cryptographic primitive, agnostic of the syndrome requirement. One example where this functionality could be helpful is where a user holds a number of CROSS keys, and wants to demonstrate in a zero-knowledge fashion that they are indeed valid restricted vectors.

First, suppose that $\mathbf{v} \in \mathbb{F}_q^n$ is a restricted vector, meaning that it satisfies $\mathrm{wt}(\mathbf{v}) = t$ with entries from $\mathcal{G} \cup \{0\}$, where $\mathcal{G}$ is a multiplicative subgroup of $\mathbb{F}_q^*$. That is, it meets the Hamming weight and entry requirements of the Restricted SDP from [10]. To prove this validity, one can employ HammR to show that $\mathbf{v} \in \mathcal{B}^t_{\lambda,\omega}$, which is exactly the necessary condition, as a consequence of Lemma 2. Indeed, this can be utilized to show that vector $\mathbf{v}$ is a valid instance of CROSS from [9], which requires it to be full-weight with elements from a subgroup of $\mathbb{F}_q^*$.

Similarly, let $\mathbf{v} \in \mathbb{F}_q^n$ be an instance of the Regular SDP, meaning that $\mathbf{v} = (\mathbf{v}_1 || \ldots || \mathbf{v}_r)$ with $\mathrm{wt}(\mathbf{v}_i) = 1$ where the non-zero entry of $\mathbf{v}_i$ takes value 1 for $i = 1, \ldots, r$. Since we have demonstrated that HammR can be batched efficiently, this protocol can be applied to prove that $\mathbf{v}_i \in \mathcal{B}^1_{1,1}$ for $i = 1, \ldots, r$, thus $\mathbf{v}$ represents a valid instance of the Regular SDP.

In fact, HammR is capable of handling an even stronger statement, showing that $\mathbf{v}$ satisfies $\mathrm{wt}(\mathbf{v}_i) = t_i$ for $i = 1, \ldots, r$. That is, that each $\mathbf{v}_i$ is a valid Template SDP instance. Or even more generally, that it meets the novel GE-Regular SDP criteria for a suitable error set $E = \mathcal{B}^t_{\lambda,\omega}$.

Similarly, HammR can showcase that $\mathbf{v}$ is a valid instance of the Rank-1 Regular SDP, a novel NP-Complete problem. The statement of this problem demands that $\mathbf{v} = (\mathbf{v}_1 || \ldots || \mathbf{v}_r)$ be such that each $\mathbf{v}_i$ is a weight-t rank-1 vector for $i = 1, \ldots, r$, or equivalently, that $\mathbf{v}_i \in \mathcal{B}^t_{\lambda,p^i-1}$.

5.2 Further Applications

We now present a number of additional potential applications for the HammR ZKP protocol framework.

Lookup instances. We showcase this first application from [52], which presents a lookup instance protocol. Let $n \leq N$, and suppose we wish to determine if all the entries of a small vector $\mathbf{a} \in \mathbb{F}_q^n$ are also entries in a large vector $\mathbf{t} \in \mathbb{F}_q^N$. This is in fact equivalent to the existence of a matrix $\mathbf{M} \in \mathbb{F}_q^{n \times N}$ such that

a) $\mathbf{M} \cdot \mathbf{t} = \mathbf{a}$,
b) The rows of $\mathbf{M}$ are standard basis vectors, ie: assuming $N < \mathrm{char}(\mathbb{F}_q)$, that $\mathbf{M} \star \mathbf{M} = \mathbf{M}$ and $\mathbf{M} \cdot \mathbb{1}^N = \mathbb{1}^n$.

Our protocol HammR can be used in this formulation of lookup instances, as all the matrix products above can be turned into inner product calculations with vectors in $\mathcal{B}^1_{1,1}$.

Proximity proofs. This second application involves a Hamming proof of proximity, similar to [4] which is concerned with bounded one-sided error. HammR can be used to demonstrate that $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n$ are proximal as vectors in a zero-knowledge fashion, by implementing the protocol on the difference.

Electronic voting. Finally, HammR has applications to electronic voting, where citizens can vote securely over the Internet instead of traveling to a polling location and voting via paper ballot, which could conceivably be damaged in transit, tampered with, or outright lost. Electronic voting has been implemented in Estonia for almost 20 years [45], and remains secure due to the use of digital signatures. A recent protocol was proposed in [46], which takes the form of a ternary lattice adaptation of the Stern protocol [87], requiring a check that a vector $\mathbf{s}$ satisfies $\mathbf{s} \star (\mathbf{s} + \mathbb{1}) \star (\mathbf{s} - \mathbb{1}) = \mathbf{0}$. HammR handles this check in a more concise manner, by checking that $\mathbf{v} \in \mathcal{B}_{1,2}^t$.

6 Conclusions

In this paper, we have presented HammR, a proof protocol demonstrating knowledge of a vector that has Hamming weight t and has entries taken from a shifted subgroup. We have proven that proofs can be efficiently batched, then that this batched protocol is complete, sound, and zero-knowledge - thus a ZKP. We have presented applications of this work to a number of well-known syndrome decoding problems, including introducing two novel problems to showcase HammR. Additionally, we showed how HammR can be leveraged to solve other problems as well, including lookup instances, vector proximity, and electronic voting protocols.
Overall, HammR represents a significant advancement in the field of zero-knowledge proofs. It offers a concise and efficient solution for proving the properties of vectors with bounded entries. Future work may explore optimizing the protocol further and extending its applications to other cryptographic problems.

Appendix A Security Assumptions

In this Appendix we define our security assumptions.
Recall that $\mathbb{1}^\lambda$ denotes the length-λ vector of all ones. We write GGen to mean a group generation algorithm that takes input $\mathbb{1}^\lambda$ with security parameter λ, and outputs the description of group $\mathbb{G}$. As usual, we reserve g to be a generator of $\mathbb{G}$. The following problem is perhaps the most famous problem to come out of classical cryptography:

Problem 8 (Discrete Logarithm Problem). The discrete logarithm problem is:

$$\text{given input } \{(g,\, g^\alpha) \,|\, (\mathbb{G},\, q,\, g) \leftarrow \mathsf{GGen}(\mathbb{1}^\lambda),\, \alpha \xleftarrow{\$} \mathbb{F}_q\}, \text{ recover } \alpha.$$

Assumption 1 (Discrete Logarithm Assumption) *We say that the discrete logarithm assumption holds if the following advantage is bounded by some negligible function for all probabilistic, polynomial-time adversaries $\mathcal{A}$:*

$$Pr[\mathcal{A}(g,\, g^{\alpha}) = \alpha \,|\, (\mathbb{G},\, q,\, g) \leftarrow \mathsf{GGen}(1^{\lambda}),\, \alpha \xleftarrow{\$} \mathbb{F}_q] \leq \mathsf{negl}(\lambda)$$

Clearly, the hardness of the Problem 8 relies on Assumption 1, ie: that an adversary's advantage is negligible.
The next two problems form the basis for the famous Diffie-Hellman key exchange.

Problem 9 (Computational Diffie-Hellman Problem). The computational Diffie-Hellman problem is:

$$\text{given input } \{(g,\, g^{\alpha},\, g^{\beta}) \,\big|\, (\mathbb{G},\, q,\, g) \leftarrow \mathsf{GGen}(1^{\lambda}),\, \alpha, \beta \xleftarrow{\$} \mathbb{F}_q\},\ \text{compute } g^{\alpha\beta}.$$

Problem 10 (Decisional Diffie-Hellman Problem). The decisional Diffie-Hellman problem is:

$$\text{given input } \{(g,\, g^{\alpha},\, g^{\beta},\, g^{\gamma}) \,\big|\, (\mathbb{G},\, q,\, g) \leftarrow \mathsf{GGen}(1^{\lambda}),\, \alpha, \beta, \gamma \xleftarrow{\$} \mathbb{F}_q\},\ \text{determine if } \gamma = \alpha\beta.$$

Both Problem 9 and Problem 10 rely on the following assumption:

Assumption 2 (Diffie-Hellman Assumption) *We say that the (decisional) Diffie-Hellman assumption holds if for all probabilistic, polynomial-time adversaries $\mathcal{A}$, the following advantage is negligible:*

$$\Bigg| Pr\Big[\mathcal{A}(g,\, g^{\alpha},\, g^{\beta},\, g^{\alpha\beta}) = 1 \,\Big|\, (\mathbb{G},\, q,\, g) \leftarrow \mathsf{GGen}(1^{\lambda}),\, \alpha, \beta \xleftarrow{\$} \mathbb{F}_q\Big]$$
$$- Pr\Big[\mathcal{A}(g,\, g^{\alpha},\, g^{\beta},\, g^{\gamma}) = 1 \,\Big|\, (\mathbb{G},\, q,\, g) \leftarrow \mathsf{GGen}(1^{\lambda}),\, \alpha, \beta, \gamma \xleftarrow{\$} \mathbb{F}_q\Big] \Bigg| \leq \mathsf{negl}(\lambda)$$

If an adversary can efficiently solve the discrete logarithm problem, then they can also solve the computational Diffie-Hellman problem. If they can efficiently solve the computational Diffie-Hellman problem, then they can also solve the decisional variant. From this, we surmise that the discrete logarithm problem is at least as hard as the computational Diffie-Hellman problem, which is at least as hard as the decisional Diffie-Hellman problem. For this document, we suppose that Assumption 1 and Assumption 2 both hold, ie: that Problem 8 and Problems 9, 10 cannot be solved efficiently by a probabilistic, polynomial-time adversary.

Appendix B Pedersen Commitments

Let $\mathbb{G}_p$ denote a group with prime p order. We write $g = (g_1, \ldots, g_\ell)$, where each $g_i \in \mathbb{G}_p$ generates a copy of $\mathbb{G}_p$, and use this to define $g \in \mathbb{G}$. In general, we will assume that p is a large enough prime, so that $p >> \ell$.

For $v \in \mathbb{F}_{p^\ell}$, we know that there exists an isomorphism $\varphi : \mathbb{F}_{p^\ell} \to \mathbb{F}_p^\ell$, so we're justified in writing $\mathbf{v} = (v_1, \ldots, v_\ell)$ with each $v_i \in \mathbb{F}_p$, up to a choice of basis.

Using these, we define $\mathbf{g}^\mathbf{v}$ for $\mathbf{g} \in \mathbb{G}$ and $\mathbf{v} \in \mathbb{F}_{p^\ell}$ by $\mathbf{g}^\mathbf{v} := (g_1^{v_1}, \ldots, g_\ell^{v_\ell})$, where each individual $g_i \in \mathbb{G}_p$ and $v_i \in \mathbb{F}_p$. This definition is constructed to behave well with Pedersen commitments: for $\mathbf{g}, \mathbf{h} \in \mathbb{G}$ and $\mathbf{v}, \mathbf{w} \in \mathbb{F}_{p^\ell}$, then

$$\mathbf{g}^\mathbf{v} \cdot \mathbf{h}^\mathbf{w} = (g_1^{v_1}, \ldots, g_\ell^{v_\ell}) \star (h_1^{w_1}, \ldots, h_\ell^{w_\ell}) = (g_1^{v_1} h_1^{w_1}, \ldots, g_\ell^{v_\ell} h_\ell^{w_\ell}).$$

Note that this Pedersen commitment over $\mathbb{G}$ looks like a vector of ℓ commitments over $\mathbb{G}_p$.

When considering vectors of commitments, let $\mathbf{g} \in \mathbb{G}^n$ and $\mathbf{v} \in \mathbb{F}_{p^\ell}^n$. Swapping to a pair of indices, where the first denotes its index in $\mathbb{G}^n$ and the second its index in $\prod_{i=1}^{\ell} \mathbb{G}_p$ we define

$$\begin{aligned}
\mathbf{g}^\mathbf{v} &= (g_1, \ldots, g_n)^{(v_1, \ldots, v_n)} \\
&= \left((g_{1,1}, \ldots, g_{1,\ell})^{(v_{1,1}, \ldots, v_{1,\ell})}, \ldots, (g_{n,1}, \ldots, g_{n,\ell})^{(v_{n,1}, \ldots, v_{n,\ell})} \right) \\
&= \left(\left(g_{1,1}^{v_{1,1}}, \ldots, g_{1,\ell}^{v_{1,\ell}} \right), \ldots, \left(g_{n,1}^{v_{n,1}}, \ldots, g_{n,\ell}^{v_{n,\ell}} \right) \right).
\end{aligned}$$

Not only does this generalize constructing Pedersen commitments over $\mathbb{G}_p$ using $\mathbb{F}_p$, it formalizes taking commitments over $\mathbb{G}$ using $\mathbb{F}_{p^\ell}$. This is necessary to standardize, as if we consider $\mathbb{F}_4 = \{0, 1, \alpha, \alpha + 1\}$, then computing g^α for a group element g may not make immediate sense.

The idea of committing to values was introduced by Blum in [21] for the purpose of coin flipping by telephone, where an individual must be able to commit to a value, then unwrap it. For an extensive overview of zero knowledge basics, we refer the interested reader to [63]. Formally, a commitment scheme as defined by [23] is two parts: the first is an efficient randomized algorithm CGen, and the second an efficient deterministic function Com. The setup algorithm CGen generates a commitment c, which defines a message space $\mathcal{M}$, a sample space $\mathcal{R}$, and a commitment space $\mathcal{C}$. We let pp denote public parameters, generated from $\mathsf{CGen}(\mathbb{1}^\lambda)$, where λ is some security parameter. The commitment function $\mathsf{Com} : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ uses a message and randomness to formulate a commitment. For a message $x \in \mathcal{M}$, the sender selects $r \xleftarrow{\$} \mathcal{R}$ uniformly at random, then forms $c := \mathsf{Com}(r, x)$.

Definition 10 (Hiding). *We say that a commitment scheme* $(\mathsf{CGen}, \mathsf{Com})$ *is computationally hiding if the commitment c doesn't reveal the secret x. This is, for every probabilistic, polynomial-time adversary $\mathcal{A}$, the following holds:*

$$\left| Pr\left[\mathcal{A}(c) = b \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{CGen}(\mathbb{1}^\lambda) \\ x_0,\, x_1 \leftarrow \mathcal{M} \\ b \xleftarrow{\$} \{0,1\} \\ r \xleftarrow{\$} \mathcal{R} \\ c := \mathsf{Com}(r, x_b) \end{array} \right] - \tfrac{1}{2} \right| \leq \mathsf{negl}(\lambda)$$

If this probability is $\frac{1}{2}$ for all $\mathcal{A}$, then we say the scheme is perfectly hiding.

Definition 11 (Binding). *We say that a commitment scheme* $(\mathsf{CGen}, \mathsf{Com})$ *is computationally binding if the commitment c can only be opened to its secret x. This is, for all polynomial-time adversaries $\mathcal{A}$:*

$$Pr\left[c_0 = c_1 \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{CGen}(\mathbb{1}^\lambda) \\ (r_0, x_0, r_1, x_1) \leftarrow \mathcal{A}(\mathsf{pp}) \\ x_0 \neq x_1 \\ c_i = \mathsf{Com}(r_i, x_i) \text{ for } i = 0, 1 \end{array} \right] \leq \mathsf{negl}(\lambda)$$

If this probability is 0 for all $\mathcal{A}$, then we say the scheme is perfectly binding.

Both of the above definitions are computational; if we remove the polynomial-time condition on the adversary, then the definitions are *statistical* hiding and binding.

Definition 12. *We say that a commitment scheme is homomorphic if for all $x_0,\, x_1 \in M$ and $r_0,\, r_1 \in R$, have that*

$$\mathsf{Com}(r_0, x_0) \cdot \mathsf{Com}(r_1, x_1) = \mathsf{Com}(r_0 + r_1, x_0 + x_1).$$

The most common commitment scheme used for zero-knowledge protocols is the Pedersen commitment, which relies on Problem 8. The Pedersen commitment takes the form $c = h^r g^x$, where g and h are group elements specified in CGen. Pedersen commitments are perfectly hiding, due to the fact that r is taken as a uniformly random; on the other hand, if an adversary is capable of breaking the binding property, then they must be able to extract discrete logarithms - in violation of Assumption 1. Since this Pedersen commitment is exactly the scalar commit protocol of Algorithm 1, we see that this protocol achieves advantageous security properties such as hiding and binding. For the vector commit protocol of Algorithm 1, we use a variant of Pedersen commitment which, for $\mathbf{x} = (x_1,\, x_2)$, takes the form $\mathsf{Com}(r, \mathbf{x}) = h^r \mathbf{g}^{\mathbf{x}} = h^r g_1^{x_1} g_2^{x_2}$, which allows us to commit to multiple values at once while still maintaining desirable properties, such as perfect hiding and computational binding under the discrete logarithm assumption.

Appendix C Zero-Knowledge Basics

A *zero knowledge proof* (ZKP) is a primitive that permits a prover to convince a verifier about the veracity of some statement without disclosing any other information surrounding their claim. One method of constructing ZKPs is in the Common Reference String model, which is what we use here - this is not the only option though, and other papers prefer the Random Oracle Model.

Let $\mathcal{L}$ be a language in NP, and let $\mathcal{R}$ be a relation that can be verified efficiently (ie: polynomial time) such that a statement s is in $\mathcal{L}$ if and only if there exists a witness w satisfying $(s; w) \in \mathcal{R}$.

A ZKP can then be thought of as a triple of algorithms $\mathit{\Pi} = (\mathsf{Gen}, \mathcal{P}, \mathcal{V})$, all of which are probabilistic, polynomial time algorithms. These represent a generator for a common reference string Gen, the prover $\mathcal{P}$, and the verifier $\mathcal{V}$. Gen accepts an input of $\mathbb{1}^\lambda$ with security parameter λ and outputs a common reference string σ. The prover's algorithm $\mathcal{P}(\sigma, (s; w)) = \pi$ accepts input of the common reference string σ, a public statement s, and a secret witness w, then produces proof π. The verifier's algorithm $\mathcal{V}(\sigma, s, \pi)$ takes in common reference string σ, statement s, and proof π, then outputs $b \in \{0, 1\}$, where $b = 1$ indicates that the proof has been accepted, and rejected else. When the prover and verifier act on their respective inputs p and v, the transcript they produce will be denoted $\tau \leftarrow \langle \mathcal{P}(p), \mathcal{V}(v) \rangle$. When this transcript is accepting, we will write $\tau = 1$.

A ZKP has three major security properties that it must satisfy: zero-knowledge, completeness, and soundness. We take these definitions from references such as [63,35,36,58,28,32,88]

Definition 13 (Public Coin Protocol). *A proof $\mathit{\Pi}$ is called a public coin protocol if the verifier's challenges are selected uniformly at random, independent of any values that appeared previously in the transcript.*

Definition 14 (Honest-verifier Zero-Knowledge). *Let ρ be the verifier's random public coin. A public coin protocol is said to obtain honest-verifier zero-knowledge for relation $\mathcal{R}$ if there exists a probabilistic, polynomial-time simulator $\mathcal{S}$ such that for all polynomial time adversary $\mathcal{A}$, the following holds for some negligible function:*

$$\left| Pr\left[\begin{matrix} \mathcal{A}(\tau) = 1 \\ (s; w) \in \mathcal{R} \end{matrix} \middle| \begin{matrix} \sigma \leftarrow \mathsf{Gen}(\mathbb{1}^\lambda) \\ ((s; w), \rho) = \mathcal{A}(\sigma) \\ \tau \leftarrow \langle \mathcal{P}(\sigma, (s; w)), \mathcal{V}(\sigma, s, \rho) \rangle \end{matrix} \right] \right.$$
$$\left. - Pr\left[\begin{matrix} \mathcal{A}(\tau') = 1 \\ (s; w) \in \mathcal{R} \end{matrix} \middle| \begin{matrix} \sigma' \leftarrow \mathsf{Gen}(\mathbb{1}^\lambda) \\ ((s; w), \rho) = \mathcal{A}(\sigma') \\ \tau' \leftarrow \mathcal{S}(s, \rho) \end{matrix} \right] \right| \leq \mathsf{negl}(\lambda)$$

When the negligible function is outright 0, this is called *perfect zero knowledge.*

Definition 15 (Completeness). *A proof Π for relation $\mathcal{R}$ is called complete if, given security parameter λ, for all $(s; w) \in \mathcal{R}$, the probability a valid proof is wrongly rejected is negligible:*

$$Pr\left[b = 0 \ but \ (s; w) \in \mathcal{R} \middle| \begin{matrix} \sigma \leftarrow \mathsf{Gen}(\mathbb{1}^\lambda) \\ \pi = \mathcal{P}(\sigma, (s; w)) \\ b = \mathcal{V}(\sigma, s, \pi) \end{matrix}\right] \leq \mathsf{negl}(\lambda)$$

Definition 16 (Soundness). *A proof Π for relation $\mathcal{R}$ is called sound if for all polynomial-time adversaries $\mathcal{A}$, the probability that their proof get wrongly accepted is negligible:*

$$Pr\left[b = 1 \ but \ (s; w) \notin \mathcal{R} \middle| \begin{matrix} \sigma \leftarrow \mathsf{Gen}(\mathbb{1}^\lambda) \\ \pi \leftarrow \mathcal{A}(\sigma, (s; w)) \\ b = \mathcal{V}(\sigma, s, \pi) \end{matrix}\right] \leq \mathsf{negl}(\lambda)$$

Note that this definition pertains specifically to long-term scenarios. The negligible function bounding above could, in a single round, take a form like $\frac{q+1}{2q}$, as found in the CROSS signature scheme from [9]. However, this is not in opposition to the definition though, as after r rounds, this value decreases exponentially in r, thereby is indeed negligible.

Witness extended emulation is a more robust version of knowledge soundness that tries to avoid certain subtle pitfalls where the *expected* polynomial-time simulator $\mathcal{S}$ gets called too often, and in total might not actually be polynomial-time! For a concrete example, see [66, Section 3.3]. In this citation, it is also shown that if one obtains knowledge soundness, then one has witness extended emulation as well. [58] adapted this framework from Proof of Knowledge to a more generic model, while [23] uses witness extended emulation as their definition for soundness.

We demonstrate witness extended emulation, as it's more convenient in our setting. Informally, this says that if an adversary $\mathcal{A}$ can generate an argument in polynomial time that has probability p of convincing the verifier, then there exists an emulator algorithm $\mathcal{E}$ that can also convince the verifier with probability p, but that moreover generates a witness.

Definition 17 (Witness Extended Emulation).
A proof Π has witness extended emulation if for all deterministic polynomial-time provers $\mathcal{P}'$, there exists a polynomial-time simulator $\mathcal{S}$ such that for all interactive adversaries $\mathcal{A}$, the following is negligible:

$$\left| Pr\left[\mathcal{A}(\tau) = 1 \middle| \begin{matrix} \sigma \leftarrow \mathsf{Gen}(\mathbb{1}^\lambda) \\ (s; u) \leftarrow \mathcal{A}(\sigma) \\ \tau \leftarrow \langle \mathcal{P}'(\sigma, (s; u)), \mathcal{V}(\sigma, s) \rangle \end{matrix}\right] - Pr\left[\begin{matrix} \mathcal{A}(\tau') = 1 \\ \tau' = 1 \Rightarrow (s; w) \in \mathcal{R} \end{matrix} \middle| \begin{matrix} \sigma \leftarrow \mathsf{Gen}(\mathbb{1}^\lambda) \\ (s; u) \leftarrow \mathcal{A}(\sigma) \\ (\tau', w) \leftarrow \mathcal{E}_{\mathcal{O}}(\sigma, s) \end{matrix}\right] \right| \leq \mathsf{negl}(\lambda)$$

where u is another witness for statement s, and where the oracle $\mathcal{O}$ is given by transcripts $\langle \mathcal{P}'(\sigma, (s; u)), \mathcal{V}(\sigma, s) \rangle$ which also allows rewinding the protocol to a certain point, introducing fresh verifier randomness, then continuing the protocol to create a new branch with the same initial values.

Finally, we recall the general forking lemma that acts as a extension for special soundness, as given in [80,15,23,27,36]. Suppose we have a public coin protocol that has $2m+1$ interactions between $\mathcal{P}$ and $\mathcal{V}$, which thusly generates challenges $x_1, \ldots, x_m$ in order. For $i \in [1, \ldots, m]$, let $n_i \geq 1$, and consider $N := \prod_{i=1}^{m} n_i$ accepting transcripts, with their corresponding challenges arranged in a tree: we initialize by generating a tree with depth m where each node of depth i has exactly n_i children, then we label the topmost root with the statement s in question. We label each node with a distinct value for the ith challenge x_i, for each of the N total leaves. This N also corresponds to the number of paths from the root statement to the bottommost leaves; ie: the number of accepting transcripts. We will call this tree a $(n_1, \ldots, n_m)$-tree of transcripts.

Lemma 7. *Let Π be a $2m + 1$ interaction public coin protocol, and suppose there exists a polynomial-time extraction algorithm $\mathcal{E}$ that can extract a witness w for statement s from a $(n_1, \ldots, n_m)$-tree of accepting transcripts with success probability $1 - \mathsf{negl}(\lambda)$, where $\mathsf{negl}(\lambda)$ is a negligible function. If $N \leq p(\lambda)$ for some polynomial p in security parameter λ, then Π obtains witness extended emulation.*

We note that taking $m = 1$ and $n = 2$ recovers the standard definition of *special soundness*, and refer to [6,5,13] for a formal definition of this concept.

References

1. Albrecht, M.R., Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., Von Maurich, I., Misoczki, R., Niederhagen, R., et al.: Classic McEliece: Conservative Code-Based Cryptography (2022), `https://classic.mceliece.org/`
2. Aragon, N., Barreto, P.L., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Ghosh, S., Gueron, S., Güneysu, T., Melchor, C.A., Misoczki, R., Persichetti, E., Richter-Brockmann, J., Sendrier, N., Tillich, J.P., Vasseur, V., Zémor, G.: BIKE - Bit Flipping Key Encapsulation. `https://bikesuite.org/`, accessed: 2024-11-18
3. Aragon, N., Blazy, O., Gaborit, P., Hauteville, A., Zémor, G.: Durandal: a Rank Metric Based Signature Scheme. Cryptology ePrint Archive, Paper 2018/1192 (2018), `https://eprint.iacr.org/2018/1192`
4. Arnon, G., Ben-David, S., Yogev, E.: Hamming Weight Proofs of Proximity with One-Sided Error. Cryptology ePrint Archive, Paper 2024/832 (2024), `https://eprint.iacr.org/2024/832`
5. Attema, T., Cramer, R., Kohl, L.: A Compressed Σ-Protocol Theory for Lattices. Cryptology ePrint Archive, Paper 2021/307 (2021), `https://eprint.iacr.org/2021/307`
6. Attema, T., Fehr, S., Klooß, M.: Fiat-Shamir Transformation of Multi-Round Interactive Proofs. Cryptology ePrint Archive, Paper 2021/1377 (2021), `https://eprint.iacr.org/2021/1377`

7. Augot, D., Finiasz, M., Sendrier, N.: A Fast Provably Secure Cryptographic Hash Function. Cryptology ePrint Archive, Paper 2003/230 (2003), https://eprint.iacr.org/2003/230
8. Baldi, M., Barenghi, A., Beckwith, L., Biasse, J.F., Esser, A., Gaj, K., Mohajerani, K., Pelosi, G., Persichetti, E., Saarinen, M.J.O., Santini, P., Wallace, R.: LESS - Linear Equivalence Signature Scheme. https://www.less-project.com/, accessed: 2024-11-18
9. Baldi, M., Barenghi, A., Bitzer, S., Karl, P., Manganiello, F., Pavoni, A., Pelosi, G., Santini, P., Schupp, J., Slaughter, F., Wachter-Zeh, A., Weger, V.: CROSS: Codes and Restricted Objects Signature Scheme. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/CROSS-spec-web.pdf, accessed: 2024-11-05
10. Baldi, M., Battaglioni, M., Chiaraluce, F., Horlemann-Trautmann, A.L., Persichetti, E., Santini, P., Weger, V.: A New Path to Code-Based Signatures via Identification Schemes with Restricted Errors (2021), https://arxiv.org/abs/2008.06403
11. Barg, A.: Some New NP-Complete Coding Problems. Probl. Peredachi Inf. **30**(3), 209–214 (1997)
12. Bartz, H., Holzbaur, L., Liu, H., Puchinger, S., Renner, J., Wachter-Zeh, A.: Rank-Metric Codes and Their Applications (2022), https://arxiv.org/abs/2203.12384
13. Battagliola, M., Longo, R., Pintore, F., Signorini, E., Tognolini, G.: Security of Fixed-Weight Repetitions of Special-Sound Multi-Round Proofs. Cryptology ePrint Archive, Paper 2024/884 (2024), https://eprint.iacr.org/2024/884
14. Baum, C., Braun, L., de Saint Guilhem, C.D., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head. Cryptology ePrint Archive, Paper 2023/996 (2023), https://eprint.iacr.org/2023/996
15. Bellare, M., Neven, G.: Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma. In: Proceedings of the 13th ACM Conference on Computer and Communications Security. pp. 390–399. Association for Computing Machinery (10 2006). https://doi.org/10.1145/1180405.1180453, https://doi.org/10.1145/1180405.1180453
16. Berlekamp, E., McEliece, R., van Tilborg, H.: On the Inherent Intractability of Certain Coding Problems (Corresp.). IEEE Transactions on Information Theory **24**(3), 384–386 (1978). https://doi.org/10.1109/TIT.1978.1055873
17. Bernhard, D., Pereira, O., Warinschi, B.: How not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. Cryptology ePrint Archive, Paper 2016/771 (2016), https://eprint.iacr.org/2016/771
18. Bettaieb, S., Bidoux, L., Gaborit, P., Kulkarni, M.: Modelings for Generic PoK and Applications: Shorter SD and PKP Based Signatures. Cryptology ePrint Archive, Paper 2024/1668 (2024), https://eprint.iacr.org/2024/1668
19. Bitzer, S., Delvaux, J., Kirshanova, E., Maaßen, S., May, A., Wachter-Zeh, A.: How to Lose Some Weight - A Practical Template Syndrome Decoding Attack. Cryptology ePrint Archive (2024), https://eprint.iacr.org/2024/621
20. Block, A.R., Garreta, A., Katz, J., Thaler, J., Tiwari, P.R., Zając, M.: Fiat-Shamir Security of FRI and Related SNARKs. Cryptology ePrint Archive, Paper 2023/1071 (2023), https://eprint.iacr.org/2023/1071
21. Blum, M.: Coin flipping by telephone a protocol for solving impossible problems. SIGACT News **15**(1), 23–27 (Jan 1983). https://doi.org/10.1145/1008908.1008911, https://doi.org/10.1145/1008908.1008911

22. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and its Applications. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. p. 103–112. STOC '88, Association for Computing Machinery, New York, NY, USA (1988). https://doi.org/10.1145/62212.62222, https://doi.org/10.1145/62212.62222
23. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting. Cryptology ePrint Archive, Paper 2016/263 (2016), https://eprint.iacr.org/2016/263
24. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing Vector OLE. Cryptology ePrint Archive, Paper 2019/273 (2019). https://doi.org/10.1145/3243734.3243868, https://eprint.iacr.org/2019/273
25. Briaud, P., Øygarden, M.: A New Algebraic Approach to the Regular Syndrome Decoding Problem and Implications for PCG Constructions. Cryptology ePrint Archive, Paper 2023/176 (2023), https://eprint.iacr.org/2023/176
26. Bui, D., Carozza, E., Couteau, G., Goudarzi, D., Joux, A.: Faster Signatures from MPC-in-the-Head. Cryptology ePrint Archive, Paper 2024/252 (2024), https://eprint.iacr.org/2024/252
27. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short Proofs for Confidential Transactions and More. Cryptology ePrint Archive, Paper 2017/1066 (2017), https://eprint.iacr.org/2017/1066
28. Bünz, B., Maller, M., Mishra, P., Tyagi, N., Vesely, P.: Proofs for Inner Pairing Products and Applications. Cryptology ePrint Archive, Paper 2019/1177 (2019), https://eprint.iacr.org/2019/1177
29. Canetti, R., Fischlin, M.: Universally Composable Commitments. Cryptology ePrint Archive, Paper 2001/055 (2001), https://eprint.iacr.org/2001/055
30. Carozza, E., Couteau, G., Joux, A.: Short Signatures from Regular Syndrome Decoding in the Head. Cryptology ePrint Archive, Paper 2023/1035 (2023), https://eprint.iacr.org/2023/1035
31. Cayrel, P.L., Véron, P., El Yousfi Alaoui, S.M.: A Zero Knowledge Identification Scheme Based on the q-ary SD Problem. In: Selected Areas in Cryptography. LNCS, vol. 6544, pp. 171–186. Springer, Waterloo, Canada (Aug 2010). https://doi.org/10.1007/978-3-642-19574-7_12, https://inria.hal.science/hal-00674249
32. Chen, Y., Ma, X., Tang, C., Au, M.H.: PGC: Pretty Good Decentralized Confidential Payment System with Auditability. Cryptology ePrint Archive, Paper 2019/319 (2019), https://eprint.iacr.org/2019/319
33. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, P., Ward, N.: Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS. Cryptology ePrint Archive, Paper 2019/1047 (2019), https://eprint.iacr.org/2019/1047
34. Chou, T., Niederhagen, R., Persichetti, E., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: Take your MEDS: Digital Signatures from Matrix Code Equivalence. Cryptology ePrint Archive, Paper 2022/1559 (2022), https://eprint.iacr.org/2022/1559
35. Christ, M., Baldimtsi, F., Chalkias, K.K., Maram, D., Roy, A., Wang, J.: SoK: Zero-Knowledge Range Proofs. Cryptology ePrint Archive, Paper 2024/430 (2024), https://eprint.iacr.org/2024/430
36. Chung, H., Han, K., Ju, C., Kim, M., Seo, J.H.: Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger. Cryptology ePrint Archive, Paper 2020/735 (2020), https://eprint.iacr.org/2020/735

37. Ciobotaru, O., Peter, M., Velichkov, V.: The Last Challenge Attack: Exploiting a Vulnerable Implementation of the Fiat-Shamir Transform in a KZG-Based SNARK. Cryptology ePrint Archive, Paper 2024/398 (2024), https://eprint.iacr.org/2024/398
38. Courtois, N., Finiasz, M., Sendrier, N.: How to Achieve a McEliece-Based Digital Signature Scheme. Cryptology ePrint Archive, Paper 2001/010 (2001), https://eprint.iacr.org/2001/010
39. Cui, H., Liu, H., Yan, D., Yang, K., Yu, Y., Zhang, K.: ReSolveD: Shorter Signatures from Regular Syndrome Decoding and VOLE-in-the-Head. Cryptology ePrint Archive, Paper 2024/040 (2024), https://eprint.iacr.org/2024/040
40. Dao, Q., Miller, J., Wright, O., Grubbs, P.: Weak Fiat-Shamir Attacks on Modern Proof Systems. Cryptology ePrint Archive, Paper 2023/691 (2023), https://eprint.iacr.org/2023/691
41. Debris-Alazard, T., Sendrier, N., Tillich, J.P.: Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes. Cryptology ePrint Archive, Paper 2018/996 (2018), https://eprint.iacr.org/2018/996
42. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model, p. 356–383. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-26951-7_13, http://dx.doi.org/10.1007/978-3-030-26951-7_13
43. Engelbert, D., Overbeck, R., Schmidt, A.: A Summary of McEliece-Type Cryptosystems and Their Security. Cryptology ePrint Archive, Paper 2006/162 (2006), https://eprint.iacr.org/2006/162
44. Esser, A., Santini, P.: Not Just Regular Decoding: Asymptotics and Improvements of Regular Syndrome Decoding Attacks. Cryptology ePrint Archive, Paper 2023/1568 (2023), https://eprint.iacr.org/2023/1568
45. of Estonia, S.E.S.: General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia. https://www.regeringen.ax/sites/default/files/attachments/page/estonia-e-voting-2017.pdf, accessed: 2024-12-4
46. Farzaliyev, V., Pärn, C., Saarse, H., Willemson, J.: Lattice-Based Zero-Knowledge Proofs in Action: Applications to Electronic Voting. Journal of Cryptology **38** (01 2025). https://doi.org/10.1007/s00145-024-09530-5, https://doi.org/10.1007/s00145-024-09530-5
47. Feneuil, T., Joux, A., Rivain, M.: Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs. Cryptology ePrint Archive, Paper 2022/188 (2022). https://doi.org/10.1007/978-3-031-15979-4_19, https://eprint.iacr.org/2022/188
48. Fiat, A., Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology — CRYPTO' 86. pp. 186–194. Springer Berlin Heidelberg, Berlin, Heidelberg (1987), https://doi.org/10.1007/3-540-47721-7_12
49. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over Lagrange-Bases for Oecumenical Noninteractive Arguments of Knowledge. Cryptology ePrint Archive, Paper 2019/953 (2019), https://eprint.iacr.org/2019/953
50. Gaborit, P., Ruatta, O., Schrek, J., Zémor, G.: RankSign: an Efficient Signature Algorithm Based on the Tank ,etric. Cryptology ePrint Archive, Paper 2013/766 (2013), https://eprint.iacr.org/2013/766

51. Gaborit, P., Girault, M.: Lightweight Code-Based Identification and Signature. In: 2007 IEEE International Symposium on Information Theory. pp. 191–195 (2007). https://doi.org/10.1109/ISIT.2007.4557225, https://ieeexplore.ieee.org/document/4557225
52. Garreta, A., Manzur, I.: FLI: Folding Lookup Instances. Cryptology ePrint Archive, Paper 2024/1531 (2024), https://eprint.iacr.org/2024/1531
53. Godard, J., Aragon, N., Gaborit, P., Loiseau, A., Maillard, J.: Single Trace Side-Channel Attack on the MPC-in-the-Head Framework. Cryptology ePrint Archive, Paper 2024/1882 (2024), https://eprint.iacr.org/2024/1882
54. Goldwasser, S., Kalai, Y.: On the (In)security of the Fiat-Shamir paradigm. In: 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings. pp. 102–113 (2003). https://doi.org/10.1109/SFCS.2003.1238185
55. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating Computation: Interactive Proofs for Muggles. J. ACM **62**(4) (Sep 2015). https://doi.org/10.1145/2699436, https://doi.org/10.1145/2699436
56. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof Systems. SIAM Journal on Computing **18**(1), 186–208 (1989). https://doi.org/10.1137/0218012, https://doi.org/10.1137/0218012
57. Groth, J.: On the Size of Pairing-Based Non-Interactive Arguments. Cryptology ePrint Archive, Paper 2016/260 (2016), https://eprint.iacr.org/2016/260
58. Groth, J., Ishai, Y.: Sub-linear Zero-Knowledge Argument for Correctness of a Shuffle. In: International Conference on the Theory and Application of Cryptographic Techniques (2008), https://doi.org/10.1007/978-3-540-78967-3_22
59. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 212–219. STOC '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237866, https://doi.org/10.1145/237814.237866
60. Hazay, C., Orsini, E., Scholl, P., Soria-Vazquez, E.: TinyKeys: A New Approach to Efficient Multi-Party Computation. Cryptology ePrint Archive, Paper 2018/208 (2018), https://eprint.iacr.org/2018/208
61. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-Knowledge Proofs from Secure Multiparty Computation. SIAM Journal on Computing **39**(3), 1121–1152 (2009). https://doi.org/10.1137/080725398, https://doi.org/10.1137/080725398, https://doi.org/10.1137/080725398
62. Jones, M.: Vac 101: Transforming an Interactive Protocol to a Noninteractive Argument. https://vac.dev/rlog/vac101-fiat-shamir/, accessed: 2024-10-15
63. Jones, M.: Zero-Knowledge Reductions and Confidential Arithmetic. Ph.D. thesis, Clemson University (2023), https://tigerprints.clemson.edu/all_dissertations/3472
64. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-Size Commitments to Polynomials and Their Applications. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010. pp. 177–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2010), https://doi.org/10.1007/978-3-642-17373-8_11
65. Li, Y.X., Deng, R., Wang, X.M.: On the Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems. IEEE Transactions on Information Theory **40**(1), 271–273 (1994). https://doi.org/10.1109/18.272496, https://ieeexplore.ieee.org/document/272496
66. Lindell, Y.: Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation. Cryptology ePrint Archive, Paper 2001/107 (2001), https://eprint.iacr.org/2001/107

67. Liu, H., Wang, X., Yang, K., Yu, Y.: The Hardness of LPN Over Any Integer Ring and Field for PCG Applications. Cryptology ePrint Archive, Paper 2022/712 (2022), https://eprint.iacr.org/2022/712
68. Liu, Q., Zhandry, M.: Revisiting Post-Quantum Fiat-Shamir. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019. pp. 326–355. Springer International Publishing, Cham (2019), https://doi.org/10.1007/978-3-030-26951-7_12
69. Lund, C., Fortnow, L., Karloff, H., Nisan, N.: Algebraic Methods for Interactive Proof Systems. J. ACM **39**(4), 859–868 (Oct 1992). https://doi.org/10.1145/146585.146605, https://doi.org/10.1145/146585.146605
70. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updateable Structured Reference Strings. Cryptology ePrint Archive, Paper 2019/099 (2019), https://eprint.iacr.org/2019/099
71. Manganiello, F., Slaughter, F.: Generic Error SDP and Generic Error CVE. Cryptology ePrint Archive, Paper 2023/717 (2023), https://eprint.iacr.org/2023/717
72. McEliece, R.J.: A Public Key Cryptosystem Based on Algebraic Coding Theory. In: Deep Space Network Progress Report (1978), https://api.semanticscholar.org/CorpusID:56502909
73. Melchor, C.A., Bettaieb, S., Blazy, O., Bos, J., Deneuville, J.C., Dion, A., Gaborit, P., Lacan, J., Persichetti, E., Robert, J.M., Véron, P., Zémor, G.: HQC - Hamming Quasi-Cyclic. https://pqc-hqc.org/, accessed: 2024-11-18
74. Miller, J.: The Frozen Heart Vulnerability in Bulletproofs. https://blog.trailofbits.com/2022/04/15/the-frozen-heart-vulnerability-in-bulletproofs/, accessed: 2024-10-15
75. Nguyen, H., Ho, U., Biryukov, A.: Fiat-Shamir in the Wild. Cryptology ePrint Archive, Paper 2024/1565 (2024), https://eprint.iacr.org/2024/1565
76. NIST: Post-Quantum Cryptography. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography, accessed: 2024-11-18
77. Ouyang, Y., Tang, D., Xu, Y.: Code-Based Zero-Knowledge from VOLE-in-the-Head and Their Applications: Simpler, Faster, and Smaller. Cryptology ePrint Archive, Paper 2024/1414 (2024), https://eprint.iacr.org/2024/1414
78. Overbeck, R., Sendrier, N.: Code-Based Cryptography, pp. 95–145. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_4, https://doi.org/10.1007/978-3-540-88702-7_4
79. Parno, B., Gentry, C., Howell, J., Raykova, M.: Pinocchio: Nearly Practical Verifiable Computation. Cryptology ePrint Archive, Paper 2013/279 (2013), https://eprint.iacr.org/2013/279
80. Pointcheval, D., Stern, J.: Security Proofs for Signature Schemes. In: Maurer, U. (ed.) Advances in Cryptology — EUROCRYPT '96. pp. 387–398. Springer Berlin Heidelberg, Berlin, Heidelberg (1996), https://doi.org/10.1007/3-540-68339-9_33
81. Puchinger, S., Renner, J., Rosenkilde, J.: Generic Decoding in the Sum-Rank Metric. In: 2020 IEEE International Symposium on Information Theory (ISIT). pp. 54–59 (2020). https://doi.org/10.1109/ISIT44484.2020.9174497, https://ieeexplore.ieee.org/document/9174497
82. Ritterhoff, S., Maringer, G., Bitzer, S., Weger, V., Karl, P., Schamberger, T., Schupp, J., Wachter-Zeh, A.: FuLeeca: A Lee-Based Signature Scheme. Cryptology ePrint Archive, Paper 2023/377 (2023), https://eprint.iacr.org/2023/377

83. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing **26**(5), 1484–1509 (Oct 1997). https://doi.org/10.1137/s0097539795293172, http://dx.doi.org/10.1137/S0097539795293172
84. Shor, P.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700, https://ieeexplore.ieee.org/document/365700
85. StarkWare: Cambrian Explosion of Cryptographic Proofs. https://medium.com/starkware/cambrian-explosion-of-cryptographic-proofs-5740a41cdbd2, accessed: 2024-11-18
86. Stern, J.: A New Paradigm for Public Key Identification. IEEE Transactions on Information Theory **42**(6), 1757–1768 (1996). https://doi.org/10.1109/18.556672, https://ieeexplore.ieee.org/document/556672
87. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) Advances in Cryptology — CRYPTO' 93. pp. 13–21. Springer Berlin Heidelberg, Berlin, Heidelberg (1994), https://doi.org/10.1007/3-540-48329-2_2
88. Thaler, J.: Proofs, Arguments, and Zero-Knowledge. https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.html, accessed: 2024-11-26
89. Véron, P.: Improved Identification Schemes Based on Error-Correcting Codes. Appl. Algebra Eng. Commun. Comput. **8**, 57–69 (01 1996). https://doi.org/10.1007/s002000050053, https://link.springer.com/article/10.1007/s002000050053
90. Weger, V., Khathuria, K., Horlemann, A.L., Battaglioni, M., Santini, P., Persichetti, E.: On the Hardness of the Lee Syndrome Decoding Problem (2022), https://arxiv.org/abs/2002.12785