Proving Knowledge of Syndrome Decoding Problems with Soundness

Felice Manganiello[1] and Freeman Slaughter[2,✉]

[1]Clemson University, Clemson SC, USA
[2]University of South Florida, Tampa FL, USA
✉fslaughter@usf.edu

Abstract. This paper introduces a novel set of code-based protocols to demonstrate algebraic relationships in zero knowledge. Specifically, we present a comprehensive collection of secure arguments of knowledge for verifying additive and multiplicative relationships between syndrome-committed secrets, including matrix products, which enable us to construct a generic arithmetic circuit framework. We leverage these primitives to formulate a rich variety of privacy-oriented primitives, such as a post-quantum range proof, lookup argument, and verifiable shuffle protocols. These contributions provide the necessary ingredients to transition advanced confidential designs, such as cryptocurrencies, into the post-quantum code-based setting.

1 Introduction

The discrete logarithm world of cryptography currently serves as the gold standard for zero-knowledge proofs: due to the homomorphic nature of Pedersen commitments, cryptographers have access to a rich toolbox with which they can easily demonstrate algebraic relationships between hidden secrets. This framework, besides being merely theoretically interesting, forms a fundamental backbone for essential privacy-preserving designs such as Bulletproofs and arithmetic circuits: mechanisms that enable modern cryptocurrencies.

In contrast, the code-based setting - while enjoying post-quantum security guarantees - lacks a cohesive and practical cookbook of analogous protocols. This work aims to bridge that gap by providing the missing schemes needed to replicate these advanced privacy features using standard coding theory assumptions, bringing us closer to a practical code-based cryptocurrency.

1.1 Contribution

In this work, we introduce a tangible collection of code-based protocols, analogues to the familiar discrete logarithm gadgets, which permit us to formulate complex zero-knowledge proof (ZKP) arguments directly from the Syndrome Decoding Problem (SDP). From the presented proofs for linear and multiplicative relationships over committed syndromes, we demonstrate that code-based

cryptography is prepared to support the advanced mechanics of next-generation, privacy-centric cryptocurrencies like Monero and Zcash.

We expand on the applications of these primitives to construct an arithmetic circuit framework, capable of proving arbitrary statements, such as the validity of a graph 3-coloring, or verifying that a list is sorted. Specifically, we introduce a method to prove a vector is binary, which allows us to embed logic predicated on Boolean statements into commitments. While existing code-based frameworks rely on assumptions like Learning Parity with Noise [50,75,12], MPC-in-the-Head [3], or expander codes [45,76], our construction relies strictly on the hardness of the Syndrome Decoding Problem. Finally, we utilize these tools to formulate a type of rudimentary post-quantum range proof - a critical component for preventing inflationary bugs in decentralized finance systems, such as cryptocurrencies.

The paper is organized as follows: in Section 2 we outline the preliminaries required to understand this paper, including notation in 2.1, code-based cryptography in 2.2, commitment schemes in 2.3, and zero-knowledge proofs in 2.4. We introduce protocols designed to verify relationships between secrets inside commitments in Section 3: in 3.1 we showcase a proof of opening, in 3.2 a proof that two commitments hide the same secret, in 3.3 a proof that a commitment hides the sum of secrets from two other commitments, and in 3.4 a proof of a matrix product, which can be adapted to matrix-vector products or even vector Schur products. Finally, we discuss applications in Section 4, including verifying Beaver triples 4.1, ensuring that CROSS keys are properly formed 4.2, an arithmetic circuit framework 4.3, a basic range proof 4.4, a lookup argument 4.5, and a verifiable shuffle 4.6. We close with conclusions in Section 5.

1.2 Technical Overview

This work addresses the gap between discrete logarithm based zero-knowledge primitives (Pedersen commitments, Bulletproofs) and the post-quantum code-based environment. It introduces a set of 5-pass ZKPs that verify algebraic properties for code-based commitments, by leveraging the hardness of the Compressed Syndrome Decoding Problem (C-SDP) and the Rank Syndrome Decoding Problem. With these results in hand, we construct secure arguments for arithmetic circuits, range proofs, and verifiable shuffles without relying on structured lattices or exotic assumptions.

We utilizes a C-SDP commitment structure $s = He + Tr$, analogous to Ajtai commitments. Security relies on the indistinguishability of the secret error e of Hamming weight t_1 and the blinder r of Hamming weight t_2 within the syndrome s. We adapt the CVE identification scheme to the C-SDP setting, resulting in a $(q-1, 2)$ identification scheme proof of opening: from a syndrome s, the prover shows that they know Hamming ball errors e and r such that $s = He + Tr$. We extend this technique into a protocol proving that a third commitment C_3 hides the sum of secrets from C_1 and C_2, i.e.: that $e_3 = e_1 + e_2$, where each error is hidden inside its respective commitment via the C-SDP. Crucially, this supports commitments defined by distinct parity-check matrices, enabling the

bridging of independent instances of the SDP. Finally, we introduce a mechanism to prove $M_3 = M_1 M_2$ for committed matrices under the rank metric, which can be adapted to matrix-vector products, or even vector Schur products. Namely, M_1 and M_2 are committed to utilizing the rank metric version of the C-SDP, then utilizing the product scheme herein, one can verify that a third commitment must hide their product. By embedding vectors into diagonal matrices, this scheme is adapted to prove vector Schur products $z = x \star y$ in the Hamming metric, serving as the multiplication gate for arithmetic circuits, though this can also be realized by replacing the standard matrix product operation with the Hadamard product in the natural way.

Leveraging these primitives, we realize several complex applications, essential for confidential transactions:

- *Beaver Triples*: Using the protocols we present, one can verify many Beaver triples at once, relying only on code-based arguments and avoiding the need for MPC techniques or trusted setup.
- *Properly Formed Keys*: These schemes can be readily applied to show that a CROSS or a WAVE key has been properly formed, as CROSS vectors have entries chosen from a multiplicative subgroup, and WAVE vectors are formed from sums of Schur products.
- *Arithmetic Circuits*: We formulate a complete arithmetic circuit framework, including a binary check, which permits the embedding of arbitrary boolean logic into a vector.
- *Post-Quantum Range Proof*: A rudimentary range proof to demonstrate that a committed integer a is contained in $[0, 2^n - 1]$ via bit decomposition. While linear in size, inefficient compared to logarithmic elliptic curve approaches like Bulletproofs, it relies strictly on code-based assumptions.
- *Lookup Arguments*: Implementing the subvector extraction method by proving knowledge of an elementary matrix M such that $MA = a$.
- *Verifiable Shuffle*: Proving a committed matrix P is a permutation by verifying it is binary ($P \star P = P$) and doubly stochastic ($P\mathbb{1} = \mathbb{1}$ and $\mathbb{1}^\top P = \mathbb{1}^\top$).

2 Preliminaries

2.1 Notation

We use $[n]$ to refer to the set $\{1, \ldots, n\}$. Let $\mathbb{F}_q$ denote the finite field with q elements, a prime power, and let $\mathbb{F}_q^n$ be vectors of length n with entries in $\mathbb{F}_q$. Denote by $\mathbb{F}_q^\star$ the multiplicative group of $\mathbb{F}_q$, or $\mathbb{F}_q \setminus \{0\}$. We use $x \xleftarrow{\$} S$ to denote that the element x has been selected uniformly at random from the set S. For x, y, we use $x \star y$ to denote the Schur (if over vectors) or Hadamard (if over matrices) product, the entrywise product. Let $\mathbb{0}$ and $\mathbb{1}$ be the vectors of all 0 or 1 entries, respectively.

For $x \in \mathbb{F}_q^n$, we reserve $\mathrm{wt}(x)$ to denote the number of non-zero entries, or $|\{i \in [n] \mid x_i \neq 0\}|$. We use this to define the Hamming ball $\mathcal{B}(t) = \{x \in \mathbb{F}_q^n \mid \mathrm{wt}(x) \leq t\}$. We define the set of monomial transformations as the set

of matrices which are permutations, but with non-zero scalars in place of 1's. Namely, $\mathrm{Mon}_n = \{P \cdot \mathrm{diag}(a_1, \ldots, a_n) \mid P \in \mathrm{Perm}_n \text{ and } a_i \in \mathbb{F}_q^\star\}$. Any $\mathbb{F}_q$-linear map taking $\mathbb{F}_q^n$ to $\mathbb{F}_q^n$ which preserves the Hamming weight is called a linear isometry; it's a basic exercise in coding theory to show that the set of monomial transformations generates the entire group of linear isometries [64].

We say that a function is negligible in λ if it decays faster than the reciprocal of any polynomial in λ, and denote this as $\mathrm{negl}(\lambda)$. We also say that an algorithm is probabilistic polynomial-time (PPT) if it possesses an output range with a probability distribution and a runtime which is bounded by some polynomial of the input length.

2.2 Code-Based Cryptography

As cryptocurrencies like Monero evolve towards Full-Chain Membership Proofs (FCMP++) to enhance scalability and privacy [61], there is a pressing need to translate these fundamental discrete logarithm based ZKPs into the post-quantum setting. Code-based cryptography, relying on the hardness of decoding random linear codes, provides a natural framework for constructing powerful zero-knowledge arguments.

Difficult Problems: We rely on the following standard hardness assumptions:

Problem 1 (Syndrome Decoding Problem (SDP)). Given parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$, syndrome $s \in \mathbb{F}_q^{n-k}$, and target weight $t \in \mathbb{N}$, find the error vector $e \in \mathcal{B}(t)$ such that $s = He$.

Problem 2 (Rank Syndrome Decoding Problem (Rank-SDP)). Given parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$, syndrome $s \in \mathbb{F}_q^{(n-k)\times m}$, and target rank $r \in \mathbb{N}$, find the error vector $e \in \mathbb{F}_q^{n\times m}$ such that $\mathrm{rank}(e) = r$ and $s = He$.

While the complexity of the rank-metric variant is not as well secured as the Hamming case, generic decoding algorithms for the rank metric generally perform worse than their Hamming counterparts [16,10,25,39,41]. It is important to note that the set of linear isometries in the rank metric, denoted here as $\mathrm{Mon}_{n\times m}$, differs from the Hamming metric; specifically, the equivalent of monomial transformations in the rank metric consists of actions equivalent to simultaneous left and right multiplication by appropriately-sized invertible matrices [65,40].

We now introduce the specific variant used for our commitment schemes:

Problem 3 (Compressed Syndrome Decoding Problem (C-SDP)). Given two parity-check matrices $H, T \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $s \in \mathbb{F}_q^{n-k}$, and target weights $t_1, t_2 \in \mathbb{N}$, find error vectors $e \in \mathcal{B}(t_1)$ and $r \in \mathcal{B}(t_2)$ such that $He + Tr = s$.

Observe that the C-SDP can be reformulated as an instance of the SDP with larger parameters:

$$\begin{pmatrix} H & T \end{pmatrix} \begin{pmatrix} e \\ r \end{pmatrix} = s.$$

Here, the total weight $t_1 + t_2$ is segregated across the concatenated error vector, a distinction that separates it from the SDP.

Commitment Structure: We utilize commitments of the form $He + Tr$, where the secret $e \in \mathcal{B}(t_1)$ is blinded by a random vector $r \xleftarrow{\$} \mathcal{B}(t_2)$. This construction mirrors Ajtai commitments [1] from lattice-based cryptography, a primitive that has been extensively developed for zero-knowledge proofs [11,57,49].

While we define the problem with both e and r chosen from Hamming balls, strict adherence to the weight constraint for the secret e is not required for security. As long as the blinder r is selected with sufficient weight to make the corresponding SDP instance difficult, the secret e remains hidden.

This flexibility is crucial for the protocols we present later in Section 3. It implies that we can select e from any arbitrary error set E, provided it satisfies unique decodability [53]. Since an adversary cannot distinguish the individual contributions of He and Tr to the syndrome s, recovering e remains hard, even if e could be easily recoverable from He by itself.

2.3 Commitment Schemes

We rely on commitment schemes as a fundamental primitive. Formally, a commitment scheme $\mathtt{com}$ allows a prover to commit to a message m using randomness r, which we denote as $c = \mathsf{com}(m, r)$. To be secure, such a scheme must satisfy two properties: it must be *binding* (this prevents the prover from altering the message later) and *hiding* (this prevents the verifier from uncovering the message before intended).

Definition 1. *A commitment scheme $\mathtt{com}$ is binding if for all PPT adversaries $\mathcal{A}$,*

$$Pr[\mathsf{com}(m, r) = \mathsf{com}(m', r') \mid (m, m', r, r') \leftarrow \mathcal{A} \text{ where } m \neq m'] \leq negl(\lambda),$$

where λ is the security parameter of the scheme.

Definition 2. *A commitment scheme $\mathtt{com}$ is hiding if for all distinct messages m and m', we have that the probability distributions*

$$\{\mathsf{com}(m, r) \mid r \xleftarrow{\$} \mathcal{R}\} \text{ and } \{\mathsf{com}(m', r) \mid r \xleftarrow{\$} \mathcal{R}\}$$

are indistinguishable, where $\mathcal{R}$ is the randomness space.

In the protocols we present later in Section 3, we instantiate $\mathtt{com}$ using a cryptographic hash function (e.g., SHA-3) modeled as a random oracle. Practically, binding is computationally equivalent to the collision resistance property, where an adversary capable of violating binding with impunity, producing distinct pairs (m, r) and (m', r') that map to the same commitment, must necessarily have found a hash collision. For $\mathtt{com}$, we assume that finding collisions is an intractable problem.

2.4 Zero-Knowledge Proofs

We begin by defining a zero-knowledge proof, a concept originally introduced by Goldwasser, Micali, and Rackoff [44]. A ZKP is an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, where at the end of their interaction, $\mathcal{V}$ is convinced of the validity of a statement, but this interaction reveals no information beyond veracity.

Formally, consider a relation $\mathcal{R}$ containing pairs $(x; w)$, where x is a public statement and w is a private witness known only to $\mathcal{P}$. The goal is for $\mathcal{P}$ to demonstrate that $(x; w) \in \mathcal{R}$.

The protocols considered in this work are 5-pass interactive proofs, an extension of the standard 3-pass Σ-protocols from Babai [7]. The interaction proceeds as follows:

1. $\mathcal{P}$ sends a commitment $\mathtt{com}_1$.
2. $\mathcal{V}$ sends a random challenge $\mathtt{ch}_1$.
3. $\mathcal{P}$ sends a second commitment $\mathtt{com}_2$.
4. $\mathcal{V}$ sends a second random challenge $\mathtt{ch}_2$.
5. $\mathcal{P}$ sends a final response $\mathtt{resp}$.

Finally, $\mathcal{V}$ considers the transcript of their interaction $\tau = (\mathtt{com}_1, \mathtt{ch}_1, \mathtt{com}_2, \mathtt{ch}_2, \mathtt{resp})$ and outputs 1 for accept, or 0 for reject.

Definition 3. *A zero-knowledge proof is an interactive protocol as defined above, but with the following properties:*

- ***Completeness**: If $(x; w) \in \mathcal{R}$, and both parties follow the protocol honestly, then $\mathcal{V}$ accepts with probability* 1.
- ***Soundness**: If there exists no w such that $(x; w) \in \mathcal{R}$, i.e.: the statement is false, then for any malicious prover $\mathcal{P}^*$, $\mathcal{V}$ only accepts with negligible probability.*
- ***Zero-Knowledge**: For any computationally efficient verifier $\mathcal{V}^*$, there exists a PPT simulator $\mathcal{S}$, without access to the witness w, which can produce a transcript indistinguishable from a genuine interaction. In this case, it is impossible for any information about w to be leaked, as they do not know a valid witness.*

In a single execution, the probability that a dishonest prover convinces the verifier is termed the *soundness error*, denoted by σ. This error can be reduced below any desired probabilistic threshold t by simply repeating the protocol r times sequentially until $\sigma^r < t$ [6].

For constructing efficient arguments, we rely on the stronger property of *special soundness* [30]. While standard soundness guarantees that a false statement cannot be proved, special soundness guarantees that a witness can be extracted from a set of accepting transcripts with colliding challenges. For more background, we refer to [37,47,52].

Definition 4. *An interactive protocol enjoys n-special soundness if there exists a PPT extractor algorithm $\mathcal{E}$ such that, when it receives statement x and a set of n accepting transcripts with the same initial commitments but distinct challenges, it can output a valid witness w such that $(x; w) \in \mathcal{R}$.*

The Fiat-Shamir Transformation: There exists a method to convert an interactive proof into a non-interactive zero-knowledge argument via the Fiat-Shamir transformation [36]. In this paradigm, the verifier's random challenges are replaced by the output of a cryptographic hash function applied to the transcript, e.g.: $\mathtt{ch}_1 = H(x, \mathtt{com}_1)$. In the Random Oracle Model, this transformation preserves the soundness and zero-knowledge properties of the underlying protocol.

Definition 5 ($(q-1,2)$ Identification Scheme [27]). *A 5-pass interactive protocol is called a $(q-1,2)$ identification scheme if the first challenge is drawn from a space of cardinality $q-1$ while the second from a space of cardinality 2, and the probability that* com *takes on any particular value is negligible*

Throughout this paper, we will shorten this to $(q-1,2)$-IS for brevity. As shown in [27], such schemes have a soundness error of $\sigma = \frac{q}{2(q-1)}$, which tends towards $1/2$ as the field size increases. Furthermore, they satisfy a generalized $(2,2)$-out-of-$(q-1,2)$ special soundness property from [5,6]: a witness can be extracted given two transcripts that collide on the first challenge, or two transcripts that share the first challenge but collide on the second.

We remark that when applying the Fiat-Shamir transform to 5-pass protocols, care must be taken to include all prior commitments in the hash function for subsequent challenges to prevent the forgery attacks described by Kales and Zaverucha [51].

3 Zero-Knowledge Algebraic Relationships

In this section, we present a collection of protocols designed to verify algebraic relationships between committed secrets by acting directly on their syndromes. Comparable protocols are ubiquitous in the discrete logarithm setting [24,74], serving as the foundation for privacy-oriented applications such as cryptocurrencies.

To contextualize our code-based constructions, we briefly recall the Pedersen commitment scheme [62], which is the gold standard in discrete logarithm cryptography due to its homomorphic properties, as well as a desirable hiding and binding tradeoff. Let $\mathbb{G}$ be a cyclic group of prime order p. A commitment to a secret $x \in \mathbb{F}_p$ is computed as $C = g^x h^r$, where $g, h \in \mathbb{G}$ are generators with no known discrete logarithm relationship, and $r \xleftarrow{\$} \mathbb{F}_p$ is a random blinder. The homomorphic nature of Pedersen commitments permit a verifier to easily confirm additive constraints on secrets from only their committed values. Our goal is to replicate these capabilities by designing protocols to prove sums, products, and other relations from code-based primitives, using only the hardness of the SDP.

3.1 Proof of Opening

We begin with the CVE identification scheme [23], a code-based zero-knowledge proof designed to demonstrate knowledge of a solution to the SDP. As a $(q-1,2)$

IS, this protocol enjoys a soundness error of $\frac{q}{2(q-1)}$, a significant improvement over the original Stern protocol [70,69], which suffered from a high soundness error of 2/3.

Functionally, this protocol serves as a proof of opening: to demonstrate knowledge of an error vector $e \in \mathcal{B}(t)$ such that $s = He$, the prover demonstrates their ability to open the commitment defined by the syndrome s, without revealing the secret e itself. This is the code-based analogue of the Schnorr protocol [66] in the discrete logarithm setting, which proves knowledge of exponents x, r for a Pedersen commitment $P = g^x h^r$.

In Algorithm 1.1, we present a generalization of the CVE scheme adapted for the C-SDP. Here, the prover demonstrates simultaneous knowledge of a secret $e \in \mathcal{B}(t_1)$ and a blinder $r \in \mathcal{B}(t_2)$ such that $s = He + Tr$, opening the commitment. This construction builds upon previously established Stern-like protocols, formulated for code-based group and blind signature frameworks [2,19].

The CVE Scheme		
Public: $q, n, k \in \mathbb{N}$, $H, T \in \mathbb{F}_q^{(n-k)\times n}$, $s = He + Tr$ **Private:** $e \in \mathcal{B}(t_1)$, $r \in \mathcal{B}(t_2)$		
PROVER		**VERIFIER**
Pick $u, v \xleftarrow{\$} \mathbb{F}_q^n$, $\pi, \rho \xleftarrow{\$} \mathrm{Mon}_n$ Set $c_1 = \mathsf{com}(\pi, \rho, Hu + Tv)$, $c_2 = \mathsf{com}(\pi(u), \pi(e), \rho(v), \rho(r))$	$\xrightarrow{(c_1, c_2)}$	
	$\xleftarrow{z}$	Sample $z \xleftarrow{\$} \mathbb{F}_q^*$
Define $y_1 = \pi(u + ze)$ Define $y_2 = \rho(v + zr)$	$\xrightarrow{y_1, y_2}$	
	$\xleftarrow{b}$	Pick $b \xleftarrow{\$} \{1, 2\}$
If $b = 1$, set $\mathtt{resp} = \{\pi, \rho\}$ Else, set $\mathtt{resp} = \{\pi(e), \rho(r)\}$	$\xrightarrow{\mathtt{resp}}$	
		If $b = 1$, accept if $\delta = H\pi^{-1}(y_1) + T\rho^{-1}(y_2) - zs$ $c_1 \stackrel{?}{=} \mathsf{com}(\pi, \rho, \delta)$ Else, accept if $\varepsilon_1 = y_1 - z\pi(e)$ $\varepsilon_2 = y_2 - z\rho(r)$ $c_2 \stackrel{?}{=} \mathsf{com}(\varepsilon_1, \pi(e), \varepsilon_2, \rho(r))$ $\mathrm{wt}(\pi(e)) = t_1$, $\mathrm{wt}(\rho(r)) = t_2$

Algorithm 1.1: The CVE code-based zero-knowledge scheme adapted to the C-SDP.

We note that the weight requirement for e is not strict; as was shown in the generic error framework of [53], the CVE scheme supports arbitrary error distributions, including those from a metric-agnostic regime. In this case, the set of isometries may differ from the standard group of monomial transformations Mon_n. As such, π should be sampled from the stabilizer of the specific error set, to ensure that the obfuscated error $\pi(e)$ remains valid.

3.2 Proof of Equality of Secrets

The Chaum-Pedersen protocol [26] is a fundamental primitive in discrete logarithm cryptography, wherein one can prove that a pair of values (g^x, h^x) share the same exponent x, relative to group generators $g, h \in \mathbb{G}$. This concept is widely deployed in privacy-preserving systems to prove that two Pedersen commitments, $C_1 = g^x h^{r_1}$ and $C_2 = g^x h^{r_2}$, hide the same secret x despite utilizing different randomness. This same concept, when defined over a group of unknown order instead of a prime-order group, is named differently: the Fujisaki-Okamoto scheme [38].

Algorithm 1.2 represents a code-based analogue to this protocol. It allows a prover to demonstrate that two syndromes, $s_1 = H_1 e + T_1 r_1$ and $s_2 = H_2 e + T_2 r_2$, commit to the same error vector $e \in \mathcal{B}(t_1)$. Critically, this property can still be demonstrated even if the commitments rely on distinct parity-check matrices (H_1, T_1) versus (H_2, T_2), as well as distinct blinders r_1, r_2. Functionally, this serves as a proof of rerandomization, an oft-utilized tool in cryptocurrency transactions for linking commitments without revealing their secrets.

Theorem 1. *Algorithm 1.2 is a $(q-1, 2)$ IS ZKP, with soundness error $\frac{q}{2(q-1)}$.*

Proof. **Completeness**: completeness can be readily seen.

Soundness: suppose that, for $\xi \neq \xi'$, we have two pairs of accepting transcripts:

$$\begin{aligned}
\tau_1 &= ((c_1, c_2), \xi, (y_i), b, (\pi_i)),\\
\tau_2 &= ((c_1, c_2), \xi, (y_i), b', (\pi_1(e), \pi_2(r_1), \pi_3(r_2)),\\
\tau_3 &= ((c_1, c_2), \xi', (y'_i), b, (\pi'_i)),\\
\tau_4 &= ((c_1, c_2), \xi', (y'_i), b', (\pi'_1(e), \pi'_2(r_1), \pi'_3(r_2)).
\end{aligned}$$

Since c_1 is a valid commitment for τ_1 and τ_3, it must be that $\mathsf{com}(\pi_i, \delta_j) = c_1 = \mathsf{com}(\pi'_i, \delta'_j)$, which, assuming hash collision are hard, implies $\pi_i = \pi'_i$ and $\delta_j = \delta'_j$. As all the monomial transformations are equivalent to their prime version, we use them interchangeably. Thus,

$$\begin{aligned}
0 = \delta_1 - \delta'_1 &= H_1 \pi_1^{-1}(y_1 - y'_1) + T_1 \pi_2^{-1}(y_2 - y'_2) - (\xi - \xi')s_1\\
0 = \delta_2 - \delta'_2 &= H_2 \pi_1^{-1}(y_1 - y'_1) + T_2 \pi_3^{-1}(y_3 - y'_3) - (\xi - \xi')s_2.
\end{aligned}$$

Now from τ_2 and τ_4, $\mathsf{com}(\varepsilon_i, \pi_1(e), \pi_{j+1}(r_j)) = c_2 = \mathsf{com}(\varepsilon'_i, \pi_1(e), \pi_{j+1}(r_j))$, as $\pi_i = \pi'_i$. Hence

$$\begin{aligned}
0 = \varepsilon_1 - \varepsilon'_1 &= (y_1 - y'_1) - (\xi - \xi')\pi_1(e), \\
0 = \varepsilon_2 - \varepsilon'_2 &= (y_2 - y'_2) - (\xi - \xi')\pi_2(r_1), \\
0 = \varepsilon_3 - \varepsilon'_3 &= (y_3 - y'_3) - (\xi - \xi')\pi_3(r_2).
\end{aligned}$$

Substituting these into the above,

$$\begin{aligned}
(\xi - \xi')s_1 &= (\xi - \xi')H_1 e + (\xi - \xi')T_1 r_1 \\
(\xi - \xi')s_2 &= (\xi - \xi')H_2 e + (\xi - \xi')T_2 r_2,
\end{aligned}$$

which, along with verifying the weight condition, verifies the prover has knowledge of a solution to the C-SDP.

Zero-Knowledge: in order to appear to pass the verifier's check, suppose the dishonest prover is preparing for the challenge $b = 1$. In this case, they select the u_i randomly, then solve the syndromes s_1, s_2 for e, r_1, r_2 ignoring the weight condition; c_1 is constructed honestly, but c_2 is random. In this manner, they will pass the challenge $b = 1$, by construction. For $b = 2$ however, they will select e, r_1, r_2 of the appropriate weight, ignoring the syndrome condition. Commitment c_1 is now random, while c_2 honest. Then, they can pass the challenge $b = 2$. As this fraudulent prover can prepare for either challenge, they can appear to pass perfectly if they know the verifier's sequence of challenges in advance - since this is indistinguishable from a genuine interaction, thus the algorithm is zero-knowledge.

3.3 Proof of Sum of Secrets

In Algorithm 1.3, we present a zero-knowledge argument for additive relationships between committed secrets. Specifically, given two commitments $C_1 = H_1 x + T_1 r_1$ and $C_2 = H_2 y + T_2 r_2$, the prover demonstrates that a third commitment C_3 conceals the sum of the secrets, $C_3 = H_3(x + y) + T_3 r_3$.

Unlike in the homomorphic setting where C_3 could be derived by simply adding $C_1 + C_2$, assuming $H_1 = H_2$ and $T_1 = T_2$, this protocol supports commitments defined by distinct parity-check matrices, enabling homomorphic addition even when commitments utilize different parameters - an important improvement compared to simple linearity. Furthermore, it natively includes a rerandomization step: the blinder r_3 for the sum may be selected independently, instead of being constrained to $r_1 + r_2$. This ensures that the third commitment leaks no information about the individual blinders of the inputs.

Note that when compared to discrete logarithm setting, a critical distinction in code-based cryptography is that the sum of two independent weight-t error vectors is generally not contained in $\mathcal{B}(t)$, which may have security ramifications. To address this, we rely on the hardness of the C-SDP. So long as the blinders r_i are selected uniformly such that they form a difficult instance of the SDP, the secrets x, y and their sum $x + y$ remains hidden, regardless of whether $x + y$ strictly satisfies the original weight-t requirement.

Theorem 2. *Algorithm 1.3 is a* $(q-1,2)$ *IS ZKP, with soundness error* $\frac{q}{2(q-1)}$.

Proof. **Completeness**: this is straightforward.

Soundness: suppose one receives valid transcripts:

$$\begin{aligned}
\tau_1 &= ((c_1,c_2),\xi,(f_j,g_i),b,(\pi_j,\rho_i)),\\
\tau_2 &= ((c_1,c_2),\xi,(f_j,g_i),b',(\pi_j(e_j),\rho_i(r_i))),\\
\tau_3 &= ((c_1,c_2),\xi',(f'_j,g'_i),b,(\pi'_j,\rho'_i)),\\
\tau_4 &= ((c_1,c_2),\xi',(f'_j,g'_i),b',(\pi'_j(e_j),\rho'_i(r_i))),
\end{aligned}$$

for distinct values of ξ and ξ'. Since c_1 is a valid commitment for both τ_1 and τ_3, it must be that

$$\mathsf{com}(\pi_j,\rho_i,\delta_i) = c_1 = \mathsf{com}(\pi'_j,\rho'_i,\delta'_i),$$

meaning that either a hash collision has occurred or else $\pi_j = \pi'_j$, $\rho_i = \rho'_i$, and $\delta_i = \delta'_i$. By considering the delta equalities, we obtain that

$$\begin{aligned}
0 = \delta_1 - \delta'_1 &= H_1\pi_1^{-1}(f_1 - f'_1) + T_1\rho_1^{-1}(g_1 - g'_1) - (\xi - \xi')s_1\\
0 = \delta_2 - \delta'_2 &= H_2\pi_2^{-1}(f_2 - f'_2) + T_2\rho_2^{-1}(g_2 - g'_2) - (\xi - \xi')s_2\\
0 = \delta_3 - \delta'_3 &= H_3\left[\pi_1^{-1}(f_1 - f'_1) + \pi_2^{-1}(f_2 - f'_2)\right] - (\xi - \xi')s_3\\
&\quad + T_3\left[\rho_1^{-1}(g_1 - g'_1) + \rho_2^{-1}(g_2 - g'_2) + \rho_3^{-1}(g_3 - g'_3)\right].
\end{aligned}$$

Similarly, comparing τ_2 and τ_4, it must be that

$$\mathsf{com}(\varepsilon_j,\phi_i) = c_2 = \mathsf{com}(\varepsilon'_j,\phi_i).$$

From this, unless a hash collision has been found, we gather that $\varepsilon_j = \varepsilon'_j$. Since we already know that $\pi_j = \pi'_j$ and $\rho_i = \rho'_i$,

$$\begin{aligned}
0 = \varepsilon_j - \varepsilon'_j &= (f_j - f'_j) - (\xi - \xi')\pi_j(e_j),\\
0 = \phi_i - \phi'_i &= (g_i - g'_i) - (\xi - \xi')\rho_i(r_i).
\end{aligned}$$

Substituting these into what we obtained from the previous transcripts, we find that

$$\begin{aligned}
(\xi - \xi')s_1 &= H_1\pi_1^{-1}((\xi - \xi')\pi_1(x)) + T_1\rho_1^{-1}((\xi - \xi')\rho_1(r_1))\\
(\xi - \xi')s_2 &= H_2\pi_2^{-1}((\xi - \xi')\pi_2(y)) + T_2\rho_2^{-1}((\xi - \xi')\rho_2(r_2))\\
(\xi - \xi')s_3 &= H_3\left[\pi_1^{-1}((\xi - \xi')\pi_1(x)) + \pi_2^{-1}((\xi - \xi')\pi_2(y))\right]\\
&\quad + T_3\left[\rho_1^{-1}((\xi - \xi')\rho_1(r_1)) + \rho_2^{-1}((\xi - \xi')\rho_2(r_2))\right.\\
&\quad\quad \left. + \rho_3^{-1}((\xi - \xi')\rho_3(r_3))\right].
\end{aligned}$$

After simplifying, $s_1 = H_1e_1 + T_1r_1$, that $s_2 = H_2e_2 + T_2r_2$, and $s_3 = H_3(e_1 + e_2) + T_3r_3$.

Thus the prover does actually have knowledge of the relationship claimed, hence this protocol is a $(q-1,2)$ IS from Definition 5, thus enjoys a soundness error of $\frac{q}{2(q-1)}$.

Zero-Knowledge: this property can be seen in the same vein as Algorithm 1.2, by preparing to respond with an error that has either the correct syndrome or the correct weight based on the verifier's challenge.

3.4 Proof of Matrix Product

In Algorithm 1.4, we introduce a protocol to verify multiplicative relationships between committed secrets. Specifically, given commitments to two matrices M_1 and M_2, the prover demonstrates that a third commitment conceals their product $M_3 = M_1 M_2$.

Change of Metric: Unlike the previous protocols in this paper which operated on vectors under the Hamming metric, this scheme utilizes matrices in $\mathbb{F}_q^{m \times n}$ and relies on the hardness of the Rank Syndrome Decoding Problem, Problem 2. Consequently, the "weight" of a secret is measured by its rank r rather than its Hamming weight t. The underlying identification scheme structure draws upon the Rank-CVE framework established in [40,14].

While designed for generic matrix multiplication, this protocol is highly versatile: it trivially supports matrix-vector products in the case $n = 1$, and significantly, can be specialized for Schur (or Hadamard) products in the obvious manner. On one hand, by embedding vectors into diagonal matrices, we can utilize this algorithm as stated to demonstrate element-wise product relationships of the form $z = x \star y$; on the other hand, exchanging matrix multiplication for the Hadamard product results in a perfectly valid ZKP for the relationship $M_1 \star M_2 = M_3$, which can then be specialised to vectors for the Schur product. This adaptation effectively bridges the gap back to the Hamming metric, enabling the binary check and arithmetic circuits we explore in Section 4.

Conceptually, this protocol is the code-based analogue of discrete logarithm arguments to verify Diffie-Hellman triples [54]. In that setting, given commitments g^x and g^y, one proves that a third value is of the form g^{xy}, showing that it is indeed a Diffie-Hellman shared secret. This idea was expanded to support Pedersen commitments in [24,74], where one can demonstrate in zero-knowledge from $C_1 = g^x h^{r_1}$ and $C_2 = g^y h^{r_2}$ that that a third commitment C_3 must be of the form $g^{xy} h^{r_3}$. Just as these primitives are essential for constructing Multi-Party Computation (MPC) protocols using Beaver triples [13], our scheme provides a post-quantum foundation for MPC, and thus, verifiable computation.

Theorem 3. *Algorithm 1.4 is a* $(3,2)$*-out-of-*$(q-1,2)$ *special sound ZKP, with soundness error* $\frac{q+1}{2(q-1)}$.

Proof. **Completeness**: this property is apparent.

Soundness: suppose that one obtains three pairs of valid transcripts:

$$\begin{aligned}
\tau_1 &= ((c_1, c_2), z, \xi, (f_j, g_i), b, (\pi_i)),\\
\tau_2 &= ((c_1, c_2), z, \xi, (f_j, g_i), b', (\pi_j(M_j), \pi_i(R_i))),\\
\tau_3 &= ((c_1, c_2), z, \xi', (f'_j, g'_i), b, (\pi'_i)),\\
\tau_4 &= ((c_1, c_2), z, \xi', (f'_j, g'_i), b', (\pi'_j(M_j), \pi'_i(R_i))),\\
\tau_5 &= ((c_1, c_2), z, \xi'', (f''_j, g''_i), b, (\pi''_i)),\\
\tau_6 &= ((c_1, c_2), z, \xi'', (f''_j, g''_i), b', (\pi'_j(M_j), \pi'_i(R_i))),
\end{aligned}$$

for distinct values of ξ, ξ', and ξ''. For commitment c_1 to be valid for both τ_1 and τ_3, it must be that $\mathsf{com}(\pi_i, \delta_i) = c_1 = \mathsf{com}(\pi'_i, \delta'_i)$, which necessitates that, presuming hash collisions are intractable, that each monomial transformation π_i is equal to its prime π'_i, and from τ_5 also equal to its double prime π''_i.

From comparing the deltas, we obtain

$$\begin{aligned}
0 = \delta_1 - \delta'_1 &= H_1\pi_1^{-1}(f_1 - f'_1) + T_1\pi_1^{-1}(g_1 - g'_1) - (\xi - \xi')s_1\\
0 = \delta_2 - \delta'_2 &= H_2\pi_2^{-1}(f_2 - f'_2) + T_2\pi_2^{-1}(g_2 - g'_2) - (\xi - \xi')s_2\\
0 = \delta_3 - \delta'_3 &= H_3\left[\pi_1^{-1}(f_1)\pi_2^{-1}(f_2) - \pi_1^{-1}(f'_1)\pi_2^{-1}(f'_2)\right] - (\xi - \xi')z\\
&\quad + T_3\left[\pi_1^{-1}(g_1)\pi_2^{-1}(g_2) - \pi_1^{-1}(g'_1)\pi_2^{-1}(g'_2) + \xi\pi_3^{-1}(g_3) - \xi'\rho_3^{-1}(g'_3)\right]\\
&\quad - (\xi^2 - \xi'^2)s_3.
\end{aligned}$$

Note $\mathsf{com}(\pi_j(M_j), \pi_i(R_i), \varepsilon_j, \phi_i) = c_2 = \mathsf{com}(\pi_j(M_j), \pi_i(R_i), \varepsilon'_j, \phi'_i)$ from comparing τ_2 and τ_4, implying $\varepsilon_j = \varepsilon'_j$ and $\phi_i = \phi'_i$. Thus,

$$\begin{aligned}
0 = \varepsilon_j - \varepsilon'_j &= (f_j - f'_j) - (\xi - \xi')\pi_j(M_j),\\
0 = \phi_i - \phi'_i &= (g_i - g'_i) - (\xi - \xi')\pi_i(R_i),
\end{aligned}$$

which can be rearranged into

$$\begin{aligned}
\pi_j^{-1}(f'_j) &= \pi_j^{-1}(f_j) - (\xi - \xi')M_j,\\
\pi_i^{-1}(g'_i) &= \pi_i^{-1}(g_i) - (\xi - \xi')R_j.
\end{aligned}$$

Substituting this into what was found earlier,

$$\begin{aligned}
0 &= (\xi - \xi')H_jM_j + (\xi - \xi')T_jR_j - (\xi - \xi')s_j\\
0 &= H_3\left[\pi_1^{-1}(f_1)\pi_2^{-1}(f_2) - (\pi_1^{-1}(f_1) - (\xi - \xi')M_1)(\pi_2^{-1}(f_2) - (\xi - \xi')M_2)\right]\\
&\quad + T_3\left[\pi_1^{-1}(g_1)\pi_2^{-1}(g_2) - (\pi_1^{-1}(g_1) - (\xi - \xi')R_1)(\pi_2^{-1}(g_2) - (\xi - \xi')R_2)\right]\\
&\quad + T_3\left[\xi\pi_3^{-1}(g_3) - \xi'(\pi_3^{-1}(g_3) - (\xi - \xi')R_3)\right] - (\xi - \xi')z\\
&\quad - (\xi^2 - \xi'^2)s_3.
\end{aligned}$$

After removing a factor of $(\xi - \xi')$, which is non-zero by distinctness of ξ and ξ', the first two equations above simplify into $H_1M_1 + T_1R_1 = s_1$ and $H_2M_2 + T_2R_2 = s_2$ exactly as claimed.

Now, consider τ_1, τ_3, and τ_5, with distinct challenges ξ, ξ', and ξ''. Because the prover's responses are all linear in their respective challenge, these are increased to quadratic at most by the verifier, thus we can uniquely interpolate along the 3 distinct challenges. Introducing the indeterminate x, we can then form linear polynomials $f_1(x)$, $f_2(x)$, $g_1(x)$, $g_2(x)$, and $g_3(x)$. Concretely, $\pi_1^{-1}(f_1(x)) = f_{1,c} + xf_{1,\ell}$, $\ldots$, $\pi_3^{-1}(g_3(x)) = g_{3,c} + xg_{3,\ell}$, identifying them by constant and linear terms.

Substituting $s_1 = H_1M_1 + T_1R_1$ into the verifier's δ_1 calculation, then

$$\begin{aligned}
\delta_1 &= H_1\pi_1^{-1}(f_1(x)) + T\rho_1^{-1}(g_1(x)) - xs_1 \\
&= H(f_{1,c} + xf_{1,\ell}) + T(g_{1,c} + xg_{1,\ell}) - xs_1 \\
&= [H(f_{1,c}) + T(g_{1,c})] + x\left[H(f_{1,\ell}) + T(g_{1,\ell}) - s_1\right] \\
&= [H(f_{1,c}) + T(g_{1,c})] + x\left[H(f_{1,\ell} - M_1) + T(g_{1,\ell} - R_1)\right].
\end{aligned}$$

For this to truly equal $d_1 = H_1u_1 + T_1v_1$ from the commitment c_1, which as a committed value must necessarily have no x term, it must be that $f_{1,\ell} = M_1$ and $g_{1,\ell} = R_1$, then also that $f_{1,c} = u_1$ and $g_{1,c} = v_1$.

Similarly for δ_2, one finds that

$$\delta_2 = [H'(f_{2,c}) + T'(g_{2,c})] + c\left[H'(f_{2,\ell} - M_2) + T'(g_{2,\ell} - R_2)\right],$$

demonstrating that $f_{2,\ell} = M_2$ and $g_{2,\ell} = R_2$, then also that $f_{2,c} = u_2$ and $g_{2,c} = v_2$.

For δ_3, applying what we found above,

$$\begin{aligned}
\delta_3 &= H_3\left[\pi_1^{-1}(f_1(x))\pi_2^{-1}(f_2(x))\right] + T_3\left[\pi_1^{-1}(g_1(x))\pi_2^{-1}(g_2(x)) + z\pi_3^{-1}(g_3(x))\right] \\
&\quad - xz - x^2s_3 \\
&= H_3\left[(u_1 + xM_1)(u_2 + xM_2)\right] + T_3\left[(v_1 + xR_1)(v_2 + xR_2) + x(g_{3,c} + xg_{3,\ell})\right] \\
&\quad - xz - x^2s_3 \\
&= (H_3\left[u_1u_2\right] + T_3\left[v_1v_2\right]) + x\left(H_3\left[M_1u_2 + u_1M_2\right] + T_3\left[R_1v_2 + v_1R_2 + g_{3,c}\right] - z\right) \\
&\quad + x^2\left(H_3\left[M_1M_2\right] + T_3\left[R_1R_2 + g_{3,\ell}\right] - s_3\right).
\end{aligned}$$

For this to equal $d_3 = H_3(u_1u_2) + T_3(v_1v_2)$ from the commitment c_1, which has no x term, it must be that the linear and quadratic terms above vanish. If the prover does pass the verifier's checks, then the verifier is convinced from the linear terms that z is formed properly. They are also convinced by the quadratic terms that the secret hidden by H_3 in s_3 is $M_1M_2 = M_3$, as purported. Additionally, it must be that $g_{3,\ell} = R_3 - R_1R_2$, hence all the prover's responses are properly formed, and the prover must possess knowledge of the secrets they claim to.

It follows then from [6,5] that Algorithm 1.4 is $(3, 2)$-out-of-$(q - 1, 2)$ special sound, with soundness error of

$$1 - \left(1 - \frac{3-1}{q-1}\right)\left(1 - \frac{2-1}{2}\right) = \frac{1}{2} + \frac{1}{q-1}.$$

Zero-Knowledge: the proof here is quite similar to the one in Algorithm 1.2, so we omit it.

4 Applications

In Section 3, we established a collection of protocols for verifying algebraic relationships between committed secrets. Algorithm 1.2 to demonstrate opening capabilities, Algorithm 1.3 to show a sum of secrets, and Algorithm 1.4 to verify matrix products are powerful primitives that can be adapted to handle matrix-vector products and, via diagonal embedding, Schur products of vectors.

In this section, we combine these schemes to construct higher-level cryptographic primitives. We begin with applications to standard cryptographic operations - verifying Beaver triples and public keys - before introducing a general arithmetic circuit framework. Finally, we demonstrate how these tools enable post-quantum analogues of advanced privacy mechanisms, including range proofs, lookup arguments, and verifiable shuffles.

4.1 Beaver Triples

A direct application of Algorithm 1.4, when thought of as a vector Schur product scheme, is the verification of Beaver triples for MPC.

In MPC, one has n communicating servers who have split a secret s between them but can only perform operations on their individual shares $[s]$, with the eventual goal of recombining them to obtain a function on the secret s without revealing it. One computational bottleneck of MPC is real-time multiplication of shares, which is possible, but inefficient and costly. Beaver's trick permits users to perform an inefficient product protocol in advance on a random triple (a, b, c) such that $c = a \star b$, then later "sacrifice" this chaff triple in order to efficiently obtain the product of shares.

Ensuring these triples are generated correctly is critical for secure MPC techniques. Our protocol allows a party to generate a batch of triples, (a_i, b_i, c_i) such that $c_i = a_i \star b_i$, then prove their validity in zero-knowledge - while ensuring the values are hidden behind the difficulty of the SDP. Unlike related frameworks [35] which require threshold cryptography, our approach avoids the need for such techniques. For further background on MPC, including post-quantum directions, we refer to [48,31,34].

4.2 Properly Formed Keys

Our protocols can also be used to verify the structure of keys in code-based cryptosystems. For instance, the CROSS signature scheme [8] utilizes full-weight error vectors with entries from a multiplicative subgroup of the form $\mathbb{E} = \{g^i \mid i = 1, \ldots, z\}$. Storing the exponents of these entries instead of the field elements themselves logarithmically reduces key size and dramatically speeds up the computation of the syndrome check.

Using the Schur product scheme, a prover can demonstrate that a third commitment contains a secret formed from the *sum* of the exponents of two other committed secrets. This corresponds to the product of the group elements $g^{x_i} \cdot g^{y_i} = g^{x_i + y_i}$. Similarly, one can prove that a committed vector is a valid

WAVE key [9], which is constructed from sums of Schur products, or verify that a CROSS key satisfies the required entry conditions.

4.3 Binary Checks & Arithmetic Circuits

The most versatile application of our framework is the construction of arithmetic circuits or Rank-1 Constraint Systems (R1CS) [15,57].

Binary Check: A foundational gadget in this framework is the proof that a committed vector $e \in \mathbb{F}_q^n$ is binary. Recall that $\mathbb{1}$ denotes the all-ones vector. Given commitments $s_1 = H_1 e + T_1 r_1$ and $s_2 = H_2(\mathbb{1} - e) + T_2 r_2$, we first leverage the sum scheme show that the underlying secrets sum to $\mathbb{1}$. We then utilize the Schur product framework to prove that the product of the secrets is zero: $e \star (\mathbb{1} - e) = \mathbb{0}$. Since $x(1-x) = 0$ if and only if $x \in \{0, 1\}$, this conclusively proves e is a binary vector.

Arithmetic Circuits: Once a binary constraint has been established, we can proceed to implement Boolean logic. For example, to enforce a conditional selection $b_i = e_i \cdot u_i + (1 - e_i) \cdot v_i$ (i.e., "if $e_i = 1$ let $b_i = u_i$, else v_i"), the prover simply commits to u, v, e, then uses the sum and product protocols to verify the arithmetic relation $b = e \star u + (\mathbb{1} - e) \star v$ predicated on e. Naturally, this can be generalized to more complex predicates such as 'if $e_i = 1$ let $b_i = u_i v_i$, else $u_i - v_i$."

This capability allows us to verify solutions to any arbitrary NP-Complete problem in zero-knowledge. Consider the graph 3-coloring problem; while all planar graphs admit a 4-coloring [4], determining if 3 colors are sufficient is an NP-Complete problem [32]. To prove a valid coloring with colors $\{1, 2, 3\}$, we represent the color of node a as a field element in $\mathbb{F}_q$ and enforce:

- **Validity:** $(1 - a)(2 - a)(3 - a) = 0$ for all nodes a.
- **Adjacency:** $a \neq b$ for all neighbors a, b. This can be arithmetized as checking the polynomial constraint $(2 - ab)(3 - ab)(6 - ab) = 0$.

While physical ZKP protocols for this exist, such as using card decks [55] or lockboxes [43], our method provides a purely code-based cryptographic solution.

One important consequence for arithmetic circuits is the possibility of a code-based cryptocurrency: in Monero, an important upcoming upgrade is FCMP++ [60]. With the protocols in this paper, mutatis mutandis, many of the FCMP++ discrete logarithm protocols can be transferred over - with care - to a code-based setting, such as the membership or ownership proofs.

4.4 A Rudimentary Post-Quantum Range Proof

Zero-knowledge range proofs [22,29,33], a framework for demonstrating that a committed integer a lies in a small interval $[0, 2^n - 1]$, are an essential primitive for preventing inflationary bugs in privacy coins such as Monero [58], MimbleWimble [73], and more [59].

We present a rudimentary code-based range proof based on committed integer comparison. Let $a, b \in \{0,1\}^n$ denote the binary decompositions of integers α and β. To prove $\alpha \geq \beta$, ensuring that a transaction does not spend more funds than it has access to in the account, we construct the following arithmetic circuit:

1. Verify a and b are binary using the binary check from the previous section.
2. Compute the vector c corresponding to the binary representation of $2^n + \alpha - \beta$.
3. Verify that the $(n+1)$-th bit of c is 1.

Mathematically, if $\alpha, \beta \in [0, 2^n - 1]$, then $2^n + \alpha - \beta \geq 2^n$ if and only if $\alpha \geq \beta$. Thus, checking the overflow bit of their difference is equivalent to a range check. While less succinct than inner-product arguments like Bulletproofs [22], this approach relies solely on code-based assumptions, like the C-SDP. For other range proving methods, see [28,42].

4.5 Lookup Arguments

Lookup arguments permit a prover to show that elements of a committed vector $a \in \mathbb{F}_q^m$ may also be found within a public table $A \in \mathbb{F}_q^n$, with $m < n$, in effect "looking up" the entries. This is a critical component for privacy-oriented cryptocurrencies such as Zcash [79] and zero-knowledge virtual machines [72,78].

We focus on the subvector extraction technique, which can be found frequently in the literature either explicitly or implicitly [77,63,78,68,67], though many other methods exist [46]. In this, the prover demonstrates knowledge of a selection matrix $M \in \mathbb{F}_q^{m \times n}$ such that $MA = a$. For M to be a valid selection matrix, selecting exactly one element per row, it must be an elementary matrix - one in which every row is all 0 entries except for a single 1 - which is equivalent to the following two properties:

1. **Binary:** M must be binary, which can be easily verified using Algorithm 1.4 by checking idempotence under the Hadamard product, i.e.: that $M \star M = M$.
2. **Stochasticity:** Each row must sum to 1; this is verified by checking that $M\mathbb{1} = \mathbb{1}$ using the matrix-vector product adaptation.

By ensuring that M has the correct form, then proving $MA = a$, we instantiate a post-quantum lookup argument compatible with code-based commitments.

4.6 Verifiable Shuffle

Finally, we can extend the logic above to prove that a committed matrix $P \in \mathbb{F}_q^{n \times n}$ is a permutation matrix. This is required for verifiable shuffles in mix-nets [20,21] and voting protocols [56,71], though can also be used to demonstrate knowledge of a solution to the Permuted Kernel Problem from [17], or variants of the Code Equivalence Problem [18].

A matrix P is a permutation if and only if it is a binary matrix with double-stochasticity, where every row and every column sums to exactly 1. Using our framework, the prover commits to P, where it remains protected by the Rank-SDP, and demonstrates:

- $P \star P = P$ (entries are binary).
- $P\mathbb{1} = \mathbb{1}$ (rows sum to 1).
- $\mathbb{1}^\top P = \mathbb{1}^\top$ (columns sum to 1).

These checks demonstrate that P is a permutation matrix without revealing the permutation itself. Considering C_1 as a commitment to e_1 and C_2 as a commitment to e_2, a verifiable shuffle technique would enable a prover to show that there exists a permutation P such that $e_1 = Pe_2$; thus, by Algorithm 1.4, we have formulated a post-quantum verifiable shuffle scheme from purely code-based methods.

5 Conclusions

In this paper, we have presented a comprehensive collection of zero-knowledge protocols designed to bridge the gap between the privacy-preserving techniques of discrete logarithm cryptography and the post-quantum security of code-based cryptography. By exploiting the hardness of the Syndrome Decoding Problem and its variants, we have designed secure arguments for the foundational algebraic relationships that permit modern cryptographic systems - proofs of opening, sums, and products. Our contributions, specifically the matrix-product protocol from Algorithm 1.4, allow us to design arbitrary arithmetic circuits, enabling the verification of predicate logic and binary constraints while ensuring zero-knowledge.

These schemes are topical and practical, offering a concrete direction for privacy-preserving designs in cryptocurrencies, like FCMP++, to evolve into their post-quantum counterparts. We have leveraged the protocols presented herein to propose code-based analogues for mechanisms such as a rudimentary range proof, a lookup argument utilizing subvector extraction, and a verifiable shuffle protocol for mix-nets. By strictly adhering to standard coding theoretic assumptions, this work confirms that code-based cryptography represents a viable candidate for the next generation of robust confidential-transaction systems.

References

1. M. Ajtai. Generating Hard Instances of Lattice Problems. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 99–108, New York, NY, USA, 1996. Association for Computing Machinery.
2. Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A Code-Based Group Signature Scheme. Cryptology ePrint Archive, Paper 2016/1119, 2016.
3. Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. Cryptology ePrint Archive, Paper 2022/1608, 2022.

4. Kenneth Appel and Wolfgang Haken. *Every Planar Map is Four Colorable*. American Mathematical Society, Providence, RI, 1989.
5. Thomas Attema, Ronald Cramer, and Lisa Kohl. A Compressed Σ-Protocol Theory for Lattices. Cryptology ePrint Archive, Paper 2021/307, 2021.
6. Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-Shamir Transformation of Multi-Round Interactive Proofs. Cryptology ePrint Archive, Paper 2021/1377, 2021.
7. L Babai. Trading Group Theory for Randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 421–429, New York, NY, USA, 1985. Association for Computing Machinery.
8. Marco Baldi, Alessandro Barenghi, Sebastian Bitzer, Patrick Karl, Felice Manganiello, Alessio Pavoni, Gerardo Pelosi, Paolo Santini, Jonas Schupp, Freeman Slaughter, Antonia Wachter-Zeh, and Violetta Weger. 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: 26-Jan-2025.
9. Gustavo Banegas, Kévin Carrier, André Chailloux, Alain Couvreur, Thomas Debris-Alazard, Philippe Gaborit, Pierre Karpman, Johanna Loyer, Ruben Niederhagen, Nicolas Sendrier, Benjamin Smith, and Jean-Pierre Tillich. Wave Signature Scheme. NIST PQC, 2023.
10. Hannes Bartz, Lukas Holzbaur, Hedongliang Liu, Sven Puchinger, Julian Renner, and Antonia Wachter-Zeh. Rank-Metric Codes and Their Applications. *Foundations and Trends in Communications and Information Theory*, 19(3):391–546, 05 2022.
11. Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafael del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-Linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits. Cryptology ePrint Archive, Paper 2018/560, 2018.
12. Carsten Baum, Alex J. Malozemoff, Marc B. Rosen, and Peter Scholl. Mac'n'Cheese: Zero-Knowledge Proofs for Boolean and Arithmetic Circuits with Nested Disjunctions. Cryptology ePrint Archive, Paper 2020/1410, 2020.
13. Donald Beaver. Efficient Multiparty Protocols Using Circuit Randomization. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 420–432, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.
14. Emanuele Bellini, Florian Caullery, Alexandros Hasikos, Marcos Manzano, and Victor Mateu. Code-based signature schemes from identification protocols in the rank metric. In Jan Camenisch and Panos Papadimitratos, editors, *Cryptology and Network Security*, pages 277–298, Cham, 2018. Springer International Publishing.
15. Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge. Cryptology ePrint Archive, Paper 2013/507, 2013.
16. Daniel J. Bernstein and Tanja Lange. Post-Quantum Cryptography. *Nature*, 549(7671):188–194, September 2017.
17. Slim Bettaieb, Loïc Bidoux, Victor Dyseryn, Andre Esser, Philippe Gaborit, Mukul Kulkarni, and Marco Palumbi. PERK: Compact Signature Scheme Based on a New Variant of the Permuted Kernel Problem. Cryptology ePrint Archive, Paper 2024/748, 2024.
18. Jean-François Biasse and Giacomo Micheli. The Code Equivalence Problem and Its Applications to Cryptography. *La Matematica*, 4(2).
19. Olivier Blazy, Philippe Gaborit, Julien Schrek, and Nicolas Sendrier. A Code-Based Blind Signature. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2718–2722, 2017.

20. Jonathan Bootle, Vadim Lyubashevsky, and Antonio Merino-Gallardo. Efficient Verifiable Mixnets from Lattices, Revisited. Cryptology ePrint Archive, Paper 2025/658, 2025.
21. Philippe Bulens, Damien Giry, and Olivier Pereira. Running Mixnet-Based Elections with Helios. In *Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'11, page 6, USA, 2011. USENIX Association.
22. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short Proofs for Confidential Transactions and More. Cryptology ePrint Archive, Paper 2017/1066, 2017.
23. Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A Zero-Knowledge Identification Scheme Based on the q-ary SD Problem. In *SAC'2010*, volume 6544 of *SAC'2010*, pages 171–186, Waterloo, Canada, August 2010. Springer.
24. Sofia Celi, Shai Levin, and Joe Rowell. CDLS: Proving Knowledge of Committed Discrete Logarithms with Soundness. Cryptology ePrint Archive, Paper 2023/1595, 2023.
25. F. Chabaud and J. Stern. The Cryptographic Security of the Syndrome Decoding Problem for Rank Distance Codes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology — ASIACRYPT '96*, pages 368–381, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
26. David Chaum and Torben Pryds Pedersen. Wallet Databases with Observers. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 89–105, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.
27. Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass MQ-based identification to MQ-based signatures. Cryptology ePrint Archive, Paper 2016/708, 2016.
28. Miranda Christ, Foteini Baldimtsi, Konstantinos Kryptos Chalkias, Deepak Maram, Arnab Roy, and Joy Wang. SoK: Zero-Knowledge Range Proofs. In Rainer Böhme and Lucianna Kiffer, editors, *6th Conference on Advances in Financial Technologies, AFT 2024, September 23-25, 2024, Vienna, Austria*, volume 316 of *LIPIcs*, pages 14:1–14:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
29. Heewon Chung, Kyoohyung Han, Chanyang Ju, Myungsun Kim, and Jae Hong Seo. Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger. Cryptology ePrint Archive, Paper 2020/735, 2020.
30. Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, January 1997.
31. Ronald Cramer, {Ivan Bjerre} Damgård, and {Jesper Buus} Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
32. David P. Dailey. Uniqueness of Colorability and Colorability of Planar 4-Regular Graphs are NP-Complete. *Discrete Mathematics*, 30(3):289–293, 1980.
33. Liam Eagen, Sanket Kanjalkar, Tim Ruffing, and Jonas Nick. Bulletproofs++: Next Generation Confidential Transactions via Reciprocal Set Membership Arguments. Cryptology ePrint Archive, Paper 2022/510, 2022.
34. David Evans, Vladimir Kolesnikov, and Mike Rosulek. A Pragmatic Introduction to Secure Multi-Party Computation. *Found. Trends Priv. Secur.*, 2(2–3):70–246, December 2018.
35. Thibauld Feneuil. Building MPCitH-based Signatures from MQ, MinRank, Rank SD and PKP. Cryptology ePrint Archive, Paper 2022/1512, 2022.

36. Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Proceedings on Advances in Cryptology—CRYPTO '86*, page 186–194, Berlin, Heidelberg, 1987. Springer-Verlag.
37. U. Fiege, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 210–217, New York, NY, USA, 1987. Association for Computing Machinery.
38. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In *Annual International Cryptology Conference*, 1997.
39. Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the Complexity of the Rank Syndrome Decoding Problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, 2016.
40. Philippe Gaborit, Julien Schrek, and Gilles Zémor. Full Cryptanalysis of the Chen Identification Protocol. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 35–50, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
41. Philippe Gaborit and Gilles Zémor. On the Hardness of the Decoding and the Minimum Distance Problems for Rank Codes. *IEEE Transactions on Information Theory*, 62(12):7245–7252, 2016.
42. Rui Gao, Zhiguo Wan, Yuncong Hu, and Huaqun Wang. A Succinct Range Proof for Polynomial-based Vector Commitment. Cryptology ePrint Archive, Paper 2024/1016, 2024.
43. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that Yield Nothing but their Validity or all Languages in NP have Zero-Knowledge Proof Systems. *J. ACM*, 38(3):690–728, July 1991.
44. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.
45. Alexander Golovnev, Jonathan Lee, Srinath Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-Time and Field-Agnostic SNARKs for R1CS. Cryptology ePrint Archive, Paper 2021/1043, 2021.
46. Hossein Hafezi, Gaspard Anthoine, Matteo Campanelli, and Dario Fiore. SoK: Lookup Table Arguments. Cryptology ePrint Archive, Paper 2025/1876, 2025.
47. Carmit Hazay and Yehuda Lindell. *Sigma Protocols and Efficient Zero-Knowledge1*, pages 147–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
48. Martin Hirt, Ueli Maurer, and Bartosz Przydatek. Efficient Secure Multi-party Computation. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, pages 143–161, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
49. Intak Hwang, Jinyeong Seo, and Yongsoo Song. Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions. Cryptology ePrint Archive, Paper 2024/306, 2024.
50. Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. Cryptology ePrint Archive, Paper 2012/513, 2012.
51. Daniel Kales and Greg Zaverucha. An Attack on Some Signature Schemes Constructed From Five-Pass Identification Schemes. Cryptology ePrint Archive, Paper 2020/837, 2020.
52. Stephan Krenn and Michele Orrù. Proposal: σ-protocols. 2021.
53. Felice Manganiello and Freeman Slaughter. Generic Error SDP and Generic Error CVE. Cryptology ePrint Archive, Paper 2023/717, 2023.

54. Ueli Maurer. Unifying Zero-Knowledge Proofs of Knowledge. In Bart Preneel, editor, *Progress in Cryptology – AFRICACRYPT 2009*, pages 272–286, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
55. Daiki Miyahara, Hiromichi Haneda, and Takaaki Mizuki. Card-Based Zero-Knowledge Proof Protocols for Graph Problems and Their Computational Model. In *Provable and Practical Security: 15th International Conference, ProvSec 2021, Guangzhou, China, November 5–8, 2021, Proceedings*, page 136–152, Berlin, Heidelberg, 2021. Springer-Verlag.
56. C. Andrew Neff. A Verifiable Secret Shuffle and its Application to e-Voting. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, CCS '01, page 116–125, New York, NY, USA, 2001. Association for Computing Machinery.
57. Ngoc Khanh Nguyen and Gregor Seiler. Practical Sublinear Proofs for R1CS from Lattices. Cryptology ePrint Archive, Paper 2022/1048, 2022.
58. Sarang Noether. Bulletproofs+ in Monero. `https://www.getmonero.org/2020/12/24/Bulletproofs+-in-Monero.html`. Accessed: 2026-Feb-3.
59. Rachel-Rose O'Leary. Aim, Fire: Bulletproofs Is a Crypto Privacy Breakthrough. `https://www.coindesk.com/markets/2018/01/02/aim-fire-bulletproofs-is-a-crypto-privacy-breakthrough`. Accessed: 2026-Feb-3.
60. Luke "Kayaba" Parker. FCMP++. `https://github.com/kayabaNerve/fcmp-plus-plus-paper/blob/develop/fcmp%2B%2B.pdf`. Accessed: 2026-Feb-4.
61. Luke "Kayaba" Parker. Full-Chain Membership Proofs Development. `https://www.getmonero.org/2024/04/27/fcmps.html`. Accessed: 2025-Oct-6.
62. Torben Pryds Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 129–140, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.
63. Jim Posen and Assimakis A. Kattis. Caulk+: Table-Independent Lookup Arguments. Cryptology ePrint Archive, Paper 2022/957, 2022.
64. Alberto Ravagnani. Coding Theory [Lecture Notes], 2024.
65. Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Hardness Estimates of the Code Equivalence Problem in the Rank Metric. Cryptology ePrint Archive, Paper 2022/276, 2022.
66. Claus-Peter Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4:161–174, 2004.
67. Srinath Setty and Justin Thaler. Twist and Shout: Faster Memory Checking Arguments via One-Hot Addressing and Increments. Cryptology ePrint Archive, Paper 2025/105, 2025.
68. Srinath Setty, Justin Thaler, and Riad Wahby. Unlocking the Lookup Singularity with Lasso. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 180–209, Cham, 2024. Springer Nature Switzerland.
69. J. Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996. `https://ieeexplore.ieee.org/document/556672`.
70. Jacques Stern. A New Identification Scheme Based on Syndrome Decoding. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 13–21, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg. `https://doi.org/10.1007/3-540-48329-2_2`.
71. Martin Strand. A Verifiable Shuffle for the GSW Cryptosystem. Cryptology ePrint Archive, Paper 2018/027, 2018.

72. Justin Thaler. FAQ on Jolt's Initial Implementation. https://a16zcrypto.com/posts/article/faqs-on-jolts-initial-implementation/. Accessed: 2026-Feb-5.
73. Tari Labs University. Bulletproofs and Mimblewimble. https://tlu.tarilabs.com/cryptography/bulletproofs-and-mimblewimble. Accessed: 2026-Feb-3.
74. Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. Doubly-Efficient zkSNARKs Without Trusted Setup. Cryptology ePrint Archive, Paper 2017/1132, 2017.
75. Chenkai Weng, Kang Yang, Jonathan Katz, and Xiao Wang. Wolverine: Fast, Scalable, and Communication-Efficient Zero-Knowledge Proofs for Boolean and Arithmetic Circuits. Cryptology ePrint Archive, Paper 2020/925, 2020.
76. Tiancheng Xie, Yupeng Zhang, and Dawn Song. Orion: Zero Knowledge Proof with Linear Prover Time. Cryptology ePrint Archive, Paper 2022/1010, 2022.
77. Arantxa Zapico, Vitalik Buterin, Dmitry Khovratovich, Mary Maller, Anca Nitulescu, and Mark Simkin. Caulk: Lookup Arguments in Sublinear Time. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 3121–3134, New York, NY, USA, 2022. Association for Computing Machinery.
78. Arantxa Zapico, Ariel Gabizon, Dmitry Khovratovich, Mary Maller, and Carla Ràfols. Baloo: Nearly Optimal Lookup Arguments. Cryptology ePrint Archive, Paper 2022/1565, 2022.
79. Zcash. The halo2 Book. https://zcash.github.io/halo2/design/proving-system/lookup.html. Accessed: 2026-Feb-5.

Equality of Secrets		
Public: $q, n, k \in \mathbb{N}$, $H_1, H_2, T_1, T_2 \in \mathbb{F}_q^{(n-k)\times n}$, $H_1 e + T_1 r_1 = s_1$, $H_2 e + T_2 r_2 = s_2$		
Private: $e \in \mathcal{B}(t_1)$, $r_1 \in \mathcal{B}(t_2)$, $r_2 \in \mathcal{B}(t_3)$		
PROVER		**VERIFIER**
Let $i \in \{1,2,3\}, j \in \{1,2\}$		
Pick $u_i \xleftarrow{\$} \mathbb{F}_q^n$, $\pi_i \xleftarrow{\$} \mathrm{Mon}_n$		
$d_j = H_j u_1 + T_j u_{j+1}$		
Set $c_1 = \mathsf{com}(\pi_i, d_j)$		
and $c_2 = \mathtt{com}(\pi_i(u_i),$		
$\pi_1(e), \pi_{j+1}(r_j))$	$\xrightarrow{(c_1,c_2)}$	
	$\xleftarrow{\xi}$	Sample $\xi \xleftarrow{\$} \mathbb{F}_q^*$
Define $y_1 = \pi_1(u_1 + \xi e)$		
and $y_{j+1} = \pi_{j+1}(u_{j+1} + \xi r_j)$	$\xrightarrow{y_i}$	
	$\xleftarrow{b}$	Pick $b \xleftarrow{\$} \{1,2\}$
If $b = 1$, set $\mathtt{resp} = \{\pi_i\}$		
Else, $\mathtt{resp} = \{\pi_1(e), \pi_{j+1}(r_j)\}$	$\xrightarrow{\mathtt{resp}}$	
		If $b = 1$, set:
		$\delta_j = H_j \pi_1^{-1}(y_1) + T_j \pi_{j+1}^{-1}(y_{j+1})$ $-\xi s_j$
		Accept if:
		$c_1 \stackrel{?}{=} \mathsf{com}(\pi_i, \delta_j)$
		Else, set:
		$\varepsilon_1 = y_1 - \xi \pi_1(e)$
		$\varepsilon_{j+1} = y_{j+1} - \xi \pi_{j+1}(r_j)$
		Accept if:
		$c_2 \stackrel{?}{=} \mathsf{com}(\varepsilon_i, \pi_1(e), \pi_{j+1}(r_j))$
		$\mathrm{wt}(\pi_1(e)) = t_1$,
		$\mathrm{wt}(\pi_{j+1}(r_j)) = t_{j+1}$

Algorithm 1.2: A code-based protocol demonstrating that two commitments hide the same secret.

The Sum Scheme		
Public: $q, n, k \in \mathbb{N}$, $H_1, H_2, H_3, T_1, T_2, T_3 \in \mathbb{F}_q^{(n-k)\times n}$, $H_1e_1 + T_1r_1 = s_1$, $H_2e_2 + T_2r_2 = s_2$, $H_3(e_1 + e_2) + T_3\widetilde{r_3} = s_3$		
Private: $e_1 \in \mathcal{B}(t_1)$, $e_2 \in \mathcal{B}(t_2)$		
PROVER		**VERIFIER**
Let $i \in \{1, 2, 3\}$ and $j \in \{1, 2\}$		
Pick $u_j, v_i \xleftarrow{\$} \mathbb{F}_q^n$,		
$\pi_j, \rho_i \xleftarrow{\$} \mathrm{Mon}_n$		
$d_j = H_j u_j + T_j v_j$,		
$d_3 = H_3(u_1 + u_2)$		
$+T_3(v_1 + v_2 + v_3)$		
Set $c_1 = \mathsf{com}(\pi_j, \rho_i, d_i)$		
Let $r_3 = \widetilde{r_3} - r_1 - r_2$		
$c_2 = \mathtt{com}(\pi_j(u_j), \rho_i(v_i),$		
$\pi_j(e_j), \rho_i(r_i))$	$\xrightarrow{(c_1, c_2)}$	
	$\xleftarrow{\xi}$	Sample $\xi \xleftarrow{\$} \mathbb{F}_q^*$
Define $f_j = \pi_j(u_j + \xi e_j)$		
and $g_i = \rho_i(v_i + \xi r_i)$	$\xrightarrow{(f_j, g_i)}$	
	$\xleftarrow{b}$	Pick $b \xleftarrow{\$} \{1, 2\}$
If $b = 1$, set $\mathtt{resp} = \{\pi_j, \rho_i\}$		
Else, $\mathtt{resp} = \{\pi_j(e_j), \rho_i(r_i)\}$	$\xrightarrow{\mathtt{resp}}$	
		If $b = 1$, set:
		$\delta_j = H_j\pi_j^{-1}(f_j) + T_j\rho_j^{-1}(g_j) - \xi s_j$
		$\delta_3 = H_3(\pi_1^{-1}(f_1) + \pi_2^{-1}(f_2)) - \xi s_3$
		$+T_3(\rho_1^{-1}(g_1) + \rho_2^{-1}(g_2) + \rho_3^{-1}(g_3))$
		Accept if:
		$c_1 = \mathsf{com}(\pi_j, \rho_i, \delta_i)$
		Else, set: $\varepsilon_j = f_j - \xi\pi_j(e_j)$
		$\phi_i = g_i - \xi\rho_i(r_i)$
		Accept if:
		$c_2 = \mathsf{com}(\varepsilon_j, \phi_i, \pi_j(e_j), \rho_i(r_i))$
		$\mathrm{wt}(\pi_j(e_j)) = t_j$

Algorithm 1.3: A code-based protocol demonstrating that the secret in a commitment is the sum of the secrets in two other commitments.

Matrix-Matrix Product Scheme	
Public: $q, \ell, m, n, k \in \mathbb{N}$, $H_1, T_1 \in \mathbb{F}_q^{(\ell-k)\times\ell}$, $H_2, T_2 \in \mathbb{F}_q^{(m-k)\times m}$, $H_3, T_3 \in \mathbb{F}_q^{(\ell-k)\times\ell}$ $H_1M_1 + T_1R_1 = s_1$, $H_2M_2 + T_2R_2 = s_2$, $H_3M_3 + T_3\widetilde{R_3} = s_3$	
Private: $M_1 \in \mathbb{F}_q^{\ell\times m}$, $M_2 \in \mathbb{F}_q^{m\times n}$ such that $M_1M_2 = M_3 \in \mathbb{F}_q^{\ell\times n}$	
PROVER	**VERIFIER**
Let $i \in \{1,2,3\}, j \in \{1,2\}$	
Pick $u_1, v_1 \xleftarrow{\$} \mathbb{F}_q^{\ell\times m}$,	
$u_2, v_2 \xleftarrow{\$} \mathbb{F}_q^{m\times n}$, $v_3 \xleftarrow{\$} \mathbb{F}_q^{\ell\times n}$,	
$\pi_1 \xleftarrow{\$} \mathrm{Mon}_{\ell\times m}, \pi_2 \xleftarrow{\$} \mathrm{Mon}_{m\times n}$	
$\pi_3 \xleftarrow{\$} \mathrm{Mon}_{\ell\times n}$	
Set $R_3 = \widetilde{R_3} - R_1R_2$,	
$d_j = H_ju_j + T_jv_j$,	
$d_3 = H_3(u_1u_2) + T_3(v_1v_2)$	
$z = H_3(u_1M_2 + M_1u_2)$ $+T_3(v_1R_2 + R_1v_2 + v_3)$	
Set $c_1 = \mathsf{com}(\pi_i, d_i)$,	
$c_2 = \mathsf{com}(\pi_j(M_j), \pi_i(R_i),$ $\pi_j(u_j), \pi_i(v_i))$	$\xrightarrow{(c_1,c_2),z}$
	$\xleftarrow{\xi}$ Sample $\xi \xleftarrow{\$} \mathbb{F}_q^*$
Set $f_j = \pi_j(u_j + \xi M_j)$,	
and $g_i = \pi_i(v_i + \xi R_i)$	$\xrightarrow{(f_j,g_i)}$
	$\xleftarrow{b}$ Pick $b \xleftarrow{\$} \{1, 2\}$
If $b = 1$, set $\mathtt{resp} = \{\pi_i\}$	
Else, $\mathtt{resp} = \{\pi_j(M_j), \pi_i(R_i)\}$	$\xrightarrow{\mathtt{resp}}$
	If $b = 1$, set:
	$\delta_j = H_j\pi_j^{-1}(f_j) + T_j\pi_j^{-1}(g_j) - \xi s_j$
	$\delta_3 = H_3\left(\pi_1^{-1}(f_1)\pi_2^{-1}(f_2)\right)$ $+T_3\left(\pi_1^{-1}(g_1)\pi_2^{-1}(g_2) + \xi\pi_3^{-1}(g_3)\right)$ $-\xi z - \xi^2 s_3$
	Accept if:
	$c_1 = \mathsf{com}(\pi_i, \delta_i)$
	Else, set: $\varepsilon_j = f_j - \xi\pi_j(M_j)$
	$\phi_i = g_i - \xi\pi_i(R_i)$
	Accept if:
	$c_2 = \mathsf{com}(\pi_j(M_j), \pi_i(R_i), \varepsilon_j, \phi_i)$

Algorithm 1.4: A code-based protocol demonstrating knowledge of a matrix-matrix product.