Code-Based Zero-Knowledge from VOLE-in-the-Head and Their Applications: Simpler, Faster, and Smaller

Ying Ouyang, Deng Tang ✉, Yanhong Xu ✉

Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
✉ {dengtang,yanhong.xu}@sjtu.edu.cn

Abstract. Zero-Knowledge (ZK) protocols allow a prover to demonstrate the truth of a statement without disclosing additional information about the underlying witness. Code-based cryptography has a long history but did suffer from periods of slow development. Recently, a prominent line of research have been contributing to designing efficient code-based ZK from MPC-in-the-head (Ishai et al., STOC 2007) and VOLE-in-the head (VOLEitH) (Baum et al., Crypto 2023) paradigms, resulting in quite efficient standard signatures. However, none of them could be directly used to construct privacy-preserving cryptographic primitives. Therefore, Stern's protocols remain to be the major technical stepping stones for developing advanced code-based privacy-preserving systems.

This work proposes new code-based ZK protocols from VOLEitH paradigm for various relations and designs several code-based privacy-preserving systems that considerably advance the state-of-the-art in code-based cryptography. Our first contribution is a new ZK protocol for proving the correctness of a regular (non-linear) encoding process, which is utilized in many advanced privacy-preserving systems. Our second contribution are new ZK protocols for concrete code-based relations. In particular, we provide a ZK of accumulated values with optimal witness size for the accumulator (Nguyen et al., Asiacrypt 2019). Our protocols thus open the door for constructing more efficient privacy-preserving systems. Moreover, our ZK protocols have the advantage of being simpler, faster, and smaller compared to Stern-like protocols. To illustrate the effectiveness of our new ZK protocols, we develop ring signature (RS) scheme, group signature (GS) scheme, fully dynamic attribute-based signature scheme from our new ZK. The signature sizes of the resulting schemes are two to three orders of magnitude smaller than those based on Stern-like protocols in various parameter settings. Finally, our first ZK protocol yields a standard signature scheme, achieving "signature size + public key size" as small as 3.05 KB, which is slightly smaller than the state-of-the-art signature scheme (Cui et al., PKC 2024) based on the regular syndrome decoding problems.

Keywords: Zero-knowledge protocols, VOLE-in-the-head, code-based cryptography, privacy-preserving schemes, signature scheme

1 Introduction

A beautiful and fundamental notion introduced by Goldwasser, Micali and Rackoff [47], zero-knowledge (ZK) proof allows to prove a statement while not revealing anything about the witness. In the last three decades or so, ZK protocols are an important tool in designing numerous cryptographic constructions. Thanks to the Fiat-Shamir heuristic [45], ZK protocols have been the basis for developing standard signatures and privacy-enhancing authentication systems, such as group signature (GS) [32], ring signature (RS) [75], attribute-based signatures (ABS) [26], anonymous credential (AC) [31], and policy-based signature (PBS) [12].

Traditional cryptographic schemes based on number-theoretic assumptions are at the risk of being broken by quantum computers. This threat motivates the research for new ZK proof techniques based on post-quantum cryptographic problems. Among all possible alternatives, code-based cryptography is one of the promising choices. Dating back to 1996, Stern [78] introduced the first ZK for syndrome decoding (SD) problem and the framework has been utilized for constructing code-based signatures and privacy-preserving systems.

However, Stern protocols and its followup works [40,59,58] have soundness error 2/3, preventing it from being practical. Therefore, numerous works (e.g., [63,85,52,20]) have been devoting to construct more efficient protocols with smaller soundness error. A recent line of research in code-based cryptography by Gueron et al. [49], Bidoux et al. [20] and Feneuil et al. [42], have independently lowered the soundness error to $1/N$ for an arbitrary N by leveraging a technique inspired from the well-known MPC-in-the-head (MPCitH) paradigm [51,53]. Since then, MPCitH and its recent variant VOLE-in-the-head (VOLEitH) [8] have achieved a high success in designing efficient code-based ZK proofs and standard signature schemes [41,30,66,67,43,33,27,1,19].

To the best of our knowledge, none of these ZK protocols could be directly used to construct advanced privacy-preserving primitives from codes, where more sophisticated algebraic structures are required. In particular, a prominent line of research in designing code-based privacy-preserving schemes employed accumulators [14] to achieve logarithmic proof sizes [71,81,70,58]. The main technical difficulty of utilizing accumulators in designing these schemes is a supporting ZK argument of valid accumulated values. This is particularly challenging for the code-based accumulators [71] built from Merkle hash trees [68]. This is because the output of each hashing has to be encoded to a small-weight vector (with respect to its dimension) before going to the next step and we have to prove that the whole recursive process is done correctly. To overcome this difficulty, Nguyen et al. [71] designed a dedicated and involved (thus inefficient) ZK protocol to prove the correctness of the encoding process within Stern's framework. We note that a recent work by Ling et al. [58] has revisited the long-established Stern's protocol and put forward a new refined framework. Theoretically interesting and beautiful, the refined framework has not yielded noteworthy efficiency improvement. Nevertheless, Stern-like protocols remain to be the major technical

stepping stone for developing code-based advanced privacy-preserving systems, even they are still far from being practical.

In this work, we aim to contribute to the development of practically efficient ZK protocols for codes, particularly for proving the knowledge of accumulated values, which can be further used to construct various advanced privacy-preserving primitives. Since all the ZK protocols presented in this work belong to the VOLEitH paradigm [8]. Let us briefly review the development of it.

Vector oblivious linear evaluation (VOLE)-based ZK protocols were initiated by Boyle [22,24]. Due to low memory consumption and linear (in the circuit) proof sizes, VOLE-based ZK protocols have recently seen a lot of progress [23,82,36,11,84,35,83,9,10,57]. At a high level, VOLE-based proofs employ preprocessed random VOLE correlations to implement highly efficient proofs via a commit-and-prove paradigm. Recently, Baum et al. [8] developed a new method, named VOLEitH, resulting in simpler, faster, and smaller proofs than related approaches based on MPCitH [51]. They then instantiated their paradigm with two protocols, one for proving statements over large fields, and the other for proving statements over small fields. In addition, they briefly mentioned how to extend their protocols to proving low-degree polynomials satisfiability via the techniques from the QuickSilver [84] protocol.

Due to the attractive features of VOLEitH paradigm, Cui et al. [33] designed a new ZK for proving the knowledge of a solution to the regular syndrome decoding (RSD) problem using this paradigm, and turned their ZK into a standard signature scheme ReSolveD. Bidoux et al. [19] also applied VOLEitH to prove solutions to rank SD and MinRank problems, and obtained efficient code-based signatures.

1.1 Our Contributions

In this work, we provide a brand new ZK protocol for proving the correctness of a regular encoding process within the VOLEitH paradigm. Built upon this core technique, we then provide efficient ZK arguments of knowledge of valid opening, of an accumulated value, and of a plaintext. As main applications of our ZK protocols, we construct efficient RS, GS, fully dynamic ABS (FDABS) schemes whose signature sizes are two to three orders of magnitude smaller than those based on Stern-like ZK protocols. In addition, our new ZK protocols naturally yield a standard signature scheme, which is as efficient as the state-of-the-art code-based ones [33,19] with a flexible tradeoff on communication and computation.

Contribution to ZK protocol for proving the correctness of a regular encoding process. Recall that Nguyen et al. [71] employed the following regular encoding function to build their accumulator. Let c be a positive integer. Given a binary vector $\mathbf{x} = (x_1, \ldots, x_c)^\top$, let $t = \sum_{h=1}^{c} 2^{c-h} \cdot x^h$ be the integer whose binary representation is exactly $\mathbf{x}$. $\mathsf{RE} : \{0,1\}^c \to \{0,1\}^{2^c}$ maps $\mathbf{x}$ to $\mathbf{y} = \mathsf{RE}(\mathbf{x})$, where $\mathbf{y}$ is the unit vector of length 2^c with the sole 1 at the $(t+1)$-th position. To demonstrate that $\mathbf{y}$ is a correct regular encoding of $\mathbf{x}$, Nguyen et al. [71] employed a dedicated permutation technique that works well in Stern's framework. However,

this permutation technique prohibits the statement about the correct regular encoding process from being proved in other more efficient MPCitH or VOLEitH frameworks. To improve the efficiency, we instead take one step back and observe that it suffices to express the regular encoding process into polynomial constraints, a set of statements that can be proved within the VOLEitH framework. To this end, we reinterpret the regular encoding process as 2^c Boolean functions, which can be seen as a special case of polynomial constraints. In addition, these Boolean functions have degree c, which is usually a small constant ranging from 2 to 8. Therefore, our targeted statement can be *efficiently* proved within the VOLEitH paradigm.

We remark that this regular encoding function is employed in designing code-based commitment schemes and accumulators [71], which are essential building blocks for many privacy-preserving schemes, e.g., [71,70,58,81]. Therefore, our new ZK protocols open the door for constructing more efficient code-based privacy-preserving schemes such as RS, GS, ABS, AC, PBS.

Contribution to ZK protocols for concrete code-based relations. Building upon the core technique of proving the correct encoding process, we propose a variety of ZK for some concrete code-based relations that are essential in constructing privacy-enhancing authentication systems. In particular, we provide a new ZK protocol for proving the knowledge of committed values for the commitment scheme [71], a ZK protocol for proving the knowledge of accumulated values for the accumulator [71], and a ZK protocol for proving the knowledge of plaintexts for a variant of McEliece cryptosystem [65,72].

All our ZK protocols are within VOLEitH paradigm. In more detail, we reduce the above tasks to proving polynomial constraints through careful transformation. Importantly, proving the correctness of the regular encoding process $\mathbf{y} = \mathsf{RE}(\mathbf{x})$ essentially implies that $\mathbf{y}$ is a regular word. This observation is a key to huge efficiency improvement. Let us elaborate it more. When proving the knowledge of an accumulated value, $\mathcal{P}$ is to prove the knowledge of $(j_1, \ldots, j_\ell)^\top \in \{0,1\}^n$, $\mathbf{v}_1, \mathbf{w}_1, \ldots, \mathbf{v}_\ell, \mathbf{w}_\ell \in \mathbb{F}_2^n$ such that

$$\forall\ i \in \{\ell-1, \ldots, 1, 0\},\ \mathbf{v}_i = \begin{cases} \mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_{i+1}) + \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ \mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_{i+1}) + \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases} \quad (1)$$

Here $\mathsf{RE} : \{0,1\}^n \to \{0,1\}^{\frac{n}{c} \cdot 2^c}$. As mentioned earlier, $\mathcal{P}$ has to prove that the above recursive steps are done correctly. Particularly, $\mathcal{P}$ has to demonstrate that $\mathsf{RE}(\mathbf{v}_i)$ is indeed a regular encoding of $\mathbf{v}_i$. This can be proved via Stern's protocol [78,71] or our ZK. However, Nguyen et al. [71] had to introduce intermediate vectors $\mathbf{v}'_{i+1} = \mathsf{RE}(\mathbf{v}_{i+1})$ and $\mathbf{w}'_{i+1} = \mathsf{RE}(\mathbf{w}_{i+1})$, thus blowing up the witness size from $\ell + 2\ell n$ bits to $(2\ell) \cdot \frac{2n}{c} \cdot 2^c + 2(\ell-1) \cdot n$ bits. This blowup also results from the involved methods they developed to remove the dependence on $j_1, \ldots, j_\ell$ when computing $\mathbf{v}_0$. Our protocol, in contrast, can achieve the optimal witness size $\ell + 2\ell n$. As a result, our new ZK protocols have the advantage of being simpler, faster, and smaller compared to Stern-like protocols.

We emphasize that our ZK of accumulated values for accumulators built from Merkle trees is the first one that achieves optimal witness size $\ell + 2\ell n$. This is one of the main reasons that our new ZK protocols outperform Stern-like protocols.

Contribution to code-based advanced privacy-preserving primitives. To further illustrate the effectiveness of our new techniques, we develop several code-based privacy-preserving primitives from these new ZK protocols. In particular, we provide new ZK protocols for the ring signature scheme [71], the group signature scheme [71], the fully dynamic attribute-based signature scheme [58]. In addition, we examine the concrete signature sizes of our ZK protocols, and compare the results with the (refined) Stern-like ZK. Details are given in Table 3, Table 4, and Table 5. The comparisons exhibit the superiority of our new ZK protocols, which are two to three orders of magnitude smaller than Stern-like protocols in various parameter settings.

Table 1. Comparison of signature sizes for different privacy-preserving primitives from different ZK protocols based on various post-quantum assumptions for 128-bit security and ring/group size 2^{10}. Note that Katz et al. [53] evaluated the signature sizes at 256-bit security. We then include the sizes of our ZK at the same level below their results. For FDABS, the maximum number of attributes is $2^{\ell} = 2^{10}$, and the size of the circuit P is $K = 2^9$.

Schemes	code-based This work	code-based (Stern-type)		hash-based [53]	lattice-based [64]
RS	60 KB	61 MB [71]	61 KB [62]	388 KB (240 KB)	13 KB
GS	75 KB	63 MB [71]	121 KB* [62]	418 KB** (297 KB)	18 KB*
FDABS	62 KB	46 MB [58]	-	-	-

*: They only achieve CPA-anonymity.
** : It only achieves selfless anonymity.

Next, we give a brief comparison between the signature sizes of privacy-preserving schemes in this work and those of some previous post-quantum constructions. The results [1] are summarized in Table 1, in which we target for 128-bit security and ring/group size 2^{10}. The comparison shows that our ZK protocols perform much better, around three orders of magnitude smaller, than Stern-like ZK in [71,58]. Compared to the code-based ring signature and group signature schemes by Liu and Wang [62], our ring signature are as efficient as theirs while our group signature sizes are around 40% smaller. Also, our ring/group signature sizes are around 30% $\sim$ 40% smaller than the state-of-the-art hash-

[1] El Kaafarani and Katsumata [37] presented a lattice-based ABS scheme without giving concrete efficiency analysis. Since they employed Stern's protocols, their ABS scheme is supposed to be less efficient than [58].

based ones by Katz et al. [53]. Moreover, our performances are comparable to the state-of-the-art lattice-based constructions [64], in which the signature sizes are 13 KB (for RS) and 18 KB (for GS) in a similar parameter setting. We stress that they [64] employed the nice features of structured lattices and specialized techniques for optimal efficiency, and the three GS constructions [62,53,64] only achieve weaker form of anonymity. In contrast, our new ZK protocols are able to design CCA-anonymous GS and more advanced primitives such as FDABS.

We remark that the applications of our ZK protocols to RS, GS, FDABS are by no means exhaustive nor optimal. In fact, it is possible to employ our ZK protocols to design more efficient code-based privacy-preserving schemes such as group encryption [54], AC, PBS. Also, one can improve the performance of our ZK by choosing less conservative parameters for the underlying accumulator [71] or smaller parameters for the McEliece encryption scheme. We leave those extensions and optimizations to future work.

A new signature scheme based on RSD problem. Finally, we give a new signature scheme ReSolveD+ (improving upon ReSolveD [33]) based on the hardness of regular syndrome decoding (RSD) problem [4,5]. The construction follows from the crucial observation that $\mathbf{y}$ is a regular word if $\mathbf{y}$ is a correct regular encoding of some secret vector $\mathbf{x}$, and from the standard methodology of turning a public-coin ZK protocol into a signature scheme via the Fiat-Shamir heuristic [45]. We provide various parameter sets that offer tradeoffs between communication and computation targeting 128-bit security. The shortest version of our signature scheme achieves "signature size + public key size" 3.05 KB, which is slightly smaller than the state-of-the-art code-based signature schemes [33,19] based on RSD and a less studied rank SD problem. We give a detailed comparison of our signature scheme with previous works in Table 8.

1.2 Technical Overview

Let us now give a high-level discussion for our contributions.

ZK for regular encoding process. Recall that we need to represent the regular encoding process $\mathsf{RE} : \{0,1\}^c \to \{0,1\}^{2^c}$ into polynomial constraints. Towards this goal, we observe that RE can be seen as 2^c Boolean functions $f_{(0,\ldots,0)}(X_1,\ldots,X_c), f_{(0,\ldots,0,1)}(X_1,\ldots,X_c),\ldots,f_{(1,\ldots,1)}(X_1,\ldots,X_c)$. So the next question is whether we could explicitly give out these Boolean functions. The answer turns out to be affirmative. Through simple yet non-trivial calculation, the truth table of $f_{(j_1,\ldots,j_c)}(X_1,\ldots,X_c)$ is exactly the unit vector $\mathbf{e}_j$, where $(j_1,\ldots,j_c)^\top$ is the binary representation of $(j-1)$. Then by Lagrange interpolation, we can explicitly express $f_{(j_1,\ldots,j_c)}(X_1,\ldots,X_c) = \prod_{h=1}^{c}(1+j_h+X_h)$. At this point, we have successfully transformed the regular encoding process into degree-c relations and the witness size is exactly c bits.

ZK of a valid opening. We now describe how to construct a new and more efficient ZK argument of knowledge of a valid opening for the commitment scheme [71]. The prover is to prove the knowledge of $\mathbf{x} \in \mathbb{F}_2^L, \mathbf{r} \in \mathbb{F}_2^k$ such that

$$\mathbf{c} = \mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{x}) + \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{r}) \in \mathbb{F}_2^n, \tag{2}$$

with $\mathbf{B}_0 \in \mathbb{F}_2^{n \times \frac{L}{c} \cdot 2^c}$ and $\mathbf{B}_1 \in \mathbb{F}_2^{n \times \frac{k}{c} \cdot 2^c}$. As we are able to represent $\mathsf{RE}(\mathbf{x})$ and $\mathsf{RE}(\mathbf{r})$ as $(f_1(\mathbf{x}), \ldots, f_{\frac{L}{c} \cdot 2^c}(\mathbf{x}))^\top$ and $(f_{\frac{L}{c} \cdot 2^c + 1}(\mathbf{r}), \ldots, f_{\frac{L+k}{c} \cdot 2^c}(\mathbf{r}))^\top$, equation (2) can be easily transformed to n polynomials that are linear combinations of f_i for $i \in [1, \frac{L+k}{c} \cdot 2^c]$ subtracted by constants.

We remark that it is possible to employ the same linear sketching techniques as in [33] to show that $\mathsf{RE}(\mathbf{x})$ and $\mathsf{RE}(\mathbf{r})$ are regular words. However, this would incur witness size $\frac{L+k}{c} \cdot 2^c$ [2] instead of the optimal witness size $L + k$ achieved by using our techniques.

ZK of an accumulated value. Recall that the goal of $\mathcal{P}$ is to prove knowledge of $(j_1, \ldots, j_\ell)^\top \in \{0,1\}^n$, $\mathbf{v}_1, \mathbf{w}_1, \ldots, \mathbf{v}_\ell, \mathbf{w}_\ell \in \mathbb{F}_2^n$ such that (1) hold. This task can be divided into three parts: (i) demonstrate that $\mathsf{RE}(\mathbf{v}_1), \ldots, \mathsf{RE}(\mathbf{v}_\ell)$, $\mathsf{RE}(\mathbf{w}_1), \ldots, \mathsf{RE}(\mathbf{w}_\ell)$ are regular words; (ii) demonstrate that the branches of the tree is correctly chosen according to $j_1, \ldots, j_\ell$; (iii) demonstrate that $\mathsf{RE}(\mathbf{v}_i)$ is a correct regular encoding of $\mathbf{v}_i$ for $i \in [1, \ell]$. We have seen that (i) can be proved via our techniques or the linear sketching techniques [33]. However, the latter would deteriorate the efficiency. In particular, the linear sketching techniques would incur witness size $2\ell \cdot \frac{n}{c} \cdot 2^c$ while our techniques only incur witness size $2\ell \cdot n$. We thus stick to our techniques. Regarding (ii), let $\overline{j_{i+1}} = 1 - j_{i+1}$, then we observe that (1) is equivalent to

$$\mathbf{v}_i = \mathbf{B}_0 \cdot \big(\overline{j_{i+1}}\mathsf{RE}(\mathbf{v}_{i+1}) + j_{i+1}\mathsf{RE}(\mathbf{w}_{i+1})\big) + \mathbf{B}_1 \cdot \big(\overline{j_{i+1}}\mathsf{RE}(\mathbf{w}_{i+1}) + j_{i+1}\mathsf{RE}(\mathbf{v}_{i+1})\big),$$

where $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{F}_2^{n \times \frac{n}{c} \cdot 2^c}$. Thus, the terms $j_{i+1} \cdot \mathsf{RE}(\mathbf{v}_{i+1})$ and $j_{i+1} \cdot \mathsf{RE}(\mathbf{w}_{i+1})$ can be represented as $(f'_1(\cdot), \ldots, f'_{\frac{n}{c} \cdot 2^c}(\cdot))^\top$ and $(f'_{\frac{n}{c} \cdot 2^c + 1}(\cdot), \ldots, f'_{\frac{2n}{c} \cdot 2^c}(\cdot))^\top$, in which the degree of each polynomial f'_i increases to $(c+1)$ due to multiplication with the secret bit j_{i+1}. Similar to the above ZK of a valid opening, equations in (1) can now be transformed to ℓn polynomials. One then observes that (iii) is naturally solved if using our ZK for proving (i). We remark that the linear sketching techniques [33] cannot be used to prove (iii). In fact, they mainly focused on proving knowledge of a regular word and did not involving any regular encoding process, let alone prove correct regular encoding process.

We would also like to stress that the above simplicity for proving (ii) and (iii) only benefits from the fact that we represent the regular encoding process as polynomials and work in the VOLEitH paradigm. In fact, in a similar setting of proving the knowledge of an accumulated value, Libert et al. [56], Nguyen et al. [71], Yang et al. [85], Derler et al. [34], Boneh et al. [21] developed quite sophisticated and dedicated techniques to prove the honest computation of $\mathbf{v}_i$ and that the whole recursive process is computed honestly. As a result, their witness sizes are all much larger than the optimal size $\ell + 2\ell n$.

ZK of a plaintext. We now introduce a ZK argument of knowledge of a plaintext for a variant of McEliece encryption scheme[65,72], where the noise is a regular

[2] They introduced an optimization that can reduce the witness size to $\frac{L+k}{c} \cdot 2^c - n$, which is still larger than $L + k$ if one sticks to a statistically hiding commitment scheme.

word. Let $\mathbf{G} \in \mathbb{F}_2^{n_e \times k_e}$ be the public key and $\mathbf{c} \in \mathbb{F}_2^{n_e}$ be a ciphertext, k_1, k_2, k be positive integers such that $k_1 + k_2 = k_e$ and $\frac{k}{c} \cdot 2^c = n_e$. The prover is to prove the knowledge of $\mathbf{u} \in \mathbb{F}_2^{k_1}$, $\mathbf{m} \in \mathbb{F}_2^{k_2}$ as well as $\mathbf{e}' \in \mathbb{F}_2^k$ such that

$$\mathbf{c} = \mathbf{G} \cdot \begin{pmatrix} \mathbf{u} \\ \mathbf{m} \end{pmatrix} + \mathsf{RE}(\mathbf{e}'). \tag{3}$$

Similarly, we prove that $\mathbf{e} = \mathsf{RE}(\mathbf{e}')$ is a regular word by demonstrating that $\mathbf{e}$ is the correct regular encoding of some vector $\mathbf{e}'$. In addition, proving the knowledge of vectors $\mathbf{u}, \mathbf{m}$ is straightforward since we can view them as the identity function on $\{0,1\}^{k_e}$.

ZK for advanced privacy-preserving primitives. Being prepared with the above ZK protocols for various code-based relations, we are able to design ZK protocols for RS scheme [71], GS scheme [71], and FDABS scheme [58]. In particular, the ZK for RS scheme is an extension of the ZK for proving the regular encoding process and for proving an accumulated value. The ZK for GS scheme is then an extension of the ZK for RS by incorporating the ZK for proving the knowledge of a plaintext for the above variant of McEliece encryption. Finally, the ZK for FDABS scheme is an extension of ZK protocols for proving an accumulated value and for proving valid opening by incorporating a ZK for circuit satisfiability as well as a ZK for proving an odd-weight vector.

2 Preliminaries

Notations. Let λ be the security parameter. We use $x \xleftarrow{\$} S$ to denote the process of sampling x uniformly at random from a finite set S. Let $[a, b) := \{a, \ldots, b-1\}$ and we often write $[1, b]$ as $[b]$. Let $\oplus$ denote the bitwise exclusive-or. For a bit j, let $\bar{j} = j \oplus 1$. Throughout this paper, all vectors are column vectors and represented by bold lowercase letters (e.g., $\mathbf{x}$). Denote by x_i and $\mathbf{x}_{[i,j]}$ the i-component of vector $\mathbf{x}$ and the vector consisting of $x_i, x_{i+1}, \ldots, x_j$. Let $(\mathbf{x} \| \mathbf{y}) \in \mathbb{F}_2^{m+n}$ and $[\mathbf{A}|\mathbf{B}] \in \mathbb{F}_2^{n \times (m+k)}$ be the concatenation of vectors $\mathbf{x} \in \mathbb{F}_2^m$ and $\mathbf{y} \in \mathbb{F}_2^n$, and matrices $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ and $\mathbf{B} \in \mathbb{F}_2^{n \times k}$. For an integer $j \in [0, 2^\ell - 1]$, denote its binary representation by $\mathsf{bin}(j) \in \{0,1\}^\ell$.

2.1 Boolean Functions

Let $f \in \mathbb{F}_2[X_1, \ldots, X_c]$ be a c-variate Boolean function: $\mathbb{F}_2^c \to \mathbb{F}_2$. Then a representation of f is by its truth table, i.e.,

$$\mathrm{TT}(f) = [f(0,0,\ldots,0), f(0,\ldots,0,1), \ldots, f(0,1,\ldots,1), f(1,1,\ldots,1)]\,.$$

Clearly, the representation is unique. It is known (see e.g., [29,73]) that any Boolean function f in c variables can be expressed in terms of a multivariate polynomial in $\mathbb{F}_2[X_1, X_2, \ldots, X_c]/(X_1^2 + X_1, X_2^2 + X_2, \ldots, X_c^2 + X_c)$:

$$f(X_1, X_2, \ldots, X_c) = \sum_{\mathbf{u} \in \mathbb{F}_2^c} a_{\mathbf{u}} \left(\prod_{j=1}^{c} X_j^{u_j} \right) = \sum_{\mathbf{u} \in \mathbb{F}_2^c} a_{\mathbf{u}} \mathbf{X}^{\mathbf{u}},$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_c)$, $\mathbf{u} = (u_1, u_2, \ldots, u_c) \in \mathbb{F}_2^c$, $a_{\mathbf{u}} \in \mathbb{F}_2$ and the term $\mathbf{X}^{\mathbf{u}} = \prod_{i=1}^{c} X_i^{u_i}$ is called a monomial. This representation is called the algebraic normal form (ANF) of f. The algebraic degree of f, denoted by $\deg(f)$, is then defined as the maximum value of $\mathsf{wt}(\mathbf{u})$ with $a_{\mathbf{u}} \neq 0$.

2.2 Code-Based Collision Resistant Hash Functions

Augot, Finiasz and Sendrier (AFS) [4,5] introduced regular syndrome decoding (RSD) and 2-regular null syndrome decoding (2-RNSD) problems and proposed a family of code-based hash functions based on the hardness of the latter problem. Later, Nguyen et al. [71] developed the AFS hash function to obtain code-based computationally binding and statistically hiding commitment scheme. We first provide some related notions following [71] and then recall the AFS hash functions.

Let k, c be positive integers and c divides k. Define the following.

- $\mathsf{Regular}(k, c)$ is the set of all vectors $\mathbf{y} = (\mathbf{y}_1 \| \ldots \| \mathbf{y}_{k/c}) \in \mathbb{F}_2^{k/c \cdot 2^c}$ consisting of k/c blocks, each of which is a unit vector of length 2^c. We call $\mathbf{y}$ a *regular word* if $\mathbf{y} \in \mathsf{Regular}(k, c)$ for some k, c.
- $\mathsf{RE} : \mathbb{F}_2^k \to \mathbb{F}_2^{k/c \cdot 2^c}$ is a regular encoding function that encodes $\mathbf{x} = (\mathbf{x}_1 \| \ldots \| \mathbf{x}_{k/c}) \in \mathbb{F}_2^k$ to $\mathbf{y} = \mathsf{RE}(\mathbf{x}) = (\mathbf{y}_1 \| \ldots \| \mathbf{y}_{k/c})$. In particular, for $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,c})^\top$, let $t_j = \sum_{k=1}^{c} x_{j,k} \cdot 2^{c-k} \in [0, 2^c - 1]$ be the integer represented by $\mathbf{x}_j$. Then $\mathbf{y}_j$ is the unit vector of length 2^c that has the sole 1 at position $t_j + 1$. It is straightforward to see that $\mathbf{y} \in \mathsf{Regular}(k, c)$.
- 2-$\mathsf{Regular}(k, c)$ is the set of all vectors $\mathbf{x} \in \mathbb{F}_2^{k/c \cdot 2^c}$ such that exist regular words $\mathbf{v}, \mathbf{w} \in \mathsf{Regular}(k, c)$ satisfying $\mathbf{x} = \mathbf{v} \oplus \mathbf{w}$. Notice that $x \in$ 2-$\mathsf{Regular}(k, c)$ if and only if it can be written as the concatenation of k/c blocks of length 2^c, each of which has Hamming weight 0 or 2. We call $\mathbf{x}$ a 2-*regular word* if $\mathbf{x} \in$ 2-$\mathsf{Regular}(k, c)$ for some k, c.

RSD and 2-RNSD problems are variants of the famous SD problem, in which the goals are to find regular words and 2-regular words. As proved in [4,5], both problems are NP-complete. We recall them below.

Definition 1 (Regular Syndrome Decoding Problem). *Let n, k, c be three positive integers, $n > c$, and $k/c \cdot 2^c > k$. Define $m = k/c \cdot 2^c$. Given a uniform random matrix $\mathbf{B} \in \mathbb{F}_2^{n \times m}$, the regular syndrome decoding $\mathsf{RSD}_{n,k,c}$ problem asks to find a $\mathbf{x} \in \mathbb{F}_2^k$ such that $\mathbf{B} \cdot \mathsf{RE}(\mathbf{x}) = \mathbf{0} \bmod 2$.*

The hardness of RSD problem. A number of works have analyzed the hardness of RSD problem under different parameter regimes, e.g., [50,41,61]. In particular, some recent works [25,39,30] have utilized the regular noise structure into account, resulting in better algebraic attacks for RSD problem. As shown in [60, Table 8], such attacks work better than other pooled Gauss attack [38] or information set decoding (ISD) attack [74] for RSD with low-noise weight. Looking ahead, we work with parameters where the exact relations between RSD and SD problems remains unclear [30,39]. Therefore, we follow the approach presented in [30] to select parameters.

Definition 2 (2-Regular Null Syndrome Decoding Problem). *Let n, k, c be three positive integers, $n > c$, and $k/c \cdot 2^c > k$. Define $m = k/c \cdot 2^c$. Given a uniform random matrix $\mathbf{B} \in \mathbb{F}_2^{n \times m}$, the 2-regular null syndrome decoding* 2-$\mathsf{RNSD}_{n,k,c}$ *problem asks to find a $\mathbf{z} \in$ 2-$\mathsf{Regular}(k, c)$ such that $\mathbf{B} \cdot \mathbf{z} = \mathbf{0} \bmod 2$.*

Note that 2-RNSD problem is equivalent to finding two different $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^k$ such that $\mathbf{B} \cdot \mathsf{RE}(\mathbf{x}) = \mathbf{B} \cdot \mathsf{RE}(\mathbf{y})$.

The hardness of 2-RNSD problem. Augot et al. [4,5] applied ISD attack and generalized birthday attack (GBA) [80] to 2-RNSD problem, as well as giving lower bound on the cost of those two attacks. Later, Augot et al. [3] improved upon previous results and proposed several parameters for achieving different security levels. Follow-up works [18,17,16] proposed further improvements. As explicitly stated in [17,16], however, the parameters chosen in [3] are too conservative so that the further improved algorithms [18,17,16] do not violate the security claims made by Augot et al. [3]. To this end, we choose parameters for 2-RNSD problem according to [3].

The AFS hash functions. Let $n, k = \Omega(\lambda)$, $k > n$, and $c|k$. The AFS family of hash functions, specified by parameters n, k, c, is the set $\{h_{\mathbf{B}} : \mathbb{F}_2^k \to \mathbb{F}_2^k,\ \mathbf{B} \in \mathbb{F}_2^{n \times 2^c \cdot k/c}\}$ that maps $\mathbf{x}$ to $\mathbf{B} \cdot \mathsf{RE}(\mathbf{x}) \bmod 2$.

It is straightfoward to see that the above hash functions are collision-resistant based on the hardness of the 2-$\mathsf{RNSD}_{n,k,c}$ problem.

The modified AFS hash function. Nguyen et al. [71] recently modified the AFS hash function family [5] so that it takes 2 inputs (instead of just 1) and hence is suitable for building Merkle hash trees. The definition is given below.

Definition 3. *Let $m = 2 \cdot 2^c \cdot n/c$. The function family $\mathcal{H}$ mapping $\mathbb{F}_2^n \times \mathbb{F}_2^n$ to $\mathbb{F}_2^n$ is defined as $\mathcal{H} = \{h_{\mathbf{B}} \mid \mathbf{B} \in \mathbb{F}_2^{n \times m}\}$, where for $\mathbf{B} = [\mathbf{B}_0 | \mathbf{B}_1]$ with $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{F}_2^{n \times m/2}$, and for any $(\mathbf{u}_0, \mathbf{u}_1) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, we have:*

$$h_{\mathbf{B}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{u}_0) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{u}_1) \in \mathbb{F}_2^n.$$

The collision resistance of the hash function family relies on the hardness of the 2-$\mathsf{RNSD}_{n,2n,c}$ problem [71].

2.3 Code-Based Commitment Scheme

The above AFS hash functions can be used to build a commitment scheme. We now recall the statistically hiding and computationally binding commitment scheme proposed in [71].

$\mathsf{CSetup}(1^\lambda)$**:** Given the security parameter 1^λ, it chooses $n = \mathcal{O}(\lambda)$, $k \geq n + 2\lambda + \mathcal{O}(1)$, and specifies the message space $\mathcal{X} = \mathbb{F}_2^L$. It also chooses $c = \mathcal{O}(1)$ that divides both k and L. Let $m_0 = 2^c \cdot L/c$ and $m_1 = 2^c \cdot k/c$. Sample $\mathbf{C}_0 \xleftarrow{\$} \mathbb{F}_2^{n \times m_0}$ and $\mathbf{C}_1 \xleftarrow{\$} \mathbb{F}_2^{n \times m_1}$. Output public parameter $\mathsf{pp} = \{\lambda, n, k, L, c, m_0, m_1, \mathbf{C}_0, \mathbf{C}_1\}$.

$\mathsf{CCom}(\mathsf{pp}, \mathbf{x})$**:** To commit to a message $\mathbf{x} \in \mathbb{F}_2^L$, this algorithm samples a randomness $\mathbf{r} \xleftarrow{\$} \mathbb{F}_2^{n \times k}$, computes $\mathbf{c} = \mathbf{C}_0 \cdot \mathsf{RE}(\mathbf{x}) \oplus \mathbf{C}_1 \cdot \mathsf{RE}(\mathbf{r})$, and outputs commitment $\mathbf{c}$ as well as the opening $\mathbf{r}$.

$\mathsf{COpen}(\mathsf{pp}, \mathbf{c}, (\mathbf{x}, \mathbf{r}))$**:** Given the inputs, it outputs 1 if $\mathbf{c} = \mathbf{C}_0 \cdot \mathsf{RE}(\mathbf{x}) \oplus \mathbf{C}_1 \cdot \mathsf{RE}(\mathbf{r})$ and 0 otherwise.

Lemma 1 ([71]). *The above commitment scheme is correct. For any $\mathbf{x} \in \mathbb{F}_2^L$, the distribution of commitment $\mathbf{c}$ is statistically close to the uniform distribution over $\mathbb{F}_2^n$. In particular, the scheme satisfies the statistical hiding property. Moreover, if $2\text{-}\mathsf{RNSD}_{n,L+k,c}$ problem is hard, then the scheme is also computationally binding.*

2.4 Updatable Code-Based Merkle-tree Accumulator

We now recall the updatable code-based Merkle-tree accumulator [71,70].

$\mathsf{TSetup}(1^\lambda)$. This algorithm first chooses $n = \mathcal{O}(\lambda)$, $c = \mathcal{O}(1)$ so that c divides n. Set $m = 2 \cdot 2^c \cdot n/c$. It then samples $\mathbf{B} \xleftarrow{\$} \mathbb{F}_2^{n \times m}$, and outputs the public parameter $\mathsf{pp} = \{\lambda, n, c, m, \mathbf{B}\}$.

$\mathsf{TAcc}(R = \{\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}\} \subseteq (\mathbb{F}_2^n)^N)$. Assume $N = 2^\ell$ without loss of generality. Re-write $\mathbf{d}_j$ as $\mathbf{u}_{\ell,j}$ and call $\mathbf{d}_j$ the leaf value of the leaf node $\mathsf{bin}(j)$ for $j \in [0, N-1]$. Build a binary tree upon N leaves $\mathbf{u}_{\ell,0}, \ldots, \mathbf{u}_{\ell,2^\ell-1}$ in the following way. For $k \in \{\ell-1, \ell-2, \ldots, 1, 0\}$ and $i \in [0, 2^k - 1]$, compute $\mathbf{u}_{k,i} = h_{\mathbf{B}}(\mathbf{u}_{k+1,2i}, \mathbf{u}_{k+1,2i+1})$. Output the accumulated value $\mathbf{u} = \mathbf{u}_{0,0}$.

$\mathsf{TWitGen}(R, \mathbf{d})$. If $\mathbf{d} \notin R$, the algorithm outputs $\perp$. Otherwise, it outputs the witness w for $\mathbf{d}$ as follows.

1. Set $\mathbf{d} = \mathbf{d}_j$ for some $j \in [0, N-1]$. Re-write $\mathbf{d}_j = \mathbf{u}_{\ell,j}$. Let $\mathsf{bin}(j) = (j_1, \ldots, j_\ell)^\top \in \{0,1\}^\ell$ be the binary representation of j.
2. Consider the path from $\mathbf{u}_{\ell,j}$ to the root $\mathbf{u}$, the witness w then consists of $\mathsf{bin}(j)$ as well as all the sibling nodes of the path. Let $w = \big(\mathsf{bin}(j), (\mathbf{w}_\ell, \ldots, \mathbf{w}_1)\big) \in \mathbb{F}_2^\ell \times \big(\mathbb{F}_2^n\big)^\ell$.

$\mathsf{TVerify}\big(\mathbf{u}, \mathbf{d}, w\big)$**.** Let w be of the following form:

$$w = \big(\ (j_1, \ldots, j_\ell)^\top, (\mathbf{w}_\ell, \ldots, \mathbf{w}_1)\big).$$

This algorithm then computes $\mathbf{v}_\ell, \ldots, \mathbf{v}_0$. Let $\mathbf{v}_\ell = \mathbf{d}$ and

$$\forall i \in \{\ell-1, \ldots, 1, 0\} : \mathbf{v}_i = \begin{cases} h_{\mathbf{B}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), \text{ if } j_{i+1} = 0; \\ h_{\mathbf{B}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), \text{ if } j_{i+1} = 1. \end{cases} \tag{4}$$

Output 1 if $\mathbf{v}_0 = \mathbf{u}$ or 0 otherwise.

$\mathsf{TUpdate}(\mathsf{bin}(j), \mathbf{d}^*)$**:** Let $\mathbf{d}_j$ be the existing leaf value of the leaf node $\mathsf{bin}(j)$. It executes the algorithm $\mathsf{TWitGen}_{\mathbf{B}}(R, \mathbf{d}_j)$, obtaining $w = (\mathsf{bin}(j), (\mathbf{w}_\ell, \ldots, \mathbf{w}_1))$. It then sets $\mathbf{v}_\ell = \mathbf{d}^*$ and recursively computes $\mathbf{v}_{\ell-1}, \ldots, \mathbf{v}_0$ as in (4). Finally, for $i \in [0, \ell]$, it sets $\mathbf{u}_{i, \lfloor \frac{j}{2^{\ell-i}} \rfloor} = \mathbf{v}_i$.

Lemma 2 ([71]). *Assume that the $2\text{-}\mathsf{RNSD}_{n,2n,c}$ problem is hard, then the given accumulator scheme is correct and secure, i.e., it is infeasible to prove that a value $\mathbf{d}^*$ was accumulated in a value $\mathbf{u}$ if it was not (see, e.g., [56,28] for formal definition).*

2.5 Randomized McEliece Encryption Schemes

Now we recall a randomized variant of the McEliece [65] encryption scheme as suggested in [72].

$\mathsf{ME.Setup}(1^\lambda)$. Let $n_e = n_e(\lambda),\ \ k_e = k_e(\lambda),\ \ t_e = t_e(\lambda)$ be the parameters for a binary $[n_e, k_e, 2t_e + 1]$ Goppa code. Choose $k_1, k_2 \in \mathbb{Z}$ such that $k_e = k_1 + k_2$. Let $\mathbb{F}_2^{k_2}$ be the plaintext space.

$\mathsf{ME.KeyGen}(n_e, k_e, t_e)$. This algorithm outputs the encryption key and decryption key for the randomized McEliece encryption scheme. It works as follows:

1. Choose a generator matrix $\mathbf{G}' \in \mathbb{F}_2^{n_e \times k_e}$ of a random $[n_e, k_e, 2t_e+1]$ Goppa code. Let $\mathbf{S} \in \mathbb{F}_2^{k_e \times k_e}$ be a random invertible matrix and $\mathbf{P} \in \mathbb{F}_2^{n_e \times n_e}$ be a random permutation matrix, compute $\mathbf{G} = \mathbf{PG'S} \in \mathbb{F}_2^{n_e \times k_e}$.
2. Output encryption key $\mathsf{pk_{ME}} = \mathbf{G}$ and decryption key $\mathsf{sk_{ME}} = (\mathbf{S}, \mathbf{G}', \mathbf{P})$.

$\mathsf{ME.Enc}(\mathsf{pk_{ME}}, \mathbf{m})$. On input a message $\mathbf{m} \in \mathbb{F}_2^{k_2}$ and $\mathsf{pk_{ME}}$, sample random $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k_1}$ and $\mathbf{e} \in \mathbb{F}_2^{n_e}$ such that the Hamming weight of $\mathbf{e}$ is exactly t_e, and then output the ciphertext $\mathbf{c} = \mathbf{G} \cdot \begin{pmatrix} \mathbf{u} \\ \mathbf{m} \end{pmatrix} \oplus \mathbf{e} \in \mathbb{F}_2^{n_e}$.

$\mathsf{ME.Dec}(\mathsf{sk_{ME}}, \mathbf{c})$**.** On input the ciphertext $\mathbf{c}$ and decryption key $\mathsf{sk_{ME}}$, it works as follows:

1. Multiply $\mathbf{P}^{-1}$ to the left of the ciphertext $\mathbf{c}$, then apply an error-correcting algorithm. Obtain $\mathbf{m}'' = \mathsf{Decode}_{\mathbf{G}'}(\mathbf{c} \cdot \mathbf{P}^{-1})$, where Decode is an error-correcting algorithm with respect to $\mathbf{G}'$. Returns $\perp$ if Decode fails.
2. Multiply $\mathbf{S}^{-1}$ to the right of the ciphertext $\mathbf{m}''$, then $\mathbf{m}' = \mathbf{S}^{-1} \cdot \mathbf{m}''$, parse $\mathbf{m}' = \begin{pmatrix} \mathbf{u} \\ \mathbf{m} \end{pmatrix}$, where $\mathbf{u} \in \mathbb{F}_2^{k_1}$ and $\mathbf{m} \in \mathbb{F}_2^{k_2}$. Return $\mathbf{m}$.

The above scheme is CPA-secure if it is infeasible to distinguish the matrix $\mathbf{G}$ from random and the decisional learning parity with (exact) noise (DLPN) problem is computationally hard.

Definition 4 (Decisional Learning Parity with Exact Noise Problem). *Let N, k, t be three integers with $N > k$ and $N > t$. The decisional learning parity with (exact) noise $\mathsf{DLPN}_{N,k,t}$ problem asks to distinguish if a given pair $(\mathbf{G}, \mathbf{r}) \in \mathbb{F}_2^{N \times k} \times \mathbb{F}_2^N$ is uniformly random or obtained by choosing $\mathbf{G} \xleftarrow{\$} \mathbb{F}_2^{N \times k}$, $\mathbf{s} \xleftarrow{\$} \mathbb{F}_2^k$, $\mathbf{e} \xleftarrow{\$} \mathbb{F}_2^t$ with exact Hamming weight t and then outputting $(\mathbf{G}, \mathbf{G} \cdot \mathbf{s} \oplus \mathbf{e})$.*

A variant with regular noise. In this work, we consider a variant of McEliece encryption scheme such that the noise $\mathbf{e}$ is a regular word. More specifically, let k, c be two integers with $c|k$ such that $\frac{k}{c} \cdot 2^c = n_e$. Then the noise is computed as $\mathbf{e} = \mathsf{RE}(\mathbf{e}')$ with $\mathbf{e}' \xleftarrow{\$} \mathbb{F}_2^k$. The security of this variant will then rely on the hardness of decisional LPN problem with regular noise, which is dual to the decisional RSD problem.

2.6 VOLE-Based Zero-Knowledge Proofs

Vector oblivious linear evaluation (VOLE). VOLE is a two-party functionality $\mathcal{F}_{\mathsf{VOLE}}^{p,r}$ between a sender and receiver. It allows the sender to obtain $\mathsf{M} \in \mathbb{F}_{p^r}^l$ and $\mathbf{u} \in \mathbb{F}_p^l$ and the receiver to obtain $\mathsf{K} \in \mathbb{F}_{p^r}^l$ and $\Delta \in \mathbb{F}_{p^r}$ such that $\mathsf{K} = \mathsf{M} + \mathbf{u} \cdot \Delta$. These VOLE correlations can be used to authenticate $\mathbf{u}$. We denote such authenticated values by $[\![\mathbf{u}]\!]$, indicating that the sender obtains $\mathbf{u}$ and M while the receiver obtains Δ and K. It is not hard to see that the sender cannot alter $\mathbf{u}$ to a different $\mathbf{u}'$ without guessing Δ correctly. It is also easy to verify that VOLE correlations are additively homomorphic. In particular, given public coefficients $c_0, \ldots, c_l \in \mathbb{F}_{p^r}$, two parties can locally compute $[\![y]\!] = \sum_{i=1}^{l} c_i \cdot [\![u_i]\!] + c_0$, where the sender computes $y := \sum_{i=1}^{l} c_i \cdot u_i + c_0$ and $\mathsf{M}_y = \sum_{i=1}^{l} c_i \cdot \mathsf{M}_{u_i}$, and the receiver computes $\mathsf{K}_y := \sum_{i=1}^{l} c_i \cdot \mathsf{K}_{u_i} + c_0 \cdot \Delta$.

VOLE-Based ZK proofs. A VOLE-based ZK protocol for circuit satisfiability works in two phases. First, two parties call the functionality $\mathcal{F}_{\mathsf{VOLE}}^{p,r}$ to obtain random VOLE correlations. Using these correlations, the two parties obtain VOLE correlations for all wire values. This is done by letting $\mathcal{P}$ commit to all input wire values and output wire values of multiplication gates. Due to the homomorphic property of VOLE, they will also obtain VOLE correlations for the output wire values of addition gates. Next, they run subprotocols to check that all multiplications gates are computed honestly. One approach proposed by Ditter et al. [36] and later improved by Yang et al. [84] employs the fact that VOLE correlations are linear relationships, and works as follows.

For i-th multiplication gate, $\mathcal{P}$ has (M_1, w_1), (M_2, w_2), $(\mathsf{M}_3, w_3) \in \mathbb{F}_p \times \mathbb{F}_{p^r}$, and the verifier $\mathcal{V}$ possesses Δ, $\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3 \in \mathbb{F}_{p^r}$ such that

$$w_3 = w_1 \cdot w_2, \quad \text{and} \quad \mathsf{K}_i = \mathsf{M}_i + w_i \cdot \Delta \quad \text{for} \quad i \in \{1, 2, 3\}. \tag{5}$$

If the circuit is computed correctly, then

$$\begin{aligned} B_i &= \underbrace{\mathsf{K}_1 \cdot \mathsf{K}_2 - \mathsf{K}_3 \cdot \Delta}_{\text{known to } \mathcal{V}} \\ &= \underbrace{\mathsf{M}_1 \cdot \mathsf{M}_2}_{\text{known to } \mathcal{P}} + \underbrace{(\mathsf{M}_2 \cdot w_1 + \mathsf{M}_1 \cdot w_2 - \mathsf{M}_3)}_{\text{known to } \mathcal{P}} \cdot \Delta + \underbrace{(w_1 \cdot w_2 - w_3)}_{\text{0 if } \mathcal{P} \text{ is honest}} \cdot \Delta^2 \\ &= A_{i,0} + A_{i,1} \cdot \Delta. \end{aligned} \tag{6}$$

Therefore, checking the quadratic constraints of multiplication gates can be converted to checking the above linear equation (6). Moreover, we can use random linear combination to reduce checking t equations (corresponding to t multiplication gates) to checking a single equation. More specifically, the verifier $\mathcal{V}$ samples a uniform vector $\boldsymbol{\chi} \in \mathbb{F}_{p^r}^t$ and sends it to $\mathcal{P}$, who returns back $A_0 = \sum_{i=1}^{t} \chi_i \cdot A_{i,0} + A_0^*$ and $A_1 = \sum_{i=1}^{t} \chi_i \cdot A_{i,1} + A_1^*$. The verifier then check if $\sum_{i=1}^{t} \chi_i \cdot B_i + B^* = A_0 + A_1 \cdot \Delta$. Here $B^* = A_0^* + A_1^* \cdot \Delta$ is another random VOLE correlation.

Vector oblivious polynomial evaluation (VOPE). VOPE, first introduced by Yang et al. [84], is an extension of VOLE, in which the sender gets $A_0, \ldots, A_d \in$

$\mathbb{F}_{p^r}$ while the receiver gets $B \in \mathbb{F}_{p^r}$ and $\Delta \in \mathbb{F}_{p^r}$ such that $B = \sum_{i\in[0,d]} A_i \cdot \Delta^i$. Such VOPE correlations are particularly efficient for proving polynomial satisfiability. As shown in [84], it is possible to prove a set of degree-d polynomials on totally l distinct variables with communication cost of $l + d$ field elements, which is independent of the number of multiplications to compute all polynomials. Let $f_1, \ldots, f_t$ be a set of l-variate degree-d polynomials over $\mathbb{F}_{p^k}$. For simplicity, we represent each polynomial in a degree-separated format, i.e., $f_i(X_1, \ldots, X_l) = \sum_{h\in[0,d]} g_{i,h}(X_1, \ldots, X_l)$ such that all terms in $g_{i,h}$ have degree exactly h. The prover wants to prove that $f_i(w_1, \ldots, w_l) = 0$ for $i \in [1, t]$ with $\mathbf{w} = (w_1, \ldots, w_l)^\top \in \mathbb{F}_p^l$. Similar to the VOLE-based ZK, $\mathcal{P}$ first commits to the witness $\mathbf{w}$, and then checks that all polynomials are satisfied. The key observation is that one can obtain a degree-$(d-1)$ constraint generalized from (6). In more detail, suppose $\mathcal{P}$ and $\mathcal{V}$ obtain $[\![w_1]\!], \ldots, [\![w_l]\!]$ such that $\mathsf{K}_i = \mathsf{M}_i + w_i \cdot \Delta$. Then

$$\begin{aligned} B_i &= \underbrace{\sum_{h=0}^{d} g_{i,h}(\mathsf{K}_1, \ldots, \mathsf{K}_l) \cdot \Delta^{d-h}}_{\text{known to } \mathcal{V}} = \sum_{h=0}^{d} g_{i,h}(\mathsf{M}_1 + w_1 \cdot \Delta, \ldots, \mathsf{M}_l + w_l \cdot \Delta) \cdot \Delta^{d-h} \\ &= \underbrace{f(w_1, \ldots, w_n)}_{0 \text{ if } \mathcal{P} \text{ is honest}} \cdot \Delta^d + \underbrace{A_{i,0}}_{\text{known to } \mathcal{P}} + \underbrace{A_{i,1}}_{\text{known to } \mathcal{P}} \cdot \Delta + \ldots + \underbrace{A_{i,d-1}}_{\text{known to } \mathcal{P}} \cdot \Delta^{d-1}. \quad (7) \end{aligned}$$

Finally, utilizing the random linear combination technique and a degree-$(d-1)$ VOPE correlation, one reduces checking t equations to checking a single equation.

2.7 VOLE-in-the-Head

A main drawback of the above VOLE-based and VOPE-based ZK proof systems is that of being inherently designated-verifier (DV) since $\mathcal{V}$ has to know its part of VOLE/VOPE correlations so as to verify the proofs. We now briefly recall the VOLE-in-the-head (VOLEitH) technique presented by Baum et al. [8] that transforms the above DVZK proofs to public-coin protocols, which in turn can be made non-interactive via the Fiat-Shamir heuristic [45].

At a high level, Baum et al. [8] employed a delayed VOLE functionality $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}$ that allows $\mathcal{P}$ to first generate its values K_i, u_i independent of Δ, M_i and to generate Δ, M_i *after* all proof messages have been "committed". This delayed VOLE functionality can then be realized via all-but-one oblivious transfer, which is further realized by GGM-based vector commitments (VC). Since in this work we focus on utilizing VOLEitH-based proof systems, we refrain from providing all the details about how to realize this delayed functionality. In the $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}$-hybrid model, they then presented two instantiations. The first one allows one to prove statements over large fields, and the second one is more tailored for proving statements over small fields. In this work, we focus on their second instantiation.

Optimizations of VOLEitH. Though being a new paradigm, VOLEitH has received significant attention from the community [57,6,33,27,19] and several

works have improved upon VOLEitH. Baum et al. [6] introduced batch all-but-one VC, rejection sampling, and proof of work at the prover's side to reduce commitment opening sizes. Also, the improved GGM-based puncturable pseudorandom function proposed by Bui et al. [27] can be used as a drop-in replacement of the GGM-based VC. These optimizations are fundamental to improving the realization of the delayed functionality $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}$. While introducing significant improvement when designing standard signature schemes, these optimizations are relatively small for designing advanced primitives such as RS and GS. Nevertheless, in the corresponding sections, we will briefly mention how these optimizations improve the signature sizes of our constructions.

3 New Techniques for Proving Regular Encoding Process

In this section, we introduce new techniques for proving the correctness of a regular encoding process within the VOLEitH paradigm [8]. Our starting point is to explicitly express the non-linear regular encoding function RE as low-degree polynomial relations, which then can be proved efficiently using VOLEitH. To this end, we first present a protocol $\Pi_{\mathsf{dD\text{-}Rep}}^t$ for proving degree-d polynomial constraints in Section 3.1, which is a generalization of the protocol $\Pi_{\mathsf{2D\text{-}Rep}}^t$ [8] for proving degree-2 constraints. Then, we show how to express the non-linear regular encoding function RE as low-degree polynomial relations in Section 3.2.

3.1 VOLE-in-the-Head for Degree-d Constraints

Let us now describe our protocol $\Pi_{\mathsf{dD\text{-}Rep}}^t$, which is a generalization of the protocol $\Pi_{\mathsf{2D\text{-}Rep}}^t$ [8] by incorporating the techniques in QuickSilver [84] for proving degree-d polynomial satisfiability. The goal of $\mathcal{P}$ is to prove the knowledge of $\mathbf{w} \in \mathbb{F}_p^l$ such that $f_i(\mathbf{w}) = 0$ for $i \in [1, t]$, where $\{f_1, \ldots, f_t\}$ are l-variate degree-d polynomials. The protocol follows the commit-and-prove paradigm and works as follows.

In the commit phase, both parties invoke the delayed functionality $\mathcal{F}_{\mathsf{sVOLE}}$. The prover obtains $\mathbf{u} \in \mathbb{F}_p^{l+(d-1)r\tau}$ and $\mathbf{V}$, while the verifer $\mathcal{V}$ will obtain $\mathbf{Q}$ and $\boldsymbol{\Delta}$ satisfying $\mathbf{Q} = \mathbf{V} + \mathbf{u} \cdot \mathbf{G}_{\mathcal{C}}\mathsf{diag}(\boldsymbol{\Delta})$ (after receiving messages from $\mathcal{P}$ in the prove phase). Next, $\mathcal{P}$ commits to its witness $\mathbf{w}$ by sending $\mathbf{d} = \mathbf{w} - \mathbf{u}_{[1,l]}$ to $\mathcal{V}$.

In the challenge phase, $\mathcal{V}$ samples uniformly random coefficients $\chi_1, \ldots, \chi_t$ and sends them to $\mathcal{P}$. These coefficients will be used for the random linear combination performed by $\mathcal{P}$ in the following phase.

In the prove phase, $\mathcal{P}$ basically reduces the task of proving $f_i(\mathbf{w}) = 0$ for $i \in [1, t]$ into the task of proving the satisfiability of (7). In the process, both parties employ the remaining $(d-1)r\tau$ relations related to $\mathbf{u}_{[l+1,l+(d-1)r\tau]}$ to generate a single VOPE correlation so as to mask a random linear combination of the t equations (7). Note that $(d-1)r\tau$ relations are required here while $(2d-1)r\tau$ are needed in QuickSilver. As pointed out by [6], this is because that QuickSilver VOPE generation should be secure against malicious verifier while VOLEitH does not have to.

Details of the generalization are given in the protocol $\Pi_{\mathsf{dD\text{-}Rep}}^t$.

Protocol 1: $\Pi^t_{\mathsf{dD\text{-}Rep}}$

PARAMETERS: Code $\mathcal{C}_{\mathsf{Rep}} = [\tau, 1, \tau]_p$ with $\mathbf{G}_{\mathcal{C}} = (1, \ldots, 1) \in \mathbb{F}_p^{1\times\tau}$. $q = p^r$. Assume there is one-to-one correspondence between elements in $\mathbb{F}_q$ and $[1, q]$. Define $S_\Delta = \mathbb{F}_q^\tau$.
INPUTS: Polynomials $f_i = \sum_{h\in[0,d]} f_{i,h} \in \mathbb{F}_{p^k}[X_1, \ldots, X_l]_{\le d}, i \in [t]$ with $k|(r\tau)$. $\mathcal{P}$ holds a witness $\mathbf{w} = (w_1, \ldots, w_l)^\top \in \mathbb{F}_p^l$ such that $f_i(w_1, \ldots, w_l) = 0$ for all $i \in [t]$.

Round 1. $\mathcal{P}$ performs the following steps.

1. Call the functionality $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},l+(d-1)r\tau,\mathcal{L}}$ and receive $\mathbf{u} \in \mathbb{F}_p^{l+(d-1)r\tau}, \mathbf{V} \in \mathbb{F}_q^{(l+(d-1)r\tau)\times\tau}$. $\mathcal{V}$ receives done.
2. Compute $\mathbf{d} = \mathbf{w} - \mathbf{u}_{[1,l]} \in \mathbb{F}_p^l$ and send $\mathbf{d}$ to $\mathcal{V}$.
3. For $i \in [l+1, l+(d-1)r\tau]$, embed the i-th element $u_i \in \mathbb{F}_p$ of $\mathbf{u}$ to $u_i \in \mathbb{F}_{q^\tau}$. For $i \in [l + (d-1)r\tau]$, lift the i-th row $\mathbf{v}_i \in \mathbb{F}_q^\tau$ of $\mathbf{V}$ into $v_i \in \mathbb{F}_{q^\tau}$. For $i \in [l]$, also embed the i-th element w_i of witness $\mathbf{w}$ to $w_i \in \mathbb{F}_{q^\tau}$.

Round 2. $\mathcal{V}$ samples uniformly random $\chi_i \xleftarrow{\$} \mathbb{F}_{q^\tau}, i \in [t]$ and sends to $\mathcal{P}$.

Round 3. After receiving $\chi_1, \ldots, \chi_t$, $\mathcal{P}$ does the following.

1. For each $i \in [t]$, compute $A_{i,0}, A_{i,1}, \ldots, A_{i,d-1} \in \mathbb{F}_{q^\tau}$ such that
$$c_i(Y) = \sum_{h=0}^{d} \overline{f_{i,h}}(v_1+w_1Y, ..., v_l+w_lY)Y^{d-h} = \overline{f_i}(w_1, ..., w_l)\cdot Y^d + \sum_{j=0}^{d-1} A_{i,j}\cdot Y^j,$$
where $\overline{f_{i,h}} \in \mathbb{F}_{q^\tau}[X_1, \ldots, X_l]$ is the embedding of $f_{i,h} \in \mathbb{F}_{p^k}[X_1, \ldots, X_l]$.
2. **Generation of a VOPE correlation.**
   a) For $j \in [1, d)$, compute
   $$u_j^* = \sum_{i\in[r\tau]} u_{l+(j-1)r\tau+i} X^{i-1} \in \mathbb{F}_{q^\tau}, \quad v_j^* = \sum_{i\in[r\tau]} v_{l+(j-1)r\tau+i} X^{i-1} \in \mathbb{F}_{q^\tau}.$$
   where $\mathbb{F}_{q^\tau} \cong \mathbb{F}_p[X]/F(X)$ with $F(X) \in \mathbb{F}_p[X]$ being an irreducible polynomial of degree $r\tau$.
   b) Define $g_1(x) = v_1^* + u_1^* \cdot x$. For $i \in [1, d-2]$, compute $g_{i+1}(x) = g_i(x)(v_{i+1}^* + u_{i+1}^* \cdot x)$. Then $\mathcal{P}$ is able to compute the coefficients $A_0^*, \ldots, A_{d-1}^* \in F_{q^\tau}$ such that $g_{d-1}(x) = \sum_{j=0}^{d-1} A_j^* \cdot x^j$.
3. For $j \in [0, d)$, compute $\tilde{a}_j = \sum_{i\in[t]} \chi_i \cdot A_{i,j} + A_j^* \in \mathbb{F}_{q^\tau}$, and send $\tilde{a}_j$ to $\mathcal{V}$.

Verification. After receiving all responses, $\mathcal{V}$ runs the following checks.

1. Call $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},l+(d-1)r\tau,\mathcal{L}}$ on input (get) and obtain $\boldsymbol{\Delta} \in S_\Delta, \mathbf{Q} \in \mathbb{F}_q^{(l+(d-1)r\tau)\times\tau}$ such that $\mathbf{Q} = \mathbf{V} + \mathbf{u}\mathbf{G}_{\mathcal{C}}\mathsf{diag}(\boldsymbol{\Delta})$. Let $\mathbf{q}_i$ be the i-th row vector of $\mathbf{Q}$, for $i \in [l+1, l+(d-1)r\tau]$.
2. Compute $\mathbf{Q}^\star = \mathbf{Q}_{[1,l]} + \mathbf{d}\cdot\mathbf{G}_{\mathcal{C}}\cdot\mathsf{diag}(\boldsymbol{\Delta})$, which is supposed to be $\mathbf{V}_{[1,l]} + \mathbf{w}\cdot\mathbf{G}_{\mathcal{C}}\cdot\mathsf{diag}(\boldsymbol{\Delta})$. Let $\mathbf{q}_1^\star, \ldots, \mathbf{q}_l^\star \in \mathbb{F}_q^\tau$ be the rows of $\mathbf{Q}^\star$.
3. Lift $\boldsymbol{\Delta} \in \mathbb{F}_q^\tau$ into $\Delta \in \mathbb{F}_{q^\tau}$. Also, lift $\mathbf{q}_1^\star, \ldots, \mathbf{q}_l^\star, \mathbf{q}_{l+1}, \ldots, \mathbf{q}_{l+(d-1)r\tau} \in \mathbb{F}_q^\tau$ into $q_1^\star, \ldots, q_l^\star, q_{l+1}, \ldots, q_{l+(d-1)r\tau} \in \mathbb{F}_{q^\tau}$.

4. **Generation of a VOPE correlation.**
   a) For $j \in [1, d)$, compute $q^*_{l+j} = \sum_{i\in[r\tau]} q_{l+(j-1)r\tau+i} X^{i-1} \in \mathbb{F}_{q^\tau}$, which should satisfy $q^*_{l+j} = v^*_j + u^*_j \cdot \Delta$.
   b) Let $B^*_1 = q^*_{l+1}$. Then for $i \in [1, d-2]$, compute $B^*_{i+1} = B^*_i \cdot q^*_{l+i+1}$. Define $B^* = B^*_{d-1}$. Then one can verify that $B^* = \sum_{j=0}^{d-1} A^*_j \cdot \Delta^j$.
5. For each $i \in [t]$, compute

$$c_i(\Delta) = \sum_{h=0}^{d} \overline{f_{i,h}}(q^\star_1, \ldots, q^\star_l) \cdot \Delta^{d-h}.$$

6. Compute $\tilde{c} = \sum_{i\in[t]} \chi_i \cdot c_i(\Delta) + B^*$ and check if $\tilde{c} = \sum_{j=0}^{d-1} \tilde{a}_j \cdot \Delta^j$.

Theorem 1. *The protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ realizes the functionality $\mathcal{F}^t_{\mathsf{dD\text{-}ZK}}$ that proves degree-d polynomial satisfiability in the $\mathcal{F}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}_{\mathsf{sVOLE}}$-hybrid model. The security holds against a malicious prover or a semi-honest verifier and the soundness error is bounded by $1/p^{r\tau} + d|S_\Delta|^{-1}$.*

Correctness of the protocol follows directly by inspection of the protocol. Details of simulation are deferred to Appendix B.

Communication cost. In $\Pi^t_{\mathsf{dD\text{-}Rep}}$, in addition to the cost of the sVOLE steps, $\mathcal{P}$ sends the initial commitment $\mathbf{d} \in \mathbb{F}^l_p$ and $\{\tilde{a}_i \in \mathbb{F}_{q^\tau}\}_{i\in[0,d-1]}$. Therefore, the total cost is summarized as follows:

$$\mathsf{Cost}_{\Pi^t_{\mathsf{dD\text{-}Rep}}} = \mathsf{Cost}_{\mathsf{sVOLE}} + l \cdot \log_2 p + d \cdot r \cdot \tau \cdot \log_2 p. \tag{8}$$

Additionally, the verifier sends t values in $\mathbb{F}_{q^\tau}$ but this can be removed via the Fiat-Shamir transform in the non-interactive setting and does not affect the final proof size. When instantiating the delayed VOLE functionality $\mathcal{F}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}_{\mathsf{sVOLE}}$ with the aforementioned GGM-based VC (see [8, Sect. 3.1, Figure 3, Figure 4] for details), the cost of sVOLE steps is

$$\begin{aligned}\mathsf{Cost}_{\mathsf{sVOLE}} = 2\lambda &+ (l + (d-1) \cdot r \cdot \tau + h) \cdot (\tau - 1) \cdot \log_2 p \\ &+ (s + s \cdot \tau) \cdot \log_2 p + (2\lambda + r \cdot \lambda) \cdot \tau.\end{aligned} \tag{9}$$

The process of the instantiation employs an $\mathbb{F}^l_p$-hiding and ϵ-universal hash function $\mathsf{H} \in \mathbb{F}^{s\times(l+(d-1)r\tau+h)}_p$ (see Definition 5 for details). Looking ahead, when calculating concrete proof sizes in Section 5, we employ formulas (8) and (9).

Fiat-Shamir transform. As shown by Baum et al. [8], applying the Fiat-Shamir transformation to $\Pi^t_{\mathsf{dD\text{-}Rep}}$ (with the functionality $\mathcal{F}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}_{\mathsf{sVOLE}}$ instantiated) results in a non-interactive zero-knowledge argument of knowledge. Ganesh et al. [46] also showed that the resulting non-interactive argument is simulation-extractable in the programmable random oracle model. At a high level, simulation-extractability guarantees that extractability holds even when the adversary sees simulated proofs, and it implies simulation-soundness [77]. Note that simulation-soundness

is required for constructing CCA2-anonymous group signatures [13] that utilizes the Naor-Yung double encryption [69]. Therefore, replacing the Stern-like ZK protocol underlying the group signature scheme [71] with the above protocol will not degrade its security.

3.2 A New Technique for Proving the Regular Encoding Process

In this section, our target is to prove the regular encoding process within the VOLEitH paradigm. We then observe that it suffices to transform the regular encoding process into low-degree polynomial constraints. For simplicity, let us focus on the regular encoding function $\mathsf{RE} : \mathbb{F}_2^c \to \mathbb{F}_2^{2^c}$.

We also observe that RE can be seen as 2^c number of c-variate Boolean functions $f_1(\cdot), \ldots, f_{2^c}(\cdot)$. If we focus on the first output bit, then the truth table of the corresponding Boolean function $f_1(\cdot)$ is the unit vector $\mathbf{e}_1 \in \mathbb{F}_2^{2^c}$ with 1 in the first position. Through Lagrange interpolation, one can obtain $f_1(\cdot) \stackrel{\triangle}{=} f_{(0,\ldots,0)}(X_1, \ldots, X_c) = \prod_{i=1}^{c}(1+0+X_i)$. Interestingly, for the j-th output bit, the truth table of $f_j(\cdot)$ is the unit vector $\mathbf{e}_j \in \mathbb{F}_2^{2^c}$, and the Boolean function is $f_j(\cdot) \stackrel{\triangle}{=} f_{(j_1,\ldots,j_c)}(X_1, \ldots, X_c) = \prod_{i=1}^{c}(1 + j_i + X_i)$, where $(j_1, \ldots, j_c)^\top = \mathsf{bin}(j-1)$.

To this end, we have successfully represented the non-linear encoding process as degree-c relations. In particular, $\mathsf{RE}(x_1, \ldots, x_c) =$

$$\left(f_{(0,\ldots,0)}(x_1, \ldots, x_c), \ldots, f_{(j_1,\ldots,j_c)}(x_1, \ldots, x_c), \ldots, f_{(1,\ldots,1)}(x_1, \ldots, x_c) \right)^\top. \quad (10)$$

When applying RE to $\mathbf{x} \in \mathbb{F}_2^n$, we simply write $\mathsf{RE}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top$ for $c|n$, $m = \frac{n}{c} \cdot 2^c$, and $\deg(f_i) = c$ for $i \in [1, m]$, without explicitly describing the details of f_i. In fact, $f_j(X_1, \ldots, X_c) = f_{2^c+j}(X_1, \ldots, X_c) = \ldots = f_{(\frac{n}{c}-1)\cdot 2^c+j}(X_1, \ldots, X_c)$ for $j \in [1, 2^c]$. Note that $f_j(\mathbf{x})$ only selects c inputs and ignores other inputs.

Therefore, to show that $\mathbf{z} = (z_1, \ldots, z_m)^\top \in \mathsf{Regular}(n, c)$ is indeed a regular encoding of $\mathbf{x} = (x_1, \ldots, x_n)^\top \in \mathbb{F}_2^n$, it suffices to show that $z_j = f_j(\mathbf{x})$ for all $j \in [1, m]$. Since these m constraints are degree-c relations, and thus can be proved in zero-knowledge using the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$.

4 New Zero-Knowledge Protocols for Various Cryptographic Building Blocks

In this section, we provide new code-based zero-knowledge protocols that are essential for constructing privacy-enhancing primitives. This includes a ZK protocol for proving the knowledge of a committed value, a ZK protocol for proving the knowledge of a secret value that is accumulated honestly, and a ZK protocol for proving the knowledge of a plaintext for a variant of McEliece cryptosystem.

4.1 ZK of a Valid Opening

We first describe a ZK of a valid opening for the commitment scheme from Section 2.3. The goal of $\mathcal{P}$ is to convince the verifier that it possesses witnesses $\mathbf{x} \in \mathbb{F}_2^L$ and $\mathbf{r} \in \mathbb{F}_2^k$ such that $\mathbf{c} = \mathbf{C}_0 \cdot \mathsf{RE}(\mathbf{x}) \oplus \mathbf{C}_1 \cdot \mathsf{RE}(\mathbf{r})$. Denote $\mathbf{C}_0 = (c_{i,j})_{i\in[n],j\in[m_0]} \in \mathbb{F}_2^{n\times m_0}$ and $\mathbf{C}_1 = (c_{i,m_0+j})_{i\in[n],j\in[m_1]} \in \mathbb{F}_2^{n\times m_1}$ with $m_0 = \frac{L}{c}\cdot 2^c$ and $m_1 = \frac{k}{c}\cdot 2^c$. The protocol essentially relies on the techniques from Section 3.2 and works as follows.

Let $\widetilde{\mathbf{x}} = (\mathbf{x}\|\mathbf{r}) \in \mathbb{F}_2^{L+k}$, and $\mathbf{c} = (c_1, \ldots, c_n)^\top$. Then $\mathbf{c} = \mathbf{C}_0\cdot\mathsf{RE}(\mathbf{x})\oplus\mathbf{C}_1\cdot\mathsf{RE}(\mathbf{r})$ is equivalent to $\mathbf{c} = [\mathbf{C}_0|\mathbf{C}_1]\cdot\mathsf{RE}(\widetilde{\mathbf{x}})$. Denote $\mathsf{RE}(\widetilde{\mathbf{x}}) = (f_1(\widetilde{\mathbf{x}}), \ldots, f_{m_0+m_1}(\widetilde{\mathbf{x}}))^\top$. The prover then prepares n polynomials of degree c:

$$\phi_i(\cdot) = \sum_{j=1}^{m_0+m_1} c_{i,j} f_j(X_1, \ldots, X_{L+k}) - c_i, \quad \forall \ \ i \in [n],$$

and the witness $\widetilde{\mathbf{x}}$. At this point, $\mathcal{P}$ runs the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and applies the Fiat-Shamir transform.

4.2 ZK of an Accumulated Value

Next, we describe a ZK of an accumulated value for the accumulator recalled in Section 2.4. The prover aims to prove knowledge of a hash chain from a secret leaf node to the root.

Specifically, the public inputs are $\mathbf{B} = [\mathbf{B}_0|\mathbf{B}_1] \in \mathbb{F}_2^{n\times m}$ and the root $\mathbf{u} \in \mathbb{F}_2^n$. The secret inputs consist of $(j_1, \ldots, j_\ell)^\top \in \{0,1\}^\ell$, $\mathbf{v}_1, \ldots, \mathbf{v}_\ell \in \mathbb{F}_2^n$, and $\mathbf{w}_1, \ldots, \mathbf{w}_\ell \in \mathbb{F}_2^n$ such that

$$\overline{j_1}\cdot(\mathbf{B}_0\cdot\mathsf{RE}(\mathbf{v}_1)\oplus\mathbf{B}_1\cdot\mathsf{RE}(\mathbf{w}_1)) + j_1\cdot(\mathbf{B}_0\cdot\mathsf{RE}(\mathbf{w}_1)\oplus\mathbf{B}_1\cdot\mathsf{RE}(\mathbf{v}_1)) = \mathbf{u}, \quad (11)$$

and for all $\theta \in [2, \ell]$:

$$\overline{j_\theta}\cdot(\mathbf{B}_0\cdot\mathsf{RE}(\mathbf{v}_\theta)\oplus\mathbf{B}_1\cdot\mathsf{RE}(\mathbf{w}_\theta)) + j_\theta\cdot(\mathbf{B}_0\cdot\mathsf{RE}(\mathbf{w}_\theta)\oplus\mathbf{B}_1\cdot\mathsf{RE}(\mathbf{v}_\theta)) = \mathbf{v}_{\theta-1}. \quad (12)$$

Denote $\mathbf{B} = (b_{i,j})_{i\in[n],j\in[m]}$, $\mathbf{x}_1 = (\mathbf{v}_1\|\mathbf{w}_1)$ and $\mathbf{y}_1 = (\mathbf{w}_1\|\mathbf{v}_1)$, and $\mathbf{u} = (u_1, \ldots, u_n)^\top \in \mathbb{F}_2^n$. We have $\mathsf{RE}(\mathbf{x}_1) = (f_1(\mathbf{x}_1), \ldots, f_m(\mathbf{x}_1))^\top$ and $\mathsf{RE}(\mathbf{y}_1) = (f_{m+1}(\mathbf{y}_1), \ldots, f_{2m}(\mathbf{y}_1))^\top$ for some polynomials $f_1, \ldots, f_{2m}$ of degree c. Then equation (11) is equivalent to the following n degree-$(c+1)$ constraints:

$$\phi_i(\cdot) = \overline{j_1}\cdot\Big(\sum_{h=1}^m b_{i,h} f_h(\mathbf{x}_1)\Big) + j_1\cdot\Big(\sum_{h=1}^m b_{i,h} f_{m+h}(\mathbf{y}_1)\Big) - u_i, \quad \forall\, i \in [n]. \quad (13)$$

Here, the extra degree is due to multiplication with j_1.

Similarly, equation (12) is equivalent to the following n degree-$(c+1)$ constraints:

$$\phi_{(\theta-1)n+i}(\cdot) = \overline{j_\theta}\cdot\Big(\sum_{h=1}^m b_{i,h} f_{2(\theta-1)m+h}(\mathbf{x}_\theta)\Big)$$

$$+ j_\theta \cdot \Big(\sum_{h=1}^{m} b_{i,h} f_{2(\theta-1)m+m+h}(\mathbf{y}_\theta) \Big) - v_{\theta,i}, \ \forall\ i \in [n], \qquad (14)$$

where $\mathbf{x}_\theta = (\mathbf{v}_\theta \| \mathbf{w}_\theta)$, $\mathbf{y}_\theta = (\mathbf{w}_\theta \| \mathbf{v}_\theta)$, $\mathbf{v}_\theta = (v_{\theta,1}, \ldots, v_{\theta,n})^\top$, and $f_{2\theta m-2m+1}$, …, $f_{2\theta m}$ are $2m$ polynomials of degree c.

To this end, $\mathcal{P}$ are prepared with ℓn polynomials $\phi_1(\cdot), \ldots, \phi_{\ell n}(\cdot)$ of degree c+1, and possesses witness $\widetilde{\mathbf{x}} = (j_1 \| \ldots \| j_\ell \| \mathbf{v}_1 \| \mathbf{w}_1 \| \ldots \| \mathbf{v}_\ell \| \mathbf{w}_\ell) \in \mathbb{F}_2^{\ell+2\ell n}$. Therefore, it can run the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and utilize the Fiat-Shamir transform to make it non-interactive. One can see that the witness size is optimal.

4.3 ZK of Plaintext Knowledge

Now, we provide a ZK of plaintext knowledge for the variant of randomized McEliece encryption schemes with regular noise described in section 2.5. The prover needs to prove knowledge of a plaintext for a given ciphertext.

Specifically, the public inputs are $\mathbf{G} \in \mathbb{F}_2^{n_e \times k_e}$ and a ciphertext $\mathbf{c} \in \mathbb{F}_2^{n_e}$, and the secret inputs consist of $\mathbf{u} \in \mathbb{F}_2^{k_1}$, $\mathbf{m} \in \mathbb{F}_2^{k_2}$ as well as $\mathbf{e}' \in \mathbb{F}_2^{k}$ with $\frac{k}{c} \cdot 2^c = n_e$ such that

$$\mathbf{c} = \mathbf{G} \cdot \begin{pmatrix} \mathbf{u} \\ \mathbf{m} \end{pmatrix} \oplus \mathsf{RE}(\mathbf{e}'). \qquad (15)$$

Let $\mathbf{u} = (u_1, \ldots, u_{k_1})^\top$, $\mathbf{m} = (m_{k_1+1}, \ldots, m_{k_1+k_2})^\top$, $\mathbf{G} = (g_{i,j})_{i\in[n_e], j\in[k_e]}$, and $\mathbf{c} = (c_1, \ldots, c_{n_e})^\top$. According to the technique in Section 3.2, we will have $\mathsf{RE}(\mathbf{e}') = (f_1(\mathbf{e}'), \ldots, f_{n_e}(\mathbf{e}'))^\top$ for some polynomials $f_1, \ldots, f_{n_e}$ of degree c. Then equation (15) is equivalent to the following n_e degree-c constraints:

$$\phi_i(\cdot) = \sum_{j=1}^{k_1} g_{i,j} \cdot u_j + \sum_{j=k_1+1}^{k_1+k_2} g_{i,j} \cdot m_j + f_i(\mathbf{e}') - c_i, \quad \forall \quad i \in [n_e]. \qquad (16)$$

To this end, $\mathcal{P}$ prepares n_e public polynomials $\phi_1(\cdot), \ldots, \phi_{n_e}(\cdot)$ of degree c, and the witness $\widetilde{\mathbf{x}} = (\mathbf{u} \| \mathbf{m} \| \mathbf{e}') \in \mathbb{F}_2^{k_e+k}$. As a result, $\mathcal{P}$ can run the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and make it non-interactive via the Fiat-Shamir transform.

5 ZK Protocols for Advanced Primitives

In this section, we provide new ZK protocols for code-based advanced privacy-preserving primitives, including ring signature scheme [71], a variant of group signature scheme [71], and fully dynamic attribute-based signature scheme [58]. Then we estimate the signature sizes of the above schemes by employing our ZK and Stern-like ZK [78]. The results show that the signature sizes utilizing our ZK protocols are two to three orders of magnitude smaller.

5.1 ZK for a Ring Signature Scheme

Being prepared with ZK protocols for proving the correctness of the regular encoding process from Section 3.2 and for proving the knowledge of an accumulated value from Section 4.2, we now provide a more efficient ZK protocol supporting the code-based ring signature scheme proposed by Nguyen et al. [71]. The construction is recalled in Appendix C.4.

This protocol is an extension of the one from Section 4.2, where $\mathcal{P}$ additionally convinces the verifier the following fact: He/She knows a secret key $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{F}_2^n$ such that

$$\mathbf{v}_\ell = \mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{x}_0) + \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{x}_1). \tag{17}$$

In fact, we have already seen how to transform the above equation (17) into n constraints of degree c in Section 4.1. More specifically, there exist some polynomials $f_{2\ell m+1}, \ldots, f_{2\ell m+m}$ of degree c such that

$$\mathsf{RE}(\mathbf{x}_0 \| \mathbf{x}_1) = (f_{2\ell m+1}(\mathbf{x}_0 \| \mathbf{x}_1), \ldots, f_{2\ell m+m}(\mathbf{x}_0 \| \mathbf{x}_1))^\top.$$

Therefore, equation (17) is equivalent to the following n degree-c relations:

$$\phi_{\ell n+i}(\cdot) = \sum_{h=1}^{m} b_{i,h} \cdot f_{2\ell m+h}(\mathbf{x}_0 \| \mathbf{x}_1) - v_{\ell,i}, \quad \forall \ \ i \in [n], \tag{18}$$

where $\mathbf{v}_\ell = (v_{\ell,1}, \ldots, v_{\ell,n})^\top$ and $[\mathbf{B}_0 | \mathbf{B}_1] = (b_{i,j})_{i\in[n], j\in[m]}$. At this point, $\mathcal{P}$ has witness $\widetilde{\mathbf{x}} = (\, j_1 \,\|\, \ldots \,\|\, j_\ell \,\|\, \mathbf{v}_1 \,\|\, \mathbf{w}_1 \,\|\, \ldots \,\|\, \mathbf{v}_\ell \,\|\, \mathbf{w}_\ell \,\|\, \mathbf{x}_0 \,\|\, \mathbf{x}_1 \,) \in \mathbb{F}_2^{\ell+2\ell n+2n}$ and the newly appeared n degree-c constraints $\phi_{\ell n+1}(\cdot), \ldots, \phi_{\ell n+n}(\cdot)$ in addition to the ℓn degree-$(c+1)$ constraints $\phi_1(\cdot), \ldots, \phi_{\ell n}(\cdot)$ from Section 4.2. Therefore, it suffices for $\mathcal{P}$ to run the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and then apply the Fiat-Shamir heuristic.

5.2 Parameters and Efficiency

We now estimate the concrete sizes of the above ring signature scheme using our ZK protocol and Stern-like protocol [71]. Note that the security of the scheme relies on the hardness of 2-$\mathsf{RNSD}_{n,2n,c}$ problem. Also, we need to make sure the underlying proof systems achieve small enough soundness errors. See Theorem 2. Details of the parameters are given in Table 2.

On the parameters of 2-RNSD problem. As discussed in Section 2.2, we choose parameters according to [3]. In particular, they chose $n = 1024, w = 128, m = 2^{21}$ and $n = 1984, w = 248, m = 31 \cdot 2^{16}$ for 128-bit and 256-bit security levels, respectively. However, we work in a setting where n, c uniquely determine w and m. Therefore, we adjust the parameters slightly by setting $c = 8$ and increasing n from 1024 to 1280 and from 1984 to 2560, respectively.

On the parameters of the VOLEitH proof system. We choose parameters according to the specification given in [8,7], for 128-bit security. Regarding the parameters for 256-bit security level, we double the repetition parameter τ.

Table 2. Parameters for the hash function $h_{\mathbf{B}}$, for the proof system from VOLEitH paradigm, and for the McEliece encryption cryptosystem that achieve 128-bit security and 256-bit security.

Parameters	Description	128-bit Security	256-bit Security
λ	Security level	128	256
n	Hash $h_{\mathbf{B}}$ output length	1280	2560
c	2-RNSD Parameter	8	8
$w = \frac{2n}{c}$	$h_{\mathbf{B}}$ input Hamming weight	320	640
$m = \frac{2n}{c} \cdot 2^c$	$h_{\mathbf{B}}$ input length	$5 \cdot 2^{14}$	$5 \cdot 2^{15}$
p	Base field $\mathbb{F}_p$	2	2
q	Extension field $\mathbb{F}_q$	2^8	2^8
τ	Repetition for VOLEitH	16	32
$s = \lambda + 16$	Universal hash parameter	144	272
$h = \lambda + 16$	Universal hash parameter	144	272
κ	Repetition for Stern	219	438
n_e	McEliece parameter	4096	8192
k_e	McEliece parameter	3328	6528
t_e	McEliece parameter	64	128

Repetition for Stern-like protocols. The underlying ZK protocol for the above ring signature schemes used by Nguyen et al. [71] is Stern-like protocol [78]. Originally, it was designed to prove knowledge of a vector with exact Hamming weight. Later, it was adapted to prove various lattice-based and code-based linear and quadratic relations, e.g. [59,55,71], giving rise to various applications such as ring signatures [56,71], group signatures [55,71], attribute-based signatures [58], group encryption [70] and so on. However, it has the main disadvantage of large soundness error 2/3. Therefore, to achieve 2^{-128} and 2^{-256} soundness errors, one needs to repeat the protocol for 219 and 438 times.

Given the above parameters, we then give a detailed comparison about signature sizes for the ring signature scheme that employs our ZK protocol presented in Section 5.1 and that employs Stern-like protocols. Theoretically, both signature sizes are logarithmic in the size of the ring. Concretely, the performance of our ZK protocol appears to be significantly better. In particular, Table 3 shows that for 128-bit and 256-bit security levels, the signature sizes of [71] are around $934\times \sim 1140\times$ larger than ours for different ring sizes.

There are several reasons for the huge differences. One reason is that one needs to repeat Stern-type protocol for 219 times and 438 times to achieve negligible soundness error, while we only need to repeat VOLEitH proofs 16 and 32 times. Another reason is that their witness size is $(2 \cdot \ell + 1) \cdot (\frac{2n}{c} \cdot 2^c) + 2 \cdot \ell \cdot n$ bits while our witness size is just $\ell + 2\ell n + 2n$ bits.

Optimizations from [6]. As mentioned in Section 2.7, Baum et al. [6] proposed several optimizations to improve the realization of $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}$. In particular, it brings the decommitment size, $\mathsf{Cost}_{\mathsf{decom}} = (2\lambda + r \cdot \lambda)\tau$ from (9), down

Table 3. Ring signature sizes by employing our ZK proof system and Stern-like ZK arguments.

Ring size	128-bit security		256-bit security	
	This paper **(KB)**	Stern-type [71] **(MB)**	This paper **(KB)**	Stern-type [71] **(MB)**
2^5	35.12	32.26	140.24	129.04
2^7	45.12	43.93	180.25	175.74
2^{10}	60.13	61.44	240.26	245.78
2^{15}	85.14	90.63	340.28	362.51
2^{20}	110.15	119.81	440.30	479.25
2^{30}	160.17	178.18	640.34	712.72

to $2\lambda \cdot \tau + \mathsf{T}_{\mathsf{open}} \cdot \lambda$, where $\mathsf{T}_{\mathsf{open}}$ is a threshold number considered in [6]. Let $\mathsf{T}_{\mathsf{open}} = 102$ and $\mathsf{T}_{\mathsf{open}} = 218$ for $\lambda = 128$ and $\lambda = 256$, then the decommitment sizes are reduced by around 416 bytes and 1212 bytes, respectively. Therefore, the figures of our constructions in Table 3 could be further improved. We, however, have to admit that these improvements are relatively small for privacy-preserving protocols, and will no longer consider them in GS and FDABS schemes.

5.3 ZK for a Group Signature Scheme

Next, we provide a more efficient ZK protocol supporting the code-based group signature scheme proposed by Nguyen et al. [71], with the modification that the McEliece scheme is replaced with one with regular noise. The construction is described in Appendix C.5 for completeness.

This protocol is extended from the one in Section 5.1, for which an encryption layer is added. Specifically, $\mathcal{P}$ additionally proves the following statement: He/She knows extra secret values $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{F}_2^{k_e-\ell}$, $\mathbf{e}'_1, \mathbf{e}'_2 \in \mathbb{F}_2^k$ with $\frac{k}{c} \cdot 2^c = n_e$ such that

$$\mathbf{c}_1 = \mathbf{G}_1 \cdot \begin{pmatrix} \mathbf{r}_1 \\ \mathsf{bin}(j) \end{pmatrix} \oplus \mathsf{RE}(\mathbf{e}'_1), \quad \text{and} \quad \mathbf{c}_2 = \mathbf{G}_2 \cdot \begin{pmatrix} \mathbf{r}_2 \\ \mathsf{bin}(j) \end{pmatrix} \oplus \mathsf{RE}(\mathbf{e}'_2), \tag{19}$$

where $\mathbf{G}_1, \mathbf{G}_2 \in \mathbb{F}_2^{n_e \times k_e}$ and $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{F}_2^{n_e}$.

We have described in Section 4.3 on transforming the above equations (19) into polynomial constraints. For $\theta \in \{1, 2\}$, let

$$\mathbf{c}_\theta = (c_{\theta,i})_{i\in[n_e]}, \quad \mathbf{G}_\theta = (g^{(\theta)}_{i,j})_{i\in[n_e],j\in[k_e]}, \quad \mathbf{r}_\theta = (r_{\theta,1}, \ldots, r_{\theta,k_e-\ell})^\top,$$
$$\mathsf{RE}(\mathbf{e}'_\theta) = (f_{2\ell m+m+n_e(\theta-1)+1}(\mathbf{e}'_\theta), \ldots, f_{2\ell m+m+n_e(\theta-1)+n_e}(\mathbf{e}'_\theta))^\top,$$

for some polynomials $f_{2\ell m+m+1}, \ldots, f_{2\ell m+m+2n_e}$ of degree c. Therefore, equation (19) is equivalent to the following $2n_e$ degree-c relations:

$$\phi_{\ell n+n+i}(\cdot) = \sum_{h=1}^{k_e-\ell} g_{i,h}^{(1)} \cdot r_{1,j} + \sum_{h=1}^{\ell} g_{i,k_e-\ell+h}^{(1)} \cdot j_h + f_{2\ell m+m+i}(\mathbf{e}_1') - c_{1,i}, \quad \forall \ \ i \in [n_e],$$

$$\phi_{\ell n+n+n_e+i}(\cdot) = \sum_{h=1}^{k_e-\ell} g_{i,h}^{(2)} \cdot r_{2,j} + \sum_{h=1}^{\ell} g_{i,k_e-\ell+h}^{(2)} \cdot j_h + f_{2\ell m+m+n_e+i}(\mathbf{e}_2') - c_{2,i}, \quad \forall \ \ i \in [n_e].$$

Now, $\mathcal{P}$ has witness $\widetilde{\mathbf{x}} \in \mathbb{F}_2^{\ell+2\ell n+2n+2(k_e-\ell)+2k}$ of the following form

$$\widetilde{\mathbf{x}} = (\, j_1 \,\|\, \ldots \,\|\, j_\ell \,\|\, \mathbf{v}_1 \,\|\, \mathbf{w}_1 \,\|\, \ldots \,\|\, \mathbf{v}_\ell \,\|\, \mathbf{w}_\ell \,\|\, \mathbf{x}_0 \,\|\, \mathbf{x}_1 \,\|\, \mathbf{r}_1 \,\|\, \mathbf{r}_2 \,\|\, \mathbf{e}_1' \,\|\, \mathbf{e}_2' \,) \tag{20}$$

and the newly appeared $2n_e$ degree-c constraints $\phi_{\ell n+n+1}(\cdot), \ldots, \phi_{\ell n+n+2n_e}(\cdot)$ in addition to the $\ell n + n$ constraints $\phi_1(\cdot), \ldots, \phi_{\ell n+n}(\cdot)$ from Section 5.1. Now $\mathcal{P}$ can proceed as before by running the protocol $\Pi_{\mathsf{dD\text{-}Rep}}^t$ and then applying the Fiat-Shamir heuristic.

5.4 Parameters and Efficiency

We now estimate the concrete sizes of group signature scheme using our ZK protocol and Stern-like protocol [71]. The security of the scheme relies on the hardness of 2-$\mathsf{RNSD}_{n,2n,c}$ problem, on the CPA-security of the McEliece encryption scheme, as well as the security of the supporting ZK protocols. See Theorem 3. We use the same parameters proposed in Table 2.

On the parameters of McEliece encryption scheme. We choose parameters for McEliece cryptosystem following the document [2] with minor adaptations. Since we modify the noise vector to be a regular vector, the CPA-security of the McEliece encryption scheme now depends on the decisional RSD problem as discussed in Section 2.5. However, this is not an issue as shown in [61, Table 1, Table 2], the usage of regular noise for LPN and SD problems does not reduce the bit security significantly. In fact, one can always choose slightly larger parameters to obtain targeted security levels. In our setting, we then slightly increase the Goppa code length n_e (and hence the dimension k_e) so that it is the form of $\frac{k}{c} \cdot 2^c$ with $\frac{k}{c} = t_e$.

With the above parameters, we then estimate signature sizes using our ZK and Stern-like ZK for various group sizes. Details are in Table 4. The results also show the superiority of our ZK protocols. In particular, for 128-bit and 256-bit security, the signature sizes of [71] are around $683\times \sim 1053\times$ larger than ours for different group sizes.

5.5 ZK for a Fully Dynamic Attribute-Based Signature Scheme

Quite recently, Nguyen et al. [58] proposed a fully dynamic attribute-based signature (FDABS) scheme from codes. The scheme employs a refined Stern-like

Table 4. Group signature sizes by employing our ZK proof system and Stern-like ZK arguments.

Group size	128-bit security		256-bit security	
	This paper **(KB)**	Stern-type [71] **(MB)**	This paper **(KB)**	Stern-type [71] **(MB)**
2^5	49.60	33.27	197.19	133.02
2^7	59.60	44.94	237.19	179.72
2^{10}	74.59	62.45	297.18	249.76
2^{15}	99.58	91.63	397.16	366.50
2^{20}	124.57	120.82	497.14	483.23
2^{30}	174.55	179.18	697.10	716.70

protocol and is proven secure in the quantum oracle model (QROM) using the variant of Unruh transform [79] presented in [44]. To the best of our knowledge, we are unaware of existing works on making VOLEitH protocol secure in QROM. A related work by Aguilar-Melchor et al. [67] presented a security proof for Hypercube-SDitH [66] in the QROM. It remains open if one can apply their techniques to the VOLEitH paradigm. Therefore, we provide a new ZK for their FDABS scheme that is only secure in the ROM and then compare efficiency with their degraded variant. The description of the scheme is recalled in Appendix C.6.

We now provide our new ZK protocol. It is an extension of the one from Section 4.2, where $\mathcal{P}$ additionally convinces the verifier the following facts: He/she knows an attribute $\mathbf{x} \in \{0,1\}^L$ together with a randomness $\mathbf{r} \in \{0,1\}^k$ such that

$$\mathbf{v}_\ell = \mathbf{C}_0 \cdot \mathsf{RE}(\mathbf{x}) \oplus \mathbf{C}_1 \cdot \mathsf{RE}(\mathbf{r}); \tag{21}$$

$$\mathsf{wt}(\mathbf{v}_\ell) = 1 \bmod 2; \tag{22}$$

$$P(\mathbf{x}) = 1, \tag{23}$$

where P is an arbitrary binary circuit with L bit inputs and K multiplication gates.

We have already shown how to transform the equation (21) into polynomial constraints. Recall that $\mathbf{C}_0 = (c_{i,j})_{i\in[n],j\in[m_0]}$, $\mathbf{C}_1 = (c_{i,m_0+j})_{i\in[n],j\in[m_1]}$ with $m_0 = \frac{L}{c} \cdot 2^c$ and $m_1 = \frac{k}{c} \cdot 2^c$. Let $\mathbf{v}_\ell = (v_{\ell,1}, \ldots, v_{\ell,n})^\top$. Then equation (21) is equivalent to the following n degree-c relations

$$\phi_{\ell n+i}(\cdot) \;=\; \sum_{h=1}^{m_0+m_1} c_{i,h} f_{2\ell m+h}(\mathbf{x}\|\mathbf{r}) - v_{\ell,i}, \quad \forall i \in [n].$$

Regarding (22), it asks to prove that the Hamming weight of $\mathbf{v}_\ell$ is odd. We then observe that this is equivalent to proving

$$v_{\ell,1} + v_{\ell,2} + \cdots + v_{\ell,n} = 1.$$

Define $\phi_{\ell n+n+1}(X_1, \ldots, X_n) = X_1 + \cdots + X_n - 1 \in \mathbb{F}_2[X_1, \ldots, X_n]$. Then equation (22) is further equivalent to the following linear polynomial

$$\phi_{\ell n+n+1}(\cdot) = v_{\ell,1} + v_{\ell,2} + \ldots + v_{\ell,n} - 1. \tag{24}$$

In terms of (23), as observed by Ling et al. [58], it is equivalent to the following K quadratic equations:

$$\begin{cases} \phi_{\ell n+n+1+1}(\cdot) = x_{\alpha(1)} \cdot x_{\beta(1)} \oplus x_{L+1} - 1, \\ \cdots \\ \phi_{\ell n+n+1+K-1}(\cdot) = x_{\alpha(K-1)} \cdot x_{\beta(K-1)} \oplus x_{L+K-1} - 1, \\ \phi_{\ell n+n+1+K}(\cdot) = x_{\alpha(K)} \cdot x_{\beta(K)} \oplus x_{L+K} - 1, \end{cases} \tag{25}$$

where $x_{L+1}, \ldots, x_{L+K}$ are the output wire values of multiplication gates and $\alpha, \beta : \{1, \ldots, K\} \to \{1, \ldots, L+K-1\}$ are two functions specifying the topology of the circuit P.

Now $\mathcal{P}$ has witness $\widetilde{\mathbf{x}} \in \mathbb{F}_2^{\ell+2\ell n+L+k+K}$ of the following form

$$\widetilde{\mathbf{x}} = (j_1 \,\|\, \ldots \|j_\ell \,\|\, \mathbf{v}_1 \,\|\, \mathbf{w}_1 \,\|\, \ldots \|\, \mathbf{v}_\ell \,\|\, \mathbf{w}_\ell \,\|\, \mathbf{x} \,\|\, \mathbf{r} \,\|\, x_{L+1} \,\|\, \ldots \|\, x_{L+K}\,), \tag{26}$$

and the newly appeared $n + 1 + K$ constraints $\phi_{\ell n+1}(\cdot), \ldots, \phi_{\ell n+n+1+K}(\cdot)$ in addition to the ℓn constraints $\phi_1(\cdot), \ldots, \phi_{\ell n}(\cdot)$ from Section 4.2. Now $\mathcal{P}$ can proceed as before by running the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and then applying the Fiat-Shamir heuristic.

We remark that our technique of handling odd Hamming weight vectors can be employed to upgrade the static GS scheme recalled in Appendix C.2 to a fully dynamic one with the same signature sizes. Specifically, we first modify the fully dynamic GS scheme [81] by restricting the user public key $\mathbf{v}_\ell$ [3] to have odd Hamming weight. Then we modify the ZK presented in Section 5.3 to additionally show that $\mathbf{v}_\ell$ has odd Hamming weight. Since this change does not increase the witness length, the signature sizes remain the same.

5.6 Parameters and Efficiency

We now estimate the concrete sizes of FDABS scheme using our ZK protocol and Stern-like protocol [58]. The security of the scheme relies on the hardness of 2-$\mathsf{RNSD}_{n,2n,c}$ and 2-$\mathsf{RNSD}_{n,L+k,c}$ problems, as well as the security of the supporting ZK protocols. See Theorem 4. To this end, we use the same parameters proposed in Table 2 and let the bit length of attribute be $L = 128$ and the bit length of randomnesses for committing the attributes be $k = n + 2\lambda$.

We calculate various signature sizes regarding different security parameters and (ℓ, K) pairs, where 2^ℓ is the maximum number of attributes allowed in the system and K is the number of multiplication gates representing the policy P. Details are in Table 5. The results further confirm the superiority of our ZK

[3] Originally, it is required to be non-zero.

protocols. For $K = 2^9$, the signature sizes of [58] are $783\times \sim 839\times$ larger than ours. For $K = 2^{16}$, the differences of the two ZK are smaller, and their signature sizes are $288\times \sim 550\times$ larger than ours. The main reason is that their witness sizes are dominated by the term $2\ell \cdot \frac{2n}{c} \cdot 2^c$ and thus less sensitive to changes of the policy size K. In contrast, our witness size is $\ell + 2\ell n + L + k + K$, and thus is susceptible to the changes of K.

Table 5. Signature sizes of the FDABS scheme by employing our ZK proof system and Stern-like ZK arguments. The bit length of the attribute in the following instances are always chosen as $L = 128$.

$(2^\ell, K)$	128-bit security		256-bit security	
	This paper **(KB)**	Stern-type [58] **(MB)**	This paper **(KB)**	Stern-type [58] **(MB)**
$(2^{10}, 2^9)$	59.38	45.41	234.76	181.29
$(2^{10}, 2^{16})$	186.38	52.20	488.76	194.87
$(2^{15}, 2^9)$	84.39	67.30	334.78	268.85
$(2^{15}, 2^{16})$	211.39	74.08	588.78	282.43
$(2^{20}, 2^9)$	109.40	89.18	434.80	356.40
$(2^{20}, 2^{16})$	236.40	95.97	688.80	369.98

6 A New Code-Based Signature Scheme

A standard paradigm to construct a signature scheme is to first design a public-coin ZK protocol for proving the knowledge of a preimage of a one-way function and then apply the Fiat-Shamir [45] transform to make it non-interactive. Therefore, the technique from Section 3.2 directly implies a code-based signature scheme, which we name ReSolveD+ and has slightly smaller signature sizes than the state-of-the-art code-based signature scheme ReSolveD [33] based on RSD problems.

6.1 Description of the Signature Scheme

The secret key is binary string $\mathbf{x} \in \mathbb{F}_2^n$, and the public key is $\mathbf{y} = \mathbf{B} \cdot \mathsf{RE}(\mathbf{x})$, where $\mathbf{B} = (b_{i,j})_{i\in[n],j\in[m]} \in \mathbb{F}_2^{n\times m}$ for $c|n$ and $m = \frac{n}{c} \cdot 2^c$ are public parameters. To sign a message $M \in \{0,1\}^*$, the signer proves knowledge of $\mathbf{x}$ such that $\mathbf{y} = (y_1, \ldots, y_n)^\top = \mathbf{B} \cdot \mathsf{RE}(\mathbf{x})$. In particular, the signer prepares n polynomials of degree c:

$$g_i(\cdot) = \sum_{j=1}^{m} b_{i,j} f_j(X_1, \ldots, X_n) - y_i, \quad \forall \ \ i \in [n],$$

and the witness $\mathbf{x} \in \mathbb{F}_2^n$. Next, it runs the protocol $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and makes it non-interactive via the Fiat-Shamir transform. Let the resultant proof be π, which would be the signature. Verification of the signature is to verify the proof π. Correctness and security directly follows from that of $\Pi^t_{\mathsf{dD\text{-}Rep}}$ and the design paradigm, based on the hardness of the $\mathsf{RSD}_{n,n,c}$ problem.

Table 6. Parameters for ReSolveD+ Signature Scheme when $c = \{2, 3, 4\}$.

Scheme	Parameter set					Estimated
	n	c	$m = \frac{n}{c} \cdot 2^c$	$w = \frac{n}{c}$	τ	Bit Security
ReSolveD + -128-Var1-2	892	2	1784	446	14	128.10
ReSolveD + -128-Var2-2	892	2	1784	446	10	128.10
ReSolveD + -128-Var1-3	453	3	1208	151	14	128.31
ReSolveD + -128-Var2-3	453	3	1208	151	10	128.31
ReSolveD + -128-Var1-4	332	4	1328	83	14	128.33
ReSolveD + -128-Var2-4	332	4	1328	83	10	128.33

6.2 Parameters and Efficiency

We follow the approach in [30] to select parameters n, c. In particular, we estimate the complexity of linearization attack, ISD attack, and birthday paradox according to formulas [30], and take their minimum as the estimation of security level. Using this estimation, we choose the smallest parameter n by fixing $c = 2, 3, 4$, respectively so that it has complexity estimation 2^{128} following the footprint of [33]. Details are in Table 6. Regarding the parameters for VOLEitH, we employ the same optimizations adopted by Cui et al. [33] (instead of the non-optimized ones from Section 3.1), to have a fair comparison with them. In Table 7, we compare the signature sizes of our signature scheme ReSolveD+ with ReSolveD for different parameter sets. The results show that we achieve smallest signature sizes when $c = 3$. In addition, our scheme ReSolveD+ has slightly shorter signature sizes than [33] when $c = 3$ and $c = 4$. Note that Baum et al. [6] recently obtained better performances for FAEST signature scheme by proposing several optimizations. Since those optimizations apply to all protocols within the VOLEitH paradigm, both signature sizes of ReSolveD and ReSolveD+ could be further reduced by a few hundred bytes. In particular, let $\mathsf{T}_{\mathsf{open}} = 112$ for $\tau = 14$ and $\mathsf{T}_{\mathsf{open}} = 102$ for $\tau = 10$. Then the signature sizes are reduced by 256 bytes and 416 bytes, respectively. The optimized signature sizes are given in Table 7 in blue color.

We also compare our signature scheme with some other post-quantum signature schemes, for 128-bit security level in Table 8. The results show that our signature sizes are competitive with those schemes and are smallest among schemes based on SD and regular SD problems. Also, Bidoux et al. [19] proposed

Table 7. Comparison of our signature schemes for different choices of $c, \tau, \mathsf{T}_{\mathsf{open}}$ with the signature scheme proposed by Cui et al. [33] for the same security levels. The percentages in parenthesis are the increases/decreases of signature sizes compared to [33].

Scheme parameters		Signature sizes in bytes			
		CLY+24 [33]	$c = 2$	$c = 3$	$c = 4$
$\tau = 14$	$\mathsf{T}_{\mathsf{open}} =$ -	4082	4572(+12.0%)	4026(−1.4%)	4040(−1.0%)
	$\mathsf{T}_{\mathsf{open}} = 112$	3826	4316(+12.9%)	3770(−1.5%)	3784(−1.1%)
$\tau = 10$	$\mathsf{T}_{\mathsf{open}} =$ -	3510	3860(+10.0%)	3470(−1.1%)	3480(−0.9%)
	$\mathsf{T}_{\mathsf{open}} = 102$	3094	3444(+11.3%)	3054(−1.3%)	3064(−1.0%)

signature schemes based on Rank SD and MinRank problems from MPCitH and VOLEitH paradigm, we only include the variants that employ the optimizations from [6] within the VOLEitH paradigm. Adj et al. [1] proposed a signature scheme based on MinRank problem preceding [19] and had slightly larger signature sizes. Therefore, we do not include their results [1] in the table.

Conclusions and Open Questions. In this work, we advanced the state-of-the-art code-based cryptography by proposing new ZK protocols from VOLEitH paradigm. In particular, we presented ZK protocols for proving the correctness of the regular encoding process and various code-based relations. Built upon these ZK protocols, we obtained privacy-preserving signatures whose signature sizes are significantly smaller than those based on Stern-like ZK protocols. We view the problem of improving the computational efficiency, particularly decreasing the number of multiplications over large finite fields required in $\Pi^{t}_{\mathsf{dD\text{-}Rep}}$ and improving the realization of $\mathcal{F}^{p,q,S_{\Delta},\mathcal{C},l,\mathcal{L}}_{\mathsf{sVOLE}}$, as fascinating opening questions for future investigations.

Acknowledgements. We would like to thank Liping Wang, Khoa Nguyen, Hongrui Cui, Hanlin Liu, Xindong Liu, Yizhou Yao, and Hongqing Liu for their valuable discussions of this work. We are also very grateful for the insightful comments and suggestions from the anonymous reviewers of ASIACRYPT 2024. This work was supported in part by the National Key Research and Development Program under Grants 2020YFA0712300 and 2022YFA1004900, and the National Natural Science Foundation of China under Grant numbers 62272303 and 12101404.

References

1. G. Adj, L. Rivera-Zamarripa, and J. A. Verbel. Minrank in the head - short signatures from zero-knowledge proofs. In *AFRICACRYPT 2023*, volume 14064 of *LNCS*, pages 3–27. Springer, 2023.

Table 8. Comparison of our scheme with some post-quantum signature schemes, targeting 128-bit security level. All the signature schemes within the VOLEitH paradigm are optimized using the techniques from [6].

Scheme	Sizes in KB			Assumptions	Paradigm
	$\|\mathsf{sig}\|$	$\|\mathsf{pk}\|$	$\|\mathsf{sig}\| + \|\mathsf{pk}\|$		
BGKM23 [20]-Sig1	24.0	0.1	24.1	SD over $\mathbb{F}_2$	Stern-type
BGKM23 [20]-Sig2	19.3	0.2	19.5	(QC)SD over $\mathbb{F}_2$	
BGKM23 [20]-Sig3	15.6	0.2	15.8	(QC)SD over $\mathbb{F}_2$	
FJR22 [41]-Var2s	11.8	0.09	11.89	SD over $\mathbb{F}_2$	MPCitH
FJR22 [41]-Var3f	11.5	0.14	11.64		
FJR22 [41]-Var3s	8.26	0.14	8.4		
CCJ23 [30]-rsd-f	12.52	0.09	12.61	RSD over $\mathbb{F}_2$	MPCitH
CCJ23 [30]-rsd-m1	9.69	0.09	9.78		
CCJ23 [30]-rsd-m2	9.13	0.09	9.22		
CCJ23 [30]-rsd-s	8.55	0.09	8.64		
MGH+23 [66]-faster	11.83	0.14	11.97	SD over $\mathbb{F}_{256}$	MPCitH
MGH+23 [66]-short	8.28	0.14	8.42		
MGH+23 [66]-shorter	6.63	0.14	6.77		
MGH+23 [66]-shortest	5.56	0.14	5.7		
[67]-Vanilla-short	8.27	0.14	8.6	SD over $\mathbb{F}_{256}$	MPCitH (QROM)
[67]-Vanilla-shorter	6.6	0.14	6.94		
[67]-Pow-short	7.78	0.14	8.11		
[67]-Pow-shorter	6.06	0.14	6.34		
BCC+24 [27]-fast	7.07	0.10	7.17	RSD over $\mathbb{F}_2$	MPCitH
BCC+24 [27]-medium	5.73	0.10	5.83		
BCC+24 [27]-compact	5.13	0.10	5.23		
[6] FAESTER-128s	4.49	0.03	4.52	OW of AES128	VOLEitH
[6] FAESTER-128f	5.91	0.03	5.94		
[6] FAESTER-EM-128s	4.07	0.03	4.10		
[6] FAESTER-EM-128f	5.32	0.03	5.35		
[6] FAESTER-d7-128s	4.27	0.03	4.30		
[6] FAESTER-d7-128f	5.60	0.03	5.63		
[19] $\mathsf{RSD_s}$-short	3.51	0.05	3.56	Rank SD	VOLEitH
[19] $\mathsf{RSD_s}$-shortest	2.84	0.05	2.89		
[19] MinRank-short	3.46	0.03	3.49	MinRank	VOLEitH
[19] MinRank-shortest	2.81	0.03	2.84		
[33] ReSolveD-128-Var1	3.74	0.08	3.82	RSD over $\mathbb{F}_2$	VOLEitH
[33] ReSolveD-128-Var2	3.02	0.08	3.10		
ReSolveD + -128-Var1-3	3.68	0.06	3.74	RSD over $\mathbb{F}_2$	VOLEitH
ReSolveD + -128-Var2-3	2.98	0.06	3.04		

2. M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. Von Maurich, R. Misoczki, R. Niederhagen, et al. Classic mceliece: conservative code-based cryptography. 2022. `https://classic.mceliece.org/nist.html`.
3. D. Augot, M. Finiasz, P. Gaborit, S. Manuel, and N. Sendrier. Sha-3 proposal: Fsb. *Submission to NIST*, pages 81–85, 2008. `https://www.rocq.inria.fr/secret/CBCrypto/fsbdoc.pdf`.
4. D. Augot, M. Finiasz, and N. Sendrier. A fast provably secure cryptographic hash function. *IACR Cryptol. ePrint Arch.*, page 230, 2003.
5. D. Augot, M. Finiasz, and N. Sendrier. A family of fast syndrome based cryptographic hash functions. In *Mycrypt 2005*, volume 3715 of *LNCS*, pages 64–83. Springer, 2005.
6. C. Baum, W. Beullens, S. Mukherjee, E. Orsini, S. Ramacher, C. Rechberger, L. Roy, and P. Scholl. One tree to rule them all: Optimizing GGM trees and owfs for post-quantum signatures. *IACR Cryptol. ePrint Arch.*, page 490, 2024.
7. C. Baum, L. Braun, C. D. de Saint Guilhem, M. Klooß, C. Majenz, S. Mukherjee, S. Ramacher, C. Rechberger, E. Orsini, L. Roy, et al. Faest: Algorithm specifications. 2023.
8. C. Baum, L. Braun, C. D. de Saint Guilhem, M. Klooß, E. Orsini, L. Roy, and P. Scholl. Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head. In *CRYPTO 2023*, volume 14085 of *LNCS*, pages 581–615. Springer, 2023.
9. C. Baum, L. Braun, A. Munch-Hansen, and P. Scholl. Moz$\mathbb{Z}_{2^k}$arella: Efficient vector-ole and zero-knowledge proofs over $\mathbb{Z}_{2^k}$. In *CRYPTO 2022*, volume 13510 of *LNCS*, pages 329–358. Springer, 2022.
10. C. Baum, S. Dittmer, P. Scholl, and X. Wang. Sok: vector ole-based zero-knowledge protocols. *Des. Codes Cryptogr.*, 91(11):3527–3561, 2023.
11. C. Baum, A. J. Malozemoff, M. B. Rosen, and P. Scholl. Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In *CRYPTO 2021*, volume 12828 of *LNCS*, pages 92–122. Springer, 2021.
12. M. Bellare and G. Fuchsbauer. Policy-based signatures. In *PKC 2014*, volume 8383 of *LNCS*, pages 520–537. Springer, 2014.
13. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003,*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.
14. J. C. Benaloh and M. de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285. Springer, 1993.
15. A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *J. Cryptol.*, 22(1):114–138, 2009.
16. D. J. Bernstein, T. Lange, R. Niederhagen, C. Peters, and P. Schwabe. Fsbday: Implementing wagner's generalized birthday attack against the sha-3 round-1 candidate fsb. In *INDOCRYPT 2009*, volume 5922 of *LNCS*, pages 18–38. Springer, 2009.
17. D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe. Faster 2-regular information-set decoding. In *IWCC 2011*, volume 6639 of *LNCS*, pages 81–98. Springer, 2011.
18. D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe. Really fast syndrome-based hashing. In *AFRICACRYPT 2011*, volume 6737 of *LNCS*, pages 134–152. Springer, 2011.

19. L. Bidoux, T. Feneuil, P. Gaborit, R. Neveu, and M. Rivain. Dual support decomposition in the head: Shorter signatures from rank SD and minrank. *IACR Cryptol. ePrint Arch.*, page 541, 2024.
20. L. Bidoux, P. Gaborit, M. Kulkarni, and V. Mateu. Code-based signatures from new proofs of knowledge for the syndrome decoding problem. *Des. Codes Cryptogr.*, 91(2):497–544, 2023.
21. D. Boneh, S. Eskandarian, and B. Fisch. Post-quantum EPID signatures from symmetric primitives. In M. Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 251–271. Springer, 2019.
22. E. Boyle, G. Couteau, N. Gilboa, and Y. Ishai. Compressing vector OLE. In *CCS 2018*, pages 896–912. ACM, 2018.
23. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, P. Rindal, and P. Scholl. Efficient two-round OT extension and silent non-interactive secure computation. In *CCS 2019*, pages 291–308. ACM, 2019.
24. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In *CRYPTO 2019*, volume 11694 of *LNCS*, pages 489–518. Springer, 2019.
25. P. Briaud and M. Øygarden. A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In *EUROCRYPT 2023*, volume 14008 of *LNCS*, pages 391–422. Springer, 2023.
26. E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *CCS 2004*, pages 132–145. ACM, 2004.
27. D. Bui, E. Carozza, G. Couteau, D. Goudarzi, and A. Joux. Short signatures from regular syndrome decoding, revisited. *IACR Cryptol. ePrint Arch.*, page 252, 2024.
28. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
29. C. Carlet, editor. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, Cambridge, 2020.
30. E. Carozza, G. Couteau, and A. Joux. Short signatures from regular syndrome decoding in the head. In *EUROCRYPT 2023*, volume 14008 of *LNCS*, pages 532–563. Springer, 2023.
31. D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.
32. D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology - EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
33. H. Cui, H. Liu, D. Yan, K. Yang, Y. Yu, and K. Zhang. Resolved: Shorter signatures from regular syndrome decoding and vole-in-the-head. In *PKC 2024*, volume 14601 of *LNCS*, pages 229–258. Springer, 2024.
34. D. Derler, S. Ramacher, and D. Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto 2018*, volume 10786 of *LNCS*, pages 419–440. Springer, 2018.
35. S. Dittmer, Y. Ishai, S. Lu, and R. Ostrovsky. Improving line-point zero knowledge: Two multiplications for the price of one. In *CCS 2022*, pages 829–841. ACM, 2022.
36. S. Dittmer, Y. Ishai, and R. Ostrovsky. Line-point zero knowledge and its applications. In *ITC 2021*, volume 199 of *LIPIcs*, pages 5:1–5:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
37. A. El Kaafarani and S. Katsumata. Attribute-based signatures for unbounded circuits in the ROM and efficient instantiations from lattices. In *PKC 2018*, volume 10770 of *LNCS*, pages 89–119. Springer, 2018.

38. A. Esser, R. Kübler, and A. May. LPN decoded. In *CRYPTO 2017*, volume 10402 of *LNCS*, pages 486–514. Springer, 2017.
39. A. Esser and P. Santini. Not just regular decoding: Asymptotics and improvements of regular syndrome decoding attacks. *IACR Cryptol. ePrint Arch.*, page 1568, 2023.
40. M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, and H. Wang. Provably secure group signature schemes from code-based assumptions. *IEEE Trans. Inf. Theory*, 66(9):5754–5773, 2020.
41. T. Feneuil, A. Joux, and M. Rivain. Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In *CRYPTO 2022*, volume 13508 of *LNCS*, pages 541–572. Springer, 2022.
42. T. Feneuil, A. Joux, and M. Rivain. Shared permutation for syndrome decoding: new zero-knowledge protocol and code-based signature. *Des. Codes Cryptogr.*, 91(2):563–608, 2023.
43. T. Feneuil and M. Rivain. Threshold linear secret sharing to the rescue of mpc-in-the-head. In *ASIACRYPT 2023*, volume 14438 of *LNCS*, pages 441–473. Springer, 2023.
44. H. Feng, J. Liu, and Q. Wu. Secure stern signatures in quantum random oracle model. In *Information Security - ISC 2019*, volume 11723 of *LNCS*, pages 425–444. Springer, 2019.
45. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
46. C. Ganesh, C. Orlandi, M. Pancholi, A. Takahashi, and D. Tschudi. Fiat-shamir bulletproofs are non-malleable (in the random oracle model). *IACR Cryptol. ePrint Arch.*, page 147, 2023.
47. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.
48. J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT 2015*, volume 9057 of *LNCS*, pages 253–280. Springer, 2015.
49. S. Gueron, E. Persichetti, and P. Santini. Designing a practical code-based signature scheme from zero-knowledge proofs with trusted setup. *Cryptogr.*, 6(1):5, 2022.
50. C. Hazay, E. Orsini, P. Scholl, and E. Soria-Vazquez. Tinykeys: A new approach to efficient multi-party computation. In *CRYPTO 2018*, volume 10993 of *LNCS*, pages 3–33. Springer, 2018.
51. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.
52. C. Jeudy, A. Roux-Langlois, and O. Sanders. Lattice signature with efficient protocols, application to anonymous credentials. In *CRYPTO 2023*, volume 14082 of *LNCS*, pages 351–383. Springer, 2023.
53. J. Katz, V. Kolesnikov, and X. Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *CCS 2018*, pages 525–537. ACM, 2018.
54. A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. In *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 181–199. Springer, 2007.
55. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*, volume 10032 of *LNCS*, pages 373–403, 2016.

56. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 1–31. Springer, 2016.
57. F. Lin, C. Xing, and Y. Yao. More efficient zero-knowledge protocols over $\mathbb{Z}_{2^k}$ via galois rings. *IACR Cryptol. ePrint Arch.*, page 150, 2023.
58. S. Ling, K. Nguyen, D. H. Phan, K. H. Tang, H. Wang, and Y. Xu. Fully dynamic attribute-based signatures for circuits from codes. In *PKC 2024*, volume 14601 of *LNCS*, pages 37–73. Springer, 2024.
59. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer, 2013.
60. H. Liu, X. Wang, K. Yang, and Y. Yu. The hardness of LPN over any integer ring and field for PCG applications. *IACR Cryptol. ePrint Arch.*, page 712, 2022.
61. H. Liu, X. Wang, K. Yang, and Y. Yu. The hardness of LPN over any integer ring and field for PCG applications. In *EUROCRYPT 2024*, volume 14656 of *LNCS*, pages 149–179. Springer, 2024.
62. X. Liu and L. Wang. Short code-based one-out-of-many proofs and applications. In *PKC 2024*, volume 14602 of *LNCS*, pages 370–399. Springer, 2024.
63. V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.
64. V. Lyubashevsky and N. K. Nguyen. BLOOM: bimodal lattice one-out-of-many proofs and applications. In S. Agrawal and D. Lin, editors, *ASIACRYPT 2022*, volume 13794 of *LNCS*, pages 95–125. Springer, 2022.
65. R. J. McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.
66. C. A. Melchor, N. Gama, J. Howe, A. Hülsing, D. Joseph, and D. Yue. The return of the sdith. In *EUROCRYPT 2023*, volume 14008 of *LNCS*, pages 564–596. Springer, 2023.
67. C. A. Melchor, A. Hülsing, D. Joseph, C. Majenz, E. Ronen, and D. Yue. Sdith in the QROM. In *ASIACRYPT 2023*, volume 14444 of *LNCS*, pages 317–350. Springer, 2023.
68. R. C. Merkle. A certified digital signature. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 218–238. Springer, 1989.
69. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC 1990*, pages 427–437. ACM, 1990.
70. K. Nguyen, R. Safavi-Naini, W. Susilo, H. Wang, Y. Xu, and N. Zeng. Group encryption: Full dynamicity, message filtering and code-based instantiation. In *PKC 2021*, volume 12711 of *LNCS*, pages 678–708. Springer, 2021. Full version is available at `https://eprint.iacr.org/2021/226`.
71. K. Nguyen, H. Tang, H. Wang, and N. Zeng. New code-based privacy-preserving cryptographic constructions. In *ASIACRYPT 2019*, volume 11922 of *LNCS*, pages 25–55. Springer, 2019.
72. R. Nojima, H. Imai, K. Kobara, and K. Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Des. Codes Cryptogr.*, 49(1-3):289–305, 2008.
73. R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.
74. E. Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8(5):5–9, 1962.
75. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.

76. L. Roy. Softspokenot: Quieter OT extension from small-field silent VOLE in the minicrypt model. In *CRYPTO 2022*, volume 13507 of *LNCS*, pages 657–687. Springer, 2022.
77. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS 1999*, pages 543–553. IEEE Computer Society, 1999.
78. J. Stern. A new paradigm for public key identification. *IEEE Trans. Inf. Theory*, 42(6):1757–1768, 1996.
79. D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015.
80. D. A. Wagner. A generalized birthday problem. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, 2002.
81. L. Wang, J. Chen, H. Dai, and C. Tao. Efficient code-based fully dynamic group signature scheme. *Theor. Comput. Sci.*, 990:114407, 2024.
82. C. Weng, K. Yang, J. Katz, and X. Wang. Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In *IEEE Symposium on Security and Privacy 2021*, pages 1074–1091. IEEE, 2021.
83. C. Weng, K. Yang, Z. Yang, X. Xie, and X. Wang. Antman: Interactive zero-knowledge proofs with sublinear communication. In *CCS 2022*, pages 2901–2914. ACM, 2022.
84. K. Yang, P. Sarkar, C. Weng, and X. Wang. Quicksilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In *CCS 2021*, pages 2986–3001. ACM, 2021.
85. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO 2019*, volume 11692 of *LNCS*, pages 147–175. Springer, 2019.

Supplementary Material

A More Preliminaries

A.1 Syntax and Security Requirements of Accumulator

An accumulator scheme is a tuple of algorithms $(\mathsf{Setup}, \mathsf{Accu}, \mathsf{WitGen}, \mathsf{Verify})$ defined as follows.

$\mathsf{TSetup}(1^\lambda)$**.** Given a security parameter 1^λ, output the public parameter pp.

$\mathsf{TAccu}_{\mathsf{pp}}(R)$**.** Given a set R with n data values as $R = \{\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}\}$, output an accumulated value $\mathbf{u}$.

$\mathsf{TWitGen}_{\mathsf{pp}}(R, \mathbf{d})$**.** It takes as inputs the set R and a value $\mathbf{d}$, and outputs a witness w such that $\mathbf{d}$ is accumulated in $\mathsf{TAcc}(R)$. If $\mathbf{d} \notin R$, it directly returns $\perp$.

$\mathsf{TVerify}_{\mathsf{pp}}(\mathbf{u}, \mathbf{d}, w)$**.** This algorithm take as inputs the accumulator value $\mathbf{u}$ and $(\mathbf{d}, w)$, and outputs 1 if $(\mathbf{d}, w)$ is valid for the accumulator value $\mathbf{u}$, otherwise return 0.

Correctness. For all $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, the accumulator is said correct if

$$\mathsf{TVerify}_{\mathsf{pp}}(\mathsf{TAccu}_{\mathsf{pp}}(R), \mathbf{d}, \mathsf{TWitGen}_{\mathsf{pp}}(R, \mathbf{d})) = 1 \text{ for all } \mathbf{d} \in R.$$

Security. An accumulator is secure, if it is infeasible to output a valid witness for a value $\mathbf{d}^*$ not chosen from the data value set, i.e., an accumulator is secure if for all PPT adversaries $\mathcal{A}$

$$\begin{aligned}\Pr[\mathsf{pp} \leftarrow &\mathsf{TSetup}(1^\lambda); (R, \mathbf{d}^*, w^*) \leftarrow \mathcal{A} : \\ &\mathsf{d}^* \notin R \ \wedge \ \mathsf{TVerify}_{\mathsf{pp}}(\mathsf{TAccu}_{\mathsf{pp}}(R), \mathbf{d}^*, w^*) = 1] = \mathsf{negl}(\lambda).\end{aligned}$$

A.2 Universal Hash function

When realizing the delayed functionality $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}$, we utilize an $\mathbb{F}_p^l$-hiding and universal linear hash function for consistency check. We recall its definitions following [76,8].

Definition 5. *A linear ϵ-almost universal family of hashes is a family of matrices $\mathcal{H} \subseteq \mathbb{F}_q^{s \times l}$ such that for any nonzero $\mathbf{v} \in \mathbb{F}_q^l$, $\Pr_{\mathsf{H} \leftarrow \mathcal{H}}[\mathsf{H}\mathbf{v} = 0] \leq \epsilon$.*

Definition 6. *Let p and $q = p^k$ be prime powers. A matrix $\mathsf{H} \in \mathbb{F}_q^{l+h}$ is $\mathbb{F}_p^l$-hiding if the distribution of $\mathsf{H} \cdot \mathbf{v}$ is independent from $\mathbf{v}_{[1,l]}$ for a uniformly random $\mathbf{v} \in \mathbb{F}_p^{l+h}$. Equivalently, if $\mathsf{H}' \in \mathbb{F}_p^{(s \cdot k) \times (l+h)}$ is H reinterpreted as an $\mathbb{F}_p$-linear map, the the column space of H' must equal the column space of $\mathsf{H}_{[l+1,l+h]}$. A hash family $\mathcal{H} \subseteq \mathbb{F}_q^{s \times (l+h)}$ is $\mathbb{F}_p^l$-hiding if every $\mathsf{H} \in \mathcal{H}$ is $\mathbb{F}_p^l$-hiding.*

A.3 Definitions of Some Functionalities

In this section, we provide details of all the functionalities mentioned in Section 2.

Functionality 1: $\mathcal{F}_{\mathsf{VOLE}}^{p,r}$

The functionality runs between two parties $\mathcal{P}$ and $\mathcal{V}$ and the adversary $\mathcal{A}$. The **Initialize** phase is run once only.

- **Initialize:** Upon receiving (init) from $\mathcal{P}$ and $\mathcal{V}$, sample $\Delta \leftarrow \mathbb{F}_{p^r}$ if $\mathcal{V}$ is honest, and receive $\Delta \in \mathbb{F}_{p^r}$ from the adversary $\mathcal{A}$ otherwise. Store Δ and send it to $\mathcal{V}$.
- **Extend:** Upon receiving (extend, l) from $\mathcal{P}$ and $\mathcal{V}$, do as follows:
  - If $\mathcal{V}$ is honest, sample $\mathsf{K} \leftarrow \mathbb{F}_{p^r}^l$. Otherwise, receive $\mathsf{K} \in \mathbb{F}_{p^r}^l$ from $\mathcal{A}$.
  - If $\mathcal{P}$ is honest, sample $\mathbf{u} \leftarrow \mathbb{F}_p^l$ and compute $\mathsf{M} := \mathsf{K} - \Delta \cdot \mathbf{u} \in \mathbb{F}_{p^r}^l$. Otherwise, receive $\mathbf{u} \in \mathbb{F}_{p^r}^l$ and $\mathsf{M} \in \mathbb{F}_{p^r}^l$ from $\mathcal{A}$, and then recompute $\mathsf{K} := \mathsf{M} + \Delta \cdot \mathbf{u}$.
  - Send $(\mathbf{u}, \mathsf{M})$ to $\mathcal{P}$ and K to $\mathcal{V}$.

Functionality 2: $\mathcal{F}_{\mathsf{VOPE}}^{p,r}$

The functionality runs with two parties $\mathcal{P}$ and $\mathcal{V}$ and the adversary $\mathcal{A}$. The **Initialize** phase is run once only.

- **Initialize:** Upon receiving (init) from $\mathcal{P}$ and $\mathcal{V}$, sample $\Delta \leftarrow \mathbb{F}_{p^r}$ if $\mathcal{V}$ is honest, and receive $\Delta \in \mathbb{F}_{p^r}$ from the adversary $\mathcal{A}$ otherwise. Store Δ and send it to $\mathcal{V}$.
- **Generate VOPE:** Upon receiving (VOPE, d) from $\mathcal{P}$ and $\mathcal{V}$, do as follows:
  - If $\mathcal{V}$ is honest, sample $B \leftarrow \mathbb{F}_{p^r}$. Otherwise, receive $B \in \mathbb{F}_{p^r}$ from $\mathcal{A}$.
  - If $\mathcal{P}$ is honest, sample $A_i \leftarrow \mathbb{F}_{p^r}$ for $i \in [d]$ and compute $A_0 := B - \sum_{i\in[d]} A_i \cdot \Delta^i$. Otherwise, receive $\{A_i\}_{i\in[0,d]}$ with $A_i \in \mathbb{F}_{p^r}$ from $\mathcal{A}$, and recompute $B := \sum_{i\in[0,d]} A_i \cdot \Delta^i$.
  - Send $\{A_i\}_{i\in[0,d]}$ to $\mathcal{P}$ and B to $\mathcal{V}$.

Functionality 3: $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C},l,\mathcal{L}}$

The functionality interacts with a sender $\mathcal{P}$, a receiver $\mathcal{V}$ and an adversary $\mathcal{A}$. It is parametrized by integers l and p, q, such that $q = p^k$, as well as an $[n_\mathcal{C}, k_\mathcal{C}, d_\mathcal{C}]_p$ linear code $\mathcal{C}$ over $\mathbb{F}_p$ with a generator matrix $\mathbf{G}_\mathcal{C} \in \mathbb{F}_p^{k_\mathcal{C} \times n_\mathcal{C}}$.

1. Upon receiving (init) from the prover $\mathcal{P}$ and the verifier $\mathcal{V}$, sample $\mathbf{U} \leftarrow \mathbb{F}_p^{l\times k_\mathcal{C}}$, $\mathbf{V} \leftarrow \mathbb{F}_p^{l\times n_\mathcal{C}}$ and $\boldsymbol{\Delta} \leftarrow S_\Delta \subseteq \mathbb{F}_q^{n_\mathcal{C}}$ and set $\mathbf{Q} := \mathbf{V} + \mathbf{U}\mathbf{G}_\mathcal{C}\mathsf{diag}(\boldsymbol{\Delta})$.
   - If $\mathcal{P}$ is corrupt, receive $\mathbf{U}, \mathbf{V}$ from the adversary $\mathcal{A}$, and recompute $\mathbf{Q}$ as above.
   - If $\mathcal{V}$ is corrupt, receive $\boldsymbol{\Delta}, \mathbf{Q}$ from the adversary $\mathcal{A}$, and compute $\mathbf{V} := \mathbf{Q} - \mathbf{U}\mathbf{G}_\mathcal{C}\mathsf{diag}(\boldsymbol{\Delta})$.
   - Send $(\mathbf{U}, \mathbf{V})$ to $\mathcal{P}$.
   - If $\mathcal{P}$ is corrupt, receive a leakage query $L \in \mathcal{L}$ from $\mathcal{A}$.
2. Upon receiving (get) from $\mathcal{V}$, if $\boldsymbol{\Delta} \notin L$, send $(\mathsf{check\text{-}failed})$ to $\mathcal{V}$ and abort. Otherwise, send $(\boldsymbol{\Delta}, \mathbf{Q})$ to $\mathcal{V}$.

Functionality 4: $\mathcal{F}^{t}_{\mathsf{dD\text{-}ZK}}$

Upon receiving $(\mathsf{prove}, \{f_i\}_{i\in[t]}, w_1, \ldots, w_l)$ from $\mathcal{P}$ and $(\mathsf{prove}, \{f_i\}_{i\in[t]})$ from $\mathcal{V}$, where $\{f_i\}_{i\in[t]}$ are degree at most d polynomials over l variables. If $f_i(w_1, \ldots, w_l) = 0$ for all $i \in [t]$, the functionality sends true to $\mathcal{V}$. Otherwise, it sends false to $\mathcal{V}$.

B Deferred Security Proof of Theorem 1

Proof. In the following, we construct simulators for the malicious prover and verifier cases to argue soundness and zero-knowledge properties respectively.

Malicious Prover. The simulator interacts with adversary $\mathcal{A}$ as follows:

1. $\mathcal{S}$ emulates $\mathcal{F}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},l+(d-1)r\tau}_{\mathsf{sVOLE}}$ for $\mathcal{A}$ by sampling τ uniform values $\Delta_i \in \mathbb{F}_q^\tau$, receiving $\mathbf{u} \in \mathbb{F}_p^{l+(d-1)r\tau}$ and $\mathbf{V} \in \mathbb{F}_q^{(l+(d-1)r\tau)\times\tau}$ from $\mathcal{A}$, and computing $\mathbf{Q} = \mathbf{V} + \mathbf{u}\mathbf{G}_{\mathcal{C}}\mathsf{diag}(\mathbf{\Delta})$.
2. When $\mathcal{A}$ sends $\mathbf{d} \in \mathbb{F}_p^l$ in step 2 of Round 1, $\mathcal{S}$ recovers $\mathbf{w} = \mathbf{d} + \mathbf{u}_{[1,l]}$.
3. $\mathcal{S}$ then executes the rest of the protocol as an honest verifier, randomly sampling the t challenges $\chi_i \in \mathbb{F}_q^\tau$ and using $\mathbf{\Delta}$ and $\mathbf{Q}$ as defined during the emulation of $\mathcal{F}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},l+(d-1)r\tau}_{\mathsf{sVOLE}}$ to execute the verification step. If the check at step 6 of the verification fails, then $\mathcal{S}$ returns $\mathbf{w} = \perp$ to the functionality; if it passes, $\mathcal{S}$ outputs $\mathbf{w}$ as a valid witness.

Due to the honest sampling of the Δ_i and χ_i values, the view of $\mathcal{A}$ simulated by $\mathcal{S}$ has the identical distribution as its view in the real-world execution. Whenever a real-word verifier rejects $\mathcal{A}$'s proof, the ideal-world verifier rejects as well since $\mathcal{S}$ return $\perp$. Thus, it only remains to bound the error probability ϵ of a real-world verifier accepting the proof when in fact the witness $\mathbf{w}$ extracted by $\mathcal{S}$ does not satisfy the constraint system f_i.

Let $f_i(\mathbf{w}) = y_i$ for some $y_i \in \mathbb{F}_{p^k}$, for each $i \in [t]$, where $\mathbf{w}$ is extracted from $\mathcal{A}$ by $\mathcal{S}$ as above. According to the definition of $c_i(\Delta)$ for $i \in [t]$, we have:

$$\begin{aligned} c_i(\Delta) &= \bar{f}_i(\mathbf{w}) \cdot \Delta^d + \sum_{j=0}^{d-1} A_{i,j} \cdot \Delta^j \\ &= y_i \cdot \Delta^d + \sum_{j=0}^{d-1} A_{i,j} \cdot \Delta^j \end{aligned}$$

where the embedding constraints $\bar{f}_i \in \mathbb{F}_{q^\tau}[X_1, \ldots, X_l]$ produce an embedded $y_i \hookrightarrow \mathbb{F}_{q^\tau}$.

In step 3 of Round 3, $\mathcal{S}$ receives $\tilde{a}'_0 = \tilde{a}_0 + e_0, \ldots, \tilde{a}'_{d-1} = \tilde{a}_{d-1} + e_{d-1}$ from $\mathcal{A}$, where $\tilde{a}_0, \ldots, \tilde{a}_{d-1}$ are computed with $\mathbf{w}$ and the additional $(d-1)r\tau$ sVOLE

correlations used for $\{\mathbf{u}_j^*, \mathbf{v}_j^*\}_{j\in[1,d-1]}$ as well as $\{A_i^*\}_{i\in[0,d-1]}$, and $e_0, \ldots, e_{d-1} \in \mathbb{F}_{q^\tau}$ are error terms chosen by $\mathcal{A}$. By expanding the computation of $\tilde{c}$, we have:

$$\begin{aligned}
\tilde{c} &= \sum_{i\in[t]} \chi_i \cdot c_i(\Delta) + B^* \\
&= \sum_{i\in[t]} \chi_i \cdot (y_i \cdot \Delta^d + \sum_{j=0}^{d-1} A_{i,j} \cdot \Delta^j) + \sum_{j=0}^{d-1} A_j^* \cdot \Delta^j \\
&= \Delta^d \cdot \sum_{i\in[t]} y_i\chi_i + \Delta^{d-1} \cdot (\sum_{i\in[t]} A_{i,d-1}\chi_i + A_{d-1}^*) + \ldots + (\sum_{i\in[t]} A_{i,0}\chi_i + A_0^*) \\
&= \Delta^d \cdot \sum_{i\in[t]} y_i\chi_i + \Delta^{d-1} \cdot (\tilde{a}'_{d-1} - e_{d-1}) + \ldots + (\tilde{a}'_0 - e_0).
\end{aligned}$$

If the real-word verifier accepts the proof, then it must hold that $\tilde{c} = \sum_{j=0}^{d-1} \tilde{a}'_j \cdot \Delta^j$. Therefore, we have the following:

$$\Delta^d \cdot \sum_{i\in[t]} y_i\chi_i - (\Delta^{d-1}e_{d-1} + \ldots + \Delta \cdot e_1 + e_0) = 0. \tag{27}$$

If the random choice of χ_i leads to $\sum_{i\in[t]} y_i\chi_i = 0$, then the optimal strategy for the prover is to set $e_0 = 0, \ldots, e_{d-1} = 0$, i.e., compute $\{\tilde{a}_i\}_{i\in[0,d-1]}$ honestly. This ensures that (27) will hold with probability 1 for any value of Δ. By assumption that the extracted witness $\mathbf{w}$ does not satisfy the constraint system, there exists at least one $y_i \neq 0$. Then the probability that $\sum_{i\in[t]} y_i\chi_i = 0$ is at most $1/p^{r\tau}$, since $\chi_i \in \mathbb{F}_{q^\tau}$ are sampled uniformly at random after the y_i are committed to.

If $\sum_{i\in[t]} y_i\chi_i \neq 0$, then equation (27) can be solved for at most d values of Δ by the simulator based on the values it received from $\mathcal{A}$. If there are no solutions, $\mathcal{S}$ aborts; otherwise, it submits the solution(s) $\Delta \in \mathbb{F}_q^\tau$ to its internally simulated $\mathcal{F}_{\mathsf{sVOLE}}$ which aborts if they differ from the sampled values. The probability that the solutions to equation (27) will equal the sampled Δ is at most $d \cdot |S_\Delta|^{-1}$, since $\Delta \leftarrow S_\Delta$ uniformly at random and it is kept secret from $\mathcal{A}$.

Using the union bound, we conclude that the soundness error is upper bounded by $1/p^{r\tau} + d|S_\Delta|^{-1}$.

Semi-honest Verifier. The simulator interacts with adversary $\mathcal{A}$ as follows:

1. $\mathcal{S}$ emulates $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},l+(d-1)r\tau}$ for $\mathcal{A}$ by receiving τ uniform values $\Delta_i \in \mathbb{F}_q$ and $l + (d-1)r\tau$ outputs $\mathbf{Q} \in \mathbb{F}_q^{(l+(d-1)r\tau)\times\tau}$.
2. $\mathcal{S}$ then simulates Round 1 of the prover by sending l uniform values $\mathbf{d} \in \mathbb{F}_p^l$ to $\mathcal{A}$.
3. To simulate $\{\tilde{a}_i\}_{i\in[0,d-1]}$ in step 3 of Round 3, $\mathcal{S}$ first computes $\tilde{c}$ based on the received $\mathbf{\Delta}$ and $\mathbf{Q}$. Subsequently, it samples $\tilde{a}_1, \ldots, \tilde{a}_{d-1}$ at random and sets $\tilde{a}_0 = \tilde{c} - (\tilde{a}_1 \cdot \Delta + \ldots + \tilde{a}_{d-1} \cdot \Delta^{d-1})$. Then, $\mathcal{S}$ sends $\{\tilde{a}_i\}_{i\in[0,d-1]}$ to $\mathcal{A}$.

Note that $\mathbf{u}$ and $\mathbf{V}$ are kept secret from $\mathcal{A}$. Therefore, we conclude that the view of $\mathcal{A}$ in the simulation is indistinguishable from the real-world execution.

C Ring, Group, and Attribute-Based Signature Schemes

In this section, we recall the RS scheme, the GS scheme proposed by Nguyen et al. [71], and the FDABS scheme by Ling et al. [58]. Before that, we recall the definitions of them as put forward in [15,48], [13], and [58].

C.1 Definition of Ring Signatures

A ring signature scheme $\mathcal{RS} = (\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify})$ consists of a quadruple of polynomial-time algorithms defines as follows.

$\mathsf{RSetup}(1^\lambda)$. On input the security parameter 1^λ, output the public parameter pp available to all users.

$\mathsf{RKgen}(\mathsf{pp})$. On input pp, output a pair of public key and the corresponding secret signing key $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{RSign}(\mathsf{pp}, (\mathsf{sk}, M, R))$. Take as inputs pp, sk, a message $M \in \{0,1\}^*$ and a ring $R = (\mathsf{pk}_0, \ldots, \mathsf{pk}_{N-1})$. Output a ring signature Σ on M with respect to the ring R.

$\mathsf{RVerify}(M, R, \Sigma)$. The deterministic algorithm verifies a purported ring signature Σ on the message M with respect to the ring of public keys R. It outputs 1 if accepting and 0 if rejecting the ring signature.

Definition 7 (Correctness). *A ring signature* $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{Rsign}, \mathsf{RVerify})$ *is correct if for any* $\mathsf{pp} \leftarrow \mathsf{RSetup}(1^\lambda)$*, any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{RKgen}(\mathsf{pp})$*, any* R *such that* $\mathsf{pk} \in R$*, any* $M \in \{0,1\}^*$*, we have* $\mathsf{RVerify}\big(M, R, \mathsf{RSign}(\mathsf{pp}, (\mathsf{sk}, M, R))\big) = 1$.

A ring signature is considered secure if it satisfies two key properties: unforgeability with respect to insider corruption, and statistical anonymity. The unforgeability with respect to insider corruption ensures that it is infeasible to forge a ring signature without controlling one of the ring members. The statistical anonymity requirement dictates that signatures generated by two adversarially chosen key are statistically indistinguishable. Readers are referred to [15,48] for formal definitions.

C.2 Definition of Group Signatures

A group signature scheme $\mathcal{GS} = (\mathsf{GKey}, \mathsf{GSign}, \mathsf{GVerify}, \mathsf{GOpen})$ consists of four polynomial-time algorithm defined as follows.

$\mathsf{GKey}(1^\lambda, 1^N)$**.** On inputs the security parameter λ and the group size N, it returns a tuple $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$ where gpk is the group publik key, gmsk is the group manager's secret key, and $\mathbf{gsk}$ is an N-vector of keys with $\mathsf{gsk}[i]$ being a secret signing key for signer $i \in [0, N-1]$.

$\mathsf{GSign}(\mathsf{gpk}, \mathsf{gsk}[i], M)$**.** On inputs gpk, $\mathsf{gsk}[i]$ for some signer $i \in [0, N-1]$ and a message M, it outputs a group signature Σ of M under $\mathsf{gsk}[i]$.

$\mathsf{GVerify}(\mathsf{gpk}, M, \Sigma)$**.** On inputs gpk, M, Σ, the deterministic group verify algorithm outputs 1 if the signature is valid or 0 otherwise.

$\mathsf{GOpen}(\mathsf{gpk}, \mathsf{gmsk}, \Sigma, M)$**.** On inputs $\mathsf{gpk}, \mathsf{gmsk}, \Sigma, M$, the deterministic opening algorithm returns an identity i or $\perp$ to indicate failure.

Definition 8 (Correctness). *A group signature is correct if for all* $\lambda, N \in \mathbb{N}$*, all* $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk}) \leftarrow \mathsf{GKgen}(1^\lambda, 1^N)$*, for any* $i \in [0, N-1]$ *and any* $M \in \{0,1\}^*$*, the following conditions hold:*

$$\mathsf{GVerify}\big(\mathsf{gpk}, M, \mathsf{GSign}(\mathsf{gpk}, \mathsf{gsk}[i], M)\big) = 1 \text{ and}$$
$$\mathsf{GOpen}\big(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{GSign}(\mathsf{gpk}, \mathsf{gsk}[i], M), M\big) = i.$$

We say that Σ is a *true* signature of M if there exists $i \in [0, N-1]$ such that $\Sigma \in \{\mathsf{GSign}(\mathsf{gpk}, \mathsf{gsk}[i], M)\}$. The first equation requires that true signatures are always valid, and the second equation requires that the opening algorithm correctly recovers the identity of signer from a true signature.

A group signature is considered secure if it satisfies two key properties: full-anonymity (or CCA2-anonymity) and full-traceability. The full-anonymity requirement states that it is infeasible for any probabilistic polynomial-time (PPT) adversary to distinguish which of two signers of its choice signed a targeted message. This must hold even if the adversary has access to all group members' secret keys, can choose the message to be signed, and can query the opening oracle for any signature except the challenge one. The full-traceability property ensures that it is infeasible for any PPT adversary to output a valid group signature that either fails in the opening algorithm or that is traced to a user who is not in the coalition set. This must hold even if the adversary is able to corrupt the group manager. Readers are referred to [13] for formal definitions.

C.3 Definition of Fully Dynamic Attribute-Based Signatures

Let $\mathcal{X}$ be the universe of possible attributes and $\mathcal{P} = \{P : \mathcal{X} \to \{0,1\}\}$ be a policy family. We say an attribute $x \in \mathcal{X}$ satisfies a policy $P \in \mathcal{P}$ if $P(x) = 1$.

An FDABS scheme consists of the following polynomial-time algorithms.

$\mathsf{Setup}_{\mathsf{init}}(1^\lambda)$**:** Run by a trusted authority, this algorithm generates public parameter pp that specifies attribute space $\mathcal{X}$, a policy family $\mathcal{P}$, a time space $\mathcal{T}$, and a message space $\mathcal{M}$.

$\mathsf{Setup}_{\mathsf{auth}}(\mathsf{pp})$**:** This algorithm is run by an attribute-issuing authority. It outputs a key pair $(\mathsf{mpk}, \mathsf{msk})$ and initialize system information info_0 and a public registration table **reg**.

$\mathsf{AttrGen}(\mathsf{msk}, x, \mathsf{info}_{\tau_{\text{current}}}, \mathbf{reg})$**:** This algorithm is invoked by a user who wishes to join the system. It is run by the authority who will generate an attribute key (or a signing key) sk_x to the user. The authority will then add a new record to the table **reg**.

$\mathsf{Update}(\mathsf{msk}, \mathcal{S}, \mathsf{info}_{\tau_{\text{current}}}, \mathbf{reg})$**:** This algorithm is run by the authority who will advance the epoch and update system information. Given the inputs, the authority computes new system information $\mathsf{info}_{\tau_{\text{new}}}$ and may also update **reg**.

$\mathsf{Sign}(\mathsf{sk}_x, M, P, \mathsf{info}_\tau)$**:** This algorithm is run by the user who possesses the signing key sk_x. Given sk_x, a message $M \in \mathcal{M}$, a policy $P \in \mathcal{P}$, and info_τ, it returns a signature Σ.

$\mathsf{Verify}(M, P, \mathsf{info}_\tau, \Sigma)$**:** This algorithm is run by any verifier. Given the inputs, it outputs a bit indicating the validity of signature Σ on message M with respect to policy P and system information info_τ.

Informally speaking, an FDABS scheme is correct if any honestly generated signature is deemed valid. Security requirements of FDABS then consist of privacy and unforgeability. Privacy demands that signatures do not give away additional information on the attribute except that it satisfies the policy. Unforgeability requires that no colluding set of signers can create valid signatures with respect to a policy with which they are not supposed to sign. Formal definitions are given in [58].

C.4 Code-Based Ring Signature Scheme by Nguyen et al.

We recall the RS scheme proposed by Nguyen et al. [71] in the following. However, we do not include the details of how the signing algorithm RSign generates a proof Π_{ring}. This is where we differ from [71].

$\mathsf{RSetup}(1^\lambda)$. Let λ be the security parameter, choose $n = \mathcal{O}(\lambda), c = \mathcal{O}(1)$ and $m = 2 \cdot \frac{n}{c} \cdot 2^c$. Sample a uniformly random matrix $\mathbf{B} \xleftarrow{\$} \mathbb{F}_2^{n \times m}$, and return $\mathsf{pp} = \{n, c, m, \mathbf{B}\}$.

$\mathsf{RKgen}(\mathsf{pp})$. On input pp, choose an uniformly random vector $\mathbf{x} = (\mathbf{x}_0 \| \mathbf{x}_1) \xleftarrow{\$} \mathbb{F}_2^{2n}$, generate $\mathbf{d} = h_{\mathbf{B}}(\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{F}_2^n$, and output $(\mathsf{pk}, \mathsf{sk}) = (\mathbf{d}, \mathbf{x})$.

$\mathsf{RSign}(\mathsf{pp}, (\mathsf{sk}, M, R))$. It takes as inputs pp, sk, a message $M \in \{0,1\}^*$, and a ring $R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1})$, and outputs a ring signature Σ on M. Specifically, let $\mathsf{sk} = (\mathbf{x}_0 \| \mathbf{x}_1) \in \mathbb{F}_2^{2n}$ such that $\mathbf{d} = h_{\mathbf{B}}(\mathbf{x}_0, \mathbf{x}_1) \in R$. This algorithm then performs the following steps.

1. First, run the algorithm $\mathsf{TAcc}(R)$ to build a Merkle tree, obtaining an accumulated value $\mathbf{u} \in \mathbb{F}_2^n$.
2. Next, run the algorithm $\mathsf{TWitGen}(R, \mathbf{d})$ to obtain the corresponding witness
$$w = \big((j_1, \ldots, j_\ell)^\top, (\mathbf{w}_\ell, \ldots, \mathbf{w}_1)\big) \in \{0,1\}^\ell \times \big(\mathbb{F}_2^n\big)^\ell,$$
indicating that $\mathbf{d}$ is accumulated in the root $\mathbf{u}$.
3. Generate a non-interactive ZK proof Π_{ring} to show the possession of a valid pair $(\mathsf{pk}, \mathsf{sk}) = (\mathbf{d}, \mathbf{x})$ such that $\mathbf{d}$ is correctly accumulated in $\mathbf{u}$ in the Merkle tree.
4. Output $\Sigma = \Pi_{\mathsf{ring}}$.

$\mathsf{RVerify}(M, R, \Sigma)$. On input pp, a message M, a ring $R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1})$ and a signature Σ, the algorithm outputs $0/1$ as follows:

1. Build a Merkle tree based on the ring R, i.e., run the algorithm $\mathsf{TAcc}(R)$ to compute the root $\mathbf{u}$.
2. Verify the validity of the proof Π_{ring}. Return 1 if Π_{ring} is valid, and 0 otherwise.

Theorem 2 ([71]). *The above RS scheme is correct. If* 2-$\mathsf{RNSD}_{n,2n,c}$ *problem is hard and the supporting zero-knowledge protocol is witness indistinguishable, then the RS scheme provides unforgeability with respect to insider corruption and anonymity in the random oracle model.*

C.5 Code-Based Group Signature Scheme by Nguyen et al.

We now describe the code-based GS scheme proposed in [71] with one difference that we employ a variant of randomized McEliece encryption with regular noise within the scheme. In addition, we do not include the details of how the signing algorithm GSign generates a proof Π_{group}. In deed, this would be where we differ from [71].

$\mathsf{GKgen}(1^\lambda, 1^N)$. On input the parameters $1^\lambda, 1^N$, the algorithm samples a uniformly random matrix $\mathbf{B} \xleftarrow{\$} \mathbb{F}_2^{n\times m}$. Then it performs as follows to get the group public key, group manager's secret key and secret signing key for each group user.

1. For each $j \in [0, N-1]$, sample a random binary vector $\mathbf{x}_j = (\mathbf{x}_{j,0} \| \mathbf{x}_{j,1}) \xleftarrow{\$} \mathbb{F}_2^{2n}$ and compute $\mathbf{d}_i = h_{\mathbf{B}}(\mathbf{x}_{j,0}, \mathbf{x}_{j,1}) \in \mathbb{F}_2^n$. $\{\mathbf{d}_j\}_{j=0}^{N-1}$ should be pairwise distict, otherwise restart the process. Then define the set $R = (\mathbf{d}_1, \ldots, \mathbf{d}_{N-1})$.
2. Run algorithm $\mathsf{TAcc}(R)$ to build a Merkle tree based on R and the hash function $h_{\mathbf{B}}$, and obtain the accumulated value $\mathbf{u} \in \mathbb{F}_2^n$.
3. Let $\ell = \lceil \log N \rceil$. For each $j \in [0, N-1]$, let $(j_1, \ldots, j_\ell)^\top$ be the binary representation of j. Run the algorithm $\mathsf{TWitGen}(R, \mathbf{d}_j)$ to output a witness $w^{(j)}$ of $\mathbf{d}_j$ such that $\mathbf{d}_j$ is correctly accumulated in $\mathbf{u}$.

$$w^{(j)} = \big((j_1, \ldots, j_\ell)^\top \in \{0,1\}^\ell, (\mathbf{w}_\ell^{(j)}, \ldots, \mathbf{w}_1^{(j)}) \in (\mathbb{F}_2^n)^\ell\big).$$

   Then define $\mathsf{gsk}[j] = (\mathbf{x}_j, \mathbf{d}_j, w^{(j)})$.
4. Run $\mathsf{ME.KeyGen}(n_e, k_e, t_e)$ twice to obtain two key-pairs $(\mathsf{pk_{ME}}^{(1)} = \mathbf{G}_1 \in \mathbb{F}_2^{n_e \times k_e}, \mathsf{sk_{ME}}^{(1)})$ and $(\mathsf{pk_{ME}}^{(2)} = \mathbf{G}_2 \in \mathbb{F}_2^{n_e \times k_e}, \mathsf{sk_{ME}}^{(2)})$.
5. Output

$$\mathsf{gpk} := \{\mathbf{B}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2\}; \quad \mathsf{gmsk} := \mathsf{sk_{ME}}^{(1)}; \quad \mathbf{gsk} := (\mathsf{gsk}[0], \ldots, \mathsf{gsk}[\mathsf{N}-1]).$$

$\mathsf{GSign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$. On input $M \in \{0,1\}^*$ and the user's secret signing key $\mathsf{gsk}[j] = (\mathbf{x}_j, \mathbf{d}_j, w^{(j)})$, where $w^{(j)} = \big((j_1, \ldots, j_\ell)^\top, (\mathbf{w}_\ell^{(j)}, \ldots, \mathbf{w}_1^{(j)})\big)$, the user performs as follows:

1. Encrypt the identity $\mathsf{bin}(j) = (j_1, \ldots, j_\ell)^\top \in \{0,1\}^\ell$ twice using the variant of randomized McEliece encryption scheme. More precisely, for each $i \in \{1,2\}$, sample $\mathbf{r}_i \xleftarrow{\$} \mathbb{F}_2^{k_e - \ell}, \mathbf{e}'_i \xleftarrow{\$} \mathbb{F}_2^k$ and compute

$$\mathbf{c}_i = \mathbf{G}_i \cdot \begin{pmatrix} \mathbf{r}_i \\ \mathsf{bin}(j) \end{pmatrix} \oplus \mathsf{RE}(\mathbf{e}'_i) \in \mathbb{F}_2^{n_e}.$$

2. Generate a non-interactive ZK proof Π_{group} to show the possessions of a valid tuple $\tau = (\mathbf{x}_j, \mathbf{d}_j, w^{(j)}, \mathbf{r}_1, \mathbf{e}'_1, \mathbf{r}_2, \mathbf{e}'_2)$, where

$$\mathbf{x}_j = (\mathbf{x}_{j,0} \,\|\, \mathbf{x}_{j,1}) \text{ and } w^{(j)} = \big((j_1, \ldots, j_\ell)^\top, (\mathbf{w}_\ell^{(j)}, \ldots, \mathbf{w}_1^{(j)})\big),$$

such that:
   (a) $h_{\mathbf{B}}(\mathbf{x}_{j,0}, \mathbf{x}_{j,1}) = \mathbf{d}_j$ and $\mathsf{TVerify}(\mathbf{u}, \mathbf{d}_j, w^{(j)}) = 1$.
   (b) $\mathbf{c}_1$ and $\mathbf{c}_2$ are both correct encryptions of $\mathsf{bin}(j) = (j_1, \ldots, j_\ell)^\top$ with randomness $(\mathbf{r}_1, \mathbf{e}'_1) \in \mathbb{F}_2^{k_e - \ell} \times \mathbb{F}_2^k$ and $(\mathbf{r}_2, \mathbf{e}'_2) \in \mathbb{F}_2^{k_e - \ell} \times \mathbb{F}_2^k$, respectively.
3. Output the group signature $\Sigma = (\Pi_{\mathsf{group}}, \mathbf{c}_1, \mathbf{c}_2)$.

$\mathsf{GVerify}(\mathsf{gpk}, M, \Sigma)$. Given the pair (M, Σ) with $\Sigma = (\Pi_{\mathsf{group}}, \mathbf{c}_1, \mathbf{c}_2)$, this algorithm simply verifies the validity of Π_{group}. Return 1 if Π_{group} is valid, and 0 otherwise.

$\mathsf{GOpen}(\mathsf{gpk}, \mathsf{gmsk}, \Sigma, M)$. On input $\mathsf{gmsk} = \mathsf{sk}_{\mathsf{ME}}{}^{(1)}$ and a group signature Σ on message M, this algorithm proceeds as follows:

1. Return $\perp$ if $\mathsf{GVerify}(\mathsf{gpk}, M, \Sigma) = 0$.
2. Run $\mathsf{ME.Dec}(\mathsf{sk}_{\mathsf{ME}}{}^{(1)}, \mathbf{c}_1)$ to decrypt $\mathbf{c}_1$. If decryption fails, return $\perp$. Otherwise, let $\mathbf{p} = (j'_1, \ldots, j'_\ell)^\top \in \{0,1\}^\ell$ be the result of decryption.
3. Output the index $j \in [0, N-1]$ that has binary representation $(j'_1, \ldots, j'_\ell)^\top$.

Theorem 3 ([71]). *The above GS scheme is correct. If* 2-RNSD *problem is hard, the variant of McEliece cryptosystem with regular noise is CPA-secure, and the supporting ZK protocol is simulation-sound, then the GS satisfies full traceability and full anonymity in the random oracle model.*

C.6 Code-Based Fully Dynamic Attribute-Based Signature Scheme by Ling et al.

In this section, we describe a variant of the FDABS scheme by Ling et al. [58] by degrading their QROM security to ROM security. Except with the difference, the following are taken verbatim from [58].

$\mathsf{Setup}_{\mathsf{init}}(1^\lambda)$**:** Given the security parameter 1^λ, this algorithm performs the following steps.

- Let $L = \mathsf{poly}(\lambda)$ be a positive integer. It then specifies the time space $\mathcal{T} = \{0, 1, 2, 3, \ldots\}$, the message space $\mathcal{M} = \{0,1\}^*$, the attribute space $\mathcal{X} = \{0,1\}^L$, and the policy family $\mathcal{P} = \{P : \mathcal{X} \to \{0,1\}\}$ that consists of all possible polynomial-size Boolean circuits with L-bit input.

- Specify an integer $\ell = \ell(\lambda)$ that determines the maximum number $N = 2^\ell = \mathsf{poly}(\lambda)$ of potential attributes.
- Choose $n = \mathcal{O}(\lambda)$, $c = \mathcal{O}(1)$ such that c divides both L and n and set $m = 2 \cdot 2^c \cdot \frac{n}{c}$.
- Sample a random matrix $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_2^{n \times m}$ that specifies a hash function $h_{\mathbf{B}}$ as in Definition 3.
- Choose $k \geq n + 2\lambda + \mathcal{O}(1)$ such that c divides k. Let $m_0 = 2^c \cdot L/c$ and $m_1 = 2^c \cdot k/c$. Sample $\mathbf{C}_0 \xleftarrow{\$} \mathbb{Z}_2^{n \times m_0}$ and $\mathbf{C}_1 \xleftarrow{\$} \mathbb{Z}_2^{n \times m_1}$ that specifies a statistically hiding and computationally binding commitment scheme.
- Let $\mathsf{COM} : \{0,1\}^* \times \{0,1\}^k \to \{0,1\}^n$ be the extended commitment scheme as described in Sextion 2.3.
- Pick a secure hash function $\mathcal{H}_{\mathsf{FS}} : \{0,1\}^* \to 1,2,3^\kappa$, where $\kappa = \mathcal{O}(\lambda)$ to be modelled as a random oracle in the Fiat-Shamir heuristic [45].

It then outputs public parameter

$$\mathsf{pp} = \{L, \mathcal{T}, \mathcal{M}, \mathcal{X}, \mathcal{P}, \ell, N, n, c, m, k, m_0, m_1, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathsf{COM}, \mathcal{H}_{\mathsf{FS}}\}.$$

$\mathsf{Setup}_{\mathsf{auth}}(\mathsf{pp})$**:** This algorithm is run by the attribute-issuing authority. On input parameter pp, it runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{AuthGen}(\mathsf{pp})$. In addition, it initializes the following.

- A registration table $\mathbf{reg} := (\mathbf{reg}[0], \ldots, \mathbf{reg}[N-1])$ so that $\mathbf{reg}[i][1] = \mathbf{0}^n$, $\mathbf{reg}[i][2] = -1$, and $\mathbf{reg}[i][3] = -1$ for all $i \in [0, N-1]$. Looking ahead, $\mathbf{reg}[i][1]$ will store a (non-zero) commitment of an attribute while $\mathbf{reg}[i][2]$ and $\mathbf{reg}[i][3]$ represent the epochs an attribute is enrolled in and removed from the system, respectively.
- A Merkle tree $\mathcal{MT}$ built on top of $\mathbf{reg}[0][1], \mathbf{reg}[1][1], \ldots, \mathbf{reg}[N-1][1]$. We remark that this $\mathcal{MT}$ is all-zero at this stage. However, it will be eventually updated either when an attribute is enrolled in or revoked from the system.
- A counter of enrolled attributes $j := 0$, initial time epoch $\tau = 0$, and initial system information $\mathsf{info}_0 = \emptyset$.

The authority will then publish public key mpk and broadcast $\mathbf{reg}$ and info_0 while keeping $\mathcal{MT}$ and j for itself. We assume that both $\mathbf{reg}$ and info are visible to everyone but only editable by a party who owns msk. It is further required that one can efficiently verify the well-formedness of $\mathbf{reg}$ and info.

$\mathsf{AttrGen}(\mathsf{msk}, \mathbf{x}, \mathsf{info}_{\tau_{\text{current}}}, \mathbf{reg})$**:** When a user requests an attribute key for his provided attribute $\mathbf{x} \in \{0,1\}^L$ at current epoch τ_{current}, the authority executes this algorithm and proceeds as follows.

1. Issue an identifier for this attribute $\mathbf{x}$ as the binary representation of j, denoted as $\mathsf{bin}(j) \in \{0,1\}^\ell$.
2. Sample randomness $\mathbf{r} \xleftarrow{\$} \{0,1\}^k$ and compute a commitment of $\mathbf{x}$ as $\mathbf{d} = \mathbf{C}_0 \cdot \mathsf{RE}(\mathbf{x}) \oplus \mathbf{C}_1 \cdot \mathsf{RE}(\mathbf{r})$. Repeat the process until the weight of $\mathbf{d}$ is odd. Return $\mathsf{sk}_{\mathbf{x}} = (\mathbf{x}, \mathbf{r}, \mathsf{bin}(j))$ to the user. From now on, we write $\mathsf{sk}_{\mathbf{x}_j} = (\mathbf{x}_j, \mathbf{r}_j, \mathsf{bin}(j))$ to distinguish signing keys of different attributes.

3. Update $\mathcal{MT}$ by running the algorithm $\mathsf{TUpdate}_{\mathbf{B}}(\mathsf{bin}(j), \mathbf{d})$, register the attribute to $\mathbf{reg}$ as $\mathbf{reg}[j][1] = \mathbf{d}$, $\mathbf{reg}[j][2] = \tau_{\text{current}}$, and increase the counter j to $j + 1$.

$\mathsf{Update}(\mathsf{msk}, \mathcal{S}, \mathsf{info}_{\tau_{\text{current}}}, \mathbf{reg})$**:** This algorithm is run by the authority to update the system and advance the epoch. Let $\mathcal{S} = \{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_r}\}$ contain attributes to be revoked. If there exists $t \in [1, r]$ so that $\mathsf{IsActive}(\mathbf{x}_{i_t}, \mathsf{info}_{\tau_{\text{current}}}) = 0$, this algorithm aborts. Otherwise it performs the following steps.

1. For each $t \in [1, r]$, run $\mathsf{TUpdate}_{\mathbf{B}}(\mathsf{bin}(i_t), \mathbf{0}^n)$ to update $\mathcal{MT}$ and set $\mathbf{reg}[i_t][3] = \tau_{\text{new}}$.
2. Note that all the zero leaves in updated $\mathcal{MT}$ are associated with either revoked attributes or potential attributes that have not registered to the system yet. In other words, only active attributes have their odd-weight commitments, denoted as $\{\mathbf{d}_j\}$, accumulated in the root $\mathbf{u}_{\tau_{\text{new}}}$ of the updated tree.
   For each j, let $w^{(j)} \in \mathbb{Z}_2^\ell \times (\mathbb{Z}_2^n)^\ell$ be the witness for the fact that $\mathbf{d}_j$ is accumulated in $\mathbf{u}_{\tau_{\text{new}}}$. (This can be obtained by running algorithm $\mathsf{TWitGen}_{\mathbf{B}}$ as described in Section 2.4). The authority then announces the updated system information as

$$\mathsf{info}_{\tau_{\text{new}}} = (\mathbf{u}_{\tau_{\text{new}}}, \{w^{(j)}\}_j).$$

As commented by the authors, it is unnecessary for a signer or a verifier to download $\mathsf{info}_{\tau_{\text{new}}}$ as a whole. In fact, a signer with an active attribute only needs to download its corresponding witness $w^{(j)}$ of $\mathcal{O}(\lambda \cdot \ell)$ bits once so as to sign messages at time τ_{new}. Meanwhile, it suffices for a verifier to download $\mathbf{u}_{\tau_{\text{new}}}$ of $\mathcal{O}(\lambda)$ bits to verify all signatures associated with τ_{new}.

$\mathsf{Sign}(\mathsf{sk}_{\mathbf{x}_j}, M, P, \mathsf{info}_\tau)$**:** This algorithm is run by a user possessing an attribute key $\mathsf{sk}_{\mathbf{x}_j} = (\mathbf{x}_j, \mathbf{r}_j, \mathsf{bin}(j))$ who wishes to sign a message with respect to a policy P. It aborts if $P(\mathbf{x}_j) = 0$ or info_τ does not include a witness containing $\mathsf{bin}(j)$. Otherwise, it proceeds as below.

1. Download $\mathbf{u}_\tau$ and the witness $w^{(j)} = (\mathsf{bin}(j), (\mathbf{w}_\ell, \ldots, \mathbf{w}_1))$ from info_τ.
2. Generate a proof to show the possession of tuple

$$\xi = (\mathbf{d}_j, \mathbf{x}_j, \mathbf{r}_j, \mathsf{bin}(j), \mathbf{w}_\ell, \ldots, \mathbf{w}_1) \tag{28}$$

   such that

   (a) $\mathbf{d}_j$ is correctly accumulated in the root $\mathbf{u}_\tau$, i.e.,

$$\mathsf{TVerify}_{\mathbf{B}}(\mathbf{u}_\tau, \mathbf{d}_j, \mathsf{bin}(j), (\mathbf{w}_\ell, \ldots, \mathbf{w}_1)) = 1.$$

   (b) $\mathbf{d}_j$ is an odd-weight commitment of $\mathbf{x}_j$, i.e.,

$$\mathbf{d}_j = \mathbf{C}_0 \cdot \mathsf{RE}(\mathbf{x}_j) \oplus \mathbf{C}_1 \cdot \mathsf{RE}(\mathbf{r}_j) \text{ and } wt(\mathbf{d}_j) = 1 \bmod 2.$$

   (c) The attribute $\mathbf{x}_j$ satisfies the claimed policy P. In other words, $P(\mathbf{x}_j) = 1$.

3. Let the resultant proof be Π. Return signature as $\Sigma = \Pi$.

$\mathsf{Verify}(M, P, \mathsf{info}_\tau, \Sigma)$**:** This algorithm checks the validity of the message signature pair (M, Σ) with respect to the policy P and time epoch τ. It returns 1 if Π is valid, and 0 otherwise.

Theorem 4 ([58]). *The above FDABS scheme is correct. If the supporting ZK protocol is ZK and sound, the Merkle tree accumulator is secure, and the commitment scheme used to commit the attributes is statistically hiding and computationally binding, then the FDABS scheme satisfies privacy and unforgeability.*