Structure-Preserving Compressing Primitives: Vector Commitments, Accumulators and Applications

Stephan Krenn[1], Omid Mir[1], and Daniel Slamanig[2]

[1] Austrian Institute of Technology
{omid.mir, stephan.krenn}@ait.ac.at
[2] Universität der Bundeswehr München, Germany
daniel.slamanig@unibw.de

Abstract. Compressing primitives such as accumulators and vector commitments, allow to represent large data sets with some compact, ideally constant-sized value. Moreover, they support operations like proving membership or non-membership with minimal, ideally also constant-sized, storage and communication overhead. In recent years, these primitives have found numerous practical applications, with many constructions based on various hardness assumptions. So far, however, it has been elusive to construct these primitives in a strictly structure-preserving setting, i.e., in a bilinear group in a way that messages, commitments and openings are all elements of the two source groups. Interestingly, backed by existing impossibility results, not even conventional commitments with such constraints are known in this setting.
In this paper we investigate whether strictly structure-preserving compressing primitives can be realized. We close this gap by presenting the first strictly structure-preserving commitment that is shrinking (and in particular constant-size). We circumvent existing impossibility results by employing a more structured message space, i.e., a variant of the Diffie-Hellman message space. Our main results are constructions of structure-preserving vector commitments (SPVC) as well as accumulators. We first discuss generic constructions and then present concrete constructions under the Diffie-Hellman Exponent assumption. To demonstrate the usefulness of our constructions, we present various applications. Most notable, we present the *first* entirely practical constant-size ring signature scheme in bilinear groups (i.e., the discrete logarithm setting). Concretely, using the popular BLS12-381 pairing-friendly curve, our ring signatures achieve a size of roughly 6500 bits.

1 Introduction

Compressing primitives like accumulators, vector commitments or key-value commitments (key-value maps) are cryptographic techniques to efficiently represent and manage large datasets in a space-efficient form. They allow to represent large data sets with a compact, ideally constant-sized, value and support operations like proving membership or non-membership with minimal storage and communication overhead (ideally constant-size witnesses). These primitives have many interesting applications for blockchain privacy, e.g., Zcash[3], and scalability, e.g., use of Verkle trees[4], stateless clients for blockchains[5], data availability sampling [57]) or batching [25]. Moreover, they are extensively used in privacy-preserving systems, such as anonymous authentication primitives, e.g., ring signatures [40] or revocation in anonymous credentials [8, 11, 36], or data outsourcing [17, 27, 81].

In this paper, we focus mainly on vector commitments [27] and accumulators [13, 18]. We recall that a vector commitment (VC) scheme enables a user to commit to a vector $\mathbf{m}$, with the ability to later open the commitment at specific positions without being able to cheat (position binding). Moreover, they might also support to update values committed at certain position. Crucially, both the commitment size and the size of the opening must remain succinct, ideally of constant size. In recent years we have seen significant advancements and growing interest in the development and applications of vector commitments. Beginning with the foundational Merkle tree [70], which leverages collision-resistant hash functions, the field has expanded to include a diverse array of algebraic constructions. These include schemes based on pairing-based assumptions [27, 52, 59, 60, 65, 66, 82] as well as those relying on assumptions over groups of unknown order, such as RSA groups or class groups [24, 27, 60]

[3] https://z.cash
[4] https://vitalik.eth.limo/general/2021/06/18/verkle.html
[5] https://ethresear.ch/t/the-stateless-client-concept/172

and post-quantum constructions based on lattices [10,62,78,83]. For an extensive overview of schemes, see [76]. As we have already mentioned above, vector commitments have been widely utilized across various applications. Moreover, generalizations of vector commitments and in particular polynomial commitments [59] and functional commitments [65], have become foundational components in many recent constructions of succinct non-interactive arguments of knowledge (SNARKs).

However, it is not clear how to apply this result to the SP setting. Moreover, when using a vector commitment in this way, it often reveals the position of elements within the set, which may expose sensitive information in certain contexts, e.g., when using them as a building block for privacy-preserving primitives such as ring signatures. Therefore, we are interested in an approach where the witness does not disclose any information about the elements.

We also recall that a (static) accumulator [13,18] is another type of compressing primitive, which allows to represent a set of elements $X = \{x_1, \ldots, x_n\}$ in the form of a succinct accumulator value acc_X. For each element $x_i \in X$, a witness wit_{x_i} can be efficiently computed to certify its membership in the set, while ensuring that it is computationally infeasible to forge witnesses for non-members $y \notin X$ (collision resistance). Universal accumulators in addition provide the functionality of non-membership witnesses for values $y \notin X$ and dynamic accumulators supporting efficient additions and deletions from the accumulated set. It is worth remarking that Catalano and Fiore have shown that any vector commitment can be used to construct universal dynamic accumulators [27]. Unfortunately, it is unclear whether this approach can be immediately applied to the SP setting or effectively hide the index, a property often required in privacy-preserving applications. Like vector commitments, accumulators have found wide-ranging applications and actually evolved into a foundational component in various advanced cryptographic constructs. They are integral to revocation in anonymous credentials [8, 11, 36], membership revocation for group signatures [22] and the construction of ring signatures [38,40,61].

Over time, cryptographic accumulators have evolved through various instantiations. Initially based on the RSA assumption [13], early schemes were later expanded upon [22, 40]. Nguyen introduced pairing-based accumulators [74], sparking further advancements in this area [11, 12, 21, 35, 50]. More recent developments include lattice-based accumulators [58, 62, 77].

Structure-Preserving Compressing Primitives. While, as outlined above, numerous constructions exist for both vector commitments and accumulators, a significant challenge remains in developing such schemes that are structure-preserving (SP). We recall that a cryptographic scheme is called structure preserving [3] if all public inputs and outputs consist of elements of the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ and functional correctness can be verified only by testing group membership, computing group operations, and evaluating pairing product equations (PPEs) [1] of the form $\prod_i \prod_j e(A_i, \hat{B}_j)^{c_{i,j}} = 1$, where $A_i \in \mathbb{G}_1$ and $\hat{B}_j \in \mathbb{G}_2$ are group elements and $c_{i,j} \in \mathbb{Z}_p$ are constants. This domain is very rich in its constructions and allows for a modular design of (advanced) cryptographic primitives, e.g., structure-preserving signatures (SPS) [3] (cf. [49] for a recent overview on the large body of works), threshold SPS [33, 73], blind signatures [3, 46], group signatures [3, 64], traceable signatures [2], homomorphic signatures [63] or delegatable anonymous credentials [34, 45, 72]. Such constructions are highly desirable because they maintain the algebraic structure of the committed values and proofs, ensuring compatibility with a broader range of cryptographic protocols and in particular the Groth-Sahai [56] and related proof systems. Moreover, especially for compressing primitives they are of concrete practical interest beyond modular protocol design, as we will discuss in this paper.

Unfortunately, there are known impossibility results of Abe et al. [6] establishing that strictly structure-preserving commitments to source-group elements cannot be smaller than the input message size and thus scale linearly with the number of group elements. In order to be compressing though, for conventional commitments, there are some approaches that relax the SP setting by allowing elements from the target group $\mathbb{G}_T$ in the commitment [5,6] (see Sec.2.1). Another approach by Abe et al. [7] is to construct SP commitments that have a relaxed notion of binding, where the message space and the verification space differ (e.g., being $\mathbb{Z}_p$ and $\mathbb{G}_1$ respectively). Nevertheless, these are not strictly structure-preserving commitments and in particular do not allow to commit to commitments, a features that is interesting for practical applications. When it comes to accumulators, to the best of our knowledge, the approach that conceptually comes closest is that of determinantal accumulators [67]. While they are in spirit of SP primitives and their combination with Groth-Sahai proofs [56], the focus of determinantal primitives is on designing primitives that can be modularity combined with algebraic NIZKs in the vein of Couteau and Hartmann [31] and Couteau et al. [32]. Consequently, they

are not structure-preserving. For vector commitments, there is an impossibility result for algebraic vector commitments in pairing-free groups [28], but this does not rule out the existence of structure-preserving ones. While [28] notes that the impossibility of strictly SP commitments [5, 6] rules out constructing succinct vector commitments in this structure-preserving setting, we are not aware of any work trying to bypass such an impossibility and thus this state of affairs is not satisfactory. This leads us to the following question:

Can we design (vector) commitments and accumulators that retains algebraic structure, i.e., being strictly structure-preserving, while being succinct?

Addressing this gap is crucial for advancing the field. Our aim is to close this gap.

1.1 Our Results

Our contributions can be summarized as follows:

- We present the first construction of group-to-group commitments that are shrinking and strictly structure-preserving, i.e., messages, commitments and openings are all elements in the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$. We obtain this by bypassing known impossibility results due to Abe et al. [5,6] as we require a more structured message space and in particular a Diffie-Hellman (DH) message space [3, 44]. Such a message space has recently shown to be relevant for various applications [33, 71].
- We present the first structure-preserving vector commitments (SPVC's). First we present a (probably folklore) construction of weak-binding WSPVC from any EUF-CMA secure SPS scheme. Then, we turn to SPVC's providing the conventional notion of position-binding and present a construction of SPVC under the Diffie-Hellman Exponent (q-DHE) assumption with a message space that is defined with respect to some global parameters (inspired by the recent work by Griffy et al. [53] and q-DHE VC schemes [52, 66]).
- We present the first constructions of structure-preserving accumulators SPA. First we discuss how our weak-binding WSPVC naturally yields an accumulator. Second, we construct a randomizable SP accumulator starting from the q-DHE based SPVC but for another and in particular DH-type message space. Here we aim to achieve index hiding, i.e., it should be possible to give out a witness and accumulated value such that they cannot be linked back to the accumulated value. This is accomplished through a randomization technique that ensures the randomized accumulated values do not reveal anything about the index of the original accumulated value and DH-type messages.
- Finally, we outline some applications that showcase the benefits of our schemes. Most notable, we present the *first* entirely practical constant-size ring signature scheme in bilinear groups (i.e., the discrete logarithm setting) and prove its security in the strongest model of Bender et al. [19]. It is inspired by the key-homomorphic signature based ring signature construction in [39] but uses an accumulator for the membership proof. We base our construction on our randomized SP accumulator and a variant of BLS signatures that we call accumulator-compatible BLS, which ensures that BLS public keys are compatible with the message space of our accumulator. Concretely, using the popular BLS12-381 pairing-friendly curve, ring signatures are of size roughly 6500 bits. Moreover, we discuss the applications of our SPVC to succinct data availability sampling and algebraic Verkle trees.

2 Preliminaries

Notations. We use $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \leftarrow \mathsf{BGSetup}(1^\lambda)$ to denote a bilinear group generator for asymmetric type-3 bilinear groups, where p is a prime of bit length λ. We use $[\ell]$ to denote the set $\{1, 2, \ldots, \ell\}$. When drawing multiple values from a set, we may omit notation for products of sets, e.g. $(x, y) \in \mathbb{Z}_p$ is the same as $(x, y) \in (\mathbb{Z}_p)^2$. For a map from the set Z to the set S, $m : Z \to S$, we will denote $m[i] \in S$ as the output of the map in S with input $i \in Z$. We use bold font to denote a vector (e.g. $\mathbf{V}$). For brevity, we will sometimes denote the elements in a vector as $\mathbf{V} = (V_i)_{i\in[\ell]} = (V_1, \ldots, V_\ell)$. Given a finite set S, we denote by $x \leftarrow S$ or $x \leftarrow\!\!\$\ S$ the sampling of an element uniformly at random from S. For an algorithm A, let $y \leftarrow A(x)$ be the process of running A on input x with access to uniformly random coins and assigning the result to y. With A^B we denote that A has oracle

access to B. We assume all algorithms are polynomial-time (PPT) unless otherwise specified and public parameters are an implicit input to all algorithms in a scheme. We use $[a, b]$ to denote the range $\{a, a+1, \ldots, b\}$. Additionally, $|N|$ represents the length of N. By the symbol $\approx$, we denote indistinguishability between two distributions $D_1 \approx D_2$.

Camenisch and Stadler Notation. We use the common notation due to Camenisch and Stadler [23] for NIZK as follows:

$$\mathsf{NIZK}\left\{(\alpha, \beta) : y = P^{\alpha} \wedge z = P^{\beta} \cdot h^{\alpha}\right\},$$

which denotes an non-interactive proof of knowledge of discrete logarithms (α, β) (the witness) satisfying the right-hand side statement about the public values y, P, z, h.

Diffie-Hellman Message Space. Over an asymmetric bilinear group, a pair $(M, \hat{N}) \in \mathbb{G}_1 \times \mathbb{G}_2$ is called a Diffie-Hellman (DH) message $\mathcal{M}_{\mathsf{DH}}$ [4, 44] if there exists $m \in \mathbb{Z}_p$ s.t. $M = P^m$ and $\hat{N} = \hat{P}^m$. One can efficiently verify whether $(M, \hat{N}) \in \mathcal{M}_{\mathsf{DH}}$ by checking $e(M, \hat{P}) = e(P, \hat{N})$. More formally, the message space are elements of the subgroup of $\mathbb{G}_1 \times \mathbb{G}_2$ defined as the image of the map $\psi' : \mathbb{Z}_p \to \mathbb{G}_1 \times \mathbb{G}_2$ as: $\psi' : x \mapsto (P^x, \hat{P}^x)$. One can easily extend the message space to a vector of Diffie-Hellman pairs $(\mathbf{M}, \hat{\mathbf{N}}) = (M_1, \ldots, M_n, \hat{N}_1, \ldots, \hat{N}_n)$ s.t. for all $i \in [n]$, $(M_i, \hat{N}_i) = (P^{m_i}, \hat{P}^{m_i}) \in \mathcal{M}_{\mathsf{DH}}$ for $m_i \in \mathbb{Z}_p$.

We note that related notations exist, such as equivalence classes [47]. Given the vector $\mathbf{M}$, the message $\mathbf{M}$ represents an equivalence class of all scaled messages $\mathbf{M}^{\mu}$. This allows for randomizing the message vector by switching between $\mathbf{M}$ and $\mathbf{M}^{\mu}$ for any non-zero μ. This concept extends to Diffie-Hellman message spaces, including Indexed DH [33] and tag-based DH message spaces [71], where randomized DH message vectors retain their validity with respect to their tags. Although latter variants are not directly related to our primitives, it provides useful context.

2.1 Structure-Preserving Commitment

A structure-preserving commitment scheme was proposed by Abe et al. [4]. The public key consists of $n+1$ group elements $(H, U_1, \ldots, U_n)$ from $\mathbb{G}_1$. To commit to $(\hat{M}_1, \ldots, \hat{M}_n) \in \mathbb{G}_2^n$, a random element $\hat{R} \in \mathbb{G}_2$ is selected, and the commitment is computed as $C = e(H, \hat{R}) \prod_{i=1}^{n} e(U_i, \hat{M}_i)$. This commitment scheme is computationally binding under the DBP assumption. Moreover, it is both a length-reducing (constant-size) scheme and a homomorphic trapdoor commitment.

2.2 Zero-Knowledge Proofs of Knowledge

We define zero-knowledge proofs of knowledge (ZKPOK) and discuss non-interactive versions thereof (NIZK).

ZKPoK. Let $L_{\mathbb{R}} = \{x \mid \exists\, w : (x, w) \in \mathbb{R}\} \subseteq \{0,1\}^*$ be a formal language, where $\mathbb{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is a binary, polynomial-time (witness) relation. For such a relation, the membership of $x \in L_{\mathbb{R}}$ can be decided in polynomial time (in $|x|$) when given a witness w of length polynomial in $|x|$ certifying $(x, w) \in \mathbb{R}$. We assume an interactive protocol $(\mathcal{P}, \mathcal{V})$ between a prover $\mathcal{P}$ and a PPT verifier $\mathcal{V}$ and denote the outcome of the protocol as $(\cdot, b) \leftarrow \big(\mathcal{P}(\cdot,\cdot), \mathcal{V}(\cdot)\big)$ where $b = 0$ indicates that $\mathcal{V}$ rejects and $b = 1$ that it accepts the conversation with $\mathcal{P}$. We require the following properties completeness, zero knowledge (ZK) and soundness. For the formal definitions, see e.g. [51].

Non-Interactive Zero-Knowedge Proofs (of Knowledge). One can use the Fiat-Shamir heuristic to transform any Sigma protocol into a non-interactive zero-knowledge proof of knowledge (NIZK). Whenever one requires multiple-extractions in a security proof, a standard measure is to opt for interactive ZKPOK. We however note that when willing to pay some extra costs, one could instead use straight-line extractable NIZK, e.g., obtained via Fischlin's transformation [42].

2.3 Assumptions

Definition 1 (The Diffie-Hellman Exponent (q-DHE) [21]). *Let $\mathbb{G}$ be a finite cyclic group of order p, P be a generator of $\mathbb{G}$. The q-DHE problem is, given a tuple of elements $(P, P_1, \ldots, P_q, P_{q+2}, \ldots, P_{2q})$ such that $P_i = P^{\alpha^i}$ for $i = 1, \ldots, q, q+2, \ldots, 2q$ and where $\alpha \leftarrow\!\!\$\ \mathbb{Z}_p^*$, to compute the missing group element $P_{q+1} = P^{\alpha^{q+1}}$ in the sequence.*

As shown in [52,66], q-DHE can be modified so as to work in asymmetric pairing configurations i.e., given $\{P^{\alpha^1}, \ldots, P^{\alpha^q}, P^{\alpha^{q+2}}, \ldots, P^{\alpha^{2q}}; \hat{P}^{\alpha^1}, \ldots, \hat{P}^{\alpha^q}; e(P,\hat{P})^{\alpha^{q+1}}\}$ it is hard to find $P^{\alpha^{q+1}}$.

Definition 2 ((Double Pairing Assumption (DBP)) [4]). *We say the double pairing assumption holds relative to* BG *if for any probabilistic polynomial-time algorithm* $\mathcal{A}$

$$\Pr\left[\begin{array}{c}\mathsf{BG} \leftarrow \mathsf{BGSetup}(1^\lambda); h_z \leftarrow\!\!\$\ \mathbb{G}_1^* \\ (\hat{Z},\hat{R}) \leftarrow \mathcal{A}(\mathsf{BG}, h_z) : (\hat{Z},\hat{R}) \in \mathbb{G}_2^* \text{ and } 1 = e(h_z,\hat{Z})e(P,\hat{R})\end{array}\right] \leq \epsilon(\lambda)$$

This assumption follows from the decisional Diffie-Hellman (DDH) assumption in $\mathbb{G}_1$. By swapping $\mathbb{G}_1$ and $\mathbb{G}_2$ (assuming DDH in $\mathbb{G}_2$), we obtain a dual assumption. Hence, if DDH holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$, DBP holds in both groups.

2.4 Vector Commitments

We give a more formal definition for Vector Commitments (VC) [27] as follows:

Definition 3 (Vector Commitment). *A vector commitment is a tuple of algorithms defined as follows:*

- $\mathsf{KeyGen}(1^\lambda, q)$: Given the security parameter λ and the size q of the committed vector, the key generation outputs some public parameters pp (which implicitly define the message space $\mathcal{M}$ and are input to all algorithms).
- $\mathsf{Commit}(m_1, \ldots, m_q)$: On input a vector of q messages $(m_1, \ldots, m_q) \in \mathcal{M}$ and the public parameters pp, the committing algorithm outputs a commitment com and auxiliary information aux.
- $\mathsf{Open}(m, i, \mathsf{aux})$: This algorithm is run by the committer to produce a proof π_i that m is the i-th committed message. In particular, notice that in the case when some updates have occurred, the auxiliary information aux can include the update information produced by these updates.
- $\mathsf{Verify}(\mathsf{com}, m, i, \pi_i)$: The verification algorithm accepts (i.e., it outputs 1) only if π_i is a valid proof that com was created for a vector $(m_1, \ldots, m_q)$ such that $m = m_i$.

In addition, some applications require an update property (to update the commitment and the corresponding openings), which is defined using two additional (and optional) algorithms:

- $\mathsf{Update}(\mathsf{com}, m, m', i)$: This algorithm is run by the committer who produced com and wants to update it by changing the i-th message to m'. Takes as input the old message m, the new message m', and the position i. Outputs a new commitment com' together with update information U.
- $\mathsf{ProofUpdate}(\mathsf{com}, \pi_j, m', i, U)$: This algorithm is run by any user who holds a proof π_j for the message at position j with respect to com. It allows the user to compute an updated proof π'_j (and an updated commitment com'), such that π'_j will be valid with respect to com', where m' is the new message at position i. The value U contains the update information needed to compute these updated values.

VC should satisfy the property of position binding as follows:

Position Binding: This property ensures that a PPT adversary (with knowledge of pp) cannot produce two proofs for the same position in a fixed commitment com that open to different values. There are two flavors of this property:

- **Weak Position Binding:** Holds only for honestly generated commitments.
- **Position Binding:** Holds even for adversarially generated commitments.

Weak binding suffices for stateless validation (e.g., in Byzantine agreement on updates [76]) and can be easily achieved using accumulators. Strong binding is essential in adversarial scenarios, like transparency logs, where commitments are generated by log servers.

Definition 4 (Position Binding [27]). *A* VC *satisfies position binding if,* $\forall i = 1, \ldots, q$*, and for every PPT adversary* $\mathcal{A}$*, the following probability (taken over all honestly generated* pp*) is at most negligible:*

$$\Pr\left[\begin{array}{c}\mathsf{Verify}(\mathsf{com}, m, i, \pi) = 1 \wedge \\ \mathsf{Verify}(\mathsf{com}, m', i, \pi') = 1 \wedge \\ m \neq m'\end{array}\middle|\ ((\mathsf{com}, \mathsf{aux}), m, m', i, \pi, \pi') \leftarrow \mathcal{A}(\mathsf{pp})\right] \leq \epsilon(\lambda)$$

If we relax the above definition to hold only for honestly generated commitments com, we obtain the weak position binding notion.

Definition 5 (Weak Position Binding [26, 52]). *A* VC *satisfies Weak position binding if* $\forall i = 1, \ldots, q$ *and for every PPT adversary* $\mathcal{A}$*, the following probability (which is taken over all honestly generated parameters) is at most negligible:*

$$\Pr\left[\begin{array}{c} \mathsf{Verify}(\mathsf{com}, m, i, \pi) = 1 \wedge \mathsf{Verify}(\mathsf{com}, m', i, \pi') = 1 \\ \wedge \quad m \neq m', (\mathsf{pp}) \leftarrow \mathsf{KeyGen}(1^\lambda), (\mathsf{com}, \mathsf{aux}) \leftarrow \mathsf{Commit}(\mathbf{m}) \\ (i, m, \pi, m', \pi') \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{com}) \end{array}\right] \leq \epsilon(\lambda).$$

Conciseness: This property requires that both the commitment and the opening for a position i are of constant size, independent of the vectors length.

Hiding. VC can also be required to be hiding, meaning that one should not be able to distinguish whether a commitment was created to a vector $(m_1, \ldots, m_q)$ or to $(m'_1, \ldots, m'_q)$, even after seeing some proofs. However, hiding is not a crucial property in the realization of VC and typically not considered.

2.5 Accumulators

We provide a formal a definition of static accumulators based on the definition given by Derler et al. in [37].

Definition 6 (Static Accumulator). *A static accumulator is a tuple of algorithms defined as follows:*

- $\mathsf{Setup}(1^\lambda, q)$: Take a security parameter λ and a parameter q. If $q \neq \infty$, then q is an upper bound on the number of elements to be accumulated. Return a key pair $(\mathsf{sk}_{acc}, \mathsf{pk}_{acc})$, where $\mathsf{sk}_{acc} = \emptyset$ if no trapdoor exists.
- $\mathsf{Eval}((\mathsf{sk}_{acc}, \mathsf{pk}_{acc}), \mathcal{X})$: This (probabilistic) algorithm takes a key pair $(\mathsf{sk}_{acc}, \mathsf{pk}_{acc})$ and a set $\mathcal{X}$ to be accumulated and returns an accumulator $\mathsf{Acc}_\mathcal{X}$ together with some auxiliary information aux.
- $\mathsf{WitCreate}((\mathsf{sk}_{acc}, \mathsf{pk}_{acc}), \mathsf{Acc}_\mathcal{X}, \mathsf{aux}, x_i)$: This algorithm takes a key pair $(\mathsf{sk}_{acc}, \mathsf{pk}_{acc})$, an accumulator $\mathsf{Acc}_\mathcal{X}$, auxiliary information aux, and a value x_i. It returns $\perp$, if $x_i \notin \mathcal{X}$, and a witness wit_{x_i} for x_i otherwise.
- $\mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}_\mathcal{X}, wit_{x_i}, x_i)$: This algorithm takes a public key pk_{acc}, an accumulator $\mathsf{Acc}_\mathcal{X}$, a witness wit_{x_i}, and a value x_i. It returns *true* if wit_{x_i} is a witness for $x_i \in \mathcal{X}$ and *false* otherwise.

The above definition focuses on static accumulators. In contrast, dynamic accumulators enable the accumulated value and witnesses to be publicly updated whenever an element is added or removed from the set. Various properties, such as being trapdoorless, universal, or supporting subset queries, are associated with this primitive. For a comprehensive list of definitions, functionalities, and properties, we refer to [15, 37]. Here, we only consider the basic properties.

Security. One requires collision resistance which states that it should be computationally infeasible to find a witness for any non-accumulated value $x \notin \mathcal{X}$.

Definition 7 (Collision resistance [15, 37]). *An accumulator scheme is said to satisfy collision resistance if for all PPT adversaries* $\mathcal{A}$*, the following advantage is negligible:*

$$\Pr\left[\begin{array}{l} (\mathsf{sk}_{acc}, \mathsf{pk}_{acc}) \leftarrow \mathsf{Setup}(1^\lambda, q), (\mathsf{Acc}_\mathcal{X}, \mathsf{aux}) \leftarrow \mathsf{Eval}((\mathsf{sk}_{acc}, \mathsf{pk}_{acc}), \mathcal{X}), \\ (\mathcal{X}, wit_{x_i}, x_i) \leftarrow \mathcal{A}^{\mathcal{O}^W}(\mathsf{pk}_{acc}) : \mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}_\mathcal{X}, wit_{x_i}, x_i) = 1 \wedge x_i \notin \mathcal{X} \end{array}\right]$$

where $\mathcal{O}^\mathrm{W}$ represent an oracle for the algorithm WitCreate. An adversary is allowed to query them an arbitrary number of times. Note that if $\mathcal{O}^\mathrm{W}$ is queried for an element $x \notin \mathcal{X}$, the oracle outputs a reject symbol $\perp$.

2.6 Ring Signatures

Ring signature (RS) schemes [79] allow members of an ad-hoc group $\mathcal{R}$ (known as a ring) to anonymously sign messages on behalf of this group defined by their public keys. While it is possible to verify a ring signature against the public keys associated with $\mathcal{R}$, determining the actual signer remains infeasible, thus guaranteeing unconditional anonymity. This characteristic makes ring signatures particularly useful in various applications, especially in whistleblowing and ensuring transaction privacy in cryptocurrencies.

Definition 8 (Ring Signature Scheme). *A ring signature scheme* RS *is a tuple of the following algorithms:*

- $\mathsf{Setup}(1^\lambda)$: Takes as input a security parameter λ and outputs public parameters pp.
- $\mathsf{KeyGen}(\mathsf{pp})$: Takes as input the public parameter pp and outputs a key pair $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_i, m, \mathcal{R})$: Takes as input the public parameters pp, a secret key sk_i, a message $m \in \mathcal{M}$, and a ring $\mathcal{R} = (\mathsf{pk}_j)_{j\in[n]}$ of n public keys such that $\mathsf{pk}_i \in \mathcal{R}$. It outputs a signature σ.
- $\mathsf{Verify}(\mathsf{pp}, m, \sigma, \mathcal{R})$: Takes as input the public parameters pp, a message $m \in \mathcal{M}$, a signature σ, and a ring $\mathcal{R}$. It outputs a bit $b \in \{0, 1\}$.

We formally define ring signature schemes in alignment with [19, 39].

Unforgeability. This property requires that without any secret key sk_i that corresponds to a public key $\mathsf{pk}_i \in \mathcal{R}$, it is infeasible to produce valid signatures with respect to arbitrary such rings $\mathcal{R}$.

Definition 9 (Unforgeability [19, 39]). *A* RS *scheme provides unforgeability if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* $\varepsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda),\\ \{(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp})\}_{i\in[\mathsf{poly}(\lambda)]},\\ \mathcal{O} \leftarrow \{\mathsf{Sig}(\cdot,\cdot,\cdot), \mathsf{Key}(\cdot)\},\\ (m^*, \sigma^*, \mathcal{R}^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\{\mathsf{pk}_i\}_{i\in[\mathsf{poly}(\lambda)]})\end{array} : \begin{array}{c}\mathsf{Verify}(m^*, \sigma^*, \mathcal{R}^*) = 1 \wedge\\ (\cdot, m^*, \mathcal{R}^*) \notin Q^{\mathsf{Sig}} \wedge\\ \mathcal{R}^* \subseteq \{\mathsf{pk}_i\}_{i\in[\mathsf{poly}(\kappa)]\setminus Q_{\mathsf{Key}}}\end{array}\right] \leq \varepsilon(\kappa),$$

where $\mathsf{Sig}(i, m, \mathcal{R}) := \mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$*,* Sig *returns* $\perp$ *if* $\mathsf{pk}_i \notin \mathcal{R} \vee i \notin [\mathsf{poly}(\kappa)]$*, and* Q^{Sig} *records the queries to* Sig*. Furthermore,* $\mathsf{Key}(i)$ *returns* sk_i *and* Q_{Key} *records the queries to* Key*.*

Anonymity. This property ensures that it is computationally infeasible to determine which ring member generated a particular signature, provided there are at least two honest members in the ring. Our anonymity notion follows the one in [39] that is corresponds to the strongest definition in [19], known as anonymity against full key exposure.

Definition 10 (Anonymity [19, 39]). *A* RS *scheme provides anonymity if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), b \leftarrow \{0,1\},\\ \{(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp})\}_{i\in[\mathsf{poly}(\lambda)]},\\ \mathcal{O} \leftarrow \{\mathsf{Sig}(\cdot,\cdot,\cdot)\},\\ (m, j_0, j_1, \mathcal{R}, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}}(\{\mathsf{pk}_i\}_{i\in[\mathsf{poly}(\lambda)]}),\\ \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_{j_b}, m, \mathcal{R}),\end{array} : \begin{array}{c} b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{st}, \sigma, \{\mathsf{sk}_i\}_{i\in[\mathsf{poly}(\lambda)]})\\ b = b' \wedge \{\mathsf{pk}_{j_0}, \mathsf{pk}_{j_1}\} \subseteq \mathcal{R}\end{array}\right] \leq \frac{1}{2} + \varepsilon(\lambda)$$

where $\mathsf{Sig}(i, m, \mathcal{R}) := \mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$*.*

3 Shrinking SP Commitments for DH Messages

We begin by constructing *constant-size* and *strict* structure-preserving commitments for group elements using a variant of Diffie-Hellman (DH) messages. By *strict*, we refer to the definition provided in [6], which means that the messages, commitments, and openings are all confined to the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

Our construction is notable for bypassing the impossibility result of Abe et al. [6], which establishes that strictly structure-preserving commitments to source-group elements cannot be smaller than the input message size and scale linearly with the number of group elements.

In contrast, in a relaxed structure-preserving setting, constant-size commitments to group elements can be achieved by allowing elements from the target group $\mathbb{G}_T$ in the commitment [5,6] (see Sec.2.1). While including $\mathbb{G}_T$ elements is acceptable when only witness indistinguishability is required for accompanying Groth-Sahai (GS) proofs, it becomes problematic when zero-knowledge is necessary (see [6]). Therefore, ensuring group-to-group commitments remain entirely within the source groups is crucial for achieving zero-knowledge, and it also allows commitments to other commitments, latter being very interesting for applications. Another way of bypassing this impossibility is done by Abe et al. in [7], who construct SP commitments that are shrinking in a relaxed binding setting, e.g., where the message space for committing is $\mathbb{Z}_p$ and the one for verification is $\mathbb{G}_1$.

Our approach is inspired by circumventing impossibility results and lower bounds in SPS [48,49] achieved by switching from arbitrary group elements in the message space to a more structured message space and in particular a Diffie-Hellman (DH) message space [3,44]. This has also recently been used to construct the first threshold SPS in [33]. In doing so we obtain shrinking strictly structure-preserving (group-to-group) commitments. The so obtained SP commitment scheme in nature is similar to the γ-binding commitment scheme in [7], but is is not trapdoor and strictly structure-preserving, i.e., messages, commitments, and openings are all elements of the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

We first define a new DH message space $\mathcal{M}_{\mathsf{DH}}$ and then present a construction for shrinking strictly structure-preserving (group-to-group) commitments based on this new DH message type.

New DH Message Space $\mathcal{M}_{\mathsf{DH}}$. We slightly adapt the DH message technique (cf. Sec. 2) as:

Definition 11 (DH Message Space ($\mathcal{M}_{\mathsf{DH}}^{new}$)). *Let the public parameters be a vector $\mathbf{X}$ of random elements in $\mathbb{G}_1$ (e.g., $\mathbf{X} = (P^{x_i})_{i\in[k]}$). We then define $\mathcal{M}_{\mathsf{DH}}$ as a DH message space, if the following property hold: For the message vector $(\mathbf{M}, \hat{\mathbf{N}}) = (M_1, \ldots, M_k, \hat{N}_1, \ldots, \hat{N}_k)$ there exist $m_i \in \mathbb{Z}_p$ ($1 \leq i \leq k$) s.t. $M_i = X_i^{m_i}$ and $\hat{N}_i = \hat{P}^{m_i}$ for all i.*

Note that membership in this message space can be efficiently checked by $e(M_i, \hat{P}) = e(X_i, \hat{N}_i)$. We can obtain the plain messages $\hat{P}^{m_i}$ in $\mathbb{G}_1$ as well by simply switching the vector $\mathbf{X}$ to reside in $\mathbb{G}_2$ and keeping the vector $\mathbf{M} = P^{m_i}$ in $\mathbb{G}_1$.

Construction. We introduce our group-to-group commitments, where the messages, commitments are all restricted to the source groups $\mathbb{G}_1$ and $\mathbb{G}_2$. Despite this, the construction remains concise and compact, achieving a compactness property.

Scheme 1 (Shrinking Strictly Structure-Preserving Commitment) *Our commitment scheme is composed of the following PPT algorithms:*

- $\mathsf{KeyGen}(1^\lambda)$: Run $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \leftarrow \mathsf{BGSetup}(1^\lambda)$, for $i = 1, \ldots, k$, choose $X_i \leftarrow\!\!\$\ \mathbb{G}_1$. Output commitment key $\mathsf{pk} := (\mathsf{BG}, X_1, \ldots, X_k)$. Note that $\mathbf{X}$ does not need to have structure, and therefore, no trusted setup is required. Indeed, we can use a random oracle to generate these elements.
- $\mathsf{Commit}(\mathsf{pk}, msg)$: Parse msg as $((M_1, \hat{N}_1), \ldots, (M_k, \hat{N}_k)) \in \mathcal{M}_{\mathsf{DH}}$ (check if they are generated correctly by $e(M_i, \hat{P}) = e(X_i, \hat{N}_i)$ for $i \in [k]$). Choose $r \leftarrow\!\!\$\ \mathbb{Z}_p$, and compute

$$\mathsf{com} := P^r \cdot \prod_{i=1}^{k} M_i \ \wedge\ \mathsf{Open} := (\hat{R} = \hat{P}^r).$$

- $\mathsf{Verify}(\mathsf{pk}, \mathsf{com}, msg, \mathsf{Open})$: Parse msg as $((M_1, \hat{N}_1), \ldots, (M_k, \hat{N}_k)) \in \mathcal{M}_{\mathsf{DH}}$, and Open as $\hat{R}$. Check for well-formedness of the messages by $e(M_i, \hat{P}) = e(X_i, \hat{N}_i)$ for $i \in [k]$, and abort otherwise. Output 1 if and only if:

$$e(\mathsf{com}, \hat{P}) = e(P, \hat{R}) \prod_{i=1}^{k} e(X_i, \hat{N}_i)\,.$$

Theorem 1. *Scheme 1 is perfectly hiding. Furthermore, it is binding under the DBP assumption.*

Proof. We provide proofs of the hiding and binding properties as follows:

Hiding. The proof of hiding is straightforward. As r is uniformly random from $\mathbb{Z}_p$ and P is a generator, P^r is a uniformly random group element, and so is com, independent of the committed message.

Binding. For an adversary $\mathcal{A}$ breaking the binding property of the commitment scheme, consider the following adversary $\mathcal{B}$: on input a DBP challenge (BG, h_z), $\mathcal{B}$ sets $X_i := h_z^{x_i}$ for random $x_i \leftarrow\!\!\$\, \mathbb{Z}_p$ for $i \in [k]$. $\mathcal{B}$ aborts if $X_i = 1$ for any i, but this happens only with negligible probability. $\mathcal{B}$ then runs $\mathcal{A}$ on $\mathsf{pk} = (\mathsf{BG}, X_1, \ldots, X_q)$. If $\mathcal{A}$ returns a commitment com together with two different valid messages and openings $(\hat{N}_1, \ldots, \hat{N}_q, \hat{R})$ and $(\hat{N}'_1, \ldots, \hat{N}'_q, \hat{R}')$, then $\mathcal{B}$ computes

$$\hat{Z}^* := \prod_{i=1}^{k} \left(\frac{\hat{N}_i}{\hat{N}'_i}\right)^{x_i}, \quad \hat{R}^* := \frac{\hat{R}}{\hat{R}'}.$$

As it holds that $e(\mathsf{com}, \hat{P}) = e(P, \hat{R}) \prod_{i=1}^{k} e(X_i, \hat{N}_i) = e(P, \hat{R}') \prod_{i=1}^{k} e(X_i, \hat{N}'_i)$, it follows that

$$1 = e\left(P, \frac{\hat{R}}{\hat{R}'}\right) \prod_{i=1}^{k} e\left(h_z^{x_i}, \frac{\hat{N}_i}{\hat{N}'_i}\right) = e\left(P, \hat{R}^*\right) e\left(h_z, \hat{Z}^*\right).$$

To see this, we first verify that the simulated inputs to $\mathcal{A}$ are correctly distributed. In KeyGen, each X_i distributes uniformly over $\mathbb{G}_1$, whereas $\mathcal{B}$ distributes uniformly over $\mathbb{G}_1^*$. Thus, the simulated parameters are statistically close to the real ones.

Moreover, note that since a valid output from $\mathcal{A}$ satisfies $msg \neq msg'$, there exists an index $i^* \in [k]$ such that $\hat{N}_{i^*} \neq \hat{N}'_{i^*}$. Thus $\hat{Z}^*$ follows the distribution of $\left(\frac{\hat{N}_{i^*}}{\hat{N}'_{i^*}}\right)^{x_i^*}$. Since $\frac{\hat{N}_{i^*}}{\hat{N}'_{i^*}} \neq 1$ and x_{i^*} is uniform over $\mathbb{Z}_p^*$, we conclude that $\hat{Z}^* \neq 1$ with overwhelming probability.

Thus $\hat{Z}^*, \hat{R}^* \in \mathbb{G}_2^*$ is a solution to the given DBP instance, and $\mathcal{B}$ succeeds with roughly the same probability as $\mathcal{A}$. □

4 Structure Preserving Vector Commitments

In this section, we introduce a Structure-Preserving Vector Commitment scheme SPVC over a message space $\mathbf{M} \in \mathbb{G}_1$ (or $\mathbb{G}_2$). As a preliminary step, we present a Weak Binding SPVC (WSPVC), where commitments are generated honestly, akin to the concept of static accumulators. We demonstrate that this can be realized through a generic construction based on any compact structure-preserving signature (SPS) scheme, effectively yielding a structure-preserving accumulator.

More precisely, we present a simple compiler in which commitments are derived from signature public keys, with auxiliary data including the corresponding secret keys and messages, while the opening/proof is simply a signature. To bind a position to a message, we introduce public information $(U_1, \ldots, U_n) \in \mathcal{M}$ into the public parameters, where random elements U_i are signed together with M_i as indices to the messages. This approach is compatible with any compact SPS. We propose an instantiation of WSPVC using the FHS signature [47], noting that message randomization is not required in this context. This message randomization can be prevented by fixing the first element of the message vector to a predetermined element U, which needs to be verified during the verification.

In Sec. 4.2, we enhance our security model to present a structure-preserving vector commitment with standard binding by incorporating message tags (identifiers). This ensures that messages will not verify unless they are computed using a compatible setup with a Common Reference String (CRS).

4.1 Weak Binding Vector Commitments

We present our weak binding vector commitment WSPVC, which, as mentioned above, can be constructed from a structure-preserving (SP) signature, assuming the committer is honest and the commitments are generated correctly.

Scheme 2 (Weak Binding VC) *A* WSPVC *is a tuple of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda, q)$: Given the security parameter λ and the size q of the committed vector, the key generation choose $(U_1, \ldots, U_q) \in \mathcal{M}$ (from the message space) and outputs public parameters $\mathsf{pp} = (U_1, \ldots, U_q, \mathsf{BG})$.

– $\mathsf{Commit}(M_1, \ldots, M_k)$: On input a vector of q messages as $(M_1, \ldots, M_q) \in \mathcal{M} \in \mathbb{G}_1^q$ and the public parameters pp, output a commitment com and auxiliary information aux as follows: Run $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{SPS.KeyGen}(1^\lambda, 2)$ and then set

$$\mathsf{com} = \mathsf{pk} \quad \wedge \quad \mathsf{aux} = (r, \sigma_1, \ldots, \sigma_n)$$

Where σ_i is a SPS signature on (M_i, U_i). For random $(r, y) \leftarrow\!\!\$\ \mathbb{Z}_p$, and using the FHS SPS as example:

$$\mathsf{pk} = (\hat{X}_1 = \hat{P}^{x_1}, \hat{X}_2 = \hat{P}^{x_2}) \wedge \sigma_i = \left(Z = (U_i^{x_1} \cdot M_i^{x_2})^{1/y}, Y = P^y, \hat{Y} = \hat{P}^y\right)$$

– $\mathsf{Open}(M, i, \mathsf{aux})$: This algorithm is run by the committer to produce a proof π_i that M is the i-th committed message. Pick the related signature $\sigma_i \leftarrow \mathsf{aux}$ and output a proof $\pi_i = \sigma_i$ for M_i.
– $\mathsf{Verify}(\mathsf{com}, M, i, \pi_i)$: The verification algorithm accepts (i.e., it outputs 1) only if π_i is a valid proof that com was created for M_i by verifying $\mathsf{SPS.Verify}(\mathsf{pk} = \mathsf{com}, M_i, \sigma_i = \pi) = 1$. For the FHS signature scheme, this is defined as follows:

$$\mathsf{SPS.Verify} : e(Z, \hat{Y}) = e(M_j, \hat{X}_2)e(U_j, \hat{X}_1) \wedge e(Y, \hat{P}) = e(P, \hat{Y})$$

– $\mathsf{Rand}(\mathsf{com}, \pi_j)$: Randomize a proof for a message M_j as: Pick a random $\mu \leftarrow\!\!\$\ \mathbb{Z}_p$:

$$\pi'_j = (\sigma'_j = (Z^{1/\mu}.Y^\mu, \hat{Y}^\mu))$$

Theorem 2. *If the SPS is unforgeable, then the* WSPVC *in Scheme 2 satisfies weak binding.*

Proof (Sketch). The proof of binding is straightforward. If the signature is unforgeable, then the commitment is position binding; specifically, a new proof requires a new signature on a different message, which would violate the unforgeability of our signature scheme. □

Remark 1. Note that since the commitment serves as a public key and is independent of the message, updating the commitments for a new message can be achieved simply by signing the new message. Moreover, the message space for the commitment can be adapted by selecting a suitable SPS. Specifically, we can configure the SPVC to handle unilateral messages, where messages are drawn solely from either $\mathbb{G}_1$ or $\mathbb{G}_2$, or bilateral messages, which allow for a mix of elements from both $\mathbb{G}_1$ and $\mathbb{G}_2$.

4.2 Vector Commitment based on q-DHE

Now we introduce our SPVC scheme within the standard position-binding model based on the q-DHE assumption, referred to as q-DHE SPVC. Before diving into the construction, we define a structured message space built on the q-DHE parameters, which can be viewed as a CRS. This design of a message structure on top of a CRS is inspired by [53] (Although their CRS and the way it is used differ from ours). Indeed, this message space enables the construction of the SPVC based on the q-DHE VC schemes [52,66] in the strong model, i.e., transforming these constructions for $m \in \mathbb{Z}_p$ into a structure-preserving commitment. Furthermore, it allows us to extract the discrete logarithm of the messages in the proof (see Lemma 1), reducing the construction to q-DHE.

Message Structure. Each message consists of vectors with $\ell + 1$ elements over $\mathbb{G}_1$, where each vector includes two main message components and a tag vector with $\ell - 1$ elements (e.g., $M_i = (\mathbf{M} = (M_{i0}, M_{i1}), \mathbf{T} = (T_{i2}, \ldots, T_{in-1}))$). The structure of the vector is determined by the q-DHE parameters specified by

$$\mathsf{pp} = \begin{pmatrix} B_1 = P^{\alpha^1}, \ldots, B_n = P^{\alpha^\ell}, B_{\ell+1} = P^{\alpha^{\ell+1}}, \ldots, B_{2\ell} = P^{\alpha^{2\ell}}; \\ \hat{B}_1 = \hat{P}^{\alpha^1}, \ldots, \hat{B}_\ell = \hat{P}^{\alpha^\ell}; g_t^{\alpha^{\ell+1}} \end{pmatrix} \tag{1}$$

The message space $\mathcal{M}^{\mathsf{pp},\ell}$ is defined according to these parameters.

$$\mathcal{M}^{\mathsf{pp},\ell} = \left\{ \begin{array}{l} ((\mathbf{M}_1, \mathbf{T}_1), \ldots, (\mathbf{M}_n, \mathbf{T}_\ell)) \mid \exists \mathbf{m} = (m_1, \ldots, m_\ell) \in \mathbb{Z}_p^\ell \text{ s.t.} \\ \forall 1 \le i \le \ell,\ M_i = (M_{i0} = P^{m_i}, M_{i1} = P^{\alpha^i \cdot m_i}, T_{ij} = B_j^{m_i}) \\ \text{where } |j| = \ell - 1 \wedge j \in [i+1, 2\ell] \wedge j \neq \ell + 1 \end{array} \right\} \tag{2}$$

In our scheme, the public parameters include additional bases $\hat{\mathbf{B}} = \{\hat{B}_1, \ldots, \hat{B}_{2\ell}\}$ that are used to verify whether a vector is in the message space and also pp does not include $B = P^{\alpha^{\ell+1}}$, meaning that no index $j = \ell + 1$ exist. Specifically, given a vector $\mathbf{M} = \{(\mathbf{M}_1, \mathbf{T}_1), (\mathbf{M}_2, \mathbf{T}_2), \ldots, (\mathbf{M}_\ell, \mathbf{T}_\ell)\}$, we use these extra bases to check that the pairing satisfies the relationship with respect to above i and j:

$$e(M_i, \hat{B}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{P}) = e(M_{i0}, \hat{B}_i) \tag{3}$$

To clarify this process, let's consider an example with $\ell = 3$. For M_1 and M_2 we have

$$M_1 = (\underbrace{P^{\alpha^1 \cdot m_1}, P^{m_1}}_{\text{main part}}, \underbrace{(B_2^{m_1}, B_3^{m_1})}_{\text{tag}}) \wedge M_2 = (\underbrace{P^{\alpha^2 \cdot m_2}, P^{m_2}}_{\text{main part}}, \underbrace{(B_3^{m_2}, B_5^{m_2})}_{\text{tag}}) =$$

$$M_1 = (P^{\alpha^1 \cdot m_1}, P^{m_1}, (P^{\alpha^2 \cdot m_1}, P^{\alpha^3 \cdot m_1})) \wedge M_2 = (P^{\alpha^2 \cdot m_2}, P^{m_2}, (P^{\alpha^3 \cdot m_2}, P^{\alpha^5 \cdot m_2}))$$

Construction. We now present a SPVC in a standard model which ensures that the commitment can be generated even in the presence of malicious behavior. In this model, tags are used to generate the proof/witness, while the main messages are used for verification and commitment. To highlight the randomization property of our primitive, we define a new algorithm Rand designed for this purpose. Our new construction is based on the q-DHE assumption (let us assume $q = \ell$).

Scheme 3 (q-DHE SPVC) *Our q-DHE* SPVC *scheme is defined as follows:*

- $\mathsf{KeyGen}(1^\lambda, \ell)$: Given the security parameter λ and the size ℓ of the committed vector, the key generation picks $\alpha \leftarrow\!\$\ \mathbb{Z}_p$ and outputs some public parameters $\mathsf{pp} = (\mathsf{BG}, B_1 = P^{\alpha^1}, \ldots, B_n = P^{\alpha^\ell}, B_{\ell+2} = P^{\alpha^{\ell+2}}, \ldots, B_{2\ell} = P^{\alpha^{2\ell}}; B_{\ell+1} = P^{\alpha^{2\ell}}, \hat{B}_1 = \hat{P}^{\alpha^1}, \ldots, \hat{B}_\ell = \hat{P}^{\alpha^\ell}; g_t^{\alpha^{\ell+1}})$.
- $\mathsf{Commit}(M_1, \ldots, M_\ell)$: On input a vector of ℓ messages parse msg as $(M_1, \ldots, M_\ell) \in \mathcal{M}^{\mathsf{pp},l}$. Check if messages are correctly generated via Equation (3) s.t. $e(M_i, \hat{B}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{P}) = e(M_{i0}, \hat{B}_i)$ for all $i \in [\ell]$, pick $r \leftarrow\!\$\ \mathbb{Z}$ and output a commitment com and auxiliary information aux:

$$\mathsf{com} = \left(C_1 = P^r \cdot \prod_{i \in [\ell]} M_{i1} \right) \ \wedge \ \mathsf{aux} = r$$

- $\mathsf{Open}((M, i, \mathsf{aux}))$: This algorithm is run by the committer to produce a proof π_i that M is the i-th committed message:

$$\pi_i = \left(B_{\ell+1-i}^r \cdot \prod_{j \neq i, k = \ell+1-i+j} T_{jk} \right)$$

- $\mathsf{Verify}(\mathsf{com}, M, i, \pi_i)$: The verification algorithm accepts (i.e., it outputs 1) only if the messages is created correctly $M_i \in \mathcal{M}^{\mathsf{pp},\ell}$ which means $e(M_{i1}, \hat{B}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_i, \hat{P}) = e(M_{i0}, \hat{B}_i)$ for all i, j in Equation(2) and also π is a valid proof that com was created for M_i:

$$e(C_1, \hat{B}_{\ell+1-i}) = e(\pi_i, \hat{P}) \cdot e(M_i, \hat{B}_{\ell+1-i})$$

Additional (and optional) properties (update the commitment and randomization), which is defined using the following algorithms:

- $\mathsf{Update}(\mathsf{com}, M, M', i)$: This algorithm is run by the committer who produced com and wants to update it by changing the i-th message to $M_i' = (\mathbf{M}' = (M_{i0}, M_{i1}), \mathbf{T}')$. Check if the new message is created correctly via Equation(3) then outputs a new commitment com' as: $C' = (C/M_{i1}) \cdot M_{i1}'$.
- $\mathsf{Rand}(\mathsf{com}, \pi_i, M_i) \rightarrow (C', \pi_i', M_i')$: Randomize the commitment and proof for a randomized message M_i' as: Pick a random $\mu \leftarrow\!\$\ \mathbb{Z}_p$ and compute:

$$C' = C^\mu \wedge \pi_i' = \pi^\mu,$$

  which is valid for the randomized message $M_i' = (\mathbf{M}^\mu, \mathbf{T}^\mu)$. Update aux with μ, i.e., set $\mathsf{aux}' = \mu \cdot \mathsf{aux}$.

Correctness: We can check that if commitments are properly generated, then proofs always satisfy the verification check. For the left hand side we have:

$$e(C_1, \hat{B}_{\ell+1-i}) = e(P^r \cdot \prod_{j\in[\ell]} M_j, \hat{B}_{\ell+1-i}) =$$

$$e(P^r \cdot P^{\sum_{j\in[\ell]} \alpha^j \cdot m_j}, \hat{P}^{\alpha^{\ell+1-i}}) =$$

$$e(P^{r\cdot\alpha^{\ell+1-i}} \cdot (P^{\sum_{j\in[\ell]} \alpha^j \cdot m_j})^{\alpha^{\ell+1-i}}, \hat{P}) = e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_{j\in[\ell]} \alpha^j \cdot m_j})^{\alpha^{\ell+1-i}}, \hat{P}) =$$

$$e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_{j\in[\ell]} m_j \alpha^{\ell+1-i+j}}), \hat{P})$$

For the right side we have

$$e((B^r_{\ell+1-i} \cdot \prod_{j\neq i} M_{\ell+1-i+j}), \hat{P}) \cdot e(M_i, \hat{B}_{\ell+1-i}) =$$

$$e(B^r_{\ell+1-i} \cdot \prod_{j\neq i} P^{m_j \cdot \alpha^{\ell+1-i+j}}, \hat{P}) \cdot e(P^{\alpha^i \cdot m_i}, \hat{P}^{\alpha^{\ell+1-i}}) =$$

$$e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_{j\neq i} m_j \cdot \alpha^{\ell+1-i+j}}, \hat{P}) \cdot e(P^{\alpha^i \cdot m_i (\alpha^{\ell+1-i})}, \hat{P}) =$$

$$e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_{j\neq i} m_j \cdot \alpha^{\ell+1-i+j}}, \hat{P}) \cdot e(P^{m_i \alpha^{\ell+1}}, \hat{P}) =$$

$$e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_{j\in[\ell]} m_j \cdot \alpha^{\ell+1-i+j}}, \hat{P})$$

This is equal to the left hand side so we have:

$$e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_j m_j \cdot \alpha^{\ell+1-i+j}}, \hat{P}) = e(B^r_{\ell+1-i}, \hat{P}) e(P^{\sum_{j\in[\ell]} m_j \alpha^{\ell+1-i+j}}, \hat{P})$$

Theorem 3. *Scheme 3 is binding in the Generic Group Model (GGM) assuming the q-DHE assumption.*

Proof of Biding: To make our proof simpler, we will first prove a lemma (Lemma 1) that we use in our proof. We can see that if α^i are random, then, verifying messages and tags using pp allows us to extract the m_i values in the generic group model.

Assume we extract the m_i and m'_i using the following lemma 1. Now we can reduce our scheme to q-DHE assumption similar to the mercurial commitment. Lets $\mathcal{A}$ comes up with a commitment (C, aux), an index $i \in \{1, \ldots, \ell\}$, a valid openings π and π' to m_i and m'_i at position i, such that $m_i \neq m'_i$. We must have

$$e(\pi_i, \hat{P}) \cdot e(P^{\alpha^i}, \hat{P}^{\alpha^{\ell+1-i}})^{m_i} = e(\pi'_i, \hat{P}) \cdot e(P^{\alpha^i}, \hat{P}^{\alpha^{\ell+1-i}})^{m'_i}$$

So that $e(\pi_i/\pi'_i, \hat{P},) = e(P^{\alpha^i}, \hat{P}^{\alpha^{\ell+1-i}})^{m'_i - m_i}$ and $e((\pi_i/\pi'_i)^{1/(m'_i - m_i)}, \hat{P}) = e(P^{\alpha^i}, \hat{P}^{\alpha^{\ell+1-i}})$. Since $m_i \neq m'_i$, the latter relation implies that $P^{\alpha^{\ell+1}} = (\pi_i/\pi'_i)^{1/(m'_i - m_i)}$ is revealed by the collision, which contradicts the q-DHE assumption.

Lemma 1 (Extraction of Discrete Logarithms from Valid Messages). *Let* pp *be a public parameter. If the messages satisfy the following conditions for all* $i \in [\ell]$*:*

$$e(M_i, \hat{B}_{j-i}) = e(T_{ij}, \hat{P}) \quad \textit{and} \quad e(M_i, \hat{P}) = e(M_{i0}, \hat{B}_i),$$

then the adversary is able to extract the discrete logarithms $\{m_i\}_{i\in[\ell]}$ *such that:*

$$m_i = \mathsf{dlog}_{B_i}(M_i) = \mathsf{dlog}_{B_j}(T_{ij}) \quad \textit{for } 1 \leq i \leq \ell,$$

where $|j| = n - 1$*,* $j \in [i+1, 2n]$*, and* $j \neq n+1$*.*

Our main technical result is to prove that our scheme satisfies binding in the generic group model (GGM) [80] for asymmetric (type-3) bilinear groups, for which there are no efficiently computable homomorphism between P and $\hat{P}$. In this model, the adversary is only given handles of group elements, which are uniform random strings. To perform group operations, it uses an oracle to which it can submit handles and receives back the handle of the sum, inversion, etc., of the group elements for which it submitted handles.

Proof. For a fixed i, consider the values $M_{i0}, M_{i1}, \{T_{ij}\}_{j=i+1,j\neq n+1}^{i+n}$ output by an adversary. With P and $\{B_u\}_{u=1,u\neq n+1}^{2n}$ being the values specified in pp, these values must now have the form:

$$M_{i0} = P^{b_0} \cdot \prod_{\substack{k=1\\k\neq n+1}}^{2n} B_k^{b_k}\,, \quad M_{i1} = P^{c_0} \cdot \prod_{\substack{k=1\\k\neq n+1}}^{2n} B_k^{c_k}\,, \text{ and } \quad T_{ij} = P^{a_{j0}} \cdot \prod_{\substack{k=1\\k\neq n+1}}^{2n} B_k^{a_{jk}}\,,$$

where all b_u, c_u, a_{ju} are known to the adversary. For the remainder of this proof, all sums and products are over $k = 1, \ldots, 2n$ with $k \neq n+1$, which will be omitted for notational convenience.

Using the structure of the B_j as defined in Section 4.2 and taking the discrete logarithm in P we obtain:

$$m_{i0} = b_0 + \sum b_k \alpha^k \tag{4}$$

$$m_{i1} = c_0 + \sum c_k \alpha^k \tag{5}$$

$$t_{ij} = a_{j0} + \sum a_{jk} \alpha^k \quad \forall j = i+1, \ldots, i+n,\ j \neq n+1\,. \tag{6}$$

Furthermore, by the verification equations $e(M_i, B_{j-i}) = e(T_{ij}, P)$ and $e(M_i, P) = e(M_{i0}, B_i)$ we obtain by a similar argument that:

$$m_{i1}\alpha^{j-i} = t_{ij} \qquad \forall j = i+1, \ldots, i+n,\ j \neq n+1 \tag{7}$$

$$m_{i1} = m_{i0}\alpha^i\,. \tag{8}$$

$\mathbf{c_0}, \ldots, \mathbf{c_{i-1}} = \mathbf{0}$: By combining Equations (4), (5) and (8) we obtain:

$$c_0 + \sum c_k \alpha^k = b_0 \alpha^i + \sum b_k \alpha^{k+i}\,.$$

Given that the lowest degree on the right hand side is i, we directly obtain that $c_0 = \cdots = c_{i-1} = 0$.

$\mathbf{c_{i+1}}, \ldots, \mathbf{c_n} = \mathbf{0}$: By combining Equations (5) to (7) we obtain for all j:

$$c_0 + \sum c_k \alpha^{k+j-i} = a_{j0} + \sum a_{jk} \alpha^k\,. \tag{9}$$

As α^{n+1} does not occur on the right hand side, the term $c_{n+1-j+i}\alpha^{n+1}$ on the left hand side must be 0 for all $j = i+1, \ldots, i+n$ satisfying $j \neq n+1$. Thus, in particular for $j = i+1, \ldots, n$, it follows that $c_{i+1} = \cdots = c_n = 0$.

$\mathbf{c_{n+2}}, \ldots, \mathbf{c_{n+i-1}}, \mathbf{c_{n+i+1}}, \ldots, \mathbf{c_{2n}} = \mathbf{0}$: In (9), the highest degree on the right hand side equals $2n$. Thus, the coefficients of α^{2n+1} on the right hand side equals 0, i.e., $c_{2n+1-j+i} = 0$ for all $j = i+1, \ldots, i+n$ satisfying $j \neq n+1$. This immediately yields $c_{n+2} = \cdots = c_{2n} = 0$ except for c_{n+i} corresponding to $j = n+1$.

$\mathbf{c_{n+i}} = \mathbf{0}$: In the case that $i = 1$, there is nothing to prove as there is no c_{n+1} in (5). For $i > 1$, consider the term $c_{n+i}\alpha n + j$ in (9) for $j = n+2$. As α^{2n+2} does not exist on the right hand side, it directly follows that $c_{n+i} = 0$.

Combining the above observations we obtain that $c_k = 0$ for all $k \neq i$. Rewriting (9) now yields for all j that:

$$c_i \alpha^j = a_{j0} + \sum a_{jk} \alpha^k\,.$$

Comparing coefficients gives us that $a_{jk} = 0$ for all $k \neq j$ and $a_{jj} = c_i$, such that $t_{ij} = c_i \alpha^j$.

Overall, this implies that $M_{i1} = P^{c_0} \cdot \prod B_k^{c_k} = B_i^{c_i}$ and $T_{ij} = P^{a_{j0}} \cdot \prod B_k^{a_{jk}} = B_j^{c_i}$, or equivalently $m_i := c_i = \mathsf{dlog}_{B_i} M_i = \mathsf{dlog}_{B_j} T_{ij}$ for all $j = i+1, \ldots, i+n$ with $j \neq n+1$. Thus, the adversary must be able to extract the discrete log of the message, and thus by induction, must always know the discrete logs of messages during the game. □

Remark 2. We note that in the context of vector commitments (VC), the commitments do not need to be hiding, which makes the inclusion of randomness r seem unnecessary. Moreover, the randomness r is not required for the binding proof. Nevertheless, we retain it here as it might be useful for other applications in the future or for achieving properties like hiding, which are not directly required in our current setting.

5 Structure-Preserving Accumulator

In this section, we introduce the concept of a structure-preserving accumulator (SPA) and demonstrate how our vector commitment can be adapted into an accumulator, resulting in the first structure-preserving accumulator of this kind.

Catalano and Fiore [27] proposed a black-box construction of accumulators based on vector commitments. Their approach involves creating a succinct commitment C to a vector $\mathbf{X} = (x_1, \ldots, x_n)$ through a vector commitment. The commitment C ensures that it is computationally infeasible to open any position i to a value x_i' different from the original x_i. In their construction, the accumulation domain is represented by the set $D = \{1, \ldots, t\}$, and the accumulator is modeled as a commitment to a binary vector of length t. Each bit i indicates whether the element $i \in D$ is included in the accumulator. Membership or non-membership of an element is verified by revealing the corresponding position i of the commitment as either 1 or 0. However, it is not clear how to apply this result to the SP setting. Moreover, when using a vector commitment in this way, it often reveals the position of elements within the set, which may expose sensitive information in certain contexts, e.g., when using them as a building block for privacy-preserving primitives such as ring signatures. Therefore, we are interested in an approach where the witness does not disclose any information about the elements.

SPA from (signature-based) Weak Binding VC. A weak binding vector commitment can naturally be expressed as an accumulator, as both schemes are essentially equivalent when the accumulator and the vector commitment are honestly generated. In the signature-based accumulator, achieving index-hiding is straightforward by omitting the index elements U_i associated with each message from the scheme. Consequently, the witness becomes a signature for the element M_i only, which is independent of any specific public parameters. However, achieving accumulators from q-DHE vector commitments is not trivial when focus on hiding the index. Thus, we need to slightly modify the VC scheme to achieve an accumulator that allows to compute witnesses in a way that they do not reveal the index of the message. We present our q-DHE type (randomizable) accumulator below.

5.1 (Randomizable) Accumulator based on the q-DHE VC

Here, we introduce a (randomizable) q-DHE Structure-Preserving Accumulator (q-DHE SPA) in which the proof (or witness in this context) discloses no information beyond the message itself—not even its index. By leveraging the vector commitment scheme outlined in Scheme 3, we can make minor adjustments to the construction to hide indices i.e.,. by hiding the index, we mean that the message is not bound or tied to any specific position or element in pp.

To achieve this, we randomize the elements that reveal the indices. Specifically, the message M_i and the associated tags can disclose the index through the relations $e(M_i, \hat{B}_{j-i}) = e(T_{ij}, \hat{P})$ and $e(M_i, \hat{P}) = e(M_{i0}, \hat{B}_i)$ (i.e., the message/tag verification needs public elements $\hat{B}_{j-i}$). To address this, we use a random value $y \in \mathbb{Z}_p$ and replace the verification equation with the following: $e(M_i, \hat{Y}) = e(T_{ij}^y, \hat{P})$ and $e(M_i, \hat{P}^y) = e(M_{i0}, \hat{B}_i^y)$, where $\hat{B}_{j-i}^y$ is applied for all j, and $\hat{\mathbf{Y}} = (\hat{P}^y, \hat{B}_i^y)_{i \in [q]}$ is included as part of the tag. In the subsequent step, the verification equation $e(C_1, \hat{B}_{\ell+1-i}) = e(\pi_i, \hat{P}) \cdot e(M_i, \hat{B}_{\ell+1-i})$ is initially designed to confirm the positions of the messages with respect to $\hat{B}_{\ell+1-i}$. We randomize this verification by picking a new random $\rho \in \mathbb{Z}_p$, modifying the equation to $e(C_1, \hat{W}_2) = e(\pi_i, \hat{P}) \cdot e(M_i, \hat{W}_2)$, where $\hat{W}_2 = (\hat{B}_{\ell+1-i})^\rho$ is part of the witness. By using the bilinear pairing property, we can appropriately randomize π_i with ρ, thereby ensuring that the verification remains valid as intended. We present the complete construction as follows:

Scheme 4 (q-DHE Accumulator) *Our q-DHE* SPA *is defined as follows:*

- $\mathsf{Setup}(1^\lambda, q)$: Given a security parameter λ and a parameter q, run $\mathsf{pp} \leftarrow \mathsf{SPVC.KeyGen}(1^\lambda, q)$ and set $\mathsf{pk}_{acc} = \mathsf{pp}$ and $\mathsf{sk}_{acc} = \perp$.
- $\mathsf{GenTag}(\mathsf{pk}_{acc}, \mathcal{X})$: Given a public key pk_{acc} and a set $\mathcal{X}$. For each $M_i \in \mathcal{X}$, where $M_i = (\mathbf{M}_i, \mathbf{T}_i)$, first check the following conditions hold for (i.e., the messages are correctly generated) for all i:

$$e(M_i, \hat{B}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{P}) = e(M_{i0}, \hat{B}_i).$$

  If the conditions hold, pick a random $y \leftarrow\!\!\$\ \mathbb{Z}_p$ and update $\mathbf{T} = \mathbf{T}^y$. Then, compute $\hat{\mathbf{Y}} = (\hat{Y}_0 = \hat{P}^y, \hat{Y}_i = \hat{B}_i^y)_{i\in[q]}$ such that: $e(M_i, \hat{Y}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{Y}_0) = e(M_{i0}, \hat{Y}_i)$. Finally, output the updated messages/tags $(\mathbf{M}, \mathbf{T}, \hat{\mathbf{Y}})$.
- $\mathsf{Eval}(\mathsf{pk}_{acc}, \mathcal{X})$: Given a public key pk_{acc} and set $\mathcal{X}$ it returns an accumulator $\mathrm{acc}_{\mathcal{X}}$ together with the aux as follows: Check $M_i = (\mathbf{M}, \mathbf{T}, \hat{\mathbf{Y}}) \in \mathcal{X}$ s.t. $e(M_i, \hat{Y}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{Y}_0) = e(M_{i0}, \hat{Y}_i)$, then run $(C, \mathsf{aux}) \leftarrow \mathsf{SPVC.Commit}(M_1, \ldots, M_\ell)$ and for a random $\rho \in \mathbb{Z}_p$ set:

$$\mathsf{Acc}_{\mathcal{X}} = C = \left(P^r \cdot \prod_{i\in[q]} M_{i1} \right) \ \wedge\ \mathsf{aux} = (r, \rho)$$

- $\mathsf{WitCreate}(\mathsf{pk}_{acc}, \mathsf{Acc}_{\mathcal{X}}, \mathsf{aux}, M_i)$: This algorithm takes a key pair pk_{acc}, an accumulator $\mathsf{Acc}_{\mathcal{X}}$, auxiliary information aux, and a value M_i. It returns $\perp$, if $M_i \notin \mathcal{X}$ or $e(M_i, \hat{Y}_{j-i}) \neq e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{Y}_0) \neq e(M_{i0}, \hat{Y}_i)$, otherwise, compute a witness wit_{M_i} for M_i: Pick $\rho \in \mathsf{aux}$, run $\pi_i \leftarrow \mathsf{SPVC.Open}((M, i, \mathsf{aux}))$ and compute

$$wit_{M_i} = \left(W_1 = \left(B_{\ell+1-i}^r \cdot \prod_{j\neq i, k=\ell+2-i+j} T_{jk} \right)^\rho \wedge \hat{W}_2 = (\hat{P}^{\alpha^{\ell+1-i}})^\rho \right)$$

- $\mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}_{\mathcal{X}}, wit_{M_i}, M_i)$: This algorithm takes a public key pk_{acc}, an accumulator $\mathsf{Acc}_{\mathcal{X}}$, a witness wit_{Mi}, and a value M_i. It returns *true* (i.e., it out- puts 1) if the message and tag are created correctly $e(M_i, \hat{Y}_{j-i}) = e(T_{ij}, \hat{P}) \wedge e(M_{i1}, \hat{Y}_0) = e(M_{i0}, \hat{Y}_i)$ and If wit_{Mi} is a valid witness for $M_i \in \mathcal{X}$ and *false* (i.e., it outputs 0) otherwise.

$$e(C_1, \hat{W}_2) = e(W_1, \hat{P}) \cdot e(M_i, \hat{W}_2)$$

- $\mathsf{Rand}(\mathsf{Acc}_{\mathcal{X}}, \pi_i, M_i, (\mu, \beta, \gamma)) \rightarrow (\mathsf{Acc}'_{\mathcal{X}}, \pi'_i, M'_i)$: On input an accumulator $\mathsf{Acc}_{\mathcal{X}}$, witness π_i, message M_i and randomness (μ, β, γ), compute a randomized accumulator and witness for a randomized message M'_i with $(\beta, \mu, \gamma) \in \mathbb{Z}_p$ as:

$$\mathsf{Acc}'_{\mathcal{X}} = \mathsf{Acc}^\mu_{\mathcal{X}} \wedge \pi'_i = (W_1^{\mu\gamma}, \hat{W}_2^\gamma) \wedge M'_i = (\mathbf{M}^\mu, \mathbf{T}^{\mu\beta}, \hat{\mathbf{Y}}^\beta).$$

  This is valid accumulator-witness pair for the randomized message. Finally, aux is updated with (μ, β, γ).

Note that one can also randomize only the witness without randomizing the messages or the accumulator, i.e., by computing $W_1^\mu, \hat{W}_2^\mu$ as randomized witnesses.

Reducing Trust in CRS. The bilinear pairing-based construction typically requires either public parameters generated in a trusted setup, which are linear in the number of elements added to the Acc, or a trusted party with a trapdoor to compute it. Trust in parameter generation can be reduced or removed using MPC protocols, such as [16]. Alternatively, Groth et al. [55] proposed updatable reference strings, which allow any party to update them securely, as demonstrated in Ethereum's 'powers of tau' ceremony [75].

Remark 3. We note that two additional properties, indistinguishability and zero-knowledge (ZK), were introduced in [37] and [15], respectively. However, for the primary applications we are targeting, these properties are not relevant, and thus we do not consider them in this work. In fact, in the context of ring signatures, accumulators do not need to be hiding (indistinguishable), making the inclusion of randomness r in the accumulator seem unnecessary. Nevertheless, we retain it here as it might be useful for other applications in the future, or for achieving properties like ZK, which are not directly required in our current setting. Moreover, if hiding the set is not a concern, the accumulator's randomness can always be provided alongside the set.

Theorem 4. *If the q-DHE Vector Commitment in Scheme 3 is position-binding, then the q-DHE Accumulator in Scheme 4 is collision resistant.*

Proof. Let $\mathcal{A}$ be an adversary against the collision resistance property of Scheme 4. We show how to build an equally efficient adversary $\mathcal{B}$ against the position-binding property of the vector commitment. $\mathcal{B}$ receives as input the parameters pp, and sets $\mathsf{pk} = \mathsf{pp}$ and $\mathsf{sk} = \perp$ and send pk to $\mathcal{A}$. Next, on input of $\mathcal{X}$, $\mathcal{B}$ first creates their tags using a random y as mentioned in GenTag: $(\mathbf{M}, \mathbf{T}, \hat{\mathbf{Y}})$) for all i. Then $\mathcal{B}$ computes accumulator $C = (P^r \cdot \prod_{i=1} M_i)$. Notice that $\mathcal{B}$ can easily answer all witness queries by computing π_i^ρ and $\hat{W}_2 = (\hat{P}^{\alpha^{\ell+1-i}})^\rho$ for the random ρ, where π_i is the same as vector commitment opening. Indeed both the accumulator C and π are the same as vector commitment and opening. At some point, $\mathcal{A}$ halts and hands to $\mathcal{B}$ a tuple $(wit'_i = (W'_1, \hat{W}'_2), M'_i)$. Notice that, being wit'_i valid in order to break collision-resistance it must hold hat:

$$\mathsf{Acc.Verify}(\mathsf{pk}_{acc}, wit', \mathrm{acc}_{\mathcal{X}}, M'_i) \wedge M'_i \notin \mathcal{X},$$

Now the reduction $\mathcal{B}$ can derandomize $(\hat{W}'_2)^{1/\rho}$, and this should be equal to $\hat{P}^{\alpha^{n+1-i}}$; if not, it returns $\perp$. Similarly, it derandomizes the tag as well, $\mathbf{T} = \mathbf{T}'^{1/y}$, to obtain $(M'_i, B_{j-i}) = (T_{ij}, \hat{P})$. It is clear that $(W'_1)^{1/\rho}$ and M'_i are now valid proofs for our vector commitment. Moreover, since $\mathsf{Acc}_{\mathcal{X}}$ was created by $\mathcal{B}$, it knows $\pi_i = \mathrm{Wit}_i$ for a M_i. This means the tuple $(\mathrm{acc}_{\mathcal{X}} = C, i, \pi_i, \pi'_i = \mathrm{Wit}'_i = (\hat{W}'_2)^{1/\rho})$ will contradict the position-binding property of the underlying vector commitment. □

Definition 12 (Perfect Randomization of Acc). *An* Acc *scheme provides perfect randomization if for all* λ*, for all* $\mathsf{pk}_{acc} \in \mathsf{Setup}(1^\lambda)$*, for all* $(\mathsf{pk}_{acc}, M_i, \mathsf{Acc}_{\mathcal{X}}, \mathsf{aux}, \pi)$*, if* $\mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}_{\mathcal{X}}, \pi_i, M_i) = 1$*, then* $(\mathsf{Acc}'_{\mathcal{X}}, \pi'_i, M'_i) \leftarrow \mathsf{Rand}(\mathsf{Acc}, \pi_i, M_i)$ *outputs uniformly random elements in the accumulator space* $(\mathsf{Acc}'_{\mathcal{X}}, \pi'_i, M'_i) \approx (\mathsf{Acc}_{\mathcal{X}}, \pi_i, M_i)$ *such that* $\mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}'_{\mathcal{X}}, \pi', M'_i)$*.*

Theorem 5. *The q-DHE Vector Commitment in Scheme 3 is perfectly randomizable.*

Proof. We provide proof of perfect randomization of accumulator as follows: Let $(\mathbf{M}, \mathbf{T}, \hat{\mathbf{Y}}) \in (\mathbb{G}_1^*)^2 \times (\mathbb{G}_1^*)^q \times (\mathbb{G}_2^*)^q$, $\mathsf{pk}_{acc} \in (\mathbb{G}_1^*)^{2n-1} \times (\mathbb{G}_2^*)^q$ and $\alpha \in \mathbb{Z}_p^*$. An accumulator, witness and messages/tags $(M_i, \mathsf{Acc}_{\mathcal{X}}, wit)$ satisfy $\mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}_{\mathcal{X}}, \pi_i, M_i) = 1$ is of the form:

$$\mathsf{Acc} = \left(P^r \cdot \prod_{i \in [q]} M_i \right), wit = \left(W_1 = (B^r_{\ell+1-i} \cdot \prod_{\substack{j \neq i \\ k=\ell+2-i+j}} T_{jk})^\rho, \hat{W}_2 = (\hat{P}^{\alpha^{\ell+1-i}})^\rho \right)$$

$$and \quad (\mathbf{M}, \mathbf{T}, \hat{\mathbf{Y}})$$

For randomness $(\mu, \beta, \gamma) \in \mathbb{Z}_p^*$, $\mathsf{Rand}(\mathsf{Acc}, \pi_i, M_i, (\mu, \beta, \gamma))$ outputs:

$$\mathsf{Acc}' = \left(P^{r\mu} \cdot \prod_{i \in [q]} M_i^\mu \right), wit' = \left(W_1 = (B^{r\mu \cdot \gamma}_{\ell+1-i} \cdot \prod_{\substack{j \neq i \\ k=\ell+2-i+j}} T_{jk}^{\mu\gamma})^\rho, \hat{W}_2 = (\hat{P}^{\alpha^{\ell+1-i}})^{\rho \cdot \gamma} \right)$$

$$and \quad M'_i = (\mathbf{M}^\mu, \mathbf{T}'^{\mu\beta}, \hat{\mathbf{Y}}^\beta)$$

which are uniformly random elements conditioned on $\mathsf{Verify}(\mathsf{pk}_{acc}, \mathsf{Acc}^\mu_{\mathcal{X}}, wit', (\mathbf{M}^\mu, \mathbf{T}'^{\mu\beta}, \hat{\mathbf{Y}}^\beta)) = 1$. Indeed, each element is perfectly randomized with fresh randomness, so it is clear that Rand and Eval are identically distributed for all $(\mathsf{Acc}', wit'_i, M'_i)$. □

6 Applications

In this section, we describe how our SPVC and SPA primitives can lead to interesting applications. As the main application, we present the first constant-size ring signature in the dlog setting. The size of the signature consists of only a few group elements, and the computation involves only group exponentiation operations along with an efficient Schnorr NIZK for a simple AND statement. We also informally discuss some other potential applications for blockchains and demonstrate how these can result in significant efficiency improvements in blockchain applications.

6.1 Constant-Size Ring Signatures

We take inspiration from the framework in [39], which employs a key-homomorphic signature and a NIWI/NIZK for an OR relation (giving linear sized signatures), with the approach introduced in [40] that employs an accumulator for the membership proof in the ring. In particular, [40] rely on an RSA accumulator and so far no construction in the discrete logarithm setting following this approach is known. Basically, the idea is to use an accumulator to compactly represent all public keys in the ring and to use a NIZK proof to demonstrate knowledge of a membership witness in this accumulator as well as the corresponding secret key, e.g., via an explicit signature.

We can achieve a constant-size ring signature in the discrete logarithm setting by integrating these two approaches—using an accumulator with a key-homomorphic signature and leveraging NIZK proofs.

We first show a variant of the BLS signature that seamlessly integrates with the message space of our accumulator. Then by leveraging the key-homomorphic property of BLS, as demonstrated in [39], we are able to construct highly efficient ring signatures. To make the scheme more compatible with BLS, we assume $M_{i0} = \mathsf{pk}_i$ and include M_{i1} in the tag, i.e., $M_{i1} = T_{i1}$.

Scheme 5 (Accumulator-Compatible BLS Scheme $\mathsf{BLS}_{\mathsf{acc}}$) $\mathsf{BLS}_{\mathsf{acc}}$ *with respect to q-DHE* SPA *accumulator* Acc *is defined by the following algorithms:*

- $\mathsf{Setup}(1^\lambda)$: Run $\mathsf{pk}_{acc} \leftarrow \mathsf{Acc.Setup}(1^\lambda, q = 2)$, choose a hash function $H : M \to \mathbb{G}_2$, output $\mathsf{pp} := (\mathsf{BG}, H, P^{\alpha^1}, P^{\alpha^2}, P^{\alpha^4}; \hat{P}^{\alpha^1}, \hat{P}^{\alpha^2}; g_t^{\alpha^3})$, where $\mathsf{pk}_{acc} = (P^{\alpha^1}, P^{\alpha^2}, P^{\alpha^4}; \hat{P}^{\alpha^1}, \hat{P}^{\alpha^2}; g_t^{\alpha^3})$.
- $\mathsf{KeyGen}(\mathsf{pp})$: Parse pp, choose $x \leftarrow\!\!\$\ \mathbb{Z}_p$, set $\mathsf{pk} = P^x$, $\mathsf{sk} = x$, and return $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{GenTag}(\mathsf{pp}, (\mathsf{sk}, \mathsf{pk}))$: Choose random $y \leftarrow\!\!\$\ \mathbb{Z}_p$ and output a tagged public-key $(\mathsf{pk}, \tau = (\mathbf{T}, \hat{\mathbf{Y}}))$ by computing: $\mathbf{T} = ({B_1}^{\mathsf{sk}\cdot y}, {B_2}^{\mathsf{sk}\cdot y}) \wedge \hat{\mathbf{Y}} = (\hat{Y}_0 = \hat{P}^y, \hat{Y}_1 = \hat{B}_1^y)$ such that $e(T_1, \hat{Y}_1) = e(T_2, \hat{P}) \wedge e(T_1, \hat{Y}_0) = e(\mathsf{pk}, \hat{B}_1^y)$.
- $\mathsf{Sign}(\mathsf{sk}, m)$: Compute and return signature as $\sigma \leftarrow H(m)^{\mathsf{sk}}$.
- $\mathsf{Verify}(\mathsf{pk}, \tau, m, \sigma)$: Parse pk as $(\mathsf{pk}, \tau = (\mathbf{T}, \hat{\mathbf{Y}}))$. Return 1 if the following holds and 0 otherwise:

$$e(P^x, H(m)) = e(P, \sigma) \wedge e(T_1, \hat{Y}_1) = e(T_2, \hat{P}) \wedge e(T_1, \hat{P}^y) = e(\mathsf{pk}, \hat{B}_1^y)$$

Theorem 6 (EUF-CMA Security of $\mathsf{BLS}_{\mathsf{acc}}$). *If the BLS signature scheme is EUF-CMA secure, then the accumulator-based variant* $\mathsf{BLS}_{\mathsf{acc}}$ *is also EUF-CMA secure.*

Proof (Sketch). To prove the EUF-CMA security of $\mathsf{BLS}_{\mathsf{acc}}$, we obtain the public key pk from the BLS challenger. Using the trapdoor α that we can freely choose, we generate the corresponding tag and provide it to the adversary. For each of the adversary's queries, we forward them to the BLS challenger. Thus, any forgery in the $\mathsf{BLS}_{\mathsf{acc}}$ scheme corresponds directly to a forgery in the BLS scheme.

Key-randomization and signature adaption $\mathsf{BLS}_{\mathsf{acc}}$. We require a functionality to randomize keys and adapt signatures to randomized keys akin to the key-homomorphic property of signatures [39]. This ensures that a signature σ on m under pk can be adapted to a signature under pk', with a well-defined relationship between pk and pk'. We also need that adapted signatures are perfectly indistinguishable from freshly generated ones. We can use the multiplicative approach for BLS in [30] for randomizing keys/tags and adapting the signature:

- $\mathsf{Adapt}(\mathsf{pk}, m, \sigma, (\mu, \beta))$: Let $(\mu, \beta) \in \mathbb{Z}_p$ and $\mathsf{pk} = P^x$. Return adapted key and signature $(\mathsf{pk}' = \mathsf{pk}^\mu, \sigma' = \sigma^\mu)$ as well as tags $(\hat{\mathbf{Y}}' = \hat{\mathbf{Y}}^\beta, \mathbf{T}' = \mathbf{T}^{\beta\mu})$.

It follows that the adapted signatures σ' are indistinguishable from freshly generated signatures under the public key pk'. Moreover, the randomized tags maintain identical distribution, such that $(\hat{\mathbf{Y}}^\beta, \mathbf{T}^{\beta\mu}) \approx (\hat{\mathbf{Y}}, \mathbf{T})$. Now we are ready to present our ring signature:

Our Construction. In Figure 1, we present our construction of ring signatures based on the $\mathsf{BLS}_{\mathsf{acc}}$ signature scheme as well as our q-DHE SPA accumulator Acc and a simulation-extractable non-interactive zero-knowledge proof system NIZK, which we instantiate with Schnorr proofs made non-interactive via the Fiat-Shamir heuristic [41].

The basic idea of the scheme is as follows: We first slightly modify the interface of $\mathsf{GenTag}(\mathsf{pp}, (\mathsf{sk}, \mathsf{pk}), i)$ and adopt an indexed approach where each user i is assigned the related part of pp. For each

user i, the corresponding public parameter B_i is used to generate their tag. Due to the use of our accumulator, we impose an upper bound on the ring size, ensuring that each user takes a designated portion of the CRS to generate their keys/tags (see Sec. 4.2 for how a message/tag, and in this case a key/tag, can be generated). Second, the users generate an accumulator for the ring $\mathcal{R}$ denoted as $\mathsf{Acc}_{\mathcal{R}}$ and a witness for their public key $\mathsf{pk} \in \mathcal{R}$ denoted as wit, along with a $\mathsf{BLS}_{\mathsf{acc}}$ signature on the message m under the pk. They then randomize the signature and pk using the Adapt algorithm of $\mathsf{BLS}_{\mathsf{acc}}$ with parameters (μ, β) to produce (pk, σ') and also randomize the witness and accumulator to obtain $(\mathsf{Acc}'_{\mathcal{R}}, wit')$. Next, they use the same (μ, β) along with an additional random value γ to randomize the accumulator and witness. Finally, they utilize a non-interactive zero-knowledge proof NIZK to demonstrate that the accumulator has been correctly randomized and that they possess the secret key corresponding to their randomized public key pk'.

- $\mathsf{Setup}(1^\lambda)$: Run $\mathsf{pp} \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Setup}(1^\lambda, q)$ and output public parameters pp.
- $\mathsf{KeyGen}(\mathsf{pp}, i)$: Run $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{KeyGen}(\mathsf{pp})$ and $\tau_i \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{GenTag}(\mathsf{pp}, (\mathsf{sk}_i, \mathsf{pk}_i), i)$ and output a keypair and its tag $((\mathsf{sk}_i, \mathsf{pk}_i), \tau_i)$. For simplicity we assume that τ_i is included as part of the pk_i so $\mathsf{pk}_i = (\mathsf{pk}_i, \tau_i)$.
- $\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_i, m, \mathcal{R})$: Choose randomness $(\mu, \beta, \gamma) \in \mathbb{Z}_p$ and do:
  - $\delta \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Sign}(\mathsf{sk}_i, m||\mathcal{R})$ // $\mathsf{BLS}_{\mathsf{acc}}$ signature on $m||\mathcal{R}$
  - $(\mathsf{Acc}_{\mathcal{R}}, \mathsf{aux}) \leftarrow \mathsf{Acc}.\mathsf{Eval}(\mathsf{pp}, \mathcal{R})$ // q-DHE SPA accumulator
  - $wit_{\mathsf{pk}_i} \leftarrow \mathsf{Acc}.\mathsf{WitCreate}(\mathsf{pp}, \mathsf{Acc}_{\mathcal{R}}, \mathsf{aux}, \mathsf{pk}_i)$ // witness for pk_i
  - $(\delta', \mathsf{pk}'_i, \tau') \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}(\mathsf{pk}_i, m, \delta, (\mu, \beta))$ // randomization
  - $(\mathsf{Acc}'_{\mathcal{R}}, wit', \mathsf{pk}'_i) \leftarrow \mathsf{Acc}.\mathsf{Rand}(\mathsf{Acc}_{\mathcal{R}}, wit, \mathsf{pk}_i, (\mu, \beta, \gamma))$ // randomization
  - $\pi = \mathsf{NIZK}\{(\mathsf{sk}' = \mathsf{sk} \cdot \mu, \mu) : \mathsf{Acc}'_{\mathcal{R}} = \mathsf{Acc}^{\mu}_{\mathcal{R}} \wedge \mathsf{pk}' = P^{\mathsf{sk}'}\}$ // NIZK proof

  Finally, output $\sigma = ((\mathsf{Acc}_{\mathcal{R}}, r), \mathsf{Acc}'_{\mathcal{R}}, wit', \delta', \mathsf{pk}', \pi)$.
- $\mathsf{Verify}(\mathsf{pp}, m, \sigma, \mathcal{R})$: Given pp, message $m \in \mathcal{M}$, signature $\sigma = (\delta, wit, \mathsf{Acc}_{\mathcal{R}})$, and ring $\mathcal{R}$, output 1 if the following holds and 0 otherwise:

$$\mathsf{Acc}.\mathsf{Verify}(\mathsf{pp}', \mathsf{Acc}'_{\mathcal{R}}, wit', \mathsf{pk}') = 1 \wedge \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Verify}(\mathsf{pk}', \tau', m||\mathcal{R}, \delta') := 1 \wedge$$

$$\mathsf{Acc}.\mathsf{Eval}(\mathsf{pp}, \mathcal{R}, r) == \mathsf{Acc} \wedge \pi \text{ is correct}$$

Note that a verifier might also recompute the accumulator locally with the knowledge of r and one can then omit to send it explicitly.

Fig. 1. Constant size ring signature in dlog setting (Σ is $\mathsf{BLS}_{\mathsf{acc}}$ and Acc is the accumulator in Sec. 5.1)

Theorem 7. *Let* $\mathsf{BLS}_{\mathsf{acc}}$ *be unforgeable, and providing adaptability of signatures,* NIZK *be simulation extractable, and q-DHE* SPA *provide collision resistance and perfect randomization, then Scheme 1 is unforgeable (Def. 9), and anonymous (Def.10).*

Roughly, unforgeability holds because, given a valid forgery of a ring signature, one can always extract a valid $\mathsf{BLS}_{\mathsf{acc}}$ signature from one of the ring members by extracting randomness form the NIZK proof which represents a valid $\mathsf{BLS}_{\mathsf{acc}}$ forgery. Anonymity is ensured by the perfect randomization of the accumulator, combined with perfect adaptation of $\mathsf{BLS}_{\mathsf{acc}}$, which guarantees that signatures from different ring members (or the same member) are indistinguishable.

Proof. We now present the formal proof of unforgeability and then anonymity.

Unforgeability. We construct a reduction $\mathcal{B}$ against $\mathsf{BLS}_{\mathsf{acc}}$ using $\mathcal{A}$ against our ring signature construction. Note that we can extract the witness (sk' and μ) from the NIZK proof and assume this will only fail with negligible probability. The challenger of $\mathsf{BLS}_{\mathsf{acc}}$ will be denoted by $\mathcal{C}$.

Initially, $\mathcal{B}$ receives from $\mathcal{C}$ values (pp) and the challenge public-key (pk, τ). We now guess an index $j \in [n]$ where $n = \mathsf{poly}(\lambda)$ is the number of users in the system. We set $\mathsf{pk}_j := \mathsf{pk}$ and generate all the other remaining $\mathsf{BLS}_{\mathsf{acc}}$ keys pk_i for $i \in [n], i \neq j$ on our own. In the following we assume that $\mathcal{A}$ does not query the oracle $\mathsf{Key}(\cdot)$ for index j. Otherwise, we abort the game and this happens with probability $1/n$. Now $\mathcal{B}$ gives pp and $\{\mathsf{pk}_i\}_{i \in [n]}$ to $\mathcal{A}$. In the following we discuss how $\mathcal{B}$ handles queries from $\mathcal{A}$.

$\mathsf{Key}(i)$: return sk_i and set $Q_{\mathsf{Key}} := Q_{\mathsf{Key}} \cup \{i\}$.

$\mathsf{Sig}(i, m, \mathcal{R})$: if $i \neq j$ we just compute the signature as in Sign. Otherwise we query $m||\mathcal{R}$ to $\mathcal{C}.\mathsf{Sign}$ and obtain a δ. We run $(\mathsf{Acc}_{\mathcal{R}}, \mathsf{aux}) \leftarrow \mathsf{Acc.Eval}(\mathsf{pp}, \mathcal{R})$ and $wit_{\mathsf{pk}_i} \leftarrow \mathsf{Acc.WitCreate}(\mathsf{pp}, \mathsf{Acc}_{\mathcal{R}}, \mathsf{aux}, \mathsf{pk}_i)$. Then randomize the accumulator and signature $(\delta', \mathsf{pk}_i', \tau') \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}(\mathsf{pk}_i, m, \delta, (\mu, \beta))$ and $(\mathsf{Acc}_{\mathcal{R}}', wit', \mathsf{pk}_i') \leftarrow \mathsf{Acc.Rand}(\mathsf{Acc}_{\mathcal{R}}, wit, \mathsf{pk}_i, (\mu, \beta, \gamma))$ for uniform randomness $(\mu, \beta, \gamma) \in \mathbb{Z}_p$ and finally we simulate the proof such that: $\pi = \mathsf{NIZK}\{(\mathsf{sk}', \mu) : \mathsf{Acc}_{\mathcal{R}}' = \mathsf{Acc}_{\mathcal{R}}^{\mu} \wedge \mathsf{pk}' = P^{\mathsf{sk}'}\}$, where $\mathsf{sk}' = \mathsf{sk} \cdot \mu$. It sets $Q^{\mathsf{Sig}} := Q^{\mathsf{Sig}} \cup (i, m, \mathcal{R})$ and outputs $\sigma = ((\mathsf{Acc}_{\mathcal{R}}, r), \mathsf{Acc}_{\mathcal{R}}', wit', \delta', \mathsf{pk}', \pi)$.

Eventually, $\mathcal{A}$ outputs a forgery $(m^*, \sigma^*, \mathcal{R}^*)$ and now by the validity criteria we know that $(\cdot, m^*, \mathcal{R}^*) \notin Q^{\mathsf{Sig}}$. Now, we extract (sk', μ) and check whether $\mathsf{pk}^{\mu} = \mathsf{pk}'^*$, i.e., whether the forgery is with respect to the j'th user. Otherwise, we abort the game and this happens with probability $1/n$. If we do not abort we note that we can compute $(\delta, \mathsf{pk}_j, \cdot) \leftarrow \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}(\mathsf{pk}'^*, m^*, \delta', (\mu^{-1}, 1))$ and output δ as a forgery for message $m^*||\mathcal{R}^*$ to $\mathcal{C}$, which completes the proof. □

Proof of Anonymity. We show that a simulation of the anonymity game for $b = 0$ is indistinguishable from a simulation of the anonymity game with $b = 1$.

- **Game 0:** The anonymity game Anon_b (Def. 10) with $b = 0$.
- **Game 1:** As in **Game 0**, except that the experiment runs Anon_b as follows: NIZK proofs in Sign are simulated.
  [**Game 0** → **Game 1**] : By the perfect zero-knowledge property of NIZK, we have $\Pr[S_1] = \Pr[S_0]$.
- **Game 2:** As in Game 1, but we now sample μ by choosing μ' uniformly at random and use $\mu = \frac{\mu' \cdot sk_1}{sk_0}$. μ' is stored internally.
  [**Game 1** → **Game 2**] : This is a perfect hop, as μ is still uniformly random.
- **Game 3:** We now switch to the (public and secret) keys for $b = 1$. Furthermore, we use μ' on behalf of μ.
  [**Game 2** → **Game 3**] : Looking at $\mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}$, we have that:

$$\begin{aligned}\mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}\ & (\mathsf{pk}_0, m||\mathcal{R}, \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Sign}(\mathsf{sk}, m||\mathcal{R}), (\mu, \beta)) = \\ & (pk_0^{\mu}, H(m||\mathcal{R})^{\mathsf{sk}_0\mu}, \hat{\mathbf{Y}}_0^{\beta}, \mathbf{T}_0^{\beta\mu}) \approx \\ & (pk_1^{\mu'}, H(m||\mathcal{R})^{sk_1\mu'}, \hat{\mathbf{Y}}_1^{\beta}, \mathbf{T}_1^{\beta\mu'}) \\ & \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}(\mathsf{pk}_1, m||\mathcal{R}, \mathsf{BLS}_{\mathsf{acc}}.\mathsf{Sign}(\mathsf{sk}, m||\mathcal{R}), (\mu', \beta))\end{aligned}$$

  To see the second equation, note that the second and the last element (corresponding to the signatures and $\mathbf{T}$) are uniquely determined by the other two elements, and thus do not need to be considered when arguing the closeness of the distributions. For the first element (corresponding to the public keys), it is easy to see that the change is purely syntactical, as $\mathsf{pk}_0^{\mu} = \mathsf{pk}_1^{\mu'}$. For the third element (corresponding to the $\hat{\mathbf{Y}}$), the distribution is identical for $\hat{\mathbf{Y}}_0$ and $\hat{\mathbf{Y}}_1$. Given that μ and β are independent, this also holds true for the joint distribution. The argument for $\mathsf{Acc.Rand}$ is identical to that of $\mathsf{BLS}_{\mathsf{acc}}.\mathsf{Adapt}$ and is as follows:

$$\begin{aligned}\mathsf{Acc.Rand}\ & (\mathsf{Acc}_{\mathcal{X}}, \pi_0, pk_0, (\mu, \beta, \gamma)) = \\ & (\mathsf{Acc}_{\mathcal{X}}^{\mu}, (W_{10}^{\mu\gamma}, \hat{W}_{20}^{\gamma}), (pk_0^{\mu}, \mathbf{T_0}^{\mu\beta}, \hat{\mathbf{Y}}_{\mathbf{0}}^{\beta})) \approx \\ & (\mathsf{Acc}_{\mathcal{X}}^{\mu'}, (W_{11}^{\mu'\gamma}, \hat{W}_{21}^{\gamma}), (pk_1^{\mu'}, \mathbf{T_1}^{\mu'\beta}, \hat{\mathbf{Y}}_{\mathbf{1}}^{\beta})) \approx \\ & \mathsf{Acc.Rand}(\mathsf{Acc}_{\mathcal{X}}, \pi_1, pk_1, (\mu', \beta, \gamma))\end{aligned}$$

  Thus together we obtain:
  $$\Pr[S_3] = \Pr[S_2].$$
- **Game 4:** Same as Game 3, except that we switch back to the honest computation of NIZK. This now corresponds to the anonymity game for $b = 1$. Again using the perfect zero-knowledge property, we get:
  $$\Pr[S_4] = \Pr[S_3].$$

Putting the individual game hops together, we obtain a transition which is valid and preserves the probabilities of success:

$$\Pr[S_4] = \Pr[S_0].$$

□

Comparison with Related Work. Looking at the recent systematization of knowledge (SoK) by Chator et al. [29], we observe several constant-size ring signatures. The work in [20], which is based on composite order groups, remains inefficient at around 100 group elements and with computations being a factor of roughly 50 times slower than in prime oder groups [43].[6] The approach by Malavolta and Schröder in [69] for building ring signatures from signatures with re-randomizable keys, when instantiated with their variant of Hofheinz and Kiltz signatures secure under the q-strong Diffie-Hellman assumption and the Groth16 zk-SNARK [54], yields constant size signatures of size $4\mathbb{G}_1 + 2\mathbb{G}_2 + \mathbb{Z}_p$. A popular pairing-friendly curve for the type-3 setting with 128 bit security is the BLS12-381 curve [14], where elements in $\mathbb{Z}_p$, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ require 32, 48, 92, and 576 bytes, respectively. For the construction in [69] this would give around 3300 bits. However, while the size is very low, using a zk-SNARK and in particular constant-sized ones such as Groth16 (or the recent Polymath [68], which is even more compact) comes with drawbacks. First they are non-universal and thus require a trusted setup per circuit (here ring size, unless one fixes an upper bound) and the concrete proving times will be significant. The signer needs to prove the verification equation of the signature scheme among $|\mathcal{R}|$ keys, involving the evaluation of a programmable hash function. In the non discrete-logarithm setting, the RSA-accumulator based ring signatures by Dodis et al. in [40] gives the most efficient solution in the factorization setting and requires around $\approx 38,500$ bits - significantly larger than in the dlog setting.

The Concrete Size of Our RS: The efficiency of our scheme can be evaluated in terms of communication costs and the bit size of the involved elements. The total communication cost is approximately $8\mathbb{G}_1 + 4\mathbb{G}_2 + 2\mathbb{Z}_p$, which corresponds to about 6500 bits for the BLS12-381 curve (without using point compression). Specifically, the elements in $\mathbb{G}_1$ consist of two elements for $\mathbf{T}$, one element for the public key pk, two for the accumulator $(\mathsf{Acc}', \mathsf{Acc})$, one for witness W_1, and two $\mathbb{G}_1$ and $\mathbb{Z}_p$ elements for the NIZK proof. In addition, the elements in $\mathbb{G}_2$ include two elements for $\hat{\mathbf{Y}}$, one element for signature δ, and one for witness $\hat{W}_2$.

Thus, our construction provides the first entirely practical and concretely efficient constant-size ring signature in the dlog setting without using heavy machinery such as SNARKs to prove bilinear group arithmetic.

6.2 Succinct Data Availability Sampling

Data availability sampling addresses a major blockchain challenge—scalability [9, 57]. This approach allows light clients to verify the availability and integrity of block data using multi-dimensional Reed-Solomon codes within an erasure coding strategy. Ethereum has shifted from fraud proofs, as highlighted in [9], to validity proofs based on polynomial commitments, leveraging their homomorphic properties. As a result, Ethereum aims to integrate this mechanism into its sharding protocol[7]. In this setting, each blockchain validator, similar to light clients, only needs to store the commitment instead of the full dataset and proof. However, within this multi-dimensional structure, clients with limited resources must store a tuple $\mathsf{com} = (C_1, \ldots, C_n)$, where each C_i is a KZG commitment [59] to a Reed-Solomon code (tensor code) [57]. We think this application can benefit from our SPVC schemes by reducing the commitment size (and, consequently, the communication size) from 256 elements per block to just one, allowing client storage to decrease from 256 KZG commitments per shard to a single commitment. This is achieved by treating the commitment com of a block of data as a single SPVC commitment.

In our approach, one can create a commitment to these KZG commitments for a more succinct representation. Using our weak binding SPVC, this can be achieved without significant modifications to the KZG scheme or the SPVC itself. Also for q-DHE PSVC, we can easily extend the public parameters of KZG with those received from the q-DHE PSVC to compute KZG commitments in different bases. For example, assume we have $(B_1, B_2, \hat{B}_1, \hat{B}_2)$ alongside $(P^{x^i})_{i\in[q]}$, where x is the KZG trapdoor. We need to compute $(B_1^{x^i})_{i\in[q]}$ and $(B_2^{x^i})_{i\in[q]}$ to derive a KZG commitment with respect to our SPVC message space.

[6] While there are approaches to convert such constructions to prime-order bilinear groups [43], this will not yield to anything near practical.

[7] https://notes.ethereum.org/@vbuterin/protodankshardingfaq

6.3 Algebraic Verkle Trees

In a stateless blockchain client model, nodes do not store the entire state but rely on "witness"—compact proofs that verify the necessary state for transaction validation. Verkle Trees[8], which improve upon Merkle Trees by using VCs at the leaf nodes, enable smaller witness sizes and more efficient verification.

From a theoretical point of view, a key challenge has been the inability to commit to commitments within the same algebraic structure due to the lack of structure-preserving VCs. Algebraic Verkle Trees (AVTs) with the WSPVC resolve this issue, enabling seamless commitment to commitments without switching between cryptographic primitives like hash functions. As already mentioned, WSPVC is sufficient for stateless validation as commitments are always honestly produced through (Byzantine) agreement on a sequence of updates. This can potentially simplify verification and reduce witness sizes. AVTs also expand the Verkle Tree in both depth and width while maintaining constant-size proofs by incrementally committing to VC commitments. Moreover, we can efficiently prove knowledge of a message without needing to prove the pre-image of a hash in a SNARK.

We believe that the use of SPVC in AVTs can pave the way for future research, particularly in optimizing scalability by integrating structure-preserving VCs into frameworks, which could lead to more efficient techniques in stateless blockchain.

7 Conclusion and Future Work

In this paper, we demonstrate that strictly structure-preserving compressing primitives can be realized. We present the first strictly structure-preserving commitment that is shrinking, and in particular, constant-size. By employing a more structured message space—specifically, a variant of the DH message space—we circumvent existing impossibility results. As our main contribution, we construct structure-preserving vector commitments (SPVC) and accumulators (SPA). We begin by discussing generic constructions and then provide concrete implementations under the Diffie-Hellman Exponent assumption. Finally, to showcase the practicality of our constructions, we present various applications. Our main application is the first entirely practical constant-size ring signature scheme in bilinear groups (i.e., the discrete logarithm setting).

We consider our work a significant first step toward developing structure-preserving compressing primitives, serving as a foundation for further exploration in this area. While our scheme offers valuable insights, the development of a fully-featured, unrestricted solution remains an open challenge for future research. One interesting direction for future work is designing a vector commitment scheme that utilizes a more natural message space—such as messages as simple as P^m —instead of relying on a CRS. Moreover, further exploration of applications for our schemes appears to be a promising research direction.

Acknowledgments

Omid Mir and Stephan Krenn were supported by the European Union's Horizon Europe project SUNRISE (grant agreement no. 101073821), PREPARED, a project funded by the Austrian security research programme KIRAS of the Federal Ministry of Finance (BMF), and the Austrian Science Fund (FWF) project number I 6650-N (REMINDER).

References

1. Abe, M.: Structure-preserving cryptography. Invited Talk at Asiacrypt 2015, https://www.iacr.org/archive/asiacrypt2015/94520356/94520356.pdf
2. Abe, M., Chow, S.S.M., Haralambiev, K., Ohkubo, M.: Double-trapdoor anonymous tags for traceable signatures. In: Lopez, J., Tsudik, G. (eds.) ACNS 11. LNCS, vol. 6715, pp. 183–200 (Jun 2011). https://doi.org/10.1007/978-3-642-21554-4_11
3. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236 (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_12

[8] Vitalik Buterin, Guillaume Ballet, Dankrad Feist. Verkle tree EIP, 2021.

4. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. Journal of Cryptology **29**(2), 363–421 (Apr 2016). https://doi.org/10.1007/s00145-014-9196-7
5. Abe, M., Groth, J., Kohlweiss, M., Ohkubo, M., Tibouchi, M.: Efficient fully structure-preserving signatures and shrinking commitments. Journal of Cryptology **32**(3), 973–1025 (Jul 2019). https://doi.org/10.1007/s00145-018-9300-5
6. Abe, M., Haralambiev, K., Ohkubo, M.: Group to group commitments do not shrink. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 301–317 (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_19
7. Abe, M., Kohlweiss, M., Ohkubo, M., Tibouchi, M.: Fully structure-preserving signatures and shrinking commitments. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 35–65 (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_2
8. Acar, T., Nguyen, L.: Revocation for delegatable anonymous credentials. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 423–440 (Mar 2011). https://doi.org/10.1007/978-3-642-19379-8_26
9. Al-Bassam, M., Sonnino, A., Buterin, V.: Fraud and data availability proofs: Maximising light client security and scaling blockchains with dishonest majorities. arXiv preprint arXiv:1809.09044 (2018)
10. Albrecht, M.R., Cini, V., Lai, R.W.F., Malavolta, G., Thyagarajan, S.A.K.: Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 102–132 (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_4
11. Au, M.H., Tsang, P.P., Susilo, W., Mu, Y.: Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 295–308 (Apr 2009). https://doi.org/10.1007/978-3-642-00862-7_20
12. Au, M.H., Wu, Q., Susilo, W., Mu, Y.: Compact e-cash from bounded accumulator. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 178–195 (Feb 2007). https://doi.org/10.1007/11967668_12
13. Bari, N., Pfitzmann, B.: Collision-free accumulators and fail-stop signature schemes without trees. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 480–494 (May 1997). https://doi.org/10.1007/3-540-69053-0_33
14. Barreto, P.S.L.M., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 02. LNCS, vol. 2576, pp. 257–267 (Sep 2003). https://doi.org/10.1007/3-540-36413-7_19
15. Barthoulot, A., Blazy, O., Canard, S.: Cryptographic accumulators: new definitions, enhanced security, and delegatable proofs. In: International Conference on Cryptology in Africa. pp. 94–119. Springer (2024)
16. Ben-Sasson, E., Chiesa, A., Green, M., Tromer, E., Virza, M.: Secure sampling of public parameters for succinct zero knowledge proofs. In: 2015 IEEE Symposium on Security and Privacy. pp. 287–304. IEEE Computer Society Press (May 2015). https://doi.org/10.1109/SP.2015.25
17. Benabbas, S., Gennaro, R., Vahlis, Y.: Verifiable delegation of computation over large datasets. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 111–131 (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9_7
18. Benaloh, J.C., de Mare, M.: One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In: Helleseth, T. (ed.) EUROCRYPT'93. LNCS, vol. 765, pp. 274–285 (May 1994). https://doi.org/10.1007/3-540-48285-7_24
19. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. Journal of Cryptology **22**(1), 114–138 (Jan 2009). https://doi.org/10.1007/s00145-007-9011-9
20. Bose, P., Das, D., Rangan, C.P.: Constant size ring signature without random oracle. In: Foo, E., Stebila, D. (eds.) ACISP 15. LNCS, vol. 9144, pp. 230–247 (Jun / Jul 2015). https://doi.org/10.1007/978-3-319-19962-7_14
21. Camenisch, J., Kohlweiss, M., Soriente, C.: An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 481–500 (Mar 2009). https://doi.org/10.1007/978-3-642-00468-1_27
22. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76 (Aug 2002). https://doi.org/10.1007/3-540-45708-9_5
23. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 410–424 (Aug 1997). https://doi.org/10.1007/BFb0052252
24. Campanelli, M., Fiore, D., Greco, N., Kolonelos, D., Nizzardo, L.: Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 3–35 (Dec 2020). https://doi.org/10.1007/978-3-030-64834-3_1

25. Campanelli, M., Fiore, D., Han, S., Kim, J., Kolonelos, D., Oh, H.: Succinct zero-knowledge batch proofs for set accumulators. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 455–469. ACM Press (Nov 2022). https://doi.org/10.1145/3548606.3560677
26. Campanelli, M., Nitulescu, A., Ràfols, C., Zacharakis, A., Zapico, A.: Linear-map vector commitments and their practical applications. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 189–219 (Dec 2022). https://doi.org/10.1007/978-3-031-22972-5_7
27. Catalano, D., Fiore, D.: Vector commitments and their applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 55–72 (Feb / Mar 2013). https://doi.org/10.1007/978-3-642-36362-7_5
28. Catalano, D., Fiore, D., Gennaro, R., Giunta, E.: On the impossibility of algebraic vector commitments in pairing-free groups. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part II. LNCS, vol. 13748, pp. 274–299 (Nov 2022). https://doi.org/10.1007/978-3-031-22365-5_10
29. Chator, A., Green, M., Tiwari, P.R.: Sok: Privacy-preserving signatures. Cryptology ePrint Archive (2023)
30. Cini, V., Ramacher, S., Slamanig, D., Striecks, C., Tairi, E.: Updatable signatures and message authentication codes. In: Garay, J. (ed.) PKC 2021, Part I. LNCS, vol. 12710, pp. 691–723 (May 2021). https://doi.org/10.1007/978-3-030-75245-3_25
31. Couteau, G., Hartmann, D.: Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 768–798 (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_27
32. Couteau, G., Lipmaa, H., Parisella, R., Ødegaard, A.T.: Efficient NIZKs for algebraic sets. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part III. LNCS, vol. 13092, pp. 128–158 (Dec 2021). https://doi.org/10.1007/978-3-030-92078-4_5
33. Crites, E.C., Kohlweiss, M., Preneel, B., Sedaghat, M., Slamanig, D.: Threshold structure-preserving signatures. In: ASIACRYPT 2023, Part II. pp. 348–382. LNCS (Dec 2023). https://doi.org/10.1007/978-981-99-8724-5_11
34. Crites, E.C., Lysyanskaya, A.: Delegatable anonymous credentials from mercurial signatures. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 535–555 (Mar 2019). https://doi.org/10.1007/978-3-030-12612-4_27
35. Damgård, I., Triandopoulos, N.: Supporting non-membership proofs with bilinear-map accumulators. Cryptology ePrint Archive, Report 2008/538 (2008), https://eprint.iacr.org/2008/538
36. Derler, D., Hanser, C., Slamanig, D.: A new approach to efficient revocable attribute-based anonymous credentials. In: Groth, J. (ed.) Cryptography and Coding - 15th IMA International Conference, IMACC 2015, Oxford, UK, December 15-17, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9496, pp. 57–74. Springer (2015). https://doi.org/10.1007/978-3-319-27239-9_4, https://doi.org/10.1007/978-3-319-27239-9_4
37. Derler, D., Hanser, C., Slamanig, D.: Revisiting cryptographic accumulators, additional properties and relations to other primitives. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 127–144 (Apr 2015). https://doi.org/10.1007/978-3-319-16715-2_7
38. Derler, D., Ramacher, S., Slamanig, D.: Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10786, pp. 419–440. Springer (2018). https://doi.org/10.1007/978-3-319-79063-3_20, https://doi.org/10.1007/978-3-319-79063-3_20
39. Derler, D., Slamanig, D.: Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge. DCC **87**(6), 1373–1413 (2019). https://doi.org/10.1007/s10623-018-0535-9
40. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in ad hoc groups. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 609–626 (May 2004). https://doi.org/10.1007/978-3-540-24676-3_36
41. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the Fiat-Shamir transform. In: Galbraith, S.D., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 60–79 (Dec 2012). https://doi.org/10.1007/978-3-642-34931-7_5
42. Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 152–168 (Aug 2005). https://doi.org/10.1007/11535218_10
43. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 44–61 (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_3
44. Fuchsbauer, G.: Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320 (2009), https://eprint.iacr.org/2009/320
45. Fuchsbauer, G.: Commuting signatures and verifiable encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 224–245 (May 2011). https://doi.org/10.1007/978-3-642-20465-4_14

46. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 233–253 (Aug 2015). https://doi.org/10.1007/978-3-662-48000-7_12
47. Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. Journal of Cryptology **32**(2), 498–546 (Apr 2019). https://doi.org/10.1007/s00145-018-9281-4
48. Ghadafi, E.: More efficient structure-preserving signatures - or: Bypassing the type-iii lower bounds. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10493, pp. 43–61. Springer (2017). https://doi.org/10.1007/978-3-319-66399-9_3, https://doi.org/10.1007/978-3-319-66399-9_3
49. Ghadafi, E.: Further lower bounds for structure-preserving signatures in asymmetric bilinear groups. In: Buchmann, J., Nitaj, A., eddine Rachidi, T. (eds.) AFRICACRYPT 19. LNCS, vol. 11627, pp. 409–428 (Jul 2019). https://doi.org/10.1007/978-3-030-23696-0_21
50. Ghosh, E., Ohrimenko, O., Papadopoulos, D., Tamassia, R., Triandopoulos, N.: Zero-knowledge accumulators and set algebra. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 67–100 (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_3
51. Goldreich, O.: Foundations of cryptography: Volume 1, basic tools, 2001. Google Scholar Google Scholar Digital Library Digital Library **5**
52. Gorbunov, S., Reyzin, L., Wee, H., Zhang, Z.: Pointproofs: Aggregating proofs for multiple vector commitments. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 2007–2023. ACM Press (Nov 2020). https://doi.org/10.1145/3372297.3417244
53. Griffy, S., Lysyanskaya, A., Mir, O., Kempner, O.P., Slamanig, D.: Delegatable anonymous credentials from mercurial signatures with stronger privacy. Cryptology ePrint Archive, Paper 2024/1216, to appear at ASICACRYPT'24 (2024), https://eprint.iacr.org/2024/1216
54. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 305–326 (May 2016). https://doi.org/10.1007/978-3-662-49896-5_11
55. Groth, J., Kohlweiss, M., Maller, M., Meiklejohn, S., Miers, I.: Updatable and universal common reference strings with applications to zk-SNARKs. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 698–728 (Aug 2018). https://doi.org/10.1007/978-3-319-96878-0_24
56. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432 (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_24
57. Hall-Andersen, M., Simkin, M., Wagner, B.: Foundations of data availability sampling. Cryptology ePrint Archive (2023)
58. Jhanwar, M.P., Safavi-Naini, R.: Compact accumulator using lattices. Cryptology ePrint Archive, Report 2014/1015 (2014), https://eprint.iacr.org/2014/1015
59. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 177–194 (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8_11
60. Lai, R.W.F., Malavolta, G.: Subvector commitments with application to succinct arguments. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 530–560 (Aug 2019). https://doi.org/10.1007/978-3-030-26948-7_19
61. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 1–31 (May 2016). https://doi.org/10.1007/978-3-662-49896-5_1
62. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. Journal of Cryptology **36**(3), 23 (Jul 2023). https://doi.org/10.1007/s00145-023-09470-6
63. Libert, B., Peters, T., Joye, M., Yung, M.: Linearly homomorphic structure-preserving signatures and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 289–307 (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_17
64. Libert, B., Peters, T., Yung, M.: Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 296–316 (Aug 2015). https://doi.org/10.1007/978-3-662-48000-7_15
65. Libert, B., Ramanna, S.C., Yung, M.: Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D. (eds.) ICALP 2016. LIPIcs, vol. 55, pp. 30:1–30:14. Schloss Dagstuhl (Jul 2016). https://doi.org/10.4230/LIPIcs.ICALP.2016.30
66. Libert, B., Yung, M.: Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 499–517 (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2_30

67. Lipmaa, H., Parisella, R.: Set (non-)membership NIZKs from determinantal accumulators. pp. 352–374. LNCS (2023). https://doi.org/10.1007/978-3-031-44469-2_18
68. Lu, D., Yu, A., Kate, A., Maji, H.: Polymath: Low-latency MPC via secure polynomial evaluations and its applications. Cryptology ePrint Archive, Report 2021/978 (2021), https://eprint.iacr.org/2021/978
69. Malavolta, G., Schröder, D.: Efficient ring signatures in the standard model. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 128–157 (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_5
70. Merkle, R.C.: A digital signature based on a conventional encryption function. In: Pomerance, C. (ed.) CRYPTO'87. LNCS, vol. 293, pp. 369–378 (Aug 1988). https://doi.org/10.1007/3-540-48184-2_32
71. Mir, O., Bauer, B., Griffy, S., Lysyanskaya, A., Slamanig, D.: Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials. In: ACM CCS 2023. pp. 30–44. ACM Press (Nov 2023). https://doi.org/10.1145/3576915.3623203
72. Mir, O., Slamanig, D., Bauer, B., Mayrhofer, R.: Practical delegatable anonymous credentials from equivalence class signatures. In: The 23rd Privacy Enhancing Technologies Symposium. pp. 488–513. de Gruyter (2023)
73. Mitrokotsa, A., Mukherjee, S., Sedaghat, M., Slamanig, D., Tomy, J.: Threshold structure-preserving signatures: Strong and adaptive security under standard assumptions. In: PKC 2024, Part I. pp. 163–195. LNCS (May 2024). https://doi.org/10.1007/978-3-031-57718-5_6
74. Nguyen, L.: Accumulators from bilinear pairings and applications. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 275–292 (Feb 2005). https://doi.org/10.1007/978-3-540-30574-3_19
75. Nikolaenko, V., Ragsdale, S., Bonneau, J., Boneh, D.: Powers-of-tau to the people: Decentralizing setup ceremonies. pp. 105–134. LNCS (Jun 2024). https://doi.org/10.1007/978-3-031-54776-8_5
76. Nitulescu, A.: Sok: Vector commitments (2021), https://www.di.ens.fr/~nitulesc/files/vc-sok.pdf
77. Papamanthou, C., Shi, E., Tamassia, R., Yi, K.: Streaming authenticated data structures. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 353–370 (May 2013). https://doi.org/10.1007/978-3-642-38348-9_22
78. Peikert, C., Pepin, Z., Sharp, C.: Vector and functional commitments from lattices. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part III. LNCS, vol. 13044, pp. 480–511 (Nov 2021). https://doi.org/10.1007/978-3-030-90456-2_16
79. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565 (Dec 2001). https://doi.org/10.1007/3-540-45682-1_32
80. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266 (May 1997). https://doi.org/10.1007/3-540-69053-0_18
81. Slamanig, D.: Dynamic accumulator based discretionary access control for outsourced storage with unlinkable access - (short paper). In: Keromytis, A.D. (ed.) FC 2012. LNCS, vol. 7397, pp. 215–222 (Feb / Mar 2012). https://doi.org/10.1007/978-3-642-32946-3_16
82. Tomescu, A., Abraham, I., Buterin, V., Drake, J., Feist, D., Khovratovich, D.: Aggregatable subvector commitments for stateless cryptocurrencies. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20. LNCS, vol. 12238, pp. 45–64 (Sep 2020). https://doi.org/10.1007/978-3-030-57990-6_3
83. Wee, H., Wu, D.J.: Succinct vector, polynomial, and functional commitments from lattices. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part III. LNCS, vol. 14006, pp. 385–416 (Apr 2023). https://doi.org/10.1007/978-3-031-30620-4_13