# On the Impossibility of Algebraic Vector Commitments in Pairing-Free Groups

Dario Catalano[1], Dario Fiore[2], Rosario Gennaro[3], and Emanuele Giunta[2,4]

[1] University of Catania, Italy.
catalano@dmi.unict.it
[2] IMDEA Software Institute, Madrid, Spain.
{dario.fiore, emanuele.giunta}@imdea.org
[3] Protocol Labs.
rosario.gennaro@protocol.ai
[4] Universidad Politecnica de Madrid, Spain.

**Abstract.** Vector Commitments allow one to (concisely) commit to a vector of messages so that one can later (concisely) open the commitment at selected locations. In the state of the art of vector commitments, *algebraic* constructions have emerged as a particularly useful class, as they enable advanced properties, such as stateless updates, subvector openings and aggregation, that are for example unknown in Merkle-tree-based schemes. In spite of their popularity, algebraic vector commitments remain poorly understood objects. In particular, no construction in standard prime order groups (without pairing) is known.

In this paper, we shed light on this state of affairs by showing that a large class of concise algebraic vector commitments in pairing-free, prime order groups are impossible to realize.

Our results also preclude any cryptographic primitive that implies the algebraic vector commitments we rule out, as special cases. This means that we also show the impossibility, for instance, of succinct polynomial commitments and functional commitments (for all classes of functions including linear forms) in pairing-free groups of prime order.

# Table of Contents


**1 Introduction** **3**
- 1.1 Our Results . . . 4
- 1.2 Our Techniques . . . 5
- 1.3 Interpretation of our impossibility and further implications . . . 7
- 1.4 Related Work . . . 7
- 1.5 Organization of the paper . . . 8

**2 Preliminaries** **9**
- 2.1 Vector Commitments . . . 9
- 2.2 Digital Signatures . . . 10

**3 Algebraic Vector Commitments** **10**
- 3.1 Generic Transformation from VCs to Signatures . . . 11
- 3.2 $\vartheta$-Unforgeability . . . 12

**4 Algebraic Signatures** **14**
- 4.1 Attack to Schemes with Strictly Linear Verification . . . 15
- 4.2 Attack to Schemes with Generic Verification . . . 21

**5 Conclusions** **22**
- 5.1 Impossibility of Algebraic Vector Commitments . . . 22
- 5.2 Impossibility of Algebraic Signatures . . . 25

**A More Preliminaries** **28**
- A.1 Linear Algebra . . . 28
- A.2 Min-Entropy . . . 29
- A.3 Arithmetic circuits and R1CS . . . 30
- A.4 Generic Group Model . . . 31

**B Examples and Constructions** **32**
- B.1 Linear vs Generic Verification for Signatures . . . 32
- B.2 Secure Signatures with Linear Verification . . . 33
- B.3 Succinct Vector Commitments with Linear Verification . . . 35
- B.4 Tightness of Algebraic VC Lower Bounds . . . 38
- B.5 Tightness of Algebraic Signatures Lower Bounds . . . 40

**C Postponed Proofs** **41**
- C.1 Generic Transformation to Signatures . . . 41
- C.2 Attack to Schemes with Strictly Linear Verification . . . 42
- C.3 Attack to Schemes with Generic Verification . . . 46

## 1 Introduction

Vector commitments [LY10, CF13] (VC) are a class of commitment schemes that allow a sender to commit to a vector $\mathbf{v}$ of $n$ messages, in such a way that she can later open the commitment at selected positions. Namely, the sender can convince anyone that the $i$-th message in the committed vector is $v_i$. A secure scheme shall satisfy *position binding*, i.e. generating valid openings to different values $v_i \neq v_i'$ for the same position $i$ is computationally infeasible.

The distinguishing feature of vector commitments is that commitments and openings must be *succinct*. In the original notion of [LY10, CF13], this means that their size is independent of $n$, the length of the vector, but a relaxed notion allowing a logarithmic dependence in $n$ may be considered, as in the case of the celebrated Merkle tree construction [Mer88].

Mainly thanks to their succinctness property, vector commitments have been shown to be a useful building block in several applications, such as zero-knowledge sets [MRK03, LY10, CF13], verifiable databases [BGV11, CF13], succinct arguments [Kil94, Mic94, BBF19, LM19], proofs of retrievability [JK07, Fis18], and stateless blockchains [CPZ18, BBF19].

Analyzing the state of the art of VC schemes, we see that VC constructions are based on two main approaches.

On one side, we have tree-based VCs, notably Merkle trees [Mer88] and their generalizations [Kus18]. These constructions have the advantage of being realizable from collision resistant hash functions, and thus can be based on the hardness of virtually any cryptographic problem including factoring, discrete logarithm, SIS and many more. In fact, we notice that VCs with logarithmic-size openings are *equivalent* to collision-resistant hash functions. The main drawback of tree-based schemes is that their openings are of size $O(\log n)$. Additionally, the tree-based approach seems to inherently impede the realization of properties such as subvector openings [BBF19, LM19] and aggregation [CFG+20], that turn useful in both theoretical and practical applications of VCs.

On the other side, we have *algebraic* vector commitments, notably based on bilinear pairings [LY10, KZG10, CF13], groups of unknown order [CF13], and lattices [PSTY13, PPS21]. Roughly speaking, an algebraic VC is one in which the commitment and verification algorithm only use algebraic operations over the group that underlies the construction (this rules out hashing group elements for example). The main advantage of these constructions is that they admit openings of constant size,[5] that are virtually optimal – a single group element in most constructions. Moreover, algebraic schemes naturally achieve useful properties such as (additive) homomorphism, stateless updatability [CF13], subvector openings [BBF19, LM19] and aggregation [CFG+20]. Yet, the powerful versatility of existing VCs with constant-size openings contrasts with the limited theoretical understanding of their foundations.

[5] We include lattice-based schemes in the 'algebraic' category although they do not perfectly fit our notion of using a group in a black box way; also, existing schemes still need (poly) logarithmic-size openings.

We see two main open questions related to algebraic VCs. The first one concerns the minimal general assumption that implies them. While tree-based schemes with logarithmic openings are well understood, being de facto equivalent to collision-resistant hash functions[6], we have no generic recipe to build algebraic VCs with constant-size openings.[7] The second question is whether algebraic VCs can be built from "standard" prime-order groups without pairings. In this setting, known constructions rely either on the tree-based approach (e.g., building a Merkle tree on top of Pedersen hash function), or on inner-product arguments in the random oracle model [BCC$^+$16, BBB$^+$18]. Both these approaches entail logarithmic-size openings and a non-algebraic verification.

We believe that settling these two questions would improve our understanding of vector commitments. In this work, we focus on the second question for two important reasons: (i) on the theoretical front, studying algebraic VCs in this minimal setting helps us understand conceptually what are the "ingredients" needed to build them; (ii) on the practical side, pairing-free groups of known order are the simplest and most efficient cryptographic setting, and yet we know of no construction of algebraic VCs there.

Our results are negative: we show that a broad class of VC schemes in this setting cannot both be succinct and satisfy position binding.

### 1.1 Our Results

We informally call a vector commitment built on top of a group $\mathbb{G}$ of prime order $q$ "algebraic" if all its procedures use $\mathbb{G}$ in a black box way, i.e. without relying on the representation of group elements. We show the following two main results.

*Impossibility of algebraic VCs with linear verification.* We start by looking at the class of algebraic VC schemes in which the verification algorithm is a set of linear equations over $\mathbb{G}$. Specifically, for a message $m$ and position $i$ the verification consists of checking that

$$A(\mathbf{z}, m, i) \cdot \mathbf{X} \stackrel{?}{=} B(\mathbf{z}, m, i) \cdot \mathbf{Y} \tag{1}$$

where $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ are the group elements appearing respectively in the public parameters and the commitment, openings are of the form $(\mathbf{Y}, \mathbf{z})$ with $\mathbf{Y}$ being a vector of group elements and $\mathbf{z}$ of field elements, and $A, B$ are functions defining matrices with coefficients in $\mathbb{F}_q$.

We believe this to be the simplest and most natural form of verification using only group operations. However we show that whenever $A$ depends affinely on $\mathbf{z}, m$ and $B$ is independent from them (we say such a scheme has *strictly* linear

[6] A Merkle tree is a VC with logarithm openings that can be realized from any CRHF. Conversely, in any non trivial VC the commitment procedure has to be shrinking and collision resistant, from which CRHF can be built.

[7] The only generic construction with constant size opening is the folklore one that combines a hash function and a constant-size SNARK; yet this is non-algebraic due to the need of encoding the hash computation in the SNARK's constraint system.

verification), then it is impossible to achieve both position binding and succinctness. More specifically we prove that if a scheme has position binding, commitments of bit-length $\ell_c$ and opening proofs of bit-length $\ell_\pi$, then asymptotically their product is lower bounded by the length of the vector we are committing to, i.e. $\ell_c \cdot \ell_\pi = \Omega(n)$. Thus either $\ell_c = \Omega(\sqrt{n})$ or $\ell_\pi = \Omega(\sqrt{n})$. Interestingly, this family of schemes captures generalizations of Pedersen commitments [BG12] which, as we show in the Appendix B.4, achieve this lower bound.

Next, we investigate how crucial are our requirements on the dependence of $A(\cdot)$ and $B(\cdot)$ on $\mathbf{z}, m$. We show they are necessary. Indeed, if we allow either $A$ to depend quadratically, or $B$ affinely, on $\mathbf{z}, m$ then there exist succinct VC constructions whose verification can be written in the above form over a group $\mathbb{G}$. We provide examples in the Appendix B.3. The schemes we find however rely on arithmetization techniques to encode arbitrary circuits as constraint systems of degree 2 over a finite field [GGPR13]. This for instance means that, for proper choice of $A$ and $B$, it is possible to express, using an algebraic verification equation as (1), computations like the validity tests for a Merkle tree path, or any arbitrary VC verification algorithm.

Despite being secure and succinct, VC schemes built this way do not satisfactorily answer our question in a positive way, as they appear to bypass the underlying group as their source of hardness. Indeed, either their security comes from problems unrelated to $\mathbb{G}$, or if they depend on $\mathbb{G}$, they must do it in a non-black-box way[8].

*Impossibility of algebraic VCs with generic group verification.* Motivated by these findings, we investigate whether VCs can be built given *only* black-box access to a cryptographic group. To study this case, we just assume the VC (which we call *algebraic with generic verification*) to use the underlying group generically, without any further constraint on its verification procedure.

Eventually we provide a black-box separation in Maurer's Generic Group Model [Mau05]. This informally implies that any VC using $\mathbb{G}$ generically and whose position binding reduces to a hard problem in $\mathbb{G}$ (such as DLP or CDH) cannot be succinct, as it must hold $\ell_c \cdot \ell_\pi = \Omega(n)$.

### 1.2 Our Techniques

Our strategy to prove our impossibility results on algebraic vector commitments consists of two main steps. (A) We show that from a VC it is possible to construct a class of signature schemes. In particular, if the VC is algebraic with linear (resp. generic) verification, the resulting signature scheme's verification has analogous algebraic properties. (B) We prove the insecurity of this class of signature schemes in pairing-free groups of known order. To achieve the latter result we build on, and extend, the recent techniques of [DHH+21], that provide negative results for a somewhat smaller family of algebraic signatures.

In what follows we give an overview on each step.

[8] For example, one may consider a Merkle-tree of Pedersen commitments which must use the group representation to go from one level to another.

*From VCs to Signatures.* Given a VC scheme for vectors of length $n$ our transformation produces a signature scheme with polynomially bounded message space $\{1, \ldots, n\}$. In a nutshell, the public key is a commitment $c$ to a vector of $n$ random values $(s_1, \ldots, s_n)$. The signature on the message $i \in [n]$ is the pair $(s_i, \pi_i)$ where $\pi_i$ is the VC opening proof that $c$ opens to $s_i$ at position $i$. Verification simply runs the VC verification algorithm to check that the opening is valid.

Conveniently, this transformation maps algebraic VCs with linear/generic verification to signature schemes with the analogous property, which we then call algebraic signatures with linear/generic verification. This happens since the verification algorithm is essentially the same in both primitives.

The resulting signature however may not be proved existentially unforgeable if it comes from a VC satisfying only position binding. Indeed the latter property does not imply that every opening proof is hard to compute. However, assuming that the scheme is also succinct, an adversary who produces 'many' correct openings should have to correctly guess the value of several messages $s_i$ used to generate the commitment. This can be shown to be information-theoretically hard if the commitment and opening proofs provided have significantly smaller bit-length than the min-entropy of those messages.

For this reason we introduce a relaxed security notion, called $\vartheta$-unforgeability, where an adversary must provide not only one but at least more than $\vartheta$-many[9] forgeries for non-queried messages. Setting $\vartheta$ as a proper function of the number of queries made by the adversary, we prove that signatures from VCs are $\vartheta$-unforgeable.

*Impossibility of Algebraic Signatures, revisited.* To conclude our impossibility result for VCs, we finally provide an impossibility result and a black-box separation for algebraic signatures with strictly linear and generic verification respectively. In particular, we show in both cases that the message space in a $\vartheta$-unforgeable construction is upper-bounded by $n + \vartheta$ with $n$ being the number of group elements in the verification key. We also show this to be tight by providing a construction that achieves this bound in Appendix B.5.

Notice that similar results were already proved in [DHH+21]. In their work signatures are assumed to be of the form $(\mathbf{Y}, t)$ with $\mathbf{Y}$ a vector of group elements and $t \in \{0,1\}^\kappa$. Moreover the verification procedure is assumed to consist of a linear check as in Equation 1. For this class of signatures, which can be shown equivalent to our notion of *algebraic with linear verification*, they provide an attack running in time $O(2^\kappa \cdot \mathsf{poly}(\lambda))$.

Thus their adversary is efficient only when $t = O(\log \lambda)$, whereas our impossibility result applies to schemes with strictly linear verification, where signatures may contain several field elements. Likewise, their black-box separation only captures schemes with *linear verification*, while we extend it to signatures where all procedures are simply required to be generic. To show that this class of schemes is indeed more general we provide examples in Appendix B.1.

[9] Where $\vartheta$ may depend on the public parameters as well as the number of queries.

We finally stress that, as in [DHH+21], our results hold in Maurer's Generic Group Model [Mau05]. For a comparison with other models of generic computation, such as Shoup's Generic Group Model [Sho97], we refer to the discussion in [Zha22].

### 1.3 Interpretation of our impossibility and further implications

As mentioned earlier, both our impossibility results specify precise bounds and conditions under which VCs cannot be built generically in pairing-free groups. The bottom line is that, whenever a position-binding VC scheme uses the group in a black box way (and relies on it for security), then it cannot be succinct, which we recall is the distinguishing feature of this primitive.

Another interesting aspect of our impossibility results is that they imply analogous impossibilities for any primitive that allows one to construct algebraic VCs (with either strictly linear or generic-group verification) in pairing-free groups. Notably, our impossibility applies to polynomial commitments [KZG10], and functional commitments [LRY16] supporting any class of functions that includes projections, i.e., $C_i(\mathbf{v}) = v_i$ (already captured by linear forms). Indeed, each of these primitives allows one to build a VC with exactly the same succinctness and type of verification.[10] Therefore we obtain that any secure functional commitment or polynomial commitment using a pairing-free group in a black-box way cannot be succinct (or, more precisely, they must satisfy $\ell_c \cdot \ell_\pi = \Omega(n)$).

Our impossibility for algebraic signatures instead can be shown to imply analogous results for verifiable random functions [MRV99] and identity-based encryption [Sha84, BF01], the latter through the Naor-trick reduction, as observed in [DHH+21]. In this way our black-box separation for signatures yield a simpler argument for the tight result in [SGS21].

An interesting question left open by our work is understanding if our results can imply the impossibility of further cryptographic primitives via a connection to the classes of algebraic signatures and vector commitments that we rule out. Another open question concerns the minimal assumptions required to describe a VC with constant-size commitment and openings. We notice that our impossibility for VCs with generic verification holds in Maurer's generic group model [Mau05]. When using Shoup's GGM [Sho97], our results may not hold as one could use the group oracle as a random oracle [ZZ21], e.g., to build a Merkle tree of Pedersen hashes (see a similar discussion for signatures in [DHH+21]). However, to the best of our knowledge all these techniques would in the best case lead to schemes with logarithmic-size openings.

### 1.4 Related Work

The study of impossibility results about the construction of cryptographic primitives in restricted models is an important area of research that provides insights on the foundations of a cryptographic problem. Starting with the seminal paper

[10] These constructions are trivial/folklore and we do not elaborate further on them.

of Impagliazzo and Rudich [IR89], a line of works study the (in)feasibility of constructing cryptographic primitives in a black-box way from general assumptions, such as one-way functions or trapdoor permutations (e.g. [Sim98, KST99, GT00, GKM+00, GMR01, GGK03]).

Another line of works (more closely related to ours), initiated by Papakonstantinou, Rackoff and Vahlis [PRV12], considers the problem of proving impossibility of cryptographic primitives that make black-box use of a cryptographic group without pairings. Specifically, [PRV12] prove that identity-based encryption (IBE) algorithms built in this model of computation cannot be secure. Following [PRV12], more recent works study the impossibility, in generic group models for pairing-free groups of known order, of other cryptographic primitives, such as verifiable delay functions [RSS20], identity-based encryption (with a result tighter than [PRV12]) [SGS21] and signature schemes [DHH+21]. In addition to proving impossibility for algebraic signatures with generic-group algorithms, [DHH+21] also prove the generic impossibility of a class of algebraic signatures whose verification is a system of linear equations over a group.

In [SGS20], Schul-Ganz and Segev prove a lower bound on the number of group operations needed to verify batch membership proofs in accumulators that make black-box use of a cryptographic group. Their lower bound applies analogously to the verification of subvector openings in vector commitments. Despite the result and the techniques of [SGS20] differ from ours, both [SGS20] and our work show certain limitations of constructing VCs in prime order groups.

Finally, we mention the work of Abe, Haralambiev and Ohkubo [AHO12] that also considers a question related to constructing vector commitments. Following a research line on structure-preserving cryptography, Abe et al. [AHO12] investigate if it is possible to construct commitment schemes in bilinear groups in which messages, keys, commitments, and decommitments are elements of bilinear groups, and whose openings are verified by pairing product equations. For this class of schemes, they prove that the commitment cannot be shrinking. Implicitly this result also implies the impossibility of constructing succinct vector commitments in this structure-preserving setting in bilinear groups.

### 1.5 Organization of the paper

In Section 3 we define algebraic VCs and show our transformation to $\vartheta$-unforgeable signatures. Section 4 presents the definition of algebraic signatures and our impossibility results for strictly linear verification and generic group verification. Finally, in Section 5 we illustrate how to relate the parameters of our VC-to-signatures transformation with those needed by the impossibility of algebraic signatures. In Appendix B we present several constructions showing: the tightness of our impossibility, secure signatures and succinct vector commitment with linear, but not strictly linear, verification.

## 2 Preliminaries

**Notation.** We denote the security parameter by $\lambda$ and negligible functions with $\mathsf{negl}(\lambda)$. We say that an algorithm is $\mathsf{PPT}$ if it runs in probabilistic polynomial time. For a positive integer $n$, $[n]$ denotes the set $\{1, \ldots, n\}$. We use $(\mathbb{G}, +)$ to denote a group of known prime order $q$ with canonical generator $G$, and $\mathbb{F}_q$ for the field of order $q$. The identity (or zero) element is denoted as $0 \in \mathbb{G}$. Given a vector $\mathbf{x} \in \mathbb{F}_q^n$, we denote $\mathbf{x} \cdot G = (x_1 G, \ldots, x_n G)$.

$\mathbb{F}_q^{n,m}$ is the space of matrices $A$ with $m$ columns and $n$ rows and entries in $\mathbb{F}_q$. $\operatorname{rk} A$ denotes the rank of $A$, i.e. the maximum number of linearly independent rows. $A^\top$ is the transposed of $A$. All $\mathbf{x} \in \mathbb{F}_q^n$ are assumed to be column vectors, whereas row vectors are denoted as $\mathbf{x}^\top$.

In what follows 'GGM' stands for Maurer's Generic Group Model [Mau05] for a group of known prime order $q$. This model can be defined through two stateful oracles $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}^0_{\mathsf{eq}}$ such that: group element are labeled with progressively increasing indices, the first being associated to the canonical generator $G$, $\mathcal{O}_{\mathsf{add}}(X, Y)$ associate the next index to the element $X + Y$ and $\mathcal{O}^0_{\mathsf{eq}}(X)$ returns 1 if $X$ equals the identity element, 0 otherwise. See Appendix A.4 for more details.

### 2.1 Vector Commitments

We recall the definition of vector commitments from [CF13].

**Definition 1 (VC).** *A Vector Commitment scheme is a tuple of algorithms* $(\mathsf{VC.Setup}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy})$ *and a message space* $\mathsf{VC.M}$ *such that*

- $\mathsf{VC.Setup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$ *generates the public parameters.*
- $\mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n) \xrightarrow{\$} c, \mathsf{aux}$ *produce a commitment to* $m_1, \ldots, m_n \in \mathsf{VC.M}$ *together with some auxiliary information.*
- $\mathsf{VC.Open}(\mathsf{pp}, m, i, \mathsf{aux}) \xrightarrow{\$} \pi$ *return an opening proof that the* $i$*-th entry of a given commitment is* $m_i$*.*
- $\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 0/1$ *verifies the opening proof's correctness.*

We require a vector commitment scheme to satisfy *perfect correctness*, that is, given public parameters $\mathsf{pp} \xleftarrow{\$} \mathsf{VC.Setup}(1^\lambda)$, commitment $c, \mathsf{aux} \xleftarrow{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$ for any $m_i \in \mathsf{VC.M}$, and opening $\pi \xleftarrow{\$} \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$, it holds

$$\Pr\left[\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 1\right] = 1$$

Moreover, to avoid trivial cases, in this paper we assume $|\mathsf{VC.M}| \geq 2$.

The main security property for a vector commitments is the so called *position binding*, which informally states that no adversary can open the same position of a given commitment to two different values. Formally

**Definition 2 (Position binding).** *A vector commitment scheme satisfies position binding if for any* PPT *adversary* $\mathcal{A}$ *there exists a negligible function* $\varepsilon(\lambda)$ *such that*

$$\Pr\left[\begin{array}{l}\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 1 \\ \mathsf{VC.Vfy}(\mathsf{pp}, c, m', i, \pi') \to 1 \\ m \neq m'\end{array} \middle| \begin{array}{l}\mathsf{pp} \leftarrow^{\$} \mathsf{VC.Setup}(1^\lambda) \\ \mathcal{A}(\mathsf{pp}) \to (c, m, m', i, \pi, \pi')\end{array}\right] \leq \varepsilon(\lambda).$$

The property that distinguishes VCs from classical binding commitments is *succinctness* Following [LY10, CF13], a VC scheme is said succinct if there is a fixed $p(\lambda) = \mathsf{poly}(\lambda)$ such that for any $n$ the size of honestly generated commitments and openings is bounded by $p(\lambda)$. One may also consider weaker notions where the size may be bounded by $p(\lambda) \log n$ or $p(\lambda, \log n)$.

Since in our work we are interested in understanding the feasibility of VCs based on their level of succinctness, we consider a parametric notion. We say that a VC has succinctness $(\ell_c, \ell_\pi)$ if for any $m_1, \ldots, m_n \in \mathsf{VC.M}$, commitment $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$ and opening $\pi \leftarrow^{\$} \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$ for any $i \in [n]$, we have that $c$ (resp. $\pi$) has bit-length $\ell_c(\lambda, n)$ (resp. $\ell_\pi(\lambda, n)$).

### 2.2 Digital Signatures

**Definition 3.** *A signature scheme is a tuple of* PPT *algorithms* (S.Setup, S.Sign, S.Vfy) *and a message space set* S.M *such that*

- $\mathsf{S.Setup}(1^\lambda) \xrightarrow{\$} (\mathsf{sk}, \mathsf{vk})$ *generates the secret and verification keys*
- $\mathsf{S.Sign}(\mathsf{sk}, m) \xrightarrow{\$} \sigma$ *returns the signature of a message* $m \in \mathsf{S.M}$
- $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 0/1$ *verifies the signature* $\sigma$ *for a message* $m \in \mathsf{S.M}$

We further require a signature scheme to satisfy *perfect correctness*, meaning that if $(\mathsf{sk}, \mathsf{vk}) \leftarrow^{\$} \mathsf{S.Setup}(1^\lambda)$ and $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$ for any $m \in \mathsf{S.M}$ then the verification algorithm accepts always, i.e.

$$\Pr\left[\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1\right] = 1.$$

## 3 Algebraic Vector Commitments

In this paper we focus on vector commitments built on a pairing-free group of known order, using it in a black box way. We start by introducing a notion of algebraic vector commitments where the verification algorithm only consists of a system of linear equations.

**Definition 4 (Algebraic VCs with linear verification).** *A vector commitment scheme is said to be* algebraic with linear verification *if the message space is* $\mathsf{VC.M} = \mathbb{F}_q$ *and*

- $\mathsf{VC.Setup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$ *such that* $\mathsf{pp} = (\mathbf{X}_1, s_1) \in \mathbb{G}^{\nu_1} \times \{0,1\}^*$.
- $\mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n) \xrightarrow{\$} c, \mathsf{aux}$ *such that* $c = (\mathbf{X}_2, s_2) \in \mathbb{G}^{\nu_2} \times \{0,1\}^*$.

– $\mathsf{VC.Open}(\mathsf{pp}, m, i, \mathsf{aux}) \to \pi$ *such that* $\pi = (\mathbf{Y}, \mathbf{z})$ *with* $\mathbf{Y} \in \mathbb{G}^k$ *and* $\mathbf{z} \in \mathbb{F}_q^h$.
– *There exist* $A : \mathbb{F}_q^{h+1} \times [n] \times \{0,1\}^* \to \mathbb{F}_q^{\ell,n}$ *and* $B : \mathbb{F}_q^{h+1} \times [n] \times \{0,1\}^* \to \mathbb{F}_q^{\ell,k}$ *matrices such that* $\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 1$ *if and only if, calling* $\mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2$ *and* $s = s_1 || s_2$

$$A(\mathbf{z}, m, i, s) \cdot \mathbf{X} = B(\mathbf{z}, m, i, s) \cdot \mathbf{Y}.$$

For the ease of presentation we will omit $s$ in $A$ and $B$ when clear from the context. Notice that the definition imposes linearity only with respect to group elements while it allows procedures $A, B$ to depend non-linearly on the field vector element $\mathbf{z}$.

As we shall see, our first impossibility result states that whenever $A$ is an affine function of $\mathbf{z}, m$ and $B$ does not depends on $\mathbf{z}, m$, then the resulting scheme cannot be both "succinct" and position binding. We call these schemes *strictly* linear since their verification equations depend linearly both in $\mathbf{z}$ and $\mathbf{Y}$.

**Definition 5 (Algebraic VCs with strictly linear verification).** *A vector commitment is said to be algebraic with strictly linear verification if it satisfies Definition 4,* $A(\mathbf{z}, m, i)$ *is an affine function*[11] *of* $\mathbf{z}, m$ *and* $B(i)$ *does not depends on* $\mathbf{z}, m$.

However, if we allow $A$ to depend quadratically, or $B$ linearly, on $\mathbf{z}, m$ then we could use arithmetization techniques, such as R1CS, to encode a circuit representing for example a Merkle tree verification into the verification equation of Definition 4. This means that we can construct algebraic VC schemes with linear verification that are succinct and position binding. We formalize this result in Appendix B.3.

This technique however either bypasses the underlying group and may reduce security to external problems, or rely on non-black-box usage of the group. An example of the latter comes by encoding a Merkle tree built using an hash function whose collision resistance is based on discrete logarithm over the same group $\mathbb{G}$, such as Pedersen hash. Note that this construction would not retain algebraic properties from the underlying group. For this reason, following an approach similar to [PRV12, DHH+21], we study whether in the Generic Group Model (GGM) the security of a VC can be reduced to hard problems on the underlying group. To this aim we provide the following more general definition.

**Definition 6 (Algebraic VCs with generic verification).** *A vector commitment scheme is said to be* algebraic with generic verification *if, in the GGM, the algorithms* $\mathsf{VC.Setup}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy}$ *are oracle machines with access to* $\mathcal{O}_{\mathsf{add}}$ *and* $\mathcal{O}^0_{\mathsf{eq}}$.

### 3.1 Generic Transformation from VCs to Signatures

The strategy we adopt to show our impossibility results is to establish a connection between vector commitments and signatures, providing a way to construct

[11] i.e. $A(\mathbf{z}, m, i) = A_0(i) + z_1 A_1(i) + \ldots + z_h A_h(i) + m A_{h+1}(i)$

the latter from the former generically. This way we will be able to bridge extensions of the impossibility results in [DHH+21] for algebraic signatures to algebraic vector commitments.

More specifically, for a given VC (not necessarily algebraic) our transformation produces a signature scheme with polynomially bounded message space $\{1, \ldots, n\}$. The high-level idea is to compute a commitment $c$ to random messages $m_1, \ldots, m_n$, and use $(\mathsf{pp}, c)$ as the verification key and the auxiliary information $\mathsf{aux}$ as the secret key. In order to sign a message $i \in \{1, \ldots, n\}$, the signer returns $m_i$ and $\pi$, the message and opening proof for the $i$-th position, while verification is performed by checking the correctness of $\pi$. A formal description of the transformation is presented in Fig. 1.

$\mathsf{S}_{\mathsf{VC}}.\mathsf{Setup}(1^\lambda)$:

1 : $\mathsf{VC.Setup}(1^\lambda) \to \mathsf{pp}$
2 : $m_1, \ldots, m_n \leftarrow^{\$} \mathsf{VC.M}$
3 : $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$
4 : $\mathsf{vk} \leftarrow (\mathsf{pp}, c)$ $\mathsf{sk} \leftarrow (\mathsf{aux}, \{m_i\}_{i=1}^n)$
5 : Return $\mathsf{vk}, \mathsf{sk}$

$\mathsf{S}_{\mathsf{VC}}.\mathsf{Sign}(\mathsf{sk}, i)$:

1 : Parse $\mathsf{sk} = (\mathsf{aux}, \{m_i\}_{i=1}^n)$
2 : $\pi \leftarrow \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$
3 : $\sigma \leftarrow (m_i, \pi)$
4 : Return $\sigma$

$\mathsf{S}_{\mathsf{VC}}.\mathsf{Vfy}(\mathsf{vk}, i, \sigma)$:

1 : Parse $\mathsf{vk} = (\mathsf{pp}, c)$ and $\sigma = (m_i, \pi)$. Return $\mathsf{VC.Vfy}(\mathsf{pp}, c, m_i, i, \pi)$

**Fig. 1.** Generic transformation from VCs to signature schemes

### 3.2 $\vartheta$-Unforgeability

In terms of security the transformation in Fig. 1 fails in general to realize a UF-CMA-secure signature scheme. Informally, the problem is that position binding and succinctness do not imply, per se, that every opening proof is hard to compute, after having seen other openings. Indeed the latter property could be easily violated, for example, by a VC where $\mathsf{VC.Open}$ attaches to every opening the proof $(m_1, \pi_1)$ for position 1. Notice that one could modify any VC to do so without violating succinctness nor position binding. Yet starting from such a VC would allow an adversary to easily forge a signature for message 1 in the scheme in Fig. 1.

Observe that, informally, if the VC scheme were *hiding*, meaning that no information about messages in unopened positions is leaked, and $|\mathsf{VC.M}|$ is large enough, then the associated signature would be secure, since an adversary would have to guess the right message in the $i$-th position. This intuition can be extended to general VC assuming that the scheme is *succinct*. Indeed, even though

the commitment $c$ or its openings $\pi$ may leak information about unopened messages among $m_1, \ldots, m_n$, if their bit length is significantly smaller than $n$, no adversary can produce "too many" forgeries given only a few openings, as correctly guessing these message would be information-theoretically hard.

For this reason we introduce a relaxed notion of unforgeability for signatures, called $\vartheta$-unforgeability, which is enough for our purposes. In a nutshell, it requires a winning adversary to produce at least $\vartheta$ forgeries on *distinct* messages, with $\vartheta$ being a function of the queries performed and the public parameters. Next, using the intuition above, we prove that signature schemes obtained through the transformation in Fig. 1 satisfy this weaker notion.

**Definition 7 ($\vartheta$-UF).** *Given a function $\vartheta : \{0,1\}^* \to \mathbb{N}$ and a signature scheme we define the $\vartheta$-Unforgeability Experiment as in Fig. 2. The advantage of an adversary $\mathcal{A}$ is defined as*

$$\mathsf{Adv}^{\vartheta\text{-}\mathsf{UF}}(\mathcal{A}) \;=\; \Pr\left[\mathsf{Exp}^{\vartheta\text{-}\mathsf{UF}}_{\mathcal{A}} = 1\right].$$

*A scheme is $\vartheta$-Unforgeable if any* PPT *adversary has negligible advantage.*

$\mathsf{Exp}^{\vartheta\text{-}\mathsf{UF}}_{\mathcal{A}}$ **with adversary** $\mathcal{A}$:

1 : Initialize $Q \leftarrow \varnothing$, generate $\mathsf{sk}, \mathsf{vk} \leftarrow^{\$} \mathsf{S.Setup}(1^\lambda)$ and send $\mathcal{A} \leftarrow \mathsf{vk}$
2 : **When** $\mathcal{A} \to m \in \mathsf{S.M}$:
3 : Sign $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$, store $Q \leftarrow Q \cup (m, \sigma)$ and send $\mathcal{A} \leftarrow \sigma$
4 : **When** $\mathcal{A} \to F$:
5 : Return 1 if the following conditions are satisfied:
6 : For all $(m, \sigma) \in F$, the signature is correct, i.e. $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1$
7 : Messages in $F$ were not queried, i.e. $(m, \sigma) \in F \;\Rightarrow\; (m, \cdot) \notin Q$
8 : $|\{m \,:\, (m, \cdot) \in F\}| > \vartheta(\mathsf{vk}, Q)$
9 : Else return 0

**Fig. 2.** $\vartheta$-Unforgeability Experiment for a given signature scheme

To provide more intuition about this notion we observe that setting $\vartheta = 0$ yields the classic unforgeability under chosen message attacks (UF-CMA) [GMR88] security definition. For higher values of $\vartheta$ we obtain progressively weaker definitions until $\vartheta(\mathsf{vk}, Q) = |\mathsf{S.M}|$, which is trivially true for any scheme. The notion of $t$-time security is also captured by our definition setting

$$\vartheta(\mathsf{vk}, Q) \;=\; \begin{cases} 0 & \text{If } |Q| \leq t \\ |\mathsf{S.M}| & \text{If } |Q| > t \end{cases}$$

Finally we can show that a signature scheme obtained from a "succinct" VC satisfy this notion. A proof appears in Appendix C.1.

**Theorem 1.** *Given a Vector Commitment with commitments of bit-length $\ell_c = \ell_c(n, \lambda)$ and opening proofs of bit-length $\ell_\pi = \ell_\pi(n, \lambda)$, then there exists a* PPT *black box reduction $\mathcal{R}$ of $\vartheta$-UF for the derived signature scheme described in Fig. 1 to the position binding property, where*

$$\vartheta(\mathsf{vk}, Q) \;=\; \frac{\lambda + \ell_c + |Q| \cdot (\ell_\pi + \log |\mathsf{VC.M}|)}{\log |\mathsf{VC.M}|}.$$

*In particular for any position binding VC, the resulting signature is $\vartheta$-UF with $\vartheta$ as specified above.*

## 4 Algebraic Signatures

Having established a connection between VC and signatures we now provide the analogous of algebraic VC with (strictly) linear/generic verification in the signature setting. The first one is equivalent to the notion of *algebraic signature* in [DHH+21] and simply constrain the verification procedure to test a system of linear equations, albeit with a minor addition: as these signatures may come in our case from a VC, we split $\mathsf{S.Setup}$ in a CRS-generator $\mathsf{S.SetupCRS}$ which returns the public parameters (a list of group elements $\mathbf{X}_1$) and the actual key generation algorithm $\mathsf{S.SetupKey}(\mathbf{X}_1)$ which produces $\mathsf{vk}$ and $\mathsf{sk}$. Note there is no loss of generality assuming this structure as $\mathsf{S.SetupCRS}$ may return an empty vector which could then be ignored by $\mathsf{S.SetupKey}$.

**Definition 8.** *A signature scheme* $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ *is said to be algebraic with linear verification if*

- $\mathsf{S.Setup}$ *is divided into two algorithms* $\mathsf{S.SetupCRS}$ *and* $\mathsf{S.SetupKey}$ *such that* $\mathsf{S.SetupCRS}(1^\lambda) \xrightarrow{\$} (\mathbf{X}_1, s_1) \in \mathbb{G}^{n_1}$ *and* $\mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1) \xrightarrow{\$} \mathsf{sk}, \mathsf{vk}$ *with*

  $$\mathsf{vk} = (\mathbf{X}, s) \in \mathbb{G}^n \times \{0,1\}^* \quad : \quad \mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2, \qquad \mathbf{X}_2 \in \mathbb{G}^{n_2}, \qquad s = s_1 || s_2.$$

- $\mathsf{S.Sign}(\mathsf{sk}, m) \xrightarrow{\$} \sigma$ *where* $\sigma = (\mathbf{Y}, \mathbf{z})$ *with* $\mathbf{Y} \in \mathbb{G}^k$ *and* $\mathbf{z} \in \mathbb{F}_q^h$.
- *There exist* $A : \mathbb{F}_q^h \times \mathsf{S.M} \times \{0,1\}^* \to \mathbb{F}_q^{\ell,n}$ *and* $B : \mathbb{F}_q^h \times \mathsf{S.M} \times \{0,1\}^* \to \mathbb{F}_q^{\ell,k}$ *matrices such that* $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1$ *if and only if* $\sigma = (\mathbf{z}, \mathbf{Y})$ *and*

  $$A(\mathbf{z}, m, s)\mathbf{X} \;=\; B(\mathbf{z}, m, s)\mathbf{Y}.$$

*Furthermore the scheme is said to have strictly linear verification if $A(\mathbf{z}, m, s)$ is an affine function of $\mathbf{z}$ and $B(m, s)$ does not depend on $\mathbf{z}$.*

When clear from the context we will omit for clarity the argument $s$ in the matrices $A, B$ above. Next we provide an analogous for algebraic vector commitments with generic verification. As in the previous definition we split the setup algorithm into a procedure that prepares the CRS and another one that uses the CRS, oblivious to any trapdoor information about it, to compute the secret and verification keys.

**Definition 9.** *A signature scheme* $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ *is said to be algebraic with generic verification if, in the GGM, all algorithms have access to* $\mathcal{O}_{\mathsf{add}}$ *and* $\mathcal{O}^0_{\mathsf{eq}}$. *Furthermore we require* $\mathsf{S.Setup}$ *to be divided into two algorithms* $\mathsf{S.SetupCRS}$ *and* $\mathsf{S.SetupKey}$ *such that* $\mathsf{S.SetupCRS}(1^\lambda) \xrightarrow{\$} (\mathbf{X}_1, s_1) \in \mathbb{G}^{n_1} \times \{0,1\}^*$ *and* $\mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1) \xrightarrow{\$} \mathsf{sk}, \mathsf{vk}$ *with*

$$\mathsf{vk} = (\mathbf{X}, s) \in \mathbb{G}^n \times \{0,1\}^* \quad : \quad \mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2, \qquad \mathbf{X}_2 \in \mathbb{G}^{n_2}, \qquad s = s_1 || s_2.$$

### 4.1 Attack to Schemes with Strictly Linear Verification

We now provide an attack for algebraic signatures with strictly linear verification. The same notation of Definition 8 will be used below without further reference.

**Theorem 2.** *Given a signature scheme with strictly linear verification, for any* $\vartheta$ *polynomially bounded such that* $n_2 + \vartheta \leq |\mathsf{S.M}|$ *there exists a* $\mathsf{PPT}$ *algorithm* $\mathcal{A}$ *that in the unforgeability experiment in Fig. 2 performs at most* $n_2$ *queries and produces* $\vartheta$ *distinct forgeries with significant probability.*

*Proof.* For the sake of presentation we build $\mathcal{A}$ describing first a subroutine $\mathcal{B}$ which could break security by doing potentially more signing queries that $n_2$. Next, we show how $\mathcal{A}$ can use $\mathcal{B}$ in a black-box way to realize the full attack with $n_2$ queries.

Similarly to the attack described in [DHH+21], upon receiving the verification key $(\mathbf{X}, s)$, the subroutine $\mathcal{B}$ (described formally in Figure 3 ) keeps track of all possible exponents of $\mathbf{X}$ in an affine space $L \subseteq \mathbb{F}_q^n$. Then for each message $m_i$ either a forgery can be produced or a new condition on $\mathbf{X}$ is found, thus decreasing $\dim L$, at the cost of a signature query. This is done by checking if the system $A(\mathbf{z}, m)\mathbf{x} = B(m)\mathbf{y}$ can be solved for a given $\mathbf{z} \in \mathbb{F}_q^h$ and all $\mathbf{x} \in L$. More specifically we define $S(L, m)$, the *solutions* set, as the collection of all those $\mathbf{z}$ for which any $\mathbf{x} \in L$ makes the systems solvable, formally

$$S(L, m) \;=\; \{\mathbf{z} \in \mathbb{F}_q^h \;:\; A(\mathbf{z}, m) \cdot L \subseteq \operatorname{Im} B(m)\}.$$

If $S(L, m)$ is easy to compute, a strategy for $\mathcal{B}$ is to check whether $S(L, m) \neq \varnothing$ and in this case to get any $\mathbf{z} \in S(L, m)$ and find, using pseudo-inverses or Gaussian elimination, a vector $\mathbf{Y} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m)\mathbf{X} = B(m)\mathbf{Y}$. Conversely, if $S(L, m) = \varnothing$, $\mathcal{B}$ may request a signature $(\mathbf{Y}, \mathbf{z})$, which implies that the exponent $\mathbf{x}$ of $\mathbf{X}$ satisfies the condition $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$. Notice that, unlike the attack presented in [DHH+21], $\mathcal{B}$ is required to be $\mathsf{PPT}$ and thus computing $S(L, m)$ efficiently is essential in our argument. This will follow as we assumed the verification to be strictly linear, implying that $S(L, m)$ is an affine space.

Although $\mathcal{B}$ effectively breaks security, we can only upper bound the number of signatures queried by $n_1 + n_2$, i.e. one for each group element in the CRS $\mathbf{X}_1$ and verification key $\mathbf{X}_2$, since initially $L = \mathbb{F}_q^{n_1+n_2}$ with dimension $n_1 + n_2$. In order to reduce the requested signatures to be at most $n_2$ we introduce a preprocessing phase to find as many linear relations among group elements of the CRS as possibile and then run $\mathcal{B}$ providing as input a refined space $L$.

Informally, if $\mathcal{B}$ is unable to find new relations among the elements of $\mathbf{X}_1$, then $\dim L$ can at most decrease by $n_2$, yielding the desired upper bound.

To conclude we then need to describe how the preprocessing is carried out: The core idea is to initialize the set of possible exponents $V = \mathbb{F}_q^{n_1}$ and execute several times $\mathcal{B}(\mathsf{vk}^*, V)$ replying to signing queries with $\mathsf{S.Sign}(\mathsf{sk}^*, \cdot)$ where $\mathsf{vk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1)$ is freshly sampled each time. If in some of those executions $\mathcal{B}$ is able to find a new relation among the group elements, then $V$ is updated accordingly (lowering its dimension by at least 1), and a new round of simulations is run. Conversely if $\mathcal{B}(\mathsf{vk}^*, V)$ fails to find new relations several times in this simulated environment, then it is executed one last time with the real verification key $\mathsf{vk}$ and signing oracle. If no new relation is found in this last execution, $\mathcal{A}$ concludes by returning the forgeries found by $\mathcal{B}$. Otherwise $\mathcal{A}$ aborts.

Informally $\mathcal{A}$ aborts with low probability since the simulated and real executions are identically distributed from $\mathcal{B}$ perspective and in particular since no relation is found among the many simulated executions, it is unlikely this will happen in the real one. Finally we remark that simulating the signature challenger in this preprocessing phase is crucial. In this way the only signature queries performed by $\mathcal{A}$ are those requested by the last execution of $\mathcal{B}$.

A more detailed proof appears in the appendix, Section C.2.

**Adversary** $\mathcal{B}(\mathsf{vk}, V)$:

1 : Set $L \leftarrow V \times \mathbb{F}_q^{n_2} \subseteq \mathbb{F}_q^{n_1+n_2}$
2 : Initialize the set of forgeries $F \leftarrow \varnothing$ and call $\theta \leftarrow n_2 + \vartheta$
3 : Sample $m_1, \ldots, m_\theta \leftarrow^{\$} \mathsf{S.M}$ distinct messages
4 : **For** $i \in \{1, \ldots, \theta\}$:
5 : **If** $S(L, m_i) \neq \varnothing$:
6 : Get a vector $\mathbf{z} \in S(L, m_i)$
7 : Find a solution $\mathbf{Y} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y}$
8 : Set $\sigma \leftarrow (\mathbf{Y}, \mathbf{z})$ and store $F \leftarrow F \cup \{(m_i, \sigma)\}$
9 : **Else**:
10 : Query $m_i$ to the challenger and get $\sigma = (\mathbf{Y}, \mathbf{z})$
11 : Update $L \leftarrow L \cap \{\mathbf{x} \in \mathbb{F}_q^n \ : \ A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)\}$
12 : Return $F, L$

**Fig. 3.** $\mathcal{B}$ breaking $\vartheta$-UF of an algebraic signature with strictly linear verification.

*Proof of Theorem 2.* Having provided the intuition behind the attacker $\mathcal{A}$ built on top of $\mathcal{B}$, we now proceed to prove the theorem through a sequence of claims. We begin by stating the following properties about $\mathcal{B}(\mathsf{vk}, V)$ where we denote $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2$, $\mathbf{x}_1$ the discrete logarithm of $\mathbf{X}_1$ and $\mathbf{x}$ the discrete

**Adversary** $\mathcal{A}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{vk})$:

1 : Parse $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1||\mathbf{X}_2$ and $s = s_1||s_2$
2 : Initialize $V \leftarrow \mathbb{F}_q^{n_1}$ the space of potential exponents of $\mathbf{X}_1$
3 : **Do**:
4 : **For** $2n_1 + 1$ times:
5 : $\mathsf{vk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1)$
6 : Execute $F^*, L^* \leftarrow^{\$} \mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk}^*,\cdot)}(\mathsf{vk}^*, V)$
7 : Set $V^* \leftarrow \{\mathbf{x}_1 : \exists \mathbf{x}_2 : \mathbf{x}_1||\mathbf{x}_2 \in L^*\}$ the projection of $L^*$ on $\mathbb{F}_q^{n_1}$
8 : **If** $V^* \neq V$:
9 : Update $V \leftarrow V^*$, **break**
10 : **Until** the for-cycle ends without interruptions
11 : Execute $F, L \leftarrow^{\$} \mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{vk}, V)$
12 : Compute $V^*$ as the projection of $L$ on $\mathbb{F}_q^{n_1}$
13 : **If** $V^* \neq V$: Return $\mathsf{fail}$
14 : **Else**: Return $F$

**Fig. 4.** $\mathcal{A}$ breaking the $\vartheta$-UF of an algebraic signature using as subroutine an algorithm $\mathcal{B}$, which is that of Fig. 3 in the case of schemes with strictly linear verification, or that of Fig. 5 in the case of schemes with generic verification.

logarithm of $\mathbf{X}$. Finally we denote $\pi : \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2} \to \mathbb{F}_q^{n_1}$ the projection on first component, i.e. $\pi(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1$.

**Claim 1** *If $L$ is an affine space, $S(L, m)$ is an affine space. Moreover an affine base for $S(L, m)$ can be computed in polynomial time.*

**Claim 2** *If $\mathbf{x}_1 \in V$ then at any step of $\mathcal{B}(\mathsf{vk}, V)$, $\mathbf{x} \in L$.*

**Claim 3** *If $\mathbf{x}_1 \in V$, $\mathcal{B}$ is $\mathsf{PPT}$ and upon returning $(F, L)$, $F$ is a set of valid forgeries.*

**Claim 4** *For a given $m_i$, if the condition at step 5 is not satisfied, i.e. $S(L, m_i) = \varnothing$, then after step 11 the dimension of $L$ decreases strictly.*

**Claim 5** *After the execution of line 1, Fig 3, $\dim L = n_2 + \dim V$ and if $\mathcal{B}(\mathsf{vk}, V)$ returns $(F, L)$ with $\pi(L) = V$ then $\dim L \geq \dim V$.*

Next we state the following properties about $\mathcal{A}$

**Claim 6** *$\mathcal{A}$ is $\mathsf{PPT}$.*

**Claim 7** *At any step of $\mathcal{A}$ execution, $\mathbf{x}_1 \in V$.*

**Claim 8** *$\mathcal{A}$ fails with probability $\Pr[\mathcal{A}(\mathsf{vk}) \to \mathsf{fail}] \leq 1/2$.*

First we observe these claims imply the thesis. Indeed by Claim 8, with probability greater than $1/2$, $\mathcal{A}$ does not return $\mathsf{fail}$. By construction, this implies that in the last execution $\mathcal{B}(\mathsf{vk}, V)$ returns $(F, L)$ with $\pi(L) = V$. Thus by Claim 5 $n_2 + \dim V \geq L \geq \dim V$ at any step of $\mathcal{B}$ during its last execution. As a consequence $\dim L$ can decrease at most $n_2$ times. Applying Claim 4 we conclude that $S(L, m_i) = \varnothing$ can happen at most $n_2$ times because each time this occurs, $\dim L$ decreases. It follows then that for at least $\theta - n_2 = \vartheta$ messages, the condition $S(L, m_i) \neq \varnothing$ is satisfied, meaning that $\mathcal{B}$ adds a new signature to the set $F$, which in the end will have cardinality $|F| \geq \vartheta$. Finally, since $\mathbf{x} \in V$ by Claim 7, we can apply Claim 3 to conclude that $F$ is a valid set of forgeries, implying that $\mathcal{A}$ breaks $\vartheta$-UF.

Next, we provide a proof for each of these claims:

*Proof of Claim 1.* We start observing that if $L$ is any set and $\mathbf{x}_1, \ldots, \mathbf{x}_d \in L$ is a base for the linear span of $L$ then $S(L, m) = \bigcap_{i=1}^{d} S(\mathbf{x}_i, m)$. By construction, $\mathbf{x}_i \in L$ implies $S(L, m) \subseteq S(\mathbf{x}_i, m)$, and in particular $S(L, m) \subseteq \cap_{i=1}^{d} S(\mathbf{x}_i, m)$. Conversely let $\mathbf{z}$ be a vector in the intersection of all $S(\mathbf{x}_i, m)$. We can find vectors $\mathbf{u}_i \in \mathbb{F}_q^k$ such that $A(\mathbf{z}, m)\mathbf{x}_i = B(m)\mathbf{u}_i$. Since $\mathbf{x}_1, \ldots, \mathbf{x}_d$ is a base for the linear span of $L$, for any $\mathbf{x} \in L$ we can express it as a linear combination $\alpha_1 \mathbf{x}_1 + \ldots + \alpha_d \mathbf{x}_d$. In conclusion

$$A(\mathbf{z}, m)\mathbf{x} \;=\; \sum_{i=1}^{d} \alpha_i A(\mathbf{z}, m)\mathbf{x}_i \;=\; \sum_{i=1}^{d} \alpha_i B(m)\mathbf{u}_i \;=\; B(m) \sum_{i=1}^{d} \alpha_i \mathbf{u}_i.$$

Thus $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$ and in particular $\mathbf{z} \in S(L, m)$.

In order to show that $S(L, m)$ is efficiently computable it suffices to show that $S(\mathbf{x}, m)$ can be computed in polynomial time for any point $\mathbf{x}$. To this aim let $f_{\mathbf{x}} : \mathbb{F}_q^h \to \mathbb{F}_q^\ell$ be such that $f(\mathbf{z}) = A(\mathbf{z}, m)\mathbf{x}$. Since the scheme has strictly linear verification (Definition 8) $A(\,\cdot\,, m)$ is an affine map and so is $f$. Furthermore by construction $S(\mathbf{x}, m) \;=\; f_{\mathbf{x}}^{-1}(\operatorname{Im} B(m))$ since $\mathbf{z} \in S(\mathbf{x}, m)$ if and only if $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$. This concludes the argument as the preimage through an affine map of a linear space is an affine space which can be computed in polynomial time.

*Proof of Claim 2.* If $\mathbf{x}_1 \in V$ then $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 \in V \times \mathbb{F}_q^{n_2}$ which by construction implies that, when $L$ is initialized, $\mathbf{x} \in L$. Next assume by induction $\mathbf{x} \in L$ in all previous steps. The only instruction in $\mathcal{B}$ that may modify $L$ is in step 11 and when this is executed, since $\sigma = (\mathbf{Y}, \mathbf{z})$ is a valid signature by perfect correctness, we have

$$A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y} \quad \Rightarrow \quad A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i).$$

*Proof of Claim 3.* To prove that $\mathcal{B}$ is a $\mathsf{PPT}$ algorithm, observe that the for-loop is executed $\theta = n_2 + \vartheta$, that is polynomially bounded, times. Inside the loop, checking $S(L, m_i) \neq \varnothing$ and possibly computing a $\mathbf{z} \in S(L, m_i)$ can be done efficiently from Claim 1 by computing a base for it. Next, calling $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$, we have that $A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$ because

$$\mathbf{z} \in S(L, m_i) \quad \Rightarrow \quad A(\mathbf{z}, m_i) \cdot L \subseteq \operatorname{Im} B(m_i) \quad \Rightarrow \quad A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$$

where the last implication follows as $\mathbf{x} \in L$ by Claim 2 and the assumption $\mathbf{x}_1 \in V$. Thus, calling $H$ a weak-inverse of $B(m_i)$, which can be computed efficiently by Proposition 2, the vector $\mathbf{Y}$ can be set as $H \cdot A(\mathbf{z}, m_i)\mathbf{X}$. Indeed, as $A(\mathbf{z}, m_i)\mathbf{X} \in \operatorname{Im} B(m_i)$ there exists a vector $\mathbf{Z} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Z}$ and in particular

$$B(m_i)\mathbf{Y} = B(m_i)HA(\mathbf{z}, m_i)\mathbf{X} = B(m_i)HB(m_i)\mathbf{Z} = B(m_i)\mathbf{Z} = A(\mathbf{z}, m_i)\mathbf{X}.$$

Finally, given the bases of two affine spaces, a base of their intersection can be computed efficiently. This conclude the proof that $\mathcal{B}$ is $\mathsf{PPT}$.

For the second part, by construction each entry in $F$ is of the form $(m_i, \mathbf{Y}, \mathbf{z})$ such that

$$A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y}.$$

Therefore, by our definition of signatures with linear verification scheme, the verifier accepts $(m_i, \mathbf{Y}, \mathbf{z})$. The claim is thus proven.

*Proof of Claim 4.* Since the condition at step 5 is not satisfied, $S(L, m_i) = \varnothing$ and in particular $\mathbf{z} \notin S(L, m_i)$ implying that $A(\mathbf{z}, m_i)\mathbf{x} \notin \operatorname{Im} B(m_i)$ for some $\mathbf{x} \in L$. Therefore $L$ is not contained in the space of all $\mathbf{x}$ such that $A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$ and in particular its dimension decreases after the execution of step 11

*Proof of Claim 5.* The first part follow as $L$ is initially $V \times \mathbb{F}_q^{n_2}$ of dimension $\dim V + n_2$. The second part follows by linear algebra since $\dim L \geq \dim \pi(L) = \dim V$.

*Proof of Claim 6.* Since $\mathsf{S.SetupKey}$, $\mathsf{S.Sign}$ and $\mathcal{B}$ are $\mathsf{PPT}$ algorithm, by Claim 3 in the last case, each step in the loop can be computed efficiently. In particular, as $2n_1+1$ is polynomially bounded, each for-loop in $\mathcal{A}$ can be performed efficiently.

Next we show that the procedure inside the Do-Until loop is repeated at most $n_1 + 1$ times. The key observation is that during the execution of $\mathcal{B}$, the space $L$ forms a monotone decreasing sequence, implying that when $\mathcal{B}(\mathsf{vk}^*, V) \to (F^*, L^*)$ then $L^* \subseteq V \times \mathbb{F}_q^{n_2}$. In particular this implies that $\pi(L^*) \subseteq \pi(V \times \mathbb{F}_q^{n_2}) = V$. Thus if at any point the for-loop is halted, $\pi(L^*) = V^* \neq V$ implies $V^* \subseteq V$. Hence the dimension of $V$ strictly decreases, and since initially $\dim(V) = n_1$, the foor-loop can be halted at most $n_1$ times.

Finally, using again that $\mathcal{B}$ is an efficient algorithm, computing $F, L$ can be done in polynomial time. It follows that $\mathcal{A}$ is $\mathsf{PPT}$.

*Proof of Claim 7.* We proceed by induction. Initially $V = \mathbb{F}_q^{n_1}$ implies $\mathbf{x}_1 \in V$. Next we observe that the value of $V$ is only changed if, within the for-loop, $V^* \neq V$ (see step 8, Fig. 4). Assume by induction that before this step is executed $\mathbf{x}_1 \in V$. Then, when this happens, $\mathcal{B}(\mathsf{vk}^*, V) \to (F^*, L^*)$ had been executed with $\mathbf{x}_1 \in V$. By Claim 2 this implies that $\mathbf{x} \in L^*$ and in particular $\mathbf{x}_1 = \pi(\mathbf{x}) \in \pi(L^*) = V^*$. Thus when $\mathcal{A}$ sets $V \leftarrow V^*$, $\mathbf{x}_1 \in V$.

*Proof of Claim 8.* Define the following events:

– $\mathcal{E}_{i,j}$ = "During the $i$-th iteration of the Do-Until loop, and the $j$-th iteration of the for loop, $\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk}^*,\cdot)}(\mathsf{vk}^*, V)$ returns $(F^*, L^*)$ such that $\pi(L^*) = V$".
– $\mathcal{E}_{\mathsf{last}}$ = "$\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{vk}, V)$ returns $F, L$ with $\pi(L) = V$".

Furthermore let $I \sim \{1, \ldots, n_1 + 1\}$ be the random variable such that $\mathcal{A}$ terminates the Do-Until loop after the $I$-th execution. Then we observe that, conditioned on $\mathbf{X}_1, s_1$ and the $V$ at iteration $i$, the event $\mathcal{E}_{i,j}$ depends only on the random coins used for $\mathcal{B}$, $\mathsf{S.SetupKey}$ and $\mathsf{S.Sign}$ which are chosen independently at each execution of $\mathcal{B}$. In particular, for a fixed $i$, the events $\{\mathcal{E}_{i,j}\}_j$ are independent and, since for $\mathcal{E}_{i,j}, \mathcal{E}_{i,k}$ with $j \neq k$ the procedure $\mathcal{B}$ is invoked with the same input

$$\Pr[\mathcal{E}_{i,j}] = \Pr[\mathcal{E}_{i,k}].$$

We may therefore define $p_i = \Pr[\mathcal{E}_{i,1}]$ as the success probability of each execution of $\mathcal{B}$ during the $i$-th loop. Similarly, if $I = i$, the vector space $V$ given in input to $\mathcal{B}$ is by construction equal to the one used during the $i$-th execution of the Do-Until loop. In particular

$$p_i = \Pr[\mathcal{E}_{\mathsf{last}} | I = i].$$

To conclude we show that

$$\begin{aligned}
\Pr[\mathcal{A} \to \mathsf{fail}] \;&=\; \Pr[\neg\mathcal{E}_{\mathsf{last}}] \;=\; \sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \Pr[I = i] \\
&\leq \sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \Pr[\mathcal{E}_{i,1} \wedge \ldots \wedge \mathcal{E}_{i,2n_1+1}] \\
&= \sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \prod_{j=1}^{2n_1+1} \Pr[\mathcal{E}_{i,j}] \\
&= \sum_{i=1}^{n_1+1} (1 - p_i) \cdot p_i^{2n_1+1} \\
&\leq \sum_{i=1}^{n_1+1} \frac{1}{2n_1 + 2} \;=\; \frac{n_1 + 1}{2n_1 + 2} \;=\; \frac{1}{2}.
\end{aligned}$$

where the first inequality comes from the fact that $I = i$ implies $\mathcal{E}_{i,j}$ for all $j \in \{1, \ldots, 2n_1 + 1\}$, while the second inequality comes from the fact that the function $f_t(x) = (1-x)x^t$ is upper bounded by $1/(t+1)$ when $x \in [0, 1]$. Indeed $f_t(0) = f_t(1) = 0$ and its derivative vanishes only at $t/(t+1)$, which has to be the maximum point, implying that

$$(1 - x) \cdot x^t \leq \left(1 - \frac{t}{t+1}\right) \cdot \left(\frac{t}{t+1}\right)^t \leq \frac{1}{t+1}.$$

### 4.2 Attack to Schemes with Generic Verification

**Theorem 3.** *Given an algebraic signature scheme with generic verification, for any $\vartheta$ such that $n_2 + \vartheta \leq |\mathsf{S.M}|$ there exists an adversary $\mathcal{A}$ that in the unforgeability experiment in Fig. 2 performs at most $n_2$ signature queries and produces $\vartheta$ distinct forgeries.*

*Moreover, calling $\kappa$ an upper bound on the signature bit-length, and $\chi$ an upper bound on the number of queries $\mathsf{S.Vfy}$ performs to $\mathcal{O}^0_{\mathsf{eq}}$, then $\mathcal{A}$ runs in time $O(\vartheta \cdot 2^\kappa \cdot 2^\chi \cdot \mathsf{poly}(\lambda))$ and performs $O(\vartheta \cdot \mathsf{poly}(\lambda))$ queries to $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}^0_{\mathsf{eq}}$.*

*Proof.* As done in Theorem 4 we begin by providing an attack $\mathcal{B}$ which breaks the scheme but performs potentially $n_1 + n_2$ signature queries.

At a high level $\mathcal{B}$, given the verification key $\mathsf{vk} = (\mathbf{X}, s)$, will keep track of all possible exponents of $\mathbf{X}$ in a set $L$ and for each message $m$ either the dimension of $L$ decreases by one or $\mathcal{B}$ finds a forgery. Assume without loss of generality that signatures are of the form $(\mathbf{Y}', t')$ with $\mathbf{Y}' \in \mathbb{G}^k$ and $t' \in \{0,1\}^\kappa$.

For any $m$, our adversary attempts to produce a forgery as follows: For all possible $t \in \{0,1\}^\kappa$, it executes the verification algorithm by simulating a generic group $\widetilde{\mathbb{G}}$ with oracles $\widetilde{\mathcal{O}}_{\mathsf{add}}$ and $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$. More specifically, since $\mathsf{S.Vfy}$ requires as input the verification key $(\mathbf{X}, s)$, the message $m$ and the signatures $(\mathbf{Y}, t)$, $\mathcal{B}$ reproduces all the group elements involved by assigning dummy indexes for $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{Y}}$ and runs $\mathsf{S.Vfy}((\widetilde{\mathbf{X}}, s), m, (\widetilde{\mathbf{Y}}, t))$. During the execution, each query to $\widetilde{\mathcal{O}}_{\mathsf{add}}$ is emulated by simply returning new incremental indexes, while to emulate $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$, $\chi$ bits $\beta_1, \ldots, \beta_\chi$ are chosen at the beginning of the execution so that the answer to the $i$-th query will be $\beta_i$. Note that each element $T_i$ the verifier queries to $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$ has to be a linear combination of the initial group elements he received, i.e. $T_i = \mathbf{a}_i^\top \widetilde{\mathbf{X}} - \mathbf{b}_i^\top \widetilde{\mathbf{Y}} - c_i \cdot \widetilde{G}$ obtained though $\widetilde{\mathcal{O}}_{\mathsf{add}}$, and $\mathcal{B}$ can extract these coefficients.

Repeating the execution of $\mathsf{S.Vfy}$ for different values of $\beta_1, \ldots, \beta_\chi$ implicitly defines a tree of height $\chi$ in which paths are determined by the replies $\mathcal{B}$ gave at the $i$-th query to $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$. If at some point a path $\beta_1, \ldots, \beta_\chi$ that makes the verifier accept is found, $\mathcal{B}$ can try to find a vector $\mathbf{Y}$ in the real GGM, such that the $i$-query $\mathsf{S.Vfy}$ would do to $\mathcal{O}^0_{\mathsf{eq}}$ will be answered with $\beta_i$. If such a $\mathbf{Y}$ is found, then $(\mathbf{Y}, t)$ will be a valid forgery for $m$.

Recalling that the $i$-th query has the form $T_i = \mathbf{a}_i \widetilde{\mathbf{X}} - \mathbf{b}_i \widetilde{\mathbf{Y}} - c_i \cdot G$, then $\mathcal{B}$ needs to find a vector $\mathbf{Y}$ such that for all $i \in \{1, \ldots, \chi\}$

$$\mathbf{a}_i^\top \mathbf{X} = \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G \text{ when } \beta_i = 1, \qquad \mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G \text{ when } \beta_i = 0$$

Regarding the equations on the left side, they can be packed up into a system $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$. Through pseudo-inverses or Gaussian elimination is easy to check if solutions exists for all $\mathbf{x} \in L$ (as in the proof of Theorem 2). If this is not the case $\mathcal{B}$ simply discards this path and continues its brute-force search. However, even if the previous condition is satisfied, for some of the points $\mathbf{x}$ in $L$ it may be the case that any vector $\mathbf{y}$ satisfying $A\mathbf{x} = B\mathbf{y} + \mathbf{c}$ fails to satisfy some of the inequalities above $\mathbf{a}_i^\top \mathbf{x} \neq \mathbf{b}_i^\top \mathbf{y} + c_i$, implying that no solution $\mathbf{Y} \in \mathbb{G}^k$ can

be found if $\mathbf{x}$ is the discrete logarithm of $\mathbf{X}$. We call these points $\mathbf{x} \in L$ *faulty* and, more specifically, the set of faulty points is defined as

$$\mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c} = \{\mathbf{x} : A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}, \quad \forall \mathbf{y} \in \mathbb{F}_q^m \quad A\mathbf{x} = B\mathbf{y} + \mathbf{c} \Rightarrow \mathbf{a}^\top \mathbf{x} = \mathbf{b}^\top \mathbf{y} + c\}.$$

Three possible cases may occur now:

- If all points in $L$ are faulty with respect to some inequality constraint, then $\mathcal{B}$ gives up on the path as the solution $\mathbf{Y}$ does not exist.
- If not all points are faulty $\mathcal{B}$ attempts to solve the system, which requires expensive queries to $\mathcal{O}_{\mathsf{add}}, \mathcal{O}^0_{\mathsf{eq}}$: if a solution $\mathbf{Y}$ satisfying all constraints is found, this is a valid forgery.
- If not all points are faulty, but no solution can be found, it means that $\mathbf{x}$, the discrete log of $\mathbf{X}$, has to be a faulty point. This information reduces the dimension of $L$ as not all points in $L$ are faulty.

Finally, if no solution can be found for any $t \in \{0,1\}^\kappa$ and path $\beta_1, \ldots, \beta_\chi$, $\mathcal{B}$ queries a signature for $m$ and uses this information to reduce the dimension of $L$. As for the proof of Theorem 2, $\mathcal{B}$ might overall query $n_1 + n_2$ signatures (as opposed to the desired $n_1$) since initially it has no information on the exponents of $\mathbf{X}$, i.e. $\dim L = n_1 + n_2$, and each signature query may reveal only one new linear combination among these group elements. To address this issue we use the same strategy presented in Theorem 2, that is, we use $\mathcal{B}$ in a black-box way inside the algorithm $\mathcal{A}$, formally described in Fig. 4. The main idea is again that $\mathcal{A}$ initially extracts linear combinations among CRS elements that could be found by $\mathcal{B}$, and finally executes $\mathcal{B}$ providing the retrieved information as input. In this way $\mathcal{B}$ will, with significant probability, only find relations among elements of $\mathbf{X}_2$, thus requesting at most $n_2$ signatures.

A detailed description of $\mathcal{A}$ appears in Fig. 5, while a more detailed proof of the Theorem appears in the appendix, Section C.3.

## 5 Conclusions

### 5.1 Impossibility of Algebraic Vector Commitments

Using both the negative results provided in the previous sections for algebraic signatures and Theorem 1 connecting the efficiency of a VC to the security of the associated signature scheme, we obtain two lower bounds for algebraic vector commitments

**Theorem 4.** *Given a position binding algebraic VC with strictly linear verification, let $\ell_c = \ell_c(n)$ and $\ell_\pi = \ell_\pi(n)$ be respectively the commitment and opening bit length to commit to a vector of $n$ entries. Then*

$$\nu_2 + \frac{\lambda + \ell_c + \nu_2 \cdot (\ell_\pi + \log |\mathsf{VC.M}|)}{\log |\mathsf{VC.M}|} \geq n.$$

**Adversary** $\mathcal{B}(\mathsf{vk}, V)$:

1 : Initialize $F \leftarrow \varnothing$ the set of forgeries
2 : Call $L = V \times \mathbb{F}_q^{n_2}$ the set of possible exponents of $\mathbf{X}$
3 : Call $\theta = n + \vartheta$ and sample $m_1, \ldots, m_\theta \leftarrow^{\$} \mathsf{S.M}$ distinct messages
4 : **For** $m \in \{m_1, \ldots, m_\theta\}$:
5 : **For** $t \in \{0,1\}^\kappa$ and $(\beta_1, \ldots, \beta_\chi) \in \{0,1\}^\chi$:
6 : Simulate a Generic Group $\widetilde{\mathbb{G}}$ with generator $\widetilde{G}$ and oracles $\widetilde{\mathcal{O}}_{\mathsf{add}}$ and $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$
7 : Assign indices for two vectors $\widetilde{\mathbf{X}} \in \widetilde{\mathbb{G}}^n$ and $\widetilde{\mathbf{Y}} \in \widetilde{\mathbb{G}}^k$
8 : Run $\mathsf{S.Vfy}((\widetilde{\mathbf{X}}, s), m, (\widetilde{\mathbf{Y}}, t))$ using $\widetilde{\mathbb{G}}$
9 : **When** $\mathsf{S.Vfy}$ queries $\widetilde{\mathcal{O}}_{\mathsf{add}}(T, S)$:
10 : Store a way to express $T + S$ as a linear combination of $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{Y}}$ and $\widetilde{G}$
11 : Return to $\mathsf{S.Vfy}$ a label for $T + S$
12 : **When** $\mathsf{S.Vfy}$ queries $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}(T_i)$ the $i$-th time:
13 : Store $\mathbf{a}_i \in \mathbb{F}_q^n$, $\mathbf{b}_i \in \mathbb{F}_q^k$ and $c_i \in \mathbb{F}_q$ such that $T_i = \mathbf{a}_i^\top \widetilde{\mathbf{X}} - \mathbf{b}_i^\top \widetilde{\mathbf{Y}} - c_i \cdot \widetilde{G}$
14 : Return $\beta_i$ to $\mathsf{S.Vfy}$
15 : **When** $\mathsf{S.Vfy}$ halts and returns $b \in \{0,1\}$:
16 : Let $A = (\mathbf{a}_i \ : \ \beta_i = 1)$, $B = (\mathbf{b}_i \ : \ \beta_i = 1)$ and $\mathbf{c} = (c_i \ : \ \beta_i = 1)$
17 : **If** $b = 0$:
18 : **Continue** cycle in line 5
19 : **Elif** $A \cdot L \nsubseteq \operatorname{Im} B + \mathbf{c}$:
20 : **Continue** cycle in line 5
21 : **Elif** $\exists i \ : \ \beta_i = 0$ and $L \subseteq \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$:
22 : **Continue** cycle in line 5
23 : **Elif** $\exists i \ : \ \beta_i = 0$ and $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$:
24 : Update $L \leftarrow L \cap \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$
25 : **Break** cycle in line 5
26 : **Else**:
27 : Find $\mathbf{Y} \in \mathbb{G}^k$ s.t. $A\mathbf{X} = B\mathbf{Y} + \mathbf{c}G$ and $\mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}_i^\top \mathbf{Y} + c_i G$ for $\beta_i = 0$
28 : Store $\sigma \leftarrow (\mathbf{Y}, t)$ and $F \leftarrow F \cup \{(m, \sigma)\}$
29 : **Break** cycle in line 5
30 : **If** the cycle ended without interruptions:
31 : Query a signature for $m$ and wait for $(\mathbf{Y}, t)$
32 : Reconstruct $A, B, \mathbf{c}$ as in step 16 using $(\mathbf{X}, s, m, \mathbf{Y}, t)$ and the group $\mathbb{G}$
33 : Update $L \leftarrow L \cap \{\mathbf{x} \in \mathbb{F}_q^n \ : \ A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}\}$
34 : Return $F, L$

**Fig. 5.** $\mathcal{B}$ breaking security of an algebraic signature scheme with generic verification.

*Proof.* Assume there exists an algebraic VC with strictly linear verification contradicting the above inequality and satisfying position binding. Then by Theorem 1 the signature scheme obtained through the transformation in Fig. 1 would satisfy $\vartheta$-UF with

$$\vartheta(\mathsf{vk}, Q) \;=\; \frac{\lambda + \ell_c + |Q| \cdot (\ell_\pi + \log |\mathsf{VC.M}|)}{\log |\mathsf{VC.M}|}$$

and its message space would have size $|\mathsf{S_{VC}.M}| = n$. Since $\mathsf{vk}$ contains $\nu_2$ group elements excluding those that belong to the CRS, i.e. the public parameters of the original Vector Commitment, the attacker $\mathcal{A}$ from Theorem 2 can produce at least $n - \nu_2$ forgeries performing at most $\nu_2$ queries. Called $Q$ the set of queries performed by $\mathcal{A}$ we would have that

$$\vartheta(\mathsf{vk}, Q) \;\leq\; \frac{\lambda + \ell_c + \nu_2 \cdot (\ell_\pi + \log |\mathsf{VC.M}|)}{\log |\mathsf{VC.M}|} \;<\; n - \nu_2$$

where we use the fact that $|Q| \leq \nu_2$ in the first inequality. This is then a contradiction since $\mathcal{A}$ would breaks the $\vartheta$-UF of the derived signature, implying that the given vector commitment was not binding.

**Theorem 5.** *Given an algebraic VC with generic verification that is position binding against unbounded adversaries performing polynomially bounded queries to the GGM oracles $\mathcal{O}_{\mathsf{add}}$, $\mathcal{O}^0_{\mathsf{eq}}$, using the same notation of Theorem 4, then*

$$\nu_2 + \frac{\lambda + \ell_c + \nu_2 \cdot (\ell_\pi + \log |\mathsf{VC.M}|)}{\log |\mathsf{VC.M}|} \;\geq\; n.$$

*Proof.* Assuming again by contradiction that the above inequality is not satisfied, Theorem 1 implies that the associated signature scheme is $\vartheta$-UF against any unbounded adversary $\mathcal{C}$ making at most polynomially many signature and group operations queries, or otherwise $\mathcal{R}^{\mathcal{C}}$ would break position binding with significant advantage. Notice that since $\mathcal{R}$ is $\mathsf{PPT}$, $\mathcal{R}^{\mathcal{C}}$ still performs polynomially many generic group operations. As in the proof of Theorem 4 then, our initial assumption implies $\vartheta \leq n - \nu_2$. Since the adversary $\mathcal{A}$ of Theorem 3 returns $n - \nu_2$ signatures performing at most $\nu_2$ queries, this contradicts the $\vartheta$-UF of the associated signature against this adversary.

**Corollary 1.** *Given an algebraic vector commitment with strictly linear verification, then $\ell_c \cdot \ell_\pi = \Omega(n)$. Analogously, given an algebraic vector commitment with generic verification position binding against unbounded adversary performing at most polynomially many queries to the GGM oracles, $\ell_c \cdot \ell_\pi = \Omega(n)$.*

Note that this lower bound implies in both cases that either $\ell_c = \Omega(\sqrt{n})$ or $\ell_\pi = \Omega(\sqrt{n})$.

### 5.2 Impossibility of Algebraic Signatures

As a by-product of our study on VC we also obtain the following two impossibility results for algebraic signatures which extend the one presented in [DHH+21] to a broader family of schemes.

**Theorem 6.** *For any UF-CMA algebraic signature scheme with strictly linear verification,* $n_1 \geq |\mathsf{S.M}|$.

**Theorem 7.** *For any algebraic signature scheme with generic verification UF-CMA secure against any unbounded adversary performing at most polynomially many queries to the GGM oracles,* $n_1 \geq |\mathsf{S.M}|$.

## Acknowledgments

This work has received funding in part from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under project PICOCRYPT (grant agreement No. 101001283), by the Spanish Government under projects SCUM (ref. RTI2018-102043-B-I00), RED2018-102321-T, and SECURING (ref. PID2019-110873RJ-I00), by the Madrid Regional Government under project BLOQUES (ref. S2018/TCS-4339), by a research grant from Nomadic Labs and the Tezos Foundation, by the Programma ricerca di ateneo UNICT 35 2020-22 linea 2 and by research gifts from Protocol Labs.

## References

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

AHO12. Masayuki Abe, Kristiyan Haralambiev, and Miyako Ohkubo. Group to group commitments do not shrink. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 301–317. Springer, Heidelberg, April 2012.

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

BBF19. Dan Boneh, Benedikt Bünz, and Ben Fisch. Batching techniques for accumulators with applications to IOPs and stateless blockchains. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 561–586. Springer, Heidelberg, August 2019.

BCC+16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

BG12. Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 263–280. Springer, Heidelberg, April 2012.

BGV11. Siavosh Benabbas, Rosario Gennaro, and Yevgeniy Vahlis. Verifiable delegation of computation over large datasets. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 111–131. Springer, Heidelberg, August 2011.

CF13. Dario Catalano and Dario Fiore. Vector commitments and their applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 55–72. Springer, Heidelberg, February / March 2013.

CFG+20. Matteo Campanelli, Dario Fiore, Nicola Greco, Dimitris Kolonelos, and Luca Nizzardo. Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 3–35. Springer, Heidelberg, December 2020.

CPZ18. Alexander Chepurnoy, Charalampos Papamanthou, and Yupeng Zhang. Edrax: A cryptocurrency with stateless transaction validation. Cryptology ePrint Archive, Report 2018/968, 2018. `https://eprint.iacr.org/2018/968`.

DHH+21. Nico Döttling, Dominik Hartmann, Dennis Hofheinz, Eike Kiltz, Sven Schäge, and Bogdan Ursu. On the impossibility of purely algebraic signatures. In *Theory of Cryptography Conference*, pages 317–349. Springer, 2021.

Fis18. Ben Fisch. PoReps: Proofs of space on useful data. Cryptology ePrint Archive, Report 2018/678, 2018. `https://eprint.iacr.org/2018/678`.

GGK03. Rosario Gennaro, Yael Gertner, and Jonathan Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *35th ACM STOC*, pages 417–425. ACM Press, June 2003.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013.

GKM+00. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 325–335. IEEE Computer Society, 2000.

GMR88. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

GMR01. Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd FOCS*, pages 126–135. IEEE Computer Society Press, October 2001.

Gol87. Oded Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 104–110. Springer, Heidelberg, August 1987.

GT00. Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 305–313. IEEE Computer Society, 2000.

IR89. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

JK07. Ari Juels and Burton S. Kaliski Jr. Pors: proofs of retrievability for large files. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 584–597. ACM Press, October 2007.

Kil94. Joe Kilian. On the complexity of bounded-interaction and noninteractive zero-knowledge proofs. In *35th FOCS*, pages 466–477. IEEE Computer Society Press, November 1994.

KST99. Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 535–542. IEEE Computer Society, 1999.

Kus18. John Kuszmaul. Verkle trees, 2018.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Heidelberg, December 2010.

LM19. Russell W. F. Lai and Giulio Malavolta. Subvector commitments with application to succinct arguments. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 530–560. Springer, Heidelberg, August 2019.

LRY16. Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *ICALP 2016*, volume 55 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl, July 2016.

LY10. Benoît Libert and Moti Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 499–517. Springer, Heidelberg, February 2010.

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005.

Mer88. Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 369–378. Springer, Heidelberg, August 1988.

Mic94. Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994.

MPZ20. Ueli Maurer, Christopher Portmann, and Jiamin Zhu. Unifying generic group models. Cryptology ePrint Archive, Report 2020/996, 2020. `https://eprint.iacr.org/2020/996`.

MRK03. Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-knowledge sets. In *44th FOCS*, pages 80–91. IEEE Computer Society Press, October 2003.

MRV99. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th FOCS*, pages 120–130. IEEE Computer Society Press, October 1999.

PPS21. Chris Peikert, Zachary Pepin, and Chad Sharp. Vector and functional commitments from lattices. In *Theory of Cryptography Conference*, pages 480–511. Springer, 2021.

PRV12. Periklis A. Papakonstantinou, Charles Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? *Electron. Colloquium Comput. Complex.*, page 167, 2012.

PSTY13. Charalampos Papamanthou, Elaine Shi, Roberto Tamassia, and Ke Yi. Streaming authenticated data structures. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 353–370. Springer, Heidelberg, May 2013.

RSS20. Lior Rotem, Gil Segev, and Ido Shahaf. Generic-group delay functions require hidden-order groups. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 155–180. Springer, Heidelberg, May 2020.

SGS20. Gili Schul-Ganz and Gil Segev. Accumulators in (and beyond) generic groups: Non-trivial batch verification requires interaction. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 77–107. Springer, Heidelberg, November 2020.

SGS21. Gili Schul-Ganz and Gil Segev. Generic-Group Identity-Based Encryption: A Tight Impossibility Result. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:23, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.

Sim98. Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology - EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998.

Zha22. Mark Zhandry. To label, or not to label (in generic groups). Cryptology ePrint Archive, Report 2022/226, 2022. `https://eprint.iacr.org/2022/226`.

ZZ21. Mark Zhandry and Cong Zhang. The relationship between idealized models under computationally bounded adversaries. Cryptology ePrint Archive, Report 2021/240, 2021. `https://eprint.iacr.org/2021/240`.

# A More Preliminaries

## A.1 Linear Algebra

We recall basic definitions and facts from linear algebra that we use in our proofs.

**Definition 10.** *Given $A \in \mathbb{F}_q^{n,m}$, a weak left inverse of $A$ is a matrix $H \in \mathbb{F}_q^{m,n}$ such that $AHA = A$.*

**Proposition 1.** *Given $A \in \mathbb{F}_q^{n,m}$ and $H \in \mathbb{F}_q^{m,n}$ then $H$ is a weak left inverse of $A$ if and only if for any $\mathbf{y} \in \operatorname{Im} A$, $AH\mathbf{y} = \mathbf{y}$.*

**Proposition 2.** *There exists a* PPT *algorithm that given a matrix $A \in \mathbb{F}_q^{n,m}$ returns a weak left inverse of $A$.*

*Proof sketch.* Calling $V = \operatorname{Im} A$, let $k = \dim V$ and $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be a base of $V$, which can be extended to $\mathbf{v}_1, \ldots, \mathbf{v}_n$ base of $\mathbb{F}_q^n$. Then, since the map $A : \mathbb{F}_q^m \to V \leq \mathbb{F}_q^n$ is surjective, there exists $\mathbf{w}_i \in \mathbb{F}_q^m$ such that $A\mathbf{w}_i = \mathbf{v}_i$ for all $i \leq k$. Then, letting $H : \mathbb{F}_q^n \to \mathbb{F}_q^m$ a linear function such that $H\mathbf{v}_i = \mathbf{w}_i$ for $i \leq k$ and $H\mathbf{v}_i = \mathbf{0}$ otherwise, we show $H$ is a weak left inverse for $A$.

To do this is enough to prove that the linear map $AH : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is the identity over $V$, which is equivalent to show $AH$ is the identity over a base of $V$. Indeed for all $\mathbf{v}_i$, $AH\mathbf{v}_i = A\mathbf{w}_i = \mathbf{v}_i$.

Finally we remark that computing a base of $\operatorname{Im} A$, extending it, finding preimages $\mathbf{w}_i$, and computing the matrix $H$ are all steps that can be performed in polynomial time.

**Definition 11.** *Given $L \leq \mathbb{F}_q^n$ be an affine space, we call its direction $\operatorname{dir} L$ the only linear subspace of $\mathbb{F}_q^n$ such that $L = \mathbf{x} + \operatorname{dir} L$ for some $\mathbf{x} \in \mathbb{F}_q^n$.*

### A.2 Min-Entropy

For a given random variable, the min-entropy measures informally how hard is it to correctly guess it without any information.

**Definition 12.** *Given $X \sim \mathcal{X}$ random variable distributed over a finite set $\mathcal{X}$ we define its min-entropy as*

$$\mathrm{H}_\infty(X) \;=\; -\log\left(\max_{x \in \mathcal{X}} \Pr\left[X = x\right]\right).$$

**Definition 13.** *Given $X \sim \mathcal{X}$ and $Y \sim \mathcal{Y}$ two random variables over finite sets, the conditional min-entropy of $X$ given $Y$ is defined as*

$$\mathrm{H}_\infty(X|Y) \;=\; -\log\left(\sum_{y \in \mathcal{Y}} \Pr\left[Y = y\right] \cdot \max_{x \in \mathcal{X}} \Pr\left[X = x | Y = y\right]\right).$$

**Proposition 3.** *Given $X \sim \mathcal{X}$ and $Y \sim \mathcal{Y}$ random variables over finite sets, then*

$$\mathrm{H}_\infty(X|Y) \geq \mathrm{H}_\infty(X) - \log|\mathcal{Y}|$$

*Proof.*

$$\begin{aligned}\sum_{y\in\mathcal{Y}} \Pr[Y=y]\cdot\max_{x\in\mathcal{X}}\Pr[X=x|Y=x] &= \sum_{y\in\mathcal{Y}}\max_{x\in\mathcal{X}}\Pr[X=x|Y=y]\Pr[Y=y]\\ &= \sum_{y\in\mathcal{Y}}\max_{x\in\mathcal{X}}\Pr[X=x,\ Y=y]\\ &\leq \sum_{y\in\mathcal{Y}}\max_{x\in\mathcal{X}}\Pr[X=x]\\ &= |\mathcal{Y}|\cdot\max_{x\in\mathcal{X}}\Pr[X=x].\end{aligned}$$

Applying $-\log(\cdot)$ to both sides yields the claimed inequality.

### A.3 Arithmetic circuits and R1CS

In this section we introduce circuit satisfiability and the rank one constraint system relations, which will be used later to provide examples of secure algebraic signatures with linear verification. Informally we define an arithmetic circuit $C$ as a tuple $(n, g_1, \ldots, g_m)$ where there first $n$ gates $(g_1, \ldots, g_n) = (1, \ldots, n)$ are input gates and the remaining gates $g_i$ are of the form $(\circ, j, k)$ where $j, k < i$ or $(\circ, j, \alpha)$ with $j < i$ and $\alpha \in \mathbb{F}_q$ and $\circ \in \{+, \cdot\}$ is a binary operator. The last gate $g_m$ is also considered to be the output gate.

Given an arithmetic circuit and an input $\mathbf{x} \in \mathbb{F}_q^n$ we can define the evaluation function $\mathsf{ev}_{\mathbf{x}}$ from the gates of $C$ to $\mathbb{F}_q$ recursively as

$$\begin{cases}\mathsf{ev}_{\mathbf{x}}(g_i) = x_i & \text{If } i \leq n\\ \mathsf{ev}_{\mathbf{x}}(g_i) = \mathsf{ev}_{\mathbf{x}}(g_j)\circ\mathsf{ev}_{\mathbf{x}}(g_k) & \text{If } g_i = (\circ, j, k)\\ \mathsf{ev}_{\mathbf{x}}(g_i) = \mathsf{ev}_{\mathbf{x}}(g_j)\circ\alpha & \text{If } g_i = (\circ, j, \alpha)\end{cases}$$

A circuit it said to return a value $y$ on input $\mathbf{x}$, in symbols $C(\mathbf{x}) = y$, if the output gate evaluates to $\mathsf{ev}_{\mathbf{x}}(g_m) = y$.

The problem of determining whether a circuit is satisfiable, i.e. if there exists an input $\mathbf{x}$ such that $C(m)$ returns 1, is notoriously $\mathsf{NP}$-complete and tightly related to the more algebraic rank one constraints system (R1CS in short) satisfiability problem. Informally an instance of R1CS is a system of quadratic equations of the form

$$M_1\mathbf{w} * M_2\mathbf{w} = M_3\mathbf{w} + \mathbf{b}$$

where $*$ is the entry-wise product between same length vectors, $M_1, M_2, M_3 \in \mathbb{F}_q^{n,m}$ and $\mathbf{b} \in \mathbb{F}_q^m$. A given instance is said to be satisfiable if the above equation admits a solution $\mathbf{w} \in \mathbb{F}_q^n$.

As both circuit satisfiability and R1CS relations are NP-complete, there exists an efficient reduction $\mathcal{T}$ that takes as input a circuit $C$ and return a R1CS instance $M_1, M_2, M_3, \mathbf{b}$, and, if provided an input $\mathbf{x} \in \mathbb{F}_q^n$ that satisfies $C$, it further returns $\mathbf{w}$ satisfying the produced R1CS. Moreover, if $M_1, M_2, M_3, \mathbf{b}$ are

obtained from $C$ as above, there also exists an efficient procedure that, given in input a witness $\mathbf{w}$ for the R1CS, returns an input $\mathbf{x}$ satisfying $C$.

A detailed description of the reduction $\mathcal{T}$ can be found in [AHIV17, GGPR13], although we provide a sketch here for completeness. Given a circuit $C = (n, g_1, \ldots, g_m)$, $\mathcal{T}$ builds the matrices $M_1, M_2, M_3 \in \mathbb{F}_q^{m,m-n+1}, \mathbf{b} \in \mathbb{F}_q^{m,m-n+1}$ adding:

- for each gate $g_i = (+, j, k)$ [resp. $g_i = (+, j, \alpha)$] the constraint $w_i = w_j + w_k$ [resp. $w_i = w_j + \alpha$]
- for each gate $g_i = (\cdot, j, k)$ [resp. $g_i = (\cdot, j, \alpha)$] the constraint $w_i = w_j \cdot w_k$ [resp. $w_i = w_j \cdot \alpha$]
- the constrain $w_m = 1$.

Furthermore if an input $\mathbf{x}$ such that $C(\mathbf{x}) = 1$ is provided, $\mathcal{T}$ returns also a solution $\mathbf{w}$ such that $w_i = \mathsf{ev}_{\mathbf{x}}(g_i)$. By the way $\mathsf{ev}$ was defined is immediate to verify by induction that $\mathbf{w}$ satisfies the given set of constraints if $C(\mathbf{x}) = 1$. Moreover if $\mathbf{w}$ is a solution to the given system, then $\mathbf{x} = (w_1, \ldots, w_n)$ is such that $C(\mathbf{x}) = 1$, implying that from any solution, a satisfying input for $C$ can be computed efficiently.

### A.4 Generic Group Model

We now introduce more in details the Generic Group Model proposed by Maurer in [Mau05] and revised by Maurer, Portmann and Zhu [MPZ20]. Even though the definitions proposed in these papers are given in higher generality to capture a broad class of generic algorithms and problems, below we only present their instantiation to the case of a group of known prime order $q$.

**Definition 14.** *A generic group of prime order $q$ is an interactive Turing machine $\mathcal{B}$ which at any step stores a list of elements $L = (V_1, \ldots, V_m)$ with $V_i \in \mathbb{F}_q$. Initially $L$ is empty and $m = 0$. Furthermore, upon receiving:*

- $(\mathsf{gen})$*: Sets $V_{m+1} \leftarrow 1$, appends $V_{m+1}$ to $L$ and increments $m \leftarrow m + 1$.*
- $(\mathsf{add}, i, j)$*: if $i, j \in \{1, \ldots, m\}$, sets $V_{m+1} \leftarrow V_i + V_j$, appends $V_{m+1}$ to $L$ and increments $m \leftarrow m + 1$.*
- $(\mathsf{eq}, i, j)$*: if $i, j \in \{1, \ldots, m\}$, sets $b$ the bit $V_i == V_j$ and returns $b$.*

A remarkable difference between this computational model and Shoup Generic Group Model [Sho97] is that a machine interacting with $\mathcal{B}$ has no access to a representation of group elements, and has to "remember" their indices as it queries them. This To avoid the issue of explicitly keeping track of them, which may render the description of generic algorithms more verbose, we may assume without loss of generality that after $(\mathsf{gen})$ or $(\mathsf{add}, i, j)$, $\mathcal{B}$ returns $m + 1$, the newly generated element's index. Furthermore, for notational simplicity, we model usage of the generic group through three oracles $\mathcal{O}_{\mathsf{gen}}$, $\mathcal{O}_{\mathsf{add}}$, $\mathcal{O}_{\mathsf{eq}}$ such that

- $\mathcal{O}_{\mathsf{gen}}()$ queries $(\mathsf{gen})$ to $\mathcal{B}$ and return the new group element's index $G$.

– $\mathcal{O}_{\mathsf{add}}(X, Y)$, queries $(\mathsf{add}, X, Y)$ to $\mathcal{B}$ and return the new group element's index $Z$
– $\mathcal{O}_{\mathsf{eq}}(X, Y)$, queries $(\mathsf{eq}, X, Y)$ to $\mathcal{B}$ and return the bit $b$ it receives back from $\mathcal{B}$.

Noticeably, without loss of generality, one may assume that $\mathcal{O}_{\mathsf{gen}}$ is queried only once at the beginning of any algorithm's execution – or equivalently that the list $L$ maintained by $\mathcal{B}$ is initialized with 1 in its first entry. Similarly one may also assume that, up to performing at most $2 \log q$ queries to $\mathcal{O}_{\mathsf{add}}$, an index for the group identity 0 is known - which could be obtained by computing $q \cdot G$. Analogously, given the index of an element $X$, its inverse can be computed in at most $2 \log q$ queries to $\mathcal{O}_{\mathsf{add}}$ by computing $(q-1)X$. Finally we observe that, since inverses can be computed efficiently, for algorithms that are bounded to perform at most a polynomial number of queries to $\mathcal{B}$, it is enough to provide access to the identity equality test oracle $\mathcal{O}^0_{\mathsf{eq}}$ defined as

– $\mathcal{O}^0_{\mathsf{eq}}(X)$: Given a precomputed index for 0, it queries $(\mathsf{eq}, X, 0)$ to $\mathcal{B}$ and returns the bit $b$ it receives back from $\mathcal{B}$.

Given access to this oracle $\mathcal{O}_{\mathsf{eq}}(X, Y)$ can then be simulated querying $\mathcal{O}^0_{\mathsf{eq}}(X-Y)$.

# B Examples and Constructions

## B.1 Linear vs Generic Verification for Signatures

In this section we compare the definitions of algebraic signature with linear and generic verification. The first one, as mentioned, is equivalent to the notion of *algebraic signatures* proposed in [DHH+21] whereas the second, seemingly more general, only requires the algorithms involved to use the group generically.

In order to prove that the second notion is strictly more general we will provide an algebraic signature scheme that has generic verification but is not equivalent to any scheme with linear verification. The core idea is that, when $\mathsf{S.Vfy}$ has no restriction it can test through $\mathcal{O}^0_{\mathsf{eq}}$ whether a given group element $X$ has discrete logarithm in base $G$ equals to 0 or 1, and continue the verification according to the result. In this way, given any signature scheme in the standard model, the signer can encode in the exponent a bit representation of a given signature while the verifier can extract these exponents and then verify the extracted string is a valid signature.

More formally, let $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ be a secure signature scheme in the standard model (e.g., stateless constructions are known from trapdoor claw-free permutations, [GMR88, Gol87]). Then an algebraic signature scheme can be constructed as shown in Fig. 6.

A PPT adversary against the unforgeability of this scheme clearly reduces to the security of the underlying signature. Thus using an existentially unforgeable scheme for this construction we would end up with a secure algebraic signature with generic verification.

To conclude, we prove that this scheme does not have linear verification

$\mathsf{S}^*.\mathsf{Setup}(1^\lambda)$

1 : $\mathsf{vk}, \mathsf{sk} \leftarrow^{\$} \mathsf{S.Setup}(1^\lambda)$
2 : Parse $\mathsf{vk} = (\beta_1, \ldots, \beta_n) \in \{0,1\}^n$
3 : Set $\mathbf{X} = (\beta_1 G, \ldots, \beta_n G)$
4 : Set $\mathsf{vk}^* \leftarrow \mathbf{X}$ and $\mathsf{sk}^* \leftarrow \mathsf{sk}$
5 : Return $\mathsf{vk}, \mathsf{sk}$

$\mathsf{S}^*.\mathsf{Sign}(\mathsf{sk}^*, m)$

1 : $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$
2 : Parse $\sigma = (\gamma_1, \ldots, \gamma_k) \in \{0,1\}^k$
3 : Set $\mathbf{Y} = (\gamma_1 G, \ldots, \gamma_k G)$
4 : Return $\mathbf{Y}$

$\mathsf{S}^*.\mathsf{Vfy}(\mathsf{vk}^*, m, \sigma^*)$

1 : Parse $\mathsf{vk} = \mathbf{X} = (X_1, \ldots, X_n)$ and $\sigma^* = \mathbf{Y} = (Y_1, \ldots, Y_k)$
2 : For each $i \in \{1, \ldots, n\}$ set $\beta_i$ the outcome of $\mathcal{O}^0_{\mathsf{eq}}(X_i - G)$
3 : For each $i \in \{1, \ldots, k\}$ set $\gamma_i$ the outcome of $\mathcal{O}^0_{\mathsf{eq}}(Y_i - G)$
4 : Set $\mathsf{vk} = (\beta_1, \ldots, \beta_n) \in \{0,1\}^n$ and $\sigma = (\gamma_1, \ldots, \gamma_k) \in \{0,1\}^k$
5 : Run $b \leftarrow^{\$} \mathsf{S.Vfy}(\mathsf{vk}, \sigma, m)$ and return $b$

**Fig. 6.** Algebraic Signature Scheme with linear verification

**Proposition 4.** *There exist no matrices $A, B$ such that $\mathsf{S.Vfy}(\mathbf{X}, m, \mathbf{Y})$ accepts if and only if*

$$A(m)\mathbf{X} = B(m)\mathbf{Y}$$

*Proof.* Assuming by contradiction such $A$ and $B$ exist, then ($\mathsf{S}^*.\mathsf{Setup}, \mathsf{S}^*.\mathsf{Sign}$, $\mathsf{S}^*.\mathsf{Vfy}$) would be an algebraic scheme with linear verification and in particular the adversary $\mathcal{A}$ described in [DHH+21] for *algebraic signatures* can be used to break the security of this construction. Note that for a general algebraic scheme with linear verification, assuming each signature is of the form $(\mathbf{Y}, t)$ with $\mathbf{Y} \in \mathbb{G}^k$ and $t \in \{0,1\}^\kappa$, the adversary $\mathcal{A}$ runs in time $O(2^\kappa \cdot \mathsf{poly}(\lambda))$. Since for the scheme described in Fig. 6 $\kappa = 0$ the adversary $\mathcal{A}$ would run in polynomial time, contradicting our assumption on the underlying signature scheme.

### B.2 Secure Signatures with Linear Verification

The first attack we describe in Theorem 2 for algebraic signature schemes with linear verification only holds if the equation tested by the verifier is of the form

$$A(\mathbf{z}, m)\mathbf{X} = B(m)\mathbf{Y}$$

with the entries $A(\mathbf{z}, m)$ being polynomials of degree one in $\mathbf{z}$, and $B(m)$ being independent from $\mathbf{z}$.

In this section we show that secure signatures with exponential message space and linear verification which do not satisfy at least one of these two hypotheses exist. To get these results we only assume the existence of a secure signature schemes ($\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy}$) in the plain model with exponential message space, e.g. stateless constructions form trapdoor claw-free permutations [GMR88, Gol87].

First we propose a scheme where in the verification equation $A(\mathbf{z}, m)$ depends quadratically from $\mathbf{z}$. The high-level idea is to observe that the satisfiability of a circuit (in this case $\mathsf{S.Vfy}(\mathsf{vk}, m, \cdot)$) can be checked by testing whether a system of equations of degree at most 2 admits solutions. In this way, using the transformation sketched in Section A.3 we can compile $C = \mathsf{S.Vfy}(\mathsf{vk}, m, \cdot)$ to an R1CS instance $M_1, M_2, M_3, \mathbf{b}$ such that, from any witness $\sigma \in \{0,1\}^k$ satisfying $C(\sigma) = 1$, a vector $\mathbf{w}$ can be computed so that

$$M_1\mathbf{w} * M_2\mathbf{w} = M_3\mathbf{w} + \mathbf{b}$$

where $*$ is the entry-wise multiplication of two vectors with the same length. Moreover we recall that given any vector $\mathbf{w}$ satisfying the above system of quadratic equations, a string $\sigma \in \{0,1\}^k$ such that $C(\sigma) = 1$ can be computed efficiently.

A detailed description of the resulting scheme is presented in Figure 7

$\mathsf{S}_1.\mathsf{Setup}(1^\lambda)$

1 : $\mathsf{vk}, \mathsf{sk} \leftarrow^{\$} \mathsf{S.Setup}(1^\lambda)$
2 : Return $\mathsf{vk}, \mathsf{sk}$

$\mathsf{S}_1.\mathsf{Sign}(\mathsf{sk}, m)$

1 : $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$
2 : Compile $\mathsf{S.Vfy}(\mathsf{vk}, m, \cdot), \sigma$ to $M_1, M_2, M_3, \mathbf{b}, \mathbf{w}$
3 : Return $\sigma^* \leftarrow \mathbf{w}$

$\mathsf{S}_1.\mathsf{Vfy}(\mathsf{vk}, m, \sigma^*)$

1 : Compile $\mathsf{S.Vfy}(\mathsf{vk}, m, \cdot)$ to $M_1, M_2, M_3, \mathbf{b}$
2 : Return 1 if $(M_1\mathbf{w} * M_2\mathbf{w} - M_3\mathbf{w} - \mathbf{b}) \cdot G = \mathbf{0}$

**Fig. 7.** Algebraic signature scheme with linear verification

**Proposition 5.** *If* $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ *is a UF-CMA secure signature, so is the scheme described in Figure 7.*

*Proof.* First of all correctness comes from the underlying scheme and the properties of the reduction between circuit satisfiability and rank one constraint systems. Indeed, for any $m \in \mathsf{S.M}$, $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$ implies $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1$. Hence $\sigma$ is a witness for that circuit, implying that $M_1\mathbf{w} * M_2\mathbf{w} = M_3\mathbf{w} + \mathbf{b}$. Thus $\mathsf{S}_1.\mathsf{Vfy}(\mathsf{vk}, m, \mathbf{w})$ accepts.

Next, if $\mathcal{A}$ is a $\mathsf{PPT}$ adversary for the scheme in Figure 7, we describe an adversary $\mathcal{B}$ for the underlying signature. Upon receiving $\mathsf{vk}$, $\mathcal{B}$ forwards $\mathsf{vk}$ to $\mathcal{A}$. Each time $\mathcal{A}$ queries a signature for $m$, $\mathcal{B}$ forwards the request to its signing oracle, getting $\sigma$. Later, it compiles $\mathsf{S.Vfy}(\mathsf{vk}, m, \cdot), \sigma$ to $M_1, M_2, M_3, \mathbf{b}, \mathbf{w}$ and returns $\mathbf{w}$. Finally, when $\mathcal{A}$ outputs $m, \mathbf{w}$, $\mathcal{B}$ compiles $\mathsf{S.Vfy}(\mathsf{vk}, m, \cdot)$ to $M_1, M_2, M_3, \mathbf{b}$ and check if $M_1\mathbf{w} * M_2\mathbf{w} = M_3\mathbf{w} + \mathbf{b}$. If that is the case it recover a witness $\sigma$ for that circuit from $\mathbf{w}$ and returns $(m, \sigma)$.

Clearly $\mathcal{B}$ perfectly simulates $\mathsf{S}_1.\mathsf{Sign}(\mathsf{sk}, \cdot)$ as it performs the same steps and, when $\mathcal{A}$ successfully return a valid forgery $(m, \mathbf{w})$, the couple $(m, \sigma)$ is a forgery for $\mathcal{B}$ since:

- $\mathcal{B}$ queries the same messages queried by $\mathcal{A}$, thus $m$ was not queried by both.
- By the property of the reduction from circuit satisfiability to rank one constraints system, $M_1\mathbf{w} * M_2\mathbf{w} = M_3\mathbf{w} + \mathbf{b}$ implies that $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1$.

Thus $\mathsf{Adv}(\mathcal{A}) = \mathsf{Adv}(\mathcal{B})$.

Next we sketch a scheme where the verification equation is of the form $A(\mathbf{z}, m)\mathbf{X} = B(\mathbf{z}, m)\mathbf{Y}$, where both $A(\mathbf{z}, m)$ and $B(\mathbf{z}, m)$ have an affine dependence on $\mathbf{z}$ (i.e. each of their entries is a polynomial of degree at most one in $\mathbf{z}$). The idea is identical to the previous construction with the exception that now the terms of degree 2 have to come from $B(\mathbf{z}, m)\mathbf{Y}$. To do this, having compiled for a given signed message $(m, \sigma)$ the circuit $\mathsf{S.Vfy}(\mathsf{vk}, m, \cdot), \sigma$ into $M_1, M_2, M_3, \mathbf{b}, \mathbf{w}$, the signer simply set $\mathbf{Y} = \mathbf{w} \cdot G$ and returns $(\mathbf{Y}, \mathbf{w})$. The verifier then locally checks

$$\mathbf{Y} = \mathbf{w} \cdot G, \qquad (M_3\mathbf{w} + \mathbf{b}) \cdot G = M_1\mathbf{w} * M_2\mathbf{Y}.$$

To see why the equation on the right hand side can be expressed in the form $B(\mathbf{z}, m)$ observe that the map $f_{\mathbf{z}} : \mathbb{G}^m \to \mathbb{G}^m$ mapping $\mathbf{Y} \mapsto M_1\mathbf{z} * M_2\mathbf{Y}$ is linear and depends linearly on $\mathbf{z}$. Moreover, if both checks are satisfied, then

$$\mathbf{Y} = \mathbf{w} \cdot G \quad \Rightarrow \quad (M_3\mathbf{w} + \mathbf{b}) \cdot G \;=\; (M_1\mathbf{w} * M_2\mathbf{w}) \cdot G$$

which immediately yields that $\mathbf{w}$ is a solution for the given R1CS.

### B.3 Succinct Vector Commitments with Linear Verification

In Section B.2 we showed how dropping the assumption of strict linear verification in algebraic signatures implies the existence of secure constructions under mild assumptions. Because of this, there is no hope to obtain lower bounds for algebraic VCs from analogous results on signatures when the verification procedure is linear but not strictly. In this section we prove that this is the case because, when either $A(\mathbf{z}, m, i)$ is allowed to depend quadratically or $B(\mathbf{z}, m, i)$ affinely on $\mathbf{z}, m$, there exist succinct VCs.

Given a (not necessarily algebraic) position binding and succinct VC ($\mathsf{VC.Setup}$, $\mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy}$) such that $\mathbb{F}_q \subseteq \mathsf{VC.M}$, the key idea as before is to exploit the quadratic terms in the verification equations to encode arbitrary circuits, in this case $C(\cdot, \cdot) = \mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot)$ which takes in input a message $m$ and an opening proof $\pi$. More specifically, the new setup algorithm simply forwards the CRS produced by $\mathsf{VC.Setup}$. Analogously a commitment to $(m_1, \ldots, m_n)$ can simply be set as the commitment $c$ returned by $\mathsf{VC.Com}$. In order to open to a message $m$ we can compute an opening $\pi$ through $\mathsf{VC.Open}$, compile $\mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot), m, \pi$ to a R1CS $M_1, M_2, M_3, \mathbf{w}$ where $w_1 = m$, and

return $\mathbf{w}$. Finally, the verification algorithm simply compiles $\mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot)$ to the same R1CS $M_1, M_2, M_3$ and, on input $m, \mathbf{w}, i$ verifies the following conditions:

$$(M_1\mathbf{w} * M_2\mathbf{w} - M_3\mathbf{w}) \cdot G = \mathbf{0}, \qquad (w_1 - m) \cdot G = 0.$$

where the last test ensure that the first input of the circuit is precisely the message $m$. A more detailed construction appears in Figure. 8

$\mathsf{VC}^*.\mathsf{Setup}(1^\lambda)$

1 : $\mathsf{ppS.Setup}(1^\lambda)$
2 : Return $\mathsf{pp}$

$\mathsf{VC}^*.\mathsf{Com}(\mathsf{pp}, m_1, \ldots, m_n)$

1 : $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$
2 : Return $(c, \mathsf{aux}||c)$

$\mathsf{VC}^*.\mathsf{Open}(\mathsf{pp}, m, i, \mathsf{aux}||c)$

1 : $\pi \leftarrow \mathsf{VC.Open}(\mathsf{pp}, m, i, \mathsf{aux})$
2 : Compile $\mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot), (m, \pi)$ to $M_1, M_2, M_3, \mathbf{w}$
3 : Return $\mathbf{w}$

$\mathsf{VC}^*.\mathsf{Vfy}(\mathsf{pp}, c, m, i, \mathbf{w})$

1 : Compile $\mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot)$ to $M_1, M_2, M_3$.
2 : Accept if the following conditions are satisfied:
3 : $(M_1\mathbf{w} * M_2\mathbf{w} - M_3\mathbf{w}) \cdot G = \mathbf{0}$
4 : $(m - w_1) \cdot G = 0$

**Fig. 8.** Algebraic VC scheme with linear verification

**Proposition 6.** *The VC scheme described in Figure 8 is algebraic with linear verification. Furthermore if* $(\mathsf{VC.Setup}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy})$ *satisfy position binding, so does this construction.*

*Proof.* The first part is immediate as $B$ can be simply set to be the zero matrix while

$$A(\mathbf{w}, m, i, (\mathsf{pp}\|c)) \;=\; (M_1\mathbf{w} * M_2\mathbf{w} - M_3\mathbf{w}) \,\|\, (m - w_1)$$

depends quadratically on $\mathbf{w}$ and $m$, and arbitrarily on $\mathsf{pp}\|c$ and $i$ which are hard-coded into the circuit that is later compiled to the R1CS $M_1, M_2, M_3$.

To prove that the resulting scheme is secure let $\mathcal{A}(\mathsf{pp})$ be a $\mathsf{PPT}$ adversary breaking positing binding. Then we can describe an adversary $\mathcal{B}$ that breaks the position binding property of the underlying VC. To do so, $\mathcal{B}(\mathsf{pp})$ runs $\mathcal{A}$ on input $\mathsf{pp}$ and waits for it to return $(c, m, m', i, \mathbf{w}, \mathbf{w}')$ with $m \neq m'$. Then it checks whether $(m, \mathbf{w})$ and $(m', \mathbf{w}')$ are both correct openings for position $i$, i.e. such that, calling $M_1, M_2, M_3$ matrices representing the R1CS obtained from

the circuit $C = \mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot)$, they satisfy

$$M_1\mathbf{w} * M_2\mathbf{w} = M_3\mathbf{w} \qquad M_1\mathbf{w}' * M_2\mathbf{w}' = M_3\mathbf{w}'.$$

and $m = w_1$, $m' = w_1'$. If this is the case it finally extracts, as detailed in Section A.3, two inputs $(\widetilde{m}, \widetilde{\pi})$ and $(\widetilde{m}', \widetilde{\pi}')$ that make the circuit $C$ evaluate to 1, i.e. such that

$$\mathsf{VC.Vfy}(\mathsf{pp}, c, \widetilde{m}, i, \widetilde{\pi}) \to 1 \qquad \mathsf{VC.Vfy}(\mathsf{pp}, c, \widetilde{m}', i, \widetilde{\pi}') \to 1.$$

and returns $(c, \widetilde{m}, \widetilde{m}', i, \widetilde{\pi}, \widetilde{\pi}')$. Note that, since the extraction of an input that makes $C$ return 1 from a solution to the related R1CS is performed by taking the first $t$ elements, where $t$ is the arity of $C$, then $\widetilde{m} = w_1 = m$ and $\widetilde{m}' = w_1' = m'$ implying that $\widetilde{m} \neq \widetilde{m}'$. This concludes the argument.

Although the construction in Figure 8 provides us a way to construct algebraic vector commitments with linear verification from any VC in the standard model, it is yet not enough to show that our lower bounds may not be extended to this case. Indeed, the reduction preserves the commitment length of the underlying VC but not the opening proof size. Indeed, as mentioned in Section A.3, the length of $\mathbf{w}$ equals the number of gates in the circuit from which we built the R1CS instance. This means that even though the underlying VC has short opening proof, if the verifier performs a significant number of operations which depends on $n$, the resulting scheme may not be succinct.

Thus, in order to show that this construction leads to VCs with constant size commitment and opening proof length, we need to provide a (non necessarily algebraic) scheme in which the circuit $\mathsf{VC.Vfy}(\mathsf{pp}, c, \cdot, i, \cdot)$ has a constant number of gates. One such example is the pairing based construction provided in [CF13] whose security is equivalent to CDH. More specifically in that case, given a pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and a generator $G \in \mathbb{G}$ the setup produces the following CRS for uniformly random $z_i \leftarrow^{\$} \mathbb{F}_q$

$$\forall i \in \{1, \ldots, n\} \quad H_i \leftarrow z_i \cdot G, \qquad \forall i, j \in \{1, \ldots, n\},\ i \neq j \quad H_{i,j} \leftarrow z_i z_j \cdot G.$$

Next a commitment to $m_1, \ldots, m_n$ is defined as $K = m_1 H_1 + \ldots + m_n H_n$ while an opening proof for position $i$ is computed as

$$\Lambda_i \leftarrow \sum_{j \neq i} m_i H_{i,j} \ = \ z_i \sum_{j \neq i} m_i H_i.$$

Finally, the verification algorithm on input $(\mathsf{pp}, K, m, i, \Lambda)$ recovers from the public parameter the group element $H_i$ and checks that

$$e(K - m_i H_i) \ = \ e(\Lambda_i, G).$$

Observe that in the circuit we want to compile to a R1CS, $\mathsf{pp}$ and $i$ are hard-coded constants, so is $H_i$, meaning that the circuit does not have to read the whole $\mathsf{pp}$ to recover it. Finally we have that in this case $\mathsf{VC.Vfy}(\mathsf{pp}, K, \cdot, i, \cdot)$

as an arithmetic circuit requires only one exponentiation $m_i \cdot H_i$, one inversion/addition over $\mathbb{G}$ to compute $K - m_i H_i$, two pairings and one equality check over $\mathbb{G}_T$. Since the number of gates required to express these operations as circuits only depends on $\lambda$, and we only need a constant number of them, we conclude that with respect to $n$ the circuit $\mathsf{VC.Vfy}(\mathsf{pp}, K, \cdot, i, \cdot)$ can be expressed with a constant number of gates. The following proposition follows

**Proposition 7.** *Given a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that CDH is hard in $\mathbb{G}$, then there exists an algebraic VC with linear verification such that $A(\mathbf{z}, m, i)$ depends quadratically in $\mathbf{z}, m$, $B(i)$ does not depends on $\mathbf{z}, m$ and, calling $\ell_c, \ell_\pi$ respectively the commitment and opening proof bit length, $\ell_c, \ell_\pi = O_\lambda(1)$.*

We finally informally remark that the same construction can be repeated if we let $A(\mathbf{z}, m, i)$ and $B(\mathbf{z}, m, i)$ both depend affinely on $\mathbf{z}, m$.

### B.4 Tightness of Algebraic VC Lower Bounds

We now show the tightness of the asymptotic lower bound $\ell_c \cdot \ell_\pi = \Omega(n)$ proved in Corollary 1 for an algebraic VC with strictly linear verification, where $n$ is the length of the vector, $\ell_c$ the commitment bit-length and $\ell_\pi$ the opening proof bit length. To this end, calling $\ell_\mathbb{G}$ the bit length of a group element representation, we describe a family of schemes parameterized by $\alpha = \alpha(n) \leq n$ where $\ell_c = \alpha \cdot \ell_\mathbb{G}$ and $\ell_\pi = \beta \cdot \ell_\mathbb{G}$ with

$$\beta(n) = \left\lceil \frac{n}{\alpha(n)} \right\rceil .$$

Since $\ell_\mathbb{G}$ is polynomially bounded in $\lambda$ but independent from $n$ we have that asymptotically $\ell_c \cdot \ell_\pi = \Theta(n)$ as

$$\begin{aligned} \frac{n}{\alpha(n)} \leq \beta(n) \leq \frac{n}{\alpha(n)} + 1 \quad &\Rightarrow \quad n \leq \alpha(n)\beta(n) \leq n + \alpha(n) \leq 2n \\ &\Rightarrow \quad \ell_\mathbb{G}^2 \cdot n \leq \ell_c \cdot \ell_\pi \leq 2\ell_\mathbb{G}^2 \cdot n. \end{aligned}$$

In particular this family of schemes achieves our asymptotic bound. The high level idea, assuming for the moment that $\alpha(n)$ divides $n$, is to perform the commitment to a vector of $n$ message $m_1, \ldots, m_n$ by dividing them into $\beta$ smaller vectors of $\alpha$ entries each, and perform a regular (non-hiding) Pedersen commitment to each sub-vector. To open a position $i$ we simply open the entire sub-vector containing the desired message, resulting in an opening proof of $\beta$ field elements.

As a final note, we remark that the security of a Pedersen commitment reduces to the hardness of the DLP in the underlying group. Thus this scheme would also be secure for the class of unbounded adversaries which perform at most a polynomial number of generic operations. In particular we conclude that even against this class of adversaries our lower bound in Corollary 1 is tight.

$\mathsf{VC.Setup}(1^\lambda, n)$

1 : $H_1, \ldots, H_\alpha \leftarrow^{\$} \mathbb{G}$
2 : $\mathsf{pp} \leftarrow (H_1, \ldots, H_\alpha)$
3 : Return $\mathsf{pp}$

$\mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$

1 : $\beta \leftarrow \lceil n/\alpha \rceil, \quad \mu \leftarrow \alpha \cdot \beta$
2 : $(m_{n+1}, \ldots, m_\mu) \leftarrow (0, \ldots, 0)$
3 : For $k \in \{0, \ldots, \beta - 1\}$:
4 : $\quad c_k \leftarrow \sum_{j=1}^{\alpha} m_{j+k\alpha} \cdot H_j$
5 : $c \leftarrow (c_k)_{k=0}^{\beta-1}, \ \mathsf{aux} \leftarrow (m_i)_{i=1}^{\mu}$
6 : Return $c, \mathsf{aux}$

$\mathsf{VC.Open}(\mathsf{pp}, m, i, \mathsf{aux})$

1 : Write $i = j + k\alpha$ with $j \in \{0, \ldots, \beta - 1\}$ and $k \in \{1, \ldots, \alpha\}$
2 : $\pi \leftarrow (m_{1+k\alpha}, \ldots, m_{(k+1)\alpha})$. Return $\pi$

$\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi)$

1 : Write $i = j^* + k\alpha$ with $j \in \{1, \ldots, \alpha\}$ and $k \in \{0, \ldots, \beta - 1\}$
2 : Parse $\pi = (m'_1, \ldots, m'_\alpha)$, $\quad c = (c_0, \ldots, c_{\beta-1})$ and $\mathsf{pp} = (H_j)_{j=1}^{\alpha}$
3 : **If** $c_k = \sum_{j=1}^{\alpha} m'_j \cdot H_j$ and $m = m'_{j^*}$:
4 : $\quad$ Return 1
5 : **Else**: Return 0

**Fig. 9.** Vector commitment over a prime order group, parameterized by $\alpha$

**Proposition 8.** *The vector commitment in Figure 9 is algebraic with strictly linear verification. Moreover if the DLP problem is hard in $\mathbb{G}$, it satisfies position binding.*

*Proof.* To show the first part of the proof we have to provide matrices $A$, $B$ that characterize the verification algorithm. Since $\mathbf{Y}$ is not provided, $B$ can only be the zero matrix, which is indeed independent from the field elements in the opening proof. Conversely, calling $\mathbf{X}_1 = (G, H_1, \ldots, H_\alpha)$, $\mathbf{X}_2 = (c_0, \ldots, c_{\beta-1})$, $\pi = (m'_1, \ldots, m'_\alpha)$, $i = j + k^*\alpha$, the matrix $A(\pi, m, i)$ is

$$A \ = \ \begin{bmatrix} m - m'_1 & 0 & \ldots & 0 & 0 \ldots & 0 & \ldots 0 \\ 0 & m'_1 & \ldots m'_\alpha & 0 \ldots & -1 & \ldots 0 \end{bmatrix}$$

where $-1$ is in the $(1 + \alpha + k)$-th column. Notice each entry is a polynomial of degree at most 1 in $m$ and $m'_i$.

Regarding position binding, given an adversary $\mathcal{A}$ which returns $(c, \widehat{m}_i, \widehat{m}'_i, i\pi_i, \pi'_i)$ with $m_i \neq m'_i$, we can trivially transform this into an adversary $\mathcal{B}$ breaking the binding property of Pedersen commitment. Having decomposed $i = j + \alpha k$ for $j \in \{1, \ldots, \alpha\}$, $k \in \{0, \ldots, \beta - 1\}$, and parsed

$$c = (c_0, \ldots, c_{\beta-1}), \qquad \pi_i = (m_1, \ldots, m_\alpha), \qquad \pi'_i = (m'_1, \ldots, m'_\alpha)$$

$\mathcal{B}$ returns $(c_k, (m_1, \ldots, m_\alpha), (m'_1, \ldots, m'_\alpha))$. If $\mathcal{A}$ succeeds in returning two valid openings for different messages, then $\mathcal{B}$ succeed as well. Indeed since $m_k = \widehat{m}_i \neq \widehat{m}'_i = m'_k$, the two vectors are distinct and, as both openings are valid, $\sum_{j=1}^{\alpha} m_j \cdot H_j = c_k = \sum_{j=1}^{\alpha} m'_j \cdot H_j$. This concludes the proof as (generalized) Pedersen commitments are binding if the DLP is hard.

### B.5 Tightness of Algebraic Signatures Lower Bounds

As done for the impossibility results of algebraic vector commitments, we show that our lower bounds for algebraic signatures with strictly linear/general verification (Theorems 6 and 7) cannot be improved. We do so by providing for each $n$ a family of UF-CMA secure signatures whose verification key contains $n$ group elements and the message space has cardinality $n$. The idea is simply to set $\mathsf{vk} = (x_1 \cdot G, \ldots, x_n \cdot G)$ and, in order to sign the message $i \in \{1, \ldots, n\}$ the signer returns $x_i$. In this way producing a forgery reduces to breaking the discrete logarithm of at least one group element in the verification key, implying intuitively that the scheme is secure.

$\mathsf{S.Setup}(1^\lambda)$:

1 : Sample $x_1, \ldots, x_n \leftarrow^{\$} \mathbb{F}_q$ and compute $H_i \leftarrow x_i \cdot G$
2 : $\mathsf{sk} \leftarrow (x_1, \ldots, x_n)$, $\mathsf{vk} \leftarrow (H_1, \ldots, H_n)$. Return $(\mathsf{sk}, \mathsf{vk})$

$\mathsf{S.Sign}(\mathsf{sk}, m)$:

1 : Parse $\mathsf{sk} = (x_1, \ldots, x_n)$. Return $x_m$

$\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma)$:

1 : Parse $\mathsf{vk} = (H_1, \ldots, H_n)$. Return $H_m == \sigma \cdot G$

**Fig. 10.** Secure algebraic signature scheme with strictly linear verification and message space $\mathsf{S.M} = \{1, \ldots, n\}$.

**Proposition 9.** *The scheme in figure 10 is algebraic with strictly linear verification. Moreover, if the DLP is hard in $\mathbb{G}$, it is UF-CMA secure.*

*Proof.* The strictly linear property of verification follows trivially by setting $\mathbf{X} = (G, H_1, \ldots, H_n)$, $A(\sigma, m) = (-\sigma, 0, \ldots, 1, \ldots, 0)$ with 1 being in the $m+1$-th position, and $B(m)$ as the zero matrix.

To prove security, we reduce any adversary $\mathcal{A}$ executed in game 2 to an adversary $\mathcal{B}$ which solves the DLP. Given a random $H \sim U(\mathbb{G})$, $\mathcal{B}$ samples an index $j \in \{1, \ldots, n\}$ which is a guess on the message that $\mathcal{A}$ will chose for his forgery, and set $H_j = H$ and $H_i = x_i \cdot G$ for all other $i$ by sampling $x_i \leftarrow^{\$} \mathbb{F}_q$. If

$\mathcal{A}$ queries $j$ $\mathcal{B}$ aborts. If it queries $i \neq j$, send $\mathcal{A} \leftarrow x_i$. Finally when $\mathcal{A}$ returns $(m, \sigma)$, abort if $m \neq j$, otherwise return $\sigma$.

Since $\mathcal{A}$ has no information on $j$ the probability it will attempt to forge that message is $1/n$. Moreover if $\mathcal{A}$ returns a correct signature $\sigma$, then $H = H_j = \sigma \cdot G$, meaning that $\mathcal{B}$ correctly solved the DLP. Hence $\mathsf{Adv}(\mathcal{A}) = n \cdot \mathsf{Adv}(\mathcal{B})$ that is negligible.

# C Postponed Proofs

## C.1 Generic Transformation to Signatures

*Proof of Theorem 1.* Given an adversary $\mathcal{A}$ breaking $\vartheta$-UF for this signature scheme, we provide a PPT algorithm $\mathcal{R}$ that breaks the underlying VC's position binding using $\mathcal{A}$ in a black-box way. For notational convenience let us denote $Q = \{(j, m_j, \pi_j)\}_{j \in S}$ the set of all queries recorded in the experiment 2 with $S$ being the set of queried positions.

**Adversary** $\mathcal{R}(\mathsf{pp})$:

1 : Samples $m_1, \ldots, m_n \leftarrow^{\$} \mathsf{VC.M}$ and get $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$
2 : Set $\mathsf{vk} \leftarrow (\mathsf{pp}, c)$, $S \leftarrow \varnothing$ and send $\mathcal{A} \leftarrow \mathsf{vk}$
3 : **When** $\mathcal{A}$ queries $j \in \{1, \ldots, n\}$:
4 : Compute the opening $\pi \leftarrow^{\$} \mathsf{VC.Open}(\mathsf{pp}, m_j, j, \mathsf{aux})$ and set $\sigma \leftarrow (m_j, \pi)$
5 : Update $S \leftarrow S \cup \{j\}$ and send $\mathcal{A} \leftarrow \sigma$
6 : **When** $\mathcal{A}$ returns a set $F$:
7 : **If** $\exists (i, \sigma) \in F \ : \ i \notin S \ \wedge \ \sigma = (\widehat{m}, \widehat{\pi}) \ \wedge \ \widehat{m} \neq m_i$:
8 : Compute $\pi_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$ and return $(c, m_i, \widehat{m}, i, \pi_i, \widehat{\pi})$
9 : **Else**: Return $\perp$

**Fig. 11.** Reduction $\mathcal{R}$ breaking position binding.

As a first step we argue that with overwhelming probability, if $\mathcal{A}$ produces more than $\vartheta$ forgeries, then at least one them contains an opening to a message that differs from the committed one.

**Claim 1** *Let $F = \{(i, \widehat{m}_i, \widehat{\pi}_i)\}_{i \in I}$ be the set of forgeries $\mathcal{A}$ produces. If $|I| \geq \vartheta(\mathsf{vk}, Q)$ and $I \cap S = \varnothing$, calling* bad *the event $\widehat{m}_i = m_i$, $\forall i \in I$, then*

$$\Pr[\mathsf{bad}] \leq 2^{-\lambda}.$$

*Proof of Claim 1.* $\{m_i\}_{i \in I}$ are uniformly distributed and independent from $\mathsf{pp}$, but may not be independent from $c$ and $\{m_j, \pi_j\}_{j \in S}$. Notice that, by the way we defined $\ell_c$ and $\ell_\pi$, we have that

$$(c, \{\pi_j\}_{j \in S}, \{m_j\}_{j \in S}) \in \{0,1\}^{\ell_c + |S| \ell_\pi} \times \mathsf{VC.M}^{|S|}.$$

Applying Proposition 3 we can then lower bound the conditional min-entropy $\mathrm{H}_\infty(\{m_i\}_{i\in I} \,|\, c, \{m_j, \pi_j\}_{j\in S}) \geq$

$$\begin{aligned}
&\geq \mathrm{H}_\infty(\{m_i\}_{i\in I}) - \ell_c - |S| \cdot \ell_\pi - |S| \cdot \log |\mathsf{VC.M}| \\
&= |I| \cdot \log |\mathsf{VC.M}| - \ell_c - |S| \cdot (\ell_\pi + \log |\mathsf{VC.M}|) \\
&\geq \vartheta(\mathsf{vk}, Q) \cdot \log |\mathsf{VC.M}| - \ell_c - |Q| \cdot (\ell_\pi + \log |\mathsf{VC.M}|) \\
&= \lambda
\end{aligned}$$

Where the first equality follows as, without conditioning on $c, \pi_j, m_j$, $m_{i i\in I}$ and $I$ are independent, the second inequality uses $|I| \geq \vartheta(\mathsf{vk}, Q)$ and $|S| = |Q|$ while the last equality applies our assumption on $\vartheta(\mathsf{vk}, Q)$. In conclusion, the probability of correctly guessing $\{m_i\}_{i\in I}$ given $c$ and $\{m_j, \pi_j\}_{j\in S}$ is smaller than $2^{-\lambda}$, proving the claim.

To conclude the Theorem's proof, if $\mathcal{A}$ wins in game 2 then all the forgeries returned are correct signatures of non-queried messages. In particular if $\mathsf{bad}$ does not occur, among the forgeries returned by $\mathcal{A}$, $\mathcal{R}$ will find $\widehat{m}_i \neq m_i$ and $\widehat{\pi}_i$ such that

$$\mathsf{VC.Vfy}(\mathsf{pp}, c, m_i, i, \pi_i) \to 1 \qquad \mathsf{VC.Vfy}(\mathsf{pp}, c, \widehat{m}_i, i, \widehat{\pi}_i) \to 1.$$

which breaks the position binding. Therefore $\mathsf{Adv}(\mathcal{A}) \leq \mathsf{Adv}(\mathcal{R}) + 2^{-\lambda}$ which is negligible.

### C.2 Attack to Schemes with Strictly Linear Verification

*Proof of Theorem 2.* Having provided the intuition behind the attacker $\mathcal{A}$ built on top of $\mathcal{B}$, we now proceed to prove the theorem through a sequence of claims. We begin by stating the following properties about $\mathcal{B}(\mathsf{vk}, V)$ where we denote $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2$, $\mathbf{x}_1$ the discrete logarithm of $\mathbf{X}_1$ and $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$. Finally we denote $\pi : \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2} \to \mathbb{F}_q^{n_1}$ the projection on first component, i.e. $\pi(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1$.

**Claim 1** *If $L$ is an affine space, $S(L, m)$ is an affine space. Moreover an affine base for $S(L, m)$ can be computed in polynomial time.*

**Claim 2** *If $\mathbf{x}_1 \in V$ then at any step of $\mathcal{B}(\mathsf{vk}, V)$, $\mathbf{x} \in L$.*

**Claim 3** *If $\mathbf{x}_1 \in V$, $\mathcal{B}$ is $\mathsf{PPT}$ and upon returning $(F, L)$, $F$ is a set of valid forgeries.*

**Claim 4** *For a given $m_i$, if the condition at step 5 is not satisfied, i.e. $S(L, m_i) = \varnothing$, then after step 11 the dimension of $L$ decreases strictly.*

**Claim 5** *After the execution of line 1, Fig 3, $\dim L = n_2 + \dim V$ and if $\mathcal{B}(\mathsf{vk}, V)$ returns $(F, L)$ with $\pi(L) = V$ then $\dim L \geq \dim V$.*

Next we state the following properties about $\mathcal{A}$

**Claim 6** *$\mathcal{A}$ is* PPT*.*

**Claim 7** *At any step of $\mathcal{A}$ execution, $\mathbf{x}_1 \in V$.*

**Claim 8** *$\mathcal{A}$ fails with probability* $\Pr[\mathcal{A}(\mathsf{vk}) \to \mathsf{fail}] \leq 1/2$.

First we observe these claims imply the thesis. Indeed by Claim 8, with probability greater than $1/2$, $\mathcal{A}$ does not return fail. By construction, this implies that in the last execution $\mathcal{B}(\mathsf{vk}, V)$ returns $(F, L)$ with $\pi(L) = V$. Thus by Claim 5 $n_2 + \dim V \geq L \geq \dim V$ at any step of $\mathcal{B}$ during its last execution. As a consequence $\dim L$ can decrease at most $n_2$ times. Applying Claim 4 we conclude that $S(L, m_i) = \varnothing$ can happen at most $n_2$ times because each time this occurs, $\dim L$ decreases. It follows then that for at least $\theta - n_2 = \vartheta$ messages, the condition $S(L, m_i) \neq \varnothing$ is satisfied, meaning that $\mathcal{B}$ adds a new signature to the set $F$, which in the end will have cardinality $|F| \geq \vartheta$. Finally, since $\mathbf{x} \in V$ by Claim 7, we can apply Claim 3 to conclude that $F$ is a valid set of forgeries, implying that $\mathcal{A}$ breaks $\vartheta$-UF.

Next, we provide a proof for each of these claims:

*Proof of Claim 1.* We start observing that if $L$ is any set and $\mathbf{x}_1, \ldots, \mathbf{x}_d \in L$ is a base for the linear span of $L$ then $S(L, m) = \bigcap_{i=1}^{d} S(\mathbf{x}_i, m)$. By construction, $\mathbf{x}_i \in L$ implies $S(L, m) \subseteq S(\mathbf{x}_i, m)$, and in particular $S(L, m) \subseteq \cap_{i=1}^{d} S(\mathbf{x}_i, m)$. Conversely let $\mathbf{z}$ be a vector in the intersection of all $S(\mathbf{x}_i, m)$. We can find vectors $\mathbf{u}_i \in \mathbb{F}_q^k$ such that $A(\mathbf{z}, m)\mathbf{x}_i = B(m)\mathbf{u}_i$. Since $\mathbf{x}_1, \ldots, \mathbf{x}_d$ is a base for the linear span of $L$, for any $\mathbf{x} \in L$ we can express it as a linear combination $\alpha_1\mathbf{x}_1 + \ldots + \alpha_d\mathbf{x}_d$. In conclusion

$$A(\mathbf{z}, m)\mathbf{x} \;=\; \sum_{i=1}^{d} \alpha_i A(\mathbf{z}, m)\mathbf{x}_i \;=\; \sum_{i=1}^{d} \alpha_i B(m)\mathbf{u}_i \;=\; B(m) \sum_{i=1}^{d} \alpha_i \mathbf{u}_i.$$

Thus $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$ and in particular $\mathbf{z} \in S(L, m)$.

In order to show that $S(L, m)$ is efficiently computable it suffices to show that $S(\mathbf{x}, m)$ can be computed in polynomial time for any point $\mathbf{x}$. To this aim let $f_{\mathbf{x}} : \mathbb{F}_q^h \to \mathbb{F}_q^\ell$ be such that $f(\mathbf{z}) = A(\mathbf{z}, m)\mathbf{x}$. Since the scheme has strictly linear verification (Definition 8) $A(\,\cdot\,, m)$ is an affine map and so is $f$. Furthermore by construction $S(\mathbf{x}, m) \;=\; f_{\mathbf{x}}^{-1}(\operatorname{Im} B(m))$ since $\mathbf{z} \in S(\mathbf{x}, m)$ if and only if $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$. This concludes the argument as the preimage through an affine map of a linear space is an affine space which can be computed in polynomial time.

*Proof of Claim 2.* If $\mathbf{x}_1 \in V$ then $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 \in V \times \mathbb{F}_q^{n_2}$ which by construction implies that, when $L$ is initialized, $\mathbf{x} \in L$. Next assume by induction $\mathbf{x} \in L$ in all previous steps. The only instruction in $\mathcal{B}$ that may modify $L$ is in step 11 and when this is executed, since $\sigma = (\mathbf{Y}, \mathbf{z})$ is a valid signature by perfect correctness, we have

$$A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y} \quad \Rightarrow \quad A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i).$$

*Proof of Claim 3.* To prove that $\mathcal{B}$ is a $\mathsf{PPT}$ algorithm, observe that the for-loop is executed $\theta = n_2 + \vartheta$, that is polynomially bounded, times. Inside the loop, checking $S(L, m_i) \neq \varnothing$ and possibly computing a $\mathbf{z} \in S(L, m_i)$ can be done efficiently from Claim 1 by computing a base for it. Next, calling $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$, we have that $A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$ because

$$\mathbf{z} \in S(L, m_i) \quad \Rightarrow \quad A(\mathbf{z}, m_i) \cdot L \subseteq \operatorname{Im} B(m_i) \quad \Rightarrow \quad A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$$

where the last implication follows as $\mathbf{x} \in L$ by Claim 2 and the assumption $\mathbf{x}_1 \in V$. Thus, calling $H$ a weak-inverse of $B(m_i)$, which can be computed efficiently by Proposition 2, the vector $\mathbf{Y}$ can be set as $H \cdot A(\mathbf{z}, m_i)\mathbf{X}$. Indeed, as $A(\mathbf{z}, m_i)\mathbf{X} \in \operatorname{Im} B(m_i)$ there exists a vector $\mathbf{Z} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Z}$ and in particular

$$B(m_i)\mathbf{Y} \;=\; B(m_i)HA(\mathbf{z}, m_i)\mathbf{X} \;=\; B(m_i)HB(m_i)\mathbf{Z} \;=\; B(m_i)\mathbf{Z} \;=\; A(\mathbf{z}, m_i)\mathbf{X}.$$

Finally, given the bases of two affine spaces, a base of their intersection can be computed efficiently. This conclude the proof that $\mathcal{B}$ is $\mathsf{PPT}$.

For the second part, by construction each entry in $F$ is of the form $(m_i, \mathbf{Y}, \mathbf{z})$ such that

$$A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y}.$$

Therefore, by our definition of signatures with linear verification scheme, the verifier accepts $(m_i, \mathbf{Y}, \mathbf{z})$. The claim is thus proven.

*Proof of Claim 4.* Since the condition at step 5 is not satisfied, $S(L, m_i) = \varnothing$ and in particular $\mathbf{z} \notin S(L, m_i)$ implying that $A(\mathbf{z}, m_i)\mathbf{x} \notin \operatorname{Im} B(m_i)$ for some $\mathbf{x} \in L$. Therefore $L$ is not contained in the space of all $\mathbf{x}$ such that $A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$ and in particular its dimension decreases after the execution of step 11

*Proof of Claim 5.* The first part follow as $L$ is initially $V \times \mathbb{F}_q^{n_2}$ of dimension $\dim V + n_2$. The second part follows by linear algebra since $\dim L \geq \dim \pi(L) = \dim V$.

*Proof of Claim 6.* Since $\mathsf{S.SetupKey}$, $\mathsf{S.Sign}$ and $\mathcal{B}$ are $\mathsf{PPT}$ algorithm, by Claim 3 in the last case, each step in the loop can be computed efficiently. In particular, as $2n_1 + 1$ is polynomially bounded, each for-loop in $\mathcal{A}$ can be performed efficiently.

Next we show that the procedure inside the Do-Until loop is repeated at most $n_1 + 1$ times. The key observation is that during the execution of $\mathcal{B}$, the space $L$ forms a monotone decreasing sequence, implying that when $\mathcal{B}(\mathsf{vk}^*, V) \to (F^*, L^*)$ then $L^* \subseteq V \times \mathbb{F}_q^{n_2}$. In particular this implies that $\pi(L^*) \subseteq \pi(V \times \mathbb{F}_q^{n_2}) = V$. Thus if at any point the for-loop is halted, $\pi(L^*) = V^* \neq V$ implies $V^* \subseteq V$. Hence the dimension of $V$ strictly decreases, and since initially $\dim(V) = n_1$, the foor-loop can be halted at most $n_1$ times.

Finally, using again that $\mathcal{B}$ is an efficient algorithm, computing $F, L$ can be done in polynomial time. It follows that $\mathcal{A}$ is $\mathsf{PPT}$.

*Proof of Claim 7.* We proceed by induction. Initially $V = \mathbb{F}_q^{n_1}$ implies $\mathbf{x}_1 \in V$. Next we observe that the value of $V$ is only changed if, within the for-loop, $V^* \neq V$ (see step 8, Fig. 4). Assume by induction that before this step is executed $\mathbf{x}_1 \in V$. Then, when this happens, $\mathcal{B}(\mathsf{vk}^*, V) \to (F^*, L^*)$ had been executed with $\mathbf{x}_1 \in V$. By Claim 2 this implies that $\mathbf{x} \in L^*$ and in particular $\mathbf{x}_1 = \pi(\mathbf{x}) \in \pi(L^*) = V^*$. Thus when $\mathcal{A}$ sets $V \leftarrow V^*$, $\mathbf{x}_1 \in V$.

*Proof of Claim 8.* Define the following events:

- $\mathcal{E}_{i,j}$ = "During the $i$-th iteration of the Do-Until loop, and the $j$-th iteration of the for loop, $\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk}^*, \cdot)}(\mathsf{vk}^*, V)$ returns $(F^*, L^*)$ such that $\pi(L^*) = V$".
- $\mathcal{E}_{\mathsf{last}}$ = "$\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk}, V)$ returns $F, L$ with $\pi(L) = V$".

Furthermore let $I \sim \{1, \ldots, n_1 + 1\}$ be the random variable such that $\mathcal{A}$ terminates the Do-Until loop after the $I$-th execution. Then we observe that, conditioned on $\mathbf{X}_1, s_1$ and the $V$ at iteration $i$, the event $\mathcal{E}_{i,j}$ depends only on the random coins used for $\mathcal{B}$, $\mathsf{S.SetupKey}$ and $\mathsf{S.Sign}$ which are chosen independently at each execution of $\mathcal{B}$. In particular, for a fixed $i$, the events $\{\mathcal{E}_{i,j}\}_j$ are independent and, since for $\mathcal{E}_{i,j}, \mathcal{E}_{i,k}$ with $j \neq k$ the procedure $\mathcal{B}$ is invoked with the same input

$$\Pr[\mathcal{E}_{i,j}] = \Pr[\mathcal{E}_{i,k}].$$

We may therefore define $p_i = \Pr[\mathcal{E}_{i,1}]$ as the success probability of each execution of $\mathcal{B}$ during the $i$-th loop. Similarly, if $I = i$, the vector space $V$ given in input to $\mathcal{B}$ is by construction equal to the one used during the $i$-th execution of the Do-Until loop. In particular

$$p_i = \Pr[\mathcal{E}_{\mathsf{last}} | I = i].$$

To conclude we show that

$$\begin{aligned}
\Pr[\mathcal{A} \to \mathsf{fail}] &= \Pr[\neg\mathcal{E}_{\mathsf{last}}] = \sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \Pr[I = i] \\
&\leq \sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \Pr[\mathcal{E}_{i,1} \wedge \ldots \wedge \mathcal{E}_{i,2n_1+1}] \\
&= \sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \prod_{j=1}^{2n_1+1} \Pr[\mathcal{E}_{i,j}] \\
&= \sum_{i=1}^{n_1+1} (1 - p_i) \cdot p_i^{2n_1+1} \\
&\leq \sum_{i=1}^{n_1+1} \frac{1}{2n_1 + 2} = \frac{n_1 + 1}{2n_1 + 2} = \frac{1}{2}.
\end{aligned}$$

where the first inequality comes from the fact that $I = i$ implies $\mathcal{E}_{i,j}$ for all $j \in \{1, \ldots, 2n_1 + 1\}$, while the second inequality comes from the fact that the

function $f_t(x) = (1-x)x^t$ is upper bounded by $1/(t+1)$ when $x \in [0,1]$. Indeed $f_t(0) = f_t(1) = 0$ and its derivative vanishes only at $t/(t+1)$, which has to be the maximum point, implying that

$$(1-x) \cdot x^t \leq \left(1 - \frac{t}{t+1}\right) \cdot \left(\frac{t}{t+1}\right)^t \leq \frac{1}{t+1}.$$

### C.3 Attack to Schemes with Generic Verification

*Proof of Theorem 3.* Having provided a description of the adversary $\mathcal{A}$, which uses in a block-box way the procedure $\mathcal{B}$ described in Fig. 5, we now show that this adversary satisfies the requirements of the Theorem. As in the previous section we break down the proof into a sequence of Claims which imply the thesis, beginning with a list of properties related to $\mathcal{B}$. In the following we denote $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2$, $\mathbf{x}_1$ the discrete logarithm of $\mathbf{X}_1$ and $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$. Moreover we call again $\pi : \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2}$ the projection on the first component, i.e. $\pi(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1$.

**Claim 1** *For all $B, \mathbf{c}$ and $\mathbf{a}, \mathbf{b}, c$, the set $\mathcal{F}_{\mathbf{a},\mathbf{b},c}^{A,B,\mathbf{c}}$ is an affine space.*

**Claim 2** *An affine base for $\mathcal{F}_{\mathbf{a},\mathbf{b},c}^{A,B,\mathbf{c}}$ is efficiently computable.*

**Claim 3** *If $\mathbf{x}_i \in V$ then at any step of $\mathcal{B}(\mathsf{vk}, V)$, $L \subseteq \mathbb{F}_q^n$ is an affine subspace and $\mathbf{x} \in L$.*

**Claim 4** *If $\mathbf{x}_i \in V$, then $\mathcal{B}(\mathsf{vk}, V)$ performs a polynomially bounded number of queries to the GGM oracles.*

**Claim 5** *Every time either condition at step 23 or 30 is satisfied, when step 24 or 33 respectively are executed, the dimension of $L$ strictly decreases.*

**Claim 6** *When step 27 is executed, then $(\mathbf{Y}, t)$ is a correct signature for $m$.*

**Claim 7** *After step 2, $\dim L = n_2 + \dim V$. Moreover, when $\mathcal{B}(\mathsf{vk}, V)$ returns $(F, L)$ with $\pi(L) = V$, then $\dim L \geq \dim V$.*

Next we provide a sequence of claims about the adversary $\mathcal{A}$, when executed with $\mathcal{B}$ as in Fig. 5.

**Claim 8** *At any given step of $\mathcal{A}(\mathsf{vk})$, $\mathbf{x}_1 \in V$.*

**Claim 9** *$\mathcal{A}$ performs a polynomially bounded number of queries to the GGM oracles.*

**Claim 10** *$\mathcal{A}(\mathsf{vk})$ fails with probability $\Pr[\mathcal{A}(\mathsf{vk}) \to \mathsf{fail}] \leq 1/2$.*

*Proof of Claim 1.* Given $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ we should show that $\mathbf{w} + (\mathbf{u} - \mathbf{v}) \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$. Let $\mathbf{d} = \mathbf{u} - \mathbf{v}$. Then $A\mathbf{u}, A\mathbf{v} \in \operatorname{Im} B + \mathbf{c}$ implies the existence of $\mathbf{r}, \mathbf{s} \in \mathbb{F}_q^m$ such that $A\mathbf{u} = B\mathbf{r} + \mathbf{c}$ and $A\mathbf{v} = B\mathbf{s} + \mathbf{c}$, therefore $A\mathbf{d} = B(\mathbf{r} - \mathbf{s})$. Moreover by construction

$$\mathbf{a}^\top \mathbf{u} = \mathbf{b}^\top \mathbf{r} + \mathbf{c}, \qquad \mathbf{a}^\top \mathbf{v} = \mathbf{b}^\top \mathbf{s} + \mathbf{c} \quad \Rightarrow \quad \mathbf{a}^\top \mathbf{d} = \mathbf{b}^\top (\mathbf{r} - \mathbf{s}).$$

Turning our attention to the vector $\mathbf{w} + \mathbf{d}$, its image through $A$ lies in $\operatorname{Im} B + \mathbf{c}$ because $A\mathbf{d} \in \operatorname{Im} B$. Furthermore for all $\mathbf{y} \in \mathbb{F}_q^m$ such that $B\mathbf{y} + \mathbf{c} = A(\mathbf{w} + \mathbf{d})$

$$\begin{aligned} B(\mathbf{y} - (\mathbf{r} - \mathbf{s})) = A\mathbf{w} \quad &\Rightarrow \quad \mathbf{a}^\top \mathbf{w} = \mathbf{b}^\top (\mathbf{y} - (\mathbf{r} - \mathbf{s})) + c \\ &\Rightarrow \quad \mathbf{a}^\top \mathbf{w} + \mathbf{a}^\top \mathbf{d} = \mathbf{b}^\top \mathbf{y} + c \\ &\Rightarrow \quad \mathbf{a}^\top (\mathbf{w} + \mathbf{d}) = \mathbf{b}^\top \mathbf{y} + c. \end{aligned}$$

Thus $\mathbf{w} + \mathbf{d} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ proving the claim.

*Proof of Claim 2.* Let $H$ a weak left inverse of $B$, then we can rewrite $\mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ as the set of all $\mathbf{x}$ such that $A\mathbf{x} \in \operatorname{Im} B + c$ and

$$\forall \mathbf{w} \qquad \mathbf{a}^\top \mathbf{x} = \mathbf{b}^\top (HA\mathbf{x} + (I - BH)\mathbf{w}) - \mathbf{b}^\top H\mathbf{c} + c$$

using the fact that all solutions to $B\mathbf{y} + \mathbf{c} = \mathbf{t}$ are of the form $H(\mathbf{t} - \mathbf{c}) + (I - BH)\mathbf{w}$ for arbitrary $\mathbf{w}$. Next we introduce an auxiliary space $\mathcal{V}_{\mathbf{r},\mathbf{s},t}$ defined as

$$\mathcal{V}_{\mathbf{r},\mathbf{s},t} \;=\; \{(\mathbf{x}, \mathbf{z}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n \;:\; \mathbf{r}^\top \mathbf{x} = \mathbf{s}^\top \mathbf{z} + t\}$$

Calling $\pi : \mathcal{V}_{\mathbf{r},\mathbf{s},t} \to \mathbb{F}_q^n$ the projection on the first component $(\mathbf{x}, \mathbf{w}) \mapsto \mathbf{x}$ then it's easy to prove that, letting $\mathbf{r} = \mathbf{a} - A^\top H^\top \mathbf{b}$, $\mathbf{s} = (I - BH)^\top \mathbf{b}$ and $t = -\mathbf{b}^\top H\mathbf{c} + c$

$$\begin{aligned} \mathbf{0} \times \mathbb{F}_q^n \nsubseteq \operatorname{dir}(\mathcal{V}_{\mathbf{r},\mathbf{s},t}) \quad &\Rightarrow \quad \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c} \;=\; \varnothing \\ \mathbf{0} \times \mathbb{F}_q^n \subseteq \operatorname{dir}(\mathcal{V}_{\mathbf{r},\mathbf{s},t}) \quad &\Rightarrow \quad \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c} \;=\; \pi(\mathcal{V}_{\mathbf{r},\mathbf{s},t}) \cap \{\mathbf{x} : A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}\} \end{aligned}$$

Since $\mathcal{V}_{\mathbf{r},\mathbf{s},t}$ is an affine hyper-plane, a base of $\operatorname{Im} B + \mathbf{c}$ is efficiently computable, and so is its preimage through $A$. Thus the thesis follows.

*Proof of Claim 3.* $L$ is an affine subspace since initially $L = V \times \mathbb{F}_q^{n_2}$. By induction, assume that, at a given step, $L$ is an affine subspace. Then it is only updated at step 24 or 33. In both cases, by Claim 1, we have that $L$ is the intersection of two affine subspaces.

For the second part, initially $\mathbf{x}_1 \in V$ implies $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 \in V \times \mathbb{F}_q^n = L$. Next assume by induction $\mathbf{x} \in L$. If step 24 is executed then the condition 23 is true, meaning that $\mathbf{x} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$. Therefore $\mathbf{x}$ still lies in $L$. Conversely, if step 33 is executed then by construction the signature satisfies $A\mathbf{X} = B\mathbf{Y} + \mathbf{c}$ meaning that $A\mathbf{x} \in \operatorname{Im} B + \mathbf{x}$. Thus again $\mathbf{x} \in L$ after step 33.

*Proof of Claim 4.* The only instructions involving generic group operation (excluding those using the simulated oracles) in the description of $\mathcal{B}$ (Fig. 5) are in line 23 to check if $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$ and in line 27 to compute $\mathbf{Y}$.

For each message $m_i$, within the cycle starting at step 5, these instructions are executed at most once. Indeed if the condition in line 23 is satisfied then the cycle is halted. Conversely if the condition is not satisfied, line 27 is executed and subsequently the cycle is once again halted. Thus it suffices to show that both these operations can be computed efficiently.

To check that $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$, given a base of this vector space (which can be computed efficiently from Claim 2) it suffices to verify that for each $\mathbf{v}$ in a base of the dual, it holds $\mathbf{v}^\top \mathbf{X} = 1$. Since the dual has dimension at most $n = n_1 + n_2$, this step can be performed using at most $n^2$ external multiplications and $n(n-1)$ additions.

Regarding the second instruction, when it is executed we have that $\mathbf{x} \in L$ by Claim 3 and $A \cdot L \subseteq \operatorname{Im} B + \mathbf{c}$ since the check at step 19 had to fail. In particular $\mathbf{Ax} \in \operatorname{Im} B + \mathbf{c}$. As shown in the proof of Claim 3, Section C.2, one can efficiently compute a vector $\mathbf{Y}_0$ such that $A\mathbf{X} + B\mathbf{Y}_0 = \mathbf{c} \cdot G$, and in particular this requires only polynomially many generic group operations.

To conclude we need to improve this solution in such a way that for all $i \in \{1, \ldots, \chi\}$ such that $\beta_i = 0$, $\mathbf{a}_i^\top \mathbf{X} + \mathbf{b}_i^\top \mathbf{Y}_0 \neq c_i$. To this aim call $I = \{i \leq \chi \,:\, \beta_i = 0\}$ and for all $i \in I$

$$W_i \;=\; \{\mathbf{y} \,:\, \mathbf{a}_i^\top \mathbf{x} + \mathbf{b_i}^\top = c_i\}.$$

First we observe that the solution for the system $A\mathbf{X} + B\mathbf{Y} = \mathbf{c} \cdot G$ is $\mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G$. Indeed for any vector $\mathbf{Y}$, using the fact that $\mathbf{Y}_0$ is a solution too,

$$A\mathbf{X} + B\mathbf{Y} = \mathbf{c} \cdot G \quad \Leftrightarrow \quad B(\mathbf{Y} - \mathbf{Y}_0) = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{Y} \in \mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G.$$

Next, calling $\mathbf{y}_0$ the discrete logarithm of $\mathbf{Y}_0$, i.e. such that $\mathbf{Y}_0 = \mathbf{y}_0 \cdot G$, then for all $i \in I$ we prove that $\mathbf{y}_0 + \operatorname{Ker} B \nsubseteq W_i$. Assuming by contradiction that for some $i$ this is not true, then, since $\mathbf{y}_0 + \operatorname{Ker} B$ is the set of all solutions to the linear system defined by $A, B, \mathbf{c}$, we would have that for all $\mathbf{y}$

$$A\mathbf{x} + B\mathbf{y} = \mathbf{c} \quad \Rightarrow \quad \mathbf{y} \in W_i \quad \Rightarrow \quad \mathbf{a}_i^\top \mathbf{x} + \mathbf{b}_i^\top \mathbf{y} = c_i.$$

In particular $\mathbf{x} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$ which implies $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$, which is a contradiction.

Observing now that $W_i$ are hyperplanes and that $\operatorname{dir} W_i = \{\mathbf{y} \,:\, \mathbf{b}_i^\top \mathbf{y} = 0\}$ we have that either $W_i \cap (\mathbf{y}_0 + \operatorname{Ker} B) = \varnothing$ or $\operatorname{Ker} B \nsubseteq \operatorname{dir} W_i$. Let now $J \subseteq I$ be the set of indices $i$ of those spaces such that $\mathbf{Y}_0 \in W_i$. If $J$ were empty, $\mathbf{Y}_0$ would be the desired solution. Otherwise, for each of these spaces $W_i$, we can efficiently find $\mathbf{u}_i$ such that $\mathbf{u}_i \in \operatorname{Ker} B \setminus \dim W_i$, i.e. $\mathbf{u}_i \in \operatorname{Ker} B$ and $\mathbf{b}_i^\top \mathbf{u}_i \neq 0$.

We claim, but not prove immediately, that a vector $\mathbf{v}$ that is not orthogonal to any $\mathbf{b}_i$ for $i \in J$ and that lies in the span of $\mathbf{u}_{i i \in J}$ can be computed in polynomial time. Assuming for the moment that the latter is true, then we can conclude that the desired solution is one of the vectors in

$$\{\mathbf{Y}_0 + (\alpha \cdot \mathbf{v}) \cdot G \,:\, \alpha \in \{1, \ldots, \chi + 1\}\}.$$

Indeed all these points (all distinct because $\chi - 1 < q$, the order of $\mathbb{G}$) lie on an affine line passing through $\mathbf{Y}_0$ and with direction $\mathbf{v}$. Since $\mathbf{u}_i \in \operatorname{Ker} B$, then also the vector $\mathbf{v} \in \operatorname{Ker} B$ as it belongs to the span of $\{\mathbf{u}_i\}_{i \in J}$, implying that the line (and in particular the set described above) is contained in $(\mathbf{y}_0 + \operatorname{Ker} B) \cdot G$. Next, by construction $\mathbf{Y}_0 \in W_i$ for $i \in J$ but $\mathbf{v} \notin \operatorname{dir} W_i$, we have that these hyperplanes intersect the line only in $\mathbf{Y}_0$. Conversely the hyperplanes $W_i$ with $i \notin J$ by definition do not contain the point $\mathbf{Y}_0$ and in particular can intersect the line in at most 1 point. As the number of these spaces is at most $\chi$, by the pigeonhole principle at least one among $\chi + 1$ points on the line does not belong in any of them. As checking membership in $W_i$ can be done with polynomially many group operations as shown before, and $\chi$ is polynomially bounded, we can find a point $\mathbf{Y} \in \mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G$ such that $\mathbf{Y} \notin W_i \cdot G$ for all $i \in I$. From the way we defined $W_i$ we thus proved that

$$\begin{aligned} \mathbf{Y} \in \mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G \quad &\Rightarrow \quad A\mathbf{X} + B\mathbf{Y} = \mathbf{c} \\ \mathbf{Y} \notin W_i \cdot G \quad &\Rightarrow \quad \mathbf{a}_i^\top \mathbf{X} + \mathbf{b}_i^\top \mathbf{Y} = c_i \cdot G. \end{aligned}$$

That is a solution to the given system.

Before concluding the proof of the claim we are left with showing an algorithm for computing $\mathbf{v}$ efficiently. This is presented if Figure 12. We show correctness

$\mathsf{ExtPoint}(\mathbf{b}_1, \ldots, \mathbf{b}_n, \mathbf{u}_1, \ldots, \mathbf{u}_n)$:

```
1 : If n = 1:    Return u_1
2 : Else:
3 :    v ← ExtPoint(b_1, ..., b_{n-1}, u_1, ..., u_n)
4 :    If v^T u ≠ 0: Return v
5 :    Else:
6 :       For α ∈ {1, ..., n}:
7 :          If (αv + u_n)^⊥ b_i = 0 for all i ∈ {1, ..., n − 1}:
8 :             Return αv + u_n
```

**Fig. 12.** ExtPoint, given $\mathbf{b}_i^\top \mathbf{u}_i \neq 0$ returns $\mathbf{v}$ a linear combination of $\mathbf{u}_i$ s.t. $\mathbf{b}_i^\top \mathbf{v} \neq 0$.

by induction. If $n = 1$, $\mathbf{v} = \mathbf{u}_1$ is not orthogonal by hypothesis to $\mathbf{b}_1$ and trivially lies on the span of $\mathbf{u}_1$. Assuming that correctness holds for $n - 1$, then the intermediate vector $\mathbf{v}$ computed is not orthogonal to any $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$ and is a linear combination of $\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}$. If $\mathbf{v}$ is also non orthogonal to $\mathbf{b}_n$, then it satisfies the desired property. Conversely, we have that for all $\alpha \in \{0, 1\}$

$$(\alpha\mathbf{v} + \mathbf{u}_n)^\top \mathbf{b}_n \;=\; \mathbf{u}_n^\top \mathbf{b}_n \;\neq\; 0.$$

Regarding the other vector observe that for each $i$ there can exists at most one $\alpha \in \mathbb{F}_q$ such that $(\alpha \mathbf{v} + \mathbf{u}_n)^\top \mathbf{b}_i = 0$, that is

$$\alpha \;=\; \frac{\mathbf{u}_n^\top \mathbf{b}_i}{\mathbf{v}^\top \mathbf{b}_i}.$$

Since $n < q$ the values of $\alpha$ we consider are all distinct and by the pigeonhole there has then to exists an $\alpha$ such that $(\alpha \mathbf{v}^\top + \mathbf{u}_n)$ is not orthogonal to any $\mathbf{b}_i$ for $i < n$. In conclusion $(\alpha \mathbf{v}^\top + \mathbf{u}_n)$ is not orthogonal to any $\mathbf{b}_i$ and it is, by the inductive hypothesis, a linear combination of $\mathbf{u}_1, \ldots, \mathbf{u}_n$. This concludes the proof.

*Proof of Claim 5.* First of all, if condition at step 23 is satisfied, then by construction the condition at step 21 is not (or step 23 is not executed). Hence $L \cap \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$ is a proper subspace of $L$, which implies that after step 24 the dimension of $L$ decreases strictly.

If instead the condition at step 33 is satisfied, as done for the proof of Claim 6, we can assume with loss of generality that $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ share the same labels of $\mathbf{X}$ and $\mathbf{Y}$. Executing $\mathsf{S.Vfy}$ on input $(\mathbf{X}, s, m, \mathbf{Y}, t)$ and calling $\beta_1, \ldots, \beta_\chi$ the bits returned by $\mathcal{O}^0_{\mathsf{eq}}$ invoked during this execution, we have that at least one of the conditions on steps 17, 19 or 21 is satisfied (since the other conditions would break the cycle). However

- If the check on step 17 is satisfied then $\mathsf{S.Vfy}$ on input $(\mathbf{X}, s, m, \mathbf{Y}, t)$ would output 0, meaning that the queried signature is rejected. This contradicts correctness of the underlying scheme.
- If the test on step 21 is satisfied, by Claim 3 $\mathbf{x} \in L$ and in particular $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$ for some $i$ such that $\beta_i = 0$. By construction this means that for any $\mathbf{Y}'$ that satisfies $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$, then $\mathbf{a}_i^\top \mathbf{X} = \mathbf{b}_i^\top \mathbf{Y}' + c_i \cdot G$.
  As $A, B, \mathbf{c}$ consist of the linear constraints tested by $\mathsf{S.Vfy}$ that $\mathbf{X}$, $\mathbf{Y}$ and $G$ verifies, then $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$. In particular $\mathbf{a}_i^\top \mathbf{X} = \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G$, which contradicts the hypothesis that $\mathcal{O}^0_{\mathsf{eq}}(\,\cdot\,)$ returns $\beta_i = 0$ for the $i$-th query.

In conclusion we obtain that $A \cdot L \nsubseteq \operatorname{Im} B + \mathbf{c}$, implying that $\{\mathbf{x} \in L \;:\; A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}\}$ is a proper subspace of $L$. Hence after step 33 the dimension of $L$ strictly decreases.

*Proof of Claim 6.* First of all we remark that step 27 can be efficiently computed since $A \cdot L \subseteq \operatorname{Im} B + \mathbf{c}$, which by Claim 3 implies that, calling $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$, $A\mathsf{x} \in \operatorname{Im} B + \mathbf{c}$, and $\mathbf{x} \notin \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$. In particular, calling $V$ the affine space of solutions $\mathbf{y} \in \mathbb{F}_q^k$ such that $A\mathbf{x} = B\mathbf{y} + \mathbf{c}$, and $V_i$ the subspace of $V$ such that $\mathbf{a}_i^\top \mathbf{x} = \mathbf{b}_i^\top \mathbf{y} + c_i$ we have two possible cases:

- $\dim V = 0$: then since condition on step 23 is not satisfied, $V_i \subsetneq V$, which means that $V_i = \varnothing$ for all $i$ such that $\beta_i = 0$. In particular $\mathbf{Y}$ is the only solution to the system $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$, which is obtained by

$$\mathbf{Y} = H \cdot (A\mathbf{X} - \mathbf{c} \cdot G)$$

with $H$ a left weak inverse of $B$, and it automatically satisfies all the conditions of the form $\mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}^\top \mathbf{Y} + c_i \cdot G$.

– $\dim V \geq 1$: then since condition on step 23 is not satisfied, $V_i \subsetneq V$ is a proper subspace of $V$. It follows that $\bigcup_{i:\beta_i=0} V_i$ contains at most $\chi \cdot q^{\dim V - 1}$ points. Since $\chi$ is polynomially bounded (or the verification algorithm would not be efficient) $\chi < q$ and in particular the set $V \setminus \bigcup_{i:\beta_i=0} V_i$ is not empty and a vector $\mathbf{Y}$ satisfying the above conditions can be obtained for instance with overwhelming probability greater than $1 - \chi \cdot q^{-1}$ by setting

$$\mathbf{Y} = H(A\mathbf{X} - \mathbf{c} \cdot G) + (I - HB)\mathbf{w} \cdot G$$

with $H$ a weak left inverse of $B$ and $\mathbf{w} \leftarrow^{\$} \mathbb{F}_q^n$.

Finally to show that $(\mathbf{Y}, t)$ is a valid signature we first observe that without loss of generality we may assume that $\mathbf{X}$ and $\mathbf{Y}$ have the same labels of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$. Indeed $\mathcal{A}$ can set $\widetilde{\mathbf{X}}$ with the same labels of $\mathbf{X}$ and $\widetilde{\mathbf{Y}}$ with large enough labels so that, for a given $\mathbf{Y} \in \mathbb{G}^k$, up to performing dummy operations (adding zeroes for instance), it is possible to obtain a new vector $\mathbf{Y}'$ representing the same group elements of $\mathbf{Y}$ and with the same labels of $\mathbf{Y}$.

As the inputs $(\mathbf{X}, s, m, \mathbf{Y}, t)$ and $(\widetilde{\mathbf{X}}, s, m, \widetilde{\mathbf{Y}}, t)$ are equal, the deterministic algorithm $\mathsf{S.Vfy}$ performs the same queries to $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}^0_{\mathsf{eq}}$. In particular, for the $i$-th query, if $\beta_i = 0$ then $\mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G$, meaning that $\mathcal{O}^0_{\mathsf{eq}}(\cdot)$ returns $0 = \beta_i$. Conversely if $\beta_i = 1$, then $\mathbf{a}^\top \mathbf{X} = \mathbf{b}^\top \mathbf{Y} + c_i \cdot G$, meaning that $\mathcal{O}^0_{\mathsf{eq}}(\cdot)$ returns $1 = \beta_i$. Hence the execution of $\mathsf{S.Vfy}$ produces the same output $b$ it returns when executed with the simulated group $\widetilde{\mathbb{G}}$. As condition on step 17 is not satisfied, $b = 1$, meaning that the signature $(\mathbf{Y}, t)$ is valid for $m$.

*Proof of Claim 7.* Initially $L = V \times \mathbb{F}_q^{n_2}$ implies $\dim L = n_2 + \dim V$. If $\mathcal{B}(\mathsf{vk}, V) \to (F, L)$ with $\pi(L) = V$ then $\dim L \geq \dim \pi(L) \geq \dim V$.

*Proof of Claim 8.* Analogous to the proof of Claim 7.

*Proof of Claim 9.* As in the proof of Claim 6, it can be shown that $\mathcal{A}$ executes its subroutine a polynomial number of times. By Claim 4 each execution requires a polynomial number of queries to the GGM and not other query is performed explicitly by $\mathcal{A}$. The thesis follows.

*Proof of Claim 10.* Analogous to the proof of Claim 8.