On the Impossibility of Algebraic NIZK In Pairing-Free Groups

Emanuele Giunta[1,2]

[1] IMDEA Software Institute, Madrid, Spain.
emanuele.giunta@imdea.org
[2] Universidad Politecnica de Madrid, Spain.

Abstract. Non-Interactive Zero-Knowledge proofs (NIZK) allow a prover to convince a verifier that a statement is true by sending only one message and without conveying any other information. In the CRS model, many instantiations have been proposed from group-theoretic assumptions. On the one hand, some of these constructions use the group structure in a black-box way but rely on pairings, an example being the celebrated Groth-Sahai proof system. On the other hand, a recent line of research realized NIZKs from sub-exponential DDH in pairing-free groups using Correlation Intractable Hash functions, but at the price of making non black-box usage of the group.

As of today no construction is known to *simultaneously* reduce its security to pairing-free group problems and to use the underlying group in a black-box way.

This is indeed not a coincidence: in this paper, we prove that for a large class of NIZK either a pairing-free group is used non black-box by relying on element representation, or security reduces to external hardness assumptions. More specifically our impossibility applies to two incomparable cases. The first one covers Arguments of Knowledge (AoK) which proves that a preimage under a given one way function is known. The second one covers NIZK (not necessarily AoK) for hard subset problems, which captures relations such as DDH, Decision-Linear and Matrix-DDH.

Table of Contents

1 Introduction

Zero-Knowledge proofs are protocols through which a *prover* can convince a *verifier* that a given statement is true without revealing any other information. *Non-interactive* zero-knowledge proofs (NIZK) [BFM88] are often preferable in concrete applications but it is well known that instantiating them in the standard model is impossible. To overcome this limitations instantiations in the random oracle model (ROM) and the common reference string (CRS) model have been studied.

In the ROM a public hash function is modeled as being truly random, used in the Fiat-Shamir transform [FS87] to replace interaction with a public coin verifier. In the CRS model a set of public parameters are set up by a trusted party before the protocol starts. In these two settings, a variety of techniques to build NIZK for NP-complete relations have been put forward. However, when applying them to languages of interest in cryptographic applications, for instance those based on prime order (pairing) groups, the reduction to a target NP-complete problem introduces overheads which limit applicability. This typically occurs when proving statements related to ElGamal ciphertexts or Pedersen commitments. For this reason NIZK that operate efficiently for group-theoretic statements, and whose hardness only relied on group-theoretic problems have been studied.

In the ROM, positive results came from the Schnorr sigma protocol [Sch90], and its subsequent generalizations [Cra96, Mau09], which eventually led to sub-linear arguments [BCC+16, BBB+18] for all NP solely based on hard problems over groups. This approach, however, can provide at best *computationally* sound arguments in the ROM and unclear security guarantees when replacing the RO with a hash function.

For these reasons a parallel line of research investigated NIZKs from group assumptions in the CRS model. Early work proved feasibility in the pairing setting [Gro06, GOS06a, GOS06b] eventually leading to the celebrated *statistically sound* Groth-Sahai proof system [GS08]. Still in the pairing setting [CH20] recently proposed different techniques to compile Schnorr-like sigma-protocols [Sch90, Cra96, Mau09] capturing linear languages. Interestingly, both constructions only make black-box usage of the underlying group, which makes the resulting NIZK simpler, randomizable and able to support aggregation for specific languages.

A more recent approach is to instantiate the Fiat-Shamir transform through correlation-intractable hash (CIH) functions [CGH04]. Recently [JJ21] realized CIH and NIZK only assuming sub-exponential DDH. This sparkled renewed interest and led to new constructions including designated-verifier NIZK from CDH [KNYY21], SNARGs for P and batch-NP [CGJ+22] and for log-space bounded-depth circuits [KLV22]. On the one hand, this proved once again that the gap between pairing-based assumptions and plain prime order group ones is thinner than we expected. On the other hand, as opposed to Groth-Sahai, all CIH-based NIZKs make non black-box usage of the underlying group by relying on element representations, for instance to hash group elements, and thus breaking the algebraic structure. Given the state of the art we therefore ask whether "best of both worlds" constructions are possible, i.e.

Do there exist NIZK based only on pairing-free prime-order group problems and that only use the group in a black-box way?

A positive answer would be interesting in practice since pairing-free groups are generally more efficient than their pairing-friendly counterpart, possibly providing faster NIZKs that Groth-Sahai. Moreover, black-box usage of the group may preserve the algebraic structure yielding NIZK with homomorphic properties, something currently not known to be possible via CIH.

Our Result. We answer the above question in the negative by showing that for a large class of relations defined over a known prime order pairing free group, any NIZK satisfies at least one of the following:

1. Relies on group element representations, thus using the group in a non black-box way.
2. Its security does not *only* depend on hard problems over the group.
3. It assumes additional algebraic structure (such as pairings).

This informally implies that the usage of pairing in black-box constructions such as Groth-Sahai cannot be removed without relying on group elements representation. Analogously, the usage of elements representation in all recent constructions from sub-exponential DDH cannot be removed without introducing more structure (like pairings or unknown order) or assumptions (like LWE).

Following the approach of recent negative results [DHH+21, CFGG23], we formalize *black-box* access to the group by assuming all procedures to be defined in Maurer's Generic Group Model (GGM) [Mau05], where the group is replaced by addition and equality check oracles, and no representation is ever given (as opposed to Shoup's Group Model [Sho97]). In this model, we isolate hard group-theoretic problems by assuming unbounded adversaries that are constrained to use the group efficiently, as done in [SGS21, DHH+21, CFGG23]. We will later refer to this class of adversaries as GPPT, standing for *Generic-Group* PPT. More in detail, we prove two independent impossibility results:

Impossibility for Arguments of Knowledge. Our first result shows that, in Maurer's GGM, given a one-way function (family) $f_k : \{0,1\}^n \to \mathbb{G}^m$ whose hardness reduces to hard group problems, there exists no NIZK-Argument of Knowledge (NIZK-AoK) to prove that a preimage x of y for the function f_k is known. In other words there exists no NIZK-AoK for the NP-relation

$$\mathcal{R} = \{((k, y), x) \;:\; f_k(x) = y\}.$$

If f_k is a one-way function family, we need to further assume that the key k only contains uniformly random group elements for the proof to go through. Although this mildly affects generality, we capture virtually all cases of interest.

Examples are discrete logarithm and linear maps $f : \mathbb{F}_q^n \to \mathbb{G}^m$, KZG [KZG10] "powers of τ" setup where $f_G : \mathbb{F}_q \to \mathbb{G}^n$ with key the group generator G and $f_G(\tau) = (\tau^i \cdot G)_{i=1}^n$, PST [PST13] multivariate CRS and range proofs.

Impossibility for Computationally Sound NIZK. Our second negative result addresses a class of relations $\mathcal{R}$ for which it is possible to sample statements inside and outside the associated language[3] so that distinguishing the two distributions is computationally hard. An example is the DDH relation, for which distinguishing DDH tuples[4] from random ones is hard. These relations were called in [GW11] *hard subset membeship problems* and for them we show that in Maurer's GGM no NIZK (not necessarily AoK) exists. This essentially captures all group-theoretic decisional problems including DDH, DLin, Matrix DDH and others.

Results Comparison. Finally, we remark that these two results are incomparable. The impossibility of NIZK-AoK for the discrete logarithm relation cannot follow from our second result, as the language associated to the discrete logarithm problem is trivial (every group element admits a discrete logarithm), and therefore is not a hard subset membership problem. Conversely, our second result is not implied by the first one because it addresses a weaker class of NIZKs, which only need to satisfy soundness but need not to be extractable.

Our Techniques. We now give an overview of the ideas and challenges involved in our result. Our starting point are two recent papers by Döttling, Hartmann, Hofeinz, Kliz, Schäge, Ursu [DHH+21] and Catalano, Fiore, Gennaro, Giunta [CFGG23] which respectively proved impossibility of Algebraic Signatures and Vector Commitments (VC) [LY10, CF13] in Maurer's GGM.

Techniques for Arguments of Knowledge. Regarding our first result, the main idea is to reduce NIZK-AoK for a one way function f to VC in such a way that the impossibility in [CFGG23] carries over to NIZKs. More specifically, in order to contradict [CFGG23] one would have to produce a VC for n messages with commitment c and opening Λ of bit length $|c|$ and $|\Lambda|$ respectively such that $|c| \cdot |\Lambda| = o(n)$. Unfortunately, the only compressing commitments known over groups are variations of Pedersen scheme [Ped92], which comply with the lower bound, and NIZKs do not seem to provide an edge for compression as we do not assume succinctness.

We overcome this limitation with the following critical observation. If the VC *hides* unopened entries, the lower bound in [CFGG23] can be improved to $\ell_c \geq n$, with ℓ_c being the number of group elements in the commitment and n the length of the committed vector. This clears the path for the following strategy: committing to a vector of field elements via Pedersen, and then relying on a

[3] Or, more generally, so that with overwhelming probability the sampled elements lies inside and outside the language respectively.

[4] i.e. tuples of the form $(G, a \cdot G, b \cdot G, ab \cdot G)$ for random a, b

NIZK-AoK to prove knowledge of the openings. Notice that opening proofs in this case may be of linear size, which does not contradict [CFGG23]. However, since the commitment would have constant size, we do violate our improved lower bound.

In order to build intuition and illustrate the challenges that arise in this construction let us present a toy instantiation for the simple case in which $f_{G_i}(x_i) = x_i \cdot G_i$ is the group exponentiation with $G_i \in \mathbb{G}$. Applying directly the idea above a commitment C to $(x_1, \ldots, x_n)$ and an opening Λ at position i should be set as

$$\begin{aligned} C &= f_{G_1}(x_1) + \ldots + f_{G_n}(x_n) = x_1 G_1 + \ldots + x_n G_n \\ \Lambda_i &= (f_{G_j}(x_i), \pi_j)_{j \neq i} = (x_j G_j, \pi_j)_{j \neq i} \end{aligned}$$

with π_j being an AoK for x_j. This approach however fails to hide x_j as it leaks $x_j G_j$. To overcome this issue we rely on the Goldreich-Levin hardcore predicate [GL89]. Their result informally says that for any OWF f, sampling an input x (viewed as a bit-string) and a random string of the same length r, is hard to guess the bit-wise inner product $x^\top r$ modulo 2 given only $f(x)$ and r. With this tool we can turn the above construction into a VC to bits, where each committed bit is a Goldreich-Levin hardcore predicate, which remains hidden even after leaking $f(x)$.

In the toy example with f being the group exponentiation, the resulting commitment for $b_1, \ldots, b_n$ is obtained by sampling x_i, r_i with $x_i^\top r_i = b_i$ and x_i field element and setting

$$C = (x_1 G_1 + \ldots + x_n G_n,\ r_1, \ldots, r_n), \qquad \Lambda_i = (x_j G_j, \pi_j)_{j \neq i}.$$

Another challenge is that we need our OWF f to be collision resistant, which is trivially true for the exponentiation, but false in general. We address this by showing a rather technical way to make any OWF family whose key consist of random group elements collision resistant. Although this transformation renders f computable only in GPPT time, more specifically in exponential space[5] but only polynomially many group operations, this suffices to obtain our result.

Techniques for Computationally Sound NIZK. Regarding our second result, we fail to provide a simple reduction to primitives known to be impossible and instead we directly adapt the approach of [DHH+21, CFGG23]. Their attack in the setting of digital signatures works as follows: initially the adversary is given a verification key vk. For each message it then attempts to extract a signature by brute-force running the verifier on all possible inputs. Eventually, either a forgery is found or a signature query reveals a linear relation on the group elements of vk. Thus after sufficiently many attempts a forgery will be found.

We begin by sketching an adaptation to the simpler case of *simulation-sound* NIZKs, where the adversary has to prove a false statement given oracle access to the simulator. In this case one may hope that replacing the signature and

[5] which explains why we do not violate the black-box separation of [Sim98].

verification procedures with the NIZK simulator and verifier, the attack would carry over by letting the adversary try to produce a proof for false statements in the same way.

One issue though is that the simulator is not guaranteed to work with *false* statements, and may simply return an error if it recognizes one as such. We therefore restrict our focus on hard subset membership problems, in which false statements can be sampled so that deciding their correctness is hard. In this way the simulator almost always returns an accepting proof, and in particular the attack eventually succeeds.

The final challenge is using this adversary, let us call it $\mathcal{A}$, to break regular soundness where no simulator oracle is provided. Our solution is reminiscent of the strategy to simulate the folklore proof for graph isomorphism against malicious verifiers: Given the crs, we begin by tossing a coin b, that is a guess on the behavior of $\mathcal{A}$: If $b = 1$ we sample a true statement with its witness (x_1, w_1), otherwise we sample a false one x_0. Next we run $\mathcal{A}$ on x_b. If $\mathcal{A}$ returns a proof and we guessed $b = 0$, then this breaks soundness. Conversely if $\mathcal{A}$ asks for a simulated proof and we guessed $b = 1$, the proof can be computed with the NIZK prover using w_1. In this second case $\mathcal{A}$'s output will contain a linear relation among the group elements in the crs. If the guess is not correct we can resample a fresh b and x_b and repeat. Since distinguishing x_0 from x_1 is hard, each guess is correct with probability close to $1/2$ and with sufficiently many guesses the attack succeeds with significant probability.

Related Work. Since the seminal work by Impagliazzo and Rudich [IR89], many papers studied the relations among cryptographic primitives through black-box reduction, and, on the negative side, black-box separations [Sim98, KST99, GT00, GKM+00, GMR01, GGK03].

This includes the study of what primitives can and cannot be built over black-box pairing-free known prime order groups, typically modeled as Maurer's or Shoup's GGM. Papakonstantinou, Rackoff and Vahlis [PRV12] were the first to study the impossibility of Identity-Based Encryption in the Shoup GGM, result later tightened by [SGS21] in Maurer's GGM and fully proved in [Zha22] in Shoup's Model. Recent works showed impossibility in Maurer's GGM for several other primitives, such as verifiable delay functions [RSS20], digital signatures [DHH+21] and vector commitments [CFGG23], where the last two are known to exist in Shoup's GGM. In this view our work places NIZK as another primitive which separates Maurer's model from Shoup's one.

Regarding general impossibility results for NIZK, Gentry and Wichs's celebrated result [GW11] shows that succinct arguments cannot be based on falsifiable assumption assuming sub-exponential hardness. We stress that our problem is orthogonal to their result as we don't assume succinctness. However, we ask for black box usage of a group, which is more restrictive than simply assuming a reduction to (falsifiable) group-theoretic problems.

Abe, Camenisch, Dowsley, Dubovitskaya proved in [ACDD19] a general impossibility result for deterministic structure-preserving primitives which captures among others PRF, VRF and unique signatures. Their result however does not cover NIZKs as proofs may not be unique.

More recently Ganesh, Khoshakhlagh, Parisella [GKP22] showed that the Couteau-Hartmann framework [CH20] to instantiate the Fiat-Shamir transform in pairing groups fails to achieve a stronger notion of extraction when no knowledge assumption is used.

We finally remark that Bellare and Goldwasser [BG90] and later Goldwasser and Ostrovsky [GO93] proved the equivalence between NIZKs and invariant signatures. This, in combination with the impossibility result for digital signatures in Maurer's GGM [DHH$^+$21, CFGG23], yields a weaker version of our second result, excluding *simulation sound* NIZKs only for certain expressive relations[6]

2 Preliminaries

2.1 Notation

In the following we denote $[n] = \{1, \ldots, n\}$. $\mathbb{F}_q$ for a prime q is the field of order q, isomorphic to the integers modulo q. $V \leq \mathbb{F}_q^n$ means that V is an affine subspace of $\mathbb{F}_q^n$. Given a group $\mathbb{G}$ of known prime order q, we call G its canonical generator. Although we will use additive notation for groups, given $a \in \mathbb{F}_q$ and $H \in \mathbb{G}$ we will refer to the operation $a \cdot H$ as an *exponentiation*. We also assume operations on vectors and sets are entry-wise, that is given $\mathbf{v} \in \mathbb{F}_q^n$, $V \leq \mathbb{F}_q^n$ and $H \in \mathbb{G}$, $\mathbf{v} \cdot G = (v_1 G, \ldots, v_n G)$ and $V \cdot G = \{\mathbf{v} \cdot G \,:\, \mathbf{v} \in V\}$. For a set $\mathcal{X}$, $x \sim \mathcal{X}$ means x is a random variable with support in $\mathcal{X}$ whereas $x \sim U(\mathcal{X})$ means x is uniformly distributed over $\mathcal{X}$. Given $x, y \sim \mathcal{X}$ we denote $\Delta(x, y)$ their statistical distance

$$\Delta(x, y) \;=\; \frac{1}{2} \sum\nolimits_{t \in \mathcal{X}} \Big| \Pr[x = t] - \Pr[y = t] \Big| .$$

2.2 Maurer's Generic Group Model

Maurer's Generic Model, introduced in [Mau05] and revised in [Zha22], is a framework to describe generic computation. Since we are interested in procedures defined over a group we will refer to this special case as the Generic Group Model (GGM). In this setting, a group $\mathbb{G}$ of known prime order q is modeled by a stateful oracle machine $\mathcal{O}$ along with an internal list of group elements V of length n. The list initially only contains one generator, i.e. $V = (G)$ and $n = 1$. Operations over the group can be performed through oracle queries $\mathcal{O}_{\mathsf{add}}$, $\mathcal{O}_{\mathsf{eq}}$ to $\mathcal{O}$. More specifically

- When $\mathcal{O}_{\mathsf{add}}(i, j)$ is queried with $i, j \leq n$, $\mathcal{O}$ computes $V_{n+1} \leftarrow V_i + V_j$ and appends the result to V.
- When $\mathcal{O}_{\mathsf{eq}}(i, j)$ is queried with $i, j \leq n$, $\mathcal{O}$ computes the bit $b \leftarrow V_i == V_j$ and returns b.

[6] More specifically given a PRF f, a perfectly binding commitment c to a PRF key k, and public inputs x and y the NIZK has to prove that $y = f_k(x)$.

We remark that the above description follows [Zha22] revision, which removes the possibility to specify in $\mathcal{O}_{\mathsf{add}}$ queries in which entry of V the result should be stored, appending it instead at the end of the list by default. Throughout the rest of this paper, to improve readability, we will use $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}_{\mathsf{eq}}$ with group elements instead of indices (implicitly associating group elements to indices).

An important class of adversaries in the GGM used to isolate hard problems in the group from other source of hardness is the following.

Definition 1. GPPT *is the class of all (unbounded) probabilistic Turing Machines with access to* $\mathcal{O}_{\mathsf{add}}, \mathcal{O}_{\mathsf{eq}}$ *whose number of oracle queries is polynomially bounded in their input length.*

This class was implicitly introduced when proving lower bound for computational and decisional problems such as discrete logarithm, DDH and CDH, as well as in recent impossibility result [RSS20, DHH+21, CFGG23].

2.3 NIZK-AoK

A Non-Interactive Zero-Knowledge argument (NIZK) for a relation $\mathcal{R}$ is a tuple of three algorithms $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ that allow a prover to convince a verifier about the validity of a statement without leaking any other information. Given $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$ and $(x, w) \in \mathcal{R}$ a valid statement, the prover can compute a proof running $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ which can later be verified by $b \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$. The proof is accepted if $b = 1$, or rejected otherwise. Below we revise formally the main properties NIZKs can satify.

Completeness: $\forall (x, w) \in \mathcal{R}$

$$\Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi) \;\middle|\; \mathsf{crs} \leftarrow \mathsf{G}(1^\lambda),\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\right] \;=\; 1.$$

Soundness: $\exists \varepsilon$ negligible such that $\forall x \;:\; \nexists w \;:\; (x, w) \in \mathcal{R}$ and $\forall \mathcal{A}$ PPT

$$\Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi) \;\middle|\; \mathsf{crs} \leftarrow \mathsf{G}(1^\lambda),\ \pi \leftarrow \mathcal{A}(\mathsf{crs}, x)\right] \;\leq\; \varepsilon(\lambda).$$

Argument of Knowledge: For any PPT adversary $\mathcal{A}$ there exists a PPT extractor E such that

1. $\exists \varepsilon$ negligible such that $\forall \mathcal{D}$ PPT, given $\mathsf{crs}_0, \mathsf{td} \leftarrow \mathsf{E}(1^\lambda),\ \mathsf{crs}_1 \leftarrow \mathsf{G}(1^\lambda)$

$$\left|\Pr\left[1 \leftarrow \mathcal{D}(\mathsf{crs}_0)\right] - \Pr\left[1 \leftarrow \mathcal{D}(\mathsf{crs}_1)\right]\right| \leq \varepsilon(\lambda).$$

2. There exists a negligible function ε such that

$$\Pr\left[\begin{matrix}\mathsf{V}(\mathsf{crs}, x, \pi) \rightarrow 1 \\ (x, w) \notin \mathcal{R}\end{matrix} \;\middle|\; \begin{matrix}\mathsf{crs}, \mathsf{td} \leftarrow \mathsf{E}(1^\lambda),\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \\ w \leftarrow \mathsf{E}(\mathsf{td}, x, \pi)\end{matrix}\right] \leq \varepsilon(\lambda).$$

Zero-Knowledge: There exists a PPT simulator S such that, up to negligible probability ε, for all $(x, w) \in \mathcal{R}$ and PPT adversary $\mathcal{A}$, given

$$\mathsf{crs}_0, \mathsf{td} \leftarrow \mathsf{S}(1^\lambda), \quad \pi_0 \leftarrow \mathsf{S}(\mathsf{td}, x), \quad \mathsf{crs}_1 \leftarrow \mathsf{G}(1^\lambda), \quad \pi_1 \leftarrow \mathsf{P}(\mathsf{crs}_1, x, w)$$

$$\Rightarrow \quad \left|\Pr\left[1 \leftarrow \mathcal{A}(\mathsf{crs}_0, \pi_0)\right] - \Pr\left[1 \leftarrow \mathcal{A}(\mathsf{crs}_1, \pi_1)\right]\right| \leq \varepsilon(1^\lambda).$$

In the rest of the paper we will say that in Maurer's Generic Group Model, a NIZK is *algebraic* if soundness and zero-knowledge hold against any GPPT adversary, i.e. with unbounded computational power but limited to perform a polynomially bounded number of queries to the GGM oracles. Analogously an Algebraic NIZK-AoK is an argument of knowledge against GPPT adversaries.

2.4 Digital Signatures

A Digital Signature scheme is a tuple of three algorithms $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ along with a set $\mathsf{S.M}$ such that

- $\mathsf{S.Setup}(1^\lambda) \to \mathsf{vk}, \mathsf{sk}$ generates the verification signature key
- $\mathsf{S.Sign}(\mathsf{sk}, m) \to \sigma$ with $m \in \mathsf{S.M}$, signs the message m.
- $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 0/1$ check the validity of the signature.

A signature scheme is correct if given vk, sk generated by the setup procedure and for all $m \in \mathsf{S.M}$, computing $\sigma \leftarrow \mathsf{S.Sign}(\mathsf{sk}, m)$ yields a valid signature, i.e. such that $\Pr[\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1] = 1$.

A signature scheme is *unforgeable* against a class of adversaries, if no adversary in this class on input vk and oracle access to $\mathsf{S.Sign}(\mathsf{sk}, \cdot)$ can return a signature on a message that was not queried with non negligible probability.

2.5 Vector Commitments

Vector Commitment (VC) [LY10, CF13] is a primitive that allows a user to commit to a vector of messages and later on reveal entries of its choice. A VC scheme consists of four algorithms $(\mathsf{VC.Setup}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy})$ and a message space $\mathsf{VC.M}$ such that

- $\mathsf{VC.Setup}(1^\lambda) \to \mathsf{pp}$ generates the public parameters.
- $\mathsf{VC.Com}(\mathsf{pp}, x_1, \dots, x_n) \to (c, \mathsf{aux})$ with $x_1, \dots x_n \in \mathsf{VC.M}$.
- $\mathsf{VC.Open}(\mathsf{pp}, i, \mathsf{aux}) \to \Lambda$ produces an opening proof for position i.
- $\mathsf{VC.Vfy}(\mathsf{pp}, c, x, i, \Lambda) \to 0/1$ check the validity of the opening proof.

A vector commitment is correct if, given pp generated by the setup algorithm, c a commitment to $(x_1, \dots, x_n)$ with auxiliary information aux and Λ an opening proof for position i, then $\Pr[\mathsf{VC.Vfy}(\mathsf{pp}, c, x, i, \Lambda) \to 1] = 1$.

The main notion of security for VC is *position binding*. This generalizes the analogous notion for standard commitments as it requires that opening a position to two different messages is computationally hard. More formally, given $\mathsf{pp} \leftarrow \mathsf{VC.Setup}(1^\lambda)$, for all adversary $\mathcal{A}$ there exists a negligible ε such that

$$\Pr\left[\begin{array}{l}\mathcal{A}(1^\lambda, \mathsf{pp}) \to (c, i, x_0, \Lambda_0, x_1, \Lambda_1), \\ \forall b \in \{0,1\} \quad \mathsf{VC.Vfy}(\mathsf{pp}, c, x_b, i, \Lambda_b) \to 1\end{array}\right] \leq \varepsilon(\lambda).$$

Although VC in applications are also required to be *succinct* we will not impose that restriction in this work. The reason behind this choice is that known impossibility results in Maurer's GGM [CFGG23] applies also to VC that are not succinct in the traditional sense (i.e. with logarithmic commitment and opening size).

3 One Way Functions in Maurer GGM

3.1 Definition

In this section we provide definitions that allow us to capture one way functions that only uses a black-box group and whose hardness reduces only to hard problems in the group. The first notion is easily captured by assuming f access the group through the oracles $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}_{\mathsf{eq}}$. To capture the second one we follow the approach of [DHH+21, CFGG23] where security is provided against all GPPT adversaries, i.e. unbounded machines restricted to perform a polynomially bounded number of queries to the GGM random oracles.

Definition 2. *We define Algebraic OWF Family a couple* (Gen, f) *of* PPT *algorithms with*

$$k \leftarrow^{\$} \mathsf{Gen}(1^\lambda), \qquad f_k : \{0,1\}^{n_1} \times \mathbb{G}^{n_2} \to \{0,1\}^{m_1} \times \mathbb{G}^{m_2}.$$

such that for all GPPT *adversaries* $\mathcal{A}$ *there exists a negligible* ε *such that*

$$\Pr\left[\mathcal{A}(y) \to z,\ \ f_k(z) = y \,\middle|\, x \leftarrow^{\$} \{0,1\}^{n_1} \times \mathbb{G}^{n_2},\ \ y \leftarrow f(x)\right] \leq \varepsilon(\lambda).$$

The simplest example of algebraic OWF is the group exponentiation $f : \mathbb{F}_q \to \mathbb{G}$ with $f(x) = x \cdot G$, whose hardness in the GGM was proved in [Mau05].

Without loss of generality we can assume that f_k outputs only group elements, as the output bits can be encoded *in the exponent.* More precisely given f as in the definition above we can define for each key k

$$f'_k : \{0,1\}^{n_1} \times \mathbb{G}^{n_2} \to \mathbb{G}^{m_1+m_2} \ :\ f_k(x) = ((b_i)_{i=1}^{m_1}, \mathbf{H}) \ \Rightarrow\ f'_k(x) = ((b_i \cdot G)_{i=1}^{m_1}, \mathbf{H}).$$

In the GGM $f'(x)$ can be computed from $f(x)$ and vice versa.

3.2 Collision Resistance

Our first impossibility result for NIZK-AoK will apply to NP relations defined for a large family of OWF, but we will need the OWF to be collision resistant. This is problematic as not every OWF is collision resistant, effectively restricting the scope of our result.

To address this issue we show that any *algebraic* OWF family with domain $\{0,1\}^n$ and Gen returning only random group elements can be transformed to achieve collision resistance by simply restricting its domain. The idea is that, with unbounded computation, we could find for a given key k a subset $\mathcal{X} \subseteq \{0,1\}^n$ such that $f_k(\mathcal{X}) = \operatorname{Im} f_k$ and f_k is injective over $\mathcal{X}$. However this would be inefficient in terms of group operations[7]. Therefore we show that if the OWF key is a vector of random group elements, it is possible to restrict the function's

[7] Each evaluation of f_k would required access to the GGM, implying exponentially many queries.

domain in GPPT time so that finding collisions implies finding linear relations among the group elements in the key.

Concretely in the following lemma we provide two GPPT algorithm, Memb and Samp, respectively testing membership with the restricted domain $x \in \mathcal{X}$ and sampling elements from $\mathcal{X}$. A proofs appears in the Appendix, Section A.1.

Lemma 1. *Given* (Gen, f) *algebraic OWF family with* $f : \{0,1\}^n \to \mathbb{G}^m$ *and* Gen *returning a uniformly distributed key* $k \sim U(\mathbb{G}^\kappa)$*, there exists a set* $\mathcal{X} \subseteq \{0,1\}^n$ *and two* GPPT *algorithm* Memb *and* Samp *such that*

- ***Correctness 1:*** $\mathsf{Memb}(x) \to 1 \;\Leftrightarrow\; x \in \mathcal{X}$.
- ***Correctness 2:*** $x \xleftarrow{\$} \mathsf{Samp}(1^\lambda) \;\Rightarrow\; x \in \mathcal{X}$.
- ***Indistinguishability:*** $\exists \varepsilon$ *negligible s.t. for* $k \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$*,* $x_1 \xleftarrow{\$} \{0,1\}^n$ *and* $x_2 \xleftarrow{\$} \mathsf{Samp}(1^\lambda)$*,*

$$\Delta\big((k, f_k(x_1)), (k, f_k(x_2))\big) \;\leq\; \varepsilon(\lambda).$$

- ***Collision Resistance:*** $\exists \varepsilon$ *negligible s.t. for all* GPPT *adversaries* $\mathcal{A}$ *, given* $k \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$*,*

$$\Pr\left[\mathcal{A}(k) \to (x_1, x_2),\;\; x_1, x_2 \in \mathcal{X},\;\; x_1 \neq x_2,\;\; f_k(x_1) = f_k(x_2)\right] \leq \varepsilon(\lambda).$$

We remark that the above Lemma could be extended to $k \sim \mathbb{G}^\kappa$ (not necessarily uniform) such that finding non-trivial linear relations among its group elements is hard for GPPT adversary. This holds as in the GGM such a vector k would be indistinguishable from $k' \sim U(\mathbb{G}^\kappa)$, implying that f can be extended to f' with key space $\mathbb{G}^\kappa$ by running the evaluation algorithm for f also for those keys for which f is not formally defined. Since Memb and Samp exists for f', they satisfy the above properties also for f (or else we could build a distinguisher for k and k').

3.3 Hard-Core Predicates

In [GL89] Goldreich and Levin proved that in the standard model, any OWF f with domain in $\{0,1\}^n$ can be transformed into another OWF $f'(\mathbf{x}, \mathbf{r}) = (f(\mathbf{x}), \mathbf{r})$ that admits the hard-core predicate $\mathbf{x}^\top \mathbf{r}$.

We observe that, given an algebraic OWF family, that is secure against any GPPT adversary, even when the function's domain is restricted as discussed in the previous section, the same result applies.

Theorem 1. *Let* (Gen, f) *an algebraic OWF familty with* $f : \{0,1\}^n \to \mathbb{G}^m$ *and* Gen *returning* $k \sim U(\mathbb{G}^\kappa)$*. Then there exists* ε *negligible such that for all* GPPT *adversaries* $\mathcal{A}$

$$\Pr\left[\mathcal{A}(k, y, \mathbf{r}) \to b, \quad b = \mathbf{x}^\top \mathbf{r} \;\middle|\; \begin{matrix} k \xleftarrow{\$} \mathsf{Gen}(1^\lambda), & \mathbf{r} \xleftarrow{\$} \{0,1\}^n \\ \mathbf{x} \xleftarrow{\$} \mathsf{Samp}(1^\lambda), & y \leftarrow f_k(\mathbf{x}) \end{matrix}\right] \leq \varepsilon(\lambda)$$

The proof is identical to the original result up to observing that in this case the function's input is sampled with a different distribution than the uniform one and that the reduction only needs black-box access to the group. For completeness a proof appears in the Appendix, Section A.2.

4 Impossibility of Algebraic NIZK-AoK

4.1 Hiding Vector Commitments

Our first step toward the impossibility of algebraic NIZK-AoK will be to prove a tighter lower bound than the one presented in [CFGG23] for the class of algebraic VC which *hides* unopened entries. In this section we define the *hiding* property in a game-based way. The approach we take is inspired by IND security for functional encryption schemes: for any two vectors $\mathbf{x}^0$, $\mathbf{x}^1$ provided by the adversary, we ask that guessing which one was committed is hard even when those positions in which $\mathbf{x}^0$ and $\mathbf{x}^1$ match are opened.

$\mathsf{ExpHide}^{\mathcal{A}}(1^\lambda)$

$\mathsf{pp} \leftarrow \mathsf{VC.Setup}(1^\lambda), \quad \beta \xleftarrow{\$} \{0,1\}$

$(\mathbf{x}^0, \mathbf{x}^1) \leftarrow \mathcal{A}(\mathsf{pp})$ such that $\mathbf{x}^0, \mathbf{x}^1 \in (\mathsf{VC.M})^n$

$c, \mathsf{aux} \leftarrow \mathsf{VC.Com}(\mathsf{pp}, \mathbf{x}^\beta), \quad \mathcal{A} \leftarrow c$

When $\mathcal{A}$ queries $i \in \{1, \dots, n\}$:

  If $x_i^0 = x_i^1$: $\Lambda_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}, i, \mathsf{aux}), \quad \mathcal{A} \leftarrow \Lambda_i$

When $\beta' \leftarrow \mathcal{A}$:

  Return $\beta' == \beta$

Fig. 1. Vector Commitment's hiding game with adversary $\mathcal{A}$

Definition 3. *Given a Vector Commitment and an adversary $\mathcal{A}$ we define its advantage at the hiding game, described in Fig. 1, as*

$$\mathsf{Adv}(\mathcal{A}) \;=\; \left| \frac{1}{2} - \Pr\left[\mathsf{ExpHide}^{\mathcal{A}}(1^\lambda) = 1\right] \right|.$$

A VC (resp. Algebraic VC) is hiding if there exists ε negligible such that for all PPT *(resp.* GPPT*) adversaries $\mathcal{A}$, $\mathsf{Adv}(\mathcal{A}) \leq \varepsilon(\lambda)$.*

As a sanity check we observe that combining any (non necessarily algebraic) VC with a commitment scheme yields an hiding VC as informally stated in [CF13]. We further notice that, viewing VCs as a special class of Functional Commitments [LRY16], the game in Fig. 3 could be rephrased for this general primitive by letting $\mathcal{A}$ query functions f and receive an opening for f only if $f(\mathbf{x}^0) = f(\mathbf{x}^1)$.

While the above definition is given in a way that can be easily generalized, when applied to VC it becomes equivalent to a simpler notion, given through the game described in Fig. 2. The two main differences from the previous definition are that $\mathbf{x}^0$ and $\mathbf{x}^1$ are allowed to differ in at most one position, and that opening proofs for all other positions are given directly without oracle queries.

$\mathsf{ExpHideVC}^{\mathcal{A}}(1^\lambda)$

$\mathsf{pp} \leftarrow \mathsf{VC.Setup}(1^\lambda), \quad \beta \xleftarrow{\$} \{0,1\}$

$(\mathbf{x}^0, \mathbf{x}^1) \leftarrow \mathcal{A}(\mathsf{pp})$ with $\mathbf{x}^0, \mathbf{x}^1$ differing only in position i

$c, \mathsf{aux} \leftarrow \mathsf{VC.Com}(\mathsf{pp}, \mathbf{x}^\beta), \quad \Lambda_j \leftarrow \mathsf{VC.Open}(\mathsf{pp}, j, \mathsf{aux})$ for all $j \neq i$.

When $\beta' \leftarrow \mathcal{A}(c, (\Lambda_j)_{j \neq i})$

  Return $\beta' == \beta$

Fig. 2. Simpler Vector Commitment's hiding game with adversary $\mathcal{A}$

Proposition 1. *A (resp. algebraic) VC is hiding if and only if there exists a negligible ε such that for each* PPT *(resp.* GPPT*) adversary $\mathcal{A}$, its advantage in the game described in Fig. 2 is*

$$\mathsf{Adv}(\mathcal{A}) \;=\; \left| \frac{1}{2} - \Pr\left[\mathsf{ExpHideVC}^{\mathcal{A}}(1^\lambda) = 1\right] \right| \;\leq\; \varepsilon(\lambda).$$

A proof of this Proposition appears in the Appendix, Section A.3.

4.2 Reduction to Signatures

Having introduced the notion of hiding VC we now show that the reduction from VC to Signatures provided in [CFGG23] transform hiding VCs into unforgeable signature schemes. We recall their construction in Fig. 3. Their idea is to sign the indices from 1 to n by letting the verification key of the scheme be the VC's common reference string and a commitment to n random messages. A signature for message i is then an opening of the i-th position, whose correctness can be verified through $\mathsf{VC.Vfy}$.

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

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

$m_1, \ldots, m_n \xleftarrow{\$} \mathsf{VC.M}$

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

$\mathsf{vk} \leftarrow (\mathsf{pp}, c), \quad \mathsf{sk} \leftarrow \mathsf{aux}$

Return vk, sk

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

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

Return $\sigma \leftarrow (m_i, \pi)$

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

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

Fig. 3. Generic transformation from VCs to signature schemes

Unforgeability is proven as follows. Assume an adversary forges a signature for message i, that is an opening of c at position i to some message m_i', with c being a commitment to $m_1, \ldots, m_n$. Then $m_i' \neq m_i$ only with negligible probability, or else $\mathcal{A}$ would break position binding. Therefore $\mathcal{A}$ can be used to break the hiding property. This is done by guessing the position i it will forge, querying in the hiding game, Figure 2, two random vectors differing in that position, answering signature queries with the received opening values and finally returning the bit corresponding to the vector containing m'_i in position i.

A more detailed proof appears in the Appendix, Section A.4.

Proposition 2. *Given a position-binding and hiding (Algebraic) Vector Commitment, the (Algebraic) Signature scheme in Fig. 3 is unforgeable.*

4.3 Lower Bound

As proposition 2 implies that Algebraic VC can be transformed into Algebraic Signatures, for which lower bounds on the parameter size are known [DHH+21, CFGG23], we now derive a lower bound for position-binding and hiding vector commitments.

In [CFGG23] it is proven that any algebraic signature satisfying a weaker security notion, which they call ϑ unforgeability, with message space of size n and a verification key with m group elements[8] must satisfy $m \geq n + \vartheta$. As observed in that paper, the standard Unforgeability notion is equivalent to their ϑ-unforgeability when $\vartheta = 0$. Moreover, in the reduction provided in Fig. 3, the verification key only consist of the public parameters (that can be given in the CRS) and one commitment. We thus conclude that

Theorem 2. *Any position-binding and hiding Algebraic Vector Commitment with* GPPT *computable procedures, whose commitment for a vector of length n contains $\ell_c = \ell_c(\lambda, n)$ group elements, satisfies $\ell_c \geq n$.*

Notice that an Algebraic VC that is both position-binding and hiding with linear opening proof size and constant commitment size would violate this Theorem, but not the results presented in [CFGG23]. Indeed in the rest of this sections we show that if Algebraic NIZK-AoK exists for certain relations, VC of this form could be built in the GGM.

We finally remark that the above Theorem also captures VC scheme that are efficient only with respect to group operations. This follows as the proof in [CFGG23] also captures this corner case and will be extremely useful in the rest of this section in order to apply our Lemma 1.

4.4 Intuition on NIZK-AoK Impossibility

The final step to obtain our claimed result on Algebraic NIZK-AoK is to show that it would allow to construct a vector commitment in GPPT violating the negative result of Theorem 2.

To build up intuition we first provide a toy construction assuming we have a NIZK-AoK $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ for the discrete logarithm relation, i.e. that given $K, H \in \mathbb{G}$

[8] Excluding those group elements contained in the CRS for which the signer has no trapdoor information.

proves knowledge of x such that $H = x \cdot K$. The idea is to tweak a regular Petersen commitment for n field elements until we make it hiding. An initial approach is, given $x_1, \ldots, x_n \in \mathbb{F}_q$, to commit to them by sending $C = x_1 K_1 + \ldots + x_n K_n$ with K_i random group elements in the public parameters. To open position i we can send, together with x_i, the elements $x_j K_j$ along with a proof of knowledge of x_j.

This would be position binding, as from any two opening of the same position a challenger can extract (using the NIZK extractor) two different representations of C in base $K_1, \ldots, K_n$, which would break the security of standard Petersen commitments. However this would not yet be hiding, even if the argument used is zero knowledge. The issue is that $x \cdot K$ does not hide x. More concretely, in the hiding game, an adversary could send $\mathbf{x}^0 = (1, 0, \ldots 0)$ and $\mathbf{x}^1 = (2, 0, \ldots, 0)$. Later, testing if C is equal to K_1 or $2K_1$, it would be able to understand which was the committed vector.

A way to address this issue is resorting to an hard-core predicate $\ell : \mathbb{F}_q \to \{0, 1\}$ for the discrete logarithm OWF, such as the least significant bit. This time, instead of committing to $x_1, \ldots, x_n$, we present a commitment to bits: given $b_1, \ldots, b_n$ the committer samples

$$x_i \leftarrow^{\$} \mathbb{F}_q \ : \ \ell(x_i) = b_i \qquad C = x_1 K_1 + \ldots + x_n K_n.$$

An opening to b_i can again be the message x_i together with $x_j K_j$ and a proof of knowledge for x_j, but now the verifier has to further verify that $\ell(x_i) = b_i$.

This scheme would be binding as before, up to observing that $\ell(x_i^0) \neq \ell(x_i^1)$ implies $x_i^0 \neq x_i^1$. Conversely the scheme is hiding because until position i is opened, nothing about x_i is revealed apart from $x_i K_i$, also because our argument is zero-knowledge. Thus guessing the message at position i reduces to the hardness of predicting the hard-core predicate ℓ.

4.5 Vector Commitments from NIZK-AoK

We now discuss how to generalize the construction in Section 4.4 to algebraic OWF families.

The first issue is that not all OWFs admit hard-core predicates. We address this using the Goldreich-Levin transformation, see Section 3.3, $f'_k(\mathbf{x}, \mathbf{r}) = (f_k(\mathbf{x}), \mathbf{r})$ which admits the hard-core bit $\mathbf{x}^\top \mathbf{r}$.

The second issue is that OWF may not be collision resistant[9]. An example is $f_K : \{0, \ldots, 2q-1\} \to \mathbb{G}$ such that $f_K(x) = x \cdot K$ where $f_K(0) = f_K(q)$. This may allow an adversary to break position-binding by finding two $\mathbf{x}, \mathbf{x}'$ with $f_k(\mathbf{x}) = f_k(\mathbf{x}')$ and different hard-core bits. To address this we introduced in Section 3.2 two GPPT procedures Memb and Samp to restrict the domain of a OWF in order to make it collision resistant and to sample from this restricted domain.

[9] In the previous example $x \mapsto x \cdot K$ is collision resistant because it is a bijection

Given these observations, in Fig. 4 we provide a complete description of the resulting VC, with (Gen, f) being an algebraic OWF where $f : \{0,1\}^\mu \to \mathbb{G}^m$ and Gen samples uniformly from $\mathbb{G}^\kappa$, and $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ is a NIZK-AoK for the relation

$$\mathcal{R} \;=\; \{((k, y), x) \;:\; f_k(x) = y\}.$$

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

1 : $\mathsf{crs}_{i,j} \leftarrow \mathsf{G}(1^\lambda)$
2 : $k_i \leftarrow \mathsf{Gen}(1^\lambda)$
3 : $\mathsf{pp} \leftarrow \{\mathsf{crs}_{i,j}, k_i \;:\; i, j \in [n]\}$

$\mathsf{VC}^*.\mathsf{Com}(\mathsf{pp}, b_1, \ldots, b_n)$:

1 : $\mathbf{x}_i \xleftarrow{\$} \mathsf{Samp}(1^\lambda)$
2 : $\mathbf{r}_i \xleftarrow{\$} \{0,1\}^\mu$ such that $\mathbf{x}_i^\top \mathbf{r}_i = b_i$
3 : $C \leftarrow \sum_{i=1}^n f_{k_i}(\mathbf{x}_i)$
4 : $\mathsf{aux} \leftarrow (\mathbf{x}_i)_{i=1}^n$
5 : Return $(C, (\mathbf{r}_j)_{j=1}^n)$, aux

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

1 : $\pi_{i,j} \leftarrow \mathsf{P}(\mathsf{crs}_{i,j}, k_j, f_{k_j}(\mathbf{x}_j), \mathbf{x}_j)$
2 : $\Lambda_i = (\mathbf{x}_i, (f_{k_j}, \pi_{i,j})_{j \neq i})$
3 : Return Λ_i

$\mathsf{VC}^*.\mathsf{Vfy}(\mathsf{pp}, (C, D), b_i, i, \Lambda_i)$:

1 : Parse $\Lambda_i = (\mathbf{x}_i, (\mathbf{Y}_j, \pi_{i,j})_{j \neq i})$
2 : Accept if and only if:
3 : $\quad C = f_{k_i}(\mathbf{x}_i) + \sum_{j \neq i} \mathbf{Y}_j$
4 : $\quad 1 = \mathsf{V}(\mathsf{crs}_j, k_j, \mathbf{Y}_j, \pi_j)$
5 : $\quad 1 = \mathsf{Memb}(\mathbf{x}_i), \quad b_i = \mathbf{x}_i^\top \mathbf{r}_i$

Fig. 4. Hiding Vector Commitment from a NIZK-AoK for $\mathcal{R}$.

Theorem 3. *If* $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ *is a NIZK-AoK, the VC described in Fig. 4 is computable in* GPPT *time, position-binding, hiding and returns commitments with* $O(1)$ *group elements.*

Proof. **Efficiency**: To show that all procedures can be computed in GPPT time it suffices to observe that all steps are efficiently computable, and by Lemma 1 $\mathsf{Samp}, \mathsf{Memb}$ are computable in GPPT time.

Constant Group-Elements Commitment: Commitments only contains m group elements in $C \in \mathbb{G}^m$ with $\mathbb{G}^m$ being the domain of f. Because m does not depend on n, the commitment only contains a constant number of group elements in n (although it may depends on λ).

Position Binding: Given $\mathcal{A}$ breaking position binding we build $\mathcal{B}$ which, given κn random group elements in $\mathbb{G}$, finds a linear relation among them. Note that this is equivalent to breaking the discrete logarithm problem, which in Maurer's GGM is known to be hard [Mau05].

$\mathcal{B}(\mathbf{V})$:

1 : Parse the input as n OWF keys $\mathbf{V} = (\mathbf{k}_1, \dots, \mathbf{k}_n) \in (\mathbb{G}^\kappa)^n$
2 : Sample with the NIZK extractor $(\mathsf{crs}_{i,j}, \mathsf{td}_{i,j}) \leftarrow \mathsf{E}(1^\lambda)$
3 : $\mathsf{pp} \leftarrow \{\mathsf{crs}_{i,j}, \mathbf{k}_i \;:\; i, j \in [n]\}$
4 : $\mathcal{A}(\mathsf{pp}) \rightarrow (c, i, b_0, \Lambda_0, b_1, \Lambda_1)$
5 : Parse $c = (C, (\mathbf{r}_j)_{j=1}^n)$ and $\Lambda_\beta = (\mathbf{x}_i^\beta, (\mathbf{Y}_j^\beta, \pi_{i,j}^\beta)_{j \neq i})$ for $\beta \in \{0, 1\}$
6 : Extract $\mathbf{x}_j^\beta \leftarrow \mathsf{E}(\mathsf{td}_{i,j}, \mathbf{k}_j, \mathbf{Y}_j^\beta, \pi_{i,j}^\beta)$ for $j \in [n] \setminus \{i\}$
7 : Compute $A_j^\beta \in \mathbb{F}_q^{m,\kappa}$ such that $f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) \;=\; A_j^\beta \cdot \mathbf{k}_j$ for $j \in [n]$
8 : Return $(A_j^0 - A_j^1)_{j=1}^n$

Fig. 5. $\mathcal{B}$ reducing position binding to the discrete logarithm problem.

We preliminary notice that in line 7, A_j^β can be computed efficiently. A way to achieve this is locally storing during the execution of $f_{\mathbf{k}_j}(\mathbf{x}_j^\beta)$ a representation for each element queried to the GGM oracle as a linear combination of the group elements in $\mathbf{k}_j$. Doing so, the matrix A_j^β is given by the output elements' representations .

Next we define the following events:

$\mathcal{B}$ wins : $\sum_{j=1}^n (A_j^0 - A_j^1)\mathbf{k}_j = \mathbf{0}$ and $(A_j^0 - A_j^1)_{j=1}^n$ is a non-zero matrix
$\mathcal{A}$ wins : $\mathcal{A}$ breaks position-binding
Ext : $f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) = \mathbf{Y}_j^\beta$ for all $j \neq i$ and $\beta \in \{0, 1\}$
Coll : $\mathbf{x}_i^0 \neq \mathbf{x}_i^1 \;\wedge\; f_{\mathbf{k}_i}(\mathbf{x}_i^0) = f_{\mathbf{k}_i}(\mathbf{x}_i^1) \;\wedge\; 1 = \mathsf{Memb}(\mathbf{x}_i^0) = \mathsf{Memb}(\mathbf{x}_i^1)$

Since $\mathcal{A}$ wins only if the openings are correct, $\pi_{i,j}^\beta$ are all accepted. Calling ε_1 the extractor error, see Section 2.3, we have that

$$\begin{aligned}
\Pr[\mathsf{Ext} \mid \mathcal{A}\ \mathsf{wins}] \;&=\; 1 - \Pr\left[\bigvee\nolimits_{j \neq i} \bigvee\nolimits_{\beta=0}^{1} f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) \neq \mathbf{Y}_j^\beta \;\middle|\; \mathcal{A}\ \mathsf{wins}\right] \\
&\geq\; 1 - \sum\nolimits_{j \neq i} \sum\nolimits_{\beta=0}^{1} \Pr\left[f_{\mathbf{k}_j}(\mathbf{x}_j) = \mathbf{Y}_j^\beta \;\middle|\; \mathcal{A}\ \mathsf{wins}\right] \\
&\geq\; 1 - (2n-2)\varepsilon_1(\lambda) \;=\; 1 - \varepsilon_1^*(\lambda).
\end{aligned}$$

with $\varepsilon_1^* = (2n-2)\varepsilon_1$ being a negligible function. By Lemma 1 we also have that $\Pr[\mathsf{Coll}] \leq \varepsilon_2(\lambda)$. Next, we notice that $\mathcal{B}$ wins if $\mathcal{A}$ wins, Ext and $\neg\mathsf{Coll}$ occurs. Indeed in this case

$$C \;=\; f_{\mathbf{k}_i}(\mathbf{x}_i^\beta) + \sum\nolimits_{j \neq i} \mathbf{Y}_j^\beta \;=\; \sum\nolimits_{j=1}^n f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) \;=\; \sum\nolimits_{j=1}^n A_j^\beta \mathbf{k}_j$$

for both $\beta \in \{0, 1\}$, where the first equality follows by $\mathcal{A}$ wins, the second one from Ext and the third one by construction. This implies $\sum_{j=1}^n (A_j^0 - A_j^1)\mathbf{k}_j = \mathbf{0}$. Moreover this relation is non trivial since, as $\mathcal{A}$ wins, we must have

$$\begin{aligned}
&b^0 \neq b^1 \quad \Rightarrow \quad (\mathbf{x}_i^0)^\top \mathbf{r}_i \neq (\mathbf{x}_i^1)^\top \mathbf{r}_i \quad \Rightarrow \quad \mathbf{x}_i^0 \neq \mathbf{x}_i^1 \quad \Rightarrow \\
&\quad \Rightarrow \quad f_{\mathbf{k}_i}(\mathbf{x}_i^0) \neq f_{\mathbf{k}_i}(\mathbf{x}_i^1) \quad \Rightarrow \quad A_i^0 \mathbf{k}_i \neq A_i^1 \mathbf{k}_i \quad \Rightarrow \quad A_i^0 - A_i^1 \neq \mathbf{0}.
\end{aligned}$$

Where the fourth implication comes from $\neg\mathsf{Coll}$. To conclude we finally bound the advantage of $\mathcal{A}$ with the probability that $\mathcal{B}$ successfully finds a linear relation.

$$\begin{aligned}\Pr[\mathcal{B}\ \mathsf{wins}] &\geq \Pr[\mathcal{A}\ \mathsf{wins},\ \mathsf{Ext},\ \neg\mathsf{Coll}]\\ &\geq \Pr[\mathcal{A}\ \mathsf{wins},\ \mathsf{Ext}] - \Pr[\mathsf{Coll}]\\ &\geq \Pr[\mathsf{Ext} \mid \mathcal{A}\ \mathsf{wins}]\cdot \Pr[\mathcal{A}\ \mathsf{wins}] - \varepsilon_2(\lambda)\\ &\geq (1-\varepsilon_1^*(\lambda))\Pr[\mathcal{A}\ \mathsf{wins}] - \varepsilon_2(\lambda).\end{aligned}$$

Since $\mathcal{B}$ succeeds with negligible probability, the advantage of $\mathcal{A}$ must be negligible as well.

Hiding: We show that given any GPPT adversary $\mathcal{A}$ executed in the game described in Fig. 2, we can build an adversary $\mathcal{B}$ guessing the Goldreich-Levin hard-core predicate for f.

The idea is, given $(\mathbf{k}, \mathbf{Y}, \mathbf{r})$, to setup the VC parameters with the simulator, and use $\mathbf{k}$ as the OWF key for a randomly guessed entry i. Next $\mathcal{A}$ proposes its two vectors of bits $\mathbf{b}_0, \mathbf{b}_1$. If they differ on the guessed position i, $\mathcal{B}$ proceeds computing the commitment, where it uses $\mathbf{Y}, \mathbf{r}$ as the OWF image for the i-th entry, and simulating the required openings. Finally, once $\mathcal{A}$ guesses a bit, it returns the same value. A detailed description appears in Fig. 6.

First we observe that due to the Zero-Knowledge property, distinguishing $\mathsf{crs}_{\ell,j}, \pi_{\ell,j}$ generated by $\mathcal{B}$ from the ones returned by real challenger of $\mathsf{ExpHideVC}$ in Fig. 2 cannot be done with advantage greater than a negligible ε by any GPPT adversary.

Next, assume $i = i'$. Calling β the hard-core predicate $\mathcal{B}$ has to guess, $\mathcal{B}$ correctly commits to $\mathbf{b}_\beta$ since its challenger sets $\mathbf{Y} = f_{\mathbf{k}}(\mathbf{x}) = f_{k_i}(\mathbf{x})$ with[10] $b_i^\beta = \beta = \mathbf{x}^\top \mathbf{r}$, $\mathbf{x} \leftarrow \mathsf{Samp}(1^\lambda)$ and $\mathbf{r} \xleftarrow{\$} \{0,1\}^\mu$. Thus $\mathcal{B}$ wins if $\mathcal{A}$ correctly guesses β, and the initial guess is correct, i.e. $i = i'$. We conclude that

$$\begin{aligned}\mathsf{Adv}(\mathcal{B}) &= |\Pr[\mathcal{A}\to 0,\ i=i' | \beta = 0] - \Pr[\mathcal{A}\to 0,\ i = i' | \beta = 1]|\\ &= \Pr[i=i']\cdot|\Pr[\mathcal{A}\to 0|\beta=0] - \Pr[\mathcal{A}\to 0|\beta=1]|\\ &\geq \frac{\mathsf{Adv}(\mathcal{A}) - \varepsilon(\lambda)}{n}.\end{aligned}$$

Where the third inequality uses $\Pr[i = i'] = 1/n$, which follows as $\mathcal{A}$ has no information on i when it computes $\mathbf{b}_0, \mathbf{b}_1$ (and in particular i').

4.6 Impossibility of Algebraic NIZK-AoK

Combining Theorem 3 and Theorem 2 we can eventually derive the following

Theorem 4. *Given* (Gen, f) *a one way function family with* Gen *returning a uniformly sampled vector in* $\mathbb{G}^\kappa$ *and* $f : \{0,1\}^\mu \to \mathbb{G}^m$*, then there exists no Algebraic NIZK-AoK for the relation*

$$\mathcal{R} = \{((k,y),x) \ :\ f_k(x) = y\}.$$

[10] note that we assumed without loss of generality $b_i^0 = 0$ and $b_i^1 = 1$.

$\mathcal{B}(\mathbf{k}, \mathbf{Y}, \mathbf{r})$:

Sample $i \xleftarrow{\$} \{1, \ldots, n\}$ a guess on the position $\mathcal{A}$ will choose
Sample $k_j \leftarrow \mathsf{Gen}(1^\lambda)$ for $j \neq i$ and set $k_i \leftarrow \mathbf{k}$
Sample with the NIZK simulator $(\mathsf{crs}_{\ell,j}, \mathsf{td}_{\ell,j}) \leftarrow \mathsf{S}(1^\lambda)$ with $\ell, j \in [n]$
$\mathsf{pp} \leftarrow \{\mathsf{crs}_{\ell,j}, k_j \ : \ \ell, j \in [n]\}$
$\mathcal{A}(\mathsf{pp}) \rightarrow \mathbf{b}^0, \mathbf{b}^1$ differing only at position i' (wlog $b^0_{i'} = 0$ and $b^1_{i'} = 1$)
If $i \neq i'$: Return $\perp$

// Simulate the commitment
For $j \neq i$:
  Sample $\mathbf{x}_j \xleftarrow{\$} \mathsf{Samp}(1^\lambda)$ and $\mathbf{r}_j \xleftarrow{\$} \{0,1\}^\mu$ with $\mathbf{x}_j^\top \mathbf{r}_j = b^0_j = b^1_j$
Set $\mathbf{r}_i \leftarrow \mathbf{r}$
Compute $C \leftarrow \mathbf{Y} + \sum_{j \neq i} f_{k_j}(\mathbf{x}_j)$
Create the commitment $c \leftarrow (C, (\mathbf{r}_j)_{j=1}^n)$

// Simulate the openings
For $\ell \neq i$:
  $\pi_{\ell,j} \leftarrow \mathsf{S}(\mathsf{td}_{\ell,j}, k_j, f_{k_j}(\mathbf{x}_j))$ for all $j \neq \ell$
  $\Lambda_\ell = (\mathbf{x}_\ell, (f_{k_j}(\mathbf{x}_j), \pi_{\ell,j})_{j \neq \ell})$

// Execute $\mathcal{A}$ to guess the hard-core bit
Execute $\mathcal{A}(\mathsf{pp}, C, (\Lambda_\ell)_{\ell \neq i}) \rightarrow b$ and return b

Fig. 6. Reduction $\mathcal{B}$ guessing the Goldwasser-Levin hardcore predicate of $f_{\mathbf{k}}$.

5 Impossibility of Algebraic NIZK

5.1 Hard Subset Membership Problem

In this section we recall the definition of Hard Subset Membership Problem, presented in [GW11]. Given an NP relation $\mathcal{R}$, its associated language $\mathcal{L}$ is the set of all statements x for which $(x, w) \in \mathcal{R}$ for some witness w. Informally, the relation $\mathcal{R}$ is a hard subset problem if there are two ways to sample from $\mathcal{L}$ and its complement $\{0,1\}^* \setminus \mathcal{L}$ that are computationally hard to distinguish. As mentioned this captures DDH since the distributions (G, aG, bG, abG) and (G, aG, bG, cG) with a, b, c random field elements and $c \neq a \cdot b$ are hard distinguish. More generally this captures decisional assumptions and their related relations such as Decision Linear and Matrix-DDH. More formally:

Definition 4. *A Subset Membership Problem is a tuple* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *with* $\mathcal{R}$ *an NP relation, and* $\mathsf{SampGood}, \mathsf{SampBad}$ *such that*

- $\mathsf{SampGood}(1^\lambda) \rightarrow (x, w) \quad \Rightarrow \quad (x, w) \in \mathcal{R}.$
- $\mathsf{SampBad}(1^\lambda) \rightarrow x \quad \Rightarrow \quad \nexists w \ : \ (x, w) \in \mathcal{R}.$

A subset membership problem is called hard (against GPPT *adversaries) if* $\exists\varepsilon$ *negligible such that for all* $\mathcal{A}$ GPPT

$$\begin{aligned} & x_0 \overset{\$}{\leftarrow} \mathsf{SampBad}(1^\lambda), \quad (x_1, w_1) \overset{\$}{\leftarrow} \mathsf{SampGood}(1^\lambda) \\ \Rightarrow \quad & \left|\Pr\left[\mathcal{A}(x_0) \to 0\right] - \Pr\left[\mathcal{A}(x_1) \to 0\right]\right| \;\leq\; \varepsilon(1^\lambda). \end{aligned}$$

In the rest of this section we will also use the following Lemma, saying that the probability of correctly guessing λ independent instances of a subset problem is negligible. A proof appears in the Appendix, Section A.5.

Lemma 2. *If* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *is a Hard Subset Membership Problem, then* $\exists\varepsilon$ *negligible such that for all* GPPT *adversaries* $\mathcal{A}$*, setting for* $i \in [\lambda]$

$$\begin{aligned} & x_i^0 \overset{\$}{\leftarrow} \mathsf{SampBad}(1^\lambda), \quad (x_i^1, w_i^1) \overset{\$}{\leftarrow} \mathsf{SampGood}(1^\lambda), \quad b_i \overset{\$}{\leftarrow} \{0,1\} \\ \Rightarrow \; & \Pr\left[\mathcal{A}(x_1^{b_1}, \dots, x_\lambda^{b_\lambda}) \to (b'_1, \dots, b'_\lambda), \qquad b_i = b'_i \quad \forall i \in [\lambda]\right] \;\leq\; \varepsilon(\lambda). \end{aligned}$$

5.2 Preliminary Adversary

Having defined relations with a hard subset problem against GPPT adversaries, in the rest of this section we show that these relations do not admit a NIZK argument in Maurer's GGM. Toward this goal we first construct an adversary $\mathcal{A}$ that, given a NIZK crs and oracle access to the simulator, either returns a proof of a false statement or it finds a linear relation among the group elements in the CRS. In order to ensure sequential executions of $\mathcal{A}$ we give it an affine space V in input, containing linear relations already found among the crs elements. Finally, we will allow $\mathcal{A}$ to fail with an arbitrary small (but non-negligible) probability $1/p$ with $p = \mathsf{poly}(\lambda)$. More formally

Lemma 3. *Let* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *be a hard subset problem and* $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ *a NIZK argument for* $\mathcal{R}$ *with simulator* S*. Then, for any* $p = \mathsf{poly}(\lambda)$*, there exists a* GPPT *adversary* $\mathcal{A}$ *such that: given*

$$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{S}(1^\lambda), \quad x \leftarrow \mathsf{SampBad}(1^\lambda) \quad : \quad \begin{aligned} \mathsf{crs} \;&=\; (\mathbf{Y}, c') \in \mathbb{G}^n \times \{0,1\}^* \\ x \;&=\; (\mathbf{Z}, z') \in \mathbb{G}^m \times \{0,1\}^* \end{aligned}$$

and $V \leq \mathbb{F}_q^n$ *such that* $\mathbf{Y} \in V \cdot G$*, calling* $\mathbf{Z} = \mathbf{z} \cdot G$ *then either:*

1. $\mathcal{A}(V, \mathsf{crs}, x) \to (\mathsf{proof}, \pi)$ *such that* $1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$.
2. $\mathcal{A}(V, \mathsf{crs}, x) \to \mathsf{query}$*. Then setting* $\pi \leftarrow \mathsf{S}(\mathsf{td}, x)$*,* $\mathcal{A}(V, \mathsf{crs}, x, \pi)$ *either aborts with probability smaller than* $1/p(\lambda)$ *or it returns* L *such that*

$$(\mathbf{Y}, \mathbf{Z}) \in L \cdot G \quad \wedge \quad L \cap \left(\mathbb{F}_q^n \times \{\mathbf{z}\}\right) \subsetneq (V \times \{\mathbf{z}\})$$

First of all we remark that the second condition simply states that the affine space L contains a new linear relation among the elements $\mathbf{Y}, \mathbf{Z}$ that is non-trivial with respect to $\mathbf{Y}$. Next, we observe that this adversary could be trivially used to break the *simulation soundness* property of the underlying NIZK. This is a stronger version of soundness in which the adversary has oracle access to a simulator and wins if it returns a proof for a false statement that was not queried. The way to use $\mathcal{A}$ is sampling $n+1$ independent elements with $\mathsf{SampBad}(1^\lambda)$ and sequentially passing them to $\mathcal{A}$, using the simulation oracle to reply query requests. At each step (assuming $\mathcal{A}$ does not abort) either $\mathcal{A}$ finds a new linear relationship on the CRS' group elements, reducing the dimension of V by 1, or it returns a proof for x breaking soundness. Calling n the number of group elements in the CRS, $\mathcal{A}$ can find at most n linear relations, implying by the pigeonhole principle that eventually it has to return a valid proof. However, note that using $\mathcal{A}$ to break the standard notion of soundness is not as trivial since in that case no simulator oracle is provided.

Although the construction of $\mathcal{A}$ is rather technical, we simply adapt the approach of [CFGG23]. First, we describe an adversary $\mathcal{B}$ that on input (crs, x) either return a proof or, with one simulation query, finds a linear relation among the group elements in (crs, x). Next, using $\mathcal{B}$ we build $\mathcal{A}$ which ensures that the linear relation found is non-trivial for those elements in the crs with probability $1 - 1/p$. A full description appears in the Appendix, Section A.6.

5.3 Attack Description

As mentioned, the main difficulty of using $\mathcal{A}$ to break soundness is the absence of a simulator oracle. In this section we explain how to circumvent this issue, describing an adversary $\mathcal{Z}$ that breaks soundness using $\mathcal{A}$, and eventually derive our second impossibility result for algebraic NIZKs.

The core idea is that NIZKs for hard subset problem allow to produce proofs in two indistinguishable ways, that is either

1. sampling $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$, $(x, w) \leftarrow^{\$} \mathsf{SampGood}(1^\lambda)$ and producing the proof using P and the witness w
2. sampling $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{S}(1^\lambda)$, $x \leftarrow^{\$} \mathsf{SampBad}(1^\lambda)$ and producing the proof using the simulator.

Thus, assuming we were able to predict whether $\mathcal{A}$ is going to return proof or query, our adversary $\mathcal{Z}$ could

1. sample $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$ when $\mathcal{A}$ is going to ask a query. In this way it can simulate $\mathsf{S}(x)$ with $\mathsf{P}(x, w)$ and get a linear relation on the CRS' elements.
2. sample $x \leftarrow \mathsf{SampBad}(1^\lambda)$ when $\mathcal{A}$ is going to return a proof π. In this way π proves a false statement and $\mathcal{Z}$ breaks soundness

Unfortunately we don't have a way to predict $\mathcal{A}$'s behavior. However, since the only difference in the two approaches above is how x is sampled, $\mathcal{A}$ cannot distinguish between them. Hence by flipping a random coin $\mathcal{Z}$ can guess $\mathcal{A}$'s reply and act accordingly. Since $\mathcal{A}$ replies almost independently from $\mathcal{Z}$'s choice, its guess is correct with probability close to $1/2$. Amplifying this in a way that makes $\mathcal{Z}$ guess correctly at least $n+1$ times allows us to conclude that $\mathcal{A}$ proves a false statement at least once, because at most n linear relations can be found on the CRS' elements. A complete description of $\mathcal{Z}$ appears in Fig. 7.

We remark that the computation of $\mathbf{z}$ in line 5 can be done in polynomial time since $\mathsf{SampBad}$ and $\mathsf{SampGood}$ are generic algorithm: Therefore, by reading their queries to the GGM oracles, it is possible to locally store the discrete logarithm in base G of any queried group element during their execution, and in particular of output's group elements.

$\mathcal{Z}(\mathsf{crs})$:

1 : Initialize $V \leftarrow \mathbb{F}_q^n$ and $\pi^* \leftarrow \perp$
2 : **For** $i \in \{1, \ldots, \lambda(n+1)\}$: // $\lambda(n+1)$ iterations to guess correctly $n+1$ times
3 :   Sample $\beta_i \xleftarrow{\$} \{0,1\}$
4 :   **If** $\beta_i = 0$: $x \leftarrow \mathsf{SampBad}(1^\lambda)$; **Else** $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$
5 :   Parse $x = (\mathbf{Z}, z') \in \mathbb{G}^m \times \{0,1\}^*$ and get $\mathbf{z}$ such that $\mathbf{Z} = \mathbf{z} \cdot G$
6 :   **If** $\mathcal{A}(V, \mathsf{crs}, x) \to \mathsf{query}$:
7 :   **If** $\beta_i = 0$: Continue the for loop
8 :   **Else**:
9 :   Create a proof $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$
10 :   Get $\mathcal{A}(V, \mathsf{crs}, x, \pi) \to L$ and let V' be s.t. $L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) = V' \times \{\mathbf{z}\}$
11 :   Update $V \leftarrow V'$
12 :   **Elif** $\mathcal{A}(V, \mathsf{crs}, x) \to (\mathsf{proof}, \pi)$:
13 :   **If** $\beta_i = 0$ and $1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$: store $\pi^* \leftarrow \pi$
14 : Return π^*

Fig. 7. GPPT Adversary $\mathcal{Z}$ breaking soundness using $\mathcal{A}$ from Lemma 3.

5.4 Impossibility of Algebraic NIZK

Given a description of the adversary $\mathcal{Z}$ we finally state and prove our second impossibility result for NIZK in Maurer's GGM.

Theorem 5. *Let* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *be a subset membership problem hard against* GPPT *adversaries. Then there exists no algebraic NIZK for* $\mathcal{R}$*.*

Proof. We show that given a complete and zero-knowledge non-interactive argument, $\mathcal{Z}$ breaks soundness. First let us fix some notation. $\mathbf{Y}$ will be the vector of group elements in crs, i.e. $\mathsf{crs} = (\mathbf{Y}, c') \in \mathbb{G}^n \times \{0,1\}^*$. $\mathcal{A}$ will be the adversary from Lemma 3 chosen with failure probability

$$\frac{1}{p(\lambda)} = \frac{1}{4\lambda(n+1)}$$

and for the i-th execution of the for-loop in $\mathcal{Z}$ we define the events:

$$\begin{array}{lcl}
\mathsf{GoodProof}_i & : & \beta_i = 0 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \to (\mathsf{proof}, \pi) \\
\mathsf{BadProof}_i & : & \beta_i = 1 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \to (\mathsf{proof}, \pi) \\
\mathsf{GoodQuery}_i & : & \beta_i = 1 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \to \mathsf{query} \\
\mathsf{BadQuery}_i & : & \beta_i = 0 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \to \mathsf{query} \\
\mathsf{Bad}_i & : & \mathsf{BadProof}_i \ \vee \ \mathsf{BadQuery}_i \\
\mathsf{Fail}_i & : & \mathcal{A}(V, \mathsf{crs}, x) \to \perp \text{ or } \mathbf{Y} \notin V \cdot G
\end{array}$$

We further define Fail the event $\exists i : \mathsf{Fail}_i$. Next, we break the proof into the following sequence of claims.

Claim 1 $\Pr[\mathsf{Fail}] \leq 1/2$.

Claim 2 *The probability of happening λ sequential* Bad *events is negligible, i.e. there exist a negligible ε_0 such that*

$$\forall j_0 \leq n\lambda \qquad \Pr\left[\bigwedge_{i=1}^{\lambda} \mathsf{Bad}_{j_0+i} \,\middle|\, \neg\mathsf{Fail}\right] \leq \varepsilon_0.$$

Claim 3 *The probability that* $\neg\mathsf{Bad}$ *occurs less than $n+1$ times is negligible, i.e.*

$$\Pr\left[\,|\{i \ : \ \mathsf{Bad}_i\}| \leq n \mid \neg\mathsf{Fail}\right] \ \leq \ (n+1) \cdot \varepsilon_0.$$

Claim 4 *If* $\mathsf{GoodQuery}_i$ *occurs, then at step 11 of Fig. 7, the dimension of V decreases with overwhelming probability, i.e. there exists a negligible ε_1 such that*

$$\Pr\left[\mathsf{GoodQuery}_i \ \wedge \ \neg(V' \lneq V) \mid \neg\mathsf{Fail}\right] \ \leq \ \varepsilon_1.$$

Claim 5 *If* $\mathsf{GoodProof}_i$ *occurs, then at step 13 of Fig. 7, the proof π is correct with overwhelming probability, i.e. there exists a negligible ε_2 such that*

$$\Pr\left[\mathsf{GoodProof}_i \ \wedge \ 0 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi) \mid \neg\mathsf{Fail}\right] \ \leq \ \varepsilon_2.$$

Before proving these claims we show they imply that with significant probability $\mathcal{Z}$ produces a proof for a false statement. From Claim 3, $1 - (n+1)\varepsilon_0 \leq$

$$\begin{aligned}
&\leq \ \Pr\left[\exists i_1, \ldots, i_{n+1} \ : \ \neg\mathsf{Bad}_{i_j} \,\middle|\, \neg\mathsf{Fail}\right] \\
&\leq \ \Pr\left[\exists i_1, \ldots, i_{n+1} \ : \ \mathsf{GoodQuery}_{i_j} \,\middle|\, \neg\mathsf{Fail}\right] + \Pr\left[\exists i \ : \ \mathsf{GoodProof}_i \mid \neg\mathsf{Fail}\right].
\end{aligned}$$

Regarding the first term, if $\mathsf{GoodQuery}$ occurs $n+1$ times, in at least one of these events the affine space returned by $\mathcal{A}$ does not yield $V' < V$, because the dimension of V can decrease at most n times. Hence, calling wrong_i the event $\neg(V' < V)$ at iteration i, we have that for some j, wrong_{i_j} occurs. Then

$$\begin{aligned}
&\Pr\left[\exists i_1, \ldots, i_{n+1} \ : \ \mathsf{GoodQuery}_{i_j} \,\middle|\, \neg\mathsf{Fail}\right] \\
= \ &\Pr\left[\exists i_1, \ldots, i_{n+1} \ : \ \mathsf{GoodQuery}_{i_j} \quad \wedge \quad \exists j : \mathsf{wrong}_{i_j} \,\middle|\, \neg\mathsf{Fail}\right] \\
\leq \ &\Pr\left[\exists i \ : \ \mathsf{GoodQuery}_i \wedge \mathsf{wrong}_i \mid \neg\mathsf{Fail}\right] \\
\leq \ &\sum_{i=1}^{\lambda(n+1)} \Pr\left[\mathsf{GoodQuery}_i \ \wedge \ \mathsf{wrong}_i \mid \neg\mathsf{Fail}\right] \ \leq \ \lambda(n+1)\varepsilon_1.
\end{aligned}$$

Regarding the second term, calling valid_i the event that a proof returned at step i is accepted by the verifier.

$$\begin{aligned}
&\Pr[\exists i \;:\; \mathsf{GoodProof}_i \mid \neg\mathsf{Fail}] \\
\leq\; &\Pr[\exists i \;:\; \mathsf{GoodProof}_i \wedge \mathsf{valid}_i \mid \neg\mathsf{Fail}] + \Pr[\exists i \;:\; \mathsf{GoodProof}_i \wedge \neg\mathsf{valid}_i \mid \neg\mathsf{Fail}] \\
\leq\; &\Pr[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi^*) \mid \neg\mathsf{Fail}] + \lambda(n+1)\varepsilon_2
\end{aligned}$$

Combining this two upper bounds together we get that $\mathcal{Z}$ returns a correct proof with probability negligibly close to 1/2.

$$\begin{aligned}
\Pr[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi^*)] \;&\geq\; (1 - (n+1)(\varepsilon_0 + \lambda\varepsilon_1 + \lambda\varepsilon_2)) \cdot \Pr[\neg\mathsf{Fail}] \\
&\geq\; \frac{1 - (n+1)(\varepsilon_0 + \lambda\varepsilon_1 + \lambda\varepsilon_2)}{2}.
\end{aligned}$$

Proof of Claim 1. Calling $\varepsilon_{\mathsf{zk}}$ the advantage of distinguishing a crs generated by G from one produced by S and $\varepsilon_{\mathcal{R}}$ the advantage of guessing an instance of a hard subset membership problem, we will show that

$$\Pr[\mathsf{Fail}_i \mid \neg\mathsf{Fail}_1 \wedge \ldots \wedge \neg\mathsf{Fail}_{i-1}] \;\leq\; \frac{1}{4\lambda(n+1)} + 2(\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}).$$

Summing all this $\lambda(n+1)$ terms will give an upper bound $\Pr[\mathsf{Fail}] \leq 1/4 + \mathsf{negl}(\lambda)$ that for sufficiently large values of λ is less that $1/2$. To show this we study two cases:

- $\beta_i = 0$. Then $\mathcal{A}$ receives (V, crs, x) with $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$ and $x \leftarrow \mathsf{SampBad}(1^\lambda)$. By Zero-Knowledge, we have that any $\mathcal{D}$ distinguishing (crs_0, π_0) generated with G and P from (crs_1, π_1) generated by S has advantage at most $\varepsilon_{\mathsf{zk}}$. This holds for any statement, and in particular also for (x, w) chosen by $\mathcal{D}$ (not depending on the crs).
  Next we sketch a distinguisher $\mathcal{D}$ using $\mathcal{A}$. Initially $\mathcal{D}$ samples (x_i, w_i) from $\mathsf{SampGood}$, set it as the challenge statement and receives (crs, π_i) either generated correctly using w_i or simulated. For the first $i-1$ rounds $\mathcal{D}$ behaves as $\mathcal{Z}$. At the i-th round if $\mathcal{A}$ outputs query it replies with π_i. If $\mathcal{A}$ fails $\mathcal{D}$ returns 1, otherwise it returns 0. When the (crs, π_i) is honestly generated, $\mathcal{A}$ fails with probability $1/p(\lambda)$ by Lemma 3. Hence when (crs, π_i) is simulated $\mathcal{A}$ fails with probability smaller than $1/p(\lambda) + \mathsf{Adv}(\mathcal{D}) \leq 1/p(\lambda) + \varepsilon_{\mathsf{zk}}$.
  In conclusion $\Pr[\mathsf{Fail}_i \mid \neg\mathsf{Fail}_1 \wedge \ldots \wedge \neg\mathsf{Fail}_{i-1}] =$
  $$\Pr\left[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \middle| \bigwedge_{j=1}^{i-1} \neg\mathsf{Fail}_j\right] \;\leq\; \frac{1}{p(\lambda)} + \varepsilon_{\mathsf{zk}} \;\leq\; \frac{1}{4\lambda(n+1)} + 2(\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}).$$
- $\beta_i = 1$. Then $\mathcal{A}$ receives (V, crs, x) with $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$, $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$. By Definition 4 the advantage of distinguishing (crs, x) from (crs, x') with $x' \leftarrow \mathsf{SampBad}(1^\lambda)$ is less than $\varepsilon_{\mathcal{R}}$. The previous argument allow us to conclude
  $$\Pr[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \mid \neg\mathsf{Fail}_1 \wedge \ldots \wedge \neg\mathsf{Fail}_{i-1}] \leq \frac{1}{4\lambda(n+1)} + \varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}.$$

Analogously, since $(\mathbf{Y}, \mathbf{Z}) \in L \cdot G$ if and only if $\mathbf{Y} \in V' \cdot G$, $\Pr[\mathbf{Y} \notin V' \cdot G] \leq \varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}$, or else $\mathcal{A}$ could be used as a distinguisher as shown before. Using a union bound yields again the claimed inequality.

Proof of Claim 2. We describe $\mathcal{M}$ using $\mathcal{A}$ to guess λ instances of a hard subset membership problem.

$\mathcal{M}(x_1, \ldots, x_\lambda)$:

Initialize $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$, $V \leftarrow \mathbb{F}_q^n$
For j_0 times: // Behave as $\mathcal{Z}$
  Sample $\beta \xleftarrow{\$} \{0,1\}$
  If $\beta = 0$: $x \leftarrow \mathsf{SampBad}(1^\lambda)$; **Else** $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$
  If $\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \mathsf{query}$:
    If $\beta = 1$:
      Create a proof $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$
      Get $\mathcal{A}(V, \mathsf{crs}, x, \pi) \rightarrow V'$ and update $V \leftarrow V'$
// After the For-loop, pass for λ times the challenges to $\mathcal{A}$
For $i \in \{1, \ldots, \lambda\}$:
  If $\mathcal{A}(V, \mathsf{crs}, x_i) \rightarrow \mathsf{proof}$: Set $b_i \leftarrow 1$
  Else: Set $b_i \leftarrow 0$
Return $(b_1, \ldots, b_\lambda)$

Fig. 8. Reduction $\mathcal{M}$ guessing λ instances of a Hard Subset Membership Problem.

By inspection $\mathcal{M}$ perfectly emulates the behavior of $\mathcal{Z}$ for the first j_0 executions of the initial For-loop. Regarding the subsequent λ calls to $\mathcal{A}$ we proceed inductively assuming $\mathcal{M}$ correctly guessed all challenges and simulated $\mathcal{Z}$ until the $(i-1)$-th step. Let b'_i be the challenger's bit, such that if $b'_i = 0$ then x_i is generated with $\mathsf{SampBad}$ or else $\mathsf{SampGood}$ was used. When $b'_i = 0$, $\mathcal{M}$ correctly executes $\mathcal{A}(V, \mathsf{crs}, x_i)$ as $\mathcal{Z}$ would with $\beta_{j_0+i} = 0$. Similarly when b'_0, $\mathcal{M}$ correctly run $\mathcal{A}(V, \mathsf{crs}, x_i)$ as $\mathcal{Z}$ would with $\beta_{j_0+i} = 1$. Thus, assuming $\neg\mathsf{Fail}$

$$\begin{aligned} b_i = b'_i \quad &\Leftrightarrow \quad (b'_i = 0 \rightarrow b_i = 0) \wedge (b'_i = 1 \rightarrow b_i = 1) \quad \Leftrightarrow \\ &\Leftrightarrow \quad \mathsf{BadQuery}_{j_0+i} \wedge \mathsf{BadProof}_{j_0+i} \quad \Leftrightarrow \quad \mathsf{Bad}_{j_0+i}. \end{aligned}$$

As a consequence, if $\mathcal{M}$ correctly guesses b'_i, not updating V keeps its behavior identical to $\mathcal{Z}$, which only updates V if $\mathsf{GoodQuery}_{j_0+i}$ occurs. Therefore

$$\begin{aligned} \Pr[b_i = b'_i,\ i \in [\lambda]] &\geq \Pr[\neg\mathsf{Fail}] \cdot \Pr[b_i = b'_i,\ i \in [\lambda] \mid \neg\mathsf{Fail}] \\ &= \Pr[\neg\mathsf{Fail}] \cdot \Pr[\neg\mathsf{Bad}_{j_0+i},\ i \in [\lambda] \mid \neg\mathsf{Fail}] \end{aligned}$$

Since by Lemma 2 the probability of $b'_i = b_i$ is negligible, and by Claim 1 $\Pr[\neg\mathsf{Fail}] \geq 1/2$ we conclude that the claim is true.

Proof of Claim 3. If $\neg\mathsf{Bad}$ occurs less than $n+1$ times, by the pigeonhole principle for at least one of the intervals $I_k = \{\lambda k+1, \ldots, \lambda(k+1)\}$ Bad_i occurs for all $i \in I_k$. A union bound yields

$$\begin{aligned}\Pr\left[\left|\{i \,:\, \neg\mathsf{Bad}_i\}\right| < n+1 \;\middle|\; \neg\mathsf{Fail}\right] \;&\leq\; \sum\nolimits_{k=0}^{n} \Pr\left[\forall i \in I_k,\ \mathsf{Bad}_i \mid \neg\mathsf{Fail}\right] \\ &\leq\; (n+1)\cdot\varepsilon(\lambda).\end{aligned}$$

Proof of Claim 4. We first observe that if (crs, x) is generated with S and $\mathsf{SampBad}$, if $\mathcal{A}(V, \mathsf{crs}, x) \to \mathsf{query}$ the affine space L it returns satisfies by Lemma 3 $L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}$. By definition then

$$V' \times \{\mathbf{z}\} \;=\; L \cap \left(\mathbb{F}_q^n \times \{\mathbf{z}\}\right) \;\lneq\; V \times \{\mathbf{z}\} \qquad \Rightarrow \qquad V' \lneq V.$$

Therefore, borrowing notation from the proof of Claim 1, when $\beta_i = 1$, the probability that $\mathsf{GoodQuery}_i \wedge \neg(V' \lneq V)$ is smaller than $\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}$, or else $\mathcal{A}$ could be used to distinguish (crs, x) from (crs', x') respectively generated with $\mathsf{G}, \mathsf{SampGood}$ and $\mathsf{S}, \mathsf{SampBad}$. Finally since $\neg\mathsf{Fail}$ occurs with significant probability,

$$\begin{aligned}\Pr\left[\mathsf{GoodQuery}_i \;\wedge\; \neg(V' \lneq V) \mid \neg\mathsf{Fail}\right] \;&\leq\; \frac{\Pr\left[\mathsf{GoodQuery}_i \;\wedge\; \neg(V' \lneq V)\right]}{\Pr\left[\neg\mathsf{Fail}\right]} \\ &\leq\; \frac{\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}}{\Pr\left[\neg\mathsf{Fail}\right]} \;\leq\; \frac{\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}}{2}.\end{aligned}$$

Proof of Claim 5. Analogous to the proof of Claim 4.

6 Conclusions

In conclusion we proved that in Maurer's GGM the following primitives are impossible:

- NIZK-AoK for the preimage relation of algebraic OWF families whose domain is the set of string $\{0,1\}^n$ and key consists of random group elements.
- NIZK for any hard subset membership problem.

Although these cover virtually all cases for which NIZKs are currently used in practice, our results leave a small gap open for technical reasons. The mainly theoretical problem of understanding whether NIZK-AoK impossibility extends to all computationally hard relations, for which finding a witness for a given statement x is (worst-case) hard for GPPT adversaries, is thus left for future work. We also leave open the analogous problem of extending our impossibility for NIZK (not necessarily AoK) to all relations $\mathcal{R}$ whose associated language is (worst-case) hard to decide for GPPT adversaries.

Acknowledgments

This work has been partially supported by SECURING Project (PID2019-110873RJ-I00/MCIN/AEI/10.13039/501100011033) and by PRODIGY Project (TED2021-132464B-I00) funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR. The authors further wish to thank the anonymous reviewers for their comments as well as Dario Fiore, Dario Catalano, David Balbas and Daniele Cozzo for the helpful discussions.

References

ACDD19. Masayuki Abe, Jan Camenisch, Rafael Dowsley, and Maria Dubovitskaya. On the impossibility of structure-preserving deterministic primitives. *Journal of Cryptology*, 32(1):239–264, January 2019.

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.

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.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

BG90. Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 194–211. Springer, Heidelberg, August 1990.

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.

CFGG23. Dario Catalano, Dario Fiore, Rosario Gennaro, and Emanuele Giunta. On the impossibility of algebraic vector commitments in pairing-free groups. In *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part II*, pages 274–299. Springer, 2023.

CGH04. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.

CGJ+22. Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. Correlation intractability and snargs from sub-exponential ddh. *Cryptology ePrint Archive*, 2022.

CH20. Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020.

Cra96. Ronald Cramer. Modular design of secure yet practical cryptographic protocols. *Ph. D.-thesis, CWI and Uni. of Amsterdam*, 1996.

DHH+21. Nico Döttling, Dominik Hartmann, Dennis Hofheinz, Eike Kiltz, Sven Schäge, and Bogdan Ursu. On the impossibility of purely algebraic signatures. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part III*, volume 13044 of *LNCS*, pages 317–349. Springer, Heidelberg, November 2021.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

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.

GKM+00. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st FOCS*, pages 325–335. IEEE Computer Society Press, November 2000.

GKP22. Chaya Ganesh, Hamidreza Khoshakhlagh, and Roberto Parisella. Niwi and new notions of extraction for algebraic languages. In *Security and Cryptography for Networks: 13th International Conference, SCN 2022, Amalfi (SA), Italy, September 12–14, 2022, Proceedings*, pages 687–710. Springer, 2022.

GL89. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.

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.

GO93. Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 228–245. Springer, Heidelberg, August 1993.

GOS06a. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006.

GOS06b. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Heidelberg, May / June 2006.

Gro06. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, December 2006.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

GT00. Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st FOCS*, pages 305–313. IEEE Computer Society Press, November 2000.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

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.

JJ21. Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 3–32. Springer, Heidelberg, October 2021.

KLV22. Yael Tauman Kalai, Alex Lombardi, and Vinod Vaikuntanathan. Snargs and ppad hardness from the decisional diffie-hellman assumption. *Cryptology ePrint Archive*, 2022.

KNYY21. Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Compact designated verifier NIZKs from the CDH assumption without pairings. *Journal of Cryptology*, 34(4):42, October 2021.

KST99. Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *40th FOCS*, pages 535–542. IEEE Computer Society Press, October 1999.

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.

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.

Mau09. Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Heidelberg, June 2009.

Ped92. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, Heidelberg, August 1992.

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

PST13. Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of correct computation. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 222–242. Springer, Heidelberg, March 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.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.

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.

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 Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, Heidelberg, May / June 1998.

Zha22. Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022.

A Postponed Proofs

A.1 Collision Resistant Algebraic OWF

Proof of Lemma 1. We will describe Memb and Samp by first providing a GPPT algorithm $\mathcal{T}$ that generates an exponentially long explicit description of the set $\mathcal{X}$ along with a probability distribution. Then Memb and Samp can check membership in $\mathcal{X}$ and sample from it without using the group at all.

In order to describe $\mathcal{T}$ we make the following preliminary observation. Given any deterministic and polynomial time oracle Turing Machine $\mathcal{F}$ that computes $f_{\mathbf{k}}(x)$ for $\mathbf{k} \in \mathbb{G}^{\kappa}$ and $x \in \{0,1\}^n$, there exists an extractor $\mathcal{E}$ that on input $\mathbf{k}, x$ returns a matrix A such that $f_{\mathbf{k}}(x) = A\mathbf{k}$. The idea is simply that $\mathcal{E}$ executes $\mathcal{F}$ forwarding all the queries to the GGM's oracles while keeping a representation of each queried element in base $\mathbf{k}$. Although there may be many matrices A such that $f_{\mathbf{k}}(x) = A\mathbf{k}$, since $\mathcal{F}$ is deterministic, so is $\mathcal{E}$, implying that there is only one such matrix returned on input $(\mathbf{k}, x)$. We can thus define

$$A_{\mathbf{k},x} \in \mathbb{F}_q^{m,\kappa} \quad : \quad \mathcal{E}(\mathbf{k}, x) \to A_{\mathbf{k},x}, \quad f_{\mathbf{k}}(x) = A_{\mathbf{k},x} \cdot \mathbf{k}.$$

The above definition works when $\mathbb{G}$ is a prime order group and $\mathbf{k}$ is uniformly sampled from $\mathbb{G}^{\kappa}$. However in the following we will use a different generic group modeling $\mathbb{G}^{\kappa}$ instead of $\mathbb{G}$ where $\mathbf{k}^* = (e_1, \ldots, e_{\kappa}) \in (\mathbb{G}^{\kappa})^{\kappa}$, with $e_1, \ldots, e_{\kappa}$ being the canonical base of $\mathbb{G}^{\kappa}$. This will be done to create an environment indistinguishable from the standard one (where the GGM models $\mathbb{G}$) in which the group elements in the key satisfy no linear relation. By executing $\mathcal{E}$ with this GGM oracle modeling $\mathbb{G}^{\kappa}$ we define for all $x \in \{0,1\}^n$

$$A_x^* \in \mathbb{F}_q^{m,\kappa} \quad : \quad \mathcal{E}(\mathbf{k}^*, x) \to A_x^*.$$

We are now ready to provide a description of $\mathcal{T}$. The idea is that computing A_x^* does not require any real GGM oracle query because $\mathcal{T}$ can simulate a GGM in which any non trivial linear relation query among the elements in $\mathbf{k}$ is answered with 0. Hence $\mathcal{T}$ can compute A_x^* for all x and insert in $\mathcal{X}$ only those element whose associated matrix does not collide with other elements already in $\mathcal{X}$. In this way we have that the map

$$\mathcal{X} \to \mathbb{F}_q^{m,\kappa} \quad : \quad x \mapsto A_x^*$$

is injective. Furthermore $\mathcal{T}$ can compute in exponential space a map $F : \{0,1\}^n \to \mathcal{X}$ such that x and $F(x)$ have the same associated matrix. This defines a distribution over $\mathcal{X}$ that is the image through F of the uniform one over $\{0,1\}^n$. A full description of $\mathcal{T}$ is provided in Figure 9, along with a Memb and Samp.

Note that $\mathcal{X}$ is well defined because $\mathcal{T}$ is deterministic, although it depends on the arbitrary choice of an ordering in $\{0,1\}^n$ and a procedure $\mathcal{F}$ to compute f, used to construct $\mathcal{E}$. Having defined $\mathcal{X}$ we can now state and prove the following claims, which trivially imply the thesis.

Claim 1 *Both correctness properties holds. I.e. for all $x \in \{0,1\}^n$, $1 \leftarrow \mathsf{Memb}(x)$ iff $x \in \mathcal{X}$ and $x \leftarrow \mathsf{Samp}(1^{\lambda})$ implies $x \in \mathcal{X}$.*

Procedure $\mathcal{T}(1^\lambda)$:

Initialize $\mathcal{X} \leftarrow \emptyset$ and $F : 0, 1^n \to \mathcal{X}$ partial function
For $x \in \{0,1\}^n$:
  Compute $A_x^* \leftarrow \mathcal{E}(\mathbf{k}^*, x)$
  If there exists $z \in \mathcal{X}$ with $A_z^* = A_x^*$:
    Set $F(x) \leftarrow z$
  Else:
    Add $\mathcal{X} \leftarrow \mathcal{X} \cup \{x\}$ and set $F(x) = x$
Return $(\mathcal{X}, F)$

$\mathsf{Memb}(1^\lambda, x)$

Run $(\mathcal{X}, F) \leftarrow \mathcal{T}$
If $x \in \mathcal{X}$: Return 1
Else: Return 0

$\mathsf{Samp}(1^\lambda)$

Run $(\mathcal{X}, F) \leftarrow \mathcal{T}$
Sample $x \leftarrow^{\$} \{0,1\}^n$
Else: Return $F(x)$

Fig. 9. Intermediate procedure $\mathcal{T}$ describing $\mathcal{X}$.

Claim 2 *There exists a negligible ε such that for any distribution $\mathcal{D}$ over $\{0,1\}^n$, given $\mathbf{k} \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$ and $x \leftarrow^{\$} \mathcal{D}$,*

$$\Pr\left[A_x^* \neq A_{\mathbf{k},x}\right] \;\leq\; \varepsilon(\lambda).$$

Claim 3 *There exists a negligible ε such that for any distribution $\mathcal{D}$ over $\{0,1\}^n$, given $\mathbf{k} \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$ and $x \leftarrow^{\$} \mathcal{D}$,*

$$\Delta((\mathbf{k}, A_x^*), (\mathbf{k}, A_{\mathbf{k},x})) \;\leq\; \varepsilon(\lambda).$$

Claim 4 *Given $\mathbf{k} \leftarrow \mathsf{Gen}(1^\lambda)$, $x_1 \leftarrow^{\$} \{0,1\}^n$ and $x_2 \leftarrow^{\$} \mathsf{Samp}(1^\lambda)$ then*

$$\Delta((\mathbf{k}, A_{x_1}^*), (\mathbf{k}, A_{x_2}^*)) \;=\; 0.$$

Claim 5 *Given $\mathbf{k} \leftarrow \mathsf{Gen}(1^\lambda)$, $x_1 \leftarrow^{\$} \{0,1\}^n$ and $x_2 \leftarrow^{\$} \mathsf{Samp}(1^\lambda)$ then*

$$\Delta((\mathbf{k}, f_{\mathbf{k}}(x_1)), (\mathbf{k}, f_{\mathbf{k}}(x_2))) \;\leq\; \varepsilon(\lambda).$$

Claim 6 *$\exists \varepsilon$ negligible s.t. for all GPPT adversaries $\mathcal{A}$, given $\mathbf{k} \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$,*

$$\Pr\left[\mathcal{A}(\mathbf{k}) \to (x_1, x_2),\;\; x_1, x_2 \in \mathcal{X},\;\; x_1 \neq x_2,\;\; f_{\mathbf{k}}(x_1) = f_{\mathbf{k}}(x_2)\right] \leq \varepsilon(\lambda).$$

Proof of Claim 1. By inspection $\mathsf{Memb}(x)$ returns 1 if and only if $x \in \mathcal{X}$ and is a GPPT algorithm since both $\mathcal{T}$ is, as no GGM query is ever performed. Analogously by construction $F : \{0,1\}^n \to \mathcal{X}$, so $\mathsf{Samp}(1^\lambda)$ always returns an element in $\mathcal{X}$.

Proof of Claim 2. Given a distribution $\mathcal{D}$, even if not efficiently sampleable, we will build an adversary $\mathcal{A}$ who tries to find a linear relation among κ group elements. The idea is that A^*_x differs from $A_{\mathbf{k},x}$ only if $\mathcal{E}$ (indirectly) queried if $\mathbf{k}$ satisfies a non-trivial linear relation and got 1 as a reply in the real GGM. This happens since $\mathcal{E}$ is deterministic, thus, if it does not find any non-trivial linear relation, it gets the same replies from both the real GGM and the generic group modeling $\mathbb{G}^\kappa$. A description of $\mathcal{A}$ appears in Figure 10.

Reduction $\mathcal{A}(\mathbf{k})$:

1 : Sample in exponential time $x \xleftarrow{\$} \mathcal{D}$
2 : Run $\mathcal{E}(\mathbf{k}, x)$
3 : **When** $\mathcal{E}$ queries $\mathcal{O}_{\mathsf{add}}(T_1, T_2)$:
4 :   Retrieve $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{F}_q^\kappa$ such that $T_i = \mathbf{a}_i^\top \mathbf{k}$
5 :   Query $\mathcal{O}_{\mathsf{add}}(T_1, T_2)$ and store $\mathbf{a}_1 + \mathbf{a}_2$ as a representation of the result
6 : **When** $\mathcal{E}$ queries $\mathcal{O}_{\mathsf{eq}}(T_1, T_2)$:
7 :   Retrieve $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{F}_q^\kappa$ such that $T_i = \mathbf{a}_i^\top \mathbf{k}$
8 :   Query $b \leftarrow \mathcal{O}_{\mathsf{eq}}(T_1, T_2)$ and return $\mathcal{E} \leftarrow b$.
9 :   *If* $b = 1$ and $\mathbf{a}_1 - \mathbf{a}_2 \neq \mathbf{0}$:
10 :   Return $\mathbf{a}_1 \neq \mathbf{a}_2$
11 : **When** $\mathcal{E}$ halts: Return $\perp$

Fig. 10. Reduction $\mathcal{A}$ finding a linear relation among κ elements $\mathbf{k}$.

First of all we observe that inductively $\mathcal{A}$ can store a representation of each element in base $\mathbf{k}$ as initially $k_i = \mathbf{e}_i^\top \mathbf{k}$, with $\mathbf{e}_i$ being 1 in the i-th position and 0 elsewhere. Next, if $\mathcal{A}$ ever executes line 10, it returns a linear relation since $b = 1$ implies $\mathbf{a}_1^\top \mathbf{k} = \mathbf{a}_2^\top \mathbf{k}$ and therefore $(\mathbf{a}_1 - \mathbf{a}_2)^\top \mathbf{k}$.

Finally, if line 10 is never executed, then $\mathcal{E}$ receives 1 from $\mathcal{O}_{\mathsf{eq}}$ only when $\mathbf{a}_1 \neq \mathbf{a}_2$, implying that $\mathcal{A}$ correctly simulates simultaneously the standard GGM and the generic group modeling $\mathbb{G}^\kappa$. Thus in this case $A^*_x = A_{\mathbf{k},x}$.

In conclusion, $(\mathcal{A} \to \perp) \Rightarrow A^*_x = A_{\mathbf{k},x}$, therefore

$$\Pr\left[\mathcal{A}(\mathbf{k}) \to \mathbf{v},\ \mathbf{v}^\top \mathbf{k} = 0\right] \ \geq\ \Pr\left[A^*_x \neq A_{\mathbf{k},x}\right].$$

Since finding linear relations on a vector of random group elements is equivalent to the discrete logarithm problem, we have that $\Pr\left[A^*_x \neq A_{\mathbf{k},x}\right]$ is negligible.

Proof of Claim 3. To simplify notation, the summations below are taken for all $\mathbf{k}_0 \in \mathbb{G}^\kappa$, $A_0 \in \mathbb{F}_q^{m,\kappa}$ and $x_0 \in \{0,1\}^n$.

$$
\begin{aligned}
&\Delta((\mathbf{k}, A^*_x), (\mathbf{k}, A_{\mathbf{k},x})) = \\
&= \frac{1}{2} \cdot \sum_{\mathbf{k}_0,\, A_0} \left|\Pr\left[\mathbf{k} = \mathbf{k}_0,\ A^*_{\mathbf{k},x} = A_0\right] - \Pr\left[\mathbf{k} = \mathbf{k}_0,\ A_{\mathbf{k},x} = A_0\right]\right| \\
&\leq \sum_{\mathbf{k}_0,\, A_0,\, x_0} \frac{1}{2} \left|\Pr\left[A^*_{\mathbf{k},x} = A_0 \middle| \mathbf{k} = \mathbf{k}_0,\ x = x_0\right] - \Pr\left[A_{\mathbf{k},x} = A_0 | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right]\right| \\
&\qquad\qquad \ldots \cdot \Pr\left[\mathbf{k} = \mathbf{k}_0,\ x = x_0\right] \\
&= \sum_{\mathbf{k}_0,\, x_0} \Pr\left[A^*_x \neq A_{\mathbf{k},x} | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right] \cdot \Pr\left[\mathbf{k} = \mathbf{k}_0,\ x = x_0\right] \\
&= \Pr\left[A^*_x \neq A_{\mathbf{k},x}\right] \ \leq\ \varepsilon
\end{aligned}
$$

where in last step we applied Claim 2 and in the third step we used the fact that

$$
\sum_{A_0} \frac{1}{2} \left|\Pr\left[A^*_{\mathbf{k},x} = A_0 \middle| \mathbf{k} = \mathbf{k}_0,\ x = x_0\right] - \Pr\left[A_{\mathbf{k},x} = A_0 | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right]\right|
$$

is equal to $\Pr\left[A^*_x \neq A_{\mathbf{k},x} | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right]$ because

- If $A^*_{x_0} = A_{\mathbf{k}_0,x_0}$ then the summation only contains terms that are 0, as A_0 is either different or equal to both matrices. Thus the above sum is 0, and so is the probability of the two matrices being different when $\mathbf{k} = \mathbf{k}_0$ and $x = x_0$.
- If $A^*_{x_0} \neq A_{\mathbf{k}_0,x_0}$ then the only non zero terms of the summation are those in which A_0 is equal to either $A^*_{x_0}$ or $A_{\mathbf{k}_0,x_0}$. This yield only two terms both equal to 1/2, implying that the sum equals 1, and so does the probability of the two matrix being different when $\mathbf{k} = \mathbf{k}_0$ and $x = x_0$.

Proof of Claim 4. We begin observing that $\mathbf{k}$ is independent from both x_1 and x_2, therefore

$$
\Delta((\mathbf{k}, A^*_{x_1}), (\mathbf{k}, A^*_{x_2})) \ =\ \Delta(\mathbf{k}, \mathbf{k}) + \Delta(A^*_{x_1}, A^*_{x_2}) \ =\ \Delta(A^*_{x_1}, A^*_{x_2}).
$$

From the way we defined Samp, see Figure 9, there exists a random variable x_3 uniformly distributed over $\{0,1\}^n$ such that $x_2 = F(x_3)$. Thus, since for each element z, F satisfy the identity $A^*_z = A^*_{F(z)}$, we have that

$$
\Delta(A^*_{x_1}, A^*_{x_2}) \ =\ \Delta(A^*_{x_1}, A^*_{F(x_3)}) \ =\ \Delta(A^*_{x_1}, A^*_{x_3}) \ =\ 0
$$

where the last equation follows as x_1 and x_3 are both uniformly distributed.

Proof of Claim 5. First of all we observe that

$$
\Delta((\mathbf{k}, f_{\mathbf{k}}(x_1)), (\mathbf{k}, f_{\mathbf{k}}(x_2))) \ \leq\ \Delta((\mathbf{k}, A_{\mathbf{k},x_1}), (\mathbf{k}, A_{\mathbf{k},x_2}))
$$

since the two distribution on the left hand can be obtained from those in the right hand through the map

$$
(\mathbf{k}, A) \ \mapsto\ (\mathbf{k}, A \cdot \mathbf{k})
$$

where we use the fact that $A_{\mathbf{k},x} \cdot \mathbf{k} = f_{\mathbf{k}}(x)$ by the way these matrices are defined. Next let z_1, z_2 be two random variables, with $z_1 \xleftarrow{\$} \{0,1\}^n$ and $z_2 \xleftarrow{\$} \mathsf{Samp}(1^\lambda)$. Applying the triangular inequality twice we get

$$
\begin{aligned}
&\Delta((\mathbf{k}, A^*_{x_1}), (\mathbf{k}, A_{x_2})) \\
\leq\ &\Delta((\mathbf{k}, A_{\mathbf{k},x_1}), (\mathbf{k}, A^*_{z_1})) \ + \ \Delta((\mathbf{k}, A^*_{z_1}), (\mathbf{k}, A^*_{z_2})) \ + \ \Delta((\mathbf{k}, A^*_{z_2}), (\mathbf{k}, A_{\mathbf{k},x_2})) \\
\leq\ &2\varepsilon
\end{aligned}
$$

where we the first and last term are smaller than ε from Claim 3 and the central term is zero from Claim 4.

Proof of Claim 6. Given a GPPT adversary $\mathcal{A}$ that given $\mathbf{k}$ returns two different points $x_1, x_2 \in \mathcal{X}$ for which $f_{\mathbf{k}}(x_1) = f_{\mathbf{k}}(x_2)$, we build a GPPT adversary $\mathcal{B}$ that finds linear relation on a vector of κ random group elements.

The idea is that if x_1 and x_2 lies in $\mathcal{X}$ are distinct, then their associated matrices $A^*_{x_1}$ and $A^*_{x_2}$ need also to be different, or else one of these two points would not be included in $\mathcal{X}$ by $\mathcal{T}$. If $\mathcal{A}$ returned inputs for which $A^*_{x_1} \neq A_{\mathbf{k},x_1}$ or $A^*_{x_2} \neq A_{\mathbf{k},x_2}$, then as done in the proof of Claim 2, we could extract a linear relation over the elements of $\mathbf{k}$. Conversely, if $A^*_{x_1} = A_{\mathbf{k},x_1}$ and $A^*_{x_2} = A_{\mathbf{k},x_2}$, then $A^*_{x_1} - A^*_{x_2}$ is a non zero matrix that vanishes on $\mathbf{k}$. A full description of $\mathcal{B}$ appears in Figure 11

Reduction $\mathcal{B}(\mathbf{k})$:

```
1 : Run A(k) → (x1, x2)
2 : If ¬(x1 ≠ x2 ∧ x1, x2 ∈ X ∧ fk(x1) ≠ fk(x2)):   Return ⊥
3 : Compute A*_{x1}, A*_{x2}, A_{k,x1}, A_{k,x2}
4 : If A*_{xb} ≠ A_{k,xb} for b ∈ {0,1}:
5 :   Compute as done in Fig. 10 a linear relation v : v^⊤ k = 0
6 :   Return v
7 : Else:
8 :   Return A*_{x1} − A*_{x2}
```

Fig. 11. Reduction $\mathcal{A}$ finding a linear relation among κ elements $\mathbf{k}$.

Let coll be the event that the condition at step 2 is not satisfied, i.e. the event in which $\mathcal{A}$ returns a valid collision, and equal be the event $A^*_{x_b} = A_{\mathbf{k},x_b}$ for $b \in \{0,1\}$. If coll and $\neg\mathsf{equal}$ occurs, $\mathcal{B}$ finds a linear relation with probability 1, as observed in the proof of Claim 2. Conversely, if coll and equal we have that

$$
f_{\mathbf{k}}(x_b) \ = \ A_{\mathbf{k},x_b} \cdot \mathbf{k} \ = \ A^*_{x_b} \cdot \mathbf{k}. \qquad f_{\mathbf{k}}(x_1) = f_{\mathbf{k}}(x_2) \quad \Rightarrow \quad A^*_{x_1}\mathbf{k} = A^*_{x_2}\mathbf{k}.
$$

Which implies that $(A^*_{x_1} - A^*_{x_2})$ vanishes on $\mathbf{k}$. Furthermore this is a non trivial relation since

$$
x_1, x_2 \in \mathcal{X},\ x_1 \neq x_2 \quad \Rightarrow \quad A^*_{x_1} \neq A^*_{x_2} \quad \Rightarrow \quad A^*_{x_1} - A^*_{x_2} \neq \mathbf{0}.
$$

We can thus conclude that if coll occurs, then $\mathcal{A}$ finds a a linear relation which implies that $\Pr[\mathsf{coll}]$ is negligible.

A.2 Hard-Core Predicates

Proof sketch of Theorem 1. We will revise the textbook proof of the result in [GL89], pointing out where necessary why it still applies in the setting of GPPT adversaries.

Assume there exists a GPPT adversary $\mathcal{A}$ such that, given $k \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$, $\mathbf{x} \leftarrow \mathsf{Samp}(1^\lambda)$, $\mathbf{r} \leftarrow^{\$} \{0,1\}^n$, with probability $\varepsilon = \mathsf{poly}(\lambda)$

$$\Pr\left[\mathcal{A}(k, f_k(\mathbf{x}), \mathbf{r}) \to b, \quad b = \mathbf{x}^\top \mathbf{r}\right] \;\geq\; \frac{1}{2} + \varepsilon$$

The first step is proving the following claim

Claim 1 *There exists a set* $\mathcal{X}_0 \subseteq \mathcal{X}$ *such that* $\Pr[\mathbf{x} \in \mathcal{X}_0] \geq 1/2$ *and for each* $\mathbf{x} \in \mathcal{X}_0$

$$\Pr\left[\mathcal{A}(k, f_k(\mathbf{x}), \mathbf{r}) \to b, \quad b = \mathbf{x}^\top \mathbf{r}\right] \;\geq\; \frac{1}{2} + \frac{\varepsilon}{2}.$$

Given this and conditioning on $\mathbf{x} \in \mathcal{X}_0$, we describe an adversary $\mathcal{B}$ that guesses through brute-force the value of ℓ independent hard-core predicate bits $b_1, \ldots, b_\ell$ for uniformly sampled $\mathbf{r}_1, \ldots, \mathbf{r}_\ell$. These are expanded to $2^\ell - 1$ bits $b'_i, \mathbf{r}'_i$ by linearity, i.e. such that

$$\forall S \subseteq [\ell],\ S \neq \varnothing \qquad \mathbf{r}'_S = \bigoplus\nolimits_{j \in S} \mathbf{r}_j, \quad b'_S = \bigoplus\nolimits_{j \in S} b_j \tag{1}$$

up to mapping non-empty subsets of $[\ell]$ to integers in $[2^\ell - 1]$. For each of its guess, $\mathcal{B}$ will try to extract the j-th bit of $\mathbf{x}$ by querying $\mathcal{A}$ on

$$\mathcal{A}(k, y, \mathbf{e}_j \oplus \mathbf{r}'_i) \to \beta'_i.$$

If the initially guessed bits are correct, which will happen after at most 2^m guesses, by linearity $\beta'_i = \mathbf{x}^\top \mathbf{r}'_i$. Thus $\mathcal{A}$ receives $2^\ell - 1$ pair-wise independent random guesses and, if it replies correctly then

$$\beta'_i \oplus b'_i \;=\; \mathbf{x}^\top (\mathbf{e}_j \oplus \mathbf{r}'_i) \;\oplus\; \mathbf{x}^\top \mathbf{r}'_i \;=\; \mathbf{x}^\top \mathbf{e}_j \;=\; x_j$$

Thus $\mathcal{B}$ will chose the majority bit among all the replies it gets from $\mathcal{A}$ as a candidate value for x_i. Note each $\beta'_i \oplus b'_i$ is correct with probability $1/2 + \varepsilon/2$. Applying Chebyshev inequality, if $2^\ell - 1 \geq \frac{2n}{\varepsilon^2}$, then the majority bit chosen by $\mathcal{A}$ is the wrong one with probability smaller than $\frac{1}{2n}$. A union bound then implies that $\mathcal{B}$ obtain the right $\mathbf{x}$ with probability grater than $1/2$. A full description of $\mathcal{B}$ appears in Figure 12

We remark that $\mathcal{B}$ runs in GPPT time since $\ell = O(\log(\lambda))$, and thus only $O(\mathsf{poly}(\lambda))$ executions of $\mathcal{A}$ and $f_k(\cdot)$ are performed. The only potentially inefficient step is the check $\mathbf{x}^* \in \mathcal{X}$, which however can be computed in GPPT time by Lemma 1.

$\mathcal{B}(k, y)$:

Chose ℓ such that $2^\ell - 1 \geq \frac{2n}{\varepsilon^2}$
Sample $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \xleftarrow{\$} \{0,1\}^n$
For $(b_1, \ldots, b_\ell) \in \{0,1\}^\ell$: // Brute-force guess ℓ hardcore bits
  For $j \in \{1, \ldots, n\}$
    Compute $b'_i, \mathbf{r}'_i$ for $i \in [2^\ell - 1]$ as in Equation 1
    Get $\beta'_i \leftarrow \mathcal{A}(k, y, \mathbf{e}_j \oplus \mathbf{r}_i)$
    Set x^*_j as the majority bit in $\{\beta'_i \oplus b'_i \ : \ i \in [2^\ell - 1]\}$
  If $f_k(\mathbf{x}^*) = y$ and $\mathbf{x}^* \in \mathcal{X}$: Return $\mathbf{x}^*$
Return $\perp$ // Abort if nothing was found

Fig. 12. Reduction $\mathcal{B}$ executed in the hiding game of Fig. 2.

A.3 Hiding VC

Proof of Proposition 1. We begin observing that one implication is trivially true: Given a VC that is hiding with respect to the game $\mathsf{ExpHide}$ in Figure 1, then any adversary $\mathcal{A}$ executed in $\mathsf{ExpHideVC}$, defined in Figure 2, also succeeds with negligible advantage.

To this goal we describe $\mathcal{B}$, executed in $\mathsf{ExpHide}$ that on input pp, executes $\mathcal{A}(\mathsf{pp})$ and forward its chosen vectors $\mathbf{x}_0, \mathbf{x}_1$ to the challenger. It then queries opening proofs Λ_j for all valid positions, i.e. all except for the i-th, the only one in which $\mathbf{x}_0$ and $\mathbf{x}_1$ are allowed to differ. Finally, it forwards $(\Lambda_j)_{j \neq i}$ to $\mathcal{A}$ and when $\mathcal{A}$ returns a bit, it output the same bit. Clearly $\mathcal{B}$ perfectly simulate the challenger defined in $\mathsf{ExpHideVC}$ and guesses correctly if and only if $\mathcal{A}$ does. Therefore $\mathsf{Adv}(\mathcal{A}) = \mathcal{B}$.

Regarding the converse we will build a sequence of hybrid games $\mathsf{H}_0, \ldots, \mathsf{H}_n$ where in H_i the challenger commits to the first i entries of $\mathbf{x}^0$ and the remaining $n - i$ entries of $\mathbf{x}^1$. In this way H_0 is equivalent to $\mathsf{ExpHideVC}$ with $\beta = 0$ and H_n is equivalent to $\mathsf{ExpHideVC}$ with $\beta = 1$. In all hybrid games, opening queries for position j are only answered if $x^0_j = x^1_j$.

To conclude, we will show that any GPPT adversary $\mathcal{D}$, its advantage at distinguishing H_{i-1} from H_i is negligible. We do this by describing a GPPT algorithm $\mathcal{B}$ that used $\mathcal{D}$ to win at the game $\mathsf{ExpHideVC}$. A description of $\mathcal{B}$ appears in Figure 13.

First of all we observe that conditioning on $x^0_i \neq x^1_i$, $\mathcal{A}$ perfectly simulates H_{i-1} if executed with challenge bit $\beta = 0$ and H_i otherwise. Next, we point out that the advantage of $\mathcal{D}$ when $x^0_i = x^1_i$ is zero as the two games becomes

$\mathcal{B}(\mathsf{pp})$

Run $\mathbf{x}^0, \mathbf{x}^1 \leftarrow \mathcal{D}(\mathsf{pp})$
If $x_i^0 = x_i^1$: Return 1 // Abort the execution
Compute and send to the challenger $\mathbf{z}^0, \mathbf{z}^1$ such that:
  $\mathbf{z}^0 = (x_1^0, \ldots, x_{i-1}^0, x_i^0, x_{i+1}^1, \ldots, x_n^1)$
  $\mathbf{z}^1 = (x_1^0, \ldots, x_{i-1}^0, x_i^1, x_{i+1}^1, \ldots, x_n^1)$
On input $(c, (\Lambda_j)_{j \neq i})$ from the challenger, forward $\mathcal{D} \leftarrow c$
When $\mathcal{D}$ queries j with $x_j^0 = x_j^1$:
  Forward the right opening proof $\mathcal{D} \leftarrow \Lambda_j$
When $b \leftarrow \mathcal{D}$: Return b

Fig. 13. Reduction $\mathcal{B}$ using $\mathcal{D}$ to win at $\mathsf{ExpHideVC}$.

identical. Thus, calling eq the event $x_i^0 = x_i^1$ and $\overline{\mathsf{eq}}$ its negation,

$$\begin{aligned}
\mathsf{Adv}(\mathcal{D}) &= |\Pr[\mathcal{D} \to 0 | \mathsf{H}_{i-1}] - \Pr[\mathcal{D} \to 1 | \mathsf{H}_i]| \\
&\leq |\Pr[\mathcal{D} \to 0 | \mathsf{H}_{i-1},\ \mathsf{eq}] - \Pr[\mathcal{D} \to 0 | \mathsf{H}_i,\ \mathsf{eq}]| \ + \\
&\qquad \ldots + \ |\Pr[\mathcal{D} \to 0 | \mathsf{H}_{i-1},\ \overline{\mathsf{eq}}] - \Pr[\mathcal{D} \to 0 | \mathsf{H}_i,\ \overline{\mathsf{eq}}]| \\
&= |\Pr[\mathcal{B} \to 0 | \beta = 0] - \Pr[\mathcal{B} \to 0 | \beta = 1]| \ \leq \ \mathsf{Adv}(\mathcal{B}).
\end{aligned}$$

Thus $\mathsf{Adv}(\mathcal{D})$ is negligible, concluding the proof.

A.4 Hiding VC to Signatures

Proof of Proposition 2. Given an adversary $\mathcal{A}$ breaking unforgeability of the signature scheme described in Fig. 3, we build in Figure 14 an adversary $\mathcal{B}$ playing against the (simpler) hiding game for Vector Commitment described in Fig. 2.

We first state the following claim, where b is the challenge bit chosen by $\mathcal{B}$'s challenger and forge the event $1 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k^*)$.

Claim 1 *Exists ε negligible such that* $\Pr\left[\mathsf{forge},\ k = i,\ x_k^* \neq x_i^0 \ \middle|\ b = 0\right] \leq \varepsilon$.

These claims implies the thesis since

$$\begin{aligned}
2 \cdot \mathsf{Adv}(\mathcal{B}) &= |\Pr[\mathcal{B} \to 0 \mid b = 0] - \Pr[\mathcal{B} \to 0 \mid b = 1]| \\
&\geq \Pr[\mathcal{B} \to 0 \mid b = 0] \\
&\geq \Pr\left[\mathsf{forge},\ k = i,\ x_k^* = x_i^0 \ \middle|\ b = 0\right] \\
&\geq \Pr[\mathsf{forge},\ k = i \mid b = 0] - \Pr\left[\mathsf{forge},\ k = i,\ x_k^* \neq x_i^0 \ \middle|\ b = 0\right] \\
&\geq \Pr[\mathsf{forge} \mid b = 0,\ k = i] \Pr[k = i \mid b = 0] - \varepsilon \\
&\geq \mathsf{Adv}(\mathcal{A}) \cdot \frac{1}{n} - \varepsilon
\end{aligned}$$

$\mathcal{B}(\mathsf{pp})$:

1 : Guess the index $\mathcal{A}$ is going to forge, $i \xleftarrow{\$} [n]$

2 : Sample $x_i^0, x_i^1 \xleftarrow{\$} \mathsf{VC.M}$ with $x_i^0 \neq x_i^1$

3 : For all $j \neq i$: $x_j \xleftarrow{\$} \mathsf{VC.M}$, and set $x_j^0 \leftarrow x_j$, $x_j^1 \leftarrow x_j$

4 : Query $\mathbf{x}^0, \mathbf{x}^1$ and wait for $(c, (\Lambda_j)_{j \neq i})$

5 : Run $\mathcal{A}(\mathsf{pp}, c)$

6 : **When** $\mathcal{A}$ queries j:

7 :   **If** $j = i$: Return 1 // i.e. abort

8 :   **Else**: $\mathcal{A} \leftarrow (x_j, \Lambda_j)$

9 : **When** $\mathcal{A}$ returns (k, x_k^*, Λ_k^*)

10 :   **If** $k \neq i$ or $0 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k^*)$ or $x_k^* = x_i^1$:

11 :   Return 1

12 :   **Else**: Return 0

Fig. 14. Reduction $\mathcal{B}$ executed in the hiding game of Fig. 2.

where we used the fact that $\mathcal{A}$ has no information on i, thus $\Pr[k = i] = 1/n$ and forge is independent on $k = i$.

Thus $\mathsf{Adv}(\mathcal{A}) \leq 2n\mathsf{Adv}(\mathcal{B}) + n\varepsilon$, that is negligible. Next we prove the Claim providing a reduction $\mathcal{C}$ that uses $\mathcal{A}$ to break position binding. The idea is that $\mathcal{C}$ can emulate the behavior of $\mathcal{B}$ when $b = 0$ until the final step. If the adversary manages to produce an opening to a message x_k^* for the right position $k = i$ but with $x_k^* \neq x_i^0$, then $\mathcal{C}$ breaks position binding returning this opening for x_k^*, and the correct one he can generate for x_i^0. A full description appears in Figure 15.

By inspection $\mathcal{C}$ simulates $\mathcal{B}$ when $b = 0$ and $\mathbf{x} = \mathbf{x}^0$ correctly. Note that there is no need to simulate $\mathbf{x}^1$ as when $b = 0$, $\mathcal{A}$ gets no information about it. Finally, if the condition at step 9 is executed, $\mathcal{C}$ breaks position binding correctly since $x_k^* \neq x_i$, $1 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k^*)$ and, by correctness, $1 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_i, i, \Lambda_i)$ with $i = k$. Hence

$$\mathsf{Adv}(\mathcal{C}) = \Pr\left[\mathsf{forge},\ k = i,\ x_k^* \neq x_i^0 \;\middle|\; b = 0\right].$$

which proves the claim.

A.5 Hard Subset Membership Problem

Proof of Lemma 2. We will show the Lemma holds using the following claim which we prove later on

Claim 1 *Given* $b_i \xleftarrow{\$} \{0,1\}$, $b_i^* \xleftarrow{\$} \{0,1\}$ *and* x_i *such that*

$$b_i = 0 \;\Rightarrow\; x_i \xleftarrow{\$} \mathsf{SampBad}(1^\lambda), \qquad b_i = 1 \;\Rightarrow\; x_i, w_i \xleftarrow{\$} \mathsf{SampGood}(1^\lambda).$$

$\mathcal{C}(\mathsf{pp})$:

// Simulate $\mathcal{B}$ and its challenger
Sample $i \xleftarrow{\$} [n]$ and $\mathbf{x} \xleftarrow{\$} \mathsf{VC.M}^n$
$(c, \mathsf{aux}) \leftarrow \mathsf{VC.Com}(\mathsf{pp}, \mathbf{x})$
Run $\mathcal{A}(\mathsf{pp}, c)$
When $\mathcal{A}$ queries j:
  If $j = i$: Return $\perp$
  Else: $\pi_j \leftarrow \mathsf{VC.Open}(\mathsf{pp}, j, \mathsf{aux})$, $\mathcal{A} \leftarrow (x_i, \Lambda_i)$
When $\mathcal{A}$ return (k, x_k^*, Λ_k^*)
  If $k \neq i$ and $\mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k)$ and $x_k^* \neq x_i$:
    $\Lambda_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}, i, \mathsf{aux})$
    Return $(c, i, x_k^*, \Lambda_k^*, x_i, \Lambda_i)$
  Else: Return $\perp$

Fig. 15. Reduction $\mathcal{C}$ breaking position binding.

for $i \in [\lambda]$*, then the distributions* $(\mathbf{x}, \mathbf{b})$ *and* $(\mathbf{x}, \mathbf{b}^*)$ *are hard to distinguish, i.e.* $\exists \varepsilon$ *negligible such that for all* GPPT *adversaries* $\mathcal{D}$

$$\mathsf{Adv}(\mathcal{D}) \;=\; \left|\Pr\left[\mathcal{D}(\mathbf{x}, \mathbf{b}) \to 0\right] \;-\; \Pr\left[\mathcal{D}(\mathbf{x}, \mathbf{b}^*) \to 0\right]\right| \;\leq\; \varepsilon.$$

Assuming the claim holds, let $\mathcal{A}$ be an adversary trying to solve λ hard subset membership challenges we can build a trivial distinguisher $\mathcal{D}$ that on input $(\mathbf{x}, \mathbf{c})$, executes $\mathcal{A}(\mathbf{x}) \to \mathbf{b}'$ and returns 0 if $\mathbf{c} = \mathbf{b}'$ and 1 otherwise. If $\mathcal{D}$ is executed with $\mathbf{c} = \mathbf{b}$, then the probability that it returns 0 is by definition $\mathsf{Adv}(\mathcal{A})$, where

$$\mathsf{Adv}(\mathcal{A}) \;=\; \Pr\left[\mathcal{A}(\mathbf{x}) \to \mathbf{b}',\; \mathbf{b}' = \mathbf{b}\right].$$

Conversely, if $\mathbf{c} = \mathbf{b}^*$ then $\mathcal{A}$ has no information on $\mathbf{b}^*$. Therefore its guess is independent from it and $\Pr\left[\mathbf{b}' = \mathbf{b}^*\right] = 2^{-\lambda}$. In conclusion we have that

$$\mathsf{Adv}(\mathcal{D}) \;=\; \left|\mathsf{Adv}(\mathcal{A}) - \frac{1}{2^\lambda}\right| \quad \Rightarrow \quad \mathsf{Adv}(\mathcal{A}) \;\leq\; \frac{1}{2^\lambda} + \mathsf{Adv}(\mathcal{D}) \;\leq\; \frac{1}{2^\lambda} + \varepsilon(\lambda).$$

which proves the Lemma.

Proof of Claim 1. With the above notation we define the following hybrid distributions

$$\sigma_i \;=\; (\mathbf{x}, b_1, \ldots, b_i, b_{i+1}^*, \ldots, b_\lambda^*)$$

and claim that distinguishing σ_{i-1} from σ_i is hard for all i, which implies the thesis. First of all, for notational convenience, we will call

$$\tau_i = (\mathbf{x}, b_1, \ldots, b_{i-1}, b_{i+1}^*, \ldots, b_\lambda^*)$$

which contains all the entries that σ_{i-1} and σ_i have in common, so that up to reordering

$$\sigma_{i-1} = (\tau_i, b_i), \quad \sigma_i = (\tau_i, b_i^*).$$

Next we study the advantage of a given GPPT algorithm $\mathcal{D}$ distinguishing σ_{i-1} from σ_i

$$\begin{aligned}
\mathsf{Adv}(\mathcal{D}) &= \left|\Pr\left[\mathcal{D}(\tau_i, b_i) \to 0\right] - \Pr\left[\mathcal{D}(\tau_i, b_i^*) \to 0\right]\right| \\
&= \Bigg|\frac{1}{2}\Pr\left[\mathcal{D}(\tau_i, 0) \to 0|b_i = 0\right] + \frac{1}{2}\Pr\left[\mathcal{D}(\tau_i, 1) \to 0|b_i = 1\right] \\
&\quad -\frac{1}{4}\Pr\left[\mathcal{D}(\tau_i, 0) \to 0|b_i = 0\right] - \frac{1}{4}\Pr\left[\mathcal{D}(\tau_i, 1) \to 0|b_i = 0\right] \\
&\quad -\frac{1}{4}\Pr\left[\mathcal{D}(\tau_i, 0) \to 0|b_i = 1\right] - \frac{1}{4}\Pr\left[\mathcal{D}(\tau_i, 1) \to 0|b_i = 1\right]\Bigg| \\
&\leq \left|\frac{1}{4} \cdot \Pr\left[\mathcal{D}(\tau_i, 0) \to 0|b_i = 0\right] - \frac{1}{4} \cdot \Pr\left[\mathcal{D}(\tau_i, 0) \to 0|b_i = 1\right]\right| + \\
&\quad + \left|\frac{1}{4} \cdot \Pr\left[\mathcal{D}(\tau_i, 1) \to 0|b_i = 0\right] - \frac{1}{4} \cdot \Pr\left[\mathcal{D}(\tau_i, 1) \to 0|b_i = 1\right]\right|.
\end{aligned}$$

where the second equation follow conditioning on all possible values of b_i and b_i^* and observing that $\mathcal{D}$ on a fixed input does not depend on b_i^* (although it does depend on b_i since x_i depends on b_i). Thus it suffice to show that this two terms are negligible. Toward this goal we build two adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$, each trying to solve a single hard subset membership problem. $\mathcal{B}_\beta(x)$ begins sampling b_j, b_j^* and x_j as in the claim statement for all $j \neq i$. Then it calls $x_i = x$ and returns the bit it gets from $\mathcal{D}(\tau_i, \beta)$ (note β is a constant value here). The probability that $\mathcal{B}_\beta$ guesses correctly is then, calling $b = b_i$ the challenge bit it has to guess

$$\begin{aligned}
&\left|\Pr\left[\mathcal{B}_\beta(x) \to 0|b = 0\right] + \Pr\left[\mathcal{B}_\beta(x) \to 0|b = 1\right]\right| \\
= &\left|\Pr\left[\mathcal{D}(\tau_i, \beta) \to 0|b = 0\right] - \Pr\left[\mathcal{D}(\tau_i, \beta) \to 0|b = 1\right]\right|
\end{aligned}$$

By Definition 4 we have that the left hand side is smaller than a negligible ε for all $\beta \in \{0, 1\}$, eventually implying that $\mathsf{Adv}(\mathcal{D}) \leq \varepsilon/2$.

A.6 Preliminary Adversary

Proof of Lemma 3. In order to provide a description of $\mathcal{A}$, we begin by building a signature scheme given a NIZK whose message space has only one element. This will allow us to use the adversary $\mathcal{B}$ described in [CFGG23] which, on an algebraic signature scheme with a single message, either produces a forgery or finds a linear relation among the group elements of the verification key. The idea is, given a NIZK for an hard subset membership problem, to set the verification key as the crs and a false statement x, and the signing key is the simulation trapdoor td. A signature for the only message 0 is then any proof π for x. In this way the signer can create a proof for x using the simulation trapdoor, while

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

$$\begin{array}{ll} 1: & \text{Sample } (\mathsf{crs},\mathsf{td}) \leftarrow \mathsf{S}(1^\lambda),\ x \leftarrow^{\$} \mathsf{SampBad}(1^\lambda) \\ 2: & \text{Set } \mathsf{vk} \leftarrow (\mathsf{crs}, x),\ \mathsf{sk} \leftarrow \mathsf{td} \text{ and Return } (\mathsf{vk},\mathsf{sk}) \end{array}$$

$\mathsf{S.Sign}(\mathsf{sk}, 0)$:

$$1: \quad \text{Return } \pi \leftarrow \mathsf{S}(\mathsf{td}, x)$$

$\mathsf{S.Vfy}(\mathsf{vk}, 0, \pi)$:

$$1: \quad \text{Return } b \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$$

Fig. 16. Signature scheme from any NIZK for a hard subset membership problem.

no adversary can provide a proof for x unless soundness does not hold. A full description of the scheme is presented in Fig. 16.

Given this signature scheme, calling $x = (\mathbf{Z}, x') \in \mathbb{G}^m \times \{0,1\}^*$ and $\mathsf{crs} = (\mathbf{Y}, c') \in \mathbb{G}^n \times \{0,1\}^*$, we claim as in [CFGG23] that

Claim 1 *There exists a* GPPT *adversary* $\mathcal{B}$ *such that, given* $V \leq \mathbb{F}_q^n$ *and* $W \leq \mathbb{F}_q^m$ *containing respectively the discrete logarithm of* $\mathbf{Y}$ *and* $\mathbf{Z}$*, either*

- $\mathcal{B}(V, W, \mathsf{vk}) \to (\pi, L)$*, with* π *a valid forgery*
- $\mathcal{B}(V, W, \mathsf{vk})$ *queries a signature for* 0 *and upon receiving a valid* π*, such that*

$$\mathcal{B}(V, W, \mathsf{vk}) \to (\perp, L) \quad \Rightarrow \quad L \lneq V \times W, \quad (\mathbf{Y}, \mathbf{Z}) \in L \cdot G.$$

Note that this adversary almost satisfies the property we wish $\mathcal{A}$ to have. However, when no forgery is found, it instead returns a linear relation among all the group elements in the verification key $(\mathbf{Y}, \mathbf{Z})$ and not only $\mathbf{Y}$. This means that linear relations found could be trivial in $\mathbf{Y}$.

In order to refine this adversary we use a technique also introduced in [CFGG23]: The idea is to run $\mathcal{B}$ several times in a simulated environment with the real statement x and a fresh crs^* generated with a trapdoor td^*. In this simulation either $\mathcal{A}$ returns a *bad* L from which $\mathcal{B}$ can extract a linear relation among the elements of x, or for sufficiently many times L satisfies

$$L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}$$

If this is the case, then $\mathcal{A}$ executes $\mathcal{B}$ one last time with the real crs it receives and, if needed, replies to the signature query from $\mathcal{B}$ using its only simulation query. Since the space L satisfied the above property for sufficiently many iterations, it will likely be satisfied also in this last one.

One issue with this approach is that $\mathbf{z}$ is not known to $\mathcal{A}$, so testing $L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}$ might be hard. The next claim address this.

Claim 2 *Given* $V \leq \mathbb{F}_q^n$*,* $W \leq \mathbb{F}_q^m$ *and* $L \leq V \times W$ *affine spaces and calling* $\eta_2 : \mathbb{F}_q^n \times \mathbb{F}_q^m \to \mathbb{F}_q^m$ *the projection on the second entry*[11]*, then for all* $\mathbf{z} \in W$

$$L \leq V \times W, \quad \eta_2(L) = W \quad \Rightarrow \quad L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}.$$

[11] i.e. $\eta(\mathbf{x}, \mathbf{y}) = \mathbf{y}$. Typically projections are denoted with π, but we have to depart from this notation to avoid any confusion as π already denotes proofs.

We are now ready to give a description of the adversary $\mathcal{A}$ parametrized by a polynomially bounded t, which appears in Fig. 17.

Given this algorithm we break the proof that $\mathcal{A}_t$ is indeed the right algorithm for some t into the following claims.

Claim 3 *$\mathcal{A}$ is* GPPT*.*

Claim 4 *For each step of the execution of $\mathcal{A}$, calling $x = (\mathbf{Z}, x') \in \mathbb{G}^m \times \{0,1\}^*$, then $\mathbf{Z} \in W \cdot G$.*

Claim 5 *For any choice of $t = \mathsf{poly}(\lambda)$*

$$\Pr\left[\mathcal{A}(V, \mathsf{crs}, x) \to \perp\right] \;\le\; \frac{m+1}{t+1}.$$

These claims imply the thesis since $\mathcal{A}$ only uses the group efficiently and, setting $t = (m+1)\cdot p(\lambda) - 1$, aborts with probability $p(\lambda)^{-1}$. Finally, if it does not abort either it returns a proof π for x, which happens without performing any query since $\mathcal{A}$ ask for a proof if and only if $\mathcal{B}$ ask for a signature, or it outputs L with $\eta_2(L) = W$. In this latter case, since $\mathbf{z} \in W$ by Claim 4, Claim 2 implies that

$$L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}$$

where we used the fact that by Claim 1, $\mathcal{A}$ returns $L \subseteq V \times W$.

Proof of Claim 2. By contradiction assume $L \cap \mathbb{F}_q^n \times \{\mathbf{z}\} = V \times \{\mathbf{z}\}$. Then for all $(\mathbf{v}, \mathbf{w}) \in V \times W$, since $\eta_2(L) = W$ there exists a point $\mathbf{u} \in V$ such that $(\mathbf{u}, \mathbf{w}) \in L$. Using the initial hypothesis we also have that

$$(\mathbf{u}, \mathbf{z}), (\mathbf{v}, \mathbf{z}) \in L \quad\Rightarrow\quad (\mathbf{u}, \mathbf{w}) + (\mathbf{v}, \mathbf{z}) - (\mathbf{u}, \mathbf{z}) \in L \quad\Rightarrow\quad (\mathbf{v}, \mathbf{w}) \in L.$$

Hence $V \times W \le L$ which is a contradiction as we assumed $L \lneq V \times W$.

Proof of Claim 3. Since t and m are polynomially bounded, it suffice to show that each individual line can be computed in GPPT time. This is evident for all commands with the exception of Line 3. That step however can be computed inefficiently, but with only polynomially many group operations. This is done first computing the conditional distribution of the *exponents* of the crs, conditioned to $\mathbf{Y} \in V \cdot G$, sampling from this distribution (which take exponential space), and finally computing crs from the sampled exponents, which takes polynomially many group operations.

Proof of Claim 4. We proceed by induction. Initially $\mathbf{Z} \in W \cdot G$ since $W = \mathbb{F}_q^m$. Next assume that until the i-th iteration of the outer for-loop, $\mathbf{Z} \in W \cdot G$. At the end of the loop either $W' = W$, implying that the thesis still holds, or $W' \neq W$. In this second case $W' = \eta_2(L)$ with L being the output of $\mathcal{B}(V, W, \mathsf{crs}^*, x)$. Since by hypothesis $\mathbf{Y} \in V \cdot G$ and $\mathbf{Z} \in W \cdot G$, we have that $(\mathbf{Y}, \mathbf{Z})$ is contained in $V \times W$. Calling $\mathbf{y}, \mathbf{z}$ the discrete logarithm respectively of $\mathbf{Y}$ and $\mathbf{Z}$, we have that

$$(\mathbf{y}, \mathbf{z}) \in L \quad\Rightarrow\quad \mathbf{z} = \eta_2(\mathbf{y}, \mathbf{z}) \in \eta_2(L) = W' \quad\Rightarrow\quad \mathbf{Z} \in W' \cdot G.$$

$\mathcal{A}_t(V, \mathsf{crs}, x)$:

Initialize $W \leftarrow \mathbb{F}_q^m$ the space of possible exponents for $\mathbf{Z}$
For $j \in \{1, \ldots, m+1\}$:
  // Each execution tries to reduce $\dim W$
  Set $W' \leftarrow W$ an affine space storing information on x gathered later on
  For $i \in \{1, \ldots, t\}$:
    Sample $(\mathsf{crs}^*, \mathsf{td}^*) \leftarrow \mathsf{S}(1^\lambda)$ with $\mathsf{crs}^* = (\mathbf{Y}^*, c^*)$ and $\mathbf{Y}^* \in V \cdot G$
    Run $\mathcal{B}(V, W, \mathsf{crs}^*, x)$
    When $\mathcal{B}$ queries a signature for 0:
      Compute a simulated proof $\pi \leftarrow \mathsf{S}(\mathsf{td}^*, x)$ and send it $\mathcal{B} \leftarrow \pi$
    When $\mathcal{B}$ returns (π, L):
      If $\eta_2(L) \neq W$: Store the result $W' \leftarrow \eta_2(L)$
  If $W' \lneq W$: Update $W' \leftarrow W$
  Else: Break the outer for-loop
// Execute $\mathcal{B}$ one last time with the real crs
Run $\mathcal{B}(V, W, \mathsf{crs}, x)$
When $\mathcal{B}$ queries a signature for 0:
  Return query and on input π send $\mathcal{B} \leftarrow \pi$
When $\mathcal{B}$ returns (π, L)
  If π is a valid proof: Return π
  Elif $\eta_2(L) = W$: Return L
  Else: Return $\perp$

Fig. 17. Adversary $\mathcal{A}_t$ parametrized by $t = \mathsf{poly}(\lambda)$.

Proof of Claim 5. We define J a random variable denoting the value index j has before terminating the outer loop by executing Line 12. Note that J is well defined since each time Line 12 is not executed, setting $W \leftarrow W'$ reduces the dimension of W by 1 which can only happens at most m times.

Next we define the event $\mathsf{E}_{i,j}$ as $J \geq j$ and at the i-th iteration of the inner for loop, the condition of Line 10 is not satisfied. We further let Fail be the event $\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp$. Note that the events $\mathsf{E}_{i,j}$ are also well defined since either $J < j$ or when $J \geq j$ the inner loop is executed for all i.

Next we observe that, for each j, the inner loop runs $\mathcal{A}$ with the same vector spaces (V, W) and equally distributed crs^* and x. Thus we have that $\Pr[\mathsf{E}_{i,j}]$ is constant for all j, allowing us to define

$$p_j \; = \; \Pr[E_{1,j}] \; = \; \ldots \; = \; \Pr[E_{t,j}] \, .$$

Next, we observe that conditioning on $J = j$, Fail occurs if and only if $\mathcal{B}(V, W, \mathsf{crs}, x)$ returns a vector space L satisfying $\eta_2(L) = W$ with crs following the same distribution simulated in Line 3. Thus

$$\Pr[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \mid J = j] \; = \; 1 - p_j$$

In conclusion

$$\begin{aligned}
\Pr[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp] \; &\leq \; \sum_{j=1}^{m+1} \Pr[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \mid J = j] \Pr[J = j] \\
&\leq \; \sum_{j=1}^{m+1} (1 - p_j) \Pr[\mathsf{E}_{1,j} \; \wedge \; \ldots \; \wedge \; \mathsf{E}_{t,j}] \\
&\leq \; \sum_{j=1}^{m+1} (1 - p_j) \cdot p_j^t \; \leq \; \sum_{j=1}^{m+1} \frac{1}{t+1} \; \leq \; \frac{m+1}{t+1} .
\end{aligned}$$

where in the third step we used the fact that the events $E_{i,j}$ for a fixed j are mutually independent and in the second to last step, we upper bound $(1 - p_j) \cdot p_j^t \leq (t+1)^{-1}$ since $p_j \in [0, 1]$.