Algebraic Group Model with Oblivious Sampling[⋆]

October 2, 2023

Helger Lipmaa$^{1[0000-0001-8393-6821]}$, Roberto Parisella$^{2[0009-0007-2241-801X]}$, and Janno Siim$^{2[0000-0001-5824-7215]}$

[1] University of Tartu, Tartu, Estonia
[2] Simula UiB, Bergen, Norway

Abstract. In the algebraic group model (AGM), an adversary has to return with each group element a linear representation with respect to input group elements. In many groups, it is easy to sample group elements obliviously without knowing such linear representations. Since the AGM does not model this, it can be used to prove the security of spurious knowledge assumptions. We show several well-known zk-SNARKs use such assumptions. We propose AGM with oblivious sampling (AGMOS), an AGM variant where the adversary can access an oracle that allows sampling group elements obliviously from some distribution. We show that AGM and AGMOS are different by studying the family of "total knowledge-of-exponent" assumptions, showing that they are all secure in the AGM, but most are not secure in the AGMOS if the DL holds. We show an important separation in the case of the KZG commitment scheme. We show that many known AGM reductions go through also in the AGMOS, assuming a novel falsifiable assumption TOFR. We prove that TOFR is secure in a version of GGM with oblivious sampling.

Keywords: Admissible encoding · algebraic group model · elliptic-curve hashing · FindRep · KZG extractablity · oblivious sampling

1 Introduction

GGM. One of the most influential idealized models of computation in cryptography is the generic group model (GGM, [Nec94,Sho97,Mau05]). GGM models the situation where an adversary $\mathcal{A}$ operates in a (usually abelian, possibly bilinear) group. In the GGM, $\mathcal{A}$'s operations on group elements are "generic" (typically addition, equality test, pairing operation), i.e., they do not depend on the concrete group. One models this either by giving the adversary access to random encodings [Sho97] or abstract handles of group elements [Mau05], together with oracles that perform group operations and equality tests on given encodings (resp., handles). $\mathcal{A}$ cannot access any other information about the group elements, including their bit representations. $\mathcal{A}$ does not even have access to the

[⋆] This is a full version of [LPS23].

group description, except (usually) its order. Most elliptic-curve cryptanalysis algorithms are generic. That is, the algorithms do not exploit any particular structure of the group.

While the GGM is used widely to argue about security in group-based settings, the GGM has several well-known weaknesses [Fis00,Den02] that have motivated researchers to propose more realistic idealized models. In particular, GGM makes the questionable assumption that the adversary cannot conduct more efficient attacks by accessing the bit-presentation of group elements.

AGM. Fuchsbauer et al. [FKL18] proposed the more realistic algebraic group model (AGM) with *algebraic* adversaries. The AGM does not assume the adversary's ignorance about the group description or bit-representation of the group elements. Instead, the AGM is a generalized knowledge assumption [Dam92], stating that an algebraic adversary $\mathcal{A}$ must know a linear representation of an output group element with respect to input group elements. Notably, a group element's creation can depend on the group description and already known group elements' bit-presentation. The knowledge of the linear representation is modeled by requiring the adversary, together with each group element, to output a linear dependence from the group elements seen thus far. More formally, given (for example) input group elements $[x_1, \ldots, x_n]_1$ [3], if the adversary outputs $[y]_1$, it has to also output integers $v_1, \ldots, v_n$ such that $[y]_1 = \sum_{i=1}^{n} v_i [x_i]_1$.

Oblivious Sampling. Since a real-life adversary is not restricted to group operations and equality tests, AGM does not always capture all (known) possible attacks. In particular, it was realized early [Bro01] in the context of GGM that one must additionally model the adversary's ability to sample group elements obliviously without knowing the linear representations.

We point out (we seem to be the first to make this connection) that in the case of elliptic curve groups, oblivious sampling is not just a theoretical possibility but concrete and *provable* (see Section 2.1 for a proof) attack due to admissible encodings [BF01,SW06,Ica09,BCI$^+$10,FT10,WB19]. Admissible encodings are efficiently computable functions E from $\mathbb{F} = \mathbb{Z}_p$ to elliptic curve groups that are regular (small preimage sizes) and preimage sampleable (given $[y]_1 \in \operatorname{Im}(E)$, one can efficiently recover its whole preimage). Admissible encodings allow an adversary $\mathcal{A}$ to sample group elements obliviously without knowing their discrete logarithms [Ica09] and even linear representations: $\mathcal{A}$ can do it by sampling $s \leftarrow\!\!\$\ \mathbb{F}$ and outputting $E(s)$. Since admissible encodings exist for all curves and are often constant-time computable, *we argue that elliptic curve group adversaries are not algebraic.* We emphasize that admissible encodings are just one approach to oblivious sampling. Many others may exist, and it is crucial to guarantee security against all of them.

[3] Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear pairing. Let the order of the groups be a prime p and denote $\mathbb{F} = \mathbb{Z}_p$. We use the standard additive bracket notation, denoting a group element as $z \cdot [1]_\kappa = [z]_\kappa \in \mathbb{G}_\kappa$, where $[1]_\kappa$ is a generator of an additive abelian group $\mathbb{G}_\kappa$, by $[z]_\kappa$. We denote the pairing by $\bullet : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Without modeling oblivious sampling, one can prove in the GGM and the AGM the security of spurious knowledge assumptions [Bro01,SPMS02,BFS16]. Consider the following *SpurKE* (*spurious knowledge of exponent*) assumption: if the adversary on input $[1]_1$ outputs $[x]_1$ then it must know x. First, we only consider groups where the DL (discrete logarithm) assumption holds since otherwise, SpurKE and many other knowledge assumptions hold trivially. As already argued in [Bro01], if the adversary samples $[x]_1$ obliviously (and DL holds), then SpurKE does not hold. However, SpurKE holds in the standard GGM and AGM (even when DL holds); in AGM, the linear representation is just x. It is a severe shortcoming of standard GGM and AGM that assumptions like SpurKE can be proven secure. SpurKE is not the only bad apple: the AGM allows one to prove the security of many similar spurious knowledge assumptions.

This shortcoming of AGM has misled researchers to use SpurKE. In the KZG polynomial commitment scheme [KZG10], the committer gets as an input a public key $\mathsf{pk} = ([1, x, \ldots, x^d]_1, [1, x]_2)$. To commit to a polynomial $f(X) = \sum_{i=0}^{d} f_i X^i$, the committer computes $[c]_1 \leftarrow \sum_{i=1}^{d} f_i [x^i]_1$. (See Section 5.3 for the full construction.) Campanelli et al. [CFF$^+$20] (the full version of [CFF$^+$21]), Section 7.3, suggest a trivial proof of knowledge for KZG commitment, where the proof is empty. They motivate this by AGM since, in AGM, the polynomial coefficients can be extracted from $[c]_1$ alone. However, extracting the polynomial directly from a commitment corresponds to the SpurKE assumption and is thus intractable.

Similarly, in the knowledge-soundness proof of Plonk [GWC19], Gabizon et al. write the following (Remark 3.2, [GWC19]): *"the algebraic group model is crucial for allowing us to model both binding and knowledge soundness in one clean game - without it, we typically cannot require E to return the polynomial immediately after A's commitment."*. Again, the authors rely on immediate extraction from the commitment, hence relying on SpurKE. This does not necessarily imply a vulnerability in Plonk. (We show later that the polynomial can be extracted if a KZG commitment is opened at some evaluation point.) However, it shows that SpurKE has been used (albeit sneakily) in well-known, widely deployed SNARKs like Plonk. In Appendix E, we give more details, showing that a common trick used to optimize quadratic tests in KZG-based zk-SNARKs results in non-extractability. We will leave it to future work to establish whether such SNARKs (including other KZG-based SNARKs in the literature) can be proven secure in AGMOS. Even if they are secure, they will need a different security proof. More generally, this shows that relying on SpurKE and equivalent assumptions is more common than expected.

Modelling Oblivious Sampling. An augmented GGM that models oblivious sampling is often called *GGM with hashing* (GGMH, [Bro01,BFS16,ALSZ21]). In GGMH, the adversary is given access to an oracle that obliviously samples from the uniform distribution over the group.

[FKL18] briefly discusses the oblivious sampling issue under the heading of "Integrating AGM with random oracles," stating that algebraic adversaries cannot do oblivious sampling. They extend AGM to the setting of protocols that explicitly use a random oracle (RO) that outputs group elements (in particular, to prove the security of the BLS signature scheme). In such cases, they consider RO answers semantically equivalent to input elements. Thus, they require that an algebraic adversary knows a representation of its output group elements as a linear combination of inputs and RO answers. We will call this extension *RO-AGM*. [FKL18] does not analyze the security of protocols where the honest participants do not use RO, but the adversary uses RO to obliviously sample group elements.

[Lip22] added oblivious sampling to the AGM. However, for this, [Lip22] uses the *fully-programmable random oracle* (FPRO, [Nie02,FLR$^+$10]) model. FPRO is even less realistic than the *non-programmable random oracle* (NPRO) model, where the reduction is only allowed to query and forward the answers of the RO. Moreover, this does *not* model admissible encodings that are real-world, well-defined, non-programmable, deterministic functions.

Let us try to understand the issues we face, including how the FPRO comes in. First, by the definition of the AGM, an algebraic adversary $\mathcal{A}$ always returns a linear representation of its output, which is typically used in a reduction proof. For instance, we can prove in AGM that Computational Diffie-Hellman (CDH) is not easier to break than DL. An algebraic CDH adversary $\mathcal{A}$, on input $[1, a, b]_\kappa$, computes $[ab]_\kappa$, and must also output integers v_1, v_2, v_3 such that $[ab]_\kappa = v_1[1]_\kappa + v_2[a]_\kappa + v_3[b]_\kappa$. A DL reduction $\mathcal{B}$ can set the DL challenge as $[a]_\kappa$, sample a random integer b, and invoke $\mathcal{A}$. Observe that if $\mathcal{A}$ succeeds in attacking CDH, the polynomial $V(X) = v_1 + v_2X + v_3y - Xb$ has a as a root. Thus, $\mathcal{B}$ can compute and return a. The discrete logarithm a will be correctly computed if V is a non-zero polynomial in X. [4] However, when proving SpurKE and similar assumptions, we should not assume that $\mathcal{A}$ returns the linear representation.

The (more challenging) second issue concerns how one constructs reductions $\mathcal{B}$ in the AGM, like the one one above. In a typical AGM proof, one analyzes an assumption with a verification polynomial $V(\boldsymbol{X})$ that depends on the discrete logarithms of the input and output group elements. $\mathcal{B}$ uses the extracted linear representation to extract all V's coefficients. The verification equation stipulates that $V(\boldsymbol{x}) = 0$. After that, one analyzes two cases. "Case A" (*algebraic*), where V is a zero polynomial: typically, either $V(\boldsymbol{X}) = 0$ is impossible (computational assumptions) or the extraction succeeds (knowledge assumptions).

Alternatively, one is in "Case X" (*$\boldsymbol{X}$-related case*) where $V(\boldsymbol{X}) \neq 0$ as a polynomial but $V(\boldsymbol{x}) = 0$. In this case, one constructs a reduction to a standard computational assumption like (d_1, d_2)-PDL (Power Discrete Logarithm, given $[1, y, \dots, y^{d_1}]_1$ and $[1, y, \dots, y^{d_2}]_2$ for random y, it is intractable to recover y).

A typical reduction $\mathcal{B}$ to PDL implicitly embeds random affine functions of $\mathcal{B}$'s input trapdoor y to all the coordinates of $\boldsymbol{x}$. (This step is only needed when V is multivariate.) This results in a *univariate* polynomial V^*, such that $V^*(Y) \neq 0$ but $V^*(y) = 0$. Using univariate polynomial factorization, $\mathcal{B}$ recovers y. Consider the case of oblivious sampling with RO. Then, $[\boldsymbol{x}]_1$ also includes

[4] The actual reduction is slightly more complicated since we need to guarantee that V is a non-zero polynomial.

RO answers, which means that a PDL reduction must implicitly embed y to the RO answers. Since now the RO must be programmed, this results in using the FPRO model in [Lip22]. See [Lip22] for a discussion on the differences between this approach and the approach of [FKL18].

Since SpurKE-like assumptions do not hold (assuming DL holds) in [Lip22]'s version of AGM, this answers one of our concerns. Unfortunately, it still relies on using FPRO. Pitfalls of FPRO are well understood; see [Nie02,FLR$^+$10] for extended discussions. Modeling admissible encodings (existing, efficient functions) via a RO would remove one of the advantages of the AGM over the GGM, the ability to argue about concrete bit representations of existing real-life objects.

Moreover, the outputs of most admissible encodings are not distributed uniformly, while the GGMH, the RO-AGM of [FKL18], and [Lip22] only consider the uniform distribution. Since both GGMH and RO-AGM only consider uniformly distributed outputs, they do not model properly admissible encodings also in this aspect. For example, Icart's admissible encoding [Ica09] has domain size $\approx 5/8$ of the group size, and thus it can be easily distinguished from an RO that outputs uniformly random group elements. In addition, one can choose a non-uniform input distribution for the admissible encoding or combine several known admissible encodings. Even if one is willing to use the FPRO model, one has the problem of embedding the input of the PDL adversary to the RO answers that come from non-uniform distributions. It is not clear how to do it generically.

Main Questions of this Work. The previous discussion leads us to the following question: *how to extend the AGM to model oblivious sampling without needing the FPRO (or even the NPRO)?* This modeling should take into account that admissible encodings can be used to sample from non-uniform distributions.

Our Contributions. Firstly, we formally establish that oblivious sampling is possible in elliptic curve groups using admissible encodings. Thus, an oblivious sampling extension to AGM is indeed needed.

Our modeling focuses on the bilinear setting; the non-bilinear setting follows directly by restricting the adversary. We consider *$(\mathcal{EF}, \mathcal{DF})$-AGMOS adversaries (AGM with oblivious sampling)* $\mathcal{A}$ that obtain group elements as inputs and output field or group elements. Here, $\mathcal{EF}$ is a family of encoding functions, and $\mathcal{DF}$ is a family of distributions. We allow $\mathcal{A}$ to use an oracle $\mathcal{O}$ to obliviously sample elements from $\mathbb{G}_1$ or $\mathbb{G}_2$. $\mathcal{A}$ inputs to $\mathcal{O}$ adversarially chosen $E \in \mathcal{EF}$ and $D \in \mathcal{DF}$. $\mathcal{O}$ samples a random $s \leftarrow\$ D$ and then outputs $E(s)$ and s. The model executes $\mathcal{O}$ correctly and honestly (it is non-programmable) without leaking any side information. With any group element $[z]_\kappa$, $\mathcal{A}$ must output a linear representation with respect to all previously seen group elements, including oracle answers from the same group. Reasonable choices for $\mathcal{EF}$ may be admissible encodings or some other oblivious sampling functions. We also describe a simpler version of AGMOS where oracles respond with uniformly random group elements and discuss the implications of that.

To tackle the issue of reductions to PDL, we define a new family of falsifiable assumptions, $(\mathcal{EF}, \mathcal{DF})$-TOFR (Tensor Oracle FindRep). We prove that $(\mathcal{EF}, \mathcal{DF})$-TOFR holds in a version of GGM augmented by a distribution oracle $\mathcal{O}$ (all previous GGMH variants consider uniform distributions), under the assumption that $\mathcal{DF}$ contains well-spread distributions (i.e., distributions with min-entropy $\omega(\log \lambda)$, where λ is the security parameter).

We prove that (a version of) the Flexible Uber assumption [BBG05,Boy08], the Power Knowledge of Exponent (PKE) assumption [DFGK14], EUF-CMA of Schnorr's signature [Sch91], and the extractability of the KZG polynomial commitment scheme [KZG10] hold in the AGMOS. Crucially, in the last case, extractability is only possible when the committer additionally opens the polynomial at some point. Immediate extraction from the commitment is intractable, as we discussed earlier. In typical AGMOS proofs, we construct two reductions: one to the new TOFR assumption and the second one to the PDL assumption. To simplify PDL reductions, we define an intermediate assumption FPR (*Find Polynomial Representation*) and reduce it to PDL. The FPR assumption hides many of the complexities of typical PDL reductions and can also be used in standard AGM proofs.

Let TotalKE be the assumption family stating that an adversary, whose input is $([1]_1, [1]_2)$, knows the discrete logarithms of all output group elements. Most of such assumptions are insecure in the standard model due to oblivious sampling (if DL holds), while they hold in the AGM. We show that such spurious assumptions do not hold in AGMOS, under the hardness of DL, obtaining a (conditional) separation with AGM.

1.1 Technical Overview

Feasibility of Oblivious Sampling. We give a more detailed proof of a claim from [Ica09] that computing the DL of a group element $G = E(s)$, given G and s, where E is an admissible encoding and s is a random input, is roughly as hard as computing the discrete logarithm of a random group element. This implies that adversaries in the elliptic curve setting can indeed sample group elements obliviously (i.e., without knowing their DLs).

Definition of AGMOS. Suppose $\mathsf{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$ is a concrete bilinear group, $\kappa \in \{1, 2\}$, and $\mathbb{F} = \mathbb{Z}_p$. Let $\mathcal{EF} = \{\mathcal{EF}_{\mathsf{p},\kappa}\}_{\mathsf{p},\kappa}$ be a set of encodings (e.g., admissible encodings), with $\mathcal{EF}_{\mathsf{p},\kappa}$ containing encodings from $\mathbb{F}$ to $\mathbb{G}_\kappa$. Let $\mathcal{DF} = \{\mathcal{DF}_{\mathsf{p}}\}_{\mathsf{p}}$ be a family of distributions over $\mathbb{F}$.

We allow $(\mathcal{EF}, \mathcal{DF})$-*AGMOS adversaries* to query (p-dependent) *non-programmable* oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Given adversarially chosen E and D as inputs, $\mathcal{O}_\kappa(E, D)$ is defined as follows: if $E \notin \mathcal{EF}_{\mathsf{p},\kappa}$ or $D \notin \mathcal{DF}_{\mathsf{p}}$, it aborts. Otherwise, it samples $s \leftarrow\$ D$, computes $[\mathbb{q}]_\kappa \leftarrow E(s)$, and returns $[\mathbb{q}]_\kappa$ and s.

We require that for every non-uniform probabilistic polynomial time (PPT) $\mathcal{A}$, there exists a non-uniform PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, such that: if the adversary returns a group element $[\mathbb{y}]_\kappa$, $\mathsf{Ext}_{\mathcal{A}}$ returns with an overwhelming probability a linear representation $(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa)$ of $[\mathbb{y}]_\kappa$ with respect to the already seen group elements (including oracle answers) from $\mathbb{G}_\kappa$. More precisely, for $\kappa \in \{1, 2\}$, $\mathsf{Ext}_{\mathcal{A}}$ extracts vectors $\boldsymbol{\gamma}_\kappa$ and $\boldsymbol{\delta}_\kappa$ and a vector of oracle answers $[\mathbb{q}_\kappa]_\kappa$, such that $\mathbb{y}_\kappa = \boldsymbol{\gamma}_\kappa^\intercal \mathbb{x}_\kappa + \boldsymbol{\delta}_\kappa^\intercal \mathbb{q}_\kappa$.

Security proofs in AGMOS. Security proofs in the AGMOS follow the general strategy of security proofs in the AGM but with some crucial differences. Since the adversary's output $\mathbb{y} = \boldsymbol{\gamma}^\intercal \mathbb{x}_\kappa + \boldsymbol{\delta}^\intercal \mathbb{q}_\kappa$ depends on the oracle answers, the usual PDL reduction strategy is not sufficient. In the AGM, the polynomial $V(\boldsymbol{X})$ corresponding to the assumption's verification (a pairing-product equation) depends only on the challenger's trapdoors (e.g., a and b in the CDH assumption). In the case of several trapdoors, the PDL reduction embeds its input $[x]_\kappa$ to all trapdoors. Since we want to avoid FPRO, in the AGMOS, the reduction cannot embed $[x]_\kappa$ to the oracle answers.

Due to that, the AGMOS proof strategy looks as follows. We work with a verification polynomial $V(\boldsymbol{X}, \mathbb{Q})$, where $\mathbb{Q}_{\kappa i}$ is an indeterminate corresponding to $[\mathbb{q}_{\kappa i}]_\kappa$ (the ith answer of $\mathcal{O}_\kappa$). This polynomial is such that $V(\boldsymbol{x}, \mathbb{q})$ is equal to 0 iff the challenger accepts the adversary's output. Note that the actual verification equation, used in the definition of the assumption, is a function of the adversary's inputs and outputs. However, since the outputs have all the form $\mathbb{y} = \boldsymbol{\gamma}^\intercal \mathbb{x}_\kappa + \boldsymbol{\delta}^\intercal \mathbb{q}_\kappa$, V can be written as a polynomial in $(\boldsymbol{X}, \mathbb{Q})$. Importantly, V's coefficients can be computed from the internal variables of the challenger and the elements extracted by the AGMOS extractor.

In the AGM proof of a computational assumption, one considers two cases. In Case A of an AGM proof, $V(\boldsymbol{X}) = 0$ as a polynomial. One typically shows that this case never materializes. In Case X of an AGM proof, $V(\boldsymbol{X}) \neq 0$ as a polynomial but $V(\boldsymbol{x}) = 0$. One then constructs a PDL reduction that embeds the challenge (given as a tuple of group elements) to $\boldsymbol{x}$ (given as group elements), obtaining a univariate polynomial $V^*(X)$. The reduction uses polynomial factorization to find $V^*(X)$'s roots. One of these roots is necessarily the discrete logarithm of the reduction's input.

An AGMOS proof strategy of computational assumptions is more complicated. In Case A of an AGMOS proof, $V(\boldsymbol{X}, \mathbb{Q}) = 0$ as a polynomial. However, V is generally more complicated than in an AGM proof, and thus one has to be more careful when showing that $V(\boldsymbol{X}, \mathbb{Q}) = 0$ is impossible. Later, we use this difference to separate AGM and AGMOS.

Assuming $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$, an AGMOS proof has more cases. Case X of an AGMOS proof corresponds to the case where we can construct a PDL adversary. Due to how the sampling oracle's answers are created, one can write $V(\boldsymbol{X}, \mathbb{Q}) = V^h(\boldsymbol{X}) + V^t(\boldsymbol{X}, \mathbb{Q})$, where V^h does not depend on $\mathbb{Q}$ while each term of V^t depends on some indeterminate $\mathbb{Q}_{\kappa i}$. In Case X, $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$ as a polynomial but $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$ as a polynomial. We divide Case X into two subcases. In Subcase $\mathsf{X}.1$, the adversary does not use oracle answers, which means that $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ as a polynomial ($\boldsymbol{\delta} = \mathbf{0}$ for all group elements output by the adversary). Thus, the non-zero polynomial $V(\boldsymbol{X}, \mathbb{Q})$ does not depend on $\mathbb{Q}$. As in the AGM, we reduce the security of the proved assumption to the PDL.

In Subcase X.2, $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$ but $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$. In this case, the coefficient of some $\mathbb{Q}_{\kappa i}$ or $\mathbb{Q}_{1i}\mathbb{Q}_{2j}$ is a non-zero polynomial in $\boldsymbol{X}$ that evaluates to zero at $\boldsymbol{X} = \boldsymbol{x}$. As in Subcase X.1, we can reduce the security to the PDL by using polynomial factorization to return PDL's input, but the reduction is different.

Case Q of an AGMOS proof is the remaining case when Cases A and X do not hold. That is, $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$ (but the verifier accepts, $V(\boldsymbol{x}, \mathbb{q}) = 0$). Now we are in a situation where $V^t \neq 0$ as a polynomial (thus, V depends nontrivially on at least one $\mathbb{Q}_{\kappa i}$) but $V(\boldsymbol{x}, \mathbb{q}) = V^h(\boldsymbol{x}) + V^t(\boldsymbol{x}, \mathbb{q}) = 0$ for $[\mathbb{q}_{\kappa i}]_\kappa$ chosen from some distribution from $\mathcal{DF}_\mathsf{p}$.

For a concrete V, the probability that $V(x, \mathbb{q}) = 0$ is negligible over the choice of $\mathbb{q}$. However, V (whose coefficients depend on the linear representations) is fixed after the adversary $\mathcal{A}$ sees the oracle's answers. Since, for any $\mathbb{q}$, one can choose a bad V so that $V(\boldsymbol{x}, \mathbb{q}) = 0$, this probabilistic argument does not work. Fortunately, $\mathcal{A}$ only knows $[\mathbb{q}_{\kappa i}]_\kappa$ as group elements. It seems reasonable to *assume* that for any adversarial input (chosen by the reduction) and for $[\mathbb{q}]_\kappa$ coming from a well-spread distribution (in particular, $\mathbb{q}$ is non-zero with an overwhelming probability), it is difficult to construct a low-degree polynomial in $\mathbb{Q}$ that evaluates to zero at the oracle answers. In a few paragraphs, we formulate this as a new assumption, TOFR. In Case Q, we construct a reduction to TOFR.

To summarize, an AGMOS proof of a computational assumption has the following structure:

- Case A: $V(\boldsymbol{X}, \mathbb{Q}) = 0$. This case is typically impossible.
- Case X: $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$ and $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$.
  - Case X.1: $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$. Reduces to PDL.
  - Case X.2: $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$. Reduces (differently) to PDL.
- Case Q: $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$ and $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$. Reduces to TOFR.

FPR assumption. To automatize PDL reductions in AGMOS proofs, we define a new intermediate assumption Find Polynomial Representation (FPR). FPR is a tautological assumption of Case X (both subcases). FPR states that it is hard to find a non-zero multivariate polynomial f that evaluates to zero at the given input trapdoor. We first reduce FPR to PDL (without using idealized models). We define two variants of FPR that have incomparable reductions to PDL; the best choice depends on the context. The actual security reductions of Case X to FPR are straightforward. The definition of FPR is an independent contribution applicable to both AGM and AGMOS proofs.

Handling Knowledge Assumptions. The AGMOS proofs of knowledge assumptions follows the above blueprint for AGMOS proofs of computational assumptions. In particular, there will be Case A, Case X, and Case Q. Case X and Case Q are similar to the case of computational assumptions. (Although, as we will see, case X may not be needed in some knowledge assumption reductions.) Depending on the case, we construct a reduction to either FPR or TOFR. Recall that Case A did not materialize in the case of computational assumptions. In the case of knowledge assumptions, in an AGMOS proof, we construct a knowledge assumption extractor Ext that uses the AGMOS extractor $\overline{\mathsf{Ext}}$ as a subroutine. We show that if $\overline{\mathsf{Ext}}$ succeeds, so does Ext.

The new assumption TOFR. The new $(\mathcal{EF}, \mathcal{DF})$-TOFR (*Tensor Oracle Find-Rep*, see Definition 2) assumption states the following. Given oracle access to the sampling oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ (that are defined w.r.t. some $(\mathcal{EF}, \mathcal{DF})$ as before), it is intractable to output a vector $\boldsymbol{v} \neq \mathbf{0}$ such that $(1 \| \mathbb{q}_1^\intercal \| \mathbb{q}_2^\intercal \| (\mathbb{q}_1 \otimes \mathbb{q}_2)^\intercal) \cdot \boldsymbol{v} = 0$. Here, $[\mathbb{q}_\kappa]_\kappa$ is the vector of $\mathcal{O}_\kappa$ answers.

TOFR generalizes the classical FindRep [Bra94] and KerMDH assumptions [MRV16]. Recall that FindRep assumes that given a uniformly random vector of group elements $[\mathbb{x}]_\kappa$, it is difficult to find a non-zero vector $\boldsymbol{v}$, such that $\boldsymbol{v}^\intercal \mathbb{x} = 0$. FindRep is tightly secure under the discrete logarithm assumption.

Our AGMOS reductions work for any family $(\mathcal{EF}, \mathcal{DF})$ for which $(\mathcal{EF}, \mathcal{DF})$-TOFR is secure, but clearly, TOFR itself is not secure for any $(\mathcal{EF}, \mathcal{DF})$. For example, it is trivial to break TOFR if the encoding function is a constant function that maps any input to $[1]_\kappa$. The adversary can easily output $\boldsymbol{v} = (1, \dots, 1, -(\ell - 1))$, where ℓ is the length of the vector $(1 \| \mathbb{q}_1^\intercal \| \mathbb{q}_2^\intercal \| (\mathbb{q}_1 \otimes \mathbb{q}_2)^\intercal)$.

However, when $(\mathcal{EF}, \mathcal{DF})$ implements oblivious sampling (adversary does not know DL of $E(s)$ for $s \leftarrow\$\, D$), we expect $(\mathcal{EF}, \mathcal{DF})$-TOFR to hold since it is similar to the FindRep assumption. Such is the case with admissible encodings.

We provide further confidence to this claim in Section 8. We first define GGM with oblivious sampling (GGMOS), a novel version of GGM where the generic adversary has (in addition to the regular operations) access to an oblivious sampling oracle. The oracle takes as an input a distribution $D \in \mathcal{OF}_\mathsf{p}$ over $\mathbb{F}$ and returns a GGM label of x sampled from D. Here D can be seen as distribution over $\mathbb{F}$, induced by $E(D')$ where $E \in \mathcal{EF}_{\mathsf{p},\kappa}$ and $D' \in \mathcal{DF}_\mathsf{p}$. Note that modeling $\mathcal{EF}_{\mathsf{p},\kappa}$ itself is not possible in GGMOS since it may depend on the concrete structure of the group (see Section 8 for further discussion).

We prove that $(\mathcal{EF}, \mathcal{DF})$-TOFR is secure in GGMOS, assuming that all distributions from $\mathcal{DF}_\mathsf{p}$ have min-entropy $\omega(\log \lambda)$, i.e., are well-spread. Thus, the strength of TOFR depends crucially on $\mathcal{DF}$. However, this proof should only be taken as implying a necessary requirement for $\mathcal{DF}$ since $\mathcal{EF}$ cannot be entirely accurately modeled in GGMOS, as mentioned above.

While GGMOS is an interesting notion by itself, due to the lack of reductions (to FPR and TOFR), it is considerably simpler to model GGM than AGM with oblivious sampling. From the three novel aspects of AGMOS proofs (modified Case A, Subcase X.2, and the new Case Q), GGMOS proofs only need to deal with the first one. Note also that GGMOS is similar to the existing GGMH [Bro01,BFS16,ALSZ21]; however, GGMH only models adversarial access to uniformly random sampling.

Example AGMOS proofs. In Section 5 and Appendix C, we present a few example AGMOS security proofs. We picked our examples such that they showcase a variety of different aspects of AGMOS. We prove that a variant of the Flexible Uber assumption [BBG05,Boy08] is secure under FPR and TOFR (see Section 5.1 for why we chose a variant). This proof follows closely the general proof strategy mentioned above. We also prove that the (bilinear) Power Knowledge of Exponent (PKE) assumption [DFGK14] is secure under TOFR. As with many knowledge assumptions, this AGMOS proof does not need to rely on FPR or PDL. Intuitively this is because knowledge assumptions (typically) do not require that computing something is hard. We prove that the KZG polynomial commitment scheme [KZG10] is extractable under FPR and TOFR. Crucially, extractability is only possible when the committer opens the polynomial at some point. Extractability based only on the commitment corresponds to SpurKE, as discussed earlier. Here, we see a combination of an extractability property and a computational hardness property (binding), which is why both assumptions are needed. Finally, we show that Schnorr's signature [Sch91] is tightly EUF-CMA secure under DL and TOFR. This one is mainly interesting because it shows another advantage of AGM/AGMOS, the ability to prove tight reductions for protocols, which are not known to be tightly secure in the standard or RO model.

Separating AGM and AGMOS. As demonstrated with SpurKE, unconditional security of knowledge assumptions in the AGM does not imply the same in the AGMOS (or in the standard model). In Section 6, we consider the TotalKE assumption that states that an adversary $\mathcal{A}$ on input $([1]_1, [1]_2)$ must know the discrete logarithm of each of its output group elements. We prove that if $\mathcal{A}$ outputs more than R elements either in $\mathbb{G}_1$ or $\mathbb{G}_2$, where R is the number of distinct (pairing-product) verification equations that define the assumption, then $\mathcal{A}$'s some output element must depend non-trivially on some $\mathbb{q}_{\kappa i}$. Thus, under the DL assumption, $\mathcal{A}$ does not know its discrete logarithm.

Interestingly, this result uses the Chevalley-Warning theorem [Che35,War35] on the number of roots a low-degree multivariate polynomial can have over finite fields. To our knowledge, this is the first use of the Chevalley-Warning theorem to prove impossibility results in the pairing-based setting. We hope our result inspires further use of this theorem in (pairing-based) cryptography.

When $R = 1$, we give a characterization of all TotalKE assumptions that can be proven secure in groups where the DL assumption holds. Since all TotalKE assumptions hold in the AGM, this separates the AGM and the AGMOS in all but a small number of TotalKE cases. Here, separation means the following: in the AGM, most of the TotalKE assumptions hold independently of the DL, while in the AGMOS, these assumptions do not hold if the DL holds.

In the GGMH and GGMOS, one is only concerned about Case A (Case X and Case Q cannot appear). However, Case A is handled similarly in GGMOS and AGMOS. Hence, we also obtain a separation between GGMH and GGMOS.

AGMOS with Uniform Oracle. We also present a more simplified version of AGMOS where the oracle responds with uniformly random group elements. This model is more restrictive (for example, admissible encodings do not produce uniform outputs) but has other benefits. In particular, it relies on a weaker version of TOFR. We prove that if we did allow programming, then that version of TOFR would be implied by the PDL assumption.

More Related Work. Rotem and Segev [RS20] formalized algebraic adversaries for decisional problems. We only focus on computational and knowledge problems in this work. Recently, Zhandry and Zhang et al. [Zha22,ZZK22] have shown un-instantiability results for AGM. Although significant issues, we are not trying to solve these problems in the current work.

Many more works combine RO and AGM [FPS20,KMSV21,GT21] to model idealized hash functions. However, the point of those works is not to strengthen AGM with oblivious sampling.

[Lip19] was an early eprint that was never published. It contained a few mistakes. In particular, it considered an initial variant of AGMOS, but without the "TOFR" case. As such, it was incomplete and has been since withdrawn. [Lip22] covers some of the results of [Lip19] (like a new variant of Groth16 SNARK and its proof in [Lip22] AGM model), but importantly it did not consider arbitrary oblivious sampling distributions and encodings. [Lip22] got over the missing TOFR case by using the FPRO. The current paper corrects and improves on another set of techniques from [Lip19]. It also does not use FPRO.

2 Preliminaries

Let $\mathbb{F} = \mathbb{Z}_p$. Vectors are, by default, column vectors. We write $(\boldsymbol{a} // \boldsymbol{b})$ to show concatenation of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. For a matrix $\boldsymbol{A}$, $\boldsymbol{A}_i$ denotes its ith row and $\boldsymbol{A}^{(j)}$ denotes its jth column. Let $\mathbf{0}_n$ be a zero vector of length n. $\mathbb{F}^{(\leq d)}[X_1, \dots, X_k]$ denotes the set of k-variate polynomials of total degree $\leq d$ over $\mathbb{F}$. For $f \in \mathbb{F}[X_1, \dots, X_m]$, $\deg(f)$ denotes the total degree of f and $\deg_{X_i}(f)$ denotes the individual degree of X_i in f. PPT denotes probabilistic polynomial-time; $\lambda \in \mathbb{N}$ is the security parameter. Let $\mathsf{negl}(\lambda)$ be an arbitrary negligible function and $\mathsf{poly}(\lambda)$ be an arbitrary polynomial function. A probability is overwhelming if it is greater $1 - \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$. A random variable X has min-entropy k, denoted $\mathbf{H}_\infty(X) = k$, if $\max_x \Pr[X = x] = 2^{-k}$. A distribution is *well-spread* if it has super-logarithmic min-entropy, $\mathbf{H}_\infty(X) = \omega(\log \lambda)$; that is, $\max_x \Pr[X = x] = 2^{-\omega(\log \lambda)} = \lambda^{-\omega(1)} = \mathsf{negl}(\lambda)$. For an algorithm $\mathcal{A}$, $\mathrm{Im}(\mathcal{A})$ is the image of $\mathcal{A}$, i.e., the set of valid outputs of $\mathcal{A}$. $\mathsf{RND}_\lambda(\mathcal{A})$ denotes the random tape of $\mathcal{A}$ (for given λ), and $r \leftarrow\$ \mathsf{RND}_\lambda(\mathcal{A})$ denotes the uniformly random choice of r from $\mathsf{RND}_\lambda(\mathcal{A})$. By $y \leftarrow \mathcal{A}(\mathbb{x}; r)$ we denote the fact that $\mathcal{A}$, given an input $\mathbb{x}$ and a randomizer r, outputs y. Let $[1, n]$ denote $\{1, 2, \dots, n\}$.

Bilinear Groups. A bilinear group generator $\mathsf{Pgen}(1^\lambda)$ returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three additive cyclic (thus, abelian) groups of prime order p, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear pairing. Recall $\mathbb{F} = \mathbb{Z}_p$. The bilinear pairing is Type-3 (there is no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$). We use the standard additive bracket notation, writing $[a]_\kappa$ to denote $a g_\kappa$ where $g_\kappa = [1]_\kappa$ is a fixed generator of $\mathbb{G}_\kappa$, $\kappa \in \{1, 2, T\}$. We denote $\hat{e}([a]_1, [b]_2)$ by $[a]_1 \bullet [b]_2$. We use the bracket notation together with matrix notation, e.g., $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{C}$ iff $[\boldsymbol{A}]_1 \bullet [\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$.

Let $d_1(\lambda), d_2(\lambda) \in \mathsf{poly}(\lambda)$. Pgen is $(d_1(\lambda), d_2(\lambda))$-PDL *(Power Discrete Logarithm [Lip12]) secure* if for any λ and non-uniform PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{pdl}}_{d_1,d_2,\mathsf{Pgen},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\mathcal{A}(\mathsf{p}, [(x^i)_{i=0}^{d_1}]_1, [(x^i)_{i=0}^{d_2}]_2) = x \mid \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), x \leftarrow\!\!\$\, \mathbb{F}\right] = \mathsf{negl}(\lambda) \enspace .$$

Algebraic Group Model. AGM [FKL18] is a recent idealized model of computation. Essentially, in the AGM, one assumes that each non-uniform PPT algorithm $\mathcal{A}$ is algebraic in the following sense. Assume $\mathcal{A}$'s input includes $\mathbb{x}_\kappa = [\boldsymbol{x}_\kappa]_\kappa$ and no other elements from the group $\mathbb{G}_\kappa$. We assume that if $\mathcal{A}$ outputs a vector $[\mathbb{y}_\kappa]_\kappa$ of group elements, then $\mathcal{A}$ knows a matrix $\boldsymbol{\gamma}_\kappa$, such that $\mathbb{y}_\kappa = \boldsymbol{\gamma}_\kappa^\intercal \mathbb{x}_\kappa$. Note that the underlying protocol can be interactive. In such a case, the outputs of earlier rounds cannot depend on the inputs of the later rounds. One can formalize this by requiring specific entries of $\boldsymbol{\gamma}_\kappa$ are zero.

Fix Pgen. More precisely, a non-uniform PPT algorithm $\mathcal{A}$ is *algebraic* if there exists a non-uniform PPT extractor $\mathsf{Ext}_\mathcal{A}$, such that for any vector of group elements $\mathbb{x} = ([\mathbb{x}_1]_1, [\mathbb{x}_2]_2)$, $\mathsf{Adv}^{\mathrm{agm}}_{\mathsf{Pgen},\mathcal{A},\mathsf{Ext}_\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{c|l} \mathbb{y}_1 \neq \boldsymbol{\gamma}_1^\intercal \mathbb{x}_1 \vee & \mathsf{p} \leftarrow\!\!\$\, \mathsf{Pgen}(1^\lambda); r \leftarrow\!\!\$\, \mathsf{RND}_\lambda(\mathcal{A}); \\ \mathbb{y}_2 \neq \boldsymbol{\gamma}_2^\intercal \mathbb{x}_2 & ([\mathbb{y}_1]_1, [\mathbb{y}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, \mathbb{x}; r); (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) \leftarrow \mathsf{Ext}_\mathcal{A}(\mathbb{x}; r) \end{array}\right] = \mathsf{negl}(\lambda) \enspace .$$

2.1 Admissible Encodings

A map $E : \mathcal{S} \to \mathcal{R}$ between finite sets is an *admissible encoding* [BCI+10] iff

Computable: E is PPT computable,

ϱ-regular: for any $y \in \mathcal{R}$, the preimage size $|E^{-1}(y)|$ of y under E is $\leq \varrho$ for a small constant ϱ ($\varrho = 4$ in [Ica09]),

Sampleable: given y in the image of E, one can efficiently compute its full preimage $E^{-1}(y)$.

Boneh and Franklin [BF01] defined admissibility slightly differently; we follow the definition of [BCI+10]. Many admissible encodings are known. In Appendix A.1, we describe two admissible encodings, one by Boneh and Franklin [BF01] and another one by Icart [Ica09].

Next, we prove a claim of Icart [Ica09] that computing the discrete logarithm of $E(s)$ is roughly as hard as computing the discrete logarithm of a uniformly random element of $\mathbb{G}$. (Icart [Ica09] only gave a proof sketch.) This result is significant since it shows that efficient oblivious sampling is possible in elliptic curve groups.

Another corollary of the sampleability is that one can efficiently recognize whether some $P \in \mathbb{G}_\kappa$ belongs to $\mathrm{Im}(E)$. Within the proof (the claim does not depend on it), we use the quantity $\psi_E := |\mathrm{Im}(E)|/|\mathbb{F}|$. As proven in [FT10], while Icart's admissible encoding has $\varrho = 4$, it has $\psi_E \approx 5/8 > 1/\varrho$.

Let D be a distribution over $\mathbb{G}_\kappa$ for some $\kappa \in \{1, 2\}$. We say that the discrete logarithm assumption over D holds in Pgen if, for any non-uniform PPT $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},D,\mathcal{A}}(\lambda) := \Pr\left[\mathcal{A}(\mathsf{p}, [x]_\kappa) = x \mid \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), [x]_\kappa \leftarrow\!\!\$\, D\right] = \mathsf{negl}(\lambda) \enspace .$$

In the following, $E(\mathbb{F})$ refers to the distribution $E(s)$ for $s \leftarrow\$ \mathbb{F}$ and $\mathbb{G}_\kappa$ is the uniform distribution over $\mathbb{G}_\kappa$.

Theorem 1. *Let $E : \mathbb{F} \to \mathbb{G}_\kappa$ be a ϱ-regular admissible encoding. For any non-uniform PPT $\mathcal{A}$, there exists a non-uniform PPT $\mathcal{C}$ such that* $\mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},E(\mathbb{F}),\mathcal{A}}(\lambda) \le \frac{\varrho|\mathbb{G}_\kappa|}{|\mathbb{F}|} \cdot \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathbb{G}_\kappa,\mathcal{C}}(\lambda)$.

Before proving Theorem 1, we state and prove the following technical lemma. Recall, $\psi_E := |\mathrm{Im}(E)|/|\mathbb{F}|$. Let $\mathrm{Im}(E)$ refers to the uniform distribution over $\mathrm{Im}(E)$.

Lemma 1. *Let $E : \mathbb{F} \to \mathbb{G}_\kappa$ be a ϱ-regular admissible encoding. For any non-uniform PPT $\mathcal{A}$, there exists a non-uniform PPT $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},E(\mathbb{F}),\mathcal{A}}(\lambda) \le \varrho\psi_E \cdot \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathrm{Im}(E),\mathcal{B}}(\lambda)$.

Proof (Lemma 1). Let $\mathcal{A}$ be a discrete logarithm adversary whose input is $[x]_\kappa = E(s)$ for $s \leftarrow\$ \mathbb{F}$. We construct a trivial adversary $\mathcal{B}$ that computes the discrete logarithm of $[x]_\kappa$ sampled from $\mathrm{Im}(E)$. $\mathcal{B}([x]_\kappa)$ just invokes $\mathcal{A}$ to compute $v \leftarrow \mathcal{A}([x]_\kappa)$ and then returns v. Let us analyze the success probability of $\mathcal{B}$.

For $i \in [1, \varrho]$, let $T_i = \{[x]_\kappa : |\{s \in \mathbb{F} : E(s) = [x]_\kappa\}| = i\}$ be the set of group elements in $\mathrm{Im}(E)$ with the preimage size i and $N_i := E^{-1}(T_i) = \{s \in \mathbb{F} : E(s) \in T_i\}$ be the set of field elements that E brings to T_i. Clearly, $|N_i| = i|T_i|$. Since $T_i = E(N_i)$, $U(T_i) = E(U(N_i))$. Since $\{T_1, \dots, T_\varrho\}$ is a partition of $\mathrm{Im}(E)$,

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathrm{Im}(E),\mathcal{B}}(\lambda) &= \Pr\left[\mathcal{A}([x]_\kappa) = x \mid [x]_\kappa \leftarrow\$ \mathrm{Im}(E)\right] \\
&= \textstyle\sum_{i=1}^{\varrho} \Pr\left[\mathcal{A}([x]_\kappa) = x \mid [x]_\kappa \leftarrow\$ T_i\right] \cdot \frac{|T_i|}{|\mathrm{Im}(E)|} \\
&= \textstyle\sum_{i=1}^{\varrho} \Pr\left[\mathcal{A}([x]_\kappa) = x \mid s \leftarrow\$ N_i; [x]_\kappa \leftarrow E(s)\right] \cdot \frac{|N_i|}{i|\mathrm{Im}(E)|} \\
&\ge \frac{1}{\varrho\psi_E} \cdot \textstyle\sum_{i=1}^{\varrho} \Pr\left[\mathcal{A}([x]_\kappa) = x \mid s \leftarrow\$ N_i; [x]_\kappa \leftarrow E(s)\right] \cdot \frac{|N_i|}{|\mathbb{F}|} \\
&= \frac{1}{\varrho\psi_E} \cdot \Pr\left[\mathcal{A}([x]_\kappa) = x \mid s \leftarrow\$ \mathbb{F}, [x]_\kappa \leftarrow E(s)\right] \\
&= \frac{1}{\varrho\psi_E} \cdot \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},E(\mathbb{F}),\mathcal{A}}(\lambda) \ . \qquad \square
\end{aligned}$$

Proof (of Theorem 1). Assume $\mathcal{B}$ is a non-uniform PPT adversary against the discrete logarithm problem for $P \leftarrow\$ \mathrm{Im}(E)$. We construct an adversary $\mathcal{C}$ against the standard discrete logarithm problem with uniformly random challenge $P \leftarrow\$ \mathbb{G}_\kappa$. $\mathcal{C}$ invokes $\mathcal{B}$ to compute the discrete logarithm v of P. $\mathcal{C}$ aborts if $\mathcal{B}$ does not succeed. Otherwise, $\mathcal{C}$ returns v. Clearly,

$$\mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathbb{G}_\kappa,\mathcal{C}}(\lambda) \ge \Pr\left[\mathcal{B}([x]_\kappa) = x \wedge [x]_\kappa \in \mathrm{Im}(E) \mid [x]_\kappa \leftarrow\$ \mathbb{G}_\kappa\right]$$

$$= \frac{|\mathrm{Im}(E)|}{|\mathbb{G}_\kappa|} \cdot \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathrm{Im}(E),\mathcal{B}}(\lambda) \ .$$

Combining this with Lemma 1, we get that for any non-uniform PPT $\mathcal{A}$, there exists a non-uniform PPT $\mathcal{B}$ and $\mathcal{C}$ such that $\mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},E(\mathbb{F}),\mathcal{A}}(\lambda) \le \varrho\psi_E \cdot \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathrm{Im}(E),\mathcal{B}}(\lambda) \le \frac{\varrho|\mathbb{G}|}{|\mathbb{F}|} \cdot \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\mathbb{G},\mathcal{C}}(\lambda)$. □

Note that the result of Theorem 1 is relevant when $(\varrho \cdot |\mathbb{G}|)/|\mathbb{F}| = \mathsf{poly}(\lambda)$. Moreover, since E is sampleable, $\mathcal{A}$ can use $E^{-1}([x]_\kappa)$ as the side information.

Note also that the elliptic curve hashing construction $E(h(s))$ from [Ica09] is an oblivious sampling function given that E is an admissible encoding and $h : \mathbb{F} \to \mathbb{F}$ is a random oracle. This is easy to see. Suppose $\mathcal{A}^h$ is a non-uniform PPT algorithm that given $(E(h(s)), s)$ for $s \leftarrow\$ \mathbb{F}$ can compute the DL of $E(s)$ with some probability ε. Additionally, $\mathcal{A}^h$ has access to a random oracle h. Then we can construct a non-uniform PPT $\mathcal{B}$ that, given $(E(s), s)$ as an input, outputs DL of $E(s)$ with the same probability ε. $\mathcal{B}$ samples a random $r \leftarrow\$ \mathbb{F}$ and a random function h such that $h(r) = s$. It then returns the output of $\mathcal{A}^h(E(s), r)$. Assuming that $\mathcal{B}$'s success probability is negligible, the success probability of $\mathcal{A}$ must also be negligible.

3 AGM with Oblivious Sampling

Next, we define AGMOS, a more realistic variant of AGMOS that gives the adversary oblivious sampling oracles that return group elements without revealing their discrete logarithm. We define AGMOS in the pairing-based setting. However, it can be restricted to a group-based setting or generalized to a multilinear-map-based setting.

Sampling oracles. Fix $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$. Let $\mathcal{EF}_{\mathsf{p},\kappa}$ be a set of (polynomially many) functions $\mathbb{F} \to \mathbb{G}_\kappa$. Let $\mathcal{DF}_{\mathsf{p}}$ be a family of distributions over $\mathbb{F}$. We introduce two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$, one for each group $\mathbb{G}_1$ and $\mathbb{G}_2$. To simplify notation, we denote $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2)$. The ith query (E, D) to $\mathcal{O}_\kappa$ consists of a function $E \in \mathcal{EF}_{\mathsf{p},\kappa}$ and a distribution $D \in \mathcal{DF}_{\mathsf{p}}$. The oracle samples a random field element $s_i \leftarrow\$ D$ and returns $[\mathbb{q}_{\kappa_i}]_\kappa \leftarrow E(s_i)$ and s_i.

We will denote the adversary's initial input (e.g., input from the challenger) in $\mathbb{G}_\kappa$ by $[\mathbb{x}_\kappa]_\kappa$. We assume $[\mathbb{x}_\kappa]_\kappa$ always includes $[1]_\kappa$. Let $\mathbb{x} = ([\mathbb{x}_1]_1, [\mathbb{x}_2]_2)$. In interactive protocols, $\mathbb{x}$ is updated sequentially (we will not formalize it). The adversary's view consists of all group elements that the adversary has seen up to the given moment. This includes the adversary's initial input, elements sent by other parties during the interaction, and oracle answers.

Definition. Let $\mathcal{O}$ be as above. We require that for any non-uniform PPT oracle adversary $\mathcal{A}^{\mathcal{O}}$, there exists a non-uniform PPT extractor $\mathsf{Ext}_{\mathcal{A}}^{\mathcal{O}}$, such that: if $\mathcal{A}^{\mathcal{O}}(\mathbb{x})$ outputs a vector of group elements $[\mathbb{y}]_\kappa$, on input $\mathbb{x} = ([\mathbb{x}_1]_1, [\mathbb{x}_2]_2)$, then with an overwhelming probability, $\mathsf{Ext}_{\mathcal{A}}^{\mathcal{O}}$ outputs field-element matrices $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, and $[\mathbb{q}_\kappa]_\kappa$ ($\mathcal{O}_\kappa$'s answer vector), such that

$$\mathbb{y} = \boldsymbol{\gamma}^\intercal \mathbb{x}_\kappa + \boldsymbol{\delta}^\intercal \mathbb{q}_\kappa \ . \tag{1}$$

Here, $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ have the natural restriction that outputted group elements should only depend on the current state (group elements, including oracle answers, seen thus far) and not on the future information.

```
$\mathcal{O}_\kappa(E, D)$
if $E \notin \mathcal{EF}_{\mathsf{p},\kappa} \vee D \notin \mathcal{DF}_{\mathsf{p}}$ then return $\perp$; fi
$s \leftarrow\!\!\$\; D; [\mathbb{q}]_\kappa \leftarrow E(s)$; return $([\mathbb{q}]_\kappa, s)$;
```

Fig. 1. The description of the oblivious sampling oracle $\mathcal{O}_\kappa$, where $\kappa \in \{1, 2\}$.

Definition 1 (AGMOS). *Let $\mathcal{EF} = \{\mathcal{EF}_{\mathsf{p},\kappa}\}$ be a collection of functions. Let $\mathcal{DF} = \{\mathcal{DF}_{\mathsf{p}}\}$ be a family of distributions. A non-uniform PPT algorithm $\mathcal{A}$ is an $(\mathcal{EF}, \mathcal{DF})$-*AGMOS adversary *for* Pgen *if there exists a non-uniform PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$*, such that for any $\mathbb{x} = (\mathbb{x}_1, \mathbb{x}_2)$,* $\mathsf{Adv}^{\mathrm{agmos}}_{\mathsf{Pgen},\mathcal{EF},\mathcal{DF},\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\lambda) :=$

$$\Pr\left[\begin{array}{c|l} \mathbb{y}_1 \neq \boldsymbol{\gamma}_1^\intercal \mathbb{x}_1 + \boldsymbol{\delta}_1^\intercal \mathbb{q}_1 \vee & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); \\ \mathbb{y}_2 \neq \boldsymbol{\gamma}_2^\intercal \mathbb{x}_2 + \boldsymbol{\delta}_2^\intercal \mathbb{q}_2 & ([\mathbb{y}_1]_1, [\mathbb{y}_2]_2) \leftarrow\!\!\$\; \mathcal{A}^{\mathcal{O}}(\mathsf{p}, \mathbb{x}; r); \\ & (\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2 \leftarrow \mathsf{Ext}_{\mathcal{A}}^{\mathcal{O}}(\mathsf{p}, \mathbb{x}; r) : \end{array}\right] = \mathsf{negl}(\lambda) \enspace .$$

$\mathcal{O}$ is the non-programmable oracle depicted in Fig. 1. Here, $[\mathbb{q}_\kappa]_\kappa$ is required to be the tuple of elements output by $\mathcal{O}_\kappa$. We denote by ql_κ the number of $\mathcal{O}_\kappa$ calls.

In Section 7, we present a simplified version of the model, where $\mathcal{O}_\kappa$ returns only uniformly random group elements.

Discussion. One can rewrite the whole framework in the usual AGM terminology, requiring that the adversary returns together with each group element an explanation of how it depends on the group elements seen thus far. This is a purely cosmetic choice.

In our proofs, we will only extract everything after the adversary has output its last group element, handling the adversary's outputs in $\mathbb{G}_\kappa$ as a vector $[\mathbb{y}_\kappa]_\kappa$. This allows for convenient matrix-vector notation. However, this choice is also purely cosmetic, and one can extract the explanations one by one.

We parameterize the AGMOS with arbitrary families $\mathcal{EF}$ and $\mathcal{DF}$. For the sake of brevity, we will often assume that the parameters are clear from the context. In modeling, we will stay agnostic to the concrete choice of the families $\mathcal{EF}$ and $\mathcal{DF}$, but as we will see, the security of the TOFR assumption will significantly depend on this choice. Functions that can be reasonably included in $\mathcal{EF}$ should satisfy two properties:

1. They should induce well-spread distributions from well-spread distributions. Thus, for each well-spread distribution D, the distribution defined by $E(s)$, where $s \leftarrow\!\!\$\; D$, should be well-spread.
2. It should be hard to compute the discrete logarithm of $E(s)$, knowing s, when s is sampled from a well-spread distribution.

See Section 8 for why it seems crucial to consider well-spread distributions. For example, if $\mathcal{EF}$ includes scalar multiplication map $[\cdot]_1 : s \mapsto [s]_1$, which does not satisfy 2, the intuition that any party cannot know the discrete logarithm of $E(s)$ does not work anymore. Thus, including $[\cdot]_1$ to $\mathcal{EF}$ will not result in a meaningful model.

Similarly to GGMH, one can consider oracles who do oblivious sampling directly by sampling from some distribution over $\mathbb{G}_\kappa$. Such a model is less realistic than using random seeds s and functions. In particular, our model allows the adversary to also learn the input of the encoding function. In particular, Theorem 1 motivates that the adversary may be able to do it without knowing their discrete logarithms. Admissible encodings are the only maps for which we are aware of concrete proofs (see Theorem 1) that relate the difficulty of computing the discrete logarithm from their image to the general discrete logarithm assumption.

3.1 Further Formalization

Next, we will introduce more notation. We assume that the adversary's input is $\mathbb{x}(\boldsymbol{x}) = (\mathbb{x}_1(\boldsymbol{x}), \mathbb{x}_2(\boldsymbol{x}))$, where $\boldsymbol{x}$ is a vector of trapdoors not known to the adversary. For simplicity, the input always includes group generators $([1]_1, [1]_2)$. Here, $\boldsymbol{x}$ includes both CRS trapdoors and the trapdoors generated by other parties (e.g., the challenger) during the protocol. Let $\boldsymbol{X}$ be a vector of indeterminates corresponding to trapdoors $\boldsymbol{x}$ and thus $\mathbb{x}(\boldsymbol{X})$ is a vector of polynomials (or possibly rational functions, though we will not analyze this case) in $\boldsymbol{X}$.

Let $\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$ be the vector of indeterminates corresponding to the concatenation of vectors of oracle outputs $[\mathbb{q}_1]_1$ and $[\mathbb{q}_2]_2$. As in Section 3, $[\mathbb{q}_{\kappa i}]_\kappa = E_{\kappa i}(s_{\kappa i})$ for $s_{\kappa i} \leftarrow\!\!\$\ D_{\kappa i}$, where $E_{\kappa i} \in \mathcal{EF}_{\mathsf{p},\kappa}$ and $D_{\kappa i} \in \mathcal{DF}_{\mathsf{p}}$. We denote the adversary's outputs in $\mathbb{G}_\kappa$ as a vector $\mathbb{y}_\kappa(\boldsymbol{x}, \mathbb{q}_\kappa)$ corresponding to a vector of polynomials $\mathbb{y}_\kappa(\boldsymbol{X}, \mathbb{Q}_\kappa)$.

In an AGMOS security proof, the proved assumption[5] is accompanied by one or more "verification polynomials" V_i. For example, on input $[1, x_1, x_2]_1$, the CDH-in-$\mathbb{G}_1$ adversary outputs $[\mathbb{y}]_1 = [\mathbb{y}(x_1, x_2, \mathbb{q}_1)]_1$, where (due to the definition of the AGMOS) $\mathbb{y}$ is an adversarially chosen polynomial. The adversary is successful if $V(x_1, x_2, \mathbb{y}) = 0$, where $V(X_1, X_2, \mathbb{y}) := X_1 X_2 - \mathbb{y}(X_1, X_2, \mathbb{Q}_1)$. In the general case, the adversary is successful if each verification polynomial evaluates to 0 at the concrete point $(\boldsymbol{x}, \mathbb{y})$, where $\mathbb{y}$ depends on $(\boldsymbol{x}, \mathbb{q})$.

Formalizing Verification Equations. In the pairing-based setting, assumptions are defined by one (or more) verification polynomial equation in the challenger input and the adversary's output. Let us call this explicit verification polynomial V^{expl}.

In AGMOS (as it was in AGM) the challenger, upon queried the knowledge extractor, checks the polynomial equation, defined by replacing adversary's outputs in V^{expl} with their linear representation. Since the adversary's outputs $\mathbb{y}_\kappa$ are affine functions in $\mathbb{Q}_\kappa$, each addend in the linear representation of $[\mathbb{y}_{\kappa i}]_\kappa$ in $\mathbb{G}_\kappa$ must depend on at most one oracle answer. Therefore, we can define the

[5] One can prove the security of a concrete assumption or a concrete primitive/protocol. We will call all things we prove in the AGMOS assumptions instead of each time saying "an assumption or the security of a protocol".

implicit verification polynomial as

$$V(\boldsymbol{X},\mathbb{Q}) := V^h(\boldsymbol{X}) + V^t(\boldsymbol{X},\mathbb{Q}) \text{ for } V^h(\boldsymbol{X}) := \boldsymbol{\gamma}^\intercal \mathbb{p}^h,\ \ V^t(\boldsymbol{X},\mathbb{Q}) := \boldsymbol{\delta}^\intercal \mathbb{p}^t \quad (2)$$

for vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ that can be computed from the outputs of the AGMOS extractor, and vectors $\mathbb{p}^h$ and $\mathbb{p}^t$ that depend on $\boldsymbol{X}$. The latter two equalities follow from Eq. (1).

Example 1 (CDH in $\mathbb{G}_1$). Recall that on input $[1, x_1, x_2]_1$, the CDH adversary outputs $[\mathbb{y}]_1 = [\mathbb{y}(x_1, x_2, \mathbb{q}_1)]_1$. The challenger checks if $[x_1]_1 \bullet x_2[1]_2 = [\mathbb{y}]_1 \bullet [1]_2$. Thus, the *explicit verification polynomial* is $V^{\mathsf{expl}}(\boldsymbol{X}, \mathbb{y}) = X_1 X_2 - \mathbb{y}$. Taking into account that $\mathbb{y} = \boldsymbol{\gamma}^\intercal \mathbb{x}_1(\boldsymbol{X}) + \boldsymbol{\delta}^\intercal \mathbb{Q}_1$ (and changing signs), we get that

$$V(\boldsymbol{X},\mathbb{Q}) = \gamma_1 + \gamma_2 X_1 + \gamma_3 X_2 - X_1 X_2 + \sum \delta_i \mathbb{Q}_{1i} \ .$$

Thus, $\mathbb{p}^h = (1, X_1, X_2, X_1X_2)^\intercal$, $\mathbb{p}^t = \mathbb{Q}_1$, $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3, -1)^\intercal$, and $\boldsymbol{\delta} = (\delta_1, \ldots)$.

Example 2 (KE). The *KE (knowledge-of-exponent) assumption* [Dam92] for Pgen in $\mathbb{G}_\kappa$ holds if for any non-uniform PPT $\mathcal{A}$, there exists a non-uniform PPT extractor $\mathsf{Ext}_\mathcal{A}$, such that $\mathsf{Adv}^{\mathsf{ke}}_{\mathsf{Pgen},\kappa,\mathcal{A},\mathsf{Ext}_\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} \mathbb{y}_2 = x\mathbb{y}_1 \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); x \leftarrow\!\!\$\, \mathbb{F}; r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); \\ \mathbb{y}_1^* \neq \mathbb{y}_1 & [\mathbb{y}_1, \mathbb{y}_2]_\kappa \leftarrow \mathcal{A}(\mathsf{p}, [x]_\kappa; r); \mathbb{y}_1^* \leftarrow \mathsf{Ext}_\mathcal{A}(\mathsf{p}, [x]_\kappa; r) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

The KE adversary outputs $\mathbb{y}_1 = \mathbb{y}_1(x, \mathbb{q}_1)$ and $\mathbb{y}_2 = \mathbb{y}_2(x, \mathbb{q}_1)$ and the extractor outputs vectors $\boldsymbol{\gamma}_i$ and $\boldsymbol{\delta}_i$, such that $\mathbb{y}_1(X, \mathbb{Q}_1) = \boldsymbol{\gamma}_1^\intercal \left(\begin{smallmatrix}1\\X\end{smallmatrix}\right) + \boldsymbol{\delta}_1^\intercal \mathbb{Q}_1 = \gamma_{11} + \gamma_{12}X + \sum_i \delta_{1i}\mathbb{Q}_{1i}$ and $\mathbb{y}_2(X, \mathbb{Q}_1) = \gamma_{21} + \gamma_{22}X + \sum_i \delta_{2i}\mathbb{Q}_{1i}$. Clearly, $V^{\mathsf{expl}}(X, \mathbb{y}_1, \mathbb{y}_2) = \mathbb{y}_2 - X\mathbb{y}_1$, while $V(X, \mathbb{Q}_1)$ can be expressed as follows:

$$\begin{aligned} V(X, \mathbb{Q}_1) =& \mathbb{y}_2(X, \mathbb{Q}_1) - X \cdot \mathbb{y}_1(X, \mathbb{Q}_1) \\ =& \gamma_{21} + (\gamma_{22} - \gamma_{11})X - \gamma_{12}X^2 + \boldsymbol{\delta}_2^\intercal \mathbb{Q}_1 - X\boldsymbol{\delta}_1^\intercal \mathbb{Q}_1 \ . \end{aligned}$$

Here, $V^h(X) = \gamma_{21} + (\gamma_{22} - \gamma_{11})X - \gamma_{12}X^2$ and $V^t(X, \mathbb{Q}_1) = \boldsymbol{\delta}_2^\intercal \mathbb{Q}_1 - X\boldsymbol{\delta}_1^\intercal \mathbb{Q}_1$.

Observe that $V^h(\boldsymbol{X})$ does not depend on $\mathbb{Q}$ while each term of $V^t(\boldsymbol{X}, \mathbb{Q})$ depends on either some indeterminate $\mathbb{Q}_{\kappa i}$ or some product $\mathbb{Q}_{1i}\mathbb{Q}_{2j}$. Since $\mathbb{Q}_{\kappa i}$ are indeterminates, it follows from $\boldsymbol{\delta} \neq \mathbf{0}$ that $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$ and hence $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$. If $\boldsymbol{\delta} = \mathbf{0}$, then the verification success does not depend on the answers of the oracles. In this case, one essentially has an AGM proof where one does not have to consider the additional details of AGMOS. On the other hand, if $\boldsymbol{\delta} \neq \mathbf{0}$ is non-zero, then the verification polynomial has at least one term, say $v\mathbb{Q}_{\kappa k}$ or $vX\mathbb{Q}_{\kappa k}$, with a non-zero coefficient v. This case is new to AGMOS and has to be analyzed separately.

4 New Assumptions

4.1 TOFR

According to Section 1.1, in Case Q of AGMOS proofs, we have to handle the case when $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$ but (since the verifier accepts) $V(\boldsymbol{x}, \mathbb{q}) = 0$. As outlined in Section 1.1, there are several differences between the AGM and the AGMOS proofs. However, only Case Q requires us to rely on a new assumption.

Intuitively, TOFR is a simplified version of the tautological assumption in Case Q of AGMOS proofs. The latter states that it is difficult to output the coefficients of a polynomial V (see Eq. (2)), such that $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$ but $V(\boldsymbol{x}, \mathbb{q}) = 0$. Recall that $V(\boldsymbol{X}, \mathbb{Q}) = \boldsymbol{\gamma}^\intercal \mathbb{x}^h + \boldsymbol{\delta}^\intercal \mathbb{x}^t$. In our AGMOS proofs, we let the TOFR reduction generate the input trapdoors $\boldsymbol{x}$. Thus, the input of a TOFR assumption is just $([1]_1, [1]_2)$; moreover, $\mathbb{x}^h = 1$, and γ is a single field element. Writing $\boldsymbol{v} := \binom{\gamma}{\boldsymbol{\delta}}$, by Eq. (2), $V^t(\boldsymbol{X}, \mathbb{Q}) = \boldsymbol{\delta}^\intercal \mathbb{x}^t \neq 0$ iff $\boldsymbol{\delta} \neq \boldsymbol{0}$ and $V(\mathbb{x}, \mathbb{q}) = 0$ iff $\boldsymbol{v}^\intercal \mathbb{x} = 0$. Here, $\mathbb{x}^t = \begin{pmatrix} \mathbb{q}_1 \\ \mathbb{q}_2 \\ \mathbb{q}_1 \otimes \mathbb{q}_2 \end{pmatrix}$, and $\mathbb{x} = \begin{pmatrix} \mathbb{x}^h \\ \mathbb{x}^t \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbb{x}^t \end{pmatrix}$. Clearly, $\boldsymbol{v}^\intercal \mathbb{x} = 0$ cannot hold if $\boldsymbol{\delta} = \boldsymbol{0}$ but $\gamma \neq 0$. Hence $\boldsymbol{\delta} \neq \boldsymbol{0}$ and $\boldsymbol{v}^\intercal \mathbb{x} = 0$ is equivalent to $\boldsymbol{v} \neq \boldsymbol{0}$ and $\boldsymbol{v}^\intercal \mathbb{x} = 0$. We get the following assumption.

Definition 2 (TOFR)**.** *Let $\mathcal{EF}$ be some family of function and $\mathcal{DF}$ a family of distributions. We say that* Pgen *is* $(\mathcal{EF}, \mathcal{DF})$-TOFR *(*Tensor Oracle FindRep*) secure if for any non-uniform PPT $\mathcal{A}$,* $\mathsf{Adv}^{\mathrm{tofr}}_{\mathsf{Pgen},\mathcal{EF},\mathcal{DF},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\boldsymbol{v} \neq \boldsymbol{0} \wedge \boldsymbol{v}^\intercal \cdot \begin{pmatrix} 1 \\ \mathbb{q}_1 \\ \mathbb{q}_2 \\ \mathbb{q}_1 \otimes \mathbb{q}_2 \end{pmatrix} = 0 \,\middle|\, \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{v} \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{p}) \right] = \mathsf{negl}(\lambda) \enspace .$$

Here, $\mathcal{O}$, $\mathbb{q}_1$, and $\mathbb{q}_2$ are as in Definition 1.

Discussion. In Section 8.2, we prove that TOFR is secure in a variant of GGM where the adversary can call an oracle that samples group elements from well-spread distributions, i.e., has more power compared to the GGM.

TOFR is related to the following well-known assumption. Let $d(\lambda) \in \mathsf{poly}(\lambda)$. Pgen is *d-FindRep (Find Representation, [Bra94]) secure in $\mathbb{G}_\kappa$* if for any non-uniform PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{findrep}}_{d,\mathsf{Pgen},\kappa,\mathcal{A}}(\lambda) := \Pr\left[\boldsymbol{v} \neq 0 \wedge \boldsymbol{v}^\intercal \boldsymbol{x} = 0 \mid \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{x} \leftarrow\!\!\$\, \mathbb{F}^d; \boldsymbol{v} \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{x}]_\kappa)\right] = \mathsf{negl}(\lambda)$.

FindRep can be tightly reduced to the discrete logarithm assumption. For the sake of completeness, we reprove this well-known result.

Lemma 2 ([Bra94]). *Let $d \geq 1$. For any non-uniform PPT $\mathcal{A}$, there exist a non-uniform PPT $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathrm{findrep}}_{d,\mathsf{Pgen},\kappa,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{dl}}_{\mathsf{Pgen},\kappa,\mathcal{B}}(\lambda) + 1/|\mathbb{F}|$.

Proof. The discrete logarithm adversary $\mathcal{B}$ embeds its challenge $[y]_\kappa$ to a FindRep challenge $[\boldsymbol{x}]_\kappa$ by sampling $\boldsymbol{r}, \boldsymbol{s} \leftarrow\!\!\$\, \mathbb{F}^d$ and then setting $[\boldsymbol{x}]_\kappa \leftarrow \boldsymbol{r}[1]_\kappa + \boldsymbol{s}[y]_\kappa$. If $\mathcal{A}$ succeeds, i.e., returns a non-zero representation $\boldsymbol{v}$, such that $\boldsymbol{v}^\intercal \boldsymbol{x} \neq 0$, then $\mathcal{B}$ returns $y' \leftarrow -\boldsymbol{v}^\intercal \boldsymbol{r} / \boldsymbol{v}^\intercal \boldsymbol{s}$. Note that $[\boldsymbol{x}]_\kappa$ is uniformly random over $\mathbb{G}_\kappa^d$ and $\boldsymbol{s}$ is independent of $\boldsymbol{x}$ and thus of $\boldsymbol{v}$. Thus, if $\boldsymbol{v}$ is non-zero, $\Pr[\boldsymbol{v}^\intercal \boldsymbol{s} = 0] = 1/|\mathbb{F}|$. Then, $0 = \boldsymbol{v}^\intercal \boldsymbol{x} = \boldsymbol{v}^\intercal(\boldsymbol{r} + \boldsymbol{s} y)$. Solving for y, we get $y = -\boldsymbol{v}^\intercal \boldsymbol{r} / \boldsymbol{v}^\intercal \boldsymbol{s}$. Thus, if $\mathcal{A}$ succeeds, then $\mathcal{B}$ works correctly, except with the probability $1/|\mathbb{F}|$. □

There are two essential differences between FindRep and TOFR. Firstly, instead of getting $[\mathbb{q}_{\kappa i}]_\kappa$ as inputs, a TOFR adversary $\mathcal{A}$ can query the oracle to obtain $[\mathbb{q}_{\kappa i}]_\kappa$ adaptively. Secondly, TOFR oracle is non-programmable. However, even if we ignore the second issue, the reduction to DL is non-obvious. We could modify the described reduction so that r_i and s_i are sampled on the fly. However, before each query, the adversary can adaptively choose a new distribution for the oracle answer. It is unclear how to choose r_i and s_i to make each $[\mathbb{q}_{\kappa i}]_\kappa$ to be from the correct distribution. We propose a more restrictive model in Section 7, where the oracle returns only uniformly random group elements. In that case, relying on a weaker version of TOFR is possible. We prove that if we allow programming, this weaker version of TOFR is equivalent to $(1,1)$-PDL.

4.2 FPR

Recall a typical step in an AGM/AGMOS security proof. According to Eq. (2), the verification polynomial has form $V(\boldsymbol{X}, \mathbb{Q}) = V^h(\boldsymbol{X}) + V^t(\boldsymbol{X}, \mathbb{Q})$. We know that $V(\boldsymbol{x}, \mathbb{q}) = 0$ for (uniformly random) $\boldsymbol{x}$ and oracle answers $\mathbb{q}$. In one of the proof branches (that we call Case X), we have $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$ but $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$. One then constructs a reduction to PDL. We automate this step by defining a tautological assumption for a typical Case X and then relate it to PDL.

Definition 3 (FPR)**.** *Let* $m \geq 1$*,* $d_1, d_2, d_T, d_g \geq 0$ *and* $\boldsymbol{d} = (d_1, d_2, d_T, d_g)$*. Let* $\boldsymbol{X} = (X_1, \ldots, X_m)$*. We say that* Pgen *is* $(\boldsymbol{d}, m)$-FPR (Find Polynomial Representation) secure *if for any non-uniform PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathrm{fpr}}_{\mathsf{Pgen}, \boldsymbol{d}, m, \mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} g(\boldsymbol{X}) \in \mathbb{F}^{(\leq d_g)}[X_1, \ldots, X_m] \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{x} \leftarrow\$\, \mathbb{F}^m; \\ g(\boldsymbol{X}) \neq 0 \wedge g(\boldsymbol{x}) = 0 & g(\boldsymbol{X}) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m}(\boldsymbol{x}, \cdot)}(\mathsf{p}) \end{array}\right] = \mathsf{negl}(\lambda) \enspace ,$$

where the oracle $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m}(\boldsymbol{x}, \cdot)$ *takes an input* (κ, f)*. If* $\kappa \in \{1, 2, T\}$ *and* $f \in \mathbb{F}[\boldsymbol{X}]$ *such that* $\deg_{X_i}(f) \leq d_\kappa$ *for all* i*, it returns* $[f(\boldsymbol{x})]_\kappa$*. Otherwise, it returns* $\perp$*. We will omit the subscript for simplicity and write* $\mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x}, \cdot)$*.*

We use techniques from [RS20,Rot22] to show that FPR reduces to PDL. Let us borrow some notation from [Rot22]. For a non-zero polynomial $f \in \mathbb{F}[X_1, \ldots, X_m]$, define $h_i \in \mathbb{F}[X_i, \ldots, X_m]$ as follows: (1) $h_1 = f$, (2) for $i \in [2, m]$: If $h_{i-1} = 0$, then $h_i := 0$. Otherwise, write $h_{i-1} = \sum_{j=0}^{d} g_j(X_i, \ldots X_m) X_{i-1}^j$ as a polynomial in $(\mathbb{F}[X_i, \ldots, X_m])[X_{i-1}]$; here, $d = \deg_{X_{i-1}} h_{i-1}$. Let j^* be the minimal index, such that g_{j^*} is a non-zero polynomial over $\mathbb{F}$. Define $h_i := g_{j^*}$. If no such index j^* exists, set $h_i = 0$. Define $\mathsf{seq}(f) := \{h_1, \ldots, h_m\}$.

Proposition 1 (Lemma 5.5 of [Rot22]). *Let* $f \in \mathbb{F}[X_1, \ldots, X_m]$ *be non-zero. Let* $\mathsf{seq}(f) = \{h_1, \ldots, h_m\}$ *be as above. Then: (1) For each* $i \in [1, m]$*,* $h_i \neq 0$*. (2) For each root* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m) \in \mathbb{F}^m$ *of* $f(\boldsymbol{X})$*, there exists* $i_0 \in [1, m]$ *such that* $v(X_{i_0}) := h_{i_0}(X_{i_0}, \alpha_{i_0+1}, \ldots, \alpha_m)$ *is a non-zero polynomial and* $v(\alpha_{i_0}) = 0$*.*

We reduce FPR to PDL in the standard model (not in the AGMOS).

Theorem 2. *Let* $m \geq 1$ *and* $d_1, d_2, d_g \geq 0$ *with* $d_g = \mathsf{poly}(\lambda)$*. Let* $\boldsymbol{d} = (d_1, d_2, d_1 + d_2, d_g)$*. If the* (d_1, d_2)-PDL *assumption holds, then the* $(\boldsymbol{d}, m)$-FPR *assumption holds.*

```
$\mathcal{B}(\mathsf{p}, [(x^i)_{i=0}^{d_1}]_1, [(x^i)_{i=0}^{d_2}]_2)$
$i^* \leftarrow\!\!\$\ [1, m]; \alpha_1, \dots, \alpha_{i^*-1}, \alpha_{i^*+1}, \dots, \alpha_m \leftarrow\!\!\$\ \mathbb{F}^{m-1}$;
Define implicitly $\alpha_{i^*} = x$;
$g(\boldsymbol{X}) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x}, \cdot)}(\mathsf{p})$;
Define $\mathsf{seq}(g) = \{h_1, \dots, h_m\}$;
$v(X) := h_{i^*}(X, \alpha_{i^*+1}, \dots, \alpha_m)$; if $v(X) = 0$ then return $\perp$; fi
Find the set of roots $S$ of $v(X)$;
return $s \in S$ such that either $s \cdot [1]_1 = [x]_1$ or $s \cdot [1]_2 = [x]_2$;
```

Fig. 2. The FPR reduction $\mathcal{B}$ to PDL assumption in Theorem 2.

Proof. Let $\mathcal{A}$ be a non-uniform PPT $(\boldsymbol{d}, m)$-FPR adversary. In Fig. 2, we depict a (d_1, d_2)-PDL adversary $\mathcal{B}$. $\mathcal{B}$ gets $(\mathsf{p}, [(x^i)_{i=0}^{d_1}]_1, [(x^i)_{i=0}^{d_2}]_2)$ as an input. $\mathcal{B}$ samples $\boldsymbol{\alpha} \leftarrow\!\!\$\ \mathbb{F}^m$, except that for a randomly chosen position i^*, it implicitly sets $\alpha_{i^*} \leftarrow x$. If $\mathcal{A}$ queries (κ, f) for some κ and f, $\mathcal{B}$ answers with

$$[f(\boldsymbol{x})]_\kappa \leftarrow \textstyle\sum_{i=0}^{d_i} f_i(\alpha_1, \dots, \alpha_{i^*-1}, \alpha_{i^*+1}, \dots, \alpha_m)[x^i]_\kappa \enspace ,$$

where f_i is defined by $f(\boldsymbol{X}) = \sum_{i=0}^{d_i} f_i(X_1, \dots, X_{i^*-1}, X_{i^*+1}, \dots, X_m)X_{i^*}^i$ and $d_i = \deg_{X_i} f$. In case of $\mathbb{G}_T$ queries, $\mathcal{B}$ can compute $[1, \dots, x^{d_1+d_2}]_T$ by pairing input elements.

If $\mathcal{A}$ succeeds, then g is a non-zero multivariate polynomial satisfying $g(\boldsymbol{x}) = 0$. By Proposition 1, there exists $i_0 \in [1, m]$ such that $v(X_{i_0}) := h_{i_0}(X_{i_0}, \alpha_{i_0+1}, \dots, \alpha_m)$ is a non-zero *univariate* polynomial. Suppose that $i_0 = i^*$, which happens with probability $1/m$. Using Proposition 1 again, $v(X)$ is a non-zero polynomial satisfying $v(x) = 0$. Thus, $\mathcal{B}$ succeeds in computing x. Here, $d_g = \mathsf{poly}(\lambda)$ since it might otherwise take superpolynomial time to find the roots. Thus, $\mathsf{Adv}^{\mathrm{fpr}}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{A}}(\lambda) \le m \cdot \mathsf{Adv}^{\mathrm{pdl}}_{d_1,d_2,\mathsf{Pgen},\mathcal{B}}(\lambda)$. □

Note that $(\boldsymbol{d}, m)$-FPR is secure even when $d_T > d_1 + d_2$. Let us denote $d_\Delta := d_T - (d_1 + d_2)$. Then $(\boldsymbol{d}, m)$-FPR reduces trivially to $(\boldsymbol{d}', m)$-FPR, where $\boldsymbol{d}' = (d_1 + d_\Delta, d_2, d_T, d_g)$. By Theorem 2, $(\boldsymbol{d}', m)$-FPR (and thus also $(\boldsymbol{d}, m)$-FPR) reduces to $(d_1 + d_\Delta, d_2)$-PDL.

In Appendix B.1, we construct another reduction to PDL, which sometimes gives a tighter reduction to PDL. See Appendix B.2 for their comparison.

5 Example AGMOS Security Proofs

For concreteness, we give four explicit security proofs in AGMOS. We will see that the proof strategy for computational assumptions (see Section 5.1) and knowledge-type assumptions (see Section 5.2) is different. We also give an extractability proof of KZG commitment scheme, which mixes both extractability and computational hardness proof types (see Section 5.3). In Appendix C, we present a tight reduction of Schnorr's signature scheme that includes the handling of additional oracles for signing and hashing.

In the rest of this section, we assume that $\mathcal{EF}$ is some family of encoding and $\mathcal{DF}$ is some family of distributions for which the $(\mathcal{EF}, \mathcal{DF})$-TOFR holds.

5.1 The Split Flexible Uber Assumption

The Flexible Uber assumption [BBG05,Boy08] is a family that covers many commonly used computational assumptions. Instead of proving that each such assumption is secure, proving the Flexible Uber assumption makes sense. However, a Flexible Uber adversary outputs a $\mathbb{G}_T$ element. Since $\mathbb{G}_T$ (a subgroup of the multiplicative group of a finite field) is not a generic group [JR10], we prefer not to handle adversaries who output $\mathbb{G}_T$ elements.[6] Instead, we prove the AGM security of a slightly weaker assumption, Split Flexible Uber.

For a vector of m-variate polynomials $\mathcal{R} = (f_1, \ldots, f_r)$ over $\mathbb{F}$ and $\boldsymbol{x} \in \mathbb{F}^m$, we denote $\mathcal{R}(\boldsymbol{x}) := (f_1(\boldsymbol{x}), \ldots, f_r(\boldsymbol{x}))$.

Definition 4 (Split Flexible Uber Assumption). *Let $m \geq 1$ be an integer and $\boldsymbol{X} = (X_1, \ldots, X_m)$. Let $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$. Let $\mathcal{R}_1 = (f_1, \ldots, f_{r_1})$, $\mathcal{R}_2 = (g_1, \ldots, g_{r_2})$, and $\mathcal{R}_T = (h_1, \ldots, h_{r_T})$ be three tuples of m-variate polynomials from $\mathbb{F}[\boldsymbol{X}]$, where $f_1 = g_1 = h_1 = 1$. The* $(\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_T, d_t)$-computational Split Uber assumption *for* Pgen*, states that for any non-uniform PPT adversary $\mathcal{A}$,* $\mathsf{Adv}^{\mathsf{sfuber}}_{\mathsf{Pgen},\mathcal{R}_1,\mathcal{R}_2,\mathcal{R}_T,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} t \in \mathbb{F}[\boldsymbol{X}] \wedge \deg t \leq d_t \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{x} \leftarrow\!\!\$\, \mathbb{F}^m; \\ t(\boldsymbol{X}) \notin \operatorname{span}\{f_i g_j\} \cup \{h_k\} \wedge & \mathsf{ck} \leftarrow ([\mathcal{R}_1(\boldsymbol{x})]_1, [\mathcal{R}_2(\boldsymbol{x})]_2, [\mathcal{R}_T(\boldsymbol{x})]_T); \\ [y_1]_1 \bullet [y_2]_2 = [t(\boldsymbol{x})]_T & (t, [y_1]_1, [y_2]_2) \leftarrow \mathcal{A}(\mathsf{ck}) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

We say t is *non-trivial* if $t \in \mathbb{F}[\boldsymbol{X}]$, $\deg t \leq d_t$, and $t(\boldsymbol{X}) \notin \operatorname{span}\{f_i g_j\} \cup \{h_k\}$.

In the Flexible Uber assumption, the adversary outputs $(t, [z]_T)$ and the requirement is that $[z]_T = [t(\boldsymbol{x})]_T$. Given $(t, [y_1]_1, [y_2]_2)$ output by a Split Flexible Uber adversary, one can construct a Flexible Uber adversary that outputs t together with $[z]_T \leftarrow [y_1]_1 \bullet [y_2]_2$. Thus, if one can break the Split Flexible Uber assumption, one can break the Flexible Uber assumption.

It is easy to see that the Split Flexible Uber assumption implies (among many other assumptions) the CDH assumption in $\mathbb{G}_1$. One sets $\mathcal{R}_1 = \{f_1, f_2\}$ where $f_1(X_1) = X_1$ and $f_2(X_2) = X_2$ (and $\mathcal{R}_2 = \mathcal{R}_T = \emptyset$). This means the adversary gets an input $[x_1, x_2]_1$ for $x_1, x_2 \leftarrow\!\!\$\, \mathbb{F}$. To break the CDH assumption, the adversary should output $[y]_1 = [x_1 x_2]_1$, which is computationally hard since otherwise, the uber adversary could output $[y]_1, [1]_2$ and $t(X_1, X_2) = X_1 \cdot X_2$.

Let us introduce some additional notation for Theorem 3. For $\kappa \in \{1, 2, T\}$, let d_κ be such that for any $f \in \mathcal{R}_\kappa$ and any $i \in [1, m]$, $\deg_{X_i}(f) \leq d_\kappa$. Let $\boldsymbol{d} = (d_1, d_2, d_T, d_g)$, where $d_g = \max(d_t, m \cdot (d_1 + d_2))$.

[6] One can extend AGMOS to allow arguing about adversarial outputs from $\mathbb{G}_T$, but it is just our preference not to do so. See [JR10] for a discussion.

Theorem 3. *If the* $(\boldsymbol{d}, m)$-FPR *and* $(\mathcal{EF}, \mathcal{DF})$-TOFR *assumptions hold, then the* $(\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_T, d_t)$*-computational Split Flexible Uber assumption holds in the AGMOS.*

Proof. Let $\mathcal{A}$ be a non-uniform PPT Split Flexible Uber assumption AGMOS adversary that with some non-negligible probability outputs t, $[y_1]_1$, and $[y_2]_2$, such that t is non-trivial and $y_1 y_2 = t(\boldsymbol{x})$. Since $\mathcal{A}$ is an AGMOS adversary, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$ that extracts $\boldsymbol{\gamma}, \boldsymbol{\delta}$, such that $[y_1]_1 = \boldsymbol{\gamma}_1^\intercal [\boldsymbol{f}(\boldsymbol{x})]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}_1]_1$ and $[y_2]_2 = \boldsymbol{\gamma}_2^\intercal [\boldsymbol{g}(\boldsymbol{x})]_2 + \boldsymbol{\delta}_2^\intercal [\mathbb{q}_2]_2$, where $[\mathbb{q}_\kappa]_\kappa$ is the tuple of sampling oracle answers in $\mathbb{G}_\kappa$. Let ql_κ be the number of oracle queries in $\mathbb{G}_\kappa$ for $\kappa \in \{1, 2\}$. Define

$$Y_1(\boldsymbol{X}, \mathbb{Q}_1) = \textstyle\sum_{f_i \in \mathcal{R}_1} \gamma_{1i} f_i(\boldsymbol{X}) + \sum \delta_{1i} \mathbb{Q}_{1i} \ ,$$
$$Y_2(\boldsymbol{X}, \mathbb{Q}_2) = \textstyle\sum_{g_i \in \mathcal{R}_2} \gamma_{2i} g_i(\boldsymbol{X}) + \sum \delta_{2i} \mathbb{Q}_{2i} \ .$$

Thus, $[y_1]_1 = [Y_1(\boldsymbol{x}, \mathbb{q}_1)]_1$ and $[y_2]_2 = [Y_2(\boldsymbol{x}, \mathbb{q}_2)]_2$. Next, assume that both $\mathcal{A}$ and $\mathsf{Ext}_{\mathcal{A}}$ succeeded. The verifier checks that $[V(\boldsymbol{x}, \mathbb{q})]_T = [0]_T$, where for $s_1(\boldsymbol{X}) := \sum_{i=1}^{r_1} \gamma_{1i} f_i(\boldsymbol{X})$ and $s_2(\boldsymbol{X}) := \sum_{i=1}^{r_2} \gamma_{2i} g_i(\boldsymbol{X})$,

$$\begin{aligned} V(\boldsymbol{X}, \mathbb{Q}) :=& Y_1(\boldsymbol{X}, \mathbb{Q}_1) Y_2(\boldsymbol{X}, \mathbb{Q}_2) - t(\boldsymbol{X}) \\ =& (s_1(\boldsymbol{X}) + \textstyle\sum \delta_{1i} \mathbb{Q}_{1i})(s_2(\boldsymbol{X}) + \sum \delta_{2i} \mathbb{Q}_{2i}) - t(\boldsymbol{X}) \\ =& V^h(\boldsymbol{X}) + V^t(\boldsymbol{X}, \mathbb{Q}) \ , \end{aligned} \tag{3}$$

where $V^h(\boldsymbol{X}) = s_1(\boldsymbol{X}) s_2(\boldsymbol{X}) - t(\boldsymbol{X})$ and

$$V^t(\boldsymbol{X}, \mathbb{Q}) = s_1(\boldsymbol{X}) \textstyle\sum_i \delta_{2i} \mathbb{Q}_{2i} + s_2(\boldsymbol{X}) \sum_i \delta_{1i} \mathbb{Q}_{1i} + \sum_{i,j} \delta_{1i} \delta_{2j} \mathbb{Q}_{1i} \mathbb{Q}_{2j} \ .$$

Observe that $\deg s_\kappa \leq m \cdot d_\kappa$ for $\kappa \in \{1, 2\}$. Note that $V^t = 0$ in an AGM proof.

Let us now consider the three AGMOS cases.

Case A: $V(\boldsymbol{X}, \mathbb{Q}) = 0$. Then also $V^h(\boldsymbol{X}) = s_1(\boldsymbol{X}) s_2(\boldsymbol{X}) - t(\boldsymbol{X}) = 0$. However, $s_1(\boldsymbol{X})$ is in the span of f_i and $s_2(\boldsymbol{X})$ is in the span of g_i. Contradiction to the assumption $t \notin \operatorname{span}\{f_i g_j\}$. Thus, this case never materializes.

Case X: $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$ and $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$. In Fig. 3, we define a $(\boldsymbol{d}, m)$-FPR adversary $\mathcal{B}_{\mathsf{fpr}}$. Recall that $\mathcal{B}_{\mathsf{fpr}}$ has access to an oracle $\mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x}, \cdot)$, where $\boldsymbol{x} \leftarrow\$ \mathbb{F}^m$ is a trapdoor vector sampled by the challenger. $\mathcal{B}_{\mathsf{fpr}}$ queries $[r(\boldsymbol{x})]_\kappa \leftarrow \mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x}, (\kappa, r))$ for various $\kappa \in \{1, 2, T\}$ and $r(\boldsymbol{X}) \in \mathcal{R}_\kappa$ to construct $\mathsf{ck} \leftarrow ([\mathcal{R}_1(\boldsymbol{x})]_1, [\mathcal{R}_2(\boldsymbol{x})]_2, [\mathcal{R}_T(\boldsymbol{x})]_T)$. Note that $\mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x}, \cdot)$ will accept those queries since $\deg_{X_i}(r) \leq d_\kappa$ for all $i \in [1, m]$. Then, $\mathcal{B}_{\mathsf{fpr}}$ will run $\mathcal{A}$ and $\mathsf{Ext}_{\mathcal{A}}$ on input ck. $\mathcal{B}_{\mathsf{fpr}}$ aborts if $\mathsf{Ext}_{\mathcal{A}}$ fails. Otherwise, $\mathcal{B}_{\mathsf{fpr}}$ learns polynomials defined Eq. (3).

Next, $\mathcal{B}_{\mathsf{fpr}}$ follows one of the two strategies. Case $\mathsf{X}.1$: if $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$, then $V(\boldsymbol{X}, \mathbb{Q}) = V^h(\boldsymbol{X}) \neq 0$, but $0 = V(\boldsymbol{x}, \mathbb{q}) = V^h(\boldsymbol{x})$. Since $\deg V^h \leq \max(m \cdot (d_1 + d_2), d_t) = d_g$, then $\mathcal{B}_{\mathsf{fpr}}$ can output V^h to break $(\boldsymbol{d}, m)$-FPR.

Case $\mathsf{X}.2$: if $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0 \wedge V^t(\boldsymbol{x}, \mathbb{Q}) = 0$ (this subcase does not occur in AGM proofs), at least one of the coefficients of some $\mathbb{Q}_{\kappa i}$ or $\mathbb{Q}_{1i} \mathbb{Q}_{2j}$ is a non-zero polynomial that evaluates to 0 at $\boldsymbol{X} = \boldsymbol{x}$. Since the coefficient of each $\mathbb{Q}_{1i} \mathbb{Q}_{2j}$ does not depend on $\boldsymbol{X}$, it must be that for some κ and i^*, $p(\boldsymbol{X}) := s_\kappa(\boldsymbol{X}) \delta_{\kappa i^*}$ is

```
$\mathcal{B}_{\mathsf{fpr}}^{\mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x},\cdot),\ \mathcal{A}^{\mathcal{O}}}(\mathsf{p})$   $\mathcal{B}_{\mathsf{tofr}}^{\mathcal{A}^{\mathcal{O}}}(\mathsf{p})$
------------------------------------------------------------
For $\kappa \in \{1,2,T\}$, $q \in \mathcal{R}_\kappa$: $[q(\boldsymbol{x})]_\kappa \leftarrow \mathcal{O}^{\mathsf{fpr}}(\boldsymbol{x},(\iota,q))$;   $\boldsymbol{x} \leftarrow_{\$} \mathbb{F}^m$;
$\mathsf{ck} \leftarrow ([\mathcal{R}_1(\boldsymbol{x})]_1, [\mathcal{R}_2(\boldsymbol{x})]_2, [\mathcal{R}_T(\boldsymbol{x})]_T)$;
$r \leftarrow \mathsf{RND}_\lambda(\mathcal{A})$; $(t, [y_1]_1, [y_2]_2) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{p}, \mathsf{ck}; r)$;
$(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^{2} \leftarrow \overline{\mathsf{Ext}}_{\mathcal{A}}^{\mathcal{O}}(\mathsf{p}, \mathsf{ck}; r)$;   // e.g., $[y_1]_1 = \boldsymbol{\gamma}_1^\intercal [\boldsymbol{f}(\boldsymbol{x})]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}_1]_1$;
if $\overline{\mathsf{Ext}}_{\mathcal{A}}$ failed then return $\perp$; fi
Define $V, V^h, V^t$ as in Eq. (3);
if $V(\boldsymbol{X}, \mathbb{Q}) \neq 0 \wedge V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ then return $V^h(\boldsymbol{X})$; fi
if $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0 \wedge V^t(\boldsymbol{x}, \mathbb{Q}) = 0$ then
    $s_1(X) \leftarrow \sum \gamma_{1i} f_i(\boldsymbol{X})$; $s_2(X) \leftarrow \sum \gamma_{2i} g_i(\boldsymbol{X})$;
    find $s_\kappa(\boldsymbol{X})\delta_{\kappa i^*} \neq 0$; return $s_\kappa(\boldsymbol{X})\delta_{\kappa i^*}$; fi
if $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$ then return $\boldsymbol{v} = (V^h(\boldsymbol{x}) // s_2(\boldsymbol{X})\boldsymbol{\delta}_1 // s_1(\boldsymbol{X})\boldsymbol{\delta}_2 // \boldsymbol{\delta}_1 \otimes \boldsymbol{\delta}_2)$;
return $\perp$;
```

Fig. 3. Flexible Uber assumption: the FPR adversary $\mathcal{B}_{\mathsf{fpr}}$ and the TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$ in Theorem 3. The differences are dashed boxed ($\mathcal{B}_{\mathsf{fpr}}$) or dotted boxed ($\mathcal{B}_{\mathsf{tofr}}$).

a non-zero polynomial that has $\boldsymbol{x}$ as a root. Observe that $\deg p(\boldsymbol{X}) \leq \max(m \cdot d_1, m \cdot d_2) \leq d_g$. Thus, $\mathcal{B}_{\mathsf{fpr}}$ can output $p(\boldsymbol{X})$ to break $(\boldsymbol{d}, m)$-FPR.

Case Q: $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$. (This case does not occur in AGM.) In Fig. 3, we depict a TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$. $\mathcal{B}_{\mathsf{tofr}}$ samples $\boldsymbol{x}$ to construct $\mathsf{ck} \leftarrow ([\mathcal{R}_1(\boldsymbol{x})]_1, [\mathcal{R}_2(\boldsymbol{x})]_2, [\mathcal{R}_T(\boldsymbol{x})]_T)$. It runs $\mathcal{A}$ to obtain $(t, [y_1]_1, [y_2]_2)$, such that t is a non-trivial polynomial and $[y_1]_1 \bullet [y_2]_2 = [t(\boldsymbol{x})]_T$ and then uses $\overline{\mathsf{Ext}}_{\mathcal{A}}$ to extract field elements $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ such that $[y_1]_1 = \boldsymbol{\gamma}_1^\intercal [\boldsymbol{f}(\boldsymbol{x})]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}_1]_1$ and $[y_2]_2 = \boldsymbol{\gamma}_2^\intercal [\boldsymbol{f}(\boldsymbol{x})]_2 + \boldsymbol{\delta}_2^\intercal [\mathbb{q}_2]_2$. If the verifier accepts,

$$0 = V(\boldsymbol{x}, \mathbb{q}) = V^h(\boldsymbol{x}) + s_2(\boldsymbol{x}) \sum \delta_{1i} \mathbb{q}_{1i} + s_1(\boldsymbol{x}) \sum \delta_{2i} \mathbb{q}_{2i} + \sum \delta_{1i} \delta_{2j} \mathbb{q}_{1i} \mathbb{q}_{2j}$$

(see Eq. (3)). Thus, $\mathcal{B}$ outputs $\boldsymbol{v} = (V^h(\boldsymbol{x}) // s_2(\boldsymbol{x})\boldsymbol{\delta}_1 // s_1(\boldsymbol{x})\boldsymbol{\delta}_2 // \boldsymbol{\delta}_1 \otimes \boldsymbol{\delta}_2)$. Since $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$, then $\boldsymbol{v} \neq \boldsymbol{0}$ and $\mathcal{B}$ breaks the TOFR assumption.

Thus, either the algebraic extractor fails, or, if it succeeds, we get one of the above cases. Hence, $\mathsf{Adv}^{\mathrm{sfuber}}_{\mathsf{Pgen},\mathcal{R}_1,\mathcal{R}_2,\mathcal{R}_T,d_t,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{agmos}}_{\mathsf{Pgen},\mathcal{EF},\mathcal{DF},\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\lambda) + \mathsf{Adv}^{\mathrm{fpr}}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathrm{tofr}}_{\mathsf{Pgen},\mathcal{EF},\mathcal{DF},\mathcal{B}_{\mathsf{tofr}}}(\lambda)$. This concludes the proof. □

5.2 The PKE Assumption

Let us recall the Power Knowledge of Exponent (PKE) assumption [DFGK14].

Definition 5. *The (asymmetric) $d(\lambda)$-PKE assumption holds for* Pgen*, if for every non-uniform PPT adversary $\mathcal{A}$, there exists a non-uniform PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$*, such that* $\mathsf{Adv}^{\mathrm{pke}}_{d,\mathsf{Pgen},\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} \mathbb{y}_1 = \mathbb{y}_2 \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); x \leftarrow_{\$} \mathbb{F}; r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); \\ \mathbb{y}_1 \neq \sum_{i=0}^{d} \gamma_i x^i & ([\mathbb{y}_1]_1, [\mathbb{y}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, ([x^i]_1, [x^i]_2)_{i=0}^{d}; r); \\ & (\gamma_i)_{i=0}^{d} \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{p}, ([x^i]_1, [x^i]_2)_{i=0}^{d}; r) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

(See [Gro10] for another variant of PKE.) Our AGMOS analysis shows that the PKE stays secure even if the adversary can sample new group elements from non-uniform distributions. We are not aware of any prior result of this type.

Theorem 4 (d-PKE). *If the $(\mathcal{EF}, \mathcal{DF})$-TOFR assumption holds, then $d(\lambda)$-PKE holds in the AGMOS.*

Proof. Let $\mathcal{A}$ be an AGMOS adversary for the d-PKE assumption that gets as an input $([x^i]_1, [x^i]_2)_{i=0}^d$. Let us denote $\mathbb{x} = (x^i)_{i=0}^d$. The challenger accepts the adversary's output if $\mathcal{A}$ outputs $[\mathbb{y}_1]_1$, $[\mathbb{y}_2]_2$ such that $\mathbb{y}_1 = \mathbb{y}_2$. The adversary wins the game if the challenger accepts, but no efficient extractor can recover $\boldsymbol{\mu}$ such that $\boldsymbol{\mu}^\intercal \mathbb{x} = \mathbb{y}_1 = \mathbb{y}_2$ with an overwhelming probability. As before, $[\mathbb{q}_1]_1$, $[\mathbb{q}_2]_2$ are vectors of query responses from the oracle.

We construct a PKE extractor $\mathsf{Ext}_\mathcal{A}$. Since $\mathcal{A}$ is an AGMOS adversary, there exists an AGMOS extractor $\overline{\mathsf{Ext}}_\mathcal{A}$ that extracts $(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa)_\kappa \in \mathbb{F}^{d+1} \times \mathbb{F}^{\mathsf{ql}_\kappa}$ for $\kappa \in \{1, 2\}$, such that $\boldsymbol{\gamma}_1^\intercal [\mathbb{x}]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}_1]_1 = [\mathbb{y}_1]_1$ and $\boldsymbol{\gamma}_2^\intercal [\mathbb{x}]_2 + \boldsymbol{\delta}_2^\intercal [\mathbb{q}_2]_2 = [\mathbb{y}_2]_2$ with an overwhelming probability. Here, ql_κ is the number of oracle queries in $\mathbb{G}_\kappa$.

In Fig. 4, we depict $\mathsf{Ext}_\mathcal{A}$. $\mathsf{Ext}_\mathcal{A}$ runs both $\mathcal{A}$ and $\overline{\mathsf{Ext}}_\mathcal{A}$. $\mathsf{Ext}_\mathcal{A}$ returns $\perp$ if $\overline{\mathsf{Ext}}_\mathcal{A}$ fails or $\boldsymbol{\delta} := \left(\begin{smallmatrix} \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \end{smallmatrix}\right) \neq \mathbf{0}_{\mathsf{ql}_1 + \mathsf{ql}_2}$. Otherwise, $\boldsymbol{\mu} := \boldsymbol{\gamma}_1 = \boldsymbol{\gamma}_2$. Thus, $\mathsf{Ext}_\mathcal{A}$ returns $\boldsymbol{\mu}$. Since $\boldsymbol{\delta} = \mathbf{0}_{\mathsf{ql}_1 + \mathsf{ql}_2}$, $\mathbb{y}_1 = \mathbb{y}_2 = \boldsymbol{\mu}^\intercal \mathbb{x}$. Hence, $\mathcal{A}$ can win iff either (1) $\overline{\mathsf{Ext}}_\mathcal{A}$ fails or (2) $\overline{\mathsf{Ext}}_\mathcal{A}$ succeeds but $\boldsymbol{\delta} \neq \mathbf{0}_{\mathsf{ql}_1 + \mathsf{ql}_2}$.

The probability $\varepsilon_0 := \Pr[\mathbb{y}_1 = \mathbb{y}_2 \wedge \overline{\mathsf{Ext}}_\mathcal{A} \text{ fails}]$ is $\varepsilon_0 \leq \mathsf{Adv}^{\mathrm{agmos}}_{\mathsf{Pgen}, \mathcal{EF}, \mathcal{A}, \overline{\mathsf{Ext}}_\mathcal{A}}(\lambda)$, which is negligible by definition for an AGMOS adversary $\mathcal{A}$. Consider now the case (2). Let $\varepsilon_1 := \Pr[\mathbb{y}_1 = \mathbb{y}_2 \wedge \overline{\mathsf{Ext}}_\mathcal{A} \text{ succeeds} \wedge \boldsymbol{\delta} \neq \mathbf{0}_{\mathsf{ql}_1 + \mathsf{ql}_2}]$. We build a reduction $\mathcal{B}_{\mathsf{tofr}}$ to TOFR assumption. Assume that $\mathcal{A}$ outputs $([\mathbb{y}_1]_1, [\mathbb{y}_2]_2)$ such that $\mathbb{y}_1 = \mathbb{y}_2$ and that $\overline{\mathsf{Ext}}_\mathcal{A}$ succeeds in extracting a linear representation $((\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa)$ for $\kappa \in \{1, 2\})$. That is, we are going to bind the probability $\Pr[\boldsymbol{\delta} \neq \mathbf{0}_{\mathsf{ql}_1 + \mathsf{ql}_2} \mid \mathbb{y}_1 = \mathbb{y}_2 \wedge \overline{\mathsf{Ext}}_\mathcal{A} \text{ succeeds}] \geq \varepsilon_1$.

The TOFR reduction $\mathcal{B}_{\mathsf{tofr}}(\mathsf{p})$ (also depicted in Fig. 4) samples $x \leftarrow\!\!\$\ \mathbb{F}$ and uses it to construct an input $([\mathbb{x}]_1, [\mathbb{x}]_2)$ for $\mathcal{A}$. It then runs $\overline{\mathsf{Ext}}_\mathcal{A}$ to obtain $\boldsymbol{\gamma}, \boldsymbol{\delta}$. Since $\mathbb{y}_1 = \mathbb{y}_2$, we get that $0 = \mathbb{y}_1 - \mathbb{y}_2 = (\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2)^\intercal \mathbb{x} + \boldsymbol{\delta}_1^\intercal \mathbb{q}_1 - \boldsymbol{\delta}_2^\intercal \mathbb{q}_2 + \mathbf{0}_{\mathsf{ql}_1 \mathsf{ql}_2}^\intercal (\mathbb{q}_1 \otimes \mathbb{q}_2)$. $\mathcal{B}_{\mathsf{tofr}}$ returns $\left((\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2)^\intercal \mathbb{x} // \boldsymbol{\delta}_1 // -\boldsymbol{\delta}_2 // \mathbf{0}_{\mathsf{ql}_1 \mathsf{ql}_2}\right)$. If $\boldsymbol{\delta} \neq \mathbf{0}_{\mathsf{ql}_1 + \mathsf{ql}_2}$, $\mathcal{B}_{\mathsf{tofr}}$ has broken TOFR assumption.

We get that $\mathsf{Adv}^{\mathrm{pke}}_{d, \mathsf{Pgen}, \mathcal{A}, \mathsf{Ext}_\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{agmos}}_{\mathsf{Pgen}, \mathcal{EF}, \mathcal{A}, \overline{\mathsf{Ext}}_\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathrm{tofr}}_{\mathsf{Pgen}, \mathcal{EF}, \mathcal{DF}, \mathcal{B}_{\mathsf{tofr}}}(\lambda) = \mathsf{negl}(\lambda)$. This concludes the proof. □

5.3 Extractability of The KZG Polynomial Commitment Scheme

In a polynomial commitment scheme (PCS, [KZG10]), the committer first commits to a polynomial $f(X)$ and then opens it to an evaluation $f(\alpha)$ at some point α chosen by the verifier. In the current paper, we focus on the non-randomized PCSs (like the first PCS construction inf [KZG10]) since such PCSs are used to construct many efficient SNARKs.

More formally, a polynomial commitment scheme over a field $\mathbb{F}$ is a tuple of PPT algorithms $\mathsf{PC} = (\mathsf{KC}, \mathsf{com}, \mathsf{open}, \mathsf{V})$, such that:

```
$\boxed{\mathsf{Ext}_{\mathcal{A}}(\mathsf{p}, [\mathbb{x}]_1, [\mathbb{x}]_2; r)}$ $\mathcal{B}_{\mathsf{tofr}}^{\mathcal{A}^{\mathcal{O}}}(\mathsf{p})$
$x \leftarrow_{\$} \mathbb{F}; r \leftarrow \mathsf{RND}_{\lambda}(\mathcal{A});$
$[\mathbb{y}_1, \mathbb{y}_2]_1 \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{p}, [\mathbb{x}]_1, [\mathbb{x}]_2; r); (\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2 \leftarrow \overline{\mathsf{Ext}}_{\mathcal{A}}^{\mathcal{O}}(\mathsf{p}, [\mathbb{x}]_1, [\mathbb{x}]_2; r);$
    $/\!\!/ \ [\mathbb{y}_1]_1 = \boldsymbol{\gamma}_1^\intercal [\mathbb{x}]_1 + \boldsymbol{\delta}_1 [\mathbb{q}_1]_1,\ [\mathbb{y}_2]_1 = \boldsymbol{\gamma}_2^\intercal [\mathbb{x}]_2 + \boldsymbol{\delta}_2 [\mathbb{q}_2]_2$
**if** $\boldsymbol{\delta} := \binom{\boldsymbol{\delta}_1}{\boldsymbol{\delta}_2} \neq \mathbf{0}_{\mathsf{ql}_1+\mathsf{ql}_2}$ **then return** $((\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2)^\intercal \mathbb{x} /\!/ \boldsymbol{\delta}_1 /\!/ - \boldsymbol{\delta}_2 /\!/ \mathbf{0}_{\mathsf{ql}_1 \cdot \mathsf{ql}_2});$
$\boxed{\textbf{if } \boldsymbol{\delta} = \mathbf{0}_{\mathsf{ql}_1+\mathsf{ql}_2} \textbf{ then return } \boldsymbol{\gamma}_1; \textbf{fi}}$
**return** $\perp$;
```

Fig. 4. $\mathsf{Ext}_{\mathcal{A}}$ and $\mathcal{B}_{\mathsf{tofr}}$ in the proof of Theorem 4 (PKE). Here, $\mathbb{x} = (x^i)_{i=0}^d$. The differences between are $\boxed{\text{boxed}}$ (Ext) or dotted boxed ($\mathcal{B}_{\mathsf{tofr}}$).

(1) $\mathsf{KC} : (1^\lambda, d) \mapsto (\mathsf{ck}, \mathsf{td})$ is a randomized commitment key generation algorithm, where d is the maximum degree of committed polynomials, ck is a commitment key and td is a trapdoor.

(2) $\mathsf{com} : (\mathsf{ck}, f(X)) \mapsto (\mathbb{c}, \mathbb{d})$ is a deterministic commitment algorithm that, given a polynomial $f(X) \in \mathbb{F}[X]$ of degree $\leq d$, outputs commitment information $\mathbb{c}$ and decommitment information $\mathbb{d}$.

(3) $\mathsf{open} : (\mathsf{ck}, \mathbb{c}, \alpha, \mathbb{d}) \mapsto (f(\alpha), \pi)$ is a deterministic opening algorithm that, given an evaluation point α, outputs $f(\alpha)$ together with opening proof π.

(4) $\mathsf{V} : (\mathsf{ck}, \mathbb{c}, \alpha, \eta, \pi) \mapsto 0/1$ is a deterministic verification algorithm that, given candidate value η for $f(\alpha)$, outputs 1 if $\eta = f(\alpha)$ and 0, otherwise.

Definition 6. *A polynomial commitment scheme* PC *is extractable for* Pgen*, if for any* $d = \mathsf{poly}(\lambda)$ *and every non-uniform PPT adversary* $\mathcal{A}$*, there exists a non-uniform PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$*, such that* $\mathsf{Adv}^{\mathrm{ext}}_{\mathsf{Pgen},\mathsf{PC},d,\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} \mathsf{V}(\mathsf{ck}, \mathbb{c}, \alpha, \eta, \pi) = 1 \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{KC}(\mathsf{p}, d); \\ \big(\mathbb{c} \neq \mathsf{com}(f(X)) \vee & r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); (\mathbb{c}, \alpha, \eta, \pi) \leftarrow \mathcal{A}(\mathsf{p}, \mathsf{ck}; r); \\ \deg f > d \ \vee \ f(\alpha) \neq \eta\big) & f(X) \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{p}, \mathsf{ck}; r) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

The KZG PCS. Let $f(X)$ be a polynomial of degree $\leq d$. In Fig. 5, we depict the famous Kate-Zaverucha-Goldberg (KZG, [KZG10]) polynomial commitment scheme. Its security is based on the fact that $(X - \alpha) \mid (f(X) - \eta) \Leftrightarrow f(\alpha) = \eta$. Next, we prove in AGMOS that KZG is extractable. Since KZG extractability is a knowledge assumption, one could expect that it is sufficient to assume TOFR just as in the proof of PKE. However, if the adversary can efficiently compute x (i.e., PDL does not hold), then one can compute an accepting opening $[\pi]_1 = [\mathbb{c} - \eta]_1/(x - \alpha)$ for any values η and α. Thus, similarly to the AGM proofs of KZG extractability, one has to assume FPR (or PDL). The extractability of a PCS combines both an extractability property (extracting $f(X)$) and a computational hardness property (it is hard to find (α, η) such that $f(\alpha) \neq \eta$).

Remark 1. In the case of KZG, the stronger extractability notion, where the adversary who only produces $\mathbb{c}$ must know f is not secure in the AGMOS (assuming

$\mathsf{KC}(1^\lambda, d)$: output $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$, $x \leftarrow\$ \mathbb{Z}_p^*$, $\mathsf{ck} = ([(x^i)_{i=0}^d]_1, [1, x]_2)$.
$\mathsf{com}(\mathsf{ck}, f(X))$: $[\mathbb{c}]_1 \leftarrow [f(x)]_1 = \sum_{j=0}^d f_j [x^j]_1$; return $([\mathbb{c}]_1, \mathbb{d} = f(X))$;
$\mathsf{open}(\mathsf{ck}, [\mathbb{c}]_1, \alpha, \mathbb{d} = f(X))$: $\pi(X) \leftarrow (f(X) - f(\alpha)) / (X - \alpha)$; $[\pi(x)]_1 \leftarrow \sum_{j=0}^{n-1} \pi_j [x^j]_1$; return $(\eta = f(\alpha), [\pi(x)]_1)$;
$\mathsf{V}(\mathsf{ck}, [\mathbb{c}]_1, \alpha, \eta, [\pi(x)]_1)$: check $[\mathbb{c} - \eta]_1 \bullet [1]_2 = [\pi(x)]_1 \bullet [x - \alpha]_2$.

Fig. 5. The KZG polynomial commitment scheme.

DL is secure); essentially, this notion is equivalent to SpurKE. This is the flaw in say [CFF+21,GWC19], mentioned in the introduction. While we leave the study of the security of such SNARKs for future work (it might be that they are secure but need a different security proof), we emphasize that one should not use the stronger extractability notion. In Appendix E, we give a concrete example why this is needed, showing that a common trick used to optimize quadratic tests in KZG-based zk-SNARKs results in non-extractability.

Below, we consider $\boldsymbol{d} = (d, 1, d+1, d_g)$.

Theorem 5 (Extractability of KZG). *If the* $(\boldsymbol{d}, m)$-FPR *and* $(\mathcal{EF}, \mathcal{DF})$-TOFR *assumptions hold, then KZG is extractable in the AGMOS.*

Proof. Let $\mathcal{A}$ be a non-uniform PPT KZG extractability AGMOS adversary that with some non-negligible probability outputs $([\mathsf{c}]_1, \alpha, \eta, [\pi]_1)$ such that $\mathsf{c} - \eta = \pi \cdot (x - \alpha)$. Let $\mathbb{x}_1 = (x^i)_{i=0}^d$. Since $\mathcal{A}$ is an AGMOS adversary, there exists an extractor $\overline{\mathsf{Ext}}_{\mathcal{A}}$ that extracts $\boldsymbol{\gamma}_1$, $\boldsymbol{\delta}_1$, $\boldsymbol{\gamma}_2$, and $\boldsymbol{\delta}_2$, such that $[\mathsf{c}]_1 = \boldsymbol{\gamma}_1^\intercal [\mathbb{x}_1]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}_1]_1$ and $[\pi]_1 = \boldsymbol{\gamma}_2^\intercal [\mathbb{x}_1]_1 + \boldsymbol{\delta}_2^\intercal [\mathbb{q}_1]_1$, where $[\mathbb{q}_1]_1$ is the tuple of sampling oracle answers in $\mathbb{G}_1$. As before, let ql_κ be the number of oracle queries in $\mathbb{G}_\kappa$ for $\kappa \in \{1, 2\}$.

Define $\mathsf{c}(X, \mathbb{Q}) = \gamma_1(X) + \sum_{i=1}^{\mathsf{ql}_1} \delta_{1i} \mathbb{Q}_{1i}$ and $\pi(X, \mathbb{Q}) = \gamma_2(X) + \sum_{i=1}^{\mathsf{ql}_1} \delta_{2i} \mathbb{Q}_{1i}$, where $\deg \gamma_1, \deg \gamma_2 \leq d$. Note $[\mathsf{c}]_1 = [\mathsf{c}(x, \mathbb{q}_1)]_1$ and $[\pi]_1 = [\pi(x, \mathbb{q}_1)]_1$. The verification ascertains that $V(x, \mathbb{q}) = 0$, where

$$V(X, \mathbb{Q}) = \mathsf{c}(X, \mathbb{Q}) - \eta - \pi(X, \mathbb{Q})(X - \alpha) = V^h(X) + V^t(X, \mathbb{Q}) \ , \qquad (4)$$

with $V^h(X) = \gamma_1(X) - \eta - \gamma_2(X)(X - \alpha)$ and $V^t(X, \mathbb{Q}) = \sum_{i=1}^{\mathsf{ql}_1} (\delta_{1i} + \alpha \delta_{2i}) \mathbb{Q}_{1i} - X \sum_{i=1}^{\mathsf{ql}_1} \delta_{2i} \mathbb{Q}_{1i}$.

We construct a KZG extractor $\mathsf{Ext}_{\mathcal{A}}$ in Fig. 6. $\mathsf{Ext}_{\mathcal{A}}$ aborts when $\overline{\mathsf{Ext}}_{\mathcal{A}}$ fails. Otherwise, it returns $\gamma_1(X) \in \mathbb{F}^{(\leq d)}[X]$. Let us argue that $\mathsf{Ext}_{\mathcal{A}}$ succeeds if there is no abort. We consider the usual three cases.

Case A: $V(X, \mathbb{Q}) = 0$. If $V(X, \mathbb{Q}) = 0$, then each of its coefficients is 0. In particular, the coefficient of $X\mathbb{Q}_{1i}$ is δ_{2i}, and thus $\delta_{2i} = 0$. The coefficient of $\mathbb{Q}_{1i}$ is $\delta_{1i} + \alpha \delta_{2i} = \delta_{1i}$, and thus $\delta_{1i} = 0$. Thus, $\delta_{1i} = \delta_{2i} = 0$. Hence, we are back in the AGM setting, with $V(X, \mathbb{Q}) = V^h(X) = 0$, meaning that $\gamma_1(X) = \eta + \gamma_2(X)(X - \alpha)$. In particular, $\mathsf{c}(\alpha) = \gamma_1(\alpha) = \eta$ and $f(X) := \gamma_1(X) \in \mathbb{F}^{(\leq d)}[X]$ returned by $\mathsf{Ext}_{\mathcal{A}}$ satisfies $[\mathsf{c}]_1 = [\mathsf{c}(x, \mathbb{q}_1)]_1 = [f(x)]_1$.

Case X: $V(X, \mathbb{Q}) \neq 0$ and $V^t(x, \mathbb{Q}) = 0$: We show that this case can happen only with negligible probability through a reduction to FPR. In Fig. 6,

```
$\boxed{\mathsf{Ext}_{\mathcal{A}}(\mathsf{p}; \mathbb{x}; r)}$ $\boxed{\mathcal{B}_{\mathsf{fpr}}^{\mathcal{A}^{\mathcal{O}}}(\mathsf{p}; \mathbb{x})}$ $\boxed{\mathcal{B}_{\mathsf{tofr}}^{\mathcal{A}^{\mathcal{O}}}(\mathsf{p}; [1]_1, [x]_2)}$   // $\mathbb{x} = ([1, \ldots, x^d]_1, [1, x]_2)$
$\boxed{\boxed{r \leftarrow\!\!\$\ \mathsf{RND}_\lambda(\mathcal{A});}}$ $\boxed{x \leftarrow\!\!\$\ \mathbb{F}}$; $([\mathbb{c}]_1, \alpha, \eta, [\pi]_1) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{p}, \mathbb{x}; r)$;   // $\mathbb{y}_1 = (\begin{smallmatrix} \mathbb{c} \\ \pi \end{smallmatrix})$
$(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2 \leftarrow \overline{\mathsf{Ext}}_{\mathcal{A}}^{\mathcal{O}}(\mathsf{p}, \mathbb{x}; r)$; if $\overline{\mathsf{Ext}}_{\mathcal{A}}$ failed then return $\perp$; fi
Define $V(X, \mathbb{Q})$, $V^h(X)$, $V^t(X, \mathbb{Q})$ as in Eq. (4);
$\boxed{\textbf{if } V(X, \mathbb{Q}) = 0 \textbf{ then return } \mathbb{c}(X) \leftarrow \sum_{i=0}^{d} \gamma_{1,i+1} X^i; \textbf{fi}}$
if $V(X, \mathbb{Q}) \neq 0 \wedge V^t(X, \mathbb{Q}) = 0$ then return $V^h(X)$; fi
if $V(X, \mathbb{Q}) \neq 0 \ \wedge\ V^t(x, \mathbb{Q}) = 0$ then
    $\beta_i(X) \leftarrow -\delta_{2i} X + \delta_{1i} + \alpha\delta_{2i}$; find $\beta_{i^*}(X) \neq 0$; return $\beta_{i^*}(X)$; fi
$\boxed{\textbf{if } V^t(x, \mathbb{Q}) \neq 0 \textbf{ then return } (V^h(x) // \boldsymbol{\delta}_1 + (\alpha - x)\boldsymbol{\delta}_2 // \mathbf{0}_{\mathsf{ql}_2 + \mathsf{ql}_1 \mathsf{ql}_2}); \textbf{fi}}$
return $\perp$;
```

Fig. 6. KZG extractability: the extractor $\mathsf{Ext}_{\mathcal{A}}$, the FPR adversary $\mathcal{B}_{\mathsf{fpr}}$, and the TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$ in the proof of Theorem 5, where the differences are either $\boxed{\textit{boxed}}$ ($\mathsf{Ext}_{\mathcal{A}}$), $\boxed{\textit{dashedboxed}}$ ($\mathcal{B}_{\mathsf{fpr}}$) or $\boxed{\textit{dottedboxed}}$ ($\mathcal{B}_{\mathsf{tofr}}$).

we depict the FPR adversary $\mathcal{B}_{\mathsf{fpr}}$ that works in this case. $\mathcal{B}_{\mathsf{fpr}}$ obtains $\mathbb{x} = ([1, x, \ldots, x^d]_1, [1, x]_2)$ from $d + 1$ calls to the FPR oracle. It then calls $\mathcal{A}$ on input $\mathbb{x}$. $\mathcal{B}_{\mathsf{fpr}}$ extracts $(\boldsymbol{\gamma}_1, \boldsymbol{\delta}_1, \boldsymbol{\gamma}_2, \boldsymbol{\delta}_2)$ by using the AGMOS extractor. If the extractor fails, $\mathcal{B}_{\mathsf{fpr}}$ aborts.

Case X.1: if $V(X, \mathbb{Q}) \neq 0$ and $V^t(X, \mathbb{Q}) = 0$, $V^h(X)$ is a non-zero polynomial with x as a root and V^h has a degree at most $d + 1 \leq d_g$. Thus, $\mathcal{B}_{\mathsf{fpr}}$ is successful.

Case X.2: if $V^t(X, \mathbb{Q}) \neq 0$ and $V^t(x, \mathbb{Q}) = 0$, then let us write $V^t(X, \mathbb{Q}) = \sum_{i=1}^{\mathsf{ql}_1} \beta_i(X) \cdot \mathbb{Q}_{1i}$, where $\beta_i(X) = -\delta_{2i} X + \delta_{1i} + \alpha\delta_{2i}$. Since $V^t(x, \mathbb{Q}) = 0$, then $\beta_i(x) = 0$ for $i \in [1, \mathsf{ql}_1]$. If all $\beta_i(X)$ have degree less than 1 (i.e., they are constants), then $\beta_i(X) = \beta_i(x) = 0$ and it contradicts the condition that $V^t(X, \mathbb{Q}) \neq 0$. Thus, there exists a degree-1($\leq d_g$ polynomial $\beta_{i^*}(X) = -\delta_{2i^*} X + \delta_{1i^*} + \alpha\delta_{2i^*}$, such that $\beta_{i^*}(x) = 0$. This shows that $\mathcal{B}_{\mathsf{fpr}}$ is successful in this case too.

In both cases, $\mathcal{B}_{\mathsf{fpr}}$ breaks the FPR assumption.

Case Q ($V^t(x, \mathbb{Q}) \neq 0$): In Fig. 6, we depict a TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$. $\mathcal{B}_{\mathsf{tofr}}$ samples x to construct $\big([1, \ldots, x^d]_1, [1, x]_2\big)$. It then runs $\mathcal{A}$ to obtain $([\mathsf{c}]_1, \alpha, \eta, [\pi]_1)$, such that $[\mathsf{c} - \eta]_1 \bullet [1]_2 = [\pi]_1 \bullet [x - \alpha]_2$ and uses $\overline{\mathsf{Ext}}_{\mathcal{A}}$ to extract field elements $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ such that $[\mathsf{c}]_1 = \boldsymbol{\gamma}_1^\intercal [\mathbb{x}_1]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}_1]_1$ and $[\pi]_1 = \boldsymbol{\gamma}_2^\intercal [\mathbb{x}_1]_1 + \boldsymbol{\delta}_2^\intercal [\mathbb{q}_1]_1$. If the verifier accepts, $0 = V(x, \mathbb{q}) = V^h(x) + \sum_{i=1}^{\mathsf{ql}_1} (\delta_{1i} + \alpha\delta_{2i}) \mathbb{q}_{1i} - x \sum_{i=1}^{\mathsf{ql}_1} \delta_{2i} \mathbb{q}_{1i}$. (see Eq. (4)). Thus, $\mathcal{B}$ outputs

$$\boldsymbol{v} \leftarrow \begin{pmatrix} V^h(x) \\ \boldsymbol{\delta}_1 - (x - \alpha)\boldsymbol{\delta}_2 \\ \mathbf{0}_{\mathsf{ql}_2 + \mathsf{ql}_1 \mathsf{ql}_2} \end{pmatrix} .$$

Since $V^t(x, \mathbb{Q}) \neq 0$, then $\boldsymbol{v} \neq \mathbf{0}$ and $\mathcal{B}$ breaks the TOFR assumption.

Thus, $\mathsf{Adv}_{\mathsf{Pgen}, \mathsf{kzg}, \mathcal{A}, \mathsf{Ext}_{\mathcal{A}}}^{\mathrm{ext}}(\lambda) \leq \mathsf{Adv}_{\mathsf{Pgen}, \mathcal{EF}, \mathcal{A}, \overline{\mathsf{Ext}}_{\mathcal{A}}}^{\mathrm{agmos}}(\lambda) + \mathsf{Adv}_{\mathsf{Pgen}, \boldsymbol{d}, m, \mathcal{B}_{\mathsf{fpr}}}^{\mathrm{fpr}}(\lambda) + \mathsf{Adv}_{\mathsf{Pgen}, \mathcal{EF}, \mathcal{DF}, \mathcal{B}_{\mathsf{tofr}}}^{\mathrm{tofr}}(\lambda)$. This concludes the proof. □

6 AGM-AGMOS Separation

Next, we will explain why some knowledge assumptions that are secure in the AGM, by definition, are not secure in the AGMOS (assuming DL is hard) while others might be secure. We will study general (publicly-verifiable) knowledge assumptions with some fixed verification equations. For the sake of simplicity, we will somewhat restrict the latter class. In particular, we only study the strongest possible knowledge assumption (the "TotalKE"). Later results of this section also only hold when the adversary's input is $([1]_1, [1]_2)$ (i.e., it does not depend on any trapdoors; this means that outputting a linear representation is the same as outputting discrete logarithms), but we will start with the general case.

We will extensively rely on the notation introduced in Section 3.1. We will also introduce matrix-vector notation to clarify the exposition; while this notation is only used in the current section, we find the tensor notation to be more cumbersome in this concrete case. Let $\mathsf{il}_\kappa/\mathsf{ol}_\kappa$ be the length of the input/output in $\mathbb{G}_\kappa$ and ql_κ be the number of oracle queries in $\mathbb{G}_\kappa$. A publicly-verifiable pairing-product verification polynomial can be written as $V^{\mathsf{expl}}(\mathbb{x}, \mathbb{y}) = \left(\begin{smallmatrix} \mathbb{x}_1 \\ \mathbb{y}_1 \end{smallmatrix}\right)^\intercal \boldsymbol{M} \left(\begin{smallmatrix} \mathbb{x}_2 \\ \mathbb{y}_2 \end{smallmatrix}\right)$ for some public matrix $\boldsymbol{M}$. Let $\boldsymbol{M} = \left(\begin{smallmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{smallmatrix}\right)$ for submatrices $\boldsymbol{M}_{ij}$. Here, say, $\boldsymbol{M}_{11} \in \mathbb{F}^{\mathsf{il}_1 \times \mathsf{il}_2}$ and $\boldsymbol{M}_{22} \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{ol}_2}$. The AGMOS extractor extracts the matrices $\boldsymbol{\gamma}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa \times \mathsf{il}_\kappa}$ and $\boldsymbol{\delta}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa \times \mathsf{ql}_\kappa}$ from $[\mathbb{y}_1]_1$ and $[\mathbb{y}_2]_2$.

Let $\boldsymbol{\varGamma}$ and $\boldsymbol{\varDelta}$ be indeterminates corresponding to $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$. Similarly to $\boldsymbol{\gamma}_\kappa$ and $\boldsymbol{\delta}_\kappa$, we think of $\boldsymbol{\varGamma}_\kappa$ and $\boldsymbol{\varDelta}_\kappa$ as ($\mathsf{ol}_\kappa \times \mathsf{il}_\kappa$ and $\mathsf{ol}_\kappa \times \mathsf{ql}_\kappa$) matrices. Let $\boldsymbol{P}_\kappa(\boldsymbol{\varGamma}, \boldsymbol{\varDelta}) := \left(\begin{smallmatrix} \boldsymbol{I}_{\mathsf{il}_\kappa} & \boldsymbol{0}_{\mathsf{il}_\kappa \times \mathsf{ql}_\kappa} \\ \boldsymbol{\varGamma}_\kappa & \boldsymbol{\varDelta}_\kappa \end{smallmatrix}\right)$ be the matrix so that $\left(\begin{smallmatrix} \mathbb{x}_\kappa \\ \mathbb{y}_\kappa \end{smallmatrix}\right) = \boldsymbol{P}_\kappa(\boldsymbol{\gamma}, \boldsymbol{\delta}) \left(\begin{smallmatrix} \mathbb{x}_\kappa \\ \mathbb{q}_\kappa \end{smallmatrix}\right)$. Define

$$\begin{aligned} \boldsymbol{N}(\boldsymbol{\varGamma}, \boldsymbol{\varDelta}) :=& \boldsymbol{P}_1^\intercal(\boldsymbol{\varGamma}, \boldsymbol{\varDelta}) \boldsymbol{M} \boldsymbol{P}_2(\boldsymbol{\varGamma}, \boldsymbol{\varDelta}) \\ =& \begin{pmatrix} \boldsymbol{I}_{\mathsf{il}_1} & \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{ql}_1} \\ \boldsymbol{\varGamma}_1 \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{il}_1} & \boldsymbol{\varDelta}_1 \end{pmatrix}^\intercal \cdot \begin{pmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{I}_{\mathsf{il}_2} & \boldsymbol{0}_{\mathsf{il}_2 \times \mathsf{ql}_2} \\ \boldsymbol{\varGamma}_2 & \boldsymbol{\varDelta}_2 \end{pmatrix} \\ =& \begin{pmatrix} \boldsymbol{M}_{11} + \boldsymbol{\varGamma}_1^\intercal \boldsymbol{M}_{21} + (\boldsymbol{M}_{12} + \boldsymbol{\varGamma}_1^\intercal \boldsymbol{M}_{22}) \boldsymbol{\varGamma}_2 & (\boldsymbol{M}_{12} + \boldsymbol{\varGamma}_1^\intercal \boldsymbol{M}_{22}) \boldsymbol{\varDelta}_2 \\ \boldsymbol{\varDelta}_1^\intercal (\boldsymbol{M}_{21} + \boldsymbol{M}_{22} \boldsymbol{\varGamma}_2) & \boldsymbol{\varDelta}_1^\intercal \boldsymbol{M}_{22} \boldsymbol{\varDelta}_2 \end{pmatrix} \enspace . \end{aligned}$$

Note that while $\boldsymbol{M}$ corresponds to V^{expl}, $\boldsymbol{N}$ corresponds to V. Let

$$\boldsymbol{f}_{\boldsymbol{M}}(\boldsymbol{\varGamma}, \boldsymbol{\varDelta}) = \begin{pmatrix} \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{il}_2} & (\boldsymbol{M}_{12} + \boldsymbol{\varGamma}_1^\intercal \boldsymbol{M}_{22}) \boldsymbol{\varDelta}_2 \\ \boldsymbol{\varDelta}_1^\intercal (\boldsymbol{M}_{21} + \boldsymbol{M}_{22} \boldsymbol{\varGamma}_2) & \boldsymbol{\varDelta}_1^\intercal \boldsymbol{M}_{22} \boldsymbol{\varDelta}_2 \end{pmatrix}$$

be equal to $\boldsymbol{N}(\boldsymbol{\varGamma}, \boldsymbol{\varDelta})$, except that its top left submatrix is $\boldsymbol{0}$. We rewrite $\boldsymbol{f}_{\boldsymbol{M}} = \boldsymbol{0}$ as the following equivalent system of polynomial equations in $(\boldsymbol{\varGamma}_1, \boldsymbol{\varGamma}_2, \boldsymbol{\varDelta}_1, \boldsymbol{\varDelta}_2)$:

$$\begin{gathered} (\boldsymbol{M}_{12} + \boldsymbol{\varGamma}_1^\intercal \boldsymbol{M}_{22}) \boldsymbol{\varDelta}_2 = \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{ql}_2} \enspace , \quad \boldsymbol{\varDelta}_1^\intercal (\boldsymbol{M}_{21} + \boldsymbol{M}_{22} \boldsymbol{\varGamma}_2) = \boldsymbol{0}_{\mathsf{ql}_1 \times \mathsf{il}_2} \enspace , \\ \boldsymbol{\varDelta}_1^\intercal \boldsymbol{M}_{22} \boldsymbol{\varDelta}_2 = \boldsymbol{0}_{\mathsf{ql}_1 \times \mathsf{ql}_2} \enspace . \end{gathered} \tag{5}$$

Clearly, for a fixed $\boldsymbol{\varGamma} = \boldsymbol{\gamma}$, Eq. (5) is a system of $\mathsf{il}_1 \mathsf{ql}_2 + \mathsf{il}_2 \mathsf{ql}_1 + \mathsf{ql}_1 \mathsf{ql}_2$ polynomial equations, where the sum of the total degrees of all polynomials is at most $\mathsf{il}_1 \mathsf{ql}_2 + \mathsf{il}_2 \mathsf{ql}_1 + 2 \mathsf{ql}_1 \mathsf{ql}_2$. Moreover, the system Eq. (5) has $\mathsf{ol}_1 \mathsf{ql}_1 + \mathsf{ol}_2 \mathsf{ql}_2$ indeterminates. Note also that $V^t(\boldsymbol{X}, \mathbb{Q}) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Q}_1 \end{pmatrix}^\intercal \boldsymbol{f}_{\boldsymbol{M}}(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \end{pmatrix}$.

Analysis of the TotalKE assumption. Let us assume $\mathsf{il}_1 = \mathsf{il}_2 = 1$, in particular, there are no input indeterminates $\boldsymbol{X}$. We will leave the general case for future work. Let TotalKE be the parameterized assumption that states the following: if the adversary, on input $([1]_1, [1]_2)$, outputs the specified number of group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ and the specified verification equation holds, then one can extract a linear representation of any output element with respect to the adversary's input elements. Since $\mathsf{il}_1 = \mathsf{il}_2 = 1$, the nontrivial linear relation is just the discrete logarithm of the output element.

Definition 7. *Let* $\mathsf{il}_1 = \mathsf{il}_2 = 1$, $\mathsf{ol}_1, \mathsf{ol}_2 \geq 1$, *and* $R \geq 1$. *Let* $\boldsymbol{M}[i] \in \mathbb{F}^{(\mathsf{ol}_1+\mathsf{il}_1)\times(\mathsf{ol}_2+\mathsf{il}_2)}$ *for* $i \in [1, R]$. *Let* $V_i(\mathrm{y}) = \binom{1}{\mathrm{y}_1}^\intercal \boldsymbol{M}[i] \binom{1}{\mathrm{y}_2}$. *The* $(\mathsf{ol}_1, \mathsf{ol}_2, \{V_i\}_{i=1}^R)$*-TotalKE assumption holds for* Pgen*, if for every non-uniform PPT adversary* $\mathcal{A}$*, there exists a non-uniform PPT extractor* $\mathsf{Ext}_\mathcal{A}$*, such that* $\mathsf{Adv}^{\mathrm{totalke}}_{\mathsf{Pgen},\mathsf{ol}_1,\mathsf{ol}_2,\{V_i\}_{i=1}^R,\mathcal{A},\mathsf{Ext}_\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} \boldsymbol{y} \in \mathbb{F}^{\mathsf{ol}_1} \wedge \boldsymbol{z} \in \mathbb{F}^{\mathsf{ol}_2} \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); \\ \forall i \in [1, R]. V_i(\boldsymbol{y}, \boldsymbol{z}) = 0 \wedge & ([\boldsymbol{y}]_1, [\boldsymbol{z}]_2) \leftarrow \mathcal{A}(\mathsf{p}, [1]_1, [1]_2; r); \\ (\boldsymbol{y}, \boldsymbol{z}) \neq (\boldsymbol{y}^*, \boldsymbol{z}^*) & (\boldsymbol{y}^*, \boldsymbol{z}^*) \leftarrow \mathsf{Ext}_\mathcal{A}(\mathsf{p}, [1]_1, [1]_2; r) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

We emphasize that for any choice of il_κ and ol_κ, TotalKE is secure in the AGM. The simplest TotalKE-type assumption is the SpurKE assumption, mentioned in the introduction: if $\mathcal{A}(\mathsf{p}, [1]_1)$ outputs $[x]_1$, then one can extract x. SpurKE holds in the AGM , but it is clearly false when one can sample obliviously. Thus, it is also false in the AGMOS, and in the standard model due to the existence of admissible encodings Section 2.1.

We are interested in for which choices of $(\mathsf{ol}_1, \mathsf{ol}_2, \{V_i\})$, the TotalKE assumption is secure in AGMOS, assuming both TOFR and DL holds.

Theorem 6. *Fix* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$ *such that DL is hard in each group. Fix* $\mathsf{il}_1 = \mathsf{il}_2 = 1$ *and* $\mathsf{ql}_1, \mathsf{ql}_2, \mathsf{ol}_1, \mathsf{ol}_2 \geq 1$. *If* $\boldsymbol{f}_{\boldsymbol{M}[1]}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \ldots = \boldsymbol{f}_{\boldsymbol{M}[R]}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \boldsymbol{0}$ *has a common solution* $(\boldsymbol{\gamma}, \boldsymbol{\delta})$ *such that* $\boldsymbol{\delta} \neq \boldsymbol{0}$*, then the* $(\mathsf{ol}_1, \mathsf{ol}_2, \{V_i\})$*-TotalKE assumption is* not *secure in the AGMOS. If this holds, we say* $\boldsymbol{M}$ *is* TotalKE-incompatible. *Otherwise, it is* TotalKE-compatible.

As a first step, we show that it is sufficient to consider one oracle query in both groups.

Lemma 3. *Let* $\boldsymbol{M} \in \mathbb{F}^{n\times m}$*, where* $n = \mathsf{il}_1 + \mathsf{ol}_1$ *and* $m = \mathsf{il}_2 + \mathsf{ol}_2$*. If the system in Eq.* (5) *has a solution* $(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa \times \mathsf{ql}_\kappa})_{\kappa=1}^2$ *for some* $\mathsf{ql}_1, \mathsf{ql}_2 > 1$ *with non-zero* $\boldsymbol{\delta}_1 \neq \boldsymbol{0}$*, then it has a non-zero solution* $(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}'_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa})_{\kappa=1}^2$ *with* $\boldsymbol{\delta}'_2 = \boldsymbol{0}_{\mathsf{ol}_2}$*. A dual claim holds for* $\boldsymbol{\delta}_2 \neq \boldsymbol{0}$.

Proof. For any $\boldsymbol{M} \in \mathbb{F}^{n\times m}$, let Eq. (5) hold for some $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\delta}_1 \in \mathbb{F}^{\mathsf{ol}_1\times\mathsf{ql}_1}$, $\boldsymbol{\delta}_2 \in \mathbb{F}^{\mathsf{ol}_2\times\mathsf{ql}_2}$, such that $\boldsymbol{\delta}_\kappa \neq \boldsymbol{0}$ for some $\kappa \in \{1, 2\}$. W.l.o.g., assume $\kappa = 1$. Then, $\boldsymbol{\delta}_1^{(k)} \neq \boldsymbol{0}$ for some k. Then, clearly, Eq. (5) has a non-zero solution when setting $\mathsf{ql}_1 = \mathsf{ql}_2 = 1$: the solution is $(\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}'_\kappa)_{\kappa=1}^2$ with $\boldsymbol{\delta}'_1 = \boldsymbol{\delta}_1^{(k)}$, $\boldsymbol{\delta}'_2 = \boldsymbol{0}_{\mathsf{ol}_2}$. □

Thus, in the rest of this subsection, we assume $\mathsf{ql}_\kappa = 1$; in particular, $\boldsymbol{\delta}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa}$. For the same reason, Theorem 6 can be equivalently stated for $\mathsf{ql}_1 = \mathsf{ql}_2 = 1$.

Assume the system $\boldsymbol{f}_{\boldsymbol{M}}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \mathbf{0}$ has a common solution $(\boldsymbol{\gamma}, \boldsymbol{\delta})$ where $\boldsymbol{\delta}$ is non-zero. We emphasize that $\boldsymbol{\delta}$ has different semantics than $\boldsymbol{\gamma}$, and, under TOFR and DL, recovering the discrete logarithm of $\boldsymbol{\delta}$ is hard for any party. Thus, the TotalKE extractor can only recover the discrete logarithms in the case $\boldsymbol{\delta} = \mathbf{0}$.

If such a non-zero solution exists, then from the fact that the verifier accepts, it does not follow that one can extract the discrete logarithms of all adversary's outputs. Thus, on our hypothesis, the TotalKE assumption is secure iff for any $\boldsymbol{\Gamma} = \boldsymbol{\gamma}$, $\boldsymbol{f}_{\boldsymbol{M}}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = 0$ has only a zero solution in $\boldsymbol{\Delta}$.

6.1 Classification of TotalKE-Compatible Matrices

We will use the following classic result. See Appendix D.1 for the proof.

Proposition 2 (Chevalley-Warning theorem [Che35,War35]). *Let $\mathbb{F}_q$ be a finite field of size q and characteristic p. If r polynomials $f_j \in \mathbb{F}[t_1, \ldots, t_n]$ satisfy $\sum \deg f_j < n$, then the number of common roots of f_j is divisible by p.*

As a corollary (that suffices for the current work), if the system of solutions has at least one solution, it must have another solution. In particular, if it has a zero solution ($X_1 = \ldots = X_n = 0$), then it must have a non-zero solution.

TotalKE-Incompatible Cases. Let $\deg \boldsymbol{f}_{\boldsymbol{M}[i]}$ be the sum of the degrees of all $\mathsf{il}_1\mathsf{ql}_2 + \mathsf{il}_2\mathsf{ql}_1 + 2\mathsf{ql}_1\mathsf{ql}_2$ polynomials involved in the system $\boldsymbol{f}_{\boldsymbol{M}[i]} = 0$, where we consider only $\boldsymbol{\Delta}$ as the indeterminates. For a fixed R, let $\deg \boldsymbol{f} := \sum_{i \le R} \deg \boldsymbol{f}_{\boldsymbol{M}[i]}$. Lemma 4 separates AGM and AGMOS.

Lemma 4. *Let $\mathsf{il}_1, \mathsf{il}_2 = 1$ and $\mathsf{ql}_1 = \mathsf{ql}_2 = 1$. For $i \in [1, R]$, fix* any $\boldsymbol{M}[i]$ *and the corresponding verification equation V_i. If either $\mathsf{ol}_1 > R$ or $\mathsf{ol}_2 > R$, then there exists a non-zero common solution with $\boldsymbol{\delta} \neq \mathbf{0}$. Thus, if DL holds, $(\mathsf{ol}_1, \mathsf{ol}_2, \{V_i\}_{i=1}^R)$-TotalKE is not secure in the AGMOS for* any $\boldsymbol{M}[i]$.

Proof. W.l.o.g., assume $\mathsf{ol}_1 \ge \mathsf{ol}_2$. Fix any $\boldsymbol{\gamma}$. Recall that for a *fixed* $\boldsymbol{\Gamma} = \boldsymbol{\gamma}$, $\boldsymbol{f}_{\boldsymbol{M}[i]}(\boldsymbol{\gamma}, \boldsymbol{\Delta}) = 0$ is a system of homogeneous polynomial equations of summatory degree $\mathsf{il}_1\mathsf{ql}_2 + \mathsf{il}_2\mathsf{ql}_1 + 2\mathsf{ql}_1\mathsf{ql}_2 \le 4$ in $\mathsf{ol}_1\mathsf{ql}_1 + \mathsf{ol}_2\mathsf{ql}_2 = \mathsf{ol}_1 + \mathsf{ol}_2$ variables. Since the polynomials in Eq. (5) do not have constant terms, the equation systems $\boldsymbol{f}_{\boldsymbol{M}[i]}(\boldsymbol{\gamma}, \boldsymbol{\Delta}) = 0$ have at least one common solution ($\boldsymbol{\delta} = \mathbf{0}$). By the Chevalley-Warning theorem, if $\mathsf{ol}_1 + \mathsf{ol}_2 > \deg \boldsymbol{f}(\boldsymbol{\gamma}, \boldsymbol{\Delta})$, then there must exist a non-zero common solution $\boldsymbol{\delta}$. This must hold for any $\boldsymbol{\gamma}$.

Recall $\mathsf{il}_1 = \mathsf{il}_2 = \mathsf{ql}_1 = \mathsf{ql}_2 = 1$. Thus, the system in Eq. (5) consists of three polynomials of degrees 1, 2, and 2, correspondingly. In the case $\mathsf{ol}_2 = 0$, most of the polynomials disappear and thus $\deg \boldsymbol{f}_{\boldsymbol{M}[i]} \le 1$. Hence, $\deg \boldsymbol{f} \le R$. The claim follows from the Chevalley-Warning theorem.

Assume now $\mathsf{ol}_2 \ge 1$. Hence, $\deg \boldsymbol{f}_{\boldsymbol{M}[i]} \le 4$ and $\deg \boldsymbol{f} \le 4R$. Because of that, we can immediately use the Chevalley-Warning theorem to get a non-tight solution with the requirement $\mathsf{ol}_1 + \mathsf{ol}_2 > 4R$.

For a tighter solution, we use the structure of the polynomials. Set $\boldsymbol{\delta}_2 \leftarrow \mathbf{0}$. Then, we have ol_1 remaining variables and $\deg \boldsymbol{f}_{\boldsymbol{M}[i]}(\boldsymbol{\gamma}, \boldsymbol{\Delta}_1, \mathbf{0}) \leq 1$. By the Chevalley-Warning theorem, the system of equations has a non-zero common solution in $\boldsymbol{\Delta}_1$ as soon as $\mathsf{ol}_1 > R$. Dually, by setting $\boldsymbol{\delta}_2 = \mathbf{0}$, the system of equations has a non-zero common solution as soon as $\mathsf{ol}_2 > R$.

Finally, finding a non-zero common solution is an algorithmically simple task. In all cases, we have a system of R linear equations in ol_1 variables that is guaranteed to have a nontrivial solution. This solution can be found efficiently by using Gaussian elimination. □

The Chevalley-Warning theorem is a very powerful tool that can be used in much more general cases than handled in Lemma 4. We consider its use to prove similar bounds another important contribution of the current paper.

TotalKE-Compatible Cases. Let $R = 1$. It follows from Lemma 4 that (if DL holds) TotalKE does not hold unless $\mathsf{ol}_1 \leq 1$ and $\mathsf{ol}_2 = 0$, or $\mathsf{ol}_1 + \mathsf{ol}_2 \leq 2$ and $\mathsf{ol}_2 \geq 1$. That is, either $\mathsf{ol}_1 = 1$ and $\mathsf{ol}_2 = 0$ (the case $\mathsf{ol}_1 = \mathsf{ol}_2 = 0$ is vacuous) or $\mathsf{ol}_1 = 1$ and $\mathsf{ol}_2 = 1$. (The case $\mathsf{ol}_1 = 0$ and $\mathsf{ol}_2 = 1$ is dual.)

In the rest of this section, we will give a list of all TotalKE-compatible matrices in the case of a single verification equation. We will leave the case of $\mathsf{il}_1 > 1$ or $\mathsf{il}_2 > 1$ or $R > 1$ for future work.

Lemma 5. *Let $\mathsf{il}_\kappa = \mathsf{ql}_\kappa = 1$ and $R = 1$ (thus there is a single matrix $\boldsymbol{M}$). Assume that DL holds.*

1. *Let $\mathsf{ol}_1 = 1$ and $\mathsf{ol}_2 = 0$. Then $\boldsymbol{M}$ is TotalKE-compatible iff $M_{21} \neq 0$. Thus, the only possibly secure TotalKE assumption involves verification equation $[y]_1 \bullet [1]_2 = M_{11}[1]_1 \bullet [1]_2$ for M_{11} chosen by the verifier.*
2. *Let $\mathsf{ol}_1 = 1$ and $\mathsf{ol}_2 = 1$. Then $\boldsymbol{M}$ is TotalKE-compatible iff $M_{22} = 0$ and either $M_{21} \neq 0$ or $M_{12} \neq 0$. Thus, the only possibly secure TotalKE assumption involves verification equation $M_{12}[y]_1 \bullet [1]_2 + [1]_1 \bullet M_{21}[z]_2 = -M_{11}[1]_1 \bullet [1]_2$ for non-zero M_{12} or M_{21}, where the verifier chooses M_{12}, M_{21}, and M_{11}.*

Proof. We recall from Theorem 6 that $\boldsymbol{M}$ is TotalKE-incompatible iff $\boldsymbol{f}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = 0$ has a solution $(\boldsymbol{\gamma}, \boldsymbol{\delta})$ such that $\boldsymbol{\delta}$ is non-zero.

(Item 1). Since $\mathsf{il}_1 = \mathsf{ol}_1 = \mathsf{ql}_1 = 1$, $\boldsymbol{\gamma}_1 = \gamma_1 \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{il}_1} = \mathbb{F}$ and $\boldsymbol{\delta}_1 = \delta_1 \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{ql}_1} = \mathbb{F}$. Since $\mathsf{ol}_2 = 0$, there is no $\boldsymbol{\delta}_2$ and thus the system Eq. (5) consists of only one polynomial, $f(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \varDelta_1 M_{21}$. Thus, $f(\boldsymbol{\gamma}, \boldsymbol{\delta}) = 0$ iff $\delta_1 M_{21} = 0$. This has a non-zero solution $\delta_1 \neq 0$ iff $M_{21} = 0$. Then, $\boldsymbol{M} = \left(\begin{smallmatrix} M_{11} \\ 0 \end{smallmatrix}\right)$ for some $M_{11} \in \mathbb{F}$. The claim follows.

(Item 2). Then γ_1, γ_2, δ_1, and δ_2 have dimension one. In this case, the equation $f(\boldsymbol{\gamma}, \boldsymbol{\delta}) = 0$ in Eq. (5) simplifies to

$$\delta_2(\gamma_1 M_{22} + M_{12}) = 0 \ , \qquad \delta_1(\gamma_2 M_{22} + M_{21}) = 0 \ , \qquad \delta_1 \delta_2 M_{22} = 0 \ .$$

For this to have a non-zero solution in (δ_1, δ_2), we need that, say, $\delta_1 \neq 0$. From the second equation, we then get $M_{21} = -\gamma_2 M_{22}$. Hence, there are only zero solutions iff $M_{22} = 0$ and $M_{21} \neq 0$ (that is, $\boldsymbol{M} = \left(\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & 0 \end{smallmatrix}\right)$ for $M_{21} \neq 0$);

in every other case, one can choose γ_2 that makes the equation hold. Choosing $\delta_2 = 0$ means no other restrictions exist.

The case $\delta_2 \neq 0$ is dual. The claim follows since $\left(\begin{smallmatrix} 1 \\ z \end{smallmatrix}\right)^\top \left(\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 \\ y \end{smallmatrix}\right) = M_{11} + M_{12}y + M_{21}z$ and we either need $M_{21} \neq 0$ or $M_{12} \neq 0$. □

Clearly, the result makes intuitive sense. For example, the verification equation in Item 2 includes one unknown group element in both groups and some constants. Since $\mathbb{Q}_{\kappa i}$ is only available in $\mathbb{G}_\kappa$, they cancel out, and thus $\boldsymbol{\delta} = \mathbf{0}$.

We will give some concrete examples in Appendix D.2.

7 AGMOS with Uniform Oracle

We propose a simplification of the general AGMOS, where the oracle produces uniformly random group elements. We call this AGM with Uniform Oblivious Sampling (AGMUOS). GGM with uniform sampling is known [BFS16,ABLZ17], and Lipmaa's variant of AGMOS [Lip22] also focused on uniform sampling.

Note that AGMUOS with uniform sampling does not accurately model, for example, admissible encodings. Since a noticeable fraction of the group is not in the image of an admissible encoding, outputs of standard admissible encodings are easily distinguishable from uniformly random group elements. Nevertheless, the uniform model is easier to state, will rely on a weaker assumption, and is still helpful as a predictor for the security of assumptions. In fact, we are unaware of any assumption that can be proven secure in the uniform model but is insecure in $(\mathcal{EF}, \mathcal{DF})$-AGMOS when $(\mathcal{EF}, \mathcal{DF})$-TOFR holds. Moreover, the standard security proof approach for AGMOS (such as in Section 5) carries over to AGMUOS, with the only difference being the underlying assumption.

We define AGMOS with uniform sampling in the pairing-based setting, just as the general model. Let $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$ be the description of the pairing. We define a uniform sampling oracle $\mathcal{U}$ that takes as an input $\kappa \in \{1, 2\}$ and returns a uniformly random group element $[\mathbb{q}]_\kappa \leftarrow\!\$\ \mathbb{G}_\kappa$. Importantly, we assume that $\mathcal{U}$ is non-programmable. That is, security reductions cannot modify outputs of $\mathcal{U}$. Besides that, the model is almost identical. For the sake of completeness, we state the complete definition below.

Definition 8 (AGMUOS). *A non-uniform PPT algorithm $\mathcal{A}$ is an* AGMUOS adversary *for* Pgen *if there exists a non-uniform PPT extractor* $\mathsf{Ext}_\mathcal{A}$*, such that for any* $\mathbb{x} = (\mathbb{x}_1, \mathbb{x}_2)$*,* $\mathsf{Adv}^{\mathrm{agmuos}}_{\mathsf{Pgen},\mathcal{A},\mathsf{Ext}_\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l} \mathbb{y}_1 \neq \boldsymbol{\gamma}_1^\top \mathbb{x}_1 + \boldsymbol{\delta}_1^\top \mathbb{q}_1 \vee \\ \mathbb{y}_2 \neq \boldsymbol{\gamma}_2^\top \mathbb{x}_2 + \boldsymbol{\delta}_2^\top \mathbb{q}_2 \end{array} \middle| \begin{array}{l} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); \\ ([\mathbb{y}_1]_1, [\mathbb{y}_2]_2) \leftarrow\!\$\ \mathcal{A}^{\mathcal{U}}(\mathsf{p}, \mathbb{x}; r); \\ (\boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2 \leftarrow \mathsf{Ext}_\mathcal{A}^{\mathcal{U}}(\mathsf{p}, \mathbb{x}; r) : \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

Here, $[\mathbb{q}_\kappa]_\kappa$ *is the tuple of elements output by* $\mathcal{U}$ *on input* $\kappa \in \{1, 2\}$*. We denote by* ql_κ *the number of* $\mathcal{U}$ *calls on input* κ*.*

As mentioned before, proofs are essentially identical in AGMUOS, except we can rely on a weaker version of TOFR. We define a uniform oracle version of the TOFR assumption that we call Uniform Tensor Oracle FindRep (UTOFR).

Definition 9 (UTOFR). *We say that* Pgen *is* UTOFR *(*Uniform Tensor Oracle FindRep*) secure if for any non-uniform PPT* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathrm{utofr}}_{\mathsf{Pgen},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\boldsymbol{v} \neq \mathbf{0} \wedge \boldsymbol{v}^\intercal \cdot \begin{pmatrix} 1 \\ \mathbb{q}_1 \\ \mathbb{q}_2 \\ \mathbb{q}_1 \otimes \mathbb{q}_2 \end{pmatrix} = 0 \,\middle|\, \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{v} \leftarrow \mathcal{A}^{\mathcal{U}}(\mathsf{p}) \right] = \mathsf{negl}(\lambda) \enspace .$$

Here, $\mathcal{U}$*,* $\mathbb{q}_1$*, and* $\mathbb{q}_2$ *are as in Definition 8.*

In fact, if we did allow programming $\mathcal{U}$, then this assumption could be reduced to the (1, 1)- PDL assumption. Although we do not allow programming in general, it does indicate that UTOFR is a relatively weak assumption.

Lemma 6. *Suppose* $\mathcal{U}$ *is programmable. Then,* $(1, 1)$*-PDL implies* UTOFR*. More formally, for any non-uniform PPT* $\mathcal{A}$*, there exist a non-uniform PPT* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathrm{utofr}}_{\mathsf{Pgen},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{pdl}}_{1,1,\mathsf{Pgen},\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda)$*.*

Proof. The discrete logarithm adversary $\mathcal{B}$ runs $\mathcal{A}$ while embedding its challenge $([y]_1, [y]_2)$ to the queries of the oracle $\mathcal{U}$. On a query κ to $\mathcal{U}$, $\mathcal{B}$ samples $r_{\kappa,i}, s_{\kappa,i} \leftarrow\!\!\$\ \mathbb{F}$ and returns $[\mathbb{q}_{\kappa i}]_\kappa \leftarrow r_{\kappa,i}[1]_\kappa + s_{\kappa i}[y]_\kappa$. If $\mathcal{A}$ succeeds, it returns a non-zero representation $\boldsymbol{v}$, such that $\boldsymbol{v}^\intercal \mathbb{q} \neq 0$. Let us denote $\boldsymbol{v} = (v_0 // \boldsymbol{v}_1 // \boldsymbol{v}_2 // \boldsymbol{v}_3)$, where subvectors correspond to 1, $\mathbb{q}_1$, $\mathbb{q}_2$ and $\mathbb{q}_1 \otimes \mathbb{q}_2$ respectively. Then,

$$\begin{aligned} 0 =& v_0 + \boldsymbol{v}_1^\intercal \mathbb{q}_1 + \boldsymbol{v}_2^\intercal \mathbb{q}_2 + \boldsymbol{v}_3^\intercal (\mathbb{q}_1 \otimes \mathbb{q}_2) \\ =& v_0 + \boldsymbol{v}_1^\intercal (\boldsymbol{r}_1 + \boldsymbol{s}_1 y) + \boldsymbol{v}_2^\intercal (\boldsymbol{r}_2 + \boldsymbol{s}_2 y) + \boldsymbol{v}_3^\intercal ((\boldsymbol{r}_1 + \boldsymbol{s}_1 y) \otimes (\boldsymbol{r}_2 + \boldsymbol{s}_2 y)) \\ =& v_0 + \boldsymbol{v}_1^\intercal \boldsymbol{r}_1 + \boldsymbol{v}_2^\intercal \boldsymbol{r}_2 + \boldsymbol{v}_3^\intercal (\boldsymbol{r}_1 \otimes \boldsymbol{r}_2) \\ &+ (\boldsymbol{v}_1^\intercal \boldsymbol{s}_1 + \boldsymbol{v}_2^\intercal \boldsymbol{s}_2 + \boldsymbol{v}_3^\intercal (\boldsymbol{r}_1 \otimes \boldsymbol{s}_2 + \boldsymbol{s}_1 \otimes \boldsymbol{r}_2)) \cdot y + \boldsymbol{v}_3^\intercal (\boldsymbol{s}_1 \otimes \boldsymbol{s}_2) \cdot y^2 \enspace . \end{aligned}$$

We can view this as a quadratic equation in y. If the coefficient of y or y^2 is non-zero, then $\mathcal{B}$ can solve the equation for y and break the PDL assumption.

Let us analyze the probability of that happening. Note that $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$ cannot be zero vectors at the same time since then also $v_0 = 0$, which implies that the whole vector $\boldsymbol{v}$ is a zero-vector.

The vectors $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are information-theoretically hidden from the adversary (they are blinded by $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ respectively). The probability that $\boldsymbol{s}_1$ or $\boldsymbol{s}_2$ contain a zero element is bounded by $(\mathsf{ql}_1 + \mathsf{ql}_2)/\mathbb{F}$. Additionally, if $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ do not contain a zero, then neither does $\boldsymbol{s}_1 \otimes \boldsymbol{s}_2$. Let us suppose that this is the case. Now, we look at two cases.

1) Suppose $\boldsymbol{v}_3 \neq \mathbf{0}$. Then, according to Schwartz-Zippel lemma, the probability that $\boldsymbol{v}_3^\intercal (\boldsymbol{s}_1 \otimes \boldsymbol{s}_2) = 0$ is bounded by $2/\mathbb{F}^{\mathsf{ql}_1 + \mathsf{ql}_2}$.

2) Suppose $\boldsymbol{v}_3 = \mathbf{0}$, but $\boldsymbol{v}_1$ or $\boldsymbol{v}_2$ are non-zero. Then the coefficient of y is $\boldsymbol{v}_1^\intercal \boldsymbol{s}_1 + \boldsymbol{v}_2^\intercal \boldsymbol{s}_2$. The probability that this coefficient is 0 is bounded by $1/\mathbb{F}^{\mathsf{ql}_1 + \mathsf{ql}_2}$.

Thus, except for negligible probability, $\mathcal{B}$ breaks $(1, 1)$-PDL. □

8 GGM with Oblivious Sampling And TOFR

Next, we will cryptanalyze TOFR using GGM with oblivious sampling (GGMOS), a more realistic version of the GGM with hashing (GGMH, [Bro01,BFS16,ALSZ21]). We base GGMOS on Shoup's GGM [Sho97] in the bilinear setting.

8.1 GGM with Oblivious Sampling

Let us recall the random representation (RR)/Shoup's GGM model, as defined in [Sho97,Zha22]. Let $\mathbf{S} \subseteq \{0,1\}^*$ be a set of binary strings of cardinality at least p. For $\kappa \in \{1,2,T\}$, let $\iota_\kappa : \mathbb{F}_p \to \mathbf{S}$ be three random injections, called labelling functions. Intuitively $[x]_\kappa = \iota_\kappa(x)$. All parties can make the following queries to the group oracle:

1. **Labelling:** The party submits (x, κ), where $x \in \mathbb{Z}_p, \kappa \in \{1,2,T\}$ and receives the string $\iota_\kappa(x)$.
2. **Group operation:** The party submits $(s_1, s_2, \alpha_1, \alpha_2, \kappa)$. The group oracle checks if there exists x_1 and x_2 such that $\iota_\kappa(x_i) = s_i$, for $i \in \{1,2\}$. If yes, it returns $\iota_\kappa(\alpha_1 x_1 + \alpha_2 x_2)$ and otherwise returns $\perp$.
3. **Pairing operation:** The party submits $(s_1, s_2, \alpha_1, \alpha_2)$. The group oracle checks if there exists x_1 and x_2 such that $\iota_1(x_1) = s_1$, and $\iota_2(x_2) = s_2$. If yes, it returns $\iota_T(\alpha_1 x_1 \alpha_2 x_2)$ and otherwise returns $\perp$.

GGMH gives the adversary an additional operation to create uniformly random group elements. With GGMOS, we will go a step further, and just like with AGMOS, we will allow oblivious sampling from potentially non-uniform distributions chosen by the adversary. More precisely, fix $\mathcal{OF} = \{\mathcal{OF}_\mathsf{p}\}$: a family of distributions over $\mathbb{F}_p$.

4. **Oblivious sampling:** The party submits (D, κ), where $D \in \mathcal{OF}_\mathsf{p}$ is a distribution and $\kappa \in \{1,2\}$. The oracle samples $x \leftarrow\!\!\$\ D$ and returns $\iota_\kappa(x)$.

All operations have unit cost.

We briefly explain why oblivious sampling is defined here differently compared to AGMOS. Suppose E is an encoding function and D is its input distribution, such as in AGMOS. Since the labeling function's representation is hidden from the adversary, how to model E in generic groups is unclear. On the one hand, if we try to model E by defining it from $\mathbb{F}_p$ to $\mathbf{S}$, then there is no relation between the seed s and the exponent x, such that $E(s) = [x]_\kappa$. On the other hand, we could define E as a function from $\mathbb{F}_p$ to $\mathbb{F}_p$ and use the labeling function on the evaluation to sample group elements. However, then we are back in the case depicted above, where we only need distributions over $\mathbb{F}_p$. In fact, the generic adversary does not see the encoding input. Thus an oblivious sampling operation where the party submits a function $E : \mathbb{F}_p \to \mathbb{F}_p$ and a distribution D is equivalent to an oblivious sampling operation, where the party submits the distribution defined by $E(D)$.

8.2 TOFR Security in the GGMOS

We prove that TOFR holds against *generic* adversaries in GGMOS, allowing sampling from any well-spread distribution.

We will need the following min-entropy version of the Schwartz-Zippel lemma.

Proposition 3 ([GV13]). *Let $F \in \mathbb{F}[X_1, \dots, X_m]$ be a non-zero polynomial of (total) degree at most d. Let D_i $(i = 1, \dots, m)$ be probability distributions on $\mathbb{F}$ such that $\mathbf{H}_\infty(D_i) \geq \log p - \tau$, where $0 \leq \tau \leq \log p$. If $x_i \leftarrow\!\!\$\ D_i$, $i \in [1, m]$, are chosen independently, then* $\Pr[F(x_1, \dots, x_m) = 0] \leq 2^\tau \cdot d/p$.

We will use Proposition 3 with the setting $\tau = \log p - \omega(\log \lambda)$. In this case, if $\mathbf{H}_\infty(D_i) \geq \omega(\log \lambda)$ then $\Pr[F(x_1, \ldots, x_m) = 0] \leq 2^{\log p - \omega(\log \lambda)} \cdot d/p = d/\lambda^{\omega(1)}$.

For each distribution D, and function E, we define the distribution $E(D)$, defined over the image of E as the one induced by first sampling $s \leftarrow\!\!\$\; D$ and then evaluating $E(s)$.

Before proving its security, we describe how to interpret TOFR in GGMOS. TOFR depends on a family of functions $\mathcal{EF}_{\mathsf{p},\kappa} = \{E : \mathbb{F} \rightarrow \mathbb{G}_\kappa\}$. As already argued, such functions seem incompatible with generic groups. For each function $E \in \mathcal{EF}_{\mathsf{p},\kappa}$, we can define E' as $E'(s) = x$, where $E(s) = [x]_\kappa$. We argue that it is sufficient to parametrize TOFR only with distributions to analyze its security against generic adversaries. In fact, let $\mathcal{OF} = \{\mathcal{OF}_{\mathsf{p}}\}$ be the family of distributions defined by $E'(D)$ for each $E \in \mathcal{EF}_{\mathsf{p},\kappa}, D \in \mathcal{DF}_{\mathsf{p}}$. When a TOFR adversary in the standard model queries the oracle with (E, D), the generic adversary will query it with the distribution $E'(D)$. Note that even if E' is not efficiently computable, we assume that oblivious sampling operations have unitary cost. Let $\{\mathcal{I}\}$ be the identity function. Then the $(\mathcal{EF}, \mathcal{DF})$-TOFR assumption and the $(\{\mathcal{I}\}, \mathcal{OF})$-TOFR are equivalent in GGMOS.

Theorem 7. *Fix* Pgen. *Let* $\mathcal{DF} = \{\mathcal{DF}_{\mathsf{p}}\}$ *be a family of well-spread distributions. Let* $\mathcal{EF} = \{\mathcal{EF}_{\mathsf{p},\kappa}\}$ *be a family of functions such that, if* $D \in \mathcal{DF}_{\mathsf{p}}$ *is a well-spread distribution, then, for each* $E \in \mathcal{EF}_{\mathsf{p},\kappa}$ *the distribution* $E'(D)$ *is well-spread. Suppose that the identity function is included in* $\mathcal{EF}$. *Let* $\mathcal{OF} = \{\mathcal{OF}_{\mathsf{p}}\}$ *be the family of functions defined by* $E'(D)$ *for each* $E \in \mathcal{EF}_{\mathsf{p},\kappa}, D \in \mathcal{DF}_{\mathsf{p}}$. *The* $(\{\mathcal{I}\}, \mathcal{OF})$-TOFR *assumption holds for* Pgen *against* $\mathcal{OF}$*-GGMOS adversaries that execute* $o(\sqrt{2^{\chi_{\mathsf{p}}}})$ *group operations, where* $\chi_{\mathsf{p}} := \min_{D \in \mathcal{OF}_{\mathsf{p}}} \mathbf{H}_\infty(D)$.

Proof. Let $\mathcal{A}$ be a $\mathcal{OF}$-GGMOS adversary against $(\{\mathcal{I}\}, \mathcal{OF})$-TOFR assumption (see Definition 2) that, given $(\mathsf{p}, [1]_1, [1]_2)$ and access to the group oracle, outputs a vector $\boldsymbol{v} \neq \mathbf{0}$, such that $\boldsymbol{v}^\intercal \cdot (1, \mathbb{q}_1^\intercal, \mathbb{q}_2^\intercal, \mathbb{q}_1^\intercal \otimes \mathbb{q}_2)^\intercal = 0$.

Here, $\mathbb{q}_{\kappa i}$ are the exponents associated to the strings returned by the TOFR oracle. Thus, the TOFR oracle on input (D, κ) for the i_κ-th oracle call, where $D \in \mathcal{OF}_{\mathsf{p},\kappa}$, queries the group oracle for an oblivious sampling operation on input (D, κ), and then receives and forwards to the adversary the string $\iota_\kappa(\mathbb{q}_i)$.

Assume that $\mathcal{A}$ makes ql queries to the TOFR oracle. Since we work in the GGMOS, after T computational steps, the adversary has strings of at most T elements, including the oracle answers. We can consider each element as the evaluation of T known but, w.l.o.g., different, ql-variate, total degree-≤ 2 polynomials F_i in $\mathbb{Q}$. Formally this procedure corresponds to a hybrid argument defined by the following group oracle.

1. **Labelling queries:** The party submits (x, κ), where $x \in \mathbb{Z}_p, \kappa \in \{1, 2, T\}$. If $\iota_\kappa(x)$ has already been defined, the oracle returns it. Otherwise, $s \leftarrow\!\!\$\; \mathbf{S}$, defines $\iota_\kappa(x) = s$, and the oracle returns s.
2. **Group operations:** The party submits $(s_1, s_2, \alpha_1, \alpha_2, \kappa)$. The group oracle checks if there exists polynomials $\chi_1(X)$ and $\chi_2(X)$ such that $\iota_\kappa(\chi_2(X)) = s_i$, for $i \in \{1, 2\}$. If this is not the case it returns $\perp$. If yes, it checks if $\iota_\kappa(\alpha_1 \chi_1(X) + \alpha_2 \chi_2(X))$ has already been assigned and returns it. Otherwise

it samples a random element in $\mathbf{S}$, defines its label as $\iota_\kappa(\alpha_1\chi_1(X)+\alpha_2\chi_2(X))$ and returns the random string.
3. **Pairing operations:** The party submits $(s_1, s_2, \alpha_1, \alpha_2)$. The group oracle checks if there exists $\chi_1(X)$ and $\chi_2(X)$ such that $\iota_1(\chi_1(X)) = s_1$ and $\iota_2(\chi_2(X)) = s_2$. If yes, it returns $\iota_T(\alpha_1\chi_1(X)\alpha_2\chi_2(X))$ and otherwise returns $\perp$.
4. **Oblivious sampling operations:** The party submits (D, κ), where $D \in \mathcal{OF}_\mathsf{p}$ is a distribution and $\kappa \in \{1, 2\}$. In its internal state, the oracle saves the distribution D associated with this query. Then, it defines a new indeterminate $\mathbb{Q}$, $s \leftarrow\!\!\$\, \mathbf{S}$, and defines $\iota_\kappa(\mathbb{Q}) = s$. Finally, it returns s.

In this case, injections ι_κ are defined "lazily" by sampling random strings on the fly when the adversary queries the oracle for group elements associated with unseen polynomials. At the end of the execution, the oracle samples variables $\mathbb{q}_{\kappa i} \leftarrow\!\!\$\, D_{\kappa i}$, where the distribution $D_{\kappa i}$ is the one received in the (κ, i)-th query.

The adversary can only win if $F_i(\mathbb{q}) = F_j(\mathbb{q})$ for two distinct polynomials F_i and F_j. Fix $i \neq j$. We define $\tau := \log p - \chi_\mathsf{p}$, then $2^\tau \cdot (2/p) = 2/2^{\chi_\mathsf{p}}$. By applying Proposition 3, the probability that $F_i(\mathbb{q}) - F_j(\mathbb{q}) = 0$ is $\leq 2/2^{\chi_\mathsf{p}}$. Thus, the probability that $F_i(\mathbb{q}) = F_j(\mathbb{q})$ for any i, j is bounded by $(2T^2)/2^{\chi_\mathsf{p}}$. Thus, $\mathcal{A}$ succeeds with probability bounded away from 0 by a constant only if the number of group operations is at least $T = \Omega(\sqrt{2^{\chi_\mathsf{p}}})$. □

Clearly, the bound $T = \Omega(\sqrt{2^{\chi_\mathsf{p}}})$ is precise. This bound explains why we require that $\mathcal{DF}$ consist of well-spread distributions.

In fact, this section aims to argue why the restriction on considering well-spread distributions looks necessary. The reader should interpret the result in this section as a piece of evidence that families $(\mathcal{EF}, \mathcal{DF})$ should be chosen such that any combination of them induces a well-spread distribution. Otherwise, there could be some generic attack the adversary can perform to recover the discrete logarithm of an element returned by the oracle. Clearly, the previous condition is not sufficient. The exponentiation that associates x to $[x]_\kappa$ should be excluded from $\mathcal{EF}$, despite mapping well-spread distributions into well-spread distributions. To add a function E, that is not an admissible encoding in $\mathcal{EF}$, one should prove a result similar to Theorem 1, showing that recovering the discrete logarithm of $E(s)$ on input s, where s is from any well-spread distribution, is as hard as the standard DL problem.

8.3 On Well-Spreadness

In Theorem 7, we obtain a GGM lower bound of $\sqrt{2^{\chi_\mathsf{p}}}$ for the time to break $(\mathcal{EF}, \mathcal{DF})$-TOFR. Importantly, this lower bound only depends on $\chi_\mathsf{p} = \min_{D \in \mathcal{DF}_\mathsf{p}} \mathbf{H}_\infty(D)$. Thus, if we only aim for polynomial security against generic adversaries, we can choose $\mathcal{DF}$ arbitrarily, as long as $\chi_\mathsf{p} = \omega(\log \lambda)$, i.e., all distributions in $\mathcal{DF}$ are well-spread.

Next, we will discuss some ramifications of the requirement that $\chi_\mathsf{p} = \min_{D \in \mathcal{DF}_\mathsf{p}} \mathbf{H}_\infty(D)$. We also discuss some possibilities of extending the AGMOS. We will leave the formalization of this discussion for future work.

If D is an arbitrary well-spread distribution on $\mathbb{G}_\kappa$, then the best-known algorithms for computing discrete logarithms of elements from D take time $|\mathrm{Im}(D)|$. When adding some structure to the distribution, it *might* be possible to apply baby-step-giant-step or Pollard's rho algorithms, with the computational complexity of approximately $\tilde{O}(\sqrt{|\mathrm{Im}(D)|})$. In particular, it is known how to solve discrete logarithms on intervals in time $(1.661 + o(1))\sqrt{|\mathrm{Im}(D)|}$ [GPR13]. Since well-spread distributions can have image size exponential in λ, it seems that computing discrete logarithms for (worst-case) well-spread distributions is an intractable problem.

The situation is different when we consider *non-uniform* adversaries. Recall that the mode mode_D is the most common output of the distribution D. Since the AGMOS extractor $\mathsf{Ext}_\mathcal{A}$ is existential and non-uniform, we can hardwire to its code the discrete logarithm of $E(\mathsf{mode}_D)$ for each distribution D actually queried by the adversary $\mathcal{A}$. (This does not work if D can depend on the random coins or the input of $\mathcal{A}$.) If the distribution is non-well-spread, then with a non-negligible probability, the input of E equals the mode. Thus the constructed non-uniform extractor succeeds in returning the hardwired discrete logarithm of the admissible encoding's output.

Hence, if we consider *weak extractors* [CD09] (i.e., extractors that succeed with some non-negligible probability[7]), then we can also handle the case where $\mathcal{DF}$ contains some non-well-spread distributions. Since here, with a non-negligible probability, $\mathsf{Ext}_\mathcal{A}$ returns the discrete logarithm as a part of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ (i.e., not as $[\mathbb{q}]_\kappa$), there is no contradiction with Theorem 7. The resulting model is sufficiently different from the AGMOS; thus, we leave precise modeling for future work.

The previous discussion is why, differently from [FKL18], we do not ask the adversary $\mathcal{A}$ itself to output values like $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, but assume the existence of a *non-uniform* extractor that does so. That is, we allow for the case that $\mathcal{A}$ itself may not know the discrete logarithms, but we can construct a non-uniform extractor (who has some values hardwired) that knows them. Importantly, weak extractors are sufficient in most non-tight security proofs, reducing the advantage of the reduction only polynomially. Alternatively, as in the DAGM, [RS20], we could require that $\mathcal{A}$ itself returns $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ but only with some probability $1/\mathsf{poly}(\lambda)$.

Acknowledgment. We thank Markulf Kohlweiss and anonymous reviewers for helpful comments.

References

ABLZ17. Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion-resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 3–33.

[7] More precisely, in this case we assume that if $\mathcal{A}$ succeeds with some probability ε, then $\mathsf{Ext}_\mathcal{A}$ succeeds with the probability $\varepsilon/\mathsf{poly}(\lambda)$

Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70700-6_1.

ALSZ21. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On subversion-resistant SNARKs. *Journal of Cryptology*, 34(3):17, July 2021. doi:10.1007/s00145-021-09379-y.

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005. doi:10.1007/11426639_26.

BCI$^+$10. Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 237–254. Springer, Heidelberg, August 2010. doi:10.1007/978-3-642-14623-7_13.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_13.

BFL20. Balthazar Bauer, Georg Fuchsbauer, and Julian Loss. A classification of computational assumptions in the algebraic group model. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 121–151. Springer, Heidelberg, August 2020. doi:10.1007/978-3-030-56880-1_5.

BFS16. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_26.

Boy08. Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, Heidelberg, September 2008. doi:10.1007/978-3-540-85538-5_3.

Bra94. Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 302–318. Springer, Heidelberg, August 1994. doi:10.1007/3-540-48329-2_26.

Bro01. Daniel R. L. Brown. The exact security of ECDSA. Contributions to IEEE P1363a, January 2001. http://grouper.ieee.org/groups/1363/.

CD09. Ran Canetti and Ronny Ramzi Dakdouk. Towards a theory of extractable functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 595–613. Springer, Heidelberg, March 2009. doi:10.1007/978-3-642-00457-5_35.

CFF$^+$20. Matteo Campanelli, Antonio Faonio, Dario Fiore, Anaïs Querol, and Hadrián Rodríguez. Lunar: a toolbox for more efficient universal and updatable zkSNARKs and commit-and-prove extensions. Cryptology ePrint Archive, Report 2020/1069, 2020. https://eprint.iacr.org/2020/1069.

CFF$^+$21. Matteo Campanelli, Antonio Faonio, Dario Fiore, Anaïs Querol, and Hadrián Rodríguez. Lunar: A toolbox for more efficient universal and updatable zkSNARKs and commit-and-prove extensions. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92078-4_1.

Che35. Claude Chevalley. Démonstration d'une hypothèse de M. Artin. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 11:73—-75, 1935. In French.

CHM+20. Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 738–768. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45721-1_26.

Dam92. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992. doi:10.1007/3-540-46766-1_36.

Den02. Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 100–109. Springer, Heidelberg, December 2002. doi:10.1007/3-540-36178-2_6.

DFGK14. George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, December 2014. doi:10.1007/978-3-662-45611-8_28.

Fis00. Marc Fischlin. A note on security proofs in the generic model. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 458–469. Springer, Heidelberg, December 2000. doi:10.1007/3-540-44448-3_35.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96881-0_2.

FLR+10. Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 303–320. Springer, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_18.

FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_3.

FT10. Pierre-Alain Fouque and Mehdi Tibouchi. Estimating the size of the image of deterministic hash functions to elliptic curves. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *LATINCRYPT 2010*, volume 6212 of *LNCS*, pages 81–91. Springer, Heidelberg, August 2010.

GPR13. Steven D. Galbraith, John M. Pollard, and Raminder S. Ruprai. Computing discrete logarithms in an interval. *Math. Comput.*, 82(282):1181–1195, 2013.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_19.

GT21. Ashrujit Ghoshal and Stefano Tessaro. Tight state-restoration soundness in the algebraic group model. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 64–93, Virtual Event, August 2021. Springer, Heidelberg. doi:10.1007/978-3-030-84252-9_3.

GV13. David Galindo and Srinivas Vivek. A practical leakage-resilient signature scheme in the generic group model. In Lars R. Knudsen and Huapeng Wu, editors, *SAC 2012*, volume 7707 of *LNCS*, pages 50–65. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-35999-6_4.

GWC19. Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. https://eprint.iacr.org/2019/953.

Ica09. Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 303–316. Springer, Heidelberg, August 2009. doi:10.1007/978-3-642-03356-8_18.

JR10. Tibor Jager and Andy Rupp. The semi-generic group model and applications to pairing-based cryptography. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 539–556. Springer, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_31.

KMSV21. Markulf Kohlweiss, Mary Maller, Janno Siim, and Mikhail Volkhov. Snarky ceremonies. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 98–127. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92078-4_4.

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. doi:10.1007/978-3-642-17373-8_11.

Lip12. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_10.

Lip19. Helger Lipmaa. Simulation-Extractable ZK-SNARKs Revisited. Technical Report 2019/612, IACR, May 31, 2019. https://ia.cr/2019/612, updated on 8 Feb 2020.

Lip22. Helger Lipmaa. A unified framework for non-universal SNARKs. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 553–583. Springer, Heidelberg, March 2022. doi:10.1007/978-3-030-97121-2_20.

LPS23. Helger Lipmaa, Roberto Parisella, and Janno Siim. Algebraic Group Model with Oblivious Sampling. In Guy Rothblum and Hoeteck Wee, editors, *TCC 2023*, volume ? of *LNCS*, pages ?–?, Taipei, Taiwan, Nov 29–Dec 2 2023. Springer, Cham. doi:?

LSZ22. Helger Lipmaa, Janno Siim, and Michal Zajac. Counting vampires: From univariate sumcheck to updatable ZK-SNARK. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 249–278. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22966-4_9.

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.

MRH04. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Heidelberg, February 2004. doi:10.1007/978-3-540-24638-1_2.

MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_27.

Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

Nie02. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126. Springer, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_8.

Rot22. Lior Rotem. Revisiting the Uber Assumption in the Algebraic Group Model: Fine-Grained Bounds in Hidden-Order Groups and Improved Reductions in Bilinear Groups. In Dana Dachman-Soled, editor, *ITC 2022*, volume 230 of *LIPIcs*, pages 13:1–13:13, Cambridge, MA, USA, July 5–7 2022. doi:10.4230/LIPIcs.ITC.2022.13.

RS20. Lior Rotem and Gil Segev. Algebraic distinguishers: From discrete logarithms to decisional uber assumptions. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 366–389. Springer, Heidelberg, November 2020. doi:10.1007/978-3-030-64381-2_13.

Sch91. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991. doi:10.1007/BF00196725.

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. doi:10.1007/3-540-69053-0_18.

SPMS02. Jacques Stern, David Pointcheval, John Malone-Lee, and Nigel P. Smart. Flaws in applying proof methodologies to signature schemes. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 93–110. Springer, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_7.

SW06. Andrew Shallue and Christiaan Van De Woestijne. Construction of Rational Points on Elliptic Curves over Finite Fields. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *ANTS 2006*, volume 4076 of *LNCS*, pages 510–524, Berlin, Germany, 23–28 July 2006. Springer, Heidelberg. ISBN 3-540-67695-3.

War35. Evald Warning. Bemerkung zur vorstehenden Arbeit von Herrn Chevalley. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 11:76—-83, 1935. In German.

WB19. Riad S. Wahby and Dan Boneh. Fast and simple constant-time hashing to the BLS12-381 elliptic curve. *IACR TCHES*, 2019(4):154–179, 2019. https://tches.iacr.org/index.php/TCHES/article/view/8348. doi:10.13154/tches.v2019.i4.154-179.

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. doi:10.1007/978-3-031-15982-4_3.

ZZK22. Cong Zhang, Hong-Sheng Zhou, and Jonathan Katz. An Analysis of the Algebraic Group Model. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022*, volume 13792 of *LNCS*, pages 310–322, Taipei, Taiwan, December 5–9, 2022. Springer, Cham. doi:10.1007/978-3-031-22966-4_9.

A Supplementary Materials for Section Section 2 (Preliminaries)

A.1 More on Admissible Encodings

Boneh and Franklin [BF01] considered supersingular curves, i.e., elliptic curves E that have exactly $q + 1$ points over a finite field $\mathbb{F}_q$. When $q \equiv 2 \pmod 3$, then the map $x \mapsto x^3$ is a bijection and thus the curve $E_b : y^2 = x^3 + b$ is supersingular, [BF01]. Define $E(u) := ((u^2 - b)^{1/3}, u)$. Moreover, define $E(0) := 0$ (the neutral element). Clearly, $E(u)$ is always a point of E_b. Finally, E can be implemented in time $O(\log^3 q)$.

Icart [Ica09] proposed the following efficient admissible encoding E. Assume $E : y^2 = x^3 + ax + b$ over the field $\mathbb{F}_q$ where $p > 3$ and $q = p^k \equiv 2 \pmod 3$. Define $E : \mathbb{F}_{p^k} \mapsto E$, $E(u) = (x, y)$, where $x = (v^2 - b - u^6/27)^{1/3} + u^2/3$ and $y = ux + v$ for $v = (3a - u^4)/(6u)$. Moreover, $E(0) = 0$. As proven in [Ica09], $E(u)$ is always a point of E and E can be implemented in time $O(\log^3 q)$. Moreover, for any point P, the solutions u of $E(u) = P$ are roots of a fixed quadratic equation. Thus, $E^{-1}_{a,b}(P)$ is computable in PPT and $|E^{-1}_{a,b}(P)| \leq 4$ for all $P \in E$. Hence, $q/4 \leq |\mathrm{Im}(E)| \leq q$. Icart [Ica09] conjectured and [FT10] proved that $\big||\mathrm{Im}(E)| - \frac{5}{8}|E(\mathbb{F}_q)|\big| \leq \lambda\sqrt{q}$ for some constant λ.

Brier et al. [BCI$^+$10] showed that given an admissible encoding $E : \mathbb{F} \to \mathbb{G}$ and a usual hash function $h : \{0,1\}^* \to \mathbb{F}$, one can efficiently construct elliptic-curve hashings $\mathsf{H} : \{0,1\}^* \to \mathrm{Im}(E)$ and $\mathsf{H} : \{0,1\}^* \to \mathbb{G}$. In particular, if h is a random oracle, then these constructions are indifferentiable from a random oracle in the sense of [MRH04].

In addition, Shallue and Van De Woestijne [SW06] proposed an admissible encoding that works with all elliptic curves $E_{a,b} : y^2 = x^3 + ax + b$, $a, b \neq 0$. Wahby and Boneh [WB19] proposed an efficient admissible encoding in the case of the standard curve BLS12-381. See [BCI$^+$10] for more discussions.

B Supplementary Materials to Section 4 (New Assumptions)

B.1 FPR* Reduction

We define a variation of FPR assumption that we call FPR*. $(\boldsymbol{d}, m)$-FPR* is just as $(\boldsymbol{d}, m)$-FPR, except that it has a new oracle $\mathcal{O}^{\mathsf{fpr}^*}(\boldsymbol{x}, \cdot)$. (We again omit

```
$\mathcal{B}(\mathsf{p}, [(\sigma^i)_{i=0}^{d_1}]_1, [(\sigma^i)_{i=0}^{d_2}]_2)$
$\boldsymbol{z}, \boldsymbol{y} \leftarrow\!\!\$\ \mathbb{F}^m$; Implicitly define $x_j := z_j + y_j\sigma$ for $j \in [1, m]$;
$g(\boldsymbol{X}) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{fpr}^*}(\boldsymbol{x},\cdot)}(\mathsf{p})$;
$P(X) := g(z_1 + y_1X, \ldots, z_m + y_mX)$;
**if** $P(X) = 0$ **then return** $\perp$; **fi**
Find the set of roots $S$ of $P(X)$;
**return** $s \in S$ such that either $s \cdot [1]_1 = [\sigma]_1$ or $s \cdot [1]_2 = [\sigma]_2$;
```

Fig. 7. The FPR* reduction $\mathcal{B}$ to PDL assumption.

the subscript.) On input (κ, f), $\mathcal{O}^{\mathsf{fpr}^*}(\boldsymbol{x}, \cdot)$ returns $[f(\boldsymbol{x})]_\kappa$ if $\kappa \in \{1, 2, T\}$ and $\deg(f) \leq d_\kappa$. That is, the oracle checks the total degree of f, not the maximum individual degree of the variables.

We recall the following result before proving the reduction to PDL.

Proposition 4 ([BFL20]). *Let $g(\boldsymbol{X})$ be an m-variate non-zero polynomial in $\mathbb{F}[\boldsymbol{X}]$ of a total degree d. Define $S(X) := g(Z_1+Y_1 \cdot X, \ldots, Z_m+Y_m \cdot X) \in R[X]$, where $R := \mathbb{F}[\boldsymbol{Z}, \boldsymbol{Y}]$. Then the highest X-degree term in $S(X)$ has a coefficient $S_{max}(\boldsymbol{Y}) \in \mathbb{F}[\boldsymbol{Y}]$ of degree d.*

Theorem 7. *Let $m, d_1, d_2, d_g \geq 0$ and $\boldsymbol{d} = (d_1, d_2, d_1 + d_2, d_g)$. If the (d_1, d_2)-*PDL *assumption holds, then the $(\boldsymbol{d}, m)$-*FPR* *assumption holds. More precisely, for any non-uniform PPT adversary $\mathcal{A}$, there exists a non-uniform PPT adversary $\mathcal{B}$, such that*

$$\mathsf{Adv}^{\mathrm{fpr}*}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{A}}(\lambda) \leq d_g/|\mathbb{F}|^m + \mathsf{Adv}^{\mathrm{pdl}}_{d_1,d_2,\mathsf{Pgen},\mathcal{B}}(\lambda) \ .$$

Proof. Let $\mathcal{A}$ be a non-uniform PPT $(\boldsymbol{d}, m)$-FPR* adversary. In Fig. 7, we depict a non-uniform PPT (d_1, d_2)-PDL adversary $\mathcal{B}$. $\mathcal{B}$ gets as an input $(\mathsf{p}, [(\sigma^i)_{i=0}^{d_1}]_1, [(\sigma^i)_{i=0}^{d_2}]_2)$. $\mathcal{B}$ defines implicitly $\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{y}\sigma$ for $\boldsymbol{z}, \boldsymbol{y} \leftarrow\!\!\$\ \mathbb{F}^m$. While $\mathcal{B}$ does not know σ, $\mathcal{B}$ can homomorphically compute $[x_1^{k_1} \cdot \cdots \cdot x_m^{k_m}]_\kappa$ for $\kappa \in \{1, 2\}$, as long as $\sum_{i=1}^m k_i \leq d_\kappa$. In the case of $\mathbb{G}_T$ elements, $\mathcal{B}$ first computes $[(\sigma^i)_{i=0}^{d_1+d_2}]_T$; thus, monomials $[x_1^{k_1} \cdot \cdots \cdot x_m^{k_m}]_T$ can have the total degree $d_1 + d_2$. $\mathcal{B}$ can simulate any $\mathcal{O}^{\mathsf{fpr}^*}$ query (κ, f) from $\mathcal{A}$ as long as $\kappa \in \{1, 2, T\}$ and $\deg f \leq d_\kappa$, where $d_T := d_1 + d_2$. Note that $x_1, \ldots, x_m$ are distributed just as in the FPR* assumption since z_j is uniformly random.

Next, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{p})$ and simulates responses of $\mathcal{O}^{\mathsf{fpr}^*}(\boldsymbol{x}, \cdot)$ as just described. Eventually $\mathcal{A}$ returns $g(\boldsymbol{X})$. If $\mathcal{A}$ is successful, then g is a non-zero m-variate polynomial of degree $\leq d_g$ such that $V_h(\boldsymbol{x}) = 0$.

We define a new polynomial $S(\boldsymbol{Y}, \boldsymbol{Z}) := g(Z_1 + Y_1 \cdot X, \ldots, Z_m + Y_m \cdot X) \in R[X]$, where $R = \mathbb{F}[\boldsymbol{Y}, \boldsymbol{Z}]$. According to the Proposition 4, the coefficient of the highest degree term (in X) is a polynomial $S_{max}(\boldsymbol{Y})$ in $\mathbb{F}[\boldsymbol{Y}]$, and it has the same total degree as $g(\boldsymbol{X})$. We define $P(X) := g(z_1 + y_1 \cdot X, \ldots, z_m + y_m \cdot X)$. Note that $\boldsymbol{y}$ is information-theoretically hidden from $\mathcal{A}$ since it is blinded by $\boldsymbol{z}$.

Hence, $S_{\max}$ is independent of $\boldsymbol{y}$. Thus, $P(X) = 0$ only if $S_{\max}(\boldsymbol{y}) = 0$, which according to Schwartz-Zippel lemma happens with probability $\leq d_g/|\mathbb{F}|^m$.

If $P(X) \neq 0$ (which happens with an overwhelming probability when $d_g \ll p$), $P(\sigma) = g(\boldsymbol{z} + \boldsymbol{y}\sigma) = 0$ and $\mathcal{B}$ can find roots of P to recover σ. Thus, $\mathsf{Adv}^{\mathrm{fpr}*}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{A}}(\lambda) \leq \frac{d_g}{|\mathbb{F}|^m} + \mathsf{Adv}^{\mathrm{pdl}}_{d_1,d_2,\mathsf{Pgen},\mathcal{B}}(\lambda)$. □

B.2 Comparison of FPR and FPR*.

Let us understand how the two assumptions and corresponding reductions differ, especially when using them in Case X of AGMOS proofs.

It is always possible to use an assumption from one of the two families for the Case X reduction since FPR and FPR* only differ in how the polynomial query oracle checks the degrees. In particular, on an input (κ, f), the oracle $\mathcal{O}^{\mathsf{fpr}}$ checks that the individual degree of each variable in f is at most d_κ and $\mathcal{O}^{\mathsf{fpr}^*}$ checks that the total degree of f is at most d_κ. If Case X reduces to $(\boldsymbol{d}, m)$-FPR, for $\boldsymbol{d} = (d_1, d_2, d_T, d_g)$, then it also reduces to $(\boldsymbol{d}^*, m)$-FPR*, where $\boldsymbol{d}^* = (m \cdot d_1, m \cdot d_2, m \cdot d_T, d_g)$. This is so since if $\deg_{X_i}(f) \leq d_\kappa$ for all $i \in [1, m]$, then $\deg(f) \leq m \cdot d_\kappa$. Vice-versa, if Case X reduces to $(\boldsymbol{d}, m)$-FPR*, for some $\boldsymbol{d}$, then it also reduces to $(\boldsymbol{d}, m)$-FPR. Clearly, if the total degree of $\deg(f) \leq d_\kappa$, then $\deg_{X_i}(f) \leq d_\kappa$ for all $i \in [1, m]$.

However, there can be a significant difference when considering reductions to PDL. Let us look at two examples.

When FPR is better. Consider an assumption where the adversary gets as an input $[(x^i y^j)^d_{i,j=0}]_1$ and has to output $[x^{d+1} y^{d+1}]_1$. When we go through the AGMOS proof, following the blueprint in Section 1.1, we find that for Case X we can construct a reduction $\mathcal{B}_{\mathsf{fpr}}$ to $(\boldsymbol{d}, m)$-FPR assumption, where $\boldsymbol{d} = (d, 0, 0, 2(d+1))$ and $m = 2$. By applying Theorem 2, we get that there exists a non-uniform PPT $\mathcal{B}_{\mathsf{pdl}}$ such that $\mathsf{Adv}^{\mathrm{fpr}}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{B}_{\mathsf{fpr}}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathrm{pdl}}_{d,0,\mathsf{Pgen},\mathcal{B}_{\mathsf{pdl}}}(\lambda)$.

However, for FPR* the natural reduction $\mathcal{B}^*_{\mathsf{fpr}}$ is to the $(\boldsymbol{d}^*, m)$-FPR* assumption, where $\boldsymbol{d}^* = (2d, 0, 0, 2(d+1))$ and $m = 2$. Then, by applying Theorem 7, we get that there exists $\mathcal{B}^*_{\mathsf{pdl}}$ such that $\mathsf{Adv}^{\mathrm{fpr}*}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{B}^*_{\mathsf{fpr}}}(\lambda) \leq (2(d+1))/|\mathbb{F}|^2 + \mathsf{Adv}^{\mathrm{pdl}}_{2d,0,\mathsf{Pgen},\mathcal{B}^*_{\mathsf{pdl}}}(\lambda)$. In the latter case, we obtain a reduction to a provably stronger [BFL20] PDL, and thus in this case, we recommend using the former reduction.

When FPR* is better. Let us consider a simple assumption where the adversary gets as an input $[x_1, \dots, x_m]_1$ for $x_1, \dots, x_m \leftarrow\$ \mathbb{F}^m$ and has to output $[\prod_{i=1}^m x_i]_1$. In this case, the total and individual degrees are the same. Thus, there are a reductions $\mathcal{B}_{\mathsf{fpr}}$ and $\mathcal{B}^*_{\mathsf{fpr}}$, which respectively reduce the Case X to $(\boldsymbol{d}, m)$-FPR and to $(\boldsymbol{d}, m)$-FPR*, where $\boldsymbol{d} = (1, 0, 0, m)$. Then, by applying Theorem 2 and Theorem 7 respectively, we get that

$$\mathsf{Adv}^{\mathrm{fpr}}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{B}_{\mathsf{fpr}}}(\lambda) \leq m \cdot \mathsf{Adv}^{\mathrm{pdl}}_{1,0,\mathsf{Pgen},\mathcal{B}_{\mathsf{pdl}}}(\lambda) \tag{6}$$

$\mathsf{Pgen}(1^\lambda)$**:** output $\mathsf{p} \leftarrow\!\!\$\ \mathsf{Pgen}$, where p cointains the description of a hash function H;
$\mathsf{KGen}(\mathsf{p})$**:** $\mathsf{sk} = x \leftarrow\!\!\$\ \mathbb{Z}_p$; output $([1]_1, \mathsf{pk} = [x]_1)$;
$\mathsf{Sign}(\mathsf{sk}, m \in \mathbb{F})$**:** $r \leftarrow\!\!\$\ \mathbb{F}$; $\mathsf{c} \leftarrow \mathsf{H}([r]_1, m, [x]_1)$; $\mathsf{z} \leftarrow \mathsf{c}x + r$; return $\sigma = (r, \mathsf{z})$;
$\mathsf{Vf}(\mathsf{pk}, m \in \mathbb{F}, \sigma = ([r]_1, \mathsf{z}))$**:** $\mathsf{c} \leftarrow \mathsf{H}([r]_1, m, [x]_1)$; Check $\mathsf{z}[1]_1 = \mathsf{c}[x]_1 + [r]_1$;

Fig. 8. Schnorr's signature scheme Σ.

and

$$\mathsf{Adv}^{\mathrm{fpr}*}_{\mathsf{Pgen},\boldsymbol{d},m,\mathcal{B}^*_{\mathsf{fpr}}}(\lambda) \leq m/|\mathbb{F}|^m + \mathsf{Adv}^{\mathrm{pdl}}_{1,0,\mathsf{Pgen},\mathcal{B}^*_{\mathsf{pdl}}}(\lambda) \ . \tag{7}$$

Here, FPR* is unquestionably a better choice for large values of m. Even more, as m increases, tightness in Eq. (6) gets linearly worse, whereas in Eq. (7), it improves slightly.

These two examples give general guidelines for choosing between FPR and FPR*. If the queried polynomials contain multivariate monomials, but m is small, then likely FPR is the better choice. However, if the queried polynomials are univariate and have a small degree, but m is large, then FPR* might be the better option. We recommend trying both reductions to see which works best for a concrete AGMOS proof.

C Schnorr's Signature Scheme

In a signature scheme, the signer (who knows the secret signing key x) signs a message m, obtaining a signature σ. The verifier (who only knows the public verification key pk) checks, given m and σ, if σ is a valid signature on m. The signature scheme is EUF-CMA (existential unforgeability under chosen-message attack) secure, if it is infeasible to create an accepting signature without knowing x, even when one is given access to an adaptive signing oracle. For the sake of convenience, we use pairing-based notation, but we emphasize that Schnorr's signature does not use pairings. That is, $\mathbb{G}_1$ can be *any* cyclic group where DL is expected to hold.

Definition 10. *A signature scheme* $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ *is* EUF-CMA secure *for* Pgen*, if for every non-uniform PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathrm{eufcma}}_{\mathsf{Pgen},\Sigma,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l} m^* \notin \mathsf{Q} \wedge \\ \mathsf{Vf}(\mathsf{pk}, m^*, \sigma^*) = 1 \end{array}\middle| \begin{array}{l} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\mathsf{p}); \\ \mathsf{Q} \leftarrow \emptyset; (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}^*}(\mathsf{pk}) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

Here, the oracle $\mathsf{Sign}^*(m)$ *sets* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$*, appends* m *to* Q*, and returns* σ*.*

Schnorr [Sch91] proposed an efficient signature scheme that is EUF-CMA secure the FPRO model and tightly EUF-CMA secure when one relies both on the AGM and the FPRO model. See Fig. 8.

An AGM security proof of Schnorr needs to emulate signing queries and hash queries. In AGMOS, we need to additionally deal with (non-emulated) sampling queries. As in [FPS20], given an input $([r]_1, m, \mathsf{pk} = [x]_1)$, the signing oracle Sign^* does the following: if the same input was received before, output the same c as before. Otherwise, sample random $\mathsf{c} \leftarrow\$ \mathbb{F}$, $\mathsf{z} \leftarrow\$ \mathbb{F}$, and choose $[r]_1$ so that the verifier accepts, i.e., $[r]_1 \leftarrow [\mathsf{z}]_1 - \mathsf{c}[x]_1$. Thus, each emulation succeeds perfectly. Similarly, the hash oracle H stores all input/output paairs but otherwise acts honestly by outputting a random field element if it was not queried before in the same inputs. (See Fig. 9.)

We will next prove that it is secure in the AGMOS + FPRO model, where one only models the hash oracle H as an FPRO. Cases A and $\mathsf{X}.1$ in our proof correspond to the AGM proof of [FPS20], while Q is new. We emphasize that more than 90% of the following proof follows [FPS20] very closely, demonstrating that in the case of more complicated AGM proofs, adding the AGMOS part to it is "relatively" easy compared to the AGM proof itself.

Theorem 8 (EUF-CMA of Schnorr). *If the* $(\boldsymbol{d}, m)$-FPR *and* $(\mathcal{EF}, \mathcal{DF})$-TOFR *assumptions hold, then Schnorr is EUF-CMA secure in the AGMOS.*

We repeat the proof intuition from [FPS20]. In the random oracle model, Schnorr signatures can be simulated without knowledge of the secret key by choosing random c and z, setting $[r]_1 := \mathsf{z}[1]_1 - \mathsf{c}[x]_1$ and then programming the random oracle so that $\mathsf{H}([r]_1, m) = \mathsf{c}$. On the other hand, an adversary that returns a signature forgery $(m^*, ([r^*]_1, \mathsf{z}^*))$ can be used to compute the discrete logarithm of the public key $\mathsf{pk} = [x]_1$. In the ROM (without AGM), extraction entails a security loss. In the AGM+ROM, extraction is straight-line and the security proof thus tight. After querying the signing oracle on messages $m_1, \dots, m_{q_s}$, the adversary obtains $([r_i]_1, \mathsf{z}_i)_{1 \le i \le q_s}$ that verify $[r_i]_1 = \mathsf{z}_i[1]_1 - \mathsf{c}_i[x]_1$ with $\mathsf{c}^* := \mathsf{H}(R^*, m^*)$. A valid forgery satisfies

$$[r^*]_1 = \mathsf{z}^*[1]_1 - \mathsf{c}^*[x]_1 \ ,$$

with $\mathsf{z}^* := \mathsf{H}([r^*]_1, m^*)$.

On the other hand, since the adversary is algebraic, when it made its first query $\mathsf{H}([r^*]_1, m^*)$, it provided a representation of $[r^*]_1$ in basis $[1, x, r_1, \dots, r_{q_s}]_1$, that is, $(\boldsymbol{\gamma}, \boldsymbol{\delta})$ with

$$[r^*]_1 = \gamma_1[1]_1 + \gamma_2[x]_1 + \textstyle\sum_{i=1}^{q_s} \gamma_{2+i}[r_i]_1 = \gamma_1[1]_1 + \gamma_2[x]_1 + \sum_{i=1}^{q_s} \gamma_{2+i}(\mathsf{z}_i[1]_1 - \mathsf{c}_i[x]_1) \ .$$

Together with the accepting verification $[r^*]_1 = \mathsf{z}^*[1]_1 - \mathsf{c}^*[x]_1$, this yields

$$\left(\mathsf{c}^* + \gamma_2 - \textstyle\sum_{i=1}^{q_s} \gamma_{2+i}\mathsf{c}_i\right)[x]_1 = \left(\mathsf{z}^* - \gamma_1 - \sum_{i=1}^{q_s} \gamma_{2+i}\mathsf{z}_i\right)[1]_1 \ . \tag{8}$$

Since c^* was chosen at random after the adversary chose $\boldsymbol{\gamma}, \boldsymbol{\delta}$, the probability that $\mathsf{c}^* + \gamma_2 - \sum_{i=1}^{q_s} \gamma_{2+i}\mathsf{c}_i \neq 0$ is overwhelming, in which case we can compute the discrete logarithm of x from the above equation.

An AGM proof also has a PDL reduction that we replace with a FPR reduction. On top of that, in the AGMOS proof, we need to consider Case Q, which in this case is relatively easy.

Proof. Let H be a FPRO. Let $\mathcal{A}$ be a EUF-CMA AGMOS adversary that makes at most q_s signature queries and q_h RO queries. We proceed by a sequence of three games.

Game_1. This is the EUF-CMA game for the Schnorr signature scheme with a random oracle H. It maintains a list Q of queried messages and T of values sampled for H. When it returns a forgery $(m^*, ([r^*]_1, \mathsf{z}^*))$, we can run the AGMOS extractor $\overline{\mathsf{Ext}}_{\mathcal{A}}$ to obtain $(\boldsymbol{\gamma}, \boldsymbol{\delta})$, such that $[r^*]_1 = \gamma_1[1]_1 + \gamma_2[x]_1 + \sum_{i=1}^{q_s} \gamma_{2+i}[r_i]_1 + \sum \delta_i[\mathbb{q}_i]_1$.

Assume that, with non-negligible probability ε_1, $\mathcal{A}$ outputs $(m^*, \sigma^* = ([r^*]_1, \mathsf{z}^*))$, such that $\mathsf{z}^*[1]_1 = \mathsf{c}^*[x]_1 + [r^*]_1$, where $\mathsf{c}^* = \mathsf{H}([r^*]_1, m, [x]_1)$. Moreover, $m^* \notin \mathsf{Q}$. By definition, $\varepsilon_1 = \mathsf{Adv}^{\mathrm{eufcma}}_{\mathsf{Pgen},\varSigma,\mathcal{A}}(\lambda)$.

Game_2. In Game_1, when the adversary calls $\mathsf{H}([r]_1, m, [x]_1)$ (say, as the jth query), we do the following change. The game uses the AGMOS extractor to obtain an explanation $(\boldsymbol{\gamma}, \boldsymbol{\delta})$ of $[r_j]_1$ in terms of previous inputs, $[r_j]_1 = \gamma_1[1]_1 + \gamma_2[x]_1 + \sum_{i=1}^{j-1} \gamma_{2+i}[r_i]_1$. It returns $\perp$ if $\mathsf{z}_i + \gamma_2 - \sum_{i=1}^{j-1} \gamma_{2+i}\mathsf{c}_i = 0$. Clearly, for each call of H, this happens with probability $1/|\mathbb{F}|$.

Since H is called at most q_h times by the adversary and once by the game when checking the signature, we get that the adversary succeeds in Game_1 with probability $\varepsilon_1 \geq \varepsilon_0 - (q_h + 1)/p$.

Game_3. In Game_3, we use the standard method for simulating the Sign oracle without the secret key by programming the random oracle. Game_3 behaves the same as Game_2, except when Sign aborts on line (*). For each signature query, $[r_j]_1$ is uniformly random, and the size of T is at most $q_s + q_h$. Thus, Game_3 aborts in line (*) wth probability $\leq (q_s + q_h)/p$. By summing over at most q_s signature queries, the difference of $\mathcal{A}$ being in successful in Game_3 satisfies $\varepsilon_2 \geq \varepsilon_1 - q_s(q_s + q_h)/p$.

Let us now consider the adversary's success in Game_3. Since $\mathcal{A}$ is an AGMOS adversary, there exists an extractor $\overline{\mathsf{Ext}}_{\mathcal{A}}$ that extracts $\boldsymbol{\gamma}, \boldsymbol{\delta}$, such that $[r^*]_1 = \boldsymbol{\gamma}^\intercal \left[\begin{smallmatrix} 1 \\ x \\ \boldsymbol{r} \end{smallmatrix}\right]_1 + \boldsymbol{\delta}^\intercal[\mathbb{q}]_1$, where $[\boldsymbol{r}]_1 = \mathbf{z} - \mathbf{c}[x]_1$ (resp.,$[\mathbb{q}]_1$) is the tuple of signing (resp., sampling) oracle answers in $\mathbb{G}_1$. Let ql_1 be the number of sampling oracle queries in $\mathbb{G}_1$. Define

$$R(X, \mathbb{Q}) = \gamma_1 + \gamma_2 X + \textstyle\sum_{i \geq 1} \gamma_{2+i}(\mathsf{z}_i - \mathsf{c}_i X) + \sum \delta_i \mathbb{Q}_i \enspace .$$

Thus, $[r]_1 = [R(x, \mathbb{q})]_1$. Next, assume that both $\mathcal{A}$ and $\overline{\mathsf{Ext}}_{\mathcal{A}}$ succeeded. The verifier checks that $[V(\boldsymbol{x}, \mathbb{q})]_T = [0]_T$, where

$$\begin{aligned} V(X, \mathbb{Q}) :=& \mathsf{z} - \mathsf{c}X - R(X, \mathbb{Q}) \\ =& \mathsf{z} - \mathsf{c}X - \gamma_1 - \gamma_2 X - \textstyle\sum_{i \geq 1} \gamma_{2+i}(\mathsf{z}_i - \mathsf{c}_i X) - \sum \delta_i \mathbb{Q}_i \\ =& V^h(\boldsymbol{X}) + V^t(\mathbb{Q}) \enspace , \end{aligned} \tag{9}$$

where $V^h(X) = \mathsf{z} - \gamma_1 - \sum_{i \geq 1} \gamma_{2+i}\mathsf{z}_i + (-\mathsf{c} - \gamma_2 + \sum_{i \geq 1} \gamma_{2+i}\mathsf{c}_i)X$ and $V^t(\mathbb{Q}) = -\sum \delta_i \mathbb{Q}_i$. Note that $V^t = 0$ in an AGM proof. Let us now consider the three AGMOS cases.

```
$\boxed{\mathcal{B}_{\mathsf{pdl}}^{\mathcal{A}^{\mathcal{O}}}(\mathsf{p}; \mathbb{x})}$ $\boxed{\mathcal{B}_{\mathsf{tofr}}^{\mathcal{A}^{\mathcal{O}}}(\mathsf{p}; [1]_1)}$    // $\mathbb{x} = ([1, x]_1$
$j \leftarrow 0; \mathsf{Q} \leftarrow \emptyset; \mathsf{T} \leftarrow \emptyset;$ $\boxed{\mathsf{U} \leftarrow \emptyset;}$
$r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A});$ $\boxed{x \leftarrow_{\$} \mathbb{F}}$; $(m^*, \sigma^* = ([r^*]_1, \mathsf{z}^*)) \leftarrow \mathcal{A}^{\mathsf{Sign}^*, \mathsf{H}, \mathcal{O}}(\mathsf{p}, \mathbb{x}; r);$    // $\mathbb{y} = ( \, r^* \, )$
$(\boldsymbol{\gamma}, \boldsymbol{\delta}, [\mathbb{q}]_1) \leftarrow \overline{\mathsf{Ext}}_{\mathcal{A}}^{\mathcal{O}}(\mathsf{p}, \mathbb{x}; r);$ if $\overline{\mathsf{Ext}}_{\mathcal{A}}$ failed then return $\perp$; fi
Define $V(X, \mathbb{Q})$, $V^h(X)$, $V^t(\mathbb{Q})$ as in Eq. (9);
if $m^* \in \mathsf{Q}$ then return $\perp$; fi
$\mathsf{c}^* \leftarrow \mathsf{H}([r^*]_1, m^*, [x]_1);$
$(\boldsymbol{\gamma}, \boldsymbol{\delta}) \leftarrow \mathsf{U}([r^*]_1, m^*);$
if $V(X, \mathbb{Q}) \neq 0 \wedge V^t(\mathbb{Q}) = 0$ then return $x \leftarrow \frac{(\mathsf{z}^* - \gamma_1 - \sum_{i=1}^{q_s} \gamma_{2+i} \mathsf{z}_i)}{(\mathsf{c}^* + \gamma_2 - \sum_{i=1}^{q_s} \gamma_{2+i} \mathsf{c}_i)}$; fi
$\boxed{\textbf{if } V^t(\mathbb{Q}) \neq 0 \textbf{ then return } \boldsymbol{v} \leftarrow \begin{pmatrix} V^h(x) \\ -\boldsymbol{\delta} \end{pmatrix} .; \textbf{fi}}$
return $\perp$;
```

```
Oracle $\tilde{\mathsf{H}}([r]_1, m)$
if return $\mathsf{T}([r]_1, m) = \perp$ then $\mathsf{T}([r]_1, m) \leftarrow_{\$} \mathbb{F}$; fi
return $\mathsf{T}([r]_1, m)$;
```

```
Oracle $\mathsf{H}([r]_1, m)$
if $\mathsf{T}([r]_1, m) = \perp$ then
   $\mathsf{T}([r]_1, m) \leftarrow_{\$} \mathbb{F}$;
Game$_{2,3}$ {
   $(\boldsymbol{\gamma}, \boldsymbol{\delta}) \leftarrow \overline{\mathsf{Ext}}_{\mathcal{A}}(\mathsf{p}, (\mathbb{x}, [r_1, \ldots, r_{|\mathsf{Q}|}]_1); r);$    // $[r]_1 = \gamma_1[1]_1 + \gamma_2[x]_1 + \sum_{i=1}^{|\mathsf{Q}|} \gamma_{2+i}[r_i]_1 + \boldsymbol{\delta}^\top [\mathbb{q}]_1$
   $\boxed{\mathsf{U}([r]_1, m) \leftarrow (\boldsymbol{\gamma}, \boldsymbol{\delta});}$
   if $\gamma_2 - \sum_{i=1}^{|\mathsf{Q}|} \gamma_{2+i} \mathsf{c}_i = -\mathsf{T}([r]_1, m)$ then return $\perp$; fi
fi
return $\mathsf{T}([r]_1, m)$;
```

```
Oracle $\mathsf{Sign}(m)$
Game$_{1,2}$ {
   $j \leftarrow j + 1$; $r_j \leftarrow_{\$} \mathbb{Z}_p$; $\mathsf{z}_j \leftarrow \tilde{\mathsf{H}}([r_j]_1, m, [x]_1)$; $\mathsf{z}_j \leftarrow r_j + \mathsf{z}_j x$;
   $\mathsf{Q} \leftarrow \mathsf{Q} \| m$; return $([r_j]_1, \mathsf{z}_j)$;
Game$_3$ {
   $j \leftarrow j + 1$; $\mathsf{c}_j, \mathsf{z}_j \leftarrow_{\$} \mathbb{Z}_p$; $[r_j]_1 \leftarrow \mathsf{z}_j[1]_1 - \mathsf{c}_j[x]_1$;
   if $\mathsf{T}([r_j]_1, m) = \perp$ then append $\mathsf{T}([r_j]_1, m) \leftarrow \mathsf{c}_j$; else return $0$; fi ;    (*)
   $\mathsf{Q} \leftarrow \mathsf{Q} \| m$; return $([r_j]_1, \mathsf{z}_j)$;
```

Fig. 9. Schnorr EUF-CMA security: the extractor $\mathsf{Ext}_{\mathcal{A}}$, the FPR adversary $\mathcal{B}_{\mathsf{fpr}}$, and the TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$ in the proof of Theorem 8, where the differences are either $\boxed{\textit{boxed}}$ ($\mathsf{Ext}_{\mathcal{A}}$), $\boxed{\textit{dashedboxed}}$ ($\mathcal{B}_{\mathsf{fpr}}$) or $\boxed{\textit{dottedboxed}}$ ($\mathcal{B}_{\mathsf{tofr}}$). We also marked the lines that are only used in some of the games.

Case A: $V(X, \mathbb{Q}) = 0$. Then also $V^h(\boldsymbol{X}) = 0$. Looking at the coefficient of X in V^h, it means $\mathsf{c} = \gamma_2 - \sum_{i \geq 1} \gamma_{2+i} \mathsf{c}_i$. However, in this case we already aborted in Game_2. Thus, this case does not materialize in Game_3.

Case X: $V(X, \mathbb{Q}) \neq 0$ and $V^t(x, \mathbb{Q}) = 0$: First, note that since V^t does not depend on x, Case $\mathsf{X}.2$ ($V^t(X, \mathbb{Q}) \neq 0$ and $V^t(x, \mathbb{Q}) = 0$) is impossible. Hence, here we only need to concentrate on Case $\mathsf{X}.1$ ($V(X, \mathbb{Q}) \neq 0$ and $V^t(\mathbb{Q}) = 0$).

We show that this case can happen only with negligible probability through a reduction to DL. In Fig. 6, we depict the DL adversary $\mathcal{B}_{\mathsf{pdl}}$ that works in this case. $\mathcal{B}_{\mathsf{pdl}}$ obtains $\mathbb{x} = [1, x]_1$. It then calls the Game_3 adversary $\mathcal{A}$ on input $\mathbb{x}$. (Note that in Game_3, $\mathcal{B}_{\mathsf{pdl}}$ does not need to know x.) It uses an auxiliary table U, storing there for every query $\mathsf{H}([r]_1, m)$ the linear representation (w.r.t. $\mathbb{x}$ and earlier values $[r_i]_1$) of $[r]_1$. Assume that $\mathcal{A}$ wins Game_3 by returning $(m^*, ([r*]_1, \mathsf{z}^*))$ and let $\mathsf{c}^* = \mathsf{T}([r^*]_1, m^*)$ (which is necessarily defined after the call $\mathsf{c}^* = \mathsf{H}([r*]_1, m^*, [x]_1)$).

Let us argue that $(\boldsymbol{\gamma}, \boldsymbol{\delta}) = \mathsf{U}([r^*]_1, m^*) \neq \perp$ and $\gamma_2 - \sum_{i=1}^{|\mathsf{Q}|} \gamma_{2+i} \mathsf{c}_i \neq -\mathsf{c}^*$. First, $m^* \notin \mathsf{Q}$, as otherwise the game would have already aborted. Hence, $\mathsf{T}([r^*]_1, m^*)$ can only have been defined either (1) during a call to H by $\mathcal{A}$, or (2) undefined when $\mathcal{A}$ stops, but defined by the game when calling $\mathsf{c}^* = \mathsf{H}([r^*]_1, m^*, [x]_1)$. In both cases, the call sets both $\mathsf{T}([r^*]_1, m^*)$ and $\mathsf{U}([r^*]_1, m^*)$. Moreover, $\gamma_2 - \sum_{i=1}^{|\mathsf{Q}|} \gamma_{2+i} \mathsf{c}_i \neq -\mathsf{c}^*$ is satisfied due to the definition of Game_2. From the argument just before the proof (see Eq. (8)), we get that $\mathcal{B}_{\mathsf{pdl}}$ computes the DL of $[x]_1$ correctly. Note that in Case X, the DL adversary succeeds with the same probability as $\mathcal{A}$ in Game_3.

Case Q: $V^t(x, \mathbb{Q}) \neq 0$. In Fig. 9, we depict a Game_3 TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$. $\mathcal{B}_{\mathsf{tofr}}$ samples x to construct $\mathbb{x}_1 = [1, x]_1$. It then runs $\mathcal{A}$ to obtain $(m^*, \sigma^* = ([r^*]_1, \mathsf{z}^*))$, such that $\mathsf{z}^*[1]_1 = \mathsf{c}^*[x]_1 + [r^*]_1$ for $\mathsf{c}^* = \mathsf{H}([r^*]_1, m^*, [x]_1)$, and uses $\overline{\mathsf{Ext}}_{\mathcal{A}}$ to extract field elements $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ such that $[r^*]_1 = \boldsymbol{\gamma}_1^\intercal \left[\begin{smallmatrix} 1 \\ x \\ \boldsymbol{r} \end{smallmatrix} \right]_1 + \boldsymbol{\delta}_1^\intercal [\mathbb{q}]_1$. If the verifier accepts, $0 = V(x, \mathbb{q}) = V^h(x) - \sum_{i=1}^{\mathsf{ql}_1} \delta_i \mathbb{q}_i$. (see Eq. (9)). Thus, $\mathcal{B}$ outputs

$$\boldsymbol{v} \leftarrow \left(\begin{smallmatrix} V^h(x) \\ -\boldsymbol{\delta} \end{smallmatrix} \right) .$$

(Note that there are $\mathsf{ql}_1 = 0$ oracle queries in $\mathbb{G}_2$.) Since $V^t(\mathbb{Q}) \neq 0$, then $\boldsymbol{v} \neq \mathbf{0}$ and $\mathcal{B}$ breaks the TOFR assumption.

Thus, $\mathsf{Adv}^{\mathrm{eufcma}}_{\mathsf{Pgen}, \Sigma, \mathcal{A}}(\lambda) \leq \frac{q_s(q_s + q_h) + q_h + 1}{|\mathbb{F}|} + \mathsf{Adv}^{\mathrm{fpr}}_{\mathsf{Pgen}, \boldsymbol{d}, m, \mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathrm{tofr}}_{\mathsf{Pgen}, \mathcal{EF}, \mathcal{DF}, \mathcal{B}_{\mathsf{tofr}}}(\lambda)$. This concludes the proof. □

D Supplementary Materials to Section 6

D.1 Proof of Proposition 2

Proof. Let Z be the set of common roots of f_j. If $y \in \mathbb{F}_q$, then $y^{q-1} = 1$ if $y \neq 0$ and $y^{q-1} = 0$ if $y = 0$. Let $\chi := \prod_{j=1}^r (1 - f_j^{q-1}) \in \mathbb{F}[t_1, \ldots, t_n]$. Then, for all $\boldsymbol{x} \in \mathbb{F}_q^{\mathsf{ol}}$, $\chi(\boldsymbol{x}) = 1$ if $x \in Z$ and $\chi(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \notin Z$. Thus, $\sum_{\boldsymbol{x} \in \mathbb{F}_q^{\mathsf{ol}}} \chi(\boldsymbol{x}) = \sharp Z$. Since $\mathbb{F}_q$ has characteristic p, $p \mid \sharp Z$ holds iff $\sum_{\boldsymbol{x} \in \mathbb{F}_q} \chi(\boldsymbol{x}) = 0$. Moreover, $\deg \chi = \sum_{j=1}^r \deg(1 - f_j^{q-1}) = (q-1) \sum_{j=1}^r d_j < (q-1)n$.

Thus, the theorem follows from the next claim: any polynomial $P \in \mathbb{F}_q[t_1, \ldots, t_n]$ of degree less than $(q-1)n$ satisfies $\sum_{\boldsymbol{x} \in \mathbb{F}_q^{\mathsf{ol}}} P(\boldsymbol{x}) = 0$. We are thus only left to prove the claim. For this, observe that $P \in \mathbb{F}_q[t_1, \ldots, t_n] \mapsto \sum_{\boldsymbol{x} \in \mathbb{F}_q^n} P(\boldsymbol{x}) \in \mathbb{F}_q$ is $\mathbb{F}_q$-linear. Thus, it is enough to show the result for a monomial $t_1^{a_1} \cdots t_n^{a_n}$ of degree less than $(q-1)n$. Next, clearly,

$$\textstyle\sum_{\boldsymbol{x} \in \mathbb{F}_q^{\mathsf{ol}}} x_1^{a_1} \cdots x_n^{a_n} = \left(\sum_{x_1 \in \mathbb{F}_q} x_1^{a_1} \right) \cdots \left(\sum_{x_1 \in \mathbb{F}_q} x_n^{a_n} \right) \enspace .$$

If $a_1 + \ldots + a_n = \deg(t_1^{a_1} \cdots t_n^{a_n}) < (q-1)n$, then we must have $a_i < q - 1$ for some i. Thus it suffices to show that if $0 \le a_i \le q - 2$, then $\sum_{x_i \in \mathbb{F}_q} x_i^{a_i} = 0$. If $a_i = 0$, then this sum is q, which is 0 in $\mathbb{F}_q$. Thus, suppose that $1 \le a_i \le q - 2$. The group $\mathbb{F}_q^*$ is cyclic, let ω be a generator. Then

$$\textstyle\sum_{x_i \in \mathbb{F}_q} x_i^{a_i} = \sum_{k=0}^{q-2} (\omega^k)^{a_i} = \frac{(\omega^{a_i})^{q-1} - 1}{\omega^{a_i} - 1} = 0 \enspace .$$

This proves the theorem. □

D.2 Examples

Let us check some concrete examples. Note that those examples do not precisely fall into the previous results. Some of them have $\mathsf{il}_1 > 1$ but this still works since the polynomials in $\mathbb{x}_1(\boldsymbol{X})$ are linearly independent. Moreover, KE is a privately verifiable assumption.

SpurKE assumption Consider a variant of TotalKE, where $\mathsf{ol}_1 = 1$, $\mathsf{ol}_2 = 0$ (the adversary outputs $[\mathbb{y}]_1$), and the verifier performs no check. Thus

$$\boldsymbol{N} = \boldsymbol{P}_1^\top \boldsymbol{M} \boldsymbol{P}_2 = \left(\begin{smallmatrix} 1 & \boldsymbol{0}^\top \\ \gamma & \boldsymbol{\Delta}^\top \end{smallmatrix} \right)^\top \cdot \begin{array}{cc} & 1 \\ 1 & 0 \\ \hline \mathbb{y} & 0 \end{array} \cdot \left(\begin{smallmatrix} 1 \end{smallmatrix} \right) = \left(\begin{smallmatrix} 0 \\ \boldsymbol{0} \end{smallmatrix} \right) \enspace .$$

Thus, from the last row of $\boldsymbol{N} = 0$, it does not follow that $\boldsymbol{\Delta} = \boldsymbol{0}$, and (if the DL assumption holds) this assumption is not secure in the AGMOS.

KE assumption: verifier checks $\mathbb{y}_2 - X\mathbb{y}_1 = 0$. Thus

$$\boldsymbol{N} = \boldsymbol{P}_1^\top \boldsymbol{M} \boldsymbol{P}_2 = \begin{pmatrix} 1 & 0 & \boldsymbol{0}^\top \\ 0 & 1 & \boldsymbol{0}^\top \\ \hline \gamma_{11} & \gamma_{12} & \boldsymbol{\Delta}_1^\top \\ \gamma_{21} & \gamma_{22} & \boldsymbol{\Delta}_2^\top \end{pmatrix}^\top \cdot \begin{array}{ccc} & 1 & X \\ 1 & 0 & 0 \\ X & 0 & 0 \\ \hline \mathbb{y}_1 & 0 & -1 \\ \mathbb{y}_2 & 1 & 0 \end{array} \cdot \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix} \right) = \begin{pmatrix} \gamma_{21} & -\gamma_{11} \\ \gamma_{22} & -\gamma_{12} \\ \boldsymbol{\Delta}_2 & -\boldsymbol{\Delta}_1 \end{pmatrix} \enspace ,$$

$$\boldsymbol{f}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \boldsymbol{\Delta}_2 & -\boldsymbol{\Delta}_1 \end{pmatrix} \enspace .$$

Thus, from the last two rows of $\boldsymbol{N}$ being 0 it follows that $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2 = \boldsymbol{0}$ and hence, if the DL assumption holds, KE *may* be secure in the AGMOS. (As we will see later, we need to deal with the Case Q for precise implication.) Note that here, there are no $\mathbb{G}_2$ queries, so $\boldsymbol{f}$ is somewhat simpler.

SpurQKE assumption: Consider a variant of TotalKE where $\mathsf{ol}_1 = 2$, $\mathsf{ol}_2 = 1$. Denoting the outputs in $\mathbb{G}_1$ by $\mathbb{y}_1, \mathbb{y}_3$ and the output in $\mathbb{G}_2$ by $\mathbb{y}_2$, the verifier checks $\mathbb{y}_3 - \mathbb{y}_1\mathbb{y}_2 = 0$. Thus,

$$\boldsymbol{N} = \left(\begin{array}{c|c} 1 & \boldsymbol{0}^\top \\ \hline \gamma_{11} & \boldsymbol{\Delta}_{11}^\top \\ \gamma_{12} & \boldsymbol{\Delta}_{12}^\top \end{array} \right) \begin{array}{c|cc} & 1 & \mathbb{y}_2 \\ 1 & 0 & 0 \\ \hline \mathbb{y}_1 & 0 & -1 \\ \mathbb{y}_3 & 1 & 0 \end{array} \left(\begin{array}{c|c} 1 & \boldsymbol{0}^\top \\ \hline \gamma_2 & \boldsymbol{\Delta}_2^\top \end{array} \right) = \left(\begin{array}{c|c} \gamma_{12} - \gamma_{11}\gamma_2 & -\gamma_{11}\boldsymbol{\Delta}_2^\top \\ \hline \boldsymbol{\Delta}_{12} - \gamma_2\boldsymbol{\Delta}_{11} & -\boldsymbol{\Delta}_{11}^\top\boldsymbol{\Delta}_2 \end{array} \right) \enspace .$$

Here, $f(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = 0$ implies that $\boldsymbol{\Delta}_2$ is zero, but it does not imply that $\boldsymbol{\Delta}_1$ (and thus $\boldsymbol{\Delta}$) is zero.

PKE assumption: verifier checks $\mathbb{y}_2 - \mathbb{y}_1 = 0$. Thus

$$\boldsymbol{N} = \boldsymbol{P}_1^\intercal \boldsymbol{M} \boldsymbol{P}_2 = \begin{pmatrix} \boldsymbol{I}_{d+1} & \mathbf{0}_{d+1}^\intercal \\ \boldsymbol{\gamma}_1 & \boldsymbol{\Delta}_1^\intercal \end{pmatrix}^\intercal \cdot \left(\begin{array}{c|cccc|c} & 1 & X & \dots & X^d & \mathbb{y}_2 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ X & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ X^d & 0 & 0 & 0 & 0 & 0 \\ \hline \mathbb{y}_1 & -1 & 0 & 0 & 0 & 0 \end{array}\right) \cdot \begin{pmatrix} \boldsymbol{I}_{d+1} & \mathbf{0}_{d+1}^\intercal \\ \boldsymbol{\gamma}_2 & \boldsymbol{\Delta}_2^\intercal \end{pmatrix}$$

$$= \left(\begin{array}{cccc|c} \gamma_{21} - \gamma_{11} & \gamma_{22} & \dots & \gamma_{2,d+1} & \boldsymbol{\Delta}_2^\intercal \\ -\gamma_{12} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -\gamma_{1d+1} & 0 & \dots & 0 & 0 \\ \hline -\boldsymbol{\Delta}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \end{array}\right) .$$

Thus, from $f(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = 0$ it follows that $\boldsymbol{\Delta} = \mathbf{0}$. Hence, if DL holds, PKE *may* be secure in the AGMOS. (As we will see later, we need to deal with the Case Q for precise implication.)

E Oblivious Sampling Attacks For KZG

KZG polynomial commitment scheme has a natural homomorphic property, $\mathsf{com}(f(X)) + \mathsf{com}(g(X)) = \mathsf{com}(f(X) + g(X))$, which often gets used in practice. We look at one example, optimizing quadratic tests, where the extractability of the commitment may not work out. While this is just one concrete example, we picked it since (a small variant of) t is used [GWC19,CHM+20,CFF+21,LSZ22].

We do not claim an attack on such zk-SNARKs. We will leave it as a future work to establish which of them are secure (and one just has to rewrite the security proofs) in the AGMOS and which need some — hopefully minor — modifications.

Example: Quadratic Test. Most of the known pairing-based Zk-SNARK verifiers use polynomial equality tests of the form $f_1(X) \cdot f_2(X) = f_3(X)$. In PCS-based zk-SNARKs, the naive way to test it is as follows. The prover sends commitments $[f_1(x)]_1, [f_2(x)]_1, [f_3(x)]_1$ and the verifier responds with a random challenge α. Then, the prover opens $v_i = f_i(\alpha)$ for $i = 1, 2, 3$ and checks that $v_1 \cdot v_2 - v_3 = 0$. Since the equality is satisfied on a random point α, with an overwhelming probability, it also holds for the polynomial equality.

The following is a common optimization of this test, first introduced in [CHM+20] and subsequently used in almost all pairing-based zk-SNARKs. First, the prover opens v_1 just as before. For the actual test, the prover and verifier homomorphically compute an intermediate commitment $[h(x)]_1 \leftarrow v_1[f_2(x)]_1 - [f_3(x)]_1$ and then open this polynomial to 0 at point α. Instead of sending v_1, v_2, and v_3 and three opening proofs, the prover now only sends v_1 and two opening proofs. (The opening proofs are usually additionally batched.)

This approach, however, has a problem with extractability. Suppose $f_1(X) = c$ is a constant polynomial. The adversary can obliviously sample $[f_2]_1$ and then set $[f_3]_1 \leftarrow c[f_2]_1$. The adversary opens $f_1(X)$ honestly to the value $v_1 = f_1(\alpha) = c$. In AGMOS, one can then extract $f_1(X)$. However, for $h(X)$, the adversary can compute an opening proof $[h(x)/(x-\alpha)]_1 = [(cf_2 - cf_2)/(x-\alpha)]_1 = [0]_1$. As in the previous example, in AGMOS, one cannot extract $f_2(X)$ or $f_3(X)$: instead only can only extract the polynomial $cf_2(X) - f_3(X)$.

An obvious solution to this problem is to accompany the commitments of $f_2(X)$ and $f_3(X)$ with knowledge components; however, this makes the zk-SNARK less efficient. We will leave it an interesting open question to investigate this problem and how it influences popular zk-SNARKs like Plonk or Marlin.