On Lifting AGM Security to the AGM with Oblivious Sampling

Juraj Belohorec[1,2], Pavel Hubáček[1,2], and Dominik Stejskal[2]

[1] Institute of Mathematics, Czech Academy of Sciences, Prague, Czech Republic
{belohorec,hubacek}@math.cas.cz
[2] Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic

Abstract. The Algebraic Group Model (AGM) is widely used to prove the security of cryptographic schemes involving groups and pairings. It requires adversaries to explain how their output group elements are obtained from the elements they have received. The AGM with Oblivious Sampling (AGMOS), introduced by Lipmaa, Parisella, and Siim (TCC 2023), additionally allows adversaries to sample group elements without learning their discrete logarithms. This models capabilities such as hashing to a group, but complicates security proofs: security in the AGM does not always imply security in the AGMOS.

We give a framework for transferring AGM security proofs to the AGMOS for a broad class of pairing-based security games. Our results separate the existing AGM argument from the additional reasoning needed to handle obliviously sampled elements. Under the Find Polynomial Representation (FPR) and Tensor Oracle Find Representation (TOFR) assumptions, this additional reasoning can be handled by general reductions. Games with further requirements, such as extractability, also require a lifting condition that ensures these requirements are preserved.

We first establish a lifting theorem for non-interactive games and apply it to KZG extractability, the multivariate Adaptive Rational Strong Diffie–Hellman assumption, and the special-soundness of the Polymath SNARK. We then extend the framework to interactive games that use random oracles and other oracles that depend on the trapdoor. In the interactive setting, we apply the theorem to the existential unforgeability of BLS signatures and strong unforgeability of BBS signatures under the FPR/PDL and TOFR assumptions. For BLS, we also provide an AGM security proof for Type-3 pairings, adapting the original Type-1 analysis.

Our framework allows existing AGM proofs to be reused in the AGMOS. To apply the framework, it is enough to check its requirements, including oracle simulation and, where needed, the lifting condition, without repeating the full security proof.

Keywords: Algebraic Group Model · Oblivious Sampling · AGM · AGMOS · Lifting Security Proofs.

Table of Contents

1 Introduction . . . 3
1.1 Our Contributions . . . 3
1.2 Technical Overview . . . 4
1.3 Other Related Work . . . 10
1.4 Organization of the Paper . . . 10
2 Preliminaries . . . 10
2.1 AGM and AGMOS . . . 12
3 Lifting for Non-Interactive Security Games . . . 14
3.1 Security Games in the AGMOS . . . 14
3.2 Additional Notation . . . 17
3.3 Lifting Theorem . . . 19
3.4 Limitations of Our Lifting Theorem . . . 24
4 Applications of the Non-Interactive Lifting Theorem . . . 25
4.1 Extractability of the KZG Polynomial Commitment Scheme . . . 25
4.2 ARSDH(n) Assumption . . . 28
4.3 Special-Soundness of Polymath . . . 31
5 Lifting for Interactive Security Games . . . 34
5.1 Interactive AGMOS Security Game . . . 34
5.2 Interactive Lifting Theorem . . . 37
6 Applications of the Interactive Lifting Theorem . . . 43
6.1 EUF-CMA Security of the BLS Signature Scheme . . . 43
6.2 SUF-CMA Security of the BBS Signature Scheme . . . 46
A EUF-CMA Security of the BLS Signature Scheme in the AGM with Type-3 Pairings . . . 52
B Interactive Security Game Polynomials . . . 56

1 Introduction

Idealized models have become common in modern provable security. By idealizing the properties of some primitives or adversaries, we can argue rigorously about schemes that we currently cannot analyze in the standard model, providing at least some level of assurance about their security. Classic examples include the Random Oracle Model (ROM) [BR93] and its quantum variant QROM [BDF+11] for hash functions, and the Generic Group Model (GGM) [Sho97, Fis00, Mau05] for hard problems over cyclic groups. Building on the GGM, Fuchsbauer, Kiltz, and Loss [FKL18] introduced the Algebraic Group Model (AGM). The AGM restricts the adversary's power by requiring every new group element it outputs to come with an explicit algebraic representation in terms of the elements it has already seen. Intuitively, this restriction captures a meaningful class of attackers that manipulate the input group elements in an "algebraic" manner. Moreover, it restores many extractor-based proof techniques and has enabled tight, natural security arguments for a wide array of schemes and protocols such as polynomial commitment schemes [CHM+20, BDFG21], SNARKs [MBKM19], and signature schemes [FPS20].

A technical limitation of the AGM is that it does not allow the adversary to sample group elements "obliviously," i.e., without knowing their discrete logarithms. However, it is well known that, in practice, the so-called admissible encodings enable such oblivious sampling without the need for any "non-algebraic" operations on group elements; see, e.g., [BF01, Ica09, WB19]. To address this technical gap, Lipmaa, Parisella, and Siim [LPS23] extended the AGM and proposed the AGM with Oblivious Sampling (AGMOS), in which the adversary also has oracle access to freshly sampled, unlinkable group elements. When introducing the AGMOS, Lipmaa et al. also presented a streamlined AGMOS proof technique using two novel assumptions: Find Polynomial Representation (FPR) and Tensor Oracle Find Representation (TOFR). They then illustrated the technique on several examples. Unsurprisingly, the resulting AGMOS proofs are often extensions of AGM proofs. To quote Lipmaa et al. [LPS23], *"[. . .] in the case of more complicated AGM proofs, adding the AGMOS part to it is 'relatively' easy compared to the AGM proof itself."* Given the general proof template of Lipmaa et al., we ask the following natural question:

Can we generically lift security proofs from AGM to AGMOS?

We give a qualified affirmative answer. A universal lifting theorem is impossible, since Lipmaa et al. [LPS23] already exhibit assumptions that are secure in the AGM but insecure in the AGMOS. Nevertheless, we show that security games based solely on pairing checks lift directly from AGM to AGMOS, while games involving additional winning conditions, such as extractability or multi-transcript consistency requirements, also lift under a sufficient algebraic condition. In our applications, this algebraic condition is simple to check. We further extend the lifting framework to interactive security games. To illustrate the interactive lifting framework, we show security of different signature scheme notions in the AGMOS.

Conceptually, our contribution is to separate the reusable AGM core of such proofs from the part that is genuinely caused by oblivious sampling. The AGM proof rules out attacks that can be explained using the group elements supplied by the challenger. The only new feature of the AGMOS is that the adversary also receives fresh random group elements, but does not know how to express them in terms of the input group elements. Our results show that these extra elements often do not help the adversary: either an attack using them also gives an attack in the ordinary AGM, or it reveals a non-trivial relation among the sampled elements. This means that, in these cases, proving security in the AGMOS does not require starting over, as one can reuse the AGM proof and check the extra conditions introduced by oblivious sampling. This clarifies the relationship between the two models and significantly simplifies the use of AGMOS in future work.

1.1 Our Contributions

For the class of assumptions considered here[3], all our lifting results are proved under the FPR and TOFR assumptions of Lipmaa et al. [LPS23]. We prove the following.

[3] As in [LPS23], by "assumption" we mean "an assumption or the security of a protocol." See Remark 3.

1. **A lifting theorem for non-interactive games.** We give general conditions under which an AGM security proof can be reused to obtain AGMOS security. Games whose winning condition is determined entirely by algebraic verification checks lift directly. Games with additional structural requirements, such as extractability or consistency across several transcripts, can also be lifted once one verifies a simple extra condition.
2. **A lifting theorem for interactive games.** We extend the framework to security games in which the adversary interacts adaptively with the challenger and additional oracles. This deals with the main limitation of the non-interactive setting and allows us to prove standard signature security notions secure in the AGMOS.
3. **Applications.** We apply the framework to obtain AGMOS security for extractability of the KZG polynomial commitment scheme [KZG10], security of the multivariate $\mathsf{ARSDH}(n)$ assumption [BDH+25], special-soundness of Polymath [Lip24], and $\mathsf{EUF\text{-}CMA}$ security of BLS signatures [BLS01]. For the BBS signature scheme, we show $\mathsf{SUF\text{-}CMA}$ security in the AGMOS by adapting the lifting proof. For the BLS application, we additionally give the required AGM proof for Type-3 pairings, adapting the original Type-1 AGM analysis of Fuchsbauer et al. [FKL18]. The applications are chosen to illustrate different features of algebraic security games: KZG captures extractor-based structural checks, ARSDH captures privately verifiable assumptions, Polymath captures multi-transcript special-soundness, BLS captures random oracle signature security with a trivial structural check, and BBS captures strong unforgeability with a non-trivial structural check. Thus, the framework is not tailored to a single example, but covers the main proof patterns that arise in current uses of the AGM and AGMOS.

The main takeaway is that AGMOS security does not require redoing the AGM proof. If the winning condition is a pairing check, AGM security together with FPR and TOFR already implies AGMOS security. If the game also has a structural check, it suffices to verify in addition the lifting condition, namely that the part of the verification polynomial involving obliviously sampled elements is identically zero only if all coefficients of these elements in the adversary's algebraic explanations are zero.

1.2 Technical Overview

In this subsection, we provide a technical overview of our results. We model each assumption as a game between a challenger and an AGMOS adversary. The challenger samples group elements and provides them to the adversary. The adversary then outputs group and field elements, and every output group element comes with an algebraic explanation relative to the inputs it has received and the obliviously sampled elements it has obtained. Our framework allows the adversary to output group elements only in $\mathbb{G}_1$ and $\mathbb{G}_2$. The adversary may still compute pairings and perform operations in $\mathbb{G}_T$, but our results do not cover target group outputs. Finally, the challenger checks whether the adversary's output satisfies the algebraic winning condition and, when relevant, an additional structural requirement. Throughout the overview, we use bracket notation for group elements. If $\mathbb{G}$ is an additive group with fixed generator G and $a \in \mathbb{F}$, then we write $[a] := a \cdot G \in \mathbb{G}$. For a vector $\mathbb{a} = (a_1, \ldots, a_\ell) \in \mathbb{F}^\ell$, we write $[\mathbb{a}] := ([a_1], \ldots, [a_\ell]) \in \mathbb{G}^\ell$ for the component-wise encoding. When the source group matters, we restore the corresponding subscript.

Structural checks. For many assumptions, winning the game amounts only to producing outputs that satisfy algebraic relations, such as pairing equations or simple equalities. These conditions are captured by verification polynomials. Some assumptions, however, impose an additional requirement that is not itself just another algebraic relation on the output group elements. Examples include whether a given extractor produces a valid witness or whether several transcripts are consistent with one another.

We formalize such a requirement by a structural check $\mathcal{S}$. Assumptions without such a condition, such as Diffie–Hellman-type assumptions, are fully described by algebraic relations alone. We say that such assumptions have a trivial structural check. Knowledge-type assumptions and multi-transcript settings, such as extractability and special-soundness, naturally lead to a non-trivial structural check. In KZG extractability, for example, the pairing check is the KZG verification equation, and the structural check asks whether a fixed

extractor, run on the game transcript, recovers a polynomial of the required degree that matches the commitment and the claimed evaluation; the adversary wins if the pairing check passes and the extraction fails. The structural check belongs to the security game rather than to the scheme, since the extractor uses the adversary's algebraic explanations, which a real verifier never sees. Our analysis turns the pairing check into a verification polynomial but never looks inside the structural check, which is why we need the compatibility and lifting conditions below to ensure that the structural check survives the projection to the AGM.

AGM proofs via verification polynomials. In a typical AGM proof, one constructs a multivariate verification polynomial $V(\boldsymbol{X}) = V(X_1, \ldots, X_n)$ from the challenger's algebraic check. The challenger samples a trapdoor vector $\boldsymbol{x}$ and gives the adversary the corresponding public input group elements $[\mathbb{x}(\boldsymbol{x})]$. The adversary outputs group elements $[\mathbb{y}]$, together with algebraic explanations expressing them as linear combinations of the public inputs, say $[\mathbb{y}] = \boldsymbol{\gamma}[\mathbb{x}(\boldsymbol{x})]$. Substituting these explanations into the challenger's algebraic check yields a polynomial equation in the unknown trapdoor variables. The variables $X_1, \ldots, X_n$ therefore represent the entries of the trapdoor vector $\boldsymbol{x}$, from which the public input group elements are derived. Whenever the adversary wins, the challenger's algebraic check holds, which is equivalent to $V(\boldsymbol{x}) = 0$. Thus, the coefficients of V are determined by the adversary's output and explanations, while the point at which V is evaluated is the challenger's secret trapdoor $\boldsymbol{x}$. The AGM proof then distinguishes two cases.

Case A: $V(\boldsymbol{X}) \equiv 0$.
Case X: $V(\boldsymbol{X}) \not\equiv 0$, but $V(\boldsymbol{x}) = 0$.

In case A, one typically shows that the adversary has not actually won. In case X, one typically builds a reduction to a variant of the Power Discrete Logarithm (PDL) assumption. When a non-trivial structural check is present, the adversary must additionally violate that requirement in order to win.

Verification in the AGMOS. The AGMOS adversary is more powerful than an AGM adversary because it also has oracle access to obliviously sampled group elements whose discrete logarithms are unknown to it. Accordingly, its output takes the form

$$[\mathbb{y}] = \boldsymbol{\gamma}[\mathbb{x}] + \boldsymbol{\delta}[\mathbb{q}],$$

where $\boldsymbol{\gamma}$ explains the dependence on the challenger inputs and $\boldsymbol{\delta}$ explains the dependence on the obliviously sampled elements. Substituting these expressions into the challenger's algebraic check yields an AGMOS verification polynomial $V(\boldsymbol{X}, \mathbb{Q})$. Here, the variables $\boldsymbol{X}$ correspond to the discrete logarithms sampled by the challenger, while the variables $\mathbb{Q}$ correspond to the unknown discrete logarithms of the obliviously sampled elements.

The AGMOS verification polynomial and its decomposition. We decompose the AGMOS verification polynomial as

$$V(\boldsymbol{X}, \mathbb{Q}) = V^h(\boldsymbol{X}) + V^t(\boldsymbol{X}, \mathbb{Q}),$$

where $V^h(\boldsymbol{X})$ collects the terms involving only $\boldsymbol{X}$ and the constant term, and $V^t(\boldsymbol{X}, \mathbb{Q})$ collects all terms depending on $\mathbb{Q}$, possibly together with $\boldsymbol{X}$. The polynomial V^h contains the terms already present in the AGM, while V^t contains the terms involving obliviously sampled elements. Whenever an AGMOS adversary succeeds, it must hold that $V(\boldsymbol{x}, \mathbb{q}) = 0$ for the sampled trapdoors $\boldsymbol{x}$ and the oracle-sampled logarithms $\mathbb{q}$. The difficulty is that this equality may now arise from cancellations involving the hidden values $\mathbb{q}$.

The four-case proof template in the AGMOS. Lipmaa et al. [LPS23] organize AGMOS proofs around four cases.

Case A: $V(\boldsymbol{X}, \mathbb{Q}) \equiv 0$.
Case X.1: $V(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{X}, \mathbb{Q}) \equiv 0$.
Case X.2: $V^t(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{x}, \mathbb{Q}) \equiv 0$.
Case Q: $V^t(\boldsymbol{x}, \mathbb{Q}) \not\equiv 0$.

The first two cases are the AGM-like branches. The last two are genuinely new in the AGMOS. If there is a structural check, the adversary wins only if it satisfies $V(\boldsymbol{x}, \mathbb{q}) = 0$ and violates the structural check.

The auxiliary assumptions FPR *and* TOFR. The FPR assumption rules out the kind of accidental vanishing in trapdoor variables that appears in case X.2; in case X.1, the same kind of vanishing is handled by the original AGM proof. It can be viewed as an abstraction of the usual PDL-style reasoning in AGM proofs. The TOFR assumption, introduced in [LPS23], rules out non-trivial linear relations among the discrete logarithms of obliviously sampled elements and handles case Q. Thus, FPR and TOFR handle the additional algebraic cases in the AGMOS. It remains to show how the cases with $V^t \equiv 0$ relate to the original AGM proof.

The lifting idea. For assumptions without a structural check, we can directly use the original AGM proof. When $V^t \equiv 0$, removing the contribution of obliviously sampled elements does not change the result of the algebraic verification. Thus, any successful AGMOS adversary in case A or X.1 gives a successful AGM adversary. For assumptions with a structural check, preserving the pairing check alone is not sufficient, because the transcript must also still violate the structural check after the projection to the AGM. We therefore require the lifting condition

$$V^t(\boldsymbol{X}, \mathbb{Q}) \equiv 0 \qquad \Longrightarrow \qquad \boldsymbol{\delta} = \mathbf{0},$$

which says that in cases A and X.1 the adversary's explanations use no obliviously sampled elements, and we require that the structural check is compatible with the AGM projection, that is, removing the AGMOS-only explanation data does not change its result. Together, the two conditions turn a successful AGMOS adversary in cases A and X.1 into a successful AGM adversary, so the original AGM security proof bounds these cases.

Expanding $V^t(\boldsymbol{X}, \mathbb{Q})$ and equating its coefficients to zero yields a system of polynomial equations that are linear or quadratic in the δ coefficients. To prove the lifting condition, we must show that $\boldsymbol{\delta} = \mathbf{0}$ is the only solution, for every allowed choice of the other explanation coefficients and transcript parameters. In many applications, including ours, this condition is easy to check.

This step uses no computational assumption. The coefficients of V and V^t are determined by the game and by the adversary's output, namely its explanation matrices and the non-group data that fix $\boldsymbol{M}$; $\boldsymbol{X}$ and $\mathbb{Q}$ are indeterminates that stand for the hidden values $\boldsymbol{x}$ and $\mathbb{q}$, exactly as the trapdoor variables do in an AGM proof. Whether $V^t \equiv 0$ is therefore decided by the adversary's output alone, and in that case the lifting condition gives $\boldsymbol{\delta} = \mathbf{0}$ directly. If the adversary instead relies on a relation that holds only for the actual hidden values, for example a non-trivial relation among the discrete logarithms of the obliviously sampled elements, then $V^t \not\equiv 0$ and the execution falls into case X.2 or Q. These are exactly the cases bounded by FPR and TOFR, while cases A and X.1 are bounded by the AGM security of the game.

The lifting condition is a sufficient criterion rather than a complete characterization. For games with a structural check, the difficulty is that $V^t \equiv 0$ may hold even when $\boldsymbol{\delta} \neq \mathbf{0}$. The obliviously sampled elements then cancel out of the verification polynomial, but removing them may change whether the structural check is violated. Known separations between the AGM and the AGMOS use this same idea.

Informal lifting theorem. Let Γ be a non-interactive AGMOS security game, and fix a corresponding AGM variant. Suppose that either both structural checks are trivial, or else $\mathcal{S}$ is compatible with this AGM variant and Γ satisfies the lifting condition above. Then, for every PPT AGMOS adversary $\mathcal{A}$, we can construct a PPT AGM adversary $\mathcal{B}$, a PPT FPR adversary $\mathcal{B}_{\mathsf{fpr}}$, and a PPT TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$ such that

$$\Pr[\mathcal{A} \text{ wins } \Gamma] \leq \Pr[\mathcal{B} \text{ wins the AGM variant of } \Gamma] + \mathsf{Adv}^{\mathsf{FPR}}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Thus, any game satisfying these conditions is secure in the AGMOS if it is secure in the AGM and the FPR and TOFR assumptions hold.

How to use the theorem. Starting from an AGM-secure game, one typically performs three short steps:

1. substitute AGMOS explanations and compute the $\mathbb{Q}$-dependent part V^t;
2. for games with a structural check, verify compatibility with the AGM projection and check that $V^t \equiv 0$ forces $\boldsymbol{\delta} = \mathbf{0}$;
3. invoke the lifting theorem together with the underlying AGM proof.

Illustration: CDH. The CDH assumption is the simplest example of a game without a structural check. In the AGM, an algebraic CDH adversary, on input $[1, a, b]$, outputs a group element $[y]$ together with coefficients $\gamma_1, \gamma_2, \gamma_3$ such that

$$[y] = \gamma_1[1] + \gamma_2[a] + \gamma_3[b].$$

The adversary wins if $[y] = [ab]$. Equivalently, the discrete logarithms satisfy

$$\gamma_1 + \gamma_2 a + \gamma_3 b - ab = 0.$$

Viewing a as the hidden trapdoor variable, this gives the verification polynomial

$$V(X) = \gamma_1 + \gamma_2 X + \gamma_3 b - Xb.$$

If the adversary succeeds, then $V(a) = 0$. The usual AGM argument then distinguishes whether $V \equiv 0$ or $V \not\equiv 0$. In the latter case, the adversary has produced a non-zero polynomial relation that vanishes at the hidden challenge a, which gives the usual reduction to the corresponding PDL/FPR-style hardness assumption.

In the AGMOS, the adversary may additionally use obliviously sampled elements, and its output can have the form

$$[y] = \gamma_1[1] + \gamma_2[a] + \gamma_3[b] + \sum_i \delta_i[\mathfrak{q}_i].$$

The verification polynomial becomes

$$V(X, \mathbb{Q}) = \gamma_1 + \gamma_2 X + \gamma_3 b - Xb + \sum_i \delta_i \mathbb{Q}_i.$$

Hence,

$$V^t(\mathbb{Q}) = \sum_i \delta_i \mathbb{Q}_i.$$

Since V^t does not depend on X, case X.2 does not occur. If $\boldsymbol{\delta} \neq \mathbf{0}$, a successful output gives the non-trivial linear relation $V^h(a) + \sum_i \delta_i \mathfrak{q}_i = 0$ between 1 and the logarithms of the obliviously sampled elements, which breaks TOFR, because the TOFR reduction samples a and b itself and can therefore compute $V^h(a)$. If $\boldsymbol{\delta} = \mathbf{0}$, the output uses only $[1, a, b]$, so it is the output of an AGM adversary, and the rest of the proof is exactly the original AGM argument above: either the verification polynomial is identically zero, or the adversary yields a non-zero polynomial relation in the hidden challenge, giving the usual reduction to an FPR/PDL-style hardness assumption.

Illustration: KZG extractability. In KZG (see Section 4.1), the commitment key is $\mathsf{ck} = \big(([x^i]_1)_{i=0}^d, [1]_2, [x]_2\big)$ for a secret trapdoor x. A commitment to a polynomial f of degree at most d is $C = [f(x)]_1$, and a proof that $f(\alpha) = \eta$ is $\pi = [(f(x) - \eta)/(x - \alpha)]_1$. The adversary outputs C, π, α, and η, and the challenger checks

$$(C - [\eta]_1) \bullet [1]_2 = \pi \bullet [x - \alpha]_2.$$

The adversary explains C and π in terms of ck and its obliviously sampled elements $[\mathfrak{q}_{1,i}]_1$; let $\delta_{1,i}$ and $\delta_{2,i}$ be the coefficients of $[\mathfrak{q}_{1,i}]_1$ in C and π, respectively. Substituting these explanations into the pairing check gives a verification polynomial whose $\mathbb{Q}$-dependent part is

$$V^t(X, \mathbb{Q}) = \sum_i \big(\delta_{1,i} + \alpha\delta_{2,i} - X\delta_{2,i}\big)\mathbb{Q}_{1,i}.$$

If $V^t \equiv 0$, the coefficients of $X\mathbb{Q}_{1,i}$ force $\delta_{2,i} = 0$, and those of $\mathbb{Q}_{1,i}$ then force $\delta_{1,i} = 0$, so the lifting condition holds. Since the extractor-based structural check is compatible with the AGM projection, the known AGM extractability proof for KZG lifts to the AGMOS once FPR and TOFR are assumed.

Interactive games. The discussion so far concerns non-interactive games, in which the algebraic basis available to the adversary is fixed at the beginning of the experiment. Many natural security notions are not of this form. In a signature security game, for example, the adversary may adaptively query a random oracle and a signing oracle before producing its final forgery, and the group elements returned during this interaction may later appear in algebraic explanations of its output. Thus, unlike in the non-interactive setting, the adversary's algebraic basis grows throughout the execution of the game.

We model this by allowing the challenger to maintain a dynamically growing set of group elements. Initially, this set contains the public group elements generated from the trapdoor. During the interaction, it is expanded by group elements returned by a random oracle, by trapdoor-dependent auxiliary oracles such as a signing oracle, and by the oblivious sampling oracle. Every new group element output by the adversary must be explained with respect to the basis available at that point in the interaction. The crucial distinction is that random oracle outputs and trapdoor-dependent oracle replies are part of the security game itself and therefore exist already in the AGM variant, whereas the obliviously sampled elements represent the additional power available only in the AGMOS.

Verification polynomials for interactive games. This distinction between interactive and non-interactive games is reflected in the verification polynomial. Besides the trapdoor variables $\boldsymbol{X}$ and the oblivious sampling variables $\mathbb{Q}$, we introduce variables $\boldsymbol{R}$ for the discrete logarithms of the group elements returned by the random oracle. The group elements returned by a trapdoor-dependent oracle are represented as functions of the trapdoors and of the random oracle outputs obtained so far, and can therefore be absorbed into the $\boldsymbol{X}$- and $\boldsymbol{R}$-dependent part of the algebraic basis. After fixing a full interaction transcript, the final pairing check again yields an implicit verification polynomial, now of the form

$$V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = V^h(\boldsymbol{X}, \boldsymbol{R}) + V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}).$$

Here, V^h contains the part already present in the interactive AGM game, while V^t contains precisely the terms involving the AGMOS-only variables $\mathbb{Q}$. It is important not to put the variables $\boldsymbol{R}$ into $\mathbb{Q}$: random oracle outputs are fresh group elements, but they are generated by the original security game rather than by the additional AGMOS oracle and must remain available in the AGM comparison.

The interactive case distinction. The additional variables $\boldsymbol{R}$ create one new branch in the AGMOS case analysis. For a successful adversary, we now distinguish the following cases:

Case A**:** $V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$;
Case X.1**:** $V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$;
Case X.2**:** $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$;
Case X.3**:** $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) \equiv 0$;
Case Q**:** $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) \not\equiv 0$.

The first two branches are again the AGM-like ones. Case X.2 is handled by an FPR reduction in the trapdoor variables $\boldsymbol{X}$, just as in the non-interactive setting. The new case X.3 arises because the verification polynomial may remain non-zero as a formal polynomial in the random oracle variables $\boldsymbol{R}$, yet vanish after the actual random oracle outputs $\boldsymbol{r}$ are substituted. We handle this branch by a second FPR reduction, now in the random oracle variables. Finally, case Q again yields a non-trivial relation among the obliviously sampled logarithms and is reduced to TOFR.

Lifting in the interactive setting. Once these extra variables are accounted for, the lifting mechanism has the same conceptual shape as before. For games without a structural check, if $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$, then the contribution of the obliviously sampled elements can be removed from an accepting transcript while preserving the final pairing check. For games with a non-trivial structural check, we use the same lifting condition as before, now in the larger set of variables:

$$V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0 \qquad \Longrightarrow \qquad \boldsymbol{\delta} = \boldsymbol{0}.$$

This guarantees that, in the AGM-like branches, the successful transcript already depends only on the algebraic basis that exists in the interactive AGM game, so both acceptance and the structural violation are preserved.

There is one further issue that does not appear in the non-interactive setting. To convert an interactive AGMOS adversary into an interactive AGM adversary, the FPR reductions in cases X.2 and X.3 must simulate the adversary's oracle interactions without knowing the trapdoor. We therefore require that the trapdoor-dependent oracle can be simulated in the reduction, and we restrict both the random oracle and the trapdoor-dependent oracle to non-group inputs. The AGM reduction can forward queries to the corresponding AGM oracles and simulate the oblivious sampling oracle itself, allowing us to use the same cancellation argument or lifting condition as before.

Informal interactive lifting theorem. Let Γ be an interactive AGMOS security game whose trapdoor-dependent oracle can be simulated as above, and fix a corresponding interactive AGM variant. Suppose that either both structural checks are trivial, or else $\mathcal{S}$ is compatible with this AGM variant and Γ satisfies the interactive lifting condition. Then, for every PPT interactive AGMOS adversary $\mathcal{A}$, we can construct a PPT adversary $\mathcal{B}$ for the interactive AGM variant of the same game, two PPT FPR adversaries $\mathcal{B}_{\mathsf{fpr},1}$ and $\mathcal{B}_{\mathsf{fpr},2}$, and a PPT TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$ such that

$$\begin{aligned}\Pr[\mathcal{A} \text{ wins } \Gamma] \le\ & \Pr[\mathcal{B} \text{ wins the interactive AGM variant of } \Gamma] \\ & + \mathsf{Adv}^{\mathsf{FPR}}_{\mathcal{B}_{\mathsf{fpr},1}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR}}_{\mathcal{B}_{\mathsf{fpr},2}}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).\end{aligned}$$

Consequently, if the relevant FPR and TOFR assumptions hold, interactive AGM security implies interactive AGMOS security for every game satisfying these conditions.

Applications to signature schemes. We illustrate the interactive theorem on two signature schemes (see Section 6) that exercise different parts of the framework. For BLS signatures, the adversary interacts with both a random oracle and a signing oracle, and the verification polynomial contains random oracle variables $\boldsymbol{R}$ in addition to the trapdoor variable. Beyond the pairing check, the EUF-CMA winning condition only requires the forged message to be fresh, which is a condition on a bit string rather than on a group element, so the structural check is trivial. The lifting theorem therefore transfers EUF-CMA security from the AGM to the AGMOS directly. Since the required AGM proof for Type-3 pairings was not available in the needed form, we additionally adapt the original Type-1 AGM analysis of Fuchsbauer et al. to the Type-3 setting.

For BBS signatures, the adversary interacts with a signing oracle but no random oracle is needed. A signature on m is $([a]_1, e)$ with a random $e \in \mathbb{F}$ and $[a]_1 = (x+e)^{-1}[c]_1$, where x is the secret key and $[c]_1$ is computed from m and public bases. It is valid if $[a]_1 \bullet (e[g_2]_2 + pk_2) = [c]_1 \bullet [g_2]_2$, where $pk_2 = [xg_2]_2$ is the public key. Strong unforgeability requires the forged pair $(m^*, ([a^*]_1, e^*))$ to be new; since this involves the group element $[a^*]_1$, the structural check is non-trivial. In the forgery, only $[a^*]_1$ can contain obliviously sampled elements, say with coefficients δ_i of $[\mathfrak{q}_{1,i}]_1$, and the verification equation pairs it with $e^*[g_2]_2 + pk_2 = [(x+e^*)g_2]_2$. Hence, with variables X and G_2 for x and g_2, we get

$$V^t(\boldsymbol{X}, \mathbb{Q}) = G_2(X + e^*) \sum_i \delta_i \mathbb{Q}_{1,i},$$

so $V^t \equiv 0$ implies $\boldsymbol{\delta} = \mathbf{0}$ and the lifting condition holds. However, the signing responses are rational functions of the trapdoor, so the interactive theorem does not apply directly. We adapt the proof by using a PDL reduction in case X.2 and simulating signing queries as in Tessaro and Zhu [TZ23]. With these changes, the known AGM proof of SUF-CMA security transfers to the AGMOS. Together, the BLS and BBS analyses illustrate the lifting approach for signature games both with and without a random oracle, and with trivial and non-trivial structural checks.

1.3 Other Related Work

Recent work shows a common pattern that motivates our results. Some works already carry out direct AGMOS analyses, while several others prove security only in the AGM and explicitly leave the AGMOS extension as future work. Although introduced only recently, the AGMOS has been recognized as an important refinement of the AGM. Some recent works used the AGMOS to analyze the security of their assumptions: Lipmaa, Parisella, and Siim [LPS24] analyzed their Adaptive Rational Strong Diffie–Hellman (ARSDH) assumption in the AGMOS. Lipmaa [Lip24] proved the special-soundness of the Polymath argument in the AGMOS. Finally, Faonio, Fiore, and Russo [FFR24] claim that PLONK [GWC19] and Marlin [CHM+20] are knowledge-sound in the AGMOS, though their proofs are not detailed. Other recent works proved security in the AGM but explicitly identified the AGMOS as an important and natural model for future extension of their results: Dellepere et al. [DMS24] proved the knowledge-soundness of their SNARKs Pari and Garuda in the AGM and stated that the results should extend to the AGMOS, but they left a formal proof for future work. Campanelli et al. [CFF+24] analyzed the protocol cq+ in the AGM with ROM, but suggested that proving the security in the AGMOS is an interesting open direction. Nitulescu, Paslis, and Ràfols [NPR26] studied their folding scheme for R1CS instances in the AGM and declared that they do not consider the AGMOS for simplicity. Finally, Wagner and Zapico [WZ24] proved position-binding and code-binding of their erasure code commitment scheme in the AGM. They proposed studying code-binding in the AGMOS as an interesting problem for future work. Compared with these direct AGMOS analyses, our goal is not to introduce a new scheme-specific proof technique. Instead, we abstract the common structure of such proofs. Existing AGMOS proofs typically begin with an AGM-style verification polynomial, and then perform an additional case analysis for the obliviously sampled elements. Our lifting theorems handle this second step: once we have an AGM proof and verify the lifting condition, FPR and TOFR handle the remaining AGMOS cases.

Independently of Lipmaa et al. [LPS23], Bauer et al. [BFHK24] studied the security of the Uber-Knowledge assumption in a related model, where the adversary can leverage hashing to the group to obliviously sample group elements. Aside from the GGM and AGM(OS), Jager and Rupp [JR10] introduced the Semi-Generic Group Model for bilinear groups, where adversaries behave generically in the source groups and arbitrarily in the target group. Jaeger and Mohan [JM24] showed that, under specific conditions, security in the AGM implies security in the GGM. As noted in [LPS25], this AGM-to-GGM lifting result also carries over to the AGMOS.

1.4 Organization of the Paper

In Section 2, we recall the AGM, the AGMOS, and the FPR and TOFR assumptions. In Section 3, we formalize the class of non-interactive games considered in this work, prove the non-interactive lifting theorem, and discuss its limitations. In Section 4, we instantiate the theorem for KZG extractability, the multivariate ARSDH assumption, and the special-soundness of Polymath.

In Section 5, we extend the framework to interactive games. In Section 6, we apply the interactive theorem to the EUF-CMA security of BLS signatures and the SUF-CMA security of BBS signatures. In Appendix A, we establish the AGM security of BLS signatures for Type-3 pairings needed by the interactive application. Finally, in Appendix B, we show how to construct the polynomials and matrices used in the interactive AGMOS framework.

2 Preliminaries

For a distribution D, we write $x \leftarrow D$ to denote that the value x is sampled according to D. A finite set S in place of D implies the uniform distribution on S. By λ, we denote the security parameter. For an algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ denotes that given input x, $\mathcal{A}$ outputs y. By $\mathsf{poly}(\lambda)$, we denote the set $\{f(\lambda) : \exists k \in \mathbb{N},\ f(\lambda) \in O(\lambda^k)\}$. We say that a function $\varepsilon\colon \mathbb{N} \to \mathbb{R}^+$ is negligible if it approaches zero faster than any inverse polynomial. In that case, we write $\varepsilon(\lambda) \in \mathsf{negl}(\lambda)$. By $i \in [a, b]$ where $a, b \in \mathbb{Z}$, we mean that

$i \in \{a, a+1, \ldots, b\}$. By PPT, we denote a probabilistic polynomial-time algorithm, that is, a randomized algorithm that runs in time polynomial in the security parameter. By DPT, we denote a deterministic polynomial-time algorithm, that is, a deterministic algorithm whose running time is polynomial in the security parameter.

A bilinear group generator $\mathsf{PGen}(1^\lambda)$ returns group parameters

$$\mathsf{gp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, [1]_1, [1]_2, [1]_T, e).$$

The groups $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, +)$ are additive cyclic groups of prime order p, generated by $[1]_1$ and $[1]_2$, respectively. The group $(\mathbb{G}_T, \odot)$ is a multiplicative cyclic group of prime order p, generated by $[1]_T$. The map $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a pairing, i.e., a non-degenerate efficiently computable bilinear map. Implicitly, gp contains information about the security parameter λ that was used to generate it. We use the notation $g \bullet h = e(g, h)$ for $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$. Pairings are commonly classified into three types depending on the relationship between $\mathbb{G}_1$ and $\mathbb{G}_2$ (see [GPS08]):

1. **Type-1 (symmetric) pairings.** In this case $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$, and the pairing has the form $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. These pairings are conceptually simple but are rarely used in modern constructions due to efficiency and security limitations.
2. **Type-2 (asymmetric with homomorphism).** Here $\mathbb{G}_1 \neq \mathbb{G}_2$, but there exists an efficiently computable group isomorphism $\varphi\colon \mathbb{G}_2 \to \mathbb{G}_1$. Some protocols can exploit this map, but Type-2 pairings are less efficient and harder to instantiate securely in practice.
3. **Type-3 (fully asymmetric) pairings.** In this setting $\mathbb{G}_1 \neq \mathbb{G}_2$, and there is no efficiently computable isomorphism between the groups. Type-3 pairings are the standard choice in modern pairing-based cryptography because they offer better efficiency and stronger security properties.

Throughout the paper, we work with Type-3 pairings. This assumption is standard in pairing-based constructions such as polynomial commitments and SNARKs, and it matches the setting used in prior work on the AGM. By $\mathbb{F}$ we denote the field $\mathbb{Z}_p$, and for indeterminates $\boldsymbol{X} = (X_1, \ldots, X_k)$,

$$\mathbb{F}^{(\leq d)}[\boldsymbol{X}] = \{f(\boldsymbol{X}) \in \mathbb{F}[\boldsymbol{X}] : \deg(f(\boldsymbol{X})) \leq d\},$$

where $\deg(f(\boldsymbol{X}))$ is the total degree of $f(\boldsymbol{X})$. For $m \in \mathbb{N}$ and $\mathbb{a} = (a_1, \ldots, a_m) \in \mathbb{F}^m$, we denote by $[\mathbb{a}]_\kappa = [a_1, \ldots, a_m]_\kappa$ the vector $(a_1[1]_\kappa, \ldots, a_m[1]_\kappa)$. Despite this notation, vectors are, by default, column vectors.

Definition 1 (Discrete Logarithm (DLOG) Assumption). *Let $\mathsf{PGen}(1^\lambda)$ output bilinear group parameters. We say that the Discrete Logarithm assumption holds for $\mathsf{PGen}(1^\lambda)$ in $\mathbb{G}_1$ if, for every PPT $\mathcal{A}$,*

$$\Pr\left[\mathcal{A}(\mathsf{gp}, [x]_1) = x \;\middle|\; \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda),\ x \leftarrow \mathbb{F}\right] \in \mathsf{negl}(\lambda).$$

An analogous definition applies to $\mathbb{G}_2$.

Definition 2 ((d_1, d_2)-PDL assumption). *Let PGen be a bilinear group generator, and $d_1(\lambda), d_2(\lambda) \in \mathsf{poly}(\lambda)$. We say the (d_1, d_2)-*Power Discrete Logarithm *((d_1, d_2)-PDL) assumption holds for PGen if, for all PPT $\mathcal{A}$,*

$$\Pr\left[\mathcal{A}\left(\mathsf{gp}, \left([x^i]_1\right)_{i=0}^{d_1}, \left([x^i]_2\right)_{i=0}^{d_2}\right) = x \;\middle|\; \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda); x \leftarrow \mathbb{F}\right] \in \mathsf{negl}(\lambda).$$

We also use two efficiency notions for families indexed by the group parameters gp output by $\mathsf{PGen}(1^\lambda)$ for $\lambda \in \mathbb{N}$. A family of distributions $\{\mathcal{D}_{\mathsf{gp}}\}_{\mathsf{gp}}$ is *efficiently sampleable* if a PPT algorithm can output samples from $\mathcal{D}_{\mathsf{gp}}$ given gp. A family of functions $\{f_{\mathsf{gp}}\}_{\mathsf{gp}}$ is *efficiently computable* if a PPT algorithm can compute $f_{\mathsf{gp}}(x)$ given gp and x.

$\mathcal{O}_{\mathsf{OS}}(\kappa, E, D)$:
If $\kappa \notin \{1,2\}$, $E \notin \mathcal{EF}_{\mathsf{gp},\kappa}$ or $D \notin \mathcal{DF}_{\mathsf{gp}}$, return $\bot$.
Sample $s \leftarrow D$.
Compute $[\mathfrak{q}]_\kappa = E(s)$.
Return $(s, [\mathfrak{q}]_\kappa)$.

Fig. 1: The $(\mathcal{EF}, \mathcal{DF})$-AGMOS oracle $\mathcal{O}_{\mathsf{OS}}$.

2.1 AGM and AGMOS

Definition 3 (AGM [FKL18]). *A probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ is* algebraic *if every group element it outputs is accompanied by an* algebraic representation *(or* explanation*) with respect to the group elements it has received in the corresponding group. Symbolically,*

$$\left([\mathbb{y}_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa\right)_{\kappa=1}^{2} \leftarrow \mathcal{A}(\mathsf{gp}, [\mathbb{x}_1]_1, [\mathbb{x}_2]_2),$$

where, for each $\kappa \in \{1,2\}$, the matrix $\boldsymbol{\gamma}_\kappa$ is over $\mathbb{F}$ and satisfies

$$\boldsymbol{\gamma}_\kappa \mathbb{x}_\kappa = \mathbb{y}_\kappa.$$

In an interactive execution, the input vector includes all group elements received so far in the corresponding group, including the group generator and group-valued oracle responses. In the AGM*, adversaries are restricted to algebraic PPT algorithms.*

Remark 1 (Model conventions). We follow the direct output convention of [FKL18]: each group element output by the adversary is accompanied by an algebraic explanation relative to the group elements received so far in the corresponding group. We use this convention also for the AGMOS, including oblivious sampling responses in the explanation basis. The original AGMOS formulation [LPS23] instead obtains these explanations through an extractor given the adversary's input and randomness.

To formalize the restriction in [FKL18, Section 2.1] on non-group inputs depending on group elements, we adopt Zhandry's type-safe game convention [Zha22, Sections 8.1–8.2], extended to our bilinear setting. The challenger treats $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ as separate types and accesses group elements through group operations, equality tests, and the pairing. It may use equality-test results in subsequent computation, but may not inspect group encodings or expose their bits as non-group inputs. These restrictions apply to the challenger; the adversary may compute on actual encodings, subject to the explanation requirement.

Definition 4 (AGMOS [LPS23]). *Fix $\mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda)$. For each $\kappa \in \{1,2\}$, let $\mathcal{EF}_{\mathsf{gp},\kappa}$ be a set of polynomially many efficiently computable functions $E\colon \mathbb{F} \to \mathbb{G}_\kappa$, and let $\mathcal{DF}_{\mathsf{gp}}$ be a family of efficiently sampleable distributions on $\mathbb{F}$. We write $\mathcal{EF} = \{\mathcal{EF}_{\mathsf{gp},\kappa}\}_{\mathsf{gp},\kappa}$ and $\mathcal{DF} = \{\mathcal{DF}_{\mathsf{gp}}\}_{\mathsf{gp}}$ for the corresponding families.*

The family $\mathcal{EF}$ specifies the allowed encodings into the source groups, while $\mathcal{DF}$ specifies the allowed distributions of their inputs. On a query (κ, E, D), the oblivious sampling oracle $\mathcal{O}_{\mathsf{OS}}$ samples $s \leftarrow D$ and returns

$$(s, E(s)) = (s, [\mathfrak{q}]_\kappa).$$

Thus, the adversary learns both the sampled input s and its encoding $E(s)$, but is not given the discrete logarithm $\mathfrak{q}$. Knowing s need not reveal $\mathfrak{q}$. The oracle is non-programmable: the adversary chooses an allowed encoding and distribution, but the oracle performs the sampling. Its precise behavior is specified in Figure 1.

*A PPT algorithm $\mathcal{A}$ is an $(\mathcal{EF}, \mathcal{DF})$-*AGMOS adversary *if, given access to $\mathcal{O}_{\mathsf{OS}}$, every source group element it outputs is accompanied by an* algebraic representation*, or* explanation*, with respect to its input elements and the obliviously sampled elements in the same group. Symbolically,*

$$([\mathbb{y}_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathfrak{q}_\kappa]_\kappa)_{\kappa=1}^{2} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{OS}}}(\mathsf{gp}, [\mathbb{x}_1]_1, [\mathbb{x}_2]_2),$$

$\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m,\boldsymbol{x}}(\kappa, f(\boldsymbol{X}))$:
If $\kappa \in \{1,2,T\}$ and $f(\boldsymbol{X}) \in \mathbb{F}[\boldsymbol{X}]$ satisfies $\deg_{X_i}(f(\boldsymbol{X})) \le d_\kappa$ for all i, then return $[f(\boldsymbol{x})]_\kappa$.
Otherwise, return $\perp$.

Fig. 2: The $(\boldsymbol{d}, m)$-FPR oracle $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m,\boldsymbol{x}}(\cdot)$.

where $[\mathfrak{q}_{1\kappa}]_\kappa$ *is the ordered list of group elements returned by* $\mathcal{O}_{\mathsf{OS}}$ *in* $\mathbb{G}_\kappa$*, and the explanation matrices over* $\mathbb{F}$ *satisfy*

$$\boldsymbol{\gamma}_\kappa \mathbb{x}_\kappa + \boldsymbol{\delta}_\kappa \mathfrak{q}_{1\kappa} = \mathbb{y}_\kappa \qquad (\kappa \in \{1,2\}).$$

Here, $\boldsymbol{\gamma}_\kappa$ *describes the contribution of the input elements, while* $\boldsymbol{\delta}_\kappa$ *describes the contribution of the obliviously sampled elements.*

Remark 2. See [LPS23] for a discussion about the choice of $\mathcal{EF}$, $\mathcal{DF}$. The authors mentioned *admissible encodings* $E \in \mathcal{EF}_{\mathsf{gp},\kappa}$ for elliptic curve groups $\mathbb{G}_\kappa$ as an example of efficient oblivious sampling. According to the definition in [LPS23], an admissible encoding is an efficiently computable function with small preimage sizes, for which the preimage of any group element in its image can be efficiently recovered. See [BF01, Ica09] for concrete examples. Not every choice of $\mathcal{EF}$, $\mathcal{DF}$ results in a meaningful model. For example, to achieve *oblivious* sampling, computing the discrete logarithm $\mathfrak{q}$ of $E(s)$ from the oracle answer $(s, [\mathfrak{q}]_\kappa)$ should be hard. In fact, proofs in the AGMOS rely on a stronger assumption TOFR, defined below.

Definition 5 (TOFR). *Let* $\mathcal{EF}$, $\mathcal{DF}$, $\mathcal{O}_{\mathsf{OS}}$, $\mathfrak{q}_\kappa$ *be as in Definition 4. For* $\kappa \in \{1,2\}$*, let* ql_κ *denote the number of oblivious sampling queries that the adversary makes in* $\mathbb{G}_\kappa$*. We say that* PGen *is* $(\mathcal{EF}, \mathcal{DF})$-TOFR (Tensor Oracle Find Representation) *secure if, for any PPT adversary* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{l|l} \boldsymbol{v} \in \mathbb{F}^{1+\mathsf{ql}_1+\mathsf{ql}_2+\mathsf{ql}_1\mathsf{ql}_2} \ \wedge \ \boldsymbol{v} \neq \boldsymbol{0} & \\ \wedge \ \boldsymbol{v}^T \begin{pmatrix} 1 \\ \mathfrak{q}_1 \\ \mathfrak{q}_2 \\ \mathfrak{q}_1 \otimes \mathfrak{q}_2 \end{pmatrix} = 0 & \begin{array}{l} \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda); \\ \boldsymbol{v} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{OS}}}(\mathsf{gp}) \end{array} \end{array}\right] \in \mathsf{negl}(\lambda),$$

where outputs $\boldsymbol{v}$ *of invalid length are rejected, and for* $\mathfrak{q}_1 = (\mathfrak{q}_{1,1}, \ldots, \mathfrak{q}_{1,\mathsf{ql}_1})^T$ *and* $\mathfrak{q}_2 = (\mathfrak{q}_{2,1}, \ldots, \mathfrak{q}_{2,\mathsf{ql}_2})^T$ *we have*

$$\mathfrak{q}_1 \otimes \mathfrak{q}_2 = \left(\mathfrak{q}_{1,1}\mathfrak{q}_{2,1},\ \mathfrak{q}_{1,1}\mathfrak{q}_{2,2},\ \ldots,\ \mathfrak{q}_{1,1}\mathfrak{q}_{2,\mathsf{ql}_2},\ \mathfrak{q}_{1,2}\mathfrak{q}_{2,1},\ \ldots,\ \mathfrak{q}_{1,\mathsf{ql}_1}\mathfrak{q}_{2,\mathsf{ql}_2}\right)^T.$$

In other words, the probability that $\mathcal{A}$ wins the following TOFR *game* is negligible. Given group parameters gp and access to the AGMOS oracle $\mathcal{O}_{\mathsf{OS}}$, defined in Figure 1, $\mathcal{A}$ must find a non-trivial linear relation between 1, $\mathfrak{q}_{1,i}$, $\mathfrak{q}_{2,j}$, $\mathfrak{q}_{1,i}\mathfrak{q}_{2,j}$, where $1 \le i \le \mathsf{ql}_1$, $1 \le j \le \mathsf{ql}_2$. Here, for $\kappa \in \{1,2\}$, ql_κ (query length) is the number of oblivious sampling queries in $\mathbb{G}_\kappa$, i.e., $\mathfrak{q}_\kappa \in \mathbb{F}^{\mathsf{ql}_\kappa}$. TOFR is the first of the two key assumptions introduced in [LPS23]; it is specific to the AGMOS, and its security depends on the parameters $\mathcal{EF}$, $\mathcal{DF}$ (see Remark 2).

Definition 6 (FPR). *Let* $m \ge 1$*,* $d_1, d_2, d_T, d_g \ge 0$ *and* $\boldsymbol{d} = (d_1, d_2, d_T, d_g)$*. Let* $\boldsymbol{X} = (X_1, \ldots, X_m)$ *be indeterminates. For* $\boldsymbol{x} \in \mathbb{F}^m$*, let* $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m,\boldsymbol{x}}(\cdot)$ *be the oracle defined in Figure 2. We say that* PGen *is* $(\boldsymbol{d}, m)$-FPR (Find Polynomial Representation) secure *if, for any PPT adversary* $\mathcal{A}$*,*

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

In other words, the probability that $\mathcal{A}$ wins the following FPR *game* is negligible. The adversary is given group parameters gp and access to the FPR oracle. The oracle, on input $(\kappa, f(\boldsymbol{X}))$, checks if $f(\boldsymbol{X})$ satisfies

certain degree bounds. If it does, it evaluates $f(\boldsymbol{X})$ on the secret trapdoor $\boldsymbol{x}$ and returns the corresponding $\mathbb{G}_\kappa$ element $[f(\boldsymbol{x})]_\kappa$. The adversary's task is to produce a non-zero polynomial $g(\boldsymbol{X})$ satisfying a certain degree bound and $g(\boldsymbol{x}) = 0$. FPR is the second new assumption from [LPS23]; Lipmaa et al. reduce it to the Power Discrete Logarithm (PDL) assumption [Lip12] in the standard model and then use it to partially automate reductions to PDL.

3 Lifting for Non-Interactive Security Games

In this section, we define the class of non-interactive security games that we analyze, introduce the notation for their verification polynomials, and prove our AGM-to-AGMOS lifting theorem. Under the FPR and TOFR assumptions, games without a structural check lift directly from the AGM to the AGMOS, and games with a structural check lift if they also satisfy the lifting condition. We end with the limitations of the theorem, including the separation result of [LPS23]. Throughout the section, we distinguish privately and publicly verifiable assumptions, and assumptions with and without a structural check.

Privately verifiable assumptions: An assumption is *privately verifiable* if, in addition to the adversary's output and the public parameters $(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2)$, the pairing check relies on additional auxiliary group elements, namely $[\mathsf{aux}_1(\boldsymbol{x})]_1$ and $[\mathsf{aux}_2(\boldsymbol{x})]_2$, in order to determine whether the algebraic winning condition has been satisfied. Importantly, these auxiliary group elements are not revealed to the adversary and are used solely by the challenger during verification. For instance, the $\mathsf{ARSDH}(n)$ assumption (see Section 4.2) is privately verifiable.

Publicly verifiable assumptions: An assumption is *publicly verifiable* if, from just the adversary's output and the public parameters $(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2)$, anyone can decide whether the pairing check has been satisfied. For instance, the DLOG assumption is publicly verifiable: given the challenge $[x]_1$, one only needs to confirm that the adversary's output y satisfies $[y]_1 = [x]_1$, which is possible with no secret information required. Thus, the public/private distinction is not whether verification is expressed as a pairing check, but whether all group elements needed to evaluate that check are public.

Without a structural check: These are assumptions where the winning condition is determined solely by a pairing check on the adversary's output group elements, possibly together with simple algebraic conditions on non-group outputs. Typical examples include the Computational Diffie–Hellman (CDH) and Strong Diffie–Hellman (SDH) assumptions.

With a structural check: Their winning condition also contains a requirement on the adversary's group elements that is not a pairing check, which we call a structural check (see Section 3.1). For example, in extractability of polynomial commitment schemes, the challenger verifies that a given extractor can recover a polynomial satisfying a required relation. Similarly, in the $\mathsf{ARSDH}(n)$ assumption, the challenger must ensure that the output group elements do not exhibit any trivial algebraic dependencies.

3.1 Security Games in the AGMOS

We now describe the non-interactive security games that we analyze. The same framework covers privately and publicly verifiable assumptions; in the publicly verifiable case, the auxiliary group elements $[\mathsf{aux}_1(\boldsymbol{x})]_1$ and $[\mathsf{aux}_2(\boldsymbol{x})]_2$ are empty. Every game includes a structural check, which is trivial for games that do not impose one. Examples with a non-trivial structural check include the extractability of the KZG polynomial commitment scheme (Section 4.1) and the privately verifiable $\mathsf{ARSDH}(n)$ assumption (Section 4.2).

AGMOS security game. The AGMOS security game that we analyze is defined as a family of games $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ indexed by the security parameter λ, where each game

$$\Gamma_\lambda = (\mathsf{PGen}, \mathsf{KPGen}, \mathsf{MGen}, \mathcal{C}, \boldsymbol{d}, \mathcal{S})$$

is specified as follows.

Here PGen is a bilinear group generator that, on input 1^λ, outputs group parameters

$$\mathsf{gp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, [1]_1, [1]_2, [1]_T, e).$$

Furthermore, KPGen outputs trapdoor-dependent polynomials satisfying the degree bounds $\boldsymbol{d} = (d_1, d_2, d_T, d_g)$ describing how the public inputs and auxiliary group elements are derived from the trapdoor. The algorithm MGen produces the verification matrix, while $\mathcal{C}$ denotes the challenger algorithm described below.

Finally, $\mathcal{S}$ denotes a *structural check*, which is a Boolean predicate on transcripts. A *transcript* tr consists of the public parameters gp, the public inputs, the adversary's oblivious sampling queries and responses in order, and its final output. Explicitly, we write

$$\mathsf{tr} = \Big(\mathsf{gp},\, ([\mathbb{x}_\kappa(\boldsymbol{x})]_\kappa)_{\kappa=1}^2,\, ([\mathsf{aux}_\kappa(\boldsymbol{x})]_\kappa)_{\kappa=1}^2,\, ([\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2,\, \big(([\mathbb{y}_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha}\big)\Big).$$

Definition 7 (Structural check). *Let $\mathcal{T}_\lambda$ denote the set of transcripts of Γ_λ. A structural check is a predicate*

$$\mathcal{S}_\lambda \colon \mathcal{T}_\lambda \to \{0,1\}$$

specifying the part of the adversary's winning condition that involves the adversary's group elements and is not expressed by the pairing check. The adversary wins if and only if the pairing check holds, the game's conditions on non-group data (if any) hold, and $\mathcal{S}_\lambda(\mathsf{tr}) = 1$. Such conditions, for example the freshness of a message in a signature game, are unaffected by the projection to the AGM and do not make a structural check non-trivial.

We call the structural check trivial if $\mathcal{S}_\lambda \equiv 1$ for every λ. We omit the subscript λ when it is clear from context.

The transcript is that of the entire security game and may contain several protocol transcripts, as in the special-soundness experiment of Section 4.3. Evaluating $\mathcal{S}$ may involve running a fixed algorithm on the transcript; in extraction games, this algorithm is an extractor, and $\mathcal{S}$ equals 1 when the extractor fails to return an output satisfying the required relation. The extractor and its input are specified in each application. A structural check may itself contain a further pairing check, provided it is compatible with the AGM variant (defined below). If a game has several pairing checks, however, we keep all of them as pairing checks rather than placing some inside the structural check, and we require the lifting condition jointly for their verification polynomials, as for Polymath in Section 4.3; this joint condition is easier to satisfy.

The game Γ_λ between a challenger and an $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$ proceeds as follows.

1. The challenger runs $\mathsf{PGen}(1^\lambda)$ to obtain gp.
2. The challenger samples a vector of trapdoors $\boldsymbol{x} \leftarrow \mathbb{F}^m$ (unknown to the adversary), and lets $\boldsymbol{X}$ denote the corresponding vector of indeterminates.
3. The challenger runs
$$(\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X})) \leftarrow \mathsf{KPGen}(1^\lambda, \mathsf{gp}),$$
such that
$$\begin{aligned} &\mathbb{x}_{11}(\boldsymbol{X}) = \mathbb{x}_{21}(\boldsymbol{X}) = 1, \qquad &&\mathbb{x}_{\kappa i}(\boldsymbol{X}) \in \mathbb{F}[\boldsymbol{X}], \\ &\deg_{X_j}\big(\mathbb{x}_{\kappa i}(\boldsymbol{X})\big) \le d_\kappa, \qquad &&\deg_{X_j}\big(\mathsf{aux}_{\kappa i}(\boldsymbol{X})\big) \le d_\kappa, \end{aligned} \tag{1}$$
for $\kappa \in \{1,2\}$, every entry i, and every $j \in [1, m]$, where these polynomials describe how the public group elements and auxiliary group elements are derived from the trapdoor, and satisfy the prescribed degree bounds. We additionally require
$$d_g \ge \max(md_1, md_2).$$
4. The adversary receives the public inputs
$$(\mathsf{gp},\, [\mathbb{x}_1(\boldsymbol{x})]_1,\, [\mathbb{x}_2(\boldsymbol{x})]_2).$$
5. The adversary makes ql_κ queries to the $(\mathcal{EF}, \mathcal{DF})$ oblivious sampling oracle in each group $\mathbb{G}_\kappa$ and receives outputs $[\mathbb{q}_\kappa]_\kappa$, for $\kappa \in \{1,2\}$.

6. The adversary outputs
$$(([\mathbb{y}_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_{1\kappa}]_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha}),$$
where $\boldsymbol{\alpha} \in \mathbb{F}^r$ collects the adversary's non-group outputs, for a length $r \in \mathsf{poly}(\lambda)$ fixed by the game, and the explanation matrices satisfy
$$\boldsymbol{\gamma}_\kappa\, \mathbb{x}_\kappa(\boldsymbol{x}) + \boldsymbol{\delta}_\kappa\, \mathbb{q}_{1\kappa} = \mathbb{y}_\kappa \qquad (\kappa \in \{1,2\}).$$
7. The challenger runs $\mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha})$ to obtain a verification matrix $\boldsymbol{M}$.
8. The adversary wins Γ_λ if and only if all of the following hold:
   (a) The algebraic verification accepts, i.e., $V^{\mathsf{expl}}(\boldsymbol{x}, \mathbb{y}_1, \mathbb{y}_2) = 0$, or equivalently,
   $$\begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \mathbb{y}_1 \\ \mathsf{aux}_1(\boldsymbol{x}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{x}) \\ \mathbb{y}_2 \\ \mathsf{aux}_2(\boldsymbol{x}) \end{pmatrix} = 0. \tag{2}$$
   (b) The structural condition is violated, i.e., $\mathcal{S}(\mathsf{tr}) = 1$.
   (c) The game's conditions on non-group data (if any) hold, for example the requirement $|S| = d'_n + 1$ in the $\mathsf{ARSDH}(n)$ game of Section 4.2.

We say that $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ is hard if for every PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the game described above is negligible, i.e.,
$$\Pr[\mathcal{A} \text{ wins } \Gamma_\lambda] \in \mathsf{negl}(\lambda).$$
We also write Γ for the family $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ and call $\mathsf{Adv}^{\Gamma}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins } \Gamma_\lambda]$ the advantage of $\mathcal{A}$ in Γ.

Remark 3. It is convenient to formulate cryptographic assumptions in terms of security games. We therefore identify an assumption with its associated family of games $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$, defined relative to an oblivious sampling oracle $(\mathcal{EF}, \mathcal{DF})$. The assumption corresponding to $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ states that, for all $\lambda \in \mathbb{N}$ and for all PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversaries $\mathcal{A}$ interacting with Γ_λ,
$$\Pr[\mathcal{A} \text{ wins } \Gamma_\lambda] \in \mathsf{negl}(\lambda).$$

Remark 4. For the AGMOS game Γ_λ, we fix a corresponding AGM game with the same setup, pairing check, and conditions on non-group data, but without access to the oblivious sampling oracle. We call it the AGM variant, denote it by Γ^{AGM}, and denote its structural check by $\mathcal{S}_{\mathrm{AGM}}$. When applying our lifting theorem, this must be the AGM game whose security we assume or establish.

Compatibility of structural checks with the AGM variant. For an AGMOS transcript tr as defined above, let
$$\mathbb{y}'_\kappa := \boldsymbol{\gamma}_\kappa \mathbb{x}_\kappa(\boldsymbol{x}), \qquad \kappa \in \{1,2\}.$$
We define its *AGM projection* by
$$\pi_{\mathrm{AGM}}(\mathsf{tr}) := \Big(\mathsf{gp},\, ([\mathbb{x}_\kappa(\boldsymbol{x})]_\kappa)_{\kappa=1}^2,\, ([\mathsf{aux}_\kappa(\boldsymbol{x})]_\kappa)_{\kappa=1}^2,\, \big(([\mathbb{y}'_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha}\big)\Big).$$
The projection does not change the public inputs and auxiliary information, removes the oblivious sampling queries and responses, and replaces the final output with its AGM explanation and the corresponding group elements. As above, auxiliary information is omitted when unused. When $\boldsymbol{\delta} = \mathbf{0}$, the output group elements remain unchanged.

Let $\mathcal{S}_{\mathrm{AGM}}$ be the structural check in the AGM variant of the game. We say that $\mathcal{S}$ is *compatible with the AGM variant* if, for every AGMOS transcript tr with $\boldsymbol{\delta} = \mathbf{0}$,
$$\mathcal{S}(\mathsf{tr}) = \mathcal{S}_{\mathrm{AGM}}\big(\pi_{\mathrm{AGM}}(\mathsf{tr})\big).$$
We require this agreement only when $\boldsymbol{\delta} = \mathbf{0}$, since the lifting condition ensures this in the cases where we apply AGM security. For a trivial structural check, we take $\mathcal{S}_{\mathrm{AGM}} \equiv 1$.

Remark 5 (Pairing check and matrix form). Pairing checks can be written as matrix equations. In the setting of Section 3.1, target group expressions are formed from pairings using group operations and scalar exponentiation. Each equality between such expressions can be written as

$$\boldsymbol{u}^T \boldsymbol{M} \boldsymbol{v} = 0,$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ contain the discrete logarithms of the source group inputs, and $\boldsymbol{M}$ contains the coefficients of the check. If verification requires s pairing checks, we use matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_s$ and require all corresponding equations to hold. These matrix equations accept exactly the same transcripts as the original pairing checks.

3.2 Additional Notation

Explicit verification polynomial of a security game. We present the notation for privately verifiable assumptions and give the changes for publicly verifiable ones at the end of this subsection. Let $\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$ and $\mathbb{Y} = (\mathbb{Y}_1, \mathbb{Y}_2)$ be the corresponding vectors of indeterminates for $\mathbb{q} = (\mathbb{q}_1, \mathbb{q}_2)$ and $\mathbb{y} = (\mathbb{y}_1, \mathbb{y}_2)$, respectively. Equation (2) may be written as $V^{\mathsf{expl}}(\boldsymbol{x}, \mathbb{y}_1, \mathbb{y}_2) = 0$, where

$$V^{\mathsf{expl}}(\boldsymbol{X}, \mathbb{Y}_1, \mathbb{Y}_2) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Y}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Y}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix}$$

is the *explicit verification polynomial*. In the examples below, we list the entries of $\mathbb{Y}_\kappa$ individually instead of writing the vectors, and omit $\mathbb{Y}_2$ when it is empty.

Implicit verification polynomial for privately verifiable assumptions. Let

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \boldsymbol{M}_{13} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \boldsymbol{M}_{23} \\ \boldsymbol{M}_{31} & \boldsymbol{M}_{32} & \boldsymbol{M}_{33} \end{pmatrix},$$

where $\boldsymbol{M}_{11} \in \mathbb{F}^{\mathsf{il}_1 \times \mathsf{il}_2}, \boldsymbol{M}_{12} \in \mathbb{F}^{\mathsf{il}_1 \times \mathsf{ol}_2}, \boldsymbol{M}_{13} \in \mathbb{F}^{\mathsf{il}_1 \times a_2}$, etc. Let $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2)$ and $\boldsymbol{\Delta} = (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2)$ be (matrix) indeterminates corresponding to $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ and $\boldsymbol{\delta} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)$, respectively. Let

$$\boldsymbol{P}_\kappa(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \boldsymbol{I}_{\mathsf{il}_\kappa} & \boldsymbol{0}_{\mathsf{il}_\kappa \times \mathsf{ql}_\kappa} & \boldsymbol{0}_{\mathsf{il}_\kappa \times a_\kappa} \\ \boldsymbol{\Gamma}_\kappa & \boldsymbol{\Delta}_\kappa & \boldsymbol{0}_{\mathsf{ol}_\kappa \times a_\kappa} \\ \boldsymbol{0}_{a_\kappa \times \mathsf{il}_\kappa} & \boldsymbol{0}_{a_\kappa \times \mathsf{ql}_\kappa} & \boldsymbol{I}_{a_\kappa} \end{pmatrix}$$

be the matrix so that

$$\begin{pmatrix} \mathbb{x}_\kappa(\boldsymbol{x}) \\ \mathbb{y}_\kappa \\ \mathsf{aux}_\kappa(\boldsymbol{x}) \end{pmatrix} = \boldsymbol{P}_\kappa(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_\kappa(\boldsymbol{x}) \\ \mathbb{q}_\kappa \\ \mathsf{aux}_\kappa(\boldsymbol{x}) \end{pmatrix}.$$

Define

$$\boldsymbol{N}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \boldsymbol{P}_1^T(\boldsymbol{\Gamma}, \boldsymbol{\Delta})\, \boldsymbol{M}\, \boldsymbol{P}_2(\boldsymbol{\Gamma}, \boldsymbol{\Delta})$$

$$= \left(\begin{array}{c|c|c} \begin{array}{c} \boldsymbol{M}_{11} + \boldsymbol{M}_{12}\boldsymbol{\Gamma}_2 + \boldsymbol{\Gamma}_1^T\boldsymbol{M}_{21} \\ + \boldsymbol{\Gamma}_1^T\boldsymbol{M}_{22}\boldsymbol{\Gamma}_2 \end{array} & (\boldsymbol{M}_{12} + \boldsymbol{\Gamma}_1^T\boldsymbol{M}_{22})\boldsymbol{\Delta}_2 & \boldsymbol{M}_{13} + \boldsymbol{\Gamma}_1^T\boldsymbol{M}_{23} \\ \hline \boldsymbol{\Delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\Gamma}_2) & \boldsymbol{\Delta}_1^T\boldsymbol{M}_{22}\boldsymbol{\Delta}_2 & \boldsymbol{\Delta}_1^T\boldsymbol{M}_{23} \\ \hline \boldsymbol{M}_{31} + \boldsymbol{M}_{32}\boldsymbol{\Gamma}_2 & \boldsymbol{M}_{32}\boldsymbol{\Delta}_2 & \boldsymbol{M}_{33} \end{array}\right).$$

Equation (2) may now be written as $V(\boldsymbol{x}, \mathbb{q}) = 0$, where

$$\begin{aligned} V(\boldsymbol{X}, \mathbb{Q}) &= V^{\mathsf{expl}}(\boldsymbol{X}, \boldsymbol{\gamma}_1\mathbb{x}_1(\boldsymbol{X}) + \boldsymbol{\delta}_1\mathbb{Q}_1, \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_2\mathbb{Q}_2) \\ &= \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Q}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{N}(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix} \end{aligned}$$

is the *implicit verification polynomial*.

Separating the $\mathbb{Q}$ terms. We decompose the polynomial $V(\boldsymbol{X}, \mathbb{Q})$ as $V(\boldsymbol{X}, \mathbb{Q}) = V^h(\boldsymbol{X}) + V^t(\boldsymbol{X}, \mathbb{Q})$, where $V^h(\boldsymbol{X})$ collects all terms that depend only on $\boldsymbol{X}$ or are constant, and $V^t(\boldsymbol{X}, \mathbb{Q})$ collects all terms that depend either on the $\mathbb{Q}_{\kappa,i}$ variables alone or jointly on both $\boldsymbol{X}$ and the $\mathbb{Q}_{\kappa,i}$ variables.

Then the $V^h(\boldsymbol{X})$ part of the implicit verification polynomial is

$$\begin{aligned} V^h(\boldsymbol{X}) = {} & \mathbb{x}_1^T(\boldsymbol{X})(\boldsymbol{M}_{11} + \boldsymbol{\Gamma}_1^T \boldsymbol{M}_{21} + \boldsymbol{M}_{12}\boldsymbol{\Gamma}_2 + \boldsymbol{\Gamma}_1^T \boldsymbol{M}_{22}\boldsymbol{\Gamma}_2)\mathbb{x}_2(\boldsymbol{X}) \\ & + \mathbb{x}_1^T(\boldsymbol{X})(\boldsymbol{M}_{13} + \boldsymbol{\Gamma}_1^T \boldsymbol{M}_{23})\mathsf{aux}_2(\boldsymbol{X}) + \mathsf{aux}_1^T(\boldsymbol{X})(\boldsymbol{M}_{31} + \boldsymbol{M}_{32}\boldsymbol{\Gamma}_2)\mathbb{x}_2(\boldsymbol{X}) \\ & + \mathsf{aux}_1^T(\boldsymbol{X})\boldsymbol{M}_{33}\mathsf{aux}_2(\boldsymbol{X}). \end{aligned} \tag{3}$$

It is easy to see that the polynomial $V^h(\boldsymbol{X})$ does not depend on the $\boldsymbol{\Delta}$'s. The $V^t(\boldsymbol{X}, \mathbb{Q})$ part of the implicit verification polynomial is

$$V^t(\boldsymbol{X}, \mathbb{Q}) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Q}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{f}_{\boldsymbol{M}}^t(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix}, \tag{4}$$

where

$$\boldsymbol{f}_{\boldsymbol{M}}^t(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{il}_2} & (\boldsymbol{M}_{12} + \boldsymbol{\Gamma}_1^T \boldsymbol{M}_{22})\boldsymbol{\Delta}_2 & \boldsymbol{0}_{\mathsf{il}_1 \times a_2} \\ \boldsymbol{\Delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\Gamma}_2) & \boldsymbol{\Delta}_1^T \boldsymbol{M}_{22}\boldsymbol{\Delta}_2 & \boldsymbol{\Delta}_1^T \boldsymbol{M}_{23} \\ \boldsymbol{0}_{a_1 \times \mathsf{il}_2} & \boldsymbol{M}_{32}\boldsymbol{\Delta}_2 & \boldsymbol{0}_{a_1 \times a_2} \end{pmatrix} \tag{5}$$

is equal to $\boldsymbol{N}(\boldsymbol{\Gamma}, \boldsymbol{\Delta})$ except that its corner submatrices are $\boldsymbol{0}$. To emphasize how $V^t(\boldsymbol{X}, \mathbb{Q})$ is constructed, let us denote

$$V_{\mathbb{x},\mathsf{aux},\boldsymbol{M},\boldsymbol{\Gamma},\boldsymbol{\Delta}}^t(\boldsymbol{X}, \mathbb{Q}) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Q}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{f}_{\boldsymbol{M}}^t(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix}, \tag{6}$$

where $\mathbb{x} = (\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}))$ and $\mathsf{aux} = (\mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X}))$. Notice that every occurrence of $\boldsymbol{\Delta}_1$ or $\boldsymbol{\Delta}_2$ appears exclusively in the component $V^t(\boldsymbol{X}, \mathbb{Q})$.

Publicly verifiable assumptions. In publicly verifiable assumptions, the auxiliary vectors are empty, and therefore the explicit verification polynomial simplifies to

$$V^{\mathsf{expl}}(\boldsymbol{X}, \mathbb{Y}_1, \mathbb{Y}_2) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Y}_1 \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Y}_2 \end{pmatrix}.$$

Therefore, the matrices $\boldsymbol{M}$ and $\boldsymbol{P}_\kappa$ for $\kappa \in \{1, 2\}$ take the form

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{pmatrix}, \qquad \boldsymbol{P}_\kappa(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \boldsymbol{I}_{\mathsf{il}_\kappa} & \boldsymbol{0}_{\mathsf{il}_\kappa \times \mathsf{ql}_\kappa} \\ \boldsymbol{\Gamma}_\kappa & \boldsymbol{\Delta}_\kappa \end{pmatrix}.$$

The implicit verification polynomial becomes

$$V(\boldsymbol{X}, \mathbb{Q}) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Q}_1 \end{pmatrix}^T \boldsymbol{N}(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \end{pmatrix},$$

where

$$\boldsymbol{N}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \left(\begin{array}{c|c} \boldsymbol{M}_{11} + \boldsymbol{M}_{12}\,\boldsymbol{\Gamma}_2 + \boldsymbol{\Gamma}_1^T\,\boldsymbol{M}_{21} + \boldsymbol{\Gamma}_1^T\,\boldsymbol{M}_{22}\,\boldsymbol{\Gamma}_2 & \left(\boldsymbol{M}_{12} + \boldsymbol{\Gamma}_1^T \boldsymbol{M}_{22}\right)\boldsymbol{\Delta}_2 \\ \hline \boldsymbol{\Delta}_1^T\left(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\,\boldsymbol{\Gamma}_2\right) & \boldsymbol{\Delta}_1^T\,\boldsymbol{M}_{22}\,\boldsymbol{\Delta}_2 \end{array} \right).$$

The $\mathbb{Q}$-dependent part reduces to

$$V_{\mathbb{x},\boldsymbol{M},\boldsymbol{\Gamma},\boldsymbol{\Delta}}^t(\boldsymbol{X}, \mathbb{Q}) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Q}_1 \end{pmatrix}^T \boldsymbol{f}_{\boldsymbol{M}}^t(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \end{pmatrix},$$

with

$$\boldsymbol{f}_{\boldsymbol{M}}^t(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{il}_2} & (\boldsymbol{M}_{12} + \boldsymbol{\Gamma}_1^T \boldsymbol{M}_{22})\boldsymbol{\Delta}_2 \\ \boldsymbol{\Delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\Gamma}_2) & \boldsymbol{\Delta}_1^T \boldsymbol{M}_{22}\boldsymbol{\Delta}_2 \end{pmatrix}.$$

3.3 Lifting Theorem

We start with an overview of the general AGMOS proof strategy from [LPS23]. We focus on the class of security games described in Section 3.1, although their proof strategy is more general. The AGMOS adversary's output is accepted if the implicit verification polynomial $V(\boldsymbol{X}, \mathbb{Q})$ satisfies $V(\boldsymbol{x}, \mathbb{q}) = 0$, where $\boldsymbol{x}$ are the trapdoors sampled by the challenger and $[\mathbb{q}_\kappa]_\kappa$ the AGMOS oracle answers. For assumptions that involve an additional structural check, the adversary must in addition violate this condition in order to win. One then analyzes four cases:

Case A: $V(\boldsymbol{X}, \mathbb{Q}) = 0$**.** Typically, this case implies that the adversary failed. Thus, it does not require treatment in the proof of security.
Case X.1: $V(\boldsymbol{X}, \mathbb{Q}) \neq 0$**, but** $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$**.** One constructs a reduction to FPR.
Case X.2: $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$**, but** $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$**.** One constructs a different reduction to FPR.
Case Q: $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$**.** One constructs a reduction to TOFR.

In the AGM, the verification polynomial does not include variables $\mathbb{Q}$, so we have $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ and only cases A and X.1 must be dealt with. In our lifting theorem, these two cases are covered by AGM security rather than by a separate reduction to FPR (see below). In the presence of a structural check, it is additionally required that any adversary considered in these cases also violates the structural check.

We prove the lifting theorem separately for security games with and without a structural check. For security games without a structural check, we turn any AGMOS adversary $\mathcal{A}$ into an AGM adversary, an FPR adversary, and a TOFR adversary. In cases A and X.1, removing the contributions of obliviously sampled elements preserves the pairing check represented by Equation (2), so the AGM adversary succeeds whenever $\mathcal{A}$ succeeds in these cases. In case X.2, we obtain a successful FPR adversary, and in case Q, a successful TOFR adversary. Thus, $\mathcal{A}$'s success probability is bounded by the sum of these three adversaries' advantages. Security in the AGM, together with the FPR and TOFR assumptions, therefore implies security in the AGMOS.

For assumptions that involve an additional structural check, it is not sufficient to preserve only the pairing check. In this case, we must also ensure that the structural violation is preserved when reducing to the AGM. To achieve this, we impose a lifting condition which guarantees that, whenever the $\mathbb{Q}$-dependent part vanishes, the adversary's explanation satisfies $\boldsymbol{\delta} = \boldsymbol{0}$. Under this condition, in the cases A and X.1, any AGMOS adversary that succeeds in breaking the assumption behaves identically to an AGM adversary. In particular, both the pairing check and the violation of the structural check are preserved, so the resulting AGM adversary also outputs a transcript that violates the structural check. Thus, the probability that an AGMOS adversary wins in cases A and X.1 is bounded by the success probability of the constructed AGM adversary, to which the AGM security bound applies.

Motivating our lifting condition. First, consider cases A and X.1, where $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$. A lifting theorem cannot hold for arbitrary security games, because in some games an AGMOS adversary passes the verification with $\boldsymbol{\delta} \neq \boldsymbol{0}$; Lipmaa et al. use this to show that, under the DLOG assumption, some assumptions that are secure in the AGM are insecure in the AGMOS (see [LPS23, Section 6] and Section 3.4). For games with a structural check, we therefore require the *lifting condition* that $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ implies $\boldsymbol{\delta} = \boldsymbol{0}$, and this condition has to be verified for each assumption that we lift. By Equations (5) and (6), the equation $V^t_{\mathbb{x},\mathsf{aux},\boldsymbol{M},\boldsymbol{\Gamma},\boldsymbol{\Delta}}(\boldsymbol{X}, \mathbb{Q}) = 0$ is a system of polynomial equations in $\boldsymbol{M}$, $\boldsymbol{\Gamma}$, and $\boldsymbol{\Delta}$ that are linear or quadratic in $\boldsymbol{\Delta}$, so it suffices to show that every solution of this system has $\boldsymbol{\Delta} = \boldsymbol{0}$ (see Section 4 for examples). Section 3.4 discusses to what extent the condition is tight and why some restriction of this form is necessary.

Handling the AGMOS-specific remaining cases. Second, consider cases X.2 and Q, which are specific to the AGMOS.

- In case X.2, consider $V^t(\boldsymbol{X}, \mathbb{Q})$ as a polynomial in the indeterminates $\mathbb{Q}$ over $\mathbb{F}[\boldsymbol{X}]$. Since $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$, one of its coefficients—let us denote it by $g(\boldsymbol{X})$—is non-zero. Since $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$, we have $g(\boldsymbol{x}) = 0$. Being able to construct such a polynomial allows us to break the FPR assumption.
- In case Q, the coefficients of $V^h(\boldsymbol{x}) + V^t(\boldsymbol{x}, \mathbb{Q}) \in \mathbb{F}[\mathbb{Q}]$ correspond to the components of $\boldsymbol{v}$ from Definition 5, allowing us to break the TOFR assumption. We deal with these two cases in Theorem 1.

Lifting theorem for security games. We now state and prove our lifting theorem for non-interactive security games. We begin by showing that for games defined in Section 3.1 without any structural check, any accepting AGMOS transcript with $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ remains accepting even after setting $\boldsymbol{\delta} = \mathbf{0}$. In other words, whenever an AGMOS adversary produces a valid transcript in this case, there exists an equivalent accepting transcript that does not depend on the obliviously sampled group elements.

Lemma 1 (Cancellation lemma). *Let $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ be the security game from Section 3.1, with $\mathcal{S} \equiv 1$. Let tr be an accepting AGMOS transcript. If its implicit verification polynomial satisfies*

$$V^t(\boldsymbol{X}, \mathbb{Q}) \equiv 0,$$

then $\pi_{\mathrm{AGM}}(\mathsf{tr})$ (as defined in Section 3.1) is an accepting AGM transcript for the same public input.

Proof. By definition of the projection, the implicit verification polynomial for $\pi_{\mathrm{AGM}}(\mathsf{tr})$ is

$$V_{\pi_{\mathrm{AGM}}(\mathsf{tr})}(\boldsymbol{X}) = V^h(\boldsymbol{X}) + V^t(\boldsymbol{X}, \mathbf{0}) = V^h(\boldsymbol{X}),$$

since setting the oblivious sampling variables to zero removes all terms in V^t. Since tr is accepting and $V^t(\boldsymbol{X}, \mathbb{Q}) \equiv 0$, we have

$$0 = V(\boldsymbol{x}, \mathbb{q}) = V^h(\boldsymbol{x}) = V_{\pi_{\mathrm{AGM}}(\mathsf{tr})}(\boldsymbol{x}).$$

Thus, $\pi_{\mathrm{AGM}}(\mathsf{tr})$ satisfies the pairing check.

The projection preserves the public input and provides valid AGM explanations for the projected outputs. Since $\mathcal{S} \equiv 1$, there is no further condition to check. Hence $\pi_{\mathrm{AGM}}(\mathsf{tr})$ is an accepting AGM transcript. □

Intuitively, the lifting condition ensures that whenever an AGMOS adversary wins in the case $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$, its behavior is already captured by an AGM adversary, i.e., it does not rely on the oblivious sampling component.

Definition 8 (Lifting condition). *Let $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of security games as defined in Section 3.1, with*

$$\Gamma_\lambda = (\mathsf{PGen}, \mathsf{KPGen}, \mathsf{MGen}, \mathcal{C}, \boldsymbol{d}, \mathcal{S}).$$

We say that this family satisfies the lifting condition *if the following holds for every security parameter λ, every gp in the support of $\mathsf{PGen}(1^\lambda)$, and every*

$$(\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X}))$$

in the support of $\mathsf{KPGen}(1^\lambda, \mathsf{gp})$: for all polynomially bounded query lengths $\mathsf{ql}_1, \mathsf{ql}_2$, all explanation matrices

$$\boldsymbol{\gamma}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa \times \mathsf{il}_\kappa}, \qquad \boldsymbol{\delta}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa \times \mathsf{ql}_\kappa} \qquad (\kappa \in \{1, 2\}),$$

all $\boldsymbol{\alpha} \in \mathbb{F}^r$, and every $\boldsymbol{M}$ in the support of $\mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha})$, we have

$$\begin{aligned} \textit{if} \quad & V^t_{\mathbb{x},\mathsf{aux},\boldsymbol{M},\boldsymbol{\gamma},\boldsymbol{\delta}}(\boldsymbol{X}, \mathbb{Q}) = 0, \\ \textit{then} \quad & \boldsymbol{\delta} = \mathbf{0}. \end{aligned}$$

Here, $\boldsymbol{X} = (X_1, \ldots, X_m)$ and $\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$ are vectors of indeterminates, with $\mathbb{Q}_\kappa = (\mathbb{Q}_{\kappa,1}, \ldots, \mathbb{Q}_{\kappa,\mathsf{ql}_\kappa})$. The identity is a formal polynomial identity in these variables.

The lifting theorem has a unified formulation that covers games with and without a structural check, and it requires the lifting condition only in the former case.

Theorem 1 (Lifting theorem). *Let Γ be an AGMOS security game as in Section 3.1 with structural check $\mathcal{S}$, and let Γ^{AGM} be its AGM variant. Suppose that*

1. *the structural checks of Γ and Γ^{AGM} are trivial, or*

2. Γ satisfies the lifting condition (Definition 8) and $\mathcal{S}$ is compatible with Γ^{AGM}.

Then for every PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$ there are PPT adversaries $\mathcal{B}$, $\mathcal{B}_{\mathsf{fpr}}$, and $\mathcal{B}_{\mathsf{tofr}}$ such that

$$\mathsf{Adv}^{\Gamma}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\Gamma^{\mathrm{AGM}}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda),$$

where $\boldsymbol{d}$ are the degree bounds of Γ and m is its number of trapdoors.

Proof. Fix a PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$ for Γ_λ.

Let Win denote the event that $\mathcal{A}$ wins the game Γ_λ, i.e., Win is the event that the pairing check Equation (2) holds, together with the game's conditions on non-group data (if any) and with $\mathcal{S}(\mathsf{tr}) = 1$ in the non-trivial-structural-check case. Let Bad denote the event that $\boldsymbol{\delta} \neq \boldsymbol{0}$ in games with a non-trivial structural check, and the event that $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$ in games with a trivial structural check.

In both cases, we decompose the winning probability as

$$\Pr[\mathsf{Win}] = \Pr[\mathsf{Win} \wedge \neg\mathsf{Bad}] + \Pr[\mathsf{Win} \wedge \mathsf{Bad}].$$

We first bound the term $\Pr[\mathsf{Win} \wedge \mathsf{Bad}]$.

In the case of games with a non-trivial structural check, the lifting condition (Definition 8) implies that

$$V^t_{\mathbb{x},\mathsf{aux},\boldsymbol{M},\boldsymbol{\gamma},\boldsymbol{\delta}}(\boldsymbol{X}, \mathbb{Q}) = 0 \;\Rightarrow\; \boldsymbol{\delta} = \boldsymbol{0}.$$

Taking the contrapositive, $\boldsymbol{\delta} \neq \boldsymbol{0}$ implies $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$. Since Bad is the event that $\boldsymbol{\delta} \neq \boldsymbol{0}$, this yields

$$\Pr[\mathsf{Win} \wedge \mathsf{Bad}] \leq \Pr[\mathsf{Win} \wedge (V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0)].$$

In the case of a trivial structural check, this bound holds by definition of Bad. Hence, in both cases, the inequality above holds.

Dropping the additional requirements, namely the conditions on non-group data and, in the non-trivial case, $\mathcal{S}(\mathsf{tr}) = 1$, can only increase the probability. Thus, in both cases,

$$\Pr[\mathsf{Win} \wedge \mathsf{Bad}] \leq \Pr[\text{AGMOS-specific}],$$

where

$$\Pr[\text{AGMOS-specific}] = \Pr\left[\begin{array}{l|l} \mathbb{y}_1 \in \mathbb{F}^{\mathsf{ol}_1} \;\wedge\; \mathbb{y}_2 \in \mathbb{F}^{\mathsf{ol}_2} \;\wedge\; \boldsymbol{\alpha} \in \mathbb{F}^r \;\wedge & \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda); \quad \boldsymbol{x} \leftarrow \mathbb{F}^m; \\ \begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \mathbb{y}_1 \\ \mathsf{aux}_1(\boldsymbol{x}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{x}) \\ \mathbb{y}_2 \\ \mathsf{aux}_2(\boldsymbol{x}) \end{pmatrix} = 0 \wedge & \big(\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X})\big) \\ & \quad \leftarrow \mathsf{KPGen}(1^\lambda, \mathsf{gp}); \\ V^t(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0 & \big(([\mathbb{y}_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, [\mathbb{q}_\kappa]_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha}\big) \\ & \quad \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{OS}}}\big(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2\big); \\ & \boldsymbol{M} \leftarrow \mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}) \end{array}\right]. \tag{7}$$

By Lemma 2,

$$\Pr[\text{AGMOS-specific}] \leq \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Thus,

$$\Pr[\mathsf{Win} \wedge \mathsf{Bad}] \leq \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Next, we bound the term $\Pr[\mathsf{Win} \wedge \neg\mathsf{Bad}]$.

We define an AGM adversary $\mathcal{B}$ as follows. On input $(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2)$, $\mathcal{B}$ runs $\mathcal{A}$ and answers its oracle queries by locally simulating the $(\mathcal{EF}, \mathcal{DF})$-AGMOS oracle $\mathcal{O}_{\mathsf{OS}}$ from Figure 1. This simulation is efficient because the functions in $\mathcal{EF}$ are efficiently computable and the distributions in $\mathcal{DF}$ are efficiently sampleable. The running time of $\mathcal{B}$ is that of $\mathcal{A}$, plus the time needed to simulate the oblivious sampling

oracle, reconstruct the projected outputs, and maintain the transcript and algebraic explanations. The FPR and TOFR reductions also require time to perform the necessary polynomial computations.

When $\mathcal{A}$ terminates, $\mathcal{B}$ proceeds as follows. If the game has a trivial structural check, then $\mathcal{B}$ replaces $\mathcal{A}$'s output by the public-input-only transcript from Lemma 1 and forwards its AGM projection to the AGM challenger. Since $\neg\mathsf{Bad}$ means precisely that $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$, Lemma 1 implies that this projected transcript remains accepting. Hence, whenever $\mathsf{Win} \wedge \neg\mathsf{Bad}$ occurs, $\mathcal{B}$ wins the AGM variant of the game.

If the game has a non-trivial structural check, then $\mathcal{B}$ checks whether $\boldsymbol{\delta} = \mathbf{0}$. If this holds, $\mathcal{B}$ outputs the projected group elements together with the explanation matrices $\boldsymbol{\gamma}_\kappa$ and the field elements $\boldsymbol{\alpha}$; otherwise, $\mathcal{B}$ outputs $\perp$. Since $\boldsymbol{\delta} = \mathbf{0}$, the projected group elements coincide with those output by $\mathcal{A}$. Whenever $\mathsf{Win} \wedge \neg\mathsf{Bad}$ occurs, we have $\boldsymbol{\delta} = \mathbf{0}$, so $\mathcal{A}$'s transcript already uses only the challenger-provided input basis. Since $\mathcal{A}$ wins on the same public instance and the same public input, AGM compatibility gives $\mathcal{S}_{\mathrm{AGM}}(\pi_{\mathrm{AGM}}(\mathsf{tr})) = \mathcal{S}(\mathsf{tr}) = 1$, and hence the projected transcript is winning also in the AGM variant of the game.

The conditions on non-group data hold for $\mathcal{B}$ whenever they hold for $\mathcal{A}$, because $\mathcal{B}$ outputs the same field elements $\boldsymbol{\alpha}$. Therefore, in both cases,

$$\Pr[\mathsf{Win} \wedge \neg\mathsf{Bad}] \leq \Pr\big[\mathcal{B} \text{ wins the AGM variant of } \Gamma_\lambda\big].$$

Combining the above bounds, we obtain

$$\Pr\big[\mathcal{A} \text{ wins } \Gamma_\lambda\big] \leq \Pr\big[\mathcal{B} \text{ wins the AGM variant of } \Gamma_\lambda\big] + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda),$$

which concludes the proof. □

Lemma 2 (Security for the AGMOS-specific cases). *Let Γ be an AGMOS security game as in Section 3.1. For every PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$ there are PPT adversaries $\mathcal{B}_{\mathsf{fpr}}$ and $\mathcal{B}_{\mathsf{tofr}}$ such that the probability in Equation (7) satisfies*

$$\Pr[\textit{AGMOS-specific}] \leq \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Proof. Assume that $\mathcal{A}$ outputs a transcript satisfying the pairing check and $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$. Using Equations (4) and (5), we derive

$$\begin{aligned} V^t(\boldsymbol{X}, \mathbb{Q}) &= \mathbb{x}_1^T(\boldsymbol{X})(\boldsymbol{M}_{12}\boldsymbol{\delta}_2 + \boldsymbol{\gamma}_1^T\boldsymbol{M}_{22}\boldsymbol{\delta}_2)\mathbb{Q}_2 + \mathbb{Q}_1^T\boldsymbol{\delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\gamma}_2)\mathbb{x}_2(\boldsymbol{X}) \\ &\quad + \mathbb{Q}_1^T\boldsymbol{\delta}_1^T\boldsymbol{M}_{22}\boldsymbol{\delta}_2\mathbb{Q}_2 + \mathbb{Q}_1^T\boldsymbol{\delta}_1^T\boldsymbol{M}_{23}\mathsf{aux}_2(\boldsymbol{X}) + \mathsf{aux}_1^T(\boldsymbol{X})\boldsymbol{M}_{32}\boldsymbol{\delta}_2\mathbb{Q}_2 \\ &= \begin{pmatrix} \boldsymbol{\delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\gamma}_2)\mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_1^T\boldsymbol{M}_{23}\mathsf{aux}_2(\boldsymbol{X}) \\ \boldsymbol{\delta}_2^T(\boldsymbol{M}_{12}^T + \boldsymbol{M}_{22}^T\boldsymbol{\gamma}_1)\mathbb{x}_1(\boldsymbol{X}) + \boldsymbol{\delta}_2^T\boldsymbol{M}_{32}^T\mathsf{aux}_1(\boldsymbol{X}) \\ \mathrm{vec}(\boldsymbol{\delta}_2^T\boldsymbol{M}_{22}^T\boldsymbol{\delta}_1) \end{pmatrix}^T \begin{pmatrix} \mathbb{Q}_1 \\ \mathbb{Q}_2 \\ \mathbb{Q}_1 \otimes \mathbb{Q}_2 \end{pmatrix}, \end{aligned} \tag{8}$$

where the operator vec denotes vectorization.

Case X.2: $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$, but $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$. In Figure 3, we define a PPT adversary $\mathcal{B}_{\mathsf{fpr}}$ for the $(\boldsymbol{d}, m)$-FPR game. The assumption (1) ensures that the FPR oracle returns the desired answers. The third block in Equation (8), namely $\mathrm{vec}(\boldsymbol{\delta}_2^T\boldsymbol{M}_{22}^T\boldsymbol{\delta}_1)$, is constant in $\boldsymbol{X}$. Since $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$ as a polynomial in $\mathbb{Q}$, this block already vanishes in case X.2. Therefore, because $V^t(\boldsymbol{X}, \mathbb{Q}) \neq 0$, one of the first two blocks contains a non-zero component $g(\boldsymbol{X})$ as described above. Since $V^t(\boldsymbol{x}, \mathbb{Q}) = 0$, we have $g(\boldsymbol{x}) = 0$. By the degree bounds from the assumption (1), $\deg(g(\boldsymbol{X})) \leq \max(md_1, md_2) \leq d_g$. Therefore, $\mathcal{B}_{\mathsf{fpr}}$ is successful.

Case Q: $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$. In Figure 4, we define a PPT adversary $\mathcal{B}_{\mathsf{tofr}}$ for the $(\mathcal{EF}, \mathcal{DF})$-TOFR game. By Equation (8), $V^t(\boldsymbol{x}, \mathbb{Q}) \neq 0$ implies $\boldsymbol{v} \neq \mathbf{0}$ and $V(\boldsymbol{x}, \mathbb{q}) = V^h(\boldsymbol{x}) + V^t(\boldsymbol{x}, \mathbb{q}) = 0$ implies

$$\boldsymbol{v}^T \begin{pmatrix} 1 \\ \mathbb{q}_1 \\ \mathbb{q}_2 \\ \mathbb{q}_1 \otimes \mathbb{q}_2 \end{pmatrix} = 0.$$

Therefore, $\mathcal{B}_{\mathsf{tofr}}$ succeeds. □

$\mathcal{B}_{\mathsf{fpr}}(\mathsf{gp})$:
Run $\mathsf{KPGen}(1^\lambda, \mathsf{gp}) \to (\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X}))$.
Query $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m,\boldsymbol{x}}(\cdot)$ to get $[\mathbb{x}_1(\boldsymbol{x})]_1$ and $[\mathbb{x}_2(\boldsymbol{x})]_2$.
Run $\mathcal{A}^{\mathcal{O}_{\mathsf{OS}}}(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2) \to ((\llbracket \mathbb{y}_\kappa \rrbracket_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, \llbracket \mathbb{q}_\kappa \rrbracket_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha})$.
Run $\mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}) \to \boldsymbol{M}$.
Find a non-zero component $g(\boldsymbol{X}) \in \mathbb{F}[\boldsymbol{X}]$ of

$$\begin{pmatrix} \boldsymbol{\delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\gamma}_2)\mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_1^T\boldsymbol{M}_{23}\mathsf{aux}_2(\boldsymbol{X}) \\ \boldsymbol{\delta}_2^T(\boldsymbol{M}_{12}^T + \boldsymbol{M}_{22}^T\boldsymbol{\gamma}_1)\mathbb{x}_1(\boldsymbol{X}) + \boldsymbol{\delta}_2^T\boldsymbol{M}_{32}^T\mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}$$

and return it.

Fig. 3: The adversary $\mathcal{B}_{\mathsf{fpr}}$ for the $(\boldsymbol{d}, m)$-FPR game in Lemma 2.

$\mathcal{B}_{\mathsf{tofr}}(\mathsf{gp})$:
Run $\mathsf{KPGen}(1^\lambda, \mathsf{gp}) \to (\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X}))$.
Sample $\boldsymbol{x} \leftarrow \mathbb{F}^m$.
Run $\mathcal{A}^{\mathcal{O}_{\mathsf{OS}}}(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2) \to ((\llbracket \mathbb{y}_\kappa \rrbracket_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa, \llbracket \mathbb{q}_\kappa \rrbracket_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha})$.
Run $\mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}) \to \boldsymbol{M}$.
Compute $V^h(\boldsymbol{X})$ from $\mathbb{x}_\kappa(\boldsymbol{X})$, $\mathsf{aux}_\kappa(\boldsymbol{X})$, $\boldsymbol{\gamma}$ and $\boldsymbol{M}$ (see Equation (3)).
Return

$$\boldsymbol{v} = \begin{pmatrix} V^h(\boldsymbol{x}) \\ \boldsymbol{\delta}_1^T(\boldsymbol{M}_{21} + \boldsymbol{M}_{22}\boldsymbol{\gamma}_2)\mathbb{x}_2(\boldsymbol{x}) + \boldsymbol{\delta}_1^T\boldsymbol{M}_{23}\mathsf{aux}_2(\boldsymbol{x}) \\ \boldsymbol{\delta}_2^T(\boldsymbol{M}_{12}^T + \boldsymbol{M}_{22}^T\boldsymbol{\gamma}_1)\mathbb{x}_1(\boldsymbol{x}) + \boldsymbol{\delta}_2^T\boldsymbol{M}_{32}^T\mathsf{aux}_1(\boldsymbol{x}) \\ \mathrm{vec}(\boldsymbol{\delta}_2^T\boldsymbol{M}_{22}^T\boldsymbol{\delta}_1) \end{pmatrix}.$$

Fig. 4: The adversary $\mathcal{B}_{\mathsf{tofr}}$ for the $(\mathcal{EF}, \mathcal{DF})$-TOFR game in Lemma 2.

Remark 6. If we set $\mathsf{aux}_1(\boldsymbol{X})$ and $\mathsf{aux}_2(\boldsymbol{X})$ to empty strings, our lifting theorem (Theorem 1) *covers publicly verifiable assumptions*. Specifically, the $\mathsf{aux}_\kappa(\boldsymbol{X})$ terms simply vanish from the reduction, yet the argument remains identical.

Remark 7 (Allowing negative powers). Some applications, such as Polymath, use negative powers of the trapdoors in their public parameters. To cover them, we allow KPGen to output Laurent polynomials, that is, elements of $\mathbb{F}[\boldsymbol{X}, \boldsymbol{X}^{-1}]$ such as $X^{-2}+X+1$. In the entries of $\mathbb{x}_\kappa$ and aux_κ, every exponent of every variable must then lie in $[-d^-, d_\kappa]$ for a fixed bound $d^- \geq 0$, which replaces the degree bound in Equation (1). The challenger samples the trapdoors from $(\mathbb{F}^*)^m$ instead of $\mathbb{F}^m$, so that the public inputs are defined, and the case distinction and the lifting condition are read as identities of Laurent polynomials.

Only case X.2 in the proof of Lemma 2 needs a change, because there $\mathcal{B}_{\mathsf{fpr}}$ obtains the public inputs from the FPR oracle, which accepts only polynomials. We therefore use the *modified* $(\boldsymbol{d}, d^-, m)$-FPR *assumption*, which differs from Definition 6 in two points. The trapdoors are sampled from $(\mathbb{F}^*)^m$, and on a query (κ, f), the oracle returns $[f(\boldsymbol{x})]_\kappa$ for every Laurent polynomial f whose exponents lie in $[-d^-, d_\kappa]$. The adversary must still output a non-zero polynomial $g \in \mathbb{F}^{(\leq d_g)}[\boldsymbol{X}]$ with $g(\boldsymbol{x}) = 0$, so for $d^- = 0$ this is the $(\boldsymbol{d}, m)$-FPR assumption with non-zero trapdoors. It is the FPR counterpart of the variant of the PDL assumption with negative exponents that Lipmaa [Lip24] uses for Polymath. Lipmaa states that this variant follows from the standard PDL assumption by reprogramming the group generators. We do not provide the details of the corresponding adaptation to the modified FPR assumption.

In case X.2, the component g found by $\mathcal{B}_{\mathsf{fpr}}$ is a non-zero Laurent polynomial with $g(\boldsymbol{x}) = 0$, and $\mathcal{B}_{\mathsf{fpr}}$ outputs $(X_1 \cdots X_m)^{d^-} g(\boldsymbol{X})$ instead. This is a non-zero polynomial of total degree at most $m(d^- + \max(d_1, d_2))$, and it vanishes at $\boldsymbol{x}$ because the trapdoors are non-zero. Hence, if the requirement $d_g \geq \max(md_1, md_2)$ is replaced by $d_g \geq m(d^- + \max(d_1, d_2))$, Theorem 1 holds with the advantage against the

modified $(\boldsymbol{d}, d^-, m)$-FPR assumption in place of $\mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr}}}(\lambda)$. The rest of the proof is unchanged, except that $\mathcal{B}_{\mathsf{tofr}}$ also samples the trapdoors from $(\mathbb{F}^*)^m$, and Section 4.3 applies this extension to Polymath.

3.4 Limitations of Our Lifting Theorem

The lifting condition on $V^t(\boldsymbol{X}, \mathbb{Q})$ from Theorem 1 does not always hold. A trivial example is when $\boldsymbol{M} = \boldsymbol{0}$, i.e., the pairing check is identically satisfied, and the lifting condition fails trivially. More generally, if $\boldsymbol{f}^t_{\boldsymbol{M}}(\boldsymbol{\gamma}, \boldsymbol{\delta}) = \boldsymbol{0}$, then $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ (see Equation (4) and Equation (5)).

Total Knowledge of Exponent. Lipmaa et al. analyzed the following family of Total Knowledge of Exponent (TotalKE) assumptions, which are trivially secure in the AGM. For consistency with Lipmaa et al., we state TotalKE using the extractor-based variant of the AGM(OS), as this is the formulation in which their separation result is naturally stated.

Definition 9 (TotalKE). *Let* $\mathsf{il}_1 = \mathsf{il}_2 = 1, \mathsf{ol}_1, \mathsf{ol}_2 \geq 1$ *and* $R \geq 1$. *Let* $\boldsymbol{M}[i] \in \mathbb{F}^{(\mathsf{il}_1+\mathsf{ol}_1)\times(\mathsf{il}_2+\mathsf{ol}_2)}$ *for* $i \in [1, R]$. *Let*

$$V_i(\boldsymbol{y}, \boldsymbol{z}) = \begin{pmatrix} 1 \\ \boldsymbol{y} \end{pmatrix}^T \boldsymbol{M}[i] \begin{pmatrix} 1 \\ \boldsymbol{z} \end{pmatrix}.$$

The $(\mathsf{ol}_1, \mathsf{ol}_2, \{V_i\}_{i=1}^R)$*-TotalKE assumption holds for* $\mathsf{PGen}(1^\lambda)$ *if for every non-uniform PPT adversary* $\mathcal{A}$*, there exists a non-uniform PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$*, such that*

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

where by $\mathsf{RND}_\lambda(\mathcal{A})$*, we denote the random tape of* $\mathcal{A}$ *(for security parameter* λ*), and by* $r \leftarrow \mathsf{RND}_\lambda(\mathcal{A})$ *the uniform choice of* r *from* $\mathsf{RND}_\lambda(\mathcal{A})$.

They hypothesized that *"[. . .] the TotalKE assumption is secure [in the AGMOS under the* DLOG *assumption] iff for any* $\boldsymbol{\Gamma} = \boldsymbol{\gamma}$, $\boldsymbol{f}^t_{\boldsymbol{M}}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \boldsymbol{0}$ *has only a zero solution in* $\boldsymbol{\Delta}$*."* Generalizing and proving a similar statement for our class of security games is an interesting line for future research.

Tightness and necessity. The lifting condition should be interpreted as a sufficient condition for a black-box AGM-to-AGMOS lift, rather than as a complete characterization of all games that lift. Nevertheless, some restriction of this kind is unavoidable: known separations between the AGM and the AGMOS show that arbitrary AGM-secure games need not remain secure once the adversary can use obliviously sampled group elements. The role of our condition is to rule out precisely the problematic case in which the $\mathbb{Q}$-dependent part of the verification polynomial vanishes while the adversary's explanation still contains a non-zero oblivious component $\boldsymbol{\delta}$. In games with a non-trivial structural check, such a component may prevent the successful AGMOS transcript from projecting to a successful AGM transcript. Thus, while we do not claim that the condition is necessary for every possible game, it captures the obstruction used by known separations and is essentially the condition one needs for our black-box projection argument.

Decisional variants. Our results apply to search games whose verification is expressed by polynomial equations. Decisional games instead ask the adversary to distinguish between two experiments. Our lifting argument does not directly apply to these games. We leave the lifting of decisional security games to future work.

Trapdoor distribution. Our games sample the trapdoors uniformly from $\mathbb{F}^m$ and derive the public inputs as polynomials in them. The reductions use both facts: the FPR game samples its trapdoors uniformly, and its oracle returns encodings of polynomials in them. Games with other trapdoor distributions are not covered by our theorems, and neither are games whose public inputs are not polynomial in the trapdoors, except for the extensions to Laurent polynomials (Remark 7) and to the rational signing responses of BBS (Section 6.2).

4 Applications of the Non-Interactive Lifting Theorem

We illustrate Theorem 1 on three assumptions: the extractability of the KZG polynomial commitment scheme, the $\mathsf{ARSDH}(n)$ assumption, and the special-soundness of Polymath. We work out each application in full detail, so that the reader can follow and check every step. Most of the length comes from notation and explicit calculations, while the underlying ideas are simple. In each case, it suffices to cast the assumption as a security game, to check that its structural check is compatible with the AGM variant, and to verify the lifting condition (for Polymath, with a small extension, see Remark 7). This is considerably simpler than proving security in the AGMOS from scratch.

4.1 Extractability of the KZG Polynomial Commitment Scheme

To illustrate their AGMOS proof technique, Lipmaa et al. showed [LPS23, Section 5.3] that if the FPR and TOFR assumptions hold (for certain parameters), the univariate KZG polynomial commitment scheme (PCS) [KZG10] is extractable in the AGMOS. To demonstrate our framework, we apply the notation from Section 3.1 to the extractability assumption for the univariate KZG PCS.

A PCS allows a prover to commit to a polynomial and later prove its evaluation at a given point. It consists of algorithms for key generation, commitment, opening, and verification. The key generation algorithm KGen produces a commitment key ck, which the prover uses to commit to a polynomial $f(X) \in \mathbb{F}[X]$ by computing

$$C = \mathsf{Com}(\mathsf{ck}, f(X)).$$

Given an evaluation point $\alpha \in \mathbb{F}$, the prover computes the value $\eta = f(\alpha)$ and an opening proof π. The verifier checks $\mathsf{Ver}(\mathsf{ck}, C, \alpha, \eta, \pi)$ to determine whether to accept the claim that the committed polynomial evaluates to η at α.

We now recall how the KZG PCS works. The commitment key is given by $\mathsf{ck} = \big(([x^i]_1)_{i=0}^d,\ [1]_2,\ [x]_2\big)$, which allows the prover to commit to univariate polynomials of degree at most d. To commit to a polynomial $f(X) = \sum_{i=0}^d f_i X^i$, the prover computes $C = \sum_{i=0}^d f_i [x^i]_1 = [f(x)]_1$, and sends the commitment C to the verifier. The verifier then selects a challenge point α and sends it to the prover. The prover responds by computing the claimed evaluation $\eta = f(\alpha)$ and the corresponding proof $\pi = [q(x)]_1$, where $q(X) = \frac{f(X)-\eta}{X-\alpha}$. If the prover is honest, the evaluation claim holds and they can compute the quotient polynomial $q(X)$, and, thus, provide a valid proof π. To verify the proof, the verifier checks the following pairing equation: $(C - [\eta]_1) \bullet [1]_2 = \pi \bullet [x - \alpha]_2$. If the equation holds, the verifier accepts the proof; otherwise, they reject.

The extractability game. Extractability for a PCS is a knowledge assumption stating that any prover who convinces the verifier must "know" a corresponding polynomial. This notion of knowledge is captured by the existence of an efficient algorithm, called the extractor, which "extracts" a valid witness from the prover, in this case a polynomial. The PCS is extractable if, for every prover, there exists an extractor that outputs a polynomial that commits to the claimed commitment and equals η at α.

In our presentation, we formulate the extractability experiment as a game between a challenger and an adversary, rather than a prover and a verifier, in order to align with our security game framework. This is only a notational change: the adversary plays the role of the prover, while the challenger runs the verification procedure and the generation of the commitment key. The only difference from the standard formulation is that the commitment key is generated by the challenger rather than some trusted third party; this distinction is irrelevant for our purposes.

To lift KZG extractability from the AGM to the AGMOS using Theorem 1, we establish two properties:

1. the extractability game fits the framework of Section 3.1, with a structural check compatible with the AGM variant;
2. the lifting condition holds (we need this because extractability is a game with a non-trivial structural check).

The structural check captures extraction failure. Thus, after the reduction to the AGM, the opening must still be accepted and the extractor must still fail.

The extractability game. The challenger generates group parameters gp, samples $x \leftarrow \mathbb{F}$, and constructs the commitment key

$$\mathsf{ck} = \big(([x^i]_1)_{i=0}^d, [1]_2, [x]_2\big).$$

In the notation of our framework, KPGen produces the formal polynomials $(1, X, \ldots, X^d)$ and $(1, X)$; evaluating them at x and encoding the results gives the public group elements used by KZG.

Given gp and ck, the adversary $\mathcal{A}$ makes oblivious sampling queries and outputs

$$\big((C, \pi, \boldsymbol{\gamma}_1, \boldsymbol{\delta}_1, [\mathfrak{q}_1]_1), \alpha, \eta\big).$$

Here, C is the commitment, π is the opening proof, and η is the claimed evaluation at α. The matrices $\boldsymbol{\gamma}_1$ and $\boldsymbol{\delta}_1$ explain C and π in terms of the commitment key elements and the obliviously sampled elements, respectively. The transcript tr contains the public inputs, the oblivious sampling queries and responses, and this final output.

The challenger checks the KZG verification equation

$$(C - [\eta]_1) \bullet [1]_2 = \pi \bullet [x - \alpha]_2.$$

An extractor Ext, given tr, attempts to recover a polynomial $f(X)$ satisfying

$$\deg(f) \le d, \qquad \mathsf{Com}(\mathsf{ck}, f) = C, \qquad f(\alpha) = \eta.$$

For this extractor, define $\mathcal{S}(\mathsf{tr}) = 1$ if extraction fails, including when the extractor returns $\perp$ or an invalid polynomial, and set it to 0 otherwise. The adversary wins exactly when the pairing check holds and $\mathcal{S}(\mathsf{tr}) = 1$. It loses if verification rejects or the extractor recovers a polynomial satisfying all three conditions.

Definition 10 (Extractability of a PCS in the AGMOS). *A PCS $\Pi = (\mathsf{KGen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Ver})$ is* extractable for PGen *if, for all $d \in \mathsf{poly}(\lambda)$ and every PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, such that*

$$\Pr\left[\begin{array}{l|l} \mathsf{Ver}(\mathsf{ck}, C, \alpha, \eta, \pi) = 1 \wedge & \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda); \mathsf{ck} \leftarrow \mathsf{KGen}(\mathsf{gp}, d); \\ (C \neq \mathsf{Com}(\mathsf{ck}, f(X)) \vee & ((C, \pi, \boldsymbol{\gamma}_1, \boldsymbol{\delta}_1, [\mathfrak{q}_1]_1), \alpha, \eta) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{OS}}}(\mathsf{ck}); \\ \ \deg f(X) > d \ \vee \ f(\alpha) \neq \eta) & \mathsf{tr} = (\mathsf{ck}, ((C, \pi, \boldsymbol{\gamma}_1, \boldsymbol{\delta}_1, [\mathfrak{q}_1]_1), \alpha, \eta)); \\ & f(X) \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{tr}) \end{array}\right] \in \mathsf{negl}(\lambda).$$

In other words, accepted openings for which extraction fails must occur only with negligible probability. To break extractability, an adversary must cause this failure with non-negligible probability for every permitted PPT extractor. Note that, even though the extractor is provided the algebraic explanation of the commitment, the above definition is not vacuous in the AGM. In particular, one needs to prove that the polynomial output by the extractor matches the evaluation claim of the adversary. Some works in the literature also consider a stronger notion of *knowledge-soundness of a PCS in the AGM(OS)* [GWC19, BDFG20], where the extractor must construct the polynomial f *only* from the commitment and its algebraic explanation, but the rest of the security experiment is unchanged.

Compatibility with the AGM variant. Fix an AGMOS adversary $\mathcal{A}$ and construct the corresponding AGM adversary $\mathcal{B}_{\mathcal{A}}$ using the projection to $[\mathbb{x}(\boldsymbol{x})]_1$ and $[\mathbb{x}(\boldsymbol{x})]_2$ from the lifting theorem. Specifically, let $\gamma_{C,i}$ and $\gamma_{\pi,i}$ denote the commitment key coefficients of C and π, respectively. Their projected group elements to $[\mathbb{x}(\boldsymbol{x})]_1$ and $[\mathbb{x}(\boldsymbol{x})]_2$ are

$$C' = \sum_{i=0}^{d} \gamma_{C,i} [x^i]_1, \qquad \pi' = \sum_{i=0}^{d} \gamma_{\pi,i} [x^i]_1.$$

The adversary $\mathcal{B}_{\mathcal{A}}$ outputs

$$\big((C', \pi', \boldsymbol{\gamma}_1), \alpha, \eta\big).$$

By AGM extractability, there exists an extractor $\mathsf{Ext}^{\mathrm{AGM}}_{\mathcal{B}_{\mathcal{A}}}$ for $\mathcal{B}_{\mathcal{A}}$. We use it to define the AGMOS extractor:

$$\mathsf{Ext}_{\mathcal{A}}(\mathsf{tr}) := \mathsf{Ext}^{\mathrm{AGM}}_{\mathcal{B}_{\mathcal{A}}}(\pi_{\mathrm{AGM}}(\mathsf{tr})),$$

where

$$\mathsf{tr} = (\mathsf{ck}, ((C, \pi, \boldsymbol{\gamma}_1, \boldsymbol{\delta}_1, [\mathbb{q}_1]_1), \alpha, \eta)) \quad \text{and} \quad \pi_{\mathrm{AGM}}(\mathsf{tr}) = (\mathsf{ck}, ((C', \pi', \boldsymbol{\gamma}_1), \alpha, \eta)).$$

Thus, the AGMOS extractor runs the AGM extractor on the projected output. We check the returned polynomial against the original commitment C and claim (α, η).

When $\boldsymbol{\delta}_1 = \mathbf{0}$, projection does not change the commitment or the proof: $C' = C$ and $\pi' = \pi$. Both extractors return the same polynomial, and we check it against the same commitment and evaluation claim. Therefore, extraction succeeds or fails in both games together. Here, $\mathcal{S}$ is the extraction-failure predicate for $\mathsf{Ext}_{\mathcal{A}}$, and $\mathcal{S}_{\mathrm{AGM}}$ is the extraction-failure predicate for the extractor $\mathsf{Ext}^{\mathrm{AGM}}_{\mathcal{B}_{\mathcal{A}}}$ supplied by AGM extractability. This proves compatibility with the AGM variant:

$$\mathcal{S}(\mathsf{tr}) = \mathcal{S}_{\mathrm{AGM}}\big(\pi_{\mathrm{AGM}}(\mathsf{tr})\big) \quad \text{when} \quad \boldsymbol{\delta}_1 = \mathbf{0}.$$

It remains to prove the lifting condition

$$V^t \equiv 0 \quad \Longrightarrow \quad \boldsymbol{\delta}_1 = \mathbf{0}.$$

In cases A and $\mathsf{X.1}$, we have $V^t \equiv 0$. The lifting condition then ensures that an accepted opening with failed extraction in the AGMOS gives an accepted opening with failed extraction in the AGM. We can therefore bound these cases using AGM extractability. The remaining cases are bounded by the FPR and TOFR advantages. The following theorem formalizes the argument above for lifting KZG extractability and knowledge-soundness from the AGM to the AGMOS. The proof follows by a straightforward verification of the lifting condition and a direct application of Theorem 1.

Theorem 2. *Let $d \in \mathsf{poly}(\lambda)$ and $\boldsymbol{d} = (d, 1, d+1, d+1)$, and assume the $(\boldsymbol{d}, 1)$-FPR and $(\mathcal{EF}, \mathcal{DF})$-$\mathsf{TOFR}$ assumptions. If the KZG PCS for univariate polynomials of degree at most d is extractable in the AGM, then it is extractable in the $(\mathcal{EF}, \mathcal{DF})$-AGMOS. The same holds for knowledge-soundness.*

Proof. Extractability of a PCS constitutes an assumption with a non-trivial structural check. As explained above, the extractor defining the structural check ignores the AGMOS-only fields, and hence the compatibility hypothesis of Theorem 1 is satisfied. Therefore, in order to apply the lifting theorem, it remains to verify the lifting condition. The extractor's allowed inputs do not change the pairing check, so the same lifting condition proof applies to both extractability and knowledge-soundness.

We now make explicit the parameters of the corresponding security game.

For an AGMOS adversary, we have $m = 1$ ($\boldsymbol{x} = x_1 = x$), $\mathsf{il}_1 = d + 1$, $\mathbb{x}_1(X) = (X^i)_{i=0}^d$, $\mathsf{il}_2 = 2$, $\mathbb{x}_2(X) = (1, X)$, $\mathsf{ol}_1 = 2$, $[\mathbb{y}_1]_1 = [\mathbb{y}_{1,1}, \mathbb{y}_{1,2}]_1 = (C, \pi)$, $\mathsf{ol}_2 = 0$, $r = 2$, $\boldsymbol{\alpha} = (\alpha, \eta)$. The explicit verification polynomial is obtained by rewriting the KZG pairing check $(C - [\eta]_1) \bullet [1]_2 = \pi \bullet [x - \alpha]_2$ in terms of discrete logarithms and rearranging all terms to one side. The resulting verification polynomial has the following form:

$$V^{\mathsf{expl}}(X, \mathbb{Y}_{1,1}, \mathbb{Y}_{1,2}) = \mathbb{Y}_{1,1} - \eta - \mathbb{Y}_{1,2}(X - \alpha).$$

The adversary $\mathcal{A}$ outputs a vector $[\mathbb{y}_1]_1 = (C, \pi) \in \mathbb{G}_1^2$ that admits a decomposition of the form $(C, \pi) = \boldsymbol{\gamma}_1[\mathbb{x}_1(\boldsymbol{x})]_1 + \boldsymbol{\delta}_1[\mathbb{q}_1]_1$, where $[\mathbb{q}_1]_1 = \big([\mathbb{q}_{1,1}]_1, \ldots, [\mathbb{q}_{1,\mathsf{ql}_1}]_1\big)$, and $\boldsymbol{\gamma}_1$ and $\boldsymbol{\delta}_1$ are matrices whose rows consist of the adversary's explanation vectors corresponding to the input elements $[\mathbb{x}_1(\boldsymbol{x})]_1$ and the obliviously sampled elements $[\mathbb{q}_1]_1$, respectively.

We now construct the implicit verification polynomial

$$V(X, \mathbb{Q}) = \sum_{i=0}^{d} \gamma_{1,i+1} X^i + \sum_{i=1}^{\mathsf{ql}_1} \delta_{1,i} \mathbb{Q}_{1,i} - \eta - \left(\sum_{i=0}^{d} \gamma_{2,i+1} X^i + \sum_{i=1}^{\mathsf{ql}_1} \delta_{2,i} \mathbb{Q}_{1,i} \right) (X - \alpha).$$

By isolating the terms involving $\mathbb{Q}$ variables, we split the polynomial $V(X, \mathbb{Q})$ into two parts: one containing no $\mathbb{Q}$ variables, and the other consisting solely of terms involving $\mathbb{Q}$ variables. The polynomial $V^t(X, \mathbb{Q})$ has the following form:

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

Now, it is straightforward to verify the lifting condition from Definition 8, i.e., $V^t(X, \mathbb{Q}) = 0 \Rightarrow \boldsymbol{\delta} = \mathbf{0}$. Considering all the monomials $X\mathbb{Q}_{1,i}$ and $\mathbb{Q}_{1,i}$ appearing in $V^t(X, \mathbb{Q})$, we construct a system of linear equations

$$\begin{aligned} \delta_{1,i} + \alpha\delta_{2,i} &= 0 \qquad \forall 1 \le i \le \mathsf{ql}_1, \\ \delta_{2,i} &= 0 \qquad \forall 1 \le i \le \mathsf{ql}_1, \end{aligned}$$

which clearly admits only the trivial solution $\boldsymbol{\delta} = \mathbf{0}$. Therefore, we can invoke Theorem 1 to conclude that, under the assumed variants of FPR and TOFR, any AGM proof of extractability for the univariate KZG PCS implies extractability in the AGMOS. □

Remark 8. By Remark 5, the explicit verification polynomial admits a representation of the form

$$V^{\mathsf{expl}}(\boldsymbol{X}, \mathbb{Y}_1, \mathbb{Y}_2) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Y}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Y}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix},$$

as described in Section 3.2.

For the extractability game of the KZG PCS, the explicit verification polynomial can be written as

$$V^{\mathsf{expl}}(X, \mathbb{Y}_1, \mathbb{Y}_2) = \begin{pmatrix} 1 & X & \dots & X^d & \mathbb{Y}_{1,1} & \mathbb{Y}_{1,2} \end{pmatrix} \underbrace{\begin{pmatrix} -\eta & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ \alpha & -1 \end{pmatrix}}_{\boldsymbol{M}} \begin{pmatrix} 1 \\ X \end{pmatrix}.$$

Lifting for other pairing-based PCS. The same method can be used for other pairing-based PCS. If the verification polynomial is linear in the adversary's outputs, setting $V^t \equiv 0$ gives a system of linear equations in $\boldsymbol{\delta}$. To prove the lifting condition, we show that its only solution is $\boldsymbol{\delta} = \mathbf{0}$ for every allowed choice of transcript parameters.

This gives a concrete starting point for studying Mercury [EG25], Samaritan [GPS25], multivariate KZG [PST13], HyperPianist [LZL+25], and Pianist [LXZ+24]. For each scheme, one must establish the required linear independence and verify the remaining hypotheses of our theorem. Once these checks are completed, the existing AGM security proof can be lifted to the AGMOS under the corresponding FPR and TOFR assumptions.

4.2 ARSDH(n) Assumption

In this subsection, we illustrate our lifting theorem on a privately verifiable assumption with a non-trivial structural check. The Adaptive Rational Strong Diffie–Hellman assumption, ARSDH, was recently used in proofs of extractability of variants of the KZG PCS in the standard model [LPS24, BDH+25]. The univariate ARSDH assumption was proposed by Lipmaa et al. [LPS24], who established its security in the AGMOS. Subsequently, Belohorec et al. [BDH+25] introduced its multivariate variant ARSDH(n), and proved its security in the AGM. Siim and Pal [SP26] recently showed that ARSDH(n) is equivalent to the univariate ARSDH

assumption. Their reductions only take linear combinations of group elements with known coefficients, so they also apply to AGMOS adversaries, and together with the result of Lipmaa et al. they give another proof of the AGMOS security of $\mathsf{ARSDH}(n)$. In this subsection, we use the results from Section 3.3 to lift the AGM security proof of the $\mathsf{ARSDH}(n)$ assumption [BDH+25] to the AGMOS.

We start by defining the assumption.

Definition 11 ($\mathsf{ARSDH}(n)$ Game). *For a PPT adversary $\mathcal{A}$, the security parameter λ, a bilinear group generator PGen, and $\mathbf{d}' = (d'_1, \ldots, d'_n) \in \mathsf{poly}(\lambda)$, the $\mathsf{ARSDH}(n)$ Game is defined as follows:*

1. *Sample $\mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda)$ and $\boldsymbol{x} = (x_1, \ldots, x_n) \leftarrow \mathbb{F}^n$.*
2. *On input $1^\lambda, \mathsf{gp}$, and $\Big(\big([x_1^{i_1} \cdots x_n^{i_n}]_1\big)_{i_1,\ldots,i_n=0}^{d'_1,\ldots,d'_n}, ([x_i]_2)_{i\in[1,n]}\Big)$, $\mathcal{A}$ outputs*

$$\{[\mathbb{y}_i]_1\}_{i=0}^{n}, \quad \{\alpha_i\}_{i=1}^{n-1} \subseteq \mathbb{F}, \quad \text{and} \quad S \subseteq \mathbb{F}.$$

$\mathcal{A}$ wins if and only if:

1. *$|S| = d'_n + 1$,*
2. *$[\mathbb{y}_0]_1 \bullet [1]_2 \neq \bigodot_{i=1}^{n-1} [\mathbb{y}_i]_1 \bullet [x_i - \alpha_i]_2$, and*
3. *$[\mathbb{y}_0]_1 \bullet [1]_2 = \Big(\bigodot_{i=1}^{n-1} [\mathbb{y}_i]_1 \bullet [x_i - \alpha_i]_2\Big) \odot [\mathbb{y}_n]_1 \bullet [Z_S(x_n)]_2$.*

Here, $Z_S(X) = \prod_{s\in S}(X - s)$ is the vanishing polynomial of the set S.

Definition 12 ($\mathsf{ARSDH}(n)$). *Let PGen be a bilinear group generator. Then the n-*variate Adaptive Rational Strong Diffie–Hellman ($\mathsf{ARSDH}(n)$) assumption *holds for PGen if, for any $\mathbf{d}' = (d'_1, \ldots, d'_n) \in \mathsf{poly}(\lambda)$ and PPT $\mathcal{A}$,*

$$\Pr[\mathcal{A} \text{ wins } \mathsf{ARSDH}(n)] \in \mathsf{negl}(\lambda).$$

The $\mathsf{ARSDH}(n)$ game is a privately verifiable assumption with a non-trivial structural check. In particular, the structural check is captured by the second verification requirement, where the challenger checks that the adversary's output does not satisfy a certain trivial algebraic relation. The first requirement, $|S| = d'_n + 1$, is a condition on non-group data, and the third requirement is the pairing check.

Formally, let

$$\mathsf{tr} = \Big(\mathsf{gp}, ([x_1^{i_1} \cdots x_n^{i_n}]_1)_{0\le i_j \le d'_j}, ([x_i]_2)_{i=1}^{n}, ([x_n^j]_2)_{j=2}^{d'_n+1},$$
$$([\mathfrak{q}]_\kappa)_{\kappa\in\{1,2\}}, ([\mathbb{y}_i]_1)_{i=0}^{n}, \boldsymbol{\gamma}_1, \boldsymbol{\delta}_1, (\alpha_i)_{i=1}^{n-1}, S\Big).$$

We define the structural predicate $\mathcal{S}(\mathsf{tr})$ by setting $\mathcal{S}(\mathsf{tr}) = 1$ exactly when

$$[\mathbb{y}_0]_1 \bullet [1]_2 \neq \bigodot_{i=1}^{n-1} [\mathbb{y}_i]_1 \bullet [x_i - \alpha_i]_2.$$

We set $\mathcal{S}(\mathsf{tr}) = 0$ otherwise. This check is essential: without this check, the adversary could win by setting all output group elements to the identity.

Next, we lift the security of the $\mathsf{ARSDH}(n)$ assumption from AGM to AGMOS using Theorem 1. The structural check depends only on the public inputs and the adversary's group and field outputs. When $\boldsymbol{\delta} = \mathbf{0}$, the AGM projection preserves these values and hence the result of the structural check. Thus, the structural check for $\mathsf{ARSDH}(n)$ is compatible with the AGM variant, as required by Theorem 1.

Theorem 3 (Lifting $\mathsf{ARSDH}(n)$ to AGMOS). *Let $d'_{\max} := \max_i\{d'_i\}$ and*

$$\boldsymbol{d} = \big(d'_{\max},\ d'_n + 1,\ d'_{\max} + d'_n + 1,\ n\max\{d'_{\max}, d'_n + 1\}\big),$$

and assume the $(\boldsymbol{d}, n)$-FPR and $(\mathcal{EF}, \mathcal{DF})$-$\mathsf{TOFR}$ assumptions. If the $\mathsf{ARSDH}(n)$ assumption holds in the AGM, then it holds in the $(\mathcal{EF}, \mathcal{DF})$-AGMOS.

Proof. Since $\mathsf{ARSDH}(n)$ is an assumption with a non-trivial structural check, applying Theorem 1 requires verifying that the lifting condition (Definition 8) holds. As the assumption is already known to be secure in the AGM [BDH+25], Theorem 1 then shows that its security carries over to the AGMOS.

In the notation of Theorem 1, we have $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n-1}, S)$, and, hence $r = d'_n + n$, $\mathsf{il}_1 = (d'_1 + 1) \cdots (d'_n + 1)$, $\mathsf{il}_2 = n+1$, $\mathsf{ol}_1 = n+1$, $\mathsf{ol}_2 = 0$. In this case, $\mathsf{aux}_1(\boldsymbol{X})$ is an empty string and $\mathsf{aux}_2(\boldsymbol{X}) = (X_n^2, \ldots, X_n^{d'_n+1}) \in \mathbb{F}[\boldsymbol{X}]^{d'_n}$. The choice $d_2 = d'_n + 1$ is needed because these auxiliary $\mathbb{G}_2$-elements have degree up to $d'_n + 1$, and the chosen value of d_g satisfies the generic requirement $d_g \geq \max(nd_1, nd_2)$.

The explicit verification polynomial is of the following form:

$$V^{\mathsf{expl}}(X_1, \ldots, X_n, \mathbb{Y}_{1,0}, \ldots, \mathbb{Y}_{1,n}) = \mathbb{Y}_{1,0} - \sum_{i=1}^{n-1} \mathbb{Y}_{1,i} \cdot (X_i - \alpha_i) - \mathbb{Y}_{1,n} \cdot Z_S(X_n),$$

where $Z_S(X_n) = \sum_{j=0}^{d'_n+1} z_j X_n^j$ is the vanishing polynomial of S, and the $\mathbb{Y}_{1,i}$, for $i \in [0, n]$, are variables that correspond to discrete logarithms of group elements output by the adversary $\mathcal{A}$. The adversary $\mathcal{A}$ produces a vector $[\mathbb{y}_1]_1 \in \mathbb{G}_1^{n+1}$ which admits a decomposition of the form $\boldsymbol{\gamma}_1[\mathbb{x}_1(\boldsymbol{x})]_1 + \boldsymbol{\delta}_1[\mathbb{q}_1]_1$, where $[\mathbb{q}_1]_1 = [\mathbb{q}_{1,1}, \ldots, \mathbb{q}_{1,\mathsf{ql}_1}]_1^T$ and $\boldsymbol{\gamma}_1, \boldsymbol{\delta}_1$ are matrices formed by the explanation vectors.

We now construct the implicit verification polynomial

$$V(\boldsymbol{X}, \mathbb{Q}) = \mathbb{Y}_{1,0}(\boldsymbol{X}, \mathbb{Q}) - \sum_{i=1}^{n-1} \mathbb{Y}_{1,i}(\boldsymbol{X}, \mathbb{Q}) \cdot (X_i - \alpha_i) - \mathbb{Y}_{1,n}(\boldsymbol{X}, \mathbb{Q}) \cdot Z_S(X_n),$$

where $\mathbb{Y}_{1,i}$ is the variable representing the discrete logarithm of the i-th coordinate of the output vector $[\mathbb{y}_1]_1$ and it can be written as a linear combination of $\mathbb{x}_1(\boldsymbol{X})$ and $\mathbb{Q}$, in the following way:

$$\mathbb{Y}_{1,i}(\boldsymbol{X}, \mathbb{Q}) = \sum_{j=1}^{\mathsf{il}_1} \gamma_{i,j} \mathbb{x}_{1,j}(\boldsymbol{X}) + \sum_{j=1}^{\mathsf{ql}_1} \delta_{i,j} \mathbb{Q}_{1,j}, \ i \in [0, n],$$

where $\mathbb{x}_{1,j}(\boldsymbol{X})$ is the j-th coordinate of the vector $\mathbb{x}_1(\boldsymbol{X})$.

When we rewrite the implicit verification polynomial in terms of $\mathbb{x}_1(\boldsymbol{X})$ and $\mathbb{Q}_1$, we get:

$$\begin{aligned} V(\boldsymbol{X}, \mathbb{Q}) = & \sum_{j=1}^{\mathsf{il}_1} \gamma_{0,j} \mathbb{x}_{1,j}(\boldsymbol{X}) + \sum_{j=1}^{\mathsf{ql}_1} \delta_{0,j} \mathbb{Q}_{1,j} \\ & - \sum_{i=1}^{n-1} \left(\sum_{j=1}^{\mathsf{il}_1} \gamma_{i,j} \mathbb{x}_{1,j}(\boldsymbol{X}) + \sum_{j=1}^{\mathsf{ql}_1} \delta_{i,j} \mathbb{Q}_{1,j} \right) (X_i - \alpha_i) \\ & - \left(\sum_{j=1}^{\mathsf{il}_1} \gamma_{n,j} \mathbb{x}_{1,j}(\boldsymbol{X}) + \sum_{j=1}^{\mathsf{ql}_1} \delta_{n,j} \mathbb{Q}_{1,j} \right) Z_S(X_n). \end{aligned}$$

By isolating the terms involving $\mathbb{Q}$ variables, we split the polynomial $V(\boldsymbol{X}, \mathbb{Q})$ into two parts: one containing no $\mathbb{Q}$ variables, and the other consisting solely of terms involving $\mathbb{Q}$ variables. The polynomial $V^t(\boldsymbol{X}, \mathbb{Q})$ has the following form

$$\begin{aligned} V^t(\boldsymbol{X}, \mathbb{Q}) &= \sum_{j=1}^{\mathsf{ql}_1} \delta_{0,j} \mathbb{Q}_{1,j} - \sum_{i=1}^{n-1} (X_i - \alpha_i) \left(\sum_{j=1}^{\mathsf{ql}_1} \delta_{i,j} \mathbb{Q}_{1,j} \right) - \left(\sum_{j=1}^{\mathsf{ql}_1} \delta_{n,j} \mathbb{Q}_{1,j} \right) Z_S(X_n) \\ &= \sum_{j=1}^{\mathsf{ql}_1} \mathbb{Q}_{1,j} \left(\delta_{0,j} - \sum_{i=1}^{n-1} (X_i - \alpha_i) \delta_{i,j} - \delta_{n,j} Z_S(X_n) \right). \end{aligned} \tag{9}$$

Now, it is straightforward to verify the lifting condition. The coefficients of $\mathbb{Q}_{1,j}$ in Equation (9) are $\delta_{0,j} - \sum_{i=1}^{n-1}(X_i - \alpha_i)\delta_{i,j} - \delta_{n,j}Z_S(X_n)$. Thus, $V^t(\boldsymbol{X}, \mathbb{Q}) = 0$ is equivalent to the following system of ql_1 linear equations in the variables $\delta_{i,j}$, where $i \in [0, n]$ and $j \in [1, \mathsf{ql}_1]$:

$$\delta_{0,j} - \sum_{i=1}^{n-1}(X_i - \alpha_i)\delta_{i,j} - \delta_{n,j}Z_S(X_n) = 0, \quad \forall 1 \le j \le \mathsf{ql}_1.$$

Since $|S| = d'_n + 1$, the polynomial $Z_S(X_n)$ is monic of degree $d'_n + 1$. Hence, for each j, the coefficient of $X_n^{d'_n+1}$ in the displayed equation is $-\delta_{n,j}$, so $\delta_{n,j} = 0$. After substituting this, the coefficients of X_i for $i \in [1, n-1]$ give $\delta_{i,j} = 0$, and the remaining constant term gives $\delta_{0,j} = 0$. Thus, $\delta_{i,j} = 0$ for all $i \in [0, n]$ and $j \in [1, \mathsf{ql}_1]$. Therefore, the lifting condition holds and we can invoke Theorem 1 to conclude that, under the specific FPR and TOFR assumptions, AGM security of $\mathsf{ARSDH}(n)$ lifts to the AGMOS. □

4.3 Special-Soundness of Polymath

We now consider security games in which the verification procedure consists of multiple pairing checks that must be satisfied simultaneously, as in special-soundness games for SNARKs. Multiple pairing checks do not introduce fundamentally new challenges compared to a single pairing check, but they require an extension of the analysis. As an application, we adapt our lifting theorem to show that the special-soundness of the interactive Polymath argument introduced in [Lip24] lifts from the AGM to the AGMOS.

Intuitively, special-soundness states that a witness can be efficiently extracted from sufficiently many accepting transcripts that share the same first prover message and have distinct verifier challenges. For protocols with multiple rounds, these transcripts form a tree, where branching corresponds to different challenges. We first define this tree of transcripts and then $\boldsymbol{k}$-out-of-$\boldsymbol{N}$ special-soundness for multi-round public-coin interactive arguments.

Definition 13 ($(k_1, \ldots, k_\mu)$-tree of transcripts). *Given $\mathbf{k} = (k_1, \ldots, k_\mu) \in \mathbb{N}^\mu$, a $\boldsymbol{k}$-tree of transcripts for a $(2\mu+1)$-move public-coin interactive argument $\Pi = (\mathsf{PGen}, \mathsf{KPGen}, \mathcal{P}, \mathcal{V})$ consists of $K = \prod_{i=1}^{\mu} k_i$ transcripts organized in a tree. In this tree, nodes represent the prover's messages and edges represent the verifier's challenges. A node at depth i has k_i children, each for distinct challenges. Transcripts map uniquely to paths from the root to a leaf.*

Definition 14 ($\boldsymbol{k}$-out-of-$\boldsymbol{N}$ special-soundness). *Let $\boldsymbol{k}, \boldsymbol{N} \in \mathbb{N}^\mu$. A $(2\mu+1)$-move public-coin interactive argument $\Pi = (\mathsf{PGen}, \mathsf{KPGen}, \mathcal{P}, \mathcal{V})$ for relation generator $\mathcal{RG}$, where $\mathcal{V}$ samples the i-th challenge from a set of cardinality $N_i \ge k_i$ for $1 \le i \le \mu$, is $\boldsymbol{k}$-out-of-$\boldsymbol{N}$ special-sound if there exists a PPT extractor Ext_{ss} such that for any λ, $n \in \mathsf{poly}(\lambda)$, $\mathcal{R} \in \mathcal{RG}(1^\lambda)$, PPT $\mathcal{A}_{ss}$,*

$$\Pr\left[\begin{array}{c|l} \text{T is a $\boldsymbol{k}$-tree of accepting} & \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda, n); \\ \text{transcripts } \wedge ((\mathsf{ck}, x), w) \notin \mathcal{R} & (\mathsf{ck}, td_{\mathsf{ck}}) \leftarrow \mathsf{KPGen}(\mathsf{gp}, \mathcal{R}); \\ & (x, T) \leftarrow \mathcal{A}_{ss}(\mathsf{ck}); \\ & w \leftarrow \mathsf{Ext}_{ss}(\mathsf{ck}, x, T) \end{array}\right] \approx_\lambda 0.$$

To prove special-soundness, we consider accepting transcripts arranged in a $(D+1, 2)$-tree, as defined in Definition 14. Here, $D = 2(d_{\max} - d_{\min})$ is the degree bound of [Lip24, Theorem 3], where $[d_{\min}, d_{\max}]$ is the range of exponents of X in the SRS. All transcripts share the first prover message. This message is followed by $D+1$ distinct first verifier challenges. For each of these challenges, the transcripts share the next prover message and then branch into two distinct second verifier challenges, each followed by an accepting prover response. Each path in this tree corresponds to one accepting transcript and gives rise to a corresponding verification polynomial, which we index by $\langle ij \rangle$, for $i \in [1, D+1]$ and $j \in [1, 2]$.

Specifically, the special-soundness adversary outputs a $(D+1, 2)$-tree of accepting transcripts, where each transcript indexed by $\langle ij \rangle$ has the form

$$\mathrm{tr}^{\langle ij \rangle} = \left([a, c]_1;\ x_1^{\langle i \rangle}, A_{x_1}^{\langle i \rangle}, x_2^{\langle ij \rangle}, [d^{\langle ij \rangle}]_1, \boldsymbol{\gamma}^{\langle ij \rangle}, \boldsymbol{\delta}^{\langle ij \rangle}\right).$$

Here, $[a, c]_1$ denotes the common first prover message, the values $x_1^{\langle i \rangle}$ and $x_2^{\langle ij \rangle}$ are verifier challenges, $A_{x_1}^{\langle i \rangle}$ is the prover response to the first challenge, $[d^{\langle ij \rangle}]_1$ is the final prover message, and $\boldsymbol{\gamma}^{\langle ij \rangle}$ and $\boldsymbol{\delta}^{\langle ij \rangle}$ are the explanation matrices corresponding to group elements in the transcript $\mathrm{tr}^{\langle ij \rangle}$. In the remainder of this subsection, we use the terminology of adversary and challenger in place of prover and verifier, respectively.

For each accepting transcript $\mathrm{tr}^{\langle ij \rangle}$, the challenger's acceptance condition can be expressed as a verification polynomial obtained by rewriting the pairing check in algebraic form. Substituting the AGMOS algebraic explanations into the pairing check of Polymath, and rearranging the resulting identity in the discrete logarithm form, yields the polynomial

$$V^{\langle ij \rangle}(\bar{\boldsymbol{X}}^{\langle ij \rangle}) := \left(\left(A(\bar{\boldsymbol{X}}) + x_2^{\langle ij \rangle} C(\bar{\boldsymbol{X}}) - \left(A_{x_1}^{\langle i \rangle} + x_2^{\langle ij \rangle} C_{x_1}^{\langle i \rangle} \right) \right) Z - D^{\langle ij \rangle}(\bar{\boldsymbol{X}}^{\langle ij \rangle}) \left(X - x_1^{\langle i \rangle} \right) \right) X^T, \tag{10}$$

which must evaluate to zero for every accepting transcript. Here, $\bar{\boldsymbol{X}}^{\langle ij \rangle}$ denotes the vector of indeterminates consisting of the trapdoor variables X and Z and the variables $\mathbb{Q}^{\langle ij \rangle}$ of the obliviously sampled elements used in $\mathrm{tr}^{\langle ij \rangle}$, and $C_{x_1}^{\langle i \rangle} = C(x_1^{\langle i \rangle})$ is the evaluation of C at the first challenge, which the verification algorithm of Polymath computes from $A_{x_1}^{\langle i \rangle}$ and the public input.

The $\mathbb{Q}$-dependent $V_t^{\langle ij \rangle}\left(\bar{\boldsymbol{X}}^{\langle ij \rangle}\right)$ polynomial has the form

$$V_t^{\langle ij \rangle}\left(\bar{\boldsymbol{X}}^{\langle ij \rangle}\right) = X^T \sum_k \beta_k^{\langle ij \rangle}(X, Z) \mathbb{Q}_k^{\langle ij \rangle}, \tag{11}$$

where the coefficients are given by

$$\beta_k^{\langle ij \rangle}(X, Z) := \left(a_{\mathbb{q}_k} + x_2^{\langle ij \rangle} c_{\mathbb{q}_k} \right) Z - d_{\mathbb{q}_k}^{\langle ij \rangle} \left(X - x_1^{\langle i \rangle} \right).$$

Here, $a_{\mathbb{q}_k}$, $c_{\mathbb{q}_k}$, and $d_{\mathbb{q}_k}^{\langle ij \rangle}$ are coefficients arising from the explanation matrix $\boldsymbol{\delta}^{\langle ij \rangle}$ corresponding to the obliviously sampled group elements, while $x_1^{\langle i \rangle}$ and $x_2^{\langle ij \rangle}$ denote the challenges appearing in the transcript.

Structural check. The structural check requires a well-formed $(D + 1, 2)$-tree for which extraction fails. Acceptance of its transcripts is enforced separately by the pairing checks. As we show below, in cases A and X.1, the lifting condition gives $\boldsymbol{\delta} = \mathbf{0}$. The AGM reduction therefore outputs the same statement and transcript tree as the AGMOS adversary. By special-soundness in the AGM, there exists an extractor that extracts a witness from such accepting trees, except with negligible probability. We use this extractor in the AGMOS as well. Since its input is unchanged, its extraction success is unchanged. Thus, the probability of extraction failure in these two cases is bounded by AGM special-soundness.

Separating cases when there are multiple verification polynomials. In Section 3, we introduced an AGMOS security game whose acceptance condition consists of a single pairing check and, where required, a structural check. However, in the context of special-soundness, we must consider multiple accepting transcripts simultaneously. Since each accepting transcript gives rise to its own pairing check, this naturally leads to a collection of pairing checks and hence a collection of verification polynomials $V_i(\boldsymbol{X}, \mathbb{Q})$, one for each transcript.

We now consider a setting with s pairing checks that must hold simultaneously. As in the single-polynomial setting, we distinguish the following cases:

1. **Case A.** For all $i \in [1, s]$, we have
$$V_i(\boldsymbol{X}, \mathbb{Q}) \equiv 0.$$
(This case does not occur.)
2. **Case X.1.** There exists $i \in [1, s]$ such that
$$V_i(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0 \quad \text{and} \quad V_i^t(\boldsymbol{X}, \mathbb{Q}) \equiv 0.$$
This means that there exists at least one index $i \in [1, s]$ such that $V_i(\boldsymbol{X}, \mathbb{Q})$ falls into case X.1, while each of the remaining polynomials falls into case A or X.1. This case is bounded by AGM security, as in the single-polynomial setting (see below).

3. **Case X.2.** There exists $i \in [1, s]$ such that

$$V_i^t(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0 \quad \text{and} \quad V_i^t(\boldsymbol{x}, \mathbb{Q}) \equiv 0.$$

   This means that there exists at least one index $i \in [1, s]$ such that $V_i(\boldsymbol{X}, \mathbb{Q})$ falls into case X.2, while each of the remaining polynomials falls into case A, X.1, or X.2. This case reduces to the FPR assumption.
4. **Case Q.** There exists $i \in [1, s]$ such that

$$V_i^t(\boldsymbol{x}, \mathbb{Q}) \not\equiv 0.$$

   This means that there exists at least one index $i \in [1, s]$ such that $V_i(\boldsymbol{X}, \mathbb{Q})$ falls into case Q, while the remaining polynomials may fall into any case. This case reduces to the TOFR assumption.

The lifting theorem still applies in this setting, provided that the lifting condition holds for the entire collection of verification polynomials. In particular, we require that

$$\begin{aligned} \text{if} \quad & V_i^t(\boldsymbol{X}, \mathbb{Q}) = 0 \quad \text{for all } i, \\ \text{then} \quad & \boldsymbol{\delta} = \mathbf{0}. \end{aligned}$$

This condition ensures that if all V^t parts vanish, then the adversary cannot make use of any obliviously sampled elements, and we get exactly the AGM variant of the security game. Apart from the fact that the reduction must now identify the relevant verification polynomial V_i, the lifting argument proceeds identically to the general case from Section 3.

Lifting condition for multiple pairing checks. Since the Polymath verifier performs multiple pairing checks, the lifting condition must hold simultaneously for all verification polynomials arising from the pairing checks. In particular, we require that

$$\begin{aligned} \text{if} \quad & V_t^{\langle ij \rangle}(\bar{\boldsymbol{X}}^{\langle ij \rangle}) = 0 \quad \text{for all } i, j, \\ \text{then} \quad & \boldsymbol{\delta} = \mathbf{0}. \end{aligned}$$

Suppose now that $V_t^{\langle ij \rangle}\left(\bar{\boldsymbol{X}}^{\langle ij \rangle}\right) = 0$ for all i, j. By the definition of $V_t^{\langle ij \rangle}\left(\bar{\boldsymbol{X}}^{\langle ij \rangle}\right)$, this implies that all its coefficients must vanish. In particular, we obtain that

$$\beta_k^{\langle ij \rangle}(X, Z) = 0 \quad \text{for all } k, i, j.$$

By definition of the coefficients $\beta_k^{\langle ij \rangle}(X, Z)$, this implies

$$a_{\mathbb{q}_k} + x_2^{\langle ij \rangle} c_{\mathbb{q}_k} = 0, \qquad d_{\mathbb{q}_k}^{\langle ij \rangle} = 0 \quad \text{for all } k, i, j.$$

Since the adversary outputs a $(D+1, 2)$-tree, for every first challenge $x_1^{\langle i \rangle}$ there exist two distinct second-round challenges $x_2^{\langle i1 \rangle}$ and $x_2^{\langle i2 \rangle}$. Consequently, for each k we obtain the system of equations

$$a_{\mathbb{q}_k} + x_2^{\langle i1 \rangle} c_{\mathbb{q}_k} = 0, \qquad a_{\mathbb{q}_k} + x_2^{\langle i2 \rangle} c_{\mathbb{q}_k} = 0.$$

Since $x_2^{\langle i1 \rangle} \neq x_2^{\langle i2 \rangle}$, it follows that

$$a_{\mathbb{q}_k} = 0 \quad \text{and} \quad c_{\mathbb{q}_k} = 0 \quad \text{for all } k.$$

Together with the fact that

$$d_{\mathbb{q}_k}^{\langle ij \rangle} = 0 \quad \text{for all } k, i, j,$$

we conclude that $\boldsymbol{\delta} = \mathbf{0}$.

Security proof in the AGMOS. In cases A and X.1, the lifting condition gives $\boldsymbol{\delta} = \mathbf{0}$, as shown above, so the AGM reduction can output the same output as the AGMOS adversary, and these cases are bounded by the security in the AGM. In case X.2, we use the modified FPR assumption described in Remark 7. The reduction uses its oracle to simulate the SRS, including the negative powers of the trapdoors, and finds a non-zero coefficient of $V^t(\boldsymbol{X}, \mathbb{Q})$ that vanishes at the trapdoors. If this coefficient contains negative powers, the reduction multiplies it by a monomial to obtain a non-zero polynomial that still vanishes at the trapdoors, and with the bounds of the theorem below, this is a solution in the modified FPR game. In case Q, the TOFR reduction follows the same argument as in our lifting proof; it samples the trapdoors itself and can therefore compute the SRS, including its negative powers.

Theorem 4 (Lifting special-soundness of Polymath to AGMOS). *Let $D = 2(d_{\max} - d_{\min})$ be as in [Lip24, Theorem 3], and assume the modified $(\boldsymbol{d}, -d_{\min}, 2)$-FPR assumption of Remark 7 with $d_1 = d_2 = d_{\max}$ and $d_g = 2(d_{\max} - d_{\min})$, and the $(\mathcal{EF}, \mathcal{DF})$-TOFR assumption. If the interactive Polymath argument with the setup and parameters of [Lip24] is $(D+1, 2)$-special-sound in the AGM, then it is $(D+1, 2)$-special-sound in the $(\mathcal{EF}, \mathcal{DF})$-AGMOS.*

Proof. The lifting condition allows us to bound cases A and X.1 by the AGM. Case X.2 is bounded using the modified FPR assumption, and case Q is bounded using the TOFR reduction. There are $2(D+1)$ paths in the tree of transcripts, so summing the bounds over all paths preserves negligibility. □

This recovers the special-soundness result proved directly in [Lip24]. It shows how our lifting proof can be adapted to Polymath: the lifting condition handles the transfer to the AGM, while the modified FPR and the TOFR assumptions handle the remaining cases.

5 Lifting for Interactive Security Games

In this section, we generalize our lifting framework to interactive AGMOS security games, where the challenger and the adversary interact adaptively. In such games, the adversary may obtain new group elements from the challenger during the execution of the game, with these elements depending on the adversary's previous queries and observations. We then prove a lifting theorem for this larger class of interactive security games.

We begin by defining the security game analyzed in the lifting theorem. We leave the detailed construction of V^h and V^t of the implicit verification polynomial to Appendix B. Next, we state and prove the lifting theorem for the interactive games described in Section 5.1. We then apply our lifting results to signature schemes in Section 6. In Section 6.1, we show that the BLS signature scheme satisfies existential unforgeability under chosen-message attacks (EUF-CMA) in the AGMOS. In Section 6.2, we show that the BBS signature scheme satisfies strong unforgeability under chosen-message attacks (SUF-CMA) in the AGMOS.

5.1 Interactive AGMOS Security Game

The interactive security game extends the non-interactive game of Section 3.1 by a random oracle, which models a hash function, and a trapdoor-dependent oracle, whose answers are group elements that depend on the trapdoor. This follows the EUF-CMA game, in which the trapdoor-dependent oracle is the signing oracle. We first describe the game informally and then define it formally.

Informal description. The challenger gives the adversary the public parameters and initial group elements. The adversary may then make polynomially many adaptive queries to the random oracle, the oblivious sampling oracle, and the trapdoor-dependent oracle. Every group element it outputs must come with an algebraic explanation relative to the group elements received so far. Finally, the adversary outputs a candidate solution. It wins if it passes the pairing check and violates the structural check, if present.

Model choice. Interactive security games can differ in their oracle access and the types of inputs their oracles accept. For example, a random oracle may accept bit strings, group elements, or both. To keep the presentation simple, we use BLS EUF-CMA security (see Section 6.1) as a guiding example. Given the public key $[x]_2$, the adversary can query a random oracle H and request signatures on chosen messages. It wins by producing a signature σ^* on a message m^* that it has not submitted to the signing oracle, such that

$$H(m^*) \bullet [x]_2 = \sigma^* \bullet [1]_2.$$

Additional structural check. As in the non-interactive setting, we distinguish games with a trivial structural check from those with a non-trivial one. For instance, BLS EUF-CMA security has a trivial structural check in our framework. Besides the pairing equation, the challenger checks that the forged message was never submitted to the signing oracle. This additional check concerns only the message and the signing queries, and is preserved by projection to the AGM. For SUF-CMA security, the adversary must output a message–signature pair (m^*, σ^*) that was never returned by the signing oracle. This requirement also concerns the group elements in the signature, so we treat it as a non-trivial structural check. For games with a non-trivial structural check, we require the same lifting condition as in the non-interactive setting:

$$V^t \equiv 0 \quad \Longrightarrow \quad \boldsymbol{\delta} = \mathbf{0}.$$

Games with a trivial structural check do not require this additional condition, but must still satisfy the other requirements of the lifting theorem.

Interactive AGMOS security game. As we have said, the interactive AGMOS security game extends the non-interactive game from Section 3.1 by giving the adversary access to a random oracle $H\colon \{0,1\}^* \to \mathbb{G}_1$ and a trapdoor-dependent oracle $\mathcal{O}_{\boldsymbol{x}}$, in addition to the oblivious sampling oracle $\mathcal{O}_{\mathsf{OS}}$. We write

$$\Gamma_\lambda = (\mathsf{PGen}, \mathsf{KPGen}, \mathsf{MGen}, \mathcal{C}, \mathcal{O}_{\boldsymbol{x}}, H, \boldsymbol{d}, \boldsymbol{d}_R, \mathcal{S}).$$

Setup and public inputs are defined as in the non-interactive game. The tuple $\boldsymbol{d}_R$ gives the degree bounds for the second FPR reduction, which we describe later.

The game follows the non-interactive security game from Section 3.1, with additional access to the oracles H and $\mathcal{O}_{\boldsymbol{x}}$. After receiving the public inputs, the adversary may make polynomially many adaptive queries to the random oracle H, $\mathcal{O}_{\boldsymbol{x}}$, and the oblivious sampling oracle $\mathcal{O}_{\mathsf{OS}}$ through the challenger. Both H and $\mathcal{O}_{\boldsymbol{x}}$ accept only non-group inputs.

The challenger records all queries and responses and maintains the lists

$$[\boldsymbol{r}]_1, \qquad ([\boldsymbol{t}_{\boldsymbol{x}}]_1, \boldsymbol{\zeta}), \qquad [\mathfrak{q}_1]_1, \qquad [\mathfrak{q}_2]_2.$$

Here, $[\boldsymbol{r}]_1$ contains the distinct random oracle outputs, $[\boldsymbol{t}_{\boldsymbol{x}}]_1$ and $\boldsymbol{\zeta}$ contain the group and field elements returned by $\mathcal{O}_{\boldsymbol{x}}$, respectively, and $[\mathfrak{q}_\kappa]_\kappa$ contains the obliviously sampled elements in $\mathbb{G}_\kappa$. Either $[\boldsymbol{t}_{\boldsymbol{x}}]_1$ or $\boldsymbol{\zeta}$ may be empty, depending on the outputs of $\mathcal{O}_{\boldsymbol{x}}$.

For each fresh hash input b, the challenger samples $r_b \leftarrow \mathbb{F}$ and stores $H(b) = [r_b]_1$; repeated inputs receive the same value. The same table is used for hash evaluations inside $\mathcal{O}_{\boldsymbol{x}}$ and during final verification. Hash values used internally by $\mathcal{O}_{\boldsymbol{x}}$ are also made available to the adversary.

Finally, the adversary outputs its candidate solution as in the non-interactive game, with algebraic explanations that may also use the group elements received during the interaction.

Output and verification. The adversary outputs

$$\big(([\mathbb{y}_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa, \boldsymbol{\delta}_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha}, \boldsymbol{b}\big),$$

where $\boldsymbol{\alpha} \in \mathbb{F}^r$ and $\boldsymbol{b} \in \{0,1\}^*$ are its non-group outputs, for a length $r \in \mathsf{poly}(\lambda)$ fixed by the game. Unlike in the non-interactive game, its explanations may also use the group elements obtained during the interaction:

$$\mathbb{y}_1 = \boldsymbol{\gamma}_1 \begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \boldsymbol{r} \\ \boldsymbol{t}_{\boldsymbol{x}} \end{pmatrix} + \boldsymbol{\delta}_1 \mathfrak{q}_1,$$

$$\mathbb{y}_2 = \boldsymbol{\gamma}_2 \mathbb{x}_2(\boldsymbol{x}) + \boldsymbol{\delta}_2 \mathfrak{q}_2.$$

The matrices have the corresponding dimensions. We write ql_h, ql_t, and ql_f for the lengths of $\boldsymbol{r}$, $\boldsymbol{t_x}$, and $\boldsymbol{\zeta}$, respectively.

The *transcript* tr consists of everything the adversary receives or outputs: the public inputs, all oracle queries and responses in their order of occurrence, and the final output together with its algebraic explanations. In particular, it includes $[\boldsymbol{r}]_1$, $[\boldsymbol{t_x}]_1$, $\boldsymbol{\zeta}$, $[\mathbb{q}_1]_1$, and $[\mathbb{q}_2]_2$.

The challenger computes

$$\boldsymbol{M} \leftarrow \mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\zeta}).$$

As in the non-interactive game, the adversary wins Γ_λ if and only if all of the following hold:

1. The algebraic verification accepts, i.e.,

$$\begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \mathbb{y}_1 \\ \mathsf{aux}_1(\boldsymbol{x}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{x}) \\ \mathbb{y}_2 \\ \mathsf{aux}_2(\boldsymbol{x}) \end{pmatrix} = 0. \tag{12}$$

2. The structural condition is violated, i.e., $\mathcal{S}(\mathsf{tr}) = 1$.
3. The game's conditions on non-group data (if any) hold, for example the freshness of the forged message in the BLS game described above.

For a trivial structural check, we take $\mathcal{S} \equiv 1$, so winning requires only the pairing check together with the conditions on non-group data. The final output also includes all non-auxiliary group elements used in the pairing check, together with their algebraic explanations. We assume these can be efficiently obtained from the public inputs, the interaction, and the available oracles, and that including them does not affect acceptance.

Interactive AGM projection and compatibility. Let tr be an interactive AGMOS transcript containing the public inputs, all oracle queries and responses in order, and the adversary's final output with its explanations. Define

$$\mathbb{y}'_1 := \boldsymbol{\gamma}_1 \begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \boldsymbol{r} \\ \boldsymbol{t_x} \end{pmatrix}, \qquad \mathbb{y}'_2 := \boldsymbol{\gamma}_2 \mathbb{x}_2(\boldsymbol{x}).$$

The *interactive AGM projection* $\pi^{\mathrm{int}}_{\mathrm{AGM}}(\mathsf{tr})$ is obtained by replacing the final output with

$$\big(([\mathbb{y}'_\kappa]_\kappa, \boldsymbol{\gamma}_\kappa)_{\kappa=1}^2, \boldsymbol{\alpha}, \boldsymbol{b}\big)$$

and removing all oblivious sampling queries and responses. The public inputs and all queries and responses involving H and $\mathcal{O}_{\boldsymbol{x}}$ are retained in their original order. In particular, the projection retains $[\boldsymbol{r}]_1$, $[\boldsymbol{t_x}]_1$, and $\boldsymbol{\zeta}$. When $\boldsymbol{\delta} = \boldsymbol{0}$, the output group elements remain unchanged.

We fix an interactive AGM variant of Γ_λ with the same setup, pairing check, conditions on non-group data, random oracle H, and trapdoor-dependent oracle $\mathcal{O}_{\boldsymbol{x}}$, but without access to the oblivious sampling oracle. As in the non-interactive setting, we denote it by Γ^{AGM} and its structural check by $\mathcal{S}_{\mathrm{AGM}}$. When applying our lifting theorem, this must be the AGM game whose security we assume or establish.

Algebraic explanations use the basis $([\mathbb{x}_1(\boldsymbol{x})]_1, [\boldsymbol{r}]_1, [\boldsymbol{t_x}]_1)$ in $\mathbb{G}_1$ and $[\mathbb{x}_2(\boldsymbol{x})]_2$ in $\mathbb{G}_2$.

We say that $\mathcal{S}$ is *compatible with the interactive AGM variant* if, for every interactive AGMOS transcript tr with $\boldsymbol{\delta} = \boldsymbol{0}$,

$$\mathcal{S}(\mathsf{tr}) = \mathcal{S}_{\mathrm{AGM}}\big(\pi^{\mathrm{int}}_{\mathrm{AGM}}(\mathsf{tr})\big).$$

For a trivial structural check, we take $\mathcal{S}_{\mathrm{AGM}} \equiv 1$.

Next, we introduce additional notation used in the analysis. We present it only for privately verifiable assumptions; the modifications for publicly verifiable assumptions are analogous to those described in Section 3.2.

Verification polynomials of an interactive security game. Let $\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$ and $\mathbb{Y} = (\mathbb{Y}_1, \mathbb{Y}_2)$ be vectors of indeterminates corresponding to the discrete logarithm vectors $\mathbb{q} = (\mathbb{q}_1, \mathbb{q}_2)$ and $\mathbb{y} = (\mathbb{y}_1, \mathbb{y}_2)$, respectively. As in the non-interactive setting, Equation (12) can be written as $V^{\mathsf{expl}}(\boldsymbol{x}, \mathbb{y}_1, \mathbb{y}_2) = 0$, where

$$V^{\mathsf{expl}}(\boldsymbol{X}, \mathbb{Y}_1, \mathbb{Y}_2) := \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \mathbb{Y}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Y}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix}$$

is the *explicit verification polynomial*.

Let $\boldsymbol{R}$ represent the random oracle output logarithms, and let $\widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R})$ collect the polynomials representing the public inputs in $\mathbb{G}_1$ and the group elements returned by H and $\mathcal{O}_{\boldsymbol{x}}$. The adversary's algebraic explanations give

$$\mathbb{Y}_1 = \boldsymbol{\gamma}_1 \widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) + \boldsymbol{\delta}_1 \mathbb{Q}_1,$$
$$\mathbb{Y}_2 = \boldsymbol{\gamma}_2 \mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_2 \mathbb{Q}_2.$$

Substituting these expressions into V^{expl} yields the *implicit verification polynomial* $V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$. The verification equation is therefore equivalent to $V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = 0$. The explicit construction of these polynomials and the associated matrices is given in Appendix B.

5.2 Interactive Lifting Theorem

In this subsection, we prove the lifting theorem for the interactive games of Section 5.1, following the blueprint of the non-interactive setting. The new feature is that the verification polynomial $V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = V^h(\boldsymbol{X}, \boldsymbol{R}) + V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$ also contains variables $\boldsymbol{R}$ for the discrete logarithms of the random oracle outputs, while V^t again collects the terms that involve $\mathbb{Q}$. This adds case X.3, in which the polynomial is non-zero in $\boldsymbol{R}$ but vanishes at the actual random oracle outputs $\boldsymbol{r}$. Given that the adversary succeeds, we have $V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = 0$, and we distinguish the following cases:

Case A: $V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$. This case is usually impossible.
Case X.1: $V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$. In this case, the polynomial does not depend on $\mathbb{Q}$, so the transcript projects to an AGM transcript, and this case is bounded by AGM security (see below).
Case X.2: $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$. This yields a non-trivial polynomial in $\boldsymbol{X}$ that vanishes at $\boldsymbol{x}$, leading to a reduction to FPR.
Case X.3: $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) \not\equiv 0$, but $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) \equiv 0$. This gives a non-trivial polynomial in $\boldsymbol{R}$, which leads to a second reduction to FPR.
Case Q: $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) \not\equiv 0$. In this case, we obtain a non-trivial relation among the $\mathbb{Q}$ variables, yielding a reduction to the TOFR assumption.

Cases A and X.1 are bounded by AGM security. For games without a non-trivial structural check, the interactive cancellation lemma (Lemma 3) shows that removing the contributions of obliviously sampled elements preserves acceptance, and for games with one, the lifting condition (Definition 15) gives $\boldsymbol{\delta} = \boldsymbol{0}$, so both the pairing check and the violation of the structural check are preserved. Cases X.2, X.3, and Q are reduced to the FPR and TOFR assumptions in Lemma 4, and Theorem 5 combines these bounds.

Lemma 3 (Interactive cancellation lemma). *Let Γ be an interactive AGMOS security game as in Section 5.1, and let* tr *be an AGMOS transcript of Γ that passes the pairing check. If $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0$, then $\pi^{\mathrm{int}}_{\mathrm{AGM}}(\mathsf{tr})$ passes the pairing check of Γ^{AGM} for the same public input and public instance.*

Proof. By definition of the implicit verification polynomial we have

$$V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = V^{\mathsf{expl}}\big(\boldsymbol{X},\, \boldsymbol{\gamma}_1 \widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) + \boldsymbol{\delta}_1 \mathbb{Q}_1,\, \boldsymbol{\gamma}_2 \mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_2 \mathbb{Q}_2\big).$$

Moreover, by the decomposition of the implicit verification polynomial,

$$V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = V^h(\boldsymbol{X}, \boldsymbol{R}) + V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}),$$

where $V^h(\boldsymbol{X}, \boldsymbol{R})$ depends only on $\boldsymbol{M}$ and $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ (see Equation (15)), while $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$ depends additionally on $(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)$ (see Equation (16) and Equation (17)).

Since we are in the case where

$$V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \equiv 0,$$

it follows that

$$V^{\mathsf{expl}}\big(\boldsymbol{X},\, \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) + \boldsymbol{\delta}_1\mathbb{Q}_1,\, \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_2\mathbb{Q}_2\big) = V^h(\boldsymbol{X}, \boldsymbol{R}).$$

Since tr passes the pairing check in the AGMOS game, we obtain

$$V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = V^{\mathsf{expl}}\big(\boldsymbol{x},\, \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{x}, \boldsymbol{r}) + \boldsymbol{\delta}_1\mathbb{q}_1,\, \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{x}) + \boldsymbol{\delta}_2\mathbb{q}_2\big) = 0.$$

Using the previous identity, this implies

$$V^h(\boldsymbol{x}, \boldsymbol{r}) = 0.$$

Now consider the projected transcript $\mathsf{tr}' = \pi^{\mathrm{int}}_{\mathrm{AGM}}(\mathsf{tr})$, whose outputs $\mathbb{y}'_1 = \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{x}, \boldsymbol{r})$ and $\mathbb{y}'_2 = \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{x})$ are obtained by setting $\boldsymbol{\delta}_1 = \boldsymbol{\delta}_2 = \boldsymbol{0}$. Substituting this into the explicit verification polynomial gives

$$V^{\mathsf{expl}}\big(\boldsymbol{X},\, \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}),\, \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{X})\big) = \widetilde{V}^h(\boldsymbol{X}, \boldsymbol{R}) + \widetilde{V}^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}).$$

Since $\widetilde{V}^h(\boldsymbol{X}, \boldsymbol{R})$ depends on the same matrix $\boldsymbol{M}$ and the same explanation matrices $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ as $V^h(\boldsymbol{X}, \boldsymbol{R})$, we have

$$\widetilde{V}^h(\boldsymbol{X}, \boldsymbol{R}) = V^h(\boldsymbol{X}, \boldsymbol{R}).$$

Furthermore, by Equations (16) and (17), setting $\boldsymbol{\delta}_1 = \boldsymbol{\delta}_2 = \boldsymbol{0}$ eliminates all $\mathbb{Q}$-dependent terms, so that

$$\widetilde{V}^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = 0.$$

Therefore,

$$V^{\mathsf{expl}}\big(\boldsymbol{X},\, \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}),\, \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{X})\big) = V^h(\boldsymbol{X}, \boldsymbol{R}).$$

Evaluating at $(\boldsymbol{x}, \boldsymbol{r})$ yields

$$V^{\mathsf{expl}}\big(\boldsymbol{x},\, \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{x}, \boldsymbol{r}),\, \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{x})\big) = V^h(\boldsymbol{x}, \boldsymbol{r}) = 0.$$

Thus,

$$V^{\mathsf{expl}}(\boldsymbol{x}, \mathbb{y}'_1, \mathbb{y}'_2) = 0,$$

showing that the outputs without the contribution of the obliviously sampled group elements satisfy the explicit verification equation.

Finally, for each $\kappa \in \{1, 2\}$, the vector $\mathbb{y}'_\kappa$ has a valid AGM explanation with respect to the public input and the outputs of the random oracle H and the trapdoor-dependent oracle $\mathcal{O}_{\boldsymbol{x}}$, namely $\boldsymbol{\gamma}_\kappa$, since

$$\mathbb{y}'_1 = \boldsymbol{\gamma}_1\widetilde{\mathbb{x}}_1(\boldsymbol{x}, \boldsymbol{r}) \quad \text{and} \quad \mathbb{y}'_2 = \boldsymbol{\gamma}_2\mathbb{x}_2(\boldsymbol{x}).$$

Hence the transformed transcript passes the pairing check in the AGM game for the same public input and public instance. □

The lifting condition for interactive games is identical to the non-interactive one, except that the verification polynomial depends on a larger set of variables.

Definition 15 (Lifting condition for interactive games). *Let $\{\Gamma_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of interactive security games as defined in Section 5.1, with*

$$\Gamma_\lambda = (\mathsf{PGen}, \mathsf{KPGen}, \mathsf{MGen}, \mathcal{C}, \mathcal{O}_{\boldsymbol{x}}, H, \boldsymbol{d}, \boldsymbol{d}_R, \mathcal{S}).$$

We say that this family satisfies the lifting condition *if the following holds for every security parameter λ, every* gp *in the support of* $\mathsf{PGen}(1^\lambda)$*, and every*

$$(\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X}))$$

in the support of $\mathsf{KPGen}(1^\lambda, \mathsf{gp})$*: for all polynomially bounded query lengths* $\mathsf{ql}_h, \mathsf{ql}_t, \mathsf{ql}_f, \mathsf{ql}_1, \mathsf{ql}_2$*, all explanation matrices*

$$\boldsymbol{\gamma}_1 \in \mathbb{F}^{\mathsf{ol}_1 \times (\mathsf{il}_1 + \mathsf{ql}_h + \mathsf{ql}_t)}, \quad \boldsymbol{\gamma}_2 \in \mathbb{F}^{\mathsf{ol}_2 \times \mathsf{il}_2}, \quad \boldsymbol{\delta}_\kappa \in \mathbb{F}^{\mathsf{ol}_\kappa \times \mathsf{ql}_\kappa} \qquad (\kappa \in \{1,2\}),$$

all scalar and bit-string data

$$\boldsymbol{\alpha} \in \mathbb{F}^r, \qquad \boldsymbol{b} \in \{0,1\}^*, \qquad \boldsymbol{\zeta} \in \mathbb{F}^{\mathsf{ql}_f}$$

satisfying the game's prescribed length bounds, every oracle interaction consistent with these data, and every $\boldsymbol{M}$ *in the support of* $\mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\zeta})$*, we have*

$$\begin{aligned} &\textit{if} \quad V^t_{\mathbb{x},\mathsf{aux},\boldsymbol{M},\boldsymbol{\gamma},\boldsymbol{\delta}}(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = 0, \\ &\textit{then} \quad \boldsymbol{\delta} = \boldsymbol{0}. \end{aligned}$$

Here, V^t is the $\mathbb{Q}$-dependent part of the implicit verification polynomial determined by the setup, the oracle interaction, the verification matrix, and the explanation matrices. The vectors

$$\boldsymbol{X} = (X_1, \ldots, X_m), \qquad \boldsymbol{R} = (R_1, \ldots, R_{\mathsf{ql}_h}), \qquad \mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2),$$

with $\mathbb{Q}_\kappa = (\mathbb{Q}_{\kappa,1}, \ldots, \mathbb{Q}_{\kappa,\mathsf{ql}_\kappa})$, consist of indeterminates, and the identity is a formal polynomial identity in these variables.

We now define simulatability of the trapdoor-dependent oracle. This means that $\mathcal{O}_{\boldsymbol{x}}$ can be simulated using only the public input, without knowing $\boldsymbol{x}$. Our reductions to the AGM require this property because the AGM adversary must simulate the oracle for the AGMOS adversary without knowing the trapdoor.

Definition 16 (Simulatability of the trapdoor-dependent oracle). *Let $\mathcal{O}_{\boldsymbol{x}}$ be a trapdoor-dependent oracle in an interactive security game Γ_λ. We say that $\mathcal{O}_{\boldsymbol{x}}$ is* simulatable for the FPR reductions *if there exist PPT stateful simulators Sim_X and Sim_R with the following properties. Both receive the public input and the polynomial descriptions produced by* KPGen*.*

Simulation with hidden trapdoors. *The simulator Sim_X has access to $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m,\boldsymbol{x}}$ but does not know $\boldsymbol{x}$. For each fresh hash input, it samples and records a logarithm $r_i \leftarrow \mathbb{F}$ and returns $[r_i]_1$. Using these values and its* FPR *oracle, it answers adaptive queries to $\mathcal{O}_{\boldsymbol{x}}$.*

Simulation with hidden hash logarithms. *The simulator Sim_R knows $\boldsymbol{x}$ and has access to $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d}_R,\mathsf{ql}_h,\boldsymbol{r}}$, but does not know $\boldsymbol{r}$. Here, ql_h bounds the total number of distinct hash inputs, including those used internally by $\mathcal{O}_{\boldsymbol{x}}$ or during final verification. For each fresh hash input, it assigns the next unused variable R_i and obtains $[r_i]_1$ by querying its* FPR *oracle on $(1, R_i)$. Using these values, the known trapdoors, and its* FPR *oracle, it answers adaptive queries to $\mathcal{O}_{\boldsymbol{x}}$.*

Each simulator uses a single hash table for all hash evaluations, including those made internally by $\mathcal{O}_{\boldsymbol{x}}$ and during final verification. Repeated inputs always receive the same value. Both simulators reproduce the complete oracle responses, including group elements, field elements, and any failure behavior, and maintain the polynomial descriptions needed to construct the implicit verification polynomial. Every FPR *query must satisfy the degree bounds of the corresponding oracle.*

For every PPT adversary, each simulation must reproduce the joint distribution of the trapdoors, the hash logarithms, and the interaction transcript in the real game. Oblivious sampling queries are answered according to the same $(\mathcal{EF}, \mathcal{DF})$-AGMOS oracle in both experiments.

Theorem 5 (Interactive AGM-to-AGMOS lifting). *Let Γ be an interactive AGMOS security game as in Section 5.1 with structural check $\mathcal{S}$ and a trapdoor-dependent oracle $\mathcal{O}_{\boldsymbol{x}}$ that is simulatable (Definition 16), and let Γ^{AGM} be its interactive AGM variant. Suppose that*

1. *the structural checks of Γ and Γ^{AGM} are trivial, or*
2. *Γ satisfies the interactive lifting condition (Definition 15) and $\mathcal{S}$ is compatible with Γ^{AGM}.*

Then for every PPT $(\mathcal{EF},\mathcal{DF})$-AGMOS adversary $\mathcal{A}$ there are PPT adversaries $\mathcal{B}$, $\mathcal{B}_{\mathsf{fpr},1}$, $\mathcal{B}_{\mathsf{fpr},2}$, and $\mathcal{B}_{\mathsf{tofr}}$ such that

$$\mathsf{Adv}^{\Gamma}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\Gamma^{\mathrm{AGM}}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr},1}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d}_R,\mathsf{ql}_h}_{\mathcal{B}_{\mathsf{fpr},2}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda),$$

where $\boldsymbol{d}$ and $\boldsymbol{d}_R$ are the degree bounds of Γ, m is its number of trapdoors, and ql_h bounds the number of distinct random oracle inputs (Definition 16).

Proof. Fix a PPT $(\mathcal{EF},\mathcal{DF})$-AGMOS adversary $\mathcal{A}$ for Γ_λ. Let Win denote the event that $\mathcal{A}$ wins the game Γ_λ, i.e., Win is the event that the pairing check Equation (12) holds, together with the game's conditions on non-group data (if any) and with $\mathcal{S}(\mathsf{tr}) = 1$ in the non-trivial-structural-check case.

Let Bad denote the event that $\boldsymbol{\delta} \neq \mathbf{0}$ in games with a non-trivial structural check, and the event that $V^t(\boldsymbol{X},\boldsymbol{R},\mathbb{Q}) \neq 0$ in games with a trivial structural check.

In both cases, we decompose the winning probability as

$$\Pr[\mathsf{Win}] = \Pr[\mathsf{Win} \wedge \neg\mathsf{Bad}] + \Pr[\mathsf{Win} \wedge \mathsf{Bad}].$$

We first bound the term $\Pr[\mathsf{Win} \wedge \mathsf{Bad}]$.

In the case of games with a non-trivial structural check, the lifting condition (Definition 15) implies that

$$V^t_{\mathbb{x},\mathsf{aux},\boldsymbol{M},\boldsymbol{\gamma},\boldsymbol{\delta}}(\boldsymbol{X},\boldsymbol{R},\mathbb{Q}) = 0 \;\Rightarrow\; \boldsymbol{\delta} = \mathbf{0}.$$

Taking the contrapositive, $\boldsymbol{\delta} \neq \mathbf{0}$ implies $V^t(\boldsymbol{X},\boldsymbol{R},\mathbb{Q}) \neq 0$. Since Bad is the event that $\boldsymbol{\delta} \neq \mathbf{0}$, this yields

$$\Pr[\mathsf{Win} \wedge \mathsf{Bad}] \le \Pr[\mathsf{Win} \wedge (V^t(\boldsymbol{X},\boldsymbol{R},\mathbb{Q}) \neq 0)].$$

In the case of a trivial structural check, this bound holds by definition of Bad. Hence, in both cases, the inequality above holds.

Dropping the additional requirements, namely the conditions on non-group data and, in the non-trivial case, $\mathcal{S}(\mathsf{tr}) = 1$, can only increase the probability. Therefore, in both the structural and non-structural cases, we can bound $\Pr[\mathsf{Win} \wedge \mathsf{Bad}]$ by

$$\Pr[\text{AGMOS-specific}] = \Pr\left[\begin{array}{l} \begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \mathbb{y}_1 \\ \mathsf{aux}_1(\boldsymbol{x}) \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \mathbb{x}_2(\boldsymbol{x}) \\ \mathbb{y}_2 \\ \mathsf{aux}_2(\boldsymbol{x}) \end{pmatrix} = 0 \wedge \\ V^t(\boldsymbol{X},\boldsymbol{R},\mathbb{Q}) \not\equiv 0 \end{array} \middle| \begin{array}{l} \mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda); \quad \boldsymbol{x} \leftarrow \mathbb{F}^m; \\ \big(\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \\ \qquad \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X})\big) \\ \qquad \leftarrow \mathsf{KPGen}(1^\lambda, \mathsf{gp}); \\ \mathsf{tr} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{OS}},\mathcal{O}_{\boldsymbol{x}},H} \\ \qquad \big(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2\big); \\ \boldsymbol{M} \leftarrow \mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\zeta}) \end{array}\right]. \tag{13}$$

We assume throughout that the adversary's output and all oracle responses have the types and dimensions specified by the game. Malformed outputs are rejected. By Lemma 4, the probability in Equation (13) is bounded by the sum of the $(\boldsymbol{d},m)$-FPR, $(\boldsymbol{d}_R,\mathsf{ql}_h)$-FPR, and $(\mathcal{EF},\mathcal{DF})$-TOFR advantages of the corresponding reductions. Thus,

$$\Pr[\mathsf{Win} \wedge \mathsf{Bad}] \le \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr},1}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d}_R,\mathsf{ql}_h}_{\mathcal{B}_{\mathsf{fpr},2}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Next, we consider the term $\Pr[\mathsf{Win} \wedge \neg\mathsf{Bad}]$. We define an adversary $\mathcal{B}$ in the interactive AGM variant of the game. On input

$$(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2),$$

and with oracle access to H and $\mathcal{O}_{\boldsymbol{x}}$, $\mathcal{B}$ runs $\mathcal{A}$ internally and simulates its interaction as follows.

1. *Random oracle queries.*
   Whenever $\mathcal{A}$ queries the random oracle H on a valid (non-group) input, the adversary $\mathcal{B}$ forwards this query to its own random oracle and obtains the corresponding group element. It then returns this element to $\mathcal{A}$ and additionally forwards it to its interactive AGM challenger, so that it is recorded as a new algebraic basis element.
2. *Trapdoor-dependent oracle $\mathcal{O}_{\boldsymbol{x}}$-queries.*
   Whenever $\mathcal{A}$ makes a query to $\mathcal{O}_{\boldsymbol{x}}$, the adversary $\mathcal{B}$ forwards the query to its own $\mathcal{O}_{\boldsymbol{x}}$-oracle and returns the response to $\mathcal{A}$. Any new group elements returned by the oracle are forwarded to the interactive AGM challenger so that they are recorded and added to the current algebraic basis.
3. *Oblivious sampling queries.*
   Since the interactive AGM game does not provide access to the oblivious sampling oracle, $\mathcal{B}$ answers these queries locally by simulating the $(\mathcal{EF}, \mathcal{DF})$-AGMOS oracle $\mathcal{O}_{\mathsf{OS}}$. This simulation is efficient because the functions in $\mathcal{EF}$ are efficiently computable and the distributions in $\mathcal{DF}$ are efficiently sampleable.

When $\mathcal{A}$ terminates, depending on whether the game had an additional structural check, the following happens. If the game Γ_λ did not have an additional structural check, then $\neg\mathsf{Bad}$ means that $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = 0$. In this case, the AGM adversary $\mathcal{B}$ applies the transformation from Lemma 3 to obtain projected group elements, and forwards them together with all relevant non-group information to its interactive AGM challenger. By Lemma 3, the projected transcript remains accepting for the same public input and the same public instance.

If the game Γ_λ has an additional structural check, $\mathcal{B}$ checks whether $\boldsymbol{\delta} = \mathbf{0}$. If $\boldsymbol{\delta} = \mathbf{0}$, then $\mathcal{B}$ forwards what $\mathcal{A}$ output to the interactive AGM challenger. Otherwise, $\mathcal{B}$ outputs $\perp$. Whenever $\mathsf{Win} \wedge \neg\mathsf{Bad}$ occurs in this branch, we have $\boldsymbol{\delta} = \mathbf{0}$ and $\mathcal{A}$'s output group elements already use only the algebraic basis available in the interactive AGM game. Moreover, compatibility of the structural check gives

$$\mathcal{S}_{\mathrm{AGM}}\big(\pi^{\mathrm{int}}_{\mathrm{AGM}}(\mathsf{tr})\big) = \mathcal{S}(\mathsf{tr}) = 1.$$

The conditions on non-group data hold for $\mathcal{B}$ whenever they hold for $\mathcal{A}$, because $\mathcal{B}$ outputs the same non-group data and forwards all of $\mathcal{A}$'s queries to H and $\mathcal{O}_{\boldsymbol{x}}$. Thus, in both the trivial and non-trivial structural-check cases, whenever $\mathsf{Win} \wedge \neg\mathsf{Bad}$ occurs, $\mathcal{B}$ wins the interactive AGM variant of the game. Hence,

$$\Pr[\mathsf{Win} \wedge \neg\mathsf{Bad}] \leq \Pr\big[\mathcal{B} \text{ wins the interactive AGM variant of the game } \Gamma_\lambda\big].$$

Combining the bounds concludes the proof. □

We now analyze the cases that are specific to the AGMOS setting. These correspond to executions in which the adversary wins the interactive security game while $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \neq 0$. As in the non-interactive setting, we show that such executions can be reduced to either the FPR assumption or the $(\mathcal{EF}, \mathcal{DF})$-$\mathsf{TOFR}$ assumption.

Lemma 4 (Security for the AGMOS-specific cases in the interactive game). *Let Γ be an interactive AGMOS security game as in Section 5.1 with a trapdoor-dependent oracle $\mathcal{O}_{\boldsymbol{x}}$ that is simulatable (Definition 16). For every PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$ there are PPT adversaries $\mathcal{B}_{\mathsf{fpr},1}$, $\mathcal{B}_{\mathsf{fpr},2}$, and $\mathcal{B}_{\mathsf{tofr}}$ such that the probability in Equation (13) satisfies*

$$\Pr[\textit{AGMOS-specific}] \leq \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr},1}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d}_R,\mathsf{ql}_h}_{\mathcal{B}_{\mathsf{fpr},2}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Proof. Assume that the transcript assembled from $\mathcal{A}$'s interaction satisfies the pairing check and $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \neq 0$. Using Equations (16) and (17), we can derive the explicit form of the polynomial $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$. However, for simplicity, we decide not to write it out explicitly.

Case X.2: $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \neq 0$, but $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) = 0$. We construct an adversary $\mathcal{B}_{\mathsf{fpr},1}$ for the $(\boldsymbol{d}, m)$-FPR game as follows.

The adversary $\mathcal{B}_{\mathsf{fpr},1}$, on input gp, first runs

$$\mathsf{KPGen}(1^\lambda, \mathsf{gp}) \to (\mathbb{x}_1(\boldsymbol{X}), \mathbb{x}_2(\boldsymbol{X}), \mathsf{aux}_1(\boldsymbol{X}), \mathsf{aux}_2(\boldsymbol{X})).$$

It then queries the FPR oracle $\mathcal{O}^{\mathsf{fpr}}_{\boldsymbol{d},m,\boldsymbol{x}}(\cdot)$ to obtain the encoded public inputs

$$[\mathbb{x}_1(\boldsymbol{x})]_1 \qquad \text{and} \qquad [\mathbb{x}_2(\boldsymbol{x})]_2.$$

Next, $\mathcal{B}_{\mathsf{fpr},1}$ runs the adversary $\mathcal{A}$ on these inputs:

$$\mathcal{A}^{\mathcal{O}_{\mathsf{OS}},H,\mathcal{O}_{\boldsymbol{x}}}(\mathsf{gp}, [\mathbb{x}_1(\boldsymbol{x})]_1, [\mathbb{x}_2(\boldsymbol{x})]_2) \to \mathsf{tr},$$

while simulating all oracle accesses of $\mathcal{A}$. In particular, it simulates the random oracle via lazy sampling (keeping a list of previously queried inputs), the oblivious sampling oracle directly, and the trapdoor-dependent oracle using its access to the FPR oracle. Since we require the trapdoor-dependent oracle to be simulatable (see Definition 16), $\mathcal{B}_{\mathsf{fpr},1}$ can simulate the $\mathcal{O}_{\boldsymbol{x}}$ oracle with access to its FPR oracle.
Using the values $\boldsymbol{\alpha}$, $\boldsymbol{b}$, and $\boldsymbol{\zeta}$ from the transcript, the adversary computes the verification matrix

$$\mathsf{MGen}(1^\lambda, \mathsf{gp}, \boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\zeta}) \to \boldsymbol{M}.$$

From the resulting transcript, together with the FPR oracle access and using Equations (16) and (17), the adversary constructs the polynomial $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$. By assumption, we have $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) \neq 0$, but $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) = 0$. Therefore, $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$ contains a non-zero monomial in the variables $\boldsymbol{R}$ and $\mathbb{Q}$, whose coefficient is a non-zero polynomial in $\boldsymbol{X}$. The adversary extracts such a non-zero coefficient $g(\boldsymbol{X}) \in \mathbb{F}[\boldsymbol{X}]$ and outputs it as its solution to the $(\boldsymbol{d}, m)$-FPR game. By the degree bounds, $\deg(g(\boldsymbol{X})) \leq \max(md_1, md_2) \leq d_g$. Therefore, $\mathcal{B}_{\mathsf{fpr},1}$ is successful. Thus, the probability of case X.2 is bounded by $\mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},m}_{\mathcal{B}_{\mathsf{fpr},1}}(\lambda)$.

Case X.3: $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) \neq 0$, but $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) = 0$.
We reduce this case to the polynomial form of the $(\boldsymbol{d}_R, \mathsf{ql}_h)$-$\mathsf{FPR}$ assumption. We construct an adversary $\mathcal{B}_{\mathsf{fpr},2}$ as follows.
The adversary $\mathcal{B}_{\mathsf{fpr},2}$ samples $\boldsymbol{x} \leftarrow \mathbb{F}^m$ and computes the public inputs $[\mathbb{x}_1(\boldsymbol{x})]_1$ and $[\mathbb{x}_2(\boldsymbol{x})]_2$, which it provides to $\mathcal{A}$. It then simulates the interaction of $\mathcal{A}$ with all oracles:

1. *Random oracle.* The random oracle H is simulated via the FPR oracle. The adversary $\mathcal{B}_{\mathsf{fpr},2}$ maintains a list L mapping oracle queries to variables. Upon a query b:
   - If b has been queried before, $\mathcal{B}_{\mathsf{fpr},2}$ returns the previously assigned value.
   - Otherwise, $\mathcal{B}_{\mathsf{fpr},2}$ assigns a fresh variable R_i, stores (b, i) in L, and queries its FPR oracle to obtain the encoding $[R_i]_1$, which it returns to $\mathcal{A}$.

   This ensures that repeated queries are answered consistently, while fresh queries yield independent uniformly random values, as guaranteed by the definition of the FPR oracle, thereby matching the distribution of a random oracle.
   To support this simulation, the adversary $\mathcal{B}_{\mathsf{fpr},2}$ requires access to an FPR oracle defined over sufficiently many variables. In particular, the number of distinct variables needed is bounded by the number ql_h of registered random oracle values. Since this quantity is polynomial in the security parameter λ, we can instantiate the $(\boldsymbol{d}_R, \mathsf{ql}_h)$-$\mathsf{FPR}$ oracle.
   The degree tuple $\boldsymbol{d}_R$ is chosen as part of the game so that the FPR oracle can return the registered random oracle outputs and all polynomial trapdoor-oracle group responses after the trapdoor variables are fixed. Its output degree bound also covers every coefficient polynomial in $\boldsymbol{R}$ that can arise from $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q})$.
2. *Oblivious sampling oracle.* This oracle is simulated directly, as the corresponding distributions are efficiently sampleable and computable.
3. *Trapdoor-dependent oracle.* Queries to $\mathcal{O}_{\boldsymbol{x}}$ are simulated directly. Since $\mathcal{B}_{\mathsf{fpr},2}$ samples the trapdoor $\boldsymbol{x}$ itself, it can answer the trapdoor-dependent part of the oracle exactly as in the real game. Moreover, whenever $\mathcal{O}_{\boldsymbol{x}}$ requires access to the random oracle, $\mathcal{B}_{\mathsf{fpr},2}$ uses the same FPR-based simulation of H described above. Thus, the simulated oracle has the same distribution as the real trapdoor-dependent oracle.

From the resulting transcript, $\mathcal{B}_{\mathsf{fpr},2}$ constructs the polynomial $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q})$. By assumption, this polynomial is non-zero, but satisfies $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) = 0$. We view $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q})$ as a polynomial in the variables $\mathbb{Q}$, whose coefficients are polynomials in $\boldsymbol{R}$. Since $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) = 0$, all these coefficient polynomials must vanish at $\boldsymbol{r}$. At the same time, since $V^t(\boldsymbol{x}, \boldsymbol{R}, \mathbb{Q}) \neq 0$, at least one of these coefficients is a non-zero polynomial $g(\boldsymbol{R})$. It follows that $g(\boldsymbol{R}) \neq 0$, but $g(\boldsymbol{r}) = 0$. The adversary $\mathcal{B}_{\mathsf{fpr},2}$ outputs this polynomial g as its solution to the $(\boldsymbol{d}_R, \mathsf{ql}_h)$-FPR game. Thus, the probability of case X.3 is bounded by $\mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d}_R,\mathsf{ql}_h}_{\mathcal{B}_{\mathsf{fpr},2}}(\lambda)$.

Case Q: $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) \neq 0$, but $V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = 0$.

We construct a $(\mathcal{EF}, \mathcal{DF})$-TOFR adversary $\mathcal{B}_{\mathsf{tofr}}$ as follows. The adversary samples $\boldsymbol{x} \leftarrow \mathbb{F}^m$, computes the corresponding public inputs, and provides them to $\mathcal{A}$. It then simulates the oracles for $\mathcal{A}$ as follows:

1. *Random oracle.* Simulated via lazy sampling.
2. *Oblivious sampling oracle.* Queries are forwarded to the $(\mathcal{EF}, \mathcal{DF})$ oblivious oracle of $\mathcal{B}_{\mathsf{tofr}}$, and responses are relayed back to $\mathcal{A}$.
3. *Trapdoor-dependent oracle.* Simulated directly using the known trapdoor $\boldsymbol{x}$ and the lazy-sampled random oracle.

From the resulting transcript, $\mathcal{B}_{\mathsf{tofr}}$ constructs the polynomial $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q})$. Since $V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) \neq 0$, it can be written as a non-trivial linear combination

$$V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{Q}) = \boldsymbol{u}^T \begin{pmatrix} \mathbb{Q}_1 \\ \mathbb{Q}_2 \\ \mathbb{Q}_1 \otimes \mathbb{Q}_2 \end{pmatrix}$$

for some non-zero vector $\boldsymbol{u}$.

Define

$$\boldsymbol{v} := \begin{pmatrix} V^h(\boldsymbol{x}, \boldsymbol{r}) \\ \boldsymbol{u} \end{pmatrix}.$$

Using that $V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = 0$, we obtain

$$\boldsymbol{v}^T \begin{pmatrix} 1 \\ \mathbb{q}_1 \\ \mathbb{q}_2 \\ \mathbb{q}_1 \otimes \mathbb{q}_2 \end{pmatrix} = V^h(\boldsymbol{x}, \boldsymbol{r}) + V^t(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = 0.$$

Since $\boldsymbol{u} \neq \boldsymbol{0}$, we have $\boldsymbol{v} \neq \boldsymbol{0}$, and thus $\mathcal{B}_{\mathsf{tofr}}$ outputs a non-trivial linear relation, breaking the $(\mathcal{EF}, \mathcal{DF})$-TOFR assumption. Thus, the probability of case Q is bounded by $\mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda)$. □

6 Applications of the Interactive Lifting Theorem

We next apply our interactive lifting framework to existential unforgeability under chosen-message attack (EUF-CMA) of BLS signatures and strong unforgeability under chosen-message attack (SUF-CMA) of BBS signatures. We explain the framework and the notions needed for interactive games in detail, but applying the theorem is often simpler. Once its requirements hold, it lifts an AGM security bound to the AGMOS under the corresponding FPR and TOFR assumptions. For games with a structural check, one must also check the lifting condition and show that the reduction preserves the structural violation. These checks require only short arguments in our applications.

6.1 EUF-CMA Security of the BLS Signature Scheme

In this subsection, we introduce the notion of EUF-CMA security and apply our lifting theorem to show that the BLS signature scheme introduced in [BLS01] is EUF-CMA secure in the AGMOS. In Appendix A, we show that the BLS signature scheme is EUF-CMA secure in the AGM when instantiated with Type-3 pairings.

Why Type-3 pairings. We focus on Type-3 pairings for two reasons. First, each output group element must have an algebraic explanation using the inputs and obliviously sampled elements in the same source group. In the Type-2 setting, a homomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ allows an adversary to transfer an element between the groups, but the resulting element does not need to have an explanation in the $\mathbb{G}_1$ basis. The Type-1 setting, where the two source groups coincide, also requires a different treatment. Type-3 pairings avoid these issues because the source groups are distinct and no efficiently computable homomorphism between them is assumed to exist. Second, Type-3 pairings are commonly used in BLS implementations for their efficiency at the required security level [BCLS22].

Let

$$H\colon \{0,1\}^* \to \mathbb{G}_1$$

be a hash function modeled as a random oracle. Since the BLS signature scheme signs by hashing messages to group elements, we treat every hash value as an element of $\mathbb{G}_1$. In the analysis, for every message m, we write

$$H(m) = [r_m]_1$$

for the corresponding discrete logarithm $r_m \in \mathbb{F}$. The variables in the verification polynomial represent the discrete logarithms of the hash oracle's group outputs. We also recall the definition of a signature scheme below and the definition of the BLS signature scheme in Figure 5.

Definition 17 (Digital signature scheme). *A digital signature scheme consists of three algorithms* $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$*:*

$\mathsf{KGen}(1^\lambda)$**:** *The key generation algorithm* KGen *takes as input the security parameter* 1^λ *and proceeds as follows. First, the algorithm* $\mathsf{PGen}(1^\lambda)$ *is invoked to generate the group parameters* gp*. Once* gp *has been generated, a key pair* (sk, pk) *is generated.*

$\mathsf{Sign}_{sk}(m)$**:** *The signing algorithm* Sign *takes as input a secret key* sk *and a message* m *from the message space, and outputs a signature* $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$ *on the message* m*.*

$\mathsf{Ver}_{pk}(m, \sigma)$**:** *The verification algorithm* Ver *takes as input a public key* pk*, a message* m*, and a signature* σ*, and outputs a bit. An output of* 1 *indicates that the signature* σ *is valid with respect to the message* m*, while an output of* 0 *indicates that the signature is invalid.*

We require that with overwhelming probability $\mathsf{Ver}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$*.*

For the BLS signature scheme of [BLS01], the original authors prove EUF-CMA security in the ROM under the CDH assumption. Later, Fuchsbauer, Kiltz, and Loss [FKL18] strengthened the analysis by giving a tight proof for BLS in the AGM + ROM under the DLOG assumption. Existential unforgeability under chosen-message attack means that an adversary cannot produce a valid message–signature pair for any message that was not previously queried to the signing oracle, even if it is allowed to obtain signatures on messages of its choice. We present the EUF-CMA security game in Figure 6.

Theorem 6 (EUF-CMA security of BLS in the AGMOS). *Let* Γ_{BLS} *be the* EUF-CMA *game of Figure 6 in the* $(\mathcal{EF}, \mathcal{DF})$*-AGMOS, with the hash function modeled as a random oracle, and let* $\Gamma^{\mathrm{AGM}}_{\mathsf{BLS}}$ *be its interactive AGM variant. Let* $\mathsf{ql}_h \in \mathsf{poly}(\lambda)$ *bound the total number of registered hash values, and let* $\boldsymbol{d} = \boldsymbol{d}_R = (1,1,1,1)$*. For every PPT* $(\mathcal{EF}, \mathcal{DF})$*-AGMOS adversary* $\mathcal{A}$ *there are PPT adversaries* $\mathcal{B}$*,* $\mathcal{B}_{\mathsf{fpr},1}$*,* $\mathcal{B}_{\mathsf{fpr},2}$*, and* $\mathcal{B}_{\mathsf{tofr}}$ *such that*

$$\mathsf{Adv}^{\Gamma_{\mathsf{BLS}}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\Gamma^{\mathrm{AGM}}_{\mathsf{BLS}}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d},1}_{\mathcal{B}_{\mathsf{fpr},1}}(\lambda) + \mathsf{Adv}^{\mathsf{FPR},\boldsymbol{d}_R,\mathsf{ql}_h}_{\mathcal{B}_{\mathsf{fpr},2}}(\lambda) + \mathsf{Adv}^{\mathsf{TOFR}}_{\mathcal{B}_{\mathsf{tofr}}}(\lambda).$$

Proof. The game in Figure 6 fits our framework described in Section 5.1. The verification polynomial corresponding to the EUF-CMA game for the BLS signature scheme is given by

$$V(X, \boldsymbol{R}, \mathbb{Q}) = X\left(\sum_{i=1}^{\mathsf{q}_s} R_i \tilde{a}_i - R^*\right) + \left(\hat{a} + \sum_{i=1}^{\mathsf{q}_h} R_i \bar{a}_i\right) + \sum_{i=1}^{\mathsf{ql}_1} \delta_i \mathbb{Q}_{1,i},$$

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

1 : $\mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda)$
2 : $sk := x \leftarrow \mathbb{F}$
3 : $pk := [x]_2$
4 : **return** (sk, pk)

$\mathsf{Sign}_{sk}(m)$

1 : $\sigma \leftarrow xH(m)$
2 : **return** σ

$\mathsf{Ver}_{pk}(m, \sigma)$

1 : **return** $(H(m) \bullet pk = \sigma \bullet [1]_2)$

Fig. 5: The BLS signature scheme.

Game G

1 : $\mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda)$
2 : $sk := x \leftarrow \mathbb{F}$
3 : $pk := [x]_2$
4 : $\mathcal{Q} := \emptyset$
5 : initialize H as undefined everywhere
6 : $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{sk}(\cdot), H(\cdot)}(\mathsf{gp}, pk)$
7 : **return** $(m^* \notin \mathcal{Q} \wedge H(m^*) \bullet [x]_2 = \sigma^* \bullet [1]_2)$

$\mathsf{Sign}_{sk}(m_i)$

1 : $\mathcal{Q} := \mathcal{Q} \cup \{m_i\}$
2 : $\sigma_i \leftarrow xH(m_i)$
3 : **return** $(H(m_i), \sigma_i)$

$H(m_i)$

1 : **if** $H(m_i)$ is defined **then**
2 : **return** $H(m_i)$
3 : **else return** $H(m_i) = H_i \leftarrow \mathbb{G}_1$

Fig. 6: BLS EUF-CMA game G.

where X denotes the variable corresponding to the discrete logarithm of the public key. The variables $\boldsymbol{R}$ represent the discrete logarithms of the random oracle outputs, including hash values obtained during signing queries, explicit hash queries, and verification of the final forgery on message m^*. We write R^* for the variable representing the hash value $H(m^*)$. We assume without loss of generality that the adversary has queried the random oracle on the forgery message m^* during the interaction. Finally, the variables $\mathbb{Q}_{1,i}$ for $i \in [1, \mathsf{q}_1]$ correspond to the discrete logarithms of the obliviously sampled group elements obtained through the oblivious sampling oracle. Here, $\hat{a}$, $\bar{a}_i$, $\tilde{a}_i$, and δ_i are the coefficients of the algebraic explanation of σ^* with respect to $[1]_1$, the hash values, the signing responses, and the obliviously sampled elements, respectively, and q_h and q_s are the numbers of explicit hash queries and signing queries (see Appendix A). The $V^h(X, \boldsymbol{R})$ and $V^t(X, \boldsymbol{R}, \mathbb{Q})$ parts of $V(X, \boldsymbol{R}, \mathbb{Q})$ are

$$V^h(X, \boldsymbol{R}) = X\left(\sum_{i=1}^{\mathsf{q}_s} R_i\tilde{a}_i - R^*\right) + \left(\hat{a} + \sum_{i=1}^{\mathsf{q}_h} R_i\bar{a}_i\right) \quad \text{and} \quad V^t(X, \boldsymbol{R}, \mathbb{Q}) = \sum_{i=1}^{\mathsf{q}_1} \delta_i\mathbb{Q}_{1,i}.$$

For BLS, every signing-oracle response has the polynomial form XR. This polynomial has total degree 2, but degree 1 in each variable family separately. Since each FPR instance bounds per-variable degrees after the other variable family has been fixed, the degree-one tuples $\boldsymbol{d} = \boldsymbol{d}_R = (1, 1, 1, 1)$ suffice. Moreover, $V^t(X, \boldsymbol{R}, \mathbb{Q})$ is independent of both X and $\boldsymbol{R}$, so cases X.2 and X.3 are vacuous for BLS; the FPR assumptions are included exactly to instantiate the generic interactive lifting theorem.

Since the verification procedure consists of a single pairing check and verifying that the message m^* does not belong to the set $\mathcal{Q}$, the structural check is trivial. Indeed, the condition $m^* \notin \mathcal{Q}$ depends only on the message and the signing queries, and is independent of any group elements. Applying Theorem 5 to BLS,

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

$\mathsf{gp} \leftarrow \mathsf{PGen}(1^\lambda)$
$[g_1]_1 \leftarrow \mathbb{G}_1^*$
$([h_1]_1, \ldots, [h_\ell]_1) \leftarrow \mathbb{G}_1^\ell$
$[g_2]_2 \leftarrow \mathbb{G}_2^*$
$\mathsf{par} \leftarrow (\mathsf{gp}, [g_1]_1, ([h_1]_1, \ldots, [h_\ell]_1), [g_2]_2)$
$x \leftarrow \mathbb{F}$
$pk_2 \leftarrow x\,[g_2]_2$
$sk \leftarrow (\mathsf{par}, x)$
$pk \leftarrow (\mathsf{par}, pk_2)$
return (sk, pk)

$\mathsf{Sign}_{sk}(m = (m_1, \ldots, m_\ell))$

$[c]_1 \leftarrow [g_1]_1 + \sum_{i=1}^{\ell} m_i\,[h_i]_1$
$e \leftarrow \mathbb{F}$
if $e = -x$ **then return** $\perp$
$[a]_1 \leftarrow (x+e)^{-1}\,[c]_1$
return $\sigma = ([a]_1, e)$

$\mathsf{Ver}_{pk}(m, \sigma = ([a]_1, e))$

$[c]_1 \leftarrow [g_1]_1 + \sum_{i=1}^{\ell} m_i\,[h_i]_1$
return $[a]_1 \bullet (e\,[g_2]_2 + pk_2)$
$\quad = [c]_1 \bullet [g_2]_2$

Fig. 7: The BBS signature scheme.

Game G

$S \leftarrow \emptyset$
$(sk, pk) \leftarrow \mathsf{KGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{sk}(\cdot)}(pk)$
$\sigma^* = ([a^*]_1, e^*)$
$[c^*]_1 \leftarrow [g_1]_1 + \sum_{i=1}^{\ell} m_i^*\,[h_i]_1$
if $(m^*, \sigma^*) \notin S \wedge$
$\quad [a^*]_1 \bullet (e^*\,[g_2]_2 + pk_2)$
$\quad = [c^*]_1 \bullet [g_2]_2$ **then**
$\quad$ **return true**
return false

Oracle $\mathsf{Sign}_{sk}(m = (m_1, \ldots, m_\ell))$

$[c]_1 \leftarrow [g_1]_1 + \sum_{i=1}^{\ell} m_i\,[h_i]_1$
$e \leftarrow \mathbb{F}$
if $e = -x$ **then return** $\perp$
$[a]_1 \leftarrow (x+e)^{-1}\,[c]_1$
$\sigma \leftarrow ([a]_1, e)$
$S \leftarrow S \cup \{(m, \sigma)\}$
return σ

Fig. 8: BBS SUF-CMA game G.

with $m = 1$, gives the claimed bound. The probability of an AGM adversary winning the EUF-CMA security game for Type-3 pairings is analyzed in Appendix A. □

6.2 SUF-CMA Security of the BBS Signature Scheme

In this subsection, we recall strong unforgeability under chosen-message attack and prove a lifting result for the BBS variant analyzed by Tessaro and Zhu [TZ23], based on the proposal of Boneh, Boyen, and Shacham [BBS04]. We assume SUF-CMA security in the AGM for the scheme. Strong unforgeability under chosen-message attack means that, even after obtaining signatures on messages of its choice, an adversary cannot produce a valid signature on a new message or a new valid signature on a message that was already signed. The scheme is described in Figure 7, and its security game is given in Figure 8. Our signing algorithm returns $\perp$ when $e = -x$; we account for this event separately.

In Theorem 5, oracle responses have discrete logarithms represented by polynomials in the hidden variables. For BBS, the discrete logarithm of the i-th signing response is

$$a_i(\boldsymbol{X}) = \frac{G_1 + \sum_{j=1}^{\ell} m_{i,j} H_j}{X + e_i},$$

where X represents the signing secret and $G_1, H_1, \ldots, H_\ell$ represent the discrete logarithms of the public bases. This is a rational function, not a polynomial. We therefore explain how to deal with these responses in the four cases of the lifting proof.

We show below that BBS satisfies the lifting condition: $V^t \equiv 0$ implies $\boldsymbol{\delta} = \mathbf{0}$. Thus, in cases A and X.1, the forged signature is explained using only public group elements and signing-oracle responses. The AGM reduction sends signing queries obtained from the AGMOS adversary to its own signing oracle and submits to the AGM game the same forgery the AGMOS adversary output. AGM security therefore bounds the probability of these two cases.

In case X.2, we reduce directly to the $(\mathsf{ql}_s, 1)$-PDL assumption from Definition 2. We use the signing-oracle simulation from [TZ23], which answers signing queries using challenge powers without knowing the signing secret. Their reduction uses a random-base version of our PDL assumption. To obtain this challenge from our $(\mathsf{ql}_s, 1)$-PDL challenge, we sample $b_1, b_2 \leftarrow \mathbb{F}^*$ independently and compute

$$([b_1 x^j]_1)_{j=0}^{\mathsf{ql}_s}, \qquad [b_2]_2, \qquad [b_2 x]_2.$$

These elements have the distribution required by the simulation in [TZ23], and recovering the exponent x solves our original PDL challenge.

In case X.2, we have $V^t \not\equiv 0$ but $V^t(\boldsymbol{x}, \mathbb{Q}) \equiv 0$. We analyze executions in which $e_i \neq -x$ for all $i \in [1, \mathsf{ql}_s]$. For BBS, case X.2 implies $x + e^* = 0$. The reduction therefore returns $-e^*$ as its $(\mathsf{ql}_s, 1)$-PDL solution. The excluded event is that $e_i = -x$ for at least one of the ql_s signing scalars. Since these scalars are sampled uniformly from $\mathbb{F}$, this event has probability at most ql_s/p, which we add to the security bound.

In case Q, the TOFR reduction samples the trapdoors. It can therefore evaluate the rational signing responses

$$a_i(\boldsymbol{x}) = \frac{g_1 + \sum_{j=1}^{\ell} m_{i,j} h_j}{x + e_i},$$

directly whenever their denominators $x + e_i$ are non-zero. The reduction then constructs the relation among obliviously sampled elements as in the case Q of the proof of Lemma 4. No random oracle is used, so case X.3 does not arise.

Theorem 7 (SUF-CMA **security of BBS in the AGMOS).** *Let Γ_{BBS} be the* SUF-CMA *game of Figure 8 for the BBS scheme of Figure 7 in the $(\mathcal{EF}, \mathcal{DF})$-AGMOS, let $\Gamma_{\mathsf{BBS}}^{\mathrm{AGM}}$ be the same game in the AGM, and let ql_s bound the number of signing queries. For every PPT $(\mathcal{EF}, \mathcal{DF})$-AGMOS adversary $\mathcal{A}$ there are PPT adversaries $\mathcal{B}$, $\mathcal{B}_{\mathsf{pdl}}$, and $\mathcal{B}_{\mathsf{tofr}}$ such that*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\Gamma_{\mathsf{BBS}}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{\Gamma_{\mathsf{BBS}}^{\mathrm{AGM}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{pdl}}}^{\mathsf{PDL},(\mathsf{ql}_s,1)}(\lambda) \\ + \mathsf{Adv}_{\mathcal{B}_{\mathsf{tofr}}}^{\mathsf{TOFR}}(\lambda) + \frac{\mathsf{ql}_s}{p}.\end{aligned}$$

In particular, if the BBS scheme is SUF-CMA *secure in the AGM and the $(\mathsf{ql}_s, 1)$-PDL and $(\mathcal{EF}, \mathcal{DF})$-TOFR assumptions hold with respect to* PGen*, then it is* SUF-CMA *secure in the $(\mathcal{EF}, \mathcal{DF})$-AGMOS.*

Proof. We follow the case distinction in the proof of Theorem 5.

We may sample ql_s independent signing scalars from $\mathbb{F}$ in advance and use them, in order, for the valid signing queries. Let Bad_e be the event that any of these scalars equals $-x$, including unused scalars. Then

$$\Pr[\mathsf{Bad}_e] \leq \frac{\mathsf{ql}_s}{p}.$$

We first consider executions conditioned on $\neg\mathsf{Bad}_e$. Let $s \le \mathsf{ql}_s$ be the number of signing queries made by the adversary. On query $m_i = (m_{i,1}, \ldots, m_{i,\ell})$, the oracle returns

$$\sigma_i = ([a_i]_1, e_i), \qquad [a_i]_1 = (x + e_i)^{-1} \left([g_1]_1 + \sum_{j=1}^{\ell} m_{i,j} [h_j]_1 \right).$$

The adversary returns a candidate $(m^*, \sigma^* = ([a^*]_1, e^*))$ with an algebraic explanation

$$[a^*]_1 = \gamma_g [1]_1 + \gamma_0 [g_1]_1 + \sum_{j=1}^{\ell} \gamma_j [h_j]_1 + \sum_{i=1}^{s} \mu_i [a_i]_1 + \sum_{k=1}^{\mathsf{ql}_1} \delta_k [\mathbb{q}_{1,k}]_1,$$

where $[\mathbb{q}_{1,k}]_1$ are the oblivious sampling responses in $\mathbb{G}_1$.

The structural check is

$$\mathcal{S} = 1 \quad \Longleftrightarrow \quad (m^*, ([a^*]_1, e^*)) \notin \{(m_i, ([a_i]_1, e_i)) : i \in [1, s]\}.$$

The pairing check is

$$[a^*]_1 \bullet (e^* [g_2]_2 + p k_2) = \left([g_1]_1 + \sum_{j=1}^{\ell} m_j^* [h_j]_1 \right) \bullet [g_2]_2.$$

Let

$$\boldsymbol{X} = (X, G_1, G_2, H_1, \ldots, H_\ell)$$

be the indeterminates corresponding to the discrete logarithms of the public bases. For a fixed execution, set

$$a_i(\boldsymbol{X}) = \frac{G_1 + \sum_{j=1}^{\ell} m_{i,j} H_j}{X + e_i}.$$

The verification function is

$$V(\boldsymbol{X}, \mathbb{Q}) = G_2 \Bigg((X + e^*) \Bigg(\gamma_g + \gamma_0 G_1 + \sum_{j=1}^{\ell} \gamma_j H_j + \sum_{i=1}^{s} \mu_i a_i(\boldsymbol{X}) + \sum_{k=1}^{\mathsf{ql}_1} \delta_k \mathbb{Q}_{1,k} \Bigg) - \left(G_1 + \sum_{j=1}^{\ell} m_j^* H_j \right) \Bigg).$$

The messages, scalars e_i, final scalar e^*, and explanation coefficients are treated as coefficients. The pairing check holds precisely when $V(\boldsymbol{x}, \mathbb{q}) = 0$, where $\boldsymbol{x} = (x, g_1, g_2, h_1, \ldots, h_\ell)$. The V^t polynomial is

$$V^t(\boldsymbol{X}, \mathbb{Q}) = G_2 (X + e^*) \sum_{k=1}^{\mathsf{ql}_1} \delta_k \mathbb{Q}_{1,k}.$$

Define

$$D(X) = \prod_{i=1}^{s} (X + e_i), \qquad \overline{V}(\boldsymbol{X}, \mathbb{Q}) = D(X) V(\boldsymbol{X}, \mathbb{Q}),$$

with an empty product equal to 1. Conditioned on $\neg\mathsf{Bad}_e$, we have $D(x) \neq 0$. Thus $\overline{V}$ is a polynomial and

$$V(\boldsymbol{x}, \mathbb{q}) = 0 \quad \Longleftrightarrow \quad \overline{V}(\boldsymbol{x}, \mathbb{q}) = 0.$$

Then

$$\overline{V}^t(\boldsymbol{X}, \mathbb{Q}) = G_2 D(X) (X + e^*) \sum_{k=1}^{\mathsf{ql}_1} \delta_k \mathbb{Q}_{1,k}.$$

Since $G_2 D(X)(X + e^*)$ is a non-zero polynomial,

$$\overline{V}^t \equiv 0 \quad \text{implies} \quad \boldsymbol{\delta} = \mathbf{0}.$$

This proves the lifting condition. When $\boldsymbol{\delta} = \mathbf{0}$, the AGM projection preserves the signing queries, responses, and final candidate, so it also preserves freshness.

We consider the following four cases:

$$\begin{aligned}
&\mathsf{A}: && \overline{V}(\boldsymbol{X}, \mathbb{Q}) \equiv 0, \\
&\mathsf{X.1}: && \overline{V}(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0, && \overline{V}^t(\boldsymbol{X}, \mathbb{Q}) \equiv 0, \\
&\mathsf{X.2}: && \overline{V}^t(\boldsymbol{X}, \mathbb{Q}) \not\equiv 0, && \overline{V}^t(\boldsymbol{x}, \mathbb{Q}) \equiv 0, \\
&\mathsf{Q}: && \overline{V}^t(\boldsymbol{x}, \mathbb{Q}) \not\equiv 0.
\end{aligned}$$

In cases A and $\mathsf{X.1}$, we have $\boldsymbol{\delta} = \mathbf{0}$. The AGM reduction forwards the AGMOS adversary's signing queries to its own signing oracle and simulates oblivious sampling locally. It submits the same forgery, with an explanation using only the public input and signing-oracle responses. The pairing check and freshness condition are preserved, so AGM security bounds the combined probability of these two cases.

In case $\mathsf{X.2}$, we reuse the signing-oracle simulation of $\mathcal{B}_1$ in [TZ23]. We omit its abort on repeated signing scalars: the simulation remains valid with repeated signing scalars, and our extraction does not require distinct scalars. We abort on Bad_e and simulate oblivious sampling locally. Conditioned on $\neg\mathsf{Bad}_e$, the simulated public key and signing responses have the same joint distribution as in the real game conditioned on $\neg\mathsf{Bad}_e$. The secret key is the PDL discrete logarithm x. In case $\mathsf{X.2}$, we have $\boldsymbol{\delta} \neq \mathbf{0}$ and

$$\overline{V}^t(\boldsymbol{x}, \mathbb{Q}) = g_2 D(x)(x + e^*) \sum_k \delta_k \mathbb{Q}_{1,k} \equiv 0.$$

Outside Bad_e, we have $g_2 D(x) \neq 0$. Since $\boldsymbol{\delta} \neq \mathbf{0}$, the polynomial $\sum_k \delta_k \mathbb{Q}_{1,k}$ is non-zero. Thus $x + e^* = 0$, and, on a well-formed candidate, the reduction returns $-e^*$ as its PDL solution. Thus this case outside Bad_e is bounded by the $(\mathsf{ql}_s, 1)$-PDL advantage.

In case Q, the TOFR reduction samples the trapdoors itself, answers signing queries honestly, and forwards oblivious sampling queries to its oracle. It can evaluate the rational signing responses because, conditioned on $\neg\mathsf{Bad}_e$, their denominators are non-zero. The probability of this case outside Bad_e is bounded by the TOFR advantage.

Finally, letting Win denote successful forgery,

$$\Pr[\mathsf{Win}] \leq \sum_{C \in \{\mathsf{A},\mathsf{X.1},\mathsf{X.2},\mathsf{Q}\}} \Pr[\mathsf{Win} \wedge C \wedge \neg\mathsf{Bad}_e] + \frac{\mathsf{ql}_s}{p}.$$

Substituting the reduction bounds proves the claim. □

Remark 9 (Alternative treatment of the signing responses). The rational signing responses of BBS play the same role as the negative powers of the trapdoors in Polymath (Remark 7). In both cases, the discrete logarithms are not polynomials in the trapdoors, and multiplying by a factor that is non-zero at the trapdoors turns them into polynomials, namely by $(X_1 \cdots X_m)^{d^-}$ in Remark 7 and by $D(X)$ in the proof of Theorem 7. Instead of the direct reduction to the $(\mathsf{ql}_s, 1)$-PDL assumption, one could therefore modify the FPR assumption as in Remark 7, so that its oracle also accepts rational functions whose denominators divide $\prod_{i=1}^{\mathsf{ql}_s}(X + e_i)$, for signing scalars $e_1, \ldots, e_{\mathsf{ql}_s}$ fixed before the first oracle query. With this oracle, the signing oracle would be simulatable for the FPR reductions (Definition 16), and Theorem 5 would apply to BBS with the same changes as in Remark 7, with $D(X)$ in place of $(X_1 \cdots X_m)^{d^-}$. We use the direct reduction because it relies on the standard $(\mathsf{ql}_s, 1)$-PDL assumption instead of a further modification of the FPR assumption.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process. During manuscript preparation, the authors used ChatGPT (OpenAI) to assist with reviewing mathematical arguments, identifying potential gaps and inconsistencies, exploring proof strategies and possible repairs, drafting and revising portions of proofs, and improving exposition and LaTeX presentation. The authors also used Claude Sonnet 5.5 and Claude Opus 5.5 (Anthropic) for proofreading and for drafting and revising passages of the text, including expository paragraphs and remarks. AI-generated suggestions were critically assessed and, where appropriate, revised and incorporated by the authors. The authors independently checked the resulting arguments and take full responsibility for all statements, proofs, and conclusions in the final manuscript.

The manuscript reflects the authors' research and intellectual contributions. AI assistance supported the development and presentation of this work; it did not replace the authors' mathematical judgment or responsibility for verifying the results.

Acknowledgments. We are grateful to Marek Sefranek and the anonymous reviewers for helpful feedback on the presentation of our results. Juraj Belohorec and Pavel Hubáček were partially supported by the Academy of Sciences of the Czech Republic (RVO 67985840), Czech Science Foundation GAČR grant No. 25-16311S, and by Zircuit. Juraj Belohorec was supported by the grant SVV-2026-260822 and by the Charles University Grant Agency project No. 490226.

References

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, Lecture Notes in Computer Science, pages 41–55. Springer, 2004.

BCLS22. Jeff Burdges, Oana Ciobotaru, Syed Lavasani, and Alistair Stewart. Efficient aggregatable BLS signatures with chaum-pedersen proofs. *IACR Cryptol. ePrint Arch.*, 2022:1611, 2022.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Berlin, Heidelberg, December 2011.

BDFG20. Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Recursive zk-snarks from any additive polynomial commitment scheme. *IACR Cryptol. ePrint Arch.*, 2020:1536, 2020.

BDFG21. Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 649–680, Virtual Event, August 2021. Springer, Cham.

BDH+25. Juraj Belohorec, Pavel Dvořák, Charlotte Hoffmann, Pavel Hubáček, Kristýna Mašková, and Martin Pastyřík. On extractability of the KZG family of polynomial commitment schemes. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VI*, volume 16005 of *Lecture Notes in Computer Science*, pages 584–616. Springer, 2025.

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, Berlin, Heidelberg, August 2001.

BFHK24. Balthazar Bauer, Pooya Farshim, Patrick Harasser, and Markulf Kohlweiss. The uber-knowledge assumption: A bridge to the AGM. *IACR Commun. Cryptol.*, 1(3):31, 2024.

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 514–532, 2001.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

CFF+24. Matteo Campanelli, Antonio Faonio, Dario Fiore, Tianyu Li, and Helger Lipmaa. Lookup arguments: Improvements, extensions and applications to zero-knowledge decision trees. In *Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part II*, pages 337–369, 2024.

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, Cham, May 2020.

DMS24. Michel Dellepere, Pratyush Mishra, and Alireza Shirzad. Garuda and pari: Faster and smaller SNARKs via equifficient polynomial commitments. *IACR Cryptol. ePrint Arch.*, page 1245, 2024.

EG25. Liam Eagen and Ariel Gabizon. MERCURY: A multilinear polynomial commitment scheme with constant proof size and no prover ffts. *IACR Cryptol. ePrint Arch.*, 2025:385, 2025.

FFR24. Antonio Faonio, Dario Fiore, and Luigi Russo. Real-world universal zkSNARKs are non-malleable. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 3138–3151. ACM Press, October 2024.

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, Berlin, Heidelberg, December 2000.

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, Cham, August 2018.

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, Cham, May 2020.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discret. Appl. Math.*, 156(16):3113–3121, 2008.

GPS25. Chaya Ganesh, Sikhar Patranabis, and Nitin Singh. Samaritan: Linear-time prover SNARK from new multilinear polynomial commitments. In *Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part V*, pages 456–489, 2025.

GWC19. Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, page 953, 2019.

Ica09. Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 303–316. Springer, Berlin, Heidelberg, August 2009.

JM24. Joseph Jaeger and Deep Inder Mohan. Generic and algebraic computation models: When AGM proofs transfer to the GGM. In *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part V*, pages 14–45, 2024.

JR10. Tibor Jager and Andy Rupp. The semi-generic group model and applications to pairing-based cryptography. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010 – 16th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 539–556. Springer, 2010.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010.

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, Berlin, Heidelberg, March 2012.

Lip24. Helger Lipmaa. Polymath: Groth16 is not the limit. In *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*, pages 170–206, 2024.

LPS23. Helger Lipmaa, Roberto Parisella, and Janno Siim. Algebraic group model with oblivious sampling. In *Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part IV*, pages 363–392, 2023.

LPS24. Helger Lipmaa, Roberto Parisella, and Janno Siim. Constant-size zk-SNARKs in ROM from falsifiable assumptions. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 34–64. Springer, 2024.

LPS25. Helger Lipmaa, Roberto Parisella, and Janno Siim. On knowledge-soundness of Plonk in ROM from falsifiable assumptions. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August*

17-21, 2025, Proceedings, Part VII, volume 16006 of *Lecture Notes in Computer Science*, pages 362–395. Springer, 2025.

LXZ+24. Tianyi Liu, Tiancheng Xie, Jiaheng Zhang, Dawn Song, and Yupeng Zhang. Pianist: Scalable zkrollups via fully distributed zero-knowledge proofs. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 1777–1793, 2024.

LZL+25. Chongrong Li, Pengfei Zhu, Yun Li, Cheng Hong, Wenjie Qu, and Jiaheng Zhang. Hyperpianist: Pianist with linear-time prover and logarithmic communication cost. In *IEEE Symposium on Security and Privacy, SP 2025, San Francisco, CA, USA, May 12-15, 2025*, pages 3383–3401, 2025.

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography. In Nigel P. Smart, editor, *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.

MBKM19. Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2111–2128. ACM Press, November 2019.

NPR26. Anca Nitulescu, Nikitas Paslis, and Carla Ràfols. Flip-and-prove R1CS. *IACR Commun. Cryptol.*, 2(4):6, 2026.

PST13. Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of correct computation. In *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 222–242, 2013.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, pages 256–266, 1997.

SP26. Janno Siim and Pritam Pal. Note on extractability of the PST polynomial commitment scheme. *IACR Cryptol. ePrint Arch.*, 2026:2398, 2026.

TZ23. Stefano Tessaro and Chenzhi Zhu. Revisiting BBS signatures. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, Lecture Notes in Computer Science, pages 691–721. Springer, 2023.

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.

WZ24. Benedikt Wagner and Arantxa Zapico. A documentation of ethereum's peerdas. *IACR Cryptol. ePrint Arch.*, page 1362, 2024.

Zha22. Mark Zhandry. To label, or not to label (in generic groups). In *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, pages 66–96, 2022.

A EUF-CMA Security of the BLS Signature Scheme in the AGM with Type-3 Pairings

In this section, we revisit the security of the BLS signature scheme [BLS01]. Fuchsbauer, Kiltz, and Loss [FKL18] proved the existential unforgeability of BLS signatures in the AGM for Type-1 pairings. We adapt their argument to the setting of Type-3 pairings. We define all the necessary notions in Section 6.1.

Security of BLS signatures in the AGM. To prove the EUF-CMA security of the BLS signature scheme in the AGM for the setting of Type-3 pairings, we closely follow the AGM proof of Fuchsbauer et al. [FKL18], which is based on the DLOG assumption. However, due to the use of Type-3 pairings in our setting, the resulting reduction relies instead on the $(1,1)$-PDL assumption.

Theorem 8. *If the* $(1,1)$-PDL *assumption holds with respect to* PGen*, then the BLS signature scheme is* EUF-CMA *secure in the AGM, when instantiated with Type-3 pairings, with the hash function modeled as a random oracle.*

Proof. Let $\mathcal{A}$ be an AGM adversary playing the EUF-CMA game of the BLS signature scheme (depicted in Figure 6). For brevity, we use the previously defined notation G_λ for the EUF-CMA security game with adversary $\mathcal{A}$ at security parameter λ. We write $G_\lambda = 1$ if $\mathcal{A}$ wins the game and $G_\lambda = 0$ otherwise. As $\mathcal{A}$ is an AGM adversary, at the end of G_λ, it returns a forgery σ^* on a message $m^* \notin \mathcal{Q}$ together with an explanation vector $\mathbf{a} = (\hat{a}, \bar{a}_1, \ldots, \bar{a}_{\mathsf{q}_h}, \tilde{a}_1, \ldots, \tilde{a}_{\mathsf{q}_s})$ such that

$$\sigma^* = xH(m^*) = \hat{a}[1]_1 + \sum_{i=1}^{\mathsf{q}_h} \bar{a}_i H_i + \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i \sigma_i. \tag{14}$$

Here, the explanation vector $\mathbf{a}$ is split (from left to right) into powers of the generator $[1]_1$, all of the answers to hash queries $H_i = H(m_i)$, $i \in [1, \mathsf{q}_h]$, and the signatures σ_i, $i \in [1, \mathsf{q}_s]$, returned by the signing oracle. We assume without loss of generality that $\mathcal{A}$ queries H on every message before submitting it to the signing oracle, so that all hash values it receives are answers to explicit hash queries. In the following, we let r^* and r_i, for $i \in [1, \mathsf{q}_h]$, be such that $H(m^*) = [r^*]_1$ and $H(m_i) = [r_i]_1$. Equation (14) is thus equivalent to

$$xr^* = x \sum_{i=1}^{\mathsf{q}_s} r_i \tilde{a}_i + \left(\hat{a} + \sum_{i=1}^{\mathsf{q}_h} r_i \bar{a}_i \right).$$

Since $\mathcal{A}$ wins G_λ, the sum $\sum_{i=1}^{\mathsf{q}_s} r_i \tilde{a}_i$ contains no signing-oracle term corresponding to m^*.

Remark 10 (Indexing convention for random oracle values). In this proof, we slightly overload the notation for random oracle outputs. When r_i appears in a sum indexed by $i \in [1, \mathsf{q}_h]$, it denotes the scalar corresponding to the i-th explicit hash query. When r_i appears in a sum indexed by $i \in [1, \mathsf{q}_s]$, it denotes the scalar corresponding to the hash value of the message in the i-th signing query. These values are sampled from the same random oracle table, and hence are consistent: if the message of a signing query coincides with the message of an explicit hash query, then the two corresponding scalars are equal. Formally, one could distinguish these values as r_i^{H} and r_i^{S}, or introduce an index map from signing queries to the random oracle table. We suppress this additional notation because no expression below mixes the two meanings without the ambient summation index making clear which query log is meant.

We now describe the behavior of adversaries $\mathcal{C}$ and $\mathcal{D}$ (depicted in Figures 9 and 10, respectively) playing in the $(1,1)$-PDL game. Each of these adversaries simulates G_λ to $\mathcal{A}$ in a different way.

Lemma 5. *Let G_λ be as defined in Figure 6 and, with $\tilde{a}_i$, r^* and r_i defined as above, let F denote the event that $\sum_i r_i \tilde{a}_i - r^* \neq 0$. Then there exist adversaries $\mathcal{C}$ and $\mathcal{D}$ playing in the $(1,1)$-PDL game such that:*

$$\mathsf{Adv}^{\mathsf{PDL}}_{\mathsf{PGen},\mathcal{C}}(\lambda) \geq \Pr[G_\lambda = 1 \wedge F],$$

$$\mathsf{Adv}^{\mathsf{PDL}}_{\mathsf{PGen},\mathcal{D}}(\lambda) \geq \frac{p-1}{p} \Pr[G_\lambda = 1 \wedge \neg F].$$

Proof. Let $([z]_1, [z]_2)$ denote the $(1,1)$-PDL instance. $\mathcal{C}$ and $\mathcal{D}$ simulate the game G_λ to the adversary $\mathcal{A}$.

Adversary $\mathcal{C}([z]_1, [z]_2)$: Adversary $\mathcal{C}$ is depicted in Figure 9 and works as follows. It sets the BLS public key $[x]_2$ as the second coordinate of its $(1,1)$-PDL challenge $([z]_1, [z]_2)$, which implicitly sets $x := z$. To answer a query $H(m_i)$, $\mathcal{C}$ first checks whether $H(m_i) = \perp$. If so, $\mathcal{C}$ samples $r_i \leftarrow \mathbb{F}$ and sets $H(m_i) := [r_i]_1$. $\mathcal{C}$ then returns $H(m_i)$. To answer a signing query $O(m_i)$, $\mathcal{C}$ first checks whether $H(m_i)$ has previously been queried. If not, $\mathcal{C}$ first calls $H(m_i)$, which defines $H(m_i)$ (and thus r_i). $\mathcal{C}$ then computes and returns $\sigma_i := r_i[z]_1$. The simulation of $\mathcal{C}$ is perfect, since the hash values are uniform in $\mathbb{G}_1$ and $\sigma_i = r_i[z]_1 = zH(m_i)$ is the BLS signature on m_i under the public key $[z]_2$. Suppose that $\mathcal{A}$ wins G_λ and that F occurs, i.e., $\sum_i r_i \tilde{a}_i - r^* \not\equiv 0 \pmod{p}$. If r^* has not been defined at this point, $\mathcal{C}$ makes an additional query $H(m^*)$, which defines r^*. Now, we can look at the verification polynomial

$$V(Z) = Z \left(\sum_{i=1}^{\mathsf{q}_s} r_i \tilde{a}_i - r^* \right) + \left(\hat{a} + \sum_{i=1}^{\mathsf{q}_h} r_i \bar{a}_i \right).$$

We now consider the usual two cases that arise in AGM security proofs.

$\mathcal{C}([z]_1, [z]_2)$

1 : $([x]_1, [x]_2) = ([z]_1, [z]_2)$
2 : $(m^*, \sigma^*) \leftarrow \mathcal{A}^{O(\cdot), H(\cdot)}(\mathsf{gp}, [x]_2)$
3 : **compute** z
4 : **return** z

$O(m_i)$

1 : **if** $H(m_i) = \perp$
2 : $\quad r_i \leftarrow \mathbb{F}$
3 : $\quad H(m_i) \leftarrow [r_i]_1$
4 : $\sigma_i = r_i[x]_1$
5 : **return** $(H(m_i), \sigma_i)$

$H(m_i)$

1 : **if** $H(m_i) \neq \perp$
2 : $\quad$ **return** $H(m_i)$
3 : $r_i \leftarrow \mathbb{F}$
4 : $H(m_i) \leftarrow [r_i]_1$
5 : **return** $H(m_i)$

Fig. 9: Game G simulated by the adversary $\mathcal{C}$.

Case A: Suppose that $V(Z)$ is the identically zero polynomial, so every coefficient is equal to zero. This contradicts our assumption that $\sum_{i=1}^{\mathsf{q}_s} r_i \tilde{a}_i - r^* \neq 0$.

Case X: Assume that $V(Z) \neq 0$ but $V(z) = 0$. Since $\sum_i r_i \tilde{a}_i - r^* \neq 0$, the adversary $\mathcal{C}$ can compute

$$z = \left(\hat{a} + \sum_{i=1}^{\mathsf{q}_h} r_i \bar{a}_i\right)\left(r^* - \sum_{i=1}^{\mathsf{q}_s} r_i \tilde{a}_i\right)^{-1}.$$

Adversary $\mathcal{D}([z]_1, [z]_2)$: Adversary $\mathcal{D}$ is depicted in Figure 10 and works as follows. It samples its own secret key $x \leftarrow \mathbb{F}$, so the public key is $[x]_2$. To answer a query $H(m_i)$, $\mathcal{D}$ first checks whether $H(m_i) = \perp$. If so, it samples $\hat{r}_i \leftarrow \mathbb{F}$ and $b_i \leftarrow \mathbb{F}$ and sets $H(m_i) := b_i[z]_1 + [\hat{r}_i]_1$, which implicitly sets $r_i := \hat{r}_i + z b_i$. $\mathcal{D}$ then returns $H(m_i)$. To answer a signing query $O(m_i)$, $\mathcal{D}$ first checks whether $H(m_i)$ has previously been queried. If not, $\mathcal{D}$ first queries $H(m_i)$, which defines $H(m_i)$ together with the values $\hat{r}_i \in \mathbb{F}$, $b_i \in \mathbb{F}$, and $r_i = \hat{r}_i + z b_i$. $\mathcal{D}$ then computes and returns $\sigma_i := xH(m_i)$. The simulation of $\mathcal{D}$ is perfect, since $H(m_i) = [\hat{r}_i + z b_i]_1$ is uniform in $\mathbb{G}_1$ because $\hat{r}_i$ is uniform, and $\sigma_i = xH(m_i)$ is the BLS signature on m_i under the public key $[x]_2$. Suppose that $\mathcal{A}$ wins G_λ and that $\neg F$ occurs, i.e., $\sum_i r_i \tilde{a}_i - r^* \equiv 0 \pmod{p}$. This means that $\mathcal{A}$ outputs a forgery (m^*, σ^*) together with an explanation vector

$$(\hat{a}, \bar{a}_1, \ldots, \bar{a}_{\mathsf{q}_h}, \tilde{a}_1, \ldots, \tilde{a}_{\mathsf{q}_s})$$

such that

$$xH(m^*) = \sigma^* = \hat{a}[1]_1 + \sum_{i=1}^{\mathsf{q}_h} \bar{a}_i H_i + \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i \sigma_i.$$

In our simulation we have $H(m_i) = [\hat{r}_i + z b_i]_1$ and $\sigma_i = xH(m_i)$, and similarly $H(m^*) = [\hat{r}^* + z b^*]_1$ and $\sigma^* = xH(m^*)$. Substituting these expressions and multiplying both sides by x^{-1} (which the reduction knows) gives

$$[\hat{r}^* + z b^*]_1 = x^{-1}\hat{a}[1]_1 + \sum_{i=1}^{\mathsf{q}_h} x^{-1}\bar{a}_i[\hat{r}_i + z b_i]_1 + \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i[\hat{r}_i + z b_i]_1.$$

Taking the corresponding scalar relation modulo p yields

$$z b^* + \hat{r}^* = x^{-1}\hat{a} + \sum_{i=1}^{\mathsf{q}_h} x^{-1}\bar{a}_i(\hat{r}_i + z b_i) + \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i(\hat{r}_i + z b_i).$$

Rearranging the terms gives

$$zb^* + \hat{r}^* = Az + B,$$

where

$$A := \sum_{i=1}^{\mathsf{q}_h} x^{-1}\bar{a}_i b_i + \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i b_i, \qquad B := x^{-1}\hat{a} + \sum_{i=1}^{\mathsf{q}_h} x^{-1}\bar{a}_i \hat{r}_i + \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i \hat{r}_i.$$

Now, let us consider the verification polynomial $V(Z)$:

$$V(Z) = Z(A - b^*) + (B - \hat{r}^*).$$

Again, we consider two cases.

Case A: In this case, we assume that $V(Z)$ is the identically zero polynomial, hence all of the coefficients are zero. In particular, $A - b^* = 0$. We distinguish two subcases, depending on whether m^* was previously queried to H. Observe that, for fixed z, we may equivalently sample $r_j, b_j \leftarrow \mathbb{F}$ independently and define $\hat{r}_j := r_j - zb_j$. The oracle responses depend on r_j and x, but not on b_j. Thus, once the group parameters, z, x, the adversary's coins, and all random oracle values are fixed, the entire interaction, including m^* and the explanation coefficients, is determined, while the masks remain unrestricted. We further fix the masks corresponding to all messages other than m^*, leaving b^* uniform over $\mathbb{F}$.

If m^* was not previously queried to H: Since the forgery message was also never signed, its hash value is first defined during final verification. The mask b^* is therefore fresh and does not occur in A. Thus

$$\Pr[b^* - A = 0] = \frac{1}{p}.$$

Consequently, Case A occurs with probability at most $1/p$ in this branch.

If m^* was previously queried to H: Let i^* be the unique hash index corresponding to m^*, so that $b_{i^*} = b^*$, and write

$$c := x^{-1}\bar{a}_{i^*}, \qquad A = cb^* + A_0.$$

Freshness ensures that no signing term contains b^*, so A_0 depends only on masks associated with messages different from m^*. If $c \neq 1$, then

$$b^* - A = (1 - c)b^* - A_0$$

is uniform over $\mathbb{F}$. Hence Case A again occurs with probability at most $1/p$.
If $c = 1$, we cannot conclude that Case A is unlikely. Instead, $\mathcal{D}$ uses $\neg F$, which gives

$$r^* = \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i r_i.$$

Define

$$A_{\mathsf{S}} := \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i b_i, \qquad B_{\mathsf{S}} := \sum_{i=1}^{\mathsf{q}_s} \tilde{a}_i \hat{r}_i.$$

Substitution yields

$$z(b^* - A_{\mathsf{S}}) = B_{\mathsf{S}} - \hat{r}^*.$$

Since m^* was never signed, A_{S} contains no occurrence of b^*. Thus $b^* - A_{\mathsf{S}}$ is non-zero with probability $1 - 1/p$, and whenever this holds, $\mathcal{D}$ computes

$$z = (B_{\mathsf{S}} - \hat{r}^*)(b^* - A_{\mathsf{S}})^{-1}.$$

This extraction works even when $V(Z)$ is identically zero.

$\mathcal{D}([z]_1, [z]_2)$

1 : $x \leftarrow \mathbb{F}$
2 : $pk := [x]_2$
3 : $(m^*, \sigma^*) \leftarrow \mathcal{A}^{O(\cdot), H(\cdot)}(\mathsf{gp}, pk)$
4 : **compute** z
5 : **return** z

$O(m_i)$

1 : **if** $H(m_i)$ is undefined **then** $H(m_i)$
2 : $\sigma_i := xH(m_i)$
3 : **return** $(H(m_i), \sigma_i)$

$H(m_i)$

1 : **if** $H(m_i)$ is defined **then**
2 : **return** $H(m_i)$
3 : $b_i, \hat{r}_i \leftarrow \mathbb{F}$
4 : $H(m_i) \leftarrow b_i[z]_1 + [\hat{r}_i]_1$
5 : **return** $H(m_i)$

Fig. 10: Game G simulated by the adversary $\mathcal{D}$.

Case X: This occurs when $V(Z) \neq 0$, but $V(z) = 0$, which means that the adversary $\mathcal{D}$ can compute $(B - \hat{r}^*)(b^* - A)^{-1} \bmod p$, which is equal to z. As we have mentioned, the probability that $b^* - A \equiv 0 \pmod{p}$ is negligible, so $\mathcal{D}$ can compute z with overwhelming probability.

□

Now, to finish the proof of Theorem 8, let an adversary $\mathcal{B}$ choose to emulate one of the adversaries $\mathcal{C}$ or $\mathcal{D}$ from Lemma 5 with probability $\frac{1}{2}$ each. Then

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{PDL}}_{\mathsf{PGen},\mathcal{B}}(\lambda) &= \frac{1}{2}\mathsf{Adv}^{\mathsf{PDL}}_{\mathsf{PGen},\mathcal{C}}(\lambda) + \frac{1}{2}\mathsf{Adv}^{\mathsf{PDL}}_{\mathsf{PGen},\mathcal{D}}(\lambda) \\
&\geq \frac{1}{2}\Pr[G_\lambda = 1 \wedge F] + \frac{p-1}{2p}\Pr[G_\lambda = 1 \wedge \neg F] \\
&\geq \frac{1}{4}\Pr[G_\lambda = 1 \wedge F] + \frac{1}{4}\Pr[G_\lambda = 1 \wedge \neg F] \\
&= \frac{1}{4}\Pr[G_\lambda = 1] \\
&= \frac{1}{4}\mathsf{Adv}_{\mathcal{A},G_\lambda}(\lambda).
\end{aligned}$$

The analysis above concludes the proof. □

B Interactive Security Game Polynomials

In this section, we give the construction of the polynomials and matrices used in the interactive AGMOS framework introduced in Section 5.1. Let

$$\boldsymbol{R} = (R_1, \ldots, R_{\mathsf{ql}_h}),$$

where $\mathsf{ql}_h \in \mathsf{poly}(\lambda)$, be indeterminates corresponding to the discrete logarithms of all registered random oracle outputs. This includes hash values obtained by direct random oracle queries, hash values registered internally by $\mathcal{O}_{\boldsymbol{x}}$, and hash values registered for the final verification step. Let

$$\boldsymbol{T} = (T_1, \ldots, T_{\mathsf{ql}_t})$$

be indeterminates corresponding to the discrete logarithms of all group elements returned by the oracle $\mathcal{O}_{\boldsymbol{x}}$.

We require that each such group-valued oracle response is computed as a polynomial function of the trapdoor vector $\boldsymbol{x}$ and the registered hash outputs available when the response is generated. Thus $\boldsymbol{T} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{R})$. Furthermore, let

$$\mathbb{Q} = (\mathbb{Q}_1, \mathbb{Q}_2)$$

denote the indeterminates corresponding to the discrete logarithms of the obliviously sampled group elements, where $\mathbb{Q}_1$ corresponds to the elements sampled in $\mathbb{G}_1$ and $\mathbb{Q}_2$ corresponds to the elements sampled in $\mathbb{G}_2$. Let also

$$\mathbb{Y} = (\mathbb{Y}_1, \mathbb{Y}_2)$$

denote the indeterminates corresponding to the discrete logarithms of the adversary outputs

$$\mathbb{y} = (\mathbb{y}_1, \mathbb{y}_2).$$

Using the algebraic explanations of the adversary outputs, we obtain

$$\mathbb{Y}_1 = \boldsymbol{\gamma}_1 \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \boldsymbol{R} \\ \boldsymbol{T} \end{pmatrix} + \boldsymbol{\delta}_1 \mathbb{Q}_1 = \boldsymbol{\gamma}_1 \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \boldsymbol{R} \\ \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{R}) \end{pmatrix} + \boldsymbol{\delta}_1 \mathbb{Q}_1,$$

and, analogously,

$$\mathbb{Y}_2 = \boldsymbol{\gamma}_2 \mathbb{x}_2(\boldsymbol{X}) + \boldsymbol{\delta}_2 \mathbb{Q}_2.$$

For convenience, define

$$\widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) = \begin{pmatrix} \mathbb{x}_1(\boldsymbol{X}) \\ \boldsymbol{R} \\ \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{R}) \end{pmatrix}.$$

Then the adversary output in $\mathbb{G}_1$ may be written compactly as

$$\mathbb{Y}_1 = \boldsymbol{\gamma}_1 \widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) + \boldsymbol{\delta}_1 \mathbb{Q}_1.$$

Now let

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \boldsymbol{M}_{13} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \boldsymbol{M}_{23} \\ \boldsymbol{M}_{31} & \boldsymbol{M}_{32} & \boldsymbol{M}_{33} \end{pmatrix},$$

where $\boldsymbol{M}_{11} \in \mathbb{F}^{\mathsf{il}_1 \times \mathsf{il}_2}$, $\boldsymbol{M}_{12} \in \mathbb{F}^{\mathsf{il}_1 \times \mathsf{ol}_2}$, $\boldsymbol{M}_{13} \in \mathbb{F}^{\mathsf{il}_1 \times a_2}$, etc. Let

$$\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2), \qquad \boldsymbol{\Delta} = (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2)$$

be matrix indeterminates corresponding to the explanation matrices

$$\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2), \qquad \boldsymbol{\delta} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2),$$

respectively.

Define the matrix $\boldsymbol{P}_1(\boldsymbol{\Gamma}, \boldsymbol{\Delta})$ as

$$\boldsymbol{P}_1(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \boldsymbol{I}_{\mathsf{il}_1} & \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{ql}_h} & \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{ql}_t} & \boldsymbol{0}_{\mathsf{il}_1 \times \mathsf{ql}_1} & \boldsymbol{0}_{\mathsf{il}_1 \times a_1} \\ \boldsymbol{\Gamma}_{1,\boldsymbol{x}} & \boldsymbol{\Gamma}_{1,\boldsymbol{r}} & \boldsymbol{\Gamma}_{1,\boldsymbol{t}} & \boldsymbol{\Delta}_1 & \boldsymbol{0}_{\mathsf{ol}_1 \times a_1} \\ \boldsymbol{0}_{a_1 \times \mathsf{il}_1} & \boldsymbol{0}_{a_1 \times \mathsf{ql}_h} & \boldsymbol{0}_{a_1 \times \mathsf{ql}_t} & \boldsymbol{0}_{a_1 \times \mathsf{ql}_1} & \boldsymbol{I}_{a_1} \end{pmatrix}.$$

The matrix $\boldsymbol{P}_1$ maps the extended algebraic basis available to the adversary to the elements appearing in the pairing check. The columns correspond respectively to:

$$(\mathbb{x}_1(\boldsymbol{x}),\ \boldsymbol{r},\ \boldsymbol{t},\ \mathbb{q}_1,\ \mathsf{aux}_1(\boldsymbol{x})).$$

The matrix $\boldsymbol{\Gamma}_1$ contains the explanation coefficients of the adversary outputs in $\mathbb{G}_1$ and is partitioned according to this basis as

$$\boldsymbol{\Gamma}_1 = \begin{pmatrix} \boldsymbol{\Gamma}_{1,\boldsymbol{x}} & \boldsymbol{\Gamma}_{1,\boldsymbol{r}} & \boldsymbol{\Gamma}_{1,\boldsymbol{t}} \end{pmatrix} \in \mathbb{F}^{\mathsf{ol}_1 \times (\mathsf{il}_1 + \mathsf{ql}_h + \mathsf{ql}_t)},$$

where:

– $\boldsymbol{\Gamma}_{1,\boldsymbol{x}} \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{il}_1}$ contains the coefficients corresponding to the challenger inputs,
– $\boldsymbol{\Gamma}_{1,\boldsymbol{r}} \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{ql}_h}$ contains the coefficients corresponding to the random oracle outputs,
– $\boldsymbol{\Gamma}_{1,\boldsymbol{t}} \in \mathbb{F}^{\mathsf{ol}_1 \times \mathsf{ql}_t}$ contains the coefficients corresponding to the "signing" oracle outputs.

Similarly, define

$$\boldsymbol{P}_2(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \boldsymbol{I}_{\mathsf{il}_2} & \boldsymbol{0}_{\mathsf{il}_2 \times \mathsf{ql}_2} & \boldsymbol{0}_{\mathsf{il}_2 \times a_2} \\ \boldsymbol{\Gamma}_2 & \boldsymbol{\Delta}_2 & \boldsymbol{0}_{\mathsf{ol}_2 \times a_2} \\ \boldsymbol{0}_{a_2 \times \mathsf{il}_2} & \boldsymbol{0}_{a_2 \times \mathsf{ql}_2} & \boldsymbol{I}_{a_2} \end{pmatrix},$$

which plays an analogous role for the group $\mathbb{G}_2$.

These matrices satisfy

$$\begin{pmatrix} \mathbb{x}_1(\boldsymbol{x}) \\ \mathbb{y}_1 \\ \mathsf{aux}_1(\boldsymbol{x}) \end{pmatrix} = \boldsymbol{P}_1(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \widetilde{\mathbb{x}}_1(\boldsymbol{x}, \boldsymbol{r}) \\ \mathbb{q}_1 \\ \mathsf{aux}_1(\boldsymbol{x}) \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbb{x}_2(\boldsymbol{x}) \\ \mathbb{y}_2 \\ \mathsf{aux}_2(\boldsymbol{x}) \end{pmatrix} = \boldsymbol{P}_2(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{x}) \\ \mathbb{q}_2 \\ \mathsf{aux}_2(\boldsymbol{x}) \end{pmatrix}.$$

Define

$$\boldsymbol{N}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \boldsymbol{P}_1^T(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) \boldsymbol{M} \boldsymbol{P}_2(\boldsymbol{\Gamma}, \boldsymbol{\Delta}).$$

Then Equation (12) may be written as

$$V(\boldsymbol{x}, \boldsymbol{r}, \mathbb{q}) = 0,$$

where

$$V(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = \begin{pmatrix} \widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) \\ \mathbb{Q}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{N}(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix}$$

is the *implicit verification polynomial*.

Explicitly, the matrix $\boldsymbol{N}(\boldsymbol{\Gamma}, \boldsymbol{\Delta})$ is

$$\left(\begin{array}{c|c|c} \begin{array}{c} \boldsymbol{M}_{11} + \boldsymbol{M}_{12}\boldsymbol{\Gamma}_2 + \\ \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T \boldsymbol{M}_{21} + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T \boldsymbol{M}_{22} \boldsymbol{\Gamma}_2 \end{array} & \boldsymbol{M}_{12}\boldsymbol{\Delta}_2 + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T \boldsymbol{M}_{22} \boldsymbol{\Delta}_2 & \boldsymbol{M}_{13} + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T \boldsymbol{M}_{23} \\ \hline \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T \boldsymbol{M}_{21} + \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T \boldsymbol{M}_{22} \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T \boldsymbol{M}_{22} \boldsymbol{\Delta}_2 & \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T \boldsymbol{M}_{23} \\ \hline \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T \boldsymbol{M}_{21} + \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T \boldsymbol{M}_{22} \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T \boldsymbol{M}_{22} \boldsymbol{\Delta}_2 & \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T \boldsymbol{M}_{23} \\ \hline \boldsymbol{\Delta}_1^T \boldsymbol{M}_{21} + \boldsymbol{\Delta}_1^T \boldsymbol{M}_{22} \boldsymbol{\Gamma}_2 & \boldsymbol{\Delta}_1^T \boldsymbol{M}_{22} \boldsymbol{\Delta}_2 & \boldsymbol{\Delta}_1^T \boldsymbol{M}_{23} \\ \hline \boldsymbol{M}_{31} + \boldsymbol{M}_{32} \boldsymbol{\Gamma}_2 & \boldsymbol{M}_{32} \boldsymbol{\Delta}_2 & \boldsymbol{M}_{33} \end{array}\right).$$

The V^h component of the verification polynomial can be expressed analogously to its formulation in Section 3.1. In particular, the corresponding matrix is independent of both $\boldsymbol{\Delta}_1$ and $\boldsymbol{\Delta}_2$.

More concretely, the polynomial $V^h(\boldsymbol{X}, \boldsymbol{R})$ can be written as

$$V^h(\boldsymbol{X}, \boldsymbol{R}) = \begin{pmatrix} \widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) \\ \mathbb{Q}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{f}_{\boldsymbol{M}}^h(\boldsymbol{\gamma}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix},$$

where $\boldsymbol{f}_{\boldsymbol{M}}^{h}(\boldsymbol{\gamma})$ denotes the matrix defined below, after we plug in the explanation matrix $\boldsymbol{\gamma}$ into $\boldsymbol{\Gamma}$.

$$\boldsymbol{f}_{\boldsymbol{M}}^{h}(\boldsymbol{\Gamma}) = \left(\begin{array}{c|c|c} \boldsymbol{M}_{11} + \boldsymbol{M}_{12}\boldsymbol{\Gamma}_2 + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T\boldsymbol{M}_{21} + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T\boldsymbol{M}_{22}\boldsymbol{\Gamma}_2 & \mathbf{0} & \boldsymbol{M}_{13} + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T\boldsymbol{M}_{23} \\ \hline \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T\boldsymbol{M}_{21} + \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T\boldsymbol{M}_{22}\boldsymbol{\Gamma}_2 & \mathbf{0} & \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T\boldsymbol{M}_{23} \\ \hline \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T\boldsymbol{M}_{21} + \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T\boldsymbol{M}_{22}\boldsymbol{\Gamma}_2 & \mathbf{0} & \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T\boldsymbol{M}_{23} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \boldsymbol{M}_{31} + \boldsymbol{M}_{32}\boldsymbol{\Gamma}_2 & \mathbf{0} & \boldsymbol{M}_{33} \end{array}\right). \tag{15}$$

Observe that this coincides exactly with the matrix obtained in the AGM variant of the game. Indeed, in the AGM setting the adversary has no access to oblivious sampling, and therefore no $\boldsymbol{\Delta}$-terms appear. As a result, only the $\boldsymbol{\Gamma}$-dependent part remains, which is precisely captured by the matrix above. Hence, as in the non-interactive setting, the fact that V^h does not depend on any $\boldsymbol{\Delta}$ carries over to the interactive case.

Separating the $\mathbb{Q}$ terms. The $V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q})$ part of the implicit verification polynomial is

$$V^t(\boldsymbol{X}, \boldsymbol{R}, \mathbb{Q}) = \begin{pmatrix} \widetilde{\mathbb{x}}_1(\boldsymbol{X}, \boldsymbol{R}) \\ \mathbb{Q}_1 \\ \mathsf{aux}_1(\boldsymbol{X}) \end{pmatrix}^T \boldsymbol{f}_{\boldsymbol{M}}^{t}(\boldsymbol{\gamma}, \boldsymbol{\delta}) \begin{pmatrix} \mathbb{x}_2(\boldsymbol{X}) \\ \mathbb{Q}_2 \\ \mathsf{aux}_2(\boldsymbol{X}) \end{pmatrix}, \tag{16}$$

where the matrix $\boldsymbol{f}_{\boldsymbol{M}}^{t}(\boldsymbol{\Gamma}, \boldsymbol{\Delta})$ has the same block structure as the matrix defined in Equation (5). In particular, the matrix $\boldsymbol{f}_{\boldsymbol{M}}^{t}(\boldsymbol{\gamma}, \boldsymbol{\delta})$ is the following.

$$\boldsymbol{f}_{\boldsymbol{M}}^{t}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \begin{pmatrix} \mathbf{0} & \boldsymbol{M}_{12}\boldsymbol{\Delta}_2 + \boldsymbol{\Gamma}_{1,\boldsymbol{x}}^T\boldsymbol{M}_{22}\boldsymbol{\Delta}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_{1,\boldsymbol{r}}^T\boldsymbol{M}_{22}\boldsymbol{\Delta}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_{1,\boldsymbol{t}}^T\boldsymbol{M}_{22}\boldsymbol{\Delta}_2 & \mathbf{0} \\ \boldsymbol{\Delta}_1^T\boldsymbol{M}_{21} + \boldsymbol{\Delta}_1^T\boldsymbol{M}_{22}\boldsymbol{\Gamma}_2 & \boldsymbol{\Delta}_1^T\boldsymbol{M}_{22}\boldsymbol{\Delta}_2 & \boldsymbol{\Delta}_1^T\boldsymbol{M}_{23} \\ \mathbf{0} & \boldsymbol{M}_{32}\boldsymbol{\Delta}_2 & \mathbf{0} \end{pmatrix}. \tag{17}$$

Note that $\boldsymbol{f}_{\boldsymbol{M}}^{h}(\boldsymbol{\Gamma}) + \boldsymbol{f}_{\boldsymbol{M}}^{t}(\boldsymbol{\Gamma}, \boldsymbol{\Delta}) = \boldsymbol{N}(\boldsymbol{\Gamma}, \boldsymbol{\Delta})$.