A preliminary version of this paper appears in the proceedings of the *20th Theory of Cryptography Conference – TCC 2022*, © IACR 2022. This is the full version.

Beyond Uber: Instantiating Generic Groups via PGGs

Balthazar Bauer[1] Pooya Farshim[2] Patrick Harasser[3(✉)] Adam O'Neill[4]

[1] IRIF, CNRS, France
balthazar.bauer@ens.fr

[2] IOHK and Durham University, UK
pooya.farshim@gmail.com

[3] Technische Universität Darmstadt, Germany
patrick.harasser@tu-darmstadt.de

[4] Manning College of Information and Computer Sciences, University of Massachusetts Amherst, USA
adamo@cs.umass.edu

November 6, 2022

Abstract. The generic-group model (GGM) has been very successful in making the analyses of many cryptographic assumptions and protocols tractable. It is, however, well known that the GGM is "uninstantiable," i.e., there are protocols secure in the GGM that are insecure when using any real-world group. This motivates the study of standard-model notions formalizing that a real-world group in some sense "looks generic."

We introduce a standard-model definition called *pseudo-generic group (PGG)*, where we require exponentiations with base an (initially) unknown group generator to result in random-looking group elements. In essence, our framework delicately lifts the influential notion of Universal Computational Extractors of Bellare, Hoang, and Keelveedhi (BHK, CRYPTO 2013) to a setting where the underlying ideal reference object is a generic group. The definition we obtain simultaneously generalizes the Uber assumption family, as group exponents no longer need to be polynomially induced. At the core of our definitional contribution is a new notion of *algebraic unpredictability*, which reinterprets the standard Schwartz–Zippel lemma as a restriction on sources. We prove the soundness of our definition in the GGM with auxiliary-input (AI-GGM).

Our remaining results focus on applications of PGGs. We first show that PGGs are indeed a generalization of Uber. We then present a number of applications in settings where exponents are not polynomially induced. In particular we prove that simple variants of ElGamal meet several advanced security goals previously achieved only by complex and inefficient schemes. We also show that PGGs imply UCEs for split sources, which in turn are sufficient in several applications. As corollaries of our AI-GGM feasibility, we obtain the security of all these applications in the presence of preprocessing attacks.

Some of our implications utilize a novel type of hash function, which we call *linear-dependence destroyers (LDDs)* and use to convert standard into algebraic unpredictability. We give an LDD for low-degree sources, and establish their plausibility for all sources by showing, via a compression argument, that random functions meet this definition.

Keywords. Generic-group model · Uber assumption · UCE · Deterministic PKE · KDM and RKA security.

Contents

1 Introduction

1.1 Background

Idealized models. A useful tool in cryptography are so-called idealized models of computation, which include the random-oracle, random-permutation, ideal-cipher, and generic-group models. In such models, all algorithms work relative to oracles that serve to implement some information-theoretically random reference object. Later, when a scheme defined in an idealized setting is used in practice, the oracles are heuristically instantiated by appropriate public, efficiently computable functions. On the one hand, idealized models are powerful because they limit the adversary's capabilities and make proofs tractable. On the other, they are subject to well-known uninstantiability results, which show the existence of (contrived) schemes that are secure in the idealized model, but provably insecure under any possible instantiation (see, e.g., [CGH04, Den02, Bla06, GKMZ16]). This indicates that idealized models are not sound in general, yet "natural" applications (with the oracles appropriately instantiated) have withstood years of scrutiny.

The generic-group model. In this work, we mainly focus on the generic-group model (GGM), where a generic group is an idealization of a finite cyclic group. It was first defined by Nechaev [Nec94] and later refined by Shoup [Sho97], who considered random encodings of group elements.[1] More specifically, for a cyclic group $(\mathsf{G}, \circ)$ of order p, Shoup's model considers a random injection $\tau\colon \mathbb{Z}_p \to \mathsf{S}$, where $\mathsf{S} \subseteq \{0,1\}^*$ with $|\mathsf{S}| \geq p$. All algorithms run on input p and encodings of application-specific group elements. To perform group operations, algorithms can query a τ oracle and an operation oracle op defined as $\mathsf{op}(h_1, h_2) \coloneqq \tau(\tau^{-1}(h_1) + \tau^{-1}(h_2))$ if $h_1, h_2 \in \mathsf{Rng}(\tau)$, and $\mathsf{op}(h_1, h_2) \coloneqq \perp$ otherwise.

Instantiating generic groups. In practice, a generic group is typically instantiated via an appropriate elliptic curve group. Indeed, for such groups no algorithms for solving discrete-logarithm-like problems more efficiently than the generic ones are known. Addressing the above-mentioned mismatch between idealized and instantiated schemes, we investigate what assumptions are being made when carrying out such an instantiation. Note that an indistinguishability-based approach formalizing the idea of "behaving like a generic group" would suffer from the same shortcomings known for random oracles [CGH04].

This line of work has been carried out with considerable success for the random oracle model (ROM), where the ideal reference object is a random function (see, e.g., [Can97, BHK13, Zha16]). For generic groups on the other hand, the most compelling formulation so far of what assumption is being made when instantiating them is given by the so-called Uber assumption [BBG05, Boy08]. At a high level, the Uber assumption speaks to the hardness of distinguishing the exponentiation $g^{T(\mathbf{s})}$ of a polynomial evaluation $T(\mathbf{s})$ from random, given exponentiations $g^{R_1(\mathbf{s})}, \ldots, g^{R_n(\mathbf{s})}$ of other polynomial evaluations. This condition holds in generic groups, and must therefore be satisfied by any concrete group that aims to "faithfully" instantiate them. However, the Uber assumption is far from the most general (standard-model) property that might hold in generic groups and thus should also be satisfied by their real-world counterparts.

Indeed, we observe that in a wide range of advanced cryptographic protocols and primitives (such as security under bad randomness, deterministic encryption, leakage resilience, and code obfuscation to name a few), inputs may not be uniformly distributed and polynomially related, but follow distributions that are, for example, only assumed to have high entropy. The Uber assumption can fall short of providing means to prove security of practical schemes in such settings. Accordingly, the main question we ask is:

Are there standard-model properties that generalize the Uber assumption and allow instantiating generic groups in a broad range of applications?

[1] An alternative formulation of the GGM is given by Maurer [Mau05]; we follow Shoup's presentation in this paper. Relations and comparisons between different flavors of the GGM are discussed in the recent work [Zha22].

We emphasize that our treatment is practice-oriented in that we aim for a notion that captures standard-model properties of groups that can be used to establish the standard-model security of *existing*, *practical* protocols in a variety of models. Further, the new definitions should combine, as far as possible, standard-model analyses with the ease of use offered by the GGM.

In order to develop the core ideas one step at a time, in this work we treat the case of simple (non-bilinear) groups[2] and focus on decisional problems. There are indeed multiple directions in which our work can be extended; we will briefly discuss some of these at the end of the Introduction.

1.2 Our Approach

Our approach is inspired by an existing framework that bridges the standard and idealized models of computation: Universal Computational Extractors (UCEs) of Bellare, Hoang, and Keelveedhi (BHK) [BHK13], a security notion for hash functions which, at a high level, requires indistinguishability from a random oracle under unpredictable inputs. Indeed, their motivation is in some sense conceptually analogous to ours. To that end, we seek to extend UCEs to structured ideal primitives, and call the resulting security notion in the case of cyclic groups *pseudo-generic groups* (PGGs). Before presenting PGGs, we give a brief overview of UCEs and refer to Section 2 for formal definitions.

Universal Computational Extractors. Let $\mathsf{H}\colon K \times D \to R$ be a keyed hash function. The UCE notion is defined via a game played by a source $\mathcal{S}$ and a distinguisher $\mathcal{D}$: Sample a challenge bit $b \twoheadleftarrow \{0,1\}$, a hash key $hk \twoheadleftarrow K$, and a RO $\rho\colon D \to R$. Then, $\mathcal{S}$ runs with access to an oracle Hash which, when queried on $x \in D$, returns $\mathsf{H}(hk, x)$ if $b = 1$, and $\rho(x)$ otherwise. Eventually, $\mathcal{S}$ outputs leakage L which is passed to $\mathcal{D}$, who is also given the hash key hk but no access to the hashing oracle. Distinguisher $\mathcal{D}$ must then guess b, and the requirement for H is that the advantage of every PPT $(\mathcal{S}, \mathcal{D})$ in this game is small.

Notice that for this definition to be meaningful, some restriction must be placed on $\mathcal{S}$ and $\mathcal{D}$: Without any additional requirement, $(\mathcal{S}, \mathcal{D})$ can win with overwhelming advantage by having $\mathcal{S}$ query Hash on any $x \in D$ with answer y, leak the pair (x, y) to $\mathcal{D}$, and then have $\mathcal{D}$ (who knows the hash key hk) check whether $y = \mathsf{H}(hk, x)$. To avoid such generic attacks, one requires $\mathcal{S}$ to be *unpredictable*, a notion formalized by asking that any predictor $\mathcal{P}$ have small advantage in the following game: Source $\mathcal{S}$ runs with access to a RO and produces leakage L. Then $\mathcal{P}$ runs on input L and wins if it can guess any of $\mathcal{S}$'s queries.

Our new notion: PGG. To port the UCE definition to the context of cyclic groups, our first idea is to let a random group generator g play the role of the hash key, and to use exponentiation with base g in place of the hash.[3] The PGG security game for a group $(\mathsf{G}, \circ)$ then follows the UCE framework: Sample a secret bit $b \twoheadleftarrow \{0,1\}$, a random generator $g \twoheadleftarrow \mathsf{G}$, and a generic-group encoding $\sigma\colon \mathbb{Z}_p \to \mathsf{G}$. Then, a source $\mathcal{S}$ interacts with an exponentiation oracle Exp which, on input $x \in \mathbb{Z}_p$, returns the real group element g^x if $b = 1$, and a generic element $\sigma(x)$ otherwise. The source can pass some leakage L to a distinguisher $\mathcal{D}$, who is also given the generator g but loses access to the oracle and has to guess b. As for UCEs, the requirement for G is that every such $(\mathcal{S}, \mathcal{D})$ has a small advantage in this game. Thus, the PGG notion captures the intuition that if an adversary does not know the random generator g of G, exponentiation with base g looks like it returns random elements from G.

As before, for this notion not to be void we must put restrictions on the queries that $\mathcal{S}$ is allowed to make. First, observe that $\mathcal{S}$ must be unpredictable, because without any such requirement $(\mathcal{S}, \mathcal{D})$ can mount the attack for UCEs sketched above. We argue now that due to the presence of a group structure on G that was missing in the UCE setting, further conditions are needed.

[2] Similar work on the algebraic-group model (discussed later) was first carried out in simple groups [FKL18] and later in bilinear ones [BFL20].

[3] Recently, Bartusek, Ma, and Zhandry (BMZ) [BMZ19] studied the "fixed-generator" and "random-generator" settings in group-based assumptions. We necessarily work in the latter since, as we shall see, otherwise attacks arise.

Algebraic unpredictability. An important question now is for what sources is PGG achievable in principle, meaning there are no "trivial" attacks. Recall that for UCEs the answer was unpredictable sources. In our context, unpredictability alone is not sufficient: Consider a source $\mathcal{S}$ that samples $x_1, x_2 \twoheadleftarrow \mathbb{Z}_p$, queries $h_i \leftarrow \text{Exp}(x_i)$, and computes $x_3 \leftarrow x_1 + x_2$ and $h'_3 \leftarrow h_1 \circ h_2$. It then queries $h_3 \leftarrow \text{Exp}(x_3)$ and passes the bit $(h_3 = h'_3)$ to $\mathcal{D}$, who simply returns it. The advantage of $(\mathcal{S}, \mathcal{D})$ in the PGG game is almost 1, even though $\mathcal{S}$ is unpredictable since x_1 and x_2 are random. The issue is that $\mathcal{S}$ can place unpredictable queries that satisfy a known linear relation and distinguish by checking if the corresponding relation holds for the oracle replies. Excluding this trivial/generic attack motivates a more refined notion of unpredictability which we call *algebraic unpredictability*. In the corresponding game, the source $\mathcal{S}$ runs with access to the ideal exponentiation oracle while querying $x_1, \ldots, x_q$, and produces leakage L. Predictor $\mathcal{P}$ runs on input L and must guess a linear combination of the queries, i.e., outputs $(\alpha_0, \alpha_1, \ldots, \alpha_q)$ not all zero and wins if $\sum_{i=1}^{q} \alpha_i x_i = \alpha_0$.

This condition excludes the attack above, since $\mathcal{P}$ can output $(0, 1, 1, -1)$ to win the game. One might try to modify the attack and let the source leak (x_3, h_3) to $\mathcal{D}$, who, given g, can compute g^{x_3} and compare it to h_3. But this also contradicts algebraic unpredictability, with a predictor returning $(x_3, 1, 1)$.

As we shall see in Section 3, due to the existence of obfuscation-based attacks (similar to those for UCEs [BFM14]), algebraic unpredictability must be *statistical* in nature; that is, we allow the algebraic predictors to run in unbounded time.

Parallel structure. It turns out that algebraic unpredictability by itself is not sufficient to rule out all generic attacks. Indeed, consider a source $\mathcal{S}$ that samples $x \twoheadleftarrow \mathbb{Z}_p$, queries $h_1 \leftarrow \text{Exp}(x)$ and $h_2 \leftarrow \text{Exp}(x^2)$, then computes $h'_2 \leftarrow h_1^x$ and passes the bit $(h_2 = h'_2)$ as leakage to $\mathcal{D}$, who decides accordingly. Again, the advantage of $(\mathcal{S}, \mathcal{D})$ in the PGG game is almost 1, and now $\mathcal{S}$ is even algebraically unpredictable. The issue here is that $\mathcal{S}$'s queries satisfy a linear relation with coefficients that are themselves *unpredictable but known to* $\mathcal{S}$ (in this case, $x \cdot x - 1 \cdot x^2 = 0$), an attack vector not ruled out by algebraic unpredictability.

To address this problem, we consider *parallel sources*. Loosely speaking, this means that $\mathcal{S}$'s Exp queries are made in parallel by single-query sources $\mathcal{S}_i(\mathsf{st})$ which, other than receiving a common initial state, do not pass state among each other. Indeed, the attack above was possible because $\mathcal{S}$ could learn more than one oracle reply. Note that, in this example, although the queries x and x^2 are allowed, the equality check $h_2 = h'_2$ requires knowledge of h_2 and h'_2 (related to different queries), and hence violates the definition of a parallel source.

Restricted post-processing. Surprisingly, even considering only parallel sources does not rule out all trivial attacks. Indeed, one can modify the source $\mathcal{S}$ from above and make it parallel by setting $\mathsf{st} \leftarrow x$, having $\mathcal{S}_1(\mathsf{st})$ compute h_1 and h'_2, and letting $\mathcal{S}_2(\mathsf{st})$ compute h_2. Leakage (h'_2, h_2) is passed to $\mathcal{D}$, which returns the bit $(h_2 = h'_2)$. This attack works because each $\mathcal{S}_i(\mathsf{st})$ still allows arbitrary post-processing of its oracle response (here, computing the exponentiation of h_1). Accordingly, we further restrict the class of sources and consider algebraically unpredictable *masking* sources, which are parallel sources where each $\mathcal{S}_i(\mathsf{st})$ is allowed only structured post-processing of its oracle replies (e.g., no post-processing or at most one group operation).

A simplification. The nature of the Exp oracle allows us to both strengthen and simplify our notion: We consider a definition of PGG whereby the distinguisher no longer receives the random generator g, and accordingly modify algebraic unpredictability to hold with $\alpha_0 = 0$ only. At a high level, this new version implies the old one, as g can be obtained by querying 1. Second, algebraic unpredictability holds for this source, as the non-simplified version allows for non-zero α_0. We discuss this simplification more thoroughly in Section 3.2.

Generalizing Uber. We give a formal definition of the resulting notion in Section 3. Note that our notion can indeed be seen as a generalization of Uber, whereby the exponents are no longer evaluations

of polynomials but may come from arbitrarily correlated distributions, as long as they adhere to the requirements set above. In particular, linear independence between polynomially induced exponents is now generalized to algebraically unpredictable sources.

GGM FEASIBILITY. Analogously to BHK who showed a RO is a UCE, we show the soundness of our definition by proving that a generic group is PGG for algebraically unpredictable masking sources. (We adopt Shoup's model for generic groups here [Sho96].) This turns out to be significantly more involved than in BHK. Typically, GGM proofs appeal to the Schwartz–Zippel lemma to carry out a lazy sampling of group elements. In our proof, we no longer use this lemma and instead rely on the *algebraic unpredictability* of sources to carry out a consistent lazy sampling. Here we use a weaker notion of *computational* algebraic unpredictability. (There is no contradiction with obfuscation-based attacks, as generic groups do not have compact representations.) A second feature of our proof is that we allow our sources to depend on the entire function table of the group encoding. This choice more accurately models computationally unbounded sources in the standard model, widens the applicability of PGGs, and due to the existence of arbitrary leakage from source, also captures the effects of preprocessing attacks (aka. auxiliary information) on the definition. We use the recent technique of decomposition of high-entropy distribution due to Coretti, Dodis, and Guo [CDG18] to handle unbounded sources.

Our GGM feasibility result, beside showing that PGGs do not suffer from structural weaknesses exposed by generic attacks, places PGGs below the GGM in the hierarchy of assumptions on groups (cf. the so-called "layered approach" to security explained in [BHK13])[4]. Indeed, using this result, one can establish security of an application in the GGM by first proving it secure under an appropriate PGG assumption (in the standard model), and then lifting the result to the GGM using the result above.

Finally, equipped with GGM feasibility, it is reasonable to conjecture that appropriate elliptic curve groups are indeed PGGs, thus allowing the framework to be applied to a variety of practical cryptosystems built using such groups.

AVOIDING UNINSTANTIABILITY. We note that PGGs circumvent a variety of uninstantiability techniques. Notably, the classical CGH-type uninstantiability results [CGH98, Den02] are avoided due to the fact that the group elements are computed wrt. *high-entropy* exponents. Furthermore, attacks due to the existence of various forms of obfuscation are avoided by requiring that the algebraic unpredictability notion be *statistical*. An analogous approach has been used in works on UCEs to avoid uninstantiability [BFM14].

We also note that Zhandry's recent AGM uninstantiability result [Zha22] inherently relies on the fact that an algebraic adversary has to return a representation of the forged tag (which then either breaks DLP or compresses random strings). This does not carry over to PGGs because adversaries are not required to be algebraic in our setting.

1.3 Applications of Pseudo-Generic Groups

We demonstrate the applicability of our definition in three ways. First, we show that the Uber assumption holds in PGGs, thereby allowing us to recover all its applications within the PGG framework. For our second set of results, recall that there are several "advanced" security models for encryption, many of which have only been obtained via inefficient schemes. We demonstrate that PGGs enable proving simple variants of the classical *ElGamal encryption scheme* secure in a number of such advanced security models. According to the discussion above, this means that these notions can be safely assumed when ElGamal is implemented using suitable elliptic curve groups. Third, we show how to construct UCEs in PGGs. As before, this

[4]The idea is to have assumptions and models organized into a hierarchy, where higher levels justify lower ones and, conversely, proving a scheme secure at some level shows that it meets higher ones as well. This allows us to identify precisely how strong an assumption is needed for a given application. Moreover, proving security of a scheme at a lower level typically gives more insight into its inner workings.

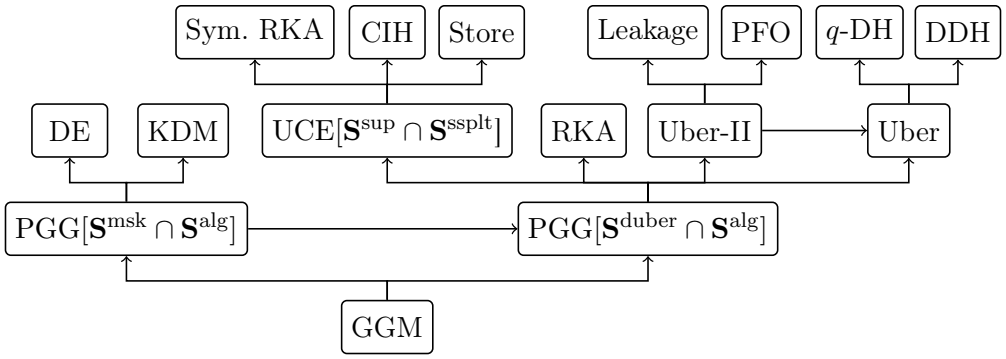

Figure 1 — Implications of PGGs. Here $\mathbf{S}^{\text{ssplt}}$ denotes the class of simple split sources (see Section 6.2), CIH stands for correlated input hashing, and store for storage auditing protocols (see [BHK13]). Results on DE (deterministic encryption), RKA and KDM or for ElGamal. Results for DE, RKA and UCE use LDDs if considering general sources.

allows us to recover all their applications within the PGG framework. We refer to Figure 1 for a schematic overview of our results.

PGGs generalize Uber. We prove that broad and even novel formulations of the Uber assumption hold in PGGs. The Uber assumption family [BBG05, Boy08] is an umbrella assumption that generalizes many hardness assumptions used to study the security of group-related schemes. Although it is commonly considered in bilinear groups, as previously mentioned, here we focus on simple groups. Nevertheless, proving that PGGs satisfy the Uber assumption allows us to recover all its applications within the PGG framework. For instance, all constructions whose security relies on the hardness of DDH (such as Diffie–Hellman key exchange, ElGamal encryption, and efficient PRFs [NR97]) or one of a number of closely related problems (e.g., q-DDHI, strong DDH, square DDH, and divisible DDH), or a randomized version of the recently introduced "Assumption 3" from [BMZ19], can be instantiated with PGGs. We further demonstrate that the Uber-II assumption, a variation of Uber with non-uniform exponents [Can97], also holds for PGGs. Specific instances of Uber-II have been used to build (composable) point-function obfuscation [CD08, BC10] and leakage-resilient PKE schemes [DHZ14].

We believe that PGGs better highlight the types of problems one expects to be hard in groups, as it places no restriction on how the exponents are sampled beyond the fact that certain trivial attacks are ruled out. In this sense, and also taking into account leakage and post-processing of exponents, PGGs go beyond the Uber assumption family.

Linear Dependence Destroyers. For some of our further implications below, we require a particular type of hash function with domain and range $\mathbb{Z}_p$ we call *linear dependence destroyer* (LDD). LDDs are defined via a game played by a source $\mathcal{S}$ and a predictor $\mathcal{A}$. Source $\mathcal{S}$ specifies a tuple of hash inputs $(x_1, \ldots, x_q)$ and state information st without seeing the random hash key hk, whereas $\mathcal{A}$ gets st and hk and returns a tuple of coefficients $(\alpha_0, \alpha_1, \ldots, \alpha_q)$. Adversary $(\mathcal{S}, \mathcal{A})$ wins if $\sum_{i=1}^{q} \alpha_i \cdot \mathsf{H}(hk, x_i) = \alpha_0$, and H is an LDD if every PPT $(\mathcal{S}, \mathcal{A})$ with $\mathcal{S}$ statistically unpredictable wins this game with negligible probability.

We show that the function $\mathsf{H}(hk, x) := 1/(x+hk)$ implicit in the work of Goyal, O'Neill, and Rao [GOR11] is an LDD when $\mathcal{S}$ is a *low-degree source*. These are sources that compute their outputs as evaluations of low-degree polynomials on points with sufficient entropy. This result, in turn, enables proofs of security for applications that use LDDs for low-degree sources. The main step in our proof is that different polynomials with random constant terms (given by the hash key) are likely to be coprime. When this is the case, the numerator of the fraction $\sum_{i=1}^{q} \alpha_i/(P_i(s_1, \ldots, s_m) + hk) - \alpha_0$ is non-zero no matter the choice of $(\alpha_0, \alpha_1, \ldots, \alpha_q)$. Winning the LDD game is thus equivalent to $(s_1, \ldots, s_m)$ being a root of this numerator, which is unlikely by the Schwartz–Zippel lemma.

Table 1 — Overview of applications of PGGs.

Application	PGG Source	Other Assumptions
Uber & Uber-II	dUber	–
RKA for ElGamal	dUber	LDD
KDM for ElGamal	Mask	–
Low-degree DE ElGamal	Mask	–
UCE for split sources	dUber	LDD
General DE for ElGamal	Mask	LDD

In fact, we conjecture that H is an LDD for *all* statistically unpredictable sources, not just for low-degree ones. To further lend plausibility to this notion, we also prove that a random function is an LDD, under mild restrictions on $\mathcal{S}$. To this end, we apply the compression technique originating from Gennaro and Trevisan [GT00] in a setting where *two* independent parties have full access to the code of the ideal object. The compression technique is commonly used in cryptography, and our extension may be of wider interest.

UCEs for split sources. A natural question is how PGGs relate to the notion of UCEs. It seems that PGGs are harder to build because they have more structure. In other words, generic groups, which PGGs instantiate, seem stronger than random oracles, which UCEs instantiate. As our first application we show that, indeed, UCEs can be constructed from PGGs for dUber sources and LDDs. The constructed UCE is for statistically unpredictable split sources. A number of applications of UCEs, such as proofs of storage, correlated-input secure hashing, and RKA security for symmetric encryption, only rely on UCE for split sources. We note that a benefit of building UCEs from PGGs is that the construction may enjoy useful algebraic properties that constructions from symmetric-key primitives do not. Once again, in the generic-group model, we show security against preprocessing attacks.

Key-dependent message security for ElGamal. Second, we show that PGGs enable proof of key-dependent message (KDM) security for a slightly tweaked version of ElGamal [BRS03, CL01]. KDM security for ElGamal does not seem to be feasible using Uber (though less efficient constructions do exist, e.g., [BHHO08, App11]). The KDM notion that we prove does not allow for adaptive queries, but it permits deriving key-dependent messages in an inefficient way. Furthermore, when combined with our GGM feasibility, we obtain KDM security against preprocessing attacks in the GGM.

Hash-then-ElGamal deterministic PKE. Moving on, we prove that ElGamal admits full instantiation of its corresponding random-oracle-model Encrypt-With-Hash (EwH) deterministic encryption scheme [BBO07], which replaces the coins in encryption with the hash of the message. Here we need that the hash function is an LDD. Preprocessing attacks are also accounted for in our definition and analysis. Note that a prior result of BHK [BHK13] also implies security of ElGamal-based EwH, but uses an assumption on the hash that makes the result arguably tautological. It is also known how to instantiate EwH for schemes meeting "lossiness" assumptions [HO13, BH15]. Our result is the first that does not require such an assumption as it shifts the security assumption with non-uniform inputs from the hash function to the underlying group.

Related-key security. We also show that ElGamal offers a form of related-key attack (RKA) security, whereby secret keys (and their corresponding public keys) are generated from related random coins. RKA security was systematically studied by Bellare, Cash, and Miller [BCM11] for PKEs. Under PGGs, and assuming LDDs (which for polynomially induced sources we show to exist) we can handle unpredictable related-key deriving functions that are claw-free (or more generally as long as the repetition pattern of secret keys does not affect unpredictability).

We summarize the above applications in Table 1. For each application, we record what type of source class is used in the reduction and whether the additional assumption of LDD is needed. For applications requiring LDDs we note that our results are modular wrt. the underlying source class. This means that for whatever source class we achieve LDDs, we also obtain an end application wrt. a corresponding source class. For example, low-degree LDDs (which we achieve unconditionally) translate to instantiations of UCEs and deterministic encryption wrt. low-degree sources and RKA security for low-degree related keys. The latter includes affine functions, which are often considered in the RKA literature.

We envision that several other security goals are also feasible under PGGs, of which we consider only a representative sample. Examples include security under bad randomness [BBN+09], joint RKA and KDM security [BDH14], randomness-dependent message security [BCPT13], related-randomness security [PSS14], and more generally application scenarios whereby the input distributions are not necessarily random and only guaranteed to come from high entropy distributions.

1.4 Other Related Work and Discussions

Public-seed pseudorandom permutations. Soni and Tessaro [ST17] define a UCE-like, standard-model notion for random (two-sided) permutations called public-seed pseudorandom permutations (psPRPs). They provide constructions of UCEs from psPRPs (for a variety of sources) by showing, for example, that the five-round Feistel [ST17] and the more efficient Naor–Reingold construction [ST18] yield psPRPs when the round functions are UCEs. Our work continues these lines of research by extending the UCE approach to defining security from random oracles and random permutations to generic groups.

Algebraic group model. An intriguing notion that has recently received considerable attention is the algebraic group model (AGM) [FKL18, BFL20]. We observe that the AGM places restrictions on adversaries that are qualitatively different compared to PGGs: Algebraic adversaries must output a representation of returned group elements, which makes the AGM a powerful and useful model since this additional information allows to carry out certain reductions.[5] Restrictions on PGG adversaries on the other hand are of standard-model type.

Nevertheless, it would be interesting to study the relation between the two notions and also to knowledge assumptions. Following work on instantiating UCEs [BM14] and on constructing groups in which the AGM can be realized and the Uber assumption holds [AHK20, KP19, AH18], another goal for future work is to construct PGGs from well-known assumptions (such as iO, dual-mode NIZKs, FHE, etc.).

Extensions of PGGs. In this work, we develop the necessary techniques and set the stage for the pseudo-generic approach to group-related assumptions. In doing so, we leave a number of directions for future research.

A natural extension to our work would be to formulate analogous PGG-type notions for bilinear groups (as considered by Boyen for the Uber assumption [Boy08] and extended via matrix DDH in [EHK+13]) or multi-linear groups. We anticipate further applications of this notion, as in the bilinear setting a host of schemes are only known to have a proof in the GGM and may be provable in PGGs.

The matrix DDH assumption (MDDH) [EHK+13] considers matrix-vector multiplication in the exponent in multi-linear groups, where a matrix is sampled from a general distribution and the vector is uniform. However, this assumption is only studied for polynomially induced distributions. As such, MDDH is not a generalization of Uber in the sense of PGG to arbitrary distributions.

Certain applications require assumptions that lie beyond the reach of PGGs as currently formulated. PGGs do not capture applications where exponents may depend on a group generator that is not random

[5] We note that our understanding of the role played by the AGM in assessing the hardness of group-related assumptions is evolving in light of recent works [KZZ22, Zha22].

(as, for example, in recent work on non-malleable point-function obfuscation [KY18, BMZ19, FF20]). The PGG framework also does not capture interactive [AH18] or knowledge-type [BP04] assumptions.

1.5 Structure of the Paper

In Section 2 we define the basic notation and recall the definition of UCEs. Section 3 contains our definitional contributions, where we define pseudo-generic groups, algebraically unpredictable and masking sources, and discuss the choices made in devising these notions. In Section 4 we prove that a generic group is a PGG, and then introduce LDDs and a candidate construction in Section 5. Section 6 contains the applications of PGGs. In Section 6.1 we show that an entropic variant of the decisional Uber assumption (and thus many implications thereof) holds in PGGs. Afterwards, we show how to apply PGG directly to the analysis of cryptosystems, by proving that PGGs and LDDs can be used to build UCEs for (simple) split sources. Further applications of PGGs are presented in Sections 6.3 to 6.5.

2 Preliminaries

Basic notation. If $n \in \mathbb{N}$, we write $[n]$ for the set $\{1, \ldots, n\}$. Unless otherwise stated, an integer $p \in \mathbb{Z}$ is assumed to be prime, and we let $\mathbb{Z}_p$ denote the field of integers modulo p. We denote the set of all bit strings of finite length by $\{0,1\}^*$, and the empty string by ε. We use boldface characters $\mathbf{x} := (x_1, \ldots, x_n)$ to denote vectors and write $\mathbf{x}[i]$, with $i \in [n]$, to denote the ith entry x_i of $\mathbf{x}$. By $x \twoheadleftarrow \mathcal{S}$ we mean sampling x according to distribution $\mathcal{S}$. Similarly, $x \twoheadleftarrow S$ means sampling x uniformly at random from a finite set S. The cardinality of a set S is denoted $|S|$. We let $L \leftarrow [\,]$ denote initializing an ordered list to empty, and $L : x$ denote appending an element x to the list L. A table T is a list of pairs (x, y), and we write $T[x] \leftarrow y$ to mean that the pair (x, y) is appended to T. We let $\mathsf{Dom}(T)$ denote the set of all values x such that $(x, y) \in T$ for some y, and similarly $\mathsf{Rng}(T)$ denote the set of all values y such that $(x, y) \in T$ for some x. For two sets D and R we denote by $\mathrm{Fun}(D, R)$ and $\mathrm{Inj}(D, R)$ the set of all functions and the set of all injections from D to R, respectively. When $|D| = |R|$, an injection is also a bijection.

Min-entropy. The min-entropy of a random variable $\mathcal{X}$ over a domain D is

$$\mathbf{H}_\infty(\mathcal{X}) := -\log \max_{x \in D} \Pr[\mathcal{X} = x]\,.$$

$\mathcal{X}$ is called a k-source if $\mathbf{H}_\infty(\mathcal{X}) \geq k$.

Polynomials and rational functions. We let $\mathbb{F}[X_1, \ldots, X_m]$ be the ring of polynomials in $m \in \mathbb{N}$ variables over a field $\mathbb{F}$, and $\mathbb{F}(X_1, \ldots, X_m)$ be the field of rational functions of the form $R(X_1, \ldots, X_m) = \hat{R}(X_1, \ldots, X_m)/\check{R}(X_1, \ldots, X_m)$, with $\hat{R}, \check{R} \in \mathbb{F}[X_1, \ldots, X_m]$ and $\check{R} \neq 0$. Here, as usual, $\hat{R}$ is called the numerator and $\check{R}$ the denominator of R. If $0 \neq R \in \mathbb{F}[X_1, \ldots, X_m]$ is a polynomial, we denote its total degree by $\deg(R)$. We extend this notation to rational functions via $\deg(R) := \deg(\hat{R}) - \deg(\check{R})$ for every $0 \neq R = \hat{R}/\check{R}$. Observe that the degree of a rational function is well-defined, since it does not depend on its representation as a fraction of polynomials. Finally, if $R_1, \ldots, R_n \in \mathbb{F}(X_1, \ldots, X_m)$ such that $R_i \neq 0$ for every $i \in [n]$, we let $\deg(R_1, \ldots, R_n) := \max_{i \in [n]} \deg(R_i)$.

Linear dependence. Let $R_1, \ldots, R_n, T \in \mathbb{F}(X_1, \ldots, X_m)$. We say that T is linearly dependent on $R_1, \ldots, R_n$ (over $\mathbb{F}$) if there exist $a_1, \ldots, a_n \in \mathbb{F}$ such that $T(X_1, \ldots, X_m) = \sum_{i=1}^{n} a_i \cdot R_i(X_1, \ldots, X_m)$.

Hash function families. A hash function family is a tuple of PPT algorithms $\mathsf{H} := (\mathsf{H.Setup}, \mathsf{H.KGen}, \mathsf{H.Eval})$. Here, algorithm $\mathsf{H.Setup}(1^\lambda)$ outputs a tuple π containing the descriptions of valid domain and range points D and R, as well as a key space K and other system-wide parameters. Algorithm $\mathsf{H.KGen}(\pi)$ is the hash key generation algorithm which returns a key $hk \in K$. The evaluation algorithm $\mathsf{H.Eval}(\pi, hk, x)$,

Game $\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$:	Proc. $\textsc{Hash}(x)$:	Game $\mathrm{Pred}^{\mathcal{P}}_{\mathsf{H},\mathcal{S}}(\lambda)$:	Proc. $\textsc{Hash}(x)$:
$b \twoheadleftarrow \{0,1\}$; $\pi \twoheadleftarrow \mathsf{H.Setup}(1^\lambda)$	if $(b = 0)$ then	$Q \leftarrow [\,]$; $\pi \twoheadleftarrow \mathsf{H.Setup}(1^\lambda)$	$Q \leftarrow Q : x$
$\rho \twoheadleftarrow \mathrm{Fun}(D, R)$	return $\rho(x)$	$\rho \twoheadleftarrow \mathrm{Fun}(D, R)$	return $\rho(x)$
$hk \twoheadleftarrow \mathsf{H.KGen}(\pi)$	return $\mathsf{H}(hk, x)$	$L \twoheadleftarrow \mathcal{S}^{\textsc{Hash}}(\pi)$	
$L \twoheadleftarrow \mathcal{S}^{\textsc{Hash}}(\pi)$; $b' \twoheadleftarrow \mathcal{D}(\pi, hk, L)$		$x \twoheadleftarrow \mathcal{P}(\pi, L)$	
return $(b = b')$		return $(x \in Q)$	

Figure 2 — *Left:* The UCE game. *Right:* The unpredictability game.

called on a hash key hk and a domain point $x \in D$, outputs a point $y \in R$. To help readability, by slight abuse of notation we will simply write $\mathsf{H}(hk, x)$ in place of $\mathsf{H.Eval}(\pi, hk, x)$.

Remark. Our definition of hash function families augments the usual syntax with a setup algorithm $\mathsf{H.Setup}$. Accordingly, we will extend the UCE definition to incorporate system parameters. Overloading notation, we allow $\mathsf{H.Setup}$ to alternatively take the description of a domain D and a range R as inputs, and let it return corresponding parameters π.

Universal computational extractors [BHK13]. Let H be a hash function family. The advantage of a pair of PPT adversaries $(\mathcal{S}, \mathcal{D})$ (called UCE source and UCE distinguisher) in the UCE game for H is defined as

$$\mathrm{Adv}^{\mathrm{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) := 2 \cdot \Pr\left[\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)\right] - 1\,,$$

where the UCE game is defined in Figure 2 (left). If $\mathbf{S}$ is a class of UCE sources, we say that H is UCE[$\mathbf{S}$] secure, if the advantage of any PPT $(\mathcal{S}, \mathcal{D})$ with $\mathcal{S} \in \mathbf{S}$ in the UCE game for H is negligible. This is usually written as $\mathsf{H} \in \mathrm{UCE}[\mathbf{S}]$.

Without any restriction on the class of sources $\mathbf{S}$, the UCE notion of security is unachievable [BHK13]. BHK exclude trivial attacks by requiring that the source be *unpredictable*, meaning that it is hard to predict any of its oracle queries when observing the leakage L. Due to the obfuscation-based attack of [BFM14], the flavor of unpredictability needs to be statistical. We recall the formal definition below.

(Statistically) unpredictable sources [BHK13, BFM14]. Let H be a hash function family and $\mathcal{S}$ a UCE source. We define the advantage of a (possibly unbounded) adversary $\mathcal{P}$ (called predictor) in the predictability game against $(\mathsf{H}, \mathcal{S})$ as

$$\mathrm{Adv}^{\mathrm{pred}}_{\mathsf{H},\mathcal{S},\mathcal{P}}(\lambda) := \Pr\left[\mathrm{Pred}^{\mathcal{P}}_{\mathsf{H},\mathcal{S}}(\lambda)\right],$$

where the game Pred is defined in Figure 2 (right). A UCE source $\mathcal{S}$ is called statistically unpredictable if the above advantage is negligible for any (possibly unbounded) predictor $\mathcal{P}$. The class of all statistically unpredictable sources is denoted $\mathbf{S}^{\mathrm{sup}}$. We say that H is UCE secure if it is UCE[$\mathbf{S}^{\mathrm{sup}}$] secure.

Schwartz–Zippel Lemma. We now recall the Schwartz–Zippel Lemma [Sch80, Zip79, DL78], a simple yet powerful tool to bound the probability of finding a root of a non-zero polynomial when evaluating it at a random point. We also generalize the standard Schwartz–Zippel lemma and obtain a more general and game-based version of this result. In this variant, the points can be chosen according to distributions with enough min-entropy, and the polynomial picked given some leakage. This version may be more suitable for use in a cryptographic setting. A proof of the game-based Schwartz–Zippel lemma is given in Appendix A.

Lemma 2.1 (Schwartz–Zippel)**.** *Let $\alpha, p \in \mathbb{N}$ with p prime, $S \subseteq \mathbb{F}_{p^\alpha}$, and let $0 \neq P \in \mathbb{F}_{p^\alpha}[X_1, \ldots, X_m]$. Then*

$$\Pr_{x_1,\ldots,x_m \twoheadleftarrow S}[P(x_1, \ldots, x_m) = 0] \leq \frac{\deg(P)}{|S|}\,.$$

Game $\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}$:	Source $\mathcal{X}_i$:	Source $\mathcal{Z}_i$:
$(\pi \coloneqq (p^\alpha, m, \mathsf{st})) \twoheadleftarrow \mathcal{A}_0$	$(\pi \coloneqq (p^\alpha, m, \mathsf{st})) \twoheadleftarrow \mathcal{A}_0$	$(\pi \coloneqq (p^\alpha, m, \mathsf{st})) \twoheadleftarrow \mathcal{A}_0$
for $i \in [m]$ do $(\mathbf{x}[i], \mathbf{z}[i]) \twoheadleftarrow \mathcal{S}_i(\pi)$	$(x, z) \twoheadleftarrow \mathcal{S}_i(\pi)$; return x	$(x, z) \twoheadleftarrow \mathcal{S}_i(\pi)$; return z
$(P, y) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{z})$; return $(y = P(\mathbf{x}))$		

Figure 3 — *Left:* The Schwartz–Zippel game. *Right:* The sources $\mathcal{X}_i$ and $\mathcal{Z}_i$.

Lemma 2.2 (Game-based Schwartz–Zippel)**.** *Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be a two-stage algorithm, where $\mathcal{A}_0$ takes no input and returns a set of public parameters $\pi \coloneqq (p^\alpha, m, \mathsf{st}) \in \mathbb{N}^2 \times \{0,1\}^*$ with $\alpha, p \in \mathbb{N}$ and p prime, and $\mathcal{A}_1$ takes π and values $z_1, \ldots, z_m \in \{0,1\}^*$ as input and returns a non-constant polynomial $P \in \mathbb{F}_{p^\alpha}[X_1, \ldots, X_m]$ with $\deg(P) \le d \in \mathbb{N}$ and a value $y \in \mathbb{F}_{p^\alpha}$. Let $\mathcal{S} \coloneqq \{\mathcal{S}_i \mid i \in \mathbb{N}\}$ be a family of sources, each taking π as input and returning values $(x, z) \in \mathbb{F}_{p^\alpha} \times \{0,1\}^*$. Then*

$$\Pr\left[\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}\right] \le d \cdot \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \mathcal{A}_0, \\ (x_1,z_1) \twoheadleftarrow \mathcal{S}_1(\pi), \ldots, (x_m,z_m) \twoheadleftarrow \mathcal{S}_m(\pi)}} \left[\frac{1}{2^{\min_{i \in [m]} \{\mathbf{H}_\infty(\mathcal{X}_i \mid (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i = z_i))\}}} \right],$$

where the game $\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}$ is defined in Figure 3 (left), and the sources $\mathcal{X}_i$ and $\mathcal{Z}_i$ are given in Figure 3 (right).

Observe that if $m = 1$, the expectation above is the prediction probability of $\mathcal{X}$ given $\mathcal{A}_0$ and $\mathcal{Z}$. In general, the minimum cannot be taken out of the expectation, because it reflects $\mathcal{A}$'s choices of which variables appear in P.

3 Pseudo-Generic Groups

We now formally define pseudo-generic groups (PGGs), where group elements are required to be indistinguishable from random, as long as their exponents satisfy a specific unpredictability condition. PGGs lift the definition of UCEs of Bellare, Hoang, and Keelveedhi [BHK13] from the setting of hash functions to that of groups. In other words, the underlying ideal object in the PGG definition, from which a concrete group is supposed to be indistinguishable, is a generic group rather than a random oracle. We start by giving some background on computational group schemes and generic groups, and then proceed to defining PGGs.

3.1 PGG Security

Computational group schemes [CS98]. A computational group scheme is a randomized algorithm Γ which, on input the security parameter 1^λ, outputs group parameters π consisting of a group operation $\circ$, an arbitrary group generator g, and a prime group order $p \in [2^{\lambda-1}, 2^\lambda)$. Implicit in these parameters is a set G such that $(\mathsf{G}, \circ)$ forms a cyclic group of order p with generator $g \in \mathsf{G}$. We write the sampling of group parameters as $(\pi \coloneqq (\circ, g, p)) \twoheadleftarrow \Gamma(1^\lambda)$, with the understanding that π implicitly defines the underlying set G. As usual, the group operation gives rise to an exponentiation algorithm $\mathsf{exp}(h, x)$ whose output is denoted as h^x. We will often omit explicitly writing the operation $\circ$.

Generic groups [Nec94, Sho97, Mau05]. Given a group $(\mathsf{G}, \circ)$ of prime order p, the generic group on G is the uniform distribution over $\mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$, where $\mathsf{S} \subseteq \{0,1\}^*$ with $|\mathsf{S}| \ge p$. Recall that every map $\tau \in \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$ allows one to define an associated group operation

$$\mathsf{op} \colon \mathsf{S} \times \mathsf{S} \to \mathsf{S} \cup \{\perp\}, \qquad (h_1, h_2) \mapsto \begin{cases} \tau\big(\tau^{-1}(h_1) + \tau^{-1}(h_2)\big) & \text{if } h_1, h_2 \in \mathsf{Rng}(\tau) \\ \perp & \text{else}. \end{cases}$$

Game $\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda)$:

$b \twoheadleftarrow \{0,1\}$
$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$r \twoheadleftarrow \mathbb{Z}_p^*$; $g \leftarrow g_0^r$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$L \twoheadleftarrow \mathcal{S}^{\mathrm{Exp}}(\pi)$
$b' \twoheadleftarrow \mathcal{D}(\pi, L)$
return $(b = b')$

Proc. $\mathrm{Exp}(x)$:

if $(b = 0)$ then return $\sigma(x)$
else return g^x

Game $\mathrm{AlgPred}_{\Gamma,\mathcal{S}}^{\mathcal{P}}(\lambda)$:

$Q \leftarrow [\,]$; $(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$L \twoheadleftarrow \mathcal{S}^{\mathrm{Exp}}(\pi)$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L)$
$[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^{q} \alpha_i x_i = 0\right)$

Proc. $\mathrm{Exp}(x)$:

$Q \leftarrow Q : x$
return $\sigma(x)$

Masking source $\mathcal{S}^{\mathrm{Exp}}(\pi)$:

$(\mathbf{x}, \mathbf{m}, L) \twoheadleftarrow \bar{\mathcal{S}}(\pi)$
for $i = 1$ to $|\mathbf{m}|$ do
  $\mathbf{y}[i] \leftarrow \mathbf{m}[i] \circ \mathrm{Exp}(\mathbf{x}[i])$
return $(\mathbf{y}, L)$

dUber source $\mathcal{S}^{\mathrm{Exp}}(\pi)$:

$(\mathbf{x}, L) \twoheadleftarrow \bar{\mathcal{S}}(\pi)$
for $i = 1$ to $|\mathbf{x}|$ do
  $\mathbf{y}[i] \leftarrow \mathrm{Exp}(\mathbf{x}[i])$
return $(\mathbf{y}, L)$

Figure 4 — *Left:* The PGG game. *Center:* The algebraic unpredictability game. *Top right:* A generic masking source. *Bottom right:* A generic dUber source.

The generic-group model is a model of computation in which all parties, honest or otherwise, are run on inputs p and encodings of application-specific elements, and have oracle access to a random encoding $\tau \in \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$ and its associated operation oracle op.

Pseudo-generic groups. Let Γ be a computational group scheme. We define the advantage of a pair of adversaries $(\mathcal{S}, \mathcal{D})$ (called PGG source and PGG distinguisher) in the PGG game for Γ as

$$\mathrm{Adv}_{\Gamma,\mathcal{S},\mathcal{D}}^{\mathrm{pgg}}(\lambda) \coloneqq 2 \cdot \Pr\left[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda)\right] - 1\,,$$

where the PGG game is defined in Figure 4 (left).[6] We require that there be polynomials q and q' such that, for every $\lambda \in \mathbb{N}$ and every $\pi \in \mathrm{supp}(\Gamma(1^\lambda))$, source $\mathcal{S}(\pi)$ makes at most $q(\lambda)$ oracle queries and outputs leakage of length at most $q'(\lambda)$. If $\mathbf{S}$ is a class of PGG sources, we say that Γ is PGG[$\mathbf{S}$] secure if the advantage of any $(\mathcal{S}, \mathcal{D})$ as above with $\mathcal{S} \in \mathbf{S}$ and $\mathcal{D}$ a PPT algorithm in the PGG game is negligible. We denote this as $\Gamma \in \mathrm{PGG}[\mathbf{S}]$.

Recall from our earlier discussion that, similarly to UCEs and psPRPs, this notion of security is not achievable without restrictions on the class of PGG sources $\mathbf{S}$. As a first step towards excluding trivial attacks, we introduce the notion of *algebraic unpredictability*, the core definition which allows us to extend UCE-type security notions beyond unstructured primitives (like hash functions and permutations). We require that no predictor be able to guess a non-trivial linear combination between the points queried by the source, as formalized below.

Algebraically unpredictable sources. Let Γ be a computational group scheme and $\mathcal{S}$ a PGG source, and assume that the leakage L produced by $\mathcal{S}$ encodes the number of Exp queries made by $\mathcal{S}$. We define the advantage of a (possibly unbounded) algorithm $\mathcal{P}$ (called predictor) in the algebraic unpredictability game against $(\Gamma, \mathcal{S})$ as

$$\mathrm{Adv}_{\Gamma,\mathcal{S},\mathcal{P}}^{\text{alg-pred}}(\lambda) \coloneqq \Pr\left[\mathrm{AlgPred}_{\Gamma,\mathcal{S}}^{\mathcal{P}}(\lambda)\right],$$

where the game AlgPred is defined in Figure 4 (center). We require that the output of $\mathcal{P}$ be different from the trivial all-zero tuple. A source $\mathcal{S}$ is called statistically algebraically unpredictable if the above advantage is negligible for any (possibly unbounded) predictor $\mathcal{P}$. We denote the class of all statistically algebraically unpredictable sources by $\mathbf{S}^{\mathrm{alg}}$. Observe that any such source must output distinct points (with high probability).

[6] Note that σ can be lazily sampled, so that the game runs in polynomial time.

MASKING AND DUBER SOURCES. Algebraic unpredictability turns out to be insufficient to rule out all trivial attacks, as explained in the Introduction. We thus restrict the set of sources for which we require PGG security even further and consider the class $\mathbf{S}^{\mathrm{msk}}$ of *masking* sources. These are sources $\mathcal{S}$ for which there exists a (possibly unbounded) auxiliary algorithm $\bar{\mathcal{S}}$ with polynomially bounded output, such that $\mathcal{S}$ takes the form in Figure 4 (top right). Here, $\bar{\mathcal{S}}$ returns vectors $\mathbf{x} \in \mathbb{Z}_p^q$ and $\mathbf{m} \in \mathsf{G}^q$ of the same length, and leakage L. Source $\mathcal{S}$ then queries EXP on all entries of $\mathbf{x}$, and multiplies the replies with the corresponding elements from $\mathbf{m}$. We also define the subclass $\mathbf{S}^{\mathrm{duber}} \subseteq \mathbf{S}^{\mathrm{msk}}$ of *distributional Uber* (dUber) sources, as shown in Figure 4 (bottom right), where we require $\mathbf{m} = (1_{\mathsf{G}}, \ldots, 1_{\mathsf{G}})$. To simplify notation, we define sources $\mathcal{S}$ in these classes via their corresponding auxiliary algorithms $\bar{\mathcal{S}}$, and call them auxiliary masking (resp., dUber) sources. Notice that masking and dUber sources always reveal the number of EXP queries through their leakage via the length $|\mathbf{y}|$ of $\mathbf{y}$.

Focusing on masking sources, and dUber sources in particular, provides a new perspective to our contribution. Indeed, dUber sources generalize the adversary in the Uber assumption insofar as oracle queries are no longer obtained as polynomial evaluations on a product distribution, but from a general distribution. To avoid trivial attacks, the target polynomial in the Uber assumption must be linearly independent from the other ones, a requirement covered by algebraic unpredictability in this setting.

PGG SECURITY. We say that a computational group scheme Γ is PGG secure if it is $\mathrm{PGG}[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}]$ secure. In order to establish confidence in this notion and show that it hides no other obvious structural weaknesses, we prove in Section 4 that PGG security is indeed achievable in the generic-group model. However, we note that, for specific applications, PGG security with respect to subclasses of $\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$ may be sufficient. On the other hand, there may exist larger, or even incomparable source classes for which PGG security is also feasible.[7]

3.2 Definitional Choices

ON SIMPLIFYING THE DEFINITION. Recall from our introductory discussion of the PGG notion that the nature of the exponentiation oracle allows to both strengthen and simplify the definition of pseudo-generic groups. We now take a closer look at this step. Consider the "generator" PGG and algebraic unpredictability games gPGG and gAlgPred defined in Figures 5 (top left) and 5 (top right). These are the PGG and AlgPred games as adapted from the UCE notion, without any further changes. In particular, notice the following differences between the two sets of games: (1) Distinguisher $\mathcal{D}$ receives the randomized generator g in the gPGG game, but not in PGG, and (2) Predictor $\mathcal{P}$ is allowed to return any $\alpha_0 \in \mathbb{Z}_p$ in the gAlgPred game, but is forced to set $\alpha_0 = 0$ in AlgPred. Advantages in the "generator" games are defined as before. Our claim that the PGG notion can be strengthened (and simplified) by moving from the "generator" games to those in Figure 4 amounts to proving the following result.

Proposition 3.1. *Let* $\mathbf{S}$ *be a source class with the following property: If* $\mathcal{S}_g \in \mathbf{S}$*, then also* $\mathcal{S} \in \mathbf{S}$*, where* $\mathcal{S}$ *is defined in Figure 5 (bottom left).*[8] *Then* $\mathrm{PGG}[\mathbf{S} \cap \mathbf{S}^{\mathrm{alg}}] \subseteq \mathrm{gPGG}[\mathbf{S} \cap \mathbf{S}^{\mathrm{galg}}]$*. More precisely, for any computational group scheme* $\Gamma \in \mathrm{PGG}[\mathbf{S} \cap \mathbf{S}^{\mathrm{alg}}]$ *and any adversary* $(\mathcal{S}_g, \mathcal{D}_g)$ *in the gPGG game for* Γ *with* $\mathcal{S}_g \in \mathbf{S} \cap \mathbf{S}^{\mathrm{galg}}$*, there are an adversary* $(\mathcal{S}, \mathcal{D})$ *in the PGG game for* Γ *and a predictor* $\mathcal{Q}_g$ *in the gAlgPred game for* $(\Gamma, \mathcal{S}_g)$ *such that*

$$\mathrm{Adv}^{\mathrm{gPGG}}_{\Gamma,\mathcal{S}_g,\mathcal{D}_g}(\lambda) \leq \mathrm{Adv}^{\mathrm{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + q(\lambda) \cdot \mathrm{Adv}^{\mathrm{galg\text{-}pred}}_{\Gamma,\mathcal{S}_g,\mathcal{Q}_g}(\lambda) + \frac{q(\lambda)+1}{2^{\lambda-1}} \,. \tag{1}$$

[7] For instance, one could allow for more expressive forms of post-processing. However, we have not yet been able to find applications of this wider class of sources.

[8] In particular, sources must always be allowed to make one extra oracle query, and to leak one additional group element. All source classes we consider in this work satisfy this property.

Game $\mathrm{gPGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda)$:	Game $\mathrm{gAlgPred}_{\Gamma,\mathcal{S}}^{\mathcal{P}}(\lambda)$:
$b \twoheadleftarrow \{0,1\}$; $(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$ $r \twoheadleftarrow \mathbb{Z}_p^*$; $g \leftarrow g_0^r$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$ $L \twoheadleftarrow \mathcal{S}^{\mathrm{Exp}}(\pi)$; $b' \twoheadleftarrow \mathcal{D}(\pi, g, L)$; return $(b = b')$	$Q \leftarrow [\,]$; $(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$ $L \twoheadleftarrow \mathcal{S}^{\mathrm{Exp}}(\pi)$; $(\alpha_0, \alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L)$ $[x_1, \ldots, x_q] \leftarrow Q$; return $\left(\sum_{i=1}^{q} \alpha_i x_i = \alpha_0\right)$

Source $\mathcal{S}^{\mathrm{Exp}}(\pi)$:	Distinguisher $\mathcal{D}(\pi, (L, h))$:	Predictor $\mathcal{P}_g(\pi, L)$:
$L \twoheadleftarrow \mathcal{S}_g^{\mathrm{Exp}}(\pi)$; $h \leftarrow \mathrm{Exp}(1)$ return (L, h)	return $\mathcal{D}_g(\pi, h, L)$	$h \twoheadleftarrow \mathsf{G}$; $(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, (L, h))$ return $(0, \alpha_1, \ldots, \alpha_q)$

Figure 5 — *Top left:* The temporary PGG game. *Top right:* The temporary algebraic unpredictability game. In both games, the Exp oracle is defined as in the corresponding game in Figure 4. *Bottom left:* Reduction from a gPGG adversary $(\mathcal{S}_g, \mathcal{D}_g)$ to a PGG adversary $(\mathcal{S}, \mathcal{D})$. *Bottom right:* Definition of the predictor $\mathcal{P}_g$.

Furthermore, $\mathcal{S} \in \mathbf{S} \cap \mathbf{S}^{\mathrm{alg}}$. More precisely, for any predictor $\mathcal{P}$ in the AlgPred game for $(\Gamma, \mathcal{S})$, there is a predictor $\mathcal{P}_g$ in the gAlgPred game for $(\Gamma, \mathcal{S}_g)$ such that

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \leq \mathrm{Adv}^{\text{galg-pred}}_{\Gamma,\mathcal{S}_g,\mathcal{P}_g}(\lambda) + q(\lambda) \cdot \mathrm{Adv}^{\text{galg-pred}}_{\Gamma,\mathcal{S}_g,\mathcal{Q}_g}(\lambda) + \frac{q(\lambda)}{2^{\lambda-1}}\,. \tag{2}$$

Here, $q(\lambda)$ is an upper bound on the number of queries made by $\mathcal{S}_g$ to the Exp *oracle.*

Proof. Fix a source class $\mathbf{S}$ as above, and let $\Gamma \in \mathrm{PGG}[\mathbf{S} \cap \mathbf{S}^{\mathrm{alg}}]$. To show that $\Gamma \in \mathrm{gPGG}[\mathbf{S} \cap \mathbf{S}^{\mathrm{galg}}]$, consider any adversary $(\mathcal{S}_g, \mathcal{D}_g)$ in the gPGG game for Γ with $\mathcal{S}_g \in \mathbf{S} \cap \mathbf{S}^{\mathrm{galg}}$, and define the PGG adversary $(\mathcal{S}, \mathcal{D})$ against Γ as shown in Figure 5 (bottom left). By assumption, $\mathcal{S} \in \mathbf{S}$.

Advantage bound. To prove Inequality (1), first recall that

$$\mathrm{Adv}^{\mathrm{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) = \Pr\left[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\middle|\, b = 1\right] - \Pr\left[\neg\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\middle|\, b = 0\right].$$

We study the two summands separately. For the first term, it is easily verified by direct inspection that the PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$ with bit $b = 1$ fixed is identical to the gPGG game for Γ played by $(\mathcal{S}_g, \mathcal{D}_g)$ with bit $b = 1$ fixed. This in particular means that $\Pr\left[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\middle|\, b = 1\right] = \Pr\left[\mathrm{gPGG}_{\Gamma}^{\mathcal{S}_g,\mathcal{D}_g}(\lambda) \,\middle|\, b = 1\right]$.

We now study the second term in the sum above. To do so, consider the following sequence of games (the formal description of which can be found in Figure 6 (top)):

- $\mathsf{Game}_0(\lambda)$ is the PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$, with bit $b = 0$ fixed and inverted winning condition. We also implement the exponentiation oracle via lazy sampling, instead of sampling the encoding σ all at once.
- $\mathsf{Game}_1(\lambda)$ is the same as $\mathsf{Game}_0(\lambda)$, but we sample h as a random element among those not yet returned by Exp. If input 1 was queried before to the oracle, we return a consistent answer.
- $\mathsf{Game}_2(\lambda)$ is the same as $\mathsf{Game}_1(\lambda)$, but we no longer check for consistency.
- $\mathsf{Game}_3(\lambda)$ is the same as $\mathsf{Game}_2(\lambda)$, but we let h be uniformly random in G. If we happen to sample h among the values already returned by Exp, we resample h and ensure that it is fresh.
- $\mathsf{Game}_4(\lambda)$ is the same as $\mathsf{Game}_3(\lambda)$, but we don't resample h. That is, h is now a random element in G.
- $\mathsf{Game}_5(\lambda)$ is the same as $\mathsf{Game}_4(\lambda)$, but we ensure that $h \neq 1_{\mathsf{G}}$ by resampling.
- $\mathsf{Game}_6(\lambda)$ is the same as $\mathsf{Game}_5(\lambda)$, we simply rewrite the sampling of h in a more compact way. By direct inspection, this game is equivalent to the gPGG game for Γ played by $(\mathcal{S}_g, \mathcal{D}_g)$, with bit $b = 0$ fixed and inverted winning condition.

We now argue that the difference between the success probabilities of subsequent games is small.

Game $\mathsf{Game}_0(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \leftarrow \text{Exp}(1)$
return $\mathcal{D}_g(\pi, h, L)$

Game $\mathsf{Game}_1(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $\boxed{h \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)}$
$\boxed{\text{if } (1 \in \mathsf{Dom}(T)) \text{ then}}$
$\quad \boxed{\mathsf{Bad}_1 \leftarrow \mathsf{true};\ h \leftarrow \text{Exp}(1)}$
return $\mathcal{D}_g(\pi, h, L)$

Comm. proc. $\text{Exp}(x)$:
if $(T[x] = \perp)$ then
$\quad g \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)$; $T[x] \leftarrow g$
return $T[x]$

Game $\mathsf{Game}_2(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)$
return $\mathcal{D}_g(\pi, h, L)$

Game $\mathsf{Game}_3(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $\boxed{h \twoheadleftarrow \mathsf{G}}$
$\boxed{\text{if } (h \in \mathsf{Rng}(T)) \text{ then}}$
$\quad \boxed{\mathsf{Bad}_2 \leftarrow \mathsf{true};\ h \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)}$
return $\mathcal{D}_g(\pi, h, L)$

Game $\mathsf{Game}_4(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \twoheadleftarrow \mathsf{G}$
return $\mathcal{D}_g(\pi, h, L)$

Game $\mathsf{Game}_5(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \twoheadleftarrow \mathsf{G}$
$\boxed{\text{if } (h = 1_\mathsf{G}) \text{ then}}$
$\quad \boxed{\mathsf{Bad}_3 \leftarrow \mathsf{true};\ h \twoheadleftarrow \mathsf{G} \setminus \{1_\mathsf{G}\}}$
return $\mathcal{D}_g(\pi, h, L)$

Game $\mathsf{Game}_6(\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \twoheadleftarrow \mathsf{G} \setminus \{1_\mathsf{G}\}$
return $\mathcal{D}_g(\pi, h, L)$

Games $\mathsf{Query1}(\lambda)/\mathsf{Query1}_i(\lambda)$:
$Q \leftarrow []$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \text{Inj}(\mathbb{Z}_p, \mathsf{G})$; $L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $[x_1, \ldots, x_q] \leftarrow Q$
$\mathsf{Query1}(\lambda)$: return $(1 \in Q)$
$\mathsf{Query1}_i(\lambda)$: if $(i \in [q])$ then return $(x_i = 1)$ else return 0

Predictor $\mathcal{Q}_{t,k,i}(\pi, L)$:
$\boldsymbol{\alpha} \leftarrow 0^{k+1}$; $\boldsymbol{\alpha}[0] \leftarrow 1$; $\boldsymbol{\alpha}[i] \leftarrow 1$
return $\boldsymbol{\alpha}$

Figure 6 — *Top:* Code of the intermediate games in the proof of the gPGG advantage bound (1). *Bottom:* Definition of the games $\mathsf{Query1}(\lambda)$ and $\mathsf{Query1}_i(\lambda)$, and of the predictor $\mathcal{Q}_{t,k,i}$. Here, the vector $\boldsymbol{\alpha}$ is indexed starting from 0 for convenience.

$\mathsf{Game}_0 \rightsquigarrow \mathsf{Game}_1$. Observe that h has the same distribution in both games: If $\mathcal{S}_g$ has queried Exp on 1 during its execution, then we set h consistently with the prior oracle reply. Otherwise, h is a random element from G different from all those previously returned by Exp. Therefore, $\mathsf{Game}_0(\lambda)$ and $\mathsf{Game}_1(\lambda)$ are indistinguishable.

$\mathsf{Game}_1 \rightsquigarrow \mathsf{Game}_2$. By definition, $\mathsf{Game}_1(\lambda)$ and $\mathsf{Game}_2(\lambda)$ are identical until Bad_1, which means that $|\Pr[\mathsf{Game}_1(\lambda)] - \Pr[\mathsf{Game}_2(\lambda)]| \le \Pr[\mathsf{Game}_1(\lambda) \text{ sets } \mathsf{Bad}_1]$ by the fundamental lemma of game playing [BR06]. To bound the latter probability, observe that

$$
\begin{aligned}
\Pr[\mathsf{Game}_1(\lambda) \text{ sets } \mathsf{Bad}_1] &= \Pr[\mathsf{Query1}(\lambda)] = \sum_{k=1}^{q(\lambda)} \Pr[\mathsf{Query1}(\lambda) \mid q = k] \Pr[q = k] \\
&\le \sum_{k=1}^{q(\lambda)} \sum_{i=1}^{k} \Pr[\mathsf{Query1}_i(\lambda) \mid q = k] \Pr[q = k] \le \sum_{k=1}^{q(\lambda)} \sum_{i=1}^{k} \Pr\big[\text{gAlgPred}_{\Gamma,\mathcal{S}_g}^{\mathcal{Q}_{t,k,i}}(\lambda) \;\big|\; q = k\big] \Pr[q = k] \\
&\le \sum_{k=1}^{q(\lambda)} k \Pr\big[\text{gAlgPred}_{\Gamma,\mathcal{S}_g}^{\mathcal{Q}_{t,k}}(\lambda) \;\big|\; q = k\big] \Pr[q = k] \le q(\lambda) \sum_{k=1}^{q(\lambda)} \Pr\big[\text{gAlgPred}_{\Gamma,\mathcal{S}_g}^{\mathcal{Q}_g}(\lambda) \;\big|\; q = k\big] \Pr[q = k] \\
&= q(\lambda) \cdot \text{Adv}_{\Gamma,\mathcal{S}_g,\mathcal{Q}_g}^{\text{galg-pred}}(\lambda)\,.
\end{aligned}
$$

Here, games $\mathsf{Query1}(\lambda)$ and $\mathsf{Query1}_i(\lambda)$, and predictors $\mathcal{Q}_{t,k,i}$ are defined in Figure 6 (bottom), predictor $\mathcal{Q}_{t,k} \in \{\mathcal{Q}_{t,k,1}, \ldots, \mathcal{Q}_{t,k,k}\}$ is the one with the largest advantage in the gAlgPred game for $(\Gamma, \mathcal{S}_g)$, and $\mathcal{Q}_g$ is the predictor that reads q off the leakage passed by $\mathcal{S}_g$ and then runs $\mathcal{Q}_{t,q}$.

$\mathsf{Game}_2 \rightsquigarrow \mathsf{Game}_3$. Similarly to the first transition, these two games are indistinguishable, because h has the same distribution in both games: In both cases it is a random element outside of $\mathsf{Rng}(T)$.

Game $\mathsf{Game}'_0(\lambda)$:
$Q \leftarrow []$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$; $L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \leftarrow \text{Exp}(1)$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L, h)$; $[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^q \alpha_i x_i = 0\right)$

Game $\mathsf{Game}'_1(\lambda)$:
$Q \leftarrow []$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$
$L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $\boxed{h \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)}$
$\boxed{\text{if } (1 \in \mathsf{Dom}(T)) \text{ then } \mathsf{Bad}_1 \leftarrow \mathsf{true};\ h \leftarrow \text{Exp}(1)}$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L, h)$; $[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^q \alpha_i x_i = 0\right)$

Comm. proc. $\text{Exp}(x)$:
$Q \leftarrow Q : x$
if $(T[x] = \bot)$ then $g \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)$; $T[x] \leftarrow g$
return $T[x]$

Game $\mathsf{Game}'_2(\lambda)$:
$Q \leftarrow []$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$; $L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L, h)$; $[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^q \alpha_i x_i = 0\right)$

Game $\mathsf{Game}'_3(\lambda)$:
$Q \leftarrow []$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$; $L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $\boxed{h \twoheadleftarrow \mathsf{G}}$
$\boxed{\text{if } (h \in \mathsf{Rng}(T)) \text{ then } \mathsf{Bad}_2 \leftarrow \mathsf{true};\ h \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T)}$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L, h)$; $[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^q \alpha_i x_i = 0\right)$

Game $\mathsf{Game}'_4(\lambda)$:
$Q \leftarrow []$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $T \leftarrow []$; $L \twoheadleftarrow \mathcal{S}_g^{\text{Exp}}(\pi)$; $h \twoheadleftarrow \mathsf{G}$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L, h)$; $[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^q \alpha_i x_i = 0\right)$

Figure 7 — Code of the intermediate games in the proof of the algebraic unpredictability bound (2).

$\mathsf{Game}_3 \rightsquigarrow \mathsf{Game}_4$. By definition, $\mathsf{Game}_3(\lambda)$ and $\mathsf{Game}_4(\lambda)$ are identical until Bad_2, which means that $|\Pr[\mathsf{Game}_3(\lambda)] - \Pr[\mathsf{Game}_4(\lambda)]| \leq \Pr[\mathsf{Game}_3(\lambda) \text{ sets } \mathsf{Bad}_2]$ by the fundamental lemma of game playing. Now recall that $\mathsf{Rng}(T)$ is a set of at most $q(\lambda)$ many elements, and since h is random in G, which contains at least $2^{\lambda-1}$ elements, we have

$$|\Pr[\mathsf{Game}_3(\lambda)] - \Pr[\mathsf{Game}_4(\lambda)]| \leq \Pr[\mathsf{Game}_3(\lambda) \text{ sets } \mathsf{Bad}_2] \leq \frac{q(\lambda)}{2^{\lambda-1}}\,.$$

$\mathsf{Game}_4 \rightsquigarrow \mathsf{Game}_5$. Again, $\mathsf{Game}_4(\lambda)$ and $\mathsf{Game}_5(\lambda)$ are identical until Bad_3, and as before the fundamental lemma of game playing gives

$$|\Pr[\mathsf{Game}_4(\lambda)] - \Pr[\mathsf{Game}_5(\lambda)]| \leq \Pr[\mathsf{Game}_5(\lambda) \text{ sets } \mathsf{Bad}_3] \leq \frac{1}{2^{\lambda-1}}\,.$$

$\mathsf{Game}_5 \rightsquigarrow \mathsf{Game}_6$. Clearly, these two games are indistinguisable, because we have only made syntactic changes in the code.

Combining the above estimates we obtain Inequality (1) for the gPGG advantage.

Algebraic unpredictability. It remains to be shown that, for every gPGG adversary $(\mathcal{S}_g, \mathcal{D}_g)$, the PGG source $\mathcal{S}$ defined as in Figure 5 (bottom left) is algebraically unpredictable. To this end, let $(\mathcal{S}_g, \mathcal{D}_g)$ be a gPGG adversary against Γ and $\mathcal{P}$ be any predictor in the algebraic unpredictability game against $\mathcal{S}$. We prove Inequality (2) via a sequence of games. As before, we give here a short description of each game, and present their formal code in Figure 7:

$\mathsf{Game}'_0(\lambda)$ is the algebraic unpredictability game for $(\Gamma, \mathcal{S})$ played by $\mathcal{P}$. We also implement the exponentiation oracle via lazy sampling, instead of sampling the encoding σ all at once.

$\mathsf{Game}'_1(\lambda)$ is the same as $\mathsf{Game}'_0(\lambda)$, but we sample h as a random element among those not yet returned by Exp. If input 1 was queried before to the oracle, we return a consistent answer.

$\mathsf{Game}'_2(\lambda)$ is the same as $\mathsf{Game}_1(\lambda)$, but we no longer check for consistency.

$\mathsf{Game}'_3(\lambda)$ is the same as $\mathsf{Game}'_2(\lambda)$, but we let h be a random element in G. We make sure that $h \notin \mathsf{Rng}(T)$ via resampling.

$\mathsf{Game}'_4(\lambda)$ is the same as $\mathsf{Game}'_3(\lambda)$, but we no longer resample h. Notice that this game is equivalent to the gAlgPred game for $(\Gamma, \mathcal{S}_g)$ played by predictor $\mathcal{P}_g$ defined in Figure 5 (bottom right).

As before, we now relate the success probabilities of subsequent games. Observe that transitions $\mathsf{Game}'_0 \rightsquigarrow \mathsf{Game}'_1$ to $\mathsf{Game}'_3 \rightsquigarrow \mathsf{Game}'_4$ are equivalent to transitions $\mathsf{Game}_0 \rightsquigarrow \mathsf{Game}_1$ to $\mathsf{Game}_3 \rightsquigarrow \mathsf{Game}_4$, respectively, which means that the same analysis as above applies to these games. Accordingly, we obtain Inequality (2) for the algebraic unpredictability advantage. □

Other definitional choices. Observe that, in the PGG game, the randomized group generator g plays the role of the hash key hk in the UCE game. The fact that g remains hidden from the source prevents it from trivially winning the PGG game by sampling $x \twoheadleftarrow \mathbb{Z}_p$, querying $h \leftarrow \textsc{Exp}(x)$, and checking if $g^x = h$. Similarly, g (or r) cannot be given to the distinguisher $\mathcal{D}$, since the source could query $h \leftarrow \textsc{Exp}(1)$, leak h to $\mathcal{D}$, who then checks if $g = h$ (resp., $g_0^r = h$).

Note also that the random injection σ that the game samples has G (the real group), and not some larger set S, as its range. This is needed because the source can check group elements for validity (e.g., using exponentiation to power $p-1$, or directly via an element validity algorithm if such a procedure is available).

Also observe that the source does not get oracle access to the operation op defined by σ. The reason is that, with such access, once again trivial attacks arise: The source samples two random group elements, then multiplies them first using the op oracle and then again locally using the input group operation $\circ$, and finally checks if the results match.[9] Removing access to the operation oracle from $\mathcal{S}$ does not restrict our ability to prove security results in the PGG model, as we shall see in Section 6.

Computational algebraic unpredictability. In [BFM14], Brzuska, Farshim and Mittelbach demonstrate an attack against UCEs with respect to a *computational* notion of unpredictability. The types of sources that we consider for PGG are analogous to the so-called split UCE sources. As BFM discuss, their iO-based attack does not extend to such sources. However, under the existence of a plausible form of obfuscation, attacks arise. In more detail, if the function mapping x to the obfuscation of the circuit $C[x]\colon h \mapsto h^x$ is one-way, the following attack emerges: The dUber source picks $x \twoheadleftarrow \mathbb{Z}_p$, defines $\mathbf{x} \leftarrow (1, x)$, and sets L to be an obfuscation of $C[x]$. The distinguisher then returns $(\mathbf{y}[2] = L(\mathbf{y}[1]))$. To avoid these types of attacks we focus on statistical algebraic unpredictability.

Despite this attack, there *is* a benefit in considering a computational notion of algebraic unpredictability when it comes to the analysis in idealized models. Indeed, as we show, PGG with respect to this wider class of sources is achievable in the GGM, and thus a wider class of applications can be proven secure in the GGM. This does not contradict potential security in the standard model since PGG with respect to computational algebraic unpredictability may still exist for sources that take specific forms.

Multi-base PGGs. For the UCE and psPRP notions, BHK and ST respectively considered multi-key extensions to cover a wider range of applications. These notions are not known to be equivalent to their simpler single-key counterparts. For pseudo-generic groups, on the other hand, a simple generator re-randomization argument shows that the multi-base and single-base notions are equivalent. We thus focus on the (single-base) PGG version above.

3.3 A First Example

To illustrate how reductions in PGGs work, we show here that the generator g_0 returned by a computational group scheme Γ can always assumed to be uniformly distributed. More involved (and interesting) applications of PGGs will be discussed in Section 6.

Proposition 3.2. *Let Γ be a computational group scheme, and consider the computational group scheme Γ_u defined in Figure 8 (top left). If Γ is $\mathrm{PGG}[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}]$ secure, then Γ_u is also $\mathrm{PGG}[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}]$ secure.*

[9] On the other hand, it is unclear how to rule this attack out using an extended notion of algebraic unpredictability that takes operation queries into account.

Group scheme $\Gamma_u(1^\lambda)$:	Predictor $\mathcal{P}_u(\pi_u, \mathbf{y}, L)$:
$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$ $s \twoheadleftarrow \mathbb{Z}_p^*$; $g_u \leftarrow g_0^s$ $\pi_u \leftarrow (\circ, g_u, p)$; return π_u	$(\circ, g_u, p) \leftarrow \pi_u$; $R := \{r \in \{0,1\}^{t(\lambda)} \mid \Gamma(1^\lambda; r) = (\circ, \cdot, p)\}$ $r \twoheadleftarrow R$; $(\tilde{\pi} := (\circ, \tilde{g}_0, p)) \twoheadleftarrow \Gamma(1^\lambda; r)$; $s \leftarrow \mathsf{DLog}_{\tilde{g}_0}(g_u)$ $(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\tilde{\pi}, \mathbf{y}, (L, s))$; return $(\alpha_1, \ldots, \alpha_q)$

Auxiliary masking source $\bar{\mathcal{S}}(\pi)$:	PGG distinguisher $\mathcal{D}(\pi, \mathbf{y}, (L, s))$:
$(\circ, g_0, p) \leftarrow \pi$; $s \twoheadleftarrow \mathbb{Z}_p^*$; $g_u \leftarrow g_0^s$; $\pi_u \leftarrow (\circ, g_u, p)$ $(\mathbf{x}, \mathbf{m}, L) \twoheadleftarrow \bar{\mathcal{S}}_u(\pi_u)$; return $(\mathbf{x}, \mathbf{m}, (L, s))$	$(\circ, g_0, p) \leftarrow \pi$; $g_u \leftarrow g_0^s$; $\pi_u \leftarrow (\circ, g_u, p)$ $b' \twoheadleftarrow \mathcal{D}_u(\pi_u, \mathbf{y}, L)$; return b'

Figure 8 — *Top left:* Definition of the computational group scheme Γ_u. *Top right:* Definition of the predictor $\mathcal{P}_u$. Here, λ is the bitlength of p, $t(\lambda)$ is an upper bound on the runtime of $\Gamma(1^\lambda)$, and $\Gamma(1^\lambda; r)$ denotes running Γ on input 1^λ and random coins r. *Bottom:* Reduction from a PGG adversary $(\mathcal{S}_u, \mathcal{D}_u)$ against Γ_u to a PGG adversary $(\mathcal{S}, \mathcal{D})$ against Γ.

More precisely, for any adversary $(\mathcal{S}_u, \mathcal{D}_u)$ in the PGG game for Γ_u with $\mathcal{S}_u \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$, there is an adversary $(\mathcal{S}, \mathcal{D})$ in the PGG game for Γ such that

$$\mathrm{Adv}^{\mathrm{pgg}}_{\Gamma_u, \mathcal{S}_u, \mathcal{D}_u}(\lambda) = \mathrm{Adv}^{\mathrm{pgg}}_{\Gamma, \mathcal{S}, \mathcal{D}}(\lambda)\,. \tag{3}$$

Furthermore, $\mathcal{S} \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$. More precisely, for any predictor $\mathcal{P}$ in the AlgPred game for $(\Gamma, \mathcal{S})$, there is a predictor $\mathcal{P}_u$ in the AlgPred game for $(\Gamma_u, \mathcal{S}_u)$ such that

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma, \mathcal{S}, \mathcal{P}}(\lambda) = \mathrm{Adv}^{\text{alg-pred}}_{\Gamma_u, \mathcal{S}_u, \mathcal{P}_u}(\lambda)\,. \tag{4}$$

Proof. Given an adversary $(\mathcal{S}_u, \mathcal{D}_u)$ in the PGG game for Γ_u with $\mathcal{S}_u \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$, define the PGG masking source $\mathcal{S}$ via auxiliary algorithm $\bar{\mathcal{S}}$ and the distinguisher $\mathcal{D}$ as shown in Figure 8 (bottom).

Masking structure. By construction, it is clear that $\bar{\mathcal{S}}$ makes no Exp query and returns vectors $\mathbf{x}$ and $\mathbf{m}$ of equal length. Thus, $\mathcal{S} \in \mathbf{S}^{\mathrm{msk}}$.

Advantage bound. By direct inspection, $\mathrm{PGG}^{\mathcal{S}_u, \mathcal{D}_u}_{\Gamma_u}(\lambda) = \mathrm{PGG}^{\mathcal{S}, \mathcal{D}}_{\Gamma}(\lambda)$, which means that the advantages in the two games also coincide, thus proving Equality (3).

Algebraic unpredictability. It remains to be shown that, for every PGG source $\mathcal{S}_u$ as above, source $\mathcal{S}$ is algebraically unpredictable. To this end, let $\mathcal{P}$ be any predictor in the algebraic unpredictability game against $\mathcal{S}$, and consider the predictor $\mathcal{P}_u$ in the algebraic unpredictability game against $\mathcal{S}_u$ defined in Figure 8 (top right). Notice that $\mathcal{P}_u$ receives as input public parameters π_u with a uniformly distributed generator, but needs parameters π as returned by Γ to run $\mathcal{P}$. To this end, $\mathcal{P}_u$ searches the randomness space of $\Gamma(1^\lambda)$ (which is possible since Γ is PPT and $\mathcal{P}_u$ can be unbounded) and randomly samples some randomness r which results in $\Gamma(1^\lambda)$ returning parameters $\tilde{\pi}$ with $(\circ, p)$ as specified in π_u. The corresponding group generator $\tilde{g}_0$ is then correctly distributed, and $\mathcal{P}_u$ can now run $\mathcal{P}$ on $\tilde{\pi}$. Again by direct inspection, we have $\mathrm{AlgPred}^{\mathcal{P}}_{\Gamma, \mathcal{S}}(\lambda) = \mathrm{AlgPred}^{\mathcal{P}_u}_{\Gamma_u, \mathcal{S}_u}(\lambda)$, which proves Equality (4). □

Remark. A similar result also holds in the opposite direction, namely, if Γ_u is $\mathrm{PGG}[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}]$ secure, then so is Γ. The proof is very similar to the one carried out above, but the roles of the source and the predictor are inverted: In this case, it is the auxiliary masking source (constructed in the proof) who searches for a correctly distributed generator $\tilde{g}_0$ (which again is possible, since masking sources can be unbounded), while the predictor simply re-randomizes the given generator.

4 Generic Groups are PGGs

In this section we show the feasibility of PGGs in the generic-group model. The importance of this result is that it rules out generic attacks against the PGG notion, thus forming a check on the soundness of the definitional framework. Furthermore, it automatically lifts the security of each application of PGGs that we consider to the GGM, as long as algebraically unpredictable, masking sources are used. This is similar to the Uber assumption family, where one relies on a specific assumption within Uber, and *reuses* the GGM hardness proved once for the whole family. As discussed above, we show GGM hardness of PGGs for a *computational* notion of algebraic unpredictability, which widens the applicability of our result.

The present section is structured as follows. We first clarify which flavor of the generic-group model we use to study the soundness of the PGG notion, and recall the bit-fixing lemma for generic groups [CDG18,BMZ19]. We then define what it means for a generic group to be PGG secure, and discuss our definitional choices. After stating our main theorem and some auxiliary lemmata, we then proceed to the proof of our result.

4.1 More on the Generic-Group Model

Formalizing the GGM. Different abstractions for "generic algorithms" on groups exist in the literature. Two prominent such models are that of Mauer [Mau05], where algorithms are given abstract "pointers" representing group elements, and the one by Shoup [Sho97], where elements are encoded as random strings. We follow the second approach, which has been more common in recent years. A thorough comparison between these two models can be found in the recent work of Zhandry [Zha22].

We note that even with random group encodings, it is still possible to define technically different models. For instance, Coretti et al. [CDG18] set the co-domain of encodings to $[M]$ (where $M \in \mathbb{N}$), whereas Shoup allows for an arbitrary co-domain $\mathsf{S} \subseteq \{0,1\}^*$. It is, however, not hard to see that by enumerating the elements of S via a fixed permutation, the two models are equivalent. That is, choosing the co-domain of the encoding does not affect the model.

We also observe that in his analysis of the discrete-logarithm problem, Shoup only allows for operations on elements that were obtained from the group encoding and operation oracles. Following recent works (see, e.g., [CDG18]), we on the other hand allow submitting any string in the co-domain of the encoding.

Bit-fixing GGM. The generic-group model offers a simple way to study the hardness (resp., security) of group-based assumptions and constructions against uniform adversaries without preprocessing. On the other hand, it performs poorly when considering either non-uniform attackers or allowing preprocessing. To address this shortcoming, Coretti, Dodis and Guo [CDG18] define the auxiliary-input GGM (AI-GGM), an extension of the ordinary GGM which takes into account this wider class of adversaries. Unfortunately, it turns out that this model is much harder to work in compared to the "plain" GGM.

Following prior work on the ROM, Coretti et al. [CDG18] also define an intermediate model, the bit-fixing GGM (BF-GGM). In this model, which is parameterized by $P \in \mathbb{N}$, the leakage of the first-stage adversary is converted into a list $\mathcal{L}$ of at most P assignments without collisions of the form $(x, y) \in \mathbb{Z}_p \times \mathsf{S}$. Then a new generic-group encoding is sampled in compatibility with the assignment $\mathcal{L}$ and is used in the online phase of the attack. This model is technically simpler and closer to the GGM than the AI-GGM, because outside of points in $\mathcal{L}$ the new encoding looks completely random to the online attacker.

Surprisingly, Coretti et al. manage to relate the two models by reducing security of any game in the AI-GGM to the BF-GGM, as shown below. The lemma we present is a reformulation of the decomposition result for injection sources [CDG18, Lemma 20], with the improvement that the function $\mathfrak{A}$ computing the leakage can be randomized, as in [BMZ19, Lemma 9]. In the following, for a set $\mathcal{L} \subseteq \mathbb{Z}_p \times \mathsf{S}$ without collisions, we denote by $\mathrm{Inj}\,(\mathbb{Z}_p, \mathsf{S} \mid \mathcal{L})$ the set of all injective functions from $\mathbb{Z}_p$ to S which extend $\mathcal{L}$.

Game AI-GG$^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S}}$:	Game BF-GG$^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S},\gamma,P}$:
$\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$; $z \twoheadleftarrow \mathfrak{A}(\tau)$	$\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$; $z \twoheadleftarrow \mathfrak{A}(\tau)$; $\mathcal{L} \leftarrow \textsc{Decomp}_{\mathfrak{A}}(\mathsf{Rng}(\tau), \gamma, P, z)$
return $\mathfrak{D}^{\tau,\tau^{-1}}(z)$	$\tau_{\mathcal{L}} \twoheadleftarrow \mathrm{Inj}\left(\mathbb{Z}_p, \mathsf{Rng}(\tau) \mid \mathcal{L}\right)$; return $\mathfrak{D}^{\tau_{\mathcal{L}},\tau_{\mathcal{L}}^{-1}}(z)$

Figure 9 — Generic games AI-GG and BF-GG from Lemma 4.1.

Lemma 4.1. *Let $p, S \in \mathbb{N}$, S a set with $|\mathsf{S}| \geq p$, and let $\mathfrak{A}$ be an algorithm taking $\tau \in \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$ as input and returning an S-bit string. Then there exists an algorithm $\textsc{Decomp}_{\mathfrak{A}}$ with the following property: For all $\gamma > 0$, $P \in \mathbb{N}$, and for any distinguisher $\mathfrak{D}$ making at most T queries to its oracles, we have*

$$\left|\Pr\left[\text{AI-GG}^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S}}\right] - \Pr\left[\text{BF-GG}^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S},\gamma,P}\right]\right| \leq \frac{2(S - \log\gamma)T}{P} + \gamma\,,$$

where the games AI-GG$^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S}}$ and BF-GG$^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S},\gamma,P}$ are defined in Figures 9 (left) and 9 (right).

Proof. Fix p, S, S, and $\mathfrak{A}$ as in the statement of the Lemma. Algorithm $\textsc{Decomp}_{\mathfrak{A}}$ then works as follows: For given T, γ, P, and z, it computes a convex combination of $\left(P, \mathsf{T}, 1 - \frac{S - \log\gamma}{P\log(p/e)}\right)$-dense sources that are γ-close to the random variable X_z, defined as the uniform distribution X on $\mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$ conditioned on $\mathfrak{A}(X) = z$. This can be done as shown in [CDG18, Lemma 13]. With this definition, the statement of the Lemma is equivalent to the one of [BMZ19, Lemma 9]. □

Remark. The lemma above holds for indistinguishability and unpredictability applications, and we will use it to relate the advantages of adversaries both in the PGG and AlgPred games, with and without resampling. We mention that Coretti et al. [CDG18] also prove another version of this result, which achieves a tighter bound but only holds for unpredictability games. Unfortunately, we cannot use this alternative version for the AlgPred game, because it only allows to bound the AI-advantage in terms of the BF-advantage (which is the way the result is intended to be used, since working in the BF-GGM is usually easier), whereas for the AlgPred game we need a bound in the opposite direction (notice that Lemma 4.1 is symmetric, so both directions hold with the looser bound).

4.2 PGGs in the GGM

We now clarify what it means for a generic group to be PGG secure. The PGG and AlgPred games in the GGM for a group of size p with target set S are presented in Figures 10 (top left) and 10 (top center), masking sources are given in Figure 10 (top right). We stress that the oracles τ and op defining the generic group and its operation are *independent* of the injection σ used to define PGG security—only the ranges of the two encodings coincide, since, as in the standard model, σ must take values in the group (which is given by $\mathsf{Rng}(\tau)$ in the GGM). Advantage terms are defined as usual:

$$\mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} := 2 \cdot \Pr\left[\text{PGG}^{\mathcal{S},\mathcal{D}}_{p,\mathsf{S}}\right] - 1\,, \qquad \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} := \Pr\left[\text{AlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}\right].$$

Recall that masking sources can be unbounded, which means that they are allowed an unlimited amount of group operations. Following [BMZ19], we mirror this in the GGM by giving $\mathcal{S}$ the entire function table of τ. Distinguisher $\mathcal{D}$ on the other hand is bounded, which means that it is only given oracles for τ and op and that the leakage L must be short. This choice of modeling more accurately reflects unbounded sources in the GGM by allowing an arbitrary number of group operations. Furthermore, it allows us to derive security in the presence of preprocessing attackers, as our sources can leak information about τ to the distinguisher.[10]

[10] In particular, this model allows a restricted class of sources that leak arbitrary information (without any unpredictability requirements), as long as the sampling of the exponents is unpredictable (e.g., random, as is the case for the DLP).

Game $\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{p,\mathsf{S}}$:

$b \twoheadleftarrow \{0,1\}$; $r \twoheadleftarrow \mathbb{Z}_p^*$; $\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$
$\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{Rng}(\tau))$
$L \twoheadleftarrow \mathcal{S}^{\mathrm{Exp}}(\tau)$
$b' \twoheadleftarrow \mathcal{D}^{\tau,\mathsf{op}}(L)$
return $(b = b')$

Proc. $\mathrm{Exp}(x)$:
if $(b = 0)$ then return $\sigma(x)$
return $\tau(rx)$

Game $\mathrm{AlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}$:

$Q \leftarrow [\,]$; $\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$
$\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{Rng}(\tau))$
$L \twoheadleftarrow \mathcal{S}^{\mathrm{Exp}}(\tau)$
$(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}^{\tau,\mathsf{op}}(L)$
$[x_1, \ldots, x_q] \leftarrow Q$
return $\left(\sum_{i=1}^{q} \alpha_i x_i = 0\right)$

Proc. $\mathrm{Exp}(x)$:
$Q \leftarrow Q : x$; return $\sigma(x)$

Masking source $\mathcal{S}^{\mathrm{Exp}}(\tau)$:

$(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}(\tau)$
for $i = 1$ to $|\mathbf{m}|$ do
  $\mathbf{y}[i] \leftarrow \mathsf{op}(\mathbf{m}[i], \mathrm{Exp}(\mathbf{x}[i]))$
return $(\mathbf{y}, \bar{L})$

Comm. proc. $\mathsf{op}(h_1, h_2)$:
return $\tau\big(\tau^{-1}(h_1) + \tau^{-1}(h_2)\big)$

Auxiliary source $\bar{\mathcal{S}}'(\tau)$:

$(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}(\tau)$; $q \leftarrow |\mathbf{x}|$; $l \leftarrow |\bar{L}|$
$\bar{L}' \leftarrow (0^{2(q_{\mathcal{S}}-q)(\lfloor \log p \rfloor + 1) + (\ell - l)}, \bar{L}, q, l)$
return $(\mathbf{x}, \mathbf{m}, \bar{L}')$

Distinguisher $\mathcal{D}'^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L}')$:

$(0^{2(q_{\mathcal{S}}-q)(\lfloor \log p \rfloor + 1) + (\ell - l)}, \bar{L}, q, l) \leftarrow \bar{L}'$
return $\mathcal{D}^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$

Predictor $\mathcal{P}^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$:

$q \leftarrow |\mathbf{y}|$; $l \leftarrow |\bar{L}|$
$\bar{L}' \leftarrow (0^{2(q_{\mathcal{S}}-q)(\lfloor \log p \rfloor + 1) + (\ell - l)}, \bar{L}, q, l)$
return $\mathcal{P}'^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L}')$

Figure 10 — *Top left:* The PGG game in the GGM. *Top center:* The AlgPred game in the GGM. *Top right:* A masking source $\mathcal{S}$ in the GGM. In all games, $|\mathsf{S}| \geq p$, and without loss of generality, all algorithms know p. *Bottom left:* Adversaries $(\mathcal{S}', \mathcal{D}')$ constructed from $(\mathcal{S}, \mathcal{D})$. *Bottom right:* Reduction from the proof of Lemma 4.2.

4.3 Preliminary Results

Before coming to the main theorem in this section, we prove two auxiliary results. In the next lemma, we show that one can turn any algebraically unpredictable, masking source $\mathcal{S}$ into a source $\mathcal{S}'$ where $\bar{\mathcal{S}}'$ has constant output length, while preserving the query complexity and the algebraic unpredictability of $\mathcal{S}$. This is needed since Lemma 4.1 holds for algorithms $\mathfrak{A}$ returning an output of fixed length. Afterwards, we study the advantage of a predictor $\mathcal{P}$ in an algebraic unpredictability game where the generic-group encoding is resampled and $\mathcal{P}$ is allowed multiple guesses. We will use Lemma 4.1 to relate $\mathcal{P}$'s advantage to a classical algebraic unpredictability advantage.

Lemma 4.2. *Let $p \in \mathbb{N}$, S a set with $|\mathsf{S}| \geq p$, and let $(\mathcal{S}, \mathcal{D})$ be an adversary in the PGG game with $\mathcal{S} \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$ making at most $q_{\mathcal{S}}$ oracle queries and returning at most ℓ bits of leakage $\bar{L}$. Then there exists a PGG adversary $(\mathcal{S}', \mathcal{D}')$ with $\mathcal{S}' \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$ making the same oracle calls as $\mathcal{S}$, with $\bar{\mathcal{S}}'$ returning exactly $2q_{\mathcal{S}}(\lfloor \log p \rfloor + 1) + \ell + \lfloor \log q_{\mathcal{S}} \rfloor + \lfloor \log \ell \rfloor + 2$ bits of output, such that $\mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} = \mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S}',\mathcal{D}'}$ and, for every predictor $\mathcal{P}'$ in the game AlgPred, there exists a predictor $\mathcal{P}$ such that $\mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S}',\mathcal{P}'} = \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}}$.*

Proof. For a PGG adversary $(\mathcal{S}, \mathcal{D})$ as in the statement of the Lemma, consider the pair $(\mathcal{S}', \mathcal{D}')$ defined in Figure 10 (bottom left). By direct inspection $\mathcal{S}' \in \mathbf{S}^{\mathrm{msk}}$, it makes the same queries as $\mathcal{S}$, and satisfies $\mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} = \mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S}',\mathcal{D}'}$. As for the (total) output length of $\bar{\mathcal{S}}$, observe that if we encode every entry of $\mathbf{x}$ and $\mathbf{m}$ with $\lfloor \log p \rfloor + 1$ bits, and the numbers $q \leq q_{\mathcal{S}}$ and $l \leq \ell$ with $\lfloor \log q_{\mathcal{S}} \rfloor + 1$ and $\lfloor \log \ell \rfloor + 1$ bits, respectively, the output $(\mathbf{x}, \mathbf{m}, \bar{L}')$ has the desired length. Note that there is no need to encode separator symbols or to repeat bits, since the additional information supplied in $\bar{L}'$ allows to uniquely recover $(\mathbf{x}, \mathbf{m}, \bar{L}')$.

Finally, notice that $\mathcal{S}'$ is algebraically unpredictable. To do so, let $\mathcal{P}'$ be any predictor playing the algebraic unpredictability game for $\mathcal{S}'$, and consider the predictor $\mathcal{P}$ defined in Figure 10 (bottom right) playing the algebraic unpredictability game for $\mathcal{S}$. Since $|\mathbf{x}| = |\mathbf{y}|$ we obviously have $\mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S}',\mathcal{P}'} = \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}}$, which concludes the proof. □

Game $\mathrm{MAlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}$:

$Q \leftarrow [\,]$; $\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$
$\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{Rng}(\tau))$; $(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}(\tau)$
for $i = 1$ to $|\mathbf{m}|$ do $\mathbf{y}[i] \leftarrow \mathsf{op}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
$((\alpha_{1,1}, \ldots, \alpha_{1,q}), \ldots, (\alpha_{\kappa,1}, \ldots, \alpha_{\kappa,q})) \twoheadleftarrow \mathcal{P}^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$
$[x_1, \ldots, x_q] \leftarrow Q$
return $\big((\exists j \in [\kappa])(\sum_{i=1}^{q} \alpha_{j,i} x_i = 0)\big)$

Comm. Proc. $\textsc{Exp}(x)$:

$Q \leftarrow Q : x$; return $\sigma(x)$

Proc. $\mathsf{op}(h_1, h_2)$:

return $\tau\big(\tau^{-1}(h_1) + \tau^{-1}(h_2)\big)$

Game Game':

$Q \leftarrow [\,]$; $\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$
$\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{Rng}(\tau))$; $(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}(\tau)$
$\mathcal{L} \leftarrow \textsc{Decomp}_{\bar{\mathcal{S}}}(\mathsf{Rng}(\tau), \gamma, P, (\mathbf{x}, \mathbf{m}, \bar{L}))$
$\tau_{\mathcal{L}} \twoheadleftarrow \mathrm{Inj}\,(\mathbb{Z}_p, \mathsf{Rng}(\tau) \mid \mathcal{L})$
for $i = 1$ to $|\mathbf{m}|$ do $\mathbf{y}[i] \leftarrow \mathsf{op}_{\mathcal{L}}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
$((\alpha_{1,1}, \ldots, \alpha_{1,q}), \ldots,$
$\quad (\alpha_{\kappa,1}, \ldots, \alpha_{\kappa,q})) \twoheadleftarrow \mathcal{P}^{\tau_{\mathcal{L}},\mathsf{op}_{\mathcal{L}}}(\mathbf{y}, \bar{L})$
$[x_1, \ldots, x_q] \leftarrow Q$
return $\big((\exists j \in [\kappa])(\sum_{i=1}^{q} \alpha_{j,i} x_i = 0)\big)$

Proc. $\mathsf{op}_{\mathcal{L}}(h_1, h_2)$:

return $\tau_{\mathcal{L}}\big(\tau_{\mathcal{L}}^{-1}(h_1) + \tau_{\mathcal{L}}^{-1}(h_2)\big)$

Algorithm $\mathfrak{A}(\tau)$:

$(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}(\tau)$; return $(\mathbf{x}, \mathbf{m}, \bar{L})$

Algorithm $\mathfrak{D}^{\tau,\tau^{-1}}(\mathbf{x}, \mathbf{m}, \bar{L})$:

$Q \leftarrow [\,]$; $T_\sigma \leftarrow [\,]$
for $i = 1$ to $|\mathbf{m}|$ do $\mathbf{y}[i] \leftarrow \mathsf{op}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
$((\alpha_{1,1}, \ldots, \alpha_{1,q}), \ldots, (\alpha_{\kappa,1}, \ldots, \alpha_{\kappa,q})) \twoheadleftarrow \mathcal{P}^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$
$[x_1, \ldots, x_q] \leftarrow Q$; return $\big((\exists j \in [\kappa])(\sum_{i=1}^{q} \alpha_{j,i} x_i = 0)\big)$

Proc. $\textsc{Exp}(x)$:

$Q \leftarrow Q : x$
if $(x \notin \mathsf{Dom}(T_\sigma))$ then
$\quad g \twoheadleftarrow \mathsf{G} \setminus \mathsf{Rng}(T_\sigma)$; $T_\sigma \leftarrow T_\sigma : (x, g)$
return $T_\sigma(x)$

Proc. $\mathsf{op}(h_1, h_2)$:

return $\tau\big(\tau^{-1}(h_1) + \tau^{-1}(h_2)\big)$

Figure 11 — *Top left:* The MAlgPred game. This is the algebraic unpredictability game, where the predictor can return multiple guesses. *Top right:* The game Game'. This is the MAlgPred game with a resampled GGM encoding. *Bottom:* Reduction from the proof of Lemma 4.3.

Lemma 4.3. *Let $p \in \mathbb{N}$ and S a set with $|\mathsf{S}| \geq p$, let $P \in \mathbb{N}$, $\gamma > 0$, and let $\mathcal{S} \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$ be a source with $\bar{\mathcal{S}}$ returning exactly $S := 2q_{\mathcal{S}}(\lfloor \log p \rfloor + 1) + \ell + \lfloor \log q_{\mathcal{S}} \rfloor + \lfloor \log \ell \rfloor + 2$ bits of output (as in Lemma 4.2). Then, for every predictor $\mathcal{P}$ making at most $q_{\mathcal{P},\tau}$ and $q_{\mathcal{P},\mathsf{op}}$ calls to its oracles,*

$$\Pr[\mathsf{Game}'] \leq \Pr\big[\mathrm{MAlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}\big] + \frac{6(S - \log \gamma)T}{P} + \gamma\,,$$

where $T := q_{\mathcal{P},\tau} + q_{\mathcal{P},\mathsf{op}} + q_{\mathcal{S}}$, and the games $\mathrm{MAlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}$ and Game' are defined in Figures 11 (top left) and 11 (top right). Furthermore, for every predictor $\mathcal{P}$ in the game MAlgPred for $\mathcal{S}$ returning at most k tuples of scalars, there exists a predictor $\mathcal{P}'$ in the AlgPred game for $\mathcal{S}$ such that $\Pr\big[\mathrm{MAlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}\big] \leq k \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}'}$.

Proof. For the first statement, we use Lemma 4.1 to prove the desired inequality. To do so, we must interpret the output of $\bar{\mathcal{S}}(\tau)$ as a bitstring, which can be done as discussed in the proof of Lemma 4.2. Now consider algorithms $\mathfrak{A}$ and $\mathfrak{D}$ defined in Figure 11 (bottom). By direct inspection, for this choice of $\mathfrak{A}$ and $\mathfrak{D}$ we have $\text{AI-GG}^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S}} = \mathrm{MAlgPred}^{\mathcal{P}}_{p,\mathsf{S},\mathcal{S}}$ and $\text{BF-GG}^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S},\gamma,P} = \mathsf{Game}'$. To conclude, notice that $\mathfrak{D}$ needs two oracle calls to implement operation op in the for-loop, and one and three oracle calls to answer queries of $\mathcal{P}$ to its oracles τ and op, respectively, which means that it makes at most $3T$ oracle queries.

For the second statement, let $\mathcal{P}'$ be the predictor that runs $\mathcal{P}$ and then outputs one of the tuples returned by $\mathcal{P}$ chosen at random. The bound follows by a standard argument. □

REMARK. Notice that the result above applies Lemma 4.1 to a version of the algebraic unpredictability game where the predictor can return multiple guesses and wins if one is correct, rather than plain algebraic unpredictability. The reason we introduce this game is that, in the proof of Theorem 4.4, a direct reduction to the AlgPred game (with resampled encoding) comes with a multiplicative loss of around T^2, which then also multiplies the additive constant due to Lemma 4.1. A reduction to Game' on the other hand has multiplicative factor 1, which means that the additive loss from Lemma 4.1 remains as is when we then move from MAlgPred to AlgPred (with loss T^2). This is crucial in order to replicate bounds obtained in prior works for specific instances of masking sources.

4.4 Generic-Group Feasibility

We are now ready to state and prove the main result of this section.

Theorem 4.4. *Let $(\mathsf{G}, \circ)$ be a group of order p, and S a set with $|\mathsf{S}| \geq p$. Then the generic group on G is $\mathrm{PGG}[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}]$ secure. More precisely, for every adversary $(\mathcal{S}, \mathcal{D})$ in the PGG game with $\mathcal{S} \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}$, there exists a predictor $\mathcal{P}$ in the game AlgPred such that*

$$\mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} \leq \mathcal{O}\left(T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} + \sqrt{\frac{ST^2}{p}} \right) ,$$

where $S := 2q_{\mathcal{S}}(\lfloor \log p \rfloor + 1) + \ell + \lfloor \log q_{\mathcal{S}} \rfloor + \lfloor \log \ell \rfloor + 2$ and $T := q_{\mathcal{S}} + q_{\mathcal{D},\tau} + q_{\mathcal{D},\mathsf{op}}$. Here $q_{\mathcal{S}}$, $q_{\mathcal{D},\tau}$, and $q_{\mathcal{D},\mathsf{op}}$ are upper bounds on the number of queries made by $\mathcal{S}$ and $\mathcal{D}$ to their respective oracles, ℓ is an upper bound on the length of the leakage $\bar{L}$ returned by $\mathcal{S}$, and we assume $T \leq \sqrt{Sp}$.

Proof. Fix any $(\mathcal{S}, \mathcal{D})$ as in the statement, and consider the corresponding pair $(\mathcal{S}', \mathcal{D}')$ defined in Lemma 4.2, where $\bar{\mathcal{S}}'$ returns exactly S bits. Since $\mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} = \mathrm{Adv}^{\mathrm{pgg}}_{p,\mathsf{S},\mathcal{S}',\mathcal{D}'}$, we estimate the latter. For the remainder of the proof, let $\gamma > 0$ and $P \in \mathbb{N}$ be numbers that will be determined later. We use the game-playing framework and consider the following sequence of games:

Game_0 is the PGG game for $(\mathcal{S}', \mathcal{D}')$ with respect to $b = 1$, i.e., where the oracle EXP uses the generic group injection τ.

Game_1 is the same as Game_0, but we additionally require that the queries $\mathcal{S}'$ makes to the EXP oracle be all distinct.

Game_2 is the same as Game_1, but we use Lemma 4.1 (with $\gamma := 1/p$) to resample τ right after the execution of $\bar{\mathcal{S}}'$, whose output is treated as leakage. Also, we do not sample the new injection $\tau_{\mathcal{L}}$ all at once, but instead implement it via lazy sampling.

Game_3 is the same as Game_2, but we replace the randomly chosen exponent $r \twoheadleftarrow \mathbb{Z}_p^*$ with a formal variable R. We also start to lazy sample the new encoding σ with the values returned by EXP (i.e., we add the sampled values to a newly initialized table T_σ), and modify the condition in the if-statement in the EXP oracle into one involving T_σ.

Game_4 is the same as Game_3, but for every EXP query $\mathbf{x}[j]$, instead of saving an entry for $R\mathbf{x}[j]$ to the encoding table, we index it with a different and independent variable Z_j.

Game_5 is the same as Game_4, but we evaluate the variables Z_j at random values $c_j \twoheadleftarrow \mathbb{Z}_p$.

Game_6 is the same as Game_5, but we insist that the values c_j be pairwise different.

Game_7 is the same as Game_6, but we no longer populate table $T_{\tau_{\mathcal{L}}}$ in oracle calls to EXP.

Game_8 is the same as Game_7, but when we lazily sample replies to EXP queries, we do so consistently with σ rather than τ. In other words, we sample group elements g at random from $\mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_\sigma)$ instead of $\mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_{\tau_{\mathcal{L}}})$. Notice that the EXP oracle now only depends on σ and is completely decoupled from τ.

Game Game_1:
$r \twoheadleftarrow \mathbb{Z}_p^*$; $\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$; $(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}'(\tau)$
if $\big((\exists 1 \le i < j \le |\mathbf{x}|)(\mathbf{x}[i] = \mathbf{x}[j])\big)$ then return 0
for $i = 1$ to $|\mathbf{x}|$ do $\mathbf{y}[i] \leftarrow \mathsf{op}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
return $\mathcal{D}'^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$

Proc. $\textsc{Exp}(x)$:
return $\tau(rx)$

Proc. $\mathsf{op}(h_1, h_2)$:
return $\tau\big(\tau^{-1}(h_1) + \tau^{-1}(h_2)\big)$

Pred. $\mathcal{Q}'^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$:
$q \leftarrow |\mathbf{y}|$; $\mathbf{v} \leftarrow 0^q$
$i \twoheadleftarrow [q]$; $j \twoheadleftarrow [q] \setminus \{i\}$
$\mathbf{v}[i] \leftarrow 1$; $\mathbf{v}[j] \leftarrow -1$
return $\mathbf{v}$

Game Game_2:
$r \twoheadleftarrow \mathbb{Z}_p^*$; $\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$; $(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}'(\tau)$
if $\big((\exists 1 \le i < j \le |\mathbf{x}|)(\mathbf{x}[i] = \mathbf{x}[j])\big)$ then
  return 0
$T_{\tau_\mathcal{L}} = [(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_k, \tilde{y}_k)]$
  $\leftarrow \textsc{Decomp}_{\bar{\mathcal{S}}'}(\mathsf{Rng}(\tau), \gamma, P, (\mathbf{x}, \mathbf{m}, \bar{L}))$
for $i = 1$ to $|\mathbf{x}|$ do
  $\mathbf{y}[i] \leftarrow \mathsf{op}_\mathcal{L}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
return $\mathcal{D}'^{\tau_\mathcal{L}, \mathsf{op}_\mathcal{L}}(\mathbf{y}, \bar{L})$

Proc. $\textsc{Exp}(x)$:
if $(rx \notin \mathsf{Dom}(T_{\tau_\mathcal{L}}))$ then
  $g \twoheadleftarrow \mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_{\tau_\mathcal{L}})$; $T_{\tau_\mathcal{L}}[rx] \leftarrow g$
return $T_{\tau_\mathcal{L}}[rx]$

Proc. $\mathsf{op}_\mathcal{L}(h_1, h_2)$:
for $i \in [2]$ do
  if $(h_i \notin \mathsf{Rng}(T_{\tau_\mathcal{L}}))$ then
    $x_i \twoheadleftarrow \mathbb{Z}_p \setminus \mathsf{Dom}(T_{\tau_\mathcal{L}})$; $T_{\tau_\mathcal{L}}[x_i] \leftarrow h_i$
  $x_i \leftarrow T_{\tau_\mathcal{L}}^{-1}[h_i]$
$x \leftarrow x_1 + x_2$
if $(x \notin \mathsf{Dom}(T_{\tau_\mathcal{L}}))$ then
  $g \twoheadleftarrow \mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_{\tau_\mathcal{L}})$; $T_{\tau_\mathcal{L}}[x] \leftarrow g$
return $T_{\tau_\mathcal{L}}[x]$

Proc. $\tau_\mathcal{L}(x)$:
if $(x \notin \mathsf{Dom}(T_{\tau_\mathcal{L}}))$ then
  $g \twoheadleftarrow \mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_{\tau_\mathcal{L}})$; $T_{\tau_\mathcal{L}}[x] \leftarrow g$
return $T_{\tau_\mathcal{L}}[x]$

Alg. $\mathfrak{A}(\tau)$:
$(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}'(\tau)$
return $(\mathbf{x}, \mathbf{m}, \bar{L})$

Alg. $\mathfrak{D}^{\tau,\tau^{-1}}(\mathbf{x}, \mathbf{m}, \bar{L})$:
$r \twoheadleftarrow \mathbb{Z}_p^*$
for $i = 1$ to $|\mathbf{x}|$ do
  $\mathbf{y}[i] \leftarrow \mathsf{op}(\mathbf{m}[i], \tau(r\mathbf{x}[i]))$
return $\mathcal{D}'^{\tau,\mathsf{op}}(\mathbf{y}, \bar{L})$

Proc. $\mathsf{op}(h_1, h_2)$:
$x_1 \leftarrow \tau^{-1}(h_1)$; $x_2 \leftarrow \tau^{-1}(h_2)$
return $\tau(x_1 + x_2)$

Figure 12 — *Top left:* The game Game_1 from the proof of Theorem 4.4. *Top right:* Reduction for the transition from Game_0 to Game_1. *Bottom left:* The game Game_2 from the proof of Theorem 4.4. *Bottom right:* Reduction for the transition from Game_1 to Game_2.

Game_9 is the same as Game_8, but we undo lazy sampling of σ and instead sample it all at once. Also, we again use Lemma 4.1 to undo resampling of τ.

Game_{10} is the same as Game_9, but we remove the constraint that all queries $\mathcal{S}$ makes are pairwise different. Doing so, we have obtained the PGG game with $b = 0$, i.e., where the oracle $\textsc{Exp}$ uses an independent encoding σ.

We now argue that the difference between the success probabilities of subsequent games are small.

$\mathsf{Game}_0 \rightsquigarrow \mathsf{Game}_1$. Denote by Bad_1 the event that in Game_0, there exist $i < j$ such that $\mathbf{x}[i] = \mathbf{x}[j]$, and observe that Game_0 and Game_1 are identical until Bad_1. To bound the probability of this event, consider the predictor $\mathcal{Q}'$ defined in Figure 12 (top right). One readily verifies that if Bad_1 occurs, then $\mathcal{Q}'$ wins the algebraic unpredictability game for source $\mathcal{S}'$, provided that $\mathcal{Q}'$ guesses a correct pair of indices where entries of $\mathbf{x}$ collide. Therefore,

$$\big|\Pr[\mathsf{Game}_0] - \Pr[\mathsf{Game}_1]\big| \le \Pr[\mathsf{Bad}_1] = \sum_{q=0}^{q_\mathcal{S}} \Pr[\mathsf{Bad}_1 \mid |\mathbf{x}| = q] \Pr[|\mathbf{x}| = q]$$
$$\le \sum_{q=0}^{q_\mathcal{S}} \binom{q}{2} \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S}',\mathcal{Q}'} \cdot \Pr[|\mathbf{x}| = q] \le \binom{q_\mathcal{S}}{2} \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S}',\mathcal{Q}'} \le T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{Q}} ,$$

where $\mathcal{Q}$ is the predictor constructed from $\mathcal{Q}'$ as in Lemma 4.2.

$\mathsf{Game}_1 \rightsquigarrow \mathsf{Game}_2$. Consider algorithms $\mathfrak{A}$ and $\mathfrak{D}$ defined in Figure 12 (bottom right). One readily verifies that $\text{AI-GG}^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S}} = \mathsf{Game}_1$, and $\Pr\big[\text{BF-GG}^{\mathfrak{A},\mathfrak{D}}_{p,\mathsf{S},\gamma,P}\big] = \Pr[\mathsf{Game}_2]$ since the only difference between these two games is that the former samples $\tau_\mathcal{L}$ at once, while the latter implements it via lazy sampling. Using Lemma 4.1, and noticing that $\mathfrak{D}$ makes no more than $3T$ queries to its oracles, we therefore obtain

$$\big|\Pr[\mathsf{Game}_1] - \Pr[\mathsf{Game}_2]\big| \le \frac{6(S - \log\gamma)T}{P} + \gamma \,.$$

$\mathsf{Game}_2 \rightsquigarrow \mathsf{Game}_3$. Consider the following three events: (1) Let Bad_2 be the event that, in Game_2, there exists i such that $r\mathbf{x}[i] \in \{\tilde{x}_1, \ldots, \tilde{x}_k\}$; (2) Let Bad_2' be the event that, at the end of the execution of $\mathcal{D}'$ in Game_3, there are polynomials $Q_1(R), Q_2(R) \in \mathsf{Dom}(T_{\tau_\mathcal{L}})$ such that $Q_1(R) \neq Q_2(R)$, but $Q_1(r) = Q_2(r)$ for a randomly chosen value $r \twoheadleftarrow \mathbb{Z}_p^*$; and (3) Let Bad_2'' be the event that, at the end of the execution of $\mathcal{D}'$ in Game_3, there is a polynomial $Q(R) \in \mathsf{Dom}(T_{\tau_\mathcal{L}})$ such that $Q(r) \in \{\tilde{x}_1, \ldots, \tilde{x}_k\}$ for a randomly chosen $r \twoheadleftarrow \mathbb{Z}_p^*$. Then notice that Game_2 and Game_3 are identical if none of these events occurs, and therefore $|\Pr[\mathsf{Game}_2] - \Pr[\mathsf{Game}_3]| \leq \Pr[\mathsf{Bad}_2] + \Pr[\mathsf{Bad}_2'] + \Pr[\mathsf{Bad}_2'']$. We bound each of these probabilities separately. For $\Pr[\mathsf{Bad}_2]$, notice that there are at most $q_\mathcal{S}$ terms $r\mathbf{x}[i]$, each of which can collide with at most P values $\tilde{x}_j$. The probability of each collision happening is $1/(p-1)$ since r is random in $\mathbb{Z}_p^*$, and therefore

$$\Pr[\mathsf{Bad}_2] \leq \frac{q_\mathcal{S} P}{p-1} \leq \frac{2q_\mathcal{S} P}{p} \leq \frac{2TP}{p}\,.$$

Moving to $\Pr[\mathsf{Bad}_2']$, observe that for Bad_2' to occur at least one of Q_1 and Q_2 must be non-constant. After the execution of $\mathcal{D}'$, $\mathsf{Dom}(T_{\tau_\mathcal{L}})$ contains at most $2q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}}$ many non-constant polynomials and at most $P + 3q_\mathcal{S} + q_{\mathcal{D},\tau} + 3q_{\mathcal{D},\mathsf{op}}$ many entries that each of the polynomials can collide with when evaluated at random $r \in \mathbb{Z}_p^*$. Each collision happens with probability $1/(p-1)$ by the Schwartz–Zippel lemma, which means

$$\Pr[\mathsf{Bad}_2'] \leq \frac{(2q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}})(P + 3q_\mathcal{S} + q_{\mathcal{D},\tau} + 3q_{\mathcal{D},\mathsf{op}})}{p-1} \leq \frac{2T(P+3T)}{p-1} \leq \frac{4T(P+3T)}{p} \leq \frac{12T(P+T)}{p}\,.$$

Finally, to bound $\Pr[\mathsf{Bad}_2'']$ notice that after the execution of $\mathcal{D}'$, $\mathsf{Dom}(T_{\tau_\mathcal{L}})$ contains at most $2q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}}$ many non-constant polynomials, each of which can collide with one of at most P elements $\tilde{x}_j$. Again, each collision happens with probability $1/(p-1)$ by the Schwartz–Zippel lemma, which yields

$$\Pr[\mathsf{Bad}_2''] \leq \frac{(2q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}})P}{p-1} \leq \frac{2TP}{p-1} \leq \frac{4TP}{p}\,.$$

Collecting the terms above, we obtain

$$\big|\Pr[\mathsf{Game}_2] - \Pr[\mathsf{Game}_3]\big| \leq \frac{2TP}{p} + \frac{12T(P+T)}{p} + \frac{4TP}{p} \leq \frac{18T(P+T)}{p}\,.$$

$\mathsf{Game}_3 \rightsquigarrow \mathsf{Game}_4$. We proceed in a similar way as above: Let Bad_3 be the event that, at the end of the execution of $\mathcal{D}'$ in Game_4, there are polynomials $Q_1(Z_1, \ldots, Z_q), Q_2(Z_1, \ldots, Z_q) \in \mathsf{Dom}(T_{\tau_\mathcal{L}})$ such that $Q_1 \neq Q_2$, but $Q_1(R\mathbf{x}[1], \ldots, R\mathbf{x}[q]) = Q_2(R\mathbf{x}[1], \ldots, R\mathbf{x}[q])$ as polynomials in R, where $q = |\mathbf{x}|$. Then Game_3 and Game_4 are identical if Bad_3 does not occur, which means $|\Pr[\mathsf{Game}_3] - \Pr[\mathsf{Game}_4]| \leq \Pr[\mathsf{Bad}_3]$. To bound the latter probability, observe that we can turn any collision as above into a non-trivial algebraic relation involving $\mathbf{x}$. Indeed, if $Q_i = \sum_{j=1}^q \alpha_{i,j} Z_j + \delta_i$, then having a collision means $\sum_{j=1}^q (\alpha_{1,j} - \alpha_{2,j})\mathbf{x}[j] = 0$, and since $Q_1 \neq Q_2$ we must have $\alpha_{1,j} \neq \alpha_{2,j}$ for some $j \in [q_\mathcal{S}]$. Therefore, consider the predictor $\mathcal{R}'$ in Figure 13 (bottom). One readily verifies that if Bad_3 occurs, then $\mathcal{R}'$ wins the game Game' in Figure 11 (top right) for source $\mathcal{S}'$. Using Lemmata 4.2 and 4.3, and observing that table T' contains at most $q_\mathcal{S} + q_{\mathcal{D}',\mathsf{op}}$ many entries, we obtain

$$\begin{aligned}\big|\Pr[\mathsf{Game}_3] - \Pr[\mathsf{Game}_4]\big| \leq \Pr[\mathsf{Bad}_3] = \Pr[\mathsf{Game}'] &\leq \Pr\big[\mathrm{MAlgPred}^{\mathcal{R}'}_{p,\mathsf{S},\mathcal{S}'}\big] + \frac{6(S - \log\gamma)T}{P} + \gamma \\ &\leq \binom{q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}}}{2} \cdot \mathrm{Adv}^{\text{alg-pred}}_{\mathsf{G},\mathcal{S}',\mathcal{R}'} + \frac{6(S - \log\gamma)T}{P} + \gamma \leq T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{R}} + \frac{6(S - \log\gamma)T}{P} + \gamma\,,\end{aligned}$$

where $\mathcal{R}$ is the predictor constructed from $\mathcal{R}'$ as in Lemma 4.2.

Game Game_3:
$\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$; $(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}'(\tau)$; $T_\sigma \leftarrow [\,]$
if $\big((\exists 1 \le i < j \le |\mathbf{x}|)(\mathbf{x}[i] = \mathbf{x}[j])\big)$ then return 0
$T_{\tau_\mathcal{L}} = [(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_k, \tilde{y}_k)] \leftarrow \textsc{Decomp}_{\bar{\mathcal{S}}'}(\mathsf{Rng}(\tau), \gamma, P, (\mathbf{x}, \mathbf{m}, \bar{L}))$
for $i = 1$ to $|\mathbf{x}|$ do $\mathbf{y}[i] \leftarrow \mathsf{op}_\mathcal{L}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
return $\mathcal{D}'^{\tau_\mathcal{L}, \mathsf{op}_\mathcal{L}}(\mathbf{y}, \bar{L})$

Proc. $\textsc{Exp}(x)$:
if $(x \notin \mathsf{Dom}(T_\sigma))$ then
  $g \twoheadleftarrow \mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_{\tau_\mathcal{L}})$
  $T_\sigma[x] \leftarrow g$; $T_{\tau_\mathcal{L}}[Rx] \leftarrow g$
return $T_\sigma[x]$

Game Game_4:
$\tau \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{S})$; $(\mathbf{x}, \mathbf{m}, \bar{L}) \twoheadleftarrow \bar{\mathcal{S}}'(\tau)$; $T_\sigma \leftarrow [\,]$; $\boxed{j \leftarrow 0}$
if $\big((\exists 1 \le i < j \le |\mathbf{x}|)(\mathbf{x}[i] = \mathbf{x}[j])\big)$ then return 0
$T_{\tau_\mathcal{L}} = [(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_k, \tilde{y}_k)] \leftarrow \textsc{Decomp}_{\bar{\mathcal{S}}'}(\mathsf{Rng}(\tau), \gamma, P, (\mathbf{x}, \mathbf{m}, \bar{L}))$
for $i = 1$ to $|\mathbf{x}|$ do $\mathbf{y}[i] \leftarrow \mathsf{op}_\mathcal{L}(\mathbf{m}[i], \textsc{Exp}(\mathbf{x}[i]))$
return $\mathcal{D}'^{\tau_\mathcal{L}, \mathsf{op}_\mathcal{L}}(\mathbf{y}, \bar{L})$

Proc. $\textsc{Exp}(x)$:
$\boxed{j \leftarrow j + 1}$
if $(x \notin \mathsf{Dom}(T_\sigma))$ then
  $g \twoheadleftarrow \mathsf{Rng}(\tau) \setminus \mathsf{Rng}(T_{\tau_\mathcal{L}})$
  $T_\sigma[x] \leftarrow g$; $\boxed{T_{\tau_\mathcal{L}}[Z_j] \leftarrow g}$
return $T_\sigma(x)$

Predictor $\mathcal{R}'^{\tau, \mathsf{op}}(\mathbf{y}, \bar{L})$:
$q \leftarrow |\mathbf{y}|$; $T' \leftarrow [(Z_1, \mathbf{y}[1]), \ldots, (Z_q, \mathbf{y}[q])]$
$b \twoheadleftarrow \mathcal{D}'^{\tau, \mathsf{op}'}(\mathbf{y}, \bar{L})$
$\{\sum_{i=1}^q \alpha_{j,i} Z_i \mid j \in [\kappa]\} \leftarrow \mathsf{Dom}(T')$
return $(\alpha_{j,1} - \alpha_{k,1}, \ldots, \alpha_{j,q} - \alpha_{k,q})_{j<k}$

Proc. $\mathsf{op}'(h_1, h_2)$:
$h \leftarrow \mathsf{op}(h_1, h_2)$
for $i \in [2]$ do
  $x_i \leftarrow T'^{-1}(h_i)$; if $(x_i = \perp)$ then $x_i \leftarrow 0$
if $(x_1 \neq 0) \vee (x_2 \neq 0)$ then $T'[x_1 + x_2] \leftarrow h$
return h

Figure 13 — *Top and Center:* The games Game_3 and Game_4 from the proof of Theorem 4.4. The oracles $\mathsf{op}_\mathcal{L}$ and $\tau_\mathcal{L}$ are the same as in Game_2. *Bottom:* Reduction for the transition from Game_3 to Game_4.

$\mathsf{Game}_4 \rightsquigarrow \mathsf{Game}_5$. Let Bad_4 be the event that, at the end of the execution of $\mathcal{D}'$ in Game_4, there are polynomials $Q_1(Z_1, \ldots, Z_q), Q_2(Z_1, \ldots, Z_q) \in \mathsf{Dom}(T_{\tau_\mathcal{L}})$ such that $Q_1 \neq Q_2$, but $Q_1(c_1, \ldots, c_q) = Q_2(c_1, \ldots, c_q)$ for randomly sampled $c_1, \ldots, c_q \twoheadleftarrow \mathbb{Z}_p$, where $q = |\mathbf{x}|$. Then Game_4 and Game_5 are equivalent until Bad_4. As before, $\mathsf{Dom}(T_{\tau_\mathcal{L}})$ will contain at most $2q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}}$ polynomials at the end of the game, with each pair resulting in a collision with probability at most $1/p$ by the Schwartz–Zippel lemma (since $\deg(Q_i) \le 1$). Hence,

$$\big|\Pr[\mathsf{Game}_4] - \Pr[\mathsf{Game}_5]\big| \le \Pr[\mathsf{Bad}_4] \le \binom{2q_\mathcal{S} + q_{\mathcal{D},\mathsf{op}}}{2} \cdot \frac{1}{p} \le \frac{2T^2}{p}\,.$$

$\mathsf{Game}_5 \rightsquigarrow \mathsf{Game}_6$. Let Bad_5 be the event that, for some call to $\textsc{Exp}$ in Game_5, the sampled value c satisfies $c \in \mathsf{Dom}(T_{\tau_\mathcal{L}})$. Again, Game_5 and Game_6 are equivalent until Bad_5, and a straightforward calculation shows that

$$\begin{aligned}\big|\Pr[\mathsf{Game}_5] - \Pr[\mathsf{Game}_6]\big| \le \Pr[\mathsf{Bad}_5] &= \sum_{q=0}^{q_\mathcal{S}} \Pr[\mathsf{Bad}_5 \mid |\mathbf{x}| = q] \Pr[|\mathbf{x}| = q] \\ &\le \sum_{q=1}^{q_\mathcal{S}} \left(\frac{P+1}{p} + \frac{P+3}{p} + \cdots + \frac{P+2q-1}{p}\right) \Pr[|\mathbf{x}| = q] \le \frac{q_\mathcal{S}(P+q_\mathcal{S})}{p} \le \frac{T(P+T)}{p}\,.\end{aligned}$$

$\mathsf{Game}_6 \rightsquigarrow \mathsf{Game}_7$. We claim that $\mathsf{Game}_6 = \mathsf{Game}_7$, which in particular implies $\Pr[\mathsf{Game}_6] = \Pr[\mathsf{Game}_7]$. Indeed, the only difference in the pseudocode of the two games is that Game_6 explicitly populates table $T_{\tau_\mathcal{L}}$ in the $\textsc{Exp}$ oracle, while Game_7 does not. On the other hand, in Game_7 this is done implicitly by the $\mathsf{op}_\mathcal{L}$-procedure, with exactly the same distribution as specified in the $\textsc{Exp}$ oracle of Game_6, since g is always picked in the complement of $\mathsf{Rng}(T_{\tau_\mathcal{L}})$.

$\mathsf{Game}_7 \rightsquigarrow \mathsf{Game}_8$. Define an intermediate game $\mathsf{Game}_{7.5}$ which is the same as Game_7, but where the $\textsc{Exp}$ oracle draws g at random from $\mathsf{Rng}(T_\tau)$. Denote by Bad_6 and Bad'_6 the events that, for some call to $\textsc{Exp}$ in game $\mathsf{Game}_{7.5}$, the random element g satisfies $g \in \mathsf{Rng}(T_{\tau_\mathcal{L}})$ (resp., $g \in \mathsf{Rng}(T_\sigma)$). Then Game_7 and $\mathsf{Game}_{7.5}$

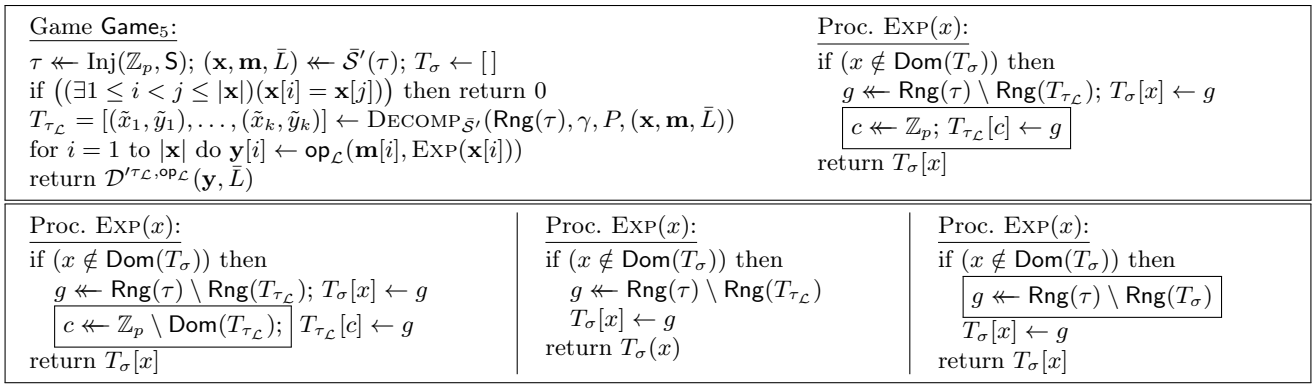

Figure 14 — *Top:* The game Game_5 from the proof of Theorem 4.4. *Bottom:* Exponentiation oracles of the games Game_6, Game_7, and Game_8. The pseudocode of the corresponding games is the same as Game_5. In all games, the oracles $\mathsf{op}_\mathcal{L}$ and $\tau_\mathcal{L}$ are the same as in Game_2.

are equivalent until Bad_6, and $\mathsf{Game}_{7.5}$ and Game_8 are equivalent until Bad'_6. An argument similar to the one for Bad_5 now yields

$$\big|\Pr[\mathsf{Game}_7] - \Pr[\mathsf{Game}_8]\big| \le \Pr[\mathsf{Bad}_6] + \Pr[\mathsf{Bad}'_6] \le \frac{q_\mathcal{S}(P + q_\mathcal{S})}{p} + \frac{q_\mathcal{S}^2}{2p} \le \frac{T(P+T)}{p} + \frac{T^2}{p}\,.$$

$\mathsf{Game}_8 \rightsquigarrow \mathsf{Game}_9$ and $\mathsf{Game}_9 \rightsquigarrow \mathsf{Game}_{10}$. Observe that we are essentially undoing the first two transitions. As a result, arguments similar to those given at the beginning of the proof apply here as well, and we have

$$\big|\Pr[\mathsf{Game}_8] - \Pr[\mathsf{Game}_9]\big| \le \frac{6(S - \log\gamma)T}{P} + \gamma\,, \qquad \big|\Pr[\mathsf{Game}_9] - \Pr[\mathsf{Game}_{10}]\big| \le T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{Q}}\,.$$

Collecting all the terms above and setting $\gamma = 1/p$, we obtain the following bound:

$$\begin{aligned}\mathrm{Adv}^{\text{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} &\le 2T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{Q}} + T^2 \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{R}} + \frac{18(S + \log p)T}{P} + \frac{20T(P+T)}{p} + 3\left(\frac{T^2+1}{p}\right) \\ &\le 3T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} + \frac{36ST}{P} + \frac{20TP}{p} + \frac{26T^2}{p} \le 3T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} + \frac{26T^2}{p} + 36T\left(\frac{S}{P} + \frac{P}{p}\right),\end{aligned}$$

where $\mathcal{P} \in \{\mathcal{Q}, \mathcal{R}\}$ is the predictor with the larger advantage. Now recall that this inequality holds for every $P \in \mathbb{N}$, which means that we can set $P :\approx \sqrt{Sp}$ to minimize the term on the right. Overall we obtain

$$\begin{aligned}\mathrm{Adv}^{\text{pgg}}_{p,\mathsf{S},\mathcal{S},\mathcal{D}} &\le 3T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} + \frac{26T^2}{p} + 72T\sqrt{\frac{S}{p}} \overset{\text{(a)}}{\le} 3T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} + 98T\sqrt{\frac{S}{p}} \\ &\le 100\left(T^2 \cdot \mathrm{Adv}^{\text{alg-pred}}_{p,\mathsf{S},\mathcal{S},\mathcal{P}} + \sqrt{\frac{ST^2}{p}}\right),\end{aligned}$$

where Inequality (a) holds by our assumption that $T \le \sqrt{Sp}$. □

Uber with preprocessing. Looking ahead, we observe that Uber and Uber-II sources are statistically algebraically unpredictable (even in the GGM) and, in particular, their algebraic unpredictability bound does not depend on the number of queries made by the predictor to the generic group oracles. This in turn

implies that when the above theorem is applied to the Uber and Uber-II sources (with q polynomials of degree at most d and at most T generic group and operation queries) we obtain

$$\mathrm{Adv}^{\text{dua-ii}}_{p,\mathcal{A}} \leq \tilde{\mathcal{O}}\left(\frac{d(T+q)^2}{p} + \sqrt{\frac{S(T+q)^2}{p}}\right)$$

in the GGM. When setting $q = 4$ and $d = 2$, the bound matches that established for the DDH problem [CK18, CDG18].

Discussion on DDH-II. As a second corollary, we obtain the hardness of the r-DDH-II assumption[11] in the GGM (here "r" stands for randomized generator). This result was also established by Bartusek, Ma, and Zhandry (BMZ) [BMZ19, Theorem 12]. Our proof, besides establishing the hardness of a winder class of assumption, is more modular and also avoids asymptotics. Furthermore, since our feasibility only relies on *computational* algebraic unpredictability, it can be applied in a setting where some group elements are directly leaked to the distinguisher.

5 From Simple to Algebraic Unpredictability: LDDs

We define a new type of hash function family called *linear-dependence destroyer* (LDD) that is useful for building schemes secure in PGGs. Intuitively, LDDs are hash functions with domain and range $\mathbb{Z}_p$ that remove, in a statistical sense, any linear dependence among a list of distinct but potentially correlated values.

Linear-Dependence Destroyers (LDDs). Let H be a hash function family with domain and range $\mathbb{Z}_p$ for some prime p. We define the advantage of a pair of adversaries $(\mathcal{S}, \mathcal{A})$ in the LDD game for H as

$$\mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{S},\mathcal{A}}(\lambda) \coloneqq \Pr\left[\mathrm{LDD}^{\mathcal{S},\mathcal{A}}_{\mathsf{H}}(\lambda)\right],$$

where the LDD game is defined in Figure 15 (top left). We require that the outputs of $\mathcal{S}$ be pairwise distinct and the output of $\mathcal{A}$ be different from the all-zero tuple. We say that H is LDD[$\mathbf{S}$] secure if the advantage of any $(\mathcal{S}, \mathcal{A})$ in the LDD game is negligible, with $\mathcal{S} \in \mathbf{S}$ and $\mathcal{A}$ an unbounded machine. We write this as $\mathsf{H} \in \mathrm{LDD}[\mathbf{S}]$.

We call an LDD source $\mathcal{S}$ statistically unpredictable if

$$\mathrm{Adv}^{\mathrm{pred}}_{\mathsf{H},\mathcal{S},\mathcal{P}}(\lambda) \coloneqq \Pr\left[\mathrm{Pred}^{\mathcal{P}}_{\mathsf{H},\mathcal{S}}(\lambda)\right]$$

is negligible for any (possibly unbounded) predictor $\mathcal{P}$, where the game Pred is defined in Figure 15 (top center). We denote the class of all statistically unpredictable LDD sources by $\mathbf{S}^{\mathrm{sup}}$. We say that H is an LDD if it is LDD[$\mathbf{S}^{\mathrm{sup}}$] secure.

In Section 5.1 we show that, for a computational group scheme Γ, the hash function family $\mathsf{H}[\Gamma]$ with domain and range $\mathbb{Z}_p$ defined in Figure 15 (bottom) is an LDD for the class of *low-degree sources* $\mathbf{S}^{\mathrm{low}}$. These are sources that compute their output as evaluations of low-degree polynomials on high-entropy points, as in Figure 15 (top right) – We refer the reader to Section 5.1 for formal definitions and proofs.

We were unable to prove that the construction in Figure 15 (bottom) is an LDD for *all* unpredictable sources, though we have not been able to break it either. We conjecture that LDDs exist for *all* statistically unpredictable sources, and not just for low-degree ones. More strongly, we conjecture that the hash function $\mathsf{H}[\Gamma]$ defined in Figure 15 (bottom) is LDD secure for all statistically unpredictable sources.

[11] Distinguish (g, g^x, g^y, g^{xy}) from (g, g^x, g^y, g^z) for a random generator g, unpredictable x, and random y and z.

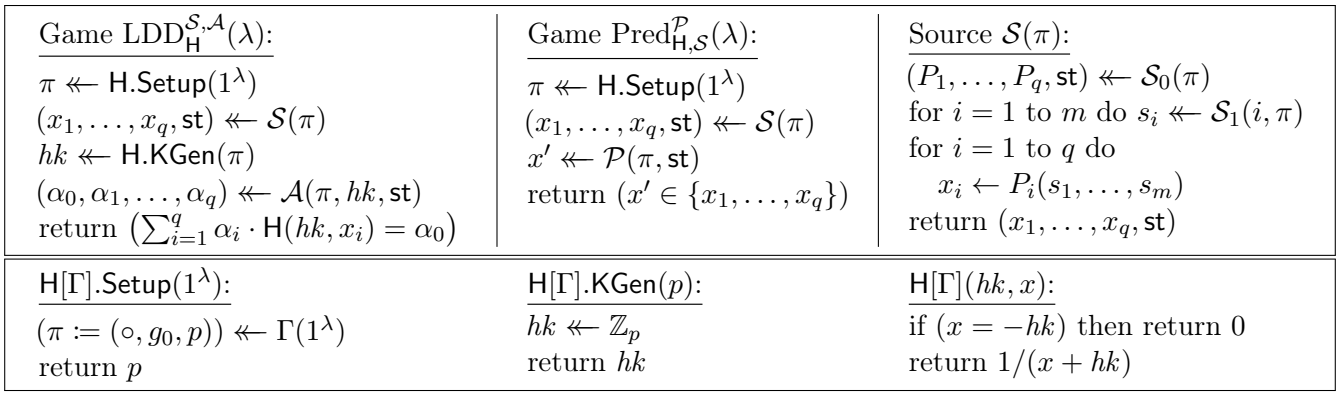

Figure 15 — *Top left:* The LDD game. *Top center:* The predictability game. *Top right:* Structure of a low-degree LDD source. *Bottom:* Candidate construction of an LDD family $\mathsf{H}[\Gamma]$ from a computational group scheme Γ.

We emphasize that LDD is an *information-theoretic* notion and thus unconditional constructions (as for randomness extractors) may exist. We note that positive results for smaller classes of sources are also meaningful, as they would translate, via our constructions and proofs, into deterministic PKE, UCEs, and RKA-secure encryption.

As evidence towards the first conjecture, we can easily prove that a RO $\rho\colon \mathbb{Z}_p^2 \to \mathbb{Z}_p$ is an LDD, if all algorithms only get polynomially bounded oracle access to ρ (rather than the entire function table). Assume that $\mathcal{A}$ makes at most n queries, and let E denote the event that $\mathcal{A}$ queries ρ on one of the points $(hk,x_1),\ldots,(hk,x_q)$. Then we can build a predictor $\mathcal{P}$ such that $\Pr[E] \le n \cdot \Pr[\mathrm{Pred}^{\mathcal{P}}_{\rho,\mathcal{S}}(p)]$, which is small by unpredictability of $\mathcal{S}$. If E does not occur, then the equation $\sum_{i=1}^q \alpha_i \cdot \mathsf{H}(hk,x_i) = \alpha_0$ is satisfied with probability at most $1/p$, because at least one coefficient α_j, $1 \le j \le q$, is non-zero, and thus the random-looking value $\mathsf{H}(hk,x_j)$ is determined by the winning condition.

We provide a stronger feasibility result for LDDs by showing that random functions are LDDs for any unpredictable source, even when *both* the source $\mathcal{S}$ and the adversary $\mathcal{A}$ have *full access* to the table of the random function.[12] This result would thus establish the existence of LDDs, similarly to that for other information-theoretic objects such as randomness extractors. At a very high level, we prove this result in two steps: First, we decompose arbitrary high-entropy sources into a convex combination of flat sources (i.e., sources that are uniform on subsets of the support of the distribution). This is a standard technique in the study of randomness extractors [Vad12]. Second, we apply a *compression-style* argument [GT00] to show that any predictor that has a high LDD-advantage against unpredictable sources can be used to compress the random function. The complete proof is rather technical, and we refer to Section 5.2 for a more detailed overview as well as the formal details.

5.1 LDDs for Low-Degree Sources

In this section we show that for a computational group scheme Γ, the hash function family $\mathsf{H}[\Gamma]$ with domain and range $\mathbb{Z}_p$ defined in Figure 15 (bottom) is an LDD for *low-degree sources*, a class of sources which we define below.

Low-degree sources. A (d,k)-source is a source $\mathcal{S}$ for which there exist PPT algorithms $\mathcal{S}_0$ and $\mathcal{S}_1$ such that $\mathcal{S}$ takes the form in Figure 15 (top right), where $\mathcal{S}_0$ outputs q non-constant polynomials in $\mathbb{Z}_p[X_1,\ldots,X_m]$ of total degree at most d and, for every $i \in \mathbb{N}$ and all public parameters π, $\mathcal{S}_1(i,\pi)$ is a k-source over $\mathbb{Z}_p$, with $k(\lambda) = \omega(\log\lambda)$. We denote the class of all such (d,k)-sources by $\mathbf{S}^{\mathrm{low}}$.

[12] Accordingly, we also impose a statistical notion of unpredictability on sources by giving predictors access to the full table.

CONSTRUCTION. Let Γ be a computational group scheme. We now prove that the hash function family $\mathsf{H}[\Gamma]$ defined in Figure 15 (bottom) is LDD[$\mathbf{S}^{\mathrm{low}}$] secure. This function was used to build a correlated-input secure hash function by Goyal, O'Neill and Rao [GOR11]. To this end, we start with a technical lemma about the set of roots of polynomials.

Lemma 5.1. *Let $m, p, \delta \in \mathbb{N}$ with p prime, and let $P \in \mathbb{Z}_p[X_1, \ldots, X_m]$ such that $d := \deg(P) \geq 1$. Denote by R the set of roots of P in $\mathbb{F}_{p^\delta}^m$. Then $|R| \leq dp^{\delta(m-1)}$. If in addition δ is such that all polynomials in one indeterminate over $\mathbb{Z}_p$ of degree at most d are split over $\mathbb{F}_{p^\delta}$, then $|R| \geq p^{\delta(m-1)}$.*

Proof. The first part of the statement follows from the Schwartz–Zippel lemma: Using Lemma 2.1 with $S := \mathbb{F}_{p^\delta}$, we obtain $|R|/p^{m\delta} \leq d/p^\delta$, i.e., $|R| \leq dp^{\delta(m-1)}$.

We prove the second part by induction on m. If $m = 1$, the lemma states that $|R| \geq p^{\delta(1-1)} = 1$. This is indeed true since P is non-constant and, by definition of δ, split over $\mathbb{F}_{p^\delta}$, which means that it has at least one root. Now let $m > 1$, and suppose that the statement holds for all $m' < m$. If P has less than m variables, the lemma follows from the induction hypothesis, so assume that this is not the case. For each $\gamma \in \mathbb{F}_{p^\delta}$, define $R_\gamma := \{(x_1, \ldots, x_m) \in R \mid x_m = \gamma\}$, and let T_γ be the set of roots of $P_\gamma := P(X_1, \ldots, X_{m-1}, \gamma)$ in $\mathbb{F}_{p^\delta}^{m-1}$. Then notice that $R = \bigsqcup_{\gamma \in \mathbb{F}_{p^\delta}} R_\gamma$, and that $R_\gamma = T_\gamma$ for every $\gamma \in \mathbb{F}_{p^\delta}$, which means

$$|R| = \sum_{\gamma \in \mathbb{F}_{p^\delta}} |R_\gamma| = \sum_{\gamma \in \mathbb{F}_{p^\delta}} |T_\gamma| \geq \sum_{\gamma \in \mathbb{F}_{p^\delta}} p^{\delta(m-2)} = p^{\delta(m-1)} .$$

Here, the inequality follows from the induction hypothesis, by observing that P_γ is a non-constant polynomial in $m - 1 < m$ variables of degree at most d. □

The main step of our proof is contained in the following result, saying that two different low-degree polynomials with random constant term are likely to be coprime. This allows us to prove in the next lemma that the numerator of any non-trivial linear combination $\sum_{i=1}^n \alpha_i/(P_i + hk) - \alpha_0$ is non-zero. LDD security of $\mathsf{H}[\Gamma]$ for low-degree sources then follows from Lemma 2.2.

Lemma 5.2. *Let $d, m, p \in \mathbb{N}$ with p prime, and let $P_1, P_2 \in \mathbb{Z}_p[X_1, \ldots, X_m]$ be such that $P_1 \neq P_2$ and $\deg(P_1, P_2) \leq d$. Then we have:*

$$\Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}[\gcd(P_1 + \alpha, P_2 + \alpha) \neq 1] \leq \frac{d}{p} .$$

Proof. Let $\delta \in \mathbb{N}$ be such that all polynomials in one indeterminate over $\mathbb{Z}_p$ of degree at most d are split over $\mathbb{F}_{p^\delta}$. Define $Q := P_1 - P_2$, and denote by R the set of its roots in $\mathbb{F}_{p^\delta}^m$. By assumption, Q is non-zero and of degree at most d, which means that $|R| \leq dp^{\delta(m-1)}$ by Lemma 5.1. Similarly, for every $\alpha \in \mathbb{Z}_p$ define $Q_\alpha := \gcd(P_1 + \alpha, P_2 + \alpha)$, and denote by R_α the set of its roots in $\mathbb{F}_{p^\delta}^m$. Again by Lemma 5.1, we know that $Q_\alpha \neq 1$ if and only if $|R_\alpha| \geq p^{\delta(m-1)}$. We can now bound the probability in the statement of the lemma as follows:

$$\begin{aligned}
\Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}[\gcd(P_1 + \alpha, P_2 + \alpha) \neq 1] &= \Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}\left[|R_\alpha| \geq p^{\delta(m-1)}\right] = \Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}\left[\sum_{\mathbf{r} \in \mathbb{F}_{p^\delta}^m} \mathbb{1}_{R_\alpha}(\mathbf{r}) \geq p^{\delta(m-1)}\right] \\
&\overset{(a)}{=} \Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}\left[\sum_{\mathbf{r} \in R} \mathbb{1}_{R_\alpha}(\mathbf{r}) \geq p^{\delta(m-1)}\right] \overset{(b)}{\leq} \frac{1}{p^{\delta(m-1)}} \mathop{\mathbb{E}}_{\alpha \twoheadleftarrow \mathbb{Z}_p}\left[\sum_{\mathbf{r} \in R} \mathbb{1}_{R_\alpha}(\mathbf{r})\right] = \frac{1}{p^{\delta(m-1)}} \sum_{\mathbf{r} \in R} \mathop{\mathbb{E}}_{\alpha \twoheadleftarrow \mathbb{Z}_p}[\mathbb{1}_{R_\alpha}(\mathbf{r})] \\
&= \frac{1}{p^{\delta(m-1)}} \sum_{\mathbf{r} \in R} \Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}[\mathbf{r} \in R_\alpha] = \frac{1}{p^{\delta(m-1)}} \sum_{\mathbf{r} \in R} \Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}[P_1(\mathbf{r}) = P_2(\mathbf{r}) = \alpha]
\end{aligned}$$

Game Game:	Game Game':
for $i = 1$ to m do $s_i \twoheadleftarrow \mathcal{S}_1(i, \pi)$	for $i = 1$ to m do $s_i \twoheadleftarrow \mathcal{S}_1(i, \pi)$
$\mathbf{s} \leftarrow (s_1, \ldots, s_m)$; $hk \twoheadleftarrow \mathbb{Z}_p$	$\mathbf{s} \leftarrow (s_1, \ldots, s_m)$; $hk \twoheadleftarrow \mathbb{Z}_p$
$(\alpha_0, \alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{A}(\pi, hk, \mathsf{st})$	$(\alpha_0, \alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{A}(\pi, hk, \mathsf{st})$; $\frac{N_{hk}}{D_{hk}} \coloneqq \sum_{i=1}^{q} \frac{\alpha_i}{P_i + hk}$
return $\left(\sum_{i=1}^{q} \frac{\alpha_i}{P_i(\mathbf{s}) + hk} = \alpha_0\right)$	return $(N_{hk}(\mathbf{s}) - \alpha_0 \cdot D_{hk}(\mathbf{s}) = 0)$

Figure 16 — The games Game and Game' from the proof of Theorem 5.4.

$$\leq \frac{1}{p^{\delta(m-1)}} \sum_{\mathbf{r} \in R} \Pr_{\alpha \twoheadleftarrow \mathbb{Z}_p}[P_1(\mathbf{r}) = \alpha] = \frac{1}{p^{\delta(m-1)}} \sum_{\mathbf{r} \in R} \frac{1}{p} = \frac{|R|}{p^{\delta(m-1)+1}} \overset{(c)}{\leq} \frac{d p^{\delta(m-1)}}{p^{\delta(m-1)+1}} = \frac{d}{p}\,.$$

Here, $\mathbb{1}_{R_\alpha}(\mathbf{r})$ is the function which returns 1 if $\mathbf{r} \in R_\alpha$, and 0 otherwise. Equality (a) holds because $R_\alpha \subseteq R$ for every $\alpha \in \mathbb{Z}_p$. Indeed, if $\mathbf{x} \in R_\alpha$ we have $(P_1 + \alpha)(\mathbf{x}) = (P_2 + \alpha)(\mathbf{x}) = 0$, which means that $(P_1 - P_2)(\mathbf{x}) = 0$, and thus $\mathbf{x} \in R$. Step (b) uses Markov's inequality, and Inequality (c) is the estimate on $|R|$ given above. □

Lemma 5.3. *Let $m, p, q \in \mathbb{N}$ with p prime, and $0 \neq P_1, \ldots, P_q \in \mathbb{Z}_p[X_1, \ldots, X_m]$ be such that at most one of them is constant and, for all $1 \leq i < j \leq q$, $\gcd(P_i, P_j) = 1$. For each $i \in [q]$ let $d_i \coloneqq \deg(P_i)$, and define $d \coloneqq \sum_{i=1}^{q} d_i$. Then, for every $\alpha_1, \ldots, \alpha_q \in \mathbb{Z}_p$ not all zero, there exist two non-zero polynomials $N, D \in \mathbb{Z}_p[X_1, \ldots, X_n]$ such that $\sum_{i=1}^{q} \frac{\alpha_i}{P_i} = \frac{N}{D}$, $\deg(N) \leq d - \min_i d_i$, and $\deg(D) \leq d$.*

Proof. For every $i \in [q]$ define the polynomial $L_i \coloneqq \prod_{1 \leq j \leq q,\, j \neq i} P_i$. If we now set $N \coloneqq \sum_{i=1}^{q} \alpha_i \cdot L_i$ and $D \coloneqq \prod_{i=1}^{q} P_i$, we obtain

$$\sum_{i=1}^{q} \frac{\alpha_i}{P_i(X_1, \ldots, X_m)} = \frac{\sum_{i=1}^{q} \alpha_i \cdot L_i(X_1, \ldots, X_m)}{\prod_{i=1}^{q} P_i(X_1, \ldots, X_m)} = \frac{N(X_1, \ldots, X_m)}{D(X_1, \ldots, X_m)}\,.$$

We now prove that $N \neq 0$. Suppose by contradiction that $N = 0$; we show that this implies $\alpha_i = 0$ for every $i \in [q]$. Fix any $i \in [q]$. If P_i is constant, then notice that $\alpha_i L_i$ is the summand of highest degree in N (because none of the other P_j is constant by assumption), which forces $\alpha_i = 0$. Suppose on the other hand that P_i is not constant. Notice that for every $j \neq i$, we have $P_i \mid L_j$ and thus $N \equiv \alpha_i L_i \pmod{P_i}$, so that $N = 0$ would mean $P_i \mid \alpha_i L_i$. Given that L_i is the product of polynomials coprime with P_i, we must have $\gcd(L_i, P_i) = 1$, and thus $P_i \mid \alpha_i$. Since α_i is a scalar, it must be $\alpha_i = 0$.

We now estimate the degrees of N and D. The bound on $\deg(D)$ is clear. For the numerator, notice that each L_i has degree $d - d_i$. Thus, the degree of N is at most $\max_i \deg(L_i) = \max_i(d - d_i) = d - \min_i d_i$. □

Theorem 5.4 (LDD for low-degree sources)**.** *Let Γ be a computational group scheme and $\mathsf{H}[\Gamma]$ the hash function family defined in Figure 15 (bottom). Then $\mathsf{H}[\Gamma] \in \mathrm{LDD}[\mathbf{S}^{\mathrm{low}}]$. More precisely, for any adversary $(\mathcal{S}, \mathcal{A})$ in the LDD game with $\mathcal{S}$ a (d, k)-source we have*

$$\mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H}[\Gamma], \mathcal{S}, \mathcal{A}}(\lambda) \leq \frac{q(\lambda) d(\lambda)}{2^{k(\lambda)}} + \frac{d(\lambda) q(\lambda)(q(\lambda) + 1)}{2^{\lambda}}\,. \tag{5}$$

Here, $d(\lambda)$ and $q(\lambda)$ are upper bounds on the degree and number of polynomials returned by $\mathcal{S}_0$, respectively, and $k(\lambda)$ is a lower bound on $\mathbf{H}_\infty(\mathcal{S}_1(i, \pi))$ for every $i \in \mathbb{N}$ and all public parameters $\pi \twoheadleftarrow \mathsf{H.Setup}(1^\lambda)$.

Proof. For every π and $(P_1, \ldots, P_q, \mathsf{st})$ returned by Γ and $\mathcal{S}_0(\pi)$, respectively, denote by $\mathsf{Game} \coloneqq \mathsf{Game}(\lambda, \pi, P_1, \ldots, P_q, \mathsf{st})$ the LDD game for $\mathsf{H}[\Gamma]$ played by $(\mathcal{S}, \mathcal{A})$ with these parameters fixed, as shown in Figure 16 (left). This means

$$\mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H}[\Gamma], \mathcal{S}, \mathcal{A}}(\lambda) = \sum_{\pi} \sum_{(P_1, \ldots, P_q, \mathsf{st})} \Pr[\mathsf{Game}] \Pr[(\Gamma(1^\lambda) = \pi) \wedge (\mathcal{S}_0(\pi) = (P_1, \ldots, P_q, \mathsf{st}))]\,.$$

Game $\mathrm{LDD}^{\mathcal{S},\mathcal{A}}_{\mathrm{RF}}(p)$:	Game $\mathrm{Pred}^{\mathcal{P}}_{\mathrm{RF},\mathcal{S}}(p)$:
$\pi \twoheadleftarrow \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$; $(x_1, \ldots, x_q, \mathsf{st}) \twoheadleftarrow \mathcal{S}(\pi)$; $hk \twoheadleftarrow \mathbb{Z}_p$ $(\alpha_0, \alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{A}(\pi, hk, \mathsf{st})$ return $\left(\sum \alpha_i \cdot \pi(hk, x_i) = \alpha_0\right)$	$\pi \twoheadleftarrow \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$; $(x_1, \ldots, x_q, \mathsf{st}) \twoheadleftarrow \mathcal{S}(\pi)$ $x' \twoheadleftarrow \mathcal{P}(\pi, \mathsf{st})$ return $(x' \in \{x_1, \ldots, x_q\})$

Figure 17 — *Left:* The LDD game with respect to a random function. *Right:* The predictability game with respect to a random function.

Fix any such π and $(P_1, \ldots, P_q, \mathsf{st})$, consider the corresponding game Game, and define $M \coloneqq \sum_{i<j} \deg(P_i, P_j)$. For every $hk \in \mathbb{Z}_p$ and $i, j \in [q]$ with $i \neq j$, define the events $E_{i,j,hk} \coloneqq (\gcd(P_i + hk, P_j + hk) = 1)$, and set $E_{hk} \coloneqq \bigcap_{1 \le i < j \le q} E_{i,j,hk}$. Then we have:

$$\begin{aligned}
\Pr[\mathsf{Game}] &\le \Pr[\mathsf{Game} \mid E_{hk}] + \Pr[\neg E_{hk}] = \Pr[\mathsf{Game} \mid E_{hk}] + \Pr\left[\bigcup_{i<j} \neg E_{i,j,hk}\right] \\
&\le \Pr[\mathsf{Game} \mid E_{hk}] + \sum_{1 \le i < j \le q} \Pr[\neg E_{i,j,hk}] \overset{(a)}{\le} \Pr[\mathsf{Game} \mid E_{hk}] + \frac{M}{p} \overset{(b)}{=} \Pr[\mathsf{Game}' \mid E_{hk}] + \frac{M}{p} \\
&\overset{(c)}{\le} \frac{q(\lambda) d(\lambda)}{2^{\min_{i \in [m]} \{\mathbf{H}_\infty(\mathcal{S}_1(i,\pi))\}}} + \frac{M}{p} \overset{(d)}{\le} \frac{q(\lambda) d(\lambda)}{2^{k(\lambda)}} + \frac{d(\lambda) q(\lambda)(q(\lambda)+1)}{2^\lambda} .
\end{aligned}$$

Here, Inequality (a) follows from Lemma 5.2. For Equality (b), notice that Game' as defined in Figure 16 (right) is the same as Game, but with a reformulated winning condition. In this game, N_{hk} and D_{hk} are defined as in Lemma 5.3, and $N_{hk} - \alpha_0 D_{hk}$ is a polynomial of degree at most qd. Finally, Inequality (c) follows from the game-based Schwartz–Zippel lemma (Lemma 2.2), and Step (d) uses the lower bound on the entropy of $\mathcal{S}_1(i, \pi)$, that $M \le dq(q+1)/2$, and that $p \ge 2^{\lambda - 1}$. □

5.2 Random Functions are LDDs

In this section we show further evidence towards the above conjecture by showing that a random function is LDD for all (possibly unbounded) sources. We first prove, via a compression argument, that a random function is LDD with respect to flat high-entropy sources.[13]

Unpredictable sources. Let $\mathcal{S}$ be a source. For $U \ge 1$ we say that $\mathcal{S}$ is U-unpredictable if, for every algorithm $\mathcal{P}$, $\Pr\left[\mathrm{Pred}^{\mathcal{P}}_{\mathrm{RF},\mathcal{S}}(p)\right] \le 1/U$, where the game Pred is defined in Figure 17 (right).

Lemma 5.5. *For every $x \in \mathbb{R}_{>0}$, denote by $\chi(x) \coloneqq \max(\mathbb{R}_{\le x} \cap \log(\mathbb{N}))$. Then $2^x - 1 \le 2^{\chi(x)} \le 2^x$.*

Proof. Fix any $x > 0$. The second inequality follows from $\chi(x) \le x$, which holds by definition. For the first inequality assume, for sake of contradiction, that $2^{\chi(x)} < 2^x - 1$, and let $y \coloneqq \log(2^{\chi(x)} + 1)$. We claim that $y \in \mathbb{R}_{\le x} \cap \log(\mathbb{N})$ and $\chi(x) < y$, thereby contradicting the maximality of $\chi(x)$. Indeed, $2^{\chi(x)} < 2^x - 1$ implies

$$\chi(x) = \log(2^{\chi(x)}) < \log(2^{\chi(x)} + 1) < \log(2^x) = x \,,$$

which shows that $\chi(x) < y < x$. To conclude, notice that $2^{\chi(x)} \in \mathbb{N}$ since $\chi(x) \in \log(\mathbb{N})$, and therefore also $y = \log(2^{\chi(x)} + 1) \in \log(\mathbb{N})$. □

[13] We were not able to prove this using the simpler bit-fixing method [CDGS18].

Theorem 5.6 (Random functions are LDDs)**.** *Let $p > 2$ be prime and let $\epsilon, U \in \mathbb{R}$. Let $\mathcal{S}$ be a U-unpredictable source, such that the number q of points it returns depends only on p, and let $\mathcal{A}$ be an algorithm with $\Pr\left[\mathrm{LDD}^{\mathcal{S},\mathcal{A}}_{\mathrm{RF}}(p)\right] \geq \epsilon$, where the game $\mathrm{LDD}^{\mathcal{S},\mathcal{A}}_{\mathrm{RF}}(p)$ is defined in Figure 17 (left). Assume that:*

$$\max\left(\frac{2^{q+3}}{p}, \frac{8}{\log(p)} + \frac{2^{q-1}}{p} + \frac{2(q+1)}{\sqrt{p}}\right) < \epsilon \leq 1\,, \qquad U \geq \frac{32}{\epsilon^2}\,, \qquad Q \coloneqq 2^q < p\,.$$

Then:

$$U \leq \frac{32}{\epsilon^2} \cdot \frac{\log(\epsilon) - 3 + \frac{\epsilon p}{2^{q-3}} - \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{8\cdot 2^{q-3}}\right)}{\log(p)\left((q+1) + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{2^q}\right) + \frac{8\epsilon p}{2^q} - 1}\,.$$

Proof Overview. To show that $\mathcal{S}$ cannot be too unpredictable, we will use it to give a compact description of many functions in $\mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$. Indeed, observe that any hk and any pair of vectors $\mathbf{x} \coloneqq (x_1, \ldots, x_q)$ and $(\alpha_0, \ldots, \alpha_q)$ returned by $\mathcal{S}$ and $\mathcal{A}$ define an affine equation with unknowns the function values $f(hk, x_i)$. As a result, we can encode these values $f(hk, x_i)$ as solutions of an affine system of equations, and they can be recovered correctly if there are enough equations so that the system has full rank. The key argument now is that each $\mathbf{x}$ can be used many times over, because the equation is true for many different vectors $(hk, \alpha_0, \ldots, \alpha_q)$. We can thus describe many equations compactly by first giving explicitly a large set F of $(hk, \alpha_0, \ldots, \alpha_q)$ with distinct hk. Then, each $\mathbf{x}$ generated by $\mathcal{S}$ gives an affine equation $\sum \alpha_i f(hk, x_i) = \alpha_0$. By giving explicitly many different $\mathbf{x}$, we can reconstruct many different equations. And we can retrieve a large number of images. Now, if $\mathcal{S}$ were too unpredictable, the above would be true for many different $\mathbf{x}$, which would allow us to compress a random function in $\mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$ beyond information-theoretic lower bounds.

In the first step of the proof (see Function Selection below), we select the "good" functions, i.e., those for which the LDD game is successful with high probability, and the source is still unpredictable. The first condition implies that most of the equations remain meaningful, while the latter means that we have a large number of vectors $\mathbf{x}$, and thus a large number of equations.

Afterwards, in the State Selection step, we want to make the adversary deterministic and independent of the source, so that we can encode the function on a cross product set. To do so, for any good function f, we fix a "good" state s_f. At this step the source and the adversary are independent. Notice that since the adversary is deterministic, $(\alpha_0, \ldots, \alpha_q)$ depends only of hk; we denote it as $(\alpha_0^{(hk)}, \ldots, \alpha_q^{(hk)})$.

Now consider the linear system we obtain for a fixed hk. Our unknowns are $y_i \coloneqq f(hk, i)$ with $i \in [p]$, and the system is of the form

$$\begin{pmatrix} m_{1,1} & \cdots & m_{1,p} \\ \vdots & \ddots & \vdots \\ m_{N,1} & \cdots & m_{N,p} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} \alpha_0^{(hk)} \\ \vdots \\ \alpha_0^{(hk)} \end{pmatrix}\,,$$

where N is the number of equations.

To understand how we determine the values $m_{i,j}$ above, let's look a small example with $q = 4$ and the first vector $\mathbf{x}$ equal to $(1, 7, 1000, 25)$. Then, we will have $m_{1,1} = \alpha_1^{(hk)}$, $m_{1,7} = \alpha_2^{(hk)}$, $m_{1,1000} = \alpha_3^{(hk)}$, $m_{1,25} = \alpha_4^{(hk)}$. All the other $m_{1,i}$ are zero.

Once we have the system, we need to make sure that the equations are linearly independent. Indeed, if this were not the case, it would not be possible to recover the function from its encoding. In other words, we need the matrix above to be invertible. As a first step, we need first to be sure which values are non-zero among $(\alpha_1^{(hk)}, \ldots, \alpha_q^{(hk)})$. This aspect changes the independence of the equations a lot, and it makes it more complex to compute the rank. We notice that the entries of the matrix being zero depends only on the vector $\boldsymbol{\alpha}$, and then of the key hk.

In the KEYS SELECTION we simplify the matrix invertibility analysis by restricting ourselves to a subset K_f of keys, such that the support C (i.e., the set of indices of non-zero coordinates) is constant. We also need to have this set big enough (to make the compression argument at the end work), and, of course, it should also be "good" in the sense that the chances of winning the LDD game on this restricted set of keys are good.

We are now in a position where all coefficients α for the subset C of size q' are non-zero. Our next step is to make a part of the equations independent, by assuring that it forms a triangular matrix if we put the equations in the right order.

Then, in INPUT SELECTION, we first create an order on the values y_i, and a function MP which associates to any equation the maximal unknown y for the order defined which appears in the equation. Function MP takes a vector $(x_1, \ldots, x_q)$ (which represents an equation), and returns the maximum x_i (representing $f(hk, x_i)$) among all the x_j with $j \in C$ for the chosen order $\prec$.

We have selected the unknowns we want to compute, then we have to select one equation for each of these unknowns. MP-Pred will select such an equation (encoded as a vector $\mathbf{x}$). The details about why the computed matrix is full rank can be found in MATRIX INVERTIBILITY.

Finally, we compute the size of our compression encoding (described in ENCODING PROCEDURE and DECODING PROCEDURE) in the CODING EFFICIENCY ANALYSIS section, and conclude deriving the bound claimed in the theorem. ■

Proof. We now carry out more formally the steps outlined in the overview above.

FUNCTION SELECTION. In the first part of our proof we define the set $F \subseteq \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$ of "good" functions f which we will encode later. A function f is deemed "good" if it has two properties: (1) Adversary $(\mathcal{S}, \mathcal{A})$ has reasonably large advantage in the LDD game when the game picks $\pi = f$, and (2) Source $\mathcal{S}(f)$ is not too predictable, i.e., choosing f does not make $\mathcal{S}$ predictable. In order to show that there is a sizable proportion of functions with both attributes, we define two subsets $G, H \subseteq \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$, each capturing one of the properties that we are interested in. We then show that both sets are "large," which means that their intersection $F \coloneqq G \cap H$ must be "large" as well. More formally, define sets

$$G \coloneqq \left\{ f \in \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p) \;\middle|\; \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)\right] \geq \frac{\epsilon}{2} \right\},$$

and, for any $0 < N \leq U$,

$$H_N \coloneqq \left\{ f \in \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p) \;\middle|\; \forall \mathcal{P} : \Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p)\right] < \frac{N}{U} \right\},$$

where the games $\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)$ and $\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p)$ are the LDD and Pred games with $\pi = f$ fixed, as in Figures 18 (top left) and 18 (top right). To show that G is "large," consider the proportion $\epsilon' \coloneqq |G|/p^{(p^2)}$ of functions $f \in \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)$ contained in G. Then:

$$\begin{aligned}
\epsilon &\leq \Pr\left[\mathrm{LDD}_{\mathrm{RF}}^{\mathcal{S},\mathcal{A}}(p)\right] = \sum_{f \in \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)} \Pr\left[\mathrm{LDD}_{\mathrm{RF}}^{\mathcal{S},\mathcal{A}}(p) \mid \pi = f\right] \Pr[\pi = f] \\
&= \sum_{f \in \mathrm{Fun}(\mathbb{Z}_p^2, \mathbb{Z}_p)} \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)\right] \Pr[\pi = f] \\
&= \sum_{f \in G} \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)\right] \Pr[\pi = f] + \sum_{f \notin G} \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)\right] \Pr[\pi = f] \\
&\leq \Pr[\pi \in G] + \frac{\epsilon}{2} \Pr[\pi \notin G] = \Pr[\pi \in G] + \frac{\epsilon}{2}(1 - \Pr[\pi \in G]) = \epsilon' + \frac{\epsilon}{2}(1 - \epsilon').
\end{aligned}$$

Game $\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)$:	Game $\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p)$:
$(x_1,\ldots,x_q,\mathsf{st}) \twoheadleftarrow \mathcal{S}(f)$; $hk \twoheadleftarrow \mathbb{Z}_p$ $(\alpha_0,\alpha_1,\ldots,\alpha_q) \twoheadleftarrow \mathcal{A}(f,hk,\mathsf{st})$ return $\left(\sum \alpha_i \cdot f(hk,x_i) = \alpha_0\right)$	$(x_1,\ldots,x_q,\mathsf{st}) \twoheadleftarrow \mathcal{S}(f)$; $x' \twoheadleftarrow \mathcal{P}(f,\mathsf{st})$ return $(x' \in \{x_1,\ldots,x_q\})$
Predictor $\mathcal{P}(f)$:	Predictor $\mathcal{P}(f,s)$:
if $(f \notin H_N)$ then return $\mathcal{P}_f(f)$ return ε	if $(s \notin W_{f,N})$ then return $\mathcal{P}_s(f,s)$ return 0

Figure 18 — *Top left:* The LDD game with fixed function $\pi = f$. *Top right:* The predictability game with fixed function $\pi = f$. *Bottom left:* Predictor against the predictability of $\mathcal{S}$ to bound $|H_N|$. *Bottom right:* Predictor for the game $\mathrm{Pred}_{f,\mathcal{S}}(p)$ to show that $W_{f,N}$ is "large."

From here we get $\epsilon' \geq \epsilon/(2-\epsilon) > \epsilon/2$, which means $|G| \geq \epsilon p^{(p^2)}/2$.

Moving on to estimating $|H_N|$ note that, by definition, for every $f \notin H_N$ there exists a predictor $\mathcal{P}_f$ such that $\Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}_f}(p)\right] \geq N/U$. We can therefore define a predictor $\mathcal{P}$ against the predictability of $\mathcal{S}$ as shown in Figure 18 (bottom left). Given that $\mathcal{S}$ is U-unpredictable, we have

$$\begin{aligned}
\frac{1}{U} &\geq \Pr\left[\mathrm{Pred}_{\mathrm{RF},\mathcal{S}}^{\mathcal{P}}(p)\right] \geq \Pr\left[\mathrm{Pred}_{\mathrm{RF},\mathcal{S}}^{\mathcal{P}}(p) \wedge (\pi \notin H_N)\right] = \sum_{f \notin H_N} \Pr\left[\mathrm{Pred}_{\mathrm{RF},\mathcal{S}}^{\mathcal{P}}(p) \wedge (\pi = f)\right] \\
&= \sum_{f \notin H_N} \Pr\left[\mathrm{Pred}_{\mathrm{RF},\mathcal{S}}^{\mathcal{P}}(p) \;\middle|\; \pi = f\right] \Pr[\pi = f] = \sum_{f \notin H_N} \Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}_f}(p)\right] \Pr[\pi = f] \geq \sum_{f \notin H_N} \frac{N}{U} \Pr[\pi = f] \\
&= \frac{N}{U} \Pr[\pi \notin H_N] = \frac{N}{U}(1 - \Pr[\pi \in H_N]) = \frac{N}{U}\left(1 - \frac{|H_N|}{p^{(p^2)}}\right),
\end{aligned}$$

from which we get $|H_N| \geq p^{(p^2)}(1 - 1/N)$.

Now fix $H \coloneqq H_{2/\epsilon'}$ (notice that this requires $U \geq 2/\epsilon'$, which is implied by $U \geq 4/\epsilon$, which holds by assumption), and consider $F \coloneqq G \cap H$. We can show that F is "large" as well by bounding its complement:

$$|\bar{F}| = |\bar{G} \cup \bar{H}| \leq |\bar{G}| + |\bar{H}| \leq p^{(p^2)}(1-\epsilon') + p^{(p^2)}\frac{\epsilon'}{2} = p^{(p^2)}\left(1 - \frac{\epsilon'}{2}\right),$$

which means that $|F| \geq \epsilon' p^{(p^2)}/2 > \epsilon p^{(p^2)}/4$. In other words, we deduce that F (i.e., the set of functions we will compress later) contains at least a proportion $\epsilon/4$ of all functions.

State Selection. Now that we have selected the "good" functions, we show how to associate to each $f \in F$ a "good" state for the source $\mathcal{S}(f)$. For $f \in F$, a source state s_f is deemed "good" if it has two properties: (1) Source $\mathcal{S}(f)$ still has large advantage in the LDD game with $\pi = f$ and $\mathsf{st} = s_f$ fixed, and (2) Source $\mathcal{S}(f)$ does not become too predictable conditioned on its state being $\mathsf{st} = s_f$. Looking ahead, we will fix $\mathsf{st} = s_f$ later in the proof; this allows us to decouple $\mathcal{S}$ and $\mathcal{A}$ (there is no communication between them), while retaining lower bounds on the unpredictability of $\mathcal{S}$ and the LDD advantage of $(\mathcal{S},\mathcal{A})$.

To prove existence of a state as above for every $f \in F$, we again define sets $T_f, W_f \subseteq \{0,1\}^*$, each capturing one of the properties we are interested in, and prove that both are "large." As a result, their intersection $S_f \coloneqq T_f \cap W_f$ cannot be empty, because we can show $\Pr[\mathsf{st} \in S_f] > 0$. We will then set s_f to be any element of the non-empty set S_f. More formally, fix any $f \in F$ and define sets

$$T_f \coloneqq \left\{ s \in \{0,1\}^* \;\middle|\; \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p) \;\middle|\; \mathsf{st} = s\right] \geq \frac{\epsilon}{4} \right\},$$

and, for any $0 < N \leq \epsilon' U/2$,

$$W_{f,N} \coloneqq \left\{ s \in \{0,1\}^* \;\middle|\; \forall \mathcal{P} : \Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p) \;\middle|\; \mathsf{st} = s\right] \leq N \frac{2}{\epsilon' U} \right\}.$$

Since $f \in F \subseteq G$, we can show as above that T_f is "large":

$$\begin{aligned}
\frac{\epsilon}{2} &\leq \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)\right] = \sum_{s \in \{0,1\}^*} \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p) \;\middle|\; \mathsf{st} = s\right] \Pr[\mathsf{st} = s] \\
&= \sum_{s \in T_f} \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p) \;\middle|\; \mathsf{st} = s\right] \Pr[\mathsf{st} = s] + \sum_{s \notin T_f} \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p) \;\middle|\; \mathsf{st} = s\right] \Pr[\mathsf{st} = s] \\
&\leq \Pr[\mathsf{st} \in T_f] + \frac{\epsilon}{4} \Pr[\mathsf{st} \notin T_f] = \Pr[\mathsf{st} \in T_f] + \frac{\epsilon}{4}(1 - \Pr[\mathsf{st} \in T_f]),
\end{aligned}$$

from which we deduce that $\epsilon'' \coloneqq \Pr[\mathsf{st} \in T_f] > \epsilon/(4-\epsilon) > \epsilon/4$.

Furthermore, to obtain a bound for $W_{f,N}$ notice that, for every $s \notin W_{f,N}$, there exists a predictor $\mathcal{P}_s$ such that $\Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}_s}(p) \;\middle|\; \mathsf{st} = s\right] > 2N/(\epsilon' U)$. Define a predictor $\mathcal{P}$ as shown in Figure 18 (bottom right). Since $f \in F \subseteq H$, we have:

$$\begin{aligned}
\frac{2}{\epsilon' U} &\geq \Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p)\right] \geq \Pr\left[\left(\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p)\right) \wedge (\mathsf{st} \notin W_{f,N})\right] = \sum_{s \notin W_{f,N}} \Pr\left[\left(\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p)\right) \wedge (\mathsf{st} = s)\right] \\
&= \sum_{s \notin W_{f,N}} \Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}}(p) \;\middle|\; \mathsf{st} = s\right] \Pr[\mathsf{st} = s] = \sum_{s \notin W_{f,N}} \Pr\left[\mathrm{Pred}_{f,\mathcal{S}}^{\mathcal{P}_s}(p) \;\middle|\; \mathsf{st} = s\right] \Pr[\mathsf{st} = s] \\
&\geq N \frac{2}{\epsilon' U} \Pr[\mathsf{st} \notin W_{f,N}] = N \frac{2}{\epsilon' U}(1 - \Pr[\mathsf{st} \in W_{f,N}]),
\end{aligned}$$

which means that $\Pr[\mathsf{st} \in W_{f,N}] > 1 - 1/N$.

Now define $W_f \coloneqq W_{f,2/\epsilon''}$ (notice that this requires $U \geq 4/(\epsilon'\epsilon'')$, which is implied by $U \geq 32/\epsilon^2$, which again holds by assumption), and set $S_f \coloneqq T_f \cap W_f$. Then

$$\Pr[\mathsf{st} \notin S_f] = \Pr[(\mathsf{st} \notin T_f) \vee (\mathsf{st} \notin W_f)] \leq \Pr[\mathsf{st} \notin T_f] + \Pr[\mathsf{st} \notin W_f] \leq 1 - \epsilon'' + \frac{\epsilon''}{2} = 1 - \frac{\epsilon''}{2} < 1 - \frac{\epsilon}{8},$$

where st is the random variable defined in the game $\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p)$. From here we get $\Pr[\mathsf{st} \in S_f] > \epsilon/8 > 0$, thus in particular $S_f \neq \emptyset$. We associate to any function $f \in F$ a "good" state $s_f \in S_f$.

Adversary Simplification. Now that we have chosen, for every "good" function $f \in F$, a "good" state s_f, we can use it to decouple $\mathcal{S}$ and $\mathcal{A}$. Indeed, conditioned on $\mathsf{st} = s_f$, $\mathcal{S}$ and $\mathcal{A}$ become independent, since there is no communication between the two algorithms. More formally, define a source $\mathcal{S}'$ that is independent of $\mathcal{A}$ such that, for every $f \in F$, distribution of $\mathcal{S}'(f)$ is the distribution of $\mathcal{S}(f)$ conditioned on $\mathsf{st} = s_f$. That is, for every $\mathbf{x} \in \mathbb{Z}_p^*$ we set

$$\Pr[\mathcal{S}'(f) = \mathbf{x}] \coloneqq \Pr[\mathcal{S}(f) = (\mathbf{x}, s_f) \mid \mathsf{st} = s_f].$$

Then, for every $f \in F$, since $s_f \in S_f \subseteq T_f$ we have

$$\Pr\left[\text{FS-LDD}_f^{\mathcal{S},\mathcal{A}}(p)\right] = \Pr\left[\mathrm{LDD}_f^{\mathcal{S},\mathcal{A}}(p) \;\middle|\; \mathsf{st} = s_f\right] \geq \frac{\epsilon}{4},$$

where the game $\text{FS-LDD}_{f,K}^{\mathcal{S},\mathcal{A}}(p)$ for a set $K \subseteq \mathbb{Z}_p$ is defined in Figure 19 (top left).

Moving on to $\mathcal{A}$, without loss of generality we can assume $\mathcal{A}$ to be deterministic. Indeed, since $\mathcal{A}$ is unbounded, it can always compute the tuple $(\alpha_0, \alpha_1, \ldots, \alpha_q)$ which maximizes the probability of winning the LDD game for given f and hk. We denote by $\left(\alpha_0^{(f,hk)}, \alpha_1^{(f,hk)}, \ldots, \alpha_q^{(f,hk)}\right)$ the deterministic output of $\mathcal{A}(f, hk, s_f)$. Notice that, by definition of an LDD adversary, these coefficients cannot be all zero.

Game FS-LDD$^{\mathcal{S},\mathcal{A}}_{f,K}(p)$:	Game FS-Pred$^{\mathcal{P}}_{f,\mathcal{S}}(p)$:
$(x_1,\ldots,x_q) \twoheadleftarrow \mathcal{S}'(f)$; $hk \twoheadleftarrow K$; $(\alpha_0,\alpha_1,\ldots,\alpha_q) \twoheadleftarrow \mathcal{A}(f,hk,s_f)$ return $\left(\sum \alpha_i \cdot f(hk,x_i) = \alpha_0\right)$	$(x_1,\ldots,x_q) \twoheadleftarrow \mathcal{S}'(f)$; $x' \twoheadleftarrow \mathcal{P}(f)$ return $(x' \in \{x_1,\ldots,x_q\})$
Game MP-LDD$^{\mathcal{S},\mathcal{A}}_{f}(p)$:	Game MP-Pred$^{\mathcal{P}}_{f,\mathcal{S}}(p)$:
$(x_1,\ldots,x_q) \twoheadleftarrow \mathrm{MP}_f^{-1}(\mathrm{MP}_f(\mathcal{S}'(f)))$; $hk \twoheadleftarrow \mathbb{Z}_p$ $(\alpha_0,\alpha_1,\ldots,\alpha_q) \twoheadleftarrow \mathcal{A}(f,hk,s_f)$ return $\left(\sum \alpha_i \cdot f(hk,x_i) = \alpha_0\right)$	$(x_1,\ldots,x_q) \twoheadleftarrow \mathcal{S}'(f)$; $x^* \leftarrow \mathrm{MP}_f(x_1,\ldots,x_q)$ $x' \twoheadleftarrow \mathcal{P}(f)$; return $(x^* = x')$

Figure 19 — *Top left:* The LDD game with fixed function $\pi = f$ and state s_f given to $\mathcal{A}$. The hash key is drawn from the set $K \subseteq \mathbb{Z}_p$. For brevity, we let FS-LDD$^{\mathcal{S},\mathcal{A}}_{f}(p) \coloneqq$ FS-LDD$^{\mathcal{S},\mathcal{A}}_{f,\mathbb{Z}_p}(p)$. *Top right:* The predictability game with fixed function $\pi = f$. *Bottom left:* The LDD game with fixed function $\pi = f$ and selection of the most probable vectors. *Bottom right:* The predictability game with fixed function $\pi = f$ and selection of the most probable x.

Keys Selection. We now come back to working towards encoding functions $f \in F$. Our encoding will actually compress f only on a subset of its domain $\mathbb{Z}_p^2$ and give the remaining function values in full. Thanks to the independence of $\mathcal{S}'$ and $\mathcal{A}$, we can let this subset be a rectangle $K_f \times X_f \subseteq \mathbb{Z}_p^2$. We discuss in the following how to choose the "good" keys K_f, and leave the definition of the "good" inputs X_f to the next part. Broadly speaking, the set K_f must have three properties: (1) It should be relatively large, so that a random key likely belongs to it; (2) We must know exactly which coefficients α_i are non-zero when $\mathcal{A}$ is run on $hk \in K_f$; and (3) Adversaries $(\mathcal{S}',\mathcal{A})$ win the FS-LDD game with relatively high probability when the hash key is drawn from K_f.

More formally, fix a function $f \in F$, its corresponding state s_f, and a parameter $N \leq p/2^q$ that will be set later. Notice that since $\mathcal{A}$ is deterministic, we can classify hash keys according to which coefficients $\alpha_i^{(f,hk)}$ are non-zero. More precisely, for any $I \subseteq [q]$, define $K_{f,I} \subseteq \mathbb{Z}_p$ as the set of all hash keys such that $\alpha_i^{(f,hk)} \neq 0$ exactly for indices $i \in I$, i.e.,

$$hk \in K_{f,I} \iff \forall i \in [q] : \left(\alpha_i^{(f,hk)} \neq 0 \iff i \in I\right).$$

Since $\alpha_i^{(f,hk)} \neq 0$ for at least one $i \in [q]$, we deduce that $K_{f,\emptyset} = \emptyset$. We also notice that $\{K_{f,I}\}$, with $I \subseteq [q]$ and $I \neq \emptyset$, is a partition of $\mathbb{Z}_p$. Denote by J_N the collection of all $I \subseteq [q]$ such that $|K_{f,I}| \geq p/(N \cdot 2^q)$, and let $U \coloneqq \bigcup_{I \in J_N} K_{f,I}$. Then we have:

$$\begin{aligned}
\frac{\varepsilon}{4} &\leq \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f}(p)\right] = \frac{|\bar{U}|}{p}\Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,\bar{U}}(p)\right] + \frac{|U|}{p}\Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right] \\
&= \frac{|\bar{U}|}{p}\Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,\bar{U}}(p)\right] + \frac{p-|\bar{U}|}{p}\Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right] \leq \frac{|\bar{U}|}{p} + \frac{p-|\bar{U}|}{p}\Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right] \\
&\leq \frac{\sum_{I \notin J_N}|K_{f,I}|}{p}\left(1 - \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right]\right) + \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right] \\
&< \frac{\sum_{I \notin J_N} 1}{N \cdot 2^q}\left(1 - \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right]\right) + \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right] \\
&= \frac{2^q - |J_N|}{N \cdot 2^q}\left(1 - \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right]\right) + \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right] \\
&\leq \frac{1}{N}\left(1 - \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right]\right) + \Pr\left[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\right],
\end{aligned}$$

from which we deduce that

$$\Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,U}(p)\big] \geq \frac{N\epsilon - 4}{4(N-1)}\,.$$

Since $U = \bigcup_{I \in J_N} K_{f,I}$ is a union of disjoint sets, the above is equivalent to

$$\sum_{I \in J_N} \frac{|K_{f,I}|}{\sum_{L \in J_N} |K_{f,L}|} \Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_{f,I}}(p)\big] \geq \frac{N\epsilon - 4}{4(N-1)}\,,$$

which in turn implies

$$\max_{I \in J_N} \Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_{f,I}}(p)\big] \geq \frac{N\epsilon - 4}{4(N-1)}\,,$$

since the sum above is a convex combination. Now consider any set $C_N \in J_N$ which maximizes the probability on the left. From the definition of J_N and the inequality above we deduce:

$$|K_{f,C_N}| \geq \frac{p}{N \cdot 2^q}\,, \qquad \Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_{f,C_N}}(p)\big] \geq \frac{N\epsilon - 4}{4(N-1)}\,.$$

Define $C \coloneqq C_{8/\epsilon}$ (note that this is allowed because $\epsilon \geq 2^{q+3}/p$ by assumption). From above we then get:

$$|K_{f,C}| \geq \frac{\epsilon p}{2^{q-3}}\,, \qquad \Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_{f,C}}(p)\big] \geq \frac{\epsilon}{8-\epsilon}\,.$$

Finally, let $K_f \subseteq K_{f,C}$ be any subset of cardinality exactly $N_1 \coloneqq \lfloor \epsilon p / 2^{q-3} \rfloor$, which maximizes the quantity $\Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K}(p)\big]$. Then we have:

$$|K_f| = \left\lfloor \frac{\epsilon p}{2^{q-3}} \right\rfloor, \qquad \Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_f}(p)\big] \geq \frac{\epsilon}{8-\epsilon} > \frac{\epsilon}{8}\,.$$

Input Selection. Having chosen the set of "good" keys, we now pick the set of "good" inputs X_f. These two sets together define the region $K_f \times X_f$, which is the set of inputs to f on which we encode the function. To define X_f, we first introduce a strict total order $\prec_f$ on $\mathbb{Z}_p$ by demanding that $a \prec_f b$ if and only if

$$\Pr_{\mathbf{x} \twoheadleftarrow \mathcal{S}'(f)}[a \in \{x_i \mid i \in C\}] < \Pr_{\mathbf{x} \twoheadleftarrow \mathcal{S}'(f)}[b \in \{x_i \mid i \in C\}]\,,$$

or if the two probabilities above coincide and $a < b$ w.r.t. the natural order on $\mathbb{Z}_p$. We also define a function $\mathrm{MP}_f \colon \mathsf{Rng}(\mathcal{S}'(f)) \to \mathsf{Rng}(\mathrm{MP}_f) \subseteq \mathbb{Z}_p$ mapping $(x_1, \ldots, x_q)$ to the maximal value of the set $\{x_i \mid i \in C\}$ w.r.t. $\prec_f$. Finally, we also consider a right inverse $\mathrm{MP}_f^{-1} \colon \mathsf{Rng}(\mathrm{MP}_f) \to \mathsf{Rng}(\mathcal{S}'(f))$ of MP_f, which on input $x \in \mathsf{Rng}(\mathrm{MP}_f)$ returns the lexicographically first vector among the preimages $(x_1, \ldots, x_q)$ of x under MP_f which maximize the probability

$$\Pr_{hk \twoheadleftarrow K_f}\Big[\sum \alpha_i^{(f,hk)} f(hk, x_i) = \alpha_0^{(f,hk)}\Big]\,.$$

Then, by construction,

$$\Pr\big[\text{MP-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_f}(p)\big] \geq \Pr\big[\text{FS-LDD}^{\mathcal{S},\mathcal{A}}_{f,K_f}(p)\big] \geq \epsilon/8\,. \tag{6}$$

We can also notice that, for any predictor $\mathcal{P}$,

$$\Pr\big[\text{MP-Pred}^{\mathcal{P}}_{f,\mathcal{S}}(p)\big] \leq \Pr\big[\text{FS-Pred}^{\mathcal{P}}_{f,\mathcal{S}}(p)\big] \leq \frac{32}{\epsilon^2 U}\,,$$

where the last inequality follows from $s_f \in S_f \subseteq W_f$.

Applying [Vad12, Lemma 6.10], we can write $\mathrm{MP}_f(\mathcal{S}'(f))$ as a convex combination of uniform distributions: There exist an integer c and $(\lambda_1, \dots, \lambda_c) \in [0,1]^c$ with $\sum \lambda_i = 1$, and sets $X_{f,1}, \dots, X_{f,c} \subseteq \mathbb{Z}_p$ with $N_2 \coloneqq |X_{f,i}| = 2^{\chi(\log(\epsilon^2 U/32))}$, such that

$$\mathrm{MP}_f(\mathcal{S}'(f)) = \sum_{i=1}^{c} \lambda_i \mathcal{U}_{X_{f,i}}\,.$$

Choose an index $1 \le r \le c$ such that

$$\Pr_{\mathbf{x} \twoheadleftarrow \mathrm{MP}_f^{-1}(\mathcal{U}_{X_{f,r}}), hk \twoheadleftarrow K_f} \Big[\sum \alpha_i^{(f,hk)} f(hk, x_i) = \alpha_0^{(f,hk)}\Big]$$

is maximal, and set $X_f \coloneqq X_{f,r}$. This implies

$$\begin{aligned}\Pr_{\substack{\mathbf{x} \twoheadleftarrow \mathrm{MP}_f^{-1}(\mathcal{U}_{X_f}) \\ hk \twoheadleftarrow K_f}} \Big[\sum \alpha_i^{(f,hk)} f(hk, x_i) = \alpha_0^{(f,hk)}\Big] &\ge \Pr_{\substack{\mathbf{x} \twoheadleftarrow \mathrm{MP}_f^{-1}(\mathrm{MP}_f(\mathcal{S}'(f))) \\ hk \twoheadleftarrow K_f}} \Big[\sum \alpha_i^{(f,hk)} f(hk, x_i) = \alpha_0^{(f,hk)}\Big] \\ &= \Pr\big[\text{MP-LDD}_{f,K_f}^{\mathcal{S},\mathcal{A}}(p)\big] > \frac{\epsilon}{8}\,.\end{aligned} \tag{7}$$

Encoding Procedure. We can now finally use the function Encoding to efficiently describe all functions in F. We can encode each function $f \in F$ by giving the set X_f as a vector $\mathbf{x}_f$ of size N_2 ordered according to $\prec_f$, the vectors of $\big\{\mathrm{MP}_f^{-1}(x) \,\big|\, x \in X_f\big\}$ as a vector of size qN_2 by ordering them according to $\prec_f$ for their preimages by MP_f^{-1}, the set K_f of size N_1, the lists of the vectors $\big(\big(\alpha_i^{(f,hk)}\big)_i\big)_{hk \in K_f}$ (in the natural order for the index hk) as a vector of size $(q+1)N_1$.

Then we describe the rectangle $\{f(hk,x) \mid (hk,x) \in K_f \times X_f\}$ by giving a subset $U \subseteq K_f \times X_f$ which corresponds to wrong equations, and we give a vector to give the concrete values for solutions of these equations $(v_{hk,x} \coloneqq f(hk,x))_{(hk,x)\in U}$.

Finally with the vector $(f(hk,x))_{(hk,x)\notin K_f \times X_f}$ we give directly all the values of the function outside the rectangle $K_f \times X_f$.

Decoding Description. The function Decoding takes as input a vector of the form

$$\Big(\mathbf{x}_f, (\mathbf{x}_1, \dots, \mathbf{x}_{N_2}), K_f, (\boldsymbol{\alpha}_{hk})_{hk\in K_f}, U, (v_a)_{a\in U}, (f_{hk,x})_{(hk,x)\notin K_f \times X_f}\Big)$$

as a function f such that:

- $f(hk,x) = f_{hk,x}$ if $hk \notin K_f$ or if $x \notin X_f \coloneqq \{\mathbf{x}_f[i] \mid i \in [N_2]\}$;
- $f(hk,x) = v_{(hk,x)}$ if $(hk,x) \in U$;
- Else for all $(hk,x) \in (K_f \times X_f) \setminus U$, we deduce the values by using all the equations $\sum_{i=1}^{q} \boldsymbol{\alpha}_{hk}[i] \cdot f(hk, \mathbf{x}_j[i]) = \alpha_0^{(f,hk)}$ for all $hk \in K_f$, and $j \in [N_2]$ (see the next part for the details).

Matrix Invertibility. We now need to show that our encoding above describes the full function. To do so, notice that for any hk, the only values of $f(hk,x)$ not explicitly given are the images for $(hk,x) \in K_f \times X_f$ for which $\sum \alpha_i^{(f,hk)} f(hk, \mathbf{x}[i]) = \alpha_0^{(f,hk)}$, with $\mathbf{x} = \mathrm{MP}^{-1}(x)$.

Let's fix the key to be hk.

Let X'_{hk} be the set of all the x's such that (hk,x) is not explicitly given.

Because $\boldsymbol{\alpha}_{hk} = \Big(\alpha_0^{(f,hk)}, \dots, \alpha_q^{(f,hk)}\Big)$ and all the $\mathrm{MP}^{-1}(x)$'s are known by the decoder (they are the $\mathbf{x}_i$'s), he has access to all these equations. We must now prove that the corresponding linear system has a unique solution.

We have the same number of equations and unknowns (call this number $N'_{hk} \coloneqq |X'_{hk}|$). We have to show that these equations are independent. Let $o_f^{(hk)}$ be the function from X'_{hk} to $[N'_{hk}]$ which gives the rank of each element of X'_{hk} for the order $\prec_f$ (i.e., $a \prec_f b \iff o_f^{(hk)}(a) < o_f^{(hk)}(b)$).

For all $x \in X'_{hk}$ and for all i such that $(\mathrm{MP}^{-1}(x))_i \in X'_{hk}$, we let $y_{o_f^{(hk)}(x)}$ be an indeterminate which is supposed to represent $f(hk, x)$, and we set $m^{(f,hk)}_{o_f^{(hk)}(x), o_f^{(hk)}((\mathrm{MP}^{-1}(x))_i)} \coloneqq \alpha_i^{(f,hk)}$, and $m_{i,j}^{(f,hk)} \coloneqq 0$ for all other indices.

For all $x \in X'_{hk}$, by calling $\mathrm{MP}^{-1}(x) = (x_1, \ldots, x_q)$, we can rewrite the equations $\sum \alpha_i^{(f,hk)} \cdot f(hk, x_i) = \alpha_0^{(f,hk)}$ as:

$$\sum_{x_i \in X'_{hk} \cap \{x_1, \ldots, x_q\}} \alpha_i^{(f,hk)} \cdot y_{o_f(x_i)} = \alpha_0^{(f,hk)} - \sum_{x_i \in \{x_1, \ldots, x_q\} \setminus X'_{hk}} \alpha_i^{(f,hk)} \cdot f(hk, x_i)\,.$$

Notice that the constant in the right side of the equality can be directly computed by the decoder.

We can rewrite all these equations with unknowns the $y_{o_f(x_i)}$'s as a matrix $M = (m_{i,j})_{1 \le i,j \le N}$.

Remark that $m_{i,j} = m_{i,j}^{(f,hk)}$.

- For all $i \in [N]$, $m_{i,i} \neq 0$. Indeed,

$$m_{i,i} = m_{i,i}^{(f,hk)} = m^{(f,hk)}_{o_f^{(hk)}(o_f^{(hk)-1}(i)), o_f^{(hk)}(o_f^{(hk)-1}(i))} = \alpha^{(f,hk)}_{n(o_f^{(hk)-1}(i))}\,,$$

  with $n(x)$ the index of x in $\mathrm{MP}^{-1}(x)$.

- If $j > i$, then $m_{i,j} = 0$. Proof: We deduce $o^{-1}(i) \prec_f o^{(hk)-1}(j)$, then $o^{(hk)-1}(j)$ can't appear in the coordinates of $\mathrm{MP}^{-1}(x)$ (because x is the maximum for this order by definition).

We deduce finally the invertibility of the matrix. With this information, we can deduce that all equations are independent, and the decoder can retrieve all of the function f.

Coding Efficiency Analysis. Let's count the functions generated by the codewords for a fixed N_1, N_2. Notice that because the equations can determine fully the rectangle, we choose for each function in this set an encoding where the number of true equations is minimal without any loss of generality (it doesn't disturb the encoding/decoding to consider a good equation as a wrong one), then $|U|$ is fixed to $N_1 N_2 - \lfloor N_1 N_2 \epsilon / 8 \rfloor$. This cardinality is at most $S(N_2, N_1)$, where:

$$\begin{aligned} S(x,y) &\le p^{x(q+1)} \cdot \binom{p}{y} \cdot p^{y(q+1)} \cdot \binom{xy}{\lfloor xy\epsilon/8 \rfloor} p^{xy - \lfloor \epsilon xy/8 \rfloor} \cdot p^{(p^2 - xy)} \\ &\le p^{x(q+1)} \cdot p^{y(q+2)} \cdot 2^{xy} p^{xy - \lfloor \epsilon xy/8 \rfloor} \cdot p^{(p^2 - xy)} \end{aligned}$$

The $p^{x(q+1)}$ is the number of possible $(\mathbf{x}, (\mathbf{x}_1, \ldots, \mathbf{x}_{N_2})$, $\binom{p}{y}$ the number of possible K_f, $p^{y(q+1)}$ the set of possible $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{N_1})$, $\binom{xy}{\lfloor xy\epsilon/8 \rfloor}$ the set of possible sets U, $p^{xy - \lfloor \epsilon xy/8 \rfloor}$ the set of possible $(v_a)_{a \in U}$, $p^{p^2 - xy}$ the number of possible $(f_{i,j}, \ldots, f_{i,j})_{(i,j) \notin K_f \times X_f}$.

We now want to upper-bound $S(N_2, N_1)$ in terms of ϵ only. It is equivalent to upper-bound $T(N_2, N_1)$, where:

$$\begin{aligned} T(\bar{x}, \bar{y}) \coloneqq \log(S(\bar{x}, \bar{y})) &\le \log(p)\left((q+1)(\bar{x} + \bar{y}) + \bar{y}\right) + \bar{x}\bar{y} - \left\lfloor \frac{\bar{x}\bar{y}\epsilon}{8} \right\rfloor \log(p) + \log(p) p^2 \\ &\le \log(p)\left((q+1)(\bar{x} + \bar{y}) + \bar{y}\right) + \bar{x}\bar{y} + \left(1 - \frac{\bar{x}\bar{y}\epsilon}{8}\right) \log(p) + \log(p) p^2\,. \end{aligned}$$

Then:

$$T(N_2,N_1) \leq \log(p)\left((q+1)\left(\left\lfloor\frac{\epsilon p}{2^{q-3}}\right\rfloor + 2^{\chi(\log(\epsilon^2 U/32))}\right) + \left\lfloor\frac{\epsilon p}{2^{q-3}}\right\rfloor + 1\right) + \left(\left\lfloor\frac{\epsilon p}{2^{q-3}}\right\rfloor \cdot 2^{\chi(\log(\epsilon^2 U/32))}\right)\left(1-\frac{\epsilon}{8}\log(p)\right) + \log(p)p^2\,.$$

Because $\frac{\epsilon}{8}\log(p) - 1 > 0$, (because by assumption $\epsilon > \frac{8}{\log(p)} + \frac{2^q}{2p} + \frac{2(q+1)}{\sqrt{p}}$), we now use the inequalities from Lemma 5.5 and the definition of $\lfloor\cdot\rfloor$:

$$T(N_2,N_1) \leq \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + (q+1)\frac{\epsilon^2 U}{32}\right) + \left(\left(\frac{\epsilon p}{2^{q-3}} - 1\right)\left(\frac{\epsilon^2 U}{32} - 1\right)\right)\left(1-\frac{\epsilon}{8}\log(p)\right) + \log(p)p^2\,. \tag{8}$$

Notice that $T(N_2,N_1) \geq \log(|F|) \geq \log\left(\epsilon p^{(p^2)}/4\right) = \log(\epsilon) + p^2\log(p) - 2$.

By applying Inequality (8), and deleting the additive term $\log(p)p^2$ the previous inequality becomes:

$$\begin{aligned}\log(\epsilon) - 2 \leq\ & \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + (q+1)\frac{\epsilon^2 U}{32}\right) + 1 + \frac{\epsilon^3 pU}{4\cdot 2^q} - \frac{\epsilon p}{2^{q-3}} - \frac{\epsilon^2 U}{32} \\ & - \left(\left(\frac{\epsilon p}{2^{q-3}} - 1\right)\left(\frac{\epsilon^2 U}{32} - 1\right)\right)\frac{\epsilon}{8}\log(p) \\ \leq\ & \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + (q+1)\frac{\epsilon^2 U}{32} - \frac{\epsilon}{8} + \frac{\epsilon^2 p}{8\cdot 2^{q-3}} + \frac{\epsilon^3 U}{8\cdot 32} - \frac{\epsilon^4 Up}{32\cdot 2^q}\right) \\ & + 1 + \frac{\epsilon^3 pU}{4\cdot 2^q} - \frac{\epsilon p}{2^{q-3}} - \frac{\epsilon^2 U}{32}\,.\end{aligned}$$

Developing further, we obtain:

$$\begin{aligned}0 \leq\ & -\log(\epsilon) + 3 + \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + (q+1)\frac{\epsilon^2 U}{32} - \frac{\epsilon}{8} + \frac{\epsilon^2 p}{8\cdot 2^{q-3}} + \frac{\epsilon^3 U}{8\cdot 32} - \frac{\epsilon^4 Up}{32\cdot 2^q}\right) \\ & + \frac{\epsilon^3 pU}{4\cdot 2^q} - \frac{\epsilon p}{2^{q-3}} - \frac{\epsilon^2 U}{32}\,.\end{aligned}$$

Then we put all the terms without any U's in the left part.

$$\begin{aligned}&\log(\epsilon) - 3 + \frac{\epsilon p}{2^{q-3}} - \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{8\cdot 2^{q-3}}\right) \\ &\qquad \leq \log(p)\left((q+1)\frac{\epsilon^2 U}{32} + \frac{\epsilon^3 U}{8\cdot 32} - \frac{\epsilon^4 Up}{32\cdot 2^q}\right) + \frac{\epsilon^3 pU}{4\cdot 2^q} - \frac{\epsilon^2 U}{32}\,.\end{aligned}$$

We divide by $\epsilon^2/32$.

$$\begin{aligned}&\frac{32}{\epsilon^2}\left(\log(\epsilon) - 3 + \frac{\epsilon p}{2^{q-3}} - \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{8\cdot 2^{q-3}}\right)\right) \\ &\qquad \leq U\left(\log(p)\left((q+1) + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{2^q}\right) + \frac{8\epsilon p}{2^q} - 1\right)\,. \end{aligned}\tag{9}$$

Define $Q \coloneqq 2^q$, and $f(x) \coloneqq \log(p)\left((q+1)x + \frac{x}{8} - \frac{x^2 p}{2^q}\right) + \frac{8xp}{2^q} - 1$.
Because we do not know if $f(\epsilon)$ is positive, we have to study it before using Equation (9).
By solving the quadratic system in ϵ we deduce that:

$$\forall \epsilon \in \left[\frac{2^{q-1}((p(2^{3-q}) + \log(p)/8) - \sqrt{(p2^{3-q} + \log(p)/8)^2 + p2^{2-q}\log(p)(q\log(p) + \log(p) - 1)})}{p\log(p)}; \right.$$
$$\left. \frac{2^{q-1}((p(2^{3-q}) + \log(p)/8) + \sqrt{(p2^{3-q} + \log(p)/8)^2 + p2^{2-q}\log(p)(q\log(p) + \log(p) - 1)})}{p\log(p)}\right]$$

implies $f(\epsilon) > 0$. By assumption, we have $\epsilon > \frac{8}{\log(p)} + \frac{2^q}{2p} + \frac{2(q+1)}{\sqrt{p}}$, then we deduce the following upper bound:

$$\epsilon > \frac{8}{\log(p)} + \frac{2^q}{2p} + \frac{2(q+1)}{\sqrt{p}}$$

We put all on the same denominator.

$$= \frac{8p + 2^q\log(p)/2 + \log(p)\sqrt{p}2(q+1)}{p\log(p)}$$
$$= \frac{2(p(2^2) + 2^q\log(p)/4) + \log(p)\sqrt{p}2(q+1)}{p\log(p)}$$
$$= \frac{2^{q-1}\left(2(p(2^{3-q}) + \log(p)/8) + \log(p)\sqrt{p}2^{2-q}(q+1)\right)}{p\log(p)}$$

We use the fact $\sqrt{n} < n$ for any $n > 1$, and $2^{2-q}(q+1) > 1$.

$$> \frac{2^{q-1}\left(2(p(2^{3-q}) + \log(p)/8) + \log(p)\sqrt{p2^{2-q}(q+1)}\right)}{p\log(p)}$$
$$= \frac{2^{q-1}\left((p(2^{3-q}) + \log(p)/8) + p2^{3-q} + \log(p)/8 + \log(p)\sqrt{p2^{2-q}(q+1)}\right)}{p\log(p)}$$
$$= \frac{2^{q-1}\left((p(2^{3-q}) + \log(p)/8) + p2^{3-q} + \log(p)/8 + \sqrt{p2^{2-q}\log^2(p)(q+1)}\right)}{p\log(p)}$$
$$= \frac{2^{q-1}\left((p(2^{3-q}) + \log(p)/8) + p2^{3-q} + \log(p)/8 + \sqrt{p2^{2-q}\log(p)(q\log(p) + \log(p))}\right)}{p\log(p)}$$
$$> \frac{2^{q-1}\left((p(2^{3-q}) + \log(p)/8) + p2^{3-q} + \log(p)/8 + \sqrt{p2^{2-q}\log(p)(q\log(p) + \log(p) - 1)}\right)}{p\log(p)}$$
$$= \frac{2^{q-1}\left((p(2^{3-q}) + \log(p)/8) + \sqrt{(p2^{3-q} + \log(p)/8)^2} + \sqrt{p2^{2-q}\log(p)(q\log(p) + \log(p) - 1)}\right)}{p\log(p)}$$

We use the fact $\sqrt{a} + \sqrt{b} > \sqrt{a+b}$.

$$> \frac{2^{q-1}\left((p(2^{3-q}) + \log(p)/8) + \sqrt{(p2^{3-q} + \log(p)/8)^2 + p2^{2-q}\log(p)(q\log(p) + \log(p) - 1)}\right)}{p\log(p)}$$

It implies $f(\epsilon) < 0$, then:

$$\frac{32}{\epsilon^2} \cdot \frac{\log(\epsilon) - 3 + \frac{\epsilon p}{2^{q-3}} - \log(p)\left((q+2)\frac{\epsilon p}{2^{q-3}} + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{8 \cdot 2^{q-3}}\right)}{\log(p)\left((q+1) + \frac{\epsilon}{8} - \frac{\epsilon^2 p}{2^q}\right) + \frac{8\epsilon p}{2^q} - 1} \geq U\,.$$

This concludes the proof. □

6 Applications of PGGs

We now present some examples of how PGGs can be used to prove the hardness of group-based assumptions and the security of practical cryptosystems under a variety of notions. As our first application, we prove that the decisional Uber assumption (DUA) family holds in PGGs. In doing so we also capture all of its implications. We then turn our attention to applications which do not seem to fall under the umbrella of the DUA. In Section 6.2 we show how to construct UCEs for split sources from PGGs and LDDs, thereby recovering several applications discussed in [BHK13]. Further applications, namely KDM-CPA and RKA-CPA security of (modified versions of) ElGamal, and security of the ElGamal-with-Hash deterministic encryption scheme, are discussed in Sections 6.3 to 6.5. Interestingly, all these applications enjoy reductions under PGGs which furthermore retain to a large extent the simplicity of proofs in the GGM. Standard-model constructions of such schemes under Uber (for example, the KDM-secure PKE scheme of Boneh et al. [BHHO08]) are often substantially more complex and less efficient.

6.1 Uber Assumption in PGGs

The Uber assumption family [BBG05, Boy08] is an umbrella assumption that generalizes many hardness assumptions used to analyze the security of concrete cryptosystems. It has been formalized for both simple and bilinear groups, and has been shown to hold in (bilinear) generic groups [BBG05]. Here we focus on simple (i.e., non-bilinear) groups and show that Uber assumptions for them fall within the PGG framework. More precisely, we show that non-interactive, generator-independent Uber assumptions hold for PGGs.

We study an entropic generalization of the decisional version of the Uber assumption, which we call DUA-II, and show that it holds for PGGs. Loosely speaking, DUA-II extends DUA by sampling the inputs to the polynomials from independent, high-entropy distributions, rather than uniformly at random. Restricted versions of DUA-II and applications thereof have previously appeared in the literature (see, for example, Canetti's DDH-II assumption in Figure 20 (bottom left)).

Decisional Uber Assumption II (DUA-II). Let Γ be a computational group scheme. We define the advantage of an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the DUA-II game for Γ as

$$\mathrm{Adv}^{\text{dua-ii}}_{\Gamma,\mathcal{A}}(\lambda) \coloneqq 2 \cdot \Pr\left[\text{DUA-II}^{\mathcal{A}}_{\Gamma}(\lambda)\right] - 1\,,$$

where the DUA-II game is defined in Figure 20 (top left). Here, $\mathcal{A}_0$ and $\mathcal{A}_1$ can be unbounded with polynomially bounded output, and $\mathcal{A}_2$ is PPT. We require that T be linearly independent from $R_1, \ldots, R_n$, and that $\mathbf{H}_\infty(\mathcal{A}_1(i,\pi)) = \omega(\log\lambda)$ for every $i \in \mathbb{N}$ and every $\pi \twoheadleftarrow \Gamma(1^\lambda)$. We say that Γ is DUA-II secure if, for any $\mathcal{A}$ as above, the advantage of $\mathcal{A}$ in the DUA-II game for Γ is negligible.

Throughout this section, we will assume that the rational functions $R_1, \ldots, R_n$ returned by $\mathcal{A}_0$ are linearly independent. This is without loss of generality: Indeed, linear (in)dependence can be checked by computing $D \coloneqq \mathrm{lcm}(\check{R}_1, \ldots, \check{R}_n)$, then writing a generic linear combination $\sum_{i=1}^n a_i \cdot R_i D = 0$, and solving the resulting linear system for $(a_1, \ldots, a_n)$. This yields a nonzero solution if and only if $R_1, \ldots, R_n$ are linearly dependent. Now observe that if R_k is linearly dependent on $R_1, \ldots, R_{k-1}$, then $g^{R_k(\mathbf{x})}$ can be

Game DUA-II$^{\mathcal{A}}_{\Gamma}(\lambda)$:

$d \twoheadleftarrow \{0,1\}$; $(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $r \twoheadleftarrow \mathbb{Z}_p^*$; $g \leftarrow g_0^r$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big)$ then return true
for $i = 1$ to n do $h_i \leftarrow g^{R_i(\mathbf{s})}$
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
$h \leftarrow g^{r'}$; $d' \twoheadleftarrow \mathcal{A}_2(\pi, h_1, \ldots, h_n, h, \mathsf{st})$; return $(d = d')$

Game DDH-II$^{\mathcal{A}}_{\Gamma}(\lambda)$:

$d \twoheadleftarrow \{0,1\}$; $(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$r \twoheadleftarrow \mathbb{Z}_p^*$; $g \leftarrow g_0^r$; $x \twoheadleftarrow \mathcal{X}(\pi)$; $y \twoheadleftarrow \mathbb{Z}_p$
if $(d = 0)$ then $z \twoheadleftarrow \mathbb{Z}_p$ else $z \leftarrow xy$
$d' \twoheadleftarrow \mathcal{A}(\pi, g, g^x, g^y, g^z)$; return $(d = d')$

Auxiliary dUber source $\bar{\mathcal{S}}(\pi)$:

$d \twoheadleftarrow \{0,1\}$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big)$ then
  return $(\varepsilon, \varepsilon)$
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
$\mathbf{x} \leftarrow (R_1(\mathbf{s}), \ldots, R_n(\mathbf{s}), r')$; $L \leftarrow (d, \mathsf{st})$
return $(\mathbf{x}, L)$

PGG distinguisher $\mathcal{D}(\pi, (\mathbf{y}, L))$:

if $(L = \varepsilon)$ then return true
$(h_1, \ldots, h_n, h) \leftarrow \mathbf{y}$; $(d, \mathsf{st}) \leftarrow L$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, h_1, \ldots, h_n, h, \mathsf{st})$; return $(d = d')$

Figure 20 — *Top left:* The decisional Uber assumption II (DUA-II) game. Here, m is an upper bound on the number of variables of the R_i, $i \in [n+1]$. The (ordinary) decisional Uber assumption (DUA) is a special case of DUA-II where $\mathcal{A}_1(i, \pi)$ is the uniform distribution over $\mathbb{Z}_p$ for all $i \in [m]$ and all π. *Bottom left:* The decisional Diffie–Hellman assumption II (DDH-II) game of Canetti [Can97], where $\mathcal{X}(\pi)$ is assumed to be a well-spread distribution for all π. *Right:* Reduction from a DUA-II adversary $\mathcal{A}$ to a PGG adversary $(\mathcal{S}, \mathcal{D})$.

computed directly by $\mathcal{A}_2$ (who knows $g^{R_1(\mathbf{x})}, \ldots, g^{R_{k-1}(\mathbf{x})}$) before guessing d', without having to separately query the challenger on $R_k(\mathbf{x})$.

We now show that the DUA-II assumption holds in pseudo-generic groups. This allows us to recover all cryptographic applications that fall under the reach of DUA and DUA-II.

Theorem 6.1 (PGG $\Longrightarrow$ DUA-II)**.** *Let Γ be a computational group scheme. If Γ is* PGG$[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{duber}}]$ *secure, then it is* DUA-II *secure. More precisely, for any adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the DUA-II game there is an adversary $(\mathcal{S}, \mathcal{D})$ in the PGG game for Γ such that*

$$\mathrm{Adv}^{\text{dua-ii}}_{\Gamma,\mathcal{A}}(\lambda) \le 2 \cdot \mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + \frac{3n(\lambda)}{2^{\lambda-1}} + \frac{3(n(\lambda)+1)\big(\hat{d}(\lambda) + 3\check{d}(\lambda)\big)}{2^{k(\lambda)}} \,. \tag{10}$$

Furthermore, $\mathcal{S} \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{duber}}$. More precisely, for any predictor $\mathcal{P}$ in the AlgPred game for $(\Gamma, \mathcal{S})$, we have

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \le \frac{n(\lambda)}{2^\lambda} + \frac{n(\lambda)^2\big(\hat{d}(\lambda) + \check{d}(\lambda) + 1\big)}{2^{k(\lambda)-2}} \,. \tag{11}$$

Here, $\hat{d}(\lambda)$ and $\check{d}(\lambda)$ are upper bounds on the degrees of the numerator and denominator polynomials, respectively, $n(\lambda)$ is an upper bound on the number of rational functions $R_1, \ldots, R_n$ returned by $\mathcal{A}_0$, and $k(\lambda)$ is a lower bound on $\mathbf{H}_\infty(\mathcal{A}_1(i, \pi))$ for every $i \in \mathbb{N}$ and every $\pi \twoheadleftarrow \Gamma(1^\lambda)$.

Proof Overview. Given an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the DUA-II game, consider a PGG adversary $(\mathcal{S}, \mathcal{D})$ defined as follows: Source $\mathcal{S}$ runs $\mathcal{A}_0$ and $\mathcal{A}_1$, and queries Exp on $R_1(\mathbf{s}), \ldots, R_n(\mathbf{s}), r'$, where r' is either $T(\mathbf{s})$ or a random value, the choice being made at random. Distinguisher $\mathcal{D}$ then runs $\mathcal{A}_2$ and checks if it did predict the choice made by $\mathcal{S}$. By construction, $\mathcal{S}$ is a dUber source.

By direct inspection, the game $\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda)$ with challenge bit $b = 1$ coincides with the game DUA-II$^{\mathcal{A}}_{\Gamma}(\lambda)$. On the other hand, when $b = 0$, the probability of $(\mathcal{S}, \mathcal{D})$ winning the PGG game is negligible. This follows

from a bad event analysis: We transition to a game where $r' \neq R_i(\mathbf{s})$ for all $i \in [n]$. Given that σ is a random injection, we can then move to a game where the corresponding reply is picked at random, independently of the random choice of $\mathcal{S}$, so that $\mathcal{A}$ has no advantage in this game.

For algebraic unpredictability, let $\mathcal{P}$ be any predictor that returns a linear combination of the queries with coefficients $\alpha_1, \ldots, \alpha_n, \alpha_{n+1}$ given the leakage computed using the ideal Exp oracle. We again transition to a game where the Exp queries are pairwise distinct, and then replace the answers with pairwise different random elements that are independent of $\mathbf{s}$. Winning the algebraic unpredictability game then means that $\mathbf{s}$ (which $\mathcal{P}$ now knows nothing about) is a root of $\alpha_1 R_1 + \cdots + \alpha_n R_n + \alpha_{n+1} r'$, which is unlikely by the Schwartz–Zippel lemma. ■

Proof. Given an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the DUA-II game, define the PGG dUber source $\mathcal{S}$ via auxiliary algorithm $\bar{\mathcal{S}}$ and the distinguisher $\mathcal{D}$ as shown in Figure 20 (right). Observe that our reduction does not directly convert the DUA-II game into the PGG experiment, since the learning phase in the former case is always wrt. the real group operation, whereas it can involve the generic encoding in the latter. Therefore, the PGG source must simulate the DUA-II game in such a way that both cases $d \in \{0, 1\}$ are covered when it plays in the real world, while gaining almost no advantage in the ideal world.

DUber structure. By construction, it is clear that $\bar{\mathcal{S}}$ makes no Exp query and returns a vector $\mathbf{x}$ and leakage L. Thus, $\mathcal{S} \in \mathbf{S}^{\mathrm{duber}}$.

Advantage bound. To prove Inequality (10), first recall that

$$\mathrm{Adv}^{\mathrm{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) = \Pr\left[\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 1\right] - \Pr\left[\neg\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 0\right].$$

We study the two summands separately. For the first term, it is easily verified by direct inspection that the PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$ with bit $b = 1$ fixed is the same as the DUA-II game for Γ played by $\mathcal{A}$. This in particular means that $\Pr\left[\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 1\right] = \Pr\left[\text{DUA-II}^{\mathcal{A}}_{\Gamma}(\lambda)\right]$, and thus

$$\begin{aligned}\mathrm{Adv}^{\text{dua-ii}}_{\Gamma,\mathcal{A}}(\lambda) &= 2 \cdot \Pr\left[\text{DUA-II}^{\mathcal{A}}_{\Gamma}(\lambda)\right] - 1 = 2 \cdot \Pr\left[\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 1\right] - 1 \\ &= 2 \cdot \mathrm{Adv}^{\mathrm{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + 2 \cdot \Pr\left[\neg\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 0\right] - 1\,.\end{aligned}$$

We now study the second term in the sum above. To do so, consider the following sequence of games (the formal description of which can be found in Figure 21):

$\mathsf{Game}_0(\lambda)$ is the PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$, with bit $b = 0$ fixed and inverted winning condition. We also set a flag $\mathsf{Bad}_0 \leftarrow \mathsf{true}$ if the randomly chosen value $r' \twoheadleftarrow \mathbb{Z}_p$ happens to coincide with $R_i(\mathbf{s})$ for some $i \in [n]$.

$\mathsf{Game}_1(\lambda)$ is the same as $\mathsf{Game}_0(\lambda)$, but we resample r' if Bad_0 occurs. This ensures that $r' \neq R_i(\mathbf{s})$ for all $i \in [n]$.

$\mathsf{Game}_2(\lambda)$ is the same as $\mathsf{Game}_1(\lambda)$, but we no longer compute h using the random injection σ, but instead sample it at random from G. In case h is one of the elements $h_i = \sigma(R_i(\mathbf{s}))$ computed before, we set the flag $\mathsf{Bad}_1 \leftarrow \mathsf{true}$ and resample h so that $h \neq h_i$ for all $i \in [n]$.

$\mathsf{Game}_3(\lambda)$ is the same as $\mathsf{Game}_2(\lambda)$, but we omit resampling h. In other words, in this game h is a uniformly random element from G.

We now argue that the difference between the success probabilities of subsequent games is small.

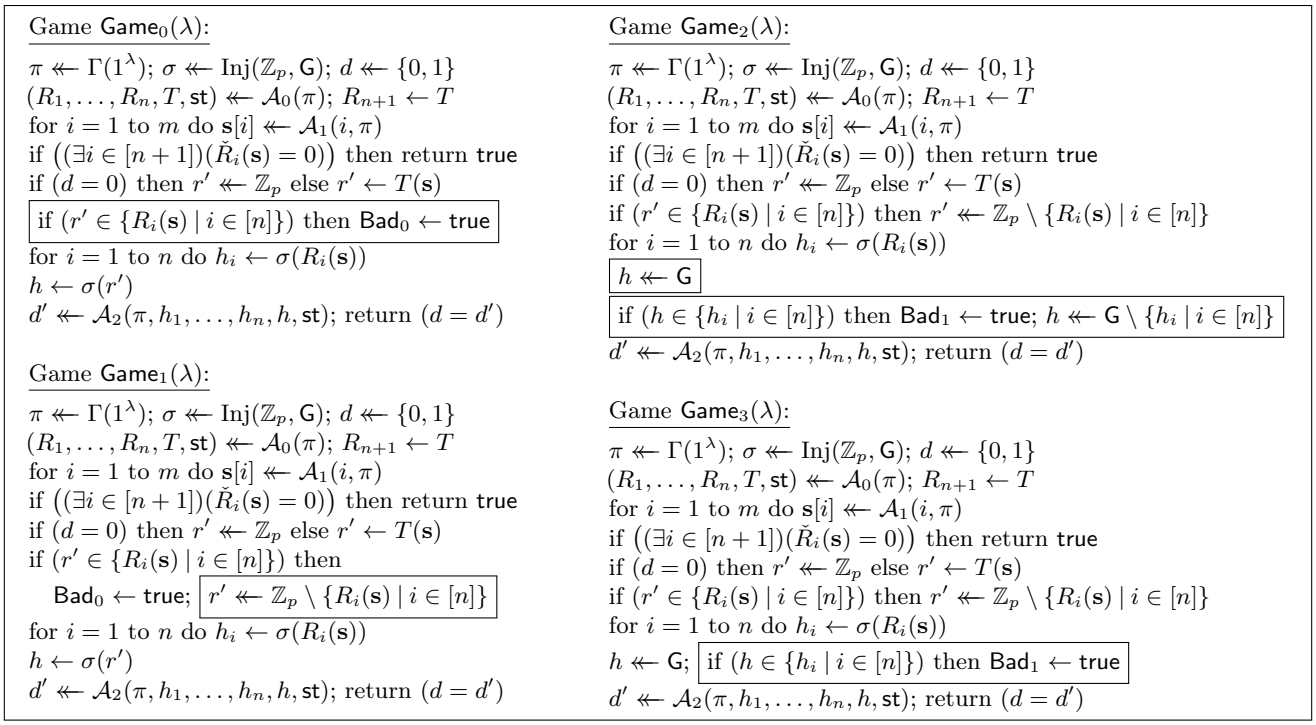

Figure 21 — Code of the intermediate games in the proof of the DUA advantage bound (10) in Theorem 6.1.

$\mathsf{Game}_0 \rightsquigarrow \mathsf{Game}_1$. By definition, $\mathsf{Game}_0(\lambda)$ and $\mathsf{Game}_1(\lambda)$ are identical until Bad_0, which means that $|\Pr[\mathsf{Game}_0(\lambda)] - \Pr[\mathsf{Game}_1(\lambda)]| \leq \Pr[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0]$ by the fundamental lemma of game playing. To bound the latter probability, observe that

$$\Pr[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0] = \frac{1}{2}\sum_{k=0}^{n(\lambda)} \Pr[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0 \mid (d=0) \wedge (n=k)] \Pr[n=k] + \frac{1}{2}\sum_{k=0}^{n(\lambda)} \Pr[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0 \mid (d=1) \wedge (n=k)] \Pr[n=k]\,,$$

where the sum over k extends only over those indices such that $\Pr[n=k] > 0$ in $\mathsf{Game}_0(\lambda)$. We now study the two conditional probabilities separately. For the case $d = 0$, notice that

$$\begin{aligned}
\Pr\left[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0 \;\middle|\; \begin{array}{l}(d=0) \wedge \\ \wedge\ (n=k)\end{array}\right] &= \Pr\left[\left(\bigwedge_{j=1}^{k+1} (\check{R}_j(\mathbf{s}) \neq 0)\right) \wedge \left(\bigvee_{i=1}^{k} (r' = R_i(\mathbf{s}))\right) \;\middle|\; \begin{array}{l}(d=0) \wedge \\ \wedge\ (n=k)\end{array}\right] \\
&\leq \sum_{i=1}^{k} \Pr\left[\left(\bigwedge_{j=1}^{k+1} (\check{R}_j(\mathbf{s}) \neq 0)\right) \wedge \big(r' = R_i(\mathbf{s})\big) \;\middle|\; (d=0) \wedge (n=k)\right] \\
&\leq \sum_{i=1}^{k} \Pr\left[r' = R_i(\mathbf{s}) \;\middle|\; \left(\bigwedge_{j=1}^{k+1} (\check{R}_j(\mathbf{s}) \neq 0)\right) \wedge (d=0) \wedge (n=k)\right] \overset{\text{(a)}}{\leq} \sum_{i=1}^{k} \frac{1}{2^{\lambda-1}} = \frac{k}{2^{\lambda-1}} \leq \frac{n(\lambda)}{2^{\lambda-1}}\,,
\end{aligned}$$

where the R_j, r', and $\mathbf{s}$ are distributed as in $\mathsf{Game}_0(\lambda)$ with $d = 0$ and $n = k$, and Inequality (a) holds because r' is distributed uniformly over $\mathbb{Z}_p$ (with $p \geq 2^{\lambda-1}$) and independently of all other random variables.

On the other hand, when $d = 1$ we have

$$\begin{aligned}
\Pr\left[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0 \;\middle|\; \begin{array}{l}(d=1)\wedge\\ \quad\wedge\ (n=k)\end{array}\right] &= \Pr\left[\left(\bigwedge_{j=1}^{k+1}(\check{R}_j(\mathbf{s})\neq 0)\right)\wedge\left(\bigvee_{i=1}^{k}(T(\mathbf{s})=R_i(\mathbf{s}))\right) \;\middle|\; \begin{array}{l}(d=1)\wedge\\ \quad\wedge\ (n=k)\end{array}\right]\\
&\le \sum_{i=1}^{k}\Pr\left[(T(\mathbf{s})=R_i(\mathbf{s}))\wedge\left(\bigwedge_{j=1}^{k+1}(\check{R}_j(\mathbf{s})\neq 0)\right)\;\middle|\;(d=1)\wedge(n=k)\right]\\
&= \sum_{i=1}^{k}\Pr\left[\left(\hat{T}(\mathbf{s})\check{R}_i(\mathbf{s})=\hat{R}_i(\mathbf{s})\check{T}(\mathbf{s})\right)\wedge\left(\bigwedge_{j=1}^{k+1}(\check{R}_j(\mathbf{s})\neq 0)\right)\;\middle|\;(d=1)\wedge(n=k)\right]\\
&\le \sum_{i=1}^{k}\Pr\left[\hat{T}(\mathbf{s})\check{R}_i(\mathbf{s})-\hat{R}_i(\mathbf{s})\check{T}(\mathbf{s})=0\;\middle|\;(d=1)\wedge(n=k)\right] \overset{(a)}{\le} \sum_{i=1}^{k}\frac{\hat{d}(\lambda)+\check{d}(\lambda)}{2^{k(\lambda)}} \le n(\lambda)\cdot\frac{\hat{d}(\lambda)+\check{d}(\lambda)}{2^{k(\lambda)}}\,.
\end{aligned}$$

Here, the R_j, T, and $\mathbf{s}$ are distributed as in $\mathsf{Game}_0(\lambda)$ with $d = 1$ and $n = k$, and Inequality (a) follows from the game-based Schwartz–Zippel lemma, since the degree of the (non-zero) polynomial $\hat{T}\check{R}_i - \hat{R}_i\check{T}$ is at most $\hat{d}(\lambda) + \check{d}(\lambda)$. Combining our two estimates above we obtain

$$\Pr[\mathsf{Game}_0(\lambda) \text{ sets } \mathsf{Bad}_0] \le \frac{1}{2}\cdot\frac{n(\lambda)}{2^{\lambda-1}} + \frac{1}{2}\cdot\frac{n(\lambda)\big(\hat{d}(\lambda)+\check{d}(\lambda)\big)}{2^{k(\lambda)}}\,.$$

$\mathsf{Game}_1 \rightsquigarrow \mathsf{Game}_2$. We argue that $\mathsf{Game}_1(\lambda)$ and $\mathsf{Game}_2(\lambda)$ are indistinguishable. Indeed, notice that the only difference between these two games is in the definition of h, but that the distribution of h is identical in both games: In $\mathsf{Game}_1(\lambda)$, h is a random group element satisfying $h \neq h_i$ for every $i \in [n]$, since $r' \neq R_i(\mathbf{s})$ for every $i \in [n]$ (by the previous game hop) and σ is a random injection. On the other hand, in $\mathsf{Game}_2(\lambda)$ the same holds by definition. Thus, $\Pr[\mathsf{Game}_1(\lambda)] = \Pr[\mathsf{Game}_2(\lambda)]$.

$\mathsf{Game}_2 \rightsquigarrow \mathsf{Game}_3$. Again, $\mathsf{Game}_2(\lambda)$ and $\mathsf{Game}_3(\lambda)$ are identical until Bad_1, which by the fundamental lemma of game playing means $|\Pr[\mathsf{Game}_2(\lambda)] - \Pr[\mathsf{Game}_3(\lambda)]| \le \Pr[\mathsf{Game}_2(\lambda) \text{ sets } \mathsf{Bad}_1]$, and

$$\begin{aligned}
\Pr[\mathsf{Game}_2(\lambda) \text{ sets } \mathsf{Bad}_1] &= \sum_{k=0}^{n(\lambda)}\Pr[\mathsf{Game}_2(\lambda) \text{ sets } \mathsf{Bad}_1 \mid n=k]\Pr[n=k]\\
&= \sum_{k=0}^{n(\lambda)}\Pr\left[\left(\bigwedge_{j=1}^{k+1}(\check{R}_j(\mathbf{s})\neq 0)\right)\wedge\left(\bigvee_{i=1}^{k}(h=h_i)\right)\;\middle|\;n=k\right]\Pr[n=k]\\
&\le \sum_{k=0}^{n(\lambda)}\sum_{i=1}^{k}\Pr\left[\left(\bigwedge_{j=1}^{k+1}(\check{R}_j(\mathbf{s})\neq 0)\right)\wedge(h=h_i)\;\middle|\;n=k\right]\Pr[n=k]\\
&\le \sum_{k=0}^{n(\lambda)}\sum_{i=1}^{k}\Pr\left[h=h_i\;\middle|\;\left(\bigwedge_{j=1}^{n+1}(\check{R}_j(\mathbf{s})\neq 0)\right)\wedge(n=k)\right]\Pr[n=k] \overset{(a)}{\le} \sum_{k=0}^{n(\lambda)}\sum_{i=1}^{k}\frac{1}{2^{\lambda-1}}\cdot\Pr[n=k] \le \frac{n(\lambda)}{2^{\lambda-1}}\,,
\end{aligned}$$

where the R_j, h, h_i and n are distributed as in $\mathsf{Game}_2(\lambda)$, and the sum over k extends only over those values such that $\Pr[n = k] > 0$. Here, Inequality (a) holds because h is sampled uniformly at random from G, a group of size $p \ge 2^{\lambda-1}$.

Game_3. We conclude by bounding the probability of winning Game_3. Observe that

$$\begin{aligned}
\Pr[\mathsf{Game}_3(\lambda)] &\leq \Pr\left[\mathsf{Game}_3(\lambda) \;\middle|\; \bigwedge_{j=1}^{n+1} (\check{R}_j(\mathbf{s}) \neq 0)\right] + \Pr\left[\bigvee_{j=1}^{n+1} (\check{R}_j(\mathbf{s}) = 0)\right] \\
&\overset{(a)}{=} \frac{1}{2} + \sum_{k=0}^{n(\lambda)} \Pr\left[\bigvee_{j=1}^{k+1} (\check{R}_j(\mathbf{s}) = 0) \;\middle|\; n = k\right] \Pr[n = k] \leq \frac{1}{2} + \sum_{k=0}^{n(\lambda)} \sum_{j=1}^{k+1} \Pr\big[\check{R}_j(\mathbf{s}) = 0 \;\big|\; n = k\big] \Pr[n = k] \\
&\overset{(b)}{\leq} \frac{1}{2} + \sum_{k=0}^{n(\lambda)} \sum_{j=1}^{k+1} \frac{\check{d}(\lambda)}{2^{k(\lambda)}} \Pr[n = k] \leq \frac{1}{2} + \frac{(n(\lambda)+1)\check{d}(\lambda)}{2^{k(\lambda)}}\,,
\end{aligned}$$

where the R_j, $\mathbf{s}$ and n are distributed as in Game_3. For Equality (a), observe that if we disregard the cases where the experiment returns true because the vector $\mathbf{s}$ is a root of one of the denominators, the challenge h is completely independent of the bit d, which means that $\mathcal{A}_2$ can do nothing but guess. Finally, Inequality (b) follows from the game-based Schwartz–Zippel lemma, recalling that $\check{R}_j$ is a (non-zero) polynomial of degree at most $\check{d}(\lambda)$ and the coordinates of $\mathbf{s}$ have min-entropy at least $k(\lambda)$.

Combining the above estimates we obtain Inequality (10) for the DUA advantage:

$$\begin{aligned}
\mathrm{Adv}^{\text{dua-ii}}_{\Gamma,\mathcal{A}}(\lambda) &= 2 \cdot \mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + 2 \cdot \Pr\big[\neg\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \;\big|\; b = 0\big] - 1 \\
&\leq 2 \cdot \mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + \frac{n(\lambda)}{2^{\lambda-1}} + \frac{n(\lambda)\big(\hat{d}(\lambda) + \check{d}(\lambda)\big)}{2^{k(\lambda)}} + \frac{2n(\lambda)}{2^{\lambda-1}} + \frac{2(n(\lambda)+1)\check{d}(\lambda)}{2^{k(\lambda)}} \\
&\leq 2 \cdot \mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + \frac{3n(\lambda)}{2^{\lambda-1}} + \frac{3(n(\lambda)+1)\big(\hat{d}(\lambda) + 3\check{d}(\lambda)\big)}{2^{k(\lambda)}}\,.
\end{aligned}$$

Algebraic unpredictability. It remains to be shown that, for every DUA-II adversary $\mathcal{A}$, the PGG source $\mathcal{S}$ defined via algorithm $\bar{\mathcal{S}}$ in Figure 20 (right) is algebraically unpredictable. To this end, let $\mathcal{A}$ be a DUA-II adversary and $\mathcal{P}$ be any predictor in the algebraic unpredictability game against $\mathcal{S}$. We prove Inequality (11) via a sequence of games. As before, we give here a short description of each game, and present their formal code in Figure 22:

$\mathsf{Game}'_0(\lambda)$ is the algebraic unpredictability game for $\mathcal{S}$ played by $\mathcal{P}$.

$\mathsf{Game}'_1(\lambda)$ is the same as $\mathsf{Game}'_0(\lambda)$, but we immediately return true if one of the denominators $\check{R}_i$ vanishes on the vector $\mathbf{s}$ returned by $\mathcal{A}_1$. We also expand this condition to include the events $(R_i(\mathbf{s}) = R_j(\mathbf{s}))$ for all $i, j \in [n]$ with $i < j$. Furthermore, we return true if the value r' coincides with $R_i(\mathbf{s})$ for some $i \in [n]$. Notice that this implies that the $R_i(\mathbf{s})$ and r' are pairwise different from here on.

$\mathsf{Game}'_2(\lambda)$ is the same as $\mathsf{Game}'_1(\lambda)$, but we rewrite the winning condition. Indeed, since we now know that no denominator $\check{R}_i(\mathbf{s})$ vanishes, in this game the Exp oracle is queried on the $n+1$ points $R_1(\mathbf{s}), \ldots, R_n(\mathbf{s}), r'$, of which the predictor must produce a valid linear combination.

$\mathsf{Game}'_3(\lambda)$ is the same as $\mathsf{Game}'_2(\lambda)$, but we implement σ via lazy sampling. In other words, we no longer compute h_i and h as evaluations of σ, but instead return random, pairwise different elements from G.

$\mathsf{Game}'_4(\lambda)$ is the same as $\mathsf{Game}'_3(\lambda)$, but we rewrite the winning condition: We clear the denominators, which is equivalent to multiplying each fraction R_i with the least common multiple D of all denominator polynomials, and evaluate the result at $\mathbf{s}$. We also set flags $\mathsf{Bad}' \leftarrow \mathsf{true}$ if one of the conditions making the experiment return true by default occurs.

$\mathsf{Game}'_5(\lambda)$ is the same as $\mathsf{Game}'_4(\lambda)$, but we remove the two conditional statements flagged with Bad'. Notice that this can be done safely, because we have cleared all denominators from the winning condition in the previous game (so that there is no danger of dividing by zero).

As before, we now relate the success probabilities of subsequent games.

Game $\mathsf{Game}'_0(\lambda)$:

$Q \leftarrow [\,]$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$; $d \twoheadleftarrow \{0,1\}$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big)$ then $L \leftarrow (\varepsilon, \varepsilon)$
else
  if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
  for $i = 1$ to n do $h_i \leftarrow \sigma(R_i(\mathbf{s}))$
  $h \leftarrow \sigma(r')$; $L \leftarrow (h_1, \ldots, h_n, h, d, \mathsf{st})$
$[x_1, \ldots, x_q] \leftarrow Q$; $(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L)$
return $\big(\sum_{i=1}^q \alpha_i x_i = 0\big)$

Game $\mathsf{Game}'_1(\lambda)$:

$Q \leftarrow [\,]$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$; $d \twoheadleftarrow \{0,1\}$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big) \vee$
  $\boxed{\big((\exists i, j \in [n])(i < j \wedge R_i(\mathbf{s}) = R_j(\mathbf{s})\big)}$
  then $\boxed{\text{return } \mathsf{true}}$
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
$\boxed{\text{if } (r' \in \{R_i(\mathbf{s}) \mid i \in [n]\}) \text{ then return } \mathsf{true}}$
for $i = 1$ to n do $h_i \leftarrow \sigma(R_i(\mathbf{s}))$
$h \leftarrow \sigma(r')$; $L \leftarrow (h_1, \ldots, h_n, h, d, \mathsf{st})$
$[x_1, \ldots, x_q] \leftarrow Q$; $(\alpha_1, \ldots, \alpha_q) \twoheadleftarrow \mathcal{P}(\pi, L)$
return $\big(\sum_{i=1}^q \alpha_i x_i = 0\big)$

Game $\mathsf{Game}'_2(\lambda)$:

$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$; $d \twoheadleftarrow \{0,1\}$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big) \vee$
  $\big((\exists i, j \in [n])(i < j \wedge R_i(\mathbf{s}) = R_j(\mathbf{s})\big)$
  then return true
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
if $(r' \in \{R_i(\mathbf{s}) \mid i \in [n]\})$ then return true
for $i = 1$ to n do $h_i \leftarrow \sigma(R_i(\mathbf{s}))$
$h \leftarrow \sigma(r')$; $L \leftarrow (h_1, \ldots, h_n, h, d, \mathsf{st})$
$\boxed{(\alpha_1, \ldots, \alpha_{n+1}) \twoheadleftarrow \mathcal{P}(\pi, L)}$
$\boxed{\text{return } \big(\sum_{i=1}^n \alpha_i R_i(\mathbf{s}) + \alpha_{n+1} r' = 0\big)}$

Game $\mathsf{Game}'_3(\lambda)$:

$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $d \twoheadleftarrow \{0,1\}$; $\boxed{S \leftarrow \emptyset}$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big) \vee$
  $\big((\exists i, j \in [n])(i < j \wedge R_i(\mathbf{s}) = R_j(\mathbf{s})\big)$
  then return true
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
if $(r' \in \{R_i(\mathbf{s}) \mid i \in [n]\})$ then return true
for $i = 1$ to n do $\boxed{h_i \twoheadleftarrow \mathsf{G} \setminus S;\ S \leftarrow S \cup \{h_i\}}$
$\boxed{h \twoheadleftarrow \mathsf{G} \setminus S;}$ $L \leftarrow (h_1, \ldots, h_n, h, d, \mathsf{st})$
$(\alpha_1, \ldots, \alpha_{n+1}) \twoheadleftarrow \mathcal{P}(\pi, L)$
return $\big(\sum_{i=1}^n \alpha_i R_i(\mathbf{s}) + \alpha_{n+1} r' = 0\big)$

Game $\mathsf{Game}'_4(\lambda)$:

$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $d \twoheadleftarrow \{0,1\}$; $S \leftarrow \emptyset$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $\big((\exists i \in [n+1])(\check{R}_i(\mathbf{s}) = 0)\big) \vee$
  $\big((\exists i, j \in [n])(i < j \wedge R_i(\mathbf{s}) = R_j(\mathbf{s})\big)$
  then $\boxed{\mathsf{Bad}' \leftarrow \mathsf{true};}$ return true
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
if $(r' \in \{R_i(\mathbf{s}) \mid i \in [n]\})$ then $\boxed{\mathsf{Bad}' \leftarrow \mathsf{true};}$ return true
for $i = 1$ to n do $h_i \twoheadleftarrow \mathsf{G} \setminus S;\ S \leftarrow S \cup \{h_i\}$
$h \twoheadleftarrow \mathsf{G} \setminus S$; $L \leftarrow (h_1, \ldots, h_n, h, d, \mathsf{st})$
$(\alpha_1, \ldots, \alpha_{n+1}) \twoheadleftarrow \mathcal{P}(\pi, L)$; $\boxed{D \leftarrow \mathrm{lcm}(\check{R}_1, \ldots, \check{R}_{n+1})}$
$\boxed{\text{return } \big(\sum_{i=1}^n \alpha_i (R_i D)(\mathbf{s}) + \alpha_{n+1} (r' D)(\mathbf{s}) = 0\big)}$

Game $\mathsf{Game}'_5(\lambda)$:

$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $d \twoheadleftarrow \{0,1\}$; $S \leftarrow \emptyset$
$(R_1, \ldots, R_n, T, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(\pi)$; $R_{n+1} \leftarrow T$
for $i = 1$ to m do $\mathbf{s}[i] \twoheadleftarrow \mathcal{A}_1(i, \pi)$
if $(d = 0)$ then $r' \twoheadleftarrow \mathbb{Z}_p$ else $r' \leftarrow T(\mathbf{s})$
for $i = 1$ to n do $h_i \twoheadleftarrow \mathsf{G} \setminus S;\ S \leftarrow S \cup \{h_i\}$
$h \twoheadleftarrow \mathsf{G} \setminus S$; $L \leftarrow (h_1, \ldots, h_n, h, d, \mathsf{st})$
$(\alpha_1, \ldots, \alpha_{n+1}) \twoheadleftarrow \mathcal{P}(\pi, L)$; $D \leftarrow \mathrm{lcm}(\check{R}_1, \ldots, \check{R}_{n+1})$
return $\big(\sum_{i=1}^n \alpha_i (R_i D)(\mathbf{s}) + \alpha_{n+1} (r' D)(\mathbf{s}) = 0\big)$

Figure 22 — Code of the intermediate games in the proof of the algebraic unpredictability bound (11) in Theorem 6.1.

$\mathsf{Game}'_0 \rightsquigarrow \mathsf{Game}'_1$. Observe that, by definition, $\mathsf{Game}'_1(\lambda)$ returns true whenever $\mathsf{Game}'_0(\lambda)$ does. Thus, clearly, $\Pr[\mathsf{Game}'_0(\lambda)] \leq \Pr[\mathsf{Game}'_1(\lambda)]$.

$\mathsf{Game}'_1 \rightsquigarrow \mathsf{Game}'_2$. Observe that $\Pr[\mathsf{Game}'_1(\lambda)] = \Pr[\mathsf{Game}'_2(\lambda)]$, because they are fundamentally identical. Indeed, we have only used the fact that in $\mathsf{Game}'_2(\lambda)$ the source makes $n+1$ queries, i.e., we know exactly which elements are in Q, how many scalars the predictor must return, and what the linear combination must look like.

$\mathsf{Game}'_2 \rightsquigarrow \mathsf{Game}'_3$. We now argue that $\Pr[\mathsf{Game}'_2(\lambda)] = \Pr[\mathsf{Game}'_3(\lambda)]$. To do so, observe that the distributions of h_i and h are identical in both games. Indeed, collisions between the $R_i(\mathbf{s})$ and r' are ruled out in both games. This means that, in both cases, h_i and h are pairwise different random elements of G: In $\mathsf{Game}'_2(\lambda)$ this holds because σ is a random injection, and in $\mathsf{Game}'_3(\lambda)$ this is true by definition.

$\mathsf{Game}'_3 \rightsquigarrow \mathsf{Game}'_4$. Notice that clearing the denominators in the equality in the return statement does not alter the winning condition, because points $\mathbf{s} \in \mathbb{Z}_p^m$ with $D(\mathbf{s}) = 0$ are already excluded by the first if-statement. Thus, $\Pr[\mathsf{Game}'_3(\lambda)] = \Pr[\mathsf{Game}'_4(\lambda)]$.

$\mathsf{Game}'_4 \rightsquigarrow \mathsf{Game}'_5$. Observe that $\mathsf{Game}'_4(\lambda)$ and $\mathsf{Game}'_5(\lambda)$ are identical until Bad'. To bound the probability of $\mathsf{Game}'_4(\lambda)$ setting Bad', observe that this event happens either if one of the denominators $\check{R}_i(\mathbf{s})$ vanishes, or if two rational functions $R_i(\mathbf{s})$ and $R_j(\mathbf{s})$ take the same value, or if r' happens to coincide with one of the $R_i(\mathbf{s})$. Call these events E_1, E_2 and E_3, respectively. Then observe that

$$\Pr[\mathsf{Game}'_4(\lambda) \text{ sets } \mathsf{Bad}'] \leq \Pr[E_1] + \Pr[E_2 \wedge \neg E_1] + \Pr[E_3 \wedge \neg E_1]\,.$$

We separately bound the three probabilities above. For the first term, observe that

$$\Pr[E_1] = \sum_{k=0}^{n(\lambda)} \Pr[E_1 \mid n = k] \Pr[n = k] \leq \sum_{k=0}^{n(\lambda)} \sum_{i=1}^{k+1} \Pr\big[\check{R}_i(\mathbf{s}) = 0 \,\big|\, n = k\big] \Pr[n = k] \leq \frac{(n(\lambda)+1)\check{d}(\lambda)}{2^{k(\lambda)}}\,,$$

where the last inequality follows from the game-based Schwartz–Zippel lemma and, as usual, the sum over k extends over all indices such that $\Pr[n = k] > 0$. For the second term, we proceed as follows:

$$\begin{aligned}
\Pr[E_2 \wedge \neg E_1] &= \sum_{k=0}^{n(\lambda)} \Pr[E_2 \wedge \neg E_1 \mid n = k] \Pr[n = k] \\
&\leq \sum_{k=0}^{n(\lambda)} \sum_{j=1}^{k} \sum_{i=1}^{j-1} \Pr\big[\big(R_i(\mathbf{s}) = R_j(\mathbf{s})\big) \wedge \neg E_1 \,\big|\, n = k\big] \Pr[n = k] \\
&\leq \sum_{k=0}^{n(\lambda)} \sum_{j=1}^{k} \sum_{i=1}^{j-1} \Pr\Big[\big(\hat{R}_i(\mathbf{s})\check{R}_j(\mathbf{s}) = \hat{R}_j(\mathbf{s})\check{R}_i(\mathbf{s})\big) \wedge \neg E_1 \,\Big|\, n = k\Big] \Pr[n = k] \qquad (12) \\
&\leq \sum_{k=0}^{n(\lambda)} \sum_{j=1}^{k} \sum_{i=1}^{j-1} \Pr\big[\hat{R}_i(\mathbf{s})\check{R}_j(\mathbf{s}) = \hat{R}_j(\mathbf{s})\check{R}_i(\mathbf{s}) \,\big|\, n = k\big] \Pr[n = k] \\
&\overset{(a)}{\leq} \sum_{k=0}^{n(\lambda)} \sum_{j=1}^{k} \sum_{i=1}^{j-1} \frac{\hat{d}(\lambda) + \check{d}(\lambda)}{2^{k(\lambda)}} \Pr[n = k] \leq \frac{n(\lambda)^2\big(\hat{d}(\lambda) + \check{d}(\lambda)\big)}{2^{k(\lambda)+1}}\,.
\end{aligned}$$

Here, Inequality (a) holds again because of the game-based Schwartz–Zippel lemma, keeping in mind that $\hat{R}_i\check{R}_j - \hat{R}_j\check{R}_i$ is a non-zero polynomial of degree at most $\hat{d}(\lambda) + \check{d}(\lambda)$ for all indices $1 \leq i, j \leq k \leq n(\lambda)$. Finally, for the third term we have

$$\Pr[E_3 \wedge \neg E_1] = \frac{1}{2}\sum_{k=0}^{n(\lambda)} \Pr[E_3 \wedge \neg E_1 \mid (d=0) \wedge (n=k)] \Pr[n=k] \\ + \frac{1}{2}\sum_{k=0}^{n(\lambda)} \Pr[E_3 \wedge \neg E_1 \mid (d=1) \wedge (n=k)] \Pr[n=k] \,.$$

We estimate the two coefficients separately. To do so, we again use the game-based Schwartz–Zippel lemma:

$$\begin{aligned}\Pr[E_3 \wedge \neg E_1 \mid (d=0) \wedge (n=k)] &\leq \sum_{i=1}^{k} \Pr\big[\big(r' = R_i(\mathbf{s})\big) \wedge \neg E_1 \;\big|\; (d=0) \wedge (n=k)\big] \\ &\leq \sum_{i=1}^{k} \Pr[r' = R_i(\mathbf{s}) \mid \neg E_1 \wedge (d=0) \wedge (n=k)] \leq \frac{k}{2^{\lambda-1}} \leq \frac{n(\lambda)}{2^{\lambda-1}} \,,\end{aligned}$$

and

$$\begin{aligned}\Pr[E_3 \wedge \neg E_1 \mid (d=1) \wedge (n=k)] &\leq \sum_{i=1}^{k} \Pr\big[\big(T(\mathbf{s}) = R_i(\mathbf{s})\big) \wedge \neg E_1 \;\big|\; (d=1) \wedge (n=k)\big] \\ &= \sum_{i=1}^{k} \Pr\big[\big(\hat{T}(\mathbf{s})\check{R}_i(\mathbf{s}) = \check{T}(\mathbf{s})\hat{R}_i(\mathbf{s})\big) \wedge \neg E_1 \;\big|\; (d=1) \wedge (n=k)\big] \\ &\leq \sum_{i=1}^{k} \Pr\big[\hat{T}(\mathbf{s})\check{R}_i(\mathbf{s}) = \check{T}(\mathbf{s})\hat{R}_i(\mathbf{s}) \;\big|\; (d=1) \wedge (n=k)\big] \\ &\leq \sum_{i=1}^{k} \frac{\hat{d}(\lambda)+\check{d}(\lambda)}{2^{k(\lambda)}} = \frac{k\big(\hat{d}(\lambda)+\check{d}(\lambda)\big)}{2^{k(\lambda)}} \leq \frac{n(\lambda)\big(\hat{d}(\lambda)+\check{d}(\lambda)\big)}{2^{k(\lambda)}} \,.\end{aligned}$$

The overall bound on the probability of $\mathsf{Game}'_4(\lambda)$ setting Bad' can be obtained by collecting the terms above:

$$\begin{aligned}\Pr[\mathsf{Game}'_4(\lambda) \text{ sets } \mathsf{Bad}'] &\leq \frac{(n(\lambda)+1)\check{d}(\lambda)}{2^{k(\lambda)}} + \frac{n(\lambda)^2\big(\hat{d}(\lambda)+\check{d}(\lambda)\big)}{2^{k(\lambda)+1}} + \frac{n(\lambda)}{2^{\lambda}} + \frac{n(\lambda)\big(\hat{d}(\lambda)+\check{d}(\lambda)\big)}{2^{k(\lambda)+1}} \\ &\leq \frac{n(\lambda)}{2^{\lambda}} + \frac{3n(\lambda)^2\big(\hat{d}(\lambda)+\check{d}(\lambda)\big)}{2^{k(\lambda)}} \,.\end{aligned}$$

Game'_5. To conclude, we are left with bounding the winning probability in $\mathsf{Game}'_5(\lambda)$. To do so, observe that, for every tuple $(\alpha_1, \ldots, \alpha_{n+1})$ returned by $\mathcal{P}$, and with $U \coloneqq X_{m+1}$ or $U \coloneqq T$ depending on d, the polynomial $\alpha_1(R_1 D) + \cdots + \alpha_n(R_n D) + \alpha_{n+1}(UD)$ is non-zero (because T is linearly independent from $R_1, \ldots, R_n$, and these are linearly independent) and of degree at most $\deg(D) + \hat{d}(\lambda) + 1 \leq n(\lambda)\check{d}(\lambda) + \hat{d}(\lambda) + 1$. Thus, by the game-based Schwartz–Zippel lemma,

$$\Pr[\mathsf{Game}'_5(\lambda)] \leq \frac{n(\lambda)\check{d}(\lambda) + \hat{d}(\lambda) + 1}{2^{k(\lambda)}} \leq \frac{n(\lambda)^2\big(\hat{d}(\lambda)+\check{d}(\lambda)+1\big)}{2^{k(\lambda)}} \,.$$

Collecting all our estimates above, we obtain the claimed bound (11). □

Remark. We observe that Inequality (11) is not tight in general. Consider for instance the Decisional q-Diffie–Hellman Inversion problem (q-DDHI, with q a polynomial), which is a special case of the DUA-II where $\mathcal{A}_0$ sets $n = q(\lambda)$ and chooses $R_i(X) \coloneqq X^i$ for $i \in [n]$ and $T(X) \coloneqq 1/X$. Then Inequality (12) can

be tightened, leading to an overall algebraic unpredictability loss that is *quadratic* in q. More precisely, recall that (12) bounds the probability that any of the events $(R_i(\mathbf{s}) = R_j(\mathbf{s}))$, with $i, j \in [n]$ and $i < j$, occur. These events could very well be all different, thus resulting in at most $\binom{n(\lambda)}{2} \leq n(\lambda)^2/2$ conditions; on the other hand, for q-DDHI many of these events overlap. Indeed, for this particular instance the event $(R_i(s) = R_j(s))$ only depends on the difference $j - i$ between the degrees of the monomials, which means that there are only $q(\lambda) - 1$ collision events to consider. Therefore, for the q-DDHI assumption the algebraic unpredictability bound becomes

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \leq \frac{q(\lambda)}{2^\lambda} + \frac{(q(\lambda)+1)(q(\lambda)+2)}{2^{k(\lambda)}}\,.$$

6.2 Building UCEs

In this section we show how to construct UCEs based on PGGs and LDDs. We consider UCEs for statistically unpredictable and *split* sources [BHK13], whose definition we recall below. Split sources are required to make distinct queries to prevent iO-based attacks. BHK use split sources to prove security of a number of applications, including RKA security, point-function obfuscation, and storage-auditing protocols, as well as several other applications that rely on computationally unpredictable split sources.

Note that split UCE sources allow for limited post-processing of the outputs of the hashing oracle. However, this feature of split sources is *not* used in any of the applications discussed by BHK: The very simple $\mathcal{S}_1$ that merely returns its input is sufficient for proving the security of all applications of split sources considered in [BHK13].[14] We call split sources of this type *simple*. Our result in this section allows to recover applications of UCEs with respect to simple split sources under PGGs and LDDs.[15]

(Simple) Split sources. A UCE source $\mathcal{S}$ is called split if there exist PPT algorithms $\mathcal{S}_0$ and $\mathcal{S}_1$ such that $\mathcal{S}$ takes the form in Figure 23 (top left). Here, $\mathcal{S}_0$ returns a vector $\mathbf{x}$ whose entries are required to be pairwise distinct, and some leakage L_0. We write $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$ if $\mathcal{S}$ is a split source constructed from algorithms $\mathcal{S}_0$ and $\mathcal{S}_1$ as above, and we denote by $\mathbf{S}^{\text{splt}}$ the class of all such split sources. We further define the class $\mathbf{S}^{\text{ssplt}} \subseteq \mathbf{S}^{\text{splt}}$ of simple split sources, which are split sources where $\mathcal{S}_1$ merely returns $L_1 = \mathbf{y}$. Similarly to PGG masking sources, we write $\bar{\mathcal{S}} := \mathcal{S}_0$ for the source defining the simple split source $\mathcal{S}$.

Construction. We construct a UCE $\mathsf{GOR}[\Gamma, \mathsf{H}]$ in a modular way in terms of a PGG Γ and an underlying LDD H as shown in Figure 23 (top right). Our construction is inspired by the correlated-input (CI) secure hash of Goyal, O'Neill, and Rao (GOR) [GOR11], where outputs of a hash function are required to look random on high-entropy, but possibly correlated, inputs. GOR show that the hash function $x \mapsto g^{1/(x+hk)}$ (associated to the conjectural LDD $(hk, x) \mapsto 1/(x + hk)$), where $hk \twoheadleftarrow \mathbb{Z}_p$ is the hash key, is non-adaptively CI secure for polynomially induced correlations under the q-DDH assumption. This assumption falls under DUA-II, and thus Theorem 6.1 allows us to recover this result. However, this falls short of achieving split UCE security, since the hash inputs are polynomially induced.

Based on the conjectured existence of LDDs for all unpredictable sources, we show that $\mathsf{GOR}[\Gamma, \mathsf{H}]$ is a fully secure UCE (beyond polynomial sources) for all statistically unpredictable and split sources. In the GGM, we can account for preprocessing too. Looking ahead into the proof, there is a close correspondence between the class of sources for which one achieves LDD security and split UCE security. That is, if H is an LDD for a certain class of sources (e.g., $\mathbf{S}^{\text{sup}}$ or $\mathbf{S}^{\text{low}}$), then $\mathsf{GOR}[\Gamma, \mathsf{H}]$ is UCE secure for an analogous source class. Thus, we obtain an unconditional result for low-degree split UCE sources, since Theorem 5.4 shows that $1/(x + hk)$ is LDD$[\mathbf{S}^{\text{low}}]$ secure.

[14] Interestingly, this simplification provides another avenue to circumvent iO-based attacks that exploit repetitions in $\mathbf{x}$.

[15] We note, however, that in iterative constructions of block-ciphers from hash functions [BHK14], or indeed in domain extenders for hash functions [ST17], adaptive calls to the hash function seem to be necessary.

Split source $\mathcal{S}^{\text{HASH}}(\pi)$:
$(\mathbf{x}, L_0) \twoheadleftarrow \mathcal{S}_0(\pi)$
for $i = 1$ to $|\mathbf{x}|$ do $\mathbf{y}[i] \leftarrow \text{HASH}(\mathbf{x}[i])$
$L_1 \twoheadleftarrow \mathcal{S}_1(\pi, \mathbf{y})$; $L \leftarrow (L_0, L_1)$
return L

$\mathsf{GOR}[\Gamma, \mathsf{H}].\mathsf{Setup}(1^\lambda)$:
$\pi \twoheadleftarrow \Gamma(1^\lambda)$; return π

$\mathsf{GOR}[\Gamma, \mathsf{H}]((h, hk), x)$:
$y \leftarrow \mathsf{H}(hk, x)$; return h^y

$\mathsf{GOR}[\Gamma, \mathsf{H}].\mathsf{KGen}(\pi)$:
$(\circ, g_0, p) \leftarrow \pi$
$r \twoheadleftarrow \mathbb{Z}_p^*$; $h \leftarrow g_0^r$
$hk \twoheadleftarrow \mathsf{H}.\mathsf{KGen}(\pi)$
return (h, hk)

Auxiliary dUber source $\bar{\mathcal{S}}'(\pi)$:
$(\circ, g_0, p) \leftarrow \pi$; $r' \twoheadleftarrow \mathbb{Z}_p^*$; $hk \twoheadleftarrow \mathsf{H}.\mathsf{KGen}(\pi)$
$(\mathbf{x}, L) \twoheadleftarrow \bar{\mathcal{S}}(\pi)$; $n \leftarrow |\mathbf{x}|$; $L' \leftarrow (L, hk)$
for $i = 1$ to n do $\mathbf{x}'[i] \leftarrow r' \cdot \mathsf{H}(hk, \mathbf{x}[i])$
$\mathbf{x}'[n+1] \leftarrow r'$; return $(\mathbf{x}', L')$

PGG distinguisher $\mathcal{D}'(\pi, (\mathbf{y}', L'))$:
$n + 1 \leftarrow |\mathbf{y}'|$; $h \leftarrow \mathbf{y}'[n+1]$
$(L, hk) \leftarrow L'$; $\mathbf{y} \leftarrow \mathbf{y}'[1..n]$
$b' \twoheadleftarrow \mathcal{D}(\pi, (h, hk), (\mathbf{y}, L))$
return b'

Figure 23 — *Top left:* Structure of the split source $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$ associated to $\mathcal{S}_0$ and $\mathcal{S}_1$. In simple split sources, algorithm $\mathcal{S}_1$ returns $L_1 = \mathbf{y}$. *Top right:* The UCE $\mathsf{GOR}[\Gamma, \mathsf{H}]$ built from a PGG Γ and an LDD H with $\mathsf{H}.\mathsf{Setup} = \Gamma$. *Bottom:* Reduction from a UCE adversary $(\mathcal{S}, \mathcal{D})$ to a PGG adversary $(\mathcal{S}', \mathcal{D}')$.

Theorem 6.2 (PGG $\wedge$ LDD $\Longrightarrow$ UCE$[\mathbf{S}^{\text{sup}} \cap \mathbf{S}^{\text{ssplt}}]$)**.** *Let Γ be a computational group scheme and H a hash function family with $\mathsf{H}.\mathsf{Setup} = \Gamma$.*[16] *Consider the hash function family $\mathsf{GOR}[\Gamma, \mathsf{H}]$ based on Γ and H as defined in Figure 23 (top right). If Γ is PGG$[\mathbf{S}^{\text{alg}} \cap \mathbf{S}^{\text{duber}}]$ secure and H is LDD$[\mathbf{S}^{\text{sup}}]$ secure, then $\mathsf{GOR}[\Gamma, \mathsf{H}]$ is UCE$[\mathbf{S}^{\text{sup}} \cap \mathbf{S}^{\text{ssplt}}]$ secure. More precisely, for any adversary $(\mathcal{S}, \mathcal{D})$ in the UCE game for $\mathsf{GOR}[\Gamma, \mathsf{H}]$, there are an adversary $(\mathcal{S}', \mathcal{D}')$ in the PGG game for Γ and an adversary $(\mathcal{T}, \mathcal{A})$ in the LDD game for H such that*

$$\text{Adv}^{\text{uce}}_{\mathsf{GOR}[\Gamma,\mathsf{H}],\mathcal{S},\mathcal{D}}(\lambda) \le \text{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S}',\mathcal{D}'}(\lambda) + q(\lambda)^2 \cdot \text{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{A}}(\lambda) + \frac{q(\lambda)^2 + 1}{2^{\lambda-1}} \,. \tag{13}$$

Furthermore, $\mathcal{S}' \in \mathbf{S}^{\text{alg}} \cap \mathbf{S}^{\text{duber}}$. More precisely, for any predictor $\mathcal{P}$ in the AlgPred game for $(\Gamma, \mathcal{S}')$, there are adversaries $(\mathcal{T}, \mathcal{A})$ and $(\mathcal{T}, \mathcal{B})$ in the LDD game for H such that

$$\text{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S}',\mathcal{P}}(\lambda) \le q(\lambda)^2 \cdot \text{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{A}}(\lambda) + \text{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{B}}(\lambda) \,. \tag{14}$$

Moreover, $\mathcal{T} \in \mathbf{S}^{\text{sup}}$. More precisely, for any predictor $\mathcal{P}$ in the Pred game for $(\mathsf{H}, \mathcal{T})$, there is a predictor $\mathcal{P}'$ in the Pred game for $(\mathsf{GOR}[\Gamma, \mathsf{H}], \mathcal{S})$ such that

$$\text{Adv}^{\text{pred}}_{\mathsf{H},\mathcal{T},\mathcal{P}}(\lambda) \le \text{Adv}^{\text{pred}}_{\mathsf{GOR}[\Gamma,\mathsf{H}],\mathcal{S},\mathcal{P}'}(\lambda) \,. \tag{15}$$

In the above, $q(\lambda)$ is an upper bound on the number of queries made by $\mathcal{S}$ to its HASH *oracle.*

Proof Overview. Let $(\mathcal{S}, \mathcal{D})$ be an adversary against UCE security of $\mathsf{GOR}[\Gamma, \mathsf{H}]$. We build $(\mathcal{S}', \mathcal{D}')$ against the PGG security of the underlying group as shown in Figure 23 (bottom).

ADVANTAGE BOUND. Let b denote the challenge bit in the PGG game. Then it is easy to see that

$$\Pr\left[\text{PGG}^{\mathcal{S}',\mathcal{D}'}_{\Gamma}(\lambda) \,\middle|\, b = 1\right] = \Pr\left[\text{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{GOR}[\Gamma,\mathsf{H}]}(\lambda) \,\middle|\, b = 1\right] .$$

Indeed, when $b = 1$ the exponentiation oracle is implemented via the real group operation. Hash values are multiplied by rr', which is random in $\mathbb{Z}_p^*$ and can thus be replaced by a random $r \in \mathbb{Z}_p^*$. Thus the UCE source and distinguisher are run as they would be in the UCE game with respect to the real hash function.

We next claim that

$$\Pr\left[\text{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{GOR}[\Gamma,\mathsf{H}]}(\lambda) \,\middle|\, b = 0\right] \le \Pr\left[\text{PGG}^{\mathcal{S}',\mathcal{D}'}_{\Gamma}(\lambda) \,\middle|\, b = 0\right] + q(\lambda)^2 \cdot \text{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{A}}(\lambda) + \frac{q(\lambda)^2 + 1}{2^{\lambda-1}} \,.$$

[16] Notice that our candidate construction from Figure 15 (bottom) can be easily modified to be of this form.

This follows from the fact that when $b = 0$, the Exp oracle returns random values subject to injectivity. We first transition to a game where Exp implements a random function. Using the PRP/PRF Switching Lemma, we incur an additive loss of $q(\lambda)^2/2^{\lambda-1}$ (since $\mathcal{S}'$ makes at most $q(\lambda)+1$ many queries). We modify this game further and replace the random function with a forgetful random function. The two games are identical unless there is a collision in the inputs to the random function. We may bound the probability of this event via the collision probability of the LDD H, which itself can be bounded in terms of the LDD advantage: Consider an adversary $\mathcal{A}$ that sets the coefficients of two indices to $+1$ and -1, the remaining coefficients to 0, and $\alpha_0 = 0$. The LDD source here is (virtually) identical to the UCE source. Thus any LDD predictor can be converted into a UCE predictor: Simply ignore the hash values and run the LDD predictor. This justifies the final inequality in the theorem. The final game that we arrive at is equivalent to the UCE game with respect to a random oracle (recall that the split source outputs distinct inputs), except that we must ensure that the group element in the hash key is a generator of G.

Algebraic unpredictability. We now show that the PGG source constructed above is algebraically unpredictable. Consider a modified algebraic prediction game where Exp returns random group elements, still subject to injectivity but not respecting equality across inputs. These two games are identical unless there is a collision among the inputs. We may bound the probability of collision via the LDD adversary $\mathcal{A}$ from above. This incurs a loss of $q(\lambda)^2$ times the LDD advantage of $\mathcal{A}$.

We now rely on the LDD security of the hash function to bound the probability of winning this modified algebraic unpredictability game. Suppose there exists an algebraic predictor $\mathcal{P}$ against $(\Gamma, \mathcal{S}')$. We construct an LDD source $\mathcal{T}$ and an LDD adversary $\mathcal{B}$ as follows. Source $\mathcal{T}$ again is virtually identical to $\mathcal{S}$, and is therefore unpredictable as shown above. Adversary $\mathcal{B}$ receives a hash key and leakage, and simulates the group elements that the algebraic predictor $\mathcal{P}$ in the modified game expects randomly but subject to injectivity. Together with the collision bound above, this establishes the second inequality stated in the theorem. ■

Proof. Given an adversary $(\mathcal{S}, \mathcal{D})$ in the UCE game for $\mathsf{GOR}[\Gamma, \mathsf{H}]$, define the PGG dUber source $\mathcal{S}'$ via auxiliary algorithm $\bar{\mathcal{S}}'$ and the distinguisher $\mathcal{D}'$ as shown in Figure 23 (bottom).

dUber structure. By construction, it is clear that $\bar{\mathcal{S}}'$ makes no Exp query and returns a vector $\mathbf{x}$ and leakage L. Thus, $\mathcal{S}' \in \mathbf{S}^{\text{duber}}$.

Advantage bound. To prove Inequality (13), first recall that

$$\mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S}',\mathcal{D}'}(\lambda) = \Pr\left[\mathrm{PGG}^{\mathcal{S}',\mathcal{D}'}_{\Gamma}(\lambda) \;\middle|\; b = 1\right] - \Pr\left[\neg\mathrm{PGG}^{\mathcal{S}',\mathcal{D}'}_{\Gamma}(\lambda) \;\middle|\; b = 0\right].$$

We study the two summands separately. For the first term, consider the following sequence of games (the formal description of which can be found in Figure 24 (top)):

$\mathsf{Game}_{1,0}(\lambda)$ is the PGG game for Γ played by $(\mathcal{S}', \mathcal{D}')$ with bit $b = 1$ fixed.
$\mathsf{Game}_{1,1}(\lambda)$ is the same as $\mathsf{Game}_{1,0}(\lambda)$, but we sample the exponent rr' of g_0, with $r, r' \in \mathbb{Z}_p^*$, as a single exponent $r \in \mathbb{Z}_p^*$. By direct inspection, this game coincides with the UCE game for $\mathsf{GOR}[\Gamma, \mathsf{H}]$ played by $(\mathcal{S}, \mathcal{D})$, with bit $b = 1$ fixed.

We now argue that these two games are indistinguishable. Indeed, notice that the exponents of g_0 in the first game are $rr'\mathsf{H}(hk, \mathbf{x}[i])$ and rr', respectively. Since r and r' are uniformly random in $\mathbb{Z}_p^*$, so is their product rr', which means that we can replace rr' with a uniformly random $r \in \mathbb{Z}_p^*$. For the first term we therefore conclude $\Pr\left[\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{GOR}[\Gamma,\mathsf{H}]}(\lambda) \;\middle|\; b = 1\right] = \Pr\left[\mathrm{PGG}^{\mathcal{S}',\mathcal{D}'}_{\Gamma}(\lambda) \;\middle|\; b = 1\right]$.

We now study the second term in the sum above. To do so, consider the following sequence of games (the formal description of which can be found in Figure 24 (bottom)):

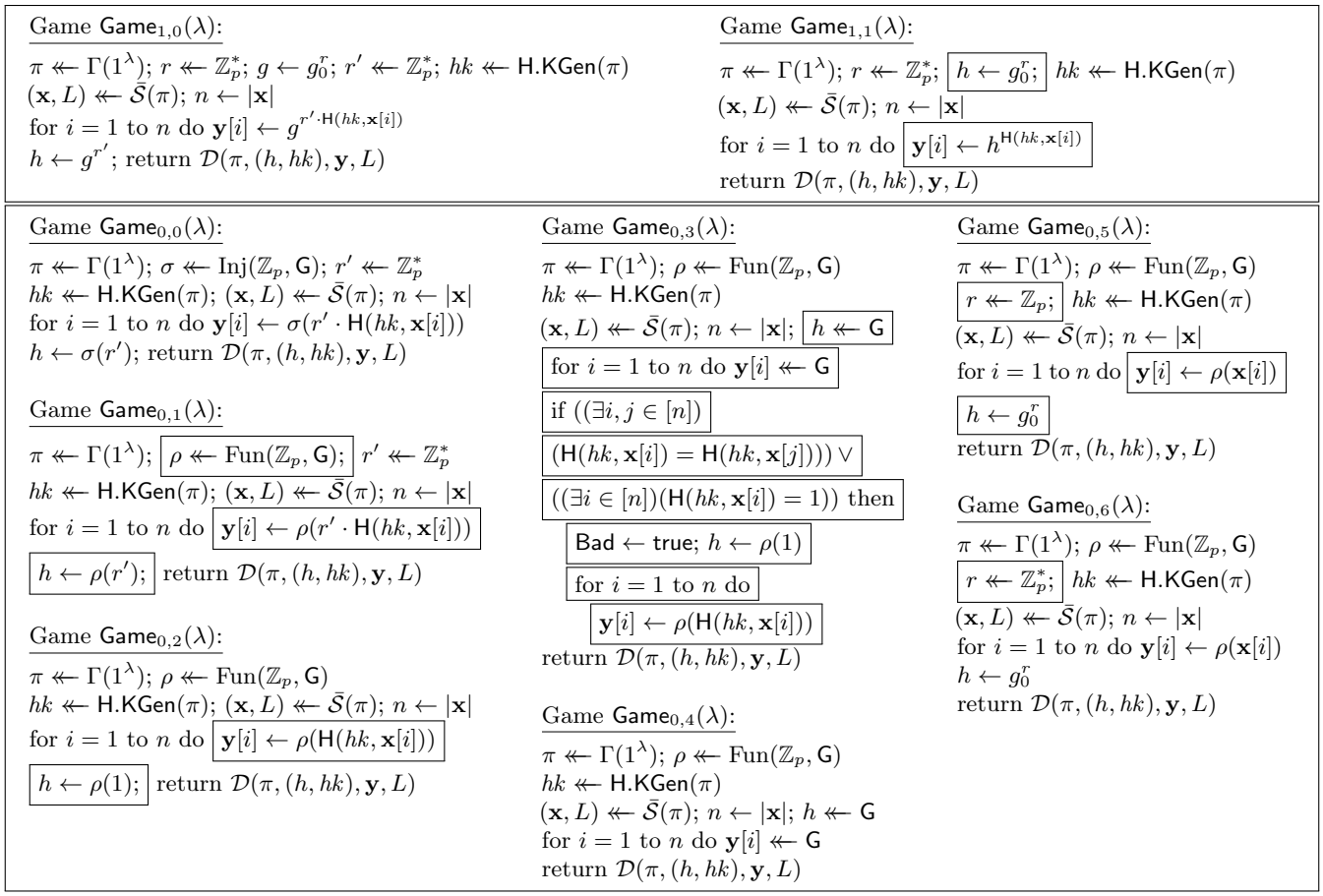

Figure 24 — Code of the intermediate games in the proof of the UCE advantage bound (13) in Theorem 6.2.

$\mathsf{Game}_{0,0}(\lambda)$ is the PGG game for Γ played by $(\mathcal{S}', \mathcal{D}')$ with bit $b = 0$ fixed and inverted winning condition.

$\mathsf{Game}_{0,1}(\lambda)$ is the same as $\mathsf{Game}_{0,0}(\lambda)$, but we substitute the random injection $\sigma \in \mathrm{Fun}(\mathbb{Z}_p, \mathsf{G})$ with a random function $\rho \in \mathrm{Fun}(\mathbb{Z}_p, \mathsf{G})$.

$\mathsf{Game}_{0,2}(\lambda)$ is the same as $\mathsf{Game}_{0,1}(\lambda)$, but we do not multiply the inputs to ρ by r'. Accordingly, we no longer sample $r' \in \mathbb{Z}_p^*$, because it isn't used anywhere in the game.

$\mathsf{Game}_{0,3}(\lambda)$ is the same as $\mathsf{Game}_{0,2}(\lambda)$, but sample h and $\mathbf{y}[i]$ at random from G, instead of evaluating ρ. Only if there is some collision among the inputs of ρ we restore the original values of h and $\mathbf{y}[i]$.

$\mathsf{Game}_{0,4}(\lambda)$ is the same as $\mathsf{Game}_{0,3}(\lambda)$, but we no longer check for collisions among the inputs of ρ.

$\mathsf{Game}_{0,5}(\lambda)$ is the same as $\mathsf{Game}_{0,4}(\lambda)$, but we compute h as $h = g_0^r$ for a random $r \in \mathbb{Z}_p$, and we let $\mathbf{y}[i]$ be $\mathbf{y}[i] = \rho(\mathbf{x}[i])$.

$\mathsf{Game}_{0,6}(\lambda)$ is the same as $\mathsf{Game}_{0,5}(\lambda)$, but we sample r at random from $\mathbb{Z}_p^*$ instead of $\mathbb{Z}_p$. This game now coincides with the UCE game for $\mathsf{GOR}[\Gamma, \mathsf{H}]$ played by $(\mathcal{S}, \mathcal{D})$, with bit $b = 0$ fixed and inverted winning condition.

As before, we now relate the success probabilities of subsequent games.

$\mathsf{Game}_{0,0} \rightsquigarrow \mathsf{Game}_{0,1}$. The difference between the success probabilites of these two games is given by the PRP/PRF Switching Lemma. Given that the random permutation/function is evaluated on at most $q(\lambda)+1$ many inputs, we have $|\Pr[\mathsf{Game}_{0,0}(\lambda)] - \Pr[\mathsf{Game}_{0,1}(\lambda)]| \leq (q(\lambda)+1)q(\lambda)/2^\lambda \leq q(\lambda)^2/2^{\lambda-1}$.

$\mathsf{Game}_{0,1} \rightsquigarrow \mathsf{Game}_{0,2}$. Notice that since $r' \in \mathbb{Z}_p^*$, multiplication by r' acts as a permutation on $\mathbb{Z}_p$. Therefore, inputs to ρ are equal in $\mathsf{Game}_{0,1}(\lambda)$ if and only if they are equal in $\mathsf{Game}_{0,2}(\lambda)$, which means that $\mathbf{y}[i]$ and h have the same distribution in both games: In both cases they are uniformly random, but repeat if hashes collide or return 1. We thus conclude that $\mathsf{Game}_{0,1}(\lambda)$ and $\mathsf{Game}_{0,2}(\lambda)$ are indistinguishable.

$\mathsf{Game}_{0,2} \rightsquigarrow \mathsf{Game}_{0,3}$. Observe that h and $\mathbf{y}'[i]$ have the same distribution in both games: If ρ is evaluated twice on the same input, then we change nothing. Otherwise, ρ is always evaluated on fresh inputs, which means that its outputs are uniformly random in G. Therefore, $\mathsf{Game}_{0,1}(\lambda)$ and $\mathsf{Game}_{0,2}(\lambda)$ are again indistinguishable.

$\mathsf{Game}_{0,3} \rightsquigarrow \mathsf{Game}_{0,4}$. By definition, $\mathsf{Game}_{0,3}(\lambda)$ and $\mathsf{Game}_{0,4}(\lambda)$ are identical until Bad, which means that $|\Pr[\mathsf{Game}_{0,3}(\lambda)] - \Pr[\mathsf{Game}_{0,4}(\lambda)]| \leq \Pr[\mathsf{Game}_{0,3}(\lambda) \text{ sets } \mathsf{Bad}]$ by the fundamental lemma of game playing. To bound the latter probability, observe that

$$
\begin{aligned}
\Pr[\mathsf{Game}_{0,3}(\lambda) \text{ sets } \mathsf{Bad}] &\leq \Pr[\mathsf{Coll}(\lambda)] = \sum_{k=1}^{q(\lambda)} \Pr[\mathsf{Coll}(\lambda) \mid n = k] \Pr[n = k] \\
&\leq \sum_{k=1}^{q(\lambda)} \sum_{1 \leq i < j \leq k+1} \Pr[\mathsf{Coll}_{i,j}(\lambda) \mid n = k] \Pr[n = k] \leq \sum_{k=1}^{q(\lambda)} \sum_{1 \leq i < j \leq k+1} \Pr\big[\mathrm{LDD}_{\mathsf{H}}^{\mathcal{T},\mathcal{A}_{i,j,k}}(\lambda) \,\big|\, n = k\big] \Pr[n = k] \\
&\leq \sum_{k=1}^{q(\lambda)} \binom{k+1}{2} \Pr\big[\mathrm{LDD}_{\mathsf{H}}^{\mathcal{T},\mathcal{A}_k}(\lambda) \,\big|\, n = k\big] \Pr[n = k] \leq q(\lambda)^2 \sum_{k=1}^{q(\lambda)} \Pr\big[\mathrm{LDD}_{\mathsf{H}}^{\mathcal{T},\mathcal{A}}(\lambda) \,\big|\, n = k\big] \Pr[n = k] \\
&= q(\lambda)^2 \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{A}}(\lambda) \,.
\end{aligned}
$$

Here, games $\mathsf{Coll}(\lambda)$ and $\mathsf{Coll}_{i,j}(\lambda)$ are defined in Figure 25 (top), LDD source $\mathcal{T}$ (specified by $\bar{\mathcal{T}}$) in Figure 25 (center left), and adversary $\mathcal{A}_{i,j,k}$ is given in Figure 25 (center right). We let $\mathcal{A}_k \in \{\mathcal{A}_{i,j,k}\}_{i<j}$ be the adversary with the largest advantage in the LDD game for H with source $\mathcal{T}$, and $\mathcal{A}$ the LDD adversary that reads n off the state passed by $\mathcal{T}$ and then runs $\mathcal{A}_n$.

$\mathsf{Game}_{0,4} \rightsquigarrow \mathsf{Game}_{0,5}$. We observe that both h and $\mathbf{y}[i]$ have the same distribution in both games, because the split UCE source $\mathcal{S}$ makes pairwise different oracle queries. Thus, $\mathsf{Game}_{0,4}(\lambda)$ and $\mathsf{Game}_{0,5}(\lambda)$ are indistinguishable.

$\mathsf{Game}_{0,5} \rightsquigarrow \mathsf{Game}_{0,6}$. Sampling r from $\mathbb{Z}_p^*$ instead of $\mathbb{Z}_p$ amounts in a loss of at most $1/2^{\lambda-1}$ in the winning probability, which means $|\Pr[\mathsf{Game}_{0,4}(\lambda)] - \Pr[\mathsf{Game}_{0,5}(\lambda)]| \leq 1/2^{\lambda-1}$.

For the second term above we therefore have

$$
\Pr\big[\neg\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{GOR}[\Gamma,\mathsf{H}]}(\lambda) \,\big|\, b = 0\big] \geq \Pr\big[\neg\mathrm{PGG}^{\mathcal{S}',\mathcal{D}'}_{\Gamma}(\lambda) \,\big|\, b = 0\big] - q(\lambda)^2 \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{A}}(\lambda) - \frac{q(\lambda)^2 + 1}{2^{\lambda-1}} \,.
$$

Combining the above estimates we obtain Inequality (13) for the UCE advantage.

Algebraic unpredictability. We now show that, for every UCE adversary $(\mathcal{S}, \mathcal{D})$, the PGG source $\mathcal{S}'$ defined via algorithm $\bar{\mathcal{S}}'$ in Figure 23 (bottom) is algebraically unpredictable. To this end, let $(\mathcal{S}, \mathcal{D})$ be a UCE adversary against $\mathsf{GOR}[\Gamma, \mathsf{H}]$ and $\mathcal{P}$ be any predictor in the algebraic unpredictability game against $\mathcal{S}'$. We prove Inequality (14) via a sequence of games. As before, we give here a short description of each game, and present their formal code in Figure 26:

$\mathsf{Game}'_0(\lambda)$ is the algebraic unpredictability game for $\mathcal{S}'$ played by $\mathcal{P}$.

Games $\mathsf{Coll}(\lambda)/\mathsf{Coll}_{i,j}(\lambda)$:

$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $hk \twoheadleftarrow \mathsf{H.KGen}(\pi)$; $(\mathbf{x}, L) \twoheadleftarrow \bar{\mathcal{S}}(\pi)$; $n \leftarrow |\mathbf{x}|$; $\mathbf{x}'[n+1] \leftarrow 1$; for $i = 1$ to n do $\mathbf{x}'[i] \leftarrow \mathsf{H}(hk, \mathbf{x}[i])$

$\mathsf{Coll}(\lambda)$: return $\big((\exists i, j \in [n+1])(\mathbf{x}'[i] = \mathbf{x}'[j])\big)$

$\mathsf{Coll}_{i,j}(\lambda)$: if $(i, j \in [n+1])$ then return $(\mathbf{x}'[i] = \mathbf{x}'[j])$
return 0

Source $\bar{\mathcal{T}}(\pi)$:

$(\mathbf{x}, L) \twoheadleftarrow \bar{\mathcal{S}}(\pi)$; return $(\mathbf{x}, (L, |\mathbf{x}|))$

Adversary $\mathcal{A}_{i,j,k}(\pi, hk, (L, k))$:

$\boldsymbol{\alpha} \leftarrow 0^{k+1}$; $\boldsymbol{\alpha}[i] \leftarrow 1$; $\boldsymbol{\alpha}[j] \leftarrow 1$; return $\boldsymbol{\alpha}$

Adversary $\mathcal{B}(\pi, hk, (L, n))$:

$S \leftarrow \emptyset$; for $i = 1$ to $n+1$ do $\mathbf{y}[i] \twoheadleftarrow \mathsf{G} \setminus S$; $S \leftarrow S \cup \{\mathbf{y}[i]\}$

$(\alpha_1, \ldots, \alpha_{n+1}) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{y}, L, hk)$; $\alpha_0 \leftarrow -\alpha_{n+1}$; return $(\alpha_0, \ldots, \alpha_n)$

Predictor $\mathcal{P}'(\pi, \mathbf{y}, L)$:

$n \leftarrow |\mathbf{y}|$

return $\mathcal{P}(\pi, (L, n))$

Figure 25 — *Top:* Definition of the games Coll and $\mathsf{Coll}_{i,j}$. *Center left:* Definition of the LDD source $\mathcal{T}$. *Center right:* Definition of the LDD adversary $\mathcal{A}_{i,j,k}$. Here, the vector $\boldsymbol{\alpha}$ is indexed starting from 0 for convenience. *Bottom left:* Definition of the LDD adversary $\mathcal{B}$. *Bottom right:* Reduction from an LDD predictor $\mathcal{P}$ to a UCE predictor $\mathcal{P}'$.

$\mathsf{Game}'_1(\lambda)$ is the same as $\mathsf{Game}'_0(\lambda)$, but we immediately return true if two hashes of distinct points queried by $\bar{\mathcal{S}}$ coincide, or if any of these hashes is equal to 1. This implies that, later in the game, the encoding σ is queried on pairwise different inputs.

$\mathsf{Game}'_2(\lambda)$ is the same as $\mathsf{Game}'_1(\lambda)$, but we implement σ via lazy sampling. In other words, we no longer compute $\mathbf{y}[i]$ as evaluations of σ, but instead return random, pairwise different elements from G. Note that there is no need to check for consistency, because of the additional check introduced in $\mathsf{Game}'_1(\lambda)$.

$\mathsf{Game}'_3(\lambda)$ is the same as $\mathsf{Game}'_2(\lambda)$, but we remove the if-statement added in $\mathsf{Game}'_1(\lambda)$. We also cancel r' from the winning condition (this can be done safely, because $r' \neq 0$ by assumption). Finally, we don't sample r' anymore, since it no longer appears anywhere in the game.

As before, we now relate the success probabilities of subsequent games.

$\mathsf{Game}'_0 \rightsquigarrow \mathsf{Game}'_1$. Observe that, by definition, $\mathsf{Game}'_1(\lambda)$ returns true whenever $\mathsf{Game}'_0(\lambda)$ does. Thus, clearly, $\Pr[\mathsf{Game}'_0(\lambda)] \leq \Pr[\mathsf{Game}'_1(\lambda)]$.

$\mathsf{Game}'_1 \rightsquigarrow \mathsf{Game}'_2$. We now argue that $\Pr[\mathsf{Game}'_1(\lambda)] = \Pr[\mathsf{Game}'_2(\lambda)]$. To do so, observe that the distributions of $\mathbf{y}[i]$ are identical in both games. Indeed, collisions between the entries $\mathbf{x}'[i]$ are ruled out in both games. This means that, in both cases, the $\mathbf{y}[i]$ are pairwise different random elements of G: In $\mathsf{Game}'_1(\lambda)$ this holds because σ is a random injection, and in $\mathsf{Game}'_2(\lambda)$ this is true by definition.

$\mathsf{Game}'_2 \rightsquigarrow \mathsf{Game}'_3$. As in the first part of the proof, $\mathsf{Game}'_2(\lambda)$ and $\mathsf{Game}'_3(\lambda)$ are identical until Bad, which by the fundamental lemma of game playing means $|\Pr[\mathsf{Game}'_2(\lambda)] - \Pr[\mathsf{Game}'_3(\lambda)]| \leq \Pr[\mathsf{Game}'_2(\lambda) \text{ sets } \mathsf{Bad}]$, and by the same reasoning as in the game hop $\mathsf{Game}_{0,3} \rightsquigarrow \mathsf{Game}_{0,4}$ above, we have

$$\Pr[\mathsf{Game}_2(\lambda) \text{ sets } \mathsf{Bad}] \leq \Pr[\mathsf{Coll}(\lambda)] \leq q(\lambda)^2 \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{A}}(\lambda)\,.$$

Game'_3. By direct inspection, we have $\mathsf{Game}'_3(\lambda) = \mathrm{LDD}^{\mathcal{T},\mathcal{B}}_{\mathsf{H}}(\lambda)$, where $\mathcal{B}$ is the LDD adversary defined in Figure 25 (bottom left). Therefore, $\Pr[\mathsf{Game}'_3(\lambda)] = \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{T},\mathcal{B}}(\lambda)$.

Collecting all our estimates above, we obtain the claimed bound (14).

Unpredictability of $\mathcal{T}$. Observe that our estimates above are only meaningful if $\mathcal{T}$ is a statistically unpredictable LDD source. To prove that this indeed is the case, let $\mathcal{P}$ be any predictor in the LDD unpredictability game for $(\mathsf{H}, \mathcal{T})$, and consider the predictor $\mathcal{P}'$ in the UCE unpredictability game for $(\mathsf{GOR}[\Gamma, \mathsf{H}], \mathcal{S})$ defined in Figure 25 (bottom right). By direct inspection, $\mathcal{P}'$ perfectly simulates the game played by $\mathcal{P}$, which means that $\mathrm{Adv}^{\mathrm{pred}}_{\mathsf{H},\mathcal{T},\mathcal{P}}(\lambda) \leq \mathrm{Adv}^{\mathrm{pred}}_{\mathsf{GOR}[\Gamma,\mathsf{H}],\mathcal{S},\mathcal{P}'}(\lambda)$. □

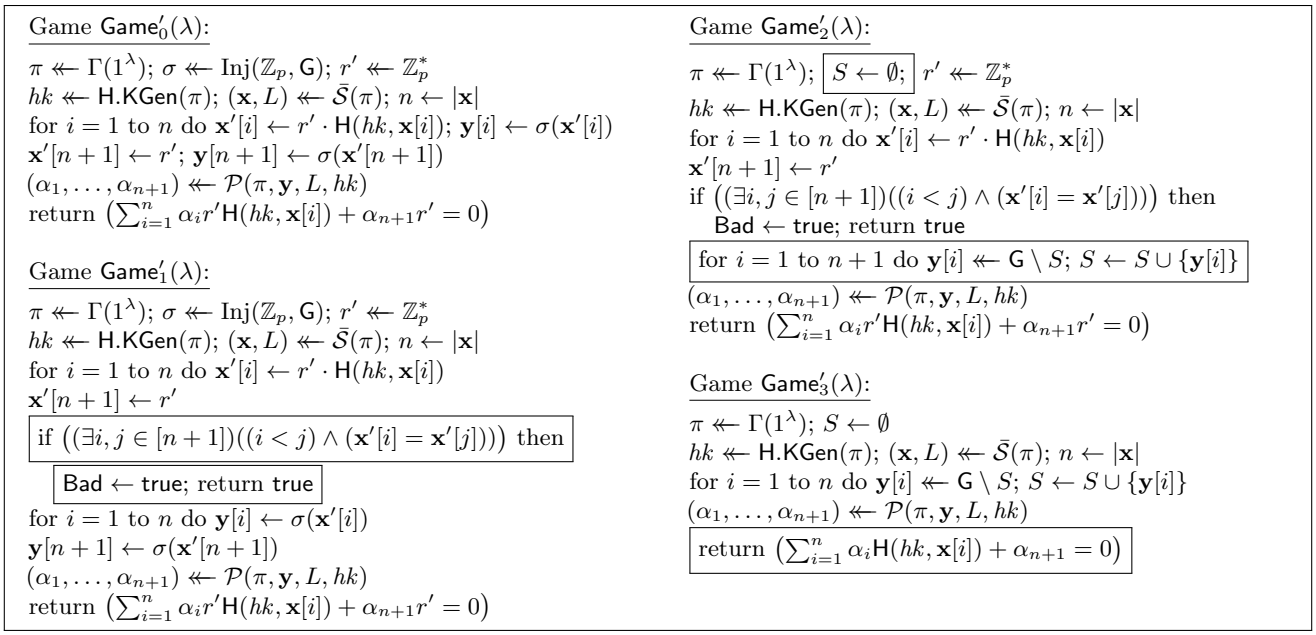

Figure 26 — Code of the intermediate games in the proof of the algebraic unpredictability bound (14) in Theorem 6.2.

Remark. We note that the above proof can be easily extended to *multi-key* UCEs [BHK13, Figure 8] for split sources by generating multiple hash keys and hash public keys via re-randomization. BHK conjectured that UCE and multi-key UCE are in general equivalent, which remains open.

An alternative construction of UCEs from PGGs would first compute g^{rx} and then chop half of the output bits so that group operations on hash outputs are no longer possible. (This was previously suggested, for example, as a way to build a RO in the GGM by Zhandry and Zhang [ZZ21].) An analysis of this construction may be made possible in the PGG framework by defining new sources that permit different forms of post-processing.

6.3 KDM-CPA Security of Modified ElGamal

Security against key-dependent plaintext attacks (KDM-CPA) is a notion of security for encryption schemes where an adversary can obtain encryptions of messages which are computed as a function of the secret key.[17] It was introduced by Black, Rogaway, and Shrimpton [BRS03] in the symmetric setting and by Camenisch and Lysyanskaya [CL01] in the public-key setting. In a breakthrough work, Boneh et al. [BHHO08] constructed KDM-secure public-key encryption for affine functions in the standard model. KDM security is also important in other contexts; for example, KDM security for IBEs from LWE, and completeness of circular security for it, have been studied in recent works [App11, GGH20, LNPT20, KM20]. Here, we focus on a non-adaptive version of this notion, where the key-dependent plaintexts can depend on the secret key and public parameters, but not on the public key or on other ciphertexts.[18]

KDM-CPA security. A PKE scheme is a tuple of PPT algorithms $\mathsf{E} := (\mathsf{E.Setup}, \mathsf{E.KGen}, \mathsf{E.Enc}, \mathsf{E.Dec})$, where $\mathsf{E.Setup}(1^\lambda)$ outputs system-wide parameters π available to all algorithms, and $\mathsf{E.KGen}$, $\mathsf{E.Enc}$, and $\mathsf{E.Dec}$ are the usual key-generation, encryption, and decryption algorithms.

[17] In the case the messages *are* the secret keys, the notion is also called *circular security*.

[18] Note that public kesy may be derived from secret keys in a randomized way. The scheme that we study here has this property.

Game KDM-CPA$^{\mathcal{A}}_{\mathsf{E}}(\lambda)$:
$d \twoheadleftarrow \{0,1\}$
$\pi \twoheadleftarrow \mathsf{E.Setup}(1^\lambda)$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do
  $(\mathbf{sk}[i], \mathbf{pk}[i]) \twoheadleftarrow \mathsf{E.KGen}(\pi)$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$
for $i = 1$ to $|\mathbf{m}_d|$ do
  $\mathbf{c}[i] \twoheadleftarrow \mathsf{E.Enc}(\pi, \mathbf{pk}[\mathbf{p}[i]], \mathbf{m}_d[i])$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}, \mathbf{c})$; return $(d = d')$

$\mathsf{MEI}[\Gamma].\mathsf{Setup}(1^\lambda)$:
$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
return π

$\mathsf{MEI}[\Gamma].\mathsf{KGen}(\pi)$:
$(\circ, g, p) \leftarrow \pi$; $s, sk \twoheadleftarrow \mathbb{Z}_p$
$pk_1 \leftarrow g^s$; $pk_2 \leftarrow g^{s \cdot sk}$
return $(sk, (pk_1, pk_2))$

$\mathsf{MEI}[\Gamma].\mathsf{Enc}(\pi, pk, m)$:
$(\circ, g, p) \leftarrow \pi$; $(pk_1, pk_2) \leftarrow pk$
$t \twoheadleftarrow \mathbb{Z}_p$; $c_1 \leftarrow pk_1^t$
$c_2' \leftarrow pk_2^t$; $c_2 \leftarrow m \circ c_2'$
return (c_1, c_2)

$\mathsf{MEI}[\Gamma].\mathsf{Dec}(\pi, sk, c)$:
$(c_1, c_2) \leftarrow c$; return $c_2 \circ c_1^{-sk}$

Auxiliary masking source $\bar{\mathcal{S}}(\pi)$:
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; for $i = 1$ to n do $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$; for $i = 1$ to q do $\mathbf{t}[i] \twoheadleftarrow \mathbb{Z}_p$
$\mathbf{x} \leftarrow (\mathbf{s}[1], \ldots, \mathbf{s}[n], \mathbf{s}[1]\mathbf{sk}[1], \ldots, \mathbf{s}[n]\mathbf{sk}[n], \mathbf{s}[\mathbf{p}[1]]\mathbf{t}[1], \ldots,$
  $\mathbf{s}[\mathbf{p}[q]]\mathbf{t}[q], \mathbf{s}[\mathbf{p}[1]]\mathbf{sk}[\mathbf{p}[1]]\mathbf{t}[1], \ldots, \mathbf{s}[\mathbf{p}[q]]\mathbf{sk}[\mathbf{p}[q]]\mathbf{t}[q])$
$\mathbf{m} \leftarrow (1_\mathsf{G}, \ldots, 1_\mathsf{G}, \mathbf{m}_d[1], \ldots, \mathbf{m}_d[q])$; $L \leftarrow (d)$; return $(\mathbf{x}, \mathbf{m}, L)$

PGG distinguisher $\mathcal{D}(\pi, (\mathbf{y}, L))$:
$(\mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2) \leftarrow \mathbf{y}$; $(d) \leftarrow L$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$
return $(d = d')$

Figure 27 — *Top left:* Our KDM-CPA model for a public-key encryption scheme E. *Top right:* The modified ElGamal encryption scheme $\mathsf{MEI}[\Gamma]$, instantiated with a computational group scheme Γ. *Bottom:* Reduction from a KDM-CPA adversary $\mathcal{A}$ to a PGG adversary $(\mathcal{S}, \mathcal{D})$.

Let E be a PKE scheme. The advantage of an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the KDM-CPA game for E is defined as

$$\mathrm{Adv}^{\text{kdm-cpa}}_{\mathsf{E},\mathcal{A}}(\lambda) := 2 \cdot \Pr\big[\text{KDM-CPA}^{\mathcal{A}}_{\mathsf{E}}(\lambda)\big] - 1\,,$$

where the KDM-CPA game is defined in Figure 27 (top left). Here, $\mathcal{A}_0$ and $\mathcal{A}_1$ can be unbounded with polynomially bounded output, and $\mathcal{A}_2$ is PPT. We require that $\mathcal{A}_1$ outputs two message vectors of equal length, which component-wise contain messages of equal lengths. We say that E is KDM-CPA secure, if the advantage of any $\mathcal{A}$ as above in the KDM-CPA game for E is negligible.

Construction. Our construction of a PKE scheme $\mathsf{MEI}[\Gamma]$ secure in the KDM model discussed above is presented in Figure 27 (top right). The scheme is a minor modification of the classical ElGamal encryption scheme: The only difference to the traditional scheme is that the generator returned by the computational group scheme is re-randomized during key generation.

We now prove that $\mathsf{MEI}[\Gamma]$ is KDM-CPA secure under PGG.

Theorem 6.3 (PGG $\Longrightarrow$ Modified ElGamal is KDM-CPA)**.** *Let Γ be a computational group scheme, and let $\mathsf{MEI}[\Gamma]$ be the modified ElGamal encryption scheme defined in Figure 27 (top right). If Γ is* PGG$[\mathbf{S}^{\text{alg}} \cap \mathbf{S}^{\text{msk}}]$ *secure, then $\mathsf{MEI}[\Gamma]$ is* KDM-CPA *secure. More precisely, for any adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the KDM-CPA game for $\mathsf{MEI}[\Gamma]$ there is an adversary $(\mathcal{S}, \mathcal{D})$ in the PGG game for Γ such that*

$$\mathrm{Adv}^{\text{kdm-cpa}}_{\mathsf{MEI}[\Gamma],\mathcal{A}}(\lambda) \le 2 \cdot \mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + \frac{(4n(\lambda)-1)(n(\lambda)+q(\lambda)) + n(\lambda) + 3q(\lambda)^2}{2^{\lambda-2}}\,. \tag{16}$$

Furthermore, $\mathcal{S} \in \mathbf{S}^{\text{alg}} \cap \mathbf{S}^{\text{msk}}$. More precisely, for any predictor $\mathcal{P}$ in the AlgPred game for $(\Gamma, \mathcal{S})$, we have

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \le \frac{6n(\lambda)^2 + 2n(\lambda)q(\lambda) + 4q(\lambda)^2 + n(\lambda)}{2^{\lambda-1}}\,. \tag{17}$$

Here, $n(\lambda)$ and $q(\lambda)$ are upper bounds on the number of key pairs requested by $\mathcal{A}_0$ and on the length of the message vectors returned by $\mathcal{A}_1$, respectively.

Proof Overview. For an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the KDM-CPA game for $\mathsf{MEI}[\Gamma]$, consider the PGG adversary $(\mathcal{S}, \mathcal{D})$ defined in Figure 27 (bottom). Notice that, by construction, $\mathcal{S}$ is a masking source.

By direct inspection, $\Pr\big[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\big|\, b = 1\big] = \Pr\big[\text{KDM-CPA}^{\mathcal{A}}_{\mathsf{MEI}[\Gamma]}(\lambda)\big]$, where b denotes the challenge bit in the PGG game. Indeed, the only difference between the two games is that randomness $\mathbf{s}[i] \in \mathbb{Z}_p$ in the KDM-CPA game corresponds to $r\mathbf{s}[i]$ in the PGG game, where r is the random exponent from the PGG game contained in the real Exp oracle. Since $r \neq 0$, these quantities are pairwise identically distributed, which implies that the winning probabilities coincide. On the other hand, $\Pr\big[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\big|\, b = 0\big]$ is negligible. This follows from a bad-event analysis: We transition to a game where we replace the Exp oracle with a forgetful random oracle, so that $\mathcal{A}_2$ has no advantage because its input is independent of d. The two games are identical unless there is a repeat query to the oracle, or there is a repetition in the outputs of the forgetful random oracle, with both events being unlikely.

Finally, to establish algebraic unpredictability of $\mathcal{S}$, consider any predictor $\mathcal{P}$ that returns a linear combination of the queries with coefficients $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q, \delta_1, \ldots, \delta_q$ given leakage $(\mathbf{y}, d)$ computed using the ideal Exp oracle. We again transition to a game where Exp is replaced with a forgetful random oracle. Unfortunately, the leakage is not yet independent of $\mathbf{sk}$, because $q = |\mathbf{c}_1| = |\mathbf{c}_2|$ depends on $\mathbf{sk}$. So we transition to a game where $\mathbf{sk}$ is picked after running $\mathcal{P}$, from the unifrom distribution on $\mathbb{Z}_p^n$ given leakage q. Notice that this distribution loses at most logarithmically many bits of min-entropy compared to the original one, so has still high min-entropy. Winning this game then means that $\mathbf{s}$, $\mathbf{sk}$ and $\mathbf{t}$ (which the predictor now knows nothing about, other than that $\mathbf{sk}$ has a slightly skewed distribution) must be a root of the non-zero polynomial

$$\sum_{i=1}^{n} X_i(\alpha_i + \beta_i Y_i) + \sum_{j=1}^{q} X_{\mathbf{p}[j]} Z_j(\gamma_j + \delta_j Y_{\mathbf{p}[j]})\,.$$

By the Schwartz–Zippel lemma, the probability that this happens (for uniform $\mathbf{s}$ and $\mathbf{t}$ and high entropy $\mathbf{sk}$) is negligible. ■

Proof. Given an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the KDM-CPA game for $\mathsf{MEI}[\Gamma]$, define the masking source $\mathcal{S}$ via auxiliary algorithm $\bar{\mathcal{S}}$ and distinguisher $\mathcal{D}$ as shown in Figure 27 (bottom). Observe that our reduction does not directly convert the KDM-CPA game into the PGG experiment, since public keys and encryptions are always computed using the real group operation in the former game, whereas they can involve the generic operation in the latter. Therefore, the source must simulate the KDM-CPA game in such a way that both cases $d \in \{0, 1\}$ are covered when it plays in the real world, while gaining almost no advantage in the ideal world.

Masking structure. By construction, it is clear that $\bar{\mathcal{S}}$ makes no Exp query and returns vectors $\mathbf{x}$ and $\mathbf{m}$ of equal length. Thus, $\mathcal{S} \in \mathbf{S}^{\mathrm{msk}}$.

Advantage bound. To prove Inequality (16), first recall that

$$\mathrm{Adv}^{\text{kdm-cpa}}_{\mathsf{MEI}[\Gamma],\mathcal{A}}(\lambda) = 2 \cdot \Pr\big[\text{KDM-CPA}^{\mathcal{A}}_{\mathsf{MEI}[\Gamma]}(\lambda)\big] - 1\,.$$

We study the term on the right. Let $\mathsf{Game}(\lambda)$ denote the PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$ with $b = 1$ fixed, as shown in Figure 28 (left). This game is virtually identical to the KDM-CPA game for $\mathsf{MEI}[\Gamma]$ played by $\mathcal{A}$, depicted in Figure 28 (right): The only major difference between the two experiments is the generator used to compute the public keys and the ciphertexts. Indeed, $\mathsf{Game}(\lambda)$ computes $\mathbf{pk}_1$, $\mathbf{pk}_2$, $\mathbf{c}_1$ and $\mathbf{c}_2$ as

$$\mathbf{pk}_1[i] := g^{\mathbf{s}[i]} = g_0^{r\mathbf{s}[i]}\,, \qquad \mathbf{c}_1[i] := g^{\mathbf{s}[\mathbf{p}[i]]\mathbf{t}[i]} = g_0^{(r\mathbf{s}[\mathbf{p}[i]])\mathbf{t}[i]}\,,$$
$$\mathbf{pk}_2[i] := g^{\mathbf{s}[i]\mathbf{sk}[i]} = g_0^{(r\mathbf{s}[i])\mathbf{sk}[i]}\,, \qquad \mathbf{c}_2[i] := \mathbf{m}_d[i] \circ g^{\mathbf{s}[\mathbf{p}[i]]\mathbf{sk}[\mathbf{p}[i]]\mathbf{t}[i]} = \mathbf{m}_d[i] \circ g_0^{(r\mathbf{s}[\mathbf{p}[i]])\mathbf{sk}[\mathbf{p}[i]]\mathbf{t}[i]}\,,$$

Game $\mathsf{Game}(\lambda)$:

$\pi \twoheadleftarrow \Gamma(1^\lambda)$; $r \twoheadleftarrow \mathbb{Z}_p^*$; $g \leftarrow g_0^r$; $d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $i = 1$ to q do $\mathbf{t}[i] \twoheadleftarrow \mathbb{Z}_p$
for $i = 1$ to n do $\mathbf{pk}_1[i] \leftarrow g^{\mathbf{s}[i]}$; $\mathbf{pk}_2[i] \leftarrow g^{\mathbf{s}[i]\mathbf{sk}[i]}$
for $i = 1$ to q do
  $\mathbf{c}_1[i] \leftarrow g^{\mathbf{s}[\mathbf{p}[i]]\mathbf{t}[i]}$; $\mathbf{c}_2[i] \leftarrow \mathbf{m}_d[i] \circ g^{\mathbf{s}[\mathbf{p}[i]]\mathbf{sk}[\mathbf{p}[i]]\mathbf{t}[i]}$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Game KDM-CPA$^{\mathcal{A}}_{\mathsf{MEI}[\Gamma]}(\lambda)$:

$d \twoheadleftarrow \{0,1\}$; $\pi \twoheadleftarrow \Gamma(1^\lambda)$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{pk}_1[i] \leftarrow g_0^{\mathbf{s}[i]}$; $\mathbf{pk}_2[i] \leftarrow g_0^{\mathbf{s}[i]\mathbf{sk}[i]}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $i = 1$ to q do
  $\mathbf{t}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{c}_1[i] \leftarrow g_0^{\mathbf{s}[\mathbf{p}[i]]\mathbf{t}[i]}$; $\mathbf{c}_2[i] \leftarrow \mathbf{m}_d[i] \circ g_0^{\mathbf{s}[\mathbf{p}[i]]\mathbf{sk}[\mathbf{p}[i]]\mathbf{t}[i]}$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Figure 28 — *Left:* The PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$ with $b = 1$ fixed. *Right:* The KDM-CPA game for $\mathsf{MEI}[\Gamma]$ played by $\mathcal{A}$.

whereas in KDM-CPA$^{\mathcal{A}}_{\mathsf{MEI}[\Gamma]}(\lambda)$ we have

$$\mathbf{pk}_1[i] \coloneqq g_0^{\mathbf{s}[i]}\,, \qquad \mathbf{pk}_2[i] \coloneqq g_0^{\mathbf{s}[i]\mathbf{sk}[i]}\,, \qquad \mathbf{c}_1[i] \coloneqq g_0^{\mathbf{s}[\mathbf{p}[i]]\mathbf{t}[i]}\,, \qquad \mathbf{c}_2[i] \coloneqq \mathbf{m}_d[i] \circ g_0^{\mathbf{s}[\mathbf{p}[i]]\mathbf{sk}[\mathbf{p}[i]]\mathbf{t}[i]}\,.$$

But observe that, in both cases, the values $\mathbf{s}[i]$ and $r\mathbf{s}[i]$ in the exponents are uniformly random in $\mathbb{Z}_p^*$, since multiplying by an invertible element gives a permutation. Thus, the public keys and the ciphertexts have the same distribution in the two experiments, which means the two games are equivalent. Therefore,

$$\begin{aligned}\mathrm{Adv}^{\text{kdm-cpa}}_{\mathsf{MEI}[\Gamma],\mathcal{A}}(\lambda) &= 2 \cdot \Pr\big[\text{KDM-CPA}^{\mathcal{A}}_{\mathsf{MEI}[\Gamma]}(\lambda)\big] - 1 = 2 \cdot \Pr\big[\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\big|\, b = 1\big] - 1 \\ &= 2 \cdot \mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + 2 \cdot \Pr\big[\neg\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\big|\, b = 0\big] - 1\,.\end{aligned}$$

We now study the second term in the sum above. To do so, we consider a sequence of games. We give here a short description of each game, and present their formal code in Figure 29.

$\mathsf{Game}_0(\lambda)$ is the PGG game for Γ played by $(\mathcal{S}, \mathcal{D})$, with bit $b = 0$ fixed and inverted winning condition. We also set a flag $\mathsf{Bad}_0 \leftarrow \mathsf{true}$ in case any of the sampled $\mathbf{s}[i]$, $\mathbf{sk}[i]$, or $\mathbf{t}[j]$ turns out to be 0.

$\mathsf{Game}_1(\lambda)$ is the same as $\mathsf{Game}_0(\lambda)$, but we use resampling to make sure that $\mathbf{s}[i], \mathbf{sk}[i], \mathbf{t}[j] \neq 0$ for all i and j. We also record all inputs to σ in a list M, and set a flag $\mathsf{Bad}_1 \leftarrow \mathsf{true}$ if σ is evaluated on repeated inputs.

$\mathsf{Game}_2(\lambda)$ is the same as $\mathsf{Game}_1(\lambda)$, but we ensure that all inputs to σ are pairwise different. We implement this by resampling $\mathbf{s}[i]$, $\mathbf{sk}[i]$, or $\mathbf{t}[j]$ if necessary.

$\mathsf{Game}_3(\lambda)$ is the same as $\mathsf{Game}_2(\lambda)$, but we lazily sample the encoding σ. In other words, the masking terms in the ciphertext are no longer obtained via evaluations of σ, but are random, pairwise distinct elements of G. This can be achieved by recording all sampled elements in a new list M'. If a collision occurs, we set $\mathsf{Bad}_2 \leftarrow \mathsf{true}$ and resample the corresponding element.

$\mathsf{Game}_4(\lambda)$ is the same as $\mathsf{Game}_3(\lambda)$, but we sample $\mathbf{pk}_1[i]$, $\mathbf{pk}_2[i]$, $\mathbf{c}_1[j]$, and $\mathbf{c}'_2[j]$ (the latter being the term masking the message $\mathbf{m}_d[j]$) at random without any further checks.

$\mathsf{Game}_5(\lambda)$ is the same as $\mathsf{Game}_4(\lambda)$, but we directly sample $\mathbf{c}_2[j]$ at random, instead of $\mathbf{c}'_2[j]$ and multiplying it with $\mathbf{m}_d[j]$.

We now study the difference between the success probabilities of subsequent games.

$\mathsf{Game}_0 \rightsquigarrow \mathsf{Game}_1$. By definition, these two games are identical until Bad_0, and by the fundamental lemma of game playing we have $|\Pr[\mathsf{Game}_0(\lambda)] - \Pr[\mathsf{Game}_1(\lambda)]| \leq \Pr[\mathsf{Game}_0 \text{ sets } \mathsf{Bad}_0] \leq (2n(\lambda) + q(\lambda))/p$. The last inequality follows from the observation that Bad_0 is set to true in Game_0 if at least one of the randomly chosen $\mathbf{s}[i]$, $\mathbf{sk}[i]$, or $\mathbf{t}[j]$ (of which there are at most $n(\lambda)$ and $q(\lambda)$, respectively) turns out to be zero, which happens with probability at most $(2n(\lambda) + q(\lambda))/p \leq (2n(\lambda) + q(\lambda))/2^{\lambda-1}$.

Game $\mathsf{Game}_0(\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$; if $(\mathbf{s}[i] = 0)$ then $\mathsf{Bad}_0 \leftarrow \mathsf{true}$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$; if $(\mathbf{sk}[i] = 0)$ then $\mathsf{Bad}_0 \leftarrow \mathsf{true}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do
  $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p$; if $(\mathbf{t}[j] = 0)$ then $\mathsf{Bad}_0 \leftarrow \mathsf{true}$
for $i = 1$ to n do
  $\mathbf{pk}_1[i] \leftarrow \sigma(\mathbf{s}[i])$; $\mathbf{pk}_2[i] \leftarrow \sigma(\mathbf{s}[i]\mathbf{sk}[i])$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \leftarrow \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Game $\mathsf{Game}_1(\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $\boxed{M \leftarrow [\,]}$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$
  if $(\mathbf{s}[i] = 0)$ then $\mathsf{Bad}_0 \leftarrow \mathsf{true}$; $\boxed{\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^*}$
  $\boxed{\text{if } (\mathbf{s}[i] \in M) \text{ then } \mathsf{Bad}_1 \leftarrow \mathsf{true}}$
  $\boxed{M \leftarrow M : \mathbf{s}[i]}$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$
  if $(\mathbf{sk}[i] = 0)$ then $\mathsf{Bad}_0 \leftarrow \mathsf{true}$; $\boxed{\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^*}$
  $\boxed{\text{if } (\mathbf{s}[i]\mathbf{sk}[i] \in M) \text{ then } \mathsf{Bad}_1 \leftarrow \mathsf{true}}$
  $\boxed{M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do
  $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p$
  if $(\mathbf{t}[j] = 0)$ then $\mathsf{Bad}_0 \leftarrow \mathsf{true}$; $\boxed{\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^*}$
  $\boxed{\text{if } (\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j] \in M) \vee (\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j] \in M) \text{ then}}$
    $\boxed{\mathsf{Bad}_1 \leftarrow \mathsf{true}}$
  $\boxed{M \leftarrow M : \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j];\ M \leftarrow M : \mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j]}$
for $i = 1$ to n do $\mathbf{pk}_1[i] \leftarrow \sigma(\mathbf{s}[i])$; $\mathbf{pk}_2[i] \leftarrow \sigma(\mathbf{s}[i]\mathbf{sk}[i])$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \leftarrow \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Game $\mathsf{Game}_2(\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $M \leftarrow [\,]$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^*$; if $(\mathbf{s}[i] \in M)$ then $\mathsf{Bad}_1 \leftarrow \mathsf{true}$; $\boxed{\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M}$
  $M \leftarrow M : \mathbf{s}[i]$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^*$
  if $(\mathbf{s}[i]\mathbf{sk}[i] \in M)$ then
    $\mathsf{Bad}_1 \leftarrow \mathsf{true}$; $\boxed{\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M}$
  $M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do
  $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^*$
  if $(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j] \in M) \vee (\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j] \in M)$ then
    $\mathsf{Bad}_1 \leftarrow \mathsf{true}$
    $\boxed{\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^* \setminus (\mathbf{s}[\mathbf{p}[j]]^{-1} \cdot M \cup (\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]])^{-1} \cdot M)}$
  $M \leftarrow M : \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j], \mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j]$
for $i = 1$ to n do $\mathbf{pk}_1[i] \leftarrow \sigma(\mathbf{s}[i])$; $\mathbf{pk}_2[i] \leftarrow \sigma(\mathbf{s}[i]\mathbf{sk}[i])$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \leftarrow \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$; $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Game $\mathsf{Game}_3(\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $M \leftarrow [\,]$; $\boxed{M' \leftarrow [\,]}$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M$; $M \leftarrow M : \mathbf{s}[i]$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M$; $M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $i = 1$ to n do
  $\boxed{\mathbf{pk}_1[i] \twoheadleftarrow \mathsf{G};\ \mathbf{pk}_2[i] \twoheadleftarrow \mathsf{G}}$
  $\boxed{\text{if } (\mathbf{pk}_1[i] \in M') \text{ then } \mathsf{Bad}_2 \leftarrow \mathsf{true};\ \mathbf{pk}_1[i] \twoheadleftarrow \mathsf{G} \setminus M'}$
  $\boxed{M' \leftarrow M' : \mathbf{pk}_1[i]}$
  $\boxed{\text{if } (\mathbf{pk}_2[i] \in M') \text{ then } \mathsf{Bad}_2 \leftarrow \mathsf{true};\ \mathbf{pk}_2[i] \twoheadleftarrow \mathsf{G} \setminus M'}$
  $\boxed{M' \leftarrow M' : \mathbf{pk}_2[i]}$
for $j = 1$ to q do
  $\boxed{\mathbf{c}_1[j] \twoheadleftarrow \mathsf{G};\ \mathbf{c}'_2[j] \twoheadleftarrow \mathsf{G}}$
  $\boxed{\text{if } (\mathbf{c}_1[j] \in M') \text{ then } \mathsf{Bad}_2 \leftarrow \mathsf{true};\ \mathbf{c}_1[j] \twoheadleftarrow \mathsf{G} \setminus M'}$
  $\boxed{M' \leftarrow M' : \mathbf{c}_1[j]}$
  $\boxed{\text{if } (\mathbf{c}'_2[j] \in M') \text{ then } \mathsf{Bad}_2 \leftarrow \mathsf{true};\ \mathbf{c}'_2[j] \twoheadleftarrow \mathsf{G} \setminus M'}$
  $\boxed{M' \leftarrow M' : \mathbf{c}'_2[j]}$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \mathbf{c}'_2[j]$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Game $\mathsf{Game}_4(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $M \leftarrow []$; $M' \leftarrow []$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M$; $M \leftarrow M : \mathbf{s}[i]$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M$; $M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $i = 1$ to n do
  $\mathbf{pk}_1[i] \twoheadleftarrow \mathsf{G}$; if $(\mathbf{pk}_1[i] \in M')$ then $\mathsf{Bad}_2 \leftarrow \mathsf{true}$
  $M' \leftarrow M' : \mathbf{pk}_1[i]$
  $\mathbf{pk}_2[i] \twoheadleftarrow \mathsf{G}$; if $(\mathbf{pk}_2[i] \in M')$ then $\mathsf{Bad}_2 \leftarrow \mathsf{true}$
  $M' \leftarrow M' : \mathbf{pk}_2[i]$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \twoheadleftarrow \mathsf{G}$; if $(\mathbf{c}_1[j] \in M')$ then $\mathsf{Bad}_2 \leftarrow \mathsf{true}$
  $M' \leftarrow M' : \mathbf{c}_1[j]$
  $\mathbf{c}'_2[j] \twoheadleftarrow \mathsf{G}$; if $(\mathbf{c}'_2[j] \in M')$ then $\mathsf{Bad}_2 \leftarrow \mathsf{true}$
  $M' \leftarrow M' : \mathbf{c}'_2[j]$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \mathbf{c}'_2[j]$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Game $\mathsf{Game}_5(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $M \leftarrow []$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M$; $M \leftarrow M : \mathbf{s}[i]$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M$; $M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $i = 1$ to n do
  $\mathbf{pk}_1[i] \twoheadleftarrow \mathsf{G}$; $\mathbf{pk}_2[i] \twoheadleftarrow \mathsf{G}$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \twoheadleftarrow \mathsf{G}$; $\boxed{\mathbf{c}_2[j] \twoheadleftarrow \mathsf{G}}$
$d' \twoheadleftarrow \mathcal{A}_2(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2)$; return $(d = d')$

Figure 29 — Code of the intermediate games in the proof of the KDM-CPA advantage bound (16) in Theorem 6.3.

$\mathsf{Game}_1 \rightsquigarrow \mathsf{Game}_2$. Notice that, by definition, the two games are identical until Bad_1. Furthermore, this flag is only triggered if $\mathbf{s}[i]$ (resp., $\mathbf{s}[i]\mathbf{sk}[i]$ or any of $\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[i]$ and $\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[i]$, all of which are uniformly random in $\mathbb{Z}_p^*$ because non-zero by the first game hop) is already contained in M. Therefore,

$$\left|\Pr[\mathsf{Game}_1(\lambda)] - \Pr[\mathsf{Game}_2(\lambda)]\right| \leq \Pr[\mathsf{Game}_1 \text{ sets } \mathsf{Bad}_1] \leq \sum_{i=0}^{n(\lambda)-1} \frac{i}{p} + \sum_{i=0}^{n(\lambda)-1} \frac{n(\lambda)+i}{p} + \sum_{i=0}^{q(\lambda)-1} \frac{2n(\lambda)+2i}{p}$$
$$\leq \frac{(2n(\lambda)-1)(n(\lambda)+q(\lambda)) + q(\lambda)^2}{2^{\lambda-1}}\,.$$

$\mathsf{Game}_2 \rightsquigarrow \mathsf{Game}_3$. We now argue that the next two games are indistinguishable. To do so, notice that the distributions of $\sigma(\mathbf{s}[i])$ and $\mathbf{pk}_1[i]$, those of $\sigma(\mathbf{s}[i]\mathbf{sk}[i])$ and $\mathbf{pk}_2[i]$, those of $\sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$ and $\mathbf{c}_1[j]$, as well as those of $\sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$ and $\mathbf{c}'_2[i]$, are identical in both games. Indeed, in Game_2 the inputs to σ are pairwise distinct, which means that the outputs are pairwise different random elements from G. This is exactly the distribution of the corresponding elements in $\mathsf{Game}_3(\lambda)$. Therefore, we have $\Pr[\mathsf{Game}_2(\lambda)] = \Pr[\mathsf{Game}_3(\lambda)]$.

$\mathsf{Game}_3 \rightsquigarrow \mathsf{Game}_4$. Again notice that, by definition, the two games are identical until Bad_2. This flag is only triggered if $\mathbf{pk}_1[i]$, $\mathbf{pk}_2[i]$, $\mathbf{c}_1[i]$, or $\mathbf{c}'_2[i]$ are already contained in M'. Thus,

$$\left|\Pr[\mathsf{Game}_3(\lambda)] - \Pr[\mathsf{Game}_4(\lambda)]\right| \leq \Pr[\mathsf{Game}_3 \text{ sets } \mathsf{Bad}_2] \leq \sum_{i=0}^{n(\lambda)-1} \frac{2i+2i+1}{p} + \sum_{i=0}^{q(\lambda)-1} \frac{2n(\lambda)+2i+2i+1}{p}$$
$$\leq \frac{2n(\lambda)^2 - n(\lambda)}{p} + \frac{q(\lambda)(2n(\lambda)+2q(\lambda)-1)}{p} \leq \frac{(2n(\lambda)-1)(n(\lambda)+q(\lambda)) + 2q(\lambda)^2}{2^{\lambda-1}}\,.$$

$\mathsf{Game}_4 \rightsquigarrow \mathsf{Game}_5$. Notice that $\Pr[\mathsf{Game}_4(\lambda)] = \Pr[\mathsf{Game}_5(\lambda)]$ because $\mathbf{c}'_2[j]$ is uniformly random in $\mathsf{Game}_4(\lambda)$, and therefore so is $\mathbf{c}_2[j]$. This means that $\mathbf{c}_2[j]$ has the same distribution in $\mathsf{Game}_4(\lambda)$ and $\mathsf{Game}_5(\lambda)$.

Game_5. Finally, observe that the advantage of $\mathcal{A}$ in $\mathsf{Game}_5(\lambda)$ is $1/2$. Indeed, the challenge bit d is completely independent of $\mathcal{A}$'s input (all input strings are random in $\mathsf{Game}_5(\lambda)$), which means that $\mathcal{A}$ gains no information about d.

Combining the above estimates we obtain Inequality (16) for the KDM-CPA advantage:

$$\begin{aligned}\mathrm{Adv}^{\text{kdm-cpa}}_{\mathsf{MEI}[\Gamma],\mathcal{A}}(\lambda) &= 2\cdot\mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + 2\cdot\Pr\left[\neg\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda)\;\middle|\; b=0\right] - 1\\ &\le 2\cdot\mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + 2\left(\frac{1}{2} + \frac{(4n(\lambda)-1)(n(\lambda)+q(\lambda)) + n(\lambda) + 3q(\lambda)^2}{2^{\lambda-1}}\right) - 1\\ &= 2\cdot\mathrm{Adv}^{\text{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + \frac{(4n(\lambda)-1)(n(\lambda)+q(\lambda)) + n(\lambda) + 3q(\lambda)^2}{2^{\lambda-2}}\,.\end{aligned}$$

Algebraic unpredictability. It remains to be shown that, for every KDM-CPA adversary $\mathcal{A}$, the masking source $\mathcal{S}$ defined via algorithm $\bar{\mathcal{S}}$ from Figure 27 (bottom) is algebraically unpredictable. So let $\mathcal{A}$ be a KDM-CPA adversary and $\mathcal{P}$ be any predictor in the algebraic unpredictability game against $\mathcal{S}$. We prove Inequality (17) via a sequence of games. As before, we give here a short description of each game, and present their formal code in Figure 30 (top).

$\mathsf{Game}'_0(\lambda)$ is the algebraic unpredictability game for $\mathcal{S}$ played by $\mathcal{P}$.
$\mathsf{Game}'_1(\lambda)$ is the same as $\mathsf{Game}'_0(\lambda)$, except that $\mathbf{s}[i], \mathbf{sk}[i], \mathbf{t}[j] \neq 0$ for all i, j.
$\mathsf{Game}'_2(\lambda)$ is the same as $\mathsf{Game}'_1(\lambda)$, but we ensure that all inputs to σ are pairwise different. We implement this by resampling $\mathbf{s}[i]$, $\mathbf{sk}[i]$, and $\mathbf{t}[j]$ if necessary.
$\mathsf{Game}'_3(\lambda)$ is the same as $\mathsf{Game}'_2(\lambda)$, but we replace evaluations of the encoding σ with random, pairwise different group elements.
$\mathsf{Game}'_4(\lambda)$ is the same as $\mathsf{Game}'_3(\lambda)$, but now we sample $\mathbf{pk}_1$, $\mathbf{pk}_2$, $\mathbf{c}_1$, and $\mathbf{c}_2$ at random, without any further checks.
$\mathsf{Game}'_5(\lambda)$ is the same as $\mathsf{Game}'_4(\lambda)$, but we revert our previous changes from $\mathsf{Game}'_2(\lambda)$ and again sample $\mathbf{s}[i]$, $\mathbf{sk}[i]$, and $\mathbf{t}[j]$ at random from $\mathbb{Z}_p^*$.
$\mathsf{Game}'_6(\lambda)$ is the same as $\mathsf{Game}'_5(\lambda)$, but we undo our first game hop and sample $\mathbf{s}[i]$, $\mathbf{sk}[i]$, and $\mathbf{t}[j]$ uniformly at random from $\mathbb{Z}_p$.
$\mathsf{Game}'_7(\lambda)$ is the same as $\mathsf{Game}'_6(\lambda)$, but with the order of a few instructions changed. More precisely, we interpret the sampling of sk and the computation of q as done jointly by the source $\mathcal{T}$ defined in Figure 30 (bottom). We can then decouple the two coordinates of $\mathcal{T}$ by defining a new source $\mathcal{T}'_1$ such that, for every $\pi \twoheadleftarrow \Gamma(1^\lambda)$, $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$, and $q \in [q(\lambda)]$, $\mathcal{T}'_1(\pi, n, q)$ is independent of $\mathcal{T}_2(\pi, n)$, and its distribution is the conditional distribution of $\mathcal{T}_1$ given $\mathcal{T}_2$ (both defined in Figure 30 (bottom)), i.e.,

$$\Pr[\mathcal{T}'_1(\pi, n, q) = \mathbf{sk}] \coloneqq \Pr[\mathcal{T}_1(\pi, n) = \mathbf{sk} \mid \mathcal{T}_2(\pi, n) = q]\,.$$

As before, we now relate the success probabilities of subsequent games.

$\mathsf{Game}'_0 \rightsquigarrow \mathsf{Game}'_4$. Notice that transitions from $\mathsf{Game}'_0(\lambda)$ to $\mathsf{Game}'_4(\lambda)$ are identical to those from $\mathsf{Game}_0(\lambda)$ to $\mathsf{Game}_5(\lambda)$ in the proof of Inequality (16) above, which means that we also incur in the same advantage loss. In other words,

$$\left|\Pr[\mathsf{Game}'_0(\lambda)] - \Pr[\mathsf{Game}'_4(\lambda)]\right| \le \frac{(4n(\lambda)-1)(n(\lambda)+q(\lambda)) + n(\lambda) + 3q(\lambda)^2}{2^{\lambda-1}}\,.$$

$\mathsf{Game}'_4 \rightsquigarrow \mathsf{Game}'_5$ and $\mathsf{Game}'_5 \rightsquigarrow \mathsf{Game}'_6$. Observe that the changes made reverse those introduced in the first two game hops, which means that we incur in the same advantage loss. From our earlier discussion we thus conclude that

$$\left|\Pr[\mathsf{Game}'_4(\lambda)] - \Pr[\mathsf{Game}'_5(\lambda)]\right| = \left|\Pr[\mathsf{Game}'_1(\lambda)] - \Pr[\mathsf{Game}'_2(\lambda)]\right| \le \frac{2n(\lambda) + q(\lambda)}{2^{\lambda-1}}\,,$$

$$\left|\Pr[\mathsf{Game}'_5(\lambda)] - \Pr[\mathsf{Game}'_6(\lambda)]\right| = \left|\Pr[\mathsf{Game}'_0(\lambda)] - \Pr[\mathsf{Game}'_1(\lambda)]\right| \le \frac{(2n(\lambda)-1)(n(\lambda)+q(\lambda)) + q(\lambda)^2}{2^{\lambda-1}}\,.$$

Game $\mathsf{Game}'_0(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p$
for $i = 1$ to n do
  $\mathbf{pk}_1[i] \leftarrow \sigma(\mathbf{s}[i])$; $\mathbf{pk}_2[i] \leftarrow \sigma(\mathbf{s}[i]\mathbf{sk}[i])$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \leftarrow \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$
$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q,$
  $\delta_1, \ldots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
  $\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_1(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do $\boxed{\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^*;\ \mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^*}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do $\boxed{\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^*}$
for $i = 1$ to n do
  $\mathbf{pk}_1[i] \leftarrow \sigma(\mathbf{s}[i])$; $\mathbf{pk}_2[i] \leftarrow \sigma(\mathbf{s}[i]\mathbf{sk}[i])$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \leftarrow \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$
$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q,$
  $\delta_1, \ldots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
  $\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_2(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$; $\sigma \twoheadleftarrow \mathrm{Inj}(\mathbb{Z}_p, \mathsf{G})$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $\boxed{M \leftarrow [\,]}$
for $i = 1$ to n do
  $\boxed{\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M;\ M \leftarrow M : \mathbf{s}[i]}$
  $\boxed{\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M;\ M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do
  $\boxed{\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^* \setminus (\mathbf{s}[\mathbf{p}[j]]^{-1} \cdot M \cup (\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]])^{-1} \cdot M)}$
  $\boxed{M \leftarrow M : \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j], \mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j]}$
for $i = 1$ to n do
  $\mathbf{pk}_1[i] \leftarrow \sigma(\mathbf{s}[i])$; $\mathbf{pk}_2[i] \leftarrow \sigma(\mathbf{s}[i]\mathbf{sk}[i])$
for $j = 1$ to q do
  $\mathbf{c}_1[j] \leftarrow \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j])$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \sigma(\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j])$
$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q,$
  $\delta_1, \ldots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
  $\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_3(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $M \leftarrow [\,]$; $\boxed{M' \leftarrow [\,]}$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M;\ M \leftarrow M : \mathbf{s}[i]$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M;\ M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do
  $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^* \setminus (\mathbf{s}[\mathbf{p}[j]]^{-1} \cdot M \cup$
    $\cup\, (\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]])^{-1} \cdot M)$
  $M \leftarrow M : \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j], \mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j]$
for $i = 1$ to n do
  $\boxed{\mathbf{pk}_1[i] \leftarrow \mathsf{G} \setminus M';\ M' \leftarrow M : \mathbf{pk}_1[i]}$
  $\boxed{\mathbf{pk}_2[i] \leftarrow \mathsf{G} \setminus M';\ M' \leftarrow M : \mathbf{pk}_2[i]}$
for $j = 1$ to q do
  $\boxed{\mathbf{c}_1[j] \leftarrow \mathsf{G} \setminus M';\ M' \leftarrow M : \mathbf{c}_1[j]}$
  $\boxed{\mathbf{c}'_2[j] \leftarrow \mathsf{G} \setminus M';\ M' \leftarrow M : \mathbf{c}'_2[j]}$
  $\mathbf{c}_2[j] \leftarrow \mathbf{m}_d[j] \circ \mathbf{c}'_2[j]$
$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q,$
  $\delta_1, \ldots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
  $\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_4(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $M \leftarrow [\,]$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus M;\ M \leftarrow M : \mathbf{s}[i]$
  $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^* \setminus \mathbf{s}[i]^{-1} \cdot M;\ M \leftarrow M : \mathbf{s}[i]\mathbf{sk}[i]$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do
  $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^* \setminus (\mathbf{s}[\mathbf{p}[j]]^{-1} \cdot M \cup$
    $\cup\, (\mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]])^{-1} \cdot M)$
  $M \leftarrow M : \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j], \mathbf{s}[\mathbf{p}[j]]\mathbf{sk}[\mathbf{p}[j]]\mathbf{t}[j]$
for $i = 1$ to n do $\boxed{\mathbf{pk}_1[i] \leftarrow \mathsf{G};\ \mathbf{pk}_2[i] \leftarrow \mathsf{G}}$
for $j = 1$ to q do $\boxed{\mathbf{c}_1[j] \leftarrow \mathsf{G};\ \mathbf{c}_2[j] \leftarrow \mathsf{G}}$
$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q,$
  $\delta_1, \ldots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
  $\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_5(\lambda)$:

$(\pi \coloneqq (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do $\boxed{\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p^*;\ \mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p^*}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do $\boxed{\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p^*}$
for $i = 1$ to n do $\mathbf{pk}_1[i] \leftarrow \mathsf{G}$; $\mathbf{pk}_2[i] \leftarrow \mathsf{G}$
for $j = 1$ to q do $\mathbf{c}_1[j] \leftarrow \mathsf{G}$; $\mathbf{c}_2[j] \leftarrow \mathsf{G}$
$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_q,$
  $\delta_1, \ldots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
  $\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_6(\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$
for $i = 1$ to n do $\boxed{\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p;\ \mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p}$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_d|$
for $j = 1$ to q do $\boxed{\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p}$
for $i = 1$ to n do $\mathbf{pk}_1[i] \leftarrow \mathsf{G}$; $\mathbf{pk}_2[i] \leftarrow \mathsf{G}$
for $j = 1$ to q do $\mathbf{c}_1[j] \leftarrow \mathsf{G}$; $\mathbf{c}_2[j] \leftarrow \mathsf{G}$
$(\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_q,$
$\delta_1, \dots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
$\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Game $\mathsf{Game}'_7(\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$d \twoheadleftarrow \{0,1\}$; $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$; $\boxed{q \twoheadleftarrow \mathcal{T}_2(\pi, n)}$
for $i = 1$ to n do $\mathbf{pk}_1[i] \leftarrow \mathsf{G}$; $\mathbf{pk}_2[i] \leftarrow \mathsf{G}$
for $j = 1$ to q do $\mathbf{c}_1[j] \leftarrow \mathsf{G}$; $\mathbf{c}_2[j] \leftarrow \mathsf{G}$
$(\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_q,$
$\delta_1, \dots, \delta_q) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)$
for $i = 1$ to n do $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$
$\boxed{\mathbf{sk} \twoheadleftarrow \mathcal{T}'_1(\pi, n, q)}$
for $j = 1$ to q do $\mathbf{t}[j] \twoheadleftarrow \mathbb{Z}_p$
return $\big(\sum_{i=1}^n \mathbf{s}[i](\alpha_i + \beta_i \mathbf{sk}[i]) +$
$\sum_{j=1}^q \mathbf{s}[\mathbf{p}[j]]\mathbf{t}[j](\gamma_j + \delta_j \mathbf{sk}[\mathbf{p}[j]]) = 0\big)$

Source $\mathcal{T}(\pi, n)$:

for $i = 1$ to n do $\mathbf{sk}[i] \twoheadleftarrow \mathbb{Z}_p$
$(\mathbf{p}, \mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{sk})$; $q \leftarrow |\mathbf{m}_0|$
return $(\mathbf{sk}, q)$

Source $\mathcal{T}_1(\pi, n)$:

$(\mathbf{sk}, q) \twoheadleftarrow \mathcal{T}(\pi, n)$
return $\mathbf{sk}$

Source $\mathcal{T}_2(\pi, n)$:

$(\mathbf{sk}, q) \twoheadleftarrow \mathcal{T}(\pi, n)$
return q

Figure 30 — *Top:* Code of the intermediate games in the proof of the algebraic unpredictability bound (17) in Theorem 6.3. *Bottom:* Definition of the source $\mathcal{T}$ and its projections $\mathcal{T}_1$ and $\mathcal{T}_2$ from the proof of Theorem 6.3. Source $\mathcal{T}'_1$ in Game'_7 is independent of $\mathcal{T}_2$ and has the distribution of $\mathcal{T}_1$ conditioned to $\mathcal{T}_2$.

$\mathsf{Game}'_6 \rightsquigarrow \mathsf{Game}'_7$. Notice that $\Pr[\mathsf{Game}'_6(\lambda)] = \Pr[\mathsf{Game}'_7(\lambda)]$ by construction of the source $\mathcal{T}'_1$. Indeed, for every $\pi \twoheadleftarrow \Gamma(1^\lambda)$, $1^n \twoheadleftarrow \mathcal{A}_0(\pi)$, $q \in [q(\lambda)]$, and $\mathbf{sk} \in \mathbb{Z}_p^n$, we have

$$\begin{aligned}\Pr[(\mathcal{T}_2(\pi, n) = q) \wedge (\mathcal{T}'_1(\pi, n, q) = \mathbf{sk})] &= \Pr[\mathcal{T}_2(\pi, n) = q] \cdot \Pr[\mathcal{T}'_1(\pi, n, q) = \mathbf{sk}] \\ &= \Pr[\mathcal{T}_2(\pi, n) = q] \cdot \Pr[\mathcal{T}_1(\pi, n) = \mathbf{sk} \mid \mathcal{T}_2(\pi, n) = q] \\ &= \Pr[(\mathcal{T}_1(\pi, n) = \mathbf{sk}) \wedge (\mathcal{T}_2(\pi, n) = q)] = \Pr[\mathcal{T}(\pi, n) = (\mathbf{sk}, q)]\,,\end{aligned}$$

which shows that $\mathbf{sk}$ and q have the same distribution in both games.

Game'_7. To conclude, we are left with bounding the winning probability of $\mathcal{P}$ in $\mathsf{Game}'_7(\lambda)$. Our strategy to do so is to apply the Schwartz–Zippel lemma, but observe that $\mathsf{Game}'_7(\lambda)$ does not quite fit the framework of Lemma 2.2, since $\mathcal{T}'_1$ is not a simple product distribution. Fortunately, the same ideas from the proof of Lemma 2.2 can be applied in this setting, to obtain a similar conclusion. Indeed, fix any π, d, n, q, $\mathbf{pk}_1$, $\mathbf{pk}_2$, $\mathbf{c}_1$, $\mathbf{c}_2$, and $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta})$ as in $\mathsf{Game}'_7(\lambda)$, and let $P_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}}$ be the polynomial defined by the coefficients returned by the predictor. Then notice that this polynomial has at most p^{2n+q-1} roots over $\mathbb{Z}_p$, since the degree of $P_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}}$ in any given variable is one. This means

$$\begin{aligned}\Pr_{\mathbf{s} \twoheadleftarrow \mathbb{Z}_p^n, \mathbf{sk} \twoheadleftarrow \mathcal{T}'_1(\pi,n,q), \mathbf{t} \twoheadleftarrow \mathbb{Z}_p^q}[P_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}}(\mathbf{s}, \mathbf{sk}, \mathbf{t}) = 0] &\le p^{2n+q-1} \cdot 2^{-\mathbf{H}_\infty(\mathcal{U}_{\mathbb{Z}_p^n}, \mathcal{T}'_1(\pi,n,q), \mathcal{U}_{\mathbb{Z}_p^q})} \\ &= p^{2n+q-1} \cdot 2^{-\mathbf{H}_\infty(\mathcal{U}_{\mathbb{Z}_p^n})} \cdot 2^{-\mathbf{H}_\infty(\mathcal{T}'_1(\pi,n,q))} \cdot 2^{-\mathbf{H}_\infty(\mathcal{U}_{\mathbb{Z}_p^q})} = p^{n-1} \cdot 2^{-\mathbf{H}_\infty(\mathcal{T}'_1(\pi,n,q))}\,,\end{aligned}$$

where $\mathcal{U}_{\mathbb{Z}_p^n}$ and $\mathcal{U}_{\mathbb{Z}_p^q}$ denote the uniform distributions over the corresponding sets. From here we then get

$$\begin{aligned}\Pr[\mathsf{Game}'_7(\lambda)] &\le \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \Gamma(1^\lambda), d \twoheadleftarrow \{0,1\}, 1^n \twoheadleftarrow \mathcal{A}_0(\pi), q \twoheadleftarrow \mathcal{T}_2(\pi,n), \mathbf{pk}_1, \mathbf{pk}_2 \twoheadleftarrow \mathsf{G}^n, \\ \mathbf{c}_1, \mathbf{c}_2 \twoheadleftarrow \mathsf{G}^q, (\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}) \twoheadleftarrow \mathcal{P}(\pi, \mathbf{pk}_1, \mathbf{pk}_2, \mathbf{c}_1, \mathbf{c}_2, d)}} \Big[p^{n-1} \cdot 2^{-\mathbf{H}_\infty(\mathcal{T}'_1(\pi,n,q))}\Big] \\ &= \mathop{\mathbb{E}}_{\pi \twoheadleftarrow \Gamma(1^\lambda), 1^n \twoheadleftarrow \mathcal{A}_0(\pi), q \twoheadleftarrow \mathcal{T}_2(\pi,n)} \Big[p^{n-1} \cdot \max_{\mathbf{sk} \in \mathbb{Z}_p^n} \Pr[\mathcal{T}'_1(\pi, n, q) = \mathbf{sk}]\Big]\end{aligned}$$

$$= \mathop{\mathbb{E}}_{\pi \twoheadleftarrow \Gamma(1^\lambda), 1^n \twoheadleftarrow \mathcal{A}_0(\pi)} \left[p^{n-1} \cdot \mathop{\mathbb{E}}_{q \twoheadleftarrow \mathcal{T}_2(\pi,n)} \left[\max_{\mathbf{sk} \in \mathbb{Z}_p^n} \Pr[\mathcal{T}_1(\pi,n) = \mathbf{sk} \mid \mathcal{T}_2(\pi,n) = q] \right] \right]$$

$$= \mathop{\mathbb{E}}_{\pi \twoheadleftarrow \Gamma(1^\lambda), 1^n \twoheadleftarrow \mathcal{A}_0(\pi)} \left[p^{n-1} \cdot \sum_{q=1}^{q(\lambda)} \max_{\mathbf{sk} \in \mathbb{Z}_p^n} \Pr[(\mathcal{T}_1(\pi,n) = \mathbf{sk}) \wedge (\mathcal{T}_2(\pi,n) = q)] \right]$$

$$\leq \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \Gamma(1^\lambda), \\ 1^n \twoheadleftarrow \mathcal{A}_0(\pi)}} \left[p^{n-1} \cdot \sum_{q=1}^{q(\lambda)} \max_{\mathbf{sk} \in \mathbb{Z}_p^n} \Pr[\mathcal{T}_1(\pi,n) = \mathbf{sk}] \right] \leq \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \Gamma(1^\lambda), \\ 1^n \twoheadleftarrow \mathcal{A}_0(\pi)}} \left[p^{n-1} \cdot q(\lambda) \cdot \frac{1}{p^n} \right] \leq \frac{q(\lambda)}{2^{\lambda-1}} \,.$$

Adding all our estimates above we obtain Inequality (17) for the algebraic unpredictability advantage. This concludes the proof. □

Remark. It is tempting to consider an alternative reduction $(\mathcal{S}, \mathcal{D})$ that uses the challenge oracle only to compute $\mathbf{c}_2$, and instead calculates $\mathbf{c}_1$ by itself using the real group operation. This appears to be a natural choice since only $\mathbf{c}_2$ depends on the challenge bit d, and would yield tighter bounds on the advantages because of a reduced number of oracle calls. Unfortunately, such a source is not guaranteed to be *statistically* algebraically unpredictable. Indeed, recall that $\mathcal{P}$ receives $\mathbf{c}_1$ and g' and, being unbounded, can compute $\mathbf{r}$ as the discrete logarithm of $\mathbf{c}_1$ to base g'. Now, if $\mathcal{A}$ is a KDM-CPA adversary returning a low-entropy $\mathbf{p}$, e.g., in the single-key setting, $\mathcal{P}$ can easily find a non-trivial linear combination.

Remark. We observe that, since our GGM feasibility result allows sources full access to the group injection, as a corollary we obtain KDM security in the presence of preprocessing attacks in the generic-group model.

6.4 IND Security of Deterministic ElGamal

Deterministic PKE and its security. We call a PKE scheme E *deterministic* if $\mathsf{E.Enc}$ is a deterministic algorithm. Deterministic PKE was introduced by Bellare, Boldyreva, and O'Neill (BBO) [BBO07]. Our security notion is based on the work of Bellare, Dowsley, and Keelveedhi [BDK15], which incorporates an additional adversary that can pass state to the second-phase adversary.[19] In keeping with our syntax, we allow the message samplers to depend on system parameters.

IND security. Let E be a deterministic PKE scheme. The advantage of an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the IND game of E is defined as

$$\mathrm{Adv}^{\mathrm{ind}}_{\mathsf{E},\mathcal{A}}(\lambda) \coloneqq 2 \cdot \Pr\left[\mathrm{IND}^{\mathcal{A}}_{\mathsf{E}}(\lambda)\right] - 1 \,,$$

where the IND game is defined in Figure 31 (top left). Here, $\mathcal{A}_0$ and $\mathcal{A}_1$ can be unbounded with polynomially bounded output, and $\mathcal{A}_2$ is PPT. We require that the vectors returned by $\mathcal{A}_1$ be of the same length, and that each vector contain pairwise distinct messages. Also, we require that $(\mathcal{A}_0, \mathcal{A}_1)$ be unpredictable, meaning that for any (possibly unbounded) predictor $\mathcal{P}$,

$$\mathrm{Adv}^{\mathrm{pred}}_{\mathsf{E},\mathcal{A}_0,\mathcal{A}_1,\mathcal{P}}(\lambda) \coloneqq \Pr\left[\mathrm{Pred}^{\mathcal{P}}_{\mathsf{E},\mathcal{A}_0,\mathcal{A}_1}(\lambda)\right]$$

is negligible, where the Pred game is given in Figure 31 (top right). We say that E is IND secure, if the advantage of any $\mathcal{A}$ as above in the IND game for E is negligible.

We note that the notions of statistical and *non-uniform* computational unpredictability in the Pred game coincide. Indeed, via a coin-fixing argument we may assume, without loss of generality, that $\mathcal{P}$ is

[19] This ensures a closer correspondence between idealized models of computation and the standard model as the state can be used for consistency and lazily sample ideal objects.

Game $\mathrm{IND}^{\mathcal{A}}_{\mathsf{E}}(\lambda)$:

$d \twoheadleftarrow \{0,1\}$; $\pi \twoheadleftarrow \mathsf{E.Setup}(1^\lambda)$
$\mathsf{st} \twoheadleftarrow \mathcal{A}_0(\pi)$; $(\mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\mathsf{st})$
$(sk, pk) \twoheadleftarrow \mathsf{E.KGen}(\pi)$; $\mathbf{c} \leftarrow \mathsf{E.Enc}(\pi, pk, \mathbf{m}_d)$
$b' \twoheadleftarrow \mathcal{A}_2(\pi, pk, \mathbf{c}, \mathsf{st})$; return $(d = d')$

Game $\mathrm{Pred}^{\mathcal{P}}_{\mathsf{E},\mathcal{A}_0,\mathcal{A}_1}(\lambda)$:

$\pi \twoheadleftarrow \mathsf{E.Setup}(1^\lambda)$; $\mathsf{st} \twoheadleftarrow \mathcal{A}_0(\pi)$
$(\mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\mathsf{st})$; $m' \twoheadleftarrow \mathcal{P}(\pi, \mathsf{st})$
return $(m' \in \mathbf{m}_0 \cup \mathbf{m}_1)$

$\mathsf{ElwH}[\Gamma, \mathsf{H}].\mathsf{Setup}(1^\lambda)$:

$(\pi := (\circ, g_0, p)) \twoheadleftarrow \Gamma(1^\lambda)$
$r' \twoheadleftarrow \mathbb{Z}_p^*$; $g' \leftarrow g_0^{r'}$
$\varpi \twoheadleftarrow \mathsf{H.Setup}(\mathsf{G}, \mathbb{Z}_p)$
return $(\pi' := (\varpi, \circ, g', p))$

$\mathsf{ElwH}[\Gamma, \mathsf{H}].\mathsf{KGen}(\pi')$:

$(\varpi, \circ, g', p) \leftarrow \pi'$
$s \twoheadleftarrow \mathbb{Z}_p^*$; $pk_1 \leftarrow g'^s$
$sk \twoheadleftarrow \mathbb{Z}_p^*$; $pk_2 \leftarrow g'^{ssk}$
$hk \twoheadleftarrow \mathsf{H.KGen}(\pi_{\mathsf{H}})$
$pk \leftarrow (pk_1, pk_2, hk)$
return (sk, pk)

$\mathsf{ElwH}[\Gamma, \mathsf{H}].\mathsf{Enc}(\pi', pk, m)$:

$(pk_1, pk_2, hk) \leftarrow pk$; $r \leftarrow \mathsf{H}(hk, m)$
$c_1 \leftarrow pk_1^r$; $c_2 \leftarrow m \circ pk_2^r$; return (c_1, c_2)

$\mathsf{ElwH}[\Gamma, \mathsf{H}].\mathsf{Dec}(\pi', sk, c)$:

$(c_1, c_2) \leftarrow c$; return $c_2 \circ c_1^{-sk}$

Auxiliary masking source $\bar{\mathcal{S}}(\pi)$:

$d \twoheadleftarrow \{0,1\}$; $(\circ, g_0, p) \leftarrow \pi$; $r' \twoheadleftarrow \mathbb{Z}_p^*$; $g' \leftarrow g_0^{r'}$
$\varpi \twoheadleftarrow \mathsf{H.Setup}(\mathsf{G}, \mathbb{Z}_p)$; $hk \twoheadleftarrow \mathsf{H.KGen}(\varpi)$; $\pi' \leftarrow (\varpi, \circ, g', p)$
$\mathsf{st} \twoheadleftarrow \mathcal{A}_0(\pi')$; $(\mathbf{m}_0, \mathbf{m}_1) \twoheadleftarrow \mathcal{A}_1(\mathsf{st})$; $t \twoheadleftarrow \mathbb{Z}_p^*$; $sk \twoheadleftarrow \mathbb{Z}_p^*$; $q \leftarrow |\mathbf{m}_0|$
for $i = 1$ to q do
  $\mathbf{x}[i] \leftarrow t \cdot \mathsf{H}(hk, \mathbf{m}_d[i])$; $\mathbf{x}[i+q] \leftarrow sk \cdot \mathbf{x}[i]$
  $\mathbf{m}[i] \leftarrow 1$; $\mathbf{m}[i+q] \leftarrow \mathbf{m}_d[i]$
$\mathbf{x}[2q+1] \leftarrow t$; $\mathbf{x}[2q+2] \leftarrow t \cdot sk$; $\mathbf{m}[2q+1] \leftarrow 1$; $\mathbf{m}[2q+2] \leftarrow 1$
$L \leftarrow (d, \mathsf{st}_0, \pi', hk)$; return $(\mathbf{x}, \mathbf{m}, L)$

PGG distinguisher $\mathcal{D}(\pi, (\mathbf{y}, L))$:

$(d, \mathsf{st}_0, \pi', hk) \leftarrow L$
$(\mathbf{c}, pk_1, pk_2) \leftarrow \mathbf{y}$
$d' \twoheadleftarrow \mathcal{A}_2(\pi', (pk_1, pk_2, hk), \mathbf{c}, \mathsf{st})$
return $(d = d')$

Figure 31 — *Top left:* The IND security game for a (deterministic) public-key encryption scheme. *Top right:* The security game defining unpredictability of the messages output by $\mathcal{A}$. *Center:* The ElGamal-with-Hash scheme $\mathsf{ElwH}[\Gamma, \mathsf{H}]$ with user-specific randomized generator included in the public key as pk_1. *bottom:* Parallel PGG source and distinguisher associated with an IND adversary $\mathcal{A}$ against ElGamal-with-Hash.

deterministic. Next we may hard-wire the value returned by $\mathcal{P}$ (whose computation may be non-polynomial time) as non-uniform advice. As such, our results in this section also hold when Pred is defined (only) with respect to non-uniform predictors.[20]

Construction. We formalize the *ElGamal-with-Hash* deterministic encryption scheme, which is a special case of the general encrypt-with-hash (EwH) construct of BBO. Let Γ be a computational group scheme, and let H be a hash function family. The associated ElGamal-with-Hash scheme $\mathsf{ElwH}[\Gamma, \mathsf{H}]$ is defined in Figure 31 (center).

We now prove that the encryption scheme $\mathsf{ElwH}[\Gamma, \mathsf{H}]$ associated to a computational group scheme Γ and a hash family H is IND secure under PGG and LDD.

Theorem 6.4 (PGG $\wedge$ LDD $\Longrightarrow$ ElGamal-with-Hash is IND)**.** *Let Γ be a computational group scheme, and H be a hash function family. If Γ is $\mathrm{PGG}[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{msk}}]$ secure and H is $\mathrm{LDD}[\mathbf{S}^{\mathrm{sup}}]$ secure, then $\mathsf{E} := \mathsf{ElwH}[\Gamma, \mathsf{H}]$ is* IND *secure. More precisely, for any adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the IND game for E, there are an adversary $(\mathcal{S}, \mathcal{D})$ in the PGG game for Γ, and an adversary $(\mathcal{S}', \mathcal{A}_{\mathrm{cr}})$ in the LDD game for H such that*

$$\mathrm{Adv}^{\mathrm{ind}}_{\mathsf{E},\mathcal{A}}(\lambda) \le 2 \cdot \mathrm{Adv}^{\mathrm{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + q(\lambda)^2 \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}_{\mathrm{cr}}}(\lambda) + \frac{4(q(\lambda)+1)}{2^{\lambda-1}} + \frac{(2q(\lambda)+2)^2}{2^{\lambda-1}} .$$

[20] This observation does not hold in the presence of parameters. Despite this, our results extend to a setting where $\mathcal{A}_0$ and $\mathcal{A}_1$ receive π but Pred is statistical.

Here, $q(\lambda)$ is an upper bound on the length of the vectors returned by $\mathcal{A}_1$. Furthermore, $\mathcal{S} \in \mathbf{S}^{\text{alg}} \cap \mathbf{S}^{\text{msk}}$ and, for any predictor $\mathcal{P}$ there exists an adversary $\mathcal{A}'$ such that

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \le 2 \cdot \mathrm{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}'}(\lambda) + \frac{q(\lambda)^2}{2} \cdot \mathrm{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}_{\text{cr}}}(\lambda) + \frac{8(q(\lambda)+1)^2}{2^\lambda} .$$

Moreover, $\mathcal{S}' \in \mathbf{S}^{\text{sup}}$ and, for any predictor $\mathcal{P}'$ there exists a predictor $\mathcal{P}''$ such that

$$\mathrm{Adv}^{\text{pred}}_{\mathsf{H},\mathcal{S}',\mathcal{P}'}(\lambda) \le \mathrm{Adv}^{\text{pred}}_{\mathsf{E},\mathcal{A}_0,\mathcal{A}_1,\mathcal{P}''}(\lambda) .$$

Proof. For the proof of this theorem we consider the source and distinguisher in Figure 31 (bottom). Here, the source computes a random generator $g_0^{r'}$ so as to be able to compute the public parameters for the scheme. The source then computes the public key and ciphertexts as in the scheme using the Exp oracle. Note, however, that these group elements will have an extra randomness component r due to the randomized generator in the PGG game, whereas they need to be computed with respect to $g_0^{r'}$ as public parameters are set up. However, since we choose a random t for the exponent of the first component of the public key, the effective randomness in the reduction is $r't/r$, which is randomly distributed. From here the proof boils down to arguing for the algebraic unpredictability of the source. For this we first invoke the Schwartz–Zippel lemma, so that any algebraic relation between the exponents queried by the source reduces to one among the exponents for the second ciphertext elements only. This, however, leads to a break of LDD security since the message sampler for is unpredictable.

More formally, given an adversary $\mathcal{A}$ in the IND game, define the masking source $\mathcal{S}$ via auxiliary algorithm $\bar{\mathcal{S}}$ and distinguisher $\mathcal{D}$ as shown in Figure 31 (bottom). It is easy to see by inspecting the code in Figure 31 (bottom) that $\bar{\mathcal{S}}$ places no Exp queries and returns vectors $\mathbf{x}$ and $\mathbf{m}$ of equal length. Thus, $\mathcal{S} \in \mathbf{S}^{\text{msk}}$.

Advantage bound. Let b denote the challenge bit in the PGG game. We claim that

$$\Pr\left[\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 1\right] = \Pr\left[\mathrm{IND}^{\mathcal{A}}_{\mathsf{E}}(\lambda)\right] .$$

When $b = 1$, the exponentiation oracle is implemented via the real group operation. The source computes the system parameters by picking a random group generator g' and a hash key hk. It then generates the two message vectors via $\mathcal{A}_0$ and $\mathcal{A}_1$. The initial phase of the source also picks random t and sk. Here, sk will act as the secret key. On the other hand, t is not the random exponent of the first element of the public key; rather, the exponent is $s = rt/r'$, where r is the exponent sampled in the PGG game.

The source computes each ciphertext via two oracle calls. These take the form $g_0^{rt\mathbf{r}[i]}$ and $g_0^{rtsk\mathbf{r}[i]}$, where $\mathbf{r}[i] = \mathsf{H}(hk, \mathbf{m}_d[i])$ is the hash of the message. It then computes the public key as (g_0^{rt}, g_0^{rtsk}). Note that these are correctly distributed as in the scheme where rt/r' is the exponent of the first element of the public key, which is uniform as t is uniform. The addition of the extra group element in the public key essentially re-randomizes the generator allowing the first stage of the attack to fully depend on the public parameters.

We now claim that

$$\Pr\left[\mathrm{PGG}^{\mathcal{S},\mathcal{D}}_{\Gamma}(\lambda) \,\middle|\, b = 0\right] \le \frac{1}{2} + \frac{q(\lambda)^2}{2}\mathrm{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}_{\text{cr}}}(\lambda) + \frac{2(q(\lambda)+1)}{2^{\lambda-1}} + \frac{(2q(\lambda)+2)^2}{2^\lambda} .$$

The public key is distributed randomly and is independent of d. We show that the rest of the group elements are also independent of d. To this end we transition to a game where we replace Exp with a random oracle. The two games are identical up to RF/RP switching advantage $(2q(\lambda)+2)^2/2^\lambda$.

Next we replace the random oracle by a forgetful function. Unless there is a point that is queried twice to Exp the two games are identical. We show this event leads to breaking the LDD property. Consider

the LDD source $\mathcal{S}'$ that runs the PGG source $\mathcal{S}$, which itself runs $(\mathcal{A}_0, \mathcal{A}_1)$. Suppose collisions in the inputs to Exp happen. Consider now an LDD adversary $\mathcal{A}_{\text{cr}}$ against the source that picks two distinct indices $i, j \in [q]$ and sets $\alpha_i = -\alpha_j = 1$ and the rest of the α's to zero. Whenever there is a collision in the outputs of H, and if i and j are guessed correctly (which happens with probability $\binom{q(\lambda)}{2}$), the adversary wins the LDD game, unless the collisions were due to t and $t \cdot sk$, which happens with probability $2(q(\lambda)+1)/2^{\lambda-1}$. Note that here we rely on the fact that $\mathcal{A}_1$ outputs distinct messages. It remains to be shown that this LDD source is unpredictable. This follows from the fact that any predictor against this source can be converted to a DE predictor against $(\mathcal{A}_0, \mathcal{A}_1)$, as we will show momentarily below.

Algebraic unpredictability. We now show that the PGG source defined above is algebraically unpredictable. To this end we show that if there is an algebraic predictor against this source, then there is an LDD adversary against an unpredictable LDD source.

We first modify the algebraic unpredictability game and transition to a new game where Exp is a forgetful oracle. As argued above this change is negligible and we pick up similar terms as those given up. In this modified game, any algebraic predictor $\mathcal{P}$ against the PGG source can be converted to an LDD adversary $\mathcal{A}'$ as follows. Algorithm $\mathcal{A}'$ runs the PGG algebraic predictor $\mathcal{P}$ on π' and st_0 and simulates the leakage for it by sampling the ciphertexts and the public key randomly from the group. It also picking a random $d \twoheadleftarrow \{0,1\}$, which is correct with probability $1/2$. Hence

$$\mathrm{Adv}^{\text{modified-alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \leq 2 \cdot \mathrm{Adv}^{\text{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}'}(\lambda)\,.$$

LDD source unpredictability. It remains to show $\mathcal{S}'$ is unpredictable. To this end, we show that any LDD predictor $\mathcal{P}'$ against $\mathcal{S}'$ can be converted to a predictor $\mathcal{P}$ against the message-sampler $\mathcal{A}_1$. This is immediate as the LDD source runs the message sampler and picks one of the message vectors randomly. Any predictor for this LDD source can be converted a predictor $\mathcal{P}''$ for the message sampler by simply running the LDD predictor and outputting the result:

$$\mathrm{Adv}^{\text{pred}}_{\mathsf{H},\mathcal{S}',\mathcal{P}'}(\lambda) \leq \mathrm{Adv}^{\text{pred}}_{\mathsf{E},\mathcal{A}_0,\mathcal{A}_1,\mathcal{P}''}(\lambda)\,.$$

This concludes proof of the theorem. □

6.5 RKA Security of ElGamal

In a related-key attack against a public-key encryption scheme, an adversary is able to obtain encryptions of messages under public keys that may be correlated.[21] This notion was introduced by Biham and Knudsen [BK98] in the context of symmetric encryption, and was later considered in the public-key setting by Goyal, O'Neill, and Rao [GOR11] and Bellare, Cash, and Miller [BCM11]. Here, we focus on a non-adaptive version of RKA security under chosen-plaintext attacks (RK-CPA) as in [GOR11], and prove that a modified version of ElGamal scheme where one hashes randomness via an LDD to derive secret keys is RKA secure. We call this scheme "shielded" ElGamal.

RK-CPA security. Let E be a PKE scheme. The advantage of a PPT adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ in the RK-CPA game for E is defined as

$$\mathrm{Adv}^{\text{rk-cpa}}_{\mathsf{E},\mathcal{A}}(\lambda) \coloneqq 2 \cdot \Pr\left[\text{RK-CPA}^{\mathcal{A}}_{\mathsf{E}}(\lambda)\right] - 1\,,$$

where the RK-CPA game is defined in Figure 32 (top left). We require that, for all $1 \leq i \leq |\mathbf{r}|$, each $\mathbf{r}[i]$ be unpredictable given st and the equality pattern of $\mathbf{r}$. Formally this means that, for any (possibly unbounded) predictor $\mathcal{P}$,

$$\mathrm{Adv}^{\text{pred}}_{\mathsf{E.Setup},\mathcal{A}_0,\mathcal{P}}(\lambda) \coloneqq \Pr\left[\text{Pred}^{\mathcal{P}}_{\mathsf{E.Setup},\mathcal{A}_0}(\lambda)\right]$$

[21] A stronger, chosen-ciphertext notion, which we omit here, allows the adversary to make decryption queries under correlated secret keys.

Game RK-CPA$^{\mathcal{A}}_{\mathsf{E}}(\lambda)$:

$b \twoheadleftarrow \{0,1\}$; $\pi \twoheadleftarrow \mathsf{E.Setup}(1^\lambda)$; $(\mathbf{r}, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(1^\lambda)$
for $i = 1$ to $|\mathbf{r}|$ do
  $(\mathbf{sk}[i], \mathbf{pk}[i]) \leftarrow \mathsf{E.KGen}(\pi; \mathbf{r}[i])$
$b' \twoheadleftarrow \mathcal{A}_1^{\mathrm{RKEnc}}(\pi, \mathbf{pk}, \mathsf{st})$; return $(b = b')$

Proc. RKEnc(i, m_0, m_1)

return $\mathsf{E.Enc}(\mathbf{pk}[i], m_b)$

Game Pred$^{\mathcal{P}}_{\mathsf{E},\mathcal{A}_0}(\lambda)$:

$\pi \twoheadleftarrow \mathsf{E.Setup}(1^\lambda)$; $(\mathbf{r}, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(1^\lambda)$
for $i = 1$ to $|\mathbf{r}|$ do
  $(\mathbf{pk}[i], \mathbf{sk}[i]) \leftarrow \mathsf{E.KGen}(\pi; \mathbf{r}[i])$
$z_{\mathsf{coll}} \leftarrow \mathsf{Colls}(\mathbf{sk})$; $(i, sk) \twoheadleftarrow \mathcal{P}(\pi, \mathsf{st}, z_{\mathsf{coll}})$
return $(\mathbf{sk}[i] = sk)$

Auxiliary dUber source $\bar{\mathcal{S}}(\pi)$:

$(\circ, g_0, p) \leftarrow \pi$; $\pi' \twoheadleftarrow \mathsf{E.Setup}(1^\lambda)$
$\pi_{\mathsf{H}} \twoheadleftarrow \mathsf{H.Setup}(\mathsf{G}, \mathbb{Z}_p)$; $hk \twoheadleftarrow \mathsf{H.KGen}(\pi_{\mathsf{H}})$
$(\mathbf{sk}, \mathsf{st}) \twoheadleftarrow \mathcal{A}_0(1^\lambda)$
for $i = 1$ to n do
  $\mathbf{s}[i] \twoheadleftarrow \mathbb{Z}_p$; $\mathbf{sk}'[i] \leftarrow \mathsf{H}(hk, \mathbf{sk}[i])$; $\mathbf{t}[i] \leftarrow \mathbf{sk}'[i] \cdot \mathbf{s}[i]$
return $((1, \mathbf{sk}', \mathbf{s}, \mathbf{t}), hk, \mathsf{st})$

PGG distinguisher $\mathcal{D}(\pi, (y, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathsf{st}))$:

$d \twoheadleftarrow \{0,1\}$; $\pi' \leftarrow (\circ, y, p, hk)$
$d' \twoheadleftarrow \mathcal{A}_1^{\mathrm{RKEnc}}(\pi', \mathbf{y}_1, \mathsf{st})$
return $(d = d')$

Proc. RKEnc(i, m_0, m_1):

return $(\mathbf{y}_2[i], m_d \circ \mathbf{y}_3[i])$

Figure 32 — *Top left:* The RK-CPA game for a public-key encryption scheme E. *Bottom left:* The unpredictability game for $\mathcal{A}_0$. *Right:* Reduction from an RK-CPA adversary $\mathcal{A}$ to a PGG adversary $(\mathcal{S}, \mathcal{D})$.

is negligible, where the Pred game is given in Figure 32 (bottom left). Here, $\mathsf{Colls}(\mathbf{sk})$ returns the repetition pattern of $\mathbf{sk}$.[22] We say that E is RK-CPA secure, if the advantage of any $\mathcal{A}$ as above in the RK-CPA game for E is negligible.

Theorem 6.5 (PGG $\Longrightarrow$ Shielded ElGamal is RK-CPA)**.** *Let Γ be a computational group scheme, and H be a hash function family. If Γ is* PGG$[\mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{duber}}]$ *secure and H is* LDD$[\mathbf{S}^{\mathrm{sup}}]$ *secure, then $\mathsf{E} \coloneqq \mathsf{El}[\Gamma, \mathsf{H}]$ is* IND *secure. More precisely, for any adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in the* RK-CPA *game for E, there are an adversary $(\mathcal{S}, \mathcal{D})$ in the PGG game for Γ, and an adversary $(\mathcal{S}', \mathcal{A}_{\mathrm{cr}})$ in the LDD game for H such that*

$$\mathrm{Adv}^{\text{rk-cpa}}_{\mathsf{E},\mathcal{A}}(\lambda) \le 2 \cdot \mathrm{Adv}^{\mathrm{pgg}}_{\Gamma,\mathcal{S},\mathcal{D}}(\lambda) + q(\lambda)^2 \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}_{\mathrm{cr}}}(\lambda) + \frac{3(q(\lambda)+1)}{2^{\lambda-1}} + \frac{(2q(\lambda)+2)^2}{2^{\lambda-1}} .$$

Furthermore, $\mathcal{S} \in \mathbf{S}^{\mathrm{alg}} \cap \mathbf{S}^{\mathrm{duber}}$ and, for any predictor $\mathcal{P}$ there exists an adversary $\mathcal{A}'$ such that

$$\mathrm{Adv}^{\text{alg-pred}}_{\Gamma,\mathcal{S},\mathcal{P}}(\lambda) \le 2 \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}'}(\lambda) + \frac{q(\lambda)^2}{2} \cdot \mathrm{Adv}^{\mathrm{ldd}}_{\mathsf{H},\mathcal{S}',\mathcal{A}_{\mathrm{cr}}}(\lambda) + \frac{8(q(\lambda)+1)^2}{2^\lambda} .$$

Moreover, $\mathcal{S}' \in \mathbf{S}^{\mathrm{sup}}$ and, for any predictor $\mathcal{P}'$ there exists a predictor $\mathcal{P}''$ such that

$$\mathrm{Adv}^{\mathrm{pred}}_{\mathsf{H},\mathcal{S}',\mathcal{P}'}(\lambda) \le \mathrm{Adv}^{\mathrm{pred}}_{\mathsf{E},\mathcal{A}_0,\mathcal{P}''}(\lambda) + \frac{3(q(\lambda)+1)}{2^\lambda} .$$

Proof Overview. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ in the RK-CPA game for $\mathsf{El}[\Gamma]$. For simplicity, we assume $\mathcal{A}_1$ makes one encryption query per index i. The proof extends to the general case but requires a bit more bookkeeping. If there are at most $q_{\mathsf{enc}}(\lambda)$ encryption queries, we need to prepare $q_{\mathsf{enc}}(\lambda)$ ciphertext components for *each* public key in case all encryption queries are requested for the same public key.

Let $|\mathbf{sk}| = n$. Define the dUber source $\mathcal{S}$ via auxiliary algorithm $\bar{\mathcal{S}}$ and distinguisher $\mathcal{D}$ as shown in Figure 32 (right). (It's easy to see by code inspection that this source is indeed dUber.)

[22] Typically RK-CPA secure is considered with respect to claw-free function. This is a special case of our definition where the collision pattern is trivial.

Note that if $b = 1$, $\mathcal{A}$ is playing the real RK-CPA game. Indeed, the ciphertexts and public keys are computed with respect to the same generator as that for the PGG group.

$$\Pr\left[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\middle|\, b = 1\right] = \mathrm{Adv}^{\text{rk-cpa}}_{\mathsf{El}[\Gamma],\mathcal{A}}(\lambda)\,.$$

Moreover, when $b = 0$, we can argue that $\mathcal{A}$ gets no information about d since the group element masking the message is uniformly distributed. In other words

$$\Pr\left[\mathrm{PGG}_{\Gamma}^{\mathcal{S},\mathcal{D}}(\lambda) \,\middle|\, b = 0\right] \approx \frac{1}{2}\,.$$

The precise upper bound proceeds similarly to the reduction for deterministic encryption by first running out collisions in Exp inputs via LDD security, and then replacing Exp by a forgetful random oracle via RF/RP switching lemma.

Algebraic unpredictability. The algebraic unpredictability of the PGG source is reduced to the unpredictability of the RKA adversary $\mathcal{A}_0$ and the security of LDD. Consider an algebraic predictor against $\mathcal{S}$. This predictor outputs values $(\alpha_0, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n)$ such that

$$\alpha_0 \cdot 1 + \sum_{i=1}^{n} \alpha_i \cdot \mathbf{sk}'[i] + \sum_{i=1}^{n} \beta_i \cdot \mathbf{s}[i] + \sum_{i=1}^{n} \gamma_i \cdot \mathbf{s}[i]\mathbf{sk}'[i] = 0\,.$$

Applying Schwartz–Zippel over $\mathbf{s}[i]$ for each i we obtain that except with negligible probability

$$\forall i : \beta_i + \gamma_i \cdot \mathbf{sk}'[i] = 0\,.$$

If for some i, $\gamma_i \neq 0$, we can use the corresponding equation to compute $\mathbf{sk}[i] = -\beta_i/\gamma_i$, and win the unpredictability game for $\mathcal{A}_0$. On the other hand, if $\gamma_i = 0$ for all i, then necessarily we also have that $\beta_i = 0$. Hence,

$$\alpha_0 + \sum_{i=1}^{n} \alpha_i \cdot \mathbf{sk}'[i] = \alpha_0 + \sum_{i=1}^{n} \alpha_i \cdot \mathsf{H}(hk, \mathbf{sk}[i]) = 0\,.$$

This, however, translates to a break of LDD security since by the security of $\mathcal{A}_0$ the keys $\mathbf{sk}[i]$ are unpredictable. ■

Acknowledgments

We thank Sogol Mazaheri for collaborating in the early stages of this work. We also thank the anonymous reviewers who helped improve the presentation of our results. Pooya Farshim was supported in part by EPSRC grant EP/V034065/1. Patrick Harasser was funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297. Adam O'Neill was supported in part by a gift from Cisco.

References

[AH18] Thomas Agrikola and Dennis Hofheinz. Interactively secure groups from obfuscation. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 341–370. Springer, Heidelberg, March 2018. (Cited on pages 9 and 10.)

[AHK20] Thomas Agrikola, Dennis Hofheinz, and Julia Kastner. On instantiating the algebraic group model from falsifiable assumptions. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 96–126. Springer, Heidelberg, May 2020. (Cited on page 9.)

[App11] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, Heidelberg, May 2011. (Cited on pages 8 and 59.)

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005. (Cited on pages 3, 7, and 44.)

[BBN+09] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, Heidelberg, December 2009. (Cited on page 9.)

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, Heidelberg, August 2007. (Cited on pages 8 and 68.)

[BC10] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 520–537. Springer, Heidelberg, August 2010. (Cited on page 7.)

[BCM11] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 486–503. Springer, Heidelberg, December 2011. (Cited on pages 8 and 71.)

[BCPT13] Eleanor Birrell, Kai-Min Chung, Rafael Pass, and Sidharth Telang. Randomness-dependent message security. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 700–720. Springer, Heidelberg, March 2013. (Cited on page 9.)

[BDH14] Florian Böhl, Gareth T. Davies, and Dennis Hofheinz. Encryption schemes secure under related-key and key-dependent message attacks. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 483–500. Springer, Heidelberg, March 2014. (Cited on page 9.)

[BDK15] Mihir Bellare, Rafael Dowsley, and Sriram Keelveedhi. How secure is deterministic encryption? In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 52–73. Springer, Heidelberg, March / April 2015. (Cited on page 68.)

[BFL20] Balthazar Bauer, Georg Fuchsbauer, and Julian Loss. A classification of computational assumptions in the algebraic group model. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 121–151. Springer, Heidelberg, August 2020. (Cited on pages 4 and 9.)

[BFM14] Christina Brzuska, Pooya Farshim, and Arno Mittelbach. Indistinguishability obfuscation and UCEs: The case of computationally unpredictable sources. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 188–205. Springer, Heidelberg, August 2014. (Cited on pages 5, 6, 11, and 18.)

[BH15] Mihir Bellare and Viet Tung Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 627–656. Springer, Heidelberg, April 2015. (Cited on page 8.)

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, Heidelberg, August 2008. (Cited on pages 8, 44, and 59.)

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 398–415. Springer, Heidelberg, August 2013. (Cited on pages 3, 4, 6, 7, 8, 11, 12, 44, 53, and 59.)

[BHK14] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Cryptography from compression functions: The UCE bridge to the ROM. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 169–187. Springer, Heidelberg, August 2014. (Cited on page 53.)

[BK98] Eli Biham and Lars R. Knudsen. Cryptanalysis of the ANSI X9.52 CBCM mode. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 100–111. Springer, Heidelberg, May / June 1998. (Cited on page 71.)

[Bla06] John Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In Matthew J. B. Robshaw, editor, *FSE 2006*, volume 4047 of *LNCS*, pages 328–340. Springer, Heidelberg, March 2006. (Cited on page 3.)

[BM14] Christina Brzuska and Arno Mittelbach. Using indistinguishability obfuscation via UCEs. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 122–141. Springer, Heidelberg, December 2014. (Cited on page 9.)

[BMZ19] James Bartusek, Fermi Ma, and Mark Zhandry. The distinction between fixed and random generators in group-based assumptions. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 801–830. Springer, Heidelberg, August 2019. (Cited on pages 4, 7, 10, 20, 21, and 29.)

[Boy08] Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, Heidelberg, September 2008. (Cited on pages 3, 7, 9, and 44.)

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289. Springer, Heidelberg, August 2004. (Cited on page 10.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006. (Cited on page 16.)

[BRS03] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Heidelberg, August 2003. (Cited on pages 8 and 59.)

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 455–469. Springer, Heidelberg, August 1997. (Cited on pages 3, 7, and 45.)

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 489–508. Springer, Heidelberg, April 2008. (Cited on page 7.)

[CDG18] Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 693–721. Springer, Heidelberg, August 2018. (Cited on pages 6, 20, 21, and 29.)

[CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 227–258. Springer, Heidelberg, April / May 2018. (Cited on page 33.)

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998. (Cited on page 6.)

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004. (Cited on page 3.)

[CK18] Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 415–447. Springer, Heidelberg, April / May 2018. (Cited on page 29.)

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001. (Cited on pages 8 and 59.)

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998. (Cited on page 12.)

[Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 100–109. Springer, Heidelberg, December 2002. (Cited on pages 3 and 6.)

[DHZ14] Ivan Damgård, Carmit Hazay, and Angela Zottarel. Short paper on the generic hardness of DDH-II, 2014. (Cited on page 7.)

[DL78] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. (Cited on page 11.)

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013. (Cited on page 9.)

[FF20] Peter Fenteany and Benjamin Fuller. Same point composable and nonmalleable obfuscated point functions. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *ACNS 20, Part II*, volume 12147 of *LNCS*, pages 124–144. Springer, Heidelberg, October 2020. (Cited on page 10.)

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018. (Cited on pages 4 and 9.)

[GGH20] Sanjam Garg, Romain Gay, and Mohammad Hajiabadi. Master-key KDM-secure IBE from pairings. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 123–152. Springer, Heidelberg, May 2020. (Cited on page 59.)

[GKMZ16] Matthew D. Green, Jonathan Katz, Alex J. Malozemoff, and Hong-Sheng Zhou. A unified approach to idealized model separations via indistinguishability obfuscation. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 587–603. Springer, Heidelberg, August / September 2016. (Cited on page 3.)

[GOR11] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, Heidelberg, March 2011. (Cited on pages 7, 31, 53, and 71.)

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st FOCS*, pages 305–313. IEEE Computer Society Press, November 2000. (Cited on pages 8 and 30.)

[HO13] Brett Hemenway and Rafail Ostrovsky. Building lossy trapdoor functions from lossy encryption. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 241–260. Springer, Heidelberg, December 2013. (Cited on page 8.)

[KM20] Fuyuki Kitagawa and Takahiro Matsuda. Circular security is complete for KDM security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 253–285. Springer, Heidelberg, December 2020. (Cited on page 59.)

[KP19] Julia Kastner and Jiaxin Pan. Towards instantiating the algebraic group model. Cryptology ePrint Archive, Report 2019/1018, 2019. https://eprint.iacr.org/2019/1018. (Cited on page 9.)

[KY18] Ilan Komargodski and Eylon Yogev. Another step towards realizing random oracles: Non-malleable point obfuscation. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 259–279. Springer, Heidelberg, April / May 2018. (Cited on page 10.)

[KZZ22] Jonathan Katz, Cong Zhang, and Hong-Sheng Zhou. An analysis of the algebraic group model. Cryptology ePrint Archive, Report 2022/210, 2022. https://eprint.iacr.org/2022/210. (Cited on page 9.)

[LNPT20] Benoît Libert, Khoa Nguyen, Alain Passelègue, and Radu Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 128–158. Springer, Heidelberg, December 2020. (Cited on page 59.)

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005. (Cited on pages 3, 12, and 20.)

[Nec94] V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994. (Cited on pages 3 and 12.)

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997. (Cited on page 7.)

[PSS14] Kenneth G. Paterson, Jacob C. N. Schuldt, and Dale L. Sibborn. Related randomness attacks for public key encryption. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 465–482. Springer, Heidelberg, March 2014. (Cited on page 9.)

[Sch80] Jack T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.*, 27(4):701–717, 1980. (Cited on page 11.)

[Sho96] Victor Shoup. On fast and provably secure message authentication based on universal hashing. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 313–328. Springer, Heidelberg, August 1996. (Cited on page 6.)

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. (Cited on pages 3, 12, and 20.)

[ST17] Pratik Soni and Stefano Tessaro. Public-seed pseudorandom permutations. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 412–441. Springer, Heidelberg, April / May 2017. (Cited on pages 9 and 53.)

[ST18] Pratik Soni and Stefano Tessaro. Naor-Reingold goes public: The complexity of known-key security. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 653–684. Springer, Heidelberg, April / May 2018. (Cited on page 9.)

[Vad12] Salil P. Vadhan. *Pseudorandomness*, volume 7 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers, Boston-Delft, 2012. (Cited on pages 30 and 40.)

[Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 479–508. Springer, Heidelberg, August 2016. (Cited on page 3.)

[Zha22] Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022. (Cited on pages 3, 6, 9, and 20.)

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and algebraic computation EUROSAM*. Springer, Berlin-New York, 1979. (Cited on page 11.)

[ZZ21] Mark Zhandry and Cong Zhang. The relationship between idealized models under computationally bounded adversaries. Cryptology ePrint Archive, Report 2021/240, 2021. https://eprint.iacr.org/2021/240. (Cited on page 59.)

A Proof of Lemma 2.2

In this section we prove the game-based version of the Schwartz–Zippel lemma. We start with a preparatory lemma, which essentially is Lemma 2.2 but for fixed parameters π, polynomial P, and $y = 0$. The general proof then follows by averaging this intermediate result. We observe that the proof of Lemma A.1 strongly resembles the proof of the classical Schwartz–Zippel lemma.

The results in this section are first presented using the (game-based) language of unpredictability, better suited to applications in cryptography, and only later reformulated in the setting of min-entropy.

Lemma A.1. *Let $\alpha, m, p \in \mathbb{N}$ with p prime, and let $\mathcal{S}_1, \ldots, \mathcal{S}_m$ be distributions over $\mathbb{F}_{p^\alpha}$ such that, for every $i \in [m]$ and every (possibly unbounded) predictor $\mathcal{P}$,*

$$\mathrm{Adv}^{\mathrm{pred}}_{\mathcal{S}_i,\mathcal{P}} := \Pr\left[\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}_i}\right] \leq k_i \in \mathbb{R}\,,$$

where the game $\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}}$ is defined in Figure 33 (left). Then, for every non-constant polynomial $P \in \mathbb{F}_{p^\alpha}[X_1, \ldots, X_m]$, we have

$$\Pr_{x_1 \twoheadleftarrow \mathcal{S}_1, \ldots, x_m \twoheadleftarrow \mathcal{S}_m}[P(x_1, \ldots, x_m) = 0] \leq \deg(P) \cdot k_P\,,$$

where $k_P := \max_{i \in T_P} k_i$ and $T_P := \{i \in [m] \mid \deg_{X_i}(P) > 0\}$.

Proof. We prove the statement via induction on m. For $m = 1$, let $r_1, \ldots, r_\ell$ be the roots of P in $\mathbb{F}_{p^\alpha}$. Then observe that, since P is non-constant, it has at most $\deg(P)$ roots in $\mathbb{F}_{p^\alpha}$ (i.e., $\ell \leq \deg(P)$), which means that

$$\Pr_{x_1 \twoheadleftarrow \mathcal{S}_1}[P(x_1) = 0] \leq \sum_{i=1}^{\ell} \Pr[\mathcal{S}_1 = r_i] = \sum_{i=1}^{\ell} \mathrm{Adv}^{\mathrm{pred}}_{\mathcal{S}_1,\mathcal{P}[r_i]} \leq \deg(P) \cdot k_1 = \deg(P) \cdot k_P\,,$$

where $\mathcal{P}[r_i]$ is the predictor that simply returns r_i. Now assume that the statement holds for all non-constant polynomials in $m \geq 1$ variables, and consider a non-constant polynomial $P \in \mathbb{F}_{p^\alpha}[X_1, \ldots, X_{m+1}]$. We distinguish two cases: If $m + 1 \notin T_P$, then $P \in \mathbb{F}_{p^\alpha}[X_1, \ldots, X_m] \subseteq \mathbb{F}_{p^\alpha}[X_1, \ldots, X_{m+1}]$, and the result follows from the inductive hypothesis because

$$\Pr_{x_1 \twoheadleftarrow \mathcal{S}_1, \ldots, x_{m+1} \twoheadleftarrow \mathcal{S}_{m+1}}[P(x_1, \ldots, x_{m+1}) = 0] = \Pr_{x_1 \twoheadleftarrow \mathcal{S}_1, \ldots, x_m \twoheadleftarrow \mathcal{S}_m}[P(x_1, \ldots, x_m) = 0] \leq \deg(P) \cdot k_P\,.$$

Now assume that $m + 1 \in T_P$. Consider the decomposition

$$P(X_1, \ldots, X_{m+1}) = \sum_{i=0}^{\deg(P)} X_{m+1}^i \cdot P_i(X_1, \ldots, X_m)\,,$$

and let $0 \leq j \leq \deg(P)$ be the maximal index i such that $P_i(X_1, \ldots, X_m) \neq 0$. Observe that P_j has total degree $\deg(P) - j$. By mutual independence of the distributions $\mathcal{S}_1, \ldots, \mathcal{S}_{m+1}$ we have

$$\begin{aligned}
&\Pr_{x_1 \twoheadleftarrow \mathcal{S}_1, \ldots, x_{m+1} \twoheadleftarrow \mathcal{S}_{m+1}}[P(x_1, \ldots, x_{m+1}) = 0] \\
&\quad= \sum_{(x_1,\ldots,x_m)} \Pr_{x_{m+1} \twoheadleftarrow \mathcal{S}_{m+1}}[P(x_1, \ldots, x_m, x_{m+1}) = 0] \cdot \Pr\left[\begin{matrix}(\mathcal{S}_1 = x_1) \wedge \cdots \\ \cdots \wedge (\mathcal{S}_m = x_m)\end{matrix}\right] \\
&\quad= \sum_{\substack{(x_1,\ldots,x_m),\\ P_j(x_1,\ldots,x_m)=0}} \Pr_{x_{m+1} \twoheadleftarrow \mathcal{S}_{m+1}}[P(x_1, \ldots, x_m, x_{m+1}) = 0] \cdot \Pr\left[\begin{matrix}(\mathcal{S}_1 = x_1) \wedge \cdots \\ \cdots \wedge (\mathcal{S}_m = x_m)\end{matrix}\right] \\
&\qquad+ \sum_{\substack{(x_1,\ldots,x_m),\\ P_j(x_1,\ldots,x_m)\neq 0}} \Pr_{x_{m+1} \twoheadleftarrow \mathcal{S}_{m+1}}[P(x_1, \ldots, x_m, x_{m+1}) = 0] \cdot \Pr\left[\begin{matrix}(\mathcal{S}_1 = x_1) \wedge \cdots \\ \cdots \wedge (\mathcal{S}_m = x_m)\end{matrix}\right],
\end{aligned}$$

Game $\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}}$:	Source $\mathcal{Z}_i(\pi)$:	Game $\mathrm{Pred}^{\mathcal{A},\mathcal{P}}_{\mathcal{S}}$:
$x \twoheadleftarrow \mathcal{S}$; $x' \twoheadleftarrow \mathcal{P}$	$(x,z) \twoheadleftarrow \mathcal{S}_i(\pi)$	$(\pi := (p^\alpha, m, \mathsf{st})) \leftarrow \mathcal{A}_0$; $(x,z) \twoheadleftarrow \mathcal{S}(\pi)$
return $(x = x')$	return z	$x' \twoheadleftarrow \mathcal{P}(\pi, z)$; return $(x = x')$

Figure 33 — *Left:* The prediction game from Lemma A.1. *Center:* Definition of the source $\mathcal{Z}_i(\pi)$. *Right:* The prediction game from Theorem A.2.

where the sum extends over all tuples $(x_1, \ldots, x_m) \in \mathbb{F}_{p^\alpha}^m$ such that the probabilities $\Pr[\mathcal{S}_1 = x_1], \ldots, \Pr[\mathcal{S}_m = x_m]$ are non-zero. We now bound the first term in the first sum with 1. Furthermore, observe that if $P_j(x_1, \ldots, x_m) \neq 0$, then $P(x_1, \ldots, x_m, X_{m+1})$ is a polynomial in one variable of degree j, so that the first term in the second sum can be bounded with jk_{m+1} by the base case. Continuing the above chain of inequalities, this yields

$$\begin{aligned}
&\leq \sum_{\substack{(x_1,\ldots,x_m),\\ P_j(x_1,\ldots,x_m)=0}} \Pr\left[\begin{array}{l}(\mathcal{S}_1 = x_1) \wedge \cdots \\ \quad \cdots \wedge (\mathcal{S}_m = x_m)\end{array}\right] + \sum_{\substack{(x_1,\ldots,x_m),\\ P_j(x_1,\ldots,x_m)\neq 0}} jk_{m+1} \cdot \Pr\left[\begin{array}{l}(\mathcal{S}_1 = x_1) \wedge \cdots \\ \quad \cdots \wedge (\mathcal{S}_m = x_m)\end{array}\right] \\
&\leq \Pr_{x_i \twoheadleftarrow \mathcal{S}_i}[P_j(x_1, \ldots, x_m) = 0] + j \cdot k_{m+1} \leq \deg(P_j) \cdot k_{P_j} + j \cdot k_{m+1} \\
&= (\deg(P) - j)k_{P_j} + j \cdot k_{m+1} \leq (\deg(P) - j)k_P + j \cdot k_P = \deg(P) \cdot k_P \,.
\end{aligned}$$

Here, the third inequality holds by the inductive hypothesis. □

Theorem A.2 (General game-based Schwartz–Zippel)**.** *Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be a two-stage algorithm, where $\mathcal{A}_0$ takes no input and returns a set of public parameters $\pi := (p^\alpha, m, \mathsf{st}) \in \mathbb{N}^2 \times \{0,1\}^*$ with $\alpha, p \in \mathbb{N}$ and p prime, and $\mathcal{A}_1$ takes π and values $z_1, \ldots, z_m \in \{0,1\}^*$ as input and returns a non-constant polynomial $P \in \mathbb{F}_{p^\alpha}[X_1, \ldots, X_m]$ and a value $y \in \mathbb{F}_{p^\alpha}$. Consider a family of sources $\mathcal{S} := \{\mathcal{S}_i \mid i \in \mathbb{N}\}$, each taking π as input and returning values $(x,z) \in \mathbb{F}_{p^\alpha} \times \{0,1\}^*$, and let $\mathcal{Z}_i(\pi)$ be the projection of $\mathcal{S}_i(\pi)$ onto the second coordinate, as shown in Figure 33 (center). Assume that, for every $i \in \mathbb{N}$, every π and z such that $\Pr[\mathcal{A}_0 = \pi] > 0$ and $\Pr[\mathcal{Z}_i(\pi) = z] > 0$, and every (possibly unbounded) predictor $\mathcal{P}$,*

$$\Pr\left[\mathrm{Pred}^{\mathcal{A},\mathcal{P}}_{\mathcal{S}_i} \;\middle|\; (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i(\pi) = z)\right] \leq k_i(\pi, z) \in \mathbb{R}\,,$$

where the game $\mathrm{Pred}^{\mathcal{A},\mathcal{P}}_{\mathcal{S}}$ is defined in Figure 33 (right). Then

$$\Pr\left[\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}\right] \leq \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \mathcal{A}_0,\\ (x_1,z_1) \twoheadleftarrow \mathcal{S}_1(\pi), \ldots, (x_m,z_m) \twoheadleftarrow \mathcal{S}_m(\pi),\\ (P,y) \twoheadleftarrow \mathcal{A}_1(\pi, z_1, \ldots, z_m)}} \left[\deg(P) \cdot k_P(\pi, z_1, \ldots, z_m)\right],$$

where the game $\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}$ is defined in Figure 3 (left), $k_P(\pi, z_1, \ldots, z_m) := \max_{i \in T_P} k_i(\pi, z_i)$, and T_P denotes the set $T_P := \{i \in [m] \mid \deg_{X_i}(P) > 0\}$.

Proof. For every $i \in \mathbb{N}$, every $\pi \in \mathbb{N}^2 \times \{0,1\}^*$ with $\Pr[\mathcal{A}_0 = \pi] > 0$, and every $z \in \{0,1\}^*$ with $\Pr[\mathcal{Z}_i(\pi) = z] > 0$, consider a source $\mathcal{X}_i(\pi, z)$ which follows the conditional distribution of $\mathcal{S}_i(\pi)$ given $\mathcal{Z}_i(\pi) = z$, i.e. verifies

$$\Pr[\mathcal{X}_i(\pi, z) = x] = \Pr[\mathcal{S}_i(\pi) = (x, z) \mid \mathcal{Z}_i(\pi) = z]$$

Game $\mathrm{SZ2}^{\mathcal{A}}_{\mathcal{Z},\mathcal{X}}$:	Game $\mathrm{SZ3}^{\mathcal{A}}_{\mathcal{Z},\mathcal{X}}$:
$(\pi \coloneqq (p^\alpha, m, \mathsf{st})) \twoheadleftarrow \mathcal{A}_0$	$(\pi \coloneqq (p^\alpha, m, \mathsf{st})) \twoheadleftarrow \mathcal{A}_0$
for $i \in [m]$ do	for $i \in [m]$ do $\mathbf{z}[i] \twoheadleftarrow \mathcal{Z}_i(\pi)$
$\quad \mathbf{z}[i] \twoheadleftarrow \mathcal{Z}_i(\pi)$; $\mathbf{x}[i] \twoheadleftarrow \mathcal{X}_i(\pi, z_i)$	$(P, y) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{z})$
$(P, y) \twoheadleftarrow \mathcal{A}_1(\pi, \mathbf{z})$	for $i \in [m]$ do $\mathbf{x}[i] \twoheadleftarrow \mathcal{X}_i(\pi, z_i)$
return $(y = P(\mathbf{x}))$	return $(y = P(\mathbf{x}))$

Figure 34 — The modified versions of the Schwartz–Zippel game.

for every $x \in \mathbb{F}_{p^\alpha}$. Then observe that, with these definitions, we can rewrite the game $\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}$ as shown in Figure 34, ignoring outcomes with probability zero. By the law of total probability,

$$\begin{aligned}
\Pr\left[\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}\right] &= \Pr\left[\mathrm{SZ2}^{\mathcal{A}}_{\mathcal{Z},\mathcal{X}}\right] = \Pr\left[\mathrm{SZ3}^{\mathcal{A}}_{\mathcal{Z},\mathcal{X}}\right] \\
&= \sum_{\pi} \sum_{\mathbf{z}} \sum_{(P,y)} \Pr\left[\mathrm{SZ3}^{\mathcal{A}}_{\mathcal{Z},\mathcal{X}} \;\middle|\; \begin{array}{c}(\mathcal{A}_0 = \pi) \wedge \bigwedge(\mathcal{Z}_i(\pi) = z_i) \\ \wedge\, (\mathcal{A}_1(\pi, \mathbf{z}) = (P, y))\end{array}\right] \Pr\left[\begin{array}{c}(\mathcal{A}_0 = \pi) \wedge \bigwedge(\mathcal{Z}_i(\pi) = z_i) \\ \wedge\, (\mathcal{A}_1(\pi, \mathbf{z}) = (P, y))\end{array}\right] \\
&\leq \sum_{\pi} \sum_{\mathbf{z}} \sum_{(P,y)} \deg(P) \cdot k_P(\pi, z_1, \ldots, z_m) \cdot \Pr\left[\begin{array}{c}(\mathcal{A}_0 = \pi) \wedge \bigwedge(\mathcal{Z}_i(\pi) = z_i) \\ \wedge\, (\mathcal{A}_1(\pi, \mathbf{z}) = (P, y))\end{array}\right] \\
&= \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \mathcal{A}_0, \\ (x_1,z_1) \twoheadleftarrow \mathcal{S}_1(\pi), \ldots, (x_m,z_m) \twoheadleftarrow \mathcal{S}_m(\pi), \\ (P,y) \twoheadleftarrow \mathcal{A}_1(\pi, z_1, \ldots, z_m)}} \left[\deg(P) \cdot k_P(\pi, z_1, \ldots, z_m)\right].
\end{aligned}$$

Here, the sums range over all π with $\Pr[\mathcal{A}_0 = \pi] > 0$, for every such π over all $\mathbf{z}$ with $\Pr[\mathcal{Z}_i(\pi) = z_i] > 0$ for every $i \in [m]$, and for every such π and $\mathbf{z}$ over all (P, y) with $\Pr[\mathcal{A}_1(\pi, \mathbf{z}) = (P, y)] > 0$. Furthermore, the inequality follows from Lemma A.1. □

The game-based Schwartz–Zippel lemma is now an immediate consequence of Theorem A.2.

Proof (of Lemma 2.2). We start by observing that, for every $i \in \mathbb{N}$ and every predictor $\mathcal{P}$,

$$\begin{aligned}
\Pr\left[\mathrm{Pred}^{\mathcal{A},\mathcal{P}}_{\mathcal{S}_i} \;\middle|\; (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i(\pi) = z)\right] &\leq \max_x \Pr[\mathcal{X}_i = x \mid (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i(\pi) = z)] \\
&= \frac{1}{2^{\mathbf{H}_\infty(\mathcal{X}_i \mid (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i(\pi) = z))}} = \frac{1}{2^{\mathbf{H}_\infty(\mathcal{X}_i \mid (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i = z))}}.
\end{aligned}$$

Using Theorem A.2 and monotonicity of the expected value we then obtain

$$\begin{aligned}
\Pr\left[\mathrm{SZ}^{\mathcal{A}}_{\mathcal{S}}\right] &\leq \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \mathcal{A}_0, \\ (x_1,z_1) \twoheadleftarrow \mathcal{S}_1(\pi), \ldots, (x_m,z_m) \twoheadleftarrow \mathcal{S}_m(\pi), \\ (P,y) \twoheadleftarrow \mathcal{A}_1(\pi, z_1, \ldots, z_m)}} \left[\deg(P) \cdot \max_{i \in T_P} \frac{1}{2^{\mathbf{H}_\infty(\mathcal{X}_i \mid (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i = z))}}\right] \\
&\leq d \cdot \mathop{\mathbb{E}}_{\substack{\pi \twoheadleftarrow \mathcal{A}_0, \\ (x_1,z_1) \twoheadleftarrow \mathcal{S}_1(\pi), \ldots, (x_m,z_m) \twoheadleftarrow \mathcal{S}_m(\pi)}} \left[\frac{1}{2^{\min_{i \in [m]} \mathbf{H}_\infty(\mathcal{X}_i \mid (\mathcal{A}_0 = \pi) \wedge (\mathcal{Z}_i = z))}}\right].
\end{aligned}$$

This concludes the proof. □

We can also use Lemma A.1 to give a direct proof of the classical Schwartz–Zippel lemma.

Proof (of Lemma 2.1). If the polynomial P is a non-zero constant, the result is obviously true, so assume that P is non-constant. Consider, for every $i \in [m]$, a source $\mathcal{S}_i$ which returns a uniformly random element from S. Then we can choose $k_i = 1/|S|$ for every $i \in [m]$, which means that also $k_P = 1/|S|$. Thus

$$\Pr_{x_1,\ldots,x_m \twoheadleftarrow S}[P(x_1,\ldots,x_m) = 0] = \Pr_{x_1 \twoheadleftarrow \mathcal{S}_1,\ldots,x_m \twoheadleftarrow \mathcal{S}_m}[P(x_1,\ldots,x_m) = 0] \leq \deg(P) \cdot k_P = \frac{\deg(P)}{|S|}$$

by Lemma A.1, and the result follows. □