Cryptographic Primitives with Hinting Property

Navid Alamati[*] Sikhar Patranabis[†]

Abstract

A *hinting* pseudorandom generator (PRG) is a potentially stronger variant of PRG with a "deterministic" form of circular security with respect to the seed of the PRG (Koppula and Waters, CRYPTO 2019). Hinting PRGs enable many cryptographic applications, most notably CCA-secure public-key encryption and trapdoor functions. In this paper, we study cryptographic primitives with the hinting property, yielding the following results:

- We present a novel and conceptually simpler approach for designing hinting PRGs from certain decisional assumptions over cyclic groups or isogeny-based group actions, which enables simpler security proofs as compared to the existing approaches for designing such primitives.

- We introduce *hinting* weak pseudorandom functions (wPRFs), a natural extension of the hinting property to wPRFs, and show how to realize circular/KDM-secure symmetric-key encryption from any hinting wPRF. We demonstrate that our simple approach for building hinting PRGs can be extended to realize hinting wPRFs from the same set of decisional assumptions.

- We propose a stronger version of the hinting property, which we call the *functional* hinting property, that guarantees security even in the presence of hints about functions of the secret seed/key. We show how to instantiate functional hinting PRGs/wPRFs for certain (families of) functions by building upon our simple techniques for realizing plain hinting PRGs/wPRFs. We also demonstrate the applicability of a functional hinting wPRF with certain algebraic properties in realizing KDM-secure public-key encryption in a black-box manner.

- We show the first black-box separation between hinting wPRFs (and hinting PRGs) from public-key encryption using simple realizations of these primitives given only a random oracle.

[*] VISA Research.
[†] IBM Research India.

1 Introduction

A pseudorandom generator (PRG) is one of the fundamental and widely studied cryptographic primitives. Informally speaking, a PRG is an expanding function with the security guarantee that the output of the PRG on a randomly chosen input (also called the "seed") is computationally indistinguishable from random. However, a plain PRG does not provide any security guarantees if the adversary has some additional "hint" with respect to the each bit of the seed.

A *hinting* PRG, introduced recently by Koppula and Waters in [KW19], is a (potentially) stronger variant of PRG that provides security even given some hinting information about each bit of the seed. This hinting property can be viewed as a "deterministic" form of circular security with respect to the seed of the PRG. We informally recall the definition of a hinting PRG to provide a more concrete view of what this hinting property actually entails, and how it encapsulates circular security with respect to the seed.

A hinting PRG is a PRG of the form $G : \{0,1\}^n \to Y^n$ that expands n-bit seed $\mathbf{s} \in \{0,1\}^n$ into a vector $\mathbf{y} = (y_1, \ldots, y_n)$ of n elements from the set Y, such that an $n \times 2$ matrix $\mathbf{Z} = \{z_{i,b}\}_{i\in[n],b\in\{0,1\}}$ distributed as follows:

$$z_{i,b} = \begin{cases} y_i & \text{if } b = s_i, \\ u_i \leftarrow Y & \text{otherwise,} \end{cases}$$

is computationally indistinguishable from a truly random matrix $\mathbf{U} \leftarrow Y^{n\times 2}$, where each element is sampled uniformly from the set Y.[1] Note that the matrix $\mathbf{Z}$ not only contains the output of the PRG, but also has some hinting information about each bit s_i of the seed $\mathbf{s}$ encoded into the arrangement of the elements in each row.

Hinting PRGs have been used as a key ingredient to construct several cryptographic primitives, such as realizing CCA-secure public-key encryption (PKE) and attribute-based encryption from their CPA-secure counterparts [KW19], trapdoor functions [KMT19a, GHMO21], black-box non-interactive non-malleable commitments [GKLW21], and CCA-compatible public-key infrastructure [KW21]. This wide range of applications motivates: (i) building hinting PRGs from a variety of mathematical assumptions, (ii) investigating some natural extensions of the hinting property to other cryptographic primitives, and (iii) studying the complexity of cryptographic primitives with hinting property.

Instantiations of Hinting PRGs. Koppula and Waters [KW19] showed how to realize hinting PRGs from the computational Diffie-Hellman (CDH) and the learning with errors (LWE) assumptions. Their constructions are based on the "missing block" framework that was introduced by Choi *et al.* [CDG$^+$17]. Later, Goyal *et al.* [GVW20] introduced a new accumulation-style framework to build hinting PRGs, and they showed (efficient) constructions of hinting PRGs from the Decisional Diffie-Hellman Inversion (DDHI) and Phi-hiding assumptions. However, despite such considerable progress, it is not known how to realize hinting PRGs from a notable class of plausibly post-quantum secure assumptions, namely isogeny-based assumptions. Note that current techniques to construct hinting PRGs either use groups with infeasible inversion or the missing-block framework, both of which seem to be out of reach based on our understanding of structural properties of isogeny-based assumptions [ADMP20]. This leads to the following question: *can we realize hinting PRGs from isogeny-based assumptions*?

On a related note, a hinting PRG is an ostensibly symmetric-key primitive, and one would expect to achieve it from decisional assumptions (such as the DDH assumption) in a considerably simpler manner than allowed by current constructions and their security proofs. In particular, the closely related notion of symmetric-key circular secure encryption [BRS03] has significantly simpler realizations and security proofs based on decisional assumptions such as the DDH assumption [BHHO08]. This leads to the question: *is there a simple construction of hinting PRGs from decisional assumptions such as DDH*? More concretely, our aim is to achieve constructions and security proofs for hinting PRGs that are simpler than those based on the missing block framework [KW19] or the accumulation framework [GVW20]. Our hope is that a simpler construction of hinting PRGs would be amenable to instantiations from decisonal isogeny-based assumptions, while also naturally enabling extensions of the hinting property to other cryptographic primitives.

[1] The original definition of hinting PRG in [KW19] uses an additional output element $z_0 \in Y$ which has no hint about the seed of the PRG. We omit this element from the definition of hinting PRG here for simplicity of exposition.

Hinting Property for Other Primitives. The authors of [KMT19a] showed that a hinting PRG can be used to build a *one-time* key-dependent message (KDM) secure symmetric-key encryption (SKE) scheme. This motivates us to ask if there exists a natural extension of hinting PRGs that implies circular/KDM security with respect to *many* encryptions of the secret key, and if so, can such an extension also be realized in a simple manner from decisional assumptions such as DDH or isogeny-based decisional assumptions. Concretely, we ask the following question: *can we instantiate natural extensions of the hinting property to other cryptographic primitives from concrete hardness assumptions?*

Functional Hinting Property. The original definition of hinting PRG, as introduced in [KW19], only considers security in the presence of hints about each bit of the PRG seed itself. A natural extension of this security property would be to guarantee PRG security in the presence of hints about each bit of *some function* of the seed. For example, for a PRG seed $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$, what if the PRG output provides hints about each bit of $f(\mathbf{s}) = (s_i \cdot s_j)_{i,j\in[n]}$, which is an n^2-length vector? This might be particularly challenging to achieve because the adversary now not only gets hints about each bit of $\mathbf{s}$ (via $s_i \cdot s_i = s_i$), but also about the pair-wise product of each bit of $\mathbf{s}$.

This strengthening of the hinting property to its functional counterpart is analogous to the strengthening of circular security to KDM security; in fact, one can view the functional hinting property with respect to a class of functions $\mathcal{F}$ as a "deterministic" form of KDM security with respect to $\mathcal{F}$. Additionally, this property also generalizes to other cryptographic primitives with the hinting property, if such primitives exist. In this paper, we ask the following question: *can we instantiate functional hinting PRG (and natural extensions of the functional hinting property to other cryptographic primitives) in a black-box way from concrete hardness assumptions?*

Complexity of Primitives with Hinting Property. Another natural direction is to investigate the complexity of a hinting PRG, and its extensions to other cryptographic primitives. Based on the current constructions of hinting PRGs, it is unclear if we necessarily need structured mathematical assumptions to realize hinting PRGs. It is seemingly hard to build a hinting PRG in a generic way from any PRG (or equivalently, any one-way function). On the other hand, a hinting PRG does not immediately entail any "public-key"-style functionalities, and we do not know if it implies PKE. This leads to the following question: *does a hinting PRG (or any of its extensions to other symmetric-key cryptographic primitives) imply PKE in a black-box way?*

Observe that the closely related notion of symmetric-key circular/KDM-secure encryption, in fact, *does not* imply PKE in a black-box way because it can be realized from a random oracle [BRS03]. However, this does not answer the question outlined above because, as the authors of [KMT19a] point out, it is not known if a hinting PRG can be realized from any symmetric-key circular secure encryption scheme in a black-box way.

1.1 Our Contributions

In this paper, we address all of the above questions by showing the following results.

Simpler Constructions of Hinting PRG from DDH or Isogenies. We propose a new approach for realizing hinting PRGs from decisional assumptions. Our approach yields significantly simpler constructions and security proofs for hinting PRGs as compared to the existing constructions and proofs based on the missing block framework [KW19] or the accumulation-style framework [GVW20]. We show how to instantiate our approach based on the DDH assumption, as well as from a recent plausibly post-quantum secure isogeny-based assumption called the linear hidden shift (LHS) assumption [ADMP20] over certain isogeny-based group actions (e.g., variants of CSIDH [CLM+18, BKV19, ADMP20]). To the best of our knowledge, prior to our work, it was not known how to securely realize a hinting PRG from any isogeny-based assumption, including the LHS assumption [ADMP20].

Building upon our technique to realize hinting PRGs from the LHS assumption, we also show a direct construction of trapdoor (one-way) functions (TDFs) from any weak pseudorandom group action (which is a plausibly post-quantum secure analogue of the DDH assumption over isogeny-based group actions, introduced in [ADMP20]) for which the LHS assumption holds. Our construction of TDFs and the corresponding proof of security are significantly simpler as compared to the previously known constructions of TDFs from such isogeny-based assumptions proposed in [ADMP20], which relied on the framework of [KMT19a]. We note that the authors of [GHMO21] proposed a construction of

TDFs given any hinting PRG and a PKE scheme with pseudorandom ciphertexts; however, their construction needs the ciphertext space to be a group, which does not hold for any isogeny-based PKE scheme.

Hinting weak PRF. We introduce a natural extension of the hinting property to another symmetric-key primitive, namely a weak pseudorandom function (wPRF). We call the resulting primitive a hinting wPRF, which is a strengthening of a hinting PRG in the sense that it guarantees weak pseudorandomness even in the presence of multiple hints with respect to the key of a wPRF. We show that a hinting wPRF can be used to construct a symmetric-key circular-secure encryption scheme (where the circular security guarantee holds with respect to multiple encryptions of the secret key) in a black-box manner (this can be amplified to achieve KDM security, albeit in a non-black-box way using known techniques [App14]). We also show that our approach for constructing hinting PRGs can be leveraged to construct hinting wPRFs. This yields simple constructions of hinting wPRFs based on either DDH or the LHS assumption.

Functional Hinting PRG/wPRF and Implications. We introduce functional hinting PRG - a strengthening of hinting PRG that guarantees PRG security in the presence of hints about each bit of some *function* of the seed. We also introduce a natural extension, namely a functional hinting wPRF, that guarantees wPRF security in the presence of hints about each bit of some (adversarially chosen) function of the secret key. We show that a functional hinting wPRF with respect to a family of functions $\mathcal{F}$ can be used to realize a symmetric-key KDM-secure encryption scheme with respect to the same function family $\mathcal{F}$ in a *black-box* manner. We then build upon our approach of realizing hinting PRGs and hinting wPRFs to realize simple constructions of functional hinting PRGs and functional hinting wPRFs for a family of quadratic functions (and functions of higher degree) based on the DDH assumption. We note that our techniques enable achieving a deterministic form of KDM-security in a black-box manner, which is a different approach as compared to prior works on KDM security [KM19, KMT19b, KM20].

Complexity of Hinting PRG/wPRF. We make progress on understanding the complexity of cryptographic primitives with the hinting property. We show the first black-box separation between hinting PRG and PKE by realizing a hinting PRG given only a random oracle. We then build upon our construction of hinting PRG to also show how to construct a hinting wPRF given only a random oracle. This additionally rules out the possibility of constructing PKE in a black-box manner from any hinting wPRF. In fact, our separation result holds even if we replaced a hinting wPRF with a hinting PRF – a strengthening of a hinting wPRF that satisfies plain/strong PRF security as opposed to weak PRF security in the presence of multiple hints with respect to the secret key.

1.2 Technical Overview

In this section, we provide an overview of our techniques. For simplicity of exposition, we focus primarily on two of our basic results – our construction of hinting PRG from DDH, and our construction of functional hinting PRG from DDH for the quadratic function $f(\mathbf{s} \in \{0,1\}^n) = \mathbf{s} \otimes \mathbf{s} \in \{0,1\}^{n^2}$. For all of our other results, we provide some high-level intuition while referring to the relevant sections in the body of the paper for details.

Hinting PRG from DDH. Let $(\mathbb{G}, g, q)$ be a DDH-hard group of prime order q with generator g. Throughout this paper, we use the notation $[\mathbf{M}]$ to denote $g^{\mathbf{M}}$ (exponentiation being applied componentwise) for any matrix $\mathbf{M} \in \mathbb{Z}_q^{m \times n}$. It was shown in [PW08, FGK$^+$10, AMP19] that for a uniformly sampled matrix $\mathbf{M} \leftarrow \mathbb{Z}_q^{n \times n}$ and a uniformly sampled binary vector $\mathbf{s} \leftarrow \{0,1\}^n$ where n is sufficiently large, we have

$$([\mathbf{M}], [\mathbf{Ms}]) \overset{c}{\approx} ([\mathbf{M}], [\mathbf{u}]) \,, \qquad (*)$$

where $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Observe that this naturally yields a PRG with public parameter $[\mathbf{M}]$ and seed $\mathbf{s}$ defined as

$$G_{[\mathbf{M}]}(\mathbf{s}) = [\mathbf{Ms}].$$

We now argue that this PRG already satisfies the hinting property. At a high level, our approach is as follows: we reduce the hinting property of G to the pseudorandomness of G, which in turn relies on the DDH assumption. We explain this in more details below.

Suppose we are given a PRG challenge of the form $([\mathbf{M}], [\mathbf{y}])$, where the vector $[\mathbf{y}]$ is either the "real" output of the PRG G, i.e., we have $[\mathbf{y}] = [\mathbf{M}\mathbf{s}]$ for some $\mathbf{s} \leftarrow \{0,1\}^n$, or $[\mathbf{y}]$ is uniformly random, i.e., we have $[\mathbf{y}] \leftarrow \mathbb{G}^n$. We construct a PPT algorithm $\mathcal{B}$ as follows: $\mathcal{B}$ takes as input a PRG challenge of the form $([\mathbf{M}], [\mathbf{y}])$ and outputs $([\mathbf{M}'], [\mathbf{Z}])$ where the matrix $[\mathbf{M}']$ is a uniformly distributed matrix in $\mathbb{G}^{n \times n}$, and $[\mathbf{Z}]$ is an $n \times 2$ matrix of group elements of the form $[\mathbf{Z}] = ([z_{i,b}])_{i \in [n], b \in \{0,1\}}$ such that:

- When $[\mathbf{y}]$ is distributed as the "real" output of the PRG G, $[\mathbf{Z}]$ is distributed as in the "real" hinting PRG game with respect to the public parameter $[\mathbf{M}']$.
- On the other hand, when $[\mathbf{y}]$ is uniformly random in $\mathbb{G}^n$, $[\mathbf{Z}]$ is distributed uniformly randomly over $\mathbb{G}^{n \times 2}$.

The main challenge here is that $\mathcal{B}$ needs to produce this output without any knowledge of the seed $\mathbf{s}$ of the PRG G. To do this, given a PRG challenge of the form $([\mathbf{M}], [\mathbf{y}])$, $\mathcal{B}$ "shifts" each diagonal entry $m_{i,i}$ of the matrix $[\mathbf{M}]$ by a random value $d_i \leftarrow \mathbb{Z}_q$ in the exponent of g, i.e., it computes the shifted diagonal element in the exponent as

$$[m'_{i,i}] = [m_{i,i}] + [d_i].$$

Let $[\mathbf{M}']$ be the corresponding matrix in $\mathbb{G}^{n \times n}$ with the shifted diagonal elements ($[\mathbf{M}']$ is identical to $[\mathbf{M}]$ in all non-diagonal entries), and define the matrix $[\mathbf{Z}] = ([z_{i,b}])_{i \in [n], b \in \{0,1\}}$ as follows: for each $i \in [n]$ and $b \in \{0,1\}$, set

$$[z_{i,b}] := \begin{cases} [y_i] & \text{if } b = 0, \\ [y_i + d_i] & \text{if } b = 1. \end{cases}$$

Suppose that $[\mathbf{y}] = [\mathbf{M}\mathbf{s}]$, and let $[\mathbf{y}'] = [\mathbf{M}'\mathbf{s}]$. If $s_i = 0$, we have

$$[z_{i,0}] = [y_i] = [y'_i], \quad [z_{i,1}] = [y'_i + d_i],$$

where the latter is uniformly random. Likewise, if $s_i = 1$, we have

$$[z_{i,1}] = [y_i + d_i] = [y'_i], \quad [z_{i,0}] = [y'_i - d_i],$$

where the latter is again uniformly random. Hence, $[\mathbf{Z}]$ is distributed as in the real hinting PRG game with respect to the public parameter $[\mathbf{M}']$, as desired. On the other hand, when $[\mathbf{y}]$ is uniformly random, so is $[\mathbf{Z}]$. We refer to Section 3.1 for a more formal description of our construction and proof.

Translation to Isogeny-based Group Actions. In the above security proof, the crux of the argument is in introducing a "shift" both in the public parameter $[\mathbf{M}]$ and in the challenge vector $[\mathbf{y}]$ when constructing $([\mathbf{M}'], [\mathbf{Z}])$, without having to solve discrete logs in the group $\mathbb{G}$. It turns out that for certain isogeny-based *effective* group actions (e.g., variants of CSIDH [CLM$^+$18, BKV19, ADMP20]), we can introduce such a "shift" using the algebraic properties of group actions without having to solve a computationally hard problem analogous to discrete log over group actions. This observation allows us to translate our technique for hinting PRGs outlined above from DDH-hard groups to group actions satisfying the LHS assumption introduced in [ADMP20]. We refer to Section 3.2 for a more formal description. In addition, we can extend this technique of publicly computable shifts in the outputs of group action computations to achieve a direct construction of TDFs from any LHS-hard weak pseudorandom effective group action. We refer to Section 3.3 for the detailed construction and proof. We point out that our construction avoids the many layers of generic transformation required by the prior construction of TDFs from such isogeny-based assumption, proposed in [ADMP20] based on the framework of [KMT19a].

Comparison with Prior Works. Our approach for realizing hinting PRGs from DDH-hard groups or LHS-hard effective group actions yields significantly simpler constructions and security proofs as compared to prior constructions and proofs for hinting PRGs based on the missing block framework [KW19] or the accumulation framework [GVW20]. Specifically, the authors of [GVW20] needed to prove a new hashing lemma, which is crucial to their proof of security, besides relying on the DDHI assumption, which is a seemingly stronger assumption as compared to DDH. Similarly, the authors of [KW19] propose a construction of hinting PRGs such that proving the hinting property itself requires multiple hybrids, where one of the intermediate hybrids relies on a statistical hashing lemma. On the other hand, in our construction, we directly reduce the hinting property of the PRG to its own pseudorandomness. We also note that neither the missing block framework of [KW19] nor the accumulation framework of [GVW20] seems amenable to realizations from isogeny-based assumptions; in particular, their techniques seem incompatible with the algebraic properties of isogeny-based group actions, especially given the long history of failed attempts to integrate standard hashing techniques into the framework of isogeny-based cryptography [BBD$^+$22]. However, our proposed technique readily extends to the setting of isogeny-based group actions, and enables the first realizations of hinting PRGs from (plausibly post-quantum secure) isogeny-based assumptions.

Hinting wPRF from DDH or LHS. For our hinting PRG construction, we used a simple proof technique that allows reducing the hinting property of the PRG to its own pseudorandomness. Observe that in this reduction, we rely on the fact that the adversary only sees a single evaluation of the hinting PRG with respect to a uniformly sampled seed. To realize hinting wPRF, we use an extension of this technique that allows similarly reducing the hinting property of the wPRF, albeit over *multiple* evaluations, to the weak pseudorandomness of the wPRF. We note that for prior approaches to constructing hinting PRGs (e.g., the construction of hinting PRGs from CDH [KW19]), such an extension to hinting wPRFs is seemingly hard to achieve. Our extension is designed to work with both DDH-hard groups as well as any LHS-hard weak pseudorandom effective group action; in particular, we preserve compatibility with the algebraic properties of group actions to enable our isogeny-based constructions of hinting wPRFs. We refer to Sections 4.1 and 4.2 for the detailed constructions and proofs of hinting wPRFs from DDH and LHS respectively, and to Section 5.3 for a simple construction of circular/KDM-secure SKE from any hinting wPRF.

Functional Hinting PRG from DDH. Our simple technique for realizing hinting PRGs from DDH is actually powerful enough to allow constructing functional hinting PRGs, which are strengthenings of hinting PRG that guarantee PRG security in the presence of hints about each bit of *some* function of the seed. For this overview, we show how to construct a functional hinting PRG from DDH, where the function f that we consider is defined as follows: given a seed $\mathbf{s} \in \{0,1\}^n$, $f(\mathbf{s}) = (s_i \cdot s_j)_{i,j \in [n]}$, which is an n^2-length vector.

The starting point of our functional hinting PRG from DDH is a stronger version of the indistinguishability $(*)$ from [PW08, FGK$^+$10, AMP19] that we prove in this paper based on the DDH assumption: for n^2 uniformly sampled matrices $\left\{\mathbf{M}_i \leftarrow \mathbb{Z}_q^{n\times n}\right\}_{i\in[n^2]}$ and a uniformly sampled vector $\mathbf{s} \leftarrow \{0,1\}^n$ (where n is sufficiently large), we have

$$\left([\mathbf{M}_i], [\mathbf{s}^t\mathbf{M}_i\mathbf{s}]\right)_{i\in[n^2]} \overset{c}{\approx} \left([\mathbf{M}_i], [u_i]\right)_{i\in[n^2]},$$

where each $u_i \leftarrow \mathbb{Z}_q$. Observe that this naturally yields a PRG with public parameter $([\mathbf{M}_1], \ldots, [\mathbf{M}_{n^2}])$ and seed $\mathbf{s}$ defined as

$$G_{([\mathbf{M}_1],\ldots,[\mathbf{M}_{n^2}])}(\mathbf{s}) = \left([\mathbf{s}^t\mathbf{M}_1\mathbf{s}], \ldots, [\mathbf{s}^t\mathbf{M}_{n^2}\mathbf{s}]\right).$$

Similar to our technique for proving the security of hinting PRG, even in this case, we can reduce the functional hinting PRG security of the above construction to its own pseudorandomness (which in turn relies on DDH) by introducing shifts on a suitable entry of each matrix $[\mathbf{M}_i]$ in the public parameter. We refer to Section 5.1 for the detailed construction and proof of security, and also for extensions of the above construction to achieve functional hinting PRGs with respect to functions of higher degree.

Functional Hinting wPRF and Applications. For our functional hinting PRG construction, we use a reduction where we rely on the fact that the adversary only sees a single evaluation of the hinting PRG with respect to a uniformly sampled seed, while only getting hints about each bit of a *single* function of the seed. Achieving a functional hinting wPRF is significantly more complicated, since not only can the adversary see multiple evaluations of the wPRF on uniformly random inputs, but also get hints about multiple functions of the secret key, where the function may be chosen adversarially from a fixed function family. In this paper, we show a construction of functional hinting wPRF from DDH with respect to the function family $\mathcal{F}$ consisting of (projective) quadratic functions (and functions of higher degree) over the bits of the key. We refer to Section 5.2 for the detailed construction and proof of functional hinting wPRFs from DDH.

In Section 5.4, we describe a simple construction of KDM-secure SKE with respect to a function family $\mathcal{F}$ from any functional hinting wPRF with respect to the same function family $\mathcal{F}$ in a *black-box* manner. We also show a strengthening of this result to obtain a construction of $\mathcal{F}$-KDM secure PKE scheme from any $\mathcal{F}$-functional hinting wPRF that additionally satisfies homomorphism between the input and output space – a property that is actually satisfied by our construction of functional hinting wPRF from DDH.

Note that the existing approaches for achieving KDM-secure PKE in a black-box way [BGK11, KMT19b] are somewhat incomparable to ours; in particular, these prior constructions are designed specifically for *arithmetic* function families that inherently require some form of algebraic structure on the secret key space, while the function family that we consider can be viewed as a certain form of boolean function family (e.g., in the case of quadratic functions, an adversary is provided with hints with respect to the conjunction/AND of each pair of bits of the secret key). Additionally, the primitive underlying our construction, namely functional hinting wPRF, provides a deterministic form of KDM security that has not been considered in prior works to the best of our knowledge. We remark that our construction of (functional) hinting wPRF from DDH/LHS essentially subsumes our construction of hinting PRG from DDH/LHS, while building upon our techniques for the latter construction. More generally, we chose to present our results in a progressive manner, where each result builds upon our techniques used to construct simpler primitives. We do this for ease of exposition, and also for highlighting the simplicity/modularity of our techniques.

Hinting (weak) PRF in the Random Oracle Model. Let $H : \{0,1\}^n \to Y^{n+1}$ be a truly random function (modeled as a random oracle), where Y is a sufficiently large set. It is easy to see that H is a PRG in the random oracle model since for any uniformly random input $\mathbf{s} \leftarrow \{0,1\}^n$, no (computationally unbounded) adversary can distinguish (with non-negligible probability) between $H(\mathbf{s} \leftarrow \{0,1\}^n)$ and $\mathbf{u} \leftarrow Y^{n+1}$ while issuing polynomially many queries to the function H. We show in Section 6 that this simple PRG in the random oracle model also satisfies the hinting property via a simple information-theoretic argument. This implies the first black-box separation between hinting PRG and PKE [IR89] to the best of our knowledge. We then build upon our construction of hinting PRG to also show how to construct a hinting PRF given only a random oracle. As mentioned earlier, a hinting PRF is a strengthening of a hinting wPRF that satisfies plain/strong PRF security as opposed to weak PRF security in the presence of multiple hints with respect to the secret key (i.e., the adversary is allowed to ask for hints with respect to the key of PRF for *arbitrarily* chosen inputs instead of randomly chosen ones). We refer to Section 6 for the detailed construction and proof. Our result also rules out the possibility of constructing PKE in a black-box way from any hinting (weak) PRF [IR89].

1.3 Outline

The rest of the paper is organized as follows. Section 2 presents preliminary background material. Section 3 presents our constructions of hinting PRGs from DDH or LHS. It also presents an extension of our techniques to realize TDFs from LHS-hard weak pseudorandom effective group actions. In Section 4, we define the notion of hinting weak PRF and we present our constructions of hinting weak PRFs from DDH or LHS. Section 5 presents our constructions of functional hinting PRGs and functional hinting weak PRFs with respect to a certain family of functions. It also presents the applications of functional hinting weak PRFs to circular/KDM security in the symmetric-key setting, and how structured functional hinting weak PRFs can be used to realize circular/KDM-secure PKE. Finally, Section 6 presents our constructions of primitives with hinting property in the random oracle model.

2 Preliminaries

Notations. For any positive integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. We may use $[a]$ to denote g^a where $a \in \mathbb{Z}_q$ and g is a generator of a cyclic group with order q. However, the difference between $[n]$ and $[a]$ will be clear from context.

We use the notation $\overset{s}{\approx}$ (respectively, $\overset{c}{\approx}$) to denote statistical (respectively, computational) indistinguishability. We denote the security parameter by λ. For a finite set S, we use $s \leftarrow S$ to sample uniformly from the set S.

Definition 2.1 (weak PRF). Let $F : K \times X \to Y$ be a function family, where each set is indexed by the security parameter. We say that F is a weak PRF if for any $Q = \mathrm{poly}(\lambda)$ it holds that

$$\big\{(x_i, F(k, x_i))\big\}_{i\in[Q]} \overset{c}{\approx} \big\{(x_i, y_i)\big\}_{i\in[Q]},$$

where $k \leftarrow K$, $x_i \leftarrow X$, and $y_i \leftarrow Y$.

Definition 2.2 (KDM-secure SKE). Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Enc})$ be a symmetric-key encryption (SKE) scheme with $\mathcal{M} = \{0,1\}^m$ and $\mathcal{K} = \{0,1\}^n$, where $\mathcal{M}$ and $\mathcal{K}$ denote the message space and the key space, respectively. Let $\mathcal{F} = \{f_I \mid f_I : \{0,1\}^n \to \{0,1\}^m\}_{I\in\mathcal{I}}$ be a family of boolean functions, and let $\bar{f} \in \mathcal{F}$ where $\bar{f}$ is the constant function $\bar{f}(\mathbf{x}) = 0^m$. We say that Π is KDM secure with respect to $\mathcal{F}$ if the advantage of any PPT adversary $\mathcal{A}$ in distinguishing the experiments $\mathsf{Exp}_0^{\mathsf{KDM}}$ and $\mathsf{Exp}_1^{\mathsf{KDM}}$ (defined in Figure 1) is negligible.

Note that KDM security for public-key encryption with respect to a function family $\mathcal{F}$ is defined similarly, except that the adversary is given public key in the beginning of the experiment.

1. The challenger samples a secret key key $\mathsf{sk} \leftarrow \{0,1\}^n$.
2. The adversary queries for a function input $f \in \mathcal{F}$.
3. If $b = 0$, the challenger responds with $\mathsf{Enc}(\mathsf{sk}, 0^m)$. Otherwise, it responds with $\mathsf{Enc}(\mathsf{sk}, f(\mathsf{sk}))$.
4. The adversary continues to make input queries as before, and each query is replied by the challenger as described above.
5. Finally, the adversary outputs a bit b'. The advantage of $\mathcal{A}$ is defined to be $\Pr[b = b']$ over all randomness in the experiment.

Figure 1: Experiment $\mathsf{Exp}_b^{\mathsf{KDM}}$

We recall the definition of hinting PRG [KW19]. We use a slightly different syntax compared to [KW19] for each block of the output of hinting PRG.[1]

Definition 2.3 (Hinting PRG). Let $n = \mathrm{poly}(\lambda)$ be an integer. Let $(\mathsf{Setup}, \mathsf{Eval})$ be a pair of algorithms such that

- $\mathsf{Setup}(1^\lambda)$ is a randomized algorithm that outputs some public parameter pp,
- $\mathsf{Eval}(\mathsf{pp}, \mathsf{s} \in \{0,1\}^n, i \in \{0\} \cup [n])$ is a deterministic algorithm that outputs (a representation of) some element y in Y, where Y is the codomain of the algorithm and $|Y| = \omega(\log \lambda)$.

We say that $(\mathsf{Setup}, \mathsf{Eval})$ defines a hinting PRG if for $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{s} \leftarrow \{0,1\}^n$ it holds that

$$(\mathsf{pp}, y_0, \mathbf{Y}) \overset{c}{\approx} (\mathsf{pp}, u_0, \mathbf{U}),$$

where these terms are distributed as

$$y_0 = \mathsf{Eval}(\mathsf{pp}, \mathsf{s}, 0), \quad y_{i,s_i} = \mathsf{Eval}(\mathsf{pp}, \mathsf{s}, i), \quad y_{i,1-s_i} \leftarrow Y, \quad u_0 \leftarrow Y, \quad \mathbf{U} \leftarrow Y^{n\times 2}.$$

[1] Specifically, the authors of [KW19] use the set $\{0,1\}^\ell$ for each block (where ℓ is fixed during the setup) whereas we use a sufficiently large (efficiently representable) set Y. Our definition allows defining hinting PRG in a setting where Y does not necessarily have a compact representation, i.e., when each element of Y is represented using more than $\log |Y|$ bits (which is the case for isogeny-based group actions). One can obtain a hinting PRG with bit-string blocks by using a suitable (statistical) extractor.

Definition 2.4 (DDH Assumption). Let $\mathbb{G}$ be a group of prime order q with generator g. We say that the DDH assumption holds over $\mathbb{G}$ if for $a \leftarrow \mathbb{Z}_q$, $b \leftarrow \mathbb{Z}_q$, $c \leftarrow \mathbb{Z}_q$ it holds that

$$(g, g^a, g^b, g^{ab}) \overset{c}{\approx} (g, g^a, g^b, g^c).$$

We will use the following special case of leftover hash lemma. We refer to [Reg09] for a proof.

Lemma 2.5. *Let G be an additively written abelian group such that $|G| = \lambda^{\omega(1)}$, and let $m > \log|G| + \omega(\log \lambda)$ be an integer. If $\mathbf{r} \leftarrow G^m$ and $\mathbf{s} \leftarrow \{0,1\}^m$, it holds that*

$$(\mathbf{r}, \sum_{i=1}^{m} s_i r_i) \overset{s}{\approx} (\mathbf{r}, u),$$

where $u \leftarrow G$ is a uniformly chosen group element.

Definition 2.6. An extractor $\mathsf{Ext} : \mathcal{S} \times X \to Y$ is a deterministic function with the seed space $\mathcal{S}$ and domain X such that if $\mathsf{seed} \leftarrow \mathcal{S}$ is sampled uniformly and x is sampled from a distribution over X with min-entropy λ^c (for some constant $0 < c < 1$), then it holds that

$$(\mathsf{seed}, \mathsf{Ext}(\mathsf{seed}, x)) \overset{s}{\approx} (\mathsf{seed}, y),$$

where $y \leftarrow Y$ is sampled uniformly.

2.1 Cryptographic Group Actions

We recall some definitions related to cryptographic group actions from [ADMP20], which provided a framework to build cryptographic primitives from certain isogeny-based assumptions (e.g., variants of CSIDH [CLM$^+$18, BKV19]).

Notations. We use $(\mathbb{G}, X, \star)$ to denote a group action $\star : \mathbb{G} \times X \to X$. Throughout the paper, we will assume that group actions are abelian and *regular*, i.e., both free and transitive (which is the case for CSIDH-style group actions). Note that for regular group actions, we have $|\mathbb{G}| = |X|$. Thus, if a group action is regular, then for any $x \in X$, the map $f_x : g \mapsto g \star x$ defines a bijection between $\mathbb{G}$ and X. We always use the additive notation $+$ to denote the group operation in $\mathbb{G}$. Since $\mathbb{G}$ is abelian, it can be viewed as a $\mathbb{Z}$-module and hence for any $z \in \mathbb{Z}$ and $g \in \mathbb{G}$, the term zg is well-defined. This property naturally extends to matrices as well, so for any matrix $\mathbf{M} \in \mathbb{G}^{m \times n}$ and any vector $\mathbf{z} \in \mathbb{Z}^n$, the term $\mathbf{Mz}$ is also well-defined. The group action extends naturally to the direct product group $\mathbb{G}^n$ for any positive integer n. If $\mathbf{g} \in \mathbb{G}^n$ and $\mathbf{x} \in X^n$, we use $\mathbf{g} \star \mathbf{x}$ to denote a vector of set elements whose ith component is $g_i \star x_i$.

Effective Group Action. We recall the definition of an effective group action (EGA) from [ADMP20]. An EGA allows us to do certain computations over $\mathbb{G}$ efficiently (e.g., group operation, inversion, and sampling uniformly), and there is an efficient procedure to compute the action of any group element on any set element. As pointed out by [ADMP20], the CSIDH-style assumption in [BKV19] (called CSI-FiSh) is an instance of EGA. We refer to [CLM$^+$18, BKV19, ADMP20] for more details on distributional properties of isogeny-based group actions.

Definition 2.7 (Effective Group Action). A group action $(\mathbb{G}, X, \star)$ is *effective* if it satisfies the following properties:

1. The group $\mathbb{G}$ is finite and there exist efficient algorithms for:
   (a) Membership testing (deciding whether a binary string represents a group element).
   (b) Equality testing and sampling uniformly in $\mathbb{G}$.
   (c) Group operation and computing inverse of any element in $\mathbb{G}$.
2. The set X is finite and there exist efficient algorithms for:

(a) Membership testing (to check if a string represents a valid set element),

(b) Unique representation (there is a canonical representation for any set element $x \in X$).

3. There exists a distinguished element $x_0 \in X$ with known representation.

4. There exists an efficient algorithm that given any $g \in \mathbb{G}$ and any $x \in X$, outputs $g \star x$.

Definition 2.8 (Weak Pseudorandom EGA). An effective group action $(\mathbb{G}, X, \star)$ is said to be a weak pseudorandom EGA (wPR-EGA) if it holds that

$$(x, y, t \star x, t \star y) \overset{c}{\approx} (x, y, u, u'),$$

where $x \leftarrow X$, $y \leftarrow X$, $t \leftarrow \mathbb{G}$, $u \leftarrow X$, and $u' \leftarrow X$.

Definition 2.9 (Linear Hidden Shift assumption [ADMP20]). Let $(\mathbb{G}, X, \star)$ be an effective group action (EGA), and let $n > \log|\mathbb{G}| + \omega(\log \lambda)$ be an integer. We say that liner hidden shift (LHS) assumption holds over $(\mathbb{G}, X, \star)$ if for any $\ell = \mathrm{poly}(\lambda)$ the following holds:

$$(\mathbf{x}, \mathbf{M}, \mathbf{Ms} \star \mathbf{x}) \overset{c}{\approx} (\mathbf{x}, \mathbf{M}, \mathbf{u}),$$

where $\mathbf{x} \leftarrow X^\ell$, $\mathbf{M} \leftarrow \mathbb{G}^{\ell \times n}$, $\mathbf{s} \leftarrow \{0,1\}^n$, and $\mathbf{u} \leftarrow X^\ell$.

3 Hinting PRG from DDH or LHS

In this section, we show how to construct a hinting PRG from either any DDH-hard group or any LHS-hard effective group action. Later, we define hinting weak PRFs and we show instantiations of this primitive based on DDH or LHS assumption by a natural extension of our techniques.

3.1 Hinting PRG from DDH

Our construction of hinting PRG from any DDH-hard group is as follows.

Construction. Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and fix some integer n such that $n > \log|\mathbb{G}| + \omega(\log \lambda)$. Given a cyclic group $\mathbb{G}$ with generator g, we use the notation $[a] = g^a$ and $[\mathbf{M}] = g^{\mathbf{M}}$ (exponentiation being applied componentwise) where $a \in \mathbb{Z}_q$ and $\mathbf{M} \in \mathbb{Z}_q^{m \times n}$ for any positive integer m and n. We use the notation $\langle \mathbf{a}, \mathbf{b} \rangle$ to denote the "dot" product of $\mathbf{a} \in \mathbb{Z}_q^n$ and $\mathbf{b} \in \mathbb{Z}_q^n$ modulo q. Our construction of hinting PRG from DDH assumption is as follows:

- $\mathsf{Setup}(1^\lambda)$: Sample $[\mathbf{M}] \leftarrow \mathbb{G}^{(n+1) \times n}$ and publish $\mathsf{pp} = [\mathbf{M}]$.
- $\mathsf{Eval}(\mathsf{pp} = [\mathbf{M}], \mathbf{s} \in \{0,1\}^n, i \in \{0\} \cup [n])$: Let $[\mathbf{m}_i]$ denote the ith[1] row of $[\mathbf{M}]$. Output $[\langle \mathbf{m}_i, \mathbf{s} \rangle]$.[2]

  Note that stacking up evaluation of the PRG on all indices $i \in \{0\} \cup [n]$ can simply be viewed as $[\mathbf{Ms}]$.

Security. We prove the security of the construction via the following theorem.

Theorem 3.1. *If $(\mathbb{G}, g, q)$ is a DDH-hard group then the construction above yields a hinting PRG.*

[1] For any matrix with $n+1$ rows, we number rows from 0 to n.

[2] Note that given any vector of group elements $[\mathbf{v}] \in \mathbb{G}^n$ and any vector $\mathbf{s} \in \{0,1\}^n$, one can efficiently compute $[\langle \mathbf{v}, \mathbf{s} \rangle]$ without the need to solve the discrete log problem.

Proof. Observe that by Lemma 3.2 (proved below) we have $([\mathbf{M}], [\mathbf{M}\mathbf{s}]) \overset{c}{\approx} ([\mathbf{M}], [\mathbf{u}])$ (where $[\mathbf{u}] \leftarrow \mathbb{G}^{n+1}$) and hence the pseudorandomness of the output in the plain PRG game follows from Lemma 3.2. Let $[\mathbf{m}_0] \in \mathbb{G}^n$ be the $0th$ row of $[\mathbf{M}]$, and let $[\bar{\mathbf{M}}]$ be all but the $0th$ row of $[\mathbf{M}]$ (i.e., bottom square matrix). To establish the security of the construction in the hinting PRG game, it is enough to show that

$$([\mathbf{m}_0], [\langle \mathbf{m}_0, \mathbf{s}\rangle], [\bar{\mathbf{M}}], [\mathbf{Y}]) \overset{c}{\approx} ([\mathbf{m}_0], [u], [\bar{\mathbf{M}}], [\mathbf{U}]), \qquad (*)$$

where $[u] \leftarrow \mathbb{G}$ and $[\mathbf{U}] \leftarrow \mathbb{G}^{n\times 2}$ are sampled uniformly and $[\mathbf{Y}] \in \mathbb{G}^{n\times 2}$ is distributed as follows

$$[y_{j,s_j}] = [\langle \mathbf{m}_j, \mathbf{s}\rangle], \quad [y_{j,1-s_j}] \leftarrow \mathbb{G}, \quad j \in [n].$$

We prove $(*)$ via a hybrid argument. Let H_0 and H_1 be the hybrids that correspond to the left-hand side and right-hand side of $(*)$, respectively (i.e., "real" game and "ideal" game). We now argue that $H_0 \overset{c}{\approx} H_1$. Let $\mathcal{A}$ be an adversary that distinguishes H_0 from H_1. We construct an adversary $\mathcal{A}'$ that distinguishes H_0' from H_1' where[1]

$$H_0' := ([\mathbf{m}_0], [\langle \mathbf{m}_0, \mathbf{s}\rangle], [\bar{\mathbf{M}}], [\bar{\mathbf{M}}\mathbf{s}]), \quad H_1' := ([\mathbf{m}_0], [u_0], [\bar{\mathbf{M}}], [\mathbf{u}]),$$

and by Lemma 3.2 it follows that the advantage of $\mathcal{A}$ should also be negligible.

Given a tuple $H_b' = ([\mathbf{m}_0], [z_0], [\bar{\mathbf{M}}], [\mathbf{z}])$, where H_b' is either distributed as H_0' or H_1', the external adversary $\mathcal{A}'$ samples a random $[\mathbf{d}] \leftarrow \mathbb{G}^n$. Let $[\mathbf{D}] \in \mathbb{G}^{n\times n}$ be a *diagonal* matrix whose diagonal is $[\mathbf{d}]$, i.e., ijth entry of $\mathbf{D}$ is 0 for any $i \neq j$. In the next step, $\mathcal{A}'$ runs $\mathcal{A}$ on the following tuple

$$([\mathbf{m}_0], [z_0], [\mathbf{M}'] := [\bar{\mathbf{M}} + \mathbf{D}], [\mathbf{Y}]),$$

where $[\mathbf{Y}]$ is an n by 2 matrix whose first and second columns are $[\mathbf{z}]$ and $[\mathbf{z} + \mathbf{d}]$ respectively. We define the output of $\mathcal{A}'$ to be the same as the output of $\mathcal{A}$.

Observe that (in the view of $\mathcal{A}$) the terms $[\mathbf{m}_0]$ and $[\mathbf{M}']$ are distributed uniformly. Moreover, if $[\mathbf{z}]$ is uniform then $[\mathbf{Y}]$ will be distributed uniformly as well. Therefore, $\mathcal{A}'$ perfectly simulates the "ideal" hybrid H_1. On the other hand, if $[\mathbf{z}] = [\bar{\mathbf{M}}\mathbf{s}]$ then from the view of $\mathcal{A}$ the matrix $[\mathbf{Y}]$ is distributed as

$$[y_{j,s_j}] = [\langle \mathbf{m}_j', \mathbf{s}\rangle], \quad [y_{j,1-s_j}] = [(-1)^{s_j} \cdot d_j + \langle \mathbf{m}_j', \mathbf{s}\rangle], \quad j \in [n].$$

To see why the relations above hold, notice that $[\langle \mathbf{m}_j', \mathbf{s}\rangle] = [\langle \bar{\mathbf{m}}_j, \mathbf{s}\rangle + s_j \cdot d_j]$ where $\mathbf{m}_j'$ and $\bar{\mathbf{m}}_j$ denote the jth row of $\mathbf{M}'$ and $\bar{\mathbf{M}}$, respectively. Because $[\mathbf{d}]$ is distributed uniformly and independently from $[\mathbf{M}']$ (in the view of $\mathcal{A}$), it follows that in the view of $\mathcal{A}$ we have

$$\left([\mathbf{M}'], \{[y_{j,s_j}]\}_{j\in n}, [y_{j,1-s_j}]\}_{j\in n}\right) \overset{s}{\approx} ([\mathbf{M}'], \{[y_{j,s_j}]\}_{j\in n}, [\mathbf{u}]),$$

where $[\mathbf{u}] \leftarrow \mathbb{G}^n$, and hence $\mathcal{A}'$ properly simulates the "real" hybrid H_0, as required. □

A generic version of the following lemma has been proved in [AMP19] for the output group of any key-homomorphic weak PRF. Below, we provide a short proof for any DDH-hard group $\mathbb{G}$.

Lemma 3.2. *Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and fix some integer ℓ and n such that $n > \log|\mathbb{G}| + \omega(\log \lambda)$ and $\ell = \text{poly}(\lambda)$. If $[\mathbf{M}] \leftarrow \mathbb{G}^{\ell\times n}$ and $\mathbf{s} \leftarrow \{0,1\}^n$, then $([\mathbf{M}], [\mathbf{M}\mathbf{s}]) \overset{c}{\approx} ([\mathbf{M}], [\mathbf{u}])$, where $[\mathbf{u}] \leftarrow \mathbb{G}^\ell$ is sampled uniformly.*

Proof. Let $[\bar{\mathbf{M}}] \in \mathbb{G}^{\ell\times n}$ be a matrix of group elements whose (i,j) entry is $[a_i \cdot b_j]$ where $a_i \leftarrow \mathbb{Z}_q, b_j \leftarrow \mathbb{Z}_q$ (for $i \in [\ell], j \in [n]$). By the leftover hash lemma, it follows that given $[\bar{\mathbf{M}}]$, the term $[\bar{\mathbf{M}}\mathbf{s}]$ is statistically indistinguishable from a fresh DDH tuple, i.e., given $[\bar{\mathbf{M}}]$ it holds that

$$[\bar{\mathbf{M}}\mathbf{s}] = \begin{pmatrix} [a_1 \cdot \langle \mathbf{b}, \mathbf{s}\rangle] \\ [a_2 \cdot \langle \mathbf{b}, \mathbf{s}\rangle] \\ \vdots \\ [a_\ell \cdot \langle \mathbf{b}, \mathbf{s}\rangle] \end{pmatrix} \overset{s}{\approx} \begin{pmatrix} [a_1 \cdot b^*] \\ [a_2 \cdot b^*] \\ \vdots \\ [a_\ell \cdot b^*] \end{pmatrix},$$

[1] This is simply a special case of Lemma 3.2 with $k = n + 1$ and we wrote the first row separately.

where $b^* \leftarrow \mathbb{Z}_q$ is chosen randomly. By a standard hybrid argument, it follows from the DDH assumption that $([\bar{\mathbf{M}}], [\bar{\mathbf{M}}\mathbf{s}]) \overset{c}{\approx} ([\bar{\mathbf{M}}], [\mathbf{u}])$. Moreover, by the DDH assumption we have $[\bar{\mathbf{M}}] \overset{c}{\approx} [\mathbf{M}]$. Therefore, it follows from a simple hybrid argument that $([\mathbf{M}], [\mathbf{M}\mathbf{s}]) \overset{c}{\approx} ([\mathbf{M}], [\mathbf{u}])$, as desired. □

3.2 Hinting PRG from LHS

We now show how to construct a hinting PRG from any LHS-hard EGA. The construction is similar to our DDH-based construction of hinting PRG, with suitable modifications to translate our techniques to the setting of EGA.

Construction. Let $(\mathbb{G}, X, \star)$ be an EGA for which LHS assumption holds. Let n be the secret dimension of the LHS assumption. We describe a construction of hinting PRG from the LHS assumption as follows. In the construction below, note that the group $\mathbb{G}$ is written additively (viewed as a $\mathbb{Z}$-module).

- $\mathsf{Setup}(1^\lambda)$: Sample $\mathbf{M} \leftarrow \mathbb{G}^{(n+1)\times n}$ and $\mathbf{x} = (x_0, x_1, \dots, x_n) \leftarrow X^{n+1}$, and publish $\mathsf{pp} = (\mathbf{M}, \mathbf{x})$.
- $\mathsf{Eval}(\mathsf{pp} = \mathbf{M}, \mathbf{s} \in \{0,1\}^n, i \in \{0\} \cup [n])$: Let $\mathbf{m}_i$ denote the ith[1] row of $\mathbf{M}$. Output $\langle \mathbf{m}_i, \mathbf{s}\rangle \star x_i$.

  Note that similar to the DDH-based construction, concatenating evaluation of the PRG on all indices $i \in \{0\} \cup [n]$ can be viewed as a larger instance of LHS assumption, i.e., $\mathbf{M}\mathbf{s} \star \mathbf{x}$.

Security. We argue the security of the construction above based on the LHS assumption as follows.

Theorem 3.3. *Let $(\mathbb{G}, X, \star)$ be an EGA. If LHS assumption holds over $(\mathbb{G}, X, \star)$ then the construction above yields a hinting PRG.*

Proof. Pseudorandomness of the output in the PRG game follows directly from the LHS assumption. Let $\mathbf{m}_0 \in \mathbb{G}^n$ be the 0*th* row of $\mathbf{M}$, and let $\bar{\mathbf{M}}$ be all but the 0*th* row of $\mathbf{M}$ (i.e., bottom square matrix). It suffices to show that

$$H_0 := (\mathbf{x}, \mathbf{m}_0, \langle \mathbf{m}_0, \mathbf{s}\rangle \star x_0, \bar{\mathbf{M}}, \mathbf{Y}) \overset{c}{\approx} (\mathbf{x}, \mathbf{m}_0, u, \bar{\mathbf{M}}, \mathbf{U}) := H_1, \qquad (**)$$

where $u \leftarrow X$ and $\mathbf{U} \leftarrow X^{n\times 2}$ are uniform and $\mathbf{Y} \in X^{n\times 2}$ is distributed as

$$y_{j,s_j} = \langle \bar{\mathbf{m}}_j, \mathbf{s}\rangle \star x_j, \quad y_{j,1-s_j} \leftarrow X, \quad j \in [n].$$

Let H_0 and H_1 be the hybrids that correspond to the left-hand side and right-hand side of $(**)$, respectively. We now argue that $H_0 \overset{c}{\approx} H_1$. Let $\mathcal{A}$ be an adversary that distinguishes H_0 from H_1, we construct another adversary $\mathcal{A}'$ that distinguishes between the following tuples

$$H_0' := (\mathbf{x}, \mathbf{m}_0, \langle \mathbf{m}_0, \mathbf{s}\rangle \star x_0, \bar{\mathbf{M}}, \bar{\mathbf{M}}\mathbf{s} \star \bar{\mathbf{x}}), \quad H_1' := (\mathbf{x}, \mathbf{m}_0, u_0, \bar{\mathbf{M}}, \mathbf{u}),$$

where $u_0 \leftarrow X$ and $\mathbf{u} \leftarrow X^n$ are sampled uniformly, and $\bar{\mathbf{x}} = (x_1, \dots, x_n)$ is the last n components of $\mathbf{x}$. Observe that the indistinguishability of H_0' and H_1' follows directly from the LHS assumption. Given a tuple of the form $H_b' = (\mathbf{x}, \mathbf{m}_0, z_0, \bar{\mathbf{M}}, \mathbf{z})$, where H_b' is either distributed as H_0' or H_1', the external adversary $\mathcal{A}'$ samples a random $\mathbf{d} \leftarrow \mathbb{G}^n$. Let $\mathbf{D} \in \mathbb{G}^{n\times n}$ be a *diagonal* matrix whose diagonal is $\mathbf{d}$, i.e., ijth entry of $\mathbf{D}$ is the identity element of $\mathbb{G}$ for any $i \neq j$. In the next step, $\mathcal{A}'$ runs $\mathcal{A}$ on the following tuple

$$(\mathbf{x}, \mathbf{m}_0, z_0, \mathbf{M}' := \bar{\mathbf{M}} + \mathbf{D}, \mathbf{Y}),$$

where $\mathbf{Y} \in X^{n\times 2}$ is a matrix whose first and second rows are $\mathbf{z}$ and $\mathbf{d} \star \mathbf{z}$, respectively. Finally, $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. It follows by inspection that $\mathcal{A}'$ perfectly simulates the "ideal" hybrid, i.e., it maps H_1' to H_1. On the other hand, if $\mathbf{z} = \bar{\mathbf{M}}\mathbf{s} \star \bar{\mathbf{x}}$ then from the view of $\mathcal{A}'$ the matrix $\mathbf{Y}$ is distributed as

$$y_{j,s_j} = \langle \mathbf{m}_j', \mathbf{s}\rangle \star x_j, \quad y_{j,1-s_j} = \left((-1)^{s_j} \cdot d_j\right) \star \left(\langle \mathbf{m}_j', \mathbf{s}\rangle \star x_j\right), \quad j \in [n].$$

Because $\mathbf{d}$ is distributed uniformly and independently from $\bar{\mathbf{M}}$ (in the view of $\mathcal{A}$), it follows that $\{y_{j,1-s_j}\}_{j\in[n]}$ is distributed uniformly in the view of $\mathcal{A}$ as well, and hence $\mathcal{A}'$ properly simulates the "real" hybrid H_0, as required. □

[1] As before, we number rows from 0 to n.

3.3 Trapdoor Functions from LHS-hard wPR-EGA

In this section, we extend our technique of publicly computable shifts (used in our construction of hinting PRG from LHS-hard EGA) to achieve a direct construction of TDFs from any LHS-hard weak pseudorandom EGA. Our construction avoids the many layers of generic transformation required by the prior construction of TDFs from such isogeny-based assumption proposed in [ADMP20] based on the framework of [KMT19a].

Construction. Let $(\mathbb{G}, X, \star)$ be a wPR-EGA such that LHS assumptions holds over $(\mathbb{G}, X, \star)$. We now describe a construction of TDF from such EGA. Let $\mathsf{Ext} : \mathcal{S} \times X \to G$ be a (statistical) extractor where $\mathcal{S}$ denotes the seed space.[1]

- $\mathsf{Gen}(1^\lambda)$: Sample $\mathbf{M} \leftarrow \mathbb{G}^{n\times n}$ where $n = n(\lambda)$ is the secret dimension of the LHS assumption. Sample $\bar{\mathbf{x}} \leftarrow X^n$, $\mathbf{x} \leftarrow X^n$, $\mathbf{t} \leftarrow \mathbb{G}^n$, $\mathsf{seed} \leftarrow \mathcal{S}$, and let $\mathbf{y} = \mathbf{t} \star \mathbf{x}$ where the action is applied componentwise. Output the tuple $\mathsf{ek} = (\mathsf{seed}, \mathbf{M}, \bar{\mathbf{x}}, \mathbf{x}, \mathbf{y})$ as evaluation key and $\mathbf{t}$ as trapdoor.
- $\mathsf{Eval}(\mathsf{ek} = (\mathsf{seed}, \mathbf{M}, \bar{\mathbf{x}}, \mathbf{x}, \mathbf{y}), (\mathbf{s} \in \{0,1\}^n, \mathbf{r} \in X^n, \mathbf{r}' \in X^n))$: To evaluate the function on the input $(\mathbf{s}, \mathbf{r}, \mathbf{r}')$, output $(\mathbf{V} \in X^{n\times 2}, \mathbf{Z} \in X^{n\times 2})$ where[2]

$$\begin{aligned} v_{i,s_i} &= \mathsf{Ext}\big(\mathsf{seed}, \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i\big) \star x_i, & v_{i,1-s_i} &= r_i, & \\ z_{i,s_i} &= \mathsf{Ext}\big(\mathsf{seed}, \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i\big) \star y_i, & z_{i,1-s_i} &= r'_i, & i \in [n]. \end{aligned}$$

- $\mathsf{Invert}(\mathbf{t}, (\mathbf{V}, \mathbf{Z}))$: To invert on the input $(\mathbf{V}, \mathbf{Z})$ using the trapdoor $\mathbf{t}$, first compute $\mathbf{s}$ as follows:

$$s_i = \begin{cases} 0 & t_i \star v_{i,0} = z_{i,0}, \\ 1 & t_i \star v_{i,1} = z_{i,1}. \end{cases}$$

  Let $\mathbf{r}$ and $\mathbf{r}'$ be two vectors such that $r_i = v_{i,1-s_i}$ and $r'_i = z_{i,1-s_i}$ for $i \in [n]$. Output $(\mathbf{s}, \mathbf{r}, \mathbf{r}')$.

Correctness of the inversion algorithm follows by inspection. We prove the one-wayness of the scheme via the following theorem.

Theorem 3.4. *If $(\mathbb{G}, X, \star)$ is an LHS-hard wPR-EGA then the construction above satisfies one-wayness.*

Proof. To prove the one-wayness it suffices to show that

$$H_0 := (\mathsf{ek}, \mathbf{V}, \mathbf{Z}) \overset{c}{\approx} (\mathsf{ek}, \mathbf{U}, \mathbf{U}') := H_3,$$

where ek, $\mathbf{V}$, $\mathbf{Z}$ are distributed as in the construction above, and $\mathbf{U}$, $\mathbf{U}'$ are two random matrices of set elements. We do the proof via a hybrid argument.

- H_0: This is the "real" game and H_0 corresponds to the tuple $(\mathsf{ek}, \mathbf{V}, \mathbf{Z})$ where ek, $\mathbf{V}$, $\mathbf{Z}$ are distributed as in the construction.
- H_1: In this hybrid we change the way two matrices are generated. Specifically, this hybrid corresponds to the tuple $(\mathsf{ek}, \mathbf{V}^{(1)}, \mathbf{Z}^{(1)})$ where $\mathbf{V}^{(1)}$ and $\mathbf{Z}^{(1)}$ are distributed as follows.

$$\begin{aligned} v^{(1)}_{i,s_i} &= \mathsf{Ext}\big(\mathsf{seed}, \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i\big) \star x_i, & v^{(1)}_{i,1-s_i} &= \rho_i \star x_i, \quad \rho_i \leftarrow \mathbb{G}, \\ z^{(1)}_{i,s_i} &= \mathsf{Ext}\big(\mathsf{seed}, \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i\big) \star y_i, & z^{(1)}_{i,1-s_i} &= \rho_i \star y_i, \quad i \in [n]. \end{aligned}$$

[1] Note that we cannot use the bit representation of an element of X to generate a group element G without using extractor, because for some EGAs (and in particular for isogeny-based group actions), elements of X do *not* have compact represenation.

[2] $\mathbf{m}_i$ denotes the ith row of $\mathbf{M}$.

- H_2: In this hybrid we use randomly chosen group elements instead of using the vector $\mathbf{s}$ to generate the output matrices. This hybrid corresponds to the tuple $(\mathsf{ek}, \mathbf{V}^{(2)}, \mathbf{Z}^{(2)})$ where $\mathbf{V}^{(2)}$ and $\mathbf{Z}^{(2)}$ are distributed as follows.

$$\begin{aligned} v^{(2)}_{i,s_i} &= \sigma_i \star x_i, & v^{(2)}_{i,1-s_i} &= \rho_i \star x_i, \quad (\sigma_i, \rho_i) \leftarrow \mathbb{G}^2, \\ z^{(2)}_{i,s_i} &= \sigma_i \star y_i, & z^{(2)}_{i,1-s_i} &= \rho_i \star y_i, \qquad i \in [n]. \end{aligned}$$

- H_3: This hybrid corresponds to the tuple $(\mathsf{ek}, \mathbf{U}, \mathbf{U}')$ where two matrices $\mathbf{U}$ and $\mathbf{U}'$ are generated randomly.

We argue the indistinguishability of consecutive hybrids as follows:

- $H_0 \overset{c}{\approx} H_1$: This follows from the weak pseudorandomness of the group action. Given a challenge tuple $(\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}')$ where $(\mathbf{x}', \mathbf{y}')$ is either uniform and independent of $(\mathbf{x}, \mathbf{y})$ or $x'_i = \rho_i \star x_i$, $y'_i = \rho_i \star y_i$ for $i \in [n]$, the reduction samples

$$\mathsf{seed} \leftarrow \mathcal{S}, \quad \mathbf{M} \leftarrow \mathbb{G}^{n \times n}, \quad \mathbf{s} \leftarrow \{0,1\}^n, \quad \bar{\mathbf{x}} \leftarrow X^n,$$

and outputs $(\mathsf{ek} = (\mathsf{seed}, \mathbf{M}, \bar{\mathbf{x}}, \mathbf{x}, \mathbf{y}), \bar{\mathbf{V}}, \bar{\mathbf{Z}})$, where $\bar{\mathbf{V}}$ and $\bar{\mathbf{Z}}$ are computed as

$$\begin{aligned} \bar{v}_{i,s_i} &= \mathsf{Ext}\big(\mathsf{seed}, \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i\big) \star x_i, & \bar{v}_{i,1-s_i} &= x'_i, \\ \bar{z}_{i,s_i} &= \mathsf{Ext}\big(\mathsf{seed}, \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i\big) \star y_i, & \bar{z}_{i,1-s_i} &= y'_i, \qquad i \in [n]. \end{aligned}$$

It follows by inspection that the reduction maps a random tuple to H_0 and a pseudorandom tuple to H_1. Thus, the hybrid H_0 is computationally indistinguishable from H_1 based on the weak pseudorandomness of EGA.

- $H_1 \overset{c}{\approx} H_2$: This follows from the security of the underlying hinting PRG. By Theorem 3.3 we know that $(\mathbf{M}, \bar{\mathbf{x}}, \mathbf{W}) \overset{c}{\approx} (\mathbf{M}, \bar{\mathbf{x}}, \mathbf{U})$, where $\mathbf{U} \leftarrow X^{n \times 2}$, $w_{i,s_i} = \langle \mathbf{m}_i, \mathbf{s}\rangle \star \bar{x}_i$, and $w_{i,1-s_i} \leftarrow X$ for $i \in [n]$. Given a challenge tuple of the form $(\mathbf{M}, \bar{\mathbf{x}}, \bar{\mathbf{W}})$ such that $\bar{\mathbf{W}}$ is either distributed as $\mathbf{W}$ or $\mathbf{U}$, the reduction samples $\mathsf{seed} \leftarrow \mathcal{S}$, $\mathbf{x} \leftarrow X^n$ and $\mathbf{y} \leftarrow X^n$, and outputs

$$(\mathsf{ek} = (\mathsf{seed}, \mathbf{M}, \bar{\mathbf{x}}, \mathbf{x}, \mathbf{y}), \bar{\mathbf{V}}, \bar{\mathbf{Z}}),$$

where $\bar{\mathbf{V}}$ and $\bar{\mathbf{Z}}$ are computed as

$$\begin{aligned} \bar{v}_{i,0} &= \mathsf{Ext}\big(\mathsf{seed}, \bar{w}_{i,0}\big) \star x_i, & \bar{v}_{i,1} &= \mathsf{Ext}\big(\mathsf{seed}, \bar{w}_{i,1}\big) \star x_i, \\ \bar{z}_{i,0} &= \mathsf{Ext}\big(\mathsf{seed}, \bar{w}_{i,0}\big) \star y_i, & \bar{z}_{i,1} &= \mathsf{Ext}\big(\mathsf{seed}, \bar{w}_{i,1}\big) \star y_i, \qquad i \in [n]. \end{aligned}$$

Observe that the reduction maps "hinting" samples $(\mathbf{W})$ to H_1, and it maps random samples $(\mathbf{U})$ to H_2. Thus, H_1 is computationally indistinguishable from H_2 based on the LHS assumption.

- $H_2 \overset{c}{\approx} H_3$: This follows from the weak pseudorandomness of the group action. The proof is similar to the proof of $H_0 \overset{c}{\approx} H_1$ and hence we omit the details. □

Remark 3.5. The input space of the TDF construction above consists of an n-bit string and $2n$ set elements. We note that for the isogeny-based instantiation, we do not know how to sample set elements directly and hence *part* of the input for our TDF construction should be accompanied with a sampling algorithm. As discussed by the authors of [HKW20], TDFs with a sampling algorithm admit a trivial construction. While our construction departs from this paradigm by having a two-part input space where each part can be sampled *independently*, it still has the drawback of requiring a sampling algorithm for one part of the input space. We refer to [HKW20] for more details.

4 Hinting weak PRF

In this section, we define hinting weak PRF and we show instantiations of this primitive based on DDH or LHS assumption. Informally, a hinting weak PRF can be viewed as an extended version of hinting PRG, where polynomially many hints of the secret key can be provided (as opposed to only one hint in the hinting PRG security game).

Definition 4.1. Let $F : K \times X \to \bar{Y}$ be a weak PRF where $K = \{0,1\}^n$ and $\bar{Y} = Y^n$ for some efficiently samplable set Y. We say that F is a hinting weak PRF if for any $Q = \mathrm{poly}(\lambda)$ it holds that

$$\big(x_i, \mathsf{S}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)})\big)_{i\in[Q]} \overset{c}{\approx} \big(x_i, \mathbf{U}_i\big)_{i\in[Q]},$$

where $\mathbf{k} \leftarrow K$, $x_i \leftarrow X$, $\mathbf{r}^{(i)} \leftarrow Y^n$, $\mathbf{U}_i \leftarrow Y^{n\times 2}$, $\mathbf{y}^{(i)} = F(\mathbf{k}, x_i)$, and $\mathsf{S}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)})$ is an n by 2 "selector" matrix (with respect to $\mathbf{k}$) defined as follows:

$$\mathsf{S}_{j,k_j}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = y_j^{(i)}, \quad \mathsf{S}_{j,1-k_j}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = r_j^{(i)}, \quad j \in [n].$$

To clarify the notation, $\mathsf{S}_{j,b}$ denotes the (j,b)th entry, k_j is the jth bit of $\mathbf{k}$, and $y_j^{(i)}$ (respectively, $r_j^{(i)}$) denotes the jth entry of the vector $\mathbf{y}^{(i)}$ (respectively, $\mathbf{r}^{(i)}$).

4.1 Hinting weak PRF from DDH

We begin by showing how to construct a hinting weak PRF from any DDH-hard group.

Construction. Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and fix some integer n such that $n > \log|\mathbb{G}| + \omega(\log\lambda)$. We use the notation from Section 3.1 to describe a construction of hinting weak PRF from DDH assumption. Our DDH-based hinting weak PRF is a function of the form $F : \{0,1\}^n \times \mathbb{G}^{n\times n} \to \mathbb{G}^n$. Thus, for any input, one group element is published per each bit of the secret key.

- $\mathsf{Gen}(1^\lambda)$: To generate a key, sample $\mathbf{k} \leftarrow \{0,1\}^n$.
- $F(\mathbf{k} = \{0,1\}^n, [\mathbf{M}] \in \mathbb{G}^{n\times n})$: To evaluate the function, output $[\mathbf{M}\mathbf{k}]$.

Security. We argue the security of the hinting weak PRF above based on the DDH assumption as follows.

Theorem 4.2. *If $(\mathbb{G}, g, q)$ is a DDH-hard group then the construction above yields a hinting weak PRF.*

Proof. Note that weak pseudorandomness of F (in the weak PRF game) follows from Lemma 3.2. To argue the hinting security property we need to show that

$$H_0 := \big([\mathbf{M}_i], \mathsf{S}\big([\mathbf{y}^{(i)}], [\mathbf{r}^{(i)}]\big)\big)_{i\in[Q]} \overset{c}{\approx} \big([\mathbf{M}_i], [\mathbf{U}_i]\big)_{i\in[Q]} =: H_1, \qquad (\diamondsuit)$$

where $[\mathbf{M}_i] \leftarrow \mathbb{G}^{n\times n}$, $\mathbf{k} \leftarrow \{0,1\}^n$, $[\mathbf{r}^{(i)}] \leftarrow \mathbb{G}^n$, $[\mathbf{U}_i] \leftarrow \mathbb{G}^{n\times 2}$, $[\mathbf{y}^{(i)}] = [\mathbf{M}_i\mathbf{k}]$, and

$$\mathsf{S}_{j,k_j}\big([\mathbf{y}^{(i)}], [\mathbf{r}^{(i)}]\big) = [y_j^{(i)}], \quad \mathsf{S}_{j,1-k_j}\big([\mathbf{y}^{(i)}], [\mathbf{r}^{(i)}]\big) = [r_j^{(i)}], \quad j \in [n].$$

To show that $(\diamondsuit)$ holds, we extend the proof of DDH-based hinting PRG to multiple instances. By Lemma 3.2 for $Q = \mathrm{poly}(\lambda)$ we have

$$H_0' := \big([\mathbf{M}_i], \big([\mathbf{M}_i\mathbf{k}]\big)_{i\in[Q]} \overset{c}{\approx} \big([\mathbf{M}_i], [\mathbf{u}_i]\big)_{i\in[Q]} =: H_1'.$$

Let $\mathcal{A}$ be an adversary that distinguishes H_0 from H_1. We construct an adversary $\mathcal{A}'$ to distinguish H_0' from H_1'. Given $H_b' = ([\mathbf{M}_i], [\mathbf{z}^{(i)}])_{i\in[Q]}$ (where H_b' is distributed as either H_0' or H_1'), the adversary $\mathcal{A}'$ samples Q uniform vectors $([\mathbf{d}^{(i)}] \leftarrow \mathbb{G}^n)_{i\in[Q]}$, and it sets $[\mathbf{M}_i'] := [\mathbf{M}_i + \mathbf{D}_i]$ where $\mathbf{D}_i$ is a diagonal matrix whose diagonal is $\mathbf{d}^{(i)}$. It then runs $\mathcal{A}$ on $([\mathbf{M}_i'], [\mathbf{Y}_i])_{i\in[Q]}$ where $[\mathbf{Y}_i]$ is an n by 2 matrix whose first (respectively, second) row is $[\mathbf{z}^{(i)}]$

(respectively, $[\mathbf{z}^{(i)} + \mathbf{d}^{(i)}]$). The output of $\mathcal{A}'$ is defined to be the same as the output of $\mathcal{A}$. It is immediate to see that $\mathcal{A}'$ maps H_1' to H_1. On the other hand, $\{[\mathbf{M}_i'], [\mathbf{d}^{(i)}]\}_{i\in[Q]}$ are uniform in the view of $\mathcal{A}$ and hence if $H_b' \equiv H_0'$ then an argument similar to the proof of DDH-based hinting PRG implies that

$$\big([\mathbf{M}_i'], [\mathbf{Y}_i]\big)_{i\in[Q]} \overset{s}{\approx} \big([\mathbf{M}_i'], \mathsf{S}\big([\mathbf{z}^{(i)}], [\mathbf{r}^{(i)}]\big)\big)_{i\in[Q]}.$$

where $[\mathbf{r}^{(i)}] \leftarrow \mathbb{G}^n$ for each i. Thus, $\mathcal{A}'$ properly maps H_0' to (a hybrid that is statistically indistinguishable from) H_0, and the proof is complete. □

4.2 Hinting weak PRF from LHS

We now show how to build a hinting weak PRF from any LHS-hard EGA. Our construction is similar to our DDH-based construction of hinting weak PRF, with suitable modifications to translate our techniques to the setting of EGA.

Construction. Let $(\mathbb{G}, X, \star)$ be an EGA for which LHS assumption holds. Let n be the secret dimension of the LHS assumption. Building upon the notation from Section 3.2, we describe a hinting weak PRF of the form $F : \{0,1\}^n \times (\mathbb{G}^{n\times n} \times X^n) \to X^n$.

- $\mathsf{Gen}(1^\lambda)$: To generate a key, sample $\mathbf{k} \leftarrow \{0,1\}^n$.
- $F(\mathbf{k} = \{0,1\}^n, (\mathbf{M} \in \mathbb{G}^{n\times n}, \mathbf{x} \in X^n))$: Output $\mathbf{M}\mathbf{k} \star \mathbf{x}$.

Security. We establish the security of the hinting weak PRF above based on the LHS assumption as follows.

Theorem 4.3. *Let $(\mathbb{G}, X, \star)$ be an EGA. If LHS assumption holds over $(\mathbb{G}, X, \star)$ then F is a hinting weak PRF.*

Proof. Weak pseudorandomness of F directly follows from the LHS assumption. To prove hinting security property, it suffices to show that

$$H_0 := \big(\mathbf{x}_i, \mathbf{M}_i, \mathsf{S}\big(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}\big)\big)_{i\in[Q]} \overset{c}{\approx} \big(\mathbf{x}_i, \mathbf{M}_i, \mathbf{U}_i\big)_{i\in[Q]} =: H_1, \qquad (\Diamond\Diamond)$$

where $\mathbf{M}_i \leftarrow \mathbb{G}^{n\times n}$, $\mathbf{k} \leftarrow \{0,1\}^n$, $\mathbf{r}^{(i)} \leftarrow X^n$, $\mathbf{U}_i \leftarrow X^{n\times 2}$, $\mathbf{y}^{(i)} = \mathbf{M}_i\mathbf{k} \star \mathbf{x}_i$, and

$$\mathsf{S}_{j,k_j}\big(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}\big) = y_j^{(i)}, \quad \mathsf{S}_{j,1-k_j}\big(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}\big) = r_j^{(i)}, \quad j \in [n].$$

In the next step, we show a reduction from the LHS assumption to $(\Diamond\Diamond)$. First, by the LHS assumption we have

$$H_0' := \big(\mathbf{x}_i, \mathbf{M}_i, \mathbf{M}_i\mathbf{k} \star \mathbf{x}_i\big)_{i\in[Q]} \overset{c}{\approx} \big(\mathbf{x}_i, \mathbf{M}_i, \mathbf{u}_i\big)_{i\in[Q]} =: H_1'.$$

Given an adversary $\mathcal{A}$ that distinguishes H_0 from H_1, we construct another adversary $\mathcal{A}'$ against the LHS assumption. Given an LHS challenge of the form $H_b' = (\mathbf{x}_i, \mathbf{M}_i, \mathbf{z}^{(i)})_{i\in[Q]}$ (where H_b' is identical to either H_0' or H_1'), the adversary $\mathcal{A}'$ samples Q uniform vectors $(\mathbf{d}^{(i)} \leftarrow \mathbb{G}^n)_{i\in[Q]}$ and it sets $\mathbf{M}_i' := \mathbf{M}_i + \mathbf{D}_i$, where $\mathbf{D}_i$ is a diagonal matrix whose diagonal is $\mathbf{d}^{(i)}$. We define the output of $\mathcal{A}'$ to be the output of $\mathcal{A}$ on $(\mathbf{x}_i, \mathbf{M}_i', \mathbf{Y}_i)_{i\in[Q]}$ where $\mathbf{Y}_i \in X^{n\times 2}$ is the matrix whose first and second rows are $\mathbf{z}^{(i)}$ and $\mathbf{d}^{(i)} \star \mathbf{z}^{(i)}$, respectively. Clearly, $\mathcal{A}'$ maps H_1' to H_1. Moreover, $(\mathbf{M}_i', \mathbf{d}^{(i)})_{i\in[Q]}$ are uniform in the view of $\mathcal{A}$ and hence if $H_b' \equiv H_0'$ then an argument similar to the proof of LHS-based hinting PRG implies that

$$\big(\mathbf{x}_i, \mathbf{M}_i', \mathbf{Y}_i\big)_{i\in[Q]} \overset{s}{\approx} \big(\mathbf{x}_i, \mathbf{M}_i', \mathsf{S}\big(\mathbf{z}^{(i)}, \mathbf{u}^{(i)}\big)\big)_{i\in[Q]},$$

and so $\mathcal{A}'$ properly maps H_0' to (a hybrid that is statistically close to) H_0. □

5 Functional Hinting Property and KDM Security

In this section, we introduce functional hinting PRG, which is a strengthening of hinting PRG that guarantees PRG security in the presence of hints about each bit of some *function* of the seed. We also introduce a natural extension, namely a functional hinting wPRF, that guarantees wPRF security in the presence of multiple hints about each bit of some (adversarially chosen) function of the secret key. We show that a functional hinting weak PRF with respect to a family of functions $\mathcal{F}$ can be used to realize a symmetric-key KDM-secure encryption scheme with respect to the same function family $\mathcal{F}$ in a *black-box* manner. We then build upon our approach of realizing hinting PRGs and hinting weak PRFs to realize simple constructions of functional hinting PRGs and functional weak PRFs for the family of projective quadratic functions (and functions of higher degree) based on the DDH assumption.

5.1 Functional Hinting PRG

We first define functional hinting PRG, which is a generalized version of hinting PRG for which the security game is defined in terms of a *function* of the seed of PRG, rather the seed itself. A plain hinting PRG can be simply viewed as a functional hinting PRG with respect to the identity function.

Definition 5.1. Let $f : \{0,1\}^n \to \{0,1\}^m$ be an efficiently computable (boolean) function. A functional hinting PRG $G_{\mathsf{pp}} : \{0,1\}^n \to \bar{Y} = Y^{m+1}$ with respect to f is defined by two algorithms $(\mathsf{Setup}, \mathsf{Eval})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^m)$: A randomized algorithm that takes the seed length n and the number of hinting blocks m, and it outputs pp as the public parameter.
- $\mathsf{Eval}(\mathsf{pp}, i \in \{0\} \cup [m], \mathsf{s} \in \{0,1\}^n)$: A deterministic algorithm that on pp and an index i, it outputs $y_i \in Y$. By stacking the outputs for all $\in \{0\} \cup [m]$, we can view the output as an element of Y^{m+1}, i.e., $G_{\mathsf{pp}}(\mathbf{s}) \in Y^{m+1}$.

We say that G_{pp} (defined by the algorithms above) is a functional hinting PRG with respect to the function $f : \{0,1\}^n \to \{0,1\}^m$, if for $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n, 1^m)$ and randomly chosen seed $\mathbf{s} \leftarrow \{0,1\}^n$ it holds that

$$\left(y_0, (y_{j,b})_{j\in[m],b\in\{0,1\}}\right) \overset{c}{\approx} \left(u_0, (u_{j,b})_{j\in[m],b\in\{0,1\}}\right),$$

where

$$\mathbf{v} := f(\mathbf{s}) \in \{0,1\}^m, \quad (y_0, y_{1,v_1}, \ldots, y_{m,v_m}) = G_{\mathsf{pp}}(\mathbf{s}) \in Y^{m+1},$$

and all other elements generated uniformly from Y, i.e.,

$$\{y_{j,1-v_j} \leftarrow Y\}_{j\in[m]}, \quad u_0 \leftarrow Y, \quad \{u_{j,b} \leftarrow Y\}_{j\in[m],b\in\{0,1\}}.$$

In the next part, we describe a construction of functional hinting PRG for the quadratic function of the seed (where the seed is viewed a vector of bits) from the DDH assumption, i.e., it is possible to (securely) provide a hint with respect to $f(\mathbf{s})$ where $f : \{0,1\}^n \to \{0,1\}^{n^2}$ defined as $f(\mathbf{s}) = \mathbf{s} \otimes \mathbf{s}$, which can be viewed as a vectorized form of $\mathbf{s}\mathbf{s}^t \in \{0,1\}^{n\times n}$.

Functional Hinting PRG for Quadratic Function from DDH. Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and let n be an integer such that $n > 2\log|\mathbb{G}| + \omega(\log\lambda)$. We use the notation from Section 3.1 to show a construction of functional hinting PRG for the quadratic function based on the DDH assumption. Our construction of functional hinting PRG $G_{\mathsf{pp}} : \{0,1\}^n \to \mathbb{G}^{n^2+1}$ from DDH is as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^{n^2})$: For each $j \in \{0\} \cup [n^2]$, sample $[\mathbf{M}_j] \leftarrow \mathbb{G}^{n\times n}$ and publish $\mathsf{pp} = \left([\mathbf{M}_j]\right)_{j\in\{0\}\cup[n^2]}$.
- $\mathsf{Eval}(\mathsf{pp}, \mathsf{s} \in \{0,1\}^n, i \in \{0\} \cup [n^2])$: Let $[\mathbf{M}_i]$ denote the ith matrix from pp. Output $[\mathbf{s}^t\mathbf{M}_i\mathbf{s}]$.[1]

[1] Note that given any matrix of group elements $[\mathbf{M}] \in \mathbb{G}^{n\times n}$ and any binary vector $\mathbf{s} \in \{0,1\}^n$, one can efficiently compute $[\mathbf{s}^t\mathbf{M}\mathbf{s}]$.

Security. We prove the security of the construction via the following theorem.

Theorem 5.2. *If $(\mathbb{G}, g, q)$ is a DDH-hard group then the construction above yields a functional hinting PRG for the quadratic function from DDH.*

Proof. First, observe that by Lemma 5.3 (proved below) for $Q = n^2 + 1$ samples we have

$$\big([\mathbf{M}_j], [\mathbf{s}^t\mathbf{M}_j\mathbf{s}]\big)_{j\in[n^2+1]} \overset{c}{\approx} \big([\mathbf{M}_j], [u_j]\big)_{j\in[n^2+1]}$$

(where $[u_j] \leftarrow \mathbb{G}$ for each $j \in [n^2+1]$) and hence the pseudorandomness of the output in the plain PRG game follows from Lemma 5.3. Let $\alpha : [n^2] \to [n]$ and $\beta : [n^2] \to [n]$ be two simple index mapping functions that map any index $i \in [n^2]$ to $(\alpha(i) = \lceil i/n \rceil, \beta(i) = i \bmod n)$. Note that α and β simply provide a way to write a vector with n^2 elements as an $n \times n$ matrix.

To establish the security of the construction in the functional hinting PRG game, it is enough to show that

$$\big([\mathbf{M}_0], [\mathbf{s}^t\mathbf{M}_0\mathbf{s}], \big([\mathbf{M}_i]\big)_{i\in[n^2]}, [\mathbf{Y}]\big) \overset{c}{\approx} ([\mathbf{M}_0], [u], [\bar{\mathbf{M}}], [\mathbf{U}]), \qquad (\square)$$

where $[u] \leftarrow \mathbb{G}$ and $[\mathbf{U}] \leftarrow \mathbb{G}^{n^2\times 2}$ are sampled uniformly and $[\mathbf{Y}] \in \mathbb{G}^{n^2\times 2}$ is distributed as follows

$$\sigma(i) = s_{\alpha(i)} \cdot s_{\beta(i)}, \quad [y_{i,\sigma(i)}] = [\mathbf{s}^t\mathbf{M}_i\mathbf{s}], \quad [y_{i,1-\sigma(i)}] \leftarrow \mathbb{G}, \quad i \in [n^2].$$

Note that $\sigma(i)$ outputs the $(\alpha(i), \beta(i))$ entry of $\mathbf{s}\mathbf{s}^t \in \{0,1\}^{n\times n}$ for any index $i \in [n^2]$. We prove $(\square)$ via a hybrid argument. Let H_0 and H_1 be the hybrids that correspond to the left-hand side and right-hand side of $(\square)$, respectively.

Let $\mathcal{A}$ be an adversary that distinguishes H_0 from H_1. We construct an adversary $\mathcal{A}'$ that distinguishes H_0' from H_1' defined as

$$H_0' := \big([\mathbf{M}_0], [\mathbf{s}^t\mathbf{M}_0\mathbf{s}], \big([\mathbf{M}_i]\big)_{i\in[n^2]}, \mathbf{y}\big), \quad H_1' := \big([\mathbf{M}_0], [u], \big([\mathbf{M}_i]\big)_{i\in[n^2]}, \mathbf{u}\big),$$

where $[y_i] = [\mathbf{s}^t\mathbf{M}_i\mathbf{s}]$ for each $i \in [n^2]$, and by Lemma 5.3 it follows that the advantage of $\mathcal{A}$ should also be negligible.

Given a tuple $H_b' = \big([\mathbf{m}_0], [z_0], \big([\mathbf{M}_i]\big)_{i\in[n^2]}, [\mathbf{z}]\big)$, where H_b' is distributed as either H_0' or H_1', the external adversary $\mathcal{A}'$ forms n^2 matrices $[\mathbf{P}_{jk}] \in \mathbb{G}^{n\times n}$ (for $j \in [n], k \in [n]$) where $[\mathbf{P}_{jk}]$ is a matrix whose all but one entry is the identity element of the group and the remaining one entry at the position (j,k) is sampled uniformly from $\mathbb{G}$. Concretely, $\mathcal{A}'$ samples a shift vector $[\mathbf{d}] \in \mathbb{G}^{n^2}$, and it sets the $(\alpha(i), \beta(i))$ entry of $[\mathbf{P}_{\alpha(i),\beta(i)}]$ as $[d_i]$ for each $i \in [n^2]$. In the next step, $\mathcal{A}'$ runs $\mathcal{A}$ on the following tuple

$$([\mathbf{m}_0], [z_0], [\mathbf{M}_i'] := [\mathbf{M}_i + \mathbf{P}_{\alpha(i),\beta(i)}], [\mathbf{Y}]),$$

where $[\mathbf{Y}]$ is an n^2 by 2 matrix whose first and second columns are $[\mathbf{z}]$ and $[\mathbf{z} + \mathbf{d}]$ respectively. We define the output of $\mathcal{A}'$ to be the same as the output of $\mathcal{A}$.

Observe that (in the view of the adversary $\mathcal{A}$) $[\mathbf{M}_0]$ and $([\mathbf{M}_i'])_{i\in[n^2]}$ are distributed uniformly. Moreover, if $[\mathbf{z}]$ is uniform then $[\mathbf{Y}]$ will be distributed uniformly as well. Thus, $\mathcal{A}'$ perfectly simulates the "ideal" hybrid H_1. On the other hand, if $[z_i] = [\mathbf{s}^t\mathbf{M}_i\mathbf{s}]$ (for each $i \in [n^2]$) then from the view of $\mathcal{A}'$ the matrix $[\mathbf{Y}]$ is distributed as

$$\sigma(i) = s_{\alpha(i)} \cdot s_{\beta(i)}, \; [y_{i,\sigma(i)}] = [\mathbf{s}^t\mathbf{M}_i'\mathbf{s}], \; [y_{i,1-\sigma(i)}] = [(-1)^{\sigma(i)} \cdot d_i + \mathbf{s}^t\mathbf{M}_i'\mathbf{s}], \quad i \in [n^2].$$

Note that the relations above hold because

$$[\mathbf{s}^t\mathbf{M}_i'\mathbf{s}] = [\mathbf{s}^t\mathbf{M}_i\mathbf{s} + s_{\alpha(i)} \cdot s_{\beta(i)} \cdot d_i], \quad i \in [n^2].$$

Since $[\mathbf{d}]$ is distributed uniformly and independently from $[\mathbf{M}']$ (in the view of $\mathcal{A}$), it follows that

$$\big(([\mathbf{M}_i'])_{i\in[n^2]}, \mathbf{Y}\big) \overset{s}{\approx} \big(([\mathbf{M}_i'])_{i\in[n^2]}, \mathbf{U}\big),$$

where $[\mathbf{U}] \leftarrow \mathbb{G}^{n^2\times 2}$, and hence $\mathcal{A}'$ properly maps the hybrid H_0' to (a hybrid that is statistically indistinguishable from) H_0, as required. □

Lemma 5.3. *Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and fix some integer ℓ and n such that $n > 2\log|\mathbb{G}| + \omega(\log\lambda)$ and $\ell = \mathrm{poly}(\lambda)$. If $\{[\mathbf{M}_i] \leftarrow \mathbb{G}^{n\times n}\}_{i\in[\ell]}$ and $\mathbf{s} \leftarrow \{0,1\}^n$, then*

$$\big([\mathbf{M}_i], [\mathbf{s}^t\mathbf{M}_i\mathbf{s}]\big)_{i\in[\ell]} \overset{c}{\approx} \big([\mathbf{M}_i], [u_i]\big)_{i\in[\ell]},$$

where $[u_i] \leftarrow \mathbb{G}$ is sampled uniformly for each $i \in [\ell]$.

Proof. Let $\bar{\mathbf{M}} \in \mathbb{G}^{n\times n}$ be a matrix of group elements whose (j,k) entry is $[a_j \cdot b_k]$ where $a_j \leftarrow \mathbb{Z}_q, b_k \leftarrow \mathbb{Z}_q$ (for $j \in [\ell], k \in [n]$). In addition, let $([\hat{\mathbf{M}}_i])_{i\in[\ell]}$ be ℓ matrices of group elements defined as

$$[\hat{\mathbf{M}}_i] = [r_i \cdot \bar{\mathbf{M}}], \quad r_i \leftarrow \mathbb{Z}_q, \quad i \in [\ell].$$

By applying the leftover hash lemma to the group $\mathbb{G}^2$, it follows that

$$([\bar{\mathbf{M}}], [\mathbf{s}^t\bar{\mathbf{M}}\mathbf{s}]) \overset{s}{\approx} ([\bar{\mathbf{M}}], [u']),$$

where $[u'] \leftarrow \mathbb{G}$, which in turn implies that

$$\big([\hat{\mathbf{M}}_i], [\mathbf{s}^t\hat{\mathbf{M}}_i\mathbf{s}]\big)_{i\in[\ell]} \overset{s}{\approx} \big([\hat{\mathbf{M}}_i], [r_i \cdot u']\big)_{i\in[\ell]}, \overset{c}{\approx} \big([\hat{\mathbf{M}}_i], [u_i]\big)_{i\in[\ell]},$$

and the computational indistinguishability follows from the DDH assumption. On the other hand, by the DDH assumption we have

$$\big([\mathbf{M}_i]\big)_{i\in[\ell]} \overset{c}{\approx} \big([\hat{\mathbf{M}}_i]\big)_{i\in[\ell]},$$

and hence a standard hybrid argument implies that

$$\big([\mathbf{M}_i], [\mathbf{s}^t\mathbf{M}_i\mathbf{s}]\big)_{i\in[\ell]} \overset{c}{\approx} \big([\mathbf{M}_i], [u_i]\big)_{i\in[\ell]},$$

as required. □

Functional Hinting PRG for Higher Degree Functions. The above construction of functional hinting PRG allows us to publish a hint with respect to the function $g(\mathbf{s}) = \mathbf{s} \otimes \mathbf{s} \in \{0,1\}^{n^2}$. Here we describe a way to obtain functional hinting PRG for functions of higher degree. One can generalize the construction above for functions of higher degree $k > 2$ by using n^k many k-dimensional array/tensor of uniformly chosen group elements as the public parameter, and the evaluation will be shrinking down each array in the public parameter to only one group element by computing a $\mathbb{G}$-linear function across each dimension using the seed $\mathbf{s}$. For instance, given n^k many k-dimensional array of uniformly chosen group elements one can construct a functional hinting PRG for degree k functions where each of n^k blocks provides a hint with respect to $s_{i_1}s_{i_2}\cdots s_{i_k}$, for $(i_1,\ldots,i_k) \in [n]^k$. The construction and proof will be similar to the quadratic case, and hence we omit the details.

5.2 Functional Hinting weak PRF

Similar to the case of hinting PRG, we define a generalized version of hinting weak PRF for which the security game is defined in terms of function(s) of the secret key, rather the key itself. Our notion of hinting weak PRF can be viewed as a functional hinting weak PRF with respect to the identity function. There are two approaches to define a functional hinting weak PRF; one approach is to guarantee security in the presence of multiple hints of a *fixed* function of the secret key (corresponding to different inputs), and another approach is to provide security in the presence of multiple hints of *different* functions of the secret key. We provide a formal definition of the latter in this section, and later we provide an instantiation based on DDH for certain family of functions.

Definition 5.4. Let $\mathcal{F} = \{f_I \mid f_I : \{0,1\}^n \to \{0,1\}^m\}_{I\in\mathcal{I}}$ be a family of boolean functions, and let $F : K \times X \to \bar{Y}$ be a weak PRF where $K = \{0,1\}^n$ and $\bar{Y} = Y^m$ for some efficiently samplable set Y. We say that F is a functional hinting weak PRF with respect to $\mathcal{F}$ if the advantage of any PPT attacker in distinguishing between the experiments $\mathsf{Exp}_0^{\mathsf{FHwPRF}}$ and $\mathsf{Exp}_1^{\mathsf{FHwPRF}}$ (described in Figure 2) is negligible.

1. The challenger samples a weak PRF key $\mathbf{k} \leftarrow \{0,1\}^n$.
2. The adversary chooses a function $f_i \in \mathcal{F}$ (corresponding to the ith query) and sends it to the challenger.
3. The challenger samples $x_i \leftarrow X$, $\mathbf{r}^{(i)} \leftarrow Y^m$, $\mathbf{U}_i \leftarrow Y^{m\times 2}$ uniformly. It then sets $\mathbf{y}^{(i)} = F(\mathbf{k}, x_i)$. Let $\mathsf{S}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)})$ be an m by 2 "selector" matrix with respect to $f_i(\mathbf{k})$ defined as follows:
$$\mathbf{v}^{(i)} = f_i(\mathbf{k}),\ \mathsf{S}_{j,v_j^{(i)}}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = y_j^{(i)},\ \mathsf{S}_{j,1-v_j^{(i)}}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = r_j^{(i)}, \quad j \in [m].$$
4. If $b = 0$, the challenger responds to the ith query with $\big(x_i, \mathsf{S}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)})\big)$.
5. If $b = 1$, the challenger responds to the ith query with $(x_i, \mathbf{U}_i)$.
6. The adversary continues to make function queries as before, and each query is replied by the challenger as described above.

Figure 2: Experiment $\mathsf{Exp}_b^{\mathsf{FHwPRF}}$with respect to $\mathcal{F}$

For a (boolean) function $g : \{0,1\}^n \to \{0,1\}^m$ we define the projective function family $\mathcal{F}_g$ as follows:
$$\mathcal{F}_g = \{f : \{0,1\}^n \to \{0,1\}^m \mid \exists \mathbf{b} \in \{0,1\}^m : f(\mathbf{x}) = (b_1 \cdot g_1(\mathbf{x}), \ldots, b_m \cdot g_m(\mathbf{x}))\},$$
where $g_i(\mathbf{x})$ denotes the ith bit of $g(\mathbf{x})$ and the condition holds for all $\mathbf{x} \in \{0,1\}^n$. We may drop the subscript g for the sake of simplicity when the function is clear from context. Informally, $\mathcal{F}$ contains all of the functions whose ith bit of the output (on any input) is either 0 or the ith output bit of g (on the same input). Note that given the function g, each function in $\mathcal{F}$ can be described by a binary vector $\mathbf{b}$. For instance, the function g itself corresponds to all-one vector $\mathbf{1}$.

In the next part of this section, we show a construction of functional hinting weak PRF for the family of projective quadratic functions based on the DDH assumption. Later, we describe how we can generalize this construction to the family of projective functions of higher degree. We note that a functional hinting weak PRF for the family of projective quadratic functions can be viewed as an extended version of a functional hinting PRG for the quadratic function $g(\mathbf{s}) = \mathbf{s} \otimes \mathbf{s}$, with an additional property that an adversary can adaptively "fix" the hint for arbitrary positions. Below, we describe a construction of functional hinting weak PRF for the family of projective quadratic functions $\mathcal{F}_g$ (as defined above) based on the DDH assumption.

Functional Hinting weak PRF for Projective Quadratic Functions. Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and let $n > 2\log|\mathbb{G}| + \omega(\log\lambda)$ be an integer. We use the notation from Section 3.1 to show a construction of functional hinting weak PRF. Consider the weak PRF $F : \{0,1\}^n \times (\mathbb{G}^{n\times n})^{n^2} \to \mathbb{G}^{n^2}$ defined as follows:

- $\mathsf{Gen}(1^\lambda)$: To generate a key, sample $\mathbf{k} \leftarrow \{0,1\}^n$.
- $F(\mathbf{k} = \{0,1\}^n, ([\mathbf{M}_i])_{i\in[n^2]} \in (\mathbb{G}^{n\times n})^{n^2})$: Output $([\mathbf{k}^t\mathbf{M}_i\mathbf{k}])_{i\in[n^2]}$.

Security. We prove the security of the construction via the following theorem.

Theorem 5.5. *If $(\mathbb{G}, g, q)$ is a DDH-hard group then the construction above yields a functional hinting weak PRF for the projective quadratic function family $\mathcal{F}_g$ from DDH.*

Proof. Weak pseudorandomness of F (in the plain weak PRF game) follows from Lemma 5.3. To establish the functional hinting security (with respect to $\mathcal{F}_g$) we need to prove that $\mathsf{Exp}_0^{\mathsf{FHwPRF}} \overset{c}{\approx} \mathsf{Exp}_1^{\mathsf{FHwPRF}}$. To show this, we extend the proof of DDH-based functional hinting PRG for quadratic function to multiple instances by keeping track of each function f_i (determined by $\mathbf{b}_i$). As mentioned before, a binary vector $\mathbf{b}_i \in \{0,1\}^{n^2}$ can be used to describe any function $f_i \in \mathcal{F}_g$ (along with g). First, by Lemma 5.3 for any $Q = \mathrm{poly}(\lambda)$ we have

$$H_0 := \Big(\big([\mathbf{M}_i^{(\ell)}] \big)_{\ell \in [n^2]}, \big([\mathbf{k}^t \mathbf{M}_i^{(\ell)} \mathbf{k}] \big)_{\ell \in [n^2]} \Big)_{i \in [Q]} \overset{c}{\approx}$$
$$H_1 := \Big(\big([\mathbf{M}_i^{(\ell)}] \big)_{\ell \in [n^2]}, [\mathbf{u}_i] \Big)_{i \in [Q]},$$

where $[\mathbf{u}_i] \leftarrow \mathbb{G}^{n^2}$. Let $\mathcal{A}$ be an adversary that distinguishes $\mathsf{Exp}_0^{\mathsf{FHwPRF}}$ from $\mathsf{Exp}_1^{\mathsf{FHwPRF}}$, and let Q be the total of queries made by $\mathcal{A}$. We construct an adversary $\mathcal{A}'$ to distinguish H_0 from H_1. Given samples of the form

$$H_b := \Big(\big([\mathbf{M}_i^{(\ell)}] \big)_{\ell \in [n^2]}, [\mathbf{z}_i] \Big)_{i \in [Q]}$$

where H_b is distributed as either H_0 or H_1, the adversary $\mathcal{A}'$ runs $\mathcal{A}$. Whenever $\mathcal{A}$ makes its ith query for a function $f_i \in \mathcal{F}_g$ determined by a binary vector $\mathbf{b}_i \in \{0,1\}^{n^2}$, the adversary $\mathcal{A}'$ responds the ith query as follows. $\mathcal{A}'$ samples $[\mathbf{d}_i] \leftarrow \mathbb{G}^{n^2}$. Let α and β be the index mapping functions from the proof of Theorem 5.2. For $\ell \in [n^2]$, the adversary $\mathcal{A}'$ sets

$$[\bar{\mathbf{M}}_i^{(\ell)}] := [\mathbf{M}_i^{(\ell)}] + [b_i^{(\ell)} \cdot d_i^{(\ell)} \cdot \mathbf{E}_{\alpha(\ell),\beta(\ell)}],$$

where $\mathbf{E}_{\alpha(\ell),\beta(\ell)}$ is an $n \times n$ matrix whose $(\alpha(\ell), \beta(\ell))$ entry is 1, and all other entries are 0. (Note that $\bar{b}_i^{(\ell)}$ and $d_i^{(\ell)}$ denote the ℓth component of $\mathbf{b}_i$ and $\mathbf{d}_i$, respectively.)

$\mathcal{A}'$ sends $\big(([\bar{\mathbf{M}}_i^{(\ell)}])_{\ell \in [n^2]}, [\mathbf{Y}_i] \big)$ to $\mathcal{A}$ as the response for the ith query, where $[\mathbf{Y}_i] \in \mathbb{G}^{n^2 \times 2}$ is the matrix whose first and columns are $[\mathbf{z}^{(i)}]$ and $[\mathbf{d}^{(i)} + \mathbf{z}^{(i)}]$. We now argue that $\mathcal{A}'$ properly maps H_b to $\mathsf{Exp}_b^{\mathsf{FHwPRF}}$ for $b \in \{0,1\}$. First, we consider the simpler case $b = 1$. Observe that the matrices $\big([\bar{\mathbf{M}}_i^{(\ell)}] \big)_{\ell \in [n^2], i \in [Q]}$ are uniformly distributed in the view of $\mathcal{A}$. Moreover, if $([\mathbf{z}_i])_{i \in [Q]}$ are distributed uniformly and independently (which happens when $b = 1$), then $([\mathbf{Y}_i])_{i \in [Q]}$ will be uniformly distributed as well and hence $\mathcal{A}'$ properly maps H_1 to $\mathsf{Exp}_1^{\mathsf{FHwPRF}}$.

If $b = 0$, based on an argument similar to the proof of DDH-based hinting PRG for the quadratic function, it can be verified that for each $i \in [Q]$ we have

$$[\mathbf{Y}_i] = \mathsf{S}(f_i, [\mathbf{y}^{(i)}], [\mathbf{u}^{(i)}]),$$

where S is the "selector" mapping (as defined in the experiment) and

$$\mathbf{v}^{(i)} := f_i(\mathbf{k}) = \mathbf{b}_i \odot g(\mathbf{k}) = \mathbf{b}_i \odot (\mathbf{k} \otimes \mathbf{k}),$$
$$[\mathbf{y}^{(i)}] := \big([\mathbf{k}^t \bar{\mathbf{M}}_i^{(\ell)} \mathbf{k}] \big)_{\ell \in [n^2]}, \quad [\mathbf{u}^{(i)}] := [(-1)^{\mathbf{v}^{(i)}} \odot \mathbf{d}^{(i)} + \mathbf{y}^{(i)}],$$
$$\mathsf{S}_{j, v_j^{(i)}}\big(f_i, [\mathbf{y}^{(i)}], [\mathbf{u}^{(i)}]\big) = y_j^{(i)}, \quad \mathsf{S}_{j, 1-v_j^{(i)}}\big(f_i, [\mathbf{y}^{(i)}], [\mathbf{u}^{(i)}]\big) = u_j^{(i)}, \quad j \in [n^2],$$

where $\odot$ denotes the component-wise/Hadamard product and $(-1)^{\mathbf{v}(i)}$ is the vector obtained by component-wise exponentiation. It follows that in the view of the adversary $\mathcal{A}$

$$\big(([\bar{\mathbf{M}}_i^{(\ell)}])_{\ell \in [n^2]}, [\mathbf{Y}_i] \big)_{i \in [Q]} \overset{s}{\approx} \mathsf{S}\big(f_i, [\mathbf{y}^{(i)}], [\mathbf{r}^{(i)}]\big)_{i \in [Q]},$$

where $[\mathbf{r}_i] \leftarrow \mathbb{G}^{n^2}$. Therefore, $\mathcal{A}'$ properly maps the hybrid H_0 to (a hybrid that is statistically indistinguishable from) $\mathsf{Exp}_0^{\mathsf{FHwPRF}}$, as required. □

Functional Hinting weak PRF for Higher Degree Function Families. The construction above allows (securely) publishing many hints with respect to the projective function family $\mathcal{F}_g$ where $g(\mathbf{s}) = \mathbf{s} \otimes \mathbf{s} \in \{0,1\}^{n^2}$. Similar to the case of hinting PRG, we briefly describe how to construct functional hinting weak PRF for the projective function family $\mathcal{F}_h$ (where h is degree k function for some $k > 2$), which enables publishing a hint in each block with respect to a projective function of $s_{i_1} s_{i_2} \cdots s_{i_k}$, for $(i_1, \ldots, i_k) \in [n]^k$. Similar to the case of functional hinting PRG, a generalized version of the construction above can be obtained using n^k many k-dimensional array/tensor of uniformly chosen group elements for each input, and the output of of F is obtained by computing a $\mathbb{G}$-linear function across each dimension using the weak PRF key $\mathbf{k}$.

5.3 Circular/KDM Security from Hinting weak PRF

We now show a construction of symmetric-key circular/KDM-secure encryption scheme from any hinting weak PRF. We note that Kitagawal *et al.* [KMT19a] demonstrated a construction of *one-time* symmetric-key KDM-secure encryption scheme from any hinting PRG. In our construction, we do not have one-time restriction and an adversary can see polynomially many encryptions of (function of) the secret key.

First, we recall the definition of symmetric-key circular-secure encryption. Note that in the definition below, we assume that the key space is a subset of message space (which is satisfied by our construction). One can also alternatively consider a definition in which each ciphertext encrypts a bit or a part of the secret key. The former is desirable in certain applications, where a single ciphertext can be used to encrypt all bits of the secret key.

Definition 5.6. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Enc})$ be a symmetric-key encryption scheme with $\mathcal{M} = \mathcal{K} = \{0,1\}^n$, where $\mathcal{M}$ and $\mathcal{K}$ denote the message space and the key space, respectively. We say that Π is circular secure (with respect to multiple encryptions) if for any $Q = \mathrm{poly}(\lambda)$ it holds that

$$\big(\mathsf{Enc}(\mathsf{sk}, \mathsf{sk}; r_i)\big)_{i\in[Q]} \overset{c}{\approx} \big(\mathsf{Enc}(\mathsf{sk}, 0^n; r_i)\big)_{i\in[Q]},$$

where $\mathsf{sk} \leftarrow \{0,1\}^n$ and each ciphertext is generated using a fresh and independent randomness r_i.

Note that *one-time* circular security is defined similarly where the attacker gets to see only one encryption of the secret key, i.e., $Q = 1$.

Construction. Let $F : K = \{0,1\}^n \times X \to \bar{Y} = Y^n$ be a hinting weak PRF. Below, we demonstrate a construction of symmetric-key circular-secure encryption scheme with $\mathcal{M} = \mathcal{K} = \{0,1\}^n$ based on F.

- $\mathsf{Gen}(1^\lambda)$: To generate a secret key sample $\mathbf{k} \leftarrow \{0,1\}^n$.
- $\mathsf{Enc}(\mathbf{k}, \mathbf{m} = (m_1, \ldots, m_n)) \in \{0,1\}^n)$: Sample $x \leftarrow X$ and $\mathbf{u} \leftarrow Y^n$ and let $\mathbf{y} = F(\mathbf{k}, x)$. Publish $\mathsf{ct} = (x, \mathbf{C}) \in X \times Y^{n\times 2}$ as the ciphertext where

$$c_{i,m_i} = \begin{cases} y_i & m_i = 0, \\ u_i & m_i = 1, \end{cases} \qquad c_{i,1-m_i} = \begin{cases} u_i & m_i = 0, \\ y_i & m_i = 1. \end{cases}$$

- $\mathsf{Dec}(\mathbf{k}, \mathsf{ct} = (x, \mathbf{C}) \in X \times Y^{n\times 2})$: Compute $\mathbf{y} = F(\mathbf{k}, x)$. For each $i \in [n]$, if $y_i = c_{i,b}$ set $m_i = b$. Output $\mathbf{m} = (m_1, \ldots, m_n)$. (If the equality check fails for some $i \in [n]$, output $\perp$).

Security. The circular security of the scheme above (for multiple encryption of the secret key) is proved via the following theorem.

Theorem 5.7. *If $F : K = \{0,1\}^n \times X \to \bar{Y} = Y^n$ is a hinting weak PRF, then the scheme above is CPA secure and it satisfies circular security.*

Proof. CPA security of the scheme follows from weak pseudorandomness of F. To prove circular security, it suffices to show that for any polynomial $Q = \mathrm{poly}(\lambda)$

$$\big((x_i, \mathbf{C}_i)\big)_{i\in[Q]} \overset{c}{\approx} \big((x_i, \mathbf{U}_i)\big)_{i\in[Q]},$$

where $\mathbf{U}_i \leftarrow Y^{n\times 2}$ and each $(x_i, \mathbf{C}_i)$ is a fresh encryption of the secret key $\mathbf{k}$. Observe that by the construction above each $\mathbf{C}_i$ is distributed as the output of "selector matrix" S (with respect to the secret key $\mathbf{k}$, see Definition 4.1) on two vectors $\mathbf{y}^{(i)} = F(\mathbf{k}, x_i)$ and $\mathbf{u}^{(i)} \leftarrow Y^n$, i.e., $\mathbf{C}_i = \mathsf{S}(\mathbf{y}^{(i)}, \mathbf{u}^{(i)})$ with respect to $\mathbf{k}$. Therefore, the hinting security property of F implies that

$$\big((x_i, \mathbf{C}_i)\big)_{i\in[Q]} \overset{c}{\approx} \big((x_i, \mathbf{U}_i)\big)_{i\in[Q]}.$$

On the other hand, the weak pseudorandomness of F implies that if $\{(x_i, \mathbf{Z}_i) \leftarrow \mathsf{Enc}(\mathbf{k}, 0^n; x_i)\}$ then

$$\big((x_i, \mathbf{U}_i)\big)_{i\in[Q]} \overset{c}{\approx} \big((x_i, \mathbf{Z}_i)\big)_{i\in[Q]},$$

and hence it follows that

$$\big(\mathsf{Enc}(\mathsf{sk}, \mathsf{sk}; x_i)\big)_{i\in[Q]} \overset{c}{\approx} \big(\mathsf{Enc}(\mathsf{sk}, 0^n; x_i)\big)_{i\in[Q]}.$$

□

Remark 5.8. We remark that one can also consider a slightly modified version of the construction for which one element from Y is published per each bit of the message, i.e., the encryption algorithm only publishes the first column of $\mathbf{C}$ (along with x). It is immediate to see that the modified construction also satisfies circular security. However, we presented the version above because it naturally corresponds to the security game of a hinting weak PRF.

Realizing KDM Security. We briefly describe two approaches to realize (symmetric-key) KDM security with respect to apriori bounded-size circuits from hinting weak PRF. Our first approach is to simply extend the construction of one-time KDM-secure scheme based on hinting PRG from [KMT19a] to a construction of (many-time) KDM-secure scheme based on hinting weak PRF. This is done by relying on the security of hinting weak PRF to (securely) provide multiple encryption of (functions of) the secret key. The construction and proof would be quite analogous to their setting, and hence we omit the details. As pointed in [KMT19a], the idea is to mask labels of garbled circuits using n blocks of output of a primitive with hinting security property (which is PRG in their work and weak PRF in ours).[1]

An alternative path is to use the generic amplification technique of [App14] to realize KDM security with respect to apriori bounded-size circuits from KDM security with respect to projection functions. It can be easily verified that the construction above also provides security for (multiple) encryptions of $k_i\mathbf{e}_i$ or $(1-k_i)\mathbf{e}_i$, where k_i denotes the ith bit of the secret key and $\mathbf{e}_i$ is the ith unit vector. Thus, based on the amplification results of [App14], we get a construction of KDM-secure SKE (with respect to bounded-size circuits) from any hinting weak PRF.

5.4 $\mathcal{F}$-KDM Security from Functional Hinting weak PRF

In this section, we show that any functional hinting weak PRF with respect to a function family $\mathcal{F}$ immediately implies a symmetric-key $\mathcal{F}$-KDM secure encryption scheme in a black-box way.

Construction. Let $F : K \times X \to \bar{Y}$ be a functional hinting weak PRF with respect to the function family $\mathcal{F} = \{f_I : \{0,1\}^n \to \{0,1\}^m\}_{I\in\mathcal{I}}$ where $K = \{0,1\}^n$ and $\bar{Y} = Y^m$. Our construction of $\mathcal{F}$-KDM secure SKE is identical to that of circular-secure SKE from (plain) hinting weak PRF with a minor difference that the message space is determined by the codomain of functions in $\mathcal{F}$, rather than the length of secret key. Below, we recall the encryption algorithm, the key generation and decryption algorithms are identical to those of Section 5.3.

- $\mathsf{Enc}(\mathbf{k} \in \{0,1\}^n, \boldsymbol{\mu} = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m)$: Sample $x \leftarrow X$ and $\mathbf{u} \leftarrow Y^m$ and let $\mathbf{y} = F(\mathbf{k}, x)$. Publish $\mathsf{ct} = (x, \mathbf{C}) \in X \times Y^{m\times 2}$ as the ciphertext where

$$c_{i,\mu_i} = \begin{cases} y_i & \mu_i = 0, \\ u_i & \mu_i = 1, \end{cases} \qquad c_{i,1-\mu_i} = \begin{cases} u_i & \mu_i = 0, \\ y_i & \mu_i = 1. \end{cases}$$

Theorem 5.9. *If $F : K \times X \to \bar{Y}$ is a functional hinting weak PRF with respect to $\mathcal{F} = \{f_I : \{0,1\}^n \to \{0,1\}^m\}_{I\in\mathcal{I}}$ where $K = \{0,1\}^n$ and $\bar{Y} = Y^m$, then the scheme described above satisfies $\mathcal{F}$-KDM security.*

Proof. The proof is similar to the proof of Theorem 5.7, and here we sketch an argument. First, observe that CPA security of the scheme follows immediately from the weak pseudorandomness of F. To argue KDM security (with respect to $\mathcal{F}$), observe that for any adversary making $Q = \mathrm{poly}(\lambda)$ adaptive queries $(f_i)_{i\in[Q]}$, the query-response pairs

[1] Note that the construction of [KMT19a] needs $n+1$ blocks whereas our definition of hinting weak PRF has n blocks as its output. However, this issue can simply be solved by evaluating the weak PRF on a fresh random input and treating the evaluation output as the block that corresponds to the index 0.

have the form $(x_i, \mathbf{C}_i)$ where $x_i \leftarrow X$ and $\mathbf{C}_i = \mathsf{S}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)})$, and the latter is the selector matrix (see Definition 5.4) with respect to $f_i(\mathbf{k})$, which is distributed as:

$$\mathbf{v}^{(i)} = f_i(\mathbf{k}),\ \mathsf{S}_{j,v_j^{(i)}}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = y_j^{(i)},\ \mathsf{S}_{j,1-v_j^{(i)}}(f_i, \mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = r_j^{(i)}, \quad j \in [m].$$

By the functional hinting property of F, it follows that

$$\big((x_i, \mathbf{C}_i)\big)_{i\in[Q]} \overset{c}{\approx} \big((x_i, \mathbf{U}_i)\big)_{i\in[Q]},$$

where each $\mathbf{U}_i \leftarrow Y^{m\times 2}$ is sampled uniformly. On the other hand, the weak pseudorandomness of F implies if $\{(x_i, \mathbf{Z}_i) \leftarrow \mathsf{Enc}(\mathbf{k}, 0^m; x_i)\}$ then

$$\big((x_i, \mathbf{U}_i)\big)_{i\in[Q]} \overset{c}{\approx} \big((x_i, \mathbf{Z}_i)\big)_{i\in[Q]},$$

and hence it follows that

$$\big((x_i, \mathbf{C}_i)\big)_{i\in[Q]} \overset{c}{\approx} \big((x_i, \mathbf{Z}_i)\big)_{i\in[Q]}.$$

□

$\mathcal{F}$-KDM Security for PKE. While a functional hinting weak PRF (with respect to some function family $\mathcal{F}$) implies a symmetric-key $\mathcal{F}$-KDM secure encryption scheme in a black-box manner, it is natural to also ask for a (black-box) construction of $\mathcal{F}$-KDM secure PKE based on a more "structured" functional weak PRF. Indeed, it can be easily verified that the DDH-based construction of functional hinting weak PRF with respect to projective quadratic functions (or higher degree functions) is homomorphic with respect to the input space, and we can exploit this property (by relying on techniques from [AMPR19]) to get a construction of $\mathcal{F}$-KDM secure *public-key* encryption from any functional hinting weak PRF that additionally satisfies input homomorphism. As a concrete example, below we describe a black-box construction of DDH-based KDM-secure PKE (with respect to projective quadratic function family), but we sketch a short proof since the analysis is quite similar to the symmetric-key case. A construction for functions of higher degree can be obtained similar to the construction of functional weak PRF for functions of higher degree from DDH.

Construction. Let $(\mathbb{G}, g, q)$ be a DDH-hard group, and fix some integer m and n such that $n > 2\log|\mathbb{G}| + \omega(\log\lambda)$ and $m = n^2$.

- $\mathsf{Gen}(1^\lambda)$: To generate a secret key sample $\mathbf{k} \leftarrow \{0,1\}^n$. Sample m matrices $([\mathbf{M}_i] \leftarrow \mathbb{G}^{n\times n})_{i\in[m]}$, and set $[y_i] = [\mathbf{k}^t\mathbf{M}_i\mathbf{k}]$ for $i \in [m]$. Output $(\mathsf{sk}, \mathsf{pk})$ as
$$\mathsf{sk} = \mathbf{k}, \quad \mathsf{pk} = \big(([\mathbf{M}_i])_{i\in[m]}, [\mathbf{y}]\big).$$
- $\mathsf{Enc}(\mathsf{pk}, \boldsymbol{\mu} = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m)$: Sample $(\mathbf{r}_i \leftarrow \{0,1\}^m)_{i\in[m]}$ and also $[\mathbf{u}] \leftarrow \mathbb{G}^m$. For each $i \in [m]$ compute
$$[\mathbf{M}_i^*] := \Big[\sum_{j=1}^m r_i^{(j)} \cdot \mathbf{M}_j\Big], \quad [y_i^*] := \Big[\sum_{j=1}^m r_i^{(j)} \cdot y_j\Big],$$
where $r_i^{(j)}$ denotes the jth bit of the vector $\mathbf{r}_i$. Output $\mathsf{ct} = \big(([\mathbf{M}_i^*])_{i\in[m]}, \mathbf{C}\big) \in (\mathbb{G}^{n\times n})^m \times \mathbb{G}^{m\times 2}$ where
$$c_{i,\mu_i} = \begin{cases} [y_i^*] & \mu_i = 0, \\ [u_i] & \mu_i = 1, \end{cases} \qquad c_{i,1-\mu_i} = \begin{cases} [u_i] & \mu_i = 0, \\ [y_i^*] & \mu_i = 1. \end{cases}$$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} = \big(([\mathbf{M}_i^*])_{i\in[m]}, \mathbf{C}\big))$: For each $i \in [m]$, compute $[y_i'] = [\mathbf{k}^t\mathbf{M}_i^*\mathbf{k}]$. If $y_i' = c_{i,b}$ set $\mu_i = b$. Output $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$. (If the equality check fails for some $i \in [m]$, output $\perp$).

Proof sketch. Let $g(\mathbf{k}) = \mathbf{k} \otimes \mathbf{k} \in \{0,1\}^m$, and let $\mathcal{F}_g$ be the family of projective quadratic functions. For any adversary with Q adaptive queries $(f_q \in \mathcal{F}_g)_{q\in[Q]}$, let the tuple $([\bar{\mathbf{M}}_i^{(q)}], [\bar{\mathbf{C}}^{(q)}])_{q\in[Q]}$ be Q ciphertexts such that

$$([\bar{\mathbf{M}}_i^{(q)}], [\bar{\mathbf{C}}^{(q)}]) \leftarrow \mathsf{Enc}(\mathsf{pk}, f_q(\mathbf{k})),$$

where $Q = \mathrm{poly}(\lambda)$. Observe that by Lemma 5.3 we have

$$(\mathsf{pk}, [\bar{\mathbf{M}}_i^{(q)}], [\bar{\mathbf{C}}^{(q)}])_{q\in[Q]} \overset{c}{\approx} (\mathsf{pk}, [\mathbf{M}_i^{(q)}], [\mathbf{C}^{(q)}])_{q\in[Q]},$$

where $[\mathbf{M}_i^{(q)}] \leftarrow \mathbb{G}^{n\times n}$ and each $[\mathbf{C}^{(q)}]$ is distributed as

$$\mathbf{v}^{(q)} := f_q(\mathbf{k}), \quad c^{(q)}_{i,v_i^{(q)}} = \begin{cases} [\mathbf{k}^t\mathbf{M}_i^{(q)}\mathbf{k}] & v_i^{(q)} = 0, \\ [u_i] \leftarrow \mathbb{G} & v_i^{(q)} = 1, \end{cases} \qquad c^{(q)}_{i,1-v_i^{(q)}} = \begin{cases} [u_i] \leftarrow \mathbb{G} & v_i^{(q)} = 0, \\ [\mathbf{k}^t\mathbf{M}_i^{(q)}\mathbf{k}] & v_i^{(q)} = 1. \end{cases}$$

Therefore, it is enough to show that

$$\begin{aligned} &([\mathbf{M}_i], [\mathbf{k}^t\mathbf{M}_i\mathbf{k}])_{i\in[m]}, ([\mathbf{M}_i^{(q)}], [\mathbf{C}^{(q)}])_{q\in[Q]} \overset{c}{\approx} \\ &([\mathbf{M}_i], [u_i])_{i\in[m]}, ([\mathbf{M}_i^{(q)}], [\mathbf{U}^{(q)}])_{q\in[Q]}, \end{aligned} \qquad (*)$$

where $[u_i] \leftarrow \mathbb{G}$ for $i \in [m]$, and $\mathbf{U}^{(q)} \leftarrow \mathbb{G}^{m\times 2}$ for $q \in [Q]$. Finally, it follows by Theorem 5.5 that the indistinguishability above $(*)$ holds, as required.

6 Realizing Hinting Property from Random Oracles

In this section, we investigate the complexity of cryptographic primitives with hinting property, and we show that a hinting (weak) PRF can be realized in the *random oracle model* (ROM). In fact, we show that the existing construction(s) of PRG and (weak) PRF in the random oracle model (with sufficiently large output length) already satisfies the hinting property. By a simple information-theoretic argument, we first show that why the (folklore) construction of PRG in the random oracle model satisfies the hinting property. We extend this solution to other primitives with hinting properties, namely weak and plain/strong PRFs.

Hinting PRG in the Random Oracle Model. Let $H : \{0,1\}^n \to Y^{n+1}$ be a truly random function (modeled as a random oracle), where Y is a sufficiently large set. As a concrete choice, the reader may assume that $Y = \{0,1\}^n$ but the argument is applicable for any set Y with superpolynomially large size, i.e., $|Y| = \lambda^{\omega(1)}$. We use $H_i(\mathbf{s})$ to the ith component of $H(\mathbf{s})$ for $i \in [n]$. It is well-known that one can treat H as a PRG in the random oracle model since any (computationally unbounded) attacker cannot distinguish between $H(\mathbf{s} \leftarrow \{0,1\}^n)$ and $\mathbf{u} \leftarrow Y^{n+1}$ with polynomially many queries to the function H. Let's consider H in the hinting PRG security game. Informally, an adversary should distinguish between two matrices $\mathbf{Y} \in Y^{n\times 2}$ and $\mathbf{U} \leftarrow Y^{n\times 2}$ where $y_{i,s_i} = H_i(\mathbf{s})$ and $y_{i,1-s_i} \leftarrow Y$.[1] First, observe that for any query $\mathbf{s}' \neq \mathbf{s}$ to the oracle H by the adversary, we know that $H(\mathbf{s}')$ is distributed uniformly and independently from $H(\mathbf{s})$. Thus, the only information that the adversary can gain about any bit of $\mathbf{s}$ is via $H(\mathbf{s})$. On the other hand (assuming that the adversary does not query $\mathbf{s}$), for all rows of $\mathbf{Y}$ we can argue that the joint distribution of y_{i,s_i} and $y_{i,1-s_i}$ is statistically indistinguishable from uniform distribution over Y^2, allowing us to argue the indistinguishability of $\mathbf{Y}$ and $\mathbf{U}$. We formalize this brief argument via the following lemma.

Lemma 6.1. *If $H : \{0,1\}^n \to Y^{n+1}$ is a function such that $n = \mathrm{poly}(\lambda)$ and Y is a superpolynomially large set (i.e., $|Y| = \lambda^{\omega(1)}$), then $H(\cdot)$ is a hinting PRG if H is modeled as a random oracle.*

Proof. Let $\mathcal{A}$ be any (computationally unbounded) adversary, and let $\mathcal{Q}$ be the set of all queries made by $\mathcal{A}$. Since $Q = \mathrm{poly}(\lambda)$, it follows that $\Pr[\mathbf{s} \in Q] \leq \mathrm{negl}$ (where $\mathbf{s}$ denotes the seed). Thus we simply focus on the event that

[1] For this brief argument, we ignore $H_0(\mathbf{s})$ for simplicity.

$\mathbf{s} \notin \mathcal{Q}$. Let $(y_0, \mathbf{Y}) \in Y^{n\times 2}$ be an element-matrix pair where $y_0 = H_0(\mathbf{s})$ and $\mathbf{Y}$ is distributed as defined above, i.e., $y_{i,s_i} = H_i(\mathbf{s})$ and $y_{i,1-s_i} \leftarrow Y$. Let $(\bar{y_0}, \bar{\mathbf{Y}}) \in Y \times Y^{n\times 2}$ be an arbitrary element-matrix pair. We will argue that in the view of $\mathcal{A}$, any element-matrix pair is equally likely to be equal to $(y_0, \mathbf{Y})$, conditioned on $\mathbf{s} \notin \mathcal{Q}$. Specifically, conditioned on $\mathbf{s} \notin \mathcal{Q}$, for any $\mathcal{A}$ we have

$$\begin{aligned}
\Pr[(y_0, \mathbf{Y}) = (\bar{y_0}, \bar{\mathbf{Y}})] &= \Pr[y_0 = \bar{y_0}] \cdot \prod_{i=1}^{n} \big(\Pr[y_{i,s_i} = \bar{y}_{i,s_i}] \cdot \Pr[y_{i,s_i} = \bar{y}_{i,1-s_i}] \big) \\
&= \Pr[H_0(\mathbf{s}) = y_0] \cdot \big(\prod_{i=1}^{n} \Pr[y_{i,1-s_i} = \bar{y}_{i,1-s_i}] \big) \cdot \big(\prod_{i=1}^{n} \Pr[H_i(\mathbf{s}) = \bar{y}_{i,s_i}] \big) \\
&= |Y|^{-n-1} \cdot \big(\prod_{i=1}^{n} \Pr[H_i(\mathbf{s}) = \bar{y}_{i,s_i}] \big) \\
&= |Y|^{-2n-1},
\end{aligned}$$

where the third line follows from the fact that $y_{i,1-s_i}$ is sampled uniformly in the game, and the last line follows from the fact that H is a random oracle. It follows that for any adversary $\mathcal{A}$ making at most $|\mathcal{Q}| = Q = \mathrm{poly}(\lambda)$ queries we have $(y_0, \mathbf{Y}) \overset{s}{\approx} (u_0, \mathbf{U})$, where $u_0 \leftarrow Y$ and $\mathbf{U} \leftarrow Y^{n\times 2}$, as required. □

Hinting PRF. We define a hinting PRF similarly to a hinting weak PRF, except that now the adversary is allowed to ask for hints with respect to the key of PRF for *arbitrarily* chosen inputs (instead of randomly chose ones). There is one minor subtlety that needs to be addressed. Specifically, an adversary should not be allowed to get multiple hints on the same input, since otherwise an attacker can immediately break the hinting security game. Below, we provide a definition of hinting PRF.

Definition 6.2. Let $F : K \times X \to \bar{Y}$ be a PRF where $K = \{0,1\}^n$ and $\bar{Y} = Y^n$ for some efficiently samplable set Y. We say that F is a hinting PRF if the advantage of any PPT attacker in distinguishing between the experiments $\mathsf{Exp}_0^{\mathsf{HPRF}}$ and $\mathsf{Exp}_1^{\mathsf{HPRF}}$ (described in Figure 3) is negligible.

1. The challenger samples a PRF key $\mathbf{k} \leftarrow \{0,1\}^n$. It also maintains a list of queries $\mathcal{Q}$, initially set to be empty.
2. The adversary queries for an input $x_i \in X$.
3. If $x_i \notin \mathcal{Q}$, the challenger adds x_i to $\mathcal{Q}$. It samples $\mathbf{r}^{(i)} \leftarrow Y^n$, $\mathbf{U}_i \leftarrow Y^{n\times 2}$ and sets $\mathbf{y}^{(i)} = F(\mathbf{k}, x_i)$. Let $\mathsf{S}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)})$ be an n by 2 "selector" matrix (with respect to $\mathbf{k}$) defined as follows:
$$\mathsf{S}_{j,k_j}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = y_j^{(i)}, \quad \mathsf{S}_{j,1-k_j}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)}) = r_j^{(i)}, \quad j \in [n].$$
4. If $b = 0$, the challenger responds to the ith query with $\big(x_i, \mathsf{S}(\mathbf{y}^{(i)}, \mathbf{r}^{(i)})\big)$.
5. If $b = 1$, the challenger responds to the ith query with $(x_i, \mathbf{U}_i)$.
6. The adversary continues to make input queries as before, and each query is replied by the challenger as described above.

Figure 3: Experiment $\mathsf{Exp}_b^{\mathsf{HPRF}}$

Hinting PRF in the Random Oracle Model. Expanding on the proof of Lemma 6.1, we show that even the stronger notion of hinting PRF can also be realized in the random oracle model. Let $H : \{0,1\}^{2n} \to Y^n$ be a truly random function (modeled as a random oracle), where Y is a sufficiently large set. For any two binary vectors $\mathbf{k} \in \{0,1\}^n$ and $\mathbf{x} \in \{0,1\}^n$, we use $H_i(\mathbf{k}, \mathbf{x})$ to denote the ith component of $H(\mathbf{k}, \mathbf{x})$.

Similar to the case of PRG, it is known that H (defined above) is a PRF if H is modeled as a random oracle. In the next lemma, we show that this construction also satisfies the hinting property, i.e., H is a hinting PRF if H is modeled as a random oracle. The proof is similar to the case of hinting PRG, with a difference that now two kinds of queries should be analyzed: (1) queries by the adversary to the random oracle, and (2) queries to the challenger in the hinting PRF game. In the hinting PRF security game, an adversary (making at most Q queries) should distinguish between $\mathcal{H}_0$ and $\mathcal{H}_1$, where $\mathcal{H}_0$ are $\mathcal{H}_1$ are defined as follows:

$$\mathcal{H}_0 := (\mathbf{Y}^{(\ell)})_{\ell \in [Q]}, \qquad \mathcal{H}_1 = (\mathbf{U}^{(\ell)})_{\ell \in [Q]},$$

$$y^{(\ell)}_{i,s_i} = H_i(\mathbf{k}, \mathbf{x}_\ell), \quad y_{i,1-s_i} \leftarrow Y, \quad \mathbf{U}^{(\ell)} \leftarrow Y^{n \times 2}, \quad \ell \in [Q], i \in [n].$$

Observe that for any query $(\mathbf{k}', \mathbf{x})$ to the oracle H by the adversary, if $\mathbf{k} \neq \mathbf{k}'$ then we know that $H(\mathbf{k}', \cdot)$ does not provide any information on $\mathbf{k}$. By an argument similar to the case of hinting PRG, we show that queries of the second kind do not leak information about $\mathbf{k}$, and hence the hinting property follows. We formalize this argument via the following lemma.

Lemma 6.3. *Let $H : \{0,1\}^{2n} \to Y^n$ be a function such that $n = \text{poly}(\lambda)$ and Y is a superpolynomially large set (i.e., $|Y| = \lambda^{\omega(1)}$), then $H(\cdot, \cdot)$ is a hinting PRF if H is modeled as a random oracle, where the first and second n bits denote the key space K and input space X, respectively.*

Proof. Let $\mathcal{A}$ be any (computationally unbounded) adversary, and let $\mathcal{Q}^O$ be the set of all random oracle queries made by $\mathcal{A}$. Let $\mathcal{Q}^{\mathsf{HP}} = \{x_i\}_{i \in [Q]}$ be the set of all *distinct* queries by $\mathcal{A}$ in the hinting PRF security game. By a simple union bound, because $|\mathcal{Q}^O| + Q \leq \text{poly}(\lambda)$ it follows that

$$\Pr[(\mathbf{k}, \mathbf{x}') \in \mathcal{Q}^O \text{ for some } x' \in X] \leq \text{negl}.$$

Thus, we focus our analysis conditioned on the event that none of oracle queries by $\mathcal{A}$ starts with $\mathbf{k}$. Conditioned on this event, it suffices to prove that $\mathcal{H}_0 \overset{s}{\approx} \mathcal{H}_1$ where

$$\mathcal{H}_0 := (\mathbf{Y}^{(\ell)})_{\ell \in [Q]}, \qquad \mathcal{H}_1 = (\mathbf{U}^{(\ell)})_{\ell \in [Q]},$$

$$y^{(\ell)}_{i,k_i} = H_i(\mathbf{k}, \mathbf{x}_\ell), \quad y^{(\ell)}_{i,1-k_i} \leftarrow Y, \quad \mathbf{U}^{(\ell)} \leftarrow Y^{n \times 2}, \quad \ell \in [Q], i \in [n].$$

In the next step, similar to the case of hinting PRG, we argue that conditioned on the event above, any tuple of ℓ matrices $(\bar{\mathbf{Y}}^{(\ell)})_{\ell \in [Q]}$ is equally likely to be equal to $(\mathbf{Y}^{(\ell)})_{\ell \in [Q]}$. Specifically, we have

$$\begin{aligned}
\Pr[(\mathbf{Y}^{(\ell)})_{\ell \in [Q]} = (\bar{\mathbf{Y}}^{(\ell)})_{\ell \in [Q]}] &= \prod_{i=1}^{n} \Big(\Pr[\forall \ell \in [Q] : y^{(\ell)}_{i,k_i} = \bar{y}^{(\ell)}_{i,k_i}] \cdot \Pr[\forall \ell \in [Q] : y^{(\ell)}_{i,k_i} = \bar{y}^{(\ell)}_{i,1-k_i}] \Big) \\
&= \Big(\prod_{i=1}^{n} \Pr[\forall \ell \in [Q] : y^{(\ell)}_{i,1-k_i} = \bar{y}^{(\ell)}_{i,1-k_i}] \Big) \cdot \big(\prod_{i=1}^{n} \Pr[\forall \ell \in [Q] : H_i(\mathbf{k}, \mathbf{x}_\ell) = \bar{y}^{(\ell)}_{i,k_i}] \big) \\
&= |Y|^{-\ell n} \cdot \big(\prod_{i=1}^{n} \Pr[\forall \ell \in [Q] : H_i(\mathbf{k}, \mathbf{x}_\ell) = \bar{y}^{(\ell)}_{i,k_i}] \big) \\
&= |Y|^{-2\ell n},
\end{aligned}$$

where the third line follows from the fact that each $y^{(\ell)}_{i,1-k_i}$ is generated uniformly and independently, and the last line follows from the fact that H is a random oracle. It follows that for any adversary $\mathcal{A}$ (making at most polynomially many queries) we have $\mathcal{H}_0 \overset{s}{\approx} \mathcal{H}_1$, as required. □

References

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *ASIACRYPT 2020, Part II*, LNCS, pages 411–439. Springer, Heidelberg, December 2020.

[AMP19] Navid Alamati, Hart Montgomery, and Sikhar Patranabis. Symmetric primitives with structured secrets. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 650–679. Springer, Heidelberg, August 2019.

[AMPR19] Navid Alamati, Hart Montgomery, Sikhar Patranabis, and Arnab Roy. Minicrypt primitives with algebraic structure and applications. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 55–82. Springer, Heidelberg, May 2019.

[App14] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. *Journal of Cryptology*, 27(3):429–451, July 2014.

[BBD$^+$22] Jeremy Booher, Ross Bowden, Javad Doliskani, Tako Boris Fouotsa, Steven D. Galbraith, Sabrina Kunzweiler, Simon-Philipp Merz, Christophe Petit, Benjamin Smith, Katherine E. Stange, Yan Bo Ti, Christelle Vincent, José Felipe Voloch, Charlotte Weitkämper, and Lukas Zobernig. Failing to hash into supersingular isogeny graphs. *IACR Cryptol. ePrint Arch.*, page 518, 2022.

[BGK11] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 201–218. Springer, Heidelberg, March 2011.

[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.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019.

[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.

[CDG$^+$17] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 33–65. Springer, Heidelberg, August 2017.

[CLM$^+$18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018.

[FGK$^+$10] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 279–295. Springer, Heidelberg, May 2010.

[GHMO21] Sanjam Garg, Mohammad Hajiabadi, Giulio Malavolta, and Rafail Ostrovsky. How to build a trapdoor function from an encryption scheme. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 220–249. Springer, 2021.

[GKLW21] Rachit Garg, Dakshita Khurana, George Lu, and Brent Waters. Black-box non-interactive non-malleable commitments. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 159–185. Springer, 2021.

[GVW20] Rishab Goyal, Satyanarayana Vusirikala, and Brent Waters. New constructions of hinting PRGs, OWFs with encryption, and more. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2020, Part I*, LNCS, pages 527–558. Springer, Heidelberg, August 2020.

[HKW20] Susan Hohenberger, Venkata Koppula, and Brent Waters. Chosen ciphertext security from injective trapdoor functions. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2020, Part I*, LNCS, pages 836–866. Springer, Heidelberg, August 2020.

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

[KM19] Fuyuki Kitagawa and Takahiro Matsuda. CPA-to-CCA transformation for KDM security. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 118–148. Springer, Heidelberg, December 2019.

[KM20] Fuyuki Kitagawa and Takahiro Matsuda. Circular security is complete for KDM security. In *ASIACRYPT 2020, Part I*, LNCS, pages 253–285. Springer, Heidelberg, December 2020.

[KMT19a] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. CCA security and trapdoor functions via key-dependent-message security. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 33–64. Springer, Heidelberg, August 2019.

[KMT19b] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. Simple and efficient KDM-CCA secure public key encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 97–127. Springer, Heidelberg, December 2019.

[KW19] Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 671–700. Springer, Heidelberg, August 2019.

[KW21] Dakshita Khurana and Brent Waters. On the CCA compatibility of public-key infrastructure. In Juan A. Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 235–260. Springer, 2021.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005.