A Unifying Umbrella for Circular-Secure Cryptographic Primitives

Fuyuki Kitagawa[1] and Takahiro Matsuda[2]

[1] NTT Social Informatics Laboratories, Tokyo, Japan, fuyuki.kitagawa@ntt.com
[2] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan, t-matsuda@aist.go.jp

Abstract

The main message of this paper is that several seemingly different circular-style primitives are *existentially equivalent*. In particular, somewhat surprisingly, we show that hinting PRGs (Koppula and Waters, CRYPTO 2019) are equivalent to secret-key encryption (SKE) schemes satisfying key-dependent-message (KDM) security. As a conceptual centerpiece, we introduce *key-dependent-shift (KDS) security* for weak pseudorandom functions (PRFs), and show that they serve as a convenient hub connecting these primitives. We also show that KDS secure weak PRFs imply other cryptographic primitives with circular-style security, such as linear-resistant PRGs (Hajiabadi et al., ITC 2023) and hinting weak PRFs (Alamati and Patranabis, ASIACRYPT 2022), all of which are thus existentially equivalent to KDM secure SKE and hinting PRGs as well. Hence, KDS secure weak PRFs can be thought of as a unifying umbrella for circular-secure cryptographic primitives. As another application of our new notion, we show that KDS security enables new constructions of public-key encryption (PKE) satisfying randomness-dependent-message (RDM) security and correlated-product secure trapdoor functions (TDFs). Our key technical contribution is a generic construction of KDS secure weak PRFs from any KDM secure SKE.

Keywords: Circular security, KDM-security, Hinting PRG, RDM security, Correlated-product secure TDF.

Contents

1 Introduction **2**
- 1.1 Backgrounds . . . 2
- 1.2 Our Results . . . 2
- 1.3 Technical Overview . . . 5
- 1.4 Difference from the Proceedings Version. . . . 7
- 1.5 Paper Organization . . . 7

2 Preliminaries **8**
- 2.1 Weak Pseudorandom Function . . . 8
- 2.2 Secret-Key Encryption . . . 8
- 2.3 Hinting PRG . . . 9
- 2.4 Linear-Resistant PRG . . . 10
- 2.5 Public-Key Encryption . . . 10
- 2.6 Garbled Circuits . . . 11

3 Key-Dependent-Shift Secure Weak PRF **12**
- 3.1 Applications . . . 13

4 Correlation RDM Secure PKE via KDS Secure Weak PRF **16**

5 Constructing a KDS Secure Weak PRF from KDM Secure SKE **22**
- 5.1 New Security Notion for SKE: Pseudorandomness of Ciphertexts against the Receiver (Receiver-PRCT Security) . . . 23
- 5.2 Constructing a SKE Scheme Satisfying KDM Security and Receiver-PRCT Security 23
- 5.3 Constructing a KDS Secure Weak PRF . . . 27

A Further Applications of KDS Secure Weak PRF **33**
- A.1 Functional Hinting PRG . . . 33
- A.2 Hinting Weak PRF . . . 35
- A.3 Functional Hinting Weak PRF . . . 36

B Proof of Theorem 5 **37**

1 Introduction

1.1 Backgrounds

Weak Pseudorandom Functions. A pseudorandom function (PRF) [GGM84] is a keyed function $\mathsf{PRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$ such that, for polynomially many adversarially chosen inputs x_i, the outputs $\mathsf{PRF}(\mathsf{k}, \mathsf{x}_i)$ are computationally indistinguishable from uniform, where $\mathsf{k} \xleftarrow{r} \{0,1\}^n$.[1] An important variant is a *weak* PRF [NR99], where pseudorandomness is required only when the inputs are sampled uniformly at random. Despite being weaker than standard PRFs, weak PRFs already suffice for many applications, including secret-key encryption (SKE).

Key-Dependent-Shift Security. We introduce a circular-style notion for weak PRFs that we call *key-dependent-shift (KDS) security*. Fix a function family $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}^\ell\}$. Informally, KDS security for $\mathcal{F}$ guarantees that for polynomially many independent samples $\mathsf{x}_i \xleftarrow{r} \{0,1\}^m$,

$$\mathsf{PRF}(\mathsf{k}, \mathsf{x}_i) \oplus f_i(\mathsf{k})$$

is computationally indistinguishable from uniform, where $\mathsf{k} \xleftarrow{r} \{0,1\}^n$ and each $f_i \in \mathcal{F}$ is chosen by an adversary. We show that KDS secure weak PRFs provide a unifying abstraction behind several circular-style primitives.

1.2 Our Results

The main message of this paper is that several seemingly different circular-style primitives are *existentially equivalent*. In particular, somewhat surprisingly, we show that *hinting PRGs [KW19] are equivalent to SKE satisfying key-dependent-message (KDM) security [BRS03]*. As a conceptual centerpiece, we introduce KDS secure weak PRFs and show that they serve as a convenient hub connecting these primitives. The key technical step is a generic construction of KDS secure weak PRFs from any KDM secure SKE. Moreover, KDS security also enables new constructions of PKE satisfying randomness-dependent-message (RDM) security [BBN+09, HO10, BCPT13] and correlated-product secure trapdoor functions (TDFs) [RS09].

Construction (KDM $\Rightarrow$ KDS). As stated above, we show that KDS secure weak PRFs can be constructed from any KDM secure SKE. Formally, we prove the following theorem.

Theorem 1 (Informal) *Assuming that there exists a SKE scheme satisfying KDM security for projection functions, there exists a KDS secure weak PRF for any functions computable by a-priori bounded polynomial-size circuits.*

Projection functions are elementary functions such that each output bit depends on at most a single bit of an input. Since projection-KDM secure SKE can be based on a number of concrete assumptions, including the learning-with-errors (LWE) and learning-parity-with noise (LPN) assumptions [ACPS09], computational Diffie–Hellman (CDH) and factoring assumptions [BLSV18], and even a group-action based assumption [ADMP20], we obtain KDS secure weak PRFs under each of these assumptions.

[1] For a set S, "$x \xleftarrow{r} S$" denotes that x is picked uniformly at random from S.

Consequences of KDS Security. We obtain the following consequences of KDS secure weak PRFs.

Hinting PRG and Its Friends: A hinting PRG [KW19] is a public-parameter-model PRG that stretches a seed $\mathsf{s} = (\mathsf{s}_1, \ldots, \mathsf{s}_n) \in \{0,1\}^n$ into $n+1$ λ-bit outputs $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \in (\{0,1\}^\lambda)^{n+1}$. It guarantees that the *hinting* distribution $\left(\mathsf{y}_0, (\mathsf{y}_{i,0}, \mathsf{y}_{i,1})_{i\in[n]}\right)$ is computationally indistinguishable from uniform, where $\mathsf{s} \xleftarrow{\mathsf{r}} \{0,1\}^n$, $(\mathsf{y}_0, \mathsf{y}_{1,\mathsf{s}_1}, \ldots, \mathsf{y}_{n,\mathsf{s}_n}) \leftarrow \mathsf{HPRG}(\mathsf{pp}, \mathsf{s})$, and $\mathsf{y}_{i,1-\mathsf{s}_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$ for each $i \in [n]$.

Koppula and Waters [KW19] introduced hinting PRGs and showed how to generically transform any CPA secure public-key encryption (PKE) scheme into a CCA secure one using a hinting PRG. Subsequently, hinting PRGs have been used in constructions of trapdoor functions (TDFs) [KMT19, GHMO21], non-interactive CCA secure commitments [GKLW21, GKLW23], anamorphic PKE [PPY24], and more.

We show that KDS secure weak PRFs for projection functions imply hinting PRGs. We also show that KDS secure weak PRFs for projection functions imply variants of a hinting PRG called a functional hinting PRG, and their "weak-PRF-analogues" called a hinting weak PRF and a functional hinting weak PRF that were introduced and studied in [AP22].

Linear-Resistant PRG: A PRG G is said to satisfy *linear resistance* if, for every $\mathbb{F}_2$-linear function f, the distribution of

$$\mathsf{G}(\mathsf{x}) \oplus f(\mathsf{x})$$

is computationally indistinguishable from uniform, where $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^n$. Hajiabadi, Khazaei, and Vahdini [HKV23] introduced linear-resistant PRGs and showed that they can be used to realize randomness-recoverable secret sharing for monotone span programs.

A KDS secure weak PRF for linear functions immediately yields a linear-resistant PRG in the public-randomness model, where G takes a uniformly chosen public parameter pp together with x as input. Such a linear-resistant PRG still suffices for the randomness-recoverable secret-sharing application.

Public-Coin KDM Secure SKE: We can construct a CPA secure SKE scheme from a weak PRF via the standard construction, where a ciphertext encrypting a plaintext m is of the form

$$\Big(\mathsf{x},\ \mathsf{PRF}(\mathsf{k},\mathsf{x}) \oplus \mathsf{m}\Big).$$

The encryption algorithm uses only public coins (randomness) x. This public-coin property implies several appealing features, such as resilience to randomness leakage and randomness-recoverability. We show that this construction becomes KDM secure if the underlying weak PRF is KDS secure. This SKE scheme is KDM-secure with respect to the same class of functions for which the underlying weak PRF is KDS-secure.

Correlation RDM Secure PKE: Let $\mathcal{F}$ be a function family. A PKE scheme is said to satisfy randomness-dependent-message (RDM) security with respect to $\mathcal{F}$ ($\mathcal{F}$-RDM security) if, for every $f \in \mathcal{F}$, the ciphertext $\mathsf{Enc}(\mathsf{pk}, f(\mathsf{r}); \mathsf{r})$ is computationally indistinguishable from an encryption of a fixed message such as $0 \cdots 0$. Following Hajiabadi and Kapron [HK15], we consider a strengthened notion that we call *correlation $\mathcal{F}$-RDM security*. Correlation $\mathcal{F}$-RDM security guarantees that, a-priori unbounded number (denoted by q) of RDM ciphertexts generated by using the same randomness r, namely,

$$\mathsf{Enc}(\mathsf{pk}_1, f_1(\mathsf{r}); \mathsf{r}), \ldots, \mathsf{Enc}(\mathsf{pk}_q, f_q(\mathsf{r}); \mathsf{r}),$$

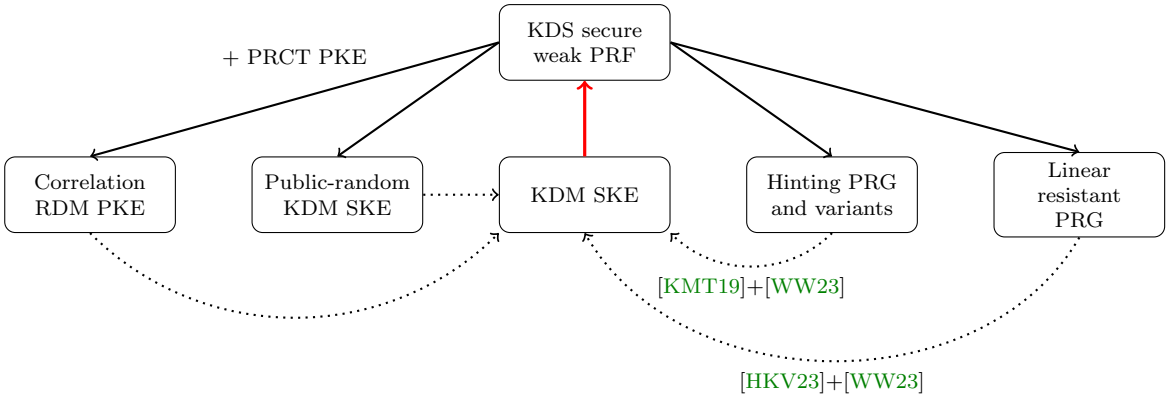

Figure 1: Equivalence of circular secure primitives

are computationally indistinguishable from encryptions of a fixed message, where $\mathsf{pk}_1, \ldots, \mathsf{pk}_q$ are independently generated public keys. Correlation $\mathcal{F}$-RDM secure PKE can also be seen as a generalization of correlated-product secure TDFs [RS09].

We show how to realize correlation $\mathcal{F}$-RDM secure PKE by combining a $\mathcal{F}$-KDS secure weak PRF with pseudorandom-ciphertext PKE.

Corollaries and Instantiations. We obtain several corollaries from the above results.

Existential Equivalence: Hinting PRGs and linear-resistant PRGs both imply one-time KDM secure SKE for projection functions [KMT19, HKV23], which in turn implies multi-time KDM secure SKE for all polynomial-time computable functions [WW23]. Furthermore, projection-KDM secure SKE can also be constructed from a circular-secure bit-SKE scheme [KM20], namely, an SKE scheme that can securely encrypt each bit of its secret key and thus satisfies arguably one of the weakest and simplest forms of KDM security. Combining these implications with our results, we show that KDM secure SKE and its public-coin variant, hinting PRG, linear-resistant PRG, and KDS secure weak PRF are all existentially equivalent. We can also prove that correlation RDM secure PKE implies KDM secure SKE,[2] which means these two primitives are existentially equivalent under the existence of pseudorandom-ciphertext PKE. We summarize these relations in Fig. 1.

New Circular Secure PRGs: Combining the above equivalence with the fact that KDM secure SKE can be realized from the LPN assumption [ACPS09], we obtain the first construction of a hinting PRG based on LPN. We also obtain the first realizations of linear-resistant PRGs based on a variety of assumptions that imply KDM secure SKE. In particular, Hajiabadi et al. [HKV23] left open the problem of constructing linear-resistant PRGs without relying on the random oracle heuristic, which is resolved by our results.

New RDM Secure PKE and Correlated Product Secure TDF: Birrell, Chung, Pass, and Telang [BCPT13] argued that it is difficult to achieve RDM secure PKE that encrypts arbitrarily long messages using only short encryption randomness. Prior to our work, achieving RDM secure PKE with short randomness relied on deterministic encryption [BBN+09] or lossy trapdoor functions [HO10, BCPT13]. This left open the existence of such schemes under the LPN assumption and under group-action-based assumptions.

[2] See Remark 1 for this implication.

Our correlation RDM security notion guarantees that arbitrarily long messages can be securely encrypted using a single short randomness. Since both of the building blocks of our correlation RDM secure PKE can be instantiated from the LPN assumption [Ale03, YZ16, ACPS09] and from group-action-based assumptions [ADMP20, AP22], we obtain the first PKE schemes satisfying a form of RDM security with short randomness from each of these two assumptions.

Moreover, these correlation RDM secure PKE schemes also yield the first correlated-product secure TDFs from the LPN assumption and from group-action-based assumptions. We further highlight that our LPN-based TDF is the first TDF that is secure for uniformly random inputs. Previous LPN-based TDFs [KMP14, KMT19] are proved secure only when part of the input is drawn from a Bernoulli distribution. Security under uniformly random inputs is required in applications where some inputs are derived from randomness-extractors, such as [HKW20].

1.3 Technical Overview

KDS Secure Weak PRF from KDM Secure SKE. We define SKE as a tuple consisting only of the encryption algorithm SEnc and the decryption algorithm SDec, for which the key generation just chooses a secret key k uniformly at random from a binary space. We also require that the message space and the ciphertext space be respectively binary spaces $\{0,1\}^m$ and $\{0,1\}^\ell$ for some m and ℓ. In our construction of a KDS secure weak PRF, we utilize a new security notion for SKE that we call *pseudorandomness of ciphertexts against the receiver (receiver-PRCT security)*. The notion guarantees that a ciphertext is computationally indistinguishable from uniform even given a secret key (hence the name of the security notion), as long as the plaintext is chosen uniformly at random, that is, $\big(\mathsf{k}, \mathsf{SEnc}(\mathsf{k},\mathsf{m})\big)$ and $\big(\mathsf{k},\mathsf{x}\big)$, where $\mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n$, $\mathsf{m} \xleftarrow{\mathsf{r}} \{0,1\}^m$, and $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^\ell$, are computationally indistinguishable.

A weak PRF satisfying KDS security with respect to a function family $\mathcal{F}$ can be easily realized from SKE satisfying receiver-PRCT security and $\mathcal{F}$-KDM security[3] simultaneously. We use a SKE scheme $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ with the key space $\{0,1\}^n$, the plaintext space $\{0,1\}^m$, and the ciphertext space $\{0,1\}^\ell$, and construct a weak PRF $\mathsf{PRF} : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^m$. The construction is highly simple: given a key k and an input x, PRF outputs $\mathsf{SDec}(\mathsf{k},\mathsf{x})$. We argue its KDS security as follows. Let $f \in \mathcal{F}$. Consider the distribution of

$$\Big(\mathsf{x},\ \mathsf{PRF}(\mathsf{k},\mathsf{x}) \oplus f(\mathsf{k})\Big) = \Big(\mathsf{x},\ \mathsf{SDec}(\mathsf{k},\mathsf{x}) \oplus f(\mathsf{k})\Big),$$

where $\mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n$ and $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^\ell$. From the receiver-PRCT security of $\mathcal{E}$, it is computationally indistinguishable from

$$\Big(\mathsf{x} = \mathsf{SEnc}(\mathsf{k}, f(\mathsf{k}) \oplus u),\ u\Big),$$

where $u \xleftarrow{\mathsf{r}} \{0,1\}^m$. Note that we use the invariance $\mathsf{SDec}(\mathsf{k},\mathsf{x}) \oplus f(\mathsf{k}) = f(\mathsf{k}) \oplus u \oplus f(\mathsf{k}) = u$. From the $\mathcal{F}$-KDM security of $\mathcal{E}$, it is computationally indistinguishable from $\big(\mathsf{x} = \mathsf{SEnc}(\mathsf{k}, v), u\big)$, where $v \xleftarrow{\mathsf{r}} \{0,1\}^m$. The final distribution is computationally indistinguishable from a uniform tuple again from receiver-PRCT security of $\mathcal{E}$.[4] We can prove that the arguments go through in the multiple-sample setting and complete the proof of the KDS security of PRF.

To connect a KDS secure weak PRF with KDM secure SKE, we have to show that we can generically add receiver-PRCT security to any KDM secure SKE. We show that this is

[3] Informally, a SKE scheme is $\mathcal{F}$-KDM secure if it can securely encrypt plaintexts of the form $f(\mathsf{k})$, where $f \in \mathcal{F}$ and k is a secret key. For the formal definition, see Section 2.2.

[4] This step can be done with the standard pseudorandom ciphertext property as well.

possible by extending the universal KDM amplification technique proposed by Waters and Wichs [WW23] using the notion of blind garbled circuits [BLSV18].

Hinting PRG from KDS Secure Weak PRF: A Warm Up with a Simpler Security Proof. We use a weak PRF $\mathsf{PRF} : \{0,1\}^n \times \{0,1\}^m \to (\{0,1\}^\lambda)^{2n+1}$ satisfying KDS security with respect to selector functions of the following form: A selector function $\mathsf{sel}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}]$ has $2n$ λ-bit strings $(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}$ hardwired, and on input $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$ outputs $(0^\lambda, w_{1,0}, w_{1,1}, \ldots, w_{n,0}, w_{n,1}) \in (\{0,1\}^\lambda)^{2n+1}$ such that $w_{1,\mathsf{k}_1} = \cdots = w_{n,\mathsf{k}_n} = 0^\lambda$, and $w_{i,1-\mathsf{k}_i} = \Delta_{i,1-\mathsf{k}_i}$ for all $i \in [n]$. A selector function can be realized by a projection function.

We construct a hinting PRG HPRG that maps an n-bit seed to $n+1$ λ-bit outputs as follows.[5] We set the public parameter as a randomly chosen string $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^m$. The evaluation algorithm, on input $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$, computes $(\mathsf{y}_0, \mathsf{y}_{1,0}, \mathsf{y}_{1,1}, \ldots, \mathsf{y}_{n,0}, \mathsf{y}_{n,1}) \leftarrow \mathsf{PRF}(\mathsf{k}, \mathsf{x})$, where each block is λ-bit long, and outputs $(\mathsf{y}_0, \mathsf{y}_{1,\mathsf{k}_1}, \ldots, \mathsf{y}_{n,\mathsf{k}_n})$. The reduction attacking the KDS secure weak PRF submits a selector function $\mathsf{sel}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}]$ for randomly chosen $\Delta_{i,v} \leftarrow \{0,1\}^\lambda$ as a KDS query, and is given $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^m$ and $\mathsf{Y} \in (\{0,1\}^\lambda)^{2n+1}$ that is either chosen uniformly at random, or

$$\begin{aligned}\mathsf{Y} &= \mathsf{PRF}(\mathsf{k}, \mathsf{x}) \oplus \mathsf{sel}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}](\mathsf{k}) \\ &= \Big(\mathsf{y}_0,\ \mathsf{y}_{1,0} \oplus w_{1,0},\ \mathsf{y}_{1,1} \oplus w_{1,1},\ \ \ldots,\ \mathsf{y}_{n,0} \oplus w_{n,0},\ \mathsf{y}_{n,1} \oplus w_{n,1}\Big),\end{aligned}$$

where $(\mathsf{y}_0, \mathsf{y}_{1,0}, \mathsf{y}_{1,1}, \ldots, \mathsf{y}_{n,0}, \mathsf{y}_{n,1}) \leftarrow \mathsf{PRF}(\mathsf{k}, \mathsf{x})$. We see that $\mathsf{y}_{i,\mathsf{k}_i} \oplus w_{i,\mathsf{k}_i} = \mathsf{y}_{i,\mathsf{k}_i}$ and $\mathsf{y}_{i,1-\mathsf{k}_i} \oplus w_{i,1-\mathsf{k}_i} = \mathsf{y}_{i,\mathsf{k}_i} \oplus \Delta_{i,1-\mathsf{k}_i}$ for each $i \in [n]$. Since $\mathsf{y}_{i,\mathsf{k}_i} \oplus \Delta_{i,1-\mathsf{k}_i}$ is distributed uniformly at random, the above distribution is identical to the hinting distribution of HPRG. Thus, the security of HPRG is reduced to that of PRF.

Hinting PRG from KDS Secure Weak PRF: A Slightly Optimized Construction. It turns out that a similar argument to the above can be conducted using a projection-KDS secure weak PRF with shorter output length $\mathsf{PRF} : \{0,1\}^n \times \{0,1\}^m \to (\{0,1\}^\lambda)^{n+1}$. Specifically, consider the construction of a hinting PRG HPRG' that on input $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n)$ simply outputs $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \leftarrow \mathsf{PRF}(\mathsf{k}, \mathsf{x})$ as it is, where x is a randomly chosen public parameter as before. This is our actual construction presented in Section 3.1.

For proving the hinting PRG security of this construction, the reduction algorithm picks $2n$ λ-bit strings $(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}$ as before, and submits a slightly different selector function $\widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}]$ that on input $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n)$ outputs $(0^\lambda, \Delta_{1,\mathsf{k}_1}, \ldots, \Delta_{n,\mathsf{k}_n}) \in (\{0,1\}^\lambda)^{n+1}$ to the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$, which returns the answer (x, z) where $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n)$. (Note that $\widetilde{\mathsf{sel}}$ can also be realized by a projection function.) Then, the reduction algorithm prepares $\mathsf{Y} = (\mathsf{z}_0, (\mathsf{z}_{i,0}, \mathsf{z}_{i,1})_{i\in[n]})$ so that $\mathsf{z}_{i,v} := \mathsf{z}_i \oplus \Delta_{i,v}$ for every $(i,v) \in [n] \times \{0,1\}$. This arrangement of Y ensures that if $\mathsf{z} = \mathsf{PRF}(\mathsf{k}, \mathsf{x}) \oplus \widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}](\mathsf{k})$, then we have

$$\mathsf{z}_0 = \mathsf{y}_0 \quad \text{and} \quad \mathsf{z}_{i,v} = \begin{cases} \mathsf{y}_i & \text{if } v = \mathsf{k}_i \\ \mathsf{y}_i \oplus \Delta_{i,0} \oplus \Delta_{i,1} & \text{if } v = 1 - \mathsf{k}_i \end{cases},$$

where $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \leftarrow \mathsf{PRF}(\mathsf{k}, \mathsf{x})$. Since the mask with $\Delta_{i,0} \oplus \Delta_{i,1}$ makes $\mathsf{z}_{i,1-\mathsf{k}_i}$ distributed uniformly at random, the distribution of Y is identical to the hinting distribution of HPRG'. On the other hand, if $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n)$ is truly random, then the XOR with $\Delta_{i,v}$ makes each element $\mathsf{z}_{i,v}$ look like a truly random string, and hence every element in Y is distributed uniformly at random. Thus, the security of HPRG' is again reduced to that of PRF.

[5] The construction of a hinting PRG here is the one presented in the proceedings of our paper.

Correlation RDM Secure PKE via KDS Secure Weak PRF. Garg, Hajiabadi, Malavolta, and Ostrovsky [GHMO21] showed how to construct TDFs from pseudorandom-ciphertext PKE and hinting PRGs. At a technical level, they built a gadget that encodes a hinting PRG seed $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$ in an invertible way using the output blocks $(\mathsf{y}_1, \ldots, \mathsf{y}_n)$, with y_0 serving as hardcore bits.

In our construction of correlation RDM secure PKE, we instead use a KDS secure weak PRF whose output is of the form $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \leftarrow \mathsf{PRF}(\mathsf{k}, \mathsf{x})$ as above, and the encryption randomness is simply the PRF key k. The public and secret keys follow the evaluation and inversion keys in [GHMO21], except that the public key additionally contains a uniformly random PRF input $\mathsf{x} \xleftarrow{r} \{0,1\}^m$.

To encrypt a randomness-dependent message $f(\mathsf{k})$, we first compute $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \leftarrow \mathsf{PRF}(\mathsf{k}, \mathsf{x})$. We then encode k into the gadget of [GHMO21] by using $(\mathsf{y}_1, \ldots, \mathsf{y}_n)$ in place of the output of a hinting PRG, and mask $f(\mathsf{k})$ using y_0.

Following the proof strategy of [GHMO21] (together with our arrangement of the values similar to what we do for the security proof of the above HPRG'), we reach a hybrid in which an adversary's view can be simulated using

$$\mathsf{PRF}(\mathsf{k}, \mathsf{x}) \oplus \left(f(\mathsf{k}),\ \widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}](\mathsf{k}) \right),$$

where $\widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[v],v\in\{0,1\}}](\mathsf{k}) := (\Delta_{1,\mathsf{k}_1}, \ldots, \Delta_{n,\mathsf{k}_n})$ and the XOR is performed component-wise. This distribution is computationally indistinguishable from uniform by the KDS security of the weak PRF, which guarantees that $f(\mathsf{k})$ is hidden from the adversary. Moreover, the multi-instance nature of KDS security allows us to complete the proof even in the correlation RDM setting.

1.4 Difference from the Proceedings Version.

In the proceedings version of this paper, in our constructions of a hinting PRG presented in Section 3.1 and a correlation RDM secure PKE scheme presented in Section 4, we used a KDS secure weak PRF whose output length is $(2n+1)\lambda$ bits, as a building block. In the current paper, we make these constructions slightly simpler and optimized using a KDS secure weak PRF with shorter output length $(n+1)\lambda$. In connection with this, we have added one paragraph for our optimized hinting PRG construction in Section 1.3.

Furthermore, all the contents omitted in the proceedings version are contained in the current paper, which includes (1) the definitions, constructions, and security proofs of a functional hinting PRG, a hinting weak PRF, and a function hinting weak PRF, (2) the proof of the correctness of our correlation RDM secure PKE scheme, and (3) the detailed descriptions and analyzes of the reduction algorithms used in the security proofs.

1.5 Paper Organization

The rest of the paper is organized as follows. In Section 2, we review the basic notation and cryptographic primitives used in the paper. In Section 3, we give the formal definition of KDS security for a weak PRF, and we give applications of KDS secure weak PRFs in Section 3.1 and Appendix A. In Section 4, we give the formal definition of correlation RDM security for PKE as well as our construction. In Section 5, we show how to construct a KDS secure weak PRF from KDM secure SKE.

2 Preliminaries

In this section, we review the basic notation and the definitions of main cryptographic primitives.

Basic Notation. For $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. For strings x and y, "$|x|$" denotes the bit-length of x, $x \leftarrow y$ denotes the assignment of y to x, "$x \| y$" denotes the concatenation x and y, and "$(x \stackrel{?}{=} y)$" is the operation that returns 1 if $x = y$ and 0 otherwise. For a discrete finite set S, "$|S|$" denotes its size, and "$x \xleftarrow{\mathsf{r}} S$" denotes choosing an element x uniformly at random from S. For a probabilistic (resp. deterministic) algorithm M, "$y \xleftarrow{\mathsf{r}} \mathsf{M}(x)$" (resp. "$y \leftarrow \mathsf{M}(x)$") denote M takes x as input, and outputs y. In particular, if M is probabilistic and we need to specify the randomness r used in M, we write "$y \leftarrow \mathsf{M}(x; r)$". If furthermore $\mathcal{O}$ is a function or an algorithm, then "$\mathsf{M}^{\mathcal{O}}$" means that M has oracle access to $\mathcal{O}$. A function $\epsilon(\lambda) : \mathbb{N} \to [0, 1]$ is said to be *negligible* if $\epsilon(\lambda) = \lambda^{-\omega(1)}$. The character "$\lambda$" always denotes a security parameter. "PPT" stands for *probabilistic polynomial time*.

2.1 Weak Pseudorandom Function

Let $\ell = \ell(\lambda)$, $m = m(\lambda)$ and $n = n(\lambda)$ be polynomials, and let $\mathsf{wPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$ be an efficiently computable function.

We say that wPRF is a *secure weak pseudorandom function (weak PRF)*, if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathtt{wPRF}}_{\mathsf{wPRF},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n;\ b \xleftarrow{\mathsf{r}} \{0,1\}; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{wPRF}}(\$)}(1^\lambda) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{wPRF}}(\$)$ is the oracle that, upon a query, picks $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^m$ and $\mathsf{y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^\ell$, computes $y_1 \leftarrow \mathsf{wPRF}(\mathsf{k}, \mathsf{x})$, and returns $(\mathsf{x}, \mathsf{y}_b)$.

Weak PRFs can be constructed from any one-way function [HILL99, GGM84].

2.2 Secret-Key Encryption

A secret-key encryption (SKE) scheme $\mathcal{E}$ consists of the two PPT algorithms $(\mathsf{SEnc}, \mathsf{SDec})$ with the following syntax:

Encryption:	ct	$\xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{sk}, \mathsf{m})$
Decryption:	$\mathsf{m} \ / \ \perp$	$\leftarrow \mathsf{SDec}(\mathsf{sk}, \mathsf{ct})$

where SDec is deterministic, sk is a secret key, and ct is a ciphertext encrypting a plaintext m under sk.

For simplicity, we assume that the security parameter λ determines the secret key space $\mathcal{K}$, plaintext space $\mathcal{M}$, and ciphertext space $\mathcal{C}$, and each element in each space has the same bit length.

We require that for all sk and m, we have $\mathsf{SDec}(\mathsf{sk}, \mathsf{SEnc}(\mathsf{sk}, \mathsf{m})) = \mathsf{m}$.

Pseudorandomness of Ciphertexts (PRCT Security). Let $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ be a SKE scheme whose secret key space is $\mathcal{K}$, plaintext space is $\mathcal{M}$, and ciphertext space is $\mathcal{C}$. We say that $\mathcal{E}$ is *PRCT secure*, if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathtt{PRCT}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{K};\ b \xleftarrow{\mathsf{r}} \{0,1\};\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{PRCT}}(\cdot)}(1^\lambda) \ : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{PRCT}}(\cdot)$ denotes the encryption oracle that takes a plaintext $\mathsf{m} \in \mathcal{M}$ as input, computes $\mathsf{ct}_1 \xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{sk}, \mathsf{m})$, picks $\mathsf{ct}_0 \xleftarrow{\mathsf{r}} \mathcal{C}$, and returns ct_b.

In this paper, we will use the well-known SKE scheme $\mathcal{E}_{\mathtt{wPRF}} = (\mathsf{SEnc}_{\mathtt{PRF}}, \mathsf{SDec}_{\mathtt{wPRF}})$ based on a weak PRF $\mathsf{wPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$, which is constructed as follows: Given a secret key $\mathsf{sk} \in \{0,1\}^n$, a plaintext $\mathsf{m} \in \{0,1\}^\ell$, and a randomness $\mathsf{r} \in \{0,1\}^m$, $\mathsf{SEnc}(\mathsf{sk}, \mathsf{m}; \mathsf{r})$ outputs $\mathsf{ct} = (\mathsf{r}, \mathsf{wPRF}(\mathsf{sk}, \mathsf{r}) \oplus \mathsf{m})$ as a ciphertext; Given a secret key $\mathsf{sk} \in \{0,1\}^\lambda$ and a ciphertext $\mathsf{ct} = (\mathsf{r}, \mathsf{y})$, $\mathsf{SDec}(\mathsf{sk}, \mathsf{ct})$ outputs $\mathsf{wPRF}(\mathsf{sk}, \mathsf{r}) \oplus \mathsf{y}$ as the decryption result. It is well-known that if wPRF is a secure weak PRF, then $\mathcal{E}_{\mathtt{wPRF}}$ is PRCT secure.

Key-Dependent Message (KDM) Security. Let $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ be a SKE scheme whose secret key space is $\mathcal{K}$ and plaintext space is $\mathcal{M}$, and let $\mathcal{F}$ be a family of functions whose domain is $\mathcal{K}$ and range is $\mathcal{M}$. We say that $\mathcal{E}$ is *$\mathcal{F}$-KDM secure* if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr\left[\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{K};\ b \xleftarrow{\mathsf{r}} \{0,1\};\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{KDM}}(\cdot,\cdot)}(1^\lambda) \ : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{KDM}}(\cdot)$ denotes the KDM-encryption oracle that takes two functions $(f_0, f_1) \in \mathcal{F}^2$ as input, and returns $\mathsf{ct} \xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{sk}, f_b(\mathsf{sk}))$.

We also say that $\mathcal{E}$ satisfies *one-time $\mathcal{F}$-KDM security* if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{OT}\text{-}\mathcal{F}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{A}}(\lambda)$ is negligible, where $\mathsf{Adv}^{\mathtt{OT}\text{-}\mathcal{F}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{A}}(\lambda)$ is defined in the same way as $\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{A}}(\lambda)$, except that $\mathcal{A}$ is allowed to make only a single KDM-encryption query.

In this paper, we will treat KDM-secure SKE schemes whose secret key space $\mathcal{K}$ and plaintext space $\mathcal{M}$ are of the form $\mathcal{K} = \{0,1\}^{\ell_{\mathsf{sk}}}$ and $\mathcal{M} = \{0,1\}^{\ell_{\mathsf{m}}}$, respectively, for some polynomials $\ell_{\mathsf{sk}} = \ell_{\mathsf{sk}}(\lambda)$ and $\ell_{\mathsf{m}} = \ell_{\mathsf{m}}(\lambda)$. The function families for KDM security that we will specifically treat are as follows:

- $\mathcal{P}$ *(Projection functions):* A function is said to be a projection function if each of its output bits depends on at most a single bit of its input. We denote by $\mathcal{P}$ the family of projection functions.
- $\mathcal{L}$ *(Affine functions):* A function $f : \{0,1\}^{\ell_{\mathsf{sk}}} \to \{0,1\}^{\ell_{\mathsf{m}}}$ is said to be an affine function if there exists a ℓ_{sk}-by-ℓ_{m} matrix $\mathsf{A} \in \{0,1\}^{\ell_{\mathsf{sk}} \times \ell_{\mathsf{m}}}$ and a vector $\mathsf{b} \in \{0,1\}^{\ell_{\mathsf{m}}}$ such that $f(\mathsf{x}) = \mathsf{x} \cdot \mathsf{A} + \mathsf{b}$ holds for all $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_{\ell_{\mathsf{sk}}}) \in \{0,1\}^{\ell_{\mathsf{sk}}}$, where the arithmetic is performed over the finite field $\mathbb{F}_2$ of order 2. We denote by $\mathcal{L}$ the family of affine functions.
- $\mathcal{B}_s$ *(Circuits of a-priori bounded size):* We denote by $\mathcal{B}_s$, where $s = s(\lambda)$ is a polynomial, the function family each of whose members can be described by a circuit of size s.

Note that we have $\mathcal{P} \subseteq \mathcal{L}$. Furthermore, if s is sufficiently large, we have $\mathcal{P} \subseteq \mathcal{B}_s$ and $\mathcal{L} \subseteq \mathcal{B}_s$.

2.3 Hinting PRG

Here, we recall the definition of a hinting PRG introduced in [KW19].

Let $n = n(\lambda)$ be a polynomial. A hinting PRG $\mathcal{H}$ with seed length n consists of the two PPT algorithms $(\mathsf{HSetup}, \mathsf{HPRG})$ with the following syntax[6]:

[6] Unlike the original paper [KW19], we adopt the syntax such that all the output blocks $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n)$ are computed in one-shot (exactly as done in a standard PRG). Furthermore, we only treat a hinting PRG such that the length of each output block is λ-bit (when pp is setup with the security parameter λ). This is without loss of generality, since if we want the length of some output block y_i to be longer, we can apply a (standard) PRG to y_i for stretching it securely.

Parameter Setup:	pp	$\xleftarrow{\mathsf{r}} \mathsf{HSetup}(1^\lambda)$
Computation:	$(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n)$	$\leftarrow \mathsf{HPRG}(\mathsf{pp}, \mathsf{s} \in \{0,1\}^n)$

where HPRG is deterministic, pp is a public parameter, $\mathsf{s} \in \{0,1\}^n$ is an input (seed), and $(\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \in (\{0,1\}^\lambda)^{n+1}$ is an output (and $y_i \in \{0,1\}^\lambda$ ($i \in \{0\} \cup [n]$) represents the i-th block of the output).

We say that $\mathcal{H}$ is a *secure hinting PRG* if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathtt{HPRG}}_{\mathcal{H},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{pp} \xleftarrow{\mathsf{r}} \mathsf{HSetup}(1^\lambda); \\ \mathsf{s} = (\mathsf{s}_1, \ldots, \mathsf{s}_n) \xleftarrow{\mathsf{r}} \{0,1\}^n; \\ (\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \leftarrow \mathsf{HPRG}(\mathsf{pp}, \mathsf{s}); \\ \forall i \in [n] : \mathsf{y}_{i,\mathsf{s}_i} \leftarrow \mathsf{y}_i; \\ \forall i \in [n] : \mathsf{y}_{i,1-\mathsf{s}_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda; \\ \mathsf{Y}_1 \leftarrow (\mathsf{y}_0, (\mathsf{y}_{i,0}, \mathsf{y}_{i,1})_{i \in [n]}); \\ \mathsf{Y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^\lambda \times (\{0,1\}^\lambda \times \{0,1\}^\lambda)^n; \\ b \xleftarrow{\mathsf{r}} \{0,1\};\ b' \xleftarrow{\mathsf{r}} \mathcal{A}(\mathsf{pp}, \mathsf{Y}_b) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible.

2.4 Linear-Resistant PRG

Here, we give the definition of a linear-resistant PRG introduced by Hajiabadi et al. [HKV23], where unlike [HKV23], we allow it to use a public parameter.

Let $m = m(\lambda)$ and $n = n(\lambda)$ be polynomials with $n < m$.

A linear-resistant PRG $\mathcal{LR}$ with seed length n and output length m consists of the two PPT algorithms $(\mathsf{LRSetup}, \mathsf{LRPRG})$ with the following syntax:

Parameter Setup:	pp	$\xleftarrow{\mathsf{r}} \mathsf{LRSetup}(1^\lambda)$
Computation:	y	$\leftarrow \mathsf{LRPRG}(\mathsf{pp}, \mathsf{s} \in \{0,1\}^n)$

where LRPRG is deterministic, pp is a public parameter, $\mathsf{s} \in \{0,1\}^n$ is an input (seed), and $\mathsf{y} \in \{0,1\}^m$ is an output.

We say that $\mathcal{LR}$ is a *secure linear-resistant PRG* if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathsf{Adv}^{\mathtt{LRPRG}}_{\mathcal{LR},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} (f \in \mathcal{L}, \mathsf{st}) \xleftarrow{\mathsf{r}} \mathcal{A}_1(1^\lambda); \\ \mathsf{pp} \xleftarrow{\mathsf{r}} \mathsf{LRSetup}(1^\lambda); \\ \mathsf{s} \xleftarrow{\mathsf{r}} \{0,1\}^n; \\ \mathsf{y}_1 \leftarrow \mathsf{LRPRG}(\mathsf{pp}, \mathsf{s}) \oplus f(\mathsf{s}); \\ \mathsf{y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^m;\ b \xleftarrow{\mathsf{r}} \{0,1\}; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}_2(\mathsf{pp}, \mathsf{y}_b, \mathsf{st}) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{L}$ denotes the family of affine functions (see Section 2.2 for its definition).

2.5 Public-Key Encryption

A public-key encryption (PKE) scheme consists of the three PPT algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

Key Generation:	$(\mathsf{pk}, \mathsf{sk})$	$\xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda)$
Encryption:	ct	$\xleftarrow{\mathsf{r}} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
Decryption:	$\mathsf{m} \ / \ \perp$	$\leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

where Dec is deterministic, $(\mathsf{pk}, \mathsf{sk})$ is a public/secret key pair, and ct is a ciphertext of a plaintext m under pk. For simplicity, we assume that λ determines the plaintext space $\mathcal{M}$, the encryption randomness space $\mathcal{R}$, and the ciphertext space $\mathcal{C}$, and each element in each space has the same bit length.

Let $\epsilon : \mathbb{N} \to [0, 1]$. We say that a PKE scheme $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is *ϵ-almost-all-keys correct* if we have

$$\mathsf{Err}_{\Pi}(\lambda) := \Pr_{(\mathsf{pk},\mathsf{sk}) \xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda)} [\exists (\mathsf{m}, \mathsf{r}) \in \mathcal{M} \times \mathcal{R} : \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; \mathsf{r})) \neq \mathsf{m}] \leq \epsilon(\lambda).$$

(A key pair $(\mathsf{pk}, \mathsf{sk})$ under which incorrect decryption could occur is called *erroneous*.) Furthermore, we just say that Π is *correct* (resp. *almost-all-keys correct*) if $\mathsf{Err}_{\Pi}(\lambda)$ is zero (resp. negligible).

Pseudorandomness of Ciphertexts (PRCT Security). We say that a PKE scheme Π satisfies *pseudorandomness of ciphertexts (PRCT security)*, if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathtt{PRCT}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda);\ b \xleftarrow{\mathsf{r}} \{0, 1\}; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{PRCT}}(\cdot)}(\mathsf{pk}) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{PRCT}}(\cdot)$ denotes the encryption oracle that takes a plaintext $\mathsf{m} \in \mathcal{M}$ as input, computes $\mathsf{ct}_1 \xleftarrow{\mathsf{r}} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$, picks $\mathsf{ct}_0 \xleftarrow{\mathsf{r}} \mathcal{C}$, and returns ct_b.

2.6 Garbled Circuits

A (circuit) garbling scheme $\mathcal{G}$ consists of the three PPT algorithms $(\mathsf{Garble}, \mathsf{GEval}, \mathsf{GSim})$ with the following syntax:

Garbling:	$(\widetilde{C}, (\mathsf{L}_{i,v})_{i\in[n],v\in\{0,1\}})$	$\xleftarrow{\mathsf{r}} \mathsf{Garble}(1^\lambda, C)$
Evaluation:	y	$\leftarrow \mathsf{GEval}(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]})$
Simulator (for Security):	$(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]})$	$\xleftarrow{\mathsf{r}} \mathsf{GSim}(1^\lambda, \mathsf{y})$

where GEval is deterministic, $C : \{0, 1\}^n \to \{0, 1\}^m$ is (the description of) a circuit, $\widetilde{C}$ is a garbled circuit, $\mathsf{L}_{i,v} \in \{0, 1\}^\lambda$ $(i \in [n], v \in \{0, 1\})$ and $\mathsf{L}'_i \in \{0, 1\}^\lambda$ $(i \in [n])$ are input labels for evaluating $\widetilde{C}$, and $\mathsf{y} \in \{0, 1\}^m$ is an evaluation result.

For simplicity, we assume that the security parameter λ determines the size $s = s(\lambda)$, the input length $n = n(\lambda)$, and the output length $m = m(\lambda)$ of circuits that can be treated by $\mathcal{G}$, and that the length of each input label is λ.

We require that for all $\lambda \in \mathbb{N}$, all circuits C with input length $n = n(\lambda)$, and all inputs $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \in \{0, 1\}^n$, the following holds: If $(\widetilde{C}, (\mathsf{L}_{i,v})_{i\in[n],v\in\{0,1\}}) \xleftarrow{\mathsf{r}} \mathsf{Garble}(1^\lambda, C)$, then we have $\mathsf{GEval}(\widetilde{C}, (\mathsf{L}_{i,\mathsf{x}_i})_{i\in[n]}) = C(x)$.

Simulation Security. A standard security notion for a garbling scheme is *simulation security* defined using the simulator GSim. The formal definition is as follows.

We say that a garbling scheme $\mathcal{G} = (\mathsf{Garble}, \mathsf{GEval}, \mathsf{GSim})$ satisfies *simulation security* if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathsf{Adv}^{\mathtt{Sim}}_{\mathcal{G},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} (C, \mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n), \mathsf{st}) \xleftarrow{\mathsf{r}} \mathcal{A}_1(1^\lambda); \\ (\widetilde{C}_1, (\mathsf{L}_{i,v})_{i\in[n],v\in\{0,1\}}) \xleftarrow{\mathsf{r}} \mathsf{Garble}(1^\lambda, C); \\ \mathsf{Labels}_1 \leftarrow (\mathsf{L}_{i,\mathsf{x}_i})_{i\in[n]}; \\ (\widetilde{C}_0, \mathsf{Labels}_0) \xleftarrow{\mathsf{r}} \mathsf{GSim}(1^\lambda, C(\mathsf{x})); \\ b \xleftarrow{\mathsf{r}} \{0, 1\};\ b' \xleftarrow{\mathsf{r}} \mathcal{A}_2(\widetilde{C}_b, \mathsf{Labels}_b, \mathsf{st}) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible.

Blindness. Brakerski et al. [BLSV18] introduced *blindness* as a security notion for a garbling scheme. Informally, it guarantees that the output of the simulator GSim is pseudorandom when it is fed with a random input. In this paper, we will treat a garbling scheme satisfying blindness, and thus we recall its formal definition below.

We say that a garbling scheme $\mathcal{G} = (\mathsf{Garble}, \mathsf{GEval}, \mathsf{GSim})$ satisfies *blindness* if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathtt{Blind}}_{\mathcal{G},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{y} \xleftarrow{\mathsf{r}} \{0,1\}^m; \\ (\widetilde{C}_1, \mathsf{Labels}_1) \xleftarrow{\mathsf{r}} \mathsf{GSim}(1^\lambda, \mathsf{y}); \\ \widetilde{C}_0 \xleftarrow{\mathsf{r}} \{0,1\}^{|\widetilde{C}_1|};\ \mathsf{Labels}_0 \xleftarrow{\mathsf{r}} (\{0,1\}^\lambda)^n; \\ b \xleftarrow{\mathsf{r}} \{0,1\};\ b' \xleftarrow{\mathsf{r}} \mathcal{A}(\widetilde{C}_b, \mathsf{Labels}_b) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible.

Brakerski et al. [BLSV18] showed a construction of a garbling scheme satisfying both simulation security and blindness, using a (standard) PRG as a building block. Hence, a garbling scheme satisfying both security notions exist if a one-way function exists [HILL99].

3 Key-Dependent-Shift Secure Weak PRF

In this section, we formally define *key-dependent-shift (KDS) security* for a weak PRF.

Definition 1 (Key-Dependent-Shift Secure Weak PRF) *Let $\ell = \ell(\lambda)$, $m = m(\lambda)$, and $n = n(\lambda)$ be polynomials. Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$ be an efficiently computable function. Let $\mathcal{F}$ be a family of functions from $\{0,1\}^n$ to $\{0,1\}^\ell$.*

We say that KwPRF is a $\mathcal{F}$-KDS secure weak PRF, if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDS}\text{-}\mathtt{wPRF}}_{\mathsf{KwPRF},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n;\ b \xleftarrow{\mathsf{r}} \{0,1\}; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{KDS}\text{-}\mathtt{wPRF}}(\cdot)}(1^\lambda) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{KDS}\text{-}\mathtt{wPRF}}(\cdot)$ is the oracle that takes a function $f \in \mathcal{F}$ as input, and proceeds as follows:

1. *Pick* $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^m$.
2. *Compute* $\mathsf{y}_1 \xleftarrow{\mathsf{r}} \mathsf{KwPRF}(\mathsf{k}, \mathsf{x}) \oplus f(\mathsf{k})$.
3. *Pick* $\mathsf{y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^\ell$.
4. *Return* $(\mathsf{x}, \mathsf{y}_b)$.

Function Families for KDS Security. In this paper, we will treat the same function families as those considered for KDM security for SKE listed in Section 2.2, namely, the family of projection functions $\mathcal{P}$, the family of affine functions $\mathcal{L}$, and the family of functions that can be computed by size-s circuits $\mathcal{B}_s$.

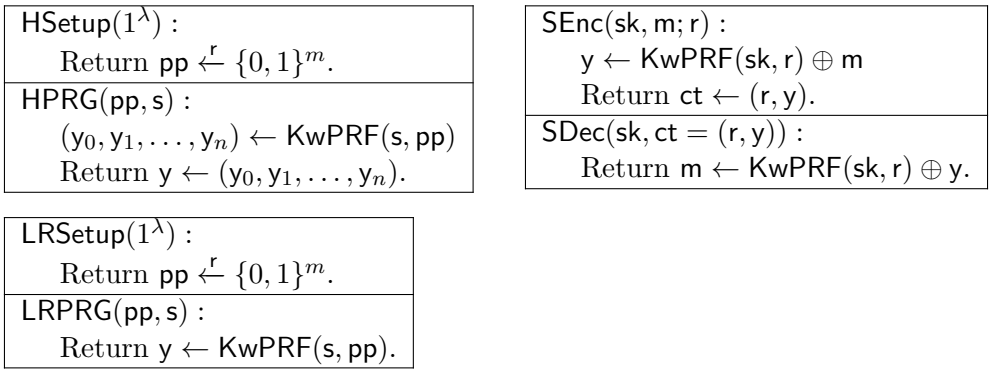

Figure 2: Applications of a KDS secure weak PRF KwPRF: Hinting PRG $\mathcal{H}$ (left-top), linear-resistant PRG $\mathcal{LR}$ (left-bottom), and public-coin KDM secure SKE $\mathcal{E}$ (right).

3.1 Applications

Here, we give applications of a KDS secure weak PRF: Hinting PRGs, linear-resistant PRGs, and public-coin KDM secure SKE. Its application to correlation RDM secure PKE will be presented in Section 4. In Appendix A, we also show how to realize variants of hinting PRG called functional hinting PRG, hinting weak PRF, and functional hinting weak PRF [AP22] using a KDS secure weak PRF.

Hinting PRG. Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{(n+1)\cdot\lambda}$ be a $\mathcal{P}$-KDS secure weak PRF, for some polynomials $n = n(\lambda)$ and $m = m(\lambda)$. We identify the output space $\{0,1\}^{(n+1)\cdot\lambda}$ with $(\{0,1\}^\lambda)^{n+1}$.

Using KwPRF, we construct a hinting PRG $\mathcal{H} = (\mathsf{HSetup}, \mathsf{HPRG})$ as in Fig. 2 (top-left).

The following theorem guarantees that $\mathcal{H}$ constructed as above is indeed a secure hinting PRG.

Theorem 2 *If* KwPRF *is a $\mathcal{P}$-KDS secure weak PRF, then $\mathcal{H}$ is a secure hinting PRG.*

Proof of Theorem 2. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the hinting PRG security of $\mathcal{H}$. Using $\mathcal{A}$, we consider the following PPT reduction algorithm $\mathcal{B}$ that attacks the $\mathcal{P}$-KDS security of KwPRF.

$\mathcal{B}^{\mathcal{O}_{\texttt{KDS-wPRF}}(\cdot)}(1^\lambda)$**:** $\mathcal{B}$ picks $\Delta_{1,0}, \Delta_{1,1}, \ldots, \Delta_{n,0}, \Delta_{n,1} \xleftarrow{r} \{0,1\}^\lambda$, and defines a projection function $g \in \mathcal{P}$ that computes the selector function $\widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}] : \{0,1\}^\lambda \to (\{0,1\}^\lambda)^{n+1}$ that has $2n$ λ-bit values hardwired, and on input $\mathsf{s} = (\mathsf{s}_1, \ldots, \mathsf{s}_n) \in \{0,1\}^n$, outputs $(0^\lambda, \Delta_{1,\mathsf{s}_1}, \ldots, \Delta_{n,\mathsf{s}_n}) \in (\{0,1\}^\lambda)^{n+1}$. (Recall that a selector function can be computed by a projection function.) Then, $\mathcal{B}$ submits a query $g \in \mathcal{P}$ to the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$, and receives the answer $(\mathsf{pp}, \mathsf{z}) \in \{0,1\}^m \times (\{0,1\}^\lambda)^{n+1}$, where $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n) \in (\{0,1\}^\lambda)^{n+1}$. Now, $\mathcal{B}$ sets $\mathsf{y}'_0 \leftarrow \mathsf{z}_0$, and $\mathsf{y}'_{i,v} \leftarrow \mathsf{z}_i \oplus \Delta_{i,v}$ for all $(i,v) \in [n] \times \{0,1\}$. Then, $\mathcal{B}$ sets $\mathsf{Y}' \leftarrow (\mathsf{y}'_0, (\mathsf{y}'_{i,0}, \mathsf{y}'_{i,1})_{i\in[n]}) \in \{0,1\}^n \times (\{0,1\}^\lambda)^{2n}$, runs $b' \xleftarrow{r} \mathcal{A}(\mathsf{pp}, \mathsf{Y}')$, sets $\beta' \leftarrow b'$, and terminates with output β'.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be the challenge bit in $\mathcal{B}$'s $\mathcal{P}$-KDS security experiment regarding KwPRF, let $\mathsf{s} = (\mathsf{s}_1, \ldots, \mathsf{s}_n) \in \{0,1\}^n$ be the key for KwPRF chosen in $\mathcal{B}$'s experiment, and let $\mathsf{y} = (\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) = \mathsf{KwPRF}(\mathsf{s}, \mathsf{pp})$, where pp is the first element of the answer to $\mathcal{B}$'s $\mathcal{O}_{\texttt{KDS-wPRF}}$-query.

It is not hard to see that if $\beta = 1$ and thus $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n) = \mathsf{KwPRF}(\mathsf{s}, \mathsf{pp}) \oplus g(\mathsf{s})$, then $\mathcal{B}$ simulates the hinting PRG security experiment with $b = 1$ perfectly for $\mathcal{A}$. Specifically, in this case, due to how we design the projection function g, we have $\mathsf{z}_0 = \mathsf{y}_0$ and $\mathsf{z}_i = \mathsf{y}_i \oplus \Delta_{i,\mathsf{s}_i}$ for all $i \in [n]$. Hence, the components of $\mathsf{Y}' = (\mathsf{y}'_0, (\mathsf{y}'_{i,0}, \mathsf{y}'_{i,1})_{i\in[n]})$ are generated as follows:

$$\mathsf{y}'_0 = \mathsf{y}_0,$$
$$\forall (i, v) \in [n] \times \{0,1\} : \quad \mathsf{y}'_{i,v} = \mathsf{z}_i \oplus \Delta_{i,v} = \begin{cases} \mathsf{y}_i & \text{if } v = \mathsf{s}_i \\ \mathsf{y}_i \oplus \Delta_{i,0} \oplus \Delta_{i,1} & \text{if } v = 1 - \mathsf{s}_i \end{cases}.$$

Note that from $\mathcal{A}$'s viewpoint, $\mathsf{y}'_{i,1-\mathsf{s}_i}$ looks like a truly random string due to the XOR with $\Delta_{i,0} \oplus \Delta_{i,1}$ that is in turn distributed uniformly at random. Hence, Y' is distributed in exactly the same way as one generated in the hinting PRG security experiment with $b = 1$.

On the other hand, if $\beta = 0$ and thus $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n)$ is just a random value in $(\{0,1\}^\lambda)^{n+1}$, then $\mathcal{B}$ simulates the hinting PRG security experiment with $b = 0$ perfectly for $\mathcal{A}$. In particular, each $\mathsf{y}'_{i,v}$ is masked with the truly random string $\Delta_{i,v}$, which makes $\mathsf{y}'_{i,v}$ look like a truly random string for every $(i, v) \in [n] \times \{0,1\}$.

Therefore, $\mathcal{B}$ is overall simulating the hinting PRG security experiment perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is that of $\mathcal{B}$'s. Since $\mathcal{B}$ uses $\mathcal{A}$'s output as it is, we have $\mathsf{Adv}^{\mathtt{HPRG}}_{\mathcal{H},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathcal{P}\text{-}\mathtt{KDS}\text{-}\mathtt{wPRF}}_{\mathsf{KwPRF},\mathcal{B}}(\lambda)$, which is negligible by our assumption that KwPRF is a $\mathcal{P}$-KDS secure weak PRF. Hence, $\mathcal{H}$ is a secure hinting PRG. □ (**Theorem 2**)

Linear-Resistant PRG (in the Public-Parameter Model). Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$ be a $\mathcal{L}$-KDS secure weak PRF, for some polynomials $n = n(\lambda)$, $m = m(\lambda)$, and $\ell = \ell(\lambda)$ with $n < \ell$.

Using KwPRF, we construct a linear resistant PRG $\mathcal{LR} = (\mathsf{LRSetup}, \mathsf{LRPRG})$ as in Fig. 2 (bottom-left).

The following theorem guarantees that $\mathcal{LR}$ constructed as above is indeed a secure linear-resistant PRG.

Theorem 3 *If* KwPRF *is a* $\mathcal{L}$*-KDS secure weak PRF, then* $\mathcal{LR}$ *is a secure linear-resistant PRG.*

Proof of Theorem 3. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the linear-resistant PRG security of $\mathcal{LR}$. Using $\mathcal{A}$, we consider the following PPT reduction algorithm $\mathcal{B}$ that attacks the $\mathcal{L}$-KDS security of KwPRF.

$\mathcal{B}^{\mathcal{O}_{\mathtt{KDS}\text{-}\mathtt{wPRF}}(\cdot)}(1^\lambda)$**:** $\mathcal{B}$ runs $(f \in \mathcal{L}, \mathsf{st}) \xleftarrow{\mathsf{r}} \mathcal{A}_1(1^\lambda)$. Then, $\mathcal{B}$ submits a query $f \in \mathcal{L}$ to the oracle $\mathcal{O}_{\mathtt{KDS}\text{-}\mathtt{wPRF}}$, and receives the answer $(\mathsf{pp}, \mathsf{y}) \in \{0,1\}^m \times \{0,1\}^\ell$. Then, $\mathcal{B}$ runs $b' \leftarrow \mathcal{A}_2(\mathsf{pp}, \mathsf{y}, \mathsf{st})$, sets $\beta' \leftarrow b'$, and terminates with output β'.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be the challenge bit in $\mathcal{B}$'s $\mathcal{L}$-KDS security experiment regarding KwPRF, and let $\mathsf{s} \in \{0,1\}^n$ be the key for KwPRF chosen in $\mathcal{B}$'s experiment. It is straightforward to see that if $\beta = 1$ and thus $\mathsf{y} = \mathsf{KwPRF}(\mathsf{s}, \mathsf{pp}) \oplus f(\mathsf{s})$, then $\mathcal{A}$ simulates the linear-resistant PRG security experiment with $b = 1$ perfectly for $\mathcal{A}$. On the other hand, if $\beta = 0$ and thus y is just a random value in $\{0,1\}^\ell$, then $\mathcal{B}$ simulates the linear-resistant PRG security experiment with $b = 0$ perfectly for $\mathcal{A}$. Therefore, $\mathcal{B}$ is simulating the linear-resistant PRG security experiment perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is that of $\mathcal{B}$'s. Since $\mathcal{B}$ uses $\mathcal{A}$'s output as it is, we have $\mathsf{Adv}^{\mathtt{LRPRG}}_{\mathcal{LR},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathcal{L}\text{-}\mathtt{KDS}\text{-}\mathtt{wPRF}}_{\mathsf{KwPRF},\mathcal{B}}(\lambda)$, which is negligible by our assumption that KwPRF is a $\mathcal{L}$-KDS secure weak PRF. Hence, $\mathcal{LR}$ is a secure linear-resistant PRG. □ (**Theorem 3**)

Public-Coin KDM Secure SKE. Let $s = s(\lambda)$ be a polynomial. Let $\mathcal{F} \in \{\mathcal{P}, \mathcal{L}, \mathcal{B}_s\}$. Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$ be a $\mathcal{F}$-KDS secure weak PRF, for some polynomials $n = n(\lambda)$, $m = m(\lambda)$, and $\ell = \ell(\lambda)$.

Using KwPRF, we construct a SKE scheme $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ such that its key space is $\{0,1\}^n$, its plaintext space is $\{0,1\}^\ell$, and the encryption randomness space is $\{0,1\}^m$, as in Fig. 2 (bottom-right). Note that this construction is exactly the weak-PRF-based SKE scheme $\mathcal{E}_{\mathtt{wPRF}}$ recalled in Section 2.2, except that the underlying (standard) weak PRF wPRF is replaced with the KDS-secure weak PRF KwPRF.

The following theorem guarantees that the SKE scheme $\mathcal{E}$ constructed as above satisfies $\mathcal{F}$-KDM security.

Theorem 4 *Let $s = s(\lambda)$ be a polynomial. Let $\mathcal{F} \in \{\mathcal{P}, \mathcal{L}, \mathcal{B}_s\}$. If KwPRF is a $\mathcal{F}$-KDS secure weak PRF, then the SKE scheme $\mathcal{E}$ satisfies $\mathcal{F}$-KDM security.*

Proof of Theorem 4. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the $\mathcal{F}$-KDM security of the SKE scheme $\mathcal{E}$. Let $\mathtt{SUC}$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in the $\mathcal{F}$-KDM security experiment regarding $\mathcal{E}$.

Using $\mathcal{A}$, we consider the following PPT reduction algorithm $\mathcal{B}$ that attacks the $\mathcal{F}$-KDS security of KwPRF.

$\mathcal{B}^{\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}(\cdot)}(1^\lambda)$**:** $\mathcal{B}$ picks $b \xleftarrow{\mathsf{r}} \{0,1\}$, and runs $\mathcal{A}(1^\lambda)$.

> For a KDM-encryption query $(f_0, f_1) \in \mathcal{F}^2$ from $\mathcal{A}$, $\mathcal{B}$ submits a query $f_b \in \mathcal{F}$ to the oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}$, and receives the answer $(\mathsf{r}, \mathsf{y}) \in \{0,1\}^m \times \{0,1\}^\ell$, and returns a ciphertext $\mathsf{ct} = (\mathsf{r}, \mathsf{y})$ to $\mathcal{A}$.
>
> When $\mathcal{A}$ teminates with output b', $\mathcal{B}$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be the challenge bit in $\mathcal{B}$'s $\mathcal{F}$-KDS security experiment regarding KwPRF, and let $\mathsf{sk} \in \{0,1\}^n$ be the key for KwPRF chosen in $\mathcal{B}$'s experiment. It is straightforward to see that if $\beta = 1$ and thus $\mathsf{y} = \mathsf{KwPRF}(\mathsf{sk}, \mathsf{r}) \oplus f_b(\mathsf{sk})$ is returned from $\mathcal{B}$'s oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}$ as the answer to $\mathcal{B}$'s query f_b, then the ciphertext $\mathsf{ct} = (\mathsf{r}, \mathsf{y})$ returned to $\mathcal{A}$ as the answer to a KDM-encryption query (f_0, f_1) is exactly $\mathsf{SEnc}(\mathsf{sk}, f_b(\mathsf{sk}); \mathsf{r})$, and thus $\mathcal{B}$ simulates the $\mathcal{F}$-KDM security experiment perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b (chosen by $\mathcal{B}$). Since $\mathcal{B}$ outputs $\beta' = 1$ only when $b' = b$ occurs, we have $\Pr[\beta' = 1 | \beta = 1] = \Pr[\mathtt{SUC}]$. On the other hand, if $\beta = 0$ and thus a random value $\mathsf{y} \in \{0,1\}^\ell$ is returned from $\mathcal{B}$'s oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}$ as the answer to $\mathcal{B}$'s query f_b, then the ciphertext $\mathsf{ct} = (\mathsf{r}, \mathsf{y})$ returned to $\mathcal{A}$ as the answer to a KDM-encryption query (f_0, f_1) is independent of b, and thus $\mathcal{A}$'s view contains no information on b. Hence, no matter what value b' is output by $\mathcal{A}$, the probability that $b' = b$ occurs (and thus $\mathcal{B}$ outputs $\beta' = 1$) is exactly $1/2$, namely, we have $\Pr[\beta' = 1 | \beta = 0] = 1/2$.

Combining the arguments so far, we have

$$\begin{aligned}\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDS\text{-}wPRF}}_{\mathsf{KwPRF},\mathcal{B}}(\lambda) &= \Big|\Pr[\beta' = 1 | \beta = 1] - \Pr[\beta' = 1 | \beta = 0]\Big| \\ &= \Big|\Pr[\mathtt{SUC}] - \frac{1}{2}\Big| = \frac{1}{2} \cdot \mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{A}}(\lambda).\end{aligned}$$

This implies $\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{A}}(\lambda) = 2 \cdot \mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{KDS\text{-}wPRF}}_{\mathsf{KwPRF},\mathcal{B}}(\lambda)$, which is negligible by our assumption that KwPRF is a $\mathcal{F}$-KDS secure weak PRF. Hence, the SKE scheme $\mathcal{E}$ satisfies $\mathcal{F}$-KDM security. □ (**Theorem 4**)

4 Correlation RDM Secure PKE via KDS Secure Weak PRF

In this section, as another application of a KDS secure weak PRF, we show how to construct a randomness-dependent-message (RDM) secure PKE scheme [HO10, BCPT13, HK15]. We first define RDM security for PKE. Following Hajiabadi and Kapron [HK15], we consider the correlated version of the definition.

Correlation RDM Security. Let $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme whose encryption randomness space is $\{0,1\}^n$ and plaintext space is $\{0,1\}^m$, and let $\mathcal{F}$ be a family of functions whose domain is $\{0,1\}^n$ and range is $\{0,1\}^m$. We say that Π satisfies *correlation $\mathcal{F}$-RDM security* if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{CRDM}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} b \xleftarrow{\mathsf{r}} \{0,1\};\ \mathsf{r} \xleftarrow{\mathsf{r}} \{0,1\}^n; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{CRDM}}(\cdot,\cdot)}(1^\lambda); \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{CRDM}}(\cdot,\cdot)$ is the oracle that takes two functions $(f_0, f_1) \in \mathcal{F}^2$ as input, and proceeds as follows:

1. Generate $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda)$.
2. Compute $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, f_b(\mathsf{r}); \mathsf{r})$.
3. Return $(\mathsf{pk}, \mathsf{ct})$.

Remark 1 We provide some remarks on our definition of correlation RDM security.

- The definition of [HK15] is recovered by restricting to the bit-circular case, where each RDM function outputs a single bit of r. Our definition also implies correlated-product secure TDFs [RS09] if the function family $\mathcal{F}$ contains the identity function.
- A PKE scheme satisfying correlation $\mathcal{F}$-RDM security implies a $\mathcal{F}$-KDM secure SKE scheme. Consider the following SKE scheme based on a PKE scheme $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$. The key-generation algorithm samples $\mathsf{r} \leftarrow \{0,1\}^n$ as a secret key. To encrypt a one-bit message $\beta \in \{0,1\}$ under key r, it generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$ and outputs the ciphertext $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \beta; \mathsf{r}))$. Decryption is done by computing $\mathsf{Enc}(\mathsf{pk}, 0; \mathsf{r})$ and $\mathsf{Enc}(\mathsf{pk}, 1; \mathsf{r})$ and comparing them with the second component of the ciphertext. The $\mathcal{F}$-KDM security of this SKE scheme follows directly from the correlation $\mathcal{F}$-RDM security of Π.
- Our notion does not capture the setting where the RDM function f depends on the current public key used for computing the challenge ciphertext $\mathsf{Enc}(\mathsf{pk}, f(\mathsf{r}); \mathsf{r})$ (although f may depend on public keys obtained from previous queries). Thus, our notion is incomparable to that of Birrell et al. [BCPT13], which allows fully public-key-dependent RDM functions but does not consider the correlation setting.

Construction. Our construction is based on the TDF construction by Garg et al. [GHMO21] who showed, among other things, how to construct a TDF from the combination of a PRCT-secure PKE scheme and a hinting PRG.

Formally, our correlation RDM secure PKE scheme is constructed as follows: Consider the following ingredients:

- Let $\ell_{\mathsf{m}} = \ell_{\mathsf{m}}(\lambda)$ be a polynomial that represents the length of a plaintext that we wish to encrypt by the PKE scheme constructed here.

$\mathsf{RKG}(1^\lambda)$:

$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{r} \mathsf{KG}(1^\lambda)$

$\mathsf{A} \xleftarrow{r} \{0,1\}^{3\lambda}$; $\mathsf{B} \xleftarrow{r} \{0,1\}^m$; $\mathsf{C}_1, \ldots, \mathsf{C}_n \xleftarrow{r} \{0,1\}^{\ell_{\mathsf{ct}}}$

$\mathsf{rpk} \leftarrow (\mathsf{pk}, \mathsf{A}, \mathsf{B}, (\mathsf{C}_i)_{i\in[n]})$; $\mathsf{rsk} \leftarrow (\mathsf{sk}, \mathsf{A}, \mathsf{B})$

Return $(\mathsf{rpk}, \mathsf{rsk})$.

$\mathsf{REnc}(\mathsf{rpk}, \mathsf{m}; \mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \in \{0,1\}^n)$:

Parse rpk as $(\mathsf{pk}, \mathsf{A}, \mathsf{B}, (\mathsf{C}_i)_{i\in[n]})$.

$(\mathsf{r}_0, (\mathsf{r}_i)_{i\in[n]}) \leftarrow \mathsf{KwPRF}(\mathsf{x}, \mathsf{B})$

$\forall i \in [n]$:

$\mathsf{ct}_{i,\mathsf{x}_i} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}_i)$

$$\mathsf{ct}_i \leftarrow \begin{cases} \mathsf{ct}_{i,0} & \text{if } \mathsf{x}_i = 0 \\ \mathsf{ct}_{i,1} \oplus \mathsf{C}_i & \text{if } \mathsf{x}_i = 1 \end{cases}$$

$\mathsf{z} \leftarrow \mathsf{r}_0 \oplus \mathsf{m}$

Return $\mathsf{rct} \leftarrow (\mathsf{z}, (\mathsf{ct}_i)_{i\in[n]})$.

$\mathsf{RDec}(\mathsf{rsk}, \mathsf{rct})$:

Parse rsk as $(\mathsf{sk}, \mathsf{A}, \mathsf{B})$.

Parse rct as $(\mathsf{z}, (\mathsf{ct})_{i\in[n]})$.

$\forall i \in [n]$:

$$\mathsf{x}_i \leftarrow \begin{cases} 0 & \text{if } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_i) = \mathsf{A} \\ 1 & \text{otherwise} \end{cases}$$

$\mathsf{x} \leftarrow (\mathsf{x}_1, \ldots, \mathsf{x}_n)$

$(\mathsf{r}_0, (\mathsf{r}_i)_{i\in[n]}) \leftarrow \mathsf{KwPRF}(\mathsf{x}, \mathsf{B})$

Return $\mathsf{m} \leftarrow \mathsf{z} \oplus \mathsf{r}_0$.

Figure 3: The RDM secure PKE scheme $\Pi_{\mathtt{RDM}}$ based on a PRCT secure PKE scheme and a KDS secure weak PRF.

- Let $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme such that, when the security parameter is λ, the plaintext space is $\{0,1\}^{3\lambda}$, the encryption randomness space is $\{0,1\}^\lambda$, and the ciphertext space is $\{0,1\}^{\ell_{\mathsf{ct}}}$ for some polynomial $\ell_{\mathsf{ct}} = \ell_{\mathsf{ct}}(\lambda)$.
- Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{\ell_{\mathsf{m}}+n\lambda}$ be a KDS-secure[7] weak PRF, for some polynomials $n = n(\lambda)$ and $m = m(\lambda)$. We will identify the output space with $\{0,1\}^{\ell_{\mathsf{m}}} \times (\{0,1\}^\lambda)^n$.

Using these primitives as building blocks, our correlation RDM secure PKE scheme $\Pi_{\mathtt{RDM}} = (\mathsf{RKG}, \mathsf{REnc}, \mathsf{RDec})$ is constructed as in Fig. 3. The plaintext space and encryption random space of $\Pi_{\mathtt{RDM}}$ are $\{0,1\}^{\ell_{\mathsf{m}}}$ and $\{0,1\}^n$, respectively.

$\Pi_{\mathtt{RDM}}$ satisfies almost-all-keys correctness, as stated below. Its proof is essentially the same as that for the TDF in [GHMO21], but we give a formal proof in Appendix B for completeness.

Theorem 5 *If the PKE scheme Π satisfies correctness, then the PKE scheme $\Pi_{\mathtt{RDM}}$ satisfies $(n+1)\cdot 2^{-\lambda}$-almost-all-keys correctness.*

The correlation RDM security of $\Pi_{\mathtt{RDM}}$ is guaranteed by the following theorem.

Theorem 6 *If the PKE scheme Π satisfies PRCT security, then the following holds:*

- *Let $\mathcal{F} \in \{\mathcal{P}, \mathcal{L}\}$. If KwPRF is a $\mathcal{F}$-KDS secure weak PRF, then the PKE scheme $\Pi_{\mathtt{RDM}}$ satisfies correlation $\mathcal{F}$-RDM security.*
- *Let $s = s(\lambda)$ be an arbitrary polynomial, and let $p = p(\lambda) = O(n\lambda)$ be the size of the circuit computing the selector function that will appear in the proof. If KwPRF is a $\mathcal{B}_{s+p}$-KDS secure weak PRF, then the PKE scheme $\Pi_{\mathtt{RDM}}$ satisfies correlation $\mathcal{B}_s$-RDM security.*

Proof of Theorem 6. Since the proofs for both of the items are essentially the same, we mainly show the proof for the second item (i.e. correlation $\mathcal{B}_s$-RDM security of $\Pi_{\mathtt{RDM}}$), and briefly

[7]The function family with respect to which KwPRF is KDS secure will affect with respect to what function family the PKE scheme $\Pi_{\mathtt{RDM}}$ constructed here is RDM secure. See Theorem 6 below.

explain how the first item (i.e. correlation $\mathcal{F}$-RDM security of $\Pi_{\mathtt{RDM}}$ for $\mathcal{F} \in \{\mathcal{P}, \mathcal{L}\}$) can be proved.

Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the correlation $\mathcal{B}_s$-RDM security of the PKE scheme $\Pi_{\mathtt{RDM}}$, and let q denote the number of queries to $\mathcal{O}_{\mathtt{CRDM}}(\cdot,\cdot)$ made by $\mathcal{A}$. We will show that for this $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ that satisfy

$$\mathsf{Adv}^{\mathcal{B}_s\text{-}\mathtt{CRDM}}_{\Pi_{\mathtt{RDM}},\mathcal{A}}(\lambda) \leq 2q \cdot \mathsf{Adv}^{\mathtt{PRCT}}_{\Pi,\mathcal{B}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathcal{B}_{s+p}\text{-}\mathtt{KDS}\text{-}\mathtt{wPRF}}_{\mathsf{KwPRF},\mathcal{B}'}(\lambda), \tag{1}$$

which will ensure that $\mathcal{A}$'s advantage is negligible by our assumption that Π satisfies PRCT security and KwPRF is a $\mathcal{B}_{s+p}$-KDS secure weak PRF, and hence imply the second item of the theorem.

Our proof is via a sequence of games argument using five games. In the following, let $\mathtt{SUC}_t$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game $t \in [4]$.

Game 1: This is the correlation $\mathcal{B}_s$-RDM security experiment regarding $\Pi_{\mathtt{RDM}}$. Specifically, the description of Game 1 is as follows:

1. Pick $b \xleftarrow{\mathsf{r}} \{0,1\}$ and $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \xleftarrow{\mathsf{r}} \{0,1\}^n$.
2. Run $\mathcal{A}^{\mathcal{O}_{\mathtt{CRDM}}(\cdot,\cdot)}(1^\lambda)$. Then, $\mathcal{A}$ starts making queries to its oracle $\mathcal{O}_{\mathtt{CRDM}}(\cdot,\cdot)$. Each query $(f_0, f_1) \in (\mathcal{B}_s)^2$ from $\mathcal{A}$ is answered as follows. For making it easier to describe the subsequent games, we change the ordering of the operations for how the public/secret key pair $(\mathsf{rpk}, \mathsf{rsk})$ and the ciphertext rct are generated, so that the distribution of the oracle's answer $(\mathsf{rpk}, \mathsf{rct})$ is identical to that in the original correlation $\mathcal{B}_s$-RDM security experiment.
   (a) Compute $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda)$.
   (b) Pick $\mathsf{A} \xleftarrow{\mathsf{r}} \{0,1\}^{3\lambda}$.
   (c) Pick $\mathsf{B} \xleftarrow{\mathsf{r}} \{0,1\}^m$, compute $(\mathsf{y}_0, (\mathsf{y}_i)_{i\in[n]}) \leftarrow \mathsf{KwPRF}(\mathsf{x}, \mathsf{B})$, and set $\mathsf{r}_0 \leftarrow \mathsf{y}_0$ and $\mathsf{r}_{i,\mathsf{x}_i} \leftarrow \mathsf{y}_i$ for all $i \in [n]$.
   (d) Pick $\mathsf{r}_{1,1-\mathsf{x}_1}, \ldots, \mathsf{r}_{n,1-\mathsf{x}_n} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$.[8]
   (e) For all $i \in [n]$, do the following:
      i. Compute $\mathsf{ct}_{i,\mathsf{x}_i} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}_{i,\mathsf{x}_i})$.
      ii. Pick $\mathsf{C}_i \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{ct}}}$.
      iii. Set $\mathsf{ct}_i \leftarrow \begin{cases} \mathsf{ct}_{i,0} & \text{if } \mathsf{x}_i = 0 \\ \mathsf{ct}_{i,1} \oplus \mathsf{C}_i & \text{if } \mathsf{x}_i = 1 \end{cases}$.
   (f) Set $\mathsf{z} \leftarrow \mathsf{r}_0 \oplus f_b(\mathsf{x})$.
   (g) Return $\mathsf{rpk} \leftarrow (\mathsf{pk}, \mathsf{A}, \mathsf{B}, (\mathsf{C}_i)_{i\in[n]})$ and $\mathsf{rct} \leftarrow (\mathsf{z}, (\mathsf{ct}_i)_{i\in[n]})$ to $\mathcal{A}$.
3. At some point, $\mathcal{A}$ terminates with output b'.

It is straightforward to see that an answer $(\mathsf{rpk}, \mathsf{rct})$ from $\mathcal{O}_{\mathtt{CRDM}}(\cdot,\cdot)$ are generated in exactly the same way as in the original security game regarding $\Pi_{\mathtt{RDM}}$. Hence, we have

$$\mathsf{Adv}^{\mathcal{B}_s\text{-}\mathtt{CRDM}}_{\Pi_{\mathtt{RDM}},\mathcal{A}}(\lambda) = 2 \cdot \left| \Pr[\mathtt{SUC}_1] - \frac{1}{2} \right|. \tag{2}$$

[8] The values $\{\mathsf{r}_{i,1-\mathsf{x}_i}\}_{i\in[n]}$ are not used in Game 1. Looking ahead, these will be used from Game 3.

Game 2: Same as Game 1, except that the value $\mathsf{C}_i \in \{0,1\}^{\ell_{\mathsf{ct}}}$ for every $i \in [n]$ that appears in the computation of $(\mathsf{rpk}, \mathsf{rct})$ by the oracle $\mathcal{O}_{\mathtt{CRDM}}$ as the response to $\mathcal{A}$'s query $(f_0, f_1) \in (\mathcal{B}_s)^2$, is generated as follows:

1. Pick $\mathsf{ct}_{i,1-\mathsf{x}_i} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{ct}}}$.
2. Set $\mathsf{C}_i \leftarrow \mathsf{ct}_{i,0} \oplus \mathsf{ct}_{i,1}$.

Note that by this change, for all $i \in [n]$, ct_i in rct now always satisfies $\mathsf{ct}_i = \mathsf{ct}_{i,0}$ (regardless of $\mathsf{x}_i = 0$ or $\mathsf{x}_i = 1$) in Game 2.

In Game 2, for all $i \in [n]$, $\mathsf{ct}_{i,1-\mathsf{x}_i}$ works as the key (mask) of the one-time pad encryption, and makes C_i distributed uniformly at random in $\{0,1\}^{\ell_{\mathsf{ct}}}$, which is exactly how this value in Game 1 is distributed. Consequently, $\mathcal{A}$'s view in Game 2 is distributed identically to that in Game 1, which implies

$$\Pr[\mathtt{SUC}_1] = \Pr[\mathtt{SUC}_2]. \tag{3}$$

Game 3: Same as Game 2, except that the value $\mathsf{ct}_{i,1-\mathsf{x}_i}$ for every $i \in [n]$ that appears in the computation of $(\mathsf{rpk}, \mathsf{rct})$ by the oracle $\mathcal{O}_{\mathtt{CRDM}}$ as the response to $\mathcal{A}$'s query $(f_0, f_1) \in (\mathcal{B}_s)^2$, is generated as $\mathsf{ct}_{i,1-\mathsf{x}_i} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}_{i,1-\mathsf{x}_i})$.

Note that by this change, for each $i \in [n]$, once $\mathsf{r}_{i,0}$ and $\mathsf{r}_{i,1}$ are generated/computed, $\mathsf{ct}_{i,v}$ is computed as $\mathsf{ct}_{i,v} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}_{i,v})$ for both $v \in \{0,1\}$.

The values $\{\mathsf{r}_{i,1-\mathsf{x}_i}\}_{i\in[n]}$ are not used in Game 2, and each of them is chosen uniformly at random. Hence, applying a query-wise hybrid argument, we can straightforwardly construct a PPT reduction algorithm $\mathcal{B}$ attacking the PRCT security of Π, such that

$$\left|\Pr[\mathtt{SUC}_2] - \Pr[\mathtt{SUC}_3]\right| = q \cdot \mathsf{Adv}^{\mathtt{PRCT}}_{\Pi,\mathcal{B}}(\lambda). \tag{4}$$

For completeness, we will provide the description and analysis of $\mathcal{B}$ after describing of the sequence of the games.

Game 4: Same as Game 3, except that the values $(\mathsf{z}, (\mathsf{r}_{i,0}, \mathsf{r}_{i,1})_{i\in[n]})$ that appear in the computation of $(\mathsf{rpk}, \mathsf{rct})$ by the oracle $\mathcal{O}_{\mathtt{CRDM}}$ as the response to $\mathcal{A}$'s query $(f_0, f_1) \in \mathcal{B}_s^2$, are picked from $\{0,1\}^{\ell_{\mathsf{m}}} \times (\{0,1\}^\lambda \times \{0,1\}^\lambda)^n$ uniformly at random.

Since the answers from the oracle $\mathcal{O}_{\mathtt{CRDM}}$ now contains no information of b, $\mathcal{A}$'s view is independent of b in Game 4. Hence, we have

$$\Pr[\mathtt{SUC}_4] = \frac{1}{2}. \tag{5}$$

Also, we can construct a PPT reduction algorithm $\mathcal{B}'$ attacking the $\mathcal{B}_{s+p}$-KDS security of KwPRF, such that

$$\left|\Pr[\mathtt{SUC}_3] - \Pr[\mathtt{SUC}_4]\right| = \mathsf{Adv}^{\mathcal{B}_{s+p}\text{-}\mathtt{KDS}\text{-}\mathtt{wPRF}}_{\mathsf{KwPRF},\mathcal{B}'}(\lambda). \tag{6}$$

We will provide the description and analysis of $\mathcal{B}'$ at the end of the proof.

Hence, Eqs. (2) to (6), together with the triangle inequality, imply Eq. (1).

We now provide the descriptions and analyses of $\mathcal{B}$ and $\mathcal{B}'$ used to derive Eqs. (4) and (6), respectively.

Firstly, the reduction $\mathcal{B}$ for Eq. (4), attacking the PRCT security of Π, is as follows:

$\mathcal{B}^{\mathcal{O}_{\mathtt{PRCT}}(\cdot)}(\mathsf{pk})$**:** $\mathcal{B}$ initially picks $\alpha \xleftarrow{\mathsf{r}} [q]$, $b \xleftarrow{\mathsf{r}} \{0,1\}$, and $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \xleftarrow{\mathsf{r}} \{0,1\}^n$, and then runs $\mathcal{A}(1^\lambda)$. When $\mathcal{A}$ makes the j-th query $(f_0^{(j)}, f_1^{(j)}) \in (\mathcal{B}_s)^2$ (where $j \in [q]$) to its oracle $\mathcal{O}_{\mathtt{CRDM}}$, $\mathcal{B}$ simulates the answer $(\mathsf{rpk}^{(j)}, \mathsf{rct}^{(j)})$ as follows:

1. If $j = \alpha$, then set $\mathsf{pk}^{(\alpha)} \leftarrow \mathsf{pk}$. Otherwise (i.e. $j \neq \alpha$), compute $(\mathsf{pk}^{(j)}, \mathsf{sk}^{(j)}) \xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda)$.
2. Pick $\mathsf{A}^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{3\lambda}$.
3. Pick $\mathsf{B}^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{ct}}}$, compute $(\mathsf{y}_0^{(j)}, (\mathsf{y}_i^{(j)})_{i\in[n]}) \leftarrow \mathsf{KwPRF}(\mathsf{x}, \mathsf{B}^{(j)})$, and set $\mathsf{r}_0^{(j)} \leftarrow \mathsf{y}_0^{(j)}$.
4. For all $i \in [n]$, do the following:
   (a) Set $\mathsf{r}_{i,\mathsf{x}_i}^{(j)} \leftarrow \mathsf{y}_i^{(j)}$, and compute $\mathsf{ct}_{i,\mathsf{x}_i}^{(j)} \leftarrow \mathsf{Enc}(\mathsf{pk}^{(j)}, \mathsf{A}^{(j)}; \mathsf{r}_{i,\mathsf{x}_i}^{(j)})$.
   (b) Compute $\mathsf{ct}_{i,1-\mathsf{x}_i}^{(j)}$ as follows.
   - If $j < \alpha$, then compute $\mathsf{ct}_{i,1-\mathsf{x}_i}^{(j)} \leftarrow \mathsf{Enc}(\mathsf{pk}^{(j)}, \mathsf{A}^{(j)}; \mathsf{r}_{i,1-\mathsf{x}_i}^{(j)})$, where $\mathsf{r}_{i,1-\mathsf{x}_i}^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$.
   - If $j = \alpha$, then submit a query $\mathsf{A}^{(\alpha)} \in \{0,1\}^{3\lambda}$ to the oracle $\mathcal{O}_{\mathtt{PRCT}}$ and receive the answer $\mathsf{ct}_{i,1-\mathsf{x}_i}^{(\alpha)}$.
   - If $j > \alpha$, then pick $\mathsf{ct}_{i,1-\mathsf{x}_i}^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{ct}}}$.

   (c) Set $\mathsf{C}_i^{(j)} \leftarrow \mathsf{ct}_{i,0}^{(j)} \oplus \mathsf{ct}_{i,1}^{(j)}$ and $\mathsf{ct}_i^{(j)} \leftarrow \mathsf{ct}_{i,0}^{(j)}$.
5. Set $\mathsf{z}^{(j)} \leftarrow \mathsf{r}_0^{(j)} \oplus f_b^{(j)}(\mathsf{x})$.
6. Set $\mathsf{rpk}^{(j)} \leftarrow (\mathsf{pk}^{(j)}, \mathsf{A}^{(j)}, \mathsf{B}^{(j)}, (\mathsf{C}_i^{(j)})_{i\in[n]})$ and $\mathsf{rct}^{(j)} \leftarrow (\mathsf{z}^{(j)}, (\mathsf{ct}_i^{(j)})_{i\in[n]})$.

When $\mathcal{A}$ terminates with output b', $\mathcal{B}$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be $\mathcal{B}$'s challenge bit in the PRCT security experiment regarding Π. It is straightforward to see that if $\alpha = 1 \wedge \beta = 0$ and thus all the values $\{\mathsf{ct}_{i,1-\mathsf{x}_i}^{(j)}\}_{i\in[n],j\in[q]}$ are just random strings, then $\mathcal{A}$'s view $\{(\mathsf{rpk}^{(j)}, \mathsf{rct}^{(j)})\}_{j\in[q]}$ is generated identically to that in Game 2, and thus $\mathcal{B}$ simulates Game 2 perfectly for $\mathcal{A}$. Since $\mathcal{B}$ outputs $\beta' = 1$ only when $\mathcal{A}$ succeeds in guessing b (chosen by $\mathcal{B}$), we have $\Pr[\beta' = 1 | \alpha = 1 \wedge \beta = 0] = \Pr[\mathtt{SUC}_2]$.

On the other hand, if $\alpha = q \wedge \beta = 1$ and thus all the values $\{\mathsf{ct}_{i,1-\mathsf{x}_i}^{(j)}\}_{i\in[n],j\in[q]}$ are computed as $\mathsf{ct}_{i,1-\mathsf{x}_i}^{(j)} \leftarrow \mathsf{Enc}(\mathsf{pk}^{(j)}, \mathsf{A}^{(j)}; \mathsf{r}_{i,1-\mathsf{x}_i}^{(j)})$ with a randomly chosen $\mathsf{r}_{i,1-\mathsf{x}_i}^{(j)}$ (where the randomnesses $\{\mathsf{r}_{i,1-\mathsf{x}_i}^{(q)}\}_{i\in[n]}$ behind $\{\mathsf{ct}_{i,1-\mathsf{x}_i}^{(q)}\}_{i\in[n]}$ are chosen by $\mathcal{B}$'s oracle $\mathcal{O}_{\mathtt{PRCT}}$), then $\mathcal{A}$'s view $\{(\mathsf{rpk}^{(j)}, \mathsf{rct}^{(j)})\}_{j\in[q]}$ is generated identically to that in Game 3, and thus $\mathcal{B}$ simulates Game 3 perfectly for $\mathcal{A}$. With a similar argument to the case $\alpha = 1 \wedge \beta = 0$, we have $\Pr[\beta' = 1 | \alpha = q \wedge \beta = 1] = \Pr[\mathtt{SUC}_3]$.

Furthermore, by our design of $\mathcal{B}$, it is easy to see that for all $j \in [q-1]$, $\mathcal{A}$'s view in the case $\alpha = j + 1 \wedge \beta = 0$ is generated identically to that in the case $\alpha = j \wedge \beta = 1$. Hence, we have $\Pr[\beta' = 1 | \alpha = j + 1 \wedge \beta = 0] = \Pr[\beta' = 1 | \alpha = j \wedge \beta = 1]$.

Moreover, since $\mathcal{B}$ chooses $\alpha \in [q]$ uniformly at random, independently of $\mathcal{B}$'s challenge bit β, for all $j \in [q]$, we have $\Pr[\alpha = j | \beta = 0] = \Pr[\alpha = j | \beta = 1] = 1/q$.

Combining the arguments so far, $\mathcal{B}$'s advantage can be calculated as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{PRCT}}_{\Pi,\mathcal{B}}(\lambda) &= \Big|\Pr[\beta' = 1|\beta = 0] - \Pr[\beta' = 1|\beta = 1]\Big| \\
&= \Big|\sum_{j\in[q]}\Big(\Pr[\beta' = 1 \wedge \alpha = j|\beta = 0] - \Pr[\beta' = 1 \wedge \alpha = j|\beta = 1]\Big)\Big| \\
&= \Big|\sum_{j\in[q]}\Big(\Pr[\beta' = 1| \wedge \alpha = j \wedge \beta = 0] \cdot \Pr[\alpha = j|\beta = 0] \\
&\qquad\qquad - \Pr[\beta' = 1|\alpha = j \wedge \beta = 1] \cdot \Pr[\alpha = j|\beta = 1]\Big)\Big| \\
&= \frac{1}{q} \cdot \Big|\sum_{j\in[q]}\Big(\Pr[\beta' = 1|\alpha = j \wedge \beta = 0] - \Pr[\beta' = 1|\alpha = j \wedge \beta = 1]\Big)\Big| \\
&= \frac{1}{q} \cdot \Big|\Pr[\beta' = 1|\alpha = 1 \wedge \beta = 0] - \Pr[\beta' = 1|\alpha = q \wedge \beta = 1]\Big| \\
&= \frac{1}{q} \cdot \Big|\Pr[\mathtt{SUC}_2] - \Pr[\mathtt{SUC}_3]\Big|,
\end{aligned}$$

which is equivalent to Eq. (4), as desired.

Finally, the reduction $\mathcal{B}'$ for Eq. (6), attacking the $\mathcal{B}_{s+p}$-KDS security of KwPRF, is as follows:

$\mathcal{B}'^{\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}(\cdot)}(1^\lambda)$**:** ($\mathcal{B}'$ will simulate Game 3 or Game 4 for $\mathcal{A}$ by implicitly regarding the key for KwPRF picked by $\mathcal{B}'$'s experiment as x in the corresponding game.) $\mathcal{B}'$ initially picks $b \xleftarrow{\mathsf{r}} \{0,1\}$, and then runs $\mathcal{A}(1^\lambda)$. When $\mathcal{A}$ makes a query $(f_0, f_1) \in (\mathcal{B}_s)^2$ to its oracle $\mathcal{O}_{\mathtt{CRDM}}$, $\mathcal{B}'$ simulates the answer $(\mathsf{rpk}, \mathsf{rct})$ as follows:

1. Compute $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\mathsf{r}} \mathsf{KG}(1^\lambda)$.
2. Pick $\mathsf{A} \xleftarrow{\mathsf{r}} \{0,1\}^{3\lambda}$.
3. Pick $\Delta_{1,0}, \Delta_{1,1}, \ldots, \Delta_{n,0}, \Delta_{n,1} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$, and construct a function $g : \{0,1\}^n \to \{0,1\}^{\ell_{\mathsf{m}}+n\lambda}$ that on input $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \in \{0,1\}^n$ outputs $(f_b(\mathsf{x}), \Delta_{1,\mathsf{x}_1}, \ldots, \Delta_{n,\mathsf{x}_n})$. Note that $f_b \in \mathcal{B}_s$ implies that $g \in \mathcal{B}_{s+p}$ holds, where p is the size of the selector function $\widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}] : \{0,1\}^n \to (\{0,1\}^\lambda)^n$ such that $\widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}](\mathsf{x}) = (\Delta_{1,\mathsf{x}_1}, \ldots, \Delta_{n,\mathsf{x}_n})$.
4. Submit a query g to the oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}$, and receive the answer $(\mathsf{B}, \mathsf{w}) \in \{0,1\}^m \times \{0,1\}^{\ell_{\mathsf{m}}+n\lambda}$, where we parse w as $(\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_n) \in \{0,1\}^{\ell_{\mathsf{m}}} \times (\{0,1\}^\lambda)^n$.
5. Set $\mathsf{z} \leftarrow \mathsf{w}_0$.
6. For all $i \in [n]$, do the following:
   (a) Set $\mathsf{r}_{i,v} \leftarrow \mathsf{w}_i \oplus \Delta_{i,v}$ for both $v \in \{0,1\}$.
   (b) Compute $\mathsf{ct}_{i,v} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}_{i,v})$ for both $v \in \{0,1\}$.
   (c) Set $\mathsf{C}_i \leftarrow \mathsf{ct}_{i,0} \oplus \mathsf{ct}_{i,1}$ and $\mathsf{ct}_i \leftarrow \mathsf{ct}_{i,0}$.
7. Set $\mathsf{rpk} \leftarrow (\mathsf{pk}, \mathsf{A}, \mathsf{B}, (\mathsf{C}_i)_{i\in[n]})$ and $\mathsf{rct} \leftarrow (\mathsf{z}, (\mathsf{ct}_i)_{i\in[n]})$.

When $\mathcal{A}$ terminates with output b', $\mathcal{B}'$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}'$.

Let $\beta \in \{0,1\}$ be $\mathcal{B}'$'s challenge bit in the $\mathcal{B}_{s+p}$-KDS security experiment regarding KwPRF, and let $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \in \{0,1\}^n$ be the key for KwPRF chosen in $\mathcal{B}'$'s experiment.

We first argue that if $\beta = 1$ and thus the value $\mathsf{w} = (\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_n) \in \{0,1\}^{\ell_\mathsf{m}} \times (\{0,1\}^\lambda)^n$ returned from $\mathcal{B}'$'s oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ is computed as $\mathsf{w} = \mathsf{KwPRF}(\mathsf{x}, \mathsf{B}) \oplus g(\mathsf{x})$, then $\mathcal{B}'$ simulates Game 3 perfectly for $\mathcal{A}$. Indeed, letting $\mathsf{y} = (\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) = \mathsf{KwPRF}(\mathsf{x}, \mathsf{B})$, we have

$$\mathsf{KwPRF}(\mathsf{x}, \mathsf{B}) \oplus g(\mathsf{x}) = (\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_n) = \Big(\mathsf{y}_0 \oplus f_b(\mathsf{x}),\ \mathsf{y}_1 \oplus \Delta_{1,\mathsf{x}_i},\ \ \ldots,\ \mathsf{y}_n \oplus \Delta_{n,\mathsf{x}_n}\Big),$$

and thus $\mathcal{B}'$ sets/computes z and $(\mathsf{r}_{i,0}, \mathsf{r}_{i,1})_{i\in[n]}$ as follows:

$$\mathsf{z} = \mathsf{w}_0 = \mathsf{y}_0 \oplus f_b(\mathsf{x}),$$
$$\forall (i, v) \in [n] \times \{0,1\}: \quad \mathsf{r}_{i,v} = \mathsf{w}_i \oplus \Delta_{i,v} = \begin{cases} \mathsf{y}_i & \text{if } v = \mathsf{x}_i \\ \mathsf{y}_i \oplus \Delta_{i,0} \oplus \Delta_{i,1} & \text{if } v = 1 - \mathsf{x}_i \end{cases}.$$

Note that from $\mathcal{A}$'s view point, $\mathsf{r}_{i,1-\mathsf{x}_i}$ looks like a truly random string due to the XOR with $\Delta_{i,0} \oplus \Delta_{i,1}$ that is in turn distributed uniformly at random. Thus, z and $(\mathsf{r}_{i,0}, \mathsf{r}_{i,1})_{i\in[n]}$ are distributed identically to those in Game 3. Hence, $\mathcal{B}'$ simulates Game 3 perfectly for $\mathcal{A}$, and we have $\Pr[\beta' = 1 | \beta = 1] = \Pr[\texttt{SUC}_3]$.

On the other hand, if $\beta = 0$ and thus $\mathsf{w} = (\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_n) \in \{0,1\}^{\ell_\mathsf{m}} \times (\{0,1\}^\lambda)^n$ is truly random, then the values $(\mathsf{z}, (\mathsf{r}_{i,0}, \mathsf{r}_{i,1})_{i\in[n]}) = (\mathsf{w}_0, (\mathsf{w}_i \oplus \Delta_{i,0}, \mathsf{w}_i \oplus \Delta_{i,1})_{i\in[n]})$ are distributed uniformly at random (due to the random choice of $\{\Delta_{i,v}\}_{i\in[n],v\in\{0,1\}}$), and thus $\mathcal{A}$'s view is generated identically to that in Game 4. Hence, we have $\Pr[\beta' = 1 | \beta = 0] = \Pr[\texttt{SUC}_4]$.

Thus, $\mathcal{B}'$'s advantage can be calculated as follows:

$$\begin{aligned} \mathsf{Adv}^{\mathcal{B}_{s+p}\text{-}\texttt{KDS-wPRF}}_{\mathsf{KwPRF},\mathcal{B}'}(\lambda) &= \Big|\Pr[\beta' = 1 | \beta = 1] - \Pr[\beta' = 1 | \beta = 0]\Big| \\ &= \Big|\Pr[\texttt{SUC}_3] - \Pr[\texttt{SUC}_4]\Big|, \end{aligned}$$

which is exactly Eq. (6), as desired. This completes the proof of the second item in the theorem statement.

Finally, we briefly explain how $\mathcal{P}$-KDS (resp. $\mathcal{L}$-KDS) security of KwPRF leads to the correlation $\mathcal{P}$-RDM (resp. $\mathcal{L}$-RDM) security of the PKE scheme $\Pi_{\texttt{RDM}}$. At this point, it should be obvious from the reduction $\mathcal{B}'$ described above that if each pair of functions (f_0, f_1) queried by $\mathcal{A}$ is a pair of projection functions (resp. affine functions), then the function g used by $\mathcal{B}'$ as its query to the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ is also a projection function (resp. an affine function). Thus, if each pair of functions (f_0, f_1) queried by $\mathcal{A}$ is a pair of projection functions (resp. affine functions), then the function used by the reduction algorithm $\mathcal{B}'$ as its query to the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ is also described by a projection function (resp. an affine function). Hence, $\mathcal{B}'$ can be straightforwardly converted to a reduction attacking the $\mathcal{P}$-KDS (resp. $\mathcal{L}$-KDS) security of KwPRF. Hence, the first item in the theorem statement can also be established. □ (**Theorem 6**)

5 Constructing a KDS Secure Weak PRF from KDM Secure SKE

In this section, we show that a KDS secure weak PRF can be constructed by using any (one-time) $\mathcal{P}$-KDM secure SKE. Our KDS secure weak PRF is built in two modular steps, using a SKE scheme with a new security notion that we call *pseudorandomness of ciphertexts against the receiver (receiver-PRCT security, for short)*, as an intermediate building block.

Specifically, first, in Section 5.1 we formally introduce the new security notion for SKE called receiver-PRCT security. Then, as the first step in our construction, we show in Section 5.2 how

to construct a SKE scheme that satisfies KDM security and receiver-PRCT security simultaneously, by using a weak PRF and a garbling scheme, and assuming the existence a one-time $\mathcal{P}$-KDM secure SKE scheme. The construction here relies on the KDM-amplification construction by Waters and Wichs [WW23]. Finally, as the second step in our construction, we show in Section 5.3 how to construct a KDS secure weak PRF from a SKE scheme that satisfies KDM security and receiver-PRCT security simultaneously.

5.1 New Security Notion for SKE: Pseudorandomness of Ciphertexts against the Receiver (Receiver-PRCT Security)

In this section, we give the formal definition of receiver-PRCT security for SKE. As the name suggests, this security notion guarantees that a ciphertext encrypting a random plaintext cannot be distinguished from a random element in the ciphertext space, even by the legitimate receiver of the ciphertext who holds a secret key.

The formal definition is as follows.

Let $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ be a SKE scheme whose secret key space is $\mathcal{K}$, plaintext space is $\mathcal{M}$, and ciphertext space is $\mathcal{C}$. We say that $\mathcal{E}$ satisfies *receiver-PRCT security* if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\text{R-PRCT}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{sk} \xleftarrow{r} \mathcal{K};\ \mathsf{m} \xleftarrow{r} \mathcal{M};\ \mathsf{ct}_1 \xleftarrow{r} \mathsf{SEnc}(\mathsf{sk}, \mathsf{m}) \\ \mathsf{ct}_0 \xleftarrow{r} \mathcal{C};\ b \xleftarrow{r} \{0,1\};\ b' \xleftarrow{r} \mathcal{A}(\mathsf{sk}, \mathsf{ct}_b) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible.

5.2 Constructing a SKE Scheme Satisfying KDM Security and Receiver-PRCT Security

Waters and Wichs [WW23] showed that for any polynomial $s = s(\lambda)$, we can construct a $\mathcal{B}_s$-KDM secure scheme, by using a SKE scheme satisfying pseudorandomness of ciphertexts (PRCT security) and a garbling scheme satisfying simulation security as building blocks, and assuming the existence of a (one-time) $\mathcal{P}$-KDM secure SKE scheme.

In this section, we show that if the underlying PRCT-secure SKE scheme is instantiated with the weak PRF-based SKE scheme (recalled in Section 2.2) and the underlying garbling scheme satisfies *blindness* [BLSV18] (in addition to simulation security), then the KDM-secure SKE scheme resulting from the Waters-Wichs construction also satisfies receiver-PRCT security.

We now formally describe the intermediate SKE construction that achieves the desired properties. Consider the following ingredients:

- Let $\mathsf{wPRF} : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a weak PRF.
- Let $\mathcal{G} = (\mathsf{Garble}, \mathsf{GEval}, \mathsf{GSim})$ be a garbling scheme.
- Let $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ be a one-time $\mathcal{P}$-KDM secure SKE scheme whose secret key space is $\{0,1\}^n$ for some polynomial $n = n(\lambda)$ and whose plaintext space is $\{0,1\}^{n+n\cdot\lambda}$. (This KDM secure SKE scheme is not directly used as a building block, but is necessary to *exist* for proving the KDM security of the Waters-Wichs construction. See Theorem 7.)
- Let $\ell_{\mathsf{m}} = \ell_{\mathsf{m}}(\lambda)$ be an arbitrary polynomial that represents the length of a plaintext that we wish to encrypt.

$\mathsf{SEnc}'(\mathsf{sk} \in \{0,1\}^{n+n\cdot\lambda}, \mathsf{m} \in \{0,1\}^{\ell_\mathsf{m}})$:
  Parse sk as $(\mathsf{z}, (\mathsf{k}_i)_{i\in[n]}) \in \{0,1\}^n \times (\{0,1\}^\lambda)^n$.
  Parse z as $(\mathsf{z}_1, \ldots, \mathsf{z}_n) \in \{0,1\}^n$.
  Let $C_\mathsf{m} : \{0,1\}^n \to \{0,1\}^{\ell_\mathsf{m}}$ be a constant circuit
    with size p such that $\forall \mathsf{x} \in \{0,1\}^n : C_\mathsf{m}(\mathsf{x}) = \mathsf{m}$.
  $(\widetilde{C}, (\mathsf{L})_{i\in[n],v\in\{0,1\}}) \xleftarrow{\mathsf{r}} \mathsf{Garble}(1^\lambda, C_\mathsf{m})$.
  $\forall i \in [n]$:
    $\mathsf{r}_{i,0}, \mathsf{r}_{i,1}, \mathsf{ct}_{i,1-\mathsf{z}_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$
    $\mathsf{ct}_{i,\mathsf{z}_i} \leftarrow \mathsf{wPRF}(\mathsf{k}_i, \mathsf{r}_{i,\mathsf{z}_i}) \oplus \mathsf{L}_{i,\mathsf{z}_i}$
  Return $\mathsf{CT} \leftarrow (\widetilde{C}, (\mathsf{r}_{i,v}, \mathsf{ct}_{i,v})_{i\in[n],v\in\{0,1\}})$.

$\mathsf{SDec}'(\mathsf{sk} \in \{0,1\}^{n+n\cdot\lambda}, \mathsf{CT})$:
  Parse sk as $(\mathsf{z}, (\mathsf{k}_i)_{i\in[n]}) \in \{0,1\}^n \times (\{0,1\}^\lambda)^n$.
  Parse z as $(\mathsf{z}_1, \ldots, \mathsf{z}_n) \in \{0,1\}^n$.
  Parse CT as $(\widetilde{C}, (\mathsf{r}_{i,v}, \mathsf{ct}_{i,v})_{i\in[n],v\in\{0,1\}})$.
  $\forall i \in [n]$:
    $\mathsf{L}'_i \leftarrow \mathsf{wPRF}(\mathsf{k}_i, \mathsf{r}_{i,\mathsf{z}_i}) \oplus \mathsf{ct}_{i,\mathsf{z}_i}$.
  Return $\mathsf{m} \leftarrow \mathsf{GEval}(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]})$.

Figure 4: The SKE scheme $\mathcal{E}'_p$ proposed by Waters and Wichs [WW23], in which the underlying PRCT secure SKE scheme is instantiated with the weak PRF-based SKE scheme.

Using them, the SKE scheme $\mathcal{E}'_p = (\mathsf{SEnc}', \mathsf{SDec}')$ proposed by Waters and Wichs [WW23], in which its underlying PRCT-secure SKE scheme is instantiated with the weak PRF-based SKE scheme, is constructed as in Fig. 4.

The constructed SKE scheme $\mathcal{E}'_p$ is parameterized by a polynomial $p = p(\lambda)$, which will be adjusted so that we can conduct the proof of its KDM security. (See Theorem 7 below.) The secret key space of $\mathcal{E}'_p$ is $\{0,1\}^{n+n\cdot\lambda}$, and its plaintext space is $\{0,1\}^{\ell_\mathsf{m}}$.

It was shown in [WW23] that $\mathcal{E}'_p$ satisfies correctness, based on the correctness of the buliding blocks. Furthermore, it was shown to satisfy KDM security as follows[9]:

Theorem 7 ([WW23]) *Assume that* wPRF *is a secure weak PRF, the garbling scheme* $\mathcal{G}$ *satisfies simulation security, and the SKE scheme* $\mathcal{E}$ *(which is used only in the proof) satisfies one-time* $\mathcal{P}$*-KDM security. Then, for any polynomial* $s = s(\lambda)$*, there exists a corresponding polynomial* $p = p(\lambda)$ *(that depends on* s *and the complexity of the algorithms of* $\mathcal{E}$*) such that the SKE scheme* $\mathcal{E}'_p$ *satisfies* $\mathcal{B}_s$*-KDM security. (Hence, adjusting the polynomial* s *appropriately,* $\mathcal{E}'_p$ *also satisfies* $\mathcal{P}$*-KDM security and* $\mathcal{L}$*-KDM security as well.)*

We now show that $\mathcal{E}'_p$ satisfies receiver-PRCT security as well.

Theorem 8 *Let* $p = p(\lambda) \geq \max\{n, \ell_\mathsf{m}\}$ *be an arbitrary polynomial. If the garbling scheme* $\mathcal{G}$ *satisfies simulation security and blindness simultaneously, then the SKE scheme* $\mathcal{E}'_p$ *satisfies receiver-PRCT security.*

Proof of Theorem 8. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the receiver-PRCT security of $\mathcal{E}'_p$. We will show that for this $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ that satisfy

$$\mathsf{Adv}^{\mathtt{R\text{-}PRCT}}_{\mathcal{E}'_p,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathtt{Sim}}_{\mathcal{G},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathtt{Blind}}_{\mathcal{G},\mathcal{B}'}(\lambda), \tag{7}$$

[9] Waters and Wichs [WW23] actually showed that their SKE construction achieves multi-key variant of KDM security (i.e. KDM security in the presence of multiple secret keys), which is stronger than KDM security in the single key setting treated in this paper. However, we do not recall this more general result since KDM security in the single key setting is sufficient for our purpose.

which will ensure that $\mathcal{A}$'s advantage is negligible by our assumption that $\mathcal{G}$ satisfies simulation security and blindness simultaneously, and hence imply the theorem.

Our proof is via a sequence of games argument using four games. In the following, let $\mathtt{T}_t$ denote the event that $\mathcal{A}$ outputs $b' = 1$ in Game $t \in [4]$.

Game 1: This is the receiver-PRCT security experiment with the challenge bit $b = 1$.

Specifically, the description of Game 1 is as follows:

1. Pick $\mathsf{sk} = (\mathsf{z} = (\mathsf{z}_1, \ldots, \mathsf{z}_n), (\mathsf{k}_i)_{i\in[n]}) \xleftarrow{\mathsf{r}} \{0,1\}^n \times (\{0,1\}^\lambda)^n$.
2. Pick $\mathsf{m} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$.
3. Let $C_\mathsf{m} : \{0,1\}^n \to \{0,1\}^{\ell_\mathsf{m}}$ be a constant circuit with size p such that $\forall \mathsf{x} \in \{0,1\}^n : C_\mathsf{m}(\mathsf{x}) = \mathsf{m}$.
4. Compute $(\widetilde{C}, (\mathsf{L}_{i,v})_{i\in[n],v\in\{0,1\}}) \xleftarrow{\mathsf{r}} \mathsf{Garble}(1^\lambda, C_\mathsf{m})$.
5. For all $i \in [n]$, do the following:
   (a) Pick $\mathsf{r}_{i,0}, \mathsf{r}_{i,1}, \mathsf{ct}_{i,1-\mathsf{z}_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$.
   (b) Compute $\mathsf{ct}_{i,\mathsf{z}_i} \leftarrow \mathsf{wPRF}(\mathsf{k}_i, \mathsf{r}_{i,\mathsf{z}_i}) \oplus \mathsf{L}_{i,\mathsf{z}_i}$.
6. Set $\mathsf{CT} \leftarrow (\widetilde{C}, (\mathsf{r}_{i,v}, \mathsf{ct}_{i,v})_{i\in[n],v\in\{0,1\}})$.
7. Run $b' \xleftarrow{\mathsf{r}} \mathcal{A}(\mathsf{sk}, \mathsf{CT})$.

Looking ahead, Game 4 will correspond to the receiver-PRCT security experiment with the challenge bit $b = 0$. That is, we have

$$\mathsf{Adv}^{\mathtt{R\text{-}PRCT}}_{\mathcal{E}'_p,\mathcal{A}}(\lambda) = \Big|\Pr[\mathtt{T}_1] - \Pr[\mathtt{T}_4]\Big|. \tag{8}$$

Game 2: Same as Game 1, except that $\widetilde{C}$ and $\{\mathsf{L}_{i,\mathsf{z}_i}\}_{i\in[n]}$ are generated by $(\widetilde{C}, (\mathsf{L}_{i,\mathsf{z}_i})_{i\in[n]}) \xleftarrow{\mathsf{r}} \mathsf{GSim}(1^\lambda, \mathsf{m})$.

Note that the labels $\{\mathsf{L}_{i,1-\mathsf{z}_i}\}_{i\in[n]}$ are not at all used in Game 1. Hence, we can straightforwardly construct a PPT reduction algorithm $\mathcal{B}$ attacking the simulation security of $\mathcal{G}$, such that

$$\Big|\Pr[\mathtt{T}_1] - \Pr[\mathtt{T}_2]\Big| = \mathsf{Adv}^{\mathtt{Sim}}_{\mathcal{G},\mathcal{B}}(\lambda). \tag{9}$$

For completeness, we will provide the description and analysis of $\mathcal{B}$ after describing the sequence of the games.

Game 3: Same as Game 2, except that $\widetilde{C}$ and $\{\mathsf{L}_{i,\mathsf{z}_i}\}_{i\in[n]}$ are replaced with truly random strings of the same length as the corresponding values in Game 2, respectively.

In Game 2, the input to GSim, i.e. $\mathsf{m} \in \{0,1\}^{\ell_\mathsf{m}}$, is chosen uniformly at random. Hence, we can again straightforwardly construct a PPT reduction algorithm $\mathcal{B}'$ attacking the blindness of $\mathcal{G}$, such that

$$\Big|\Pr[\mathtt{T}_2] - \Pr[\mathtt{T}_3]\Big| = \mathsf{Adv}^{\mathtt{Blind}}_{\mathcal{G},\mathcal{B}'}(\lambda). \tag{10}$$

Again for completeness, we will provide the description and analysis of $\mathcal{B}'$ after describing of the sequence of the games.

Game 4: Same as Game 3, except that for all $i \in [n]$, $\mathsf{ct}_{i,\mathsf{z}_i} \in \{0,1\}^\lambda$ is chosen uniformly at random.

In Game 3, $\mathsf{ct}_{i,\mathsf{z}_i}$ is computed as $\mathsf{ct}_{i,\mathsf{z}_i} = \mathsf{wPRF}(\mathsf{k}_i, \mathsf{r}_{i,\mathsf{z}_i}) \oplus \mathsf{L}_{i,\mathsf{z}_i}$, and $\mathsf{L}_{i,\mathsf{z}_i}$ is chosen uniformly at random (independently of any other values). Hence, for all $i \in [n]$, $\mathsf{L}_{i,\mathsf{z}_i}$ works as the key (mask) of the one-time pad encryption, and makes $\mathsf{ct}_{i,\mathsf{z}_i}$ look like a truly random string in $\{0,1\}^\lambda$. Consequently, $\mathcal{A}$'s view in Game 4 is distributed identically to that in Game 3, which implies

$$\Pr[\mathtt{T}_3] = \Pr[\mathtt{T}_4]. \tag{11}$$

In Game 4, all the values in $\mathcal{A}$'s view $(\mathsf{sk}, \mathsf{CT} = (\widetilde{C}, (\mathsf{r}_{i,v}, \mathsf{ct}_{i,v})_{i\in[n],v\in\{0,1\}}))$ are just truly random strings, which is exactly how these values are generated in the receiver-PRCT security experiment with the challenge bit $b = 0$. (Thus, we indeed obtain Eq. (8).)

Hence, Eqs. (8) to (11), together with the triangle inequality, imply Eq. (7).

For completeness, we now provide the descriptions and analyses of $\mathcal{B}$ and $\mathcal{B}'$ used to derive Eqs. (9) and (10), respectively.

Firstly, the reduction $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ for Eq. (9), attacking the simulation security of $\mathcal{G}$, is as follows:

$\mathcal{B}_1(1^\lambda)$**:** $\mathcal{B}_1$ picks $\mathsf{m} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$ and $\mathsf{z} = (\mathsf{z}_1, \ldots, \mathsf{z}_n) \xleftarrow{\mathsf{r}} \{0,1\}^n$, and generates a constant circuit $C_\mathsf{m} : \{0,1\}^n \to \{0,1\}^{\ell_\mathsf{m}}$ exactly as done in SEnc', sets $\mathsf{st}_\mathcal{B}$ as all the information known to $\mathcal{B}_1$, and terminates with output $(C_\mathsf{m}, \mathsf{z}, \mathsf{st}_\mathcal{B})$.

$\mathcal{B}_2(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]}, \mathsf{st}_\mathcal{B})$**:** $\mathcal{B}_2$ picks $\mathsf{k}_1, \ldots, \mathsf{k}_n \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$. Next, for all $i \in [n]$, $\mathcal{B}_2$ picks $\mathsf{r}_{i,0}, \mathsf{r}_{i,1}, \mathsf{ct}_{i,1-\mathsf{z}_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$, and computes $\mathsf{ct}_{i,\mathsf{z}_i} \leftarrow \mathsf{wPRF}(\mathsf{k}_i, \mathsf{r}_{i,\mathsf{z}_i}) \oplus \mathsf{L}'_i$. Then, $\mathcal{B}_2$ sets $\mathsf{sk} \leftarrow (\mathsf{z}, (\mathsf{k}_i)_{i\in[n]})$ and $\mathsf{CT} \leftarrow (\widetilde{C}, (\mathsf{r}_{i,v}, \mathsf{ct}_{i,v})_{i\in[n],v\in\{0,1\}})$, and runs $b' \xleftarrow{\mathsf{r}} \mathcal{A}(\mathsf{sk}, \mathsf{CT})$. Finally, $\mathcal{B}_2$ sets $\beta' \leftarrow b'$, and terminates with output β'.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be $\mathcal{B}$'s challenge bit in the simulation security experiment regarding $\mathcal{G}$. It is straightforward to see that if $\beta = 1$ and thus $\widetilde{C}$ and $(\mathsf{L}'_i)_{i\in[n]}$ are generated as $(\widetilde{C}, (\mathsf{L}_{i,v})_{i\in[n],v\in\{0,1\}}) \xleftarrow{\mathsf{r}} \mathsf{Garble}(1^\lambda, C_\mathsf{m})$ and $(\mathsf{L}'_i)_{i\in[n]} \leftarrow (\mathsf{L}_{i,\mathsf{z}_i})_{i\in[n]}$, then $\mathcal{A}$'s view $(\mathsf{sk}, \mathsf{CT})$ is generated identically to that in Game 1. Hence, we have $\Pr[\beta' = 1 | \beta = 1] = \Pr[\mathtt{T}_1]$. On the other hand, if $\beta = 0$ and thus $\widetilde{C}$ and $(\mathsf{L}'_i)_{i\in[n]}$ are generated as $(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]}) \xleftarrow{\mathsf{r}} \mathsf{GSim}(1^\lambda, \mathsf{m})$ with a random $\mathsf{m} \in \{0,1\}^{\ell_\mathsf{m}}$ (chosen by $\mathcal{B}_1$), then $\mathcal{A}$'s view $(\mathsf{sk}, \mathsf{CT})$ is generated identically to that in Game 2. Hence, we have $\Pr[\beta' = 1 | \beta = 0] = \Pr[\mathtt{T}_2]$. Thus, $\mathcal{B}$'s advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{Sim}}_{\mathcal{G},\mathcal{B}}(\lambda) = \Big|\Pr[\beta' = 1 | \beta = 1] - \Pr[\beta' = 1 | \beta = 0]\Big| = \Big|\Pr[\mathtt{T}_1] - \Pr[\mathtt{T}_2]\Big|,$$

which is exactly Eq. (9), as desired.

Finally, the reduction $\mathcal{B}'$ for Eq. (10), attacking the blindness of $\mathcal{G}$, is as follows:

$\mathcal{B}'(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]})$**:** $\mathcal{B}'$ picks $\mathsf{z} = (\mathsf{z}_1, \ldots, \mathsf{z}_n) \xleftarrow{\mathsf{r}} \{0,1\}^n$ and $\mathsf{k}_1, \ldots, \mathsf{k}_n \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$. Next, for all $i \in [n]$, $\mathcal{B}'$ picks $\mathsf{r}_{i,0}, \mathsf{r}_{i,1}, \mathsf{ct}_{i,1-\mathsf{z}_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$, and computes $\mathsf{ct}_{i,\mathsf{z}_i} \leftarrow \mathsf{wPRF}(\mathsf{k}_i, \mathsf{r}_{i,\mathsf{z}_i}) \oplus \mathsf{L}'_i$. Then, $\mathcal{B}'$ sets $\mathsf{sk} \leftarrow (\mathsf{z}, (\mathsf{k}_i)_{i\in[n]})$ and $\mathsf{CT} \leftarrow (\widetilde{C}, (\mathsf{r}_{i,v}, \mathsf{ct}_{i,v})_{i\in[n],v\in\{0,1\}})$, and runs $b' \xleftarrow{\mathsf{r}} \mathcal{A}(\mathsf{sk}, \mathsf{CT})$. Finally, $\mathcal{B}'$ sets $\beta' \leftarrow b'$, and terminates with output β'.

The above completes the description of $\mathcal{B}'$.

Let $\beta \in \{0,1\}$ be $\mathcal{B}'$'s challenge bit in the blindness experiment regarding $\mathcal{G}$. It is straightforward to see that if $\beta = 1$ and thus $\widetilde{C}$ and $(\mathsf{L}'_i)_{i\in[n]}$ are generated as $(\widetilde{C}, (\mathsf{L}'_i)_{i\in[n]}) \xleftarrow{r} \mathsf{GSim}(1^\lambda, \mathsf{m})$ for a random $\mathsf{m} \in \{0,1\}^{\ell_\mathsf{m}}$ (chosen by $\mathcal{B}'$'s experiment), then $\mathcal{A}$'s view $(\mathsf{sk}, \mathsf{CT})$ is generated identically to that in Game 2. Hence, we have $\Pr[\beta' = 1|\beta = 1] = \Pr[\mathtt{T}_2]$. On the other hand, if $\beta = 0$ and thus $\widetilde{C}$ and $(\mathsf{L}'_i)_{i\in[n]}$ are just random strings, then $\mathcal{A}$'s view $(\mathsf{sk}, \mathsf{CT})$ is generated identically to that in Game 3. Hence, we have $\Pr[\beta' = 1|\beta = 0] = \Pr[\mathtt{T}_3]$. Thus, $\mathcal{B}'$'s advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{Blind}}_{\mathcal{G},\mathcal{B}'}(\lambda) = \Big|\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0]\Big| = \Big|\Pr[\mathtt{T}_2] - \Pr[\mathtt{T}_3]\Big|,$$

which is exactly Eq. (10), as desired. □ (**Theorem 8**)

5.3 Constructing a KDS Secure Weak PRF

In this section, we show how to construct a KDS secure weak PRF from a SKE scheme that satisfies KDM security and receiver-PRCT security simultaneously.

Let $\mathcal{E} = (\mathsf{SEnc}, \mathsf{SDec})$ be a SKE scheme whose secret key space is $\{0,1\}^n$, plaintext space is $\{0,1\}^{\ell_\mathsf{m}}$, and ciphertext space is $\{0,1\}^{\ell_\mathsf{ct}}$, for some polynomials $n = n(\lambda)$, $\ell_\mathsf{m} = \ell_\mathsf{m}(\lambda)$, and $\ell_\mathsf{ct} = \ell_\mathsf{ct}(\lambda)$. For simplicity, we assume that $\mathsf{SDec}(\mathsf{sk}, \mathsf{ct})$ always returns a string in $\{0,1\}^{\ell_\mathsf{m}}$ as long as $\mathsf{sk} \in \{0,1\}^n$ and $\mathsf{ct} \in \{0,1\}^{\ell_\mathsf{ct}}$.[10]

Using $\mathcal{E}$ as a building block, our KDS secure weak PRF $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^{\ell_\mathsf{ct}} \to \{0,1\}^{\ell_\mathsf{m}}$ is defined by

$$\forall(\mathsf{k}, \mathsf{x}) \in \{0,1\}^n \times \{0,1\}^{\ell_\mathsf{ct}} : \quad \mathsf{KwPRF}(\mathsf{k}, \mathsf{x}) := \mathsf{SDec}(\mathsf{k}, \mathsf{x}).$$

The KDS security of KwPRF is guaranteed by the following theorem.

Theorem 9 *If the SKE scheme $\mathcal{E}$ satisfies receiver-PRCT security, then the following holds:*

- *Let $\mathcal{F} \in \{\mathcal{P}, \mathcal{L}\}$. If the SKE scheme $\mathcal{E}$ additionally satisfies $\mathcal{F}$-KDM security, then KwPRF is a $\mathcal{F}$-KDS secure weak PRF.*
- *Let $s = s(\lambda)$ be an arbitrary polynomial. If the SKE scheme $\mathcal{E}$ additionally satisfies $\mathcal{B}_{s+\ell_\mathsf{m}}$-KDM security, then KwPRF is a $\mathcal{B}_s$-KDS secure weak PRF.*

Proof of Theorem 9. Since the proofs for both of the items are essentially the same, we mainly show the proof for the second item (i.e. $\mathcal{B}_s$-KDS security of KwPRF), and briefly explain how the first item (i.e. $\mathcal{F}$-KDS security of KwPRF for $\mathcal{F} \in \{\mathcal{P}, \mathcal{L}\}$) can be proved.

Let $s = s(\lambda)$ be an arbitrary polynomial. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the $\mathcal{B}_s$-KDS security of KwPRF and makes q queries to the oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}(\cdot)$. We will show that for this $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ that satisfy

$$\mathsf{Adv}^{\mathcal{B}_s\text{-}\mathtt{KDS\text{-}wPRF}}_{\mathsf{KwPRF},\mathcal{A}}(\lambda) \le 2q \cdot \mathsf{Adv}^{\mathtt{R\text{-}PRCT}}_{\mathcal{E},\mathcal{B}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathcal{B}_{s+\ell_\mathsf{m}}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{B}'}(\lambda), \tag{12}$$

which will ensure that $\mathcal{A}$'s advantage is negligible by our assumption that $\mathcal{E}$ satisfies receiver-PRCT security and $\mathcal{B}_{s+\ell_\mathsf{m}}$-KDM security simultaneously, and hence imply the second item of the theorem.

Our proof is via a sequence of games argument using four games. In the following, let $\mathtt{SUC}_t$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game $t \in [4]$.

[10] This requirement on SDec is without loss of generality, since if the output of SDec does not follow this requirement, we can replace it with some fixed value $\mathsf{m}_0 \in \{0,1\}^{\ell_\mathsf{m}}$. (Note that this modification does no harm to correctness, KDM security, and receiver-PRCT security of the original SKE scheme.)

Game 1: This is the $\mathcal{B}_s$-KDS security experiment regarding KwPRF.

Specifically, the description of Game 1 is as follows:

1. Pick $\mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n$ and $b \xleftarrow{\mathsf{r}} \{0,1\}$.
2. Run $\mathcal{A}^{\mathcal{O}_{\texttt{KDS-wPRF}}(\cdot)}(1^\lambda)$. Then, $\mathcal{A}$ starts making queries to its oracle $\mathcal{O}_{\texttt{KDS-wPRF}}(\cdot)$. Each query $f \in \mathcal{B}_s$ from $\mathcal{A}$ is answered as follows:
   (a) Pick $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{ct}}}$.
   (b) Compute $\mathsf{y}_1 \leftarrow \mathsf{SDec}(\mathsf{k}, \mathsf{x}) \oplus f(\mathsf{k})$.
   (c) Pick $\mathsf{y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{m}}}$.
   (d) Return $(\mathsf{x}, \mathsf{y}_b)$.
3. At some point, $\mathcal{A}$ terminates with output b'.

By definition, we have

$$\mathsf{Adv}^{\mathcal{B}_s\text{-}\texttt{KDS-wPRF}}_{\mathsf{KwPRF},\mathcal{A}}(\lambda) = 2 \cdot \left|\Pr[\texttt{SUC}_1] - \frac{1}{2}\right|. \tag{13}$$

Game 2: Same as Game 1, except that $\mathsf{x} \in \{0,1\}^{\ell_{\mathsf{ct}}}$ and $\mathsf{y}_1 \in \{0,1\}^{\ell_{\mathsf{m}}}$ that appear in the computation by the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ as the response to $\mathcal{A}$'s query $f \in \mathcal{B}_s$, are generated as follows:

1. Pick $\mathsf{y} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{m}}}$.
2. Compute $\mathsf{y}_1 \leftarrow \mathsf{y} \oplus f(\mathsf{k})$.
3. Compute $\mathsf{x} \xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{k}, \mathsf{y})$.

Comparing the way x and y_1 are generated in Game 2 and that in Game 1 and applying a query-wise hybrid argument, we can straightforwardly construct a PPT reduction algorithm $\mathcal{B}$ attacking the receiver-PRCT security of $\mathcal{E}$, such that

$$\left|\Pr[\texttt{SUC}_1] - \Pr[\texttt{SUC}_2]\right| = q \cdot \mathsf{Adv}^{\texttt{R-PRCT}}_{\mathcal{E},\mathcal{B}}(\lambda). \tag{14}$$

For completeness, we will provide the description and analysis of $\mathcal{B}$ after describing of the sequence of the games.

Game 3: Same as Game 2, except that $\mathsf{y}_1 \in \{0,1\}^{\ell_{\mathsf{m}}}$ and $\mathsf{y} \in \{0,1\}^{\ell_{\mathsf{m}}}$ that appear in the computation by the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ as the response to $\mathcal{A}$'s query $f \in \mathcal{B}_s$, are generated as follows:

1. Pick $\mathsf{y}_1 \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{m}}}$.
2. Compute $\mathsf{y} \leftarrow \mathsf{y}_1 \oplus f(\mathsf{k})$.

The distribution of $(\mathsf{y}, \mathsf{y}_1)$ in Game 3 and that in Game 2 are clearly identical, and thus we have

$$\Pr[\texttt{SUC}_2] = \Pr[\texttt{SUC}_3]. \tag{15}$$

Game 4: Same as Game 3, except that x that appears in the computation by the oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ as the response to $\mathcal{A}$'s query $f \in \mathcal{B}_s$, is generated as $\mathsf{x} \xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{k}, 1^{\ell_{\mathsf{m}}})$.

Notice that in Game 3, given $\mathsf{y}_1 \in \{0,1\}^{\ell_{\mathsf{m}}}$, the "plaintext" $\mathsf{y} = \mathsf{y}_1 \oplus f(\mathsf{k})$ encrypted in the "ciphertext" x, can be computed by a circuit with size $s + \ell_{\mathsf{m}}$ (since f can be computed by

a size-s circuit and XORing with $\mathsf{y}_1 \in \{0,1\}^{\ell_\mathsf{m}}$ can be realized by ℓ_m XOR gates), using the "secret key" $\mathsf{k} \in \{0,1\}^n$ as an input. Hence, we can again straightforwardly construct a PPT reduction algorithm $\mathcal{B}'$ attacking the $\mathcal{B}_{s+\ell_\mathsf{m}}$-KDM security of $\mathcal{E}$, such that

$$\left|\Pr[\mathtt{SUC}_3] - \Pr[\mathtt{SUC}_4]\right| = \mathsf{Adv}^{\mathcal{B}_{s+\ell_\mathsf{m}}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{B}'}(\lambda). \tag{16}$$

Again for completeness, we will provide the description and analysis of $\mathcal{B}'$ after describing of the sequence of the games.

In Game 4, x and y_1 generated as an answer (to $\mathcal{A}$'s query $f \in \mathcal{B}_s$) by the oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}$ are generated independently. Furthermore, already in Game 3, y_1 and y_0 are just independent random strings and thus distributed identically. Hence, the challenge bit b is information-theoretically hidden from $\mathcal{A}$'s view in Game 4, which implies

$$\Pr[\mathtt{SUC}_4] = \frac{1}{2}. \tag{17}$$

Hence, Eqs. (13) to (17), together with the triangle inequality, imply Eq. (12).

For completeness, we now provide the descriptions and analyses of $\mathcal{B}$ and $\mathcal{B}'$ used to derive Eqs. (14) and (16), respectively.

Firstly, the reduction $\mathcal{B}$ for Eq. (14), attacking the receiver-PRCT security of $\mathcal{E}$, is as follows:

$\mathcal{B}(\mathsf{sk},\mathsf{ct})$**:** $\mathcal{B}$ initially picks $\alpha \xleftarrow{\mathsf{r}} [q]$, and then sets $\mathsf{k} \leftarrow \mathsf{sk}$. Then, $\mathcal{B}$ picks $b \xleftarrow{\mathsf{r}} \{0,1\}$, and runs $\mathcal{A}(1^\lambda)$.

When $\mathcal{A}$ makes the j-th query $f^{(j)} \in \mathcal{B}_s$ (where $j \in [q]$), $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ generates $\mathsf{x}^{(j)} \in \{0,1\}^{\ell_\mathsf{ct}}$ and $\mathsf{y}^{(j)} \in \{0,1\}^{\ell_\mathsf{m}}$ as follows:
   - If $j < \alpha$, then pick $\mathsf{y}^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$, and compute $\mathsf{x}^{(j)} \xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{k},\mathsf{y}^{(j)})$.
   - If $j = \alpha$, then set $\mathsf{x}^{(\alpha)} \leftarrow \mathsf{ct}$ (where this ct is the one that $\mathcal{B}$ initially receives as input), and compute $\mathsf{y}^{(\alpha)} \leftarrow \mathsf{SDec}(\mathsf{k},\mathsf{x}^{(\alpha)})$.
   - If $j > \alpha$, then pick $\mathsf{x}^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{ct}}$, and compute $\mathsf{y}^{(j)} \leftarrow \mathsf{SDec}(\mathsf{k},\mathsf{x}^{(j)})$.
2. Then, $\mathcal{B}$ computes $\mathsf{y}_1^{(j)} \leftarrow \mathsf{y}^{(j)} \oplus f^{(j)}(\mathsf{k})$, picks $\mathsf{y}_0^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$, and then returns $(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})$ to $\mathcal{A}$ as the answer to the j-th query.

When $\mathcal{A}$ terminates with output b', $\mathcal{B}$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be $\mathcal{B}$'s challenge bit in the receiver-PRCT security experiment regarding $\mathcal{E}$. We first note that if $\alpha = 1 \wedge \beta = 0$ and thus $\mathsf{k} = \mathsf{sk}$ and $\mathsf{x}^{(1)} = \mathsf{ct}$ are just random strings, then $\mathcal{A}$'s view $\{(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})\}_{j\in[q]}$ is distributed identically to that in Game 1, and thus $\mathcal{B}$ simulates Game 1 perfectly for $\mathcal{A}$. Since $\mathcal{B}$ outputs $\beta' = 1$ only when $b' = b$ occurs, we have $\Pr[\beta' = 1 | \alpha = 1 \wedge \beta = 0] = \Pr[\mathtt{SUC}_1]$.

Next, we argue that if $\alpha = q \wedge \beta = 1$ and thus $\mathsf{k} = \mathsf{sk}$ and $\mathsf{x}^{(q)} = \mathsf{ct}$ are generated as $\mathsf{sk} \xleftarrow{\mathsf{r}} \{0,1\}^n$, $\mathsf{y} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$, and $\mathsf{ct} \xleftarrow{\mathsf{r}} \mathsf{SEnc}(\mathsf{sk},\mathsf{y})$, then $\mathcal{A}$'s view $\{(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})\}_{j\in[q]}$ is distributed identically to that in Game 2. To see this, notice that the answers to the first to $(q-1)$-th queries, i.e. $\{(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})\}_{j\in[q-1]}$, are generated exactly as in Game 2. Furthermore, regarding the answer to the q-th query $f^{(q)}$, $\mathsf{SDec}(\mathsf{k},\mathsf{x}^{(q)}) = \mathsf{y}$ holds by the correctness of $\mathcal{E}$, and this y is then used as $\mathsf{y}^{(q)}$. Hence, $\mathsf{y}^{(q)}$ that $\mathcal{B}$ computes during the course of generating $\mathsf{y}_1^{(q)}$, is just a random string (chosen by $\mathcal{B}$'s experiment), which is used to compute $\mathsf{y}_1^{(q)} = \mathsf{y}^{(q)} \oplus f^{(q)}(\mathsf{k})$.

Thus, the distribution of all of $\{(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})\}_{j\in[q]}$ generated by $\mathcal{B}$ is identical to that of $\mathcal{A}$'s view in Game 2, and thus $\mathcal{B}$ simulates Game 2 perfectly for $\mathcal{A}$. With a similar argument to the case $\alpha = 1 \wedge \beta = 0$, we have $\Pr[\beta' = 1 | \alpha = q \wedge \beta = 1] = \Pr[\mathtt{SUC}_2]$.

Furthermore, due to how $\mathcal{B}$ answers $\mathcal{A}$'s queries, for all $j \in [q-1]$, $\mathcal{A}$'s view in the case $\alpha = j+1 \wedge \beta = 0$, is identical to that in the case $\alpha = j \wedge \beta = 1$. Hence, we have $\Pr[\beta' = 1 | \alpha = j+1 \wedge \beta = 0] = \Pr[\beta' = 1 | \alpha = j \wedge \beta = 1]$ for all $j \in [q-1]$.

Moreover, since $\mathcal{B}$ chooses $\alpha \in [q]$ uniformly at random, independently of $\mathcal{B}$'s challenge bit β, for all $j \in [q]$, we have $\Pr[\alpha = j | \beta = 0] = \Pr[\alpha = j | \beta = 1] = 1/q$.

Combining the arguments so far, $\mathcal{B}$'s advantage can be calculated as follows:

$$
\begin{aligned}
\mathsf{Adv}^{\text{R-PRCT}}_{\mathcal{E},\mathcal{B}}(\lambda) &= \Big|\Pr[\beta' = 1 | \beta = 0] - \Pr[\beta' = 1 | \beta = 1]\Big| \\
&= \Big|\sum_{j\in[q]} \Big(\Pr[\beta' = 1 \wedge \alpha = j | \beta = 0] - \Pr[\beta' = 1 \wedge \alpha = j | \beta = 1]\Big)\Big| \\
&= \Big|\sum_{j\in[q]} \Big(\Pr[\beta' = 1 | \wedge \alpha = j \wedge \beta = 0] \cdot \Pr[\alpha = j | \beta = 0] \\
&\qquad\qquad - \Pr[\beta' = 1 | \alpha = j \wedge \beta = 1] \cdot \Pr[\alpha = j | \beta = 1]\Big)\Big| \\
&= \frac{1}{q} \cdot \Big|\sum_{j\in[q]} \Big(\Pr[\beta' = 1 | \alpha = j \wedge \beta = 0] - \Pr[\beta' = 1 | \alpha = j \wedge \beta = 1]\Big)\Big| \\
&= \frac{1}{q} \cdot \Big|\Pr[\beta' = 1 | \alpha = 1 \wedge \beta = 0] - \Pr[\beta' = 1 | \alpha = q \wedge \beta = 1]\Big| \\
&= \frac{1}{q} \cdot \Big|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]\Big|,
\end{aligned}
$$

which is equivalent to Eq. (14), as desired.

Finally, the reduction $\mathcal{B}'$ for Eq. (16), attacking the $\mathcal{B}_{s+\ell_\mathsf{m}}$-KDM security of $\mathcal{E}$, is as follows:

$\mathcal{B}'^{\mathcal{O}_{\mathtt{KDM}}(\cdot,\cdot)}(1^\lambda)$**:** $\mathcal{B}'$ picks $b \xleftarrow{\mathsf{r}} \{0,1\}$, and runs $\mathcal{A}(1^\lambda)$, where $\mathcal{B}'$ answers $\mathcal{A}$'s queries as follows:

> ($\mathcal{B}'$ here implicitly regards the secret key sk chosen by $\mathcal{B}'$'s experiment as $\mathsf{k} \in \{0,1\}^n$ for KwPRF.) Upon the j-th query $f^{(j)} \in \mathcal{B}_s$ from $\mathcal{A}$ (where $j \in [q]$), $\mathcal{B}'$ picks $\mathsf{y}_1^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$, and then defines the function $g_0^{(j)} : \{0,1\}^n \to \{0,1\}^{\ell_\mathsf{m}}$ that, on input $\mathsf{sk} = \mathsf{k} \in \{0,1\}^n$, outputs $\mathsf{y}^{(j)} = \mathsf{y}_1 \oplus f^{(j)}(\mathsf{k}) \in \{0,1\}^{\ell_\mathsf{m}}$. (Note that given $\mathsf{y}_1^{(j)}$, $g_0^{(j)}$ can be described by using the circuit for computing $f^{(j)}$ and additional ℓ_m XOR gates, and thus $g_0^{(j)} \in \mathcal{B}_{s+\ell_\mathsf{m}}$.) $\mathcal{B}'$ also defines the constant function $g_1^{(j)} : \{0,1\}^n \to \{0,1\}^{\ell_\mathsf{m}}$ of size $s + \ell_\mathsf{m}$ (thus $g_1^{(j)} \in \mathcal{B}_{s+\ell_\mathsf{m}}$) that always outputs the fixed value 1^{ℓ_m}. $\mathcal{B}'$ then submits a KDM-encryption query $(g_0^{(j)}, g_1^{(j)}) \in (\mathcal{B}_{s+\ell_\mathsf{m}})^2$ to $\mathcal{O}_{\mathtt{KDM}}$, and receives a ciphertext $\mathsf{ct}^{(j)}$. $\mathcal{B}'$ now sets $\mathsf{x}^{(j)} \leftarrow \mathsf{ct}^{(j)}$ and picks $\mathsf{y}_0^{(j)} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_\mathsf{m}}$, and returns $(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})$ to $\mathcal{A}$ as the answer to the j-th query.
>
> When $\mathcal{A}$ terminates with output b', $\mathcal{B}'$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}'$.

Let $\beta \in \{0,1\}$ be the challenge bit in $\mathcal{B}'$'s $\mathcal{B}_{s+\ell_\mathsf{m}}$-KDM security experiment regarding $\mathcal{E}$. It is straightforward to see that if $\beta = 0$ and thus $\mathsf{x}^{(j)} = \mathsf{ct}^{(j)}$ is an encryption of $\mathsf{y}_1^{(j)} \oplus f^{(j)}(\mathsf{k})$ for all $j \in [q]$, then $\mathcal{A}$'s view $\{(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})\}_{j\in[q]}$ is generated identically to that in Game 3, and thus $\mathcal{B}$ simulates Game 3 perfectly for $\mathcal{A}$. Since $\mathcal{B}$ outputs $\beta' = 1$ only when $b' = b$ occurs, we have $\Pr[\beta' = 1 | \beta = 0] = \Pr[\mathtt{SUC}_3]$. On the other hand, if $\beta = 1$ and thus $\mathsf{x}^{(j)} = \mathsf{ct}^{(j)}$ is an encryption of 1^{ℓ_m} (independent of y_1 and $\mathcal{A}$'s query $f^{(j)}$) for all $j \in [q]$, then $\mathcal{A}$'s view $\{(\mathsf{x}^{(j)}, \mathsf{y}_b^{(j)})\}_{j\in[q]}$ is

generated identically to that in Game 4, and thus $\mathcal{B}'$ simulates Game 4 perfectly for $\mathcal{A}$. With a similar argument to the case $\beta = 0$, we have $\Pr[\beta' = 1|\beta = 1] = \Pr[\mathtt{SUC}_4]$. Thus, $\mathcal{B}'$'s advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathcal{B}_{s+\ell_{\mathsf{m}}}\text{-}\mathtt{KDM}}_{\mathcal{E},\mathcal{B}'}(\lambda) = \Big|\Pr[\beta' = 1|\beta = 0] - \Pr[\beta' = 1|\beta = 1]\Big| = \Big|\Pr[\mathtt{SUC}_3] - \Pr[\mathtt{SUC}_4]\Big|,$$

which is exactly Eq. (16), as desired.

This completes the proof of the second item in the theorem statement.

Finally, we briefly explain how $\mathcal{P}$-KDM (resp. $\mathcal{L}$-KDM) security of the SKE scheme $\mathcal{E}$ leads to the $\mathcal{P}$-KDS (resp. $\mathcal{L}$-KDS security) of KwPRF. At this point, it should be obvious from the description of the reduction $\mathcal{B}'$ above that if $\mathcal{A}$'s query $f^{(j)}$ is a projection function (resp. an affine function), then the functions $g_0^{(j)}$ and $g_1^{(j)}$ used by $\mathcal{B}'$ as its KDM-encryption queries can also be described by projection functions (resp. affine functions). Hence, $\mathcal{B}'$ can be straightforwardly converted to a reduction attacking the $\mathcal{P}$-KDM (resp. $\mathcal{L}$-KDM) security of the SKE scheme $\mathcal{E}$. Hence, the first item in the theorem statement can also be established. □ (**Theorem 9**)

Acknowledgement This work was partially supported by JST K Program JPMJKP24U3 and JST CREST JPMJCR22M1.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Berlin, Heidelberg, August 2009.

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Cham, December 2020.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307. IEEE Computer Society Press, October 2003.

[AP22] Navid Alamati and Sikhar Patranabis. Cryptographic primitives with hinting property. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 33–62. Springer, Cham, December 2022.

[BBN+09] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, Berlin, Heidelberg, December 2009.

[BCPT13] Eleanor Birrell, Kai-Min Chung, Rafael Pass, and Sidharth Telang. Randomness-dependent message security. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 700–720. Springer, Berlin, Heidelberg, March 2013.

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 535–564. Springer, Cham, April / May 2018.

[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, Berlin, Heidelberg, August 2003.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

[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, Cham, December 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, Cham, October 2021.

[GKLW23] Rachit Garg, Dakshita Khurana, George Lu, and Brent Waters. On non-uniform security for black-box non-interactive CCA commitments. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 173–204. Springer, Cham, April 2023.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HK15] Mohammad Hajiabadi and Bruce M. Kapron. Reproducible circularly-secure bit encryption: Applications and realizations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 224–243. Springer, Berlin, Heidelberg, August 2015.

[HKV23] Mohammad Hajiabadi, Shahram Khazaei, and Behzad Vahdani. Randomness recoverable secret sharing schemes. In Kai-Min Chung, editor, *ITC 2023*, volume 267 of *LIPIcs*, pages 12:1–12:25. Schloss Dagstuhl, June 2023.

[HKW20] Susan Hohenberger, Venkata Koppula, and Brent Waters. Chosen ciphertext security from injective trapdoor functions. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 836–866. Springer, Cham, August 2020.

[HO10] Brett Hemenway and Rafail Ostrovsky. Building injective trapdoor functions from oblivious transfer. *Electron. Colloquium Comput. Complex.*, TR10-127, 2010.

[KM20] Fuyuki Kitagawa and Takahiro Matsuda. Circular security is complete for KDM security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 253–285. Springer, Cham, December 2020.

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise LPN. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, March 2014.

[KMT19] 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, Cham, August 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, Cham, August 2019.

[NR99] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

[PPY24] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Public-key anamorphism in (CCA-secure) public-key encryption and beyond. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part II*, volume 14921 of *LNCS*, pages 422–455. Springer, Cham, August 2024.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, Berlin, Heidelberg, March 2009.

[WW23] Brent Waters and Daniel Wichs. Universal amplification of KDM security: From 1-key circular to multi-key KDM. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 674–693. Springer, Cham, August 2023.

[YZ16] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 214–243. Springer, Berlin, Heidelberg, August 2016.

A Further Applications of KDS Secure Weak PRF

We show how to construct variants of hinting PRG using a KDS secure weak PRF.

A.1 Functional Hinting PRG

Here, we first give the definition of a functional hinting PRG introduced in [AP22].[11]

Let $m = m(\lambda)$, $n = n(\lambda)$, and $t = t(\lambda)$ be polynomials. Let $\mathcal{F}$ be a family of functions whose domain is $\{0,1\}^n$ and range is $\{0,1\}^t$. Let $\mathcal{FH} = (\mathsf{FHSetup}, \mathsf{FHPRG})$ be a hinting PRG with seed length n, such that $\mathsf{FHPRG}(\mathsf{pp}, \cdot) : \{0,1\}^n \to \{0,1\}^{(t+1)\cdot\lambda}$.

We say that $\mathcal{FH}$ is a *secure functional hinting PRG with respect to* $\mathcal{F}$ *($\mathcal{F}$-functional hinting*

[11] The actual security definition of a functional hinting PRG given in [AP22] is with respect to a single function f, but we naturally extend it to one with respect to a function family $\mathcal{F}$ in line with the security definition for a functional hinting weak PRF given in Section A.3.

PRG) if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{FHPRG}}_{\mathcal{FH},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} (f \in \mathcal{F}, \mathsf{st}) \xleftarrow{\mathsf{r}} \mathcal{A}_1(1^\lambda); \\ \mathsf{pp} \xleftarrow{\mathsf{r}} \mathsf{FHSetup}(1^\lambda); \\ \mathsf{s} \xleftarrow{\mathsf{r}} \{0,1\}^n; \\ (\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_t) \leftarrow \mathsf{FHPRG}(\mathsf{pp}, \mathsf{s}); \\ \alpha = (\alpha_1, \ldots, \alpha_t) \leftarrow f(\mathsf{s}); \\ \forall i \in [t] : \mathsf{y}_{i,\alpha_i} \leftarrow \mathsf{y}_i; \\ \forall i \in [t] : \mathsf{y}_{i,1-\alpha_i} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda; \\ \mathsf{Y}_1 \leftarrow (\mathsf{y}_0, (\mathsf{y}_{i,0}, \mathsf{y}_{i,1})_{i\in[t]}); \\ \mathsf{Y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^\lambda \times (\{0,1\}^\lambda \times \{0,1\}^\lambda)^t; \\ b \xleftarrow{\mathsf{r}} \{0,1\};\ b' \xleftarrow{\mathsf{r}} \mathcal{A}_2(\mathsf{pp}, \mathsf{Y}_b, \mathsf{st}) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible.

Note that if $t = n$ and $\mathcal{F}$ consists only of the identity function, then the definition is equivalent to that for a hinting PRG.

For functional hinting PRGs, we will consider the same function families as those considered for KDM security for SKE listed in Section 2.2, in particular, the family of functions that can be computed by size-s circuits $\mathcal{B}_s$.

Construction. Let $n = n(\lambda)$, $m = m(\lambda)$, and $t = t(\lambda)$ be polynomials. Let $s = s(\lambda)$ be a polynomial such that $s = \Omega(n+t)$. Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{(t+1)\cdot\lambda}$ be a $\mathcal{B}_{s+p}$-KDS secure weak PRF (where $p = p(\lambda)$ is a polynomial specified in Theorem 10). We identify the output space $\{0,1\}^{(t+1)\cdot\lambda}$ with $\{0,1\}^\lambda \times (\{0,1\}^\lambda)^t$.

Using KwPRF, we construct a $\mathcal{B}_s$-functional hinting PRG $\mathcal{FH} = (\mathsf{FHSetup}, \mathsf{FHPRG})$ as follows:

$\mathsf{FHSetup}(1^\lambda)$**:** Pick $\mathsf{pp} \xleftarrow{\mathsf{r}} \{0,1\}^m$ and return pp.

$\mathsf{FHPRG}(\mathsf{pp}, \mathsf{s} \in \{0,1\}^n)$**:** Compute $\mathsf{y} = (\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) \leftarrow \mathsf{KwPRF}(\mathsf{s}, \mathsf{pp})$ and return y.

The following theorem guarantees that $\mathcal{FH}$ constructed as above is indeed a secure $\mathcal{B}_s$-functional hinting PRG.

Theorem 10 *Let $s = s(\lambda)$ be a polynomial such that $s = \Omega(n + t)$, and let $p = p(\lambda) = O(t\lambda)$ be the size of the circuit computing the selector function that will appear in the proof. If* KwPRF *is a $\mathcal{B}_{s+p}$-KDS secure weak PRF, then $\mathcal{FH}$ is a secure $\mathcal{B}_s$-functional hinting PRG.*

Proof of Theorem 10. Since the proof is almost the same as that for Theorem 2, we only explain the difference. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an arbitrary PPT adversary that attacks the $\mathcal{B}_s$-functional hinting PRG security of $\mathcal{FH}$. Using $\mathcal{A}$, we consider the PPT reduction algorithm $\mathcal{B}'$ that initially runs $(f \in \mathcal{B}_s, \mathsf{st}) \xleftarrow{\mathsf{r}} \mathcal{A}_1(1^\lambda)$, and then works in almost the same way as $\mathcal{B}$ used in the proof of Theorem 2 (where $\mathcal{B}'$ here finally runs $\mathcal{A}_2(\mathsf{pp}, \mathsf{Y}_b, \mathsf{st})$ instead of $\mathcal{A}(\mathsf{pp}, \mathsf{Y}_b)$). The important difference from $\mathcal{B}$ is that $\mathcal{B}'$ here picks $2t$ random values $(\Delta_{i,v})_{i\in[t],v\in\{0,1\}}$, and prepares a function $g : \{0,1\}^n \to \{0,1\}^t$ that on input s, first computes $\alpha = (\alpha_1, \ldots, \alpha_t) \leftarrow f(\mathsf{s})$, and then computes the selector function $\mathsf{sel}[(\Delta_{i,v})_{i\in[t],v\in\{0,1\}}](\cdot)$ that on input α outputs $(\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_n) = (0^\lambda, \Delta_{1,\alpha_1}, \ldots, \Delta_{t,\alpha_t})$. (That is, we have $g(\mathsf{s}) = \mathsf{sel}[(\Delta_{i,v})_{i\in[t],v\in\{0,1\}}](f(\mathsf{s}))$.) Clearly, g can be computed by a circuit of size $s+p$ and thus we have $g \in \mathcal{B}_{s+p}$. The rest of the analysis remains exactly the same as that for $\mathcal{B}$ in the proof of Theorem 2. □ (**Theorem 10**)

A.2 Hinting Weak PRF

Here, we first recall the definition of a hinting weak PRF introduced in [AP22].

Let $m = m(\lambda)$ and $n = n(\lambda)$ be polynomials, and let $\mathsf{HwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{(n+1)\cdot\lambda}$ be an efficiently computable function.[12]

We say that HwPRF is a *secure hinting weak PRF*, if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathtt{HwPRF}}_{\mathsf{HwPRF},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n;\ b \xleftarrow{\mathsf{r}} \{0,1\}; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\mathtt{HwPRF}}(\$)}(1^\lambda) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\mathtt{HwPRF}}(\$)$ is the oracle that, upon a query, proceeds as follows:

1. Parse k as $(\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$.
2. Pick $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^m$, and compute $\mathsf{y} = (\mathsf{y}_0, (\mathsf{y}_{i,\mathsf{k}_i})_{i\in[n]}) \leftarrow \mathsf{HwPRF}(\mathsf{k}, \mathsf{x})$.
3. Pick $\mathsf{y}_{1,1-\mathsf{k}_1}, \ldots, \mathsf{y}_{n,1-\mathsf{k}_n} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$.
4. Set $\mathsf{Y}_1 \leftarrow (\mathsf{y}_0, (\mathsf{y}_{i,v})_{i\in[n],v\in\{0,1\}})$.
5. Pick $\mathsf{Y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^\lambda \times (\{0,1\}^\lambda \times \{0,1\}^\lambda)^n$.
6. Return $(\mathsf{x}, \mathsf{Y}_b)$.

Construction. Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{(n+1)\cdot\lambda}$ be a $\mathcal{P}$-KDS secure weak PRF, for some polynomials $n = n(\lambda)$ and $m = m(\lambda)$. We identify the output space $\{0,1\}^{(n+1)\cdot\lambda}$ with $(\{0,1\}^\lambda)^{n+1}$.

The following theorem guarantees that KwPRF is itself secure also as a hinting weak PRF.

Theorem 11 *If* KwPRF *is a* $\mathcal{P}$*-KDS secure weak PRF, then* KwPRF *is also a secure hinting weak PRF.*

Proof of Theorem 11. The proof goes very similarly to that of the proof of Theorem 2. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the hinting weak PRF security of KwPRF. Using $\mathcal{A}$, we consider the following PPT reduction algorithm $\mathcal{B}$ that attacks the $\mathcal{P}$-KDS security of KwPRF.

$\mathcal{B}^{\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}(\cdot)}(1^\lambda)$**:** $\mathcal{B}$ runs $\mathcal{A}(1^\lambda)$, and then responds to each query from $\mathcal{A}$ as follows:

> $\mathcal{B}$ picks $\Delta_{1,0}, \Delta_{1,1}, \ldots, \Delta_{n,0}, \Delta_{n,1} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$, and defines a projection function $g \in \mathcal{P}$ that computes the selector function $\widetilde{\mathsf{sel}}[(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}] : \{0,1\}^\lambda \to (\{0,1\}^\lambda)^{n+1}$ that has $2n$ λ-bit values $(\Delta_{i,v})_{i\in[n],v\in\{0,1\}}$ hardwired, and on input $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$, outputs $(0^\lambda, \Delta_{1,\mathsf{k}_1}, \ldots, \Delta_{n,\mathsf{k}_n}) \in (\{0,1\}^\lambda)^{n+1}$. (Recall that a selector function can be computed by a projection function.) Then, $\mathcal{B}$ submits a query $g \in \mathcal{P}$ to the oracle $\mathcal{O}_{\mathtt{KDS\text{-}wPRF}}$, and receives the answer $(\mathsf{x}, \mathsf{z}) \in \{0,1\}^m \times (\{0,1\}^\lambda)^{n+1}$, where $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n) \in (\{0,1\}^\lambda)^{n+1}$. Now, $\mathcal{B}$ sets $\mathsf{y}'_0 \leftarrow \mathsf{z}_0$, and $\mathsf{y}'_{i,v} \leftarrow \mathsf{z}_i \oplus \Delta_{i,v}$ for all $(i,v) \in [n] \times \{0,1\}$. Then, $\mathcal{B}$ sets $\mathsf{Y}' \leftarrow (\mathsf{y}'_0, (\mathsf{y}'_{i,0}, \mathsf{y}'_{i,1})_{i\in[n]}) \in \{0,1\}^n \times (\{0,1\}^\lambda)^{2n}$, and returns $(\mathsf{x}, \mathsf{Y}')$ to $\mathcal{A}$ as the answer to $\mathcal{A}$'s query.
>
> When $\mathcal{A}$ terminates with output b', $\mathcal{B}$ sets $\beta' \leftarrow b'$, and terminates with output β'.

[12] Similarly to the case of a hinting PRG, for simplicity, we only consider hinting weak PRFs such that the length of each output block is λ-bit.

The above completes the description of $\mathcal{B}$.

Let $\beta \in \{0,1\}$ be the challenge bit in $\mathcal{B}$'s $\mathcal{P}$-KDS security experiment regarding KwPRF, and let $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$ be the key for KwPRF chosen in $\mathcal{B}$'s experiment.

We first argue that if $\beta = 1$ and thus the value $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n) \in (\{0,1\}^\lambda)^{n+1}$ returned from $\mathcal{B}$'s oracle $\mathcal{O}_{\texttt{KDS-wPRF}}$ is computed as $\mathsf{z} = \mathsf{KwPRF}(\mathsf{k}, \mathsf{x}) \oplus g(\mathsf{k})$, then $\mathcal{B}$ simulates the hinting weak PRF security experiment with $b = 1$ perfectly for $\mathcal{A}$. Indeed, letting $\mathsf{y} = (\mathsf{y}_0, \mathsf{y}_1, \ldots, \mathsf{y}_n) = \mathsf{KwPRF}(\mathsf{k}, \mathsf{x})$, we have $\mathsf{z}_0 = \mathsf{y}_0$ and $\mathsf{z}_i = \mathsf{y}_i \oplus \Delta_{i,\mathsf{k}_i}$ for all $i \in [n]$. Hence, the components of $\mathsf{Y}' = (\mathsf{y}'_0, (\mathsf{y}'_{i,0}, \mathsf{y}'_{i,1})_{i\in[n]})$ are generated as follows:

$$\mathsf{y}'_0 = \mathsf{y}_0,$$
$$\forall (i,v) \in [n] \times \{0,1\} : \quad \mathsf{y}'_{i,v} = \mathsf{z}_i \oplus \Delta_{i,v} = \begin{cases} \mathsf{y}_i & \text{if } v = \mathsf{k}_i \\ \mathsf{y}_i \oplus \Delta_{i,0} \oplus \Delta_{i,1} & \text{if } v = 1 - \mathsf{k}_i \end{cases}.$$

Note that from $\mathcal{A}$'s viewpoint, $\mathsf{y}'_{i,1-\mathsf{k}_i}$ looks like a truly random string due to the XOR with $\Delta_{i,0} \oplus \Delta_{i,1}$ that is in turn distributed uniformly at random. Hence, Y' is distributed in exactly the same way as one generated in the hinting weak PRF security experiment with $b = 1$.

On the other hand, if $\beta = 0$ and thus $\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_n) \in (\{0,1\}^\lambda)^{n+1}$ is truly random, then $\mathcal{B}$ simulates the hinting weak PRF security experiment with $b = 0$ perfectly for $\mathcal{A}$. In particular, each $\mathsf{y}'_{i,v}$ is masked with the truly random string $\Delta_{i,v}$, which makes $\mathsf{y}'_{i,v}$ look like a truly random string for every $(i,v) \in [n] \times \{0,1\}$.

Therefore, $\mathcal{B}$ is overall simulating the hinting weak PRF security experiment perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is that of $\mathcal{B}$'s. Since $\mathcal{B}$ uses $\mathcal{A}$'s output as it is, we have $\mathsf{Adv}^{\mathsf{HwPRF}}_{\mathsf{KwPRF},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathcal{P}\text{-}\texttt{KDS-wPRF}}_{\mathsf{KwPRF},\mathcal{B}}(\lambda)$, which is negligible by our assumption that KwPRF is a $\mathcal{P}$-KDS secure weak PRF. Hence, KwPRF is a secure hinting weak PRF. □ (**Theorem 11**)

A.3 Functional Hinting Weak PRF

Here, we first recall the definition of a functional hinting weak PRF introduced in [AP22].

Let $m = m(\lambda)$, $n = n(\lambda)$, and $t = t(\lambda)$ be polynomials, and let $\mathcal{F}$ be a family of functions whose domain is $\{0,1\}^n$ and range is $\{0,1\}^t$. Let $\mathsf{FHwPRF} : \{0,1\}^n \times \{0,1\}^m \times \{0,1\}^{(t+1)}$ be an efficiently computable function.

We say that FHwPRF is a *secure functional hinting weak PRF with respect to $\mathcal{F}$ ($\mathcal{F}$-hinting weak PRF)*, if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathcal{F}\text{-}\texttt{FHwPRF}}_{\mathsf{FHwPRF},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{k} \xleftarrow{\mathsf{r}} \{0,1\}^n;\ b \xleftarrow{\mathsf{r}} \{0,1\}; \\ b' \xleftarrow{\mathsf{r}} \mathcal{A}^{\mathcal{O}_{\texttt{FHwPRF}}(\$)}(1^\lambda) \end{array} : b' = b \right] - \frac{1}{2} \right|$$

is negligible, where $\mathcal{O}_{\texttt{FHwPRF}}(\$)$ is the oracle that, upon a query, proceeds as follows:

1. Parse k as $(\mathsf{k}_1, \ldots, \mathsf{k}_n) \in \{0,1\}^n$ and compute $\alpha = (\alpha_1, \ldots, \alpha_t) \leftarrow f(\mathsf{k})$.
2. Pick $\mathsf{x} \xleftarrow{\mathsf{r}} \{0,1\}^m$, and compute $\mathsf{y} = (\mathsf{y}_0, (\mathsf{y}_{i,\alpha_i})_{i\in[n]}) \leftarrow \mathsf{FHwPRF}(\mathsf{k}, \mathsf{x})$.
3. Pick $\mathsf{y}_{1,1-\alpha_1}, \ldots, \mathsf{y}_{t,1-\alpha_t} \xleftarrow{\mathsf{r}} \{0,1\}^\lambda$.
4. Set $\mathsf{Y}_1 \leftarrow (\mathsf{y}_0, (\mathsf{y}_{i,v})_{i\in[t],v\in\{0,1\}})$.
5. Pick $\mathsf{Y}_0 \xleftarrow{\mathsf{r}} \{0,1\}^\lambda \times (\{0,1\}^\lambda \times \{0,1\}^\lambda)^t$.
6. Return $(\mathsf{x}, \mathsf{Y}_b)$.

Note that if $t = n$ and f is the identity function, then the definition is equivalent to that for a hinting weak PRF.

Construction. The analogy between the hinting PRG given in Section 3.1 and the functional hinting PRG given in Section A.1 can be established also for a hinting weak PRF and a functional hinting weak PRF. Specifically, let $t = t(\lambda)$, $n = n(\lambda)$, and $m = m(\lambda)$ be polynomials. Let $s = s(\lambda)$ be a polynomial such that $s = \Omega(n + t)$. Let $\mathsf{KwPRF} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{(t+1)\cdot\lambda}$ be a $\mathcal{B}_{s+p}$-KDS secure weak PRF (where $p = p(\lambda)$ is a polynomial specified in Theorem 12).

The following theorem guarantees that KwPRF is a $\mathcal{B}_s$-functional weak PRF. We omit its formal proof since it is a straightforward mix of the reductions used for proving the security of the $\mathcal{B}_s$-functional hinting PRG in Section A.1 and the hinting weak PRF in Section A.2.

Theorem 12 *Let $s = s(\lambda)$ be a polynomial such that $s = \Omega(n + t)$, and let $p = (\lambda) = O(t\lambda)$ be the size of the circuit computing the selector function that is defined in exactly the same way as the one used in the proof of Theorem 10. If KwPRF is a $\mathcal{B}_{s+p}$-KDS secure weak PRF, then KwPRF is also a secure $\mathcal{B}_s$-functional hinting weak PRF.*

B Proof of Theorem 5

Let $(\mathsf{pk}, \mathsf{sk})$ be a key pair output by $\mathsf{KG}(1^\lambda)$. For each $\mathsf{A}' \in \{0,1\}^{3\lambda} \cup \{\bot\}$, we define the set $\mathcal{C}_{\mathsf{A}'} \subset \{0,1\}^{\ell_{\mathsf{ct}}}$ by

$$\mathcal{C}_{\mathsf{A}'} := \left\{ \mathsf{ct} \in \{0,1\}^{\ell_{\mathsf{ct}}} \;\middle|\; \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{A}' \right\}.$$

Furthemore, for each $\mathsf{A}' \in \{0,1\}^{3\lambda}$, we define the function $f_{\mathsf{A}'} : \{0,1\}^\lambda \times \mathcal{C}_{\mathsf{A}'} \to \{0,1\}^{\ell_{\mathsf{ct}}}$ by

$$\forall (\mathsf{r}, \mathsf{ct}) \in \{0,1\}^\lambda \times \mathcal{C}_{\mathsf{A}'} : \quad f_{\mathsf{A}'}(\mathsf{r}, \mathsf{ct}) := \mathsf{Enc}(\mathsf{pk}, \mathsf{A}'; \mathsf{r}) \oplus \mathsf{ct}.$$

We refer to a pair of a public key $\mathsf{rpk} = (\mathsf{pk}, \mathsf{A}, \mathsf{B}, (\mathsf{C}_i)_{i\in[n]})$ and a secret key $\mathsf{rsk} = (\mathsf{sk}, \mathsf{A}, \mathsf{B})$ as *bad* if there exists $i \in [n]$ such that $\mathsf{C}_i \in \mathrm{Im}(f_{\mathsf{A}})$.

Now, we first show that if $(\mathsf{rpk} = (\mathsf{pk}, \mathsf{A}, \mathsf{B}, (\mathsf{C}_i)_{i\in[n]}), \mathsf{rsk} = (\mathsf{sk}, \mathsf{A}, \mathsf{B}))$ is not bad, then $(\mathsf{rpk}, \mathsf{rsk})$ is not errorneous, namely, $\mathsf{RDec}(\mathsf{rsk}, \mathsf{REnc}(\mathsf{rpk}, \mathsf{m}; \mathsf{x})) = \mathsf{m}$ holds for all $\mathsf{m} \in \{0,1\}^{\ell_{\mathsf{m}}}$ and $\mathsf{x} \in \{0,1\}^n$. To this end, letting $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n) \in \{0,1\}^n$ and $\mathsf{rct} = (\mathsf{z}, (\mathsf{ct}_i)_{i\in[n]}) = \mathsf{REnc}(\mathsf{rpk}, \mathsf{m}; \mathsf{x})$, it is sufficient to show that the following two conditions hold (since RDec correctly recovers x bit-by-bit, from which RDec can recover m):

(a) For all $i \in [n]$ for which $\mathsf{x}_i = 0$ holds, we have $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_i) = \mathsf{A}$ (and thus $\mathsf{ct}_i \in \mathcal{C}_{\mathsf{A}}$).

(b) For all $i \in [n]$ for which $\mathsf{x}_i = 1$ holds, we have $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_i) \neq \mathsf{A}$ (and thus $\mathsf{ct}_i \notin \mathcal{C}_{\mathsf{A}}$).

Note that the condition (a) trivially holds due to the correctness of Π, since by the design of REnc, $\mathsf{ct}_i = \mathsf{ct}_{i,0}$ is just an encryption of A for all $i \in [n]$ with $\mathsf{x}_i = 0$.

To see that the condition (b) also holds, assume towards a contradiction (with the condition that $(\mathsf{rpk}, \mathsf{rsk})$ is not bad) that there exists $i \in [n]$ for which $\mathsf{x}_i = 1$ and $\mathsf{ct}_i \in \mathcal{C}_{\mathsf{A}}$ hold simultaneously. By the design of REnc, we have $\mathsf{ct}_i = \mathsf{ct}_{i,1} \oplus \mathsf{C}_i$, and $\mathsf{ct}_{i,1}$ is an encryption of A. Thus, there exists some $\mathsf{r} \in \{0,1\}^\lambda$ that satisfies the following:

$$\mathsf{ct}_i = \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}) \oplus \mathsf{C}_i \quad \Longleftrightarrow \quad \mathsf{C}_i = \mathsf{Enc}(\mathsf{pk}, \mathsf{A}; \mathsf{r}) \oplus \mathsf{ct}_i = f_{\mathsf{A}}(\mathsf{r}, \mathsf{ct}_i), \tag{18}$$

which implies $\mathsf{C}_i \in \mathrm{Im}(f_{\mathsf{A}})$. However, this is a contradiction with the condition that $(\mathsf{rpk}, \mathsf{rsk})$ is not bad and thus $\mathsf{C}_i \notin \mathrm{Im}(f_{\mathsf{A}})$ must hold. Hence, the condition (b) holds as well.

We have seen so far that if the pair $(\mathsf{rpk}, \mathsf{rsk})$ is not bad, then it is not errorneous. Next, we show that when $(\mathsf{rpk}, \mathsf{rsk}) \xleftarrow{\mathsf{r}} \mathsf{RKG}(1^\lambda)$ is executed, the probability that $(\mathsf{rpk}, \mathsf{rsk})$ becomes bad is bounded by $(n + 1) \cdot 2^{-\lambda}$, which will complete the proof of the theorem.

For a preparation, we define the set $\mathtt{FAT} \subseteq \{0,1\}^{3\lambda}$ by

$$\mathtt{FAT} := \left\{ \mathsf{A}' \in \{0,1\}^{3\lambda} \;\middle|\; |\mathcal{C}_{\mathsf{A}'}| > 2^{\ell_{\mathsf{ct}}-2\lambda} \right\}.$$

Then, the probability that a randomly chosen $\mathsf{A} \in \{0,1\}^{3\lambda}$ belongs to $\mathtt{FAT}$ is at most $2^{-\lambda}$, as shown below:

$$\begin{aligned}
\Pr_{\mathsf{A} \xleftarrow{r} \{0,1\}^{3\lambda}}[\mathsf{A} \in \mathtt{FAT}] &= \Pr_{\mathsf{A} \xleftarrow{r} \{0,1\}^{3\lambda}}\left[|\mathcal{C}_{\mathsf{A}}| > 2^{\ell_{\mathsf{ct}}-2\lambda} \right] \\
&\leq (2^{\ell_{\mathsf{ct}}-2\lambda})^{-1} \cdot \mathop{\mathbf{E}}_{\mathsf{A} \xleftarrow{r} \{0,1\}^{3\lambda}}\left[|\mathcal{C}_{\mathsf{A}}| \right] \qquad (19) \\
&= 2^{-\ell_{\mathsf{ct}}+2\lambda} \cdot 2^{-3\lambda} \cdot \sum_{\mathsf{A}' \in \{0,1\}^{3\lambda}} |\mathcal{C}_{\mathsf{A}'}| \\
&\leq 2^{-\lambda}, \qquad (20)
\end{aligned}$$

where Eq. (19) follows from the Markov inequality; Eq. (20) is obtained by the fact that $\sum_{\mathsf{A}' \in \{0,1\}^{3\lambda}} |\mathcal{C}_{\mathsf{A}'}| = 2^{\ell_{\mathsf{ct}}} - |\mathcal{C}_{\perp}| \leq 2^{\ell_{\mathsf{ct}}}$ holds, which is in turn due to the fact that $\{\mathcal{C}_{\mathsf{A}'}\}_{\mathsf{A}' \in \{0,1\}^{3\lambda} \cup \{\perp\}}$ is a partition of $\{0,1\}^{\ell_{\mathsf{ct}}}$ by the definition of $\mathcal{C}_{\mathsf{A}'}$.

Furthermore, note that the size of $\mathrm{Im}(f_{\mathsf{A}'})$ is at most $2^{\lambda} \cdot |\mathcal{C}_{\mathsf{A}'}|$ for all $\mathsf{A}' \in \{0,1\}^{3\lambda}$. Hence, for all $\mathsf{A}' \in \{0,1\}^{3\lambda} \setminus \mathtt{FAT}$, we have

$$\Pr_{\mathsf{C} \xleftarrow{r} \{0,1\}^{\ell_{\mathsf{ct}}}}[\mathsf{C} \in \mathrm{Im}(f_{\mathsf{A}'})] \leq 2^{\lambda} \cdot |\mathcal{C}_{\mathsf{A}'}| \cdot 2^{-\ell_{\mathsf{ct}}} \overset{(*)}{\leq} 2^{\lambda} \cdot (2^{\ell_{\mathsf{ct}}-2\lambda}) \cdot 2^{-\ell_{\mathsf{ct}}} = 2^{-\lambda}, \qquad (21)$$

where the inequality (*) uses the fact that we are considering A' that does not belong to $\mathtt{FAT}$, for which $|\mathcal{C}_{\mathsf{A}'}| \leq 2^{\ell_{\mathsf{ct}}-2\lambda}$ holds.

Combining the arguments so far, we can estimate an upper bound of the probability of $(\mathsf{rpk}, \mathsf{rsk})$ being bad when $(\mathsf{rpk}, \mathsf{rsk}) \xleftarrow{r} \mathsf{RKG}(1^{\lambda})$ is executed, as follows:

$$\begin{aligned}
\Pr[(\mathsf{rpk}, \mathsf{rsk}) \text{ is bad}] &= \Pr[\exists i \in [n] : \mathsf{C}_i \in \mathrm{Im}(f_{\mathsf{A}})] \\
&\leq \Pr[\mathsf{A} \in \mathtt{FAT}] + \Pr[\exists i \in [n] : \mathsf{C}_i \in \mathrm{Im}(f_{\mathsf{A}}) | \mathsf{A} \notin \mathtt{FAT}] \\
&\leq \Pr[\mathsf{A} \in \mathtt{FAT}] + \sum_{i \in [n]} \Pr[\mathsf{C}_i \in \mathrm{Im}(f_{\mathsf{A}}) | \mathsf{A} \notin \mathtt{FAT}] \\
&\overset{(\dagger)}{\leq} 2^{-\lambda} + n \cdot 2^{-\lambda} \\
&= (n+1) \cdot 2^{-\lambda},
\end{aligned}$$

where we have used Eqs. (20) and (21) for deriving the inequality (†).

In summary, when $(\mathsf{rpk}, \mathsf{rsk}) \xleftarrow{r} \mathsf{RKG}(1^{\lambda})$ is executed, except with probability at most $(n+1) \cdot 2^{-\lambda}$, the public/secret key pair $(\mathsf{rpk}, \mathsf{rsk})$ is not errorneous, and thus $\Pi_{\mathtt{RDM}}$ satisfies $(n+1) \cdot 2^{-\lambda}$-almost-all-keys correctness. □ (**Theorem 5**)