Minicrypt Primitives with Algebraic Structure and Applications

Navid Alamati* Hart Montgomery † Sikhar Patranabis‡ Arnab Roy§

February 5, 2019

Abstract

Algebraic structure lies at the heart of much of Cryptomania as we know it. An interesting question is the following: instead of building (Cryptomania) primitives from concrete assumptions, can we build them from *simple* Minicrypt primitives endowed with additional *algebraic* structure? In this work, we affirmatively answer this question by adding algebraic structure to the following Minicrypt primitives:

- One-Way Function (OWF)
- Weak Unpredictable Function (wUF)
- Weak Pseudorandom Function (wPRF)

The algebraic structure that we consider is group homomorphism over the input/output spaces of these primitives. We also consider a "bounded" notion of homomorphism where the primitive only supports an a priori bounded number of homomorphic operations in order to capture lattice-based and other "noisy" assumptions. We show that these structured primitives can be used to construct many cryptographic protocols. In particular, we prove that:

- (Bounded) *Homomorphic OWFs* (HOWFs) imply collision-resistant hash functions, Schnorr-style signatures, and chameleon hash functions.
- (Bounded) *Input-Homomorphic weak UFs* (IHwUFs) imply CPA-secure PKE, non-interactive key exchange, trapdoor functions, blind batch encryption (which implies anonymous IBE, KDM-secure and leakage-resilient PKE), CCA2 deterministic PKE, and hinting PRGs (which in turn imply transformation of CPA to CCA security for ABE/1-sided PE).
- (Bounded) *Input-Homomorphic weak PRFs* (IHwPRFs) imply PIR, lossy trapdoor functions, OT and MPC (in the plain model).

In addition, we show how to realize any CDH/DDH-based protocol with certain properties in a generic manner using IHwUFs/IHwPRFs, and how to instantiate such a protocol from many concrete assumptions. We also consider primitives with substantially richer structure, namely *Ring IHwPRFs* and *L-composable IHwPRFs*. In particular, we show the following:

- Ring IHwPRFs with certain properties imply FHE.
- 2-composable IHwPRFs imply (black-box) IBE, and L-composable IHwPRFs imply non-interactive $(L+1)$-party key exchange.

Our framework allows us to categorize many cryptographic protocols based on which structured Minicrypt primitive implies them. In addition, it potentially makes showing the *existence* of many cryptosystems from novel assumptions substantially easier in the future.

*University of Michigan. Most of the work was done while the author was an intern at Fujitsu Laboratories of America. Email: alamati@umich.edu

†Fujitsu Laboratories of America. Email: hmontgomery@us.fujitsu.com

‡Indian Institute of Technology, KGP. The work was done while the author was an intern at Fujitsu Laboratories of America. Email: sikharpatranabis@gmail.com

§Fujitsu Laboratories of America. Email: aroy@us.fujitsu.com

Contents

1 Introduction

An important question in the theory of cryptography is also one of the simplest to state: what implies public-key cryptography? Ever since the (public) invention of public-key encryption [DH76, RSA78], people have debated this important question.

The history of symmetric-key cryptography goes back millenia–the Caesar cipher is a classic example of old cryptography–and it has continued to evolve through the centuries in different ways. There is a long list of ciphers, notably including the Viginère cipher, the Enigma machine, and even modern ciphers like AES, that can be thought of as the output of an enormous amount of human effort to build secure symmetric-key encryption.

On the other hand, public-key cryptography is a very recent development compared to symmetric-key cryptography. Many people thought that public-key cryptography was impossible before the seminal work by Diffie and Hellman [DH76]. Although we can build symmetric-key ciphers from many different assumptions, including some very simple ones, the known methods for realizing public-key cryptography require at least some kind of mathematical structure. This has led many to conjecture that public-key cryptography does, in fact, require some mathematical structure.

Barak ruminated on this question in his recent work "The Complexity of Public Key Cryptography" [Bar17]. As he puts it, "... it seems that you can't throw a rock without hitting a one-way function" but public-key cryptography is somehow "special". Barak implicitly argues that there is some mathematical structure inherent in public-key cryptography: "One way to phrase the question we are asking is to understand what type of structure is needed for public-key cryptography."

But many cryptosystems that interest people today are substantially more complicated than basic public-key encryption (PKE). In recent years, primitives like identity-based encryption [BF01, Coc01], fully homomorphic encryption [Gen09], and functional encryption [BSW11] have captivated cryptographers. It is natural to ask: is there any sort of mathematical structure that is inherent to these primitives as well? While there has been a substantial amount of work relating relatively similar primitives, to our knowledge no one has attempted to comprehensively examine the relationship between a broader collection of these higher-level primitives.

In a celebrated work, Impagliazzo [Imp95] proposed "five worlds" of relative complexity, which range from *Algorithmica*–where "efficient" algorithms for all (worst-case) problems in NP exist and cryptography is essentially nonexistent–to *Cryptomania*, a world in which public-key cryptography exists. Only two of these worlds allow for cryptography: *Minicrypt*, where symmetric cryptographic primitives exist but public-key cryptography does not, and the aforementioned *Cryptomania*.

It turns out that Minicrypt is a fairly simple world. A number of famous works have shown how to build the most commonly studied and used Minicrypt primitives from one-way functions in a generic manner. For instance, one-way functions imply pseudorandom generators [BM82, HILL99], which in turn can be used to build pseudorandom functions [GGM84]. From these primitives, it has long been known how to generically build symmetric-key encryption schemes and digital signature schemes [Rom90].

On the other hand, Cryptomania is a significantly more complicated class. It contains primitives that are very different, and it seems difficult to relate them in a generic manner. We cannot expect to, say, build FHE from PKE in a black-box manner, and there are many black-box separation results for cryptosystems in Cryptomania (we discuss this more in our related work section). In fact, recently it has even become popular to separate Cryptomania into two worlds: a world where indistinguishability obfuscation (iO) [BGI+01, GGH+13b] doesn't exist, and a world called Obfustopia [GPSZ17] where it does.

This, of course, raises a fundamental question in the complexity of public-key cryptography: can we construct classes of primitives within Cryptomania (i.e. "continents" of Cryptomania) that are tightly tied to each other through generic constructions? Ideally, we would want these "continents" to have strong relationships with a particular primitive (similar to the relationship between one-way functions and Minicrypt) where all of the cryptographic algorithms in the class could be built from the given primitive in a generic manner, and the given primitive would be conceptually the simplest function in the class.

The fact that most of the concrete assumptions that imply PKE (and also many other cryptographic

primitives) have some algebraic structure seems to imply that perhaps we can classify cryptosystems by the algebraic structure necessary for them to function. This leads us to the following question:

> Is it possible to construct Cryptomania primitives from simple Minicrypt primitives that are additionally equipped with some algebraic structure?

1.1 Our Contributions

In this work, we provide a constructive answer to the question of building PKE (and other primitives in Cryptomania) from Minicrypt primitives with algebraic structure. Let's start by considering the following Minicrypt primitives:

1. One-way Functions
2. Weak Unpredictable Functions
3. Weak Pseudorandom Functions

To add *algebraic structure* to the mentioned primitives, we assume that they are *(Input-)Homomorphic*: the input and output spaces of the primitive are groups, and the primitive is (bounded) homomorphic with respect to an efficiently computable group homomorphism. We use the following primitives and abbreviations throughout the paper:[1]

- *Homomorphic One-way Functions* (HOWFs)[2]
- *Input-Homomorphic Weak Unpredictable Functions* (IHwUFs)
- *Input-Homomorphic Weak Pseudorandom Functions* (IHwPRFs)[3]

In the body of the paper we also consider "bounded" homomorphisms, where the number of allowed homomorphisms is bounded by some function $\gamma = \gamma(\lambda)$ where λ is the security parameter, which lets us work with lattice-based and other "noisy" cryptographic assumptions.

At this point we can informally state our main contribution: we present a framework for building cryptographic primitives from HOWFs/IHwUFs/IHwPRFs (see Figure 1). This framework lets us categorize cryptographic primitives by the type of structured Minicrypt primitive that implies them. However, we need to be able to instantiate the above *general* primitives from *concrete* assumptions to have a useful framework. It turns out that we can instantiate our primitives (in most cases) from a wide variety of assumptions, typically including the assumptions that would be expected for such applications.

Instantiations from Concrete Assumptions. We show that "mainstream" cryptographic assumptions such as DDH and LWE naturally imply (bounded) HOWFs/IHwUFs/IHwPRFs. We also show that a (bounded) group-homomorphic PKE implies a (bounded) IHwPRF. This allows instantiating these primitives from any concrete assumption that implies a (bounded) homomorphic PKE (e.g. QR and DCR). Unfortunately, there is a caveat to this: the transformation from homomorphic PKE to IHwPRF comes with a disadvantage that the input space may *depend* on the key.[4] The reader may refer to Figure 2 for an overview of instantiations from concrete assumptions.[5]

[1] We define these primitives precisely in Section 2.

[2] When the function does not have a key (i.e. a one-way function) we will drop the "I" and refer to the function as simply homomorphic.

[3] In case of IHwUFs/IHwPRFs we do not assume any homomorphism on the key space.

[4] This property is necessary to realize certain cryptographic primitives from IHwUFs or IHwPRFs. We refer to Appendix 3.4 for a discussion on this property and the details of the instantiations.

[5] Notice that search to decision reductions are mostly for Gaussian-like distributions, and there are certain distributions for which search to decision reduction is not available.

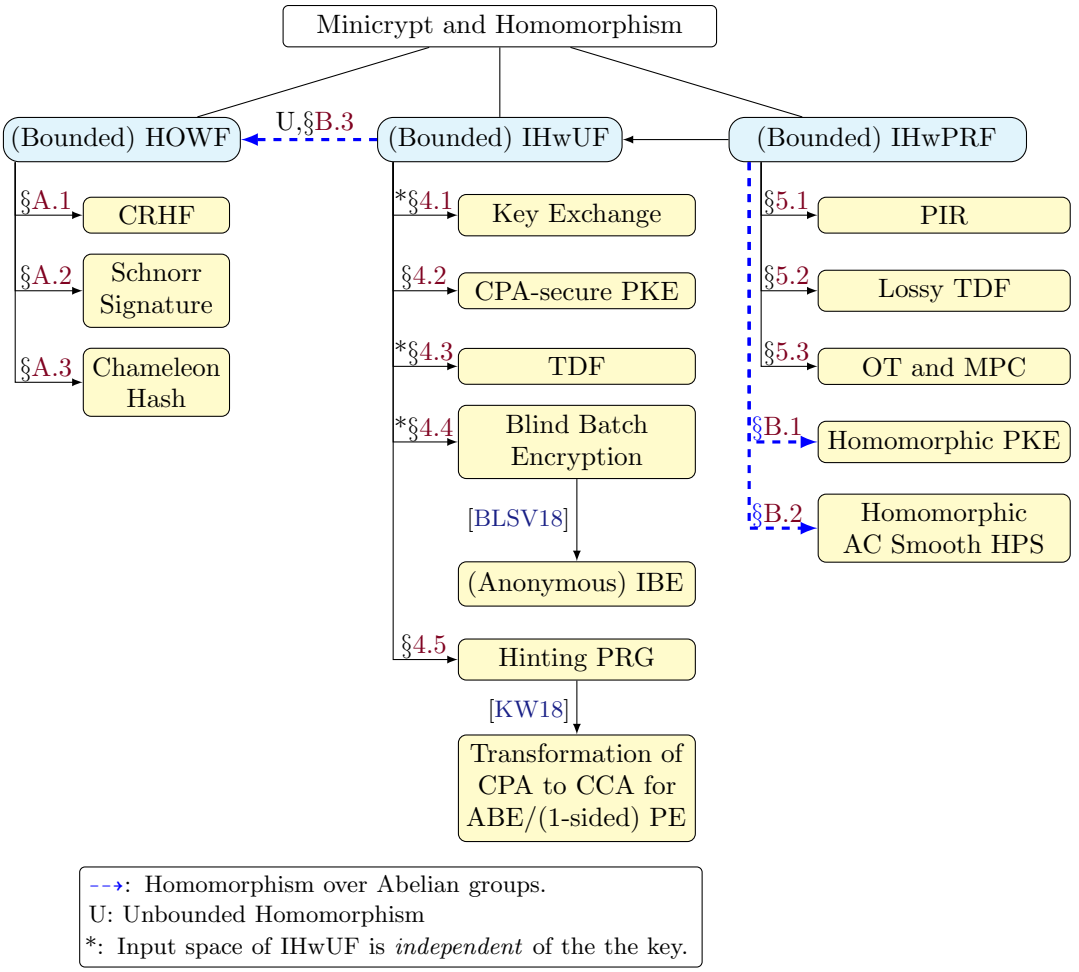

Figure 1: Cryptographic primitives from Minicrypt and Homomorphism.

Building Cryptosystems from New Assumptions. One of the benefits of our work is the implications for new assumptions. Rather than manually building lots of different cryptosystems from a new assumption, researchers only need to build one (or more) of our simple structured primitives, and the existence of a whole host of cryptosystems immediately follows.

To illustrate how this might be useful, let's look at the history of lattice-based cryptography: Ajtai and Dwork [AD97] gave a lattice-based PKE (following Ajtai's worst-case to average-case reductions for lattice problems [Ajt96]), but lattice cryptography may began in earnest with Regev's LWE paper [Reg05] in 2005. This work, in addition to introducing the LWE problem, showed how to build a basic PKE scheme from LWE as well. However, it took a while for the cryptographic community to "catch up" to other group-based cryptosystems: for instance, the first private information retrieval scheme from lattices was presented in [AMG07], and the first identity-based encryption was given in [GPV08].

These works used sophisticated techniques on lattices in order to extend the range of lattice-based cryptosystems. With our work, the existence of all of these types of cryptosystems based on the LWE assumption follows immediately from the simple observation that LWE implies a (bounded) IHwPRF. While the necessary tools for many of our constructions were not around in 2008 (particularly [DG17b] and the line of work following it), we do hope that this paper is useful for public-key cryptography assumptions in the future in terms of *feasibility* results. Ideally, it will be easy to show the existence of many types of cryptosystems for new assumptions using the tools from this paper.

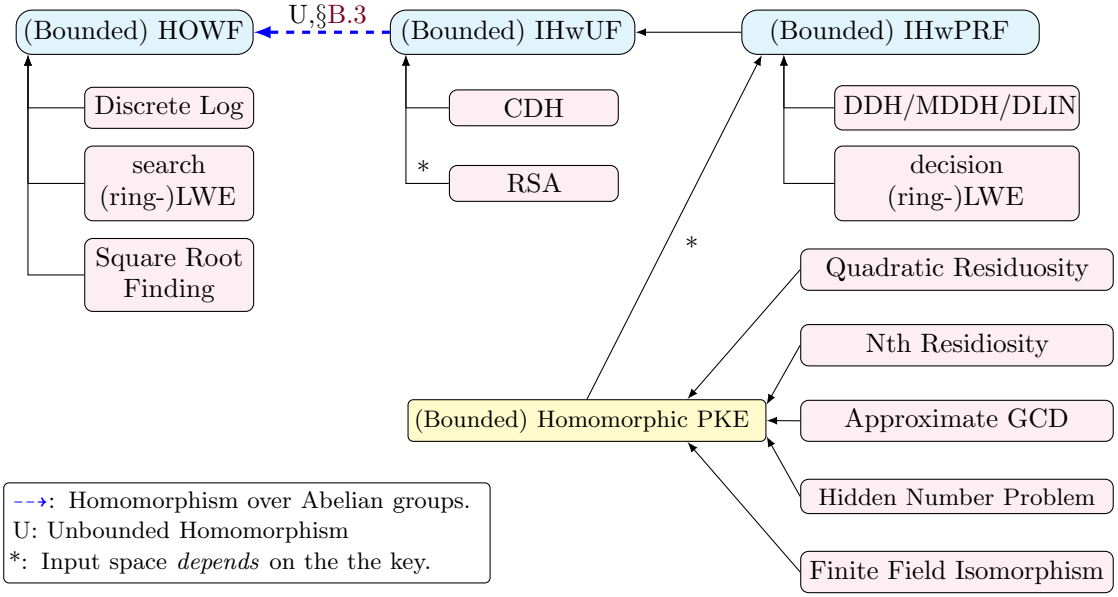

Figure 2: Instantiations from Concrete Assumptions

More Primitives from Richer Structures. Although the main focus of this work is to construct many cryptographic primitives from IHwUFs/IHwPRFs, one might ask: what if we consider richer structures? For instance, what would happen if we have a *ring homomorphism* for an IHwPRF instead of just a group homomorphism? To partially answer this question, we consider two additional structures over wPRFs:

- *Ring Homomorphism*: We consider Ring IHwPRFs (RIHwPRFs) where the input and output spaces are rings, and the homomorphism is with respect to ring operations (instead of just group operations).
- *L-composability*: We consider L-composable IHwPRFs, where L levels of IHwPRF operations compose with each other under certain conditions.

We summarize our results for these richly structured primitives in Figure 3. We remark that "*" means the order of the output ring of RIHwPRF is polynomial in the security parameter. In Appendices C and D, we provide a thorough treatment of these two primitives.

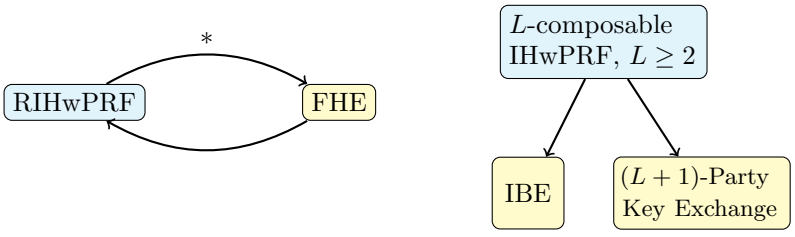

Figure 3: Cryptographic primitives from richer structures.

While the structure of 2-composability appears similar to that of bilinear pairing groups, we partially explore a possible separation between the two. We show in Appendix C that 2-composability suffices to achieve

three-party non-interactive key exchange and simple black-box constructions of IBE. Subsequently, we also present a discussion on why this primitive does not naturally yield other cryptographic protocols implied by bilinear pairings, e.g., NIZK and unique signatures. This leaves open the interesting question of whether there exists some concrete assumption that implies 2-composability but not bilinear pairings. The separation seemingly extends to the general L-composability setting, in the sense that the structure of L-composability appears to be weaker than that of a full-fledged multilinear map [GGH13a].

On the Categorization of Primitives. This work enables us to categorize different primitives based upon which structured Minicrypt primitive implies them. But it is also possible to ask whether a given cryptosystem may be constructed from some other structured Minicrypt primitive. For instance, is it possible to construct PKE from a HOWF? A positive answer would imply that one can base PKE on the discrete log problem, a long-standing (and potentially possible) goal in cryptography. We can build PKE from IHwUFs, but can we hope to do better? Our work gives rise to interesting questions like this for future work, and we discuss this more later in the paper.

It is easy to see that none of the three primitives HOWF/IHwUF/IHwPRF can be built from PKE in a *black-box* manner [HHRS07], as all of them imply collision-resistant hash functions. In addition to input homomorphisms, one may consider other structures on Minicrypt primitives.

One of the simplest structures is what we term *dual-computable*. This notion is certainly folklore, and some earlier works on PKE and key exchange implicitly constructed this primitive. A *dual-computable* primitive is a tuple of keyed functions (F_1, G_1, F_2, G_2) such that $G_1(k_1, F_2(k_2, x)) = G_2(k_2, F_1(k_1, x))$ where x represents the input and k_i represent keys. The reader may notice that this primitive is almost an abstraction of key exchange if the functions are unpredictable. It is not clear what kind of (minimal) structure over OWFs would imply dual-computable functions.

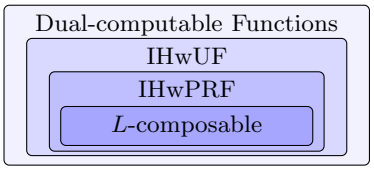

Figure 4: Implication Landscape

1.2 Related Works

Realizing public-key cryptography via some form of structure and hardness has been studied seemingly since its invention. However, several recent works have discussed this relationship in more detail. For instance, [BDV17] examined the relationship of structure and hardness through obfuscation lens, while a recent work by Berman *et al.* showed that laconic zero-knowledge protocols imply PKE [BDRV18]. Pietrzak and Sjdin [PS08] showed that a certain input property of weak PRFs implies PKE. A recent survey [BR17] briefly discusses structure and PKE through the lens of (strengthened) PRFs.

A number of works have shown how to build certain cryptosystems from cryptographic primitives with algebraic structure. These include commitment schemes, CRHF, IND-CCA secure PKE, PIR, and key-dependent message (KDM) secure PKE [IKO05, HO12, KO97, HKS16]. Of particular relevance to us is the work of Hajiabadi *et al.* [HKS16] on using homomorphic weak PRFs to build KDM secure PKE.[1]

There are other related black-box constructions (or implications in a non-black-box way) between cryptographic primitives, some of which we utilize in our work. For instance, Ishai *et al.* showed how to construct secure computation protocols from enhanced trapdoor functions (or homomorphic PKE) [IKLP06]. Rothblum [Rot11] showed a transformation of a secret-key encryption (SKE) scheme with some special form of weak homomorphism into a PKE that has similar properties. Black-box constructions have been shown for resettable zero-knowledge arguments [OSV15] and cryptographic accumulators [DHS15]. Many cryptographic primitives have been realized in a black-box manner from lossy trapdoor functions [PW08, BHY09, GPR16]. Very recently, Friolo *et al.* [FMV18] showed how to build secure multi-party computation from what they call strongly uniform key agreement and Fischlin and Harasser [FH18] showed the equivalence of invisible sanitizable signatures and PKE.

[1] As mentioned earlier, we refer to this primitive as Input-Homomorphic weak PRF (IHwPRF) to emphasize that the homomorphism is on the input space and not on the key space.

Understanding the complexity of various public-key primitives also requires knowledge of black-box separations, which have been extensively studied in the literature. This (non-exhaustively) includes studies separating IBE from CRHFs (and thus FHE) [MM16], separating indistinguishability obfuscation (iO) from certain primitives (for instance, CRHFs) [AS15, MMN+16], separating succinct non-interactive arguments from falsifiable assumptions [GW11], and showing that garbling of circuits having one-way function gates are not sufficient to realize PKE [GHMM18]. These separations (and related works) allow us to clearly see that some primitives are *not* equivalent, at least modulo certain assumptions. We refer the reader to [RTV04, Fis12, BBF13] for a survey on black-box reductions and separations.

1.3 Paper Outline

The rest of the paper is organized as follows: in Section 1.4, we explain the intuition of our constructions at a high level. In addition, we show how to construct a PKE from an IHwUF/IHwPRF and how to instantiate this construction with the DDH assumption. We also illustrate the power of our framework by showing constructions of recently proposed primitives such as batch encryption [BLSV18] from an IHwUF. In Section 1.5, we conclude by offering some more commentary on our work and suggest some promising directions for future work. The preliminaries and detailed constructions (and proofs) are organized as follows:

- Section 2 presents preliminary background material.
- Appendix A formalizes the definition of (bounded) HOWFs, describes how to construct many primitives from them, and discusses instantiations from concrete assumptions.
- Section 3 formally defines (bounded) IHwUFs/IHwPRFs, describes a general protocol to build primitives from them, and discusses instantiations from concrete assumptions.
- Section 4/5 shows constructions of different primitives from (bounded) IHwUFs/IHwPRFs, and describes how to instantiate them from the general protocol.
- Appendix B shows more applications from IHwUFs/IHwPRFs over abelian groups.
- Appendices C and D formally define composable IHwPRFs and Ring IHwPRFs, respectively, and show applications of these primitives. We also included a discussion on a potential separation of composable IHwPRFs from algebraic maps in Appendix C.

1.4 Technical Overview

In this section, we aim to explain some of the intuition behind our results. We will start by focusing on one particular primitive–the input homomorphic weak PRF–and some of its applications. The results for other primitives are not exactly the same, but the general structure of how we build cryptosystems from these other primitives is relatively similar. We will discuss these other primitives later in this section.

1.4.1 PKE from IHwUFs/IHwPRFs

Let's start by considering the notion of a general input-homomorphic weak PRF, or, as we have been abbreviating, an IHwPRF, which we will define as a function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. Recall that, informally speaking, a weak PRF is a function that is indistinguishable from a random function *with respect to uniformly sampled inputs*. This "weakness" as compared to a regular PRF will be critical.

We will also endow our weak PRF F with a homomorphism over the input. Suppose our input space $\mathcal{X}$ and our output space $\mathcal{Y}$ are groups with group operations $\oplus$ and $\otimes$, respectively. Roughly speaking, an IHwPRF is just a regular weak PRF with the following property:

$$F\left(k, x_1 \oplus x_2\right) = F\left(k, x_1\right) \otimes F\left(k, x_2\right).$$

We also consider what we call γ-bounded IHwPRFs. These IHwPRFs have a homomorphism that can only be computed a maximum of γ times, where γ is a pre-determined parameter. This concept lets us consider noisy assumptions like LWE, which are only approximately homomorphic. The notion is very similar to definitions of the *almost* key-homomorphic PRFs of [BLMR13]. γ-bounded IHwPRFs work for almost all of the applications that we consider in almost the same way that full IHwPRFs do. For the rest of this technical overview, though, we will assume we have a "full" IHwPRF. Also, we occasionally refer to an Input-Homomorphic weak Unpredictable Function (IHwUF), which has the same properties as IHwPRF except for the fact that its output on a uniformly random input is just *unpredictable* and not necessarily *pseudorandom*.

DDH-based Instantiation of IHwPRF. In general, it is simple to build IHwPRFs from assumptions that are widely used in cryptography. We defer most of these to Section 3.4, but we show how to build an IHwPRF from the DDH assumption here. Let $\mathbb{G}$ be a group of prime order q where the DDH problem is hard. For a uniformly sampled key $k \leftarrow \mathbb{Z}_q$ and an input $x \in \mathbb{G}$, consider the following function:

$$F(k, x) = x^k.$$

If we are only allowed to see the evaluation of F on random inputs x_i (as the weak PRF definition requires), then it is easy to see that F is a weak PRF based on the DDH assumption. Moreover, the homomorphism property is also satisfied:

$$x_1^k \cdot x_2^k = (x_1 \cdot x_2)^k.$$

Thus F is an IHwPRF. Building a *bounded* IHwPRF from LWE is similarly straightforward, but we defer this to later in the paper.

On the Input Space. It is useful to note that the "discrete logarithm problem" on the input space of an IHwPRF must be hard by its weak pseudorandomness property. Concretely, given two evaluations $(x_1, F(k, x_1))$ and $(x_2, F(k, x_2))$, an adversary can compute some value c such that $x_1^c = x_2$, then they can check if

$$F(k, x_1)^c = F(k, x_2)$$

and use this to break the (weak) pseudorandomness of F. In the context of (bounded) IHwPRFs over arbitrary groups, we note that there must exist an equivalent "discrete log" problem that allows us to capture the aforementioned property.[1] This property is crucial to the security of nearly all constructions presented in this paper.

PKE Construction. We now illustrate how to construct a CPA-secure PKE given an IHwPRF. To provide more intuition, we will present an instantiation of the encryption scheme using the DDH assumption in parallel. The construction from IHwPRF is highlighted for clarity.

Setup:

- **IHwPRF Construction:** Select an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ over groups $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ with key space $\mathcal{K}$, input space $\mathcal{X}$, and output space $\mathcal{Y}$ and some integer $n > 3\log(|\mathcal{X}|)$. Select a set X of $2n$ uniform "base elements" from $\mathcal{X}$ as

  $$\mathrm{X} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}.$$

  Select a random key $k \leftarrow \mathcal{K}$. Create a tuple Y of $2n$ elements from $\mathcal{Y}$ as

  $$\mathrm{Y} = \{y_{j,b}\}_{j \in [n], b \in \{0,1\}}$$

[1] For our LWE-based bounded IHwPRF, the "discrete log" problem equivalent is the ISIS problem.

such that $y_{j,b} = F(k, x_{j,b})$. Output the secret key and public key as:[1]

$$\mathsf{sk} = k, \quad \mathsf{pk} = (\mathrm{X}, \mathrm{Y}).$$

- **DDH Instantiation:** Let $F : \mathbb{Z}_q \times \mathbb{G} \to \mathbb{G}$ be the function defined as $F(k \in \mathbb{Z}_q, g \in \mathbb{G}) = g^k$. Select a set G of $2n$ randomly sampled elements from $\mathbb{G}$ as

$$\mathrm{G} = \{g_{j,b} \leftarrow \mathbb{G}\}_{j \in [n], b \in \{0,1\}}.$$

Select a random key $k \leftarrow \mathbb{Z}_q$. Create a tuple H of $2n$ elements from $\mathbb{G}$ as

$$\mathrm{H} = \{h_{j,b}\}_{j \in [n], b \in \{0,1\}}$$

such that $h_{j,b} = g_{j,b}^k$. Output the secret key and the public key as

$$\mathsf{sk} = k, \quad \mathsf{pk} = (\mathrm{G}, \mathrm{H}).$$

Encrypt:

- **IHwPRF Construction:** On input a message $\mathsf{m} \in \mathcal{Y}$, sample a vector $\mathbf{s} = (s_1, \ldots, s_n) \leftarrow \{0,1\}^n$. Set

$$x^* = \bigoplus_{j \in [n]} x_{j,s_j}, \quad y^* = \bigotimes_{j \in [n]} y_{j,s_j}.$$

Output the ciphertext $\mathsf{ct} = (x^*, y^* \otimes \mathsf{m})$.

- **DDH Instantiation:** On input a message $\mathsf{m} \in \mathbb{G}$, sample a vector $\mathbf{s} = (s_1, \ldots, s_n) \leftarrow \{0,1\}^n$. Set

$$g^* = \prod_{j=1}^{n} g_{j,s_j}, \quad h^* = \prod_{j=1}^{n} h_{j,s_j}.$$

Output the ciphertext $\mathsf{ct} = (g^*, h^* \cdot \mathsf{m})$.

By the leftover hash lemma, our "subset sum" process gives us outputs that are statistically close to uniform for arbitrary groups. This may be viewed as a generalization of the "exponentiation" operation to arbitrary groups.

Decrypt:

- **IHwPRF Construction:** On input a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2) \in \mathcal{X} \times \mathcal{Y}$, output

$$\mathsf{m}' = [F(k, \mathsf{ct}_1)]^{-1} \otimes \mathsf{ct}_2.$$

If $(\mathsf{ct}_1, \mathsf{ct}_2) = (x^*, y^* \otimes \mathsf{m})$, we have

$$\mathsf{m}' = [F(k, \mathsf{ct}_1)]^{-1} \otimes \mathsf{ct}_2 = (y^*)^{-1} \otimes (y^* \otimes \mathsf{m}) = \mathsf{m}.$$

[1] We implicitly assume that the description of IHwPRF is publicly available. This is similar to the assumption that in a DDH-based encryption scheme like ElGamal, the description of the cyclic group $\mathbb{G}$ is public.

- **DDH Instantiation:** On input a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2) \in \mathbb{G} \times \mathbb{G}$, output

$$\mathsf{m}' = \left(\mathsf{ct}_1^k\right)^{-1} \cdot \mathsf{ct}_2.$$

If $(\mathsf{ct}_1, \mathsf{ct}_2) = (g^*, h^* \cdot \mathsf{m})$, we have

$$\mathsf{m}' = \left(\mathsf{ct}_1^k\right)^{-1} \cdot \mathsf{ct}_2 = (h^*)^{-1} \cdot (h^* \cdot \mathsf{m}) = \mathsf{m}.$$

Note that the decryption in the IHwPRF construction works even when $\mathcal{X}$ and $\mathcal{Y}$ are non-abelian groups.

We summarize the main steps in the construction of PKE from IHwPRF in Figure 5, and compare it with the DDH-instantiation over cyclic groups of prime order. Observe that the DDH-based PKE described above is very similar to ElGamal encryption [ElG84]. In fact, it can be viewed as a form of ElGamal encryption where we use a less efficient method to create the group elements (g, h) and (g^*, h^*): namely, in order to get a random element, we take a subset product of many public elements rather than just raising a single element to a random power.

Setup:

$F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$	$\rightsquigarrow$	$(\mathbb{G}, q)$
$\mathrm{X} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$	$\rightsquigarrow$	$\mathrm{G} = \{g_{j,b} \leftarrow \mathbb{G}\}_{j\in[n],b\in\{0,1\}}$
$k \leftarrow \mathcal{K}$	$\rightsquigarrow$	$k \leftarrow \mathbb{Z}_q$
$\mathrm{Y} = \{y_{j,b} = F(k, x_{j,b})\}_{j\in[n],b\in\{0,1\}}$	$\rightsquigarrow$	$\mathrm{H} = \left\{h_{j,b} = g_{j,b}^k\right\}_{j\in[n],b\in\{0,1\}}$
$\boxed{\mathsf{sk} = k, \quad \mathsf{pk} = (\mathrm{X}, \mathrm{Y})}$	$\rightsquigarrow$	$\boxed{\mathsf{sk} = k, \quad \mathsf{pk} = (\mathrm{G}, \mathrm{H})}$

Encrypt:

$\mathsf{m} \in \mathcal{Y}$	$\rightsquigarrow$	$\mathsf{m} \in \mathbb{G}$
$\mathsf{s} \leftarrow \{0,1\}^n$	$\rightsquigarrow$	$\mathsf{s} \leftarrow \{0,1\}^n$
$x^* = \bigoplus_{j\in[n]} x_{j,s_j}$	$\rightsquigarrow$	$g^* = \prod_{j=1}^n g_{j,s_j}$
$y^* = \bigotimes_{j\in[n]} y_{j,s_j}$	$\rightsquigarrow$	$h^* = \prod_{j=1}^n h_{j,s_j}$
$\boxed{\mathsf{ct} = (x^*, y^* \otimes \mathsf{m})}$	$\rightsquigarrow$	$\boxed{\mathsf{ct} = (g^*, h^* \cdot \mathsf{m})}$

Decrypt:

$\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$	$\rightsquigarrow$	$\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$
$\boxed{\mathsf{m}' = [F(k, \mathsf{ct}_1)]^{-1} \otimes \mathsf{ct}_2}$	$\rightsquigarrow$	$\boxed{\mathsf{m}' = \left(\mathsf{ct}_1^k\right)^{-1} \cdot \mathsf{ct}_2}$

Figure 5: PKE from IHwPRF and DDH Instantiation

This leads us to the following question: how far can we go if we take traditional DDH-based schemes and write them as IHwPRFs? For schemes that require two exponentiations, we could write the first exponentiation as a "subset sum", and then the second as a IHwPRF evaluation. This is essentially how our

DDH-based instantiation of PKE from IHwPRF works. In what follows, we illustrate this comparison via a non-interactive key exchange protocol.

We show a non-interactive key exchange protocol from IHwPRFs in Figure 6. For illustration, we compare it with the Diffie-Hellman key exchange protocol. In the IHwPRF setting, the (randomly sampled) "base elements" $\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ are publicly available to both parties at the beginning of the protocol. Given the "base elements", there are two ways to arrive at the final secret y^*. The first way is to apply the IHwPRF on the "base elements", followed by applying a "subset-product" in the output space of the IHwPRF. The second way is to first do a "subset-sum" on the base elements, and then apply the IHwPRF. The two parties involved in the protocol each use one of these strategies. Security of the protocol follows from the weak pseudorandomness of F and one-wayness of "subset-sums" in its input space, where the latter is also implied by the weak pseudorandomness of F.

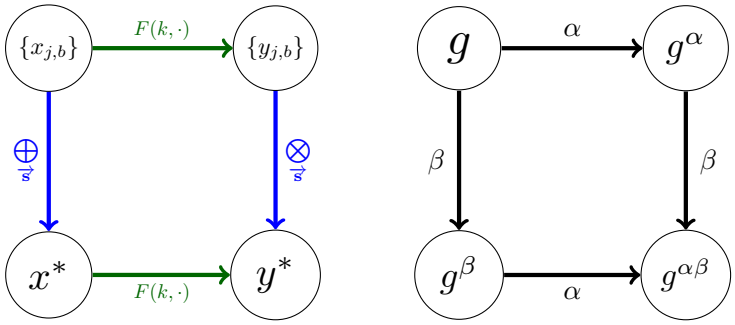

Figure 6: Visualization of Non-Interactive Key Exchange from IHwPRF

Finally, the reader may observe that the protocol is secure even if the function F is an IHwUF instead of an IHwPRF, provided that both parties extract a "hardcore bit" from the secret y^* and use it as the key.[1] Similarly, one can construct a CPA-secure PKE from IHwUF by using the hardcore bit of the secret y^* to mask the message bit.

1.4.2 Extending the Scheme with a General Protocol

It turns out that we can do substantially more than just PKE, as an examination of the above protocol might suggest. It turns out we can take *any* one-round[2] CDH/DDH-based protocol and convert it into a (less efficient) protocol using a general IHwUF/IHwPRF. We defer the details to Section 3.3, but the basic idea is the following: visualize one-round CDH/DDH schemes as protocols played by two parties with the following four phases. Below is a rough description of this protocol (see Section 3.3 for the precise mathematical flow and definitions):

- **Initialization**: Setting up the group and any random elements needed for the protocol.
- **Pre-Evaluation**: The first party exponentiates some (or all) of the random elements from the initialization stage and sends some (or all) of these to the second player.
- **Evaluation**: The second party exponentiates some of the elements from the first player and potentially some of the elements from initialization as well. The second player potentially publishes some of these elements as well.

[1] Note that the protocol assumes that the input space of the IHwUF/IHwPRF is independent of the choice of key. See Section 4.1 for more details.

[2] Informally, in our context this means a protocol that can be "played" by two parties with a simple out-and-back communication flow, along with any PPT computation the parties choose to do before, during, or after the communication.

- **Post-Evaluation**: Either party can multiply/invert/process the elements, and may publish some outputs of these.

It turns out that the vast majority of CDH/DDH-based cryptosystems fall into this archetype, and thus we can build them using an IHwUF/IHwPRF. Among other implications, this approach encompasses recent constructions such as (anonymous) IBE from CDH/DDH and a number of other works in the same vein [DG17b, DG17a, BLSV18, DGHM18, GH18, KW18, GGH18]. Although these works use many novel techniques, we show that the CDH/DDH-related portion of the constructions can be boiled down to something that fits within the above framework. The few protocols that cannot be handled involve at least three exponentiations (and cannot be rewritten as less efficient protocols with two or less exponentiations).

We can use our general protocol and the ideas around it to build many cryptosystems. In the following subsection, we outline some of the constructions that we consider interesting.

1.4.3 Batch Encryption from IHwUFs

In a recent work, Brakerski *et al.* [BLSV18] introduced and formalized a powerful cryptographic primitive called *batch encryption*. Roughly speaking, the basic idea of batch encryption is the following: a user encrypts a $2 \times N$ matrix of bits, and decryption *selectively* reveals only N of these bits–one in each column. For a given column, which bit is revealed depends on the value of the secret key used for decryption.

Brakerski *et al.* showed that batch encryption can be used in conjunction with garbled circuits to construct identity-based encryption (IBE).[1] In fact, when equipped with a stronger property called "blinding", batch encryption was shown to imply anonymous IBE, KDM-CPA secure PKE, and leakage resilient PKE [BLSV18]. The authors of [BLSV18] showed how to construct batch encryption from concrete assumptions, so it is natural to ask the following question: is there a generic primitive that implies batch encryption?

In this subsection, we answer this question in the affirmative by showing that IHwUFs are sufficient to construct blind batch encryption. This in turn implies that IHwUFs are sufficient to construct anonymous IBE, KDM-secure PKE and leakage-resilient PKE as well.[2] We begin by defining blind batch encryption informally, and then illustrate how to construct the same from any IHwUF family. [3]

Batch Encryption. A batch encryption scheme is a public-key encryption scheme in which the key generation algorithm Gen "projects" a secret string $\mathbf{s} \in \{0,1\}^n$ onto a corresponding hash value $h \in \{0,1\}^\ell$, such that $\ell < n$. Corresponding to this "projection" function, there should exist encryption and decryption algorithms such that:

- The encryption algorithm $\mathsf{Enc}(\mathsf{pp}, h, i, (\mathsf{m}_0, \mathsf{m}_1))$ takes as input the public parameter pp associated with the projection function, a hash $h \in \{0,1\}^\ell$, a position index $i \in [n]$ and a pair of message-bits $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\}^2$, and outputs a ciphertext ct.
- The decryption algorithm $\mathsf{Dec}(\mathsf{pp}, \mathbf{s}, i, \mathsf{ct})$ takes as input a ciphertext ct and a secret string $\mathbf{s}$, and then recovers m_{s_i} where s_i is the value of the i^{th}-bit of $\mathbf{s}$, provided that ct was generated using $h = \mathsf{Gen}(\mathsf{pp}, \mathbf{s})$.

In other words, a decryptor can use the knowledge of the preimage $\mathbf{s}$ of a hash output string $h \in \{0,1\}^\ell$ to decrypt *exactly one* of the two encrypted messages, depending on the i^{th}-bit of $\mathbf{s}$. The security requirement is roughly that the distributions

$$\{\mathsf{pp}, \mathbf{s}, \mathsf{Enc}(\mathsf{pp}, h = \mathsf{Gen}(\mathsf{pp}, \mathbf{s}), i, (\mathsf{m}_{s_i}, \mathsf{m}_{1-s_i}))\}_{\mathbf{s} \in \{0,1\}^n} \quad \text{and}$$
$$\{\mathsf{pp}, \mathbf{s}, \mathsf{Enc}(\mathsf{pp}, h = \mathsf{Gen}(\mathsf{pp}, \mathbf{s}), i, (\mathsf{m}_{s_i}, \mathsf{m}^*))\}_{\mathbf{s} \in \{0,1\}^n, \mathsf{m}^* \leftarrow \{0,1\}}$$

[1] An equivalent cryptosystem, named as *hash encryption*, was introduced by Döttling *et al.*in [DGHM18].

[2] The construction of anonymous IBE requires an additional primitive - "blind garbled circuits" besides blind batch encryption. However, blind garbled circuits are implied by any one-way function, and are hence also implied by IHwUFs.

[3] We can analogously construct blind batch encryption from γ-bounded IHwUFs. For simplicity, we show the construction from a "full" IHwUF here. The reader may refer to Section 4.4 for the details of the construction from γ-bounded IHwUFs.

are computationally indistinguishable. In fact, as Brakerski et al. pointed out in [BLSV18], a weaker *selective* notion of security suffices, where the adversary commits to a string $\mathbf{s} \in \{0,1\}^n$ and an index $i \in [n]$ before the public parameter pp is published.

Note that the adaptive security guarantee implicitly requires the projection function to be collision-resistant; otherwise, a PPT adversary could distinguish an encryption of m_{1-s_i} from random with non-negligible probability simply by generating a different preimage $\mathbf{s}'$ of h such that $s'_i \neq s_i$.

An additional security requirement, called "blindness" was formalized with respect to batch encryption in [BLSV18]. Roughly, a batch encryption scheme is said to be blind if the ciphertext ct can be decomposed into parts $(\mathsf{ct}_1, \mathsf{ct}_2)$ such that the marginal distribution of ct_1 is independent of both the image string h and the message pair $(\mathsf{m}_0, \mathsf{m}_1)$, while the marginal distribution of ct_2 is uniform whenever the message pair $(\mathsf{m}_0, \mathsf{m}_1)$ is uniform in $\{0,1\}^2$.

Projection Function from IHwUF. The first step in instantiating a batch encryption scheme is to realize the projection function. Given an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, we define $\mathsf{Gen}_{\mathrm{IHwUF}}(\mathsf{pp}, \mathbf{s})$ to output

$$h = \bigoplus_{j \in [n]} x_{j,s_j},$$

where $\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ is a set of uniformly random elements in the input group of the IHwUF, published as part of the public parameter pp. We claim that this function is both one-way and collision resistant, provided that $n > 3\log|\mathcal{X}|$.[1]

One-wayness. To see that this function is one-way, consider a PPT adversary $\mathcal{A}$ that, given uniformly random group elements $\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ and a "target" element x^*, outputs a vector $\mathbf{s} \in \{0,1\}^n$ such that

$$x^* = \bigoplus_{j \in [n]} x_{j,s_j}.$$

One can then construct a PPT algorithm $\mathcal{B}$ that on input $\{x_{j,b}, F(k, x_{j,b})\}_{j\in[n],b\in\{0,1\}}$ (where each $x_{j,b}$ is uniformly random) and a uniformly random target element x^*, invokes $\mathcal{A}$ as a subroutine on the tuple $\left\{x^*, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right\}$ to obtain $\mathbf{s} \in \{0,1\}^n$ and outputs

$$F(k, x^*) = \bigotimes_{j \in [n]} F\left(k, x_{j,s_j}\right),$$

which violates the weak unpredictability of the function F. Note that the reduction is valid since for $n > 3\log|\mathcal{X}|$, the existence of some $\mathbf{s} \in \{0,1\}^n$ such that $x^* = \bigoplus_{j\in[n]} x_{j,s_j}$ is guaranteed for almost all $x^* \in \mathcal{X}$ by the leftover hash lemma [IZ89].

Collision-Resistance. To see that this function is collision-resistant, consider a PPT adversary $\mathcal{A}$ that, given uniformly random group elements $\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$, outputs $(\mathbf{s}, \mathbf{s}') \in \{0,1\}^n \times \{0,1\}^n$ such that $\mathbf{s} \neq \mathbf{s}'$ and

$$\bigoplus_{j \in [n]} x_{j,s_j} = \bigoplus_{j \in [n]} x_{j,s'_j}.$$

One can then construct a PPT algorithm $\mathcal{B}$ that on input $\{x_{j,b}, F(k, x_{j,b})\}_{j\in[n],b\in\{0,1\}}$ (where each $x_{j,b}$ is uniformly random) and a random target element x^*, uniformly guesses $i \leftarrow [n]$, resets $x_{i,0} := x^*$ and invokes $\mathcal{A}$ as a subroutine on the modified set $\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ to obtain a collision $(\mathbf{s}, \mathbf{s}')$. If $s_i = s'_i$, it aborts. Otherwise, it exploits the homomorphism of the function F to output $F(k, x^*)$. Since the probability that $\mathbf{s}$ and $\mathbf{s}'$ differ in the i^{th} bit is at least $1/n$, $\mathcal{B}$ breaks the weak unpredictability of F.

[1] We note that it is possible to use a smaller constant, but we use 3 through the whole paper for the sake of simplicity.

Encryption and Decryption. Corresponding to the projection function as described above, we realize our encryption procedure $\mathsf{Enc}_{\mathrm{IHwUF}}(\mathsf{pp}, h, i, (\mathsf{m}_0, \mathsf{m}_1))$ as follows: sample $k_0, k_1 \leftarrow \mathcal{K}$ and set the following

$$\begin{aligned} y^{(0)}_{j,0} &= F(k_0, x_{j,b}) \quad , \quad y^{(1)}_{j,1} = F(k_1, x_{j,b}) \quad \text{for } j \in [n] \setminus \{i\}, b \in \{0,1\} \\ y^{(0)}_{i,0} &= F(k_0, x_{i,0}) \quad , \quad y^{(1)}_{i,0} = \perp, \\ y^{(0)}_{i,1} &= \perp \quad , \quad y^{(1)}_{i,1} = F(k_1, x_{i,1}). \end{aligned}$$

Next, mask the messages $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\} \times \{0,1\}$ as follows:[1]

$$\begin{aligned} \mathsf{e}_0 &= \mathrm{XOR}(\mathrm{HardCore}(F(k_0, h)), \mathsf{m}_0) \\ \mathsf{e}_1 &= \mathrm{XOR}(\mathrm{HardCore}(F(k_1, h)), \mathsf{m}_1). \end{aligned}$$

Output the ciphertext as

$$\mathsf{ct} = \left(\mathsf{ct}_1 = \left\{y^{(0)}_{j,b}, y^{(1)}_{j,b}\right\}_{j \in [n], b \in \{0,1\}}, \mathsf{ct}_2 = (\mathsf{e}_0, \mathsf{e}_1)\right).$$

Given a preimage string $\mathbf{s}$, our decryption algorithm $\mathsf{Dec}_{\mathrm{IHwUF}}(\mathsf{pp}, \mathbf{s}, i, \mathsf{ct})$ now recovers m_{s_i} as

$$\mathsf{m}_{s_i} = \mathrm{XOR}\left(\mathrm{HardCore}\left(\bigotimes_{j \in [n]} y^{(s_i)}_{j,s_j}\right), \mathsf{e}_{s_i}\right).$$

Correctness follows from the homomorphic property of the function F. Observe that irrespective of the value of the bit s_i, m_{s_i} can always be recovered as the decryptor has access to $y^{(b)}_{i,b}$ for each $b \in \{0,1\}$. However, it cannot recover m_{1-s_i} since it does not have access to $y^{(b)}_{i,1-b}$ for either $b = 0$ or $b = 1$. In addition, we note that, unlike existing constructions, our construction does not require the groups $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ to be abelian for correctness to hold.

Security. We now sketch our security proof. Suppose we are given an adversary $\mathcal{A}$ that breaks the security of this scheme. We construct a PPT algorithm $\mathcal{B}$ that breaks the weak unpredictability of the function F. We assume that $\mathcal{B}$ has oracle access to an IHwUF F with key k.

In our security game, $\mathcal{B}$ receives a uniformly random challenge element x^* and a bit $\mathsf{e}^* \in \{0,1\}$ such that $\mathsf{e}^* = \mathrm{HardCore}(F(k, x^*))$ (the "real" case) or e^* is a uniform bit (the "random" case). The goal of $\mathcal{B}$ is to output a bit b, such that

$$b = \begin{cases} 0 & \text{if } \mathsf{e}^* = \mathrm{HardCore}(F(k, x^*)) \\ 1 & \text{if } \mathsf{e}^* \leftarrow \{0,1\} \end{cases}$$

In other words, $\mathcal{B}$ must distinguish the hardcore bit associated with the output of $F(k, x^*)$ from random (which is equivalent to constructing the entire output $F(k, x^*)$)[2] using the adversary $\mathcal{A}$.

We note here that the exact value of n is typically chosen by the adversary $\mathcal{A}$ at the beginning of the game, subject to the restriction that $n > 3 \log |\mathcal{X}|$. For simplicity, we describe the interaction between $\mathcal{B}$ and $\mathcal{A}$ after the value of n has been chosen.

- The adversary $\mathcal{A}$ chooses an arbitrary preimage string $\mathbf{s} \in \{0,1\}^n$ and an index $i \in [n]$, and provides $(\mathbf{s}, i)$ to $\mathcal{B}$.

[1] We assume that each group element $y \in \mathcal{Y}$ has a deterministic hardcore bit, denoted as HardCore(y). If a deterministic hardcore bit is not known then we can use the Goldreich-Levin [GL89] construction.

[2] By the Goldreich-Levin Theorem [GL89], this can be used to build an algorithm that constructs $F(k, x^*)$ with only polynomial loss in advantage.

- $\mathcal{B}$ queries the IHwUF F a total of $2n$ times, getting a tuple of the form

$$\{x_{j,b}, F(k, x_{j,b})\}_{j\in[n],b\in\{0,1\}}.$$

- $\mathcal{B}$ now resets

$$x_{i,s_i} := \left(\bigoplus_{j\in[i-1]} x_{j,s_j}\right)^{-1} \oplus x^* \oplus \left(\bigoplus_{j\in[i+1,n]} x_{j,s_j}\right)^{-1},$$

and provides $\mathsf{pp} = \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ to $\mathcal{A}$. In other words, $\mathcal{B}$ fixes x^* to be the image of $\mathbf{s}$ under the projection function parameterized by pp.

- The adversary $\mathcal{A}$ generates $\mathbf{m}^{(0)} = \left(\mathsf{m}_0^{(0)}, \mathsf{m}_1^{(0)}\right)$ and $\mathbf{m}^{(1)} = \left(\mathsf{m}_0^{(1)}, \mathsf{m}_1^{(1)}\right)$ such that $\mathsf{m}_{s_i}^{(0)} = \mathsf{m}_{s_i}^{(1)}$, and sends them to $\mathcal{B}$.

- In response, $\mathcal{B}$ samples $k' \leftarrow \mathcal{K}$, and implicitly fixes $k_{s_i} := k'$ and $k_{1-s_i} := k$. It then sets the following

$$\begin{aligned}
y_{j,s_j}^{(s_i)} = F\left(k', x_{j,s_j}\right) \quad &, \quad y_{j,s_j}^{(1-s_i)} = F\left(k, x_{j,s_j}\right) \quad \text{for } j \in [n]\setminus\{i\}, b \in \{0,1\},\\
y_{i,s_i}^{(s_i)} = F\left(k', x_{i,s_i}\right) \quad &, \quad y_{i,s_i}^{(1-s_i)} = \perp,\\
y_{i,1-s_i}^{(s_i)} = \perp \quad &, \quad y_{i,1-s_i}^{(1-s_i)} = F\left(k, x_{i,1-s_i}\right).
\end{aligned}$$

To mask the messages, $\mathcal{B}$ sets the following

$$\begin{aligned}
\mathsf{e}_{s_i}^{(0)} &= \mathrm{XOR}\left(\mathrm{HardCore}\left(F\left(k', x^*\right)\right), \mathsf{m}_{s_i}^{(0)}\right), \quad \mathsf{e}_{1-s_i}^{(0)} = \mathrm{XOR}\left(\mathsf{e}^*, \mathsf{m}_{1-s_i}^{(0)}\right),\\
\mathsf{e}_{s_i}^{(1)} &= \mathrm{XOR}\left(\mathrm{HardCore}\left(F\left(k', x^*\right)\right), \mathsf{m}_{s_i}^{(1)}\right), \quad \mathsf{e}_{1-s_i}^{(1)} = \mathrm{XOR}\left(\mathsf{e}^*, \mathsf{m}_{1-s_i}^{(1)}\right).
\end{aligned}$$

Finally, $\mathcal{B}$ samples $b^* \leftarrow \{0,1\}$ and sends ct to $\mathcal{A}$ where

$$\mathsf{ct} = \left(\mathsf{ct}_1 = \left\{y_{j,b}^{(0)}, y_{j,b}^{(1)}\right\}_{j\in[n],b\in\{0,1\}}, \mathsf{ct}_2 = \left(\mathsf{e}_0^{(b^*)}, \mathsf{e}_1^{(b^*)}\right)\right).$$

- $\mathcal{A}$ outputs a bit b'. If $b^* = b'$, $\mathcal{B}$ outputs 1. Otherwise it outputs 0.

Note that when $\mathsf{e}^* = \mathrm{HardCore}\left(F\left(k, x^*\right)\right)$, the challenge ciphertext is generated perfectly. On the other hand, when e^* is a uniform bit, the adversary $\mathcal{A}$ has no advantage since $\mathsf{m}_{s_i}^{(0)} = \mathsf{m}_{s_i}^{(1)}$ by definition. Hence, the advantage of $\mathcal{B}$ is negligibly different from the advantage of $\mathcal{A}$.

Blindness. The aforementioned batch encryption scheme is additionally "blind". This follows from the fact that the ciphertext component ct_1 is independent of both the image string h and the message-pair $(\mathsf{m}_0, \mathsf{m}_1)$. Additionally, if $(\mathsf{m}_0, \mathsf{m}_1)$ is uniform in $\{0,1\}^2$, then the distribution of ct_2 is also uniform.

1.4.4 More Primitives

Recyclable OWFE. In a recent work, Garg and Hajiabadi [GH18] introduced a cryptographic primitive called recyclable *one-way function with encryption* (OWFE), and showed that recyclable OWFEs imply trapdoor functions (TDFs) with negligibly small inversion error. They also showed how to construct recyclable OWFE from the CDH assumption, which in turn gave the first TDF construction from the CDH assumption. In a more recent follow-up, Garg *et al.* [GGH18] introduced a strengthened version of recyclable OWFE called *smooth* recyclable OWFE, and showed how to realize the same from CDH assumption. They showed that

this strengthened primitive implies TDFs with almost-perfect correctness and CCA2-secure deterministic encryption, where the CCA2-security holds with respect to plaintexts sampled from distributions with super-logarithmic min-entropy.

We show in Section 4.3 that IHwUFs imply smooth recyclable OWFE, thereby answering the question of whether this cryptosystem can be constructed from a generic primitive. This shows that IHwUFs also imply TDFs with almost-perfect correctness and CCA2-secure deterministic encryption for plaintexts sampled from distributions with super-logarithmic min-entropy. The techniques for this construction are similar to those presented for batch encryption. We refer the reader to Section 4.3 for the details of the construction.

Hinting PRG. A "hinting PRG" is a stronger variant of traditional PRGs introduced by Koppula and Waters in [KW18], who show that hinting PRGs can be used to generically transform any CPA-secure attribute-based encryption scheme or one-sided predicate encryption scheme into a CCA-secure counterpart. Informally, a hinting PRG takes n bits as input and outputs $n \cdot \ell$ output bits with the restriction that no PPT adversary can distinguish between $2n$ uniformly random strings and $2n$ strings such that half the strings are output by the PRG, and the remaining half are uniformly random, where the strings are arranged as a $2 \times n$ matrix as follows: in the i^{th} column of this matrix, the top entry is pseudorandom and the bottom entry is random if the i^{th} bit of the seed is 0; otherwise the bottom entry is pseudorandom and top entry is random.

Koppula and Waters [KW18] showed explicit constructions of hinting PRG families from the CDH and LWE assumptions. We show in Section 4.3 that any IHwUF family can be used to construct a hinting PRG, thereby answering the question of whether hinting PRGs can be constructed from a generic primitive. The techniques for our construction are also similar to those presented for batch encryption. We refer the reader to Section 4.3 for the details of the construction.

CRHF and More from HOWF. Informally, a HOWF is just a one-way function $f : \mathcal{X} \to \mathcal{Y}$ with the following additional properties: the input space $\mathcal{X}$ and the output space $\mathcal{Y}$ are groups with group operations $\oplus$ and $\otimes$, respectively, and

$$f(x_1 \oplus x_2) = f(x_1) \otimes f(x_2).$$

In this paper, we show that any HOWF can used to construct a collision-resistant hash function (CRHF) family that maps bit strings to elements in the output space of the HOWF. In addition, we show constructions of Schnorr signatures and chameleon hash functions from HOWFs. The reader may refer to Appendix A for the details of these constructions, and instantiations of HOWF from concrete assumptions.[1]

1.4.5 Richer Structures

As mentioned earlier, we can also consider richer structures than just a group homomorphism over a Minicrypt primitive. In this section, we provide more details for two of these more structured primitives, namely Ring IHwPRFs and L-composable IHwPRFs.

Ring IHwPRFs. We first informally define a Ring Input-Homomorphic weak PRF (RIHwPRF). Let $(R, +, \times)$ and $(\boxed{R}, \boxplus, \boxtimes)$ be two efficiently samplable rings such that the ring operations are efficiently computable. An RIHwPRF is a weak PRF

$$F : \mathcal{K} \times \boxed{R} \to R$$

(with input space $\boxed{R}$ and output space R) such that for every key $k \in \mathcal{K}$ the mapping $F(k, \cdot) : \boxed{R} \to R$ is a ring homomorphism from $\boxed{R}$ to R.[2]

[1] Here we use "full" HOWF for simplicity. We also consider "bounded" HOWFs for which only a bounded number of homomorphic operations is allowed. The notion of bounded HOWFs works for all of the applications that we consider in almost the same way that full HOWFs do.

[2] It is also possible to define (bounded) RIHwPRFs similar to IHwPRFs, but we only consider unbounded homomorphism here for the sake of simplicity.

We outline a simple construction of symmetric-key FHE from an RIHwPRF F provided that the size of output space of F is polynomial in the security parameter, i.e., $|R| \leq \text{poly}(\lambda)$. Using the generic transformation in [Rot11], one can obtain a public-key FHE from a symmetric-key FHE. For the description of the scheme, see Appendix D. The construction is as follows:

- Given an RIHwPRF $F : \mathcal{K} \times \boxed{R} \to R$, publish its description as the public parameters. To generate a secret key sample a key $k \leftarrow \mathcal{K}$.
- To encrypt a bit $\mathsf{m} \in \{0,1\}$ under key k, sample a preimage $\mathsf{ct} \leftarrow \boxed{R}$ such that $F(k, \mathsf{ct}) = \mathsf{m}_R$ and publish ct as the ciphertext.[1] (Notice that 0_R and 1_R are the multiplicative and the additive identity elements of R, respectively.)
- To decrypt a ciphertext $\mathsf{ct} \in \boxed{R}$ under key k, output m' where

$$\mathsf{m}' = \begin{cases} 0 & \text{if } F(k, \boxed{r}) = 0_R \\ 1 & \text{if } F(k, \boxed{r}) = 1_R \\ \perp & \text{otherwise.} \end{cases}$$

- To evaluate a (homomorphic) $\mathsf{NAND}(\mathsf{ct}, \mathsf{ct}')$ operation, output $\boxed{1} \boxminus \mathsf{ct} \boxtimes \mathsf{ct}'$ where $\boxed{1}$ is the identity element of $\boxed{R}$ with respect to addition, and $\boxminus$ is the subtraction in the ring $\boxed{R}$.

The security of the scheme follows from a standard hybrid argument. Observe that by ring-homomorphism of F, if ct and ct' are valid ciphertexts encrypting m and m' respectively, decrypting $\boxed{1} \boxminus \mathsf{ct} \boxtimes \mathsf{ct}'$ gives $\mathsf{NAND}(\mathsf{m}, \mathsf{m}')$.

L-Composable IHwPRFs. We first describe 2-Composable IHwPRFs before generalizing to $L \geq 2$. Informally, a two-composable IHwPRF is a collection of two functions and two "composers"

$$\begin{aligned} F_1 : \mathcal{K} \times \mathcal{X}_1 \to \mathcal{Y}_1 \quad , \qquad F_2 : \mathcal{K} \times \mathcal{X}_2 \to \mathcal{Y}_2, \\ C_1 : \mathcal{Y}_1 \times \mathcal{X}_2 \to \mathcal{Z} \quad , \qquad C_2 : \mathcal{Y}_2 \times \mathcal{X}_1 \to \mathcal{Z}. \end{aligned}$$

such that the functions are IHwPRFs and the composers are weak PRFs. Additionally, the following composition property holds: for every $k \in \mathcal{K}$ and for every $x_1, x_2 \in \mathcal{X}$, we have:

$$C_1\left(F_1\left(k, x_1\right), x_2\right) = C_2\left(F_2\left(k, x_2\right), x_1\right), \text{ both denoted } F_T(k, (x_1, x_2)).$$

This primitive gives us 3-party non-interactive key exchange (NIKE) in the following way: the public key includes vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. Two of the parties generate secret subsets $\mathbf{s}_1$ and $\mathbf{s}_2$, and publish the group elements

$$\bigoplus_{j \in [n]} x^{(1)}_{j, s_{1,j}}, \quad \bigoplus_{j \in [n]} x^{(2)}_{j, s_{2,j}},$$

respectively. The 3rd party generates a secret key k and publishes $F_1(k, \mathbf{x}^{(1)})$ and $F_2(k, \mathbf{x}^{(2)})$. Each party computes the shared key:

$$F_T\left(k, \left(\bigoplus_{j \in [n]} x^{(1)}_{j, s_{1,j}}, \bigoplus_{j \in [n]} x^{(2)}_{j, s_{2,j}}\right)\right),$$

which can be computed from any party's secret and the other parties' outputs, using the composition property and input homomorphism of F_1 and F_2. Security follows by the weak PRF properties and LHL.

We give formal definitions and proofs in Appendix C, where we also show that an analog of Boneh-Franklin IBE [BF01] can be constructed from 2-composable IHwPRFs. However, we still argue in Appendix C that

[1] Such a preimage can be efficiently sampled by weak pseudorandomness of F and the fact that the order of the ring is polynomial.

2-composable IHwPRFs are seemingly much weaker than bilinear pairing groups. Specifically, we argue that the general abstraction of dual system groups (DSG [CGW15]) is hard to capture in the 2-Composable IHwPRF setting due to the following limitations:

1. DSG seems to require properties that translate to the requirement of key homomorphism in the 2-composable IHwPRF setting.
2. DSG also requires algebraic interaction on both of the coordinates. Realizing this in the IHwPRF setting forces both the coordinate domains $\mathcal{X}_1$ and $\mathcal{X}_2$ to be *ring homomorphic* on a single ring, where all the algebra can take place.

The currently known constructions of rich ABEs like fuzzy IBEs [SW05], spatial encryption [BH08] and monotone span program ABEs [GPSW06] from bilinear groups all require at least one of the properties just described. Since the only instantiation of 2-composable IHwPRFs we know of are bilinear groups, it seems difficult to achieve these rich ABEs without restricting 2-composable IHwPRFs to almost traditional bilinear groups.

Thus we see a seeming separation in the amount of structure that we need for 3-party NIKE and simple IBE (in RO) from that seemingly necessary for NIZKs (without RO) and rich ABEs. This poses a tantalizing question: *Can we construct a 3-party NIKE protocol from a weaker primitive than bilinear pairing groups?* In other words, can we achieve the structure of 2-composability from concrete assumptions, e.g., lattice-based assumptions, that do not naturally imply bilinear pairings?

Generalizing to L $\geq$ 2. In the general setting, which we formalize in Appendix C.3, we consider L inner IHwPRFs F_i and L different composers which satisfy an analogous composition property as the 2-composable setting. By a straightforward generalization, we get an $(L+1)$-party non-interactive key exchange from an L-Composable IHwPRF, which is not known from any $(< L)$-Composable IHwPRFs. We also do not know how to construct such a protocol from any hard $(< L)$-multilinear group. We still observe an analogous seeming separation in the amount of structure that we need for multi-party non-interactive key exchange from that seemingly necessary for circuit ABEs and iOs. The corresponding open question is whether we can build the former from weaker primitives that may lack the structure needed for the latter.

1.5 Conclusion and Future Work

In this paper, we presented a framework to build many cryptosystems from Minicrypt primitives with structure. Our framework allows us to categorize many cryptosystems based on which structured Minicrypt primitive implies them, and potentially makes showing the *existence* of many cryptosystems from novel assumptions substantially easier in the future. In addition, some of our constructions are novel in their own right. Although our framework does yield new constructions from less studied assumptions, the main focus of this work is to investigate what kind of structure, when added to to simple and natural Minicrypt primitives, implies advanced cryptosystems like IBE. Hence, we are not explicitly examining new constructions from a mainstream assumption. We believe that our work opens up a substantial number of questions, some of which we mention here.

Primitives from Weaker Assumptions. A pertinent open question is: can we build some of the Cryptomania primitives discussed in this paper from weaker Minicrypt primitives with structure. For instance, can we build PKE from HOWFs (which would imply PKE from discrete log)? Can we build PIR/lossy TDFs from IHwUFs (which would imply the first PIR/lossy TDFs from CDH)? Is it possible to build round-optimal OT and MPC in the plain model from IHwUFs/IHwPRFs?

More Primitives. While we constructed many popularly used Cryptomania primitives from our framework, we could not encompass many others. These (non-exhaustively) include primitives implied by bilinear pairings such as NIZK, unique signatures, VRFs, ABE and PE, and primitives known from specific assumptions such

as worst-case smooth hash proof systems, KDM-CCA secure PKE and dual-mode cryptosystems. It is open to construct one or more of these primitives from simple Minicrypt primitives with structure.

New Assumptions. One of the nicest aspects of our work is the implications for new assumptions. If a new assumption implies one of the Minicrypt primitives with structure discussed in this paper, then it immediately implies a whole host of cryptographic primitives. We leave it open to build HOWFs/IHwUFs/IHwPRFs from new concrete assumptions, which in conjunction with our framework would allow building a large number of Cryptomania primitives from such assumptions.

"Continents" of Cryptomania. We leave it open to explore if there are even weaker forms of structure that, when endowed upon Minicrypt primitives, lead to interesting implications in Cryptomania. It is also interesting to explore non-trivial separations between these structured primitives, e.g., between HOWFs and IHwUFs. Such separations would potentially allow us to divide the world of Cryptomania into many "continents" of primitives, where each "continent" is entirely implied by some simple Minicrypt primitive with structure.

2 Preliminaries

Notation. For any positive integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. We use λ for the security parameter. We use the symbols $\oplus$ and $\otimes$ as group operations defined in the context. For two equal-length strings s_1 and s_2 we denote their bitwise XOR as $\mathrm{XOR}(s_1, s_2)$. For a finite set S, we use $s \leftarrow S$ to sample uniformly from the set S.

Unpredictable Functions. Informally, an efficiently computable function is called unpredictable if there exists no PPT adversary that can predict its output on a new input that has not been queried. More formally, a UF family is an efficiently computable function family $\{F(k, \cdot) : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$ (where K, $\mathcal{X}$ and $\mathcal{Y}$ are indexed by the security parameter λ) such that for all PPT adversaries $\mathcal{A}$ we have

$$\Pr[\mathcal{A}^{F(k,\cdot)} = (x^*, F(k, x^*))] \leq \mathrm{negl}(\lambda),$$

where $k \leftarrow \mathcal{K}$ and x^* is any arbitrary group element in $\mathcal{X}$.

Weak Unpredictable Functions. A weaker notion of unpredictability requires the unpredictability guarantee to hold when the adversary sees evaluations of the UF on uniformly random inputs, and should predict its output on a new uniformly random input. More formally, let $F^{\$}(k)$ be a *randomized* oracle that when queried samples $x \leftarrow \mathcal{X}$ and outputs $(x, F(k, x))$. A weak UF (wUF) family is an efficiently computable function family $\{F(k, \cdot) : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$ (where K, $\mathcal{X}$ and $\mathcal{Y}$ are indexed by the security parameter λ) such that for all PPT adversaries $\mathcal{A}$ we have

$$\Pr[\mathcal{A}^{F^{\$}(k)}(x^*) = F(k, x^*)] \leq \mathrm{negl}(\lambda),$$

where $k \leftarrow \mathcal{K}$ and x^* is an uniformly randomly sampled group element in $\mathcal{X}$.

Pseudorandom Functions. Informally, an efficiently computable function is called pseudorandom if there exists no PPT adversary that can distinguish it from a truly random function. More formally, a PRF family is an efficiently computable function family $\{F(k, \cdot) : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$ (where K, $\mathcal{X}$ and $\mathcal{Y}$ are indexed by the security parameter λ) such that for all PPT adversaries $\mathcal{A}$ we have

$$\left|\Pr[\mathcal{A}^{F(k,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(1^\lambda) = 1]\right| \leq \mathrm{negl}(\lambda),$$

where $k \leftarrow \mathcal{K}$ and $f : \mathcal{X} \to \mathcal{Y}$ is a (truly) random function.

Weak Pseudorandom Functions. A weaker variant of pseudorandomnness requires the aforementioned indistinguishability guarantee to hold only when the PRF is fed with uniformly random inputs. More formally, let $F^{\$}(k,\cdot)$ be a *randomized* oracle that responds to queries by sampling $x \leftarrow \mathcal{X}$ and outputting $(x, F(k,x))$. A weak PRF (wPRF) family is an efficiently computable function family $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ (where K, $\mathcal{X}$ and $\mathcal{Y}$ are indexed by the security parameter λ) such that for all PPT adversaries $\mathcal{A}$ we have

$$\left|\Pr[\mathcal{A}^{F^{\$}(k,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(1^\lambda) = 1]\right| \leq \mathrm{negl}(\lambda),$$

where $k \leftarrow \mathcal{K}$ and $f : \mathcal{X} \to \mathcal{Y}$ is a (truly) random function.

The Leftover Hash Lemma. Let $(\mathcal{X}, \oplus)$ be a finite group of size $|\mathcal{X}|$, and let $n > 3\log|\mathcal{X}|$. For any fixed $2n$-vector of group elements $\mathbf{x} = \{x_{j,b}\}_{j\in[n],b\in\{0,1\}} \in \mathcal{X}^{2n}$, denote by $\mathcal{S}_{\mathbf{x}}$ the following distribution:

$$\mathcal{S}_{\mathbf{x}} = \left\{ \bigoplus_{j\in[n]} x_{j,r_j} : (r_1, \cdots, r_n) \leftarrow \{0,1\}^n \right\}$$

Also, let $\mathcal{U}_{\mathcal{X}}$ denote the uniform distribution over $\mathcal{X}$, and let $\mathsf{SD}(\mathcal{D}_1, \mathcal{D}_2)$ denote the statistical distance between the distributions $\mathcal{D}_1$ and $\mathcal{D}_2$. We will use the following special case of leftover hash lemma [IZ89]. The proof is almost identical to the proof of Claim 5.3 in [Reg05].[1]

Lemma 2.1. (Leftover Hash Lemma.) *For any finite group $(\mathcal{X}, \oplus)$ and $n > 3\log|\mathcal{X}|$, for all but at most a $\left(1/\sqrt{|\mathcal{X}|}\right)$-fraction of vectors $\mathbf{x} \in \mathcal{X}^{2n}$, it holds that the distribution $\mathcal{S}_{\mathbf{x}}$ is statistically $\left(1/\sqrt{|\mathcal{X}|}\right)$-close to the uniform distribution over $\mathcal{X}$. In other words, the following holds:*

$$\Pr_{\mathbf{x}\in\mathcal{X}^{2n}}\left[\mathsf{SD}(\mathcal{S}_{\mathbf{x}}, \mathcal{U}_{\mathcal{X}}) > 1/\sqrt{|\mathcal{X}|}\right] \leq 1/\sqrt{|\mathcal{X}|}$$

3 A Framework Based on Homomorphic Keyed Functions

3.1 Building Blocks

Definition 3.1. (Input-Homomorphic Weak UF.) We call a family of functions $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ an IHwUF family if the following conditions hold:

1. $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is a *weak UF* family.
2. $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are both efficiently samplable groups.
3. The group operations $\oplus$ and $\otimes$, and inverse operation in each group are efficiently computable.
4. For every $k \in \mathcal{K}$ and for every $x_1, x_2 \in \mathcal{X}$, we have

$$F(k, x_1 \oplus x_2) = F(k, x_1) \otimes F(k, x_2).$$

Definition 3.2. (Input-Homomorphic Weak PRF.) We call a function family $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ an IHwPRF family if it satisfies the aforementioned requirements and additionally, F is a *weak PRF* family.

We also consider a notion of *bounded* homomorphism, in the sense that input-homomorphism is preserved only for an apriori bounded number of group operations in the source group of the UF/PRF. We formally describe this notion as γ-bounded homomorphism, where the parameter γ reflects the maximum number of group operations that the homomorphism can tolerate.

[1] Notice that the proof of Claim 5.3 in [Reg05] also works for *non-abelian* groups if the "summation" of group elements is always done in a *fixed* order.

Definition 3.3. (γ-Bounded Input-Homomorphic Weak UF.) We call a function family $\{F(k,\cdot):\mathcal{X}\to\mathcal{Y}\}_{k\in\mathcal{K}}$ a γ-bounded IHwUF family if there exists a universal mapping $\mathcal{R}:\mathcal{Y}\to\mathcal{Z}$ such that:

1. $\{F(k,\cdot):\mathcal{X}\to\mathcal{Y}\}_{k\in\mathcal{K}}$ and $\{\mathcal{R}(F(k,\cdot)):\mathcal{X}\to\mathcal{Z}\}_{k\in\mathcal{K}}$ are weak UF families.
2. $(\mathcal{X},\oplus)$ and $(\mathcal{Y},\otimes)$ are both efficiently samplable groups.
3. The group operations $\oplus$ and $\otimes$, and inverse operations in each group are efficiently computable.
4. For every L-length input vector $(x_1,\ldots,x_L)\in\mathcal{X}^L$, we have:

$$\mathcal{R}\Big(F\Big(k,\bigoplus_{j\in[L]}x_j\Big)\Big)=\mathcal{R}\Big(\bigotimes_{j\in[L]}F(k,x_j)\Big)$$

   subject to the restriction that $L\le\gamma$.

Definition 3.4. (γ-Bounded Input-Homomorphic Weak PRF.) A function family $\{F(k,\cdot):\mathcal{X}\to\mathcal{Y}\}_{k\in\mathcal{K}}$ is called a γ-bounded IHwPRF family if it satisfies the aforementioned requirements and additionally, F is a *weak PRF* family.

We state two useful lemmas that are used in the security proofs of cryptographic primitives constructed from (γ-Bounded) IHwUFs/IHwPRFs.The proofs of these lemmas are a straightforward application of leftover hash lemma.

Lemma 3.5. *Let $F:\mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ be an* IHwUF*, and let $n>3\log|\mathcal{X}|$. Let $\mathbf{r}=(r_1,\ldots,r_n)\leftarrow\{0,1\}^n$ and $k\leftarrow\mathcal{K}$ be sampled randomly. Given $2n$ pairs $\{(x_{j,b},F(k,x_{j,b}))\}_{j\in[n],b\in\{0,1\}}$ where $x_{j,b}\leftarrow\mathcal{X}$ and a challenge $x^*=\bigoplus_{j\in[n]}x_{j,r_j}$, no PPT adversary can predict $F(k,x^*)$ with non-negligible probability.*

Lemma 3.6. *Let $F:\mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ be an* IHwPRF*, and let $n>3\log|\mathcal{X}|$. Let $\mathbf{r}=(r_1,\ldots,r_n)\leftarrow\{0,1\}^n$ and $k\leftarrow\mathcal{K}$ be sampled randomly. Given $2n$ pairs $\{(x_{j,b},F(k,x_{j,b}))\}_{j\in[n],b\in\{0,1\}}$ where $x_{j,b}\leftarrow\mathcal{X}$ and a challenge $x^*=\bigoplus_{j\in[n]}x_{j,r_j}$, no PPT adversary can distinguish between $F(k,x^*)$ and a random element $y\leftarrow\mathcal{Y}$ with non-negligible probability.*

It is easy to see that the aforementioned lemmas hold equivalently for γ-bounded IHwUFs/PRFs if $\gamma\ge n$.

3.2 A Family of Collision-Resistant One-Way Functions

The starting point of the PKC framework is a family of CR-OWFs from IHwUFs (and hence, IHwPRFs). Informally, given an IHwUF $F:\mathcal{K}\times\mathcal{X}\to\mathcal{Y}$, subset-sum on uniformly random vectors $\mathbf{x}\in\mathcal{X}^n$ is both one-way and collision-resistant when n is sufficiently large. This family has a few interesting features. Firstly, the evaluation procedure involves n homomorphic operations, where n is apriori bounded, meaning that the family is equivalently implied by any γ-bounded IHwUF for $\gamma\ge n$. Secondly, evaluation correctness, one-wayness and collision-resistance hold even if the input and output group of the IHwUF are *non-abelian*. Thirdly, the resulting function family does *not* use the key space of F explicitly. As we will see later, this feature will be helpful to construct asymmetric cryptographic primitives. Notice that given any weak unpredictable function (not necessarily input homomorphic) $\widetilde{F}:\widetilde{\mathcal{K}}\times\widetilde{\mathcal{X}}\to\widetilde{\mathcal{Y}}$, one can easily define a one-way function $f_{\tilde{x}}:\widetilde{\mathcal{K}}\to\widetilde{\mathcal{Y}}$ as $f_{\tilde{x}}(\tilde{k}):=\widetilde{F}(\tilde{k},\tilde{x})$ for some *randomly chosen* $\tilde{x}\in\widetilde{\mathcal{X}}$.

Construction. Let $F:\mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ be an IHwUF. Fix $n>3\log|\mathcal{X}|$ and sample $2n$ uniformly random group elements from $\mathcal{X}$ as:

$$\mathbf{x}=\{x_{j,b}\leftarrow\mathcal{X}\}_{j\in[n],b\in\{0,1\}}.$$

Define the family of functions $\mathrm{OWF}_{\mathbf{x}}:\{0,1\}^n\to\mathcal{X}$ as

$$\mathrm{OWF}_{\mathbf{x}}(\mathbf{r}=(r_1,\cdots,r_n))=\bigoplus_{j\in[n]}x_{j,r_j}.$$

One-Wayness. We first prove the one-wayness of the aforementioned family of functions. Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an IHwUF. For a fixed $n > 3\log|\mathcal{X}|$, define the experiment $\mathsf{Expt}^{\mathrm{OWF-IHwUF}}$ as in Figure 7.

1. The challenger uniformly samples group elements $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ and a binary string $\mathbf{r} = (r_1, \cdots, r_n) \leftarrow \{0,1\}^n$.
2. The challenger computes $x^* = \bigoplus_{j\in[n]} x_{j,r_j}$ and sends the following tuple to the adversary $\mathcal{A}$:
$$\left(\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, x^*\right)$$
3. Eventually, the adversary $\mathcal{A}$ outputs a bit-string $\mathbf{r}' = (r'_1, \cdots, r'_n) \in \{0,1\}^n$.

Figure 7: Experiment for OWF-IHwUF security.

For any PPT adversary $\mathcal{A}$ we define $\mathbf{Adv}^{\mathrm{OWF-IHwUF}}(\mathcal{A})$ to be the probability of $x^* = \bigoplus_{j\in[n]} x_{j,r'_j}$.

Lemma 3.7. *For all PPT adversaries $\mathcal{A}$, we have $\mathbf{Adv}^{\mathrm{OWF-IHwUF}}(\mathcal{A}) = \mathrm{negl}(\lambda)$.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ such that $\mathbf{Adv}^{\mathrm{OWF-IHwUF}}(\mathcal{A})$ is non-negligible. We construct a PPT algorithm $\mathcal{B}$ such that $\mathcal{B}$ breaks the weak unpredictability of F. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ queries its oracle $2n$ times and receives $\{x_{j,b}, y_{j,b} = F(k, x_{j,b})\}_{j\in[n],b\in\{0,1\}}$. The algorithm $\mathcal{B}$ also gets a (uniformly random) challenge query $x^* \in \mathcal{X}$.
2. $\mathcal{B}$ forwards the tuple $\left(\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, x^*\right)$ to the adversary $\mathcal{A}$.
3. Eventually, $\mathcal{A}$ outputs a bit-string $\mathbf{r} = (r_1, \cdots, r_n) \in \{0,1\}^n$.
4. If $x^* = \bigoplus_{j\in[n]} x_{j,r_j}$, $\mathcal{B}$ outputs $y^* = \bigotimes_{j\in[n]} y_{j,r_j}$. Otherwise $\mathcal{B}$ outputs a uniformly random $y^* \leftarrow \mathcal{Y}$.

By the leftover hash lemma, given a uniformly random vector $\mathbf{x} \in \mathcal{X}^{2n}$, a uniform element $x^* \in \mathcal{X}$, and a uniform binary string $\mathbf{r} \in \{0,1\}^n$ we know that $(\mathbf{x}, \bigoplus_{j\in[n]} x_{j,r_j})$ is statistically indistinguishable from $(\mathbf{x}, x^*)$. Hence $\mathcal{B}$ correctly simulates the one-wayness game for $\mathcal{A}$. By input-homomorphism of F, we have

$$F(k, x^*) = F\left(k, \bigoplus_{j\in[n]} x_{j,r_j}\right) = \bigotimes_{j\in[n]} y_{j,r_j} = y^*.$$

It follows that $\mathbf{Adv}^{\mathrm{IHwUF}}(\mathcal{B})$ is negligibly different from $\mathbf{Adv}^{\mathrm{OWF-IHwUF}}(\mathcal{A})$, as desired.

Collision-Resistance. We prove that the aforementioned OWF family is also collision-resistant. Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an IHwUF. For a fixed $n > 3\log|\mathcal{X}|$, define the experiment $\mathsf{Expt}^{\mathrm{CRHF-IHwUF}}$ as in Figure 8.

1. The challenger uniformly samples group elements $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ sends them to the adversary $\mathcal{A}$.
2. $\mathcal{A}$ outputs $\mathbf{r} = (r'_1, \cdots, r'_n) \in \{0,1\}^n$ and $\mathbf{r}' = (r'_1, \cdots, r'_n) \in \{0,1\}^n$.

Figure 8: Experiment for CRHF-IHwUF security.

For any PPT adversary $\mathcal{A}$ we define $\mathbf{Adv}^{\text{CRHF}-\text{IHwUF}}(\mathcal{A})$ to be the probability of the event that

$$\bigoplus_{j\in[n]} x_{j,r_j} = \bigoplus_{j\in[n]} x_{j,r'_j}.$$

Lemma 3.8. *For all PPT adversaries $\mathcal{A}$, we have $\mathbf{Adv}^{\text{CRHF}-\text{IHwUF}}(\mathcal{A}) = \text{negl}(\lambda)$.*

Proof. Assume the existence of a PPT adversary $\mathcal{A}$ such that $\mathbf{Adv}^{\text{CRHF}-\text{IHwUF}}(\mathcal{A})$ is non-negligible. We construct a PPT algorithm $\mathcal{B}$ with the same advantage in breaking the weak unpredictability of F. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ queries its oracle $2n$ times and receives $\{x_{j,b}, y_{j,b} = F(k, x_{j,b})\}_{j\in[n],b\in\{0,1\}}$. The algorithm $\mathcal{B}$ also gets a (uniformly random) challenge query $x^* \in \mathcal{X}$.
2. $\mathcal{B}$ uniformly randomly picks $i \leftarrow [n]$ and $b^* \leftarrow \{0,1\}$, and sets $x_{i,b^*} := x^*$. It then forwards $\{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ to the adversary $\mathcal{A}$.
3. $\mathcal{A}$ outputs $\mathbf{r} = (r_1, \cdots, r_n) \in \{0,1\}^n$ and $\mathbf{r}' = (r'_1, \cdots, r'_n) \in \{0,1\}^n$.
4. $\mathcal{B}$ proceeds as follows:
   - If $\bigoplus_{j\in[n]} x_{j,r_j} \neq \bigoplus_{j\in[n]} x_{j,r'_j}$ or $r_i = r'_i$, $\mathcal{B}$ outputs a random $y^* \leftarrow \mathcal{Y}$.
   - Otherwise, assume wlog that $r_i = b^*$. Then, the following must hold:

     $$x^* = \left(\bigoplus_{j\in[i-1]} x_{j,r_j}\right)^{-1} \oplus \left(\bigoplus_{j\in[n]} x_{j,r'_j}\right) \oplus \left(\bigoplus_{j\in[i+1,n]} x_{j,r_j}\right)^{-1},$$

     where the right-hand side is independent of x^*. $\mathcal{B}$ now outputs y^* as

     $$y^* = \left(\bigotimes_{j\in[i-1]} y_{j,r_j}\right)^{-1} \otimes \left(\bigotimes_{j\in[n]} y_{j,r'_j}\right) \otimes \left(\bigotimes_{j\in[i+1,n]} y_{j,r_j}\right)^{-1}.$$

     By the input-homomorphism of F, we have $F(k, x^*) = y^*$.

Observe that if $\mathcal{A}$ outputs a valid collision $(\mathbf{r}, \mathbf{r}')$, the probability that $\mathbf{r}$ and $\mathbf{r}'$ differ in the ith bit for a randomly chosen $i \leftarrow [n]$ is at least $1/n$. It follows that

$$\mathbf{Adv}^{\text{IHwUF}}(\mathcal{B}) \geq \left(\frac{\mathbf{Adv}^{\text{CRHF}-\text{IHwUF}}(\mathcal{A})}{n}\right),$$

which is non-negligible, as desired.

Note 3.9. The aforementioned OWF/CRHF family may be analogously instantiated using a γ-bounded IHwUF family, subject to the restriction that $n \leq \gamma$. The proofs of one-wayness and collision resistance also follow similarly.

3.3 The Framework

In this section, we present a framework for designing public-key cryptographic primitives from a combination of IHwUF (or IHwPRF) operations and "subset-sums" over group elements. The protocol consists of four phases — initialization, pre-evaluation, evaluation and post-evaluation. The evaluation phase is based on UF (or PRF) operations, while the pre-evaluation and post-evaluation phases are based on "subset-sum" operations over group elements. For the sake of clarity, we exemplify each phase of the framework using a "conceptually" equivalent framework in the discrete-log based group-theoretic setting. More specifically, we show how our framework subsumes a generic methodology of constructing cryptographic primitives from the CDH/DDH assumptions over prime order groups.

1. **Initialization:**

   - Publish $\left(\mathcal{G}, g, N^*, \bar{N}, \widehat{N}\right)$, where $\mathcal{G}$ describes a cyclic group $(\mathbb{G}, \cdot)$ of prime order p with uniform generator g, and $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$ are fixed functions.
   - Let $\mathcal{F}$ be the description of an IHwUF/IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ over groups $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$. Fix $n > 3\log(|\mathcal{X}|)$, $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, and publish $\left(\mathcal{F}, \mathrm{X}, n, N^*, \bar{N}, \widehat{N}\right)$, where X is a tuple of $2n$ uniform "base elements" from $\mathcal{X}$ as
   $$\mathrm{X} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n], b\in\{0,1\}}\,.$$

2. **Pre-Evaluation:**

   - Given the generator g, sample N^* random elements from $\mathbb{Z}_q$ as
   $$\{\alpha_{n^*} \leftarrow \mathbb{Z}_q\}_{n^*\in[N^*]}\,,$$
   and output the tuple
   $$\mathrm{A} = \{g_{n^*} = g^{\alpha_{n^*}}\}_{n^*\in[N^*]}\,.$$
   - Given the tuple of "base elements" X, sample N^* random strings as
   $$\{\mathbf{s}_{n^*} = (s_{n^*,1} \ldots, s_{n^*,n}) \leftarrow \{0,1\}^n\}_{n^*\in[N^*]}\,,$$
   and output the tuple
   $$\mathrm{X}^* = \left\{x^*_{n^*} = \bigoplus_{j=1}^{n} x_{j,s_{n^*,j}}\right\}_{n^*\in[N^*]}\,.$$

3. **Evaluation:**

   - Given the generator g and the pre-evaluation output A, sample $\bar{N}$ elements $\beta_1, \ldots, \beta_{\bar{N}} \leftarrow \mathbb{Z}_q$ and output B, B^* where
   $$\mathrm{B} = \left\{h_{\bar{n}} = g^{\beta_{\bar{n}}}\right\}_{\bar{n}\in[\bar{N}]}\,, \quad \mathrm{B}^* = \left\{h_{\bar{n},n^*} = (g^*_{n^*})^{\beta_{\bar{n}}}\right\}_{\bar{n}\in[\bar{N}], n^*\in[N^*]}\,.$$

- Given "base elements" X and the pre-evaluation output X^*, sample $\bar{N}$ keys $k_1, \ldots, k_{\bar{N}} \leftarrow \mathcal{K}$ and output the tuple $(\mathrm{Y}, \mathrm{Y}^*)$, where

$$\begin{aligned} \mathrm{Y} &= \left\{ y_{j,b}^{(\bar{n})} = F\left(k_{\bar{n}}, x_{j,b}\right) \right\}_{\bar{n}\in[\bar{N}], j\in[n], b\in\{0,1\}}, \\ \mathrm{Y}^* &= \left\{ y^*_{\bar{n},n^*} = F\left(k_{\bar{n}}, x^*_{n^*}\right) \right\}_{\bar{n}\in[\bar{N}], n^*\in[N^*]}. \end{aligned}$$

4. **Post-Evaluation:**

- Given the generator g and the evaluation outputs $(h_1, \ldots, h_{\bar{N}})$, sample $\widehat{N}$ random elements from $\mathbb{Z}_q$ as

$$\{\gamma_{\hat{n}} \leftarrow \mathbb{Z}_q\}_{\hat{n}\in[\widehat{N}]},$$

and output

$$\mathrm{C} = \left\{ \hat{g}_{\hat{n}} = g^{\gamma_{\hat{n}}}, \hat{h}_{\bar{n},\hat{n}} = h_{\bar{n}}^{\gamma_{\hat{n}}} \right\}_{\bar{n}\in[\bar{N}], \hat{n}\in[\widehat{N}]}.$$

- Given X and the evaluation output $\mathrm{Y} = \left\{ y_{j,b}^{(1)}, \ldots, y_{j,b}^{(\bar{N})} \right\}_{j\in[n], b\in\{0,1\}}$, sample $\widehat{N}$ random binary vectors as

$$\{\mathbf{r}_{\hat{n}} = (r_{\hat{n},1} \ldots, r_{\hat{n},n}) \leftarrow \{0,1\}^n\}_{\hat{n}\in[\widehat{N}]},$$

and output $(\hat{\mathrm{X}}, \hat{\mathrm{Y}})$, where

$$\hat{\mathrm{X}} = \left\{ \hat{x}_{\hat{n}} = \bigoplus_{j=1}^{n} x_{j,r_{\hat{n},j}} \right\}_{\hat{n}\in[\widehat{N}]}, \quad \hat{\mathrm{Y}} = \left\{ \hat{y}_{\bar{n},\hat{n}} = \bigotimes_{j=1}^{n} y_{j,r_{\hat{n},j}}^{(\bar{n})} \right\}_{\bar{n}\in[\bar{N}], \hat{n}\in[\widehat{N}]}.$$

Functionality. The following claim is a direct consequence of the input-homomorphism of the function F.

Claim 3.10. *For each $n^* \in [N^*], \hat{n} \in [\widehat{N}], \bar{n} \in [\bar{N}]$, we have:*

$$\begin{aligned} y^*_{\bar{n},n^*} &= F\left(k_{\bar{n}}, x^*_{n^*}\right) = \bigotimes_{j\in[n]} y^{(\bar{n})}_{j,s_{n^*,j}} \\ \hat{y}_{\bar{n},\hat{n}} &= \bigotimes_{j\in[n]} y^{(\bar{n})}_{j,s_{\hat{n},j}} = F\left(k_{\bar{n}}, \hat{x}_{\hat{n}}\right) \end{aligned}$$

Remark 3.11. If F is a γ-bounded IHwUF/IHwPRF for $\gamma > n$, the relations above are modified as follows:

$$\begin{aligned} \mathcal{R}\left(y^*_{\bar{n},n^*}\right) &= \mathcal{R}\left(F\left(k_{\bar{n}}, x^*_{n^*}\right)\right) = \mathcal{R}\left(\bigotimes_{j\in[n]} y^{(\bar{n})}_{j,s_{n^*,j}} \right) \\ \mathcal{R}\left(\hat{y}_{\bar{n},\hat{n}}\right) &= \mathcal{R}\left(\bigotimes_{j\in[n]} y^{(\bar{n})}_{j,s_{\hat{n},j}} \right) = \mathcal{R}\left(F\left(k_{\bar{n}}, \hat{x}_{\hat{n}}\right)\right) \end{aligned}$$

Security (IHwPRF). The following theorem captures the security guarantees provided by the general framework when instantiated using an IHwPRF.

Theorem 3.12. *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an* IHwPRF*. Then for any $n > 3\log(|\mathcal{X}|)$, for all functions $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, and for any PPT adversary $\mathcal{A}$, it holds that*

$$\left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}, \mathrm{Y}^*, \hat{\mathrm{Y}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{U}_{\mathrm{X}}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] \right|$$

is negligible in the security parameter λ, where:

- *The tuple $(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}, \mathrm{Y}^*, \hat{\mathrm{Y}})$ is as defined in the protocol above.*
- *$\mathrm{U}_{\mathrm{X}}, \mathrm{U}_{\mathrm{X}^*}$ and $\mathrm{U}_{\hat{\mathrm{X}}}$ are tuples of $2n, N^*$ and $\widehat{N}$ uniformly sampled group elements from $\mathcal{X}$.*
- *U_{Y}, $\mathrm{U}_{\mathrm{Y}^*}$ and $\mathrm{U}_{\hat{\mathrm{Y}}}$ are tuples of $(2n \cdot \bar{N})$, $(N^* \cdot \bar{N})$ and $(\widehat{N} \cdot \bar{N})$ uniform group elements from $\mathcal{Y}$.*

Proof. The proof proceeds in two stages. The first stage applies the leftover hash lemma, while the second stage relies on the pseudorandomness of F.

Applying LHL. Let $\mathrm{U}_{\mathrm{X}^*} = \{u_1^*, \ldots, u_{N^*}^*\}$ be a tuple of N^* uniformly sampled group elements in $\mathcal{X}$, and let

$$\widetilde{\mathrm{U}}_{\mathrm{Y}^*} = \left\{y_{1,n^*}^*, \ldots, y_{\bar{N},n^*}^*\right\}_{n^* \in [N^*]},$$

where $y_{\bar{n},n^*}^* = F(k_{\bar{n}}, u_{n^*}^*)$ for each $\bar{n} \in [\bar{N}], n^* \in [N^*]$, and $k_1, \ldots, k_{\bar{N}} \in \mathcal{K}$ are the *same* PRF keys as used in the "real" protocol. We use the following claim.

Lemma 3.13. *For any $n > 3\log(|\mathcal{X}|)$, for all functions $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, and for any PPT adversary $\mathcal{A}$, we have*

$$\left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}, \mathrm{Y}^*, \hat{\mathrm{Y}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \hat{\mathrm{X}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \hat{\mathrm{Y}}\right) = 1\right] \right|$$

is negligible in the security parameter λ. The proof of this lemma follows from Lemma 3.6. Next, let $\mathrm{U}_{\hat{\mathrm{X}}} = \left\{\hat{u}_1, \ldots, \hat{u}_{\widehat{N}}\right\}$ be a tuple of $\widehat{N}$ uniformly sampled group elements in $\mathcal{X}$, and let

$$\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}} = \left\{\hat{y}_{1,n^*}, \ldots, \hat{y}_{\bar{N},\hat{n}}\right\}_{\hat{n} \in [\widehat{N}]},$$

where $\hat{y}_{\bar{n},n^*} = F(k_{\bar{n}}, \hat{u}_{n^*})$ for each $\bar{n} \in [\bar{N}]$ and each $\hat{n} \in [\widehat{N}]$, and $k_1, \ldots, k_{\bar{N}} \in \mathcal{K}$ are the *same* PRF keys as used in the "real" protocol. We also use the following claim.

Lemma 3.14. *For any $n > 3\log(|\mathcal{X}|)$, for all functions $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, and for any PPT adversary $\mathcal{A}$, we have*

$$\left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \hat{\mathrm{X}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \hat{\mathrm{Y}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}\right) = 1\right] \right|$$

is negligible in the security parameter λ. The proof of this lemma again follows from Lemma 3.6.

Pseudorandomness. We now use the pseudorandomness of F to prove the following lemma.

Lemma 3.15. *For any $n > 3\log(|\mathcal{X}|)$, for all functions $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, and for any PPT adversary $\mathcal{A}$, we have*

$$\left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] \right|$$

is negligible in the security parameter λ, where U_{Y}, $\mathrm{U}_{\mathrm{Y}^*}$ and $\mathrm{U}_{\hat{\mathrm{Y}}}$ denote tuples of $(2n \cdot \bar{N})$, $(N^* \cdot \bar{N})$ and $(\widehat{N} \cdot \bar{N})$ uniform elements in $\mathcal{Y}$, respectively, while $\widetilde{\mathrm{U}}_{\mathrm{Y}^*}$ and $\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}$ are as described in Lemmas 3.13 and 3.14, respectively. We prove this lemma through a sequence of hybrid arguments. Define the collection $\{\mathrm{Y}^{(\bar{n})}\}_{m \in [0,\bar{N}]}$ as follows.

- $\mathrm{Y}^{(0)}$ is identical to Y as output by the "real world" protocol.
- $\mathrm{Y}^{(\bar{N})}$ is identical to U_{Y}.
- For each $\bar{n} \in [\bar{N}]$, $\mathrm{Y}^{(\bar{n})}$ is identical to $\mathrm{Y}^{(\bar{n}-1)}$ except that the sub-tuple of elements $\{y_{j,b}^{(\bar{n})}\}_{j \in [n], b \in \{0,1\}}$ is replaced by $2n$ elements sampled uniformly at random from $\mathcal{Y}$.

Similarly, define the collection $\{\widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n})}\}_{m \in [0,\bar{N}]}$ as follows.

- $\widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(0)}$ is identical to $\widetilde{\mathrm{U}}_{\mathrm{Y}^*}$ as described in Lemma 3.13.
- $\widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{N})}$ is identical to $\mathrm{U}_{\mathrm{Y}^*}$.
- For each $\bar{n} \in [\bar{N}]$, $\widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n})}$ is identical to $\widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n}-1)}$ except that the sub-tuple of elements $\{y_{\bar{n},n^*}^*\}_{n^* \in [N^*]}$ is replaced by N^* elements sampled uniformly at random from $\mathcal{Y}$.

Finally, define the collection $\{\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n})}\}_{m \in [0,\bar{N}]}$ as follows.

- $\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(0)}$ is identical to $\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}$ as described in Lemma 3.14.
- $\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{N})}$ is identical to $\mathrm{U}_{\hat{\mathrm{Y}}}$.
- For each $\bar{n} \in [\bar{N}]$, $\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n})}$ is identical to $\widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n}-1)}$ except that the sub-tuple of elements $\{\hat{y}_{\bar{n},\hat{n}}\}_{\hat{n} \in [\widehat{N}]}$ is replaced by $\widehat{N}$ elements sampled uniformly at random from $\mathcal{Y}$.

We first prove the following auxiliary lemma.

Lemma 3.16. *For any $n > 3\log(|\mathcal{X}|)$, for all functions $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, and for any PPT adversary $\mathcal{A}$,*

$$\left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}^{(\bar{n}-1)}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n}-1)}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n}-1)}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}^{(\bar{n})}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n})}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n})}\right) = 1\right] \right|$$

is negligible in the security parameter λ.

To prove this lemma, suppose that there exists a PPT adversary $\mathcal{A}$ such that

$$\left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}^{(\bar{n}-1)}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n}-1)}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n}-1)}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}^{(\bar{n})}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}^{(\bar{n})}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}^{(\bar{n})}\right) = 1\right] \right|$$

is non-negligible for some $\bar{n} \in [\bar{N}]$. We construct a PPT algorithm $\mathcal{B}$ that breaks the weak pseudorandomness of F. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ queries its oracle $(2n + N^* + \widehat{N})$ times and receives the following tuples

$$\left\{ \{x_{j,b}, y_{j,b}\}_{j\in[n],b\in\{0,1\}}, \{x^*_{n^*}, y^*_{n^*}\}_{n^*\in[N^*]}, \{\hat{x}_{\hat{n}}, \hat{y}_{\hat{n}}\}_{\hat{n}\in[N^*]} \right\}.$$

2. It then samples $(\bar{N} - \bar{n})$ PRF keys as $k_{\bar{n}+1}, \ldots, k_{\bar{N}} \leftarrow \mathcal{K}$ and sets the following for each $n'_k \in [\bar{N}]$:

$$y^{(n'_k)}_{j,b} = \begin{cases} y_{j,b} & \text{if } n'_k = \bar{n} \\ F(k_{n'_k}, x_{j,b}) & \text{if } n'_k > \bar{n} \\ y \leftarrow \mathcal{Y} & \text{otherwise.} \end{cases} \quad \text{for } j \in [n], b \in \{0,1\}$$

$$y^*_{n'_k,n^*} = \begin{cases} y^*_{n^*} & \text{if } n'_k = \bar{n} \\ F(k_{n'_k}, x^*_{n^*}) & \text{if } n'_k > \bar{n} \\ y \leftarrow \mathcal{Y} & \text{otherwise.} \end{cases} \quad \text{for } n^* \in [N^*]$$

$$\hat{y}_{n'_k,\hat{n}} = \begin{cases} \hat{y}_{\hat{n}} & \text{if } n'_k = \bar{n} \\ F(k_{n'_k}, \hat{x}_{\hat{n}}) & \text{if } n'_k > \bar{n} \\ y \leftarrow \mathcal{Y} & \text{otherwise.} \end{cases} \quad \text{for } \hat{n} \in [\widehat{N}]$$

3. $\mathcal{B}$ then sets the following:

$$\mathrm{X} = \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, \quad \mathrm{U}_{\mathrm{X}^*} = \{x^*_{n^*}\}_{n^*\in[N^*]}, \quad \mathrm{U}_{\hat{\mathrm{X}}} = \{\hat{x}_{\hat{n}}\}_{\hat{n}\in[N^*]}$$

$$\mathrm{Y}' = \left\{ y^{(1)}_{j,b}, \ldots, y^{(\bar{N})}_{j,b} \right\}_{j\in[n],b\in\{0,1\}}$$

$$\widetilde{\mathrm{U}}'_{\mathrm{Y}^*} = \left\{ y^*_{1,n^*}, \ldots, y^*_{\bar{N},n^*} \right\}_{n^*\in[N^*]}, \quad \widetilde{\mathrm{U}}'_{\hat{\mathrm{Y}}} = \left\{ \hat{y}_{1,\hat{n}}, \ldots, \hat{y}_{\bar{N},\hat{n}} \right\}_{\hat{n}\in[N^*]}$$

and sends the tuple $\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}', \widetilde{\mathrm{U}}'_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}'_{\hat{\mathrm{Y}}}\right)$ to $\mathcal{A}$.

4. Eventually, $\mathcal{A}$ outputs a bit b. $\mathcal{B}$ outputs the same bit b.

Observe the following:

- When $\mathcal{B}$ interacts with the "real" PRF oracle, the distribution of $\left(\mathrm{Y}', \widetilde{\mathrm{U}}'_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}'_{\hat{\mathrm{Y}}}\right)$ is identical to the distribution of $\left(\mathrm{Y}^{(\bar{n}-1)}, \widetilde{\mathrm{U}}^{(\bar{n}-1)}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}^{(\bar{n}-1)}_{\hat{\mathrm{Y}}}\right)$.
- When $\mathcal{B}$ interacts with a "random" oracle, the distribution of $\left(\mathrm{Y}', \widetilde{\mathrm{U}}'_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}'_{\hat{\mathrm{Y}}}\right)$ is identical to the distribution of $\left(\mathrm{Y}^{(\bar{n})}, \widetilde{\mathrm{U}}^{(\bar{n})}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}^{(\bar{n})}_{\hat{\mathrm{Y}}}\right)$.

It follows that $\mathbf{Adv}^{\mathrm{IHwUF}}(\mathcal{B})$ is negligibly different from the advantage of $\mathcal{A}$, which completes the proof of Lemma 3.16.
To prove Lemma 3.15, observe that

$$\begin{aligned} & \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] \right| \\ \leq & \sum_{m=1}^{\bar{N}} \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}^{(\bar{n}-1)}, \widetilde{\mathrm{U}}^{(\bar{n}-1)}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}^{(\bar{n}-1)}_{\hat{\mathrm{Y}}}\right) = 1\right] - \right. \\ & \left. \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}^{(\bar{n})}, \widetilde{\mathrm{U}}^{(\bar{n})}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}^{(\bar{n})}_{\hat{\mathrm{Y}}}\right) = 1\right] \right| \leq \mathrm{negl}(\lambda) \end{aligned}$$

This completes the proof of Lemma 3.15.

Putting Everything Together. Using Lemmas 3.13, 3.14, and 3.15 we have

$$\begin{aligned}
& \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}, \mathrm{Y}^*, \hat{\mathrm{Y}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{U}_{\mathrm{X}}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] \right| \\
\leq & \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}, \mathrm{Y}^*, \hat{\mathrm{Y}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \hat{\mathrm{X}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \hat{\mathrm{Y}}\right) = 1\right] \right| + \\
& \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \hat{\mathrm{X}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \hat{\mathrm{Y}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}\right) = 1\right] \right| + \\
& \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{Y}, \widetilde{\mathrm{U}}_{\mathrm{Y}^*}, \widetilde{\mathrm{U}}_{\hat{\mathrm{Y}}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] \right| + \\
& \left| \Pr\left[\mathcal{A}\left(\mathrm{X}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathrm{U}_{\mathrm{X}}, \mathrm{U}_{\mathrm{X}^*}, \mathrm{U}_{\hat{\mathrm{X}}}, \mathrm{U}_{\mathrm{Y}}, \mathrm{U}_{\mathrm{Y}^*}, \mathrm{U}_{\hat{\mathrm{Y}}}\right) = 1\right] \right| \\
\leq & \ \mathrm{negl}(\lambda)
\end{aligned}$$

since X and U_{X} are both distributed uniformly over $\mathcal{X}^{2n}$. This completes the proof of Theorem 3.12.

Security (IHwUF). The following theorem captures the security guarantees provided by the general framework when instantiated using an IHwUF.

Theorem 3.17. *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an* IHwUF. *Then for any $n > 3\log(|\mathcal{X}|)$, for all functions $N^* = N^*(\lambda)$, $\bar{N} = \bar{N}(\lambda)$ and $\widehat{N} = \widehat{N}(\lambda)$, for any arbitrary $(n^*, \hat{n}, \bar{n}) \in [N^*] \times [\widehat{N}] \times [\bar{N}]$, and for any PPT adversary $\mathcal{A} = (\mathcal{A}^*, \hat{\mathcal{A}})$, we have*

$$\Pr\left[\mathcal{A}^*\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = y^*_{\bar{n}, n^*}\right] + \Pr\left[\hat{\mathcal{A}}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = \hat{y}_{\bar{n}, \hat{n}}\right] \leq \mathrm{negl}(\lambda),$$

where the tuple $(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}, y^*_{\bar{n}, n^*}, \hat{y}_{\bar{n}, \hat{n}})$ for $(n^*, \hat{n}, \bar{n}) \in [N^*] \times [\widehat{N}] \times [\bar{N}]$ is as defined in the protocol above.

Proof. Suppose that there exists a PPT adversary $\mathcal{A} = (\mathcal{A}^*, \hat{\mathcal{A}})$ such that

$$\Pr\left[\mathcal{A}^*\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = y^*_{\bar{n}, n^*}\right] + \Pr\left[\hat{\mathcal{A}}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = \hat{y}_{\bar{n}, \hat{n}}\right]$$

is non-negligible for some $(n^*, \hat{n}, \bar{n}) \in [N^*] \times [\widehat{N}] \times [\bar{N}]$.

- Assume there exists $(n^*, \bar{n}) \in [N^*] \times [\bar{N}]$ such that $\Pr\left[\mathcal{A}^*\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = y^*_{\bar{n}, n^*}\right]$ is non-negligible. We construct a PPT algorithm $\mathcal{B}$ that breaks the weak unpredictability of F. $\mathcal{B}$ proceeds as follows:
  1. $\mathcal{B}$ queries its oracle $2n$ times and receives the tuple $\{x_{j,b}, y_{j,b}\}_{j \in [n], b \in \{0,1\}}$, and a challenge $x^* \in \mathcal{X}$.
  2. It then samples $(\bar{N} - 1)$ PRF keys as $k_1, \ldots, k_{\bar{n}-1}, k_{\bar{n}+1}, \ldots, k_{\bar{N}} \leftarrow \mathcal{K}$ and sets the following for each $n'_k \in [\bar{N}]$, $j \in [n]$, $b \in \{0,1\}$:
  $$y^{(n'_k)}_{j,b} = \begin{cases} y_{j,b} & \text{if } n'_k = \bar{n} \\ F(k_{n'_k}, x_{j,b}) & \text{otherwise.} \end{cases}$$
  3. $\mathcal{B}$ then samples $(N^* - 1)$ random group elements as
  $$x^*_1, \ldots, x^*_{n^*-1}, x^*_{n^*+1}, x^*_{N^*} \leftarrow \mathcal{X}$$
  and sets $x^*_{n^*} = x^*$, where x^* is the input challenge to $\mathcal{B}$.

4. $\mathcal{B}$ additionally samples $\hat{N}$ group elements $\{\hat{x}_{\hat{n}}\}_{\hat{n}\in\hat{N}}$ and sends the tuple $\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right)$ to $\mathcal{A}$, where

$$\mathrm{X} = \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, \quad \mathrm{X}^* = (x^*_1, \ldots, x^*_{N^*}), \quad \hat{\mathrm{X}} = \left(\hat{x}_1, \ldots, \hat{x}_{\hat{N}}\right),$$

$$\mathrm{Y} = \left\{y^{(1)}_{j,b}, \ldots, y^{(\bar{N})}_{j,b}\right\}_{j\in[n],b\in\{0,1\}}.$$

5. Eventually, $\mathcal{A}$ outputs $y^* \in \mathcal{Y}$. $\mathcal{B}$ outputs the same y^*.

By the leftover hash lemma, the distributions of X^* and $\hat{\mathrm{X}}$ are statistically indistinguishable from that in the "real" protocol, while the distributions of X and Y are identical to that in the "real" protocol from the definition of the UF-oracle in the weak unpredictability experiment. It follows that $\mathbf{Adv}^{\mathrm{IHwUF}}(\mathcal{B})$ is negligibly different from the advantage of $\mathcal{A}^*$.

- A similar argument shows that if there exists $(\hat{n}, \bar{n}) \in [\hat{N}] \times [\bar{N}]$ such that $\Pr\left[\hat{\mathcal{A}}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = \hat{y}_{\bar{n},\hat{n}}\right]$ is non-negligible, then there exists an attacker against the weak unpredictability of F with non-negligible advantage.

Combining the aforementioned inferences, for any $(n^*, \hat{n}, \bar{n}) \in [N^*] \times [\widehat{N}] \times [\bar{N}]$, and for any PPT adversary $\mathcal{A} = (\mathcal{A}^*, \hat{\mathcal{A}})$, we have

$$\Pr\left[\mathcal{A}^*\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = y^*_{\bar{n},n^*}\right] + \Pr\left[\hat{\mathcal{A}}\left(\mathrm{X}, \mathrm{X}^*, \hat{\mathrm{X}}, \mathrm{Y}\right) = \hat{y}_{\bar{n},\hat{n}}\right] \leq \mathrm{negl}(\lambda),$$

which completes the proof of Theorem 3.17.

3.4 Instantiations from Cryptographic Assumptions

In this subsection, we provide instantiations of IHwUFs/IHwPRFs from concrete assumptions. We start with Diffie-Hellman assumption as an easy example. We then provide an example of IHwPRF where the input space *depends* on the secret key. After that we provide an example of IHwPRF where a bounded number of homomorphisms are allowed. Finally, we prove that any (group-)homomorphic PKE implies an IHwPRF family, providing constructions of IHwPRF family from different concrete assumptions.

CDH/DDH. Given a group $(\mathbb{G}, \cdot)$ of order q with generator g, define the function $F : \mathbb{Z}_q \times \mathbb{G} \to \mathbb{G}$ as $F(k, h) = h^k$. We prove that F is an IHwPRF assuming DDH assumption. It is easy to see that for any $h_1, h_2 \in \mathbb{G}$ we have $F(k, h_1 \cdot h_2) = F(k, h_1) \cdot F(k, h_2)$. Let $\mathcal{A}$ be an attacker against the weak pseudorandomness of F, and let n be the number of queries of the attacker. It is enough to show that

$$((g^{x_1}, g^{kx_1}), (g^{x_2}, g^{kx_2}), \ldots, (g^{x_n}, g^{kx_n})) \overset{c}{\approx} ((g^{x_1}, g^{r_1}), (g^{x_2}, g^{r_2}), \ldots, (g^{x_n}, g^{r_n})),$$

where $k, x_i, r_i \leftarrow \mathbb{Z}_q$ are uniform and independent for $i \in [n]$, and (g^{x_i}, g^{kx_i}) is the answer to the ith query.

Given a DDH-challenge tuple (g^k, g^{x^*}, y) where y is either g^{kx^*} or a random element of $\mathbb{G}$, the reduction first samples n pairs of random elements: $\{(s_i, t_i) \leftarrow \mathbb{Z}_q^2\}_{i\in[n]}$. It then outputs the following tuple:

$$\left((g^{s_1}(g^{x^*})^{t_1}, (g^k)^{s_1} y^{t_1}), \ldots, (g^{s_n}(g^{x^*})^{t_n}, (g^k)^{s_n} y^{t_n})\right).$$

Observe that when $y = g^{kx^*}$ the tuple above is identical to the "real" game where all queries answered as PRF output, and y random corresponds to the "ideal" game where all queries answered as a random function. Therefore, an attacker against the weak pseudorandomness of F with advantage ε implies a DDH distinguisher with advantage ε. A similar argument shows that F is an IHwUF under CDH assumption.

Matrix-DDH. The DDH problem has been generalized to several different algebraic problems, like Decisional Linear (DLIN [BBS04]) and k-linear [HK07, Sha07]. Escala et al [EHK+13] generalized all these assumptions into one framework called the matrix-DDH assumptions. Given a cyclic group $(\mathbb{G}, \cdot)$ of order q with generator g and a $\mathbb{Z}_q$-matrix $\mathbf{A}$, they adopted the notation $[\mathbf{A}]$ to denote the component-wise exponentiation $g^{\mathbf{A}}$. The matrix-DDH problem is parameterized by a distribution $\mathcal{D}_{l,k}$ on $\mathbb{Z}_q^{l\times k}$ matrices with $(l > k)$, where the top $k \times k$ matrix, denoted $\bar{\mathbf{A}}$, is overwhelmingly invertible. The remaining $(l-k)\times k$ bottom matrix is denoted $\underline{\mathbf{A}}$. The $\mathcal{D}_{l,k}$ matrix-DDH assumption states that with $\mathbf{A} \leftarrow \mathcal{D}_{l,k}$ and $(\mathbf{w}, \mathbf{r}) \leftarrow \mathbb{Z}_q^k \times \mathbb{Z}_q^{l-k}$:

$$\left([\bar{\mathbf{A}}\mathbf{w}], [\underline{\mathbf{A}}\mathbf{w}]\right) \overset{c}{\approx} \left([\bar{\mathbf{A}}\mathbf{w}], [\mathbf{r}]\right),$$

We now show that any $\mathcal{D}_{l,k}$-matrix-DDH assumption can give us an IHwPRF. The key sampling algorithm is as follows: first sample $\mathbf{A} \leftarrow \mathcal{D}_{l,k}$ and then compute key $\mathbf{K} = \bar{\mathbf{A}}^{-1}\underline{\mathbf{A}}$. Now we define the function $F : \mathbb{Z}_q^{k\times(l-k)} \times \mathbb{G}^k \to \mathbb{G}^{l-k}$ as $F(\mathbf{K}, [\mathbf{x}]) = \mathbf{K}[\mathbf{x}] = [\mathbf{K}\mathbf{x}]$. We prove that F is an IHwPRF assuming the $\mathcal{D}_{l,k}$-Matrix-DDH assumption. It is easy to see that for any $[\mathbf{x}_1], [\mathbf{x}_2] \in \mathbb{G}^k$ we have

$$F(\mathbf{K}, [\mathbf{x}_1 + \mathbf{x}_2]) = F(\mathbf{K}, [\mathbf{x}_1]) + F(\mathbf{K}, [\mathbf{x}_2]).$$

Let $\mathcal{A}$ be an attacker against the weak pseudorandomness of F, and let n be the number of queries of the attacker. It is enough to show that

$$(([\mathbf{x}_1], [\mathbf{K}\mathbf{x}_1]), \ldots, ([\mathbf{x}_n], [\mathbf{K}\mathbf{x}_n])) \overset{c}{\approx} (([\mathbf{x}_1], [\mathbf{r}_1]), \ldots, ([\mathbf{x}_n], [\mathbf{r}_n])),$$

where $\mathbf{x}_i, \mathbf{r}_i$ are uniform and independent for $i \in [n]$, and $[\mathbf{x}_i], [\mathbf{K}\mathbf{x}_i]$ is the answer to the i-th query.

It was shown in [EHK+13] that by the random self-reducibility of matrix-DDH samples, the distinguishing advantage of the above distributions is bounded by a multiplicative factor of $(l-k)$ over a single sample matrix-DDH advantage. Therefore, an attacker against the weak pseudorandomness of F with advantage ε implies a Matrix-DDH distinguisher with advantage $\varepsilon/(l-k)$. Instead of decisional, if we assume that F is unpredictable, then a similar argument shows that F is an IHwUF under the computational matrix-DDH assumption.

Quadratic Residuosity. Let $N = pq$ be a composite modulus where p and q are randomly generated equal-size primes, and let $\mathcal{J}_N^{+1}$ be the set of all elements in $\mathbb{Z}_N^*$ with Jacboi symbol 1. Define $F(k = (p, q), x \in \mathcal{J}_N^{+1})$ as follows:

$$F(k, x) = \begin{cases} 0 & \text{if } x \in \mathcal{QR}_N \\ 1 & \text{if } x \notin \mathcal{QR}_N \end{cases}$$

First, given the factorization of N one can efficiently determine whether an element $x \in \mathcal{J}_N^{+1}$ is a quadratic residue. Moreover, Observe that for any $x_1, x_2 \in \mathcal{J}_N^{+1}$ we have $F(k, x_1x_2) = F(k, x_1) + F(k, x_2)$ where x_1x_2 is the product of x_1 and x_2 in $\mathbb{Z}_N^*$, and $+$ is addition modulo 2.[1] A simple hybrid argument similar to the case of DDH construction implies the weak pseudorandomness of F. It follows that F is an IHwPRF under QR assumption.

Remark 3.18. We note that although F is an IHwPRF (and hence IHwUF), it has a limitation that the input space *depends* on the key. In some applications, it is necessary to know the input space *before* generating the key.

By a similar argument it is also possible to construct an IHwUF from RSA assumption. However, like the case of QR, the input space (implicitly) depends on the choice of the key.

[1] Recall that for any $x_1, x_2 \in \mathcal{J}_N^{+1}$, the product x_1x_2 is a quadratic non-residue if and only if exactly one of them is a quadratic non-residue.

LWE. We now sketch the construction of IHwPRF from LWE asumption [Reg05]. Let n, q be the parameters of the LWE assumption where n is the dimension of the secret and q is the modulus. Also, let χ denote the (Gaussian-like) noise distribution. Let GSamp be a Gaussian sampler algorithm that on a uniformly random chosen input $u \leftarrow \{0,1\}^\ell$, outputs a sample according to χ. First, we use a weak PRF $F_N : \mathcal{K}_N \times \mathbb{Z}_q^n \to \{0,1\}^\ell$ to generate the randomness for GSamp algorithm.[1] We define the bounded IHwPRF[2] $F : \mathcal{K} \times \mathbb{Z}_q^n \to \mathbb{Z}_q$ as

$$F((k,\mathbf{s}),\mathbf{a}) = \langle \mathbf{s}, \mathbf{a} \rangle + \mathsf{GSamp}(F_N(k,\mathbf{a}))$$

$$\mathcal{R}(b \in \mathbb{Z}_q) = \begin{cases} 0 & |b| \le q/4 \\ 1 & |b| > q/4 \end{cases}$$

where $\mathcal{K} = \mathcal{K}_N \times \mathbb{Z}_q^n, k \leftarrow \mathcal{K}_N$ and $\mathbf{s} \leftarrow \mathbb{Z}_q^n$. The weak pseudorandomness of F follows from a simple hybrid argument, and we omit the details here. As for bounded homomorphism, observe that if q is sufficiently large (superpolynomial in n), for any $\gamma \le q/n$ we have

$$\mathcal{R}\left(F((k,\mathbf{s}), \sum_{i\in[\gamma]} \mathbf{a}_i)\right) = \mathcal{R}\left(\sum_{i\in[\gamma]} F((k,\mathbf{s}),\mathbf{a}_i)\right).$$

Remark 3.19. It is easy to see that if q is polynomial, the probability that the equality above does not hold is bounded by $1/\operatorname{poly}(n)$. We remark that for almost all of the applications in the paper (except the case of non-interactive key exchange) one can use polynomial modulus simply by repeating cryptographic protocol with independent randomness.[3] Observe that one can analogously construct a bounded IHwPRF family based on the Ring LWE assumption [LPR10].

Note 3.20. A similar argument shows that the function family $f_{\mathbf{A}}(\mathbf{s},\mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}$ where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}$ and $s \leftarrow \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \chi^m$, is a bounded HOWF family based on search LWE. [4]

DCR/FFI/AGCD/HNP. We now show that an IHwUF/IHwPRF is implied by any assumption that yields a (group-)homomorphic PKE. Informally, the decryption algorithm of any homomorphic PKE can be viewed as an IHwPRF, where the ciphertext space and the message space are the input space and the output space of the IHwPRF, respectively. We stress that here we use a slight generalization of the definition of weak pseudorandomness where the input/key is sampled according to some efficiently samplable distribution over the input/key space and these distributions are not necessarily uniform. However, most of the instantiations from concrete assumptions results in a uniform distribution over the key/input space.

Lemma 3.21. *Let* $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ *be a CPA-secure homomorphic PKE. Let* $\mathcal{K}$, $(\mathcal{M},\otimes)$, *and* $(\mathcal{C},\oplus)$ *be the key space, message space, and ciphertext space of* Π, *respectively. The function family* F *defined as*

$$(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda), \quad F(\mathsf{sk} \in \mathcal{K}, c \in \mathcal{C}_{\mathsf{pk}}) = \mathsf{Dec}(\mathsf{sk}, c) = m \in \mathcal{M},$$

is an IHwPRF family, where $\mathcal{C}_{\mathsf{pk}}$ *denotes to set of all valid ciphertexts under the public key* pk.

Proof. Observe that by homomorphism of Π, for any $c_1, c_2 \in \mathcal{C}_{\mathsf{pk}}$ we have

$$F(\mathsf{sk}, c_1 \oplus c_2) = F(\mathsf{sk}, c_1) \otimes F(\mathsf{sk}, c_2),$$

[1] Note that we do not need any homomorphism property for F_N, and it is just used to generated the noise for LWE samples.
[2] See Definition 3.4 for a formal definition of bounded IHwPRF.
[3] As an example, for the case of PKE, the encryptor publishes polynomially many encryptions of the same message, and the decryptor can recover the message with probability $1 - \operatorname{negl}(\lambda)$ simply by taking a majority over the decrypted messages.
[4] Note that although there are a variety of search to decision reductions for LWE (for Gaussian-like distributions), there are certain distributions for which decision LWE is easy, but search LWE is hard.

which implies the homomorphism of F. Now we show the weak pseudorandomness of F. We define a distribution $\mathcal{D}$ over $\mathcal{C}_{\mathsf{pk}}$ as follows. To sample according to $\mathcal{D}$, first generate a uniform $m \leftarrow \mathcal{M}$ and let $c = \mathsf{Enc}(\mathsf{pk}, m)$ be the encryption of m using a fresh randomness. Let $\mathcal{A}$ be an adversary against the weak pseudorandomness of F, and let n be the number of queries made by $\mathcal{A}$. We define $n+1$ hybrids as follows. Let H_j be a hybrid that the first $j-1$ queries of the adversary are answered as $(c_i \leftarrow \mathcal{D}, F(\mathsf{sk}, c_i))$ for $i \in [j-1]$, and the remaining queries are answered as (c_i, m_i) where m_i is generated randomly and independent of c_i. It is enough to show that for each $i \in [n]$ the hybrids H_{i-1} and H_i are computationally indistinguishable. To do so, given an attacker $\mathcal{A}$ that distinguishes H_{i-1} and H_i, we build an attacker $\mathcal{B}$ that breaks the semantic security of Π. First $\mathcal{B}$ asks its challenger and receives the public key pk. It then runs $\mathcal{A}$. The attacker $\mathcal{B}$ answers jth query of $\mathcal{A}$ as follows:

- If $j \in [i-1]$, $\mathcal{B}$ samples $m \leftarrow M$ and computes $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$. It then sends (c, m) to $\mathcal{A}$.
- If $i = j$, $\mathcal{B}$ samples two uniform messages $m^{(0)}, m^{(1)} \leftarrow \mathcal{M}$ and sends them to its challenger. Upon receiving c^* (challenge ciphertext), $\mathcal{B}$ sends $(c^*, m^{(1)})$ to $\mathcal{A}$.
- If $i+1 \leq j \leq n$, $\mathcal{B}$ samples $m \leftarrow M$ and computes $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$. It then sends (c, r) to $\mathcal{A}$ where r is sampled independently and uniformly over $\mathcal{M}$.

If $\mathcal{A}$ outputs 1, $\mathcal{B}$ also outputs 1. Otherwise, $\mathcal{B}$ outputs 0. Since both of $m^{(0)}$ and $m^{(1)}$ generated uniformly at random c^* is distributed according to $\mathcal{D}$. If c^* is an encryption of $m^{(1)}$ we have $F(\mathsf{sk}, c^*) = m^{(1)}$ and hence the reduction maps encryption of $m^{(1)}$ to a valid weak PRF output. On the other hand, if c^* is an encryption of $m^{(0)}$, then c^* is independent of $m^{(0)}$ and hence the reduction maps encryption of $m^{(0)}$ to a random pair $(c^*, m^{(0)})$ where c^* is distributed according to $\mathcal{D}$ and $m^{(0)}$ is uniform. Therefore, the advantage of $\mathcal{B}$ in the CPA security game is equal to the advantage of $\mathcal{A}$ in distinguishing H_{i-1} and H_i. □

Observe that a similar proof also shows that a γ-bounded homomorphic PKE implies a bounded IHwPRF. Therefore, Lemma 3.21 immediately yields an IHwPRF from the Decisional Composite Residuosity [Pai99]. It also yields constructions of (γ-bounded) IHwPRF family from several assumptions, e.g., Approximate GCD [How01], Finite Field Isomorphism [DHP+18], Hidden Number Problem [BV96].

4 Primitives from IHwUF

In this section, we present constructions of various cryptographic primitives from (γ-bounded) IHwUFs. Interestingly, none of these constructions require the source or target groups of the IHwUF to be abelian.

4.1 Two-Party Non-Interactive Key Exchange

We present a non-interactive key exchange protocol between non-uniform PPT algorithms $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$. It allows exchange of a single key-bit and is obtained from an IHwUF in a black-box manner (note that $\mathcal{A}_b$ and $\mathcal{B}_b$ operate in parallel for $b \in \{0,1\}$).

- $\mathsf{Setup}(1^\lambda)$: Given the security parameter λ, the setup algorithm creates a description $\mathcal{F}_{\mathrm{IHwUF}}$ for an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. It uniformly samples $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ for a fixed $n > 3\log(|\mathcal{X}|)$, and outputs the public parameter pp as

$$\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHwUF}} \ , \ \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right).$$

- $\mathcal{A}_0(\mathsf{pp})$: On input $\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHwUF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$, the algorithm $\mathcal{A}_0$ first samples a uniformly random $\mathbf{s} = (s_1, \ldots, s_n) \leftarrow \{0,1\}^n$ and then outputs $(\mathsf{st}_{\mathcal{A}}, x^*_{\mathcal{A}})$, where

$$\mathsf{st}_{\mathcal{A}} = \mathbf{s} \quad , \quad x^*_{\mathcal{A}} = \bigoplus_{j\in[n]} x_{j,s_j}.$$

- $\mathcal{B}_0(\mathsf{pp})$: On input $\mathsf{pp} = \left(\mathcal{F}_{\text{IHwUF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$, the algorithm $\mathcal{B}_0$ first samples a key $k \leftarrow \mathcal{K}$ and computes $y_{j,b} = F(k, x_{j,b})$ for each $j \in [n]$ and $b \in \{0,1\}$. It then outputs $(\mathsf{st}_{\mathcal{B}}, \mathbf{y}_{\mathcal{B}})$, where

$$\mathsf{st}_{\mathcal{B}} = k \quad , \quad \mathbf{y}_{\mathcal{B}} = \left(\{y_{j,b}\}_{j\in[n],b\in\{0,1\}}\right).$$

- $\mathcal{A}_1(\mathsf{pp}, \mathsf{st}_{\mathcal{A}}, \mathbf{y}_{\mathcal{B}})$: On input $\mathsf{st}_{\mathcal{A}} = \mathbf{s}$ and $\mathbf{y}_{\mathcal{B}}$, the algorithm $\mathcal{A}_1$ computes the final key-bit as

$$\mathsf{k}^* = \text{HardCore}\left(\bigotimes_{j\in[n]} y_{j,s_j}\right).$$

- $\mathcal{B}_1(\mathsf{pp}, \mathsf{st}_{\mathcal{B}}, x^*_{\mathcal{A}})$: On input $\mathsf{st}_{\mathcal{B}} = k$ and $x^*_{\mathcal{A}}$, the algorithm $\mathcal{B}_1$ computes the final key-bit as

$$\mathsf{k}^* = \text{HardCore}\left(F\left(k, x^*_{\mathcal{A}}\right)\right).$$

Instantiation from General Protocol. The aforementioned NIKE scheme can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with description $\mathcal{F}_{\text{IHwUF}}$, a fixed $n > 3\log|\mathcal{X}|$, $N^* = 1$, $\bar{N} = 1$ and $\hat{N} = 0$. Set $\mathsf{pp} = \text{X}$.
- **Pre-Evaluation:** Let $\text{X}^* = \{x^*\}$ be the output of the pre-evaluation phase. Set $x^*_{\mathcal{A}} = x^*$.
- **Evaluation:** Let Y and $\text{Y}^* = \{y^*\}$ be the outputs of the evaluation phase. Set $\mathbf{y}_{\mathcal{B}} = \text{Y}$ and $\mathsf{k}^* = \text{HardCore}(y^*)$.

Correctness and security of the scheme follow from Claim 3.10 and Theorem 3.17, respectively.

Note 4.1. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwUF if $\gamma \geq n$, with the following minor modification to the final key-generation step:

$$\mathsf{k}^* = \text{HardCore}\left(\mathcal{R}\left(\bigotimes_{j\in[n]} y_{j,s_j}\right)\right) = \text{HardCore}\left(\mathcal{R}\left(F\left(k, x^*_{\mathcal{A}}\right)\right)\right),$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.3).

Note 4.2. The aforementioned NIKE protocol can only be instantiated using an IHwUF family for which the input space is *independent* of the choice of key. [1]

4.2 CPA-Secure PKE

We present a CPA-secure public-key encryption scheme from any IHwUF. First we provide a formal definition of a CPA-secure PKE scheme and next we state the construction.

Definition 4.3. (CPA-Secure PKE.) Let $\Pi = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Π is said to be CPA-secure if for all PPT adversaries $\mathcal{A}$, the views of $\mathcal{A}$ in the games $\mathsf{Expt}_0^{\text{ind-cpa}}$ and $\mathsf{Expt}_1^{\text{ind-cpa}}$ are computationally indistinguishable.

[1] Note that this property does not hold for some instantiations from concrete assumptions, e.g., QR. See Section 3.4 for more details.

Experiment $\mathsf{Expt}_b^{\text{ind-cpa}}$:

1. The challenger runs the setup algorithm and generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp})$, and provides pk to the adversary $\mathcal{A}$.
2. The adversary $\mathcal{A}$ issues a challenge encryption query for a pair of messages $(\mathsf{m}_0, \mathsf{m}_1)$. The challenger creates the challenge ciphertext
$$\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b),$$
and sends ct^* to the adversary $\mathcal{A}$.

We now present a CPA-secure PKE from any IHwUF family.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm creates a description $\mathcal{F}_{\text{IHwUF}}$ for an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. It also fixes some integer $n > 3 \log |\mathcal{X}|$. The algorithm outputs $\mathcal{F}_{\text{IHwUF}}$ and n as the public parameter pp.
- $\mathsf{Gen}(\mathsf{pp})$: The key-generation algorithm uniformly samples $2n$ elements in $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ and a key $k \leftarrow \mathcal{K}$, and outputs
$$\mathsf{sk} = k \quad , \quad \mathsf{pk} = \{x_{j,b}, y_{j,b}\}_{j\in[n],b\in\{0,1\}},$$
where $y_{j,b} = F(k, x_{j,b})$ for each $j \in [n]$ and $b \in \{0,1\}$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$: Given the public-key $\mathsf{pk} = \{x_{j,b}, y_{j,b}\}_{j\in[n],b\in\{0,1\}}$ and a message-bit $\mathsf{m} \in \{0,1\}$, the encryption algorithm uniformly samples an n-bit string $\mathbf{r} = (r_1, \ldots, r_n) \leftarrow \{0,1\}^n$ and outputs the ciphertext $\mathsf{ct} = (c, \mathsf{e})$, where
$$c = \bigoplus_{j\in[n]} x_{j,r_j} \quad , \quad \mathsf{e} = \text{XOR}\left(\text{HardCore}\left(\bigotimes_{j\in[n]} y_{j,r_j}\right), \mathsf{m}\right).$$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: Given the secret-key $\mathsf{sk} = k$ and the ciphertext $\mathsf{ct} = (c, \mathsf{e})$, the algorithm outputs the bit
$$\mathsf{m}' = \text{XOR}\left(\text{HardCore}\left(F(k, c)\right), \mathsf{e}\right).$$

Instantiation from General Protocol. The aforementioned PKE scheme can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with description $\mathcal{F}_{\text{IHwUF}}$, a fixed $n > 3 \log |\mathcal{X}|$, $N^* = 0$, $\bar{N} = 1$ and $\hat{N} = 1$. Set $\mathsf{pk}_1 = \text{X}$.
- **Evaluation:** Set $\mathsf{sk} = k$ (where $k \in \mathcal{K}$ is the IHwUF key) and $\mathsf{pk}_2 = \text{Y}$, where Y is the output of the evaluation phase. Output $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2))$.
- **Post-Evaluation:** Let $\hat{\text{X}} = \{\hat{x}\}$ and $\hat{\text{Y}} = \{\hat{y}\}$ be the outputs of the post-evaluation phase. Set:
$$c = \hat{x}, \quad \mathsf{e} = \text{XOR}\left(\text{HardCore}(\hat{y}), \mathsf{m}\right),$$
where m is the message-bit.

Correctness and security of the scheme follow from Claim 3.10 and Theorem 3.17, respectively.

Note 4.4. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwUF if $\gamma \geq n$, with the following minor modification to the encryption algorithm:

$$\mathsf{e} = \mathrm{XOR}\left(\mathrm{HardCore}\left(\mathcal{R}\left(\bigotimes_{j\in[n]} y_{j,r_j}\right)\right), \mathsf{m}\right),$$

and the following minor modification to the decryption algorithm:

$$\mathsf{m}' = \mathrm{XOR}\left(\mathrm{HardCore}\left(\mathcal{R}\left(F(k,c)\right)\right), \mathsf{e}\right)$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.3).

Note 4.5. The aforementioned PKE can also be instantiated using an (γ-bounded) IHwUF family for which the input space is dependent of the choice of key with the following minor modification: the setup algorithm only outputs the description of the key space and the output space of the IHwUF, while the description of the input space is published along with the public key by the key generation algorithm.

4.3 Trapdoor Functions

In this subsection, we show that IHwUFs imply trapdoor functions with almost-perfect correctness. Garg and Hajiabadi [GH18] introduced a primitive called recyclable *one-way function with encryption* (OWFE), and they showed that recyclable OWFEs imply TDFs (in a black-box way) with negligibly small inversion error. In this section, we demonstrate how to construct recyclable OWFE from IHwUF. We begin by presenting the formal definition of recyclable OWFE from [GH18], followed by our construction.

Definition 4.6. (Recyclable One-Way Function with Encryption.) A recyclable OWFE scheme is a tuple of four PPT algorithms $\mathrm{OWFE} = (\mathsf{Setup}, \mathsf{OWF}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

- $\mathsf{Setup}\left(1^\lambda\right)$: Given the security parameter λ, it sets $n = n(\lambda)$ and $\ell = \ell(\lambda)$ for some fixed polynomial functions and outputs pp for a one-way function $\mathsf{OWF} : \{0,1\}^n \to \{0,1\}^\ell$.
- $\mathsf{OWF}\left(\mathsf{pp}, \mathbf{s}\right)$: Given the public parameter pp, it maps a string $\mathbf{s} \in \{0,1\}^n$ to an image $h \in \{0,1\}^\ell$.
- $\mathsf{Enc}\left(\mathsf{pp}, h, (i, b^*)\right)$: Given the public parameter pp, an image $h \in \{0,1\}^\ell$, an index $i \in [n]$ and a bit $b^* \in \{0,1\}$, it outputs a ciphertext ct and an additional bit $\mathsf{e} \in \{0,1\}$.
- $\mathsf{Dec}\left(\mathsf{pp}, \mathbf{s}, (i, b^*), \mathsf{ct}\right)$:[1] Given the public parameter pp, a preimage string $\mathbf{s}$, an index $i \in [n]$, a bit $b^* \in \{0,1\}$ and a ciphertext ct, it outputs $\mathsf{e}' \in \{0,1\} \cup \{\perp\}$.

The following correctness and security properties must be satisfied:

- **Correctness:** If $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$, then for all $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and all $i \in [n]$, letting $h = \mathsf{OWF}\left(\mathsf{pp}, \mathbf{s}\right)$ and $b^* = s_i$, it holds with overwhelming probability over the randomness of Enc that if $(\mathsf{ct}, \mathsf{e}) \leftarrow \mathsf{Enc}\left(\mathsf{pp}, h, (i, b^*)\right)$, then we have

$$\mathsf{Dec}\left(\mathsf{pp}, \mathbf{s}, (i, b^*), \mathsf{ct}\right) = \mathsf{e}.$$

- **One-Wayness:** For any PPT adversary $\mathcal{A}$ we have

$$\Pr[\mathrm{OWF}(\mathsf{pp}, \mathcal{A}(h)) = h] \leq \mathrm{negl}(\lambda),$$

where $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathbf{s} \leftarrow \{0,1\}^n$ and $h = \mathrm{OWF}(\mathsf{pp}, \mathbf{s})$.

[1] Notice that although e is part of the output of the encryption algorithm, the decryption algorithm does *not* take e as part of its input. The aim of the decryption algorithm is in fact to output e given only the ciphertext ct. This property is used in the construction of TDFs. See [GH18] for details.

- **Security:** For $b \in \{0,1\}$, define the experiment $\mathsf{Expt}_b^{\text{ind-OWFE}}$ between a challenger and an adversary $\mathcal{A}$ as follows:

Experiment $\mathsf{Expt}_b^{\text{ind-OWFE}}$:

1. The adversary $\mathcal{A}$ takes as input 1^n and 1^ℓ, and sends a string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and an index $i \in [n]$ to the challenger.
2. The challenger generates the public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$. It computes $h = \mathsf{OWF}\left(\mathsf{pp}, \mathbf{s}\right)$ and $(\mathsf{ct}^*, \mathsf{e}_0^*) \leftarrow \mathsf{Enc}\left(\mathsf{pp}, h, (i, 1-s_i)\right)$. Finally, it samples $\mathsf{e}_1^* \leftarrow \{0,1\}$ and sends $(\mathsf{pp}, \mathsf{ct}^*, \mathsf{e}_b^*)$ to the adversary.

An OWFE encryption scheme $(\mathsf{Setup}, \mathsf{OWF}, \mathsf{Enc}, \mathsf{Dec})$ is said to be secure if for all PPT adversaries $\mathcal{A}$, the views of the adversary in $\mathsf{Expt}_0^{\text{ind-OWFE}}$ and $\mathsf{Expt}_1^{\text{ind-OWFE}}$ are computationally indistinguishable.

- **Recyclability:** An OWFE scheme is said to be recyclable if for all $\mathsf{pp} \in \mathsf{Setup}\left(1^\lambda\right)$, all $\mathbf{s}_1, \mathbf{s}_2 \in \{0,1\}^n$, all $i \in [n]$, all $b^* \in \{0,1\}$ and all randomness r, letting $(\mathsf{ct}_1, \mathsf{e}_1) \leftarrow \mathsf{Enc}\left(\mathsf{pp}, h_1 = \mathsf{OWF}\left(\mathsf{pp}, \mathbf{s}_1\right), (i, b^*); r\right)$ and $(\mathsf{ct}_2, \mathsf{e}_2) \leftarrow \mathsf{Enc}\left(\mathsf{pp}, h_2 = \mathsf{OWF}\left(\mathsf{pp}, \mathbf{s}_2\right), (i, b^*); r\right)$ we have $\mathsf{ct}_1 = \mathsf{ct}_2$. [1]

Recently, Garg *et al.* [GGH18] introduced an enhanced version of recyclable OWFE called *smooth* recyclable OWFE. A recyclable OWFE $= (\mathsf{Setup}, \mathsf{OWF}, \mathsf{Enc}, \mathsf{Dec})$ is said to be (ℓ, n)-*smooth* if for any two (ℓ, n)-sources $\mathcal{S}_1$ and $\mathcal{S}_2$ and for any PPT adversary $\mathcal{A}$, we have

$$\left|\Pr\left[\mathcal{A}(\mathsf{pp}, \mathrm{OWF}(\mathsf{pp}, \mathbf{s}_1)) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{pp}, \mathrm{OWF}(\mathsf{pp}, \mathbf{s}_2)) = 1\right]\right| \leq \mathrm{negl}(\lambda),$$

where $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathbf{s}_1 \leftarrow \mathcal{S}_1$ and $\mathbf{s}_2 \leftarrow \mathcal{S}_2$.

Construction from IHwUF. We show a black-box construction of smooth recyclable OWFE from any IHwUF family.

- $\mathsf{Setup}\left(1^\lambda\right)$: Given the security parameter λ the setup algorithm creates a description $\mathcal{F}_{\text{IHwUF}}$ for an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, and fixes some integer $n = n(\lambda) > 3\log|\mathcal{X}|$. It samples $2n$ uniform elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n], b\in\{0,1\}}$ and outputs the public parameter pp as

$$\mathsf{pp} = \left(\mathcal{F}_{\text{IHwUF}}, \{x_{j,b}\}_{j\in[n], b\in\{0,1\}}\right).$$

- $\mathsf{OWF}\left(\mathsf{pp}, \mathbf{s}\right)$: Given pp and a string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$, generate the corresponding image as

$$h = \bigoplus_{j\in[n]} x_{j,s_j}.$$

- $\mathsf{Enc}\left(\mathsf{pp}, h, (i, b^*)\right)$: Given pp, an image h, an index $i \in [n]$ and $b^* \in \{0,1\}$, the encryption algorithm randomly samples $k \leftarrow \mathcal{K}$ and computes the following

$$y_{i,b^*} = F\left(k, x_{i,b^*}\right), \quad y_{i,1-b^*} = \perp,$$

$$y_{j,b} = F\left(k, x_{j,b}\right) \text{ for } j \in [n] \setminus \{i\}, b \in \{0,1\}.$$

It finally outputs the pair

$$(\mathsf{ct}, \mathsf{e}) = \left(\{y_{j,b}\}_{j\in[n], b\in\{0,1\}}, \mathrm{HardCore}\left(F\left(k, h\right)\right)\right).$$

[1] Informally, this says that the ct component is independent of the image h.

- $\mathsf{Dec}(\mathsf{pp}, \mathbf{s}, (i, b^*), \mathsf{ct})$: Given pp, a string $\mathbf{s} = (s_1, \ldots, s_n)$ and a ciphertext $\mathsf{ct} = \left(\{y_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$, the decryption algorithm outputs

$$\mathsf{e}' = \begin{cases} \text{HardCore}\left(\bigotimes_{j\in[n]} y_{j,s_j}\right) & \text{if } s_i = b^* \\ \perp & \text{otherwise.} \end{cases}$$

Instantiation from General Protocol. The aforementioned OWFE scheme can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with description $\mathcal{F}_{\text{IHwUF}}$, a fixed $n > 3\log|\mathcal{X}|$, $N^* = 1$, $\bar{N} = 1$ and $\hat{N} = 0$. Set $\mathsf{pp} = (\mathcal{F}_{\text{IHwUF}}, \mathrm{X})$.
- **Pre-Evaluation.** In the generic protocol, the pre-evaluation phase samples a uniformly random binary string $\mathbf{s} \in \{0,1\}^n$. One may view this as an input to the OWF. Consequently, if $\mathrm{X}^* = \{x^*\}$ is the output of the post-evaluation phase, set the image $h = x^*$.
- **Evaluation:** Let $\mathrm{Y} = \{y_{j,b}\}_{j\in[n],b\in\{0,1\}}$ and $\mathrm{Y}^* = \{y^*\}$ be the outputs of the evaluation phase. For a given $i \in [n]$ and $b^* \in \{0,1\}^n$, set the ciphertext $\mathsf{ct} = \mathrm{Y} \setminus \{y_{i,1-b^*}\}$ and the bit $\mathsf{e} = \text{HardCore}(y^*)$.

To see that the instantiation satisfies the desired properties of a recyclable OWFE scheme, consider the following:

- Correctness follows from Claim 3.10. More specifically, given a binary string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and ct such that $(\mathsf{ct}, e) = \mathsf{Enc}(\mathsf{pp}, h, (i, b^*))$ for $h = \mathsf{OWF}(\mathsf{pp}, \mathbf{s})$ and $s_i = b^*$, the decryption algorithm does not need $y_{i,1-b^*}$ to recover the bit e.
- One-wayness follows from Lemma 3.7.
- Security follows from Theorem 3.17.
- Recyclability follows from the fact that ct does not depend on the image-string h.

Finally, the aforementioned OWFE scheme is (ℓ, n)-smooth for any choice of $\ell \geq \log|\mathcal{X}| + \omega(\log\lambda)$. This follows directly from the leftover hash lemma. More specifically, let $(\mathcal{S}_1, \mathcal{S}_2)$ be (ℓ, n)-sources for $\ell \geq \log|\mathcal{X}| + \omega(\log\lambda)$. Then, for any $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathbf{s}_1 \leftarrow \mathcal{S}_1$ and $\mathbf{s}_2 \leftarrow \mathcal{S}_2$, the distributions of $\mathrm{OWF}(\mathsf{pp}, \mathbf{s}_1)$ and $\mathrm{OWF}(\mathsf{pp}, \mathbf{s}_2)$ are statistically negligibly close to uniform by Lemma 2.1.

Implications. Garg *et al.* [GGH18] showed that an (ℓ, n)-smooth recyclable OWFE scheme implies:

1. TDFs with almost-perfect correctness, which is an improvement over TDFs with negligible inversion error (see [GH18] and [GGH18] for details).
2. CCA2-secure deterministic encryption, where the CCA2-security guarantee holds w.r.t. plaintexts sampled from (ℓ, n)-sources.

Note 4.7. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwUF if $\gamma \geq n$, with the following minor modification to the encryption algorithm:

$$\mathsf{e} = \text{HardCore}\left(\mathcal{R}\left(F\left(k, h\right)\right)\right),$$

and the following minor modification to the decryption algorithm:

$$\mathsf{e}' = \begin{cases} \text{HardCore}\left(\mathcal{R}\left(\bigotimes_{j\in[n]} y_{j,s_j}\right)\right) & \text{if } s_i = b^* \\ \perp & \text{otherwise.} \end{cases}$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.3). Finally, the aforementioned construction can only be instantiated from an IHwUF family for which the input space is independent of the choice of key.

4.4 Blind Batch Encryption

In this subsection, we show that IHwUFs imply "batch encryption", a cryptographic primitive introduced by Brakerski et al. in [BLSV18]. [1] We now present the formal definition of blind batch encryption, followed by our construction.

Definition 4.8. (Batch Encryption.) A batch encryption scheme is a tuple of four PPT algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

- $\mathsf{Setup}\left(1^\lambda\right)$: Given the security parameter λ, it outputs the public parameter pp.
- $\mathsf{Gen}\left(\mathsf{pp}, \mathbf{s}\right)$: Given pp, it projects the string $\mathbf{s} \in \{0,1\}^n$ to a hash value h where $n = n(\lambda)$ is some fixed polynomial included in pp.
- $\mathsf{Enc}\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1)\right)$: Given pp, a hash value h, an index $i \in [n]$ and a message pair $(\mathsf{m}_0, \mathsf{m}_1)$, it outputs a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$.
- $\mathsf{Dec}\left(\mathsf{pp}, \mathbf{s}, i, \mathsf{ct}\right)$: Given pp, a string $\mathbf{s}$, an index $i \in [n]$ and a ciphertext ct, it outputs a string m.

The following completeness, succinctness, security, and blindness properties must be satisfied:

- **Correctness:** If $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$, then for all $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$, all $i \in [n]$, and all message-pairs $(\mathsf{m}_0, \mathsf{m}_1)$, letting $h = \mathsf{Gen}\left(\mathsf{pp}, \mathbf{s}\right)$ and $\mathsf{ct} \leftarrow \mathsf{Enc}\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1)\right)$ it holds with overwhelming probability over the randomness of Enc that
$$\mathsf{Dec}\left(\mathsf{pp}, \mathbf{s}, i, \mathsf{ct}\right) = \mathsf{m}_{s_i}.$$
- **Succinctness.** A batch encryption scheme is fully succinct if for some string $\mathbf{s} \in \{0,1\}^n$, letting $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$ and $h = \mathsf{Gen}\left(\mathsf{pp}, \mathbf{s}\right)$, we have $|h| \leq \mathrm{poly}\left(\lambda\right)$ for some *fixed* polynomial in the security parameter λ.
- **Security:** For each bit $b \in \{0,1\}$, define the following experiment $\mathsf{Expt}_b^{\text{ind-batch}}$ between a challenger and an adversary $\mathcal{A}$:

Experiment $\mathsf{Expt}_b^{\text{ind-batch}}$:

1. The adversary $\mathcal{A}$ takes as input 1^λ and 1^n. It chooses an index $i \in [n]$ and a binary string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$, and sends $(\mathbf{s}, i)$ to the challenger.
2. The challenger generates $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$, and sends pp to the adversary $\mathcal{A}$.
3. The adversary $\mathcal{A}$ generates $\mathbf{m}^{(0)} = \left(\mathsf{m}_0^{(0)}, \mathsf{m}_1^{(0)}\right)$ and $\mathbf{m}^{(1)} = \left(\mathsf{m}_0^{(1)}, \mathsf{m}_1^{(1)}\right)$ such that $\mathsf{m}_{s_i}^{(0)} = \mathsf{m}_{s_i}^{(1)}$, and sends them to the challenger.
4. The challenger computes the hash $h = \mathsf{Gen}\left(\mathsf{pp}, \mathbf{s}\right)$, generates the ciphertext
$$\mathsf{ct}^* \leftarrow \mathsf{Enc}\left(\mathsf{pp}, h, \left(i, \mathsf{m}_0^{(b)}, \mathsf{m}_1^{(b)}\right)\right),$$
and sends ct^* to the adversary $\mathcal{A}$.

A batch encryption scheme $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be secure if for all PPT adversaries $\mathcal{A}$, the views of the adversary in $\mathsf{Expt}_0^{\text{ind-batch}}$ and $\mathsf{Expt}_1^{\text{ind-batch}}$ are computationally indistinguishable.

[1] An equivalent cryptosystem, nomenclatured as *hash encryption*, was introduced by Döttling and Garg in [DGHM18].

- **Blindness:** Let $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a batch encryption scheme, such that one can view a ciphertext produced by the encryption algorithm as $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$, where ct_1 is produced by the sub-routine Enc_1 and ct_2 is produced by the sub-routine Enc_2. Also, for $b \in \{0,1\}$, define the experiment $\mathsf{Expt}_b^{\text{blind-batch}}$ between a challenger and an adversary $\mathcal{A}$ as follows:

Experiment $\mathsf{Expt}_b^{\text{blind-batch}}$:

1. The adversary $\mathcal{A}$ takes as input 1^λ and 1^n, and sends a binary preimage string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n, i \in [n]$ to the challenger.
2. The challenger generates $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$ and $h = \mathsf{Gen}\left(\mathsf{pp}, \mathbf{s}\right)$.
3. The challenger randomly generates $\mathbf{m} = (\mathsf{m}_0, \mathsf{m}_1)$ and creates:

$$\begin{aligned} \mathsf{ct}_1^* &\leftarrow \mathsf{Enc}_1\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1)\right) \\ \mathsf{ct}_2^* &\leftarrow \mathsf{Enc}_2\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1)\right). \end{aligned}$$

   - If $b = 0$, the challenger sets $\mathsf{ct}^* = (\mathsf{ct}_1^*, \mathsf{ct}_2^*)$.
   - If $b = 1$, the challenger sets $\mathsf{ct}^* = (\mathsf{ct}_1^*, \sigma^*)$ where $\sigma^* \leftarrow \{0,1\}^{|\mathsf{ct}_2^*|}$.
4. Finally, the challenger sends $(\mathsf{pp}, \mathsf{ct}^*)$ to the adversary $\mathcal{A}$.

A batch encryption scheme $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be blind if:

1. The encryption subroutine Enc_1 does not depend on either the hash value h or the message pair $(\mathsf{m}_0, \mathsf{m}_1)$. Hence we can write the first subroutine as $\mathsf{Enc}_1\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1); r\right) = \mathsf{Enc}_1\left(\mathsf{pp}, i; r\right)$.
2. For all PPT adversaries $\mathcal{A}$, the views of the adversary in $\mathsf{Expt}_0^{\text{blind-batch}}$ and $\mathsf{Expt}_1^{\text{blind-batch}}$ are computationally indistinguishable.

Construction from IHwUF. We present a construction of fully succinct blind batch encryption from any IHwUF family.

- $\mathsf{Setup}\left(1^\lambda\right)$: Given the security parameter λ the setup algorithm creates a description $\mathcal{F}_{\text{IHwUF}}$ for an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. It then fixes some integer $n = n(\lambda) > 3\log|\mathcal{X}|$ and samples $2n$ uniform elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ and outputs the public parameter pp as

$$\mathsf{pp} = \left(\mathcal{F}_{\text{IHwUF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right).$$

- $\mathsf{Gen}\left(\mathsf{pp}, \mathbf{s}\right)$: Given $\mathsf{pp} = \left(\mathcal{F}_{\text{IHwUF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$ and a binary string $\mathbf{s} = (s_1, \ldots, s_n)$, generate the corresponding hash value h as

$$h = \bigoplus_{j\in[n]} x_{j,s_j}.$$

- $\mathsf{Enc}\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1)\right)$: Given $\mathsf{pp} = \left(\mathcal{F}_{\text{IHwUF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$, a hash value h, an index $i \in [n]$ and $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\} \times \{0,1\}$, the encryption algorithm randomly samples $k_0, k_1 \leftarrow \mathcal{K}$ and computes the

following

$$\begin{aligned}
y_{i,0}^{(0)} &= F\left(k_0, x_{i,0}\right), \quad y_{i,1}^{(1)} = F\left(k_1, x_{i,1}\right) \\
y_{i,1}^{(0)} &= y_{i,0}^{(1)} = \perp \\
y_{j,b}^{(0)} &= F\left(k_0, x_{j,b}\right) \text{ for } j \in [n] \setminus \{i\}, b \in \{0,1\} \\
y_{j,b}^{(1)} &= F\left(k_1, x_{j,b}\right) \text{ for } j \in [n] \setminus \{i\}, b \in \{0,1\} \\
\mathsf{e}_0 &= \mathrm{XOR}\left(\mathrm{HardCore}\left(F\left(k_0, h\right)\right), \mathsf{m}_0\right) \\
\mathsf{e}_1 &= \mathrm{XOR}\left(\mathrm{HardCore}\left(F\left(k_1, h\right)\right), \mathsf{m}_1\right).
\end{aligned}$$

It finally outputs the ciphertext

$$\mathsf{ct} = \left(\mathsf{ct}_1 = \left\{y_{j,b}^{(b')}\right\}_{j\in[n],(b,b')\in\{0,1\}\times\{0,1\}}, \mathsf{ct}_2 = (\mathsf{e}_0, \mathsf{e}_1)\right).$$

- $\mathsf{Dec}\left(\mathsf{pp}, \mathbf{s}, i, \mathsf{ct}\right)$: Given pp, a string $\mathbf{s} = (s_1, \ldots, s_n)$ and a ciphertext ct the decryption algorithm outputs

$$\mathsf{m}'_{s_i} = \mathrm{XOR}\left(\mathrm{HardCore}\left(\bigotimes_{j\in[n]} y_{j,s_j}^{(s_i)}\right), \mathsf{e}_{s_i}\right),$$

where $\mathsf{ct} = \left(\left\{y_{j,b}^{(b')}\right\}_{j\in[n],(b,b')\in\{0,1\}\times\{0,1\}}, (\mathsf{e}_0, \mathsf{e}_1)\right)$.

Instantiation from General Protocol. The aforementioned BE scheme can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with description $\mathcal{F}_{\mathrm{IHwUF}}$, a fixed $n > 3\log|\mathcal{X}|$, $N^* = 1$, $\bar{N} = 2$ and $\hat{N} = 0$. Set $\mathsf{pp} = (\mathcal{F}_{\mathrm{IHwUF}}, \mathrm{X})$, where X is the set of base elements.
- **Pre-Evaluation.** In the general protocol, the pre-evaluation phase samples a uniformly random binary string $\mathbf{s} \in \{0,1\}^n$. One may view this as an input to the generation algorithm. Consequently, if $\mathrm{X}^* = \{x^*\}$ is the output of the post-evaluation phase, set the image $h = x^*$.
- **Evaluation:** Let $\mathrm{Y} = \{y_{j,b}^{(0)}, y_{j,b}^{(1)}\}_{j\in[n],b\in\{0,1\}}$ and $\mathrm{Y}^* = (y_0^*, y_1^*)$ be the tuples output by the evaluation phase. For a given $i \in [n]$, set $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$ where

$$\begin{aligned}
\mathsf{ct}_1 &= \mathrm{Y} \setminus \left\{y_{i,1}^{(0)}, y_{i,0}^{(1)}\right\}, \\
\mathsf{ct}_2 &= \left(\mathsf{e}_0 = \mathrm{HardCore}\left(y_0^*\right), \mathsf{e}_1 = \mathrm{HardCore}\left(y_1^*\right)\right).
\end{aligned}$$

To see that the instantiation satisfies the properties of a blind batch encryption scheme, consider the following:

- Correctness follows from Claim 3.10. More specifically, given a binary string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and a ciphertext $\mathsf{ct} = \mathsf{Enc}\left(\mathsf{pp}, h, (i, \mathsf{m}_0, \mathsf{m}_1)\right)$ such that $h = \mathsf{Gen}(\mathsf{pp}, \mathbf{s})$ and $s_i = b^*$, the decryption algorithm does not need $y_{i,1-b^*}^{(b^*)}$ to recover the message m.
- One-wayness follows from Lemma 3.7.
- Security follows from Theorem 3.17.
- Blindness follows from the fact that the ciphertext component ct_1 does not depend on the image h.

Implications. Brakerski et al. [BLSV18] showed that fully succinct blind batch encryption scheme along with blind garbled circuit (which can be constructed from any one-way function) imply:

1. Anonymous IBE,
2. Bounded KDM-secure PKE,
3. Leakage-resilient PKE with resilience to leakage of a $(1 - o(1))$-fraction of the secret-key.

We showed that IHwUF implies blind batch encryption. As IHwUF is also enough to construct blind garbled circuits, it follows that any IHwUF implies the above three primitives.

Note 4.9. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwUF if $\gamma \geq n$, with the following minor modification to the encryption algorithm:

$$\begin{aligned} \mathsf{e}_0 &= \mathrm{XOR}\left(\mathrm{HardCore}\left(\mathcal{R}\left(F\left(k_0, h\right)\right)\right), \mathsf{m}_0\right), \\ \mathsf{e}_1 &= \mathrm{XOR}\left(\mathrm{HardCore}\left(\mathcal{R}\left(F\left(k_1, h\right)\right)\right), \mathsf{m}_1\right). \end{aligned}$$

and the following minor modification to the decryption algorithm:

$$\mathsf{m}'_{s_i} = \mathrm{XOR}\left(\mathrm{HardCore}\left(\mathcal{R}\left(\bigotimes_{j\in[n]} y_{j,s_j}^{(s_i)}\right)\right), \mathsf{e}_{s_i}\right),$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.3). Finally, the aforementioned construction can only be instantiated from an IHwUF family for which the input space is independent of the choice of key.

4.5 Hinting PRGs

In this subsection, we show that IHwUFs imply hinting PRGs, which are a stronger variant of traditional PRGs introduced by Koppula and Waters in [KW18]. Hinting PRGs can be used to generically transform any CPA-secure ABE into a CCA-secure one.

Informal Description. Informally, a hinting PRG takes n bits as input and outputs $n \cdot \ell$ output bits for some fixed polynomials $n = n(\lambda)$ and $\ell = \ell(\lambda)$, such that no PPT adversary can distinguish between $2n$ uniformly random strings and $2n$ strings such that half the strings are output by the PRG, and the remaining half are uniformly random, *even if* the strings are arranged as a $2 \times n$ matrix as the follows: in the i^{th} column of this matrix, the top entry is pseudorandom if the i^{th} bit of the seed is 0; else, the bottom entry is pseudorandom. Note that such a matrix-based arrangement carries some information (or "hint") about the seed, and the indistinguishability guarantee in the presence of such an arrangement is what makes a hinting PRG stronger than a traditional PRG.

Definition 4.10. (Hinting PRG.) A hinting PRG is a tuple of PPT algorithms HPRG = $(\mathsf{Setup}, \mathsf{Eval})$ defined as follows:

- $\mathsf{Setup}\left(1^\lambda\right)$: Given the security parameter λ, it sets $n = n(\lambda)$ and $\ell = \ell(\lambda)$ for some fixed polynomial functions and outputs (pp, n, ℓ), where pp is the public parameter.
- $\mathsf{Eval}(\mathsf{pp}, \mathbf{s}, i^*)$: Given the public parameter pp, a seed $\mathbf{s} \in \{0,1\}^n$ and an index $i^* \in [n] \cup \{0\}$, it outputs a string $\mathbf{e} \in \{0,1\}^\ell$.

Security. For $b \in \{0,1\}$, define the experiment $\mathsf{Expt}_b^{\mathrm{HPRG}}$ between a challenger and an adversary $\mathcal{A}$ as follows:

Experiment $\mathsf{Expt}_b^{\mathrm{HPRG}}$:

1. The challenger generates $(\mathsf{pp}, n, \ell) \leftarrow \mathsf{Setup}\left(1^\lambda\right)$ and provides the same to the adversary $\mathcal{A}$.
2. The challenger uniformly samples $\mathbf{s} = (s_1, \ldots, s_n) \leftarrow \{0,1\}^n$ and sets the following.

$$\begin{aligned} y_0^{(0)} &= \mathsf{Eval}(\mathsf{pp}, \mathbf{s}, 0), \quad y_0^{(1)} \leftarrow \{0,1\}^\ell, \\ y_{i,s_i}^{(0)} &= \mathsf{Eval}(\mathsf{pp}, \mathbf{s}, i), \quad y_{i,1-s_i}^{(1)} \leftarrow \{0,1\}^\ell \text{ for each } i \in [n], \\ y_{i,b'}^{(1)} &\leftarrow \{0,1\}^\ell \text{ for each } i \in [n], b' \in \{0,1\}. \end{aligned}$$

and sends $\left(y_0^{(b)}, \{y_{j,b'}^{(b)}\}_{j \in [n], b' \in \{0,1\}}\right)$ to the adversary $\mathcal{A}$.

An HPRG $(\mathsf{Setup}, \mathsf{Eval})$ is secure if for all PPT adversaries $\mathcal{A}$, the views of the adversary in $\mathsf{Expt}_0^{\mathrm{HPRG}}$ and $\mathsf{Expt}_1^{\mathrm{HPRG}}$ are computationally indistinguishable.

Construction from IHwUF. We show a black-box construction of HPRG from any IHwUF family.

- $\mathsf{Setup}\left(1^\lambda\right)$: Given the security parameter λ the setup algorithm creates a description $\mathcal{F}_{\mathrm{IHwUF}}$ for an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, and fixes some integer $n = n(\lambda) > 3 \log |\mathcal{X}|$ and some integer $\ell = \ell(\lambda)$. It samples $2n$ group elements from $\mathcal{X}$ and $2(n+1) \cdot \ell$ keys from $\mathcal{K}$ as

$$\{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}, \quad \{k_{i_1,i_2,\beta} \leftarrow \mathcal{K}\}_{i_1 \in [n] \cup \{0\}, i_2 \in [\ell], \beta \in \{0,1\}}.$$

  For each $i_2 \in [\ell]$, it creates a $2 \times n$ matrix $\mathbf{H}_{0,i_2} \in \mathcal{Y}^{2 \times n}$ such that

$$\mathbf{H}_{0,i_2}[b,j] = F\left(k_{0,i_2,0}, x_{j,b}\right) \text{ for each } j \in [n], b \in \{0,1\}.$$

  Additionally, for each $i_1 \in [n], i_2 \in [\ell], \beta \in \{0,1\}$, it creates a $2 \times n$ matrix $\mathbf{H}_{i_1,i_2,\beta} \in \mathcal{Y}^{2 \times n}$ such that for each $j \in [n], b \in \{0,1\}$, we have

$$\mathbf{H}_{i_1,i_2,\beta}[b,j] = \begin{cases} \perp & \text{if } (i_1, \beta) = (j, b) \\ F\left(k_{i_1,i_2,\beta}, x_{j,b}\right) & \text{otherwise.} \end{cases}$$

  Finally, it outputs (pp, n, ℓ), where [1]

$$\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHwUF}}, \{\mathbf{H}_{0,i_2}\}_{i_2 \in [\ell]}, \{\mathbf{H}_{i_1,i_2,\beta}\}_{i_1 \in [n], i_2 \in [\ell], \beta \in \{0,1\}}\right).$$

- $\mathsf{Eval}(\mathsf{pp}, \mathbf{s}, i^*)$: On input $\mathsf{pp} = \left(\{\mathbf{H}_{0,i_2}\}_{i_2 \in [\ell]}, \{\mathbf{H}_{i_1,i_2,\beta}\}_{i_1 \in [n], i_2 \in [\ell], \beta \in \{0,1\}}\right)$ and $\mathbf{s} = (s_1, \ldots, s_n)$, the evaluation algorithm outputs $\mathbf{e} = (\mathsf{e}_1, \ldots, \mathsf{e}_\ell)$, where for each $i_2 \in [\ell]$, we have

$$\mathsf{e}_{i_2} = \begin{cases} \mathrm{HardCore}\left(\bigotimes_{j \in [n]} \mathbf{H}_{0,i_2}[s_j, j]\right) & \text{if } i^* = 0 \\ \mathrm{HardCore}\left(\bigotimes_{j \in [n]} \mathbf{H}_{i^*,i_2,s_{i^*}}[s_j, j]\right) & \text{otherwise.} \end{cases}$$

[1] In [KW18], Koppula and Waters explicitly include the randomness for generating hardcore bits in pp. Here, we implicitly assume that every element in the output group of the IHwUF has a deterministic hardcore bit. If this is not the case, the randomness for hardcore bit generation should be included in pp.

Instantiation from General Protocol. The aforementioned HPRG can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwUF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with description $\mathcal{F}_{\text{IHwUF}}$, a fixed $n > 3 \log |\mathcal{X}|$, $N^* = 0$, $\bar{N} = 2(n+1) \cdot \ell$ and $\hat{N} = 1$. Set $\mathsf{pp}_1 = (\mathcal{F}_{\text{IHwUF}}, \mathrm{X})$.

- **Evaluation:** Let $\mathrm{Y} = \left\{ y_{j,b}^{(\bar{n})} \right\}_{\bar{n} \in [\bar{N}], j \in [n], b \in \{0,1\}}$ be the tuple of elements output by the evaluation phase. For each $i_2 \in [\ell]$, create a $2 \times n$ matrix $\mathbf{H}_{0,i_2} \in \mathcal{Y}^{2 \times n}$ such that

$$\mathbf{H}_{0,i_2}[b,j] = y_{j,b}^{(i_2)} \text{ for each } j \in [n], b \in \{0,1\}.$$

  Additionally, for each $i_1 \in [n], i_2 \in [\ell], \beta \in \{0,1\}$, create a $2 \times n$ matrix $\mathbf{H}_{i_1,i_2,\beta} \in \mathcal{Y}^{2 \times n}$ such that for each $j \in [n], b \in \{0,1\}$, we have

$$\mathbf{H}_{i_1,i_2,\beta}[b,j] = \begin{cases} \perp & \text{if } (i_1, \beta) = (j, b) \\ y_{j,b}^{(2(i_1 \cdot \ell + i_2) + (\beta+1))} & \text{otherwise.} \end{cases}$$

  Set the public parameter pp for the HPRG as $\mathsf{pp} = (\mathsf{pp}_1, \mathsf{pp}_2)$, where

$$\mathsf{pp}_2 = \left(\{\mathbf{H}_{0,i_2}\}_{i_2 \in [\ell]}, \{\mathbf{H}_{i_1,i_2,\beta}\}_{i_1 \in [n], i_2 \in [\ell], \beta \in \{0,1\}} \right).$$

- **Post-Evaluation.** Recall that in the general protocol, the post-evaluation phase samples a uniformly random binary string $\mathbf{r} \in \{0,1\}^n$. One may view this as the input string to the evaluation algorithm of the HPRG, along with the auxiliary input index $i^* \in [n] \cup \{0\}$. Let $\hat{\mathrm{Y}} = \{\hat{y}_{\bar{n}}\}_{\bar{n} \in [\bar{N}]}$ be the output of the post-evaluation phase. For each $i_2 \in [\ell]$, set

$$\mathsf{e}_{i_2} = \begin{cases} \text{HardCore}\left(\hat{y}_{i_2}\right) & \text{if } i^* = 0 \\ \text{HardCore}\left(\hat{y}_{(2(i^* \cdot \ell + i_2) + (\beta+1))}\right) & \text{otherwise.} \end{cases}$$

  and set the evaluation output for the HPRG as $\mathbf{e} = (\mathsf{e}_1, \ldots, \mathsf{e}_\ell)$.

Finally, security of the HPRG follows from Theorem 3.17.

Note 4.11. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwUF if $\gamma \geq n$, with the following minor modification to the evaluation algorithm:

$$\mathsf{e}_{i_2} = \begin{cases} \text{HardCore}\left(\mathcal{R}\left(\bigotimes_{j \in [n]} \mathbf{H}_{0,i_2}[s_j, j]\right)\right) & \text{if } i^* = 0 \\ \text{HardCore}\left(\mathcal{R}\left(\bigotimes_{j \in [n]} \mathbf{H}_{i^*,i_2,s_{i^*}}[s_j, j]\right)\right) & \text{otherwise.} \end{cases}$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.3).

Note 4.12. This construction can be instantiated using an (γ-bounded) IHwUF family for which the input space is dependent of the choice of key with the following minor modification to the setup algorithm: each matrix output by the setup algorithm is built using a different $2n$-vector, sampled from a different input space corresponding to the choice of key for that matrix. This in turn allows instantiating the hinting PRG scheme from all concrete assumptions that give rise to (γ-bounded) IHwUFs, including the ones with key-dependent input space such as QR and DCR (see Section 3.4).

5 Primitives from IHwPRF

In this section, we present constructions of various cryptographic primitives from (γ-bounded) IHwPRFs. Once again, none of these constructions require the input or output groups of the IHwPRF to be abelian.

5.1 Private Information Retrieval

A (single-database) private information retrieval (PIR) scheme is a two-party protocol between a sender and a receiver. The sender holds a public database (say, for concreteness, a string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$), and the receiver wishes to query an item in the database (say, the bit s_i for some $i \in [n]$) without revealing which item was queried (that is, i is not revealed to the sender). Note that in this model, the database is public, which implies that the unqueried items/bits need not be hidden from the receiver. A trivial solution is where the sender sends $\mathbf{s}$ to the receiver in the clear, which of course preserves receiver privacy. The total communication in such a protocol, measured as the number of bits exchanged between the sender and the receiver, is n. A *non-trivial* PIR protocol is one that securely achieves the aforementioned functionality with communication strictly smaller than n bits, where n is the size of the database. Black-box constructions of PIR protocols are known from different assumptions, e.g., group-homomorphic encryption [KO97], smooth subgroup assumptions [CMS99, GR05], and trapdoor permutations [KO00]

As a warm-up, we first demonstrate an inefficient PIR protocol that has a communication overhead of $O(n \cdot \ell(\lambda))$ bits, where ℓ is the maximum number of bits needed to encode a group element in either $\mathcal{X}$ or $\mathcal{Y}$. While this is even worse than the trivial protocol, we subsequently show how the efficiency of this protocol may be boosted to achieve a non-trivial PIR protocol, without any additional assumptions.

Inefficient PIR from IHwPRFs. Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an IHwPRF:

1. On input an index $i \in [n]$, the receiver uniformly samples 2 elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ and $k \leftarrow \mathcal{K}$. It also samples a uniform $\tilde{y}$ in $\mathcal{Y}$ subject to the restriction that $\tilde{y}$ is not the identity element of $\mathcal{Y}$. It then sets the following

$$\begin{aligned} y_{i,0} &= F(k, x_{i,0}) \\ y_{i,1} &= F(k, x_{i,1}) \otimes \tilde{y} \\ y_{j,b} &= F(k, x_{j,b}) \text{ for } j \in [n] \setminus \{i\}, b \in \{0,1\} \end{aligned}$$

   and sends $\left(\{x_{j,b}, y_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$ to the sender.

2. The sender, on input a string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and $\left(\{x_{j,b}, y_{j,b}\}_{j\in[n],b\in\{0,1\}}\right)$, sends (x^*, y^*) to the receiver where

$$(x^*, y^*) = \left(\bigoplus_{j\in[n]} x_{j,s_j}, \bigotimes_{j\in[n]} y_{j,s_j}\right).$$

3. The receiver retrieves the bit s_i as

$$s_i = \begin{cases} 0 & \text{if } y^* = F(k, x^*) \\ 1 & \text{otherwise.} \end{cases}$$

Boosting Efficiency. We now apply a generic efficiency-boosting technique introduced in [KO97] to convert the inefficient protocol into a PIR protocol with a communication overhead of $O(\sqrt{n} \cdot \ell(\lambda))$ bits. Quite evidently, such a PIR protocol is non-trivial in the sense that the overall communication complexity is strictly smaller than n bits for sufficiently large n. The idea is to view the database string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ as a binary matrix $\mathbf{S} \in \{0,1\}^{\sqrt{n}\times\sqrt{n}}$ such that:

$$\mathbf{S}_{j_1,j_2} = s_{(j_1-1)\sqrt{n}+j_2} \text{ for } j_1, j_2 \in [\sqrt{n}]$$

The receiver now sends across only $2\sqrt{n}$ group elements in its first message to the sender, as opposed to $2n$ in the inefficient protocol, while the receiver performs $\sqrt{n}$ "subset-sum" operations over these elements (one per column of the matrix $\mathbf{S}$) and sends back $2\sqrt{n}$ group elements to the receiver. The detailed protocol is as follows:

1. On input an index $i \in [n]$, the receiver uniformly samples $2\sqrt{n}$ elements from $\mathcal{X}$ as $\{x_{j_1,b} \leftarrow \mathcal{X}\}_{j_1 \in [\sqrt{n}], b \in \{0,1\}}$ and $k \leftarrow \mathcal{K}$. It also samples a uniform $\tilde{y}$ from $\mathcal{Y}$ subject to restriction that $\tilde{y}$ is not the identity element of $\mathcal{Y}$. Let $i_1 = \lceil i/\sqrt{n} \rceil$. The receiver sets the following

$$\begin{aligned} y_{i_1,0} &= F\left(k, x_{i_1,0}\right) \\ y_{i_1,1} &= F\left(k, x_{i_1,1} \oplus \tilde{x}\right) \\ y_{j_1,b} &= F\left(k, x_{j_1,b}\right) \text{ for } j_1 \in [\sqrt{n}] \setminus \{i_1\}, b \in \{0,1\} \end{aligned}$$

   and sends $\left(\{x_{j_1,b}, y_{j_1,b}\}_{j_1 \in [\sqrt{n}], b \in \{0,1\}}\right)$ to the sender.

2. The sender, on input a string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and $\left(\{x_{j_1,b}, y_{j_1,b}\}_{j_1 \in [\sqrt{n}], b \in \{0,1\}}\right)$, creates a binary matrix $\mathbf{S} \in \{0,1\}^{\sqrt{n} \times \sqrt{n}}$ where

$$\mathbf{S}_{j_1,j_2} = s_{\sqrt{n}(j_1-1)+j_2} \text{ for } j_1, j_2 \in [\sqrt{n}].$$

   It then sends $\left(\{x^*_{j_2}, y^*_{j_2}\}_{j_2 \in [\sqrt{n}]}\right)$ to the receiver where

$$\begin{aligned} x^*_{j_2} &= \bigoplus_{j_1 \in [\sqrt{n}]} x_{j_1, \mathbf{S}_{j_1,j_2}} \text{ for } j_2 \in [\sqrt{n}] \\ y^*_{j_2} &= \bigotimes_{j_1 \in [\sqrt{n}]} y_{j_1, \mathbf{S}_{j_1,j_2}} \text{ for } j_2 \in [\sqrt{n}]. \end{aligned}$$

3. The receiver computes $i_2 = i \bmod \sqrt{n}$ and retrieves the bit s_i as

$$s_i = \begin{cases} 0 & \text{if } y^*_{i_2} = F\left(k, x^*_{i_2}\right) \\ 1 & \text{otherwise.} \end{cases}$$

Note 5.1. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwPRF if $\gamma \geq n$, with the following minor modification to the final step:

$$s_i = \begin{cases} 0 & \text{if } \mathcal{R}\left(y^*_{i_2}\right) = \mathcal{R}\left(F\left(k, x^*_{i_2}\right)\right) \\ 1 & \text{otherwise.} \end{cases}$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.4).

Note 5.2. The aforementioned construction can also be instantiated using a (γ-bounded) IHwPRF family for which the input space is dependent of the choice of key. In particular, since the receiver chooses the PRF key, it can set up the input space accordingly, and sample a random $2n$-vector of elements from this space. This in turn allows instantiating the PIR scheme from all concrete assumptions that give rise to (γ-bounded) IHwPRFs, including the ones with key-dependent input space (see Section 3.4).

Instantiation from General Protocol. Let $|\mathrm{DB}|$ denote the size of the database string in the PIR scheme. This scheme can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, a fixed $n = \sqrt{|\mathrm{DB}|}$ such that $n > 3 \log |\mathcal{X}|$, $N^* = 0$, $\bar{N} = 1$ and $\hat{N} = \sqrt{|\mathrm{DB}|}$.
- **Evaluation:** Let $\mathrm{Y} = \{y_{j,b}\}_{j \in [\sqrt{|\mathrm{DB}|}], b \in \{0,1\}}$ be the output of the evaluation phase. For a given $i \in [|\mathrm{DB}|]$, set $i_1 = \lceil i / \sqrt{|\mathrm{DB}|} \rceil$ and reset $y_{i_1,1} := y_{i_1,1} \otimes \tilde{y}$, where $\tilde{y}$ is a uniform non-identity element in $\mathcal{Y}$. Set the first message of the receiver to the sender as (X, Y).
- **Post-Evaluation:** Recall that in the protocol, the post-evaluation phase samples $\hat{n} = \sqrt{|\mathrm{DB}|}$ binary strings, each of size $n = \sqrt{|\mathrm{DB}|}$. One may view this as a random binary matrix of size $\sqrt{|\mathrm{DB}|} \times \sqrt{|\mathrm{DB}|}$. Consequently, if $\hat{\mathrm{X}}$ and $\hat{\mathrm{Y}}$ are the outputs of the post-evaluation phase, set the message from the sender to the receiver as $(\hat{\mathrm{X}}, \hat{\mathrm{Y}})$.

Correctness and security of the scheme follow from Claim 3.10 and Theorem 3.12, respectively.

5.2 Lossy Trapdoor Functions

We begin with the formal definition of a lossy trapdoor function family from [PW08], and then we show how to construct lossy TDF family from IHwPRFs.

Definition 5.3. (Lossy Trapdoor Function.) A lossy trapdoor function family is a tuple of four PPT algorithms $\mathsf{LTDF} = (\mathsf{GenInjective}, \mathsf{GenLossy}, \mathsf{Eval}, \mathsf{Invert})$ defined as follows:

- $\mathsf{GenInjective}\left(1^\lambda\right)$: Given the security parameter λ, the algorithm outputs the public parameter pp for an *injective* function, along with a trapdoor t.
- $\mathsf{GenLossy}\left(1^\lambda\right)$: Given the security parameter λ, the algorithm outputs the public parameter pp for a *lossy* function. It does not produce a trapdoor. (See below for a formal definition of lossiness.)
- $\mathsf{Eval}\left(\mathsf{pp}, \mathbf{s}\right)$: Given the public parameter pp and a preimage string $\mathbf{s} \in \{0,1\}^n$ (where $n = n(\lambda)$ is included in pp), the evaluation algorithm outputs the corresponding image h.
- $\mathsf{Invert}\left(\mathsf{t}, h\right)$: Given the trapdoor t and an image h, the inversion algorithm outputs $\mathbf{s}' \in \{0,1\}^n$.

The following completeness and security properties must be satisfied:

- **Completeness:** If $(\mathsf{pp}, \mathsf{t}) \leftarrow \mathsf{GenInjective}\left(1^\lambda\right)$, then for all preimage strings $\mathbf{s} \in \{0,1\}^n$, it holds with overwhelming probability over the random coins of $\mathsf{GenInjective}$ that [1]

$$\mathsf{Invert}\left(\mathsf{t}, h = \mathsf{Eval}\left(\mathsf{pp}, \mathbf{s}\right)\right) = \mathbf{s}.$$

- **One-Wayness without Trapdoors:** For any PPT adversary $\mathcal{A}$ we have

$$\Pr[\mathsf{Eval}(\mathsf{pp}, \mathcal{A}(h)) = h] \leq \mathrm{negl}(\lambda),$$

  where $\mathsf{pp} \leftarrow \mathsf{GenInjective}(1^\lambda)$, $\mathbf{s} \leftarrow \{0,1\}^n$ and $h = \mathsf{Eval}(\mathsf{pp}, \mathbf{s})$.

- **Lossiness:** A TDF family $(\mathsf{GenInjective}, \mathsf{GenLossy}, \mathsf{Eval}, \mathsf{Invert})$ is said to be ε-lossy if for any *unbounded* adversary $\mathcal{A}$ we have $\Pr[\mathcal{A}(h) = \mathbf{s}] \leq \varepsilon$ where $\mathsf{pp} \leftarrow \mathsf{GenLossy}(1^\lambda)$, $\mathbf{s} \leftarrow \{0,1\}^n$ and $h = \mathsf{Eval}(\mathsf{pp}, \mathbf{s})$.
- **Indistinguishability of Modes:** For any PPT adversary $\mathcal{A}$ we have

$$\left| \Pr[\mathcal{A}(\mathsf{pp}_0) = 1] - \Pr[\mathcal{A}(\mathsf{pp}_1) = 1] \right| \leq \mathrm{negl}(\lambda),$$

  where $\mathsf{pp}_0 \leftarrow \mathsf{GenInjective}(1^\lambda)$ and $\mathsf{pp}_1 \leftarrow \mathsf{GenLossy}(1^\lambda)$.

[1] Note that if a string h does not lie in the image space of the TDF, then the output behavior of $\mathsf{Invert}\left(\mathsf{t}, h\right)$ is unspecified. Hence, in certain applications, one may need to verify the output of the inversion algorithm on a random image string h.

Construction from IHwPRFs. We present a black-box construction of a lossy TDF family from any IHwPRF family. The construction is inspired by the DDH-based lossy TDF family proposed by Peikert and Waters in [PW08].

- $\mathsf{GenInjective}\left(1^{\lambda}\right)$: In the injective mode, the algorithm creates a description $\mathcal{F}_{\mathrm{IHwPRF}}$ for an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, fixes some $n = n(\lambda) > 3\log|\mathcal{X}|$ and samples $2n$ uniform elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$. It samples n uniform keys as $\{k_i \leftarrow \mathcal{K}\}_{i\in[n]}$ and n uniform *non-identity* elements from $\mathcal{Y}$ as $\{\mathsf{m}_i \leftarrow \mathcal{Y} \setminus \{y_{\mathrm{id}}\}\}_{i\in[n]}$, and sets the following:

$$\begin{aligned} y_{i,i,0} &= F\left(k_i, x_{i,0}\right) \text{ for } i \in [n] \\ y_{i,i,1} &= F\left(k_i, x_{i,1}\right) \otimes \mathsf{m}_i \text{ for } i \in [n] \\ y_{i,j,b} &= F\left(k_i, x_{j,b}\right) \text{ for } i, j \in [n], i \neq j, b \in \{0,1\}\,. \end{aligned}$$

  It outputs the public parameter pp and the trapdoor t as

$$\begin{aligned} \mathsf{pp} &= \left(\mathcal{F}_{\mathrm{IHwPRF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, \{y_{i,j,b}\}_{i,j\in[n],b\in\{0,1\}}\right) \\ \mathsf{t} &= \left(\{k_i\}_{i\in[n]}\right). \end{aligned}$$

- $\mathsf{GenLossy}\left(1^{\lambda}\right)$: The algorithm creates a description $\mathcal{F}_{\mathrm{IHwPRF}}$ for an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, and fixes some integer $n = n(\lambda) > 3\log|\mathcal{X}|$. It samples $2n$ uniform elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ and n uniform keys as $\{k_i \leftarrow \mathcal{K}\}_{i\in[n]}$, and sets the following

$$y_{i,j,b} = F\left(k_i, x_{j,b}\right) \text{ for } i, j \in [n], b \in \{0,1\}\,.$$

  It outputs the public parameter pp as

$$\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHwPRF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, \{y_{i,j,b}\}_{i,j\in[n],b\in\{0,1\}}\right).$$

- $\mathsf{Eval}\left(\mathsf{pp}, \mathbf{s}\right)$: Given $\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHwPRF}}, \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, \{y_{i,j,b}\}_{i,j\in[n],b\in\{0,1\}}\right)$ and $\mathbf{s} \in \{0,1\}^n$, the evaluation algorithm computes

$$\begin{aligned} x^* &= \bigoplus_{j\in[n]} x_{j,s_j} \\ y_i^* &= \bigotimes_{j\in[n]} y_{i,j,s_j} \text{ for } i \in [n]. \end{aligned}$$

  It then outputs the image $h = \left(x^*, \{y_i^*\}_{i\in[n]}\right)$.

- $\mathsf{Invert}\left(\mathsf{t}, h\right)$: Given the trapdoor $\mathsf{t} = \left(\{k_i\}_{i\in[n]}\right)$ and $h = \left(x^*, \{y_i^*\}_{i\in[n]}\right)$, the inversion algorithm recovers the preimage bit s_i for each $i \in [n]$ as:

$$s_i = \begin{cases} 0 & \text{if } y_i^* = F\left(k_i, x^*\right) \\ 1 & \text{otherwise.} \end{cases}$$

  Finally, it outputs the recovered string $\mathbf{s} = (s_1, \ldots, s_n)$.

Instantiation from General Protocol. The lossy TDF family described above can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with description $\mathcal{F}_{\text{IHwPRF}}$, a fixed n such that $n > 3 \log |\mathcal{X}|$, $N^* = 0$, $\bar{N} = n$ and $\hat{N} = 1$.
- **Evaluation:** Let $\mathrm{Y} = \left\{ y_{j,b}^{(1)}, \ldots, y_{j,b}^{(n)} \right\}_{j \in [n], b \in \{0,1\}}$ be the tuple output by the evaluation phase. Sample a uniform *non-identity* element m from $\mathcal{Y}$.
  - In the lossy mode, output $\mathsf{pp} = (\mathcal{F}_{\text{IHwPRF}}, \mathrm{X}, \mathrm{Y})$.
  - In the injective mode, reset $y_{i,1}^{(i)} := y_{i,1}^{(i)} \otimes \mathsf{m}$ for each $i \in [n]$ and output $\mathsf{pp} = (\mathcal{F}_{\text{IHwPRF}}, \mathrm{X}, \mathrm{Y})$.
- **Post-Evaluation:** Recall that in the protocol, the post-evaluation phase uniformly samples a binary string $\mathbf{s} \in \{0,1\}^n$. One may view this as an input to the lossy TDF. Consequently, if $\hat{\mathrm{X}}$ and $\hat{\mathrm{Y}}$ are the outputs of the post-evaluation phase, set the evaluation output of the lossy TDF as $(\hat{\mathrm{X}}, \hat{\mathrm{Y}})$.

To see that the instantiation satisfies the properties of a lossy TDF family, consider the following:

- Completeness in the injective mode follows from Claim 3.10.
- One-wayness without trapdoors follows from Lemma 3.7.
- Indistinguishability of modes follows from Theorem 3.12.
- To argue lossiness, let $h = (x^*, \{y_i^*\}_{i \in [n]})$ be the output of the evaluation algorithm on input $\mathbf{s} \in \{0,1\}^n$ in the lossy mode. By Claim 3.10, for each $i \in [n]$ we have $y_i^* = F(k_i, x^*)$ where x^* is a function of the input $\mathbf{s}$, and k_i is fixed by the public parameter pp. Therefore, the number of possible outputs in the lossy mode is upper bounded by $|\mathcal{X}|$, while the number of possible inputs is 2^n for some fixed $n > 3 \log |\mathcal{X}|$. Hence, the "lossiness" of the TDF family is at least $n - \log |\mathcal{X}| \geq 2 \log |\mathcal{X}|$.

Implications. The following are some implications of lossy TDFs.

- **CCA-Secure PKE.** Peikert and Waters [PW08] showed that CCA2-secure PKE can be constructed from any lossy TDF family. Their construction uses a primitive called All-But-One TDF (which can be built from any lossy TDF family). The decryption algorithm in the resulting PKE is witness recovering. Since (γ-bounded) IHwPRFs imply lossy TDF family, it immediately follows that (γ-bounded) IHwPRFs are sufficient to construct CCA2-secure PKE.
- **Selective Opening Attack (SOA)-Secure PKE.** Bellare *et al.* [BHY09] showed that PKE schemes that are secure against selective opening attacks can be constructed from any lossy TDF family. [1] It follows that (γ-bounded) IHwPRFs are sufficient to construct SOA-secure PKE.
- **Non-Interactive Statistically Binding Commitments.** Suppose that LTDF be a lossy trapdoor function family. Then $\mathsf{LTDF} = (\mathsf{GenInjective}, \mathsf{GenLossy}, \mathsf{Eval}, \mathsf{Invert})$ yields a non-interactive statistically binding commitment scheme as follows:
  - Commitment: To commit to a string $\mathbf{s} \in \{0,1\}^n$, the committer samples $(\mathsf{pp}, \mathsf{t}) \leftarrow \mathsf{GenInjective}\left(1^\lambda\right)$ and outputs $(\mathsf{pp}, h = \mathsf{Eval}(\mathsf{pp}, \mathbf{s}))$.
  - Open: To decommit, the committer outputs $\mathbf{s}'$.
  - Verification: Given a commitment (pp, h) and a decommitment $\mathbf{s}'$, check if $h = \mathsf{Eval}(\mathsf{pp}, \mathbf{s}')$.

[1] See [BHY09] for the details of the construction.

Statistical binding follows from the fact that pp sampled using GenInjective describes a family of injective one-way functions, while computational hiding follows from the "one-wayness without trapdoors" property of any lossy TDF in the injective mode. It follows that (γ-bounded) IHwPRFs are sufficient to construct non-interactive statistically binding commitment schemes.

- **Three-Round Non-Malleable Commitments.** Goyal *et al.* [GPR16] showed that any non-interactive statistically binding commitment scheme yields a black-box construction of non-malleable commitments with three rounds of interaction. Hence, any (γ-bounded) IHwPRF family implies a three-round non-malleable commitment scheme.

Note 5.4. The aforementioned construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated equivalently using a γ-bounded IHwPRF if $\gamma \geq n$, with the following minor modification to the inversion algorithm:

$$s_i = \begin{cases} 0 & \text{if } \mathcal{R}(y_i^*) = \mathcal{R}(F(k_i, x^*)) \\ 1 & \text{otherwise.} \end{cases}$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.4).

Note 5.5. The aforementioned lossy TDF construction can be instantiated using an (γ-bounded) IHwPRF family for which the input space is dependent of the choice of key with the following minor modification to the setup algorithms: in both the lossy and the injective modes, each column of the $\mathcal{Y}$-matrix output by the setup algorithm is built using a different $2n$-vector, sampled from a different input space corresponding to the choice of key for that column.

5.3 Oblivious Transfer and Multi-Party Computation

In this subsection we show how to construct maliciously secure OT[1] and MPC in the plain model from IHwPRFs. We use a recent result of Friolo *et al.* [FMV18] in which they showed that: [2]

1. A CPA-secure PKE with pseudorandom public keys (i.e., the distribution of public keys should be computationally indistinguishable from the uniform distribution over an efficiently samplable group) implies (in a black-box way) a two-round *strongly uniform* key-exchange protocol, where the distribution of messages sent by one of the parties is computationally indistinguishable from the uniform distribution over an efficiently samplable group, even when the other party is malicious.

2. For any $t \in \mathbb{N}$, a t-round strongly uniform secure key-exchange protocol is black-box equivalent to a t-round strongly uniform semi-honestly secure OT protocol in the plain model, where the distribution of all the messages sent by the receiver are computationally indistinguishable from the uniform distribution over an efficiently samplable group, even when the sender is malicious.

3. For any odd $t \in \mathbb{N}$, a t-round strongly uniform semi-honestly secure OT protocol in the plain model, together with a non-interactive statistically binding commitment scheme, implies (in a black-box manner) a $(t+1)$-round maliciously secure OT protocol in the plain model.

A recent breakthrough result of Benhamouda and Lin [BL18] showed that for any $t \in \mathbb{N}$ such that $t \geq 5$, a t-round fully maliciously secure OT protocol implies a t-round fully maliciously secure MPC protocol.

[1] One can construct a 2-round semi-honest OT in the plain model using lossy TDFs [BL18], which can be based on IHwPRFs.
[2] For the sake of succinctness, we state the results informally. See [FMV18] for the formal description.

Construction from IHwPRFs. We now demonstrate that the framework of Friolo *et al.* [FMV18] can be instantiated using any IHwPRF family.

- **PKE with pseudorandom public keys.** We use an IHwPRF (instead of an IHwUF) in the construction of 4.2 to get a PKE with pseudorandom public keys. As a result, pk and sk for the modified scheme will have the form

$$\mathsf{sk} = k, \quad \mathsf{pk} = \{(x_{j,b}, y_{j,b} = F(k, x_{j,b}))\}_{j\in[n],b\in\{0,1\}},$$

  where $y_{j,b}$ are the evaluations of the IHwPRF. It is easy to see that the new scheme is still CPA-secure. In addition, any PPT algorithm that can efficiently distinguish the public key in the modified scheme from a unifomly random tuple in $(\mathcal{X} \times \mathcal{Y})^{2n}$ breaks the weak pseudorandomness of F.

- **Non-interactive statistically binding commitments.** We demonstrated in Section 5.2 that any IHwPRF family implies a non-interactive statistically binding commitment scheme.

Combining these observations with the results of Friolo et al. and Benhamouda et al. leads to the following implications:

- Any IHwPRF implies a 4-round maliciously secure OT protocol in the plain model.
- Any IHwPRF implies a 5-round maliciously secure MPC protocol in the plain model.

References

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *29th ACM STOC*, pages 284–293. ACM Press, May 1997.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[AMG07] C. Aguilar-Melchor and P. Gaborit. A lattice-based computationally-efficient private information retrieval protocol. In *Western European Workshop on Research in Cryptology*. Citeseer, 2007.

[AS15] G. Asharov and G. Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In V. Guruswami, editor, *56th FOCS*, pages 191–209. IEEE Computer Society Press, October 2015.

[Bar17] B. Barak. The complexity of public-key cryptography. Cryptology ePrint Archive, Report 2017/365, 2017. https://eprint.iacr.org/2017/365.

[BBF13] P. Baecher, C. Brzuska, and M. Fischlin. Notions of black-box reductions, revisited. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 296–315. Springer, Heidelberg, December 2013.

[BBS04] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, August 2004.

[BDRV18] I. Berman, A. Degwekar, R. D. Rothblum, and P. N. Vasudevan. From laconic zero-knowledge to public-key cryptography - extended abstract. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 674–697. Springer, Heidelberg, August 2018.

[BDV17] N. Bitansky, A. Degwekar, and V. Vaikuntanathan. Structure vs. hardness through the obfuscation lens. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 696–723. Springer, Heidelberg, August 2017.

[BF01] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

[BGI+01] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[BH08] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 455–470. Springer, Heidelberg, December 2008.

[BHY09] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35. 2009.

[BL18] F. Benhamouda and H. Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In *EUROCRYPT*, pages 500–532. 2018.

[BLMR13] D. Boneh, K. Lewi, H. W. Montgomery, and A. Raghunathan. Key homomorphic PRFs and their applications. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Heidelberg, August 2013.

[BLSV18] Z. Brakerski, A. Lombardi, G. Segev, and V. Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 535–564. Springer, Heidelberg, April / May 2018.

[BM82] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd FOCS*, pages 112–117. IEEE Computer Society Press, November 1982.

[BR17] A. Bogdanov and A. Rosen. Pseudorandom functions: Three decades later. Cryptology ePrint Archive, Report 2017/652, 2017. https://eprint.iacr.org/2017/652.

[BSW11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

[BV96] D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 129–142. 1996.

[CA89] D. Chaum and H. V. Antwerpen. Undeniable signatures. In *CRYPTO*, pages 212–216. 1989.

[CGGM00] R. Canetti, O. Goldreich, S. Goldwasser, and S. Micali. Resettable zero-knowledge (extended abstract). In *STOC*, pages 235–244. 2000.

[CGW15] J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015.

[CMS99] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT*, pages 402–414. 1999.

[Coc01] C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, Heidelberg, December 2001.

[CS02] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64. 2002.

[DG17a] N. Döttling and S. Garg. From selective IBE to full IBE and selective HIBE. In Y. Kalai and L. Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 372–408. Springer, Heidelberg, November 2017.

[DG17b] N. Döttling and S. Garg. Identity-based encryption from the Diffie-Hellman assumption. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 537–569. Springer, Heidelberg, August 2017.

[DGHM18] N. Döttling, S. Garg, M. Hajiabadi, and D. Masny. New constructions of identity-based and key-dependent message secure encryption schemes. In M. Abdalla and R. Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 3–31. Springer, Heidelberg, March 2018.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DHP+18] Y. Doröz, J. Hoffstein, J. Pipher, J. H. Silverman, B. Sunar, W. Whyte, and Z. Zhang. Fully homomorphic encryption from the finite field isomorphism problem. In *PKC*, pages 125–155. 2018.

[DHS15] D. Derler, C. Hanser, and D. Slamanig. Revisiting cryptographic accumulators, additional properties and relations to other primitives. In K. Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 127–144. Springer, Heidelberg, April 2015.

[EHK+13] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

[ElG84] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.

[ER65] P. Erdös and A. Rényi. Probabilistic methods in group theory. *Journal d'Analyse Mathématique*, 14(1):127–138, 1965.

[FH18] M. Fischlin and P. Harasser. Invisible sanitizable signatures and public-key encryption are equivalent. In B. Preneel and F. Vercauteren, editors, *ACNS 18*, volume 10892 of *LNCS*, pages 202–220. Springer, Heidelberg, July 2018.

[Fis12] M. Fischlin. Black-box reductions and separations in cryptography (invited talk). In A. Mitrokotsa and S. Vaudenay, editors, *AFRICACRYPT 12*, volume 7374 of *LNCS*, pages 413–422. Springer, Heidelberg, July 2012.

[FMV18] D. Friolo, D. Masny, and D. Venturi. Secure multi-party computation from strongly uniform key agreement. Cryptology ePrint Archive, Report 2018/473, 2018. http://eprint.iacr.org/.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGH13a] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.

[GGH+13b] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH+13c] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499. Springer, Heidelberg, August 2013.

[GGH18] S. Garg, R. Gay, and M. Hajiabadi. New techniques for efficient trapdoor functions and applications. Cryptology ePrint Archive, Report 2018/872, 2018. http://eprint.iacr.org/.

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

[GH18] S. Garg and M. Hajiabadi. Trapdoor functions from the computational diffie-hellman assumption. In *CRYPTO*, pages 362–391. 2018.

[GHMM18] S. Garg, M. Hajiabadi, M. Mahmoody, and A. Mohammed. Limits on the power of garbling techniques for public-key encryption. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 335–364. Springer, Heidelberg, August 2018.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.

[GPR16] V. Goyal, O. Pandey, and S. Richelson. Textbook non-malleable commitments. In *STOC*. 2016.

[GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

[GPSZ17] S. Garg, O. Pandey, A. Srinivasan, and M. Zhandry. Breaking the sub-exponential barrier in obfustopia. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 156–181. Springer, Heidelberg, April / May 2017.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GR05] C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In *ICALP*, pages 803–815. 2005.

[GW11] C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In L. Fortnow and S. P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

[HHRS07] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th FOCS*, pages 669–679. IEEE Computer Society Press, October 2007.

[HILL99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HK07] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, Heidelberg, August 2007.

[HKS16] M. Hajiabadi, B. M. Kapron, and V. Srinivasan. On generic constructions of circularly-secure, leakage-resilient public-key encryption schemes. In C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang, editors, *PKC 2016, Part II*, volume 9615 of *LNCS*, pages 129–158. Springer, Heidelberg, March 2016.

[HO12] B. Hemenway and R. Ostrovsky. On homomorphic encryption and chosen-ciphertext security. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 52–65. Springer, Heidelberg, May 2012.

[How01] N. Howgrave-Graham. Approximate integer common divisors. In *Cryptography and Lattices, International Conference (CaLC)*, pages 51–66. 2001.

[IKLP06] Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank. Black-box constructions for secure computation. In J. M. Kleinberg, editor, *38th ACM STOC*, pages 99–108. ACM Press, May 2006.

[IKO05] Y. Ishai, E. Kushilevitz, and R. Ostrovsky. Sufficient conditions for collision-resistant hashing. In J. Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 445–456. Springer, Heidelberg, February 2005.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. June 1995. ISSN 1063-6870.

[IZ89] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *30th FOCS*, pages 248–253. IEEE Computer Society Press, October / November 1989.

[JR13] C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2013.

[KO97] E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *FOCS*, pages 364–373. 1997.

[KO00] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *EUROCRYPT*, pages 104–121. 2000.

[KR00] H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS*. 2000.

[KW15] E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015.

[KW18] V. Koppula and B. Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. Cryptology ePrint Archive, Report 2018/847, 2018. http://eprint.iacr.org/.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

[Mau09] U. M. Maurer. Unifying zero-knowledge proofs of knowledge. In B. Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Heidelberg, June 2009.

[MM16] M. Mahmoody and A. Mohammed. On the power of hierarchical identity-based encryption. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 243–272. Springer, Heidelberg, May 2016.

[MMN+16] M. Mahmoody, A. Mohammed, S. Nematihaji, R. Pass, and A. Shelat. Lower bounds on assumptions behind indistinguishability obfuscation. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 49–66. Springer, Heidelberg, January 2016.

[OK93] W. Ogata and K. Kurosawa. On claw free families. In H. Imai, R. L. Rivest, and T. Matsumoto, editors, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 111–123. Springer, Heidelberg, November 1993.

[OSV15] R. Ostrovsky, A. Scafuro, and M. Venkitasubramaniam. Resettably sound zero-knowledge arguments from OWFs - the (semi) black-box way. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 345–374. Springer, Heidelberg, March 2015.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238. 1999.

[PS00] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[PS08] K. Pietrzak and J. Sjödin. Weak pseudorandom functions in minicrypt. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 423–436. Springer, Heidelberg, July 2008.

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

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Rom90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

[Rot11] R. Rothblum. Homomorphic encryption: From private-key to public-key. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 219–234. Springer, Heidelberg, March 2011.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RTV04] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, Heidelberg, February 2004.

[Sha07] H. Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. http://eprint.iacr.org/2007/074.

[SW05] A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

[Wat09] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, August 2009.

A Homomorphic One-Way Functions

Definition A.1. (Homomorphic One-Way Function.) A homomorphic one-way function (HOWF) f over an input group $(\mathcal{X}, \oplus)$ and an output group $(\mathcal{Y}, \otimes)$ is a one-way function $f : \mathcal{X} \to \mathcal{Y}$ such that for any $x_1, x_2 \in \mathcal{X}$, we have $f(x_1 \oplus x_2) = f(x_1) \otimes f(x_2)$.

We also consider a notion of *bounded* homomorphism, in the sense that input-homomorphism is preserved for an apriori bounded number of group operations in the source group of the OWF. We formally describe this notion as γ-bounded homomorphism, where the parameter γ reflects the maximum number of group operations that the homomorphism can tolerate.

Definition A.2. (γ-Bounded Homomorphic OWF.) A γ-bounded homomorphic one-way function (γ-bounded HOWF) over an input group $(\mathcal{X}, \oplus)$ and an output group[1] $(\mathcal{Y}, \otimes)$ is a one-way function $f : \mathcal{X} \to \mathcal{Y}$ if for any L-length input vector $(x_1, \ldots, x_L) \in \mathcal{X}^L$, we have:

$$f\left(\bigoplus_{j\in[L]} x_j\right) = \bigotimes_{j\in[L]} f(x_j)$$

subject to the restriction that $L \leq \gamma$.

Instantiations from Cryptographic Assumptions. It is easy to see that the following assumptions yields HOWF family. Specifically:

- The function family defined by $f_g(x) = g^x$ is an HOWF family based on discrete log assumption since $f_g(x_1 + x_2) = g^{x_1} \cdot g^{x_2} = f_g(x_1) \cdot f_g(x_2)$.
- Let $N = pq$ be an RSA modulus where p and q are equal-size prime numbers. The function family defined by $f_N(x) = x^2$ is an HOWF family based on square finding assumption since $f_N(x_1x_2) = f_N(x_1) \cdot f_N(x_2)$. (Operations are done modulo N)
- Let $N = pq$ be an RSA modulus as above, and let $e \leftarrow \mathbb{Z}^*_{\varphi(N)}$. The function family defined by $f_{N,e}(x) = x^e$ is an HOWF family based on RSA assumption since $f_{N,e}(x_1x_2) = f_{N,e}(x_1) \cdot f_{N,e}(x_2)$.

It is also easy to describe instantiations of (γ-bounded) HOWF family from assumptions other than the ones mentioned above. See Section 3.4 for more details.

In what follows, we present constructions of various cryptographic primitives from (γ-bounded) HOWFs, including collision-resistant hash functions (CRHFs), Schnorr signatures and chameleon hash functions. Constructing CRHFs (and Schnorr-like protocols) from structured primitives have been around for many years. Ogata and Kurosawa [OK93] demonstrated that *homomorphic one-way permutations* imply claw-free permutations and hence CRHFs. The authors of [IKO05] constructed CRHFs from homomorphic encryption and homomorphic one-way commitments. Maurer [Mau09] showed Schnorr-style zero-knowledge proof of knowledge protocols from (unbounded) HOWFs. For the constructions presented in this section, we explicitly describe how to instantiate them from both unbounded and bounded HOWFs. In addition, none of these constructions require the input or output groups of the HOWF to be abelian.

A.1 CRHF from HOWFs

In this subsection, we show that any HOWF induces a collision-resistant hash function family. Given any HOWF $f : \mathcal{X} \to \mathcal{Y}$, let $\mathbf{x} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ be a vector of $2n$ uniform elements for some fixed $n = n(\lambda)$. Define $\mathbf{y}$ as

$$\mathbf{y} = \{y_{j,b} = f(x_{j,b})\}_{j\in[n],b\in\{0,1\}}.$$

[1] We don't need $\mathcal{Y}$ to be a group. It is easy to see that the definition (and also applications) also work if image of $\mathcal{X}$ under f is a group. However, we assume $\mathcal{Y}$ to be a group for simplicity.

Now, define the function family $\mathcal{H}_{\mathbf{y}} : \{0,1\}^n \to \mathcal{Y}$ as

$$\mathcal{H}_{\mathbf{y}}\left(\mathbf{r} = (r_1, \cdots, r_n)\right) = \bigotimes_{j \in [n]} y_{j,r_j}.$$

Collision-Resistance. We now show that the aforementioned function family is collision-resistant. Let $f : \mathcal{X} \to \mathcal{Y}$ be an HOWF. For some fixed function $n = n(\lambda)$, define the experiment $\mathsf{Expt}^{\mathrm{CRHF-HOWF}}$ as in Figure 9.

1. The challenger uniformly samples $\{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}$.
2. The challenger sets $y_{j,b} = f(x_{j,b})$ for $j \in [n], b \in \{0,1\}$ and sends $\{y_{j,b}\}_{j \in [n], b \in \{0,1\}}$ to the adversary $\mathcal{A}$.
3. The adversary $\mathcal{A}$ outputs two bit-strings $\mathbf{r} = (r_1, \cdots, r_n) \in \{0,1\}^n$ and $\mathbf{r}' = (r'_1, \cdots, r'_n) \in \{0,1\}^n$.

Figure 9: Experiment for CRHF-HOWF security.

For any PPT adversary $\mathcal{A}$ we define $\mathbf{Adv}^{\mathrm{CRHF-HOWF}}(\mathcal{A})$ to be the probability of the event that

$$\bigoplus_{j \in [n]} y_{j,r_j} = \bigoplus_{j \in [n]} y_{j,r'_j}.$$

Lemma A.3. *For all PPT adversaries $\mathcal{A}$, we have $\mathbf{Adv}^{\mathrm{CRHF-HOWF}}(\mathcal{A}) = \mathrm{negl}(\lambda)$.*

Proof. Let $\mathcal{A}$ be a PPT adversary such that $\mathbf{Adv}^{\mathrm{CRHF-HOWF}}(\mathcal{A})$ is non-negligible. We construct a PPT algorithm $\mathcal{B}$ that breaks the one-wayness of f. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ receives a challenge query $y^* \in \mathcal{Y}$, such that $y^* = f(x^*)$ for some (uniformly random) $x^* \in \mathcal{X}$.
2. $\mathcal{B}$ samples $2n$ uniformly random elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}$ and sets $y_{j,b} = f(x_{j,b})$ for $j \in [n], b \in \{0,1\}$.
3. $\mathcal{B}$ uniformly randomly picks $i \leftarrow [n]$ and $b^* \leftarrow \{0,1\}$, and sets $y_{i,b^*} := y^*$. It then forwards $\{y_{j,b}\}_{j \in [n], b \in \{0,1\}}$ to the adversary $\mathcal{A}$.
4. $\mathcal{A}$ outputs $\mathbf{r} = (r_1, \cdots, r_n) \in \{0,1\}^n$ and $\mathbf{r}' = (r'_1, \cdots, r'_n) \in \{0,1\}^n$.
5. $\mathcal{B}$ proceeds as follows:
   - If $\bigotimes_{j \in [n]} y_{j,r_j} \neq \bigotimes_{j \in [n]} y_{j,r'_j}$ or $r_i = r'_i$, it outputs a uniformly random $x^* \leftarrow \mathcal{X}$.
   - Otherwise, assume wlog that $r_i = b^*$. Then, the following must hold

     $$y^* = \left(\bigotimes_{j \in [i-1]} y_{j,r_j}\right)^{-1} \otimes \left(\bigotimes_{j \in [n]} y_{j,r'_j}\right) \otimes \left(\bigotimes_{j \in [i+1,n]} y_{j,r_j}\right)^{-1}.$$

     where the right-hand side is independent of y^*. $\mathcal{B}$ now outputs x^* as

     $$x^* = \left(\bigoplus_{j \in [i-1]} x_{j,r_j}\right)^{-1} \oplus \left(\bigoplus_{j \in [n]} x_{j,r'_j}\right) \oplus \left(\bigoplus_{j \in [i+1,n]} x_{j,r_j}\right)^{-1}$$

     By the input-homomorphism of f, we have $f(x^*) = y^*$.

Observe that if $\mathcal{A}$ outputs a valid collision $(\mathbf{r}, \mathbf{r}')$, the probability that $\mathbf{r}$ and $\mathbf{r}'$ differ in the ith bit for a randomly chosen $i \leftarrow [n]$ is at least $1/n$. It follows that

$$\mathbf{Adv}^{\mathrm{HOWF}}(\mathcal{B}) \geq \left(\frac{\mathbf{Adv}^{\mathrm{CRHF-HOWF}}(\mathcal{A})}{n} \right),$$

which is non-negligible, as desired.

Note A.4. We implicitly assumed that the distribution of the input in the OWF game is uniform. For some constructions of HOWFs, this is not the case. With a sligh modification of the proof, one can show that HOWFs (in general) imply CRHFs. In addition, the aforementioned CRHF family may be equivalently instantiated from any γ-bounded HOWF family, subject to the restriction that $n \leq \gamma$.

A.2 Schnorr-style Digital Signature from HOWFs

We show how to construct a Schnorr-style signature scheme from any HOWF family. The signature scheme is existentially unforgeable against adaptive chosen-message attacks in the programmable random oracle model.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm samples a HOWF $f : \mathcal{X} \to \mathcal{Y}$. It also fixes some poly-bounded integer $n = n(\lambda)$. Finally, it chooses a hash function $H : \mathcal{Y} \times \{0,1\}^* \to \{0,1\}^n$ (modeled as a random oracle in the security proof). The algorithm outputs the public parameter pp as

$$\mathsf{pp} = (f, n, H).$$

- $\mathsf{Gen}(\mathsf{pp})$: The key-generation algorithm uniformly samples $2n$ elements in $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n],b\in\{0,1\}}$ and computes $y_{j,b} = f(x_{j,b})$ for $j \in [n]$ and $b \in \{0,1\}$. It outputs the signing key sk and the verification key vk as

$$\mathsf{sk} = \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}, \quad \mathsf{vk} = \{y_{j,b}\}_{j\in[n],b\in\{0,1\}}.$$

- $\mathsf{Sign}(\mathsf{sk}, \mathsf{m})$: Given the signing key $\mathsf{sk} = \{x_{j,b}\}_{j\in[n],b\in\{0,1\}}$ and a message $\mathsf{m} \in \{0,1\}^*$, the algorithm uniformly samples $x^* \leftarrow \mathcal{X}$ and sets $y^* = f(x^*)$. It then sets the vector $\mathbf{r} = (r_1, \ldots, r_n) \in \{0,1\}^n$ as $\mathbf{r} = H(y^*, \mathsf{m})$. Finally, it outputs the signature $\sigma = (\mathbf{r}, \hat{x}, y^*) \in \{0,1\}^n \times \mathcal{X} \times \mathcal{Y}$, where

$$\hat{x} = x^* \oplus \left(\bigoplus_{j\in[n]} x_{j,r_j} \right)^{-1}.$$

- $\mathsf{Ver}(\mathsf{vk}, \mathsf{m}, \sigma)$: Given the verification key $\mathsf{vk} = \{y_{j,b}\}_{j\in[n],b\in\{0,1\}}$, a message $\mathsf{m} \in \{0,1\}^*$ and a signature $\sigma = (\mathbf{r}, \hat{x}, y^*)$, where $\mathbf{r} = (r_1, \ldots, r_n) \in \{0,1\}^n$, the verification algorithm checks if both of the following conditions hold.

$$y^* = f(\hat{x}) \otimes \left(\bigotimes_{j\in[n]} y_{j,r_j} \right), \quad \mathbf{r} = H(y^*, \mathsf{m}).$$

  If yes, it validates the signature. Otherwise, it outputs $\perp$.

Correctness follows from the homomorphism of f. In order to prove existential unforgeability under an adaptively chosen-message attack in the programmable random oracle model, we resort to the forking lemma [PS00]. We first prove the following lemma.

Lemma A.5. *If H is modeled as a random oracle, there exists a PPT simulator $\mathcal{S}$ that produces, with non-negligible probability, a signature $\tilde{\sigma}$ on any arbitrary message m without the knowledge of the signing key sk such that the distribution of $\tilde{\sigma}$ is statistically indistinguishable from that of $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$.*

Proof. The simulator $\mathcal{S}$ receives the verification key $\mathsf{vk} = \{y_{j,b}\}_{j\in[n],b\in\{0,1\}}$ and proceeds as follows:

- The simulator $\mathcal{S}$ uniformly samples $\tilde{\mathbf{r}} = (\tilde{r}_1, \ldots, \tilde{r}_n) \leftarrow \{0,1\}^n$ and $\tilde{x} \leftarrow \mathcal{X}$.
- It sets $\tilde{y}^* = f(\tilde{x}) \otimes \left(\bigotimes_{j\in[n]} y_{j,\tilde{r}_j} \right)$ and returns the signature $\tilde{\sigma} = (\tilde{\mathbf{r}}, \tilde{x}, \tilde{y}^*)$.

Observe that in the simulation, we must have $\tilde{y}^* = f(\tilde{x}^*)$, where

$$\tilde{x}^* = \tilde{x} \oplus \left(\bigoplus_{j\in[n]} x_{j,\tilde{r}_j} \right).$$

Since $\tilde{x}$ is uniform in $\mathcal{X}$, so is $\tilde{x}^*$. Hence, the distribution of $(\tilde{x}, \tilde{y}^* = f(\tilde{x}^*))$ in the simulation is statistically indistinguishable from that of $(\hat{x}, y^* = f(x^*))$ in the "real" signing algorithm. Finally, under the assumption that H is a random oracle, the distribution of the string $\tilde{\mathbf{r}}$ in the simulation is also statistically indistinguishable from that of the string $\mathbf{r} = H(y^*, m)$ in the "real" signing algorithm. This completes the proof of Lemma A.5.

Let $\mathcal{A}$ be a PPT adversary that performs an existential forgery under an adaptively chosen-message attack against the aforementioned signature scheme with probability ε, while making $Q_1 = Q_1(\lambda)$ signing queries and $Q_2 = Q_2(\lambda)$ random oracle queries, such that $\varepsilon \geq 10(Q_1+1)(Q_1+Q_2)/2^\lambda$. Let (m, σ) be the message-signature pair corresponding to this forgery, where $\sigma = (\mathbf{r}, \hat{x}, y^*)$. As shown by Pointcheval and Stern in [PS00], Lemma A.5 implies the following "forking lemma".

Lemma A.6. *A poly-time replay of the adversary $\mathcal{A}$, where its interactions with the signing oracle are replaced by interactions with the simulator $\mathcal{S}$ as described above, produces with non-negligible probability two valid message-signature pairs*

$$(m, \sigma = (\mathbf{r}, \hat{x}, y^*)), \quad (m, \sigma' = (\mathbf{r}', \hat{x}', y^*))$$

on the same message m, such that $\mathbf{r} \neq \mathbf{r}'$, and hence, $\sigma \neq \sigma'$.

Finally, given a PPT adversary that forges a pair of non-identical signatures on the same message with non-negligible probability, one can construct a PPT adversary that induces collisions on the family of CRHFs described in Section A.1 with the same probability. This completes the proof of existential unforgeability for our signature scheme.

Note A.7. The aforementioned signature scheme has an apriori bounded number of homomorphic operations, which allows it to be instantiated using a γ-bounded HOWF family, subject to the restriction that $n \leq \gamma$.

A.3 Chameleon Hash Functions from HOWFs

We now show how to construct a chameleon hash function family from any HOWF. The formal definition of chameleon hash functions is presented below.

Definition A.8. (Chameleon Hash Functions.) A chameleon hash function family is defined as a tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{CHash}, \mathsf{TrpCollision})$ described below.

- $\mathsf{Setup}\left(1^\lambda\right)$: Given λ, it outputs the public parameter pp and a trapdoor t.
- $\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right)$: Given pp, a string $\mathbf{s} \in \{0,1\}^n$ (where $n = n(\lambda)$ is included in pp) and randomness r, it outputs a hash h.
- $\mathsf{TrpCollision}\left(\mathsf{t}, (\mathbf{s}, r), \mathbf{s}'\right)$: Given the trapdoor t, a string $\mathbf{s} \in \{0,1\}^n$, some randomness r, and a string $\mathbf{s}' \in \{0,1\}^n$, it outputs some randomness r'.

The following properties must be satisfied:

- **Uniformity:** If $(\mathsf{pp}, \mathsf{t}) \leftarrow \mathsf{Setup}\left(1^\lambda\right)$, then for all string pairs $\mathbf{s}, \mathbf{s}' \in \{0,1\}^n$ and all uniformly sampled randomness pairs (r, r'), the two distributions $\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right)$ and $\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}'; r'\right)$ are statistically indistinguishable.

- **Collision-Resistance:** For all PPT adversaries $\mathcal{A}$, letting $\mathsf{pp} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$ and $((\mathbf{s}, r), (\mathbf{s}', r')) \leftarrow \mathcal{A}(\mathsf{pp})$ (such that $(\mathbf{s}, r) \neq (\mathbf{s}', r')$), we have

$$\Pr[\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right) = \mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}'; r'\right)] \leq \mathrm{negl}(\lambda).$$

- **Trapdoor Collisions:** If $(\mathsf{pp}, \mathsf{t}) \leftarrow \mathsf{Setup}\left(1^\lambda\right)$, then for all $\mathbf{s}, \mathbf{s}' \in \{0,1\}^n$ and randomness r, it holds that

$$\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right) = \mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}'; r'\right),$$

  where $r' = \mathsf{TrpCollision}\left(\mathsf{pp}, (\mathbf{s}, r), \mathbf{s}'\right)$.

Construction from HOWFs. We construct a chameleon hash function family from any HOWF as follows.

- $\mathsf{Setup}\left(1^\lambda\right)$: The setup algorithm samples a HOWF $f : \mathcal{X} \to \mathcal{Y}$. It samples $2n$ elements from $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}$ for some polynomial $n = n(\lambda)$ and outputs the public parameter pp and a trapdoor t as:

$$\mathsf{pp} = \left(f, \{y_{j,b}\}_{j \in [n], b \in \{0,1\}}\right), \ \mathsf{t} = \{x_{j,b}\}_{j \in [n], b \in \{0,1\}}$$

  where $y_{j,b} = f\left(x_{j,b}\right)$ for $j \in [n]$ and $b \in \{0,1\}$.

- $\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right)$: Given the public parameter $\mathsf{pp} = \left(f, \{y_{j,b}\}_{j \in [n], b \in \{0,1\}}\right)$, a string $\mathbf{s} \in \{0,1\}^n$, and randomness $r \leftarrow \mathcal{X}$, the hashing algorithm outputs the hash h as

$$h = \left(\bigotimes_{j \in [n]} y_{j,s_j}\right) \otimes f\left(r\right).$$

- $\mathsf{TrpCollision}\left(\mathsf{t}, (\mathbf{s}, r), \mathbf{s}'\right)$: Given the trapdoor $\mathsf{t} = \{x_{j,b}\}_{j \in [n], b \in \{0,1\}}$, a string $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$, some randomness $r \in \mathcal{X}$, and another string $\mathbf{s}' = (s'_1, \ldots, s'_n) \in \{0,1\}^n$, the equivocation algorithm outputs $r' \in \mathcal{X}$ as

$$r' = \left(\bigoplus_{j \in [n]} x_{j,s'_j}\right)^{-1} \oplus \left(\bigoplus_{j \in [n]} x_{j,s_j}\right) \oplus r.$$

We now argue that the aforementioned construction satisfies the desired properties of a chameleon hash function.

- **Uniformity:** Let $\mathbf{s} = (s_1, \ldots, s_n) \in \{0,1\}^n$ and $\mathbf{s}' = (s'_1, \ldots, s'_n) \in \{0,1\}^n$ be arbitrary binary strings, and let $r, r' \leftarrow \mathcal{X}$ be uniformly random elements in $\mathcal{X}$. Let

$$x^* = \left(\bigoplus_{j \in [n]} x_{j,s_j}\right) \oplus r, \quad x'^* = \left(\bigoplus_{j \in [n]} x_{j,s'_j}\right) \oplus r'.$$

  It is easy to see that both x^* and x'^* are uniformly distributed over $\mathcal{X}$ so long as r and r' are uniform. This in turn implies that the distributions

$$\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right) = f\left(x^*\right), \quad \mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}'; r'\right) = f\left(x'^*\right),$$

  are both statistically close to the distribution $\{f(x)\}_{x \leftarrow \mathcal{X}}$, and are hence statistically indistinguishable. This completes the proof of uniformity.

- **Collision-Resistance.** Suppose that there exists a PPT adversary $\mathcal{A}$ that produces with non-negligible probability ε a tuple $((\mathbf{s}, r), (\mathbf{s}', r'))$ such that $(\mathbf{s}, r) \neq (\mathbf{s}', r')$ and

$$\mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}; r\right) = \mathsf{CHash}\left(\mathsf{pp}, \mathbf{s}'; r'\right).$$

  An argument very similar to the one used in proof of Theorem A.3 can be used to demonstrate the existence of a a PPT algorithm $\mathcal{B}$ that breaks the one-wayness of f with non-negligible probability.

- **Trapdoor Collisions:** Finally, it is straightforward to verify that trapdoor collisions produced by the scheme are valid.

Note A.9. The aforementioned chameleon hash construction has an apriori bounded number of homomorphic operations, which allows it to be instantiated similarly using a γ-bounded HOWF family, subject to the restriction that $n \leq \gamma$.

Implications. The following are some implications of chameleon hash functions:

- Chameleon hash functions imply statistically hiding and computationally binding (non-interactive) trapdoor commitment schemes [KR00], which in turn imply resettable zero-knowledge proofs for NP [CGGM00].
- Chameleon hash functions yield *chameleon signature schemes* [KR00], which are non-interactive undeniable signatures [CA89], and guarantee both non-repudiation and non-transferability.

All of the aforementioned primitives are therefore implied by any (γ-bounded) HOWF family.

B Homomorphism over Abelian Groups

In this section, we present constructions of primitives from IHwUFs/IHwPRFs that require the underlying input and output groups to be abelian.

B.1 Group-Homomorphic PKE

Definition B.1. (Group-Homomorphic PKE.) A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ over a message group $(\mathcal{M}, \oplus)$ and a ciphertext group $(\mathcal{C}, \otimes)$ is group-homomorphic if it satisfies the following: letting $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}\left(1^\lambda\right)$, for all messages $\mathsf{m}_1, \mathsf{m}_2 \in \mathcal{M}$, it holds with overwhelming probability over the randomness of Enc that

$$\mathsf{Dec}\left(\mathsf{sk}, \mathsf{Enc}\left(\mathsf{pk}, \mathsf{m}_1 \oplus \mathsf{m}_2\right)\right) = \mathsf{Dec}\left(\mathsf{sk}, \mathsf{Enc}\left(\mathsf{pk}, \mathsf{m}_1\right)\right) \otimes \mathsf{Dec}\left(\mathsf{sk}, \mathsf{Enc}\left(\mathsf{pk}, \mathsf{m}_2\right)\right)$$

Construction of Group-Homomorphic PKE. We present a group-homomorphic CPA-Secure PKE from any IHwPRF with abelian input and output groups.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm creates a description $\mathcal{F}_{\mathrm{IHwPRF}}$ for an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. It also fixes some integer $n > 3 \log |\mathcal{X}|$. The algorithm outputs $\mathcal{F}_{\mathrm{IHwPRF}}$ and n as the public parameter pp.
- $\mathsf{Gen}\left(\mathsf{pp}\right)$: The key-generation algorithm uniformly samples $2n$ elements in $\mathcal{X}$ as $\{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}$ and a key $k \leftarrow \mathcal{K}$, and outputs

$$\mathsf{sk} = k, \quad \mathsf{pk} = \{x_{j,b}, y_{j,b}\}_{j \in [n], b \in \{0,1\}},$$

  where $y_{j,b} = F(k, x_{j,b})$ for each $j \in [n]$ and $b \in \{0,1\}$.

- $\mathsf{Enc}\,(\mathsf{pk}, \mathsf{m})$: Given the public-key $\mathsf{pk} = \{x_{j,b}, y_{j,b}\}_{j\in[n],b\in\{0,1\}}$ and a message $\mathsf{m} \in \mathcal{Y}$, the encryption algorithm uniformly samples an n-bit string $\mathbf{r} = (r_1, \ldots, r_n) \leftarrow \{0,1\}^n$ and outputs the ciphertext $\mathsf{ct} = (c, \mathsf{e})$, where

$$c = \bigoplus_{j\in[n]} x_{j,r_j}, \quad \mathsf{e} = \left(\bigotimes_{j\in[n]} y_{j,r_j}\right) \otimes \mathsf{m}.$$

- $\mathsf{Dec}\,(\mathsf{sk}, \mathsf{ct})$: Given the secret-key $\mathsf{sk} = k$ and the ciphertext $\mathsf{ct} = (c, \mathsf{e})$, the algorithm outputs

$$\mathsf{m}' = (F(k, c))^{-1} \otimes \mathsf{e}.$$

Instantiation from General Protocol. The aforementioned PKE scheme can be instantiated from the general protocol described in Section 3.3 as follows:

- **Initialization.** Instantiate the protocol using an IHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are abelian groups, a fixed $n > 3\log|\mathcal{X}|$, $N^* = 0$, $\bar{N} = 1$ and $\hat{N} = 1$. Set $\mathsf{pk}_1 = \mathrm{X}$.
- **Evaluation:** Set $\mathsf{sk} = k$ (where $k \in \mathcal{K}$ is the IHwUF key) and $\mathsf{pk}_2 = \mathrm{Y}$, where Y is the output of the evaluation phase. Output $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2))$.
- **Post-Evaluation:** Let $\hat{\mathrm{X}} = \{\hat{x}\}$ and $\hat{\mathrm{Y}} = \{\hat{y}\}$. Set:

$$c = \hat{x}, \quad \mathsf{e} = \hat{y} \otimes \mathsf{m},$$

  where $\mathsf{m} \in \mathcal{Y}$ is the message.

Correctness and security of the scheme follow from Claim 3.10 and Theorem 3.17, respectively.

Group-Homomorphism. To see that the aforementioned PKE scheme is group-homomorphic, consider a pair of ciphertexts $(\mathsf{ct}_1, \mathsf{ct}_2) = ((c_1, \mathsf{e}_1), (c_2, \mathsf{e}_2))$, where

$$c_1 = \bigoplus_{j\in[n]} x_{j,r_{j,1}} \quad , \quad \mathsf{e}_1 = \left(\bigotimes_{j\in[n]} y_{j,r_{j,1}}\right) \otimes \mathsf{m}_1,$$
$$c_2 = \bigoplus_{j\in[n]} x_{j,r_{j,2}} \quad , \quad \mathsf{e}_2 = \left(\bigotimes_{j\in[n]} y_{j,r_{j,2}}\right) \otimes \mathsf{m}_2.$$

for some $\mathsf{m}_1, \mathsf{m}_2 \in \mathcal{Y}$. Now consider the ciphertext $\mathsf{ct}_3 = (c_3, \mathsf{e}_3)$, where

$$c_3 = c_1 \oplus c_2, \quad \mathsf{e}_3 = \mathsf{e}_1 \otimes \mathsf{e}_2.$$

By the leftover hash lemma, there exists a string $\mathbf{r}_3 = (r_{1,3}, \ldots, r_{n,3}) \in \{0,1\}^n$ such that

$$c_3 = \bigoplus_{j\in[n]} x_{j,r_{j,3}}, \quad \mathsf{e}_3 = F\left(k, \bigoplus_{j\in[n]} x_{j,r_{j,3}}\right) \otimes (\mathsf{m}_1 \otimes \mathsf{m}_2),$$

where the second equality additionally exploits the abelian nature of the group $(\mathcal{Y}, \otimes)$. Quite evidently, ct_3 is a valid ciphertext for the message $\mathsf{m}_3 = \mathsf{m}_1 \otimes \mathsf{m}_2$.

Note B.2. The aforementioned construction may be instantiated using a γ-bounded IHwPRF over abelian groups provided that γ is sufficiently larger than n [1], with the following minor modification to the encryption algorithm:

$$\mathsf{e} = \mathcal{R}\left(\bigotimes_{j\in[n]} y_{j,r_j}\right) \odot \mathsf{m},$$

[1] The number of homomorphic operations allowed for the PKE depends on the values of γ (and n).

and the following minor modification to the decryption algorithm:

$$\mathsf{m}' = \mathcal{R}\left(F(k,c)\right)^{-1} \odot \mathsf{e}$$

where $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ is a universal map (see Definition 3.4), with the additional property that $(\mathcal{Z}, \odot)$ is an abelian group and the homomorphism also propagates through $\mathcal{R}$ (in addition to the function F). Note that the message space is now $\mathcal{Z}$ instead of $\mathcal{Y}$.

B.2 Hash Proof Systems

In this subsection, we show that any IHwPRF family implies an average-case smooth projective hash proof system. We begin by briefly recalling the notion of projective hash proof systems as defined in [CS02].

Definition B.3. (Projective Hash Proof System.) Let $\mathcal{H} : \mathcal{HK} \times \Sigma_1 \to \Sigma_2$ be an efficiently computable function, and let $\mathcal{L} \subset \Sigma_1$. Also, let $\alpha : \mathcal{HK} \to \mathcal{HP}$ be a "projection" function. We say that the tuple $\text{HPS} = (\mathcal{H}, \mathcal{HP}, \mathcal{HK}, \Sigma_1, \Sigma_2, \mathcal{L})$ is a projective hash proof system if the following four properties hold:

1. There exist efficient algorithms to sample uniformly from Σ_1, uniformly from $\mathcal{HK}$, and uniformly from $\mathcal{L}$ along with a witness w that proves membership in $\mathcal{L}$.
2. Given any uniformly random $\sigma \leftarrow \Sigma_1$, no PPT algorithm can efficiently decide if $\sigma \in \mathcal{L}$.
3. For any $\mathsf{hk} \in \mathcal{HK}$ and $y \in \mathcal{L}$, the value of $\mathcal{H}(\mathsf{hk}, \sigma)$ is determined entirely by (σ, hp), where $\mathsf{hp} = \alpha(\mathsf{hk})$.
4. There exists an efficient "public evaluation" algorithm that on input $\sigma \in \mathcal{L}$, a witness w for the statement that $\sigma \in \mathcal{L}$, and the projection $\mathsf{hp} = \alpha(\mathsf{hk})$, outputs $\mathcal{H}(\mathsf{hk}, \sigma)$.

Definition B.4. (Average-Case Smooth Projective HPS.) Given $\text{HPS} = (\mathcal{H}, \mathcal{HP}, \mathcal{HK}, \Sigma_1, \Sigma_2, \mathcal{L})$, define the following distributions:

$$\begin{aligned} \mathcal{D}_{\text{HPS,real}} &= \left\{\alpha(\mathsf{hk}), \sigma, \mathcal{H}(\mathsf{hk}, \sigma)\right\}_{\mathsf{hk} \leftarrow \mathcal{HK}, \sigma \leftarrow \Sigma_1} \\ \mathcal{D}_{\text{HPS,rand}} &= \left\{\alpha(\mathsf{hk}), \sigma, \widetilde{\sigma}\right\}_{\mathsf{hk} \leftarrow \mathcal{HK}, \sigma \leftarrow \Sigma_1, \widetilde{\sigma} \leftarrow \Sigma_2} \end{aligned}$$

The projective hash proof system HPS is said to be ϵ-average-case smooth if we have the property $\mathsf{SD}\left(\mathcal{D}_{\text{HPS,real}}, \mathcal{D}_{\text{HPS,rand}}\right) < \epsilon$.

Definition B.5. (Homomorphic Projective HPS.) A projective hash proof system $\text{HPS} = (\mathcal{H}, \mathcal{HP}, \mathcal{HK}, \Sigma_1, \Sigma_2, \mathcal{L})$ is homomorphic if the following two properties hold:

1. There exist efficiently computable group operations $\oplus$ and $\otimes$ such that $(\Sigma_1, \oplus)$ and $(\Sigma_2, \otimes)$ are efficiently samplable groups.
2. For every $\mathsf{hk} \in \mathcal{HK}$ and for every $\sigma_1, \sigma_2 \in \Sigma_1$, we have

$$\mathcal{H}(\mathsf{hk}, \sigma_1 \oplus \sigma_2) = \mathcal{H}(\mathsf{hk}, \sigma_1) \otimes \mathcal{H}(\mathsf{hk}, \sigma_2).$$

Projective HPS from IHwPRFs. Given an IHwPRF $F : K \times \mathcal{X} \to \mathcal{Y}$, fix an $n > 4(\log|\mathcal{X}| + \log|\mathcal{Y}|)$ and sample $2n$ group elements from $\mathcal{X}$ as

$$\mathbf{x} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j \in [n], b \in \{0,1\}}.$$

Define the language $\mathcal{L}_{\mathbf{x}} \subset \mathcal{Y}^{2n}$ as

$$\mathcal{L}_{\mathbf{x}} = \left\{ \{y_{j,b} = F(k, x_{j,b})\}_{j \in [n], b \in \{0,1\}} \right\}_{k \in \mathcal{K}}$$

Next, define the hash function $\mathcal{H} : \{0,1\}^n \times \mathcal{Y}^{2n} \to \mathcal{Y}$ as

$$\mathcal{H}\left(\mathbf{r} = (r_1, \ldots, r_n), \mathbf{y} = \{y_{j,b}\}_{j\in[n], b\in\{0,1\}}\right) = \bigotimes_{j\in[n]} y_{j,r_j}.$$

Finally, define the projection function $\alpha_{\mathbf{x}} : \{0,1\}^n \to \mathcal{X}$ as

$$\alpha_{\mathbf{x}}\left(\mathbf{r} = (r_1, \ldots, r_n)\right) = \bigoplus_{j\in[n]} x_{j,r_j}.$$

To see that $\left(\mathcal{H}, \mathcal{X}, \{0,1\}^n, \mathcal{Y}^{2n}, \mathcal{Y}, \mathcal{L}_{\mathbf{x}}\right)$ is a projective HPS, observe the following.

- One can efficiently sample a random element of $\mathcal{L}_{\mathbf{x}}$, along with a witness for its membership, by generating $k \leftarrow \mathcal{K}$ and computing $y_{j,b} = F\left(k, x_{j,b}\right)$ for each $j \in [n]$ and $b \in \{0,1\}$.
- If there exists a PPT algorithm $\mathcal{A}$ such that, given a uniformly random vector $2n$-vector $\mathbf{y} \in \mathcal{Y}$, $\mathcal{A}$ can efficiently decide with non-negligible probability if $\mathbf{y} \in \mathcal{L}_{\mathbf{x}}$, then one can construct a PPT adversary $\mathcal{B}$ that breaks the weak pseudorandomness of F with non-negligible probability.
- Consider an $2n$-vector $\mathbf{y} = \{y_{j,b}\}_{j\in[n], b\in\{0,1\}}$ such that there exists a key $k \in \mathcal{K}$ for which $y_{j,b} = F\left(k, x_{j,b}\right)$ for $j \in [n]$ and $b \in \{0,1\}$. The "public evaluation" algorithm takes as input a projection $\alpha_{\mathbf{x}}\left(\mathbf{r}\right)$ of a string $\mathbf{r}$ and the "witness" key k, and outputs $\mathcal{H}\left(\mathbf{r}, \mathbf{y}\right) = F\left(k, \alpha_{\mathbf{x}}\left(\mathbf{r}\right)\right)$.

Average-Case Smoothness. Let $\mathbf{x}$ and $\mathbf{y}$ be random vectors in $\mathcal{X}^{2n}$ and $\mathcal{Y}^{2n}$, and let $x^* = \alpha_{\mathbf{x}}(\mathbf{r})$ for $\mathbf{r} \leftarrow \{0,1\}^n$. The following lemma implies average-case smoothness for our HPS scheme.

Lemma B.6. *Letting $\mathbf{x} \leftarrow \mathcal{X}^{2n}$, $\mathbf{y} \leftarrow \mathcal{Y}^{2n}$, $x^* = \alpha_{\mathbf{x}}(\mathbf{r})$ and $y^* = \mathcal{H}\left(\mathbf{r}, \mathbf{y}\right)$ for $\mathbf{r} \leftarrow \{0,1\}^n$, the following holds for any unbounded adversary $\mathcal{A}$:*

$$\Pr\left[\mathcal{A}\left(\mathbf{x}, \mathbf{y}, x^*\right) = y^*\right] \leq \mathrm{negl}(\lambda).$$

To prove this lemma, we first recall the following lemma due to Erdos and Renyi [ER65]:

Lemma B.7. *Let $(\mathcal{X}, \oplus)$ be a finite group, and let $\mathbf{x} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n], b\in\{0,1\}}$ be a vector of $2n$ random elements. Given an element $x \in \mathcal{X}$, let $\kappa_{\mathbf{x}}(x)$ be a random variable that denotes the number of all binary strings $\mathbf{r} = (r_1, \ldots, r_n)$ such that $x = \bigoplus x_{j,r_j}$. If $n > 2\log(1/\mathcal{X}) + 2\log(1/\varepsilon) + \log(1/\delta)$, then for any $x \in \mathcal{X}$*

$$\Pr\left[(1-\varepsilon)\frac{2^n}{|\mathcal{X}|} \leq \kappa_{\mathbf{x}}(x) \leq (1+\varepsilon)\frac{2^n}{|\mathcal{X}|}\right] \geq 1 - \delta.$$

This lemma (together with the leftover hash lemma) implies that given two groups $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$, two random vectors $\mathbf{x} = \{x_{j,b} \leftarrow \mathcal{X}\}_{j\in[n], b\in\{0,1\}}$ and $\mathbf{y} = \{y_{j,b} \leftarrow \mathcal{Y}\}_{j\in[n], \{0,1\}\in b}$, and a random binary string $\mathbf{r} \leftarrow \{0,1\}^n$ where $n > 4(\log|\mathcal{X}| + \log|\mathcal{Y}|)$, for any *unbounded* adversary $\mathcal{A}$ we have

$$\Pr_{\mathbf{x},\mathbf{y},\mathbf{r}}[\mathcal{A}(\mathbf{x}, \mathbf{y}, x^*) = y^*] \leq \mathrm{negl}(\lambda),$$

where

$$x^* = \bigoplus_{j\in[n]} x_{j,r_j}, \quad y^* = \bigotimes_{j\in[n]} y_{j,r_j}.$$

To see this, fix two elements $x^* \in \mathcal{X}$ and $y^* \in \mathcal{Y}$ and apply the previous lemma on the direct product group $\mathcal{X} \times \mathcal{Y}$. Now given random vectors $\mathbf{x}$ and $\mathbf{y}$, the number of $\mathbf{r}$'s such that $\bigoplus x_{j,r_j} = x^*$ is at least $(1 - \mathrm{negl}(\lambda)) \cdot |\mathcal{X}|^3|\mathcal{Y}|^4$. In addition, the number of $\mathbf{r}$'s such that $\bigoplus x_{j,r_j} = x^*$ *and* $\bigoplus y_{j,r_j} = y^*$ is at most $(1 + \mathrm{negl}(\lambda)) \cdot |\mathcal{X}|^3|\mathcal{Y}|^3$. It follows that there at least $(1 - \mathrm{negl}(\lambda)) \cdot |\mathcal{Y}|$ distinct y's such that $\bigoplus x_{j,r_j} = x^*$ *and* $\bigoplus y_{j,r_j} = y$. Since the distribution of $\bigoplus x_{j,r_j}$ and $\bigotimes y_{j,r_j}$ is negligibly close to uniform by the leftover hash lemma, it follows that given $(\mathbf{x}, \mathbf{y}, x^*)$ any unbounded adversary cannot guess y^* with non-negligible probability. This in turn implies Lemma B.6 and completes the proof of average-case smoothness for our HPS scheme.

Homomorphism. Observe that if $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are abelian, the HPS is homomorphic. Define the operation between pairs of $2n$-vectors $\left(\mathbf{y}^{(1)} = \left\{y^{(1)}_{j,b}\right\}_{j\in[n],b\in\{0,1\}}, \mathbf{y}^{(2)} = \left\{y^{(2)}_{j,b}\right\}_{j\in[n],b\in\{0,1\}}\right) \in \mathcal{Y}^{2n} \times \mathcal{Y}^{2n}$ as

$$\mathbf{y}^{(1)} \odot \mathbf{y}^{(2)} := \left(\left\{y^{(1)}_{j,b} \otimes y^{(2)}_{j,b}\right\}_{j\in[n],b\in\{0,1\}}\right).$$

Since $\mathcal{Y}$ is abelian with respect to the operation $\otimes$, we have

$$\mathcal{H}\left(\mathbf{r}, \mathbf{y}^{(1)} \odot \mathbf{y}^{(2)}\right) = \mathcal{H}\left(\mathbf{r}, \mathbf{y}^{(1)}\right) \otimes \mathcal{H}\left(\mathbf{r}, \mathbf{y}^{(2)}\right).$$

Note B.8. The aforementioned construction requires an IHwPRF family for which the key space does not depend on the input space. However, since the number of homomorphic operations needed to compute a hash is upper bounded by n, it maybe equivalently instantiated using a γ-bounded IHwPRF family, subject to the restriction that $n \leq \gamma$.

B.3 HOWFs from IHwUFs

In this subsection, we show that IHwUFs with *unbounded* homomorphism over abelian groups imply HOWFs.

Construction of HOWF. Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an IHwUF such that $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are abelian groups. Let $\mathbf{x} = \{x_j \leftarrow \mathcal{X}\}_{j\in[n]}$ be n uniform elements for $n > 3\log|\mathcal{X}|$. Define the family of functions $\mathrm{HOWF}_{\mathbf{x}} : \mathbb{Z}^n_{|\mathcal{X}|} \to \mathcal{X}$ as

$$\mathrm{HOWF}_{\mathbf{x}}\left(\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)\right) = \bigoplus_{j\in[n]} [\alpha_j]\, x_j.$$

where $[\alpha_j]\, x_j$ denotes operating x_j with itself α_j times by applying the group operation $\oplus$. Homomorphism follows directly from the abelian property of the group $(\mathcal{X}, \oplus)$. For $n > 3\log|\mathcal{X}|$, consider the following experiment $\mathsf{Expt}^{\text{HOWF-IHwUF}}_{|\mathcal{X}|}$:

1. The challenger samples n group elements as $\{x_j \leftarrow \mathcal{X}\}_{j\in[n]}$ and a vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n) \leftarrow \mathbb{Z}^n_{|\mathcal{X}|}$.
2. The challenger computes $x^* = \bigoplus_{j\in[n]} [\alpha_j]\, x_j$ and sends the tuple $\left(\{x_j\}_{j\in[n]}, x^*\right)$ to the adversary.
3. Eventually, the adversary $\mathcal{A}$ outputs $\vec{\alpha}' = (\alpha'_1, \ldots, \alpha'_n) \in \mathbb{Z}^n_{|\mathcal{X}|}$.

For any PPT adversary $\mathcal{A}$ we define $\mathbf{Adv}^{\text{HOWF-IHwUF}}(\mathcal{A})$ to be the probability of $x^* = \bigoplus_{j\in[n]} \left[\alpha'_j\right] x_j$ over all random coins in the experiment.

Lemma B.9. *For all PPT adversaries we have* $\mathbf{Adv}^{\text{HOWF-IHwUF}}(\mathcal{A}) \leq \mathrm{negl}(\lambda)$.

Proof. The proof is similar to the proof of Lemma 3.7.

C Composable IHwPRFs

In this section, we extend our homomorphism-based framework to allow pairs of IHwPRFs that compose with each other. We refer to such IHwPRFs as *two-composable* IHwPRFs. The formal definition is presented below.

Definition C.1 (Two-Composable IHwPRF). A two-composable IHwPRF is a tuple of two functions and two "composers"

$$F_1 : \mathcal{K} \times \mathcal{X}_1 \to \mathcal{Y}_1 \quad , \quad F_2 : \mathcal{K} \times \mathcal{X}_2 \to \mathcal{Y}_2,$$
$$C_1 : \mathcal{Y}_1 \times \mathcal{X}_2 \to \mathcal{Z} \quad , \quad C_2 : \mathcal{Y}_2 \times \mathcal{X}_1 \to \mathcal{Z}.$$

such that the following conditions hold:

1. $(\mathcal{X}_1, \oplus_1)$, $(\mathcal{X}_2, \oplus_2)$, $(\mathcal{Y}_1, \otimes_1)$, $(\mathcal{Y}_2, \otimes_2)$ and $(\mathcal{Z}, \odot)$ are efficiently samplable groups.
2. The group operations $\oplus_1$, $\oplus_2$, $\otimes_1$, $\otimes_2$ and $\odot$, and the inverse operations in each group, are efficiently computable.
3. The functions $F_1 : \mathcal{K} \times \mathcal{X}_1 \to \mathcal{Y}_1$ and $F_2 : \mathcal{K} \times \mathcal{X}_2 \to \mathcal{Y}_2$ are IHwPRFs.
4. The "composers" $C_1 : \mathcal{Y}_1 \times \mathcal{X}_2 \to \mathcal{Z}$ and $C_2 : \mathcal{Y}_2 \times \mathcal{X}_1 \to \mathcal{Z}$ are weak PRFs.
5. For every $k \in \mathcal{K}$ and for every $x_1, x_2 \in \mathcal{X}$, we have:

$$C_1\left(F_1\left(k, x_1\right), x_2\right) = C_2\left(F_2\left(k, x_2\right), x_1\right).$$

   We will denote either of the above equal quantities as $F_T(k, (x_1, x_2))$.

Note C.2. We do not impose any homomorphism requirements on the composers C_1 and C_2.

Notations. We adopt some notations in this section to simplify the exposition. Given a vector $\mathbf{x} \in \mathcal{G}^{2n}$ indexed by $(i \in [n], b \in \{0,1\})$, where $(\mathcal{G}, +)$ is any group, and a string $\mathbf{s} \in \{0,1\}^n$, we define a subset-summing notation as follows:

$$\langle\!\langle \mathbf{x}, \mathbf{s} \rangle\!\rangle = \sum_{i \in [n]} x_{i, s_i}$$

We now state a useful lemma. All constructions presented in this section depend on this lemma for their security.

Lemma C.3 (Two-Composable IHwPRF Lemma). *Let (F_1, F_2, C_1, C_2) be a two-composable* IHwPRF*. Then for any $n_1, n_2 > 3\log\left(\max\left(|\mathcal{X}_1|, |\mathcal{X}_2|\right)\right)$ and for any PPT adversary $\mathcal{A}$, the following holds:*

$$\left|\Pr\left[\mathcal{A}\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, x_1^*, x_2^*, z^*\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, u_1^*, u_2^*, v^*\right) = 1\right]\right| \leq \mathrm{negl}(\lambda),$$

where:

- *The vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are uniform in $\mathcal{X}_1^{2n_1}$ and $\mathcal{X}_2^{2n_2}$, respectively.*
- *For some uniformly sampled bit strings $\mathbf{r}_1 \leftarrow \{0,1\}^{n_1}$ and $\mathbf{r}_2 \leftarrow \{0,1\}^{n_2}$, we have*

$$x_1^* = \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{r}_1 \right\rangle\!\right\rangle, \quad x_2^* = \left\langle\!\left\langle \mathbf{x}^{(2)}, \mathbf{r}_2 \right\rangle\!\right\rangle.$$

- *For some $k \in \mathcal{K}$, we have*

$$\mathbf{y}^{(1)} = F_1\left(k, \mathbf{x}^{(1)}\right), \quad \mathbf{y}^{(2)} = F_2\left(k, \mathbf{x}^{(2)}\right), \quad z^* = F_T\left(k, (x_1^*, x_2^*)\right).$$

- *The group elements u_1^*, u_2^* and v^* are uniform in $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{Z}$ respectively.*

Proof. The proof of this lemma follows immediately from the following lemmas:

Lemma C.4. *For any $n_1, n_2 > 3\log(\max(|\mathcal{X}_1|, |\mathcal{X}_2|))$ and for any PPT adversary $\mathcal{A}$, the following holds:*

$$\left|\Pr\left[\mathcal{A}\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, x_1^*, x_2^*, z^*\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, u_1^*, u_2^*, \tilde{z}^*\right) = 1\right]\right| \leq \mathrm{negl}(\lambda),$$

where $\tilde{z}^* = F_T(k, (u_1^*, u_2^*))$.

Lemma C.5. *Let F_1 be an IHwPRF and and C_1 be a wPRF. Then, for any PPT adversary $\mathcal{A}$, the following holds:*

$$\left|\Pr\left[\mathcal{A}\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, x_1^*, x_2^*, \tilde{z}^*\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, u_1^*, u_2^*, v^*\right) = 1\right]\right| \leq \mathrm{negl}(\lambda),$$

The proof of Lemma C.4 follows from the leftover hash lemma, while the proof of Lemma C.5 follows from the weak pseudorandomness of F_1 and C_1.

In the following subsections, we show that two-composable IHwPRFs imply (in a black-box way) certain cryptoprimitives that do not have known black-box realizations from standard IHwPRFs.

C.1 Non-Interactive Three-Party Key-Exchange

We present a black-box non-interactive three-party key-exchange protocol from any two-composable IHwPRF. Note that three-party NIKE is currently known only from bilinear maps. Our protocol is in the standard model, and involves three non-uniform PPT algorithms $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$ and $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1)$, where $\mathcal{A}_b$, $\mathcal{B}_b$ and $\mathcal{C}_b$ operate in parallel for $b \in \{0,1\}$.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm creates a description $\mathcal{F}_{\mathrm{IHCwPRF}}$ for a two-composable IHwPRF consisting of the following functions and "composers":

  $$F_1 : \mathcal{K} \times \mathcal{X}_1 \to \mathcal{Y}_1 \quad , \quad F_2 : \mathcal{K} \times \mathcal{X}_2 \to \mathcal{Y}_2,$$
  $$C_1 : \mathcal{Y}_1 \times \mathcal{X}_2 \to \mathcal{Z} \quad , \quad C_2 : \mathcal{Y}_2 \times \mathcal{X}_1 \to \mathcal{Z}.$$

  It uniformly samples $\mathbf{x}^{(1)} \leftarrow \mathcal{X}_1^{2n}$ and $\mathbf{x}^{(2)} \leftarrow \mathcal{X}_2^{2n}$, for $n > 3\log(\max(|\mathcal{X}_1|, |\mathcal{X}_2|))$, and outputs the public parameter pp as

  $$\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHCwPRF}}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right).$$

- $\mathcal{A}_0(\mathsf{pp})$: On input pp, the algorithm $\mathcal{A}_0$ samples $\mathbf{a} \leftarrow \{0,1\}^n$ and outputs $(\mathsf{st}_{\mathcal{A}}, x_{\mathcal{A}}^*)$, where

  $$\mathsf{st}_{\mathcal{A}} = \mathbf{a}, \quad x_{\mathcal{A}}^* = \left\langle\left\langle \mathbf{x}^{(1)}, \mathbf{a}\right\rangle\right\rangle.$$

- $\mathcal{B}_0(\mathsf{pp})$: On input pp, the algorithm $\mathcal{B}_0$ samples $\mathbf{b} \leftarrow \{0,1\}^n$ and outputs $(\mathsf{st}_{\mathcal{B}}, x_{\mathcal{B}}^*)$, where

  $$\mathsf{st}_{\mathcal{B}} = \mathbf{b}, \quad x_{\mathcal{B}}^* = \left\langle\left\langle \mathbf{x}^{(2)}, \mathbf{b}\right\rangle\right\rangle.$$

- $\mathcal{C}_0(\mathsf{pp})$: On input pp, the algorithm $\mathcal{C}_0$ samples a key $k \leftarrow \mathcal{K}$ and computes $\mathbf{y}^{(1)} = F_1\left(k, \mathbf{x}^{(1)}\right)$ and $y^{(2)} = F_2\left(k, \mathbf{x}^{(2)}\right)$. It then outputs $(\mathsf{st}_{\mathcal{C}}, \mathbf{y}_{\mathcal{C}})$, where

  $$\mathsf{st}_{\mathcal{C}} = k, \quad \mathbf{y}_{\mathcal{C}} = \left(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}\right).$$

- $\mathcal{A}_1(\mathsf{pp}, \mathsf{st}_{\mathcal{A}}, x_{\mathcal{B}}^*, \mathbf{y}_{\mathcal{C}})$: Given $\mathsf{st}_{\mathcal{A}} = \mathbf{a}$, $x_{\mathcal{B}}^*$, and $\mathbf{y}_{\mathcal{C}}$, the algorithm $\mathcal{A}_1$ computes the final key as

  $$\mathsf{k}^* = C_1\left(\left\langle\left\langle \mathbf{y}^{(1)}, \mathbf{a}\right\rangle\right\rangle, x_{\mathcal{B}}^*\right).$$

- $\mathcal{B}_1\left(\mathsf{pp}, \mathsf{st}_{\mathcal{B}}, x^*_{\mathcal{A}}, \mathbf{y}_{\mathcal{C}}\right)$: Given $\mathsf{st}_{\mathcal{B}} = \mathbf{b}$, $x^*_{\mathcal{A}}$, and $\mathbf{y}_{\mathcal{C}}$, the algorithm $\mathcal{B}_1$ computes the final key as

$$\mathsf{k}^* = C_2\left(\left\langle\!\left\langle \mathbf{y}^{(2)}, \mathbf{b} \right\rangle\!\right\rangle, x^*_{\mathcal{A}}\right).$$

- $\mathcal{C}_1\left(\mathsf{pp}, \mathsf{st}_{\mathcal{C}}, x^*_{\mathcal{A}}, x^*_{\mathcal{B}}\right)$: Given $\mathsf{st}_{\mathcal{C}} = k$, $x^*_{\mathcal{A}}$, and $x^*_{\mathcal{B}}$, the algorithm $\mathcal{C}_1$ computes the final key as

$$\mathsf{k}^* = C_1\left(F_1\left(k, x^*_{\mathcal{A}}\right), x^*_{\mathcal{B}}\right),$$

  or equivalently, as

$$\mathsf{k}^* = C_2\left(F_2\left(k, x^*_{\mathcal{B}}\right), x^*_{\mathcal{A}}\right).$$

Correctness and Security. Correctness follows from the properties of a two-composable IHwPRF. When the protocol is correctly executed, in all cases the final key computed is:

$$\mathsf{k}^* = F_T\left(k, \left(\left\langle\!\left\langle (\mathbf{x}^{(1)}, \mathbf{a} \right\rangle\!\right\rangle, \left\langle\!\left\langle \mathbf{x}^{(2)}, \mathbf{b} \right\rangle\!\right\rangle\right)\right).$$

Security follows from Lemma C.3.

C.2 Black-Box IBE

We present a black-box construction of an IBE scheme from any two-composable IHwPRF. Note that the only IBE construction from IHwPRFs is non-black-box since it requires garbled circuits (Section 4.4). Black-box constructions of IBE (even in the random oracle model) are currently known only from group-theoretic/lattice assumptions, such as bilinear Diffie-Hellman [BF01], QR [Coc01] and LWE [GPV08], and the techniques used for these constructions are closely tied to the underlying assumption. Our construction, on the other hand, makes black-box use of a generic primitive that seems to be strictly weaker than bilinear maps.

Definition C.6. (Identity-Based Encryption.) An IBE scheme over an identity space $\mathcal{ID}$ and message space $\mathcal{M}$ is a tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

- $\mathsf{Setup}(1^\lambda)$: Given the security parameter λ, outputs the public parameter pp and the master secret-key msk.
- $\mathsf{Ext}\left(\mathsf{pp}, \mathsf{msk}, \mathsf{id}\right)$: Given the public parameter pp, the master-secret-key msk and an identity $\mathsf{id} \in \mathcal{ID}$, outputs a secret-key $\mathsf{sk}_{\mathsf{id}}$.
- $\mathsf{Enc}\left(\mathsf{pp}, \mathsf{id} \in \mathcal{ID}, \mathsf{m}\right)$: Given the public parameter pp, an identity $\mathsf{id} \in \mathcal{ID}$ and a message $\mathsf{m} \in \mathcal{M}$, outputs a ciphertext ct.
- $\mathsf{Dec}\left(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct}\right)$: Given a secret key $\mathsf{sk}_{\mathsf{id}}$ and a ciphertext ct, outputs a decrypted message m'.

The following correctness and security properties must be satisfied:

- **Correctness:** If $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}\left(1^\lambda\right)$, then for all $\mathsf{id} \in \mathcal{ID}$ and all $\mathsf{m} \in \mathcal{M}$, it holds with overwhelming probability over the randomness of Ext and Enc that if $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{Ext}\left(\mathsf{pp}, \mathsf{msk}, \mathsf{id}\right)$ and $\mathsf{ct} \leftarrow \mathsf{Enc}\left(\mathsf{pp}, \mathsf{id} \in \mathcal{ID}, \mathsf{m}\right)$, then we have

$$\mathsf{Dec}\left(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct}\right) = \mathsf{m}.$$

- **Anonymous-CPA Security:** For $b \in \{0, 1\}$, define the experiment $\mathsf{Expt}_b^{\text{ano-cpa}}$ between a challenger and an adversary $\mathcal{A}$ as in Figure 10:

1. The challenger samples $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and provides pp to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ adaptively issues key-generation queries. For each query identity id, the challenger responds with
$$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{Ext}(\mathsf{msk}, \mathsf{id}).$$
3. The adversary $\mathcal{A}$ outputs identity-message pairs $(\mathsf{id}_0^*, \mathsf{m}_0^*)$ and $(\mathsf{id}_1^*, \mathsf{m}_1^*)$, such that $\mathsf{id}_{b^*}^* \neq \mathsf{id}$ for each identity id queried previously and each $b^* \in \{0,1\}$. The challenger responds to the adversary $\mathcal{A}$ with the ciphertext
$$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_b^*, \mathsf{m}_b^*).$$
4. The adversary $\mathcal{A}$ continues to adaptively issue key-generation queries, subject to the aforementioned restrictions. The challenger responds as above.

Figure 10: Experiment for the Anonymous CPA security of IBE.

An IBE scheme $(\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ is said to be anonymous-CPA-secure if for all PPT adversaries $\mathcal{A}$, the views of the adversary in $\mathsf{Expt}_0^{\text{ano-cpa}}$ and $\mathsf{Expt}_1^{\text{ano-cpa}}$ are computationally indistinguishable.

Construction from 2-Composable IHwPRF. Our construction is inspired by the seminal Boneh-Franklin IBE [BF01] and is both anonymous and message-hiding against adaptive adversaries in the random oracle model.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm creates a description $\mathcal{F}_{\text{IHCwPRF}}$ for a two-composable IHwPRF consisting of the following functions and "composers":
$$F_1 : \mathcal{K} \times \mathcal{X}_1 \to \mathcal{Y}_1 \quad , \qquad F_2 : \mathcal{K} \times \mathcal{X}_2 \to \mathcal{Y}_2,$$
$$C_1 : \mathcal{Y}_1 \times \mathcal{X}_2 \to \mathcal{Z} \quad , \qquad C_2 : \mathcal{Y}_2 \times \mathcal{X}_1 \to \mathcal{Z}.$$
It fixes an $n > 3\log(\max(|\mathcal{X}_1|, |\mathcal{X}_2|))$ and a hash function $H : \mathcal{ID} \to \mathcal{X}_2$ (modeled as a random oracle in the security proof), where $\mathcal{ID}$ is the identity space for the IBE scheme. It uniformly samples a key $k \leftarrow \mathcal{K}$ and $2n$ elements from $\mathcal{X}_1$ as $\mathbf{x}^{(1)} \leftarrow \mathcal{X}_1^{2n}$, and sets $\mathbf{y}^{(1)} = F_1\left(k, \mathbf{x}^{(1)}\right)$. Finally, it outputs the public parameter pp and the master secret-key msk as
$$\mathsf{pp} = \left(\mathcal{F}_{\text{IHCwPRF}}, H, \mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \quad \mathsf{msk} = k.$$
- $\mathsf{Ext}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$: Given the public parameter pp, the master secret-key $\mathsf{msk} = k$ and an identity $\mathsf{id} \in \mathcal{ID}$, the extraction algorithm outputs the secret-key $\mathsf{sk}_{\mathsf{id}}$ as
$$\mathsf{sk}_{\mathsf{id}} = F_2(k, H(\mathsf{id})).$$
- $\mathsf{Enc}(\mathsf{pp}, \mathsf{id} \in \mathcal{ID}, \mathsf{m})$: Given the public parameter pp, an identity $\mathsf{id} \in \mathcal{ID}$ and a message $\mathsf{m} \in \mathcal{Z}$, the encryption algorithm uniformly samples a string $\mathbf{r} = (r_1, \ldots, r_n) \leftarrow \{0,1\}^n$ and outputs the ciphertext $\mathsf{ct} = (c_1, c_2)$, where
$$c_1 = \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{r} \right\rangle\!\right\rangle, \quad c_2 = C_1\left(\left\langle\!\left\langle \mathbf{y}^{(1)}, \mathbf{r} \right\rangle\!\right\rangle, H(\mathsf{id})\right) \odot \mathsf{m}.$$
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct})$: Given the secret-key $\mathsf{sk}_{\mathsf{id}}$ and the ciphertext $\mathsf{ct} = (c_1, c_2)$, the decryption algorithm outputs the message
$$\mathsf{m}' = C_2(\mathsf{sk}_{\mathsf{id}}, c_1)^{-1} \odot c_2.$$

Correctness. Correctness of the scheme may be verified as follows:

$$\begin{aligned} c_2 &= C_1\left(\left\langle\!\left\langle \mathbf{y}^{(1)}, \mathbf{r} \right\rangle\!\right\rangle, H(\mathsf{id})\right) \odot \mathsf{m} \\ &= C_1\left(F_1\left(k, \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{r} \right\rangle\!\right\rangle\right), H\left(\mathsf{id}\right)\right) \odot \mathsf{m} \\ &= C_2\left(F_2\left(k, H\left(\mathsf{id}\right)\right), \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{r} \right\rangle\!\right\rangle\right) \odot \mathsf{m} \\ &= C_2\left(\mathsf{sk}_{\mathsf{id}}, c_1\right) \odot \mathsf{m} \end{aligned}$$

Security. Suppose that there exists a PPT adversary $\mathcal{A}$ that can distinguish between its views in the experiments $\mathsf{Expt}_0^{\text{ano-cpa}}$ and $\mathsf{Expt}_1^{\text{ano-cpa}}$ with non-negligible probability ε, while making a maximum of Q_1 hash queries and Q_2 secret key queries. We construct a PPT algorithm $\mathcal{B}$ with the advantage negligibly smaller than ε. Suppose $\mathcal{B}$ receives as input a tuple

$$\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \left\{x_\ell^{(2)}, y_\ell^{(2)}\right\}_{\ell \in [Q_1+Q_2+1]}, x_1^*, x_2^*, z^*\right),$$

as described in Lemma C.3 and interacts with $\mathcal{A}$ as follows.

- In the setup phase, $\mathcal{B}$ provides the tuple $\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right)$ to the adversary $\mathcal{A}$ as the public parameter. It also initializes a counter $\mathsf{cnt} := 0$ and a look-up table $\mathsf{T} := \phi$. The table T stores tuples of the form

  $$(\ell, \mathsf{id}_\ell, H(\mathsf{id}_\ell), \mathsf{e}_\ell) \quad \text{for } \ell \in \mathbb{N}, \mathsf{id}_\ell \in \mathcal{ID}, \mathsf{e}_\ell \in \{0,1\},$$

  and is used by $\mathcal{B}$ to answer random oracle queries issued by $\mathcal{A}$.

- When $\mathcal{A}$ issues a hash query on an identity id, $\mathcal{B}$ first checks if an entry of the form $(\ell, \mathsf{id}, H(\mathsf{id}), \mathsf{e})$ already exists in the table T.

  - If yes, it provides $H(\mathsf{id})$ to $\mathcal{A}$.
  - If not, it samples a bit $\mathsf{e} \in \{0,1\}$ such that $\Pr[\mathsf{e}=1] = \delta$ for some $\delta \in (0,1)$ and sets $H(\mathsf{id})$ as

    $$H(\mathsf{id}) = \begin{cases} x_{\mathsf{cnt}}^{(2)} & \text{if } \mathsf{e}=1 \\ x_2^* \oplus x_{\mathsf{cnt}}^{(2)} & \text{otherwise.} \end{cases}$$

    It then increments the counter as $\mathsf{cnt} := \mathsf{cnt} + 1$ and updates the table T as $\mathsf{T} := \mathsf{T} \cup \{(\mathsf{cnt}, \mathsf{id}, H(\mathsf{id}), \mathsf{e})\}$. Finally, it provides $H(\mathsf{id})$ to $\mathcal{A}$.

- When $\mathcal{A}$ issues a secret key query on an identity id_ℓ, $\mathcal{B}$ retrieves/creates the tuple $(\ell, \mathsf{id}_\ell, H(\mathsf{id}_\ell, \mathsf{e}_\ell))$ as described above.

  - If $\mathsf{e}_\ell = 0$, it outputs $\perp$ and aborts.
  - If $\mathsf{e}_\ell = 1$, it outputs $\mathsf{sk}_{\mathsf{id}_\ell} = y_\ell^{(2)}$.

  Quite evidently, when $\mathcal{B}$ does not abort, the distribution of $\mathsf{sk}_{\mathsf{id}_\ell}$ is identical to that in the "real" experiment.

- When $\mathcal{A}$ outputs the challenges $(\mathsf{id}_0^*, \mathsf{m}_0^*)$ and $(\mathsf{id}_1^*, \mathsf{m}_1^*)$, $\mathcal{B}$ uniformly samples a bit $b \leftarrow \{0,1\}$ and retrieves/creates the tuple $(\ell_b^*, \mathsf{id}_b^*, H(\mathsf{id}_b^*), \mathsf{e}_b^*)$ as described above.

  - If $\mathsf{e}_\ell = 1$, it outputs $\perp$ and aborts.
  - If $\mathsf{e}_\ell = 0$, it outputs the challenge ciphertext

    $$\mathsf{ct}^* = \left(x_1^*, z^* \odot C_2\left(y_{\ell_b^*}^{(2)}, x_1^*\right) \odot \mathsf{m}_b^*\right).$$

- $\mathcal{A}$ continues to issue hash and secret-key queries adaptively. $\mathcal{B}$ responds as described above.
- Eventually, $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{B}$ outputs 1. Else, it outputs 0.

When $z^* = C_1\left(F_1\left(k, x_1^*\right), x_2^*\right)$, we have

$$\begin{aligned} z^* \odot C_2\left(y_{\ell_b^*}^{(2)}, x_1^*\right) &= C_1\left(F_1\left(k, x_1^*\right), x_2^*\right) \odot C_2\left(y_{\ell_b^*}^{(2)}, x_1^*\right) \\ &= C_1\left(F_1\left(k, x_1^*\right), x_2^*\right) \odot C_2\left(F_2\left(k, x_{\ell_b^*}^{(2)}\right), x_1^*\right) \\ &= C_1\left(F_1\left(k, x_1^*\right), x_2^*\right) \odot C_1\left(F_1\left(k, x_1^*\right), x_{\ell_b^*}^{(2)}\right) \\ &= C_1\left(F_1\left(k, x_1^*\right), x_2^* \oplus x_{\ell_b^*}^{(2)}\right) \\ &= C_1\left(F_1\left(k, x_1^*\right), H(\mathsf{id}_b^*)\right) \end{aligned}$$

and hence, the ciphertext ct^* is well-formed w.r.t. the bit b chosen by $\mathcal{B}$. On the other hand, when z^* is uniform in $\mathcal{Z}$, the ciphertext ct^* is independent of the bit b chosen by $\mathcal{B}$. Hence, the advantage of $\mathcal{B}$ in breaking the two-composable IHwPRF assumption may be quantified as $\mathbf{Adv}_{\mathcal{B}} = \varepsilon - \Pr[\mathsf{abort}]$, where the probability that $\mathcal{B}$ aborts is upper bounded as $\Pr[\mathsf{abort}] \leq \delta(1-\delta)^{Q_2}$, which is negligible for $Q_2 = \mathrm{poly}(\lambda)$. This completes the proof of anonymous-CPA security of our IBE scheme from two-composable IHwPRFs.

C.3 L-Composable IHwPRFs

In this section, we generalize two-composable IHwPRFs introduced in Section C to *L-composable* IHwPRFs for any $L \geq 2$. The formal definition is presented below.

Definition C.7 (L-Composable IHwPRF). An L-composable IHwPRF is a collection of L functions and L "composers"

$$\left\{F_\ell : \mathcal{K} \times \mathcal{X}_\ell \to \mathcal{Y}_\ell\right\}_{\ell \in [L]} \quad , \quad \left\{C_\ell : \mathcal{Y}_\ell \times \mathcal{X}_{L\setminus\{\ell\}} \to \mathcal{Z}\right\}_{\ell \in [L]},$$

where

$$\mathcal{X}_{L\setminus\{\ell\}} := \mathcal{X}_1 \times \ldots \times \mathcal{X}_{\ell-1} \times \mathcal{X}_{\ell+1} \times \ldots \times \mathcal{X}_L \quad \text{for each } \ell \in [L]$$

such that the following conditions hold:

1. $\left\{\left(\mathcal{X}_\ell, \oplus_\ell\right), \left(\mathcal{Y}_\ell, \otimes_\ell\right)\right\}_{\ell \in [L]}$ and $\left(\mathcal{Z}, \odot\right)$ are efficiently samplable groups.
2. The group operations $\{\oplus_\ell, \otimes_\ell\}_{\ell \in [L]}$ and $\odot$, and the inverse operations in each group, are efficiently computable.
3. For each $\ell \in [L]$, the function $F_\ell : \mathcal{K} \times \mathcal{X}_\ell \to \mathcal{Y}_\ell$ is an *IHwPRF*.
4. For each $\ell \in [L]$, the "composer" $C_\ell : \mathcal{Y}_\ell \times \mathcal{X}_{L\setminus\{\ell\}} \to \mathcal{Z}$ is a *wPRF*.
5. For every $k \in \mathcal{K}$, for every $(x_1, \ldots, x_L) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_L$, and for any choice of $\ell_1, \ell_2 \in [L]$ the following quantities are equal:

   $$C_{\ell_1}\left(F_{\ell_1}\left(k, x_{\ell_1}\right), x_1, \ldots, x_{\ell_1-1}, x_{\ell_1+1}, \ldots, x_L\right) \quad , \quad C_{\ell_2}\left(F_{\ell_2}\left(k, x_{\ell_2}\right), x_1, \ldots, x_{\ell_2-1}, x_{\ell_2+1}, \ldots, x_L\right).$$

   We will denote each of the above equal quantities as $F_T(k, (x_1, \ldots, x_L))$.

Note C.8. As in the two-composable setting, we do not impose any homomorphism requirements on the composers $C_1, \ldots, C_L$.

Notations. We adopt the same notations as in the two-linear section. Given a vector $\mathbf{x} \in \mathcal{G}^{2n}$ indexed by $(i \in [n], b \in \{0,1\})$, where $(\mathcal{G}, +)$ is any group, and a string $\mathbf{s} \in \{0,1\}^n$, we use the subset-summing notation as follows:

$$\langle\langle \mathbf{x}, \mathbf{s} \rangle\rangle = \sum_{i \in [n]} x_{i,s_i}$$

Lemma C.9 (L-Composable IHwPRF Lemma). *Let $\left(\{F_\ell, C_\ell\}_{\ell \in [L]}\right)$ be an L-composable* IHwPRF*. Then for any $n_1, \ldots, n_L > 3\log\left(\max\left(|\mathcal{X}_1|, \ldots, |\mathcal{X}_L|\right)\right)$ and for any PPT adversary $\mathcal{A}$, the following holds:*

$$\left|\Pr\left[\mathcal{A}\left(\left\{\mathbf{x}^{(\ell)}, \mathbf{y}^{(\ell)}, x^*_\ell\right\}_{\ell \in [L]}, z^*\right) = 1\right] - \Pr\left[\mathcal{A}\left(\left\{\mathbf{x}^{(\ell)}, \mathbf{y}^{(\ell)}, u^*_\ell\right\}_{\ell \in [L]}, v^*\right) = 1\right]\right| \leq \mathrm{negl}(\lambda),$$

where:

- *For each $\ell \in [L]$, the vector $\mathbf{x}^{(\ell)}$ is uniform in $\mathcal{X}_\ell^{2n_\ell}$.*
- *For each $\ell \in [L]$, we have*
$$x^*_\ell = \left\langle\left\langle \mathbf{x}^{(\ell)}, \mathbf{r}_\ell \right\rangle\right\rangle.$$
where $\mathbf{r}_\ell \leftarrow \{0,1\}^{n_\ell}$ is a uniformly sampled bit string.
- *For some $k \in \mathcal{K}$, we have*
$$\mathbf{y}^{(\ell)} = F_\ell\left(k, \mathbf{x}^{(\ell)}\right) \text{ for each } \ell \in [L], \quad z^* = F_T\left(k, (x^*_1, \ldots, x^*_L)\right).$$
- *For each $\ell \in [L]$, the group element u^*_ℓ is uniform in $\mathcal{X}_\ell$.*
- *The group element v^* is uniform in $\mathcal{Z}$.*

Proof. The proof of this lemma is essentially an extension of the proof of Lemma C.3.

Non-Interactive $(L+1)$-Party Key-Exchange. We present a black-box non-interactive $(L+1)$-party key-exchange protocol from any L-composable IHwPRF. Our protocol is in the standard model, and involves $(L+1)$ non-uniform PPT algorithms $\{\mathcal{A}_\ell = (\mathcal{A}_{\ell,0}, \mathcal{A}_{\ell,1})\}_{\ell \in [0,L]}$ such that the algorithms $\{\mathcal{A}_{\ell,b}\}_{\ell \in [0,L]}$ operate in parallel for $b \in \{0,1\}$.

- $\mathsf{Setup}(1^\lambda)$: The algorithm creates a description $\mathcal{F}^{(L)}_{\mathrm{IHCwPRF}}$ for an L-composable IHwPRF consisting of the following functions and "composers":
$$\{F_\ell : \mathcal{K} \times \mathcal{X}_\ell \to \mathcal{Y}_\ell\}_{\ell \in [L]} \quad , \quad \left\{C_\ell : \mathcal{Y}_\ell \times \mathcal{X}_{L \setminus \{\ell\}} \to \mathcal{Z}\right\}_{\ell \in [L]}.$$
It fixes $n > 3\log\left(\max\left(|\mathcal{X}_1| \ldots, |\mathcal{X}_L|\right)\right)$, uniformly samples $\mathbf{x}^{(\ell)} \leftarrow \mathcal{X}_\ell^{2n}$ for each $\ell \in [L]$, and outputs the public parameter pp as
$$\mathsf{pp} = \left(\mathcal{F}_{\mathrm{IHCwPRF}}, \left\{\mathbf{x}^{(\ell)}\right\}_{\ell \in [L]}\right).$$
- $\mathcal{A}_{0,0}(\mathsf{pp})$: The algorithm $\mathcal{A}_{0,0}$ samples $k \leftarrow \mathcal{K}$ and computes $\mathbf{y}^{(\ell)} = F_\ell\left(k, \mathbf{x}^{(\ell)}\right)$ for each $\ell \in [L]$. It then outputs $(\mathsf{st}_0, \mathbf{y})$, where
$$\mathsf{st}_0 = k, \quad \mathbf{y} = \left\{\mathbf{y}^{(\ell)}\right\}_{\ell \in [L]}.$$

- $\mathcal{A}_{\ell,0}\left(\mathsf{pp}\right)$: For each $\ell \in [L]$, the algorithm $\mathcal{A}_{\ell,0}$ samples $\mathbf{a}^{(\ell)} \leftarrow \{0,1\}^n$ and outputs $(\mathsf{st}_\ell, x^*_\ell)$, where
$$\mathsf{st}_{\mathcal{A}} = \mathbf{a}^{(\ell)}, \quad x^*_\ell = \left\langle\!\left\langle \mathbf{x}^{(\ell)}, \mathbf{a}^{(\ell)} \right\rangle\!\right\rangle.$$
- $\mathcal{A}_{0,1}\left(\mathsf{pp}, \mathsf{st}_0, \{x^*_\ell\}_{\ell\in[L]}\right)$: The algorithm $\mathcal{A}_{0,1}$ chooses some $\ell \in [L]$ to compute the final key as
$$\mathsf{k}^* = C_\ell\left(F_\ell\left(k, x^*_\ell\right), x^*_1, \ldots, x^*_{\ell-1}, x^*_{\ell+1}, \ldots, x^*_L\right).$$
- $\mathcal{A}_{\ell,1}\left(\mathsf{pp}, \mathsf{st}_\ell, \{x^*_\ell\}_{\ell\in L\setminus\{\ell\}}, \mathbf{y}\right)$: For each $\ell \in [L]$, the algorithm $\mathcal{A}_{\ell,1}$ computes the final key as
$$\mathsf{k}^* = C_\ell\left(\left\langle\!\left\langle \mathbf{y}^{(\ell)}, \mathbf{a}^{(\ell)} \right\rangle\!\right\rangle, x^*_1, \ldots, x^*_{\ell-1}, x^*_{\ell+1}, \ldots, x^*_L\right).$$

Correctness and Security. Correctness follows from the properties of an L-composable IHwPRF. When the protocol is correctly executed, in all cases the final key computed is:
$$\mathsf{k}^* = F_T\left(k, \left(\left\langle\!\left\langle (\mathbf{x}^{(1)}, \mathbf{a}^{(1)} \right\rangle\!\right\rangle, \ldots, \left\langle\!\left\langle \mathbf{x}^{(L)}, \mathbf{a}^{(L)} \right\rangle\!\right\rangle\right)\right).$$
Security follows from Lemma C.9.

On Separation from Algebraic Maps. We have seen in the previous sections that weak PRFs, when endowed with the structure of group homomorphism over the input/output spaces, imply a wide range of sophisticated cryptographic primitives. A further strengthening of IHwPRFs via adding a 2-composability structure allowed us to obtain 3-party non-interactive key-exchange, which is not known from standard IHwPRFs. We also obtained a simple black-box construction of IBE analogous to [BF01] from this primitive, while standard IHwPRFs are known to imply IBE only in a non-black-box way. Historically, these two primitives were the seeds to using *bilinear pairing* groups.

This leads us to ponder the following questions: can we get primitives and patterns of construction that are traditionally instantiated by bilinear pairing groups from 2-composable IHwPRFs? What are the structural similarities and differences between 2-composable IHwPRFs and bilinear maps? In this section, we partially answer these questions.

We first observe that the following three properties are common between 2-composable IHwPRFs and bilinear pairing groups, and are instrumental in giving rise to the 3-party KE and Boneh-Franklin-style IBEs:

1. Homomorphism in both coordinates:
$$F_1\left(k, \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{s}_1 \right\rangle\!\right\rangle\right) = \left\langle\!\left\langle F_1(k, \mathbf{x}^{(1)}), \mathbf{s}_1 \right\rangle\!\right\rangle \quad \leftrightsquigarrow \quad (g_1^{kx})^a = g_1^{k(xa)}$$
$$F_2\left(k, \left\langle\!\left\langle \mathbf{x}^{(2)}, \mathbf{s}_2 \right\rangle\!\right\rangle\right) = \left\langle\!\left\langle F_2(k, \mathbf{x}^{(2)}), \mathbf{s}_2 \right\rangle\!\right\rangle \quad \leftrightsquigarrow \quad (g_2^{kx})^b = g_2^{k(xb)}$$
2. Composition:
$$C(F_1(k, x^{(1)}), x^{(2)}) = C(F_2(k, x^{(2)}), x^{(1)}) \quad \leftrightsquigarrow \quad e(g_1^{kx}, g_2^y) = e(g_1^x, g_2^{ky})$$
3. Pseudorandomness of composed result:
$$\begin{array}{ccc}
\left(\begin{array}{c} \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \\ \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{s}_1 \right\rangle\!\right\rangle, F_1(k, \mathbf{x}^{(1)}), \\ \left\langle\!\left\langle \mathbf{x}^{(2)}, \mathbf{s}_2 \right\rangle\!\right\rangle, F_2(k, \mathbf{x}^{(2)}), \\ F_T(k, (\left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{s}_1 \right\rangle\!\right\rangle, \left\langle\!\left\langle \mathbf{x}^{(2)}, \mathbf{s}_2 \right\rangle\!\right\rangle)) \end{array}\right) & \leftrightsquigarrow & \left(\begin{array}{c} g_1, g_2, \\ g_1^x, g_1^k, \\ g_2^y, g_2^k, \\ e(g_1, g_2)^{kxy} \end{array}\right) \\
\overset{c}{\approx} & & \overset{c}{\approx} \\
\left(\begin{array}{c} \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \\ \left\langle\!\left\langle \mathbf{x}^{(1)}, \mathbf{s}_1 \right\rangle\!\right\rangle, F_1(k, \mathbf{x}^{(1)}), \\ \left\langle\!\left\langle \mathbf{x}^{(2)}, \mathbf{s}_2 \right\rangle\!\right\rangle, F_2(k, \mathbf{x}^{(2)}), \\ \$ \end{array}\right) & \leftrightsquigarrow & \left(\begin{array}{c} g_1, g_2, \\ g_1^x, g_1^k, \\ g_2^y, g_2^k, \\ \$ \end{array}\right)
\end{array}$$

In the bilinear groups setting, the pseudorandomness property essentially follows from an asymmetric version of the Bilinear DDH assumption [BF01].

We now consider the more sophisticated class of dual-system [Wat09] based constructions that can be instantiated from bilinear pairing groups. One of the simplest such constructions is the Quasi Adaptive NIZK (QA-NIZK [JR13]) construction given by [KW15]. We sketch the construction from bilinear pairings below, with a *potential* counterpart construction from 2-Composable IHwPRFs presented alongside. Following [EHK+13], the notation $[a]_i$ means g_i^a and it is naturally extended to vectors and matrices.

Language:

$$\begin{array}{ccc} \text{Parameter } \mathbf{M} \in \mathbb{Z}_q^{n\times t} & \leftrightsquigarrow & \text{Parameter } \mathbf{m} \in \mathcal{X}_1^{2n} \\ L_{\mathbf{M}} = \{[\mathbf{Mw}]_1 : \mathbf{w} \in \mathbb{Z}_q^t\} & \leftrightsquigarrow & L_{\mathbf{m}} = \{\langle\!\langle \mathbf{m}, \mathbf{w} \rangle\!\rangle : \mathbf{w} \in \{0,1\}^n\} \end{array}$$

CRS:

$$\begin{array}{lccc} \text{Sample:} & \mathbf{K} \leftarrow \mathbb{Z}_q^{t\times(k+1)}, \mathbf{A} \leftarrow \mathcal{D}_k & \leftrightsquigarrow & k \leftarrow \mathcal{K}, a \leftarrow \mathcal{X}_2 \\ \text{Prover CRS:} & [\mathbf{M}^\top \mathbf{K}]_1 & \leftrightsquigarrow & F_1(k, \mathbf{m}) \\ \text{Verifier CRS:} & [\mathbf{A}]_2, [\mathbf{KA}]_2 & \leftrightsquigarrow & a, F_2(k, a) \end{array}$$

Prover:

$$\begin{array}{lccc} \text{Input:} & \mathbf{x} = [\mathbf{Mw}]_1 & \leftrightsquigarrow & x = \langle\!\langle \mathbf{m}, \mathbf{w} \rangle\!\rangle \\ \text{Proof:} & \pi = \mathbf{w}^\top [\mathbf{M}^\top \mathbf{K}]_1 & \leftrightsquigarrow & \pi = \langle\!\langle F_1(k, \mathbf{m}), \mathbf{w} \rangle\!\rangle \end{array}$$

Verifier:

$$\begin{array}{lccc} \text{Word and proof:} & \mathbf{x} \in \mathbb{G}_1^n,\ \pi \in \mathbb{G}_1^{k+1} & \leftrightsquigarrow & x \in \mathcal{X}_1,\ \pi \in \mathcal{Y}_1 \\ \text{Check:} & e(\mathbf{x}^\top, [\mathbf{KA}]_2) = e(\pi, [\mathbf{A}]_2) & \leftrightsquigarrow & C_2(F_2(k,a), x) = C_1(\pi, a) \end{array}$$

Zero-Knowledge:

$$\begin{array}{lccc} \text{Trapdoor:} & \mathbf{K} & \leftrightsquigarrow & k \\ \text{Simulator input:} & \mathbf{x} \in \mathbb{G}_1^n & \leftrightsquigarrow & x \in \mathcal{X}_1 \\ \text{Simulated proof:} & \pi = \mathbf{x}^\top \mathbf{K} & \leftrightsquigarrow & \pi = F_1(k, x) \end{array}$$

As we see, the above construction satisfies correctness and zero-knowledge. However, there are several issues and limitations:

1. The language $L_{\mathbf{m}} = \{\langle\!\langle \mathbf{m}, \mathbf{w} \rangle\!\rangle : \mathbf{w} \in \{0,1\}^n\}$ may actually be trivial for large enough (but still poly) n. That is, it may be equal to almost all of $\mathcal{X}_1$. In applications we usually require the language to be proved to be a hard subset. To make the language non-trivial, we choose n appropriately so that $|L_{\mathbf{m}}|/|\mathcal{X}_1| = \text{negl}(\lambda)$.

2. Soundness condition is that for $x \notin L_{\mathbf{m}}$, it is hard to find π such that $C_2(F_2(k,a),x) = C_1(\pi,a)$, or equivalently, $C_1(F_1(k,x),a) = C_1(\pi,a)$. This at least requires that it is hard to find $x \notin L_{\mathbf{m}}$ and π, such that $F_1(k,x) = \pi$. However, this is not true as $2m_1 \notin L_{\mathbf{m}}$ with high probability and $F_1(k, 2m_1) = 2F_1(k, m_1)$, which is easy to compute. Essentially, not being a subset-sum doesn't still rule out other scalar products with non-0,1 coefficients, and many of these quantities will still have easy to compute proofs.

3. We observe that since k is the only quantity that acts on both coordinates, the most natural construction is the one we took - that is, having k play the role of the $\mathbf{K}$ in the bilinear setting. Deviating from this choice would require substantial departure from the flavor of the construction.

To work around the soundness aspect, let's allow $\mathbf{w}$ to be an integer vector and subset-summing operation to be replaced by inner product. The requirement then becomes the following: given $(\mathbf{m}, a, F_1(k, \mathbf{m}), F_2(k, a))$, it is hard to find $x \notin \mathit{GeneratedSubset}(\mathbf{m})$ and π, such that $C_1(F_1(k, x), a) = C_1(\pi, a)$. This is implied by the $\mathcal{D}_k$ Matrix-DDH assumption in the bilinear group setting along with the fact that the wPRF instantiation in this setting is *key homomorphic*.

Unfortunately, this hardness is not captured by the wPRF security alone of C_1 and F_1. The reason is similar to why it's difficult to construct worst case SPHFs from IHwPRFs. Since x is adversarially chosen, it could potentially belong to a small subgroup of $\mathcal{X}_1$, where $F_1(k, x)$ is easy to compute - either regardless of k or after seeing $(\mathbf{m}, F_1(k, \mathbf{m}))$.

Given this more concrete background, we argue that the general abstraction of Dual Systems Groups (DSG [CGW15]) is hard to capture in the 2-Composable IHwPRF setting:

1. The set that parallels the notion of keys in the 2-Composable IHwPRF setting is denoted by $\mathcal{W}$ in [CGW15]. Essentially objects from $\mathcal{W}$ are the ones that interact with both the coordinates. The paper defines something called "predicate encodings" for several different types of Attribute-based Encryptions (ABE). All the interesting ABEs beyond IBE essentially require algebra in the $\mathcal{W}$ space. This translates to the requirement of key homomorphism in the IHwPRF setting.

2. The property of "parameter hiding" required in proving security in the DSG setting requires algebraic interaction of both the coordinates. To realize this in the IHwPRF setting forces both the coordinate domains $\mathcal{X}_1$ and $\mathcal{X}_2$ to be *ring homomorphic* to a single ring, where all the algebra can take place.

The currently known constructions of rich ABEs like fuzzy IBEs [SW05], spatial encryption [BH08] and monotone span program ABEs [GPSW06] from bilinear groups all require at least one of the properties just described. Since the only instantiation of 2-composable IHwPRFs we know of are bilinear groups, it seems difficult to achieve these rich ABEs without restricting 2-composable IHwPRFs to almost traditional bilinear groups.

Thus we see a seeming separation in the amount of structure that we need for 3-party key exchange and simple IBE (in RO) from that seemingly necessary for NIZKs (without RO) and rich ABEs. This poses a tantalizing question:

> Can we construct a 3-party non-interactive key exchange protocol from a weaker primitive than bilinear pairing groups?

In other words, can we achieve the structure of 2-composability from concrete assumptions, e.g., lattice-based assumptions, that do not naturally imply bilinear pairings?

L-Composable IHwPRFs vs. L-multilinear maps. Similar separations come to the fore as we raise the composability/multilinearity levels. On the positive side, we get $(L + 1)$-party key exchange from an L-Composable IHwPRF, which is not known from $(< L)$-Composable IHwPRFs. We also do not know how to construct such a protocol from hard $(< L)$-multilinear groups.

On the other hand, we have constructions of ABE for general circuits [GGH+13c] and indistinguishability obfuscation (iO) for NC^1 from hard multilinear groups [GGH+13b]. These constructions require algebraic interaction of the different coordinates, and hence, just like the DSG setting for bilinear groups, require the input domains to all be ring-homomorphic to a single ring. In fact, some computations like "Killian"-izing Barrington matrices for iO requires the input domains to be field-homomorphic to a single field, so that matrix inversions can be performed.

Thus we have an analogous seeming separation in the amount of structure that we need for L-party key exchange from that seemingly necessary for circuit ABEs and iOs. The corresponding open question is whether we can build the former from weaker primitives that may lack the structure needed for the latter.

D Ring IHwPRF and FHE

In this part, we show that Ring IHwPRF and fully homomorphic encryption (FHE) are equivalent provided that the order of the output ring is polynomial in the security parameter. First we provide a formal definition of Ring IHwPRF.

Definition D.1. (Ring Input-Homomorphic Weak PRF.) We call a family of functions $\{F(k,\cdot) : \boxed{R} \to R\}_{k\in\mathcal{K}}$ a Ring IHwPRF (RIHwPRF) family if the following conditions hold:

1. $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is a *weak PRF* family.
2. $\left(\boxed{R}, \boxplus, \boxtimes\right)$ and $(R,+,\times)$ are both efficiently samplable rings.
3. The ring operations $(\boxplus, \boxtimes)$ and $(+,\times)$, and the inverse operation of the additive group in each ring are efficiently computable.
4. For every $k \in \mathcal{K}$, the mapping $F(k,\cdot) : \boxed{R} \to R$ is a ring homomorphism from $\boxed{R}$ to R.

Remark D.2. A γ-bounded RIHwPRF is defined similar to Definition 3.4 where we have $\gamma = (\gamma_+, \gamma_\times)$, and γ_+ (respectively $\gamma_\times$) is an apriori bound on the number of possible homomorphic operations for the operation $+$ (respectively $\times$).

It is easy to see that a slight modification of Lemma 3.21 shows that a ring-homomorphic public-key encryption scheme (which is implied by an FHE) yields an RIHwPRF family. Below, we show that an RIHwPRF family implies a symmetric-key FHE if the order of output ring is polynomial in the security parameter. Notice that one can construct a public-key FHE scheme from a private-key one using the transformation given in [Rot11].

Lemma D.3. *Let $F : \mathcal{K} \times \boxed{R} \to R$ be a (γ-bounded) RIHwPRF family where $\gamma = (\gamma_+, \gamma_\times)$, and let R be a ring such that $|R| \le \mathrm{poly}(\lambda)$ where λ is the security parameter. Given F, there is a black-box construction of a (leveled) symmetric-key FHE such that the maximum number of allowed NAND operations is $\min\{\gamma_+, \gamma_\times\}/2$. In particular, an RIHwPRF with unbounded homomorphism implies an FHE scheme with unbounded number of allowed NAND operations.*

Proof. We sketch the construction and omit the details of the security proof. The security follows from a standard reduction that constructs a distinguisher against the weak pseduradomness of F given any attacker with non-negliginle advantage against the security of the symmetric-key encryption scheme.

- $\mathsf{Gen}(1^\lambda)$: Sample a key $k \leftarrow \mathcal{K}$ from the key space of the Ring IHwPRF. Publish a description $\mathcal{F}_{\mathrm{RIHwPRF}}$ of the Ring IHwPRF as the public parameter, along with the key k.
- $\mathsf{Enc}(k \in \mathcal{K}, \mathsf{m} \in \{0,1\})$: Let 0_R (and 1_R) be the identity element of the ring R with respect to addition (and multiplication). To encrypt a bit m under the secret key k, sample a preimage of m_R in $\boxed{R}$. Such a preimage can be efficiently sampled since the order of the ring is polynomial in λ, and by the weak pseudorandomness of F we have

$$\left| \Pr_{k\leftarrow\mathcal{K}, \boxed{r}\leftarrow\boxed{R}} [F(k,\boxed{r}) = 0_R] - 1/|R| \right| \le \mathrm{negl}(\lambda).$$

  The algorithm outputs $\mathsf{ct} = \boxed{r_{\mathsf{m}}}$ as the ciphertext where $\boxed{r_{\mathsf{m}}}$ is a (randomly) sampled preimage of m_R.

- $\mathsf{Dec}(k \in \mathcal{K}, \boxed{r})$: The decryption algorithm outputs $\mathsf{m}' \in \{0,1\} \cap \{\perp\}$ where

$$\mathsf{m}' = \begin{cases} 0 & \text{if } F(k, \boxed{r}) = 0_R \\ 1 & \text{if } F(k, \boxed{r}) = 1_R \\ \perp & \text{otherwise.} \end{cases}$$

- $\mathsf{NAND}(\mathsf{ct}, \mathsf{ct}')$: Given $\mathsf{ct} \in \boxed{R}$ and $\mathsf{ct}' \in \boxed{R}$, output

$$\boxed{1} \boxminus \mathsf{ct} \boxtimes \mathsf{ct}',$$

  where $\boxed{1}$ is the identity element of $\boxed{R}$ with respect to addition, and $\boxminus$ is the subtraction in the ring $\boxed{R}$. By ring-homomorphism of F, it is easy to see that if ct and ct' are well-formed ciphertexts such that $\mathsf{ct} \leftarrow \mathsf{Enc}(k, \mathsf{m} \in \{0,1\})$ and $\mathsf{ct}' \leftarrow \mathsf{Enc}(k, \mathsf{m}' \in \{0,1\})$ we have

$$\mathsf{Dec}(k, \mathsf{NAND}(\mathsf{ct}, \mathsf{ct}')) = \mathsf{Dec}(k, \boxed{1} \boxminus \mathsf{ct} \boxtimes \mathsf{ct}') = \mathsf{NAND}(\mathsf{m}, \mathsf{m}').$$

  Observe that to do a NAND operation, we need one subtraction and one multiplication in the ring $\boxed{R}$, hence the maximum number of allowed NAND operations is $\min\{\gamma_+, \gamma_\times\}/2$. □

Remark D.4. It is easy to see that the construction also works for an RIHwPRF $F : \mathcal{K} \times \boxed{R} \to R$ with an arbitrarily large output ring if (1) the output ring R has an ideal $\mathcal{I}$ of polynomial index and (2) the secret key for RIHwPRF determines the coset that corresponds to output of the RIHwPRF. In this case, one can define another RIHwPRF F' with polynomial order output ring where $F' : \mathcal{K} \times \boxed{R} \to R/\mathcal{I}$, and use the elements $\mathcal{I}$ and $1_R + \mathcal{I}$ of the quotient ring to simulate NAND operation.