Symmetric Primitives with Structured Secrets

Navid Alamati* Hart Montgomery † Sikhar Patranabis‡

Abstract

Securely managing encrypted data on an untrusted party is a challenging problem that has motivated the study of a variety of cryptographic primitives. A special class of such primitives allows an untrusted party to transform a ciphertext encrypted under one key to a ciphertext under another key, using some auxiliary information that does not leak the underlying data. Prominent examples of such primitives in the symmetric-key setting are key-homomorphic PRFs, updatable encryption, and proxy re-encryption. Although these primitives differ significantly in terms of their constructions and security requirements, they share two important properties: (a) they have *secrets with structure or extra functionality*, and (b) all known constructions of these primitives satisfying reasonably strong definitions of security are based on *concrete* public-key assumptions, e.g., DDH and LWE.

This raises the question of whether these objects inherently belong to the world of public-key primitives, or they can potentially be built from simple symmetric-key objects such as pseudorandom functions. In this work, we show that the latter possibility is unlikely. More specifically, we show that:

- Any (bounded) key-homomorphic *weak* PRF with an abelian output group implies a (bounded) *input-homomorphic* weak PRF, which has recently been shown to imply not only public-key encryption (PKE), but also a variety of primitives such as PIR, lossy TDFs, and even IBE.
- Any ciphertext-independent updatable encryption scheme that is forward and post-compromise secure implies PKE. Moreover, any symmetric-key proxy re-encryption scheme with reasonably strong security guarantees implies a forward and post-compromise secure ciphertext-independent updatable encryption, and hence PKE.

In addition, we show that unbounded (or exact) key-homomorphic weak PRFs over abelian groups are *impossible* in the quantum world. In other words, over abelian groups, bounded key-homomorphism is the best that we can hope for in terms of post-quantum security. Our attack also works over other structured primitives with abelian groups and exact homomorphisms, including homomorphic one-way functions and input-homomorphic weak PRFs.

*University of Michigan.
†Fujitsu Laboratories of America.
‡Indian Institute of Technology, Kharagpur. Part of the work was done while the author was an intern at Fujitsu Laboratories of America.

1 Introduction

Examining the practicality and security of cryptographic primitives has always been one of the most important aspects of cryptographic research. When a new cryptographic protocol is developed, it is often somewhat inefficient and relies on a relatively strong assumption. We might ask a question that captures the essence of "lower bounds" for cryptographic algorithms: is it possible to improve this cryptosystem, or is the proposed scheme close to optimal?

A plausible approach for understanding the gap between known constructions and "reasonable" lower bounds is to determine the power of a cryptographic primitive, i.e., what other cryptographic objects can be built from it in a generic way. For instance, if a certain primitive is known to imply public-key encryption (PKE), then it does not seem likely that this primitive can be built in a generic manner from one-way functions (OWFs) [IR89, GHMM18]. However, for certain classes of primitives, this gap might be substantial.

One such class of primitives that has been studied considerably is what we will term *symmetric primitives with structured secrets*. Perhaps the most iconic member of this (informal) class, and one we will use to illustrate our points here, is the *key-homomorphic PRF*. Recall that, informally, a key-homomorphic PRF is a function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with key space $\mathcal{K}$ and output space $\mathcal{Y}$ endowed with group operations $\oplus$ and $\otimes$, respectively, that meets all of the requirements of a pseudorandom function with the following extra property:[1]

$$F(k_1, x) \otimes F(k_2, x) = F(k_1 \oplus k_2, x).$$

Key-homomorphic PRFs (KHPRFs) were first implicitly shown in [NPR99] in the random oracle model and then formally defined and constructed in the standard model in [BLMR13]. There are a number of interesting applications of KHPRFs, including primitives like distributed PRFs [NPR99, BLMR13, LST18], updatable encryption [EPRS17, LT18], and PRFs that are secure against related key attacks [LMR14].

Since [BLMR13], there have been a number of works constructing improved variants of KHPRFs [BP14, BV15, BFP+15]. However, despite this quantity of research, the known constructions of KHPRFs still require powerful assumptions. For instance, we only know how to build exact key-homomorphic PRFs in the standard model from multilinear maps or related assumptions [BLMR13]. If we relax these requirements to almost KHPRFs in the standard model, all known constructions still require an LWE assumption with superpolynomial modulus. Even constructions in the random oracle model require public-key assumptions like DDH [NPR99].

All of these assumptions and constructions are seemingly very heavyweight for an ostensibly symmetric-key primitive that is typically targeted for applications in the symmetric-key setting. This leads us to a natural question: can we construct more efficient key-homomorphic PRFs, or is there some fundamental lower bound limiting their efficiency? Boneh et. al state, optimistically, "Another interesting area of research is to construct key-homomorphic PRFs whose performance is comparable to real-world block ciphers such as AES," [BLMR13] but so far there are no known realizations of such a construction.

However, key-homomorphic PRFs are far from the only symmetric primitive with structured secrets for which the gap between known constructions and lower bounds appears to be relatively large. There are a number of other seemingly symmetric-key primitives that are only known to be implementable from concrete public-key assumptions.

Updatable Encryption. Suppose that Alice wants to perform key rotation on encrypted data in the cloud, but does not trust the cloud with her secret key. *Updatable encryption*, first defined in [BLMR13] as an application of key-homomorphic PRFs, allows third parties to periodically rotate encryption keys by moving ciphertexts from an old key to a new one, without actually learning the contents of the ciphertexts.

Boneh *et al.* [BLMR13] proposed the first formal definitions and concrete realizations of updatable encryption, which were subsequently refined by Everspaugh *et al.* in [EPRS17]. In a more recent work, Lehmann and Tackmann [LT18] introduced stronger security notions for updatable encryption that are desirable for real-world applications, and also pointed out that none of the existing constructions satisfy these notions. They addressed this issue by presenting a new, non-KHPRF updatable encryption protocol called RISE that achieves these stronger security requirements.

However, all of the constructions from the stronger security assumptions in [LT18] are either built from key-homomorphic PRFs or from concrete public-key assumptions. Yet again, the question remains: can we build similar

[1] We note that this equality can be relaxed to achieve *approximate* key-homomorphic PRFs, and these (approximate) key-homomorphic PRFs can be built from lattice-based assumptions, like LWE.

schemes using simple symmetric-key primitives? Lehmann and Tackmann [LT18] are pessimistic, "secure updatable encryption schemes seem to inherently require techniques from the public-key world" but no formal bounds were given.

Proxy Re-Encryption. A *proxy re-encryption* scheme is a cryptosystem where, given a special update token, a third party can transform a ciphertext encrypted under Alice's public key to a ciphertext encrypted under Bob's public key, while learning nothing about the underlying message. Proxy re-encryption was initially developed in [BBS98] and then formalized in [AFGH05, AFGH06]. A number of subsequent works proposed improved schemes, including CCA-secure proxy re-encryption [CH07], identity-based proxy re-encryption [GA07], and CCA-secure unidirectional proxy re-encryption [LV08].

Proxy re-encryption has also been studied extensively in the symmetric-key setting [SNS11]. In particular, many of the proposed definitions and security notions associated with proxy re-encryption [nBL17, DKL+18, FKKP19] can be adopted to the symmetric-key setting. Interestingly, while some of the simpler definitions of security may be realized from known symmetric primitives, the stronger definitions only have known realizations from public-key assumptions like DDH and LWE [ABPW13, CCL+14]. This leads to the following question: are these stronger definitions of symmetric-key proxy re-encryption achievable from symmetric-key encryption?

Downside of Structure. A common property underlying each of the cryptographic primitives discussed so far is that they have structured secrets. While the presence of structure potentially allows building rich cryptosystems from simple primitives, it may also make these primitives vulnerable to potential attacks [Bar17]. This motivates us to pose the following question: can the structure inherent in KHwPRFs (and related primitives) lead to attacks?

1.1 Our Contributions

We show that the answer to many of these questions is negative. Our results can be summarized as follows:

Key-Homomorphic Weak PRFs. We show that any key-homomorphic *weak* PRF (KHwPRF) $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with an abelian output group $\mathcal{Y}$ implies PKE. In fact, we show that KHwPRFs with abelian output groups imply a much stronger primitive called *input-homomorphic weak PRF* (IHwPRF) which, by the recent work of [AMPR19], implies a large number of public-key primitives, including identity-based encryption [Sha84], private information retrieval [KO97], and lossy trapdoor functions [PW08]. In essence, our results indicate that it is seemingly unlikely that KHwPRFs, and hence KHPRFs, with abelian output groups are implied by symmetric-key primitives [IR89, GHMM18]. Our results also hold for *bounded* KHwPRFs with abelian output groups (encompassing nearly all applications of *almost* KHPRFs from lattice-based assumptions). To our knowledge, all existing constructions of KHwPRFs (and almost KHwPRFs) have abelian output groups.

Interestingly, our constructions of PKE and IHwPRF *only* use the output group $\mathcal{Y}$ of the KHwPRF. We use the security of the KHwPRF to argue security of our constructions. It may be possible that this seemingly novel construction technique has other applications.

These results on KHwPRFs lend evidence to support the idea that many "symmetric-key" cryptosystems that are currently only known from KHPRFs, do, in fact, belong to the world of public-key primitives. We note that some primitives (such as distributed PRFs) have KHPRF-based constructions that require abelian key and output groups, further strengthening the argument that these constructions are unlikely to be built from symmetric-key primitives.

Finally, we show how to construct a Naor-Reingold style PRF [NR97] from any key-homomorphic weak PRF. As we explain in Section 3.4, this allows us to construct highly parallel and potentially efficient PRFs from any KHwPRF. To the best of our knowledge, prior to this work, it was not known how to construct a Naor-Reingold style PRF from a *generic* primitive.

Updatable Encryption. We show that any ciphertext-independent updatable encryption scheme that satisfies the adaptive notions of forward and post-compromise security proposed in [LT18] implies PKE. As pointed out in [LT18], forward and post-compromise security are desirable for real-world applications, since they guarantee that message confidentiality is preserved even in the presence of temporary key compromise. Our result confirms the pessimism

expressed in [LT18] that updatable encryption schemes with desirable security properties inherently belong to the class of asymmetric primitives.

Proxy Re-Encryption. We show that any symmetric-key proxy re-encryption scheme that satisfies an adaptive notion of update indistinguishability implies updatable encryption with forward and post-compromise security, and hence PKE. We remark that our security definition for symmetric-key proxy re-encryption is a slight upgradation of the definition presented in [FKKP19] in the sense that our definition captures indistinguishability of updates in addition to the traditionally desirable properties of proxy re-encryption, such as unidirectionality and adaptive indistinguishability of encryption. Our definition also unifies the security notions achieved by a large number of existing constructions [AFGH05, AFGH06, ABH09, CCL+14].

Quantum Attacks on Primitives with Structure. We show that any exact (not bounded) homomorphic one-way function (HOWF) with abelian input and output groups can be broken in polynomial time using a quantum computer. This immediately rules out the existence of abelian, exact KHwPRFs (and hence KHPRFs) in the quantum setting. In other words, over abelian groups, KHwPRFs (and KHPRFs) with bounded homomorphism are the best that we can hope for in the quantum world.

We can also extend this attack to essentially all exact input-homomorphic weak unpredictable functions (IHwUFs) and IHwPRFs over abelian groups using the results from [AMPR19], which in turn yields quantum attacks on essentially all exact (group-)homomorphic encryption schemes over abelian groups. We note that a similar result with respect to homomorphic encryption was achieved in [AGKP14], albeit using different techniques.

1.2 Related Works

We have already discussed a number of papers related to key-homomorphic PRFs, updatable encryption, and proxy re-encryption. However, we want to note the construction [DKPW12] of Dodis *et al.* which showed how to build efficient MACs from key-homomorphic weak PRFs, even predating [BLMR13].

Previous works have studied the relationship between cryptographic primitives and structure. Recently, [AMPR19] examined simple primitives with structured inputs. In a work on a similar topic, Pietrzak and Sjödin [PS08] showed that weak PRFs with a certain input property imply PKE. In a different line of works that show PKE from other primitives, Berman *et al.* showed that laconic zero-knowledge protocols imply PKE [BDRV18], Fischlin and Harasser [FH18] showed that PKE is implied by invisible sanitizable signatures, and Rothblum [Rot11] demonstrated (homomorphic) PKE from a secret-key encryption scheme with some form of weak homomorphism. For a comprehensive treatment of PRFs and related primitives, see [BR17].

1.3 Technical Overview

In this section, we explain at a high level the techniques behind our constructions and proofs.

Key-Homomorphic Weak PRFs. Informally, a key-homomorphic weak PRF is a function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with keyspace $\mathcal{K}$ and output space $\mathcal{Y}$ endowed with group operations $\oplus$ and $\otimes$, respectively, that meets the definition of a weak pseudorandom function[1] with the following extra property:

$$F(k_1, x) \otimes F(k_2, x) = F(k_1 \oplus k_2, x).$$

As a warm-up, we first show that a KHwPRF with an abelian output group $\mathcal{Y}$ implies PKE. To illustrate how this works, we will show how our construction works with a simple DDH-based KHwPRF from [NPR99][2] in parallel with a generic construction. First, we use the following notation for two weak PRFs:

[1] A weak PRF is a PRF for which the pseudorandomness guarantee holds when the inputs are sampled uniformly at random.

[2] This was originally envisioned by the authors of [NPR99] as a PRF in the random oracle model, but we note that it is equivalent to a weak PRF in the standard model.

Generic KHwPRF	DDH Instantiation
$F(k \in \mathcal{K}, x \in \mathcal{X}) \in \mathcal{Y}$	$F_{DDH}(g \in \mathbb{G}, k \in \mathbb{Z}_q) \in \mathbb{G}$
$F(k, x) = y$	$F_{DDH}(g, k) = g^k$

Now consider many instances of the same KHwPRF in parallel, with different keys. By a hybrid argument, we know that such a set of KHwPRF outputs is still indistinguishable from random. One can visualize this as follows:

Generic KHwPRF				DDH Instantiation			
$F(k_1, x_1)$	$F(k_2, x_1)$	$\dots$	$F(k_\ell, x_1)$	$g_1^{k_1}$	$g_1^{k_2}$	$\dots$	$g_1^{k_\ell}$
$F(k_1, x_2)$	$F(k_2, x_2)$	$\dots$	$F(k_\ell, x_2)$	$g_2^{k_1}$	$g_2^{k_2}$	$\dots$	$g_2^{k_\ell}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$F(k_1, x_m)$	$F(k_2, x_m)$	$\dots$	$F(k_\ell, x_m)$	$g_m^{k_1}$	$g_m^{k_2}$	$\dots$	$g_m^{k_\ell}$

Now suppose we take a random "subset sum"[1] of the columns of these many instances of KHwPRFs in parallel. If $\mathbf{s} = (s_1, \dots, s_\ell) \in \{0,1\}^\ell$ is a random vector denoting our subset sum choice, we get new "columns" as follows:

Generic KHwPRF	DDH Instantiation
$\bigotimes_{j=1}^{\ell} s_j \cdot F(k_j, x_1) = F(k^*, x_1)$	$\prod_{j=1}^{\ell} g_1^{s_j \cdot k_j} = g_1^{k^*}$
$\bigotimes_{j=1}^{\ell} s_j \cdot F(k_j, x_2) = F(k^*, x_2)$	$\prod_{j=1}^{\ell} g_2^{s_j \cdot k_j} = g_2^{k^*}$
$\vdots$	$\vdots$
$\bigotimes_{j=1}^{\ell} s_j \cdot F(k_j, x_m) = F(k^*, x_m)$	$\prod_{j=1}^{\ell} g_m^{s_j \cdot k_j} = g_m^{k^*}$

If $\ell > 3 \log |\mathcal{K}|$, then the distribution of $k^* = \oplus_{j=1}^{\ell} s_j k_j$ will be statistically close to uniform over $\mathcal{K}$. This can be shown by a relatively simple application of the leftover hash lemma [IZ89]. It now follows by the pseudorandomness of the KHwPRFs F and F_{DDH} that these new columns are computationally indistinguishable from random, even given the outputs of the other columns.

We now present the critical step of our argument: if such subset sums of KHwPRF outputs are indistinguishable from random even if the randomness for the subset sum is reused, then, by a series of hybrid arguments, it follows that similar subset sums of randomly chosen elements of the output group $\mathcal{Y}$ are indistinguishable from random: otherwise, we would have a distinguisher for the original KHwPRF. In other words, the following must hold:

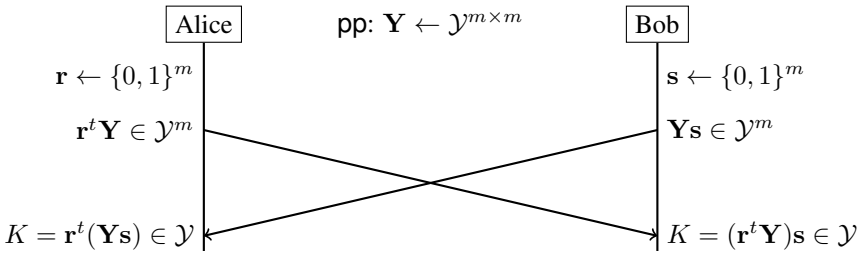
Generic KHwPRF	DDH Instantiation
$\mathbf{Y} \leftarrow \mathcal{Y}^{m \times \ell}, \mathbf{s} \leftarrow \{0,1\}^\ell$	$\mathbf{G} \leftarrow \mathbb{G}^{m \times \ell}, \mathbf{s} \leftarrow \{0,1\}^\ell$
$(\mathbf{Y}, \mathbf{Ys}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{u})$	$(\mathbf{G}, \mathbf{Gs}) \overset{c}{\approx} (\mathbf{G}, \mathbf{h})$
where $\mathbf{u} \leftarrow \mathcal{Y}^m$.	where $\mathbf{h} \leftarrow \mathbf{G}^m$.

Note that for a matrix of group elements $\mathbf{Y} \in \mathcal{Y}^{m \times \ell}$ and a vector $\mathbf{s} \in \{0,1\}^\ell$, we denote by $\mathbf{Ys} \in \mathcal{Y}^m$ the vector of group elements

$$\left(\bigotimes_{j=1}^{\ell} s_j \cdot y_{1,j}, \dots, \bigotimes_{j=1}^{\ell} s_j \cdot y_{m,j} \right).$$

Given this hard problem, which is based on the weak pseudorandomness of F, it is simple to construct a two-party noninteractive key exchange protocol (which is sufficient for PKE), as visualized in the following figure (note that the public parameter pp consists of an $m \times m$ matrix of uniformly chosen group elements for a fixed $m > 3 \log |\mathcal{K}|$).

[1] We use the term "subset sum" loosely to essentially indicate subset group-operation over the output space of the KHwPRF. Depending on whether the group is additive or multiplicative, we perform either subset sums or subset products.



It turns out that the technique described above is actually versatile enough to construct a number of stronger cryptographic primitives. More specifically, we show how to build an input-homomorphic weak PRF (IHwPRF), which by [AMPR19] implies a variety of public-key primitives.

Informally, an IHwPRF is a function $F' : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with input space $\mathcal{X}$ and output space $\mathcal{Y}$ endowed with group operations $\oplus$ and $\otimes$, respectively, that also meets the definition of a weak pseudorandom function. However, the homomorphism is over the input space rather than the key space:

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

First, note that the DDH-based KHwPRF is already input homomorphic. But the DDH assumption is very special in this regard, and we cannot guarantee that other constructions of KHwPRFs are also implicitly IHwPRFs. In general, for a KHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, the input space $\mathcal{X}$ might not even be a group.

We now illustrate the construction of an IHwPRF $F' : \{0,1\}^\ell \times \mathcal{Y}^\ell \to \mathcal{Y}$ from any KHwPRF with an abelian output group $\mathcal{Y}$ (where $\ell > 3 \log |\mathcal{K}|$):

Generic KHwPRF	DDH Instantiation
$F' : \{0,1\}^\ell \times \mathcal{Y}^\ell \to \mathcal{Y}$	$F'_{\text{DDH}} : \{0,1\}^\ell \times \mathbb{G}^\ell \to \mathbb{G}$
$F'(\mathbf{s}, (y_1, ..., y_\ell)) = \bigotimes_{j=1}^{\ell} s_j \cdot y_j$	$F'_{\text{DDH}}(\mathbf{s}, (g_1, ..., g_\ell)) = \prod_{j=1}^{\ell} g_j^{s_j}$

First, note that the input homomorphism of F' and F'_{DDH} follows from that fact that the underlying groups $\mathcal{Y}$ and $\mathbb{G}$ are abelian, respectively. If $\mathcal{Y}$ is not abelian, then F' would still be pseudorandom, but not input homomorphic. It is an interesting open problem to remove this restriction on $\mathcal{Y}$ while retaining input-homomorphism.

Notice that in the actual *constructions* of PKE and IHwPRF, we do not explicitly use the key space or the input space of the underlying KHwPRF; we essentially use the pseudorandomness of the KHwPRF to argue their security. In Section 3, we present the detailed constructions and proofs, and extend our techniques to work for *almost* KHwPRFs.

On the negative side, we rule out the existence of *exact* KHwPRFs with output groups over which a system of linear equations (with binary variables) can be solved efficiently, because such an algorithm can be used to break the hard problem instance described above, and hence to break the pseudorandomness of the underlying exact KHwPRF.[1]

Updatable Encryption. We show that any ciphertext-independent updatable encryption (UE) scheme that meets the notion of "adaptive indistinguishability of updates" formalized by Lehmann and Tackmann in [LT18] implies a PKE scheme. Recall that a UE scheme allows publishing an update token $\Delta_{0,1}$ that can be used by a third party to transform a ciphertext encrypted under a key sk_0 to a ciphertext encrypted under another key sk_1, without knowing the underlying message.

In our PKE construction from UE, the public key consists of a pair of UE ciphertexts encrypting 0 and 1 respectively under a key sk_0, and an update token $\Delta_{0,1}$. Depending on the plaintext bit b, the encryption algorithm updates one of the two ciphertexts, and the decryption algorithm in turn decrypts using the updated key sk_1. To prove CPA security, we show a reduction in which the challenge ciphertext for the UE game is transformed into the public key for the PKE game, which then allows us to switch between knowledge of secrets and knowledge of update tokens. The detailed construction and proof of security are presented in Section 4.1.

[1] We remark that known algorithms to solve systems of linear equations over abelian groups need an *explicit representation* of the group, see [GR02] for more details. For example, such an explicit representation is not known to an adversary against a DDH-hard group $\mathbb{G}$.

As a side note, our construction of PKE assumes that the update algorithm of the underlying UE scheme is *randomized*. We point out that all existing UE schemes satisfying the notion of update indistinguishability (notably, the RISE scheme in [LT18]) have randomized update algorithms.

Proxy Re-Encryption. We show that any symmetric-key proxy re-encryption scheme that satisfies the indistinguishability-based security notions with respect to encryption and update operations implies a ciphertext-independent UE scheme with indistinguishability of updates. By the result mentioned above, it thus implies a PKE scheme.

Our construction of UE from a symmetric-key PRE essentially maps PRE secret keys associated with different identifiers to UE secret keys associated with different epochs. To prove security, we show a reduction where any *valid* oracle query from the adversary in the UE game can be mapped into a corresponding *valid* oracle query to the challenger in the PRE game. The detailed construction and proof of security are presented in Section 4.2.

We remark that while existing PRE schemes typically support *multi-hop updates* [AFGH05, FKKP19], UE schemes as formalized in [LT18] support a more sequential flavor of updates. It is unlikely that such UE schemes would imply PRE schemes with desirable security properties, unless the definitions for UE are further strengthened to encompass functionalities similar to multi-hop-updates. We elaborate more on this in Section 4.2.

Quantum Attacks on Generic Primitives. In the body of the paper, we show that there exist quantum attacks on a number of generic exact primitives over abelian groups. However, since all of these attacks essentially follow from our attack on an exact homomorphic one-way function (HOWF) over an abelian group, we will focus our attention here on this attack. Informally, an HOWF is a function $f : \mathcal{X} \to \mathcal{Y}$ with input group $(\mathcal{X}, \oplus)$ and output group $(\mathcal{Y}, \otimes)$ (where both group operations are efficiently computable), that meets the definition of a one-way function with the following extra property:

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

Our attack relies on the fact that there exists a quantum algorithm such that given black-box access to an abelian group $\mathcal{G}$ with certain properties, it outputs an *explicit representation* of the group; in other words, it outputs an isomorphism $\psi : \mathcal{G} \to \mathbb{Z}_{q_1} \oplus \cdots \oplus \mathbb{Z}_{q_m}$ such that both ψ and ψ^{-1} are efficiently computable (see [CM01] and Section 6.2 of [Chi17] for more details).

At a high level, our attack works as follows: given an exact HOWF $f : \mathcal{X} \to \mathcal{Y}$ such that $\mathcal{X}$ and $\mathcal{Y}$ are both abelian groups, we use their explicit representations to construct *linear systems of modular equations*, and efficiently solve them to find a preimage for any given HOWF output. The detailed description of the attack is presented in Section 5.

1.4 Open Problems

We believe that our work opens up a substantial number of questions, some of which we mention here.

Security Notions for Truly Symmetric-Key UE and PRE. In this paper, we show that "good" notions of security for UE and PRE necessarily imply public-key encryption. But in practice, people may be unwilling to spend the extra resources necessary for such strong cryptography. What are the best security notions that we can achieve for UE and PRE using truly symmetric-key primitives? Lehmann and Tackmann [LT18] examine double-encryption (their "2Enc" protocol) as an example of a construction from purely symmetric-key primitives. Can we build anything with stronger security using (fast) symmetric primitives?

Constructing KHPRFs and Non-abelian Primitives. The current constructions we have of KHPRFs are quite limited: for instance, we do not know of any exact KHPRF in the standard model. Is it possible to build an (exact) KHPRF from some DDH-like assumption? Our work here suggests that building cryptosystems from inherently non-abelian assumptions would have interesting implications. It is an interesting open problem to build a non-abelian KHwPRF (or even IHwPRF) from an underlying assumption over non-abelian groups.

1.5 Paper Outline

The rest of this paper is organized as follows:

- We describe notations and preliminary background material in Section 2.

- In Section 3, we show the constructions of PKE and (bounded) IHwPRFs from (bounded) KHwPRFs with abelian output groups. We also describe our construction of Naor-Reingold style PRFs from KHwPRFs with abelian output groups. Section 5 shows that any homomorphic one-way function with exact/unbounded homomorphism over abelian groups can be broken using a quantum algorithm.
- In Section 4.1, we show that any ciphertext-independent updatable encryption scheme with forward and post-compromise security implies PKE. In Section 4.2, we show that any symmetric-key proxy re-encryption scheme with reasonably strong security guarantees implies a ciphertext-independent updatable encryption scheme with forward and post-compromise security, and hence PKE.

2 Preliminaries

2.1 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 a finite set S, we use $s \leftarrow S$ to sample uniformly from the set S.

Let $(\mathcal{Y}, \otimes)$ be an efficiently samplable group, such that the group operation is efficiently computable. Let $\mathbf{Y} \in \mathcal{Y}^{m \times \ell}$ be an $m \times \ell$ matrix of group elements sampled from $\mathcal{Y}$. Also, let $\mathbf{s} = (s_1, \ldots, s_\ell) \in \{0,1\}^\ell$ be an arbitrary binary vector. We denote by $\mathbf{Ys} \in \mathcal{Y}^m$ the vector of group elements

$$\left(\bigotimes_{j:s_j=1} y_{1,j}, \ldots, \bigotimes_{j:s_j=1} y_{m,j} \right).$$

Similarly, let $\mathbf{S} = [s_{j,j'}] \in \{0,1\}^{\ell \times \ell'}$ be an arbitrary binary matrix. We denote by $\mathbf{YS} \in \mathcal{Y}^{m \times \ell'}$ the matrix of group elements

$$\begin{bmatrix} \bigotimes_{j:s_{j,1}=1} y_{1,j} & \cdots & \bigotimes_{j:s_{j,\ell'}=1} y_{1,j} \\ \vdots & \ddots & \vdots \\ \bigotimes_{j:s_{j,1}=1} y_{m,j} & \cdots & \bigotimes_{j:s_{j,\ell'}=1} y_{m,j} \end{bmatrix}.$$

2.2 Cryptographic Primitives

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. 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 pseudorandom function (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.

Definition 2.1. (Homomorphic One-Way Function.) A homomorphic one-way function (HOWF) is a function $f : \mathcal{X} \to \mathcal{Y}$ with input group $(\mathcal{X}, \oplus)$ and output group $(\mathcal{Y}, \otimes)$ (where both group operations are efficiently computable), that meets the definition of a one-way function with the following extra property:

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

Definition 2.2. (Key-Homomorphic Functions.) A function family $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is key-homomorphic if the following conditions hold:

- $(\mathcal{K}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are efficiently samplable groups, and the group operations and the inverse operation in each group are efficiently computable.
- For any pair of keys $k_1, k_2 \in \mathcal{K}$ and any input $x \in \mathcal{X}$, we have

$$F(k_1, x) \otimes F(k_2, x) = F(k_1 \oplus k_2, x).$$

A key-homomorphic weak PRF (KHwPRF) family is a weak PRF family that is also key homomorphic. Similarly, a key-homomorphic PRF (KHPRF) family is a PRF family that is also key homomorphic.

Definition 2.3. (Input-Homomorphic Weak PRF.) A weak pseudorandom function family $\{F'(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is an IHwPRF family if the following conditions are satisfied:

- $(\mathcal{X}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are efficiently samplable groups, and the group operations and the inverse operation in each group are efficiently computable.
- For any pair of inputs $x_1, x_2 \in \mathcal{X}$ and any key $k \in \mathcal{K}$, we have

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

Definition 2.4. (γ-Bounded IHwPRF.) A weak pseudorandom function family $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is a γ-bounded IHwPRF family if there is an (efficiently computable) universal mapping $\mathcal{R} : \mathcal{Y} \to \mathcal{Z}$ such that

- $(\mathcal{K}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are efficiently samplable groups, and the group operations and the inverse operation in each group are efficiently computable.
- For a randomly chosen input vector $(x_1, \ldots, x_L) \leftarrow \mathcal{X}^L$ such that $L \le \gamma$, and a randomly chosen key $k \leftarrow \mathcal{K}$, the following holds with overwhelming probability:

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

3 Key-Homomorphic weak PRFs and Implications

In this section we show how to construct PKE and input-homomorphic weak PRF (IHwPRF) from a key-homomorphic weak PRF (KHwPRF). First, we introduce a hardness assumption over the *output group* of a KHwPRF. This hardness assumption has the advantage that it does not directly involve the input set $\mathcal{X}$ of the KHwPRF, which may be algebraically unstructured. Here (and in the following two subsections), we assume that the KHwPRF has unbounded (or exact) homomorphism. Later in this section, we show how to extend our results to "almost" KHwPRFs. Finally, we provide a construction of Naor-Reingold style PRF from KHwPRFs.

Theorem 3.1. *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a KHwPRF, and let $m = \mathrm{poly}(\lambda)$ be an (arbitrary) positive integer. Assume that $d = \mathrm{poly}(\lambda)$ be a positive integer such that $d > 3\log|\mathcal{K}|$. Let $\mathbf{Y} \in \mathcal{Y}^{m\times d}$ be a matrix of group elements such that each entry $y_{i,j}$ (for $i \in [m], j \in [d]$) is drawn uniformly and independently from $\mathcal{Y}$. If $\mathbf{s} \leftarrow \{0,1\}^d$, then for any PPT adversary we have*

$$(\mathbf{Y}, \mathbf{Ys}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{u}),$$

where $\mathbf{u} \leftarrow \mathcal{Y}^m$ is a a vector of m uniformly chosen elements from $\mathcal{Y}$.

Proof. Let $\mathbf{F} \in \mathcal{Y}^{m\times d}$ be a matrix formed in the following way: first sample m uniform elements from $\mathcal{X}$ as $\{x_i \leftarrow \mathcal{X}\}_{i\in[m]}$, and generate d uniform elements from $\mathcal{K}$ as $\{k_j \leftarrow \mathcal{K}\}_{j\in[d]}$. Now we set $\mathbf{F}_{i,j} = F(k_j, x_i)$, i.e., each row (respectively, column) has the same input (respectively, key).

In the first part we prove that $\mathbf{F} \overset{c}{\approx} \mathbf{Y}$. We define the hybrids $\mathcal{H}_j$ over the columns as follows: let $\mathcal{H}_j$ be the hybrid that the first j columns are generated using the weak PRF and the remaining columns are generated using uniform and independent values. By construction, we have $\mathcal{H}_0 \equiv \mathbf{Y}$ and $\mathcal{H}_d \equiv \mathbf{F}$. It is enough to show that $\mathcal{H}_{j-1} \overset{c}{\approx} \mathcal{H}_j$ for each $j \in [d]$. Given access to an oracle $\mathcal{O}$ which is either F or a truly random function, the reduction invokes its oracle m times and receives $\{x_{i'}, \mathcal{O}(x_{i'})\}_{i'\in[m]}$. It then samples $j-1$ keys as $\{k_{j'} \leftarrow \mathcal{K}\}_{j'\in[j-1]}$ and forms the matrix $\mathbf{M} \in \mathcal{Y}^{m\times d}$ as follows:

- If $j' < j$, set $\mathbf{M}_{i',j'} = F(k_{j'}, x_{i'})$.
- If $j' = j$, set $\mathbf{M}_{i',j'} = \mathcal{O}(x_{i'})$.
- If $j' > j$, for each $i' \in [m]$ and $j' \in \{j+1, \ldots, d\}$ sample a fresh $y \leftarrow \mathcal{Y}$ and set $\mathbf{M}_{i',j'} = y$.

Observe that $\mathbf{M} \equiv \mathcal{H}_{j-1}$ if $\mathcal{O}$ corresponds to a truly random function, and $\mathbf{M} \equiv \mathcal{H}_j$ if $\mathcal{O}$ corresponds to the pseudorandom function F. It follows that $\mathcal{H}_{j-1} \overset{c}{\approx} \mathcal{H}_j$.

In the second part of the proof, we show that $(\mathbf{F}, \mathbf{Fs}) \overset{c}{\approx} (\mathbf{F}, \mathbf{u})$. Given an attacker $\mathcal{A}$ that distinguishes $(\mathbf{F}, \mathbf{Fs})$ from $(\mathbf{F}, \mathbf{u})$, we describe an attacker $\mathcal{B}$ against the weak pseudorandomness of F. Given access to an oracle $\mathcal{O}$ which is either F or a truly random function, $\mathcal{B}$ invokes its oracle m times and receives $\{x_i, \mathcal{O}(x_i)\}_{i\in[m]}$. The reduction then samples d keys as $\{k_j \leftarrow \mathcal{K}\}_{j\in[d]}$ and forms the matrix $\mathbf{F}$ as $\mathbf{F}_{i,j} = F(k_j, x_i)$. Define the vectors $\mathbf{y}^* \in \mathcal{Y}^m$ and $\mathbf{k} \in \mathcal{K}^d$ as

$$\mathbf{k} = (k_1, \ldots, k_d), \qquad \mathbf{y}^* := (\mathcal{O}(x_1), \ldots, \mathcal{O}(x_m)) .$$

Finally, $\mathcal{B}$ runs $\mathcal{A}$ on the input $(\mathbf{F}, \mathbf{y}^*)$ and $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. It is easy to see that if $\mathcal{O}$ is a truly random function we have $(\mathbf{F}, \mathbf{y}^*) \equiv (\mathbf{F}, \mathbf{u})$. Observe that by the leftover hash lemma, we have $(\mathbf{k}, \bigoplus_{\mathbf{s}} \mathbf{k}) \overset{s}{\approx} (\mathbf{k}, k^*)$ where k^* is uniform over $\mathcal{K}$. If $\mathbf{y}^*$ corresponds to the weak PRF outputs ($\mathcal{O}$ is the weak PRF), by key homomorphism of F we have

$$\mathbf{Fs} = \begin{pmatrix} F\left(\bigoplus_{\mathbf{s}} \mathbf{k}, x_1\right) \\ F\left(\bigoplus_{\mathbf{s}} \mathbf{k}, x_2\right) \\ \vdots \\ F\left(\bigoplus_{\mathbf{s}} \mathbf{k}, x_m\right) \end{pmatrix} \overset{s}{\approx} \begin{pmatrix} F\left(k^*, x_1\right) \\ F\left(k^*, x_2\right) \\ \vdots \\ F\left(k^*, x_m\right) \end{pmatrix} \equiv \mathbf{y}^*.$$

Therefore, the advantage of $\mathcal{B}$ (in the weak PRF game) is negligibly different from the advantage of $\mathcal{A}$. It follows that $(\mathbf{F}, \mathbf{Fs}) \overset{c}{\approx} (\mathbf{F}, \mathbf{u})$, as required.

Using the first part of the proof by a straightforward reduction we have $(\mathbf{Y}, \mathbf{Ys}) \overset{c}{\approx} (\mathbf{F}, \mathbf{Fs})$ and $(\mathbf{F}, \mathbf{u}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{u})$. Using the second part, it follows that

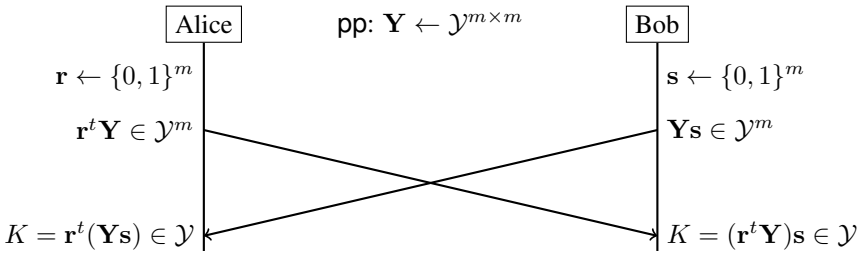
$$(\mathbf{Y}, \mathbf{Ys}) \overset{c}{\approx} (\mathbf{F}, \mathbf{Fs}) \overset{c}{\approx} (\mathbf{F}, \mathbf{u}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{u}),$$

and hence we get $(\mathbf{Y}, \mathbf{Ys}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{u})$. □

3.1 Public-Key Encryption

Now we describe a non-interactive key exchange protocol (which is sufficient to realize PKE) based on any KHwPRF. Later, we explain construction of an IHwPRF from any KHwPRF, which in turn implies a variety of cryptographic primitives. We first start with an inefficient protocol, and then we show how to improve its efficiency.

Given a KHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that $\mathcal{Y}$ is an abelian group, fix some integer $m > 3\log|\mathcal{K}|$ and let $\mathbf{Y} \in \mathcal{Y}^{m\times m}$ be a matrix of uniformly chosen group elements from $\mathcal{Y}$. Alice (respectively, Bob) chooses binary vector $\mathbf{r} \leftarrow \{0,1\}^m$ (respectively, $\mathbf{s} \leftarrow \{0,1\}^m$), and sends $\mathbf{r}^t\mathbf{Y}$ (respectively, $\mathbf{Ys}$) to Bob (respectively, Alice). The final secret will be $\mathbf{r}^t(\mathbf{Ys}) = (\mathbf{r}^t\mathbf{Y})\mathbf{s} \in \mathcal{Y}$. The following figure is a simple visualization of the key exchange protocol.



We sketch the security proof for the mentioned protocol. It is enough to show

$$(\mathbf{Y}, \mathbf{r}^t\mathbf{Y}, \mathbf{Ys}, \mathbf{r}^t\mathbf{Ys}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{y}_1, \mathbf{y}_2, y),$$

where $\mathbf{Y} \leftarrow \mathcal{Y}^{m\times m}, \mathbf{r} \leftarrow \{0,1\}^m, \mathbf{s} \leftarrow \{0,1\}^m, \mathbf{y}_1 \leftarrow \mathcal{Y}^m, \mathbf{y}_2 \leftarrow \mathcal{Y}^m, y \leftarrow \mathcal{Y}$.

Observe that by Theorem 3.1 and a simple hybrid argument we can replace $\mathbf{r}^t\mathbf{Y}$ with a random vector $\mathbf{u} \leftarrow \mathcal{Y}^m$ and so

$$(\mathbf{Y}, \mathbf{u}, \mathbf{Ys}, \mathbf{us}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{y}_1, \mathbf{y}_2, y).$$

Now let $\hat{\mathbf{Y}} \in \mathcal{Y}^{(m+1)\times m}$ be the matrix that has $\mathbf{Y}$ as its top submatrix and $\mathbf{u}$ as its last row. By applying Theorem 3.2 again, it follows that

$$(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}\mathbf{s}) \overset{c}{\approx} (\hat{\mathbf{Y}}, y),$$

as required.

The reader may notice that the aforementioned key exchange protocol is too expensive in terms of communication complexity, i.e, to agree on some group element the parties need to exchange $2m^2$ group elements. Using the following lemma, we immediately get a key exchange protocol for which the whole cost of communication is twice the size of the final secret (like DDH).

Lemma 3.2. *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a KHwPRF, and let $m > 3\log|\mathcal{K}|$ be a positive integer. For any PPT adversary we have*

$$(\mathbf{Y}, \mathbf{RY}, \mathbf{YS}, \mathbf{RYS}) \overset{c}{\approx} (\mathbf{Y}, \mathbf{Y}', \mathbf{Y}'', \mathbf{Y}'''),$$

where $\mathbf{Y}, \mathbf{Y}', \mathbf{Y}'', \mathbf{Y}'''$ are matrices of uniform group elements in $\mathcal{Y}^{m\times m}$, and $\mathbf{S}$, $\mathbf{R}$ are uniform binary matrices, i.e., $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$ and $\mathbf{S} \leftarrow \{0,1\}^{m\times m}$.[1]

Proof. The lemma follows from Theorem 1, and a standard hybrid argument. □

3.2 Input-Homomorphic weak PRF

Here we show a simple construction of an IHwPRF from any KHwPRF. We remark that although an IHwPRF implies a variety of cryptographic primitives, the constructions will not be necessarily efficient. More efficient constructions can be obtained by directly building the primitive using the assumption in Lemma 3.2.

Lemma 3.3. *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a KHwPRF. If $d > 3\log|\mathcal{K}|$ be a positive integer and $\mathcal{Y}$ is an abelian group, the function $\tilde{F} : \{0,1\}^d \times \mathcal{Y}^d \to \mathcal{Y}$ defined as*

$$\tilde{F}\big(\mathbf{s} = (s_1, \ldots, s_d), \mathbf{y} = (y_1, \ldots, y_d)\big) = \bigotimes_{\mathbf{s}} \mathbf{y} = \bigotimes_{j:s_j=1} y_j$$

is an IHwPRF.

[1] Notice that for the correctness of key exchange, we require the group $\mathcal{Y}$ to be abelian.

Proof. First, observe that $\tilde{F}$ is input homomorphic since for any $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}^d$ and $\mathbf{s} \in \{0,1\}^d$ we have

$$\begin{aligned}\tilde{F}(\mathbf{s},\mathbf{y}) \otimes \tilde{F}(\mathbf{s},\mathbf{y}') &= \left(\bigotimes_{\mathbf{s}} \mathbf{y}\right) \otimes \left(\bigotimes_{\mathbf{s}} \mathbf{y}'\right) \\ &= \left(\bigotimes_{j:s_j=1} y_j\right) \otimes \left(\bigotimes_{j:s_j=1} y'_j\right) \\ &= \bigotimes_{j:s_j=1} (y_j \otimes y'_j) \\ &= \bigotimes_{\mathbf{s}} (\mathbf{y} \otimes \mathbf{y}') = \tilde{F}(\mathbf{s}, \mathbf{y} \otimes \mathbf{y}').\end{aligned}$$

Given m (where $m = \mathrm{poly}(\lambda)$) samples of the form $(\mathbf{y}_i, \mathcal{O}(\mathbf{y}_i))$, form the matrix $\mathbf{Y} \in \mathcal{Y}^{m \times d}$ such that the i'th row of $\mathbf{Y}$ is $\mathbf{y}_i$. In addition, define $\mathbf{y}^*$ as $\mathbf{y}^* := (\mathcal{O}(\mathbf{y}_1), \ldots, \mathcal{O}(\mathbf{y}_m))$. Observe that if $\mathcal{O}$ is a truly random function then $\mathbf{y}^*$ is uniformly distributed in $\mathcal{Y}^m$. On the other hand, if $\mathcal{O}$ is the weak PRF, we have $\mathbf{y}^* = \mathbf{Y}\mathbf{s}$ for some uniform $\mathbf{s} \in \{0,1\}^d$. By applying Theorem 3.1 and observing the fact that $m = \mathrm{poly}(\lambda)$, it follows that F is a weak PRF. □

Implications. By plugging in the results of [AMPR19], and using the Lemma 3.3 it follows that KHwPRFs imply noninteractive key exchange, private information retrieval [KO97], lossy trapdoor functions [PW08], identity-based encryption (in a non-blackbox manner) [DG17b, DG17a, BLSV18], and hinting PRGs [KW19].

We remark that KHwPRFs trivially imply homomorphic one-way functions (HOWFs) and hence using the results of [AMPR19], KHwPRFs imply collision-resistant hash functions, Schnorr signatures, and chameleon hash functions.

3.3 Asymmetric Primitives from Bounded KHwPRFs

In this part, we show that the "approximate" (some papers called it "almost") version of key-homomorphic weak PRFs with certain properties imply a variety of asymmetric primitives, such as public-key encryption (PKE). Approximate KHwPRFs have the property that $F_{k \oplus k'}(x)$ is *close* to $F_k(x) \otimes F_{k'}(x)$ where closeness is measured with respect to some distance function.

An Algebraic Definition. Formalizing a general definition for "approximate" homomorphism requires a somewhat involved *geometric* definition that needs a distance function, which also does not nicely fit into the recent (algebraic) framework of [AMPR19]. In this work, we provide a natural *algebraic* definition for *bounded* key-homomorphic weak PRFs, which is similar to the definition of bounded IHwPRFs of [AMPR19].

We remark that all existing constructions of approximate KHwPRFs with an appropriate choice of parameters can be viewed as bounded KHwPRFs.

Definition 3.4. A weak pseudorandom function family $\{F(k,\cdot) : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$ is a γ-bounded KHwPRF family if there exists (efficiently computable) universal mappings $\mathcal{R}_{\mathsf{in}} : \mathcal{Y} \to \mathcal{Z}_{\mathsf{in}}$ and $\mathcal{R}_{\mathsf{out}} : \mathcal{Z}_{\mathsf{in}} \to \mathcal{Z}_{\mathsf{out}}$ such that

- $(\mathcal{K}, \oplus)$, $(\mathcal{Y}, \otimes)$, and $(\mathcal{Z}_{\mathsf{in}}, \odot)$ are efficiently samplable groups, and the group operations and the inverse operation in each group are efficiently computable.
- For a randomly chosen key vector $(k_1, \ldots, k_L) \leftarrow \mathcal{K}^L$ such that $L \leq \gamma$, and a randomly chosen input $x \leftarrow \mathcal{X}$, the following holds with overwhelming probability:

$$\mathcal{R}_{\mathsf{in}}\Big(F\Big(\bigoplus_{j \in [L]} k_j, x\Big)\Big) = \mathcal{R}_{\mathsf{in}}\Big(\bigotimes_{j \in [L]} F(k_j, x)\Big),$$

$$\mathcal{R}_{\mathsf{out}}\Big(\bigodot_{j \in [L]} \mathcal{R}_{\mathsf{in}}\Big(F(k_j, x)\Big)\Big) = \mathcal{R}_{\mathsf{out}}\Big(\mathcal{R}_{\mathsf{in}}\Big(\bigotimes_{j \in [L]} F(k_j, x)\Big)\Big).$$

Bounded KHwPRFs and LWR. All of the currently known instantiations of "approximate" key-homomorphic (weak) PRFs use Learning With Rounding (LWR) [BPR12] as their underlying assumption. It is easy to see that if the *output* group of some LWR-based KHwPRF is $\mathbb{Z}_p^n$ for some *superpolynomial* modulus p and some dimension n, we can define the mapping $\mathcal{R}_{\mathsf{in}}$ (respectively, $\mathcal{R}_{\mathsf{out}}$) to be rounding with respect to some modulus p_{in} (respectively, p_{out}) such that p/p_{in} and $p_{\mathsf{in}}/p_{\mathsf{out}}$ are both superpolynomial.

This immediately yields bounded KHwPRFs from approximate KHwPRFs that have the mentioned property. We remark that this property seems to be necessary for most of the applications of KH-PRFs in [BLMR13] (and in some cases to get an efficient construction). The reader may note that the resulting construction of bounded KHwPRFs from LWR has a triple rounding, one that is embedded in the (weak) PRF F and one for each mapping $\mathcal{R}_{\mathsf{out}}$ and $\mathcal{R}_{\mathsf{in}}$ defined above. Although this property is inherent for the LWR-based construction, in general there may not be any similarity between F and $\mathcal{R}_{\mathsf{in}}$ or $\mathcal{R}_{\mathsf{out}}$ for a bounded KHwPRF.

PKE Construction from Bounded KHwPRF. Using the definition above, we now construct a public-key encryption scheme from a *bounded* KHwPRF. The construction is almost identical to the case of unbounded KHwPRFs, with the difference being applying the mappings $\mathcal{R}_{\mathsf{in}}$ and $\mathcal{R}_{\mathsf{out}}$ of the bounded KHwPRF. The argument for the security is also very similar to the exact/unbounded case, and we omit the details.

Given a γ-bounded KHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ (with mappings $\mathcal{R}_{\mathsf{in}}$ and $\mathcal{R}_{\mathsf{out}}$ defined as above) such that $\mathcal{Y}$ and $\mathcal{Z}_{\mathsf{in}}$ are abelian groups and $\gamma > 3\log|\mathcal{K}|$, fix some integer $m > 3\log|\mathcal{K}|$ and let $\mathbf{Y} \in \mathcal{Y}^{m\times m}$ be a matrix of uniformly chosen group elements from $\mathcal{Y}$. Alice (respectively, Bob) chooses binary vector $\mathbf{r} \leftarrow \{0,1\}^m$ (respectively, $\mathbf{s} \leftarrow \{0,1\}^m$), and sends $\mathcal{R}_{\mathsf{in}}(\mathbf{r}^t\mathbf{Y})$ (respectively, $\mathcal{R}_{\mathsf{in}}(\mathbf{Y}\mathbf{s})$) to Bob (respectively, Alice). The final secret will be

$$\mathcal{R}_{\mathsf{out}}\big(\mathbf{r}^t\mathcal{R}_{\mathsf{in}}(\mathbf{Y}\mathbf{s})\big) = \mathcal{R}_{\mathsf{out}}\big(\mathcal{R}_{\mathsf{in}}(\mathbf{r}^t\mathbf{Y})\mathbf{s}\big) \in \mathcal{Z}_{\mathsf{out}}.$$

Bounded IHwPRF from Bounded KHwPRF. Using the definition above, we now construct a bounded IHwPRF from a *bounded* KHwPRF. The construction is almost identical to the case of unbounded KHwPRFs, with the difference being applying the mappings $\mathcal{R}_{\mathsf{in}}$ and $\mathcal{R}_{\mathsf{out}}$ of the bounded KHwPRF.

Given a γ-bounded KHwPRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ (with mappings $\mathcal{R}_{\mathsf{in}}$ and $\mathcal{R}_{\mathsf{out}}$ defined as above) such that $\mathcal{Y}$ and $\mathcal{Z}_{\mathsf{in}}$ are abelian groups and $\gamma > 3\log|\mathcal{K}|$, fix some integer d such that $3\log|\mathcal{K}| < d \le \gamma$ we define a bounded IHwPRF $\tilde{F} : \{0,1\}^d \times \mathcal{Y}^d \to \mathcal{Z}_{\mathsf{in}}$ with its associated mapping $\tilde{\mathcal{R}} : \mathcal{Z}_{\mathsf{in}} \to \mathcal{Z}_{\mathsf{out}}$ as

$$\tilde{F}\big(\mathbf{s} = (s_1,\ldots,s_d), \mathbf{y} = (y_1,\ldots,y_d)\big) = \mathcal{R}_{\mathsf{in}}\big(\bigotimes_{\mathbf{s}} \mathbf{y}\big) = \mathcal{R}_{\mathsf{in}}\big(\bigotimes_{j:s_j=1} y_j\big),$$

where $\tilde{\mathcal{R}}$ (the associated mapping with $\tilde{F}$) is identical to $\mathcal{R}_{\mathsf{out}}$. The security proof is very similar to the exact/unbounded case, and hence we omit the details.

3.4 Naor-Reingold PRF

Here we show a construction of Naor-Reingold style PRF from any KHwPRF. Before we do so, however, we will provide some background on the Naor-Reingold PRF and explain why PRFs in this style are important. We start by recalling the original Naor-Reingold PRF [NR97]:

Let $\mathbb{G}$ be a group of order p, and let $F_{NR} : (\mathbb{Z}_p^{(\ell+1)} \times \mathbb{G}) \times \{0,1\}^\ell \to \mathbb{G}$ be the function defined by

$$F_{NR}\left(\{\alpha_j\}_{j\in[0,\ell]} \in \mathbb{Z}_p^{(\ell+1)}, g \in \mathbb{G}, \mathbf{x} \in \{0,1\}^\ell\right) = g^{\alpha_0 \prod_{i=1}^{\ell} \alpha_i^{x_i}},$$

where the values $\alpha_0, \alpha_1, \ldots, \alpha_\ell$ form the key and $\mathbf{x}$ is the input. Informally, a Naor-Reingold style PRF requires a constant number of computations (for instance, the exponentiation in F_{NR}) on which the assumption related to its hardness depends, while all of the operations that scale with the length of the input (for instance, the integer multiplications in the exponent of F_{NR}) are less expensive. This feature allows Naor-Reingold style PRFs to be potentially efficient. In particular, assuming that the underlying operations have reasonably low circuit depth, such PRFs typically have polylogarithmic evaluation circuits.

We now show a simple construction of Naor-Reingold style PRF from any exact KHwPRF with abelian output group. Our construction involves a subset product of binary matrices and one "multiplication" of a group matrix and an integer matrix. The depth of the PRF evaluation circuit is polylogarithmic provided that the group operation can be done efficiently.

Theorem 3.5. *Let $\tilde{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a KHwPRF, and fix some $m > 3\log|\mathcal{K}|$. Let $\mathbf{Y} \in \mathcal{Y}^{m\times m}$ be a (public) matrix of group elements such that each entry $y_{i,j}$ (for $i \in [m], j \in [m]$) is drawn uniformly and independently from $\mathcal{Y}$. The function $F : \mathcal{Y}^{(\ell+1)\times m^2} \times \{0,1\}^\ell \to \mathcal{Y}^{m\times m}$ defined as*

$$F\Big((\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell), \mathbf{x} = (x_1, \ldots, x_\ell)\Big) = \mathbf{Y}\mathbf{S}_0 \prod_{i=1}^{\ell} \mathbf{S}_i^{x_i}$$

is a pseudorandom function where $\mathbf{S}_i \leftarrow \{0,1\}^{m\times m}$ for $i \in \{0, \ldots, \ell\}$.

Proof. To prove this theorem, we use the following lemma.

Lemma 3.6. *Let $\mathbf{Y}_1, \ldots, \mathbf{Y}_Q \in \mathcal{Y}^{m\times m}$ be matrices with uniformly and independently sampled entries from $\mathcal{Y}$ for some $Q = \mathrm{poly}(\lambda)$, and let $\mathbf{S} \leftarrow \{0,1\}^{m\times m}$ be a uniformly sampled binary matrix. Then for any PPT adversary we have*

$$\{(\mathbf{Y}_q, \mathbf{Y}_q\mathbf{S})\}_{q\in[Q]} \overset{c}{\approx} \{(\mathbf{Y}_q, \mathbf{U}_q)\}_{q\in[Q]}.$$

where for each $q \in [Q]$, $\mathbf{U}_q \leftarrow \mathcal{Y}^{m\times m}$ is a matrix of uniformly chosen elements from $\mathcal{Y}$.

This lemma follows directly from Theorem 1, and a standard hybrid argument over the columns of $\mathbf{S}$. The proof of pseudorandomness now proceeds via a series of $(\ell+1)$ hybrid games, where for each $j \in [0, \ell]$, the j^{th} game is as described below.

1. The challenger samples $(\ell - j)$ uniform binary matrices as $\mathbf{S}_i \leftarrow \{0,1\}^{m\times m}$ for $i \in [j+1, \ell]$. It also maintains a list $\mathcal{L}$ of $m \times m$ matrices over the group $\mathcal{Y}$. Initially, this list is empty. The challenger also creates and stores an $m \times m$ matrix $\mathbf{Y}_0$ consisting of uniformly and independently sampled entries from $\mathcal{Y}$.

2. The adversary adaptively issues a maximum of $Q = \mathrm{poly}(\lambda)$ PRF queries of the form $\mathbf{x}_1, \ldots, \mathbf{x}_Q$, where for each $q \in [Q]$, we have $\mathbf{x}_q = (x_{1,q}, \ldots, x_{\ell,q})$. For ease of representation, we divide each query string as $\mathbf{x}_q = (\mathbf{x}_q^{(0)}, \mathbf{x}_q^{(1)})$, where

$$\mathbf{x}_q^{(0)} = (x_{1,q}, \ldots, x_{j,q}), \quad \mathbf{x}_q^{(1)} = (x_{j+1,q}, \ldots, x_{\ell,q}).$$

3. Upon receipt of the q^{th} query, the challenger proceeds as follows:

   (a) If $j = 0$, it sets $\mathbf{Y}_q = \mathbf{Y}_0$.

   (b) Otherwise, it checks if there exists a $q' < q$ such that $\mathbf{x}_q^{(0)} = \mathbf{x}_{q'}^{(0)}$.

      i. If yes, it sets $\mathbf{Y}_q = \mathbf{Y}_{q'}$.

      ii. Otherwise, it sets $\mathbf{Y}_q$ to be an $m \times m$ matrix with uniformly and independently sampled entries from $\mathcal{Y}$.

   (c) It updates the list $\mathcal{L}$ as $\mathcal{L} = \mathcal{L} \cup \{\mathbf{Y}_q\}$ and responds to the q^{th} query as

$$f_{j,q} = \mathbf{Y}_q \prod_{i=j+1}^{\ell} \mathbf{S}_i^{x_{i,q}}.$$

Note that in the zero$^{\text{th}}$ hybrid, we replaced the component $\mathbf{Y}\mathbf{S}_0$ in the original PRF construction by an $m \times m$ matrix $\mathbf{Y}_0$ consisting of uniformly and independently sampled entries from $\mathcal{Y}$. It follows from Theorem 1 that this hybrid is indistinguishable from the real PRF experiment.

Now, for each $j \in [0, \ell]$, let $\mathcal{F}_j = \{f_{j,q}\}_{q\in[Q]}$ be the set of responses generated by the challenger in the j^{th} game. The proof of Theorem 3.5 now follows immediately from the following claim:

Claim 3.7. *For each $j \in [0, \ell-1]$ and for any PPT adversary we have*

$$\mathcal{F}_j \overset{c}{\approx} \mathcal{F}_{j+1}.$$

Let $\mathcal{A}$ be a PPT adversary such that for some $j \in [\ell]$, $\mathcal{A}$ efficiently distinguishes between $\mathcal{F}_j$ and $\mathcal{F}_{j+1}$. We construct an attacker $\mathcal{B}$ against the assumption in Lemma 3.6. $\mathcal{B}$ receives as input a tuple of the form $\{(\mathbf{Y}_q, \mathbf{Z}_q)\}_{q\in[Q']}$ for some $Q' > Q$, where either each $\mathbf{Z}_q$ is of the form $\mathbf{Y}_q\mathbf{S}_j$ for some uniformly random $m \times m$ binary matrix $\mathbf{S}_j$, or each $\mathbf{Z}_q$ is a uniformly random matrix over $\mathcal{Y}^{m\times m}$. It proceeds as follows:

1. $\mathcal{B}$ samples $(\ell - j - 1)$ uniform binary matrices as $\mathbf{S}_i \leftarrow \{0,1\}^{m\times m}$ for $i \in [j+2, \ell]$. It also maintains a counter variable cnt. Initially, $\mathsf{cnt} = 1$.

2. $\mathcal{A}$ adaptively issues a maximum of $Q = \mathrm{poly}(\lambda)$ PRF queries of the form $\mathbf{x}_1, \ldots, \mathbf{x}_Q$, where for each $q \in [Q]$, we have $\mathbf{x}_q = (x_{1,q}, \ldots, x_{\ell,q})$. Again, for ease of representation, we divide each query string as $\mathbf{x}_q = (\mathbf{x}_q^{(0)}, \mathbf{x}_q^{(1)})$, where
$$\mathbf{x}_q^{(0)} = (x_{1,q}, \ldots, x_{j,q}), \quad \mathbf{x}_q^{(1)} = (x_{j+1,q}, \ldots, x_{\ell,q}).$$

3. Upon receipt of the q^{th} query, $\mathcal{B}$ checks if there exists a $q' < q$ such that $\mathbf{x}_q^{(0)} = \mathbf{x}_{q'}^{(0)}$.
   (a) If yes, it sets $\widetilde{\mathbf{Y}}_q = \widetilde{\mathbf{Y}}_{q'}$ and $\widetilde{\mathbf{Z}}_q = \widetilde{\mathbf{Z}}_{q'}$.
   (b) Otherwise, it sets $\widetilde{\mathbf{Y}}_q = \mathbf{Y}_{\mathsf{cnt}}$ and $\widetilde{\mathbf{Z}}_q = \mathbf{Z}_{\mathsf{cnt}}$, and updates $\mathsf{cnt} = \mathsf{cnt} + 1$.

4. $\mathcal{B}$ now responds to the q^{th} query as
$$\tilde{f}_{j,q} = \begin{cases} \widetilde{\mathbf{Y}}_q \prod_{i=j+2}^{\ell} \mathbf{S}_i^{x_{i,q}} & \text{if } x_{j+1,q} = 0 \\ \widetilde{\mathbf{Z}}_q \prod_{i=j+2}^{\ell} \mathbf{S}_i^{x_{i,q}} & \text{if } x_{j+1,q} = 1. \end{cases}$$

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

Let $\widetilde{\mathcal{F}} = \{\tilde{f}_{j,q}\}_{q\in[Q]}$ be the set of responses generated by $\mathcal{B}$. It is easy to see the following:

- If each $\mathbf{Z}_q$ is of the form $\mathbf{Y}_q\mathbf{S}_j$ for some uniformly random $m \times m$ binary matrix $\mathbf{S}_j$, then the distribution of $\widetilde{\mathcal{F}}$ is identical to that of $\mathcal{F}_j$.
- On the other hand, if each $\mathbf{Z}_q$ is a uniformly random matrix over $\mathcal{Y}^{m\times m}$, then the distribution of $\widetilde{\mathcal{F}}$ is identical to that of $\mathcal{F}_{j+1}$.

It now follows that the advantage of $\mathcal{B}$ is identical to that of $\mathcal{A}$. This completes the proof of Claim 3.7. The proof of Theorem 3.5 follows immediately. □

NR-style PRFs from Bounded KHwPRFs. Our definition of bounded KHwPRFs does not allow a direct construction of NR-style PRFs. However, there are known constructions of NR-style PRFs from lattice-based assumptions. A notable example is the lattice-based KHPRF from [BLMR13], which proves security by progressively rounding further at each hybrid argument to ensure that "exactness" holds at each step. However, while actually computing the PRF, this is simulated by rounding once to a specially chosen modulus. In practical scenarios, this construction seems substantially less efficient than related pseudorandom synthesizer constructions [NR95, Mon18].

Our algebraic definition of bounded KHwPRFs does not encompass multiple levels of "rounding" (or any other compressing operation), since it seemingly makes bounded KHwPRFs inherently inefficient for constructing NR-style PRFs. Thus, we omit constructing NR-style PRFs from bounded KHwPRFs.

4 Updatable Encryption and Symmetric PRE

In this section, we show that any ciphertext-independent updatable encryption scheme that satisfies the adaptive notions of forward and post-compromise security proposed in [LT18] implies PKE. We also show that any symmetric-key proxy re-encryption scheme that satisfies the adaptive notion of indistinguishability-based security formalized in [FKKP19] implies updatable encryption with forward and post-compromise security, and hence PKE.

4.1 PKE from Updatable Encryption

An updatable encryption scheme is, informally speaking, a symmetric key encryption scheme with the following extra property: a user with a secret key k_1 can provide an update token $\sigma_{1,2}$ that maps ciphertexts encrypted under key k_1 to new ciphertexts encrypted under some other key k_2. The main application of updatable encryption is handling key rotation of data in the cloud where a data owner does not trust the cloud owner enough to provide them with a secret key in the clear.

Updatable encryption was first defined in [BLMR13] as an application of KHPRFs. The definitions proposed in [BLMR13] were subsequently refined by Everspaugh *et al.* in [EPRS17]. In a more recent work, Lehmann and Tackmann [LT18] introduced more rigorous security notions for updatable encryption that are desirable for real-world applications, and also pointed out that none of the existing constructions satisfy these notions. In this section, we show that updatable encryption with adaptive update indistinguishability (IND-UPD) as defined by Lehmann and Tackmann [LT18] implies public-key encryption. We start by defining the general functionality of any UE scheme. Note that all of the definitions we use here are from [LT18].

Definition 4.1. (Updatable Encryption). An updatable encryption scheme UE for a message space $\mathcal{M}$ is a tuple of five PPT algorithms $(\mathsf{Setup}, \mathsf{Next}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Update})$ defined as follows:

- $\mathsf{Setup}(1^\lambda)$: Given the security parameter λ, it generates a secret key sk_0.
- $\mathsf{Next}(\mathsf{sk}_e)$: On input a secret key sk_e for epoch e, it generates a new secret key sk_{e+1} and a new update token $\Delta_{e,e+1}$ for epoch $(e+1)$.
- $\mathsf{Enc}(\mathsf{sk}_e, \mathsf{m})$: On input a secret key sk_e for epoch e and a message $\mathsf{m} \in \mathcal{M}$, it generates a ciphertext ct_e.
- $\mathsf{Dec}(\mathsf{sk}_e, \mathsf{ct}_e)$: On input a secret key sk_e and a ciphertext ct_e for some epoch e, it either outputs a message $\mathsf{m}' \in \mathcal{M}$ or $\perp$.
- $\mathsf{Update}(\Delta_{e,e+1}, \mathsf{ct}_e)$: On input an update token $\Delta_{e,e+1}$ and a ciphertext ct_e for some epoch e, it outputs an updated ciphertext ct_{e+1} for epoch $(e+1)$.

Correctness. For any message $\mathsf{m} \in \mathcal{M}$, for any $\mathsf{sk}_0 \leftarrow \mathsf{Setup}(1^\lambda)$, and for any sequence of key/update token pairs $(\mathsf{sk}_1, \Delta_{0,1}), \ldots, (\mathsf{sk}_e, \Delta_{e-1,e})$ obtained recursively as $(\mathsf{sk}_j, \Delta_{j-1,j}) \leftarrow \mathsf{Next}(\mathsf{sk}_{j-1})$ for each $j \in [e]$, we have

$$\mathsf{Dec}(\mathsf{sk}_j, \mathsf{ct}_j) = \mathsf{m},$$

for any $j \in [e]$, where the sequence of ciphertexts $\mathsf{ct}_0, \mathsf{ct}_1, \ldots, \mathsf{ct}_e$ is obtained as $\mathsf{ct}_0 = \mathsf{Enc}(\mathsf{sk}_0, \mathsf{m})$ and $\mathsf{ct}_j \leftarrow \mathsf{Update}(\Delta_{j-1,j}, \mathsf{ct}_{j-1})$ for each $j \in [e]$.

Security Notions for UE. In their paper [LT18], Lehmann and Tackmann define several notions of security for updatable encryption. Previous works had somewhat non-accurate notions of security, so we consider the definitions from [LT18] to be the only suitable ones currently known for UE. In this section, we focus on their IND-UPD security definition, which we explain below. In our opinion, this definition reflects the security needs of a user storing data and updating ciphertexts in an untrusted cloud. However, debating the definitions of UE is out of scope of this paper, and we refer to the sections 3 and 4 of [LT18] for a discussion of notions of UE security.

Forward and Post-Compromise Security. We adopt the definition of post-compromise security for UE schemes proposed and formalized by Lehmann and Tackmann in a recent work [LT18]. More specifically, we focus on the notion

of *adaptive update indistinguishability*, or IND-UPD in short (also referred to as *unlinkability*), which ensures that an updated ciphertext obtained via the Update algorithm does not reveal any information about the previous ciphertext to a PPT adversary $\mathcal{A}$, even when $\mathcal{A}$ adaptively compromises polynomially many keys and tokens before and after the challenge epoch.

Adaptive Update Indistinguishability. We recall the formal definitions for adaptive update indistinguishability from [LT18]. We assume that the adversary has access to the following oracles (e is an epoch counter initialized to 0 and $\mathcal{L}$ is a list initialized to empty):

1. $\mathcal{O}_{\mathsf{Enc}}$: On input a message m, this oracle outputs $\mathsf{ct}_e \leftarrow \mathsf{Enc}(\mathsf{sk}_e, \mathsf{m})$, where sk_e is the secret key corresponding to the current epoch e, and adds the tuple (ct_e, e) to the list $\mathcal{L}$.
2. $\mathcal{O}_{\mathsf{Next}}$: When queried, this oracle generates a new key/update token pair as $(\mathsf{sk}_{e+1}, \Delta_{e,e+1}) \leftarrow \mathsf{Next}(\mathsf{sk}_e)$, updates the epoch counter to $(e+1)$ and adds $(e+1, \mathsf{sk}_{e+1}, \Delta_{e,e+1})$ to the global state of the challenger. If issued post challenge-query phase, it also updates the challenge ciphertext to the new epoch as $\mathsf{ct}^*_e \leftarrow \mathsf{Update}(\Delta_{e-1,e}, \mathsf{ct}^*_{e-1})$, and adds (ct^*_e, e) to the list $\mathcal{L}$.
3. $\mathcal{O}_{\mathsf{Update}}$: On input a ciphertext ct_{e-1} such that $(\mathsf{ct}_{e-1}, e-1) \in \mathcal{L}$ (i.e., the input ciphertext is honestly generated during the previous epoch), this oracle outputs $\mathsf{ct}_e \leftarrow \mathsf{Update}(\Delta_{e-1,e}, \mathsf{ct}_{e-1})$, and adds (ct_e, e) to the list $\mathcal{L}$.
4. $\mathcal{O}_{\mathsf{corrupt}}$: This oracle takes as input an epoch $e' \leq e$ (where e is the current epoch) and either key or token. On input (e', key), it outputs the key $\mathsf{sk}_{e'}$. On input (e', token), it outputs the token $\Delta_{e'-1,e'}$.
5. $\mathcal{O}_{\mathsf{challenge}}$: This oracle returns the current challenge ciphertext ct^*_e from the list $\mathcal{L}$.

For each bit $b \in \{0,1\}$, define the following experiment $\mathsf{Expt}^{\text{ind-upd}}_b$ between a challenger and an adversary $\mathcal{A}$:

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

1. The challenger generates $\mathsf{sk}_0 \leftarrow \mathsf{Setup}(1^\lambda)$.
2. The challenger maintains an epoch counter e, a challenge epoch counter e^* and a list $\mathcal{L}$. Initially, $e = 0$, $e^* = \perp$ and $\mathcal{L} = \phi$.
3. The adversary $\mathcal{A}$ adaptively issues any number of queries to the $\mathcal{O}_{\mathsf{Enc}}$, $\mathcal{O}_{\mathsf{Next}}$, $\mathcal{O}_{\mathsf{Update}}$ and $\mathcal{O}_{\mathsf{corrupt}}$ oracles. These oracles update the epoch counter e and the list $\mathcal{L}$ as described above.
4. The adversary $\mathcal{A}$ eventually outputs a pair of ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$, subject to the restriction that $(\mathsf{ct}_0, e-1), (\mathsf{ct}_1, e-1) \in \mathcal{L}$ and $|\mathsf{ct}_0| = |\mathsf{ct}_1|$.
5. The challenger queries the $\mathcal{O}_{\mathsf{Next}}$ oracle to obtain the key/update token pair $(\mathsf{sk}_e, \Delta_{e-1,e})$.
6. The challenger sets $\mathsf{ct}^*_e \leftarrow \mathsf{Update}(\Delta_{e-1,e}, \mathsf{ct}_b)$ and adds the tuple (ct^*_e, e) to the list $\mathcal{L}$. It also sets $e^* = e$.
7. The adversary $\mathcal{A}$ continues to adaptively issue any number of queries to the $\mathcal{O}_{\mathsf{Enc}}$, $\mathcal{O}_{\mathsf{Next}}$, $\mathcal{O}_{\mathsf{Update}}$, $\mathcal{O}_{\mathsf{corrupt}}$ and $\mathcal{O}_{\mathsf{challenge}}$ oracles, albeit subject to the following restrictions:
   (a) $\mathcal{A}$ has not made an *update-query* to the $\mathcal{O}_{\mathsf{corrupt}}$ oracle during the challenge epoch e^*, that is, it does not know the update token Δ_{e^*-1,e^*}.
   (b) If $\mathcal{E}^*_0$ is the set of all epochs during which $\mathcal{A}$ has queried the $\mathcal{O}_{\mathsf{challenge}}$ oracle and $\mathcal{E}^*_1$ is the set of all epochs during which $\mathcal{A}$ has made *key-queries* to the $\mathcal{O}_{\mathsf{corrupt}}$ oracle, then $\mathcal{E}^*_0 \cap \mathcal{E}^*_1 = \{\}$.

Definition 4.2. (IND-UPD secure Updatable Encryption). An updatable encryption scheme $(\mathsf{Setup}, \mathsf{Next}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Update})$ is said to be IND-UPD-secure if for all PPT adversaries $\mathcal{A}$, the views of $\mathcal{A}$ in the experiments $\mathsf{Expt}_0^{\text{ind-upd}}$ and $\mathsf{Expt}_1^{\text{ind-upd}}$ are computationally indistinguishable. (Note that in the aforementioned definition, we implicitly assumed that the update algorithm of the underlying UE scheme is randomized.)

We now show that any updatable encryption scheme that satisfies adaptive update indistinguishability implies a PKE scheme. More formally, let $\mathsf{UE} = (\mathsf{Setup}, \mathsf{Next}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Update})$ be an IND-UPD secure scheme. We construct a PKE scheme as follows.

- Key Generation: The key generation algorithm receives as input the security parameter λ. It first generates a secret key for the UE scheme as $\mathsf{sk}_0 \leftarrow \mathsf{Setup}(1^\lambda)$. It then recursively updates this secret key $(e+1)$ times for some arbitrarily chosen epoch e, as

  $$(\mathsf{sk}_j, \Delta_{j-1,j}) \leftarrow \mathsf{Next}(\mathsf{sk}_{j-1}) \text{ for each } j \in [e+1].$$

  Finally, it chooses two messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ such that $\mathsf{m}_0 \neq \mathsf{m}_1$, sets

  $$\mathsf{ct}_0^* = \mathsf{Enc}(\mathsf{sk}_e, \mathsf{m}_0), \quad \mathsf{ct}_1^* = \mathsf{Enc}(\mathsf{sk}_e, \mathsf{m}_1),$$

  and outputs the secret key/public key pair $(\mathsf{sk}_{\text{PKE}}, \mathsf{pk}_{\text{PKE}})$ as

  $$\mathsf{sk}_{\text{PKE}} = \mathsf{sk}_{e+1}, \quad \mathsf{pk}_{\text{PKE}} = ((\mathsf{m}_0, \mathsf{ct}_0^*), (\mathsf{m}_1, \mathsf{ct}_1^*), \Delta_{e,e+1}) .$$

- Encryption: To encrypt a bit $b \in \{0,1\}$, the encryption algorithm outputs a randomized update of ct_b^* to the epoch $e+1$ using the publicly available update token $\Delta_{e,e+1}$. More formally, on input a bit $b \in \{0,1\}$, the encryption algorithm outputs

  $$\mathsf{ct}_{\text{PKE}} \leftarrow \mathsf{Update}(\Delta_{e,e+1}, \mathsf{ct}_b^*).$$

- Decryption: On input a ciphertext ct_{PKE}, the decryption algorithm computes

  $$\mathsf{m}' = \mathsf{Dec}(\mathsf{sk}_{\text{PKE}}, \mathsf{ct}_{\text{PKE}}).$$

  If $\mathsf{m}' = \mathsf{m}_b$ for some $b \in \{0,1\}$, it outputs b. Otherwise, it outputs $\perp$.

Correctness is straightforward to verify. We now formally prove the following theorem.

Theorem 4.3. *If* UE *is* IND-UPD *secure, then the aforementioned* PKE *scheme is* IND-CPA *secure.*

Proof. Let $\mathcal{A}$ be an adversary that breaks the IND-CPA security of the PKE scheme with non-negligible advantage ε. We construct an algorithm $\mathcal{B}$ that breaks the IND-UPD security of UE with advantage $\varepsilon' = \varepsilon/2$. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ plays the IND-UPD security game with the challenger for UE till some epoch $e-1$ for some arbitrarily chosen challenge epoch e. At this point, $\mathcal{B}$ chooses a pair of arbitrary messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ such that $\mathsf{m}_0 \neq \mathsf{m}_1$, and queries the $\mathcal{O}_{\mathsf{Enc}}$ oracle to obtain $(\mathsf{ct}_0, \mathsf{ct}_1)$, where

   $$\mathsf{ct}_0 = \mathsf{Enc}(\mathsf{sk}_{e-1}, \mathsf{m}_0), \quad \mathsf{ct}_1 = \mathsf{Enc}(\mathsf{sk}_{e-1}, \mathsf{m}_1).$$

2. $\mathcal{B}$ outputs $(\mathsf{ct}_0, \mathsf{ct}_1)$ as the pair of challenge ciphertexts, and receives the challenge ciphertext ct_e^*, which is a randomized update of ct_b to the epoch e for $b \leftarrow \{0,1\}$.

3. Next, $\mathcal{B}$ issues the following additional queries:

   (a) It queries the $\mathcal{O}_{\mathsf{Update}}$ oracle to receive $\widetilde{\mathsf{ct}}_e^*$, which is a randomized update of ct_1 to the epoch e.

(b) It issues a query to the $\mathcal{O}_{\mathsf{Next}}$ oracle, followed by a *token-query* to the $\mathcal{O}_{\mathsf{corrupt}}$ oracle, to obtain an update token $\Delta_{e,e+1}$.

Note that none of the aforementioned queries violate any of the constraints described in the IND-UPD security experiment.

4. $\mathcal{B}$ now provides the adversary $\mathcal{A}$ with the public key $\mathsf{pk}_{\mathrm{PKE}}$, where

$$\mathsf{pk}_{\mathrm{PKE}} = \left((\mathsf{m}_0, \mathsf{ct}_e^*), (\mathsf{m}_1, \widetilde{\mathsf{ct}}_e^*), \Delta_{e,e+1}\right).$$

5. $\mathcal{B}$ uniformly samples $b' \leftarrow \{0,1\}$ and outputs the challenge ciphertext $\mathsf{ct}_{\mathrm{PKE}}^*$, where

$$\mathsf{ct}_{\mathrm{PKE}}^* = \begin{cases} \mathsf{Update}(\Delta_{e,e+1}, \mathsf{ct}_e^*) & \text{if } b' = 0, \\ \mathsf{Update}(\Delta_{e,e+1}, \widetilde{\mathsf{ct}}_e^*) & \text{if } b' = 1. \end{cases}$$

6. $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

Observe that when $b = 0$, the distribution of the public key $\mathsf{pk}_{\mathrm{PKE}}$ in the view of $\mathcal{A}$ is exactly as in the real IND-CPA experiment. On the other hand, if $b = 1$, then ct_e^* and $\widetilde{\mathsf{ct}}_e^*$ are sampled from the same distribution. It follows that the advantage of $\mathcal{B}$ in the IND-UPD experiment is $\varepsilon' = \varepsilon/2$. This completes the proof of Theorem 4.3. □

4.2 PKE from Symmetric PRE

A symmetric-key proxy re-encryption (SK-PRE) scheme is a symmetric-key encryption scheme that allows the holder of a key sk_0 to derive a re-encryption token $\Delta_{0,1}$ for any other key sk_1. This re-encryption token lets anyone transform ciphertexts encrypted under sk_0 into ciphertexts encrypted under sk_1, *without having to know the underlying message*. Additionally, it is often desirable for the transformations to be *unidirectional*, i.e., given the token $\Delta_{0,1}$, transforming ciphertexts under sk_1 to ciphertexts under sk_0 should not be possible.

The security notions for symmetric-key PRE are very similar in spirit to those studied extensively in the literature for their public-key counterparts. Security is typically defined in a multi-user setting against a PPT adversary that can issue oracle queries to obtain ciphertexts under the users' secret keys. The adversary can also request for re-encryption tokens, and can *corrupt* a set of users by gaining access to their secret keys. The core security requirement closely resembles indistinguishability of symmetric-key encryption under chosen plaintext attacks - encryptions of any two messages m_0 and m_1 should be indistinguishable, provided that the adversary cannot trivially decrypt them given the obtained secret and re-encryption keys.

In this paper, we adopt the *adaptive* security definitions for PRE presented by Fuchsbauer *et al.* in [FKKP19] to the symmetric-key setting. This definition captures two desirable security properties for PRE, namely - unidirectionality of re-encryption tokens and security against adversaries that can *adaptively* issue queries to the encryption, token generation and corruption oracles. This definition also unifies the security notions for PRE prescribed in a large number of existing works [AFGH05, AFGH06, ABH09, CCL+14].

Definition 4.4. (Symmetric-Key PRE). A symmetric-key PRE scheme PRE for a message space $\mathcal{M}$ is a tuple of five PPT algorithms $(\mathsf{Setup}, \mathsf{RK}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{REnc})$ defined as follows:

- $\mathsf{Setup}(1^\lambda)$: Given the security parameter λ, it generates a secret key sk.
- $\mathsf{RK}(\mathsf{sk}_i, \mathsf{sk}_j)$: On input a "source" secret key sk_i and a "destination" secret key sk_j, it generates a unidirectional re-encryption token $\Delta_{i,j}$.
- $\mathsf{Enc}(\mathsf{sk}, (\mathsf{m}, \ell))$: On input a secret key sk, a message $\mathsf{m} \in \mathcal{M}$ and a level $\ell \in [\lambda]$, it generates a level-ℓ ciphertext (ct, ℓ).
- $\mathsf{Dec}(\mathsf{sk}, (\mathsf{ct}, \ell))$: On input a secret key sk and a level-ℓ ciphertext ct for some level $\ell \in [\lambda]$, it either outputs a message $\mathsf{m}' \in \mathcal{M}$ or $\perp$.

- $\mathsf{REnc}(\Delta_{i,j}, (\mathsf{ct}_i, \ell))$: On input a re-encryption token $\Delta_{i,j}$ and a level-ℓ ciphertext ct_i under a "source" secret key sk_i for some level $\ell \in [\lambda - 1]$, it outputs a transformed level $(\ell + 1)$ ciphertext $(\mathsf{ct}_j, \ell + 1)$ under a "destination" secret key sk_j.

Correctness. A symmetric-key PRE scheme is correct with respect to a message space $\mathcal{M}$ if each of the following conditions are satisfied:

1. For any message $\mathsf{m} \in \mathcal{M}$, any level $\ell \in [\lambda]$, and any $\mathsf{sk} \leftarrow \mathsf{Setup}(1^\lambda)$, we have (with all but negligible probability)
$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, (\mathsf{m}, \ell))) = \mathsf{m}.$$
2. For any message $\mathsf{m} \in \mathcal{M}$, any level $\ell \in [\lambda - 1]$, any pair of secret keys $\mathsf{sk}_i, \mathsf{sk}_j \leftarrow \mathsf{Setup}(1^\lambda)$ and any re-encryption token $\Delta_{i,j} \leftarrow \mathsf{RK}(\mathsf{sk}_i, \mathsf{sk}_j)$, we have (with all but negligible probability)
$$\mathsf{Dec}(\mathsf{sk}_j, \mathsf{REnc}(\Delta_{i,j}, \mathsf{Enc}(\mathsf{sk}_i, (\mathsf{m}, \ell)))) = \mathsf{m}.$$

Security. We adopt the definition of *adaptive* security for PRE schemes proposed and formalized by Fuchsbauer *et al.* in [FKKP19] to the symmetric-key setting. We assume that the adversary has access to the following oracles ($\mathcal{L}$ is a list initialized to empty, while $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ are secret keys generated uniformly at the beginning of the experiment):

1. $\mathcal{O}_{\mathsf{Enc}}$: On input a message m, a level ℓ and an index i, this oracle outputs $(\mathsf{ct}, \ell) \leftarrow \mathsf{Enc}(\mathsf{sk}_i, (\mathsf{m}, \ell))$, where sk_i is the i^{th} secret key, and adds the tuple (ct, ℓ) to the list $\mathcal{L}$.
2. $\mathcal{O}_{\mathsf{RK}}$: When queried with a tuple (i, j), this oracle outputs a re-encryption token $\Delta_{i,j} \leftarrow \mathsf{RK}(\mathsf{sk}_i, \mathsf{sk}_j)$.
3. $\mathcal{O}_{\mathsf{REnc}}$: On input a tuple (i, j) and a level-ℓ ciphertext (ct_i, ℓ) such that $(\mathsf{ct}_i, \ell) \in \mathcal{L}$, this oracle outputs a transformed level-$(\ell + 1)$ ciphertext $(\mathsf{ct}_j, \ell + 1) \leftarrow \mathsf{REnc}(\Delta_{i,j}, (\mathsf{ct}_i, \ell))$, and adds $(\mathsf{ct}_j, \ell + 1)$ to the list $\mathcal{L}$.
4. $\mathcal{O}_{\mathsf{corrupt}}$: This oracle takes as input an index $i \in [n]$ and outputs the secret key sk_i.

For each bit $b \in \{0, 1\}$, define the following experiment $\mathsf{Expt}_b^{\text{ind-pre}}$ between a challenger and an adversary $\mathcal{A}$:

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

1. The challenger generates $\mathsf{sk}_1, \ldots, \mathsf{sk}_n \leftarrow \mathsf{Setup}(1^\lambda)$, and it maintains a list $\mathcal{L}$. Initially, $\mathcal{L} = \{\}$.
2. The adversary $\mathcal{A}$ adaptively issues any number of queries to the $\mathcal{O}_{\mathsf{Enc}}$, $\mathcal{O}_{\mathsf{RK}}$, $\mathcal{O}_{\mathsf{REnc}}$ and $\mathcal{O}_{\mathsf{corrupt}}$ oracles. These oracles update the list $\mathcal{L}$ as described above.
3. The adversary $\mathcal{A}$ eventually outputs a pair of challenge indices (i^*, j^*), a challenge level ℓ^*, and a pair of ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$, subject to the restriction that $(\mathsf{ct}_0, \ell^* - 1), (\mathsf{ct}_1, \ell^* - 1) \in \mathcal{L}$ and $|\mathsf{ct}_0| = |\mathsf{ct}_1|$.
4. The challenger queries the $\mathcal{O}_{\mathsf{RK}}$ oracle to obtain the update token Δ_{i^*, j^*}.
5. The challenger sets $\mathsf{ct}^* \leftarrow \mathsf{REnc}(\Delta_{i^*, j^*}, \mathsf{ct}_b^*)$ and adds the tuple (ct^*, ℓ^*) to the list $\mathcal{L}$. It provides the adversary with ct^* as the challenge ciphertext.
6. The adversary $\mathcal{A}$ continues to adaptively issue any number of queries to the $\mathcal{O}_{\mathsf{Enc}}$, $\mathcal{O}_{\mathsf{RK}}$, $\mathcal{O}_{\mathsf{REnc}}$ and $\mathcal{O}_{\mathsf{corrupt}}$ oracles, subject to the following restriction: if $\mathcal{E}_0^*$ is the set of all identifiers on which $\mathcal{A}$ has queried the $\mathcal{O}_{\mathsf{corrupt}}$ oracle, and $\mathcal{E}_1^*$ is the set of all (i, j) pairs on which $\mathcal{A}$ has queried the $\mathcal{O}_{\mathsf{RK}}$ oracle, then $i^*, j^* \notin \mathcal{E}_0^*$, and there exists no finite sequence of tuples
$$(i_0, i_1), (i_1, i_2), \ldots, (i_{k-1}, i_k) \in \mathcal{E}_1^*,$$
such that $i_0 = i^*$ and $i_k \in \mathcal{E}_0^*$.

Definition 4.5. (IND-UPD secure Symmetric-Key PRE). A symmetric-key PRE scheme $(\mathsf{Setup}, \mathsf{RK}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{REnc})$ is said to be adaptively IND-UPD secure if for all PPT adversaries $\mathcal{A}$, the views of $\mathcal{A}$ in the experiments $\mathsf{Expt}_0^{\text{ind-pre}}$ and $\mathsf{Expt}_1^{\text{ind-pre}}$ are computationally indistinguishable.

We now show that any symmetric-key PRE scheme that satisfies adaptive update indistinguishability implies an updatable encryption scheme with adaptive update indistinguishability. Formally, let $\mathsf{PRE} = (\mathsf{Setup}, \mathsf{RK}, \mathsf{PRE}, \mathsf{Dec}, \mathsf{REnc})$ be an IND-UPD secure symmetric-key PRE scheme. We construct an updatable encryption scheme as follows:

- $\mathsf{UE.Setup}(1^\lambda)$: Given the security parameter λ, output a secret key sk_0 as
$$\mathsf{sk}_0 \leftarrow \mathsf{PRE.Setup}(1^\lambda).$$
- $\mathsf{UE.Next}(\mathsf{sk}_e)$: On input a secret key sk_e for epoch e, generate a new secret key sk_{e+1} and a new update token $\Delta_{e,e+1}$ for epoch $(e+1)$ as
$$\mathsf{sk}_{e+1} \leftarrow \mathsf{PRE.Setup}(1^\lambda), \quad \Delta_{e,e+1} \leftarrow \mathsf{PRE.RK}(\mathsf{sk}_e, \mathsf{sk}_{e+1}).$$
- $\mathsf{UE.Enc}(\mathsf{sk}_e, \mathsf{m})$: On input a secret key sk_e for epoch e and a message $\mathsf{m} \in \mathcal{M}$, output the ciphertext ct_e, where
$$(\mathsf{ct}_e, e) \leftarrow \mathsf{PRE.Enc}(\mathsf{sk}_e, (\mathsf{m}, e)).$$
- $\mathsf{UE.Dec}(\mathsf{sk}_e, \mathsf{ct}_e)$: On input a secret key sk_e and a ciphertext ct_e for some epoch e, it outputs
$$\mathsf{m}' = \mathsf{PRE.Dec}(\mathsf{sk}_e, (\mathsf{ct}_e, e))$$
- $\mathsf{UE.Update}(\Delta_{e,e+1}, \mathsf{ct}_e)$: On input an update token $\Delta_{e,e+1}$ and a ciphertext ct_e for some epoch e, output an updated ciphertext ct_{e+1} for epoch $(e+1)$, where
$$(\mathsf{ct}_{e+1}, e+1) \leftarrow \mathsf{PRE.REnc}(\Delta_{e,e+1}, (\mathsf{ct}_e, e)).$$

Correctness is straightforward to verify. We now formally prove the following theorem.

Theorem 4.6. *If* PRE *is adaptively* IND-UPD *secure, then* UE *is* IND-UPD *secure.*

Proof. Let $\mathcal{A}$ be an adversary that breaks the IND-UPD security of UE with non-negligible advantage ε. We construct an algorithm $\mathcal{B}$ that breaks the IND-UPD security of PRE with advantage ε' such that ε' is negligibly close to ε. $\mathcal{B}$ maintains two lists $\mathcal{L}_0$ and $\mathcal{L}_1$ (both initially empty), and proceeds as follows:

1. Suppose that $\mathcal{A}$ issues an encryption query corresponding to some epoch e and a message $\mathsf{m} \in \mathcal{M}$. $\mathcal{B}$ issues an encryption query to the challenger in the PRE game with message m, level $\ell = e$ and identifier $i = e$. Upon receipt of the ciphertext
$$(\mathsf{ct}_e, e) \leftarrow \mathsf{PRE.Enc}(\mathsf{sk}_e, (\mathsf{m}, e)),$$
$\mathcal{B}$ adds the tuple $(\mathsf{m}, \mathsf{ct}_e, e)$ to its list $\mathcal{L}_0$ and forwards ct_e to $\mathcal{A}$.
2. Suppose that $\mathcal{A}$ issues a *next* query corresponding to some epoch e. $\mathcal{B}$ issues a re-encryption token generation query to the challenger in the PRE game with the tuple $(e, e+1)$. Upon receipt of the corresponding token
$$\Delta_{e,e+1} = \mathsf{PRE.RK}(\mathsf{sk}_i, \mathsf{sk}_j),$$
$\mathcal{B}$ adds the tuple $(e, e+1, \Delta_{e,e+1})$ to the list $\mathcal{L}_1$.

3. Suppose that $\mathcal{A}$ issues an update query on a ciphertext ct_{e-1} corresponding to the epoch $e-1$. $\mathcal{B}$ first checks its list $\mathcal{L}_0$ for an entry of the form $(\mathsf{m}, \mathsf{ct}_{e-1}, e-1)$. If such an entry is not found, ct_{e-1} cannot be an honestly generated ciphertext, and $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ issues re-encryption query to the challenger in the PRE game with the tuple $(e-1, e, (\mathsf{ct}_{e-1}, e-1))$, receives the updated ciphertext ct_e, where

$$(\mathsf{ct}_e, e) \leftarrow \mathsf{PRE.REnc}(\Delta_{e-1,e}, (\mathsf{ct}_{e-1}, e-1)),$$

adds the tuple $(\mathsf{m}, \mathsf{ct}_e, e)$ to its list $\mathcal{L}_0$ and responds to $\mathcal{A}$ with ct_e.

4. Suppose that $\mathcal{A}$ issues a key-corruption query corresponding to some epoch e, subject to the restrictions mentioned in the update-indistinguishability experiment. $\mathcal{B}$ in turn issues a corruption query to the challenger in the PRE game for the identifier e, receives sk_e and forwards the same to $\mathcal{A}$.

5. Suppose that $\mathcal{A}$ issues a token-corruption query corresponding to some epoch e, subject to the restrictions mentioned in the update-indistinguishability experiment. $\mathcal{B}$ looks up the list $\mathcal{L}_1$ to retrieve the re-encryption token $\Delta_{e-1,e}$. If such a token does not exist, it is obtained by issuing re-encryption token generation query to the challenger in the PRE game with the tuple $(e-1, e)$. $\mathcal{B}$ then provides $\mathcal{A}$ with $\Delta_{e-1,e}$.

6. Suppose that at some epoch $e = e^* - 1$, $\mathcal{A}$ outputs a pair of challenge ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$. $\mathcal{B}$ issues a re-encryption query to the challenger in the PRE game with the tuple $(i^*, j^*\ell^*, \mathsf{ct}_0, \mathsf{ct}_1)$, where $i^* = e^* - 1$, and $j^* = \ell^* = e^*$. It receives in response a challenge ciphertext (ct^*, e^*), where

$$(\mathsf{ct}^*, e^*) \leftarrow \mathsf{PRE.REnc}(\Delta_{e^*-1,e^*}, (\mathsf{ct}^*, e^* - 1)),$$

adds the tuple $(\{\}, \mathsf{ct}^*, e^*)$ to its list $\mathcal{L}_0$ and responds to $\mathcal{A}$ with ct^*.

7. $\mathcal{A}$ eventually outputs a bit b'. $\mathcal{B}$ outputs the same bit b'.

Observe that by definition of the update-indistinguishability experiment, $\mathcal{A}$ is restricted from issuing a key-corruption query on any epoch e where it has issued a challenge query. In the simulation, this translates to $\mathcal{B}$ never having to issue a key-corruption query corresponding to any identifier under which it knows an encryption/re-encryption of the challenge ciphertext. In other words, any query from $\mathcal{A}$ that does not violate the restrictions in the update-indistinguishability experiment requires $\mathcal{B}$ to issue a corresponding query that does not violate the restrictions in the PRE indistinguishability experiment. It follows that the advantage of $\mathcal{B}$ is negligibly close to the advantage of $\mathcal{A}$, as required. □

In the aforementioned construction above, we showed how to construct a UE scheme with desirable security properties from any symmetric-key PRE scheme. A natural question to ask is whether the reverse direction is plausible. Unfortunately, based upon the currently accepted definitions for PRE and UE, this seems unlikely.

First, note that while existing definitions for PRE schemes almost always encompass *multi-hop updates* [AFGH05, FKKP19], UE schemes as formalized in [LT18] support a more sequential flavor of updates. In addition, most of the existing UE constructions have bidirectional update tokens (a feature that allows for more efficient constructions in practice), implying that they do not adhere to the "unidirectionality of updates" requirements that PRE schemes are typically expected to meet. This seems to indicate that, as compared to UE, symmetric-key PRE schemes are expected to be more flexible in terms of functionality, while also meeting somewhat stronger security requirements.

We note here that certain early works on UE, notably [BLMR13, EPRS17], modeled their security requirements in the flavor of PRE, meaning that they allowed for key rotation and ciphertext updates across arbitrary epochs. However, as Lehmann and Tackmann explained in in [LT18], such flexibility actually provides the adversary with greater power than is expected in practical applications of key rotation, and also makes achieving post-compromise security for UE significantly harder. Their arguments indicate that it is natural to model UE and its security in a sequential setting, which in turn makes it unlikely that such a UE scheme would imply symmetric-key PRE with reasonable security guarantees.

It is, however, an interesting problem to strengthen the definitions for UE to support "out-of-sequence updates", and to investigate if such an upgrade actually makes it equivalent to symmetric-key PRE.

5 Negative Results in Quantum Setting

In this section we show that any homomorphic one-way function (HOWF) with exact/unbounded homomorphism over abelian groups can be broken using a quantum algorithm. Since exact (or unbounded) KHwPRFs (and hence KHPRFs) over abelian groups trivially imply unbounded HOWFs, it follows that there is no secure construction of an unbounded KHPRF/KHwPRF in quantum world. As a result, a secure KHwPRF either needs to have an approximate homomorphism, or the homomorphism should hold over a non-abelian group.

At a high level, given any abelian group with certain conditions there are known quantum algorithms to determine the structure of the group. That is, given an abelian group $\mathcal{G}$, there is an efficient quantum algorithm to find (an efficiently computable) isomorphism $\psi : \mathcal{G} \to \mathbb{Z}_{q_1} \oplus \cdots \oplus \mathbb{Z}_{q_m}$. We apply this to both the input and output group of a candidate HOWF f. Then we show a simple classic algorithm that given these isomorphisms over the input and output group of f, one can simply break one-wayness of f.

Theorem 5.1. *Let $f : \mathcal{X} \to \mathcal{Y}$ be a (classic) HOWF such that $\mathcal{X}$ and $\mathcal{Y}$ are abelian groups, and there exists an efficient algorithm to find a generating set for $\mathcal{Y}$. There exists a polynomial quantum algorithm that breaks the one-wayness of f with non-negligible advantage.*[1]

First we recall the following fact from algebra. A proof can be found in any standard textbook.

Theorem 5.2. *Any finite abelian group is isomorphic to a direct sum of cyclic groups, and each cyclic group has a prime power order.*

We also rely on the following quantum algorithm (see [CM01] and Section 6.2 of [Chi17] for more details).

Theorem 5.3. *Let $\mathcal{G}$ be a finite abelian group such that (1) each element of $\mathcal{G}$ has a unique decoding, (2) there is an efficient algorithm to do group operations on the elements of $\mathcal{G}$, and (3) there is an efficient algorithm to find a generating set for $\mathcal{G}$. There is a polynomial time quantum algorithm such that decomposes the group $\mathcal{G}$ as*

$$\mathcal{G} = \langle g_1 \rangle \oplus \cdots \oplus \langle g_M \rangle,$$

in terms of the generators $g_1, \ldots, g_M$, and for every $m, m' \in [M]$ such that $m \neq m'$ we have $\langle g_m \rangle \cap \langle g_{m'} \rangle = \{e\}$, where e is the identity element of $\mathcal{G}$. Moreover, the isomorphism

$$\psi : \mathcal{G} \to \mathbb{Z}_{|\langle g_1 \rangle|} \oplus \cdots \oplus \mathbb{Z}_{|\langle g_M \rangle|},$$

(in both ways) can be computed efficiently.

Now we are ready to proceed to the proof of Theorem 5.1. Let $f : \mathcal{X} \to \mathcal{Y}$ be an unbounded HOWF such that $\mathcal{X}$ and $\mathcal{Y}$ are abelian groups. Given a challenge $y^* \in \mathcal{Y}$ such that $y^* := f(x^*)$ for some uniform $x^* \leftarrow \mathcal{X}$, we want to find a preimage x such that $f(x) = y^*$. Let

$$\tilde{\mathcal{X}} := \mathbb{Z}_{p_1} \oplus \cdots \oplus \mathbb{Z}_{p_M} \quad , \quad \tilde{\mathcal{Y}} := \mathbb{Z}_{q_1} \oplus \cdots \oplus \mathbb{Z}_{q_N}$$

be the decomposition of groups $\mathcal{X}$ and $\mathcal{Y}$, respectively, where p_i (respectively, q_j) is a prime power for $m \in [M]$ (respectively, $n \in [N]$). We fix some arbitrary order for the cyclic groups, and we call $\tilde{\mathcal{X}}$ an *explicit representation* of $\mathcal{X}$. Using Theorem 5.3, we can efficiently compute the isomorphisms $\psi_{\mathcal{X}}, \psi_{\mathcal{Y}}$ (and their inverses) for any element in the domain of the isomorphism where

$$\psi_{\mathcal{X}} : \mathcal{X} \to \tilde{\mathcal{X}} \quad , \quad \psi_{\mathcal{Y}} : \mathcal{Y} \to \tilde{\mathcal{Y}}.$$

We define $\tilde{f} : \tilde{\mathcal{X}} \to \tilde{\mathcal{Y}}$ as the analog of f over the explicit representations of $\mathcal{X}$ and $\mathcal{Y}$, i.e., define

$$\tilde{f}(\tilde{x}) = \psi_{\mathcal{Y}}(f(\psi_{\mathcal{X}}^{-1}(\tilde{x}))).$$

It is not hard to see that $f(x) = y$ is equivalent to $\tilde{f}(\psi_{\mathcal{X}}(x)) = \psi_{\mathcal{Y}}(y)$. Because the isomorphisms $\psi_{\mathcal{X}}, \psi_{\mathcal{Y}}$ and their inverses are efficiently computable, it is enough to show an attack against one-wayness of $\tilde{f}$.

For each $n \in [N]$, we define $\mathbf{e}_n \in \mathbb{Z}_{p_1} \oplus \cdots \oplus \mathbb{Z}_{p_N}$ to be the (unit) vector whose nth component is 1, and all other components are 0.[2] For an element $\tilde{y} \in \tilde{\mathcal{Y}}$, let $[\tilde{y}]_m \in \mathbb{Z}_{q_m}$ be the mth component of $\tilde{y}$. We compute the index set I_m

[1] Notice that it is almost always the case that $\mathcal{Y}$ is an efficiently samplable group. By Theorem 5 of [AGKP14], a set of uniform elements with size $3 \log|\mathcal{Y}|$ forms a generating set for $\mathcal{Y}$ with an overwhelming probability.

[2] Notice that each component may live in a different cyclic group.

for each $m \in M$ as

$$I_m = \{n \in [N] \mid [\tilde{f}(\mathbf{e}_n)]_m \neq 0\}.$$

All index sets $\{I_m\}_{m\in[M]}$ can be computed efficiently since both N and M are polynomially bounded. Define a vector of variables $\mathbf{z} = (z_1, \ldots, z_N) \in \mathbb{Z}^N$, and for each $m \in [M]$, consider the following system of modular equations where $\{z_i\}_{i\in I_m}$ are the (unknown) variables:

$$S_m : \quad \sum_{i\in I_m} z_i[\tilde{f}(\mathbf{e}_i)]_m \equiv [\tilde{y}]_m \pmod{q_m}$$

Consider the following observations:

- Without loss of generality we can assume that for two distinct $m, m' \in [M]$, we have $\gcd(q_m, q'_m) = 1$. If $q_m = q'_m$, we can simply merge S_m and $S_{m'}$. If $q_m < q_{m'}$ and $\gcd(q_m, q'_m) > 1$, we can "lift" the equation in $S_{m'}$ simply by multiplying the both sides by $q^{m'}/q^m$ and adding the resulting equation to $S_{m'}$. We refer to this part as "merging step".
- Observe that for any two *prime powers* p and q, if there is a non-trivial homomorphism from $\mathbb{Z}_p$ to $\mathbb{Z}_q$ then either $p \mid q$ or $q \mid p$. Therefore, if z_n appears in S_m (or equivalently $n \in I_m$), we either have $p_n \mid q_m$ or $q_m \mid p_n$.

Let $\overline{M} \subseteq M$ be the set of indices after the "merging step". Using the previous observations, it follows that

- For any two distinct $\overline{m}_1, \overline{m}_2 \in \overline{M}$, we have $\gcd(q_{\overline{m}_1}, q_{\overline{m}_2}) = 1$.
- For any $n \in N$, there is at most one $\overline{m} \in \overline{M}$ such that the variable z_n appears in $S_{\overline{m}}$.

Each system of equation(s) $S_{\overline{m}}$ can be seen as a system of linear equation(s) over the group $\mathbb{Z}_{q_{\overline{m}}}$, and it can solved using the known algorithms for solving linear equations over finite abelian groups, e.g., [GR02]. One can equivalently interpret each $S_{\overline{m}}$ as a system of equations over the finite *ring* $\mathbb{Z}_{q_{\overline{m}}}$ (since $q_{\overline{m}}$ is not necessarily prime).

By solving each system $S_{\overline{m}}$, we can determine the vector $\mathbf{z} \in \mathbb{Z}^N$. Finally, we output $\tilde{x}$ as the preimage of $\tilde{y}$ the attacker where

$$\tilde{x} = (z_1 \bmod p_1, \ldots, z_N \bmod p_n).$$

By construction, we know that the vector $\mathbf{z}$ satisfies all system of equation(s) $\{S_m\}_{m\in M}$. It follows that $\tilde{f}(\tilde{x}) = \tilde{y}$, as required.

Building Quantum-Secure Primitives from Abelian Groups. Our results here may have some implications for the construction of quantum-secure primitives over abelian groups. For instance, in [AMPR19], the authors showed that many public-key cryptosystems can be built from generic primitives with exact homomorphism. Our results here give evidence that such constructions are not going to be quantum-secure when instantiated with new assumptions that rely on abelian groups. Lattice-based primitives do not support exact homomorphisms, which makes them immune to a wide class of quantum attacks. However, there do exist other assumptions relying on abelian groups, such as isogeny-based assumptions [JD11], for which similar notions of homomorphism are yet to be explored [dOPS18].

References

[ABH09] G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy re-encryption. In M. Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 279–294. Springer, Heidelberg, April 2009.

[ABPW13] Y. Aono, X. Boyen, L. T. Phong, and L. Wang. Key-private proxy re-encryption under LWE. In G. Paul and S. Vaudenay, editors, *INDOCRYPT 2013*, volume 8250 of *LNCS*, pages 1–18. Springer, Heidelberg, December 2013.

[AFGH05] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *NDSS 2005*. The Internet Society, February 2005.

[AFGH06] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.

[AGKP14] F. Armknecht, T. Gagliardoni, S. Katzenbeisser, and A. Peter. General impossibility of group homomorphic encryption in the quantum world. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 556–573. Springer, Heidelberg, March 2014.

[AMPR19] N. Alamati, H. Montgomery, S. Patranabis, and A. Roy. Minicrypt primitives with algebraic structure and applications. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 55–82. Springer, Heidelberg, May 2019.

[Bar17] B. Barak. The complexity of public-key cryptography. In *Tutorials on the Foundations of Cryptography*, pages 45–77. 2017.

[BBS98] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 127–144. Springer, Heidelberg, May / June 1998.

[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.

[BFP+15] A. Banerjee, G. Fuchsbauer, C. Peikert, K. Pietrzak, and S. Stevens. Key-homomorphic constrained pseudorandom functions. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 31–60. Springer, Heidelberg, March 2015.

[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.

[BP14] A. Banerjee and C. Peikert. New and improved key-homomorphic pseudorandom functions. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 353–370. Springer, Heidelberg, August 2014.

[BPR12] A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012.

[BR17] A. Bogdanov and A. Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography*, pages 79–158. 2017.

[BV15] Z. Brakerski and V. Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 1–30. Springer, Heidelberg, March 2015.

[CCL+14] N. Chandran, M. Chase, F.-H. Liu, R. Nishimaki, and K. Xagawa. Re-encryption, functional re-encryption, and multi-hop re-encryption: A framework for achieving obfuscation-based security and instantiations from lattices. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 95–112. Springer, Heidelberg, March 2014.

[CH07] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, editors, *ACM CCS 2007*, pages 185–194. ACM Press, October 2007.

[Chi17] A. M. Childs. Lecture notes on quantum algorithms, 2017. https://www.cs.umd.edu/~amchilds/qa/qa.pdf.

[CM01] K. K. H. Cheung and M. Mosca. Decomposing finite abelian groups. *Quantum Information & Computation*, 1(3):26–32, 2001.

[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.

[DKL+18] D. Derler, S. Krenn, T. Lorünser, S. Ramacher, D. Slamanig, and C. Striecks. Revisiting proxy re-encryption: Forward secrecy, improved security, and applications. In M. Abdalla and R. Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 219–250. Springer, Heidelberg, March 2018.

[DKPW12] Y. Dodis, E. Kiltz, K. Pietrzak, and D. Wichs. Message authentication, revisited. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 355–374. Springer, Heidelberg, April 2012.

[dOPS18] C. de Saint Guilhem, E. Orsini, C. Petit, and N. P. Smart. Secure oblivious transfer from semi-commutative masking. Cryptology ePrint Archive, Report 2018/648, 2018. https://eprint.iacr.org/2018/648.

[EPRS17] A. Everspaugh, K. G. Paterson, T. Ristenpart, and S. Scott. Key rotation for authenticated encryption. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 98–129. Springer, Heidelberg, August 2017.

[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.

[FKKP19] G. Fuchsbauer, C. Kamath, K. Klein, and K. Pietrzak. Adaptively secure proxy re-encryption. In D. Lin and K. Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 317–346. Springer, Heidelberg, April 2019.

[GA07] M. Green and G. Ateniese. Identity-based proxy re-encryption. In J. Katz and M. Yung, editors, *ACNS 07*, volume 4521 of *LNCS*, pages 288–306. Springer, Heidelberg, June 2007.

[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.

[GR02] M. Goldmann and A. Russell. The complexity of solving equations over finite groups. *Inf. Comput.*, 178(1):253–262, 2002.

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

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

[JD11] D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In B.-Y. Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011.

[KO97] E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th FOCS*, pages 364–373. IEEE Computer Society Press, October 1997.

[KW19] V. Koppula and B. Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In *CRYPTO (To Appear)*. 2019.

[LMR14] K. Lewi, H. W. Montgomery, and A. Raghunathan. Improved constructions of PRFs secure against related-key attacks. In I. Boureanu, P. Owesarski, and S. Vaudenay, editors, *ACNS 14*, volume 8479 of *LNCS*, pages 44–61. Springer, Heidelberg, June 2014.

[LST18] B. Libert, D. Stehlé, and R. Titiu. Adaptively secure distributed PRFs from LWE. In A. Beimel and S. Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 391–421. Springer, Heidelberg, November 2018.

[LT18] A. Lehmann and B. Tackmann. Updatable encryption with post-compromise security. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 685–716. Springer, Heidelberg, April / May 2018.

[LV08] B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In R. Cramer, editor, *PKC 2008*, volume 4939 of *LNCS*, pages 360–379. Springer, Heidelberg, March 2008.

[Mon18] H. Montgomery. More efficient lattice PRFs from keyed pseudorandom synthesizers. In D. Chakraborty and T. Iwata, editors, *INDOCRYPT 2018*, volume 11356 of *LNCS*, pages 190–211. Springer, Heidelberg, December 2018.

[nBL17] E. L. (née Berners-Lee). Improved security notions for proxy re-encryption to enforce access control. Cryptology ePrint Archive, Report 2017/824, 2017. https://eprint.iacr.org/2017/824.

[NPR99] M. Naor, B. Pinkas, and O. Reingold. Distributed pseudo-random functions and KDCs. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 327–346. Springer, Heidelberg, May 1999.

[NR95] M. Naor and O. Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. In *36th FOCS*, pages 170–181. IEEE Computer Society Press, October 1995.

[NR97] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[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.

[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.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

[SNS11] A. Syalim, T. Nishide, and K. Sakurai. Realizing proxy re-encryption in the symmetric world. In A. Abd Manaf, A. Zeki, M. Zamani, S. Chuprat, and E. El-Qawasmeh, editors, *Informatics Engineering and Information Science*, pages 259–274. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.