A preliminary version of this paper appears in the Proceedings of CRYPTO 2010. This is the full version.

Pseudorandom Functions and Permutations Provably Secure Against Related-Key Attacks

Mihir Bellare[1] David Cash[2]

July 2010

Abstract

This paper fills an important foundational gap with the first proofs, under standard assumptions and in the standard model, of the existence of pseudorandom functions (PRFs) and pseudorandom permutations (PRPs) resisting rich and relevant forms of related-key attacks (RKA). An RKA allows the adversary to query the function not only under the target key but under other keys derived from it in adversary-specified ways. Based on the Naor-Reingold PRF we obtain an RKA-PRF whose keyspace is a group and that is proven, under DDH, to resist attacks in which the key may be operated on by *arbitrary* adversary-specified group elements. Previous work was able only to provide schemes in idealized models (ideal cipher, random oracle), under new, non-standard assumptions, or for limited classes of attacks. The reason was technical difficulties that we resolve via a new approach and framework that, in addition to the above, yields other RKA-PRFs including a DLIN-based one derived from the Lewko-Waters PRF. Over the last 15 years cryptanalysts and blockcipher designers have routinely and consistently targeted RKA-security; it is visibly important for abuse-resistant cryptography; and it helps protect against fault-injection sidechannel attacks. Yet ours are the first significant proofs of existence of secure constructs. We warn that our constructs are proofs-of-concept in the foundational style and not practical.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: `mihir@cs.ucsd.edu`. URL: `http://www.cs.ucsd.edu/users/mihir`. Supported in part by NSF grants CNS-0627779 and CCF-0915675.

[2] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: `cdc@ucsd.edu`. URL: `http://www.cs.ucsd.edu/users/cdcash`. Supported in part by NSF grants CNS-0627779 and CCF-0915675.

Contents

1 Introduction

Alarmed by the number of successful related-key attacks (RKAs) against real blockciphers [16, 18, 17, 42, 46, 21, 10, 11, 13, 12, 54, 59, 32, 40, 14, 43, 38], theoreticians have stepped back to ask to what extent the underlying goal of RKA-secure PRFs and PRPs is achievable at all. The question is made challenging by the unusual nature of the attack model which allows the adversary to manipulate the key. Previous works providing RKA-secure PRFs and PRPs have bypassed rather than overcome the core technical difficulties by using the ideal cipher or random oracle models, making non-standard assumptions themselves "related-key" in nature, or limiting attackers to weak classes of RKAs for which the problem disappears [6, 50]. We provide a new technical approach based on which we obtain the first designs of PRFs and PRPs secure against non-trivial and application-relevant forms of RKAs under standard assumptions (DDH) and in the standard model. Our constructions are not practical, providing, instead, in-principle proofs of achievability of the goals in the classical foundational style.

The model. RKAs were introduced by Biham and Knudsen [8, 9, 44] and formalized by Bellare and Kohno (BK) [6]. Referring to any $\phi\colon \mathcal{K} \rightarrow \mathcal{K}$ as a related-key deriving (RKD) function, the latter define what it means for a family of functions $F\colon \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ to be a Φ-RKA-PRF, where Φ is a class (set) of RKD functions. The game begins by picking a random challenge bit b, a random target key $K \in \mathcal{K}$ and, for each $L \in \mathcal{K}$, a random function $G_L\colon \mathcal{D} \rightarrow \mathcal{R}$. The adversary is allowed multiple queries to an oracle that, given a pair $(\phi, x) \in \Phi \times \mathcal{D}$, returns $F_{\phi(K)}(x)$ if $b = 1$ and $G_{\phi(K)}(x)$ if $b = 0$, and its advantage is $2\Pr[b = b'] - 1$, where b' is the bit it outputs. The definition of a family of permutations (blockcipher) $F\colon \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{D}$ being a Φ-RKA-PRP is analogous, the difference being that each G_L is a random permutation on $\mathcal{D}$ rather than a random function. Note that when Φ consists of just the identity function, we recover the standard PRF [36] and PRP [49] notions.

Group-induced classes. We must beware of inherent limitations. It is observed in [6] that some Φ are "impossible" in the sense that *no* F can be a Φ-RKA-PRF or a Φ-RKA-PRP. Indeed, any Φ that contains a constant function $\phi(\cdot) = C$, for some attacker-known constant $C \in \mathcal{K}$, is impossible. (For some x, just query the RK-oracle with (ϕ, x) and return 1 if the response is $F_C(x)$.) The class of all RKD functions is impossible, and so is the class of all permutations. The basic foundational question, then, is to identify *specific* classes Φ, as rich, interesting and relevant as possible, for which we can prove "possibility," meaning existence of Φ-RKA-PRFs and Φ-RKA-PRPs. But which classes are good candidates?

BK [6] showed the (standard model) possibility of any class Φ whose member RKDs modify only the second half of the given key, and Lucks [50] gave, for the same class, an alternative construction with better concrete security. But if part of the key is unmodified, we can just use it as the "actual" key and put the rest in the input, meaning RKA-security here is for "trivial" reasons. For the proof-of-concept results in which we are interested, we seek candidate classes where the core technical difficulties cannot be bypassed in this way.

Luckily, Lucks [50] has already pinpointed a worthy target. His group-induced classes are elegant, appealing, non-trivial and application-relevant. If $(\mathcal{K}, *)$ is a group under an operation "$*$", the associated group-induced class is $\mathsf{rkd}[\mathcal{K}, *] = \{ \phi^*_\Delta : \Delta \in \mathcal{K} \}$ where $\phi^*_\Delta(K) = K * \Delta$ for all $K \in \mathcal{K}$. These classes are rich because all group actions are included. They also have what in [35] is called the completeness property and viewed as important to non-triviality of the class, namely that for any $K, K' \in \mathcal{K}$ there is a $\phi \in \mathsf{rkd}[\mathcal{K}, *]$ such that $\phi(K) = K'$. Security relative to these classes suffices for applications and cannot be established by tricks such as the above. The quest that emerges is to find (non-trivial) groups $(\mathcal{K}, *)$ for which we can show the possibility of $\mathsf{rkd}[\mathcal{K}, *]$, meaning exhibit $\mathsf{rkd}[\mathcal{K}, *]$-RKA-PRFs and $\mathsf{rkd}[\mathcal{K}, *]$-RKA PRPs $F\colon \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ whose keyspace is $\mathcal{K}$.

Previous work. Results of [6] imply that ideal ciphers achieve $\mathsf{rkd}[\mathcal{K}, *]$-RKA-PRP security for *any* large enough group $(\mathcal{K}, *)$. Also, one can easily strengthen a given PRF or PRP to be a $\mathsf{rkd}[\mathcal{K}, *]$-RKA one by hashing the key with a random oracle before use [50]. However, it is unclear how to instantiate the ideal primitives here to get "real" constructions for even a *single* group [28, 19]. For certain composite numbers M, Lucks [50] provides $\mathsf{rkd}[\mathbb{Z}_M, +]$-RKA-PRFs for the group $(\mathbb{Z}_M, +)$, where $+$ is addition modulo M, but the assumptions on which he bases security are not only interactive and novel but also themselves "related-key" in nature and uncomfortably close to just assuming the construct itself is secure, making the value of the proofs debatable from the point of view of security assurance. Existing PRFs such as the DDH-based one of Naor and Reingold [52] or the DLIN-based one of Lewko and Waters [47] are subject to simple attacks showing they provide no RKA-security. (Nonetheless they will be a starting point for our constructs.) Research has expanded to consider RKA-security of other primitives while leaving the goal unachieved for the more basic PRF, PRP and PRG ones [1, 35].

The salient fact that emerges from this previous work is that we do not have *even a single example* of a group $(\mathcal{K}, *)$ for which we can prove the existence of a $\mathsf{rkd}[\mathcal{K}, *]$-RKA-PRF or $\mathsf{rkd}[\mathcal{K}, *]$-RKA-PRP under standard assumptions in the standard model. The reason for the lack of progress is technical obstacles. The attack models underlying standard definitions of standard primitives do not allow any key-manipulation by the adversary. This makes it unclear how one can do any reductions, which seem to require applying RKD functions to an unknown key. This difficulty is appreciated, with Goldenberg and Liskov [35, Section 4] saying "The major open problem in related-secret security is whether or not related-key secure blockciphers exist ... related-secret pseudorandom bits cannot be constructed using traditional techniques. This leaves a significant open problem ... can fundamentally new techniques be found to create related-secret pseudorandom bits?"

New RKA-PRFs. We fill the above gap, providing the first constructions, under the standard DDH assumption and in the standard model, of Φ-RKA-PRFs where Φ is group-induced. We obtain and analyze our designs via a general framework using two new primitives which may be of independent interest, namely key-malleable PRFs and key fingerprints. However, (surprisingly) at least one of our constructions, that we call the multiplicative DDH based RKA-PRF, is compact enough to state here. Let NR: $(\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ denote the Naor-Reingold PRF [52] that given key $\mathbf{a} = (\mathbf{a}[0], \ldots, \mathbf{a}[n]) \in (\mathbb{Z}_p^*)^{n+1}$ and input $x = x[1] \ldots x[n] \in \{0,1\}^n$ returns

$$\mathrm{NR}(\mathbf{a}, x) \;=\; g^{\mathbf{a}[0] \prod_{i=1}^{n} \mathbf{a}[i]^{x[i]}} \;, \tag{1}$$

where $\mathbb{G} = \langle g \rangle$ is a group of prime order p. The keyspace $\mathcal{K} = (\mathbb{Z}_p^*)^{n+1}$ is a group under the operation $*$ of component-wise multiplication modulo p, but simple attacks [6] show that NR is not itself a $\mathsf{rkd}[\mathcal{K}, *]$-RKA-PRF. Let h be a collision-resistant hash function with domain $\{0,1\}^n \times \mathbb{G}^{n+1}$ and range $\{0,1\}^{n-2}$. Given key $\mathbf{a}$ and input x, our construct F: $(\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ returns

$$F(\mathbf{a}, x) \;=\; \mathrm{NR}(\mathbf{a}, 11 \| h(x, (g^{\mathbf{a}[0]}, g^{\mathbf{a}[0]\mathbf{a}[1]}, \ldots, g^{\mathbf{a}[0]\mathbf{a}[n]}))) \;,$$

where "$\|$" denotes concatenation. Theorem 4.2 says that F is a $\mathsf{rkd}[(\mathbb{Z}_p^*)^{n+1}, *]$-RKA-PRF under the DDH assumption. The difficulty such a proof had to overcome was how the "simulator," given $\mathbf{d}$, can answer queries for F on keys of the form $\mathbf{a} * \mathbf{d}$ without itself knowing $\mathbf{a}$ and without contradicting RKA security by enabling an attack.

This and other results are obtained via a general framework hinging on two new primitives. We call a PRF M: $\mathcal{K} \times \mathcal{D} \to \mathcal{R}$ *key-malleable* relative to a class Φ of RKD functions on $\mathcal{K}$ if there is an efficient algorithm that given $(\phi, x) \in \Phi \times \mathcal{D}$ and oracle access to M_K returns $M_{\phi(K)}(x)$. That this could be useful for building a Φ-RKA-PRF is, on the one, hand, intuitive, because it allows us to simulate an oracle for $M(\phi(K), \cdot)$ via an oracle for $M(K, \cdot)$. But it is, on the other hand, counter-intuitive, because the same property immediately gives rise to an attack showing that M is *not* a Φ-RKA-PRF! Something else is necessary. This turns out to be the new concept of a *key fingerprint*, a vector $\mathbf{w}$ over $\mathcal{D}$ that uniquely identifies a key in the sense that for all $(\phi, \phi', K) \in \Phi \times \Phi \times \mathcal{K}$ we have $M_{\phi(K)}(\mathbf{w}) \neq M_{\phi'(K)}(\mathbf{w})$ whenever $\phi \neq \phi'$, where we have extended M to vector second arguments on which it operates component-wise. Given $M, \mathbf{w}$ and a collision-resistant hash function, our general construction shows how to build F: $\mathcal{K} \times \mathcal{D} \to \mathcal{R}$ that we can show is a Φ-RKA-PRF (cf. Theorem 3.1).

The multiplicative DDH based RKA-PRF noted above is obtained by showing that NR is a key-malleable PRF relative to $\mathsf{rkd}[(\mathbb{Z}_p^*)^{n+1}, *]$ and then finding a key fingerprint for it. It is interesting that we turn malleability [31], typically viewed as a "bad" property, into a "good" property that we can exploit.

Two more constructs emanate from this framework. There are groups where DDH is easy but the Decision Linear (DLIN) problem of [25] still seems hard. Lewko and Waters [47] provide a DLIN-based analogue of the Naor-Reingold PRF, commenting that they know of no "closed-form" rendition of it akin to the above Equation (1) for NR. Using matrices, we provide in Equation (21) such a closed-form, and then, restricting attention to invertible matrices and slightly modifying the function, we obtain in Equation (23) a PRF that we can show is key-malleable and admits a key fingerprint. Our framework then yields the DLIN-based RKA-PRF of Theorem 5.3.

The group $(\mathbb{Z}_p^*)^{n+1}$ underlying our multiplicative DDH-based RKA-PRF is, as the name indicates, multiplicative. Providing a DDH-based $\mathsf{rkd}[\mathbb{Z}^{n+1}, *]$-RKA-PRF where $*$ is component-wise addition modulo p is more difficult. We provide in Theorem 6.3 a solution that involves first modifying the Naor-Reingold PRF and then applying our framework. However, the running time of our reduction is exponential in the input size. Theoretically, this means we must assume hardness of DDH against exponential-time algorithms. In practice, one can get security by using larger groups. This situation parallels that for the BB IBE scheme [23].

From RKA-PRFs to RKA-PRPs. Practical interest centers on RKA-secure blockciphers, meaning PRPs, and the constructions above are RKA-PRFs. It is not clear how one might modify the constructions to get RKA-PRPs. We use a different approach. Using deterministic extractors [27, 33, 29], we convert our Φ-RKA-PRFs into Φ-RKA-PRGs with bitstring outputs. When these are used as key-derivation functions to key an ordinary (not RKA) PRP, we obtain a Φ-RKA-PRP. (This second, composition step extends similar ones from [50, 35]). For each class Φ for which we have a Φ-RKA-PRF, this not only yields a CPA-secure Φ-RKA-PRP but even a CCA-secure one.

Related work and techniques. Based on the Boneh-Boyen short signature scheme [24], Dodis and Yampolskiy [30] define a PRF BBDY: $\mathbb{Z}_p \times S \to \mathbb{G}$ via $\mathrm{BBDY}(k,x) = \mathbf{e}(g,g)^{1/(k+x)}$, where $\mathbf{e}\colon \langle g\rangle \times \langle g\rangle \to \mathbb{G}$ is a bilinear map and $S \subseteq \mathbb{Z}_p$. This had seemed to us promising towards building a $\mathsf{rkd}[\mathbb{Z}_p, +]$-RKA-PRF, but (disappointingly) did not lead there. To begin with, BBDY is easily shown by attack to not itself be a $\mathsf{rkd}[\mathbb{Z}_p, +]$-RKA-PRF. (Adding 1 to k or to x yields the same outcome.) By exploiting the symmetry between k and x and using the composition paradigm, it turns out one can show how to construct a $\mathsf{rkd}[\mathbb{Z}_p, +]$-RKA-PRF if BBDY was a (plain) PRF, but *only if the input domain S was equal to $\mathbb{Z}_p$*. The problem is that the q-DBDHI-based proof of [30, 24] requires S to be "small" and in particular delivers nothing at all when $S = \mathbb{Z}_p$. We comment that there is no attack showing BBDY is not a PRF when $S = \mathbb{Z}_p$ and one might prove this in the generic model, but there seems little reason to pursue a generic group model solution when we already have a standard model, DDH-based solution. (In fact, since DDH is hard in the generic group model [58], our results already imply a generic model solution anyway.)

RKA-security is much easier for randomized primitives than deterministic ones. From the ElGamal scheme over a group of prime order p, one can easily get a (randomized) $\mathsf{rkd}[\mathbb{Z}_p, +]$-RKA-CPA-secure DDH-based symmetric encryption scheme. Applebaum [1] presents a more efficient $\mathsf{rkd}[\{0,1\}^n, \oplus]$-RKA-CPA-secure (still randomized) symmetric encryption scheme assuming hardness of the LPN problem. There seems to be no simple way, from these techniques, to get the full-fledged group-induced RKA-PRFs that we target, where the computation is deterministic. That the deterministic case is more difficult than the randomized one is not surprising or unusual. In analogy, DDH based injective trapdoor functions [53] were discovered much later than DDH-based public-key encryption schemes, and fully PRIV-secure deterministic public-key encryption remains an open problem [4, 22].

Goldenberg and Liskov [35] broaden the scope to consider related-secret security. As with Lucks [50] they can, via composition, reduce the design of Φ-RKA-PRFs to the design of Φ-RKA-PRGs, but provide no new constructions of the latter and hence of the former. They have negative results indicating the difficulty of getting these for non-trivial classes Φ, and comment [35, Section 1] that "This leads us to the conclusion that if related-secret pseudorandomness (including related-key blockciphers) are possible, they must be proven either based on other related-secret pseudorandomness assumptions, or a dramatically new way of creating pseudorandomness from hardness must be developed." Our results are answers to these questions, showing that one can in fact obtain related-key pseudorandomness under standard assumptions. (Our RKA-PRFs of course directly yield RKA-PRGs.) Their negative results are in a limited model of computation and do not apply in our context.

RKA-security and encryption of key-dependent messages [20] have in common the technical difficulty of how to do reductions without access to the keys one is trying to attack, but no connection between the two is known.

Context. Conceived with the goal of studying the strength of blockcipher key-schedules [8, 9, 44], RKAs quickly became mainstream. RKA-security is viewed as necessary for the collision-resistance of blockcipher-based compression functions [55]. (But one should note that this view has no formal justification.) RKA-resistance was a stated design goal of AES and remains so for other modern ciphers. A successful RKA is universally viewed by cryptanalysts as a break of the cipher. The recent attention-grabbing attacks on AES-192 and AES-256 [18, 17, 16] were RKAs, and far from unique in this regard: a look at the literature shows that RKAs abound [42, 46, 21, 10, 11, 13, 12, 54, 59, 32, 40, 14, 43, 38]. Several higher-level cryptographic constructs, including HMAC [5, 2], the 3GPP confidentiality and integrity algorithms f8,f9 [39], and RMAC [41, 45], use related keys and thus rely for their (standard, not RKA) security on RKA-security of the underlying compression function or blockcipher.

The most direct use of RKA-security is for very cheap, simple and natural ways to rekey or tweak block ciphers. Subkeys of K for use with modes of operation of a blockcipher E might be derived in standard usage via $E_K(\Delta_1), E_K(\Delta_2), \ldots$ where $\Delta_1, \Delta_2, \ldots$ are constants. If E is a RKA-PRP one can just use instead $K * \Delta_1, K * \Delta_2, \ldots$, where $*$ is a group operation, saving many blockcipher operations. On the other hand if E is a $\mathsf{rkd}[\mathcal{K}, *]$-RKA-PRP, then $F_K^T(x) = E_{K*T}(x)$ is shown in [6] to be a tweakable blockcipher, a primitive that has proven to be of great importance both conceptually and in applications [48, 56]. More designs would probably use related keys if it were possible to do so safely. Non-expert (in practice, most!) designers do it anyway, making RKA-security, in the words of Biryukov, Dunkelman, Keller, Khovratovich and Shamir [16], central to abuse-resistant cryptography.

Beyond this, RKA-security provides resistance to fault injection attacks [26, 15] where the attacker can inject faults that change bits of a hardware-stored key and observe the outputs of the cryptographic primitive under the modified key, putting RKAs under the umbrella of sidechannel attacks. This sidechannel connection is captured by the tamper-proof security model of Gennaro, Lysyanskaya, Malkin, Micali and Rabin [34]. (They were apparently not aware of the prior model of [6] and the cryptanalytic literature on RKAs. We hope our current paper helps connect these two lines of work.)

Overall, the motivation for the theoretical study of RKA-security is not just powerful but unusual in coming from so many different parts of cryptography, namely foundations, cryptanalysis, protocol design and resistance to sidechannel attacks.

2 Basic definitions

A family of functions $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ takes a key $K \in \mathcal{K}$ and input $x \in \mathcal{D}$ and returns an output $F_K(x) = F(K, x) \in \mathcal{R}$. Let $\mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R})$ be the set of all families of functions $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$. For sets X, Y let $\mathsf{Fun}(X, Y)$ be the set of all functions mapping X to Y. If S is a (finite) set then $s \overset{\$}{\leftarrow} S$ denotes the operation of picking s from S at random and $|S|$ is the size of S. We denote by $y \overset{\$}{\leftarrow} A(x_1, x_2, \ldots)$ the operation of running randomized algorithm A on inputs $x_1, x_2, \ldots$ and fresh coins and letting y denote the output. If $\mathbf{v}$ is a vector then $|\mathbf{v}|$ denotes the number of its coordinates and $\mathbf{v}[i]$ denotes its i-th coordinate, meaning $\mathbf{v} = (\mathbf{v}[1], \ldots, \mathbf{v}[|\mathbf{v}|])$. A (binary) string x is identified with a vector over $\{0, 1\}$ so that $|x|$ is its length and $x[i]$ is its i-th bit. If $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is a family of functions and $\mathbf{x}$ is a vector over $\mathcal{D}$ then $F(K, \mathbf{x})$ denotes the vector $(F(K, \mathbf{x}[1]), \ldots, F(K, \mathbf{x}[|\mathbf{x}|]))$. Read the term "efficient" as meaning "polynomial-time" in the natural asymptotic extension of our concrete framework.

GAMES. Some of our definitions and proofs are expressed via code-based games [7]. Recall that such a game —see Figure 1 for an example— consists of an (optional) INITIALIZE procedure and procedures to respond to adversary oracle queries. A game G is executed with an adversary A as follows. First, INITIALIZE (if present) executes. Then A executes, its oracle queries being answered by the corresponding procedures of G. When A terminates, its output, denoted G^A, is called the output of the game, and we let "$\mathrm{G}^A \Rightarrow 1$" denote the event that this game output takes value 1. Boolean flags are assumed initialized to false. The running time of an adversary by convention is the worst case time for the execution of the adversary with any of the games defining its security, so that the time of the called game procedures is included. When (as often) we describe a game in text and say the game "begins" by doing something, we are describing how INITIALIZE works.

PRFS. The advantage of an adversary A in attacking the (standard) prf security of a family of functions $F\colon\ \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is defined via

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(A) \;=\; \Pr\left[\,\mathrm{PRFReal}^A_F \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{PRFRand}^A_F \Rightarrow 1\,\right] . \tag{2}$$

Game $\mathrm{PRFReal}_F$ begins by picking $K \xleftarrow{\$} \mathcal{K}$ and responds to oracle query $\mathrm{F{\scriptsize N}}(x)$ via $F(K, x)$. Game $\mathrm{PRFRand}_F$ begins by picking $f \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$ and responds to oracle query $\mathrm{F{\scriptsize N}}(x)$ via $f(x)$.

RKA-PRFS. We recall definitions from [6]. Let $F\colon\ \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a family of functions and $\Phi \subseteq \mathsf{Fun}(\mathcal{K}, \mathcal{K})$. The members of Φ are called RKD (related-key deriving) functions. An adversary is said to be Φ-restricted if its oracle queries (ϕ, x) satisfy $\phi \in \Phi$. The advantage of a Φ-restricted adversary A in attacking the prf-rka security of F is defined via

$$\mathbf{Adv}^{\mathrm{prf\text{-}rka}}_{\Phi,F}(A) \;=\; \Pr\left[\,\mathrm{RKPRFReal}^A_F \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{RKPRFRand}^A_F \Rightarrow 1\,\right] . \tag{3}$$

Game $\mathrm{RKPRFReal}_F$ begins by picking $K \xleftarrow{\$} \mathcal{K}$ and responds to oracle query $\mathrm{RKF{\scriptsize N}}(\phi, x)$ via $F(\phi(K), x)$. Game $\mathrm{RKPRFRand}_F$ begins by picking $K \xleftarrow{\$} \mathcal{K}$ and $G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R})$, and responds to oracle query $\mathrm{RKF{\scriptsize N}}(\phi, x)$ via $G(\phi(K), x)$.

CR HASH FUNCTIONS. The advantage of C in attacking the cr (collision-resistance) security of $H\colon\ \mathcal{D} \to \mathcal{R}$ is

$$\mathbf{Adv}^{\mathrm{cr}}_{H}(C) \;=\; \Pr\left[\, x \neq x' \text{ and } H(x) = H(x')\,\right]$$

where the probability is over $(x, x') \xleftarrow{\$} C$. For simplicity and to better reflect practice, we view hash functions as unkeyed. This means there always *exists* an efficient C whose cr-advantage is 1, but that does not mean we can find it, and our results remain meaningful because the proofs give *explicit* constructions of cr-adversaries from other adversaries [57]. We could extend our treatment to let hash functions be families, which would be more rigorous. We can't make the hash key part of the PRF key because then it would be subject to the RKA, but since its secrecy is not needed for security, we can make it a public parameter. Thus, keyed hash functions require an extended syntax for function families in which functions in the family depended on a public parameter, and we have chosen to avoid this.

3 General Construction of RKA-PRFs and RKA-PRPs

In this section we describe and analyze our general RKA-PRF construction. We begin by defining the notions of key-malleability and key fingerprints, on which the general construction is based. Theorem 3.1 states the general construction and proves its security. In subsequent sections we show how to instantiate the general construction to obtain DDH based RKA-PRFs for group-induced classes as well as other

RKA-PRFs.

3.1 Key-Malleability

Suppose $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is a family of functions and $\Phi \subseteq \mathsf{Fun}(\mathcal{K},\mathcal{K})$ is a set of RKD functions. Suppose T is a deterministic algorithm that given an oracle $f\colon \mathcal{D} \to \mathcal{R}$ and inputs $(\phi, x) \in \Phi \times \mathcal{D}$ returns a point $\mathsf{T}^f(\phi, x) \in \mathcal{R}$. We say that T is a *key-transformer* for (M, Φ) if it satisfies two conditions. The first, called *correctness*, asks that $M(\phi(K), x) = \mathsf{T}^{M(K,\cdot)}(\phi, x)$ for every $(\phi, K, x) \in \Phi \times \mathcal{K} \times \mathcal{D}$. This is a relatively straightforward condition saying that one can compute $M(\phi(K), x)$ from ϕ, x if one has an oracle for $M(K, \cdot)$. The second condition, called *uniformity*, is more subtle. Roughly, it says that if the oracle provided to T is random then the outputs of T on any input sequence $(\phi_1, x_1), \ldots, (\phi_q, x_q)$ are uniformly and independently distributed *as long as* $x_1, \ldots, x_q$ *are distinct.* Formally, game $\text{KTReal}_{\mathsf{T}}$ begins by picking $f \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$ and responds to oracle query $\text{KTF}\textsc{n}(\phi, x)$ via $\mathsf{T}^f(\phi, x)$ while game $\text{KTRand}_{\mathsf{T}}$ makes no initializations and responds to oracle query $\text{KTF}\textsc{n}(\phi, x)$ by picking and returning a random point in $\mathcal{R}$. Let us say a Φ-restricted adversary is *unique input* if, in its oracle queries $(\phi_1, x_1), \ldots, (\phi_q, x_q)$, the points $x_1, \ldots, x_q$ are always distinct, where by "always" we mean with probability one regardless of how oracle queries are answered and what are the coins of the adversary. The uniformity requirement is that

$$\Pr\left[\,\text{KTReal}_{\mathsf{T}}^{U} \Rightarrow 1\,\right] \;=\; \Pr\left[\,\text{KTRand}_{\mathsf{T}}^{U} \Rightarrow 1\,\right] \tag{4}$$

for every unique-input Φ-restricted adversary U against the uniformity of T. We say M is Φ-*key-malleable* if there exists an efficient key transformer for (M, Φ).

That key-malleability might be useful to obtain RKA-PRFs is, on the one hand, intuitive, because the correctness property clearly allows us to simulate queries to $M(\phi(K), \cdot)$ via queries to $M(K, \cdot)$. It is, on the other hand, counter-intuitive, because the same correctness property immediately yields an attack showing that M is *not* a Φ-RKA-PRF as long as Φ contains the identity function id and a function ϕ satisfying $\phi(K) \neq K$ for all $K \in \mathcal{K}$, conditions met by any group-induced Φ. Indeed, consider Φ-restricted adversary A that, for some $x \in \mathcal{D}$, makes query $y \leftarrow \text{RKF}\textsc{n}(\phi, x)$. Then it runs T on inputs ϕ, x to get an output z, answering any oracle query w made in this computation by $\text{RKF}\textsc{n}(\text{id}, w)$. It returns 1 if $y = z$ and 0 otherwise. Correctness says that A always returns 1 in game RKPRFReal_M. But the assumption on ϕ implies that A returns 1 with probability at most $1/|\mathcal{R}|$ in game RKPRFRand_M. So $\mathbf{Adv}^{\text{prf-rka}}_{M,\Phi}(A)$ is almost 1.

Although a key-malleable M is not a Φ-RKA-PRF, one can show that it is an RKA-PRF versus unique-input adversaries. (The adversary of the above attack need not be unique-input.) This leaves two questions. The first is how to bridge the gap to arbitrary adversaries, which we do via the concept of key fingerprints discussed below. The second is how to obtain key-malleable PRFs, which we will do later via the Naor-Reingold [52] and Lewko-Waters [47] constructs.

3.2 Key fingerprints

Suppose $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is a family of functions and $\Phi \subseteq \mathsf{Fun}(\mathcal{K},\mathcal{K})$ is a set of RKD functions. Let $\mathbf{w}$ be vector over $\mathcal{D}$ and let $m = |\mathbf{w}|$. We say that $\mathbf{w}$ is a *key fingerprint* for (M, Φ) if

$$(M(\phi(K), \mathbf{w}[1]), \ldots, M(\phi(K), \mathbf{w}[m])) \neq \big(M(\phi'(K), \mathbf{w}[1]), \ldots, M(\phi'(K), \mathbf{w}[m])\big) \tag{5}$$

for all $K \in \mathcal{K}$ and all distinct $\phi, \phi' \in \Phi$.

Let's call a class $\Phi \subseteq \mathsf{Fun}(\mathcal{K},\mathcal{K})$ of RKD functions *claw-free* if $\phi(K) \neq \phi'(K)$ for every key $K \in \mathcal{K}$ and every distinct $\phi, \phi' \in \Phi$ [50, 6]. We note that if (M, Φ) has a key fingerprint then it follows automatically that Φ is claw-free. Indeed, if there is a K and ϕ, ϕ' such that $\phi(K) = \phi'(K)$ then there can be no $\mathbf{w}$ for which Equation (5) is true. We will use this frequently below.

We say that $\mathbf{w}$ is a *strong key fingerprint* for (M, Φ) if

$$(M(K, \mathbf{w}[1]), \ldots, M(K, \mathbf{w}[m])) \neq \big(M(K', \mathbf{w}[1]), \ldots, M(K', \mathbf{w}[m])\big) \tag{6}$$

for all distinct $K, K' \in \mathcal{K}$. If Φ is claw-free then a strong key fingerprint for (M, Φ) is also a key fingerprint for (M, Φ), which we will use in analyzing our constructs. If Φ is complete —recall this means that for every $K, K' \in \mathcal{K}$ there is a $\phi \in \Phi$ such that $\phi(K) = K'$— then any key fingerprint for (M, Φ) is also a strong key fingerprint for (M, Φ). In general, however, the existence of a key fingerprint may not imply the existence of a strong key fingerprint.

3.3 Construction

Let $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a key-malleable family of functions and T a key transformer for (M, Φ). Let $\mathbf{w} \in \mathcal{D}^m$ be a key-fingerprint for (M, Φ). We say that a point $w \in \mathcal{D}$ is a *possible oracle query* for T relative to $(M, \Phi, \mathbf{w})$ if there exists $(f, \phi, i) \in \mathsf{Fun}(\mathcal{D}, \mathcal{R}) \times \Phi \times \{1, \ldots, m\}$ such that the computation $\mathsf{T}^f(\phi, \mathbf{w}[i])$ makes oracle query w. We let $\mathsf{Qrs}(\mathsf{T}, M, \Phi, \mathbf{w})$ be the set of all possible oracle queries w for T relative to $(M, \Phi, \mathbf{w})$. Let $\overline{\mathcal{D}} = \mathcal{D} \times \mathcal{R}^m$. A hash function H with domain $\overline{\mathcal{D}}$ is said to be *compatible* with $(\mathsf{T}, M, \Phi, \mathbf{w})$ if its range is $\mathcal{D} \setminus \mathsf{Qrs}(\mathsf{T}, M, \Phi, \mathbf{w})$. That is, possible oracle queries of T relative to $(M, \Phi, \mathbf{w})$ are not allowed to be outputs of H. With this, we can say what are the ingredients of our construction of a Φ-RKA-PRF: (1) a Φ-key-malleable PRF, meaning a family of functions $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ such that, on the one hand, M is a PRF and, on the other hand, there exists a key transformer T for (M, Φ); (2) a key fingerprint $\mathbf{w}$ for (M, Φ); and (3) a collision-resistant hash function $H\colon \overline{\mathcal{D}} \to \mathcal{D} \setminus \mathsf{Qrs}(\mathsf{T}, M, \Phi, \mathbf{w})$ that is compatible with $(\mathsf{T}, M, \Phi, \mathbf{w})$. We combine them to build $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ that on input K, x computes $\overline{\mathbf{w}} \leftarrow M(K, \mathbf{w})$ —recall that, as per our notational conventions, $M(K, \mathbf{w})$ is the vector whose i-th component is $M(K, \mathbf{w}[i])$ for $1 \leq i \leq m$— and then returns $M(K, H(x, \overline{\mathbf{w}}))$. The following theorem says that F is a Φ-RKA-PRF assuming M is a PRF and H is collision-resistant. No assumptions are made on Φ beyond those implied by the conditions stated here.

Theorem 3.1 *Let $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a family of functions and $\Phi \subseteq \mathsf{Fun}(\mathcal{K}, \mathcal{K})$ a class of RKD functions. Let T be a key-transformer for (M, Φ) making Q_{T} oracle queries, and let $\mathbf{w} \in \mathcal{D}^m$ be a key fingerprint for (M, Φ). Let $\overline{\mathcal{D}} = \mathcal{D} \times \mathcal{R}^m$ and let $H\colon \overline{\mathcal{D}} \to S$ be a hash function that is compatible with $(\mathsf{T}, M, \Phi, \mathbf{w})$, so that $S = \mathcal{D} \setminus \mathsf{Qrs}(\mathsf{T}, M, \Phi, \mathbf{w})$. Define $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ by*

$$F(K, x) \;=\; M(K, H(x, M(K, \mathbf{w}))) \tag{7}$$

for all $K \in \mathcal{K}$ and $x \in \mathcal{D}$. Let A be a Φ-restricted adversary against the prf-rka security of F that makes $Q_A \leq |S|$ oracle queries. Then we can construct an adversary B against the prf-security of M and an adversary C against the cr-security of H such that

$$\mathbf{Adv}^{\mathrm{prf\text{-}rka}}_{\Phi,F}(A) \;\leq\; \mathbf{Adv}^{\mathrm{prf}}_{M}(B) + \mathbf{Adv}^{\mathrm{cr}}_{H}(C)\ . \tag{8}$$

Adversary B makes $(m+1) \cdot Q_{\mathsf{T}} Q_A$ oracle queries, and B and C have the same running time as A. ▌

Proof of Theorem 3.1: We use the game sequence of Figure 1, in the analysis below abbreviating by W_i the event "$G_i^A \Rightarrow 1$". We assume (wlog) that A never repeats an oracle query. Game G_0 simply instantiates game $\mathrm{RKPRFReal}_F$ of the definition of Section 2 with our construction F, so

$$\Pr\left[\mathrm{RKPRFReal}_F^A \Rightarrow 1\right] \;=\; \Pr[W_0]\ . \tag{9}$$

Game G_1, which does not include the boxed code, introduces some book-keeping, keeping track of hash values in a set D and setting a flag bad to true if it ever sees a repeat. The book-keeping does not affect the values returned by RKFn so

$$\Pr[W_1] = \Pr[W_0]\ . \tag{10}$$

Game G_2 adds the boxed code which "corrects" a hash value repetition by picking instead a value that, being drawn from $S \setminus D$, will not repeat any previous one. The addition of this "artificial" step, leading

```
proc Initialize // $G_0$
01 $K \xleftarrow{\$} \mathcal{K}$
proc RKFn$(\phi, x)$ // $G_0$
02 For $i = 1, \ldots, |\mathbf{w}|$ do
03     $\overline{\mathbf{w}}[i] \leftarrow M(\phi(K), \mathbf{w}[i])$
04 $h \leftarrow H(x, \overline{\mathbf{w}})$
05 $y \leftarrow M(\phi(K), h)$
06 Return $y$
```

```
proc Initialize // $G_1, G_2$
11 $K \xleftarrow{\$} \mathcal{K}$ ; $D \leftarrow \emptyset$
proc RKFn$(\phi, x)$ // $G_1$, $\boxed{G_2}$
12 For $i = 1, \ldots, |\mathbf{w}|$ do
13     $\overline{\mathbf{w}}[i] \leftarrow M(\phi(K), \mathbf{w}[i])$
14 $h \leftarrow H(x, \overline{\mathbf{w}})$
15 If $h \in D$ then
16     bad $\leftarrow$ true ; $\boxed{h \xleftarrow{\$} S \setminus D}$
17 $D \leftarrow D \cup \{h\}$
18 $y \leftarrow M(\phi(K), h)$
19 Return $y$
```

```
proc Initialize // $G_3$
31 $D \leftarrow \emptyset$
proc RKFn$(\phi, x)$ // $G_3$
32 For $i = 1, \ldots, |\mathbf{w}|$ do
33     $\overline{\mathbf{w}}[i] \leftarrow \mathsf{T}^{M(K,\cdot)}(\phi, \mathbf{w}[i])$
34 $h \leftarrow H(x, \overline{\mathbf{w}})$
35 If $h \in D$ then $h \xleftarrow{\$} S \setminus D$
36 $D \leftarrow D \cup \{h\}$
37 $y \leftarrow \mathsf{T}^{M(K,\cdot)}(\phi, h)$
38 Return $y$
```

```
proc Initialize // $G_4$
41 $D \leftarrow \emptyset$ ; $f \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$
proc RKFn$(\phi, x)$ // $G_4$
42 For $i = 1, \ldots, |\mathbf{w}|$ do
43     $\overline{\mathbf{w}}[i] \leftarrow \mathsf{T}^{f}(\phi, \mathbf{w}[i])$
44 $h \leftarrow H(x, \overline{\mathbf{w}})$
45 If $h \in D$ then $h \xleftarrow{\$} S \setminus D$
46 $D \leftarrow D \cup \{h\}$
47 $y \leftarrow \mathsf{T}^{f}(\phi, h)$
48 Return $y$
```

```
proc Initialize // $G_5$
51 $D \leftarrow \emptyset$ ; $f, g \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$
proc RKFn$(\phi, x)$ // $G_5$
52 For $i = 1, \ldots, |\mathbf{w}|$ do
53     $\overline{\mathbf{w}}[i] \leftarrow \mathsf{T}^{g}(\phi, \mathbf{w}[i])$
54 $h \leftarrow H(x, \overline{\mathbf{w}})$
55 If $h \in D$ then $h \xleftarrow{\$} S \setminus D$
56 $D \leftarrow D \cup \{h\}$
57 $y \leftarrow \mathsf{T}^{f}(\phi, h)$
58 Return $y$
```

```
proc RKFn$(\phi, x)$ // $G_6$
61 $y \xleftarrow{\$} \mathcal{R}$
62 Return $y$
```

```
proc Initialize // $G_7$
71 $K \xleftarrow{\$} \mathcal{K}$
72 $G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R})$
proc RKFn$(\phi, x)$ // $G_7$
73 $y \leftarrow G(\phi(K), x)$
74 Return $y$
```

Figure 1: Games for the proof of Theorem 3.1. Game G_2 includes the boxed code and game G_1 does not.

to a game different from the "real" one, is to ensure that the values of h on which $\mathsf{T}^f(\phi, h)$ is later called (lines 37,47,57) are distinct, putting us in a position to exploit the uniformity of T and replace the outputs by random values. This, however, is some distance away. For the moment we observe that games G_1, G_2 are identical until bad —differ only in code following the setting of bad to true— and hence the fundamental lemma of game playing [7] implies that

$$\Pr[W_1] \;\leq\; \Pr[W_2] + \Pr[B_1] \tag{11}$$

where B_1 denotes the event that the execution of A with game G_1 sets the flag bad to true. Making crucial use of the assumption that $\mathbf{w}$ is a key fingerprint for (M, Φ), we design adversary C attacking the cr-security of H such that

$$\Pr[B_1] \;\leq\; \mathbf{Adv}^{\mathrm{cr}}_{H}(C)\ . \tag{12}$$

Adversary C begins by picking $K \xleftarrow{\$} \mathcal{K}$ and initializing a counter $j \leftarrow 0$. It then runs A. When the latter makes a RKFn-query (ϕ, x), adversary C responds via

```
For $i = 1, \ldots, |\mathbf{w}|$ do $\overline{\mathbf{w}}[i] \leftarrow M(\phi(K), \mathbf{w}[i])$
$j \leftarrow j + 1$ ; $\phi_j \leftarrow \phi$ ; $x_j \leftarrow x$ ; $\overline{\mathbf{w}}_j \leftarrow \overline{\mathbf{w}}$ ; $h_j \leftarrow H(x, \overline{\mathbf{w}})$ ; $y \leftarrow M(\phi(K), h)$ ; Return $y$
```

When A halts, C searches for a, b satisfying $1 \leq a < b \leq j$ such that $h_a = h_b$ and, if it finds them, outputs $(x_a, \overline{\mathbf{w}}_a), (x_b, \overline{\mathbf{w}}_b)$ and halts. Towards justifying Equation (12) the main question is, why are $(x_a, \overline{\mathbf{w}}_a), (x_b, \overline{\mathbf{w}}_b)$ distinct? The assumption that A never repeats an oracle query means that $(\phi_a, x_a) \neq (\phi_b, x_b)$. Now consider two cases. First, if $\phi_a = \phi_b$ then we must have $x_a \neq x_b$ whence of course

$(x_a, \overline{\mathbf{w}}_a) \neq (x_b, \overline{\mathbf{w}}_b)$. Second, if $\phi_a \neq \phi_b$ then the assumption that $\mathbf{w}$ is a key fingerprint for (M, Φ) means, by Equation (5), that $\overline{\mathbf{w}}_a \neq \overline{\mathbf{w}}_b$ and again $(x_a, \overline{\mathbf{w}}_a) \neq (x_b, \overline{\mathbf{w}}_b)$.

In game G_3, we use the key transformer T, given by the assumed Φ-key-malleability of M, to compute $M(\phi(K), \cdot)$ via oracle calls to $M(K, \cdot)$, both at line 33 and at line 37. The correctness property of the key transformer implies

$$\Pr[W_2] \;=\; \Pr[W_3]\;. \tag{13}$$

Game G_4 replaces the oracle given to T by a random function. We design adversary B attacking the prf-security of M such that

$$\Pr[W_3] - \Pr[W_4] \;\leq\; \mathbf{Adv}^{\mathrm{prf}}_{M}(B)\;. \tag{14}$$

This is possible because the games make only oracle access to $M(K, \cdot)$ and f, respectively. In detail, adversary B runs A. When the latter makes a RKFN-query (ϕ, x), adversary B responds via

```
For $i = 1, \ldots, |\mathbf{w}|$ do $\overline{\mathbf{w}}[i] \leftarrow \mathsf{T}^{\mathrm{FN}}(\phi, \mathbf{w}[i])$ ; $h \leftarrow H(x, \overline{\mathbf{w}})$ ; $y \leftarrow \mathsf{T}^{\mathrm{FN}}(\phi, h)$ ; Return $y$
```

where FN is B's own oracle. When A halts, B halts with the same output. Then

$$\Pr\left[\mathrm{PRFReal}^{B}_{M} \Rightarrow 1\right] = \Pr[W_3] \qquad \text{and} \qquad \Pr\left[\mathrm{PRFRand}^{B}_{M} \Rightarrow 1\right] = \Pr[W_4]$$

so Equation (14) follows from Equation (2).

Rather than return $y = \mathsf{T}^f(\phi, h)$ as at lines 47,48, we would like to pick and return a random y, as at lines 61,62 of game G_6, saying this makes no difference by the uniformity of T. But we have to be careful, because line 47 is not the only place f is used in G_4. Oracle f is also being queried in the computation $\mathsf{T}^f(\phi, \mathbf{w}[i])$ at line 43, and if a f-query made here equals an input h at line 47, then it is unclear we can argue randomness of the line 47 output y based on the uniformity of T. The assumed compatibility of H with $(\mathsf{T}, M, \Phi, \mathbf{w})$ comes to the rescue. It says the queries to f in the computation $\mathsf{T}^f(\phi, \mathbf{w}[i])$ at line 43, which fall within the set $\mathsf{Qrs}(\mathsf{T}, M, \Phi, \mathbf{w})$, are not in the set S that is the range of H. Thus, the calls to f at lines 43 and 47 can be answered with different, independent random functions without affecting the distribution of the procedure output. In other words, considering game G_5, which switches f to g at line 53 but not at line 57, the compatibility of H with $(\mathsf{T}, M, \Phi, \mathbf{w})$ implies that

$$\Pr[W_4] \;=\; \Pr[W_5]\;. \tag{15}$$

We will now exploit the uniformity of T to show that

$$\Pr[W_5] \;=\; \Pr[W_6]\;. \tag{16}$$

To do this we design *unique-input* Φ-restricted adversary U against the uniformity of T such that

$$\Pr\left[\mathrm{KTReal}^{U}_{M} \Rightarrow 1\right] = \Pr[W_5] \qquad \text{and} \qquad \Pr\left[\mathrm{KTRand}^{U}_{M} \Rightarrow 1\right] = \Pr[W_6]\;. \tag{17}$$

Equation (16) follows from Equation (4). Adversary U begins by initializing set $D \leftarrow \emptyset$ and picking $g \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$. (Adversary U of the uniformity condition is not required to be efficient so picking g like this is okay but in any case we could make U efficient if we liked by simulating g via lazy sampling rather than picking it upfront.) It then runs A. When the latter makes a RKFN-query (ϕ, x), adversary U responds via

```
For $i = 1, \ldots, |\mathbf{w}|$ do $\overline{\mathbf{w}}[i] \leftarrow \mathsf{T}^g(\phi, \mathbf{w}[i])$
$j \leftarrow j + 1$ ; $\phi_j \leftarrow \phi$ ; $h_j \leftarrow H(x, \overline{\mathbf{w}})$ ; If $h_j \in D$ then $h_j \xleftarrow{\$} S \setminus D$
$y \leftarrow \mathrm{KTFN}(\phi_j, h_j)$ ; Return $y$
```

where KTFN is U's own oracle. The delicate question is, why is U unique-input? The boxed code introduced at line 16, carried through to line 55, and reflected by the "If" statement in the code for U

above, ensures that $h_1, \ldots, h_j$ are all distinct at the end of U's computation as long as $Q_A \leq |S|$, which the theorem assumed. Equation (17) follows.

The claw-freeness of Φ —recall this follows from the assumption that (M, Φ) has a key fingerprint— implies that if $(\phi, x) \neq (\phi', x')$ then $(\phi(K), x) \neq (\phi'(K), x')$. This together with the assumption that A does not repeat an oracle query imply

$$\Pr[W_6] \ = \ \Pr[W_7] \ = \ \Pr\left[\,\mathrm{RKPRFRand}_F^A \Rightarrow 1\,\right] \ . \tag{18}$$

Equation (8) follows from Equations (9), (10), (11), (12), (13), (14), (15), (16), (18), (3). ▌

4 Multiplicative DDH-based RKA-PRF

We instantiate our general construction to get a DDH-based Φ-RKA-PRF where Φ is group induced. Let $\mathbb{G}$ be a (multiplicatively written) group of prime order p, and let $g \in \mathbb{G}$ be an arbitrary generator of $\mathbb{G}$. The classic Naor-Reingold [52] PRF NR: $\mathbb{Z}_p^{n+1} \times \{0,1\}^n \rightarrow \mathbb{G}$ is defined via

$$\mathrm{NR}(\mathbf{a}, x) \ = \ g^{\mathbf{a}[0] \prod_{i=1}^{n} \mathbf{a}[i]^{x[i]}} \tag{19}$$

for all $\mathbf{a} \in \mathbb{Z}_p^{n+1}$ and $x \in \{0,1\}^n$. Recall the advantage of an adversary B against the DDH problem in $\mathbb{G}$ is

$$\mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(B) \ = \ \Pr\Big[\,B(g^a, g^b, g^{ab}) \Rightarrow 1\,\Big] - \Pr\Big[\,B(g^a, g^b, g^c) \Rightarrow 1\,\Big],$$

where the probabilities are over $a, b, c \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$. The following result of [52] says that NR is a PRF if DDH is hard in $\mathbb{G}$.

Lemma 4.1 [52] *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and* NR: $\mathbb{Z}_p^{n+1} \times \{0,1\}^n \rightarrow \mathbb{G}$ *the family of functions defined via Equation (19). Let A an adversary against the prf-security of* NR *that makes Q oracle queries. Then we can construct an adversary B against the DDH problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}_{\mathrm{NR}}^{\mathrm{prf}}(A) \leq n \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(B) \ . \tag{20}$$

The running time of B is that of A plus the time required for $4 \cdot Q$ exponentiations in $\mathbb{G}$. ▌

We are now ready to instantiate the ingredients of our general construction and obtain our first concrete construction.

Group-induced class. Define operation $*$ by $\mathbf{a} * \mathbf{d} = (\mathbf{a}[0]\mathbf{d}[0], \ldots, \mathbf{a}[n]\mathbf{d}[n])$ where operations on components are multiplications modulo p. Then the set $\mathcal{K} = (\mathbb{Z}_p^*)^{n+1}$ is a group under $*$. Let $\phi_{\mathbf{d}}^*$: $\mathcal{K} \rightarrow \mathcal{K}$ be defined by $\phi_{\mathbf{d}}^*(\mathbf{a}) = \mathbf{a} * \mathbf{d}$ for all $\mathbf{a}, \mathbf{d} \in \mathcal{K}$. Let $\Phi = \mathsf{rkd}[(\mathbb{Z}_p^*)^{n+1}, *]$ be the class of all $\phi_{\mathbf{d}}^*$ as $\mathbf{d}$ ranges over $\mathcal{K}$. This class is group-induced, the group being $(\mathcal{K}, *)$.

Key malleability. We claim that NR is Φ-key-malleable. The key-transformer T, given oracle f: $\{0,1\}^n \rightarrow \mathbb{G}$ and inputs $\phi_{\mathbf{d}}^*, x$, returns $f(x)^{\mathbf{d}[0] \prod_{i=1}^{n} \mathbf{d}[i]^{x[i]}}$. Correctness holds because

$$\mathsf{T}^{\mathrm{NR}(\mathbf{a}, \cdot)}(\phi_{\mathbf{d}}^*, x) \ = \ \mathrm{NR}(\mathbf{a}, x)^{\mathbf{d}[0] \prod_{i=1}^{n} \mathbf{d}[i]^{x[i]}} \ = \ \mathrm{NR}(\mathbf{a} * \mathbf{d}, x) \ .$$

In game $\mathrm{KTReal}_{\mathrm{NR}}$, the responses received by unique-input, Φ-restricted adversary U to KTFn-queries $(\phi_{\mathbf{d}_1}^*, x_1), \ldots, (\phi_{\mathbf{d}_q}^*, x_q)$ are $f(x_1)^{\mathbf{d}_1[0] \prod_{i=1}^{n} \mathbf{d}_1[i]^{x[i]}}, \ldots, f(x_q)^{\mathbf{d}_q[0] \prod_{i=1}^{n} \mathbf{d}_q[i]^{x[i]}}$ where $f \stackrel{\$}{\leftarrow} \mathsf{Fun}(\{0,1\}^n, \mathbb{G})$ was chosen by the game. Since $x_1, \ldots, x_q$ are distinct and the exponents are non-zero, these responses are randomly and independently distributed over $\mathbb{G}$. We have verified the uniformity condition.

Key fingerprint. For $i = 1, \ldots, n$ let $\mathbf{w}[i] = 0^{i-1} \,\|\, 1 \,\|\, 0^{n-i}$ be the string that is all zeros except at position i, where it has a one. Let $\mathbf{w}[0] = 0^n$. We claim that $\mathbf{w}$ is a strong key fingerprint for (NR, Φ). To see this, first note that $(\mathrm{NR}(\mathbf{a}, \mathbf{w}[0]), \mathrm{NR}(\mathbf{a}, \mathbf{w}[1]) \ldots, \mathrm{NR}(\mathbf{a}, \mathbf{w}[n])) = (g^{\mathbf{a}[0]}, g^{\mathbf{a}[0]\mathbf{a}[1]}, \ldots, g^{\mathbf{a}[0]\mathbf{a}[n]})$. Now if $\mathbf{a}, \mathbf{a}' \in \mathcal{K}$ are distinct keys and $\mathbf{a}[0] \neq \mathbf{a}'[0]$ then $g^{\mathbf{a}[0]} \neq g^{\mathbf{a}'[0]}$. On the other hand if $\mathbf{a}[0] = \mathbf{a}'[0]$ and $\mathbf{a}[i] \neq \mathbf{a}'[i]$ for some $i > 0$, then $g^{\mathbf{a}[0]\mathbf{a}[i]} \neq g^{\mathbf{a}'[0]\mathbf{a}'[i]}$. The claim follows from the definition of Equation (6) with $M = \mathrm{NR}$. Since Φ is claw-free, $\mathbf{w}$ is also a key fingerprint for (NR, Φ), satisfying Equation (5) with $M = \mathrm{NR}$.

Compatible hash function. The set of possible oracle queries of T relative to $(\mathrm{NR}, \Phi, \mathbf{w})$ is $\mathsf{Qrs}(\mathsf{T}, \mathrm{NR}, \Phi, \mathbf{w}) = \{\, \mathbf{w}[i] \,:\, 0 \leq i \leq n \,\}$ because on inputs ϕ, x the only oracle query made by T is x itself. Let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^{n+1}$. If $h\colon\ \overline{\mathcal{D}} \to \{0,1\}^{n-2}$ is collision resistant, then $H\colon\ \overline{\mathcal{D}} \to \{0,1\}^n \setminus \mathsf{Qrs}(\mathsf{T}, \mathrm{NR}, \Phi, \mathbf{w})$ defined by $H(x, \mathbf{z}) = 11 \,\|\, h(x, \mathbf{z})$ is collision resistant and compatible with $(\mathsf{T}, \mathrm{NR}, \Phi, \mathbf{w})$ because all members of $\mathsf{Qrs}(\mathsf{T}, \mathrm{NR}, \Phi, \mathbf{w})$ have Hamming weight at most 1 while outputs of H have Hamming weight at least 2.

We have all the ingredients. The following theorem combines the above with Theorem 3.1 and Lemma 4.1 to present our DDH-based Φ-RKA-PRF for group-induced Φ and specify its security.

Theorem 4.2 *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and $\mathrm{NR}\colon\ \mathbb{Z}_p^{n+1} \times \{0,1\}^n \to \mathbb{G}$ the family of functions defined via Equation (19). Let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^{n+1}$ and let $h\colon\ \overline{\mathcal{D}} \to \{0,1\}^{n-2}$ be a hash function. Define $F\colon\ (\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ by*

$$F(\mathbf{a}, x) \;=\; \mathrm{NR}(\mathbf{a}, 11 \,\|\, h(x, (g^{\mathbf{a}[0]}, g^{\mathbf{a}[0]\mathbf{a}[1]}, \ldots, g^{\mathbf{a}[0]\mathbf{a}[n]})))$$

for all $\mathbf{a} \in (\mathbb{Z}_p^*)^{n+1}$ and $x \in \{0,1\}^n$. Let $\Phi = \mathsf{rkd}[(\mathbb{Z}_p^*)^{n+1}, *]$ where $*$ is the operation of component-wise multiplication modulo p. Let A be a Φ-restricted adversary against the prf-rka security of F that makes $Q_A \leq 2^{n-2}$ oracle queries. Then we can construct an adversary B against the DDH problem in $\mathbb{G}$ and an adversary C against the cr-security of h such that

$$\mathbf{Adv}^{\text{prf-rka}}_{\Phi,F}(A) \;\leq\; n \cdot \mathbf{Adv}^{\text{ddh}}_{\mathbb{G}}(B) + \mathbf{Adv}^{\text{cr}}_{h}(C)\ .$$

The running time of B is that of A plus the time required for $4 \cdot Q$ exponentiations in $\mathbb{G}$. C has the same running time as A. ▌

5 DLIN-based RKA-PRF

There are groups where DDH is easy but the DLIN problem still seems hard, which motivated Lewko and Waters [47] to find a DLIN-based PRF. In the same vein, we seek a DLIN-based RKA-PRF.

Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p. Lewko and Waters [47] describe their DLIN-based PRF as having key a randomly chosen tuple $(y_0, z_0, y_1, z_1, w_1, v_1, \ldots, y_n, z_n, w_n, v_n) \in \mathbb{Z}_p^{4n+2}$ and then on input $x \in \{0,1\}^n$ computing its output via

```
$a \leftarrow y_0$ ; $b \leftarrow z_0$
For $i = 1, \ldots, n$ do
    If $x[i] = 1$ then $a \leftarrow a y_i + b z_i$ ; $b \leftarrow a w_i + b v_i$
Return $g^a$
```

Lewko and Waters [47, Section 1] comment that "the additional complexity required to accommodate the weaker assumptions means that our functions can no longer be described by closed-form formulas like . . . ," referring, in the ". . . ," to the formula for NR that we have given as Equation (19). We provide such a closed-form formula based on matrices. (This will put us in a position, via a slight modification of the construction, to apply Theorem 3.1 and obtain a RKA-PRF.) Let $\mathrm{AL}_2(p)$ denote the set of all 2 by 2 matrices over $\mathbb{Z}_p$. If $\mathbf{M} \in \mathrm{AL}_2(p)$ and $b \in \{0,1\}$ then $\mathbf{M}^b$ is the identity matrix if $b = 0$ and is of course just $\mathbf{M}$ if $b = 1$. If $\mathbf{u} = (\mathbf{u}[1], \mathbf{u}[2])$ is a 2-vector over $\mathbb{Z}_p$ then $\mathbf{u} \cdot \mathbf{M}$ denotes the 2-vector obtained by the vector-matrix product in which $\mathbf{u}$ is viewed as a 1 by 2 matrix. We define $\mathrm{LW}\colon\ \mathrm{AL}_2(p)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ via

$$\mathrm{LW}(\mathbf{A}, x) \;=\; g^{\mathbf{y}[1]} \qquad \text{where} \qquad \mathbf{y} \;=\; (1,0) \cdot \mathbf{A}[0] \textstyle\prod_{i=1}^{n} \mathbf{A}[i]^{x[i]} \tag{21}$$

for all $\mathbf{A} \in \mathrm{AL}_2(p)^{n+1}$ and $x \in \{0,1\}^n$. Here the key is an $(n+1)$-vector $\mathbf{A} = (\mathbf{A}[0], \ldots, \mathbf{A}[n])$ of 2 by 2 matrices over $\mathbb{Z}_p$. The formula left-multiplies the matrix product by the 2-vector $(1, 0)$ to get a 2-vector $\mathbf{y}$ whose first component $\mathbf{y}[1]$ becomes the exponent to which g is raised to get the function output. We claim LW is exactly the function described by the code above. (To verify this it helps to recall that matrix multiplication is associative. Strictly speaking the LW key is longer, being $4n + 4$ elements of $\mathbb{Z}_p$, but the second row of $\mathbf{A}[0]$ is effectively unused due to the product with $(1, 0)$ so the effective key is $4n + 2$ points in $\mathbb{Z}_p$, as in the original construct.) Comparing with Equation (19), the closed-form formulation of Equation (21) makes clearer how LW is an analogue of NR. Recall the advantage of an adversary B against the Decision Linear (DLIN) problem [25] is

$$\mathbf{Adv}^{\mathrm{dlin}}_{\mathbb{G}}(B) \;=\; \Pr\left[B(h, k, h^b, k^b, g^{a+b}) \Rightarrow 1\right] - \Pr\left[B(h, k, h^b, k^b, g^c) \Rightarrow 1\right],$$

where the probabilities are over $h, k \xleftarrow{\$} \mathbb{G}$ and $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$. The result of [47] says LW is PRF under DLIN:

Lemma 5.1 [47] *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and let* LW: $\mathrm{AL}_2(p)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ *be defined via Equation (21). Let A an adversary against the prf-security of* LW *that makes Q oracle queries. Then we can construct an adversary B against the DLIN problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{LW}}(A) \leq nQ \cdot \mathbf{Adv}^{\mathrm{dlin}}_{\mathbb{G}}(B)\ . \tag{22}$$

The running time of B is equal to the running time of A. ▌

To obtain a key-malleable PRF admitting a key fingerprint, we need two modifications. (The modifications are in fact to get the key fingerprint, not the key malleability.) First, we restrict the keyspace, drawing the matrices from $\mathrm{GL}_2(p) \subset \mathrm{AL}_2(p)$ rather than $\mathrm{AL}_2(p)$, where $\mathrm{GL}_2(p)$ is the set of invertible matrices in $\mathrm{AL}_2(p)$, usually referred to as the general linear group. Second, if $\mathbf{y}[1] = 0$, we use $\mathbf{y}[2]$, which we will be able to guarantee is not 0 in this case, in its place. In detail, define LW*: $\mathrm{GL}_2(p)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ via

$$\mathrm{LW}^*(\mathbf{A}, x) \;=\; \begin{cases} g^{\mathbf{y}[1]} & \text{if } \mathbf{y}[1] \neq 0 \\ g^{\mathbf{y}[2]} & \text{otherwise} \end{cases} \qquad \text{where} \qquad \mathbf{y} \;=\; (1,0) \cdot \mathbf{A}[0] \textstyle\prod_{i=1}^{n} \mathbf{A}[i]^{x[i]} \tag{23}$$

for all $\mathbf{A} \in \mathrm{GL}_2(p)^{n+1}$ and $x \in \{0,1\}^n$. The modifications are "low-probability" enough to leave unchanged the status of the function as a DLIN-based PRF upto a new term in the bound.

Lemma 5.2 *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and let* LW*: $\mathrm{GL}_2(p)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ *be defined via Equation (23). Let A an adversary against the prf-security of* LW* *that makes Q oracle queries. Then we can construct an adversary B against the DLIN problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{LW}^*}(A) \leq nQ \cdot \mathbf{Adv}^{\mathrm{dlin}}_{\mathbb{G}}(B) + \frac{Q + 2n + 2}{p}\ . \tag{24}$$

The running time of B is equal to the running time of A. ▌

Proof: A random matrix from $\mathrm{AL}_2(p)$ is in $\mathrm{GL}_2(p)$ with probability

$$\frac{(p^2-1)(p^2-p)}{p^4} \;\geq\; 1 - \frac{2}{p}\ .$$

The probability that a random function with range $\mathbb{G}$ returns the identity element of $\mathbb{G}$ across Q queries to it is Q/p. A short game sequence can use these facts in conjunction with Lemma 5.1 to conclude. ▌

We proceed to instantiate our general construction.

THE CLASS Φ. For $\mathbf{d} \in (\mathbb{Z}_p^*)^{n+1}$ let $\phi_{\mathbf{d}}$: $\mathrm{GL}_2(p)^{n+1} \to \mathrm{GL}_2(p)^{n+1}$ be defined by $\phi_{\mathbf{d}}(\mathbf{A}) = (\mathbf{d}[0]\mathbf{A}[0], \ldots, \mathbf{d}[n]\mathbf{A}[n])$. (When a matrix is multiplied by a scalar as here, each entry of the matrix is multiplied by the scalar.) Let $\Phi = \{ \phi_{\mathbf{d}} \ : \ \mathbf{d} \in (\mathbb{Z}_p^*)^{n+1} \}$. Φ is not a group-induced class, and is also not complete. Despite this, Φ is interesting in that the entire key can be modified. Previous work did not provide constructions of RKA-PRFs under standard assumptions for classes with this property. On the other hand, Φ is claw-free. This follows from the restrictions we put on the keyspace.

Key malleability. We claim that LW* is Φ-key-malleable. The key-transformer T, given oracle $f\colon \{0,1\}^n \to \mathbb{G}$ and inputs $\phi_{\mathbf{d}}, x$, returns $f(x)^{\mathbf{d}[0]\prod_{i=1}^{n}\mathbf{d}[i]^{x[i]}}$, just as for NR. Noting that $s = \mathbf{d}[0]\prod_{i=1}^{n}\mathbf{d}[i]^{x[i]} \neq 0$ we have

$$\mathsf{T}^{\mathrm{LW}^*(\mathbf{A},\cdot)}(\phi_{\mathbf{d}}, x) \ = \ \begin{cases} g^{s\mathbf{y}[1]} & \text{if } \mathbf{y}[1] \neq 0 \\ g^{s\mathbf{y}[2]} & \text{otherwise} \end{cases}$$

where $\mathbf{y} = (1,0) \cdot \mathbf{A}[0]\prod_{i=1}^{n}\mathbf{A}[i]^{x[i]}$. But

$$s\mathbf{y} \ = \ \left(\mathbf{d}[0]\prod_{i=1}^{n}\mathbf{d}[i]^{x[i]}\right)\left((1,0)\cdot\mathbf{A}[0]\prod_{i=1}^{n}\mathbf{A}[i]^{x[i]}\right) \ = \ (1,0)\cdot(\mathbf{d}[0]\mathbf{A}[0])\prod_{i=1}^{n}(\mathbf{d}[i]\mathbf{A}[i])^{x[i]} \ ,$$

and correctness follows. Since the formula defining T has not changed, the argument for uniformity is the same as the one we gave for NR.

Key fingerprint. Making crucial use of both the modifications we made to the LW construct, we can show that the same key fingerprint as for NR continues to work for LW*. Namely, for $i = 1, \ldots, n$ let $\mathbf{w}[i] = 0^{i-1} \,\|\, 1 \,\|\, 0^{n-i}$ be the string that is all zeros except at position i, where it has a one. Let $\mathbf{w}[0] = 0^n$. We claim that $\mathbf{w}$ is a key fingerprint for (LW*, Φ). (We do not, as before, claim it is a strong key fingerprint, but this was not necessary for Theorem 3.1.) Indeed, suppose $\mathbf{d}, \mathbf{d}' \in (\mathbb{Z}_p^*)^{n+1}$ are distinct and $\mathbf{A} \in \mathrm{GL}_2(p)^{n+1}$. We consider two cases, the first being that $\mathbf{d}[0] \neq \mathbf{d}'[0]$. Since $\mathbf{A}[0]$ is non-singular, the 2-vector $\mathbf{a}$ comprising its first row is not $(0,0)$. If $\mathbf{a}[1] \neq 0$ then the 2-vectors $\mathbf{d}[0]\mathbf{a}, \mathbf{d}'[0]\mathbf{a}$ have distinct, non-zero first components $\mathbf{d}[0]\mathbf{a}[1], \mathbf{d}'[0]\mathbf{a}[1]$, and thus

$$\mathrm{LW}^*(\phi_{\mathbf{d}}(\mathbf{A}), \mathbf{w}[0]) \ = \ g^{\mathbf{d}[0]\mathbf{a}[1]} \neq g^{\mathbf{d}'[0]\mathbf{a}[1]} \ = \ \mathrm{LW}^*(\phi_{\mathbf{d}'}(\mathbf{A}), \mathbf{w}[0]) \ .$$

On the other hand if the first component of $\mathbf{a}$ is 0 then the second, $\mathbf{a}[2]$, is non-zero, and we have

$$\mathrm{LW}^*(\phi_{\mathbf{d}}(\mathbf{A}), \mathbf{w}[0]) \ = \ g^{\mathbf{d}[0]\mathbf{a}[2]} \neq g^{\mathbf{d}'[0]\mathbf{a}[2]} \ = \ \mathrm{LW}^*(\phi_{\mathbf{d}'}(\mathbf{A}), \mathbf{w}[0]) \ .$$

The second case is that $\mathbf{d}[0] = \mathbf{d}'[0]$ and $\mathbf{d}[i] \neq \mathbf{d}'[i]$ for some $i > 0$. The matrix $\mathbf{M} = \mathbf{A}[0]\mathbf{A}[i]$ is non-singular and hence its first row $\mathbf{a}$ is not $(0,0)$. Let $d = \mathbf{d}[0]\mathbf{d}[i]$ and $d' = \mathbf{d}'[0]\mathbf{d}'[i]$. If $\mathbf{a}[1] \neq 0$ then the 2-vectors $d\mathbf{a}, d'\mathbf{a}$ have distinct, non-zero first components $d\mathbf{a}[1], d'\mathbf{a}[1]$, and thus

$$\mathrm{LW}^*(\phi_{\mathbf{d}}(\mathbf{A}), \mathbf{w}[i]) \ = \ g^{d\mathbf{a}[1]} \neq g^{d\mathbf{a}[1]} \ = \ \mathrm{LW}^*(\phi_{\mathbf{d}'}(\mathbf{A}), \mathbf{w}[i]) \ .$$

On the other hand if the first component of $\mathbf{a}$ is 0 then the second, $\mathbf{a}[2]$, is non-zero, and we have

$$\mathrm{LW}^*(\phi_{\mathbf{d}}(\mathbf{A}), \mathbf{w}[i]) \ = \ g^{d\mathbf{a}[2]} \neq g^{d'\mathbf{a}[2]} \ = \ \mathrm{LW}^*(\phi_{\mathbf{d}'}(\mathbf{A}), \mathbf{w}[i]) \ .$$

Compatible hash function. No changes here. We continue to have $\mathsf{Qrs}(\mathsf{T}, \mathrm{LW}^*, \Phi, \mathbf{w}) = \{ \mathbf{w}[i] \ : \ 0 \leq i \leq n \}$. Let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^{n+1}$. If $h\colon \overline{\mathcal{D}} \to \{0,1\}^{n-2}$ is collision resistant, then $H\colon \overline{\mathcal{D}} \to \{0,1\}^n \setminus \mathsf{Qrs}(\mathsf{T}, \mathrm{LW}^*, \Phi, \mathbf{w})$ defined by $H(x, \mathbf{z}) = 11 \,\|\, h(x, \mathbf{z})$ is collision resistant and compatible with $(\mathsf{T}, \mathrm{LW}^*, \Phi, \mathbf{w})$.

Combining the above with Theorem 3.1 and Lemma 5.2 we obtain a DLIN-based Φ-RKA-PRF:

Theorem 5.3 *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p. Let* LW*: $\mathrm{GL}_2(p)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ *be the family of functions defined via Equation (23). Let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^{n+1}$ and let $h\colon \overline{\mathcal{D}} \to \{0,1\}^{n-2}$ be a hash function. For $i = 1, \ldots, n$ let $\mathbf{w}[i] = 0^{i-1} \,\|\, 1 \,\|\, 0^{n-i}$ and let $\mathbf{w}[0] = 0^n$. Define $F\colon$* $\mathrm{GL}_2(p)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ *by*

$$F(\mathbf{A}, x) \ = \ \mathrm{LW}^*(\mathbf{A}, 11 \,\|\, h(x, \mathrm{LW}^*(\mathbf{A}, \mathbf{w})))$$

for all $\mathbf{A} \in \mathrm{GL}_2(p)^{n+1}$ *and* $x \in \{0,1\}^n$. *Let* $\Phi = \{ \phi_{\mathbf{d}} : \mathbf{d} \in (\mathbb{Z}_p^*)^{n+1} \}$ *where* $\phi_{\mathbf{d}}(\mathbf{A}) = (\mathbf{d}[0]\mathbf{A}, \ldots, \mathbf{d}[n]\mathbf{A}[n])$ *for all* $\mathbf{A} \in \mathrm{GL}_2(p)^{n+1}$. *Let A be a Φ-restricted adversary against the prf-rka security of F that makes $Q_A \leq 2^{n-2}$ oracle queries. Then we can construct an adversary B against the DLIN problem in $\mathbb{G}$ and an adversary C against the cr-security of h such that*

$$\mathbf{Adv}^{\text{prf-rka}}_{\Phi,F}(A) \ \leq \ n \cdot \mathbf{Adv}^{\text{dlin}}_{\mathbb{G}}(B) + \mathbf{Adv}^{\text{cr}}_{h}(C) + \frac{Q+2n+2}{p}$$

where $Q = (n+2) \cdot Q_A$. B has running time equal to that of A plus the time required to compute $\mathcal{O}((n+2) \cdot Q_A)$ exponentiations in $\mathbb{G}$. C has the same running time as A. ▌

6 Additive DDH-based RKA-PRF

The group operation on $(\mathbb{Z}_p^*)^{n+1}$ in our multiplicative DDH-based RKA-PRF is component-wise multiplication modulo p. In this section we show how to instantiate our general construction to give RKA security when the related-key attack can component-wise *add* modulo p instead of multiply. This leads to a more complicated analysis that uses the full power of our definition of key-malleable PRFs.

Our additive construction will use a variant of the Naor-Reingold [52] PRF, denoted $\mathrm{NR}^*\colon \mathbb{Z}_p^n \times \{0,1\}^n \to \mathbb{G}$, which is defined via

$$\mathrm{NR}^*(\mathbf{a}, x) \ = \ g^{\prod_{i=1}^{n} \mathbf{a}[i]^{x[i]}} \tag{25}$$

for all $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n$. We have modified NR in two ways. The first is that we are now working over $\mathbb{Z}_p$ instead of $\mathbb{Z}_p^*$, but this is only to make addition modulo p a group, and it will not affect things because a random element from $\mathbb{Z}_p$ is non-zero with high probability anyway. The second change is that the key no longer includes $\mathbf{a}[0]$, which is accordingly left out of the evaluation of the function. Of course, this means the function is no longer a PRF because $\mathrm{NR}^*(\mathbf{a}, 0^n) = g$ for all $\mathbf{a}$. But if we exclude 0^n from the set of allowed inputs, then one can prove PRF security.

Lemma 6.1 *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and* $\mathrm{NR}^*\colon \mathbb{Z}_p^n \times (\{0,1\}^n \setminus \{0^n\}) \to \mathbb{G}$ *the family of functions defined via Equation (25). Let A an adversary against the prf-security of* NR^* *that makes Q oracle queries. Then we can construct an adversary B against the DDH problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}^{\text{prf}}_{\mathrm{NR}^*}(A) \leq n \cdot \mathbf{Adv}^{\text{ddh}}_{\mathbb{G}}(B) \ . \tag{26}$$

The running time of B is equal to the running time of A, plus the time required to compute $c \cdot Q$ exponentiations in $\mathbb{G}$, where c is a small constant. ▌

The proof is a very simple adaptation of the original security proof for NR, which we give in the appendix. We note that our multiplicative construction can also be modified to use NR* instead of NR, but there we preferred to use the well-known NR function as a “black-box” instead of delving into its proof, which seems necessary here.

We proceed by instantiating the ingredients for our general construction and then applying that analysis.

Group-induced class. We will use component-wise addition. Define operation $+$ on $\mathbb{Z}_p^n$ by $\mathbf{a} + \mathbf{d} = (\mathbf{a}[1] + \mathbf{d}[1], \ldots, \mathbf{a}[n] + \mathbf{d}[n])$ where operations on components are additions modulo p. Then the set $\mathcal{K} = \mathbb{Z}_p^n$ is a group under $+$. Let $\phi^+_{\mathbf{d}}\colon \mathcal{K} \to \mathcal{K}$ be defined by $\phi^+_{\mathbf{d}}(\mathbf{a}) = \mathbf{a} + \mathbf{d}$ for all $\mathbf{a}, \mathbf{d} \in \mathcal{K}$. Let $\Phi = \mathsf{rkd}[\mathbb{Z}_p^n, *]$ be the class of all $\phi^+_{\mathbf{d}}$ as $\mathbf{d}$ ranges over $\mathcal{K}$. This class is group-induced, the group being $(\mathcal{K}, +)$.

Key malleability. We claim that NR* is Φ-key-malleable. For bit-strings x, y, let us introduce the notation $S(x) = \{i : x[i] = 1\}$ and $y \preceq x$ if $S(y) \subseteq S(x)$. The key-transformer T_+, given oracle

$f\colon \{0,1\}^n \to \mathbb{G}$ and inputs $\phi^+_{\mathbf{d}}, x$, returns

$$\mathsf{T}^f_+(\phi^+_{\mathbf{d}}, x) = g^{\prod_{i\in S(x)} \mathbf{d}[i]} \cdot \prod_{y\preceq x, y\neq 0^n} f(y)^{\prod_{j\in S(x)\setminus S(y)} \mathbf{d}[j]}. \tag{27}$$

To verify correctness, observe that $\mathrm{NR}^*(\mathbf{a}, y) \;=\; g^{\prod_{i\in S(y)} \mathbf{a}[i]}$, so

$$\begin{aligned}\mathsf{T}^{\mathrm{NR}^*(\mathbf{a},\cdot)}_+(\phi^+_{\mathbf{d}}, x) \;&=\; g^{\prod_{i\in S(x)} \mathbf{d}[i]} \cdot \prod_{y\preceq x, y\neq 0^n} g^{\prod_{i\in S(y)} \mathbf{a}[i] \prod_{j\in S(x)\setminus S(y)} \mathbf{d}[j]} \;=\; \prod_{R\subseteq S(x)} g^{\prod_{i\in R} \mathbf{a}[i] \prod_{j\in S(x)\setminus R} \mathbf{d}[j]} \\ &=\; g^{\sum_{R\subseteq S(x)} \prod_{i\in R} \mathbf{a}[i] \prod_{j\in S(x)\setminus R} \mathbf{d}[j]} \;=\; g^{\prod_{i\in S(x)} \mathbf{a}[i]+\mathbf{d}[i]} \;=\; \mathrm{NR}^*(\mathbf{a}+\mathbf{d}, x).\end{aligned}$$

Verifying the uniformity condition is more subtle, so we start with a sketch of the proof before giving the details. We need to show that when f is a random function in Equation (27), then $\mathsf{T}^f_+(\cdot,\cdot)$ appears to be a random function to all unique-input adversaries. Our proof will proceed inductively, where we show that all queries with at least $n-i$ bits set to 1 can equivalently be responded to with random samples, assuming that queries with more 1-bits are answered with random samples. The key observation to enable this proof is that T_+, when run on inputs $\phi^+_{\mathbf{d}}, x$, queries its oracle at x exactly once to get $f(x)$, and then multiplies $f(x)$ by constants and other values $f(y)$ for other points $y \preceq x$, $y \neq x$, where all of the y strictly fewer 1 bits than x. This means that we can start the induction with $i = 0$ by observing that T_+ will use the value $f(1^n)$ at most once, and so this value "masks out" everything else when multiplied in the response.

Lemma 6.2 *Let p be prime, $\mathbb{G} = \langle g\rangle$ be a group of prime order p, $\Phi = \mathsf{rkd}[\mathbb{Z}^n_p, +]$, and $\mathrm{NR}^* : \mathbb{Z}^n_p \times (\{0,1\}^n \setminus \{0^n\}) \to \mathbb{G}$ be defined via $\mathrm{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$. Suppose that T_+ is defined via*

$$\mathsf{T}^f_+(\phi^+_{\mathbf{d}}, x) = g^{\prod_{i\in S(x)} \mathbf{d}[i]} \cdot \prod_{y\preceq x, y\neq 0^n} f(y)^{\prod_{j\in S(x)\setminus S(y)} \mathbf{d}[j]}.$$

Then T_+ is a key-transformer for (NR^*, Φ).

Proof: We have already verified the correctness condition, so all that remains to prove is the uniformity condition. We will use the sequence of games in Figure 2. In the description of the games we use $\mathsf{hw}(x)$ to denote the Hamming weight of a bit-string x.

In G_0 the "If" test in KTFN will never pass, as $\mathsf{hw}(x) \leq n$ for all bit strings of length n. This gives

$$\Pr\left[\, G^A_0 \Rightarrow 1 \,\right] = \Pr\left[\, \mathrm{KTReal}^A \Rightarrow 1 \,\right]. \tag{28}$$

We claim that for all $0 \leq i \leq n-1$,

$$\Pr\left[\, G^A_i \Rightarrow 1 \,\right] = \Pr\left[\, G^A_{i+1} \Rightarrow 1 \,\right]. \tag{29}$$

The only difference between G_i and G_{i+1} is in the way bit-strings x of Hamming weight $n-i-1$ are handled. In G_i such a string is fed to $\mathsf{T}^f_+(\phi^+_{\mathbf{d}}, x)$, which computes

$$\mathsf{T}^f_+(\phi^+_{\mathbf{d}}, x) = g^{\prod_{i\in S(x)} \mathbf{d}[i]} \cdot f(x) \cdot \prod_{y\preceq x, y\neq 0^n, y\neq x} f(y)^{\prod_{j\in S(y)\setminus S(x)} \mathbf{d}[i]}. \tag{30}$$

This is the *only* time that G_i will query $f(x)$ because A is a unique-input adversary and all other queries to f will be at other points with this Hamming weight or at points with strictly less Hamming weight. Since this is the only time $f(x)$ is used, the entire value computed in Equation (30) can equivalently be set to an independent random value, which is accomplished by instead querying $g(x)$. But this is what is done with x in G_{i+1}, and the claim follows.

Finally, we have

$$\Pr\left[\, G^A_n \Rightarrow 1 \,\right] = \Pr\left[\, \mathrm{KTRand}^A \Rightarrow 1 \,\right] \tag{31}$$

```
$\underline{\text{proc Initialize}}$  // $G_i, i = 0, \ldots, n$
$f, g \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$
$\underline{\text{proc KTFn}(\phi^+_{\mathbf{d}}, x)}$  // $G_i, i = 0, \ldots, n$
If $\mathsf{hw}(x) > n - i$ then $y \leftarrow g(x)$
Else $y \leftarrow \mathsf{T}^f_+(\phi^+_{\mathbf{d}}, x)$
Return $y$
```

Figure 2: Games for the proof of Lemma 6.2.

because the "If" statement will always pass, as every bit-string except 0^n satisfies $\mathsf{hw}(x) > 0$, and the adversary is not allowed queries with $x = 0^n$. The theorem is proved by collecting Equations (28), (29), and (31) to get

$$\Pr\left[\,\text{KTReal}^A \Rightarrow 1\,\right] = \Pr\left[\,G_0^A \Rightarrow 1\,\right] = \cdots = \Pr\left[\,G_n^A \Rightarrow 1\,\right] = \Pr\left[\,\text{KTRand}^A \Rightarrow 1\,\right].$$

▮

Key fingerprint. Let $\mathbf{w}^*$ be the key fingerprint $\mathbf{w}$ from the multiplicative construction, except with $\mathbf{w}[0]$ omitted. We claim that $\mathbf{w}^*$ is a strong key-fingerprint for (NR^*, Φ). We have $(\text{NR}^*(\mathbf{a}, \mathbf{w}[1]), \ldots, \text{NR}(\mathbf{a}, \mathbf{w}[n])) = (g^{\mathbf{a}[1]}, \ldots, g^{\mathbf{a}[n]})$, and if $\mathbf{a}, \mathbf{a}' \in \mathcal{K}$ are distinct keys, then $\mathbf{a}[i] \neq \mathbf{a}'[i]$ for some i, so $g^{\mathbf{a}[i]} \neq g^{\mathbf{a}'[i]}$. The claim then follows from the definition of Equation (6) with $M = \text{NR}^*$. Since Φ is claw-free, $\mathbf{w}^*$ is also a key fingerprint for (NR^*, Φ), satisfying Equation (5) with $M = \text{NR}^*$.

Compatible hash function. The set of possible oracle queries of T_+ relative to $(\text{NR}^*, \Phi, \mathbf{w}^*)$ is $\mathsf{Qrs}(\mathsf{T}_+, \text{NR}^*, \Phi, \mathbf{w}^*) = \{\, \mathbf{w}[i] \;:\; 1 \le i \le n \,\}$. We can use the same hash function that was used in the multiplicative construction. Namely, let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^n$ and suppose $h\colon\ \overline{\mathcal{D}} \to \{0,1\}^{n-2}$ is collision resistant, then $H\colon\ \overline{\mathcal{D}} \to \{0,1\}^n \setminus \mathsf{Qrs}(\mathsf{T}_+, \text{NR}^*, \Phi, \mathbf{w}^*)$ defined by $H(x, \mathbf{z}) = 11 \,\|\, h(x, \mathbf{z})$ is collision resistant and compatible with $(\mathsf{T}_+, M, \Phi, \mathbf{w}^*)$ because all members of $\mathsf{Qrs}(\mathsf{T}_+, \text{NR}^*, \Phi, \mathbf{w}^*)$ have at most one 1 bit while outputs of H have at least two 1 bits, which also means that the output of H is never 0^n, so its output is always in the domain of NR^*.

We have all the ingredients for our additive construction. The following theorem combines the ingredients above with Theorem 3.1 and Lemma 4.1 to give our additive DDH-based Φ-RKA-PRF and prove its security.

Theorem 6.3 *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and $\text{NR}^*\colon\ \mathbb{Z}_p^n \times \{0,1\}^n \to \mathbb{G}$ the family of functions defined via Equation (25). Let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^n$ and let $h\colon\ \overline{\mathcal{D}} \to \{0,1\}^{n-2}$ be a hash function. For $i = 1, \ldots, n$ let $\mathbf{w}^*[i] = 0^{i-1} \,\|\, 1 \,\|\, 0^{n-i}$. Define $F\colon\ \mathbb{Z}_p^n \times \{0,1\}^n \to \mathbb{G}$ by*

$$F(\mathbf{a}, x) \;=\; \text{NR}^*(\mathbf{a}, 11 \,\|\, h(x, \text{NR}^*(\mathbf{a}, \mathbf{w}^*)))$$

for all $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n$. Let $\Phi = \mathsf{rkd}[\mathbb{Z}_p^{n+1}, +]$ where $+$ is the operation of component-wise multiplication modulo p. Let A be a Φ-restricted adversary against the prf-rka security of F that makes Q_A oracle queries. Then we can construct an adversary B against the DDH problem in $\mathbb{G}$ and an adversary C against the cr-security of h such that

$$\mathbf{Adv}^{\text{prf-rka}}_{\Phi,F}(A) \;\le\; n \cdot \mathbf{Adv}^{\text{ddh}}_{\mathbb{G}}(B) + \mathbf{Adv}^{\text{cr}}_{h}(C).$$

The running time of B is equal to the running time of A, plus the time required to compute $4 \cdot 2^n \cdot Q_A$ exponentiations in $\mathbb{G}$. C has the same running time as A. ▮

We remark that this concrete security statement is still meaningful even though B runs in time exponential in n. This is because n is independent of p, so we are free to adjust p to sizes where it is still plausible to assume an adversary like B has small DDH advantage. This is similar to the situation for an IBE scheme of Boneh and Boyen [23].

7 From RKA-PRFs to RKA-PRPs

Cryptanalytic interest has mostly been in RKA-secure blockciphers, meaning, families of permutations. It is not clear how one might directly modify the constructions of Section 3, which are families of functions, to make them families of permutations. We use, instead, a simple but powerful composition approach that produces a Φ-RKA-PRP from a given Φ-RKA-PRG and an ordinary PRP. We obtain appropriate Φ-RKA-PRGs by combining our Φ-RKA-PRFs with deterministic extractors [29, 33, 27]. This approach not only yields RKA-secure PRPs under chosen-plaintext attack (CPA) but even under chosen-ciphertext attack (CCA).

Some definitions. A family of functions $E\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is a family of permutations, or blockcipher, if $\mathcal{R} = \mathcal{D}$ and $E_K\colon \mathcal{D} \to \mathcal{D}$ is a permutation for all $K \in \mathcal{K}$, in which case $E^{-1}\colon \mathcal{K} \times \mathcal{D} \to \mathcal{D}$ denotes its inverse. Let $\mathsf{PF}(\mathcal{K}, \mathcal{D}) \subseteq \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{D})$ be the set of all families of permutations $E\colon \mathcal{K} \times \mathcal{D} \to \mathcal{D}$ and let $\mathsf{Perm}(X) \subseteq \mathsf{Fun}(X, X)$ be the set of all permutations on X. The advantage of an adversary A in attacking the (standard) prp-security of a family of permutations $E\colon \mathcal{K} \times \mathcal{D} \to \mathcal{D}$ is defined via

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(A) \;=\; \Pr\left[\,\mathrm{PRPReal}^{A}_{E} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{PRPRand}^{A}_{E} \Rightarrow 1\,\right] . \tag{32}$$

Game $\mathrm{PRPReal}_F$ begins by picking $K \stackrel{\$}{\leftarrow} \mathcal{K}$. It responds to oracle query $\textsc{Fn}(x)$ via $E(K, x)$ and to oracle query $\textsc{Fn}^{-1}(y)$ via $E^{-1}(K, y)$. Game $\mathrm{PRFRand}_F$ begins by picking $f \stackrel{\$}{\leftarrow} \mathsf{Perm}(\mathcal{D})$. It responds to oracle query $\textsc{Fn}(x)$ via $f(x)$ and to oracle query $\textsc{Fn}^{-1}(y)$ via $f^{-1}(y)$. The oracles are referred to as the encryption and decryption oracles, respectively, and while the definition captures cca attacks, we can recover the cpa case by considering adversaries that make no decryption queries. The advantage of a Φ-restricted adversary A in attacking the prp-rka security of a family of permutations $\overline{E}\colon \overline{\mathcal{K}} \times \mathcal{D} \to \mathcal{D}$ is defined via

$$\mathbf{Adv}^{\mathrm{prp\text{-}rka}}_{\Phi,\overline{E}}(A) \;=\; \Pr\left[\,\mathrm{RKPRPReal}^{A}_{\overline{E}} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{RKPRPRand}^{A}_{\overline{E}} \Rightarrow 1\,\right] . \tag{33}$$

Game $\mathrm{RKPRPReal}_{\overline{E}}$ begins by picking $\overline{K} \stackrel{\$}{\leftarrow} \overline{\mathcal{K}}$. It responds to oracle query $\mathrm{RK}\textsc{Fn}(\phi, x)$ via $\overline{E}(\phi(\overline{K}), x)$ and to oracle query $\mathrm{RK}\textsc{Fn}^{-1}(\phi, y)$ via $\overline{E}^{-1}(\phi(\overline{K}), y)$. Game $\mathrm{RKPRPRand}_{\overline{E}}$ begins by picking $\overline{K} \stackrel{\$}{\leftarrow} \overline{\mathcal{K}}$ and $G \stackrel{\$}{\leftarrow} \mathsf{PF}(\overline{\mathcal{K}}, \mathcal{D})$. It responds to oracle query $\mathrm{RK}\textsc{Fn}(\phi, x)$ via $G(\phi(\overline{K}), x)$ and to oracle query $\mathrm{RK}\textsc{Fn}^{-1}(\phi, y)$ via $G^{-1}(\phi(\overline{K}), y)$. Again, the oracles are referred to as the encryption and decryption oracles, respectively, and while the definition captures cca attacks, we can recover the cpa case by considering adversaries that make no decryption queries.

Our constructs rely on RKA-PRGs. Let $S\colon \overline{\mathcal{K}} \to \mathcal{K}$. An adversary against S is said to be Φ-restricted if its oracle queries are functions $\phi \in \Phi$. The advantage of such an adversary A in attacking the prg-rka security of S is defined via

$$\mathbf{Adv}^{\mathrm{prg\text{-}rka}}_{\Phi,S}(A) \;=\; \Pr\left[\,\mathrm{RKGReal}^{A}_{S} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{RKGRand}^{A}_{S} \Rightarrow 1\,\right] . \tag{34}$$

Game $\mathrm{RKGReal}_S$ begins by picking $\overline{K} \stackrel{\$}{\leftarrow} \overline{\mathcal{K}}$. It responds to oracle query $\mathrm{G}\textsc{Fn}(\phi)$ via $S(\phi(\overline{\mathcal{K}}))$. Game $\mathrm{RKGRand}_S$ begins by picking $\overline{K} \stackrel{\$}{\leftarrow} \overline{\mathcal{K}}$ and $R \stackrel{\$}{\leftarrow} \mathsf{Fun}(\overline{\mathcal{K}}, \mathcal{K})$. It responds to oracle query $\mathrm{G}\textsc{Fn}(\phi, x)$ via $R(\phi(\overline{K}))$.

Construction. Let $E\colon \mathcal{K} \times \mathcal{D} \to \mathcal{D}$ be a PRP and $S\colon \overline{\mathcal{K}} \to \mathcal{K}$ a Φ-RKA-PRG. Theorem 7.1 says that $\overline{E}\colon \overline{\mathcal{K}} \times \mathcal{D} \to \mathcal{R}$ defined by $\overline{E}(\overline{K}, x) = E(S(\overline{K}), x)$ is a Φ-RKA-PRP. This holds for any Φ, the only (mild) restriction being that it is claw-free.

Theorem 7.1 *Let $E\colon \mathcal{K} \times \mathcal{D} \to \mathcal{D}$ be family of permutations and let $S\colon \overline{\mathcal{K}} \to \mathcal{K}$. Define $\overline{E}\colon \overline{\mathcal{K}} \times \mathcal{D} \to \mathcal{R}$ by $\overline{E}(\overline{K}, x) = E(S(\overline{K}), x)$. Let $\Phi \subseteq \mathsf{Fun}(\overline{\mathcal{K}}, \overline{\mathcal{K}})$ be any claw-free class of RKD functions. Let A be a*

Φ-restricted adversary attacking the prp-rka security of $\overline{E}$. Assume it makes q_e encryption and q_d decryption queries, with the number of different RKD functions involved in its oracle queries being $q \le q_e + q_d$. Then we can construct a Φ-restricted adversary A_1 attacking the prg-security of S and an adversary A_2 attacking the prp-security of E such that

$$\mathbf{Adv}^{\text{prp-rka}}_{\Phi,\overline{E}}(A) \;\le\; \mathbf{Adv}^{\text{prg-rka}}_{\Phi,S}(A_1) + q \cdot \mathbf{Adv}^{\text{prp}}_{E}(A_2) + \frac{q^2}{|\mathcal{K}|} \,.$$

Adversaries A_1, A_2 have the same running time as A. Adversary A_1 makes $q_e + q_d$ oracle queries and A_2 makes q_e encryption queries and q_d decryption queries.

Proof: Game G_0 begins by picking $\overline{K} \xleftarrow{\$} \mathcal{K}$. It responds to RKFN-query (ϕ, x) with $E(S(\phi(\overline{K})), x)$ and to RKFN^{-1}-query (ϕ, y) with $E^{-1}(S(\phi(\overline{K})), y)$. Game G_1, having initially picked $\overline{K} \xleftarrow{\$} \mathcal{K}$ and $R \xleftarrow{\$} \mathsf{Fun}(\mathcal{K}, \mathcal{K})$, responds to RKFN-query (ϕ, x) with $E(R(\phi(\overline{K})), x)$ and to RKFN^{-1}-query (ϕ, y) with $E^{-1}(R(\phi(\overline{K})), y)$. Game G_2, having initially picked $\overline{K} \xleftarrow{\$} \mathcal{K}$, $R \xleftarrow{\$} \mathsf{Fun}(\mathcal{K}, \mathcal{K})$ and $H \xleftarrow{\$} \mathsf{PF}(\mathcal{K}, \mathcal{D})$, responds to RKFN-query (ϕ, x) with $H(R(\phi(\overline{K})), x)$ and to RKFN^{-1}-query (ϕ, y) with $H^{-1}(R(\phi(\overline{K})), y)$. Game G_3, having initially picked $\overline{K} \xleftarrow{\$} \mathcal{K}$ and $G \xleftarrow{\$} \mathsf{PF}(\overline{\mathcal{K}}, \mathcal{D})$, responds to RKFN-query (ϕ, x) with $G(\phi(\overline{K}), x)$ and to RKFN^{-1}-query (ϕ, y) with $G^{-1}(\phi(\overline{K}), y)$. Letting W_i abbreviate the event "$G_i^A \Rightarrow 1$," we have

$$\begin{aligned}\mathbf{Adv}^{\text{prp-rka}}_{\Phi,\overline{E}}(A) &= \Pr[W_0] - \Pr[W_3] \\ &= (\Pr[W_0] - \Pr[W_1]) + (\Pr[W_1] - \Pr[W_2]) + (\Pr[W_2] - \Pr[W_3]) \,.\end{aligned}$$

A straightforward simulation allows us to construct A_1 such that

$$\Pr[W_0] - \Pr[W_1] \;\le\; \mathbf{Adv}^{\text{prg-rka}}_{\Phi,S}(A_1) \,.$$

Also,

$$\Pr[W_2] - \Pr[W_3] \;\le\; \frac{q^2}{|\mathcal{K}|} \,.$$

Finally we use a hybrid argument to construct A_2 such that

$$\Pr[W_1] - \Pr[W_2] \;\le\; q \cdot \mathbf{Adv}^{\text{prp}}_{E}(A_2) \,.$$

It is this last step that relies on the assumption that Φ is claw-free. We omit the details. ▌

Any of our Φ-RKA-PRFs —let's take the multiplicative DDH-based F: $(\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ for concreteness— can of course easily be turned into a Φ-RKA-PRG, in this case S': $(\mathbb{Z}_p^*)^{n+1} \to \mathbb{G}$ defined by $S'(\overline{K}) = F(\overline{K}, C)$ for some fixed, public constant $C \in \{0,1\}^n$. (This function is not length-increasing as with a normal PRG, but that doesn't matter here.) And, of course PRPs E exist aplenty. But we are not done. We would like to let S' play the role of S in Theorem 7.1. The problem is that we need E, S which "match up" in the sense that the keyspace of E is the range of S. But the range of F, and hence of S' in this construction, is a group. Getting a PRP whose keyspace is this group isn't immediate. The best choices for PRPs have keyspace a set $\{0,1\}^r$ of bitstrings. (For example [49, 51], one-way function based PRPs via [37, 36, 49], or even blockciphers like AES.) To get a Φ-RKA-PRG with range $\{0,1\}^r$, we can apply an extractor to the output of S'. But randomized extractors (LHL) are ruled out because the key has nowhere to go other than in the key of the RKA-PRG and would then become subject to the RKA. Luckily, there exist deterministic extractors Ext: $\mathbb{G} \to \{0,1\}^r$ for certain groups $\mathbb{G}$ and we can let $S(\overline{K}) = \mathrm{Ext}(S'(\overline{K}))$. Now to permit extraction of suitable quality we need to choose appropriate parameters. Let p, q be primes such that $q = 2p+1$ has bitlength $2r+4$, let $\mathbb{G}$ be an order p subgroup of $\mathbb{Z}_q^*$, and let $\mathrm{Ext}(h)$ return the r least significant bits of the integer $h \in \mathbb{G}$. Then [29, Theorem 8] says that the statistical distance between the distribution $\mathrm{Ext}(h)$, when h is drawn at random from $\mathbb{G}$, and the uniform distribution on $\{0,1\}^r$, is at most $5 \cdot 2^{-r/2}$.

The following lemma summarizes how we can get a Φ-RKA-PRG with bit outputs from one whose range is an appropriate group. Let $\mathrm{bin}(h)$ denote the binary representation of an integer h. Combining this with Theorem 7.1 and our RKA-PRFs constructions we get RKA-PRPs.

Lemma 7.2 *Let p, q be primes such that $q = 2p + 1$ has bitlength $2r + 4$, let $\mathbb{G}$ be an order p subgroup of $\mathbb{Z}_q^*$, and let $\mathrm{Ext}(h)$ return the r least significant bits of $\mathrm{bin}(h)$ for $h \in \mathbb{G}$. Let S': $\overline{\mathcal{K}} \to \mathbb{G}$. Define S: $\overline{\mathcal{K}} \to \{0,1\}^r$ by $S(\overline{K}) = \mathrm{Ext}(S(\overline{K}))$. Let $\Phi \subseteq \mathsf{Fun}(\overline{\mathcal{K}}, \overline{\mathcal{K}})$ be any claw-free class of RKD functions. Let A be a Φ-restricted adversary attacking the prg-rka security of S. Assume it makes q oracle queries. Then we can construct a Φ-restricted adversary A_1 attacking the prg-security of S' such that*

$$\mathbf{Adv}^{\text{prg-rka}}_{\Phi,S}(A) \;\leq\; \mathbf{Adv}^{\text{prg-rka}}_{\Phi,S'}(A_1) + \frac{5q}{2^{r/2}} \,.$$

Adversaries A_1 has the same running time as A and makes the same number of oracle queries as A.

Acknowledgments

We thank Xavier Boyen and Damien Vergnaud for explaining that the q-BDHI-based proof of PRF-security of BBDY from [30, 24] only works when the domain of the function is small, rendering abortive our attempts to obtain RKA-security via this function. We thank Mira Belenkiy for suggesting that BBDY might help for RKA-security in the first place. We thank Mira and Tolga Acar for discussions on the practical relevance of RKAs that rekindled the first author's interest in this area.

References

[1] B. Applebaum. Fast cryptographic primitives based on the hardness of decoding random linear code. Technical Report TR-845-08, Princeton University, 2008. (Cited on page 2, 3.)

[2] M. Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 602–619. Springer, Aug. 2006. (Cited on page 4.)

[3] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, May 2000. (Cited on page 22.)

[4] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, Aug. 2007. (Cited on page 3.)

[5] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Aug. 1996. (Cited on page 4.)

[6] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer, May 2003. (Cited on page 1, 2, 4, 5, 6.)

[7] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. (Cited on page 5, 8.)

[8] E. Biham. New types of cryptoanalytic attacks using related keys (extended abstract). In T. Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 398–409. Springer, May 1993. (Cited on page 1, 4.)

[9] E. Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994. (Cited on page 1, 4.)

[10] E. Biham, O. Dunkelman, and N. Keller. Related-key boomerang and rectangle attacks. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 507–525. Springer, May 2005. (Cited on page 1, 4.)

[11] E. Biham, O. Dunkelman, and N. Keller. A related-key rectangle attack on the full KASUMI. In B. K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 443–461. Springer, Dec. 2005. (Cited on page 1, 4.)

[12] E. Biham, O. Dunkelman, and N. Keller. Related-key impossible differential attacks on 8-round AES-192. In D. Pointcheval, editor, *CT-RSA 2006*, volume 3860 of *LNCS*, pages 21–33. Springer, Feb. 2006. (Cited on page 1, 4.)

[13] E. Biham, O. Dunkelman, and N. Keller. A simple related-key attack on the full SHACAL-1. In M. Abe, editor, *CT-RSA 2007*, volume 4377 of *LNCS*, pages 20–30. Springer, Feb. 2007. (Cited on page 1, 4.)

[14] E. Biham, O. Dunkelman, and N. Keller. A unified approach to related-key attacks. In K. Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 73–96. Springer, Feb. 2008. (Cited on page 1, 4.)

[15] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In B. S. K. Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 513–525. Springer, Aug. 1997. (Cited on page 4.)

[16] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir. Key recovery attacks of practical complexity on aes variants with up to 10 rounds. In *EUROCRYPT 2010*, May 2010. To appear. (Cited on page 1, 4.)

[17] A. Biryukov and D. Khovratovich. Related-key cryptanalysis of the full aes-192 and aes-256. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, Dec. 2009. (Cited on page 1, 4.)

[18] A. Biryukov, D. Khovratovich, and I. Nikolic. Distinguisher and related-key attack on the full AES-256. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 231–249. Springer, Aug. 2009. (Cited on page 1, 4.)

[19] J. Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In M. J. B. Robshaw, editor, *FSE 2006*, volume 4047 of *LNCS*, pages 328–340. Springer, Mar. 2006. (Cited on page 2.)

[20] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Aug. 2003. (Cited on page 4.)

[21] M. Blunden and A. Escott. Related key attacks on reduced round KASUMI. In M. Matsui, editor, *FSE 2001*, volume 2355 of *LNCS*, pages 277–285. Springer, Apr. 2001. (Cited on page 1, 4.)

[22] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, Aug. 2008. (Cited on page 3.)

[23] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, May 2004. (Cited on page 3, 17.)

[24] D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, May 2004. (Cited on page 3, 19.)

[25] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Aug. 2004. (Cited on page 3, 12.)

[26] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 37–51. Springer, May 1997. (Cited on page 4.)

[27] R. Canetti, J. B. Friedlander, S. V. Konyagin, M. Larsen, D. Lieman, and I. Shparlinski. On the statistical properties of Diffie-Hellman distributions. *Israel J Math.*, 120:23–46, 2000. (Cited on page 3, 17.)

[28] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998. (Cited on page 2.)

[29] C. Chevalier, P.-A. Fouque, D. Pointcheval, and S. Zimmer. Optimal randomness extraction from a Diffie-Hellman element. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 572–589. Springer, Apr. 2009. (Cited on page 3, 17, 18.)

[30] Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In S. Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 416–431. Springer, Jan. 2005. (Cited on page 3, 19.)

[31] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000. (Cited on page 3.)

[32] O. Dunkelman, N. Keller, and J. Kim. Related-key rectangle attack on the full SHACAL-1. In E. Biham and A. M. Youssef, editors, *SAC 2006*, volume 4356 of *LNCS*, pages 28–44. Springer, Aug. 2006. (Cited on page 1, 4.)

[33] P.-A. Fouque, D. Pointcheval, J. Stern, and S. Zimmer. Hardness of distinguishing the MSB or LSB of secret keys in Diffie-Hellman schemes. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 240–251. Springer, July 2006. (Cited on page 3, 17.)

[34] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 258–277. Springer, Feb. 2004. (Cited on page 4.)

[35] D. Goldenberg and M. Liskov. On related-secret pseudorandomness. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 255–272. Springer, Feb. 2010. (Cited on page 1, 2, 3, 4.)

[36] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986. (Cited on page 1, 18.)

[37] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. (Cited on page 18.)

[38] S. Hong, J. Kim, S. Lee, and B. Preneel. Related-key rectangle attacks on reduced versions of SHACAL-1 and AES-192. In H. Gilbert and H. Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, pages 368–383. Springer, Feb. 2005. (Cited on page 1, 4.)

[39] T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 427–445. Springer, Feb. 2004. (Cited on page 4.)

[40] G. Jakimoski and Y. Desmedt. Related-key differential cryptanalysis of 192-bit key AES variants. In M. Matsui and R. J. Zuccherato, editors, *SAC 2003*, volume 3006 of *LNCS*, pages 208–221. Springer, Aug. 2004. (Cited on page 1, 4.)

[41] É. Jaulmes, A. Joux, and F. Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. In J. Daemen and V. Rijmen, editors, *FSE 2002*, volume 2365 of *LNCS*, pages 237–251. Springer, Feb. 2002. (Cited on page 4.)

[42] J. Kelsey, B. Schneier, and D. Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Y. Han, T. Okamoto, and S. Qing, editors, *ICICS 97*, volume 1334 of *LNCS*, pages 233–246. Springer, Nov. 1997. (Cited on page 1, 4.)

[43] J. Kim, S. Hong, and B. Preneel. Related-key rectangle attacks on reduced AES-192 and AES-256. In A. Biryukov, editor, *FSE 2007*, volume 4593 of *LNCS*, pages 225–241. Springer, Mar. 2007. (Cited on page 1, 4.)

[44] L. R. Knudsen. Cryptanalysis of LOKI91. In J. Seberry and Y. Zheng, editors, *AUSCRYPT'92*, volume 718 of *LNCS*, pages 196–208. Springer, Dec. 1992. (Cited on page 1, 4.)

[45] L. R. Knudsen and T. Kohno. Analysis of RMAC. In T. Johansson, editor, *FSE 2003*, volume 2887 of *LNCS*, pages 182–191. Springer, Feb. 2003. (Cited on page 4.)

[46] Y. Ko, S. Hong, W. Lee, S. Lee, and J.-S. Kang. Related key differential attacks on 27 rounds of XTEA and full-round GOST. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 299–316. Springer, Feb. 2004. (Cited on page 1, 4.)

[47] A. B. Lewko and B. Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *ACM CCS 09*, pages 112–120. ACM Press, Nov. 2009. (Cited on page 2, 3, 6, 11, 12.)

[48] M. Liskov, R. L. Rivest, and D. Wagner. Tweakable block ciphers. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46. Springer, Aug. 2002. (Cited on page 4.)

[49] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988. (Cited on page 1, 18.)

[50] S. Lucks. Ciphers secure against related-key attacks. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 359–370. Springer, Feb. 2004. (Cited on page 1, 2, 3, 6.)

[51] M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999. (Cited on page 18.)

[52] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004. (Cited on page 2, 6, 10, 14, 22.)

[53] 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. (Cited on page 3.)

[54] R. C.-W. Phan. Related-key attacks on triple-DES and DESX variants. In T. Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 15–24. Springer, Feb. 2004. (Cited on page 1, 4.)

[55] B. Preneel, R. Govaerts, and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. In D. R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 368–378. Springer, Aug. 1994. (Cited on page 4.)

[56] P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Dec. 2004. (Cited on page 4.)

[57] P. Rogaway. Formalizing human ignorance. In P. Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06*, volume 4341 of *LNCS*, pages 211–228. Springer, Sept. 2006. (Cited on page 5.)

[58] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, May 1997. (Cited on page 3.)

[59] W. Zhang, W. Wu, L. Zhang, and D. Feng. Improved related-key impossible differential attacks on reduced-round AES-192. In E. Biham and A. M. Youssef, editors, *SAC 2006*, volume 4356 of *LNCS*, pages 15–27. Springer, Aug. 2006. (Cited on page 1, 4.)

A Analysis of NR^*

We define the function $\mathrm{NR}^* : \mathbb{Z}_p^n \times \{0,1\}^n \to \mathbb{G}$ by the formula $\mathrm{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$. Below we show that if we disallow queries with $x = 0^n$, then one can prove PRF security. This fact does not seem to be a special case of the original security theorem for NR, so we provide a proof which is a small modification of the original analysis.

In the proof, we will use the following result from [3], which formalizes a type of random self-reducibility property of DDH, a fact due to Shoup that was used in [52].

Lemma A.1 *Let p be prime and $\mathbb{G} = \langle g \rangle$ be a group of order p. Then there exists a randomized algorithm R that takes inputs from $\mathbb{G}^3$ and produces outputs in $\mathbb{G}^2$ with the following properties. First, if $(X, Y, Z) \xleftarrow{\$} \mathbb{G}^3$ and $(\hat{Y}, \hat{Z}) \xleftarrow{\$} \mathsf{R}(X, Y, Z)$, then $\hat{Y}$ and $\hat{Z}$ are uniformly random over $\mathbb{G}$, and they are independent of X, Y, Z and each other. Second, if $X = g^a, Y = g^b, Z = g^{ab}$ for $a, b \xleftarrow{\$} \mathbb{Z}_p$, and $(\hat{Y}, \hat{Z}) \xleftarrow{\$} \mathsf{R}(X, Y, Z)$, then $\hat{Y}$ is uniformly random and independent of X, Y, Z, and if $\hat{Y} = g^{\hat{b}}$, then $\hat{Z} = g^{a\hat{b}}$.*

Moreover, the running time of R is equal to the time required to compute 4 exponentiations in $\mathbb{G}$.

Lemma A.2 *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and NR^*: $\mathbb{Z}_p^n \times (\{0,1\}^n \setminus \{0^n\}) \to \mathbb{G}$ the family of functions defined via Equation (19). Let A an adversary against the prf-security of NR^* that makes Q oracle queries. Then we can construct an adversary B against the DDH problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{NR}^*}(A) \leq n \cdot \mathbf{Adv}^{\mathrm{ddh}}_{\mathbb{G}}(B) . \tag{35}$$

The running time of B is equal to the running time of A plus the time required to compute $4Q$ exponentiations in $\mathbb{G}$.▌

Proof: Let A be a PRF adversary that issues at most q queries. We will use the games described in Figure 3. The figure describes games G_j, H_j, K_j where $j = 1, \ldots, n$. In each of these games the adversary is given oracle access to $\textsc{Fn}$, which it queries with bit strings which have length j and are not equal to 0^j.

Game G_j implements a version of the PRF experiment with NR^* with input-size and key-size j instead of n.

Game H_j a table $\mathsf{T}[\cdot]$ that is indexed by bit-strings of length j and initialized with $\perp$ at each index. It initially selects r from $\mathbb{Z}_p$ and initializes the indices $\mathsf{T}[0^j]$ and $\mathsf{T}[0^{j-1} \,\|\, 1]$ with g and g^r respectively. On a query to $\textsc{Fn}(x)$, the game returns $\mathsf{T}[x]$ if it is not $\perp$. Otherwise, it takes y to be the first $j-1$ bits of x and assigns values to both $\mathsf{T}[y \,\|\, 0]$ and $\mathsf{T}[y \,\|\, 1]$. The first position gets a random group element h, and the second h^r. The string x will always be one of $y \,\|\, 0$ or $y \,\|\, 1$, and thus $\mathsf{T}[x]$ will be set before returning. We note that, once initialized, the same table index is never written to again.

Game K_j essentially implements a "random oracle" that takes bit-strings of length j as input. On each query it samples an element from $\mathbb{G}$ and returns. It uses the table to ensure that the same value is returned if the same x is queried twice.

By inspection we get that G_n and K_n correspond to the real and random experiments for NR^*. This gives

$$\Pr\left[\,\mathrm{PRFReal}^A_{\mathrm{NR}^*} \Rightarrow 1\,\right] = \Pr\left[\,G^A_n \Rightarrow 1\,\right]$$
$$\Pr\left[\,\mathrm{PRFRand}^A_{\mathrm{NR}^*} \Rightarrow 1\,\right] = \Pr\left[\,K^A_n \Rightarrow 1\,\right].$$

We will show that for every adversary A and $j = 1, \ldots, n$ there exist adversaries $B_1, \ldots, B_j$ such that

$$\Pr\left[\,G^A_j \Rightarrow 1\,\right] - \Pr\left[\,K^A_j \Rightarrow 1\,\right] \leq \sum_{i=1}^{j} \mathbf{Adv}^{\mathrm{ddh}}_{\mathbb{G}}(B_i). \tag{36}$$

The theorem is proved by taking $j = n$. The adversary B in the statement of the theorem can be obtained by picking $j \xleftarrow{\$} \{1, \ldots, n\}$ and running B_j. We omit the standard details here.

We proceed by induction on $j = 1, \ldots, n$. For $j = 1$ we have

$$\Pr\left[\,G^A_1 \Rightarrow 1\,\right] - \Pr\left[\,K^A_1 \Rightarrow 1\,\right] = 0$$

because both G_1 and K_1 allow only a single value of $x = 1$ in queries, and both return a random and independent value for this query. This completes the base case for the induction. For $j > 1$, we add and subtract $\Pr[H^A_j \Rightarrow 1]$ to get

$$\begin{aligned} \Pr\left[\,G^A_j \Rightarrow 1\,\right] - \Pr\left[\,K^A_j \Rightarrow 1\,\right] < \\ (\Pr\left[\,G^A_j \Rightarrow 1\,\right] - \Pr\left[\,H^A_j \Rightarrow 1\,\right]) + (\Pr\left[\,H^A_j \Rightarrow 1\,\right] - \Pr\left[\,K^A_j \Rightarrow 1\,\right]) \end{aligned} \tag{37}$$

We deal with the two addends in (37) separately. We claim that for every adversary A there exists an adversary C such that

$$\Pr\left[\,G^A_j \Rightarrow 1\,\right] - \Pr\left[\,H^A_j \Rightarrow 1\,\right] < \Pr\left[\,G^C_{j-1} \Rightarrow 1\,\right] - \Pr\left[\,K^C_{j-1} \Rightarrow 1\,\right]. \tag{38}$$

We now describe C. It has access to an $\textsc{Fn}$ oracle that takes inputs from $\{0,1\}^{j-1}$ is processed according to either G_{j-1} or K_{j-1}. It starts by picking $s \xleftarrow{\$} \mathbb{Z}_p$ and then running A. When A queries its $\textsc{Fn}$ oracle with $x \in \{0,1\}^j$, C processes it according to the following procedure.

```
If $\mathsf{T}[x] = \perp$ then
    $y \leftarrow x[1, \ldots, j-1]$ ; $k \leftarrow \textsc{Fn}(y)$
    $\mathsf{T}[y \,\|\, 0] \leftarrow k$ ; $\mathsf{T}[y \,\|\, 1] \leftarrow k^s$
Return $\mathsf{T}[x]$
```

If C is interacting with K_{j-1} then is straightforward to verify that it simulates H_{j-1}. If instead it is interacting with G_{j-1}, then C simulates G_j with s playing the role of $\mathbf{a}[j]$. The only difference is that C does some potentially unnecessary preprocessing when populating the table.

We can now use the inductive hypothesis to get

$$\Pr\left[\, G_{j-1}^{C} \Rightarrow 1 \,\right] - \Pr\left[\, K_{j-1}^{C} \Rightarrow 1 \,\right] < \sum_{i=1}^{j-1} \mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(B_i). \tag{39}$$

for some efficient B_i (that depend on C).

We now bound the second addend in (37) by

$$\Pr\left[\, H_j^{A} \Rightarrow 1 \,\right] - \Pr\left[\, K_j^{A} \Rightarrow 1 \,\right] < \mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(B_j), \tag{40}$$

where B_j is an efficient adversary depending on A. B_j gets as input $(X, Y, Z) \in \mathbb{G}^3$. It initializes $\mathsf{T}[0^j] \leftarrow g$ and $\mathsf{T}[0^{j-1} \,\|\, 1] \leftarrow X$ and runs A. It answers queries for A by running the following procedure.

```
If $\mathsf{T}[x] = \perp$ then
    $y \leftarrow x[1, \ldots, j-1]$ ; $(\hat{Y}, \hat{Z}) \xleftarrow{\$} \mathsf{R}(X, Y, Z)$
    $\mathsf{T}[y \,\|\, 0] \leftarrow \hat{Y}$ ; $\mathsf{T}[y \,\|\, 1] \leftarrow \hat{Z}$
Return $\mathsf{T}[x]$
```

We claim that if (X, Y, Z) were selected at random from $\mathbb{G}^3$, then B_j simulates K_j. This is true because the values returned to A are either X or generated by $(\hat{Y}, \hat{Z}) \xleftarrow{\$} \mathsf{R}(X, Y, Z)$. By the first property of R, we have that all of these values are random and independent, as would be the case in K_j.

If, on the other hand, X, Y are random, $X = g^a, Y = g^b$ and $Z = g^{ab}$, then we claim that B_j simulates H_j. This is straightforward to verify. Intuitively, a is playing the role of r from H_j, and R produces a random and independent $\hat{Y}$ to play the role of h, and by the second property of R, $\hat{Z}$ properly plays the role of h^r. This completes the proof of the last claim.

Substituting (39) into (38) and adding (40) gives (36). This completes the inductive step, and hence the proof of the theorem. ▌

```
proc Initialize  // $G_j$
$\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^j$
proc Fn($x$)  // $G_j, x \in \{0,1\}^j$
$y \leftarrow \Pi_{i=1}^{j} \mathbf{a}[i]^{x[i]}$
Return $g^y$
```

```
proc Initialize  // $H_j$
$r \xleftarrow{\$} \mathbb{Z}_p$
$\mathsf{T}[0^j] \leftarrow g$ ; $\mathsf{T}[0^{j-1} \,\|\, 1] \leftarrow g^r$
proc Fn($x$)  // $H_j, x \in \{0,1\}^j$
If $\mathsf{T}[x] = \perp$ then
    $y \leftarrow x[1, \ldots, j-1]$ ; $h \xleftarrow{\$} \mathbb{G}$
    $\mathsf{T}[y \,\|\, 0] \leftarrow h$ ; $\mathsf{T}[y \,\|\, 1] \leftarrow h^r$
Return $\mathsf{T}[x]$
```

```
proc Initialize  // $K_j$
$\mathsf{T}[0^j] \leftarrow g$
proc Fn($x$)  // $K_j, x \in \{0,1\}^j$
If $\mathsf{T}[x] = \perp$ then
    $h \xleftarrow{\$} \mathbb{G}$ ; $\mathsf{T}[x] \xleftarrow{\$} h$
Return $\mathsf{T}[x]$
```

Figure 3: Games for the proof of Theorem A.2.