# Related-Key Security for Pseudorandom Functions Beyond the Linear Barrier

Michel Abdalla[1], Fabrice Benhamouda[1],
Alain Passelègue[1], and Kenneth G. Paterson[2]

[1] Département d'Informatique, École normale supérieure, Paris, France
{michel.abdalla,fabrice.ben.hamouda,alain.passelegue}@ens.fr
http://www.di.ens.fr/users/{mabdalla,fbenhamo,passeleg}
[2] Information Security Group, Royal Holloway, University of London, Surrey, UK
kenny.paterson@rhul.ac.uk
http://www.isg.rhul.ac.uk/~kp/

**Abstract.** Related-key attacks (RKAs) concern the security of cryptographic primitives in the situation where the key can be manipulated by the adversary. In the RKA setting, the adversary's power is expressed through the class of related-key deriving (RKD) functions which the adversary is restricted to using when modifying keys. Bellare and Kohno (Eurocrypt 2003) first formalised RKAs and pin-pointed the foundational problem of constructing RKA-secure pseudorandom functions (RKA-PRFs). To date there are few constructions for RKA-PRFs under standard assumptions, and it is a major open problem to construct RKA-PRFs for larger classes of RKD functions. We make significant progress on this problem. We first show how to repair the Bellare-Cash framework for constructing RKA-PRFs and extend it to handle the more challenging case of classes of RKD functions that contain claws. We apply this extension to show that a variant of the Naor-Reingold function already considered by Bellare and Cash is an RKA-PRF for a class of affine RKD functions under the DDH assumption, albeit with an exponential-time security reduction. We then develop a second extension of the Bellare-Cash framework, and use it to show that the same Naor-Reingold variant is actually an RKA-PRF for a class of degree $d$ polynomial RKD functions under the stronger decisional $d$-Diffie-Hellman inversion assumption. As a significant technical contribution, our proof of this result avoids the exponential-time security reduction that was inherent in the work of Bellare and Cash and in our first result.

**Keywords:** Related-Key Security, Pseudorandom Functions.

## 1 Introduction

**Background and Context.** A common approach to prove the security of a cryptographic scheme, known as provable security, is to relate its security to one of its underlying primitives or to an accepted hard computational problem. While this approach is now standard and widely accepted, there is still a significant gap

J.A. Garay and R. Gennaro (Eds.): CRYPTO 2014, Part I, LNCS 8616, pp. 77–94, 2014.

between the existing models used in security proofs and the actual environment in which these cryptosystems are deployed. For example, most of the existing security models assume that the adversary has no information about the user's secret key. However, it is well known that this is not always true in practice: the adversary may be able to learn partial information about the secrets using different types of side-channel attacks, such as the study of energy consumption, fault injection, or timing analysis. In the particular case of fault injection, for instance, an adversary can learn not only partial information about the secret key, but he may also be able to force a cryptosystem to work with different but related secret keys. Then, if he can observe the outcome of this cryptosystem, he may be able to break it. This is what is known in the literature as a related-key attack (RKA).

Most primitives are designed without taking related-key attacks into consideration so their security proofs do not provide any guarantee against such attacks. Hence, a cryptographic scheme that is perfectly safe in theory may be completely vulnerable in practice. Indeed, many such attacks were found during the last decade, especially against practical blockciphers [10–14, 18]. Inspired by this cryptanalytic work, some years ago, theoreticians started to develop appropriate security models and search for cryptographic primitives which can be proven RKA secure.

**Formal Foundations of RKA Security.** Though RKAs were first introduced by Biham and Knudsen [9, 19] in the early 1990s, it was only in 2003 that Bellare and Kohno [6] began the formalisation of the theoretical foundations for RKA security. We recall their security definition for RKA security of PRFs here. Let $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a family of functions for a security parameter $k$, and let $\Phi = \{\phi\colon \mathcal{K} \to \mathcal{K}\}$ be a set of functions on the key space $\mathcal{K}$, called a related-key deriving (RKD) function set. We say that $F$ is a $\Phi$-RKA-PRF if for any polynomial-time adversary, its advantage in the following game is negligible. The game starts by picking a random challenge bit $b$, a random target key $K \in \mathcal{K}$ and a random function $G\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$. The adversary can repeatedly query an oracle that, given a pair $(\phi, x) \in \Phi \times \mathcal{D}$, returns either $F(\phi(K), x)$, if $b = 1$, or $G(\phi(K), x)$, if $b = 0$. Finally, the adversary outputs a bit $b'$, and its advantage is defined by $2\Pr[\, b = b' \,] - 1$. Note that if the class $\Phi$ of RKD functions contains only the identity function, then this notion matches standard PRF security.

Bellare and Cash [3] designed the first RKA-PRFs secure under standard assumptions, by adapting the Naor-Reingold PRF [21]. Their RKA-PRFs are secure for RKA function classes consisting of certain multiplicative and additive classes. To explain their results, let us begin by recalling the definition of the Naor-Reingold PRF. Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$. Let $\mathsf{NR}\colon (\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ denote the Naor-Reingold PRF that given a 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

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

The keyspace of the Naor-Reingold PRF is $\mathcal{K} = (\mathbb{Z}_p^*)^{n+1}$, which has a group structure under the operation of component-wise multiplication modulo $p$,

denoted $*$. Now let $\Phi_*$ denote the class of component-wise multiplicative functions on $(\mathbb{Z}_p^*)^{n+1}$, that is $\Phi_* = \{\phi\colon\ \mathbf{a} \in (\mathbb{Z}_p^*)^{n+1} \mapsto \mathbf{b} * \mathbf{a} \mid \mathbf{b} \in (\mathbb{Z}_p^*)^{n+1}\}$. It is easy to see that $\mathsf{NR}$ is not itself a $\Phi_*$-RKA-PRF, since it suffers from simple algebraic attacks, but using a collision-resistant hash function $h\colon\ \{0,1\}^n \times \mathbb{G}^{n+1} \to \{0,1\}^{n-2}$, Bellare and Cash were able to show that a simple modification of the Naor-Reingold PRF does yield a $\Phi_*$-RKA-PRF under the DDH assumption. Specifically, they defined $F\colon\ (\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ by:

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

and showed that this $F$ is indeed a $\Phi_*$-RKA-PRF under the DDH assumption. A second construction in [3] uses similar techniques to build an RKA-PRF under the DLIN assumption.

In the original version of their paper, Bellare and Cash also used a variant of the Naor-Reingold PRF, $\mathsf{NR}^*\colon\ (\mathbb{Z}_p)^n \times \{0,1\}^n \backslash \{0^n\} \to \mathbb{G}$, defined by:

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

to obtain a third RKA-PRF, this one for additive RKD functions. In more detail, the keyspace $\mathcal{K} = (\mathbb{Z}_p)^n$ of $\mathsf{NR}^*$, has a natural group structure under the operation of component-wise addition modulo $p$. We define $\Phi_+$ to be the class of functions, $\Phi_+ = \{\phi\colon\ \mathbf{a} \in (\mathbb{Z}_p)^n \mapsto \mathbf{a} + \mathbf{b} \mid \mathbf{b} \in (\mathbb{Z}_p)^n\}$. Then, Bellare and Cash claimed that the function $F\colon\ (\mathbb{Z}_p)^n \times (\{0,1\}^n \setminus 0^n) \to \mathbb{G}$ with

$$F(\mathbf{a}, x) = \mathsf{NR}^*(\mathbf{a}, 11 \| h(x, (g^{\mathbf{a}[1]}, g^{\mathbf{a}[2]}, \ldots, g^{\mathbf{a}[n]})))$$

is a $\Phi_+$-RKA-PRF under the DDH assumption, when the function $h\colon\ \{0,1\}^n \times \mathbb{G}^n \to \{0,1\}^{n-2}$ is a collision-resistant hash function. The running time of their security reduction in this case was exponential in the input size.

These foundational results of [3] were obtained by applying a single, elegant, general framework to the Naor-Reingold PRFs. The framework hinges on two main tools, key-malleability and key-fingerprints for PRFs and associated RKD function classes $\Phi$. The former property means that there is an efficient deterministic algorithm, called a *key-transformer*, that enables one to transform an oracle for computing $F(K, x)$ into one for computing $F(\phi(K), x)$ for any $\phi \in \Phi$ and any input $x$ (the technical requirements are in fact somewhat more involved than these). The latter provides a means to ensure that, in the Bellare-Cash construction for an RKA-PRF from a (normal) PRF $F$, all adversarial queries to the putative $\Phi$-RKA-PRF get appropriately separated before being processed by $F$. In combination, these two features enable a reduction to be made to the PRF security of the underlying function $F$.

Unfortunately, it was recently discovered that the original framework of [3] has a bug, in that a technical requirement on the key-transformer, called hash function compatibility, was too weak to enable the original security proof of the Bellare-Cash construction to go through. When hash function compatibility is appropriately strengthened to enable a proof, it still holds for the key-transformers used

in the analysis of their two main constructions, the multiplicative DDH and DLIN-based RKA-PRF constructions. However, the new compatibility definition no longer holds for the key-transformer used in their additive, DDH-based RKA-PRF construction. With respect to their framework and, specifically, their additive, DDH-based RKA-PRF construction, Bellare and Cash note in the latest version of their paper [4]: *We see no easy way to fill the gap within our current framework and accordingly are retracting our claims about this construction and omitting it from the current version.*

**Main Question.** A natural question that arises from the work of Bellare-Cash is whether it is possible to go further, to obtain RKA-PRFs for larger classes of RKD function than $\Phi_*$ and $\Phi_+$. This is important in understanding whether there are yet to be discovered fundamental barriers in achieving RKA security for PRFs, as well as bringing the current state of the art for RKA security closer to practical application. This question becomes even more relevant in the light of the results of Bellare, Cash and Miller [5], who showed that RKA-security can be transferred from PRFs to several other primitives, including identity-based encryption (IBE), signatures, as well as symmetric (SE) and public-key encryption (PKE) secure against chosen-ciphertext attacks. Their results illustrate the central role that RKA-PRFs play in related-key security more generally: any advance in constructing RKA-PRFs for broader classes would immediately transfer to these other primitives via the results of [5]. A subsidiary question is whether it is possible to repair the Bellare-Cash framework without requiring stronger hash compatibility conditions on the key-transformer. This, if achievable, would reinstate their $\Phi_+$-RKA-PRF.

A partial answer to the first question was provided by Goyal, O'Neill and Rao [17], who proposed RKA-secure weak-PRF and symmetric encryption schemes for polynomial functions using the Decisional Truncated q-ADHE problem. RKA-secure weak-PRFs, however, are significantly weaker than standard RKA-PRFs since their security only holds with respect to random inputs. Wee [22] provided RKA-secure PKE for linear functions, while Bellare, Paterson, and Thomson [7] proposed a framework for obtaining RKA-secure IBE for affine and polynomial RKD function sets, from which RKA security for signatures, PKE (and more) for the same RKD function sets follows using the results of [5] and extensions thereof. However, in respect of these works, it should be noted that achieving RKA security for randomised primitives appears to be substantially easier than for PRFs which are deterministic objects. An extended discussion on this point can be found in [3, Section 1].

In parallel work to ours, Lewi et al. [20] showed that the key homomorphic PRFs from Boneh et al. [15] (and slight extensions of them) are RKA-secure. Specifically, they show RKA-security for a strict subset of $\Phi_+$ for the PRF of [15] that is based on the Learning with Error (LWE) problem, and against a claw-free class of affine functions for the PRF of [15] that is based on multilinear maps. They also showed that, if the adversary's queries are restricted to unique inputs, these two PRFs are RKA-secure for larger classes, namely a class of affine RKD functions (with a low-norm for the "linear" part) for the LWE-based

PRF and a class of polynomial RKD functions for the PRF based on multilinear maps. These classes are not really comparable to our classes $\Phi_{\mathsf{aff}}$ and $\Phi_d$ of affine and polynomial functions defined below, because the secret-key structures are slightly different. However, we remark that Lewi et al. [20] do not deal with claw-free classes and do not show ways to leverage unique-input RKA security to full RKA security. We handle both of these issues in our paper, and it may be possible to extend our solutions to their setting. It should also be remarked that the construction of Barnahee and Peikert [2] may also yield another RKA-secure PRF based on LWE.

**Our Contributions.** In this paper, we make substantial progress on the main question above, obtaining RKA-PRFs for substantially larger classes of RKD functions than were previously known. Along the way, we recover the original Bellare-Cash framework, showing that their original technical conditions on the key-transformer are in fact *already* sufficient to enable a (different) proof of RKA security to go through. Let us first introduce our main results on specific RKA-PRFs, and then explain the technical means by which they are obtained.

For $p$ prime and $n, d \geq 1$, let $\Phi_d$ denote the class of functions from $\mathbb{Z}_p^n$ to $\mathbb{Z}_p^n$ each of whose component functions is a non-constant polynomial in one variable of degree at most $d$. That is, we have:

$$\Phi_d = \left\{ \phi\colon\ \mathbb{Z}_p^n \to \mathbb{Z}_p^n \;\middle|\; \begin{array}{r} \phi = (\phi_1, \ldots, \phi_n); \phi_i : A_i \mapsto \sum_{j=0}^{d} \alpha_{i,j} \cdot A_i^j, \\ \forall i = 1, \ldots, n,\ (\alpha_{i,1}, \ldots, \alpha_{i,d}) \neq 0^d \end{array} \right\} .$$

For the special case $d = 1$, we denote $\Phi_1$ by $\Phi_{\mathsf{aff}}$ ($\mathsf{aff}$ for affine functions). Note that $\Phi_+ \subset \Phi_{\mathsf{aff}}$.

We will construct RKA-PRFs for the RKD-function classes $\Phi_{\mathsf{aff}}$ and $\Phi_d$ for each $d$. To this end, let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$, 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. Let $\mathbf{w}[i] = 0^{i-1} \| 1 \| 0^{n-i}$, for $i = 1, \ldots, n$. Define $F\colon\ \mathbb{Z}_p^n \times (\{0,1\}^n \setminus 0^n) \to \mathbb{G}$ by:

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

for all $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n$. This is the same $F$ as in the withdrawn construction of [3]. Theorems 7 and 13 show that this function is an RKA-PRF for both the RKD-function classes $\Phi_{\mathsf{aff}}$ and $\Phi_d$ (for each $d$), under reasonable hardness assumptions.

For our first result on the $\Phi_{\mathsf{aff}}$-RKA-PRF security of $F$, we recover and extend the withdrawn result of Bellare and Cash [3], under the same hardness assumption that they required, namely the standard DDH assumption. Here our proof, like that in [3], requires an exponential-time reduction. We then develop a further extension of the Bellare-Cash framework enabling us to circumvent their use of key-transformers having a key malleability property. We use this framework to modularise our proof that $F$ is also a $\Phi_d$-RKA-PRF. As part of this proof, we require the decisional $d$-Diffie-Hellman Inversion ($d$-DDHI) assumption, introduced in [17]. Informally, the $d$-DDHI problem in a group $\mathbb{G}$ of prime order $p$ consists of deciding, given inputs $(g, g^a, \ldots, g^{a^d})$ and $z$, where $g$ is a generator

of $\mathbb{G}$, whether $z$ is equal to $g^{\frac{1}{a}}$ or to a random group element. Notably, in our analysis of the $\Phi_d$-RKA-PRF security of $F$, we are able to avoid an exponential-time reduction. This puts the RKA-PRF $F$ on the same footing as the surviving constructions in [3].

Let us now expand on the technical aspects of our contributions.

**Proof Barriers and Techniques.** We first show how the Bellare-Cash framework can be modified to deal with RKD functions that are *not* claw-free, meaning that there exist pairs of different RKD functions $\phi_1$ and $\phi_2$ and a key $K \in \mathcal{K}$, such that $\phi_1(K) = \phi_2(K)$. Up to now, only claw-free classes have been considered for RKA-PRFs. But classes $\Phi$ underlying practical attacks such as fault injections have no reason to be claw-free, so dealing with non-claw-free classes of RKD functions is important in advancing RKA security towards practice. Moreover, both our RKD function classes of interest, $\Phi_{\mathsf{aff}}$ and $\Phi_d$, do contain claws. The lack of claw-freeness poses a problem in security proofs because, if an adversary is able to find two RKD functions which lead to the same derived key, he can detect this via his queries, and then the equation $\phi_1(K) = \phi_2(K)$ may leak information on $K$ sufficient to enable the adversary to break RKA-PRF security in a particular construction.

We overcome the lack of claw-freeness in our adaptation of the Bellare-Cash framework by introducing two new concepts, $\Phi$-Key-Collision Security for PRFs and $\Phi$-Statistical-Key-Collision Security. The former is a property similar to the identity-collision-resistance property defined in [5] in the context of pseudorandom generators and refers to the non-existence of an adversary who can find a colliding key (i.e. $\phi_1(K) = \phi_2(K)$ for $\phi_1, \phi_2 \in \Phi$), when given oracle access to the PRF under related keys $\phi(K)$. The latter concept is essentially the same, but now oracle access to the PRF is replaced by oracle access to a random function. These properties are just the right ingredients necessary to generalise the Bellare-Cash framework to the non-claw-free case.

At the same time as dealing with claws, we are able to repair the gap in the proof for the original Bellare-Cash framework, showing that their original hash function compatibility condition required of the key-transformer is already strong enough to enable an alternative proof of RKA security. Our new proof introduces a slightly different sequence of game hops in order to avoid the apparent impasse in the original proof. Our main theorem establishing the RKA-PRF security of functions arising from this framework is Theorem 1. It repairs and extends the corresponding main theorem in [3]. Our theorem is then combined with an analysis of the specific function $\mathsf{NR}^*$ to obtain Theorem 7 concerning the $\Phi_{\mathsf{aff}}$-RKA-PRF security of $F$.

To show that $F$ is also an RKA-PRF for $\Phi_d$, we still have a second major difficulty to overcome. While $\Phi_d$-Key-Collision Security and $\Phi_d$-Statistical-Key-Collision Security can still be proven for $F$, we no longer have the key-transformer component that is critical to the Bellare-Cash framework. Instead, in Section 5, we introduce a further extension of their framework, replacing the key-transformer with a stronger pseudorandomness condition on the base PRF $M$ used in the construction, which we call $(S, \Phi)$-unique-input-prf-rka security. The new requirement

essentially states that $M$ should already act as a $\Phi$-RKA-PRF on a restricted domain $S$, provided the queries $(\phi_1, x_1), \ldots, (\phi_k, x_k)$ made by the $\Phi$-RKA-PRF adversary to its oracle with $x_i \in S$ are all for *distinct* $x_i$. Under this condition, we are able to prove Theorem 8 establishing the security of RKA-PRFs arising from our further extension of the Bellare-Cash framework. This theorem then enables us to prove in a modular fashion that $F$ is also an RKA-PRF for $\Phi_d$.

The final technical challenge is in proving that $\mathsf{NR}^*$, playing the role of $M$, satisfies the relevant $(S, \Phi)$-unique-input-prf-rka security property so as to allow the application of Theorem 8. This is done in a crucial lemma, Lemma 12, whose proof involves a delicate series of hybrids in which we gradually replace the oracle responses to queries $(\phi_i, x_i)$ for $x_i$ in a suitable set $S$ with random values. We exploit the algebraic nature of the function $\mathsf{NR}^*$ to ensure that the hybrids are close under a particular pair of hardness assumptions the $(N, d)$-Polynomial DDH and $(N, d)$-EDDH assumptions). We also make use of an efficient, approximate (but close to perfect) procedure to detect linear dependencies arising in the simulation from the adversary's oracle queries. This procedure is key to making the entire proof efficient (rather than exponential-time). Finally, we provide a series of reductions relating our pair of hardness assumptions to the $d$-DDHI assumption. Examining the details of the proof shows that we can recover our result concerning $\Phi_{\mathsf{aff}}$-RKA-PRF security of $F$ under DDH (rather than 1-DDHI), but now without an exponential-time reduction.

## 2 Definitions

**Notations and Conventions.** Let $\mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$ be the set of all families of functions $F\colon\ \mathcal{K} \times \mathcal{D} \to \mathcal{R}$. A family of functions $F\colon\ \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ takes a key $K \in \mathcal{K}$ and an input $x \in \mathcal{D}$ and returns an output $F(K, x) \in \mathcal{R}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes its length, and $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[|\mathbf{x}|])$. A binary string $x$ is identified with a vector over $\{0,1\}$ so that $|x|$ denotes its length, $x[i]$ its $i$-th bit and, for $i, j \in \{1, \ldots, n\}$, $i \leq j$, $x[i, \ldots, j]$ the binary string $x[i] \| \ldots \| x[j]$. For a binary string $x \in \{0,1\}^n$ and an integer $d$, we denote by $d \cdot x$ the string $y = y[1] \| \ldots \| y[n] \in \{0, d\}^n$ defined by $y[i] = d \cdot x[i]$ for $i = 1, \ldots, n$. For two strings $x, y \in \{0, \ldots, d\}^n$, we denote by $y \preceq x$ the fact that $y[i] \leq x[i]$, $\forall i = 1, \ldots, n$ and we denote by $S(x)$ the set $\{i \mid x[i] \neq 0\}$. If $\phi$ is a vector of functions from $S_1$ to $S_2$ with $|\phi| = n$ and $\mathbf{a} \in S_1^n$ then we denote by $\phi(\mathbf{a})$ the vector $(\phi[1](\mathbf{a}[1]), \ldots, \phi[n](\mathbf{a}[n])) \in S_2^n$. 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 $(F(K, \mathbf{x}[1]), \ldots, F(K, \mathbf{x}[|\mathbf{x}|]))$. If $S$ is a set, then $|S|$ denotes its size. We denote by $s \xleftarrow{\$} S$ the operation of picking at random $s$ in $S$. If $\mathsf{A}$ is a randomized algorithm, we denote by $y \xleftarrow{\$} \mathsf{A}(x_1, x_2, \ldots)$ the operation of running $\mathsf{A}$ on inputs $(x_1, x_2, \ldots)$ with fresh coins and letting $y$ denote the output.

**Games.** Some of our definitions and proofs use code-based game-playing [8]. Recall that a game has an **Initialize** procedure, procedures to respond to adversary's oracle queries, and a **Finalize** procedure. A game G is executed with an adversary $A$ as follows. First, **Initialize** executes and its outputs are the

inputs to $A$. Then $A$ executes, its oracle queries being answered by the corresponding procedures of G. When $A$ terminates, its outputs become the input to the **Finalize** procedure. The output of the latter, denoted $\mathrm{G}^A$ is called the output of the game, and we let "$\mathrm{G}^A \Rightarrow 1$", abbreviated $W$ in the proofs, denote the event that this game output takes the value 1. Boolean flags are assumed initialized to $\mathsf{false}$. Games $\mathrm{G}_i$, $\mathrm{G}_j$ are identical until $\mathsf{flag}$ if their code differs only in statements that follow the setting of $\mathsf{flag}$ to $\mathsf{true}$. 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.

**PRFs.** PRFs were introduced by [16]. A PRF is a family of functions $F\colon\ \mathcal{K}\times\mathcal{D}\to\mathcal{R}$ which is efficiently computable and so that it is hard to distinguish a function chosen randomly from the PRF family from a random function, which is formally defined as the fact that the advantage of any efficient adversary in attacking the standard prf security of $F$ is negligible. 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

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

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

**RKA-PRFs.** We recall the 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 $\Phi$ are called RKD (Related-Key Deriving) functions. An adversary is said to be $\Phi$-restricted if its oracle queries $(\phi,x)$ satisfy $\phi\in\Phi$. The advantage of a $\Phi$-restricted adversary $A$ in attacking the prf-rka security of $F$ is defined via

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

Game $\mathrm{RKPRFReal}_F$ begins by picking $K \xleftarrow{\$} \mathcal{K}$ and then responds to oracle query $\mathrm{RKFn}(\phi,x)$ via $F(\phi(K),x)$. Game $\mathrm{RKPRFRand}_F$ begins by picking $G \xleftarrow{\$} \mathsf{Fun}(\mathcal{K}\times\mathcal{D},\mathcal{R})$ and responds to oracle query $\mathrm{RKFn}(\phi,x)$ via $G(\phi(K),x)$. We say that $F$ is a $\Phi$-RKA-secure PRF if for any $\Phi$-restricted, efficient adversary, its advantage in attacking the prf-rka security is negligible.

**Strong Key Fingerprint.** A strong key fingerprint is a tool used in proofs to detect whether a key arises more than once in a simulation, even if we do not have any information about the key itself. We recall the definition from [3]. Suppose $F\colon\ \mathcal{K}\times\mathcal{D}\to\mathcal{R}$ is a family of functions. Let $\mathbf{w}$ be a vector over $\mathcal{D}$ and let $n=|\mathbf{w}|$. We say that $\mathbf{w}$ is a *strong key fingerprint* for $F$ if

$$(F(K,\mathbf{w}[1]),\ldots,F(K,\mathbf{w}[n])) \neq (F(K',\mathbf{w}[1]),\ldots,F(K',\mathbf{w}[n]))$$

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

**Key-Malleability.** As defined in [3], let $F\colon \mathcal{K}\times\mathcal{D}\to\mathcal{R}$ be a family of functions and $\Phi$ be a class of RKD functions. Suppose $\mathsf{T}$ 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 $\mathsf{T}^f(\phi,x)\in\mathcal{R}$. $\mathsf{T}$ is said to be a *key-transformer* for $(F,\Phi)$ if it satisfies the *correctness* and *uniformity* conditions. *Correctness* asks that $\mathsf{T}^{F(K,\cdot)}(\phi,x)=F(\phi(K),x)$ for every $(\phi,K,x)\in\Phi\times\mathcal{K}\times\mathcal{D}$. Let us say that a $\Phi$-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. *Uniformity* requires that for any (even inefficient) $\Phi$-restricted, unique-input adversary $U$,

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

where game $\mathrm{KTReal}_{\mathsf{T}}$ is initialized by picking $f\xleftarrow{\$}\mathsf{Fun}(\mathcal{D},\mathcal{R})$ and responds to query $\mathrm{KTFn}(\phi,x)$ via $\mathsf{T}^f(\phi,x)$, while $\mathrm{KTRand}_{\mathsf{T}}$ has no initialization and responds to oracle query $\mathrm{KTFn}(\phi,x)$ by returning a value $y\xleftarrow{\$}\mathcal{R}$ chosen uniformly at random in $\mathcal{R}$. If such a key-transformer exists, we say that $F$ is a $\Phi$-key-malleable PRF.

**Compatible Hash Function.** Let $F\colon \mathcal{K}\times\mathcal{D}\to\mathcal{R}$ be a family of functions and $\Phi$ be a class of RKD functions, such that there is a key-transformer $\mathsf{T}$ for $(F,\Phi)$. Let $\mathbf{w}\in\mathcal{D}^m$ and let $\overline{\mathcal{D}}=\mathcal{D}\times\mathcal{R}^m$. We denote by $\mathsf{Qrs}(\mathsf{T},F,\Phi,\mathbf{w})$ the set of all $w\in\mathcal{D}$ such that there exists $(f,\phi,i)\in\mathsf{Fun}(\mathcal{D},\mathcal{R})\times\Phi\times\{1,\ldots,m\}$ such that the computation of $\mathsf{T}^f(\phi,\mathbf{w}[i])$ makes oracle query $w$. Then, we say that a hash function $H\colon \overline{\mathcal{D}}\to S$ is *compatible* with $(\mathsf{T},F,\Phi,\mathbf{w})$, if $S=\mathcal{D}\backslash\mathsf{Qrs}(\mathsf{T},F,\Phi,\mathbf{w})$. Note that this definition is the same as that given in the original Bellare-Cash framework [3] rather than the stronger one used in the authors' repaired version [4].

**CR hash functions.** The advantage of $C$ in attacking the collision-resistance security of $H\colon \mathcal{D}\to\mathcal{R}$ is

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

where the probability is over $(x,x')\xleftarrow{\$}C$.

**Hardness Assumptions.** Our proofs make use of the $d$-Strong Discrete Logarithm ($d$-SDL) and Decisional $d$-Diffie-Hellman Inversion ($d$-DDHI) problems given in [17] and described in Fig. 1. We define the advantage of an adversary $D$ against the $d$-SDL problem in $\mathbb{G}$ as

$$\mathbf{Adv}^{d\text{-}\mathsf{sdl}}_{\mathbb{G}}(D)=\Pr\left[\,d\text{-}\mathrm{SDL}^{D}_{\mathbb{G}}\Rightarrow\mathsf{true}\,\right]$$

where the probability is over the choices of $a\in\mathbb{Z}_p$, $g\in\mathbb{G}$, and the random coins used by the adversary. The advantage of an adversary $D$ against the $d$-DDHI problem in $\mathbb{G}$ is defined to be

$$\mathbf{Adv}^{d\text{-}\mathsf{ddhi}}_{\mathbb{G}}(D)=\Pr\left[\,d\text{-}\mathrm{DDHI\text{-}Real}^{D}_{\mathbb{G}}\Rightarrow 1\,\right]-\Pr\left[\,d\text{-}\mathrm{DDHI\text{-}Rand}^{D}_{\mathbb{G}}\Rightarrow 1\,\right]$$

| | |
|---|---|
| **proc Initialize** // $d$-SDL<br>$g \xleftarrow{\$} \mathbb{G}$ ; $a \xleftarrow{\$} \mathbb{Z}_p^*$<br>Return $(g, g^a, \ldots, g^{a^d})$ | **proc Finalize**(a') // $d$-SDL<br>Return $(g^a = g^{a'})$ |
| **proc Initialize** // $d$-DDHI-Real<br>$g \xleftarrow{\$} \mathbb{G}$ ; $a \xleftarrow{\$} \mathbb{Z}_p^*$<br>Return $(g, g^a, \ldots, g^{a^d}, g^{1/a})$<br>**proc Finalize**(b) // $d$-DDHI-Real<br>Return $b$ | **proc Initialize** // $d$-DDHI-Rand<br>$g \xleftarrow{\$} \mathbb{G}$ ; $a \xleftarrow{\$} \mathbb{Z}_p^*$ ; $z \xleftarrow{\$} \mathbb{Z}_p^*$<br>Return $(g, g^a, \ldots, g^{a^d}, g^z)$<br>**proc Finalize**(b) // $d$-DDHI-Rand<br>Return $b$ |

**Fig. 1.** Games defining the $d$-SDL and $d$-DDHI problems in $\mathbb{G}$

where the probabilities are over the choices of $a, z \in \mathbb{Z}_p$, $g \in \mathbb{G}$, and the random coins used by the adversary. We have two assumptions corresponding to the hardness of these problems, the $d$-SDL assumption and the $d$-DDHI assumption. Setting $d = 1$ in the $d$-SDL problem, we recover the usual definition of the DL problem in $\mathbb{G}$.

## 3 Repairing and Extending the Bellare-Cash Framework

Here, we give a method to deal with classes of RKD functions that are not claw-free, such as affine classes, by repairing and extending the general framework of Bellare and Cash from [3]. Our approach still relies on key-malleability, meaning that it is not generally applicable since almost all the known PRFs are not key-malleable for interesting classes of functions. However, as we shall see, it *does* provide an easy way to obtain a $\Phi_{\mathsf{aff}}$-RKA-secure PRF, using the variant $\mathsf{NR}^*$ of the Naor-Reingold PRF. In Section 5, we will present a further extension of the Bellare-Cash approach that enables us to deal with PRFs that are not key-malleable.

To deal with non-claw-freeness, we first introduce two new notions. The first one is called $\Phi$-*Key-Collision Security* and captures the likelihood that an adversary finds two RKD functions which lead to the same derived key in a given PRF construction. The second one, called $\Phi$-*Statistical-Key-Collision Security*, is similar, but replaces the oracle access to the PRF with an oracle access to a random function.

**$\Phi$-Key-Collision ($\Phi$-kc) Security.** Let $\Phi$ be a class of RKD functions. We define the advantage of an adversary $A$ against the $\Phi$-key-collision security of a PRF $M$: $\mathcal{K} \times \mathcal{D} \to \mathcal{R}$, denoted by $\mathbf{Adv}^{\mathsf{kc}}_{\Phi,M}(A)$, to be the probability of success in the game on the left side of Fig. 2, where the functions $\phi$ appearing in $A$'s queries are restricted to lie in $\Phi$.

**$\Phi$-Statistical-Key-Collision ($\Phi$-skc) Security.** Let $\Phi$ be a class of RKD functions. We define the advantage of an adversary $A$ against the $\Phi$-statistical-key-collision security for $\mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$, denoted by $\mathbf{Adv}^{\mathsf{skc}}_{\Phi}(A)$, to be the probability of success in the game on the right side of Fig. 2. Here the functions $\phi$ appearing in $A$'s queries are again restricted to lie in $\Phi$.

**proc Initialize**
$K \xleftarrow{\$} \mathcal{K}$
**proc RKFn**$(\phi, x)$
$y \leftarrow M(\phi(K), x)$
Return $y$
**proc Finalize**$(\phi_1, \phi_2)$
Return $(\phi_1 \neq \phi_2$ and $\phi_1(K) = \phi_2(K))$

**proc Initialize**
$K \xleftarrow{\$} \mathcal{K}$ ; $D \leftarrow \emptyset$ ; $E \leftarrow \emptyset$
$F \xleftarrow{\$} \mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$ ; $b' \leftarrow 0$
**proc RKFn**$(\phi, x)$
If $\phi(K) \in E$ and $\phi \notin D$ then $b' \leftarrow 1$
$D \leftarrow D \cup \{\phi\}$ ; $E \leftarrow E \cup \{\phi(K)\}$
$y \leftarrow F(\phi(K), x)$
Return $y$
**proc Finalize**
Return $(b' = 1)$

**Fig. 2.** Game defining the $\Phi$-key-collision security of a PRF $M$ on the left and $\Phi$-statistical-key-collision security for $\mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$ on the right

Using these notions, we can now prove the following theorem, which both repairs and extends the main result of [3].

**Theorem 1.** *Let* $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ *be a family of functions and* $\Phi$ *be a class of* RKD *functions that contains the identity function* $\mathsf{id}$. *Let* $\mathsf{T}$ *be a key-transformer for* $(M, \Phi)$ *making* $Q_{\mathsf{T}}$ *oracle queries, and let* $\mathbf{w} \in \mathcal{D}^m$ *be a strong key fingerprint for* $M$. *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* $(\mathsf{T}, M, \Phi, \mathbf{w})$. *Define* $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ *by*

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

*for all* $K \in \mathcal{K}$ *and* $x \in \mathcal{D}$. *Let* $A$ *be a* $\Phi$*-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 standard prf security of* $M$, *an adversary* $C$ *against the cr security of* $H$, *an adversary* $D$ *against the* $\Phi$*-kc security of* $M$ *and an adversary* $E$ *against* $\Phi$*-skc security for* $\mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$ *such that*

$$\mathbf{Adv}^{\mathsf{prf\text{-}rka}}_{\Phi,F}(A) \;\leq\; \mathbf{Adv}^{\mathsf{prf}}_{M}(B) + \mathbf{Adv}^{\mathsf{cr}}_{H}(C) + \mathbf{Adv}^{\mathsf{kc}}_{\Phi,M}(D) + \mathbf{Adv}^{\mathsf{skc}}_{\Phi}(E)\;. \quad (1)$$

*Adversaries* $C$, $D$ *and* $E$ *have the same running time as* $A$. *Adversary* $B$ *has the same running time as* $A$ *plus the time required for* $Q_A \cdot (m + 1)$ *executions of the key-transformer* $\mathsf{T}$.

Note that if the class $\Phi$ is claw-free, then the advantage of any adversary in breaking $\Phi$-kc security of $M$ or $\Phi$-skc security for $\mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$ is zero. In this case Theorem 1 matches exactly the main theorem of [3], under the original and weaker definition of hash function compatibility from [3]. This justifies our claim of repairing the Bellare-Cash framework.

**Overview of the Proof.** The proof of the above theorem is detailed in the full version [1]. Here we provide a brief overview. Since the RKD functions that we consider in our case may have claws, we start by dealing with possible collisions on the related-keys in the RKPRFReal case, using the key-collision notion. Then, we deal with possible collisions on hash values in order to ensure that the

hash values $h$ used to compute the output $y$ are pairwise distinct so the attacker is unique-input. Then, using the properties of the key-transformer and the compatibility condition, we show that it is hard to distinguish the output from a uniformly random output based on the standard prf security of $M$. Finally, we use the statistical-key-collision security notion to deal with possible key collisions in the RKPRFRand case so that the last game matches the description of the RKPRFRand game.

*Remark 2.* It is worth noting that we deviate from the original proof of [3] in games $\mathrm{G}_5$–$\mathrm{G}_7$, filling the gap in their original proof, but under the same technical conditions on compatibility. Unlike in their proof, we are able to show that the output of $F$ is already indistinguishable from a uniformly random output as soon as one replaces the underlying PRF $M$ with a random function $f$ due to the uniformity condition of the transformer. In order to build a unique-input adversary against the uniformity condition, the main trick is to precompute the values of $f(w)$ for all $w \in \mathsf{Qrs}(\mathsf{T}, M, \Phi, \mathbf{w})$ and use these values to compute $\mathsf{T}^f(\phi, \mathbf{w}[i])$, for $i = 1, \ldots, |\mathbf{w}|$ and $\phi \in \Phi$, whenever needed. This avoids the need to query the oracle in the uniformity game twice on the same input when computing the fingerprint.

## 4 Related-Key Security for Affine RKD Functions

In this section, we apply the above framework to the variant $\mathsf{NR}^*$ of the Naor-Reingold PRF. Recall that $\mathsf{NR}^*$: $(\mathbb{Z}_p)^n \times \{0,1\}^n \backslash \{0^n\} \to \mathbb{G}$ was defined in [3] by:

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

for all $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \backslash \{0^n\}$. We recall the definition of $\Phi_{\mathsf{aff}}$ $(= \Phi_1)$ from the introduction. Using the above theorem, we prove that $\mathsf{NR}^*$ can be used to build a $\Phi_{\mathsf{aff}}$-RKA-secure PRF under the DDH assumption, thereby recovering and strengthening the withdrawn result from [3]. We first recall the following lemma from [3].

**Lemma 3.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^{n} \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \setminus \{0^n\}$. Let $A$ be an adversary against the standard prf security of $\mathsf{NR}^*$ that makes $Q_A$ oracle queries. Then we can construct an adversary $B$ against the DDH problem such that*

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

*The running time of $B$ is equal to the running time of $A$, plus the time required to compute $O(Q_A)$ exponentiations in $\mathbb{G}$.*

In what follows, we prove the properties needed to apply Theorem 1 to $\mathsf{NR}^*$. The proofs of the above lemmas are detailed in the full version [1].

**Strong Key Fingerprint.** Let $\mathbf{w}[i] = 0^{i-1} \| 1 \| 0^{n-i}$, for $i = 1, \ldots, n$. Then $\mathbf{w}$ is a strong key fingerprint for $\mathsf{NR}^*$. Indeed, we have $(\mathsf{NR}^*(\mathbf{a}, \mathbf{w}[1]), \ldots, \mathsf{NR}^*(\mathbf{a}, \mathbf{w}[n]))$

$= (g^{\mathbf{a}[1]}, \ldots, g^{\mathbf{a}[n]})$, so if $\mathbf{a} \neq \mathbf{a}'$ are two distinct keys in $\mathcal{K} = \mathbb{Z}_p^n$, then there exists $i \in \{1, \ldots, n\}$ such that $\mathbf{a}[i] \neq \mathbf{a}'[i]$, so $g^{\mathbf{a}[i]} \neq g^{\mathbf{a}'[i]}$.

**Compatible Hash Function.** We have $\mathsf{Qrs}(\mathsf{T}_{\mathsf{aff}}, \mathsf{NR}^*, \Phi_{\mathsf{aff}}, \mathbf{w}) = \{\mathbf{w}[1], \ldots, \mathbf{w}[n]\}$, so 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 collision resistant hash function. Then the hash function defined by $H(x, \mathbf{z}) = 11 \| h(x, \mathbf{z})$ is a collision resistant hash function that is compatible with $(\mathsf{T}_{\mathsf{aff}}, \mathsf{NR}^*, \Phi_{\mathsf{aff}}, \mathbf{w})$ since every element of $\mathsf{Qrs}(\mathsf{T}_{\mathsf{aff}}, \mathsf{NR}^*, \Phi_{\mathsf{aff}}, \mathbf{w})$ has at most one 1 bit and every output of $H$ has at least two 1 bits. Note that in particular the output of $H$ is never $0^n$, so it is always in the domain of $\mathsf{NR}^*$.

**Lemma 4.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \setminus \{0^n\}$. Let $D$ be an adversary against the $\Phi_{\mathsf{aff}}$-key-collision security of $\mathsf{NR}^*$ that makes $Q_D$ oracle queries. Then we can construct an adversary $C$ against the* DL *problem in $\mathbb{G}$ with the same running time as that of $D$ such that*

$$\mathbf{Adv}^{\mathsf{kc}}_{\Phi_{\mathsf{aff}}, \mathsf{NR}^*}(D) \leq n \cdot \mathbf{Adv}^{\mathsf{dl}}_{\mathbb{G}}(C).$$

Since the hardness of DDH implies the hardness of DL, the above lemma does not introduce any additional hardness assumptions beyond DDH.

**Lemma 5.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \setminus \{0^n\}$. Let $\mathsf{T}_{\mathsf{aff}}$ be defined via*

$$\mathsf{T}^f_{\mathsf{aff}}(\phi, x) = g^{\prod_{i \in S(x)} \mathbf{c}[i]} \cdot \prod_{y \preceq x, y \neq 0^n} f(y)^{\prod_{j \in S(y)} \mathbf{b}[j] \prod_{k \in S(x) \setminus S(y)} \mathbf{c}[k]}$$

*where $\phi = (\phi_1, \ldots, \phi_n) \in \Phi_{\mathsf{aff}}$, with $\phi_i\colon\ a \to \mathbf{b}[i]a + \mathbf{c}[i]$, $\mathbf{b}[i] \neq 0$, for $i = 1, \ldots, n$. Then $\mathsf{T}_{\mathsf{aff}}$ is a key-transformer for $(\mathsf{NR}^*, \Phi_{\mathsf{aff}})$. Moreover, the worst-case running time of this key-transformer is the time required to compute $O(2^n)$ exponentiations in $\mathbb{G}$.*

**Lemma 6.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$. Let $A$ be an adversary against the $\Phi_{\mathsf{aff}}$-statistical-key-collision security for $\mathsf{Fun}(\mathbb{Z}_p^n \times \{0,1\}^n, \mathbb{G})$ making $Q_A$ queries. Then we have*

$$\mathbf{Adv}^{\mathsf{skc}}_{\Phi_{\mathsf{aff}}}(A) \leq \frac{Q_A^2}{2p}\,.$$

We now have everything we need to apply Theorem 1 to $\mathsf{NR}^*$. Combining Theorem 1, Lemmas 3–6 and the above properties, we obtain the following theorem.

**Theorem 7.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \setminus \{0^n\}$. 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. Let $\mathbf{w}[i] = 0^{i-1} \| 1 \| 0^{n-i}$, for $i = 1, \ldots, n$. Define $F\colon\ \mathbb{Z}_p^n \times \{0,1\}^n \to \mathbb{G}$ by*

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

*for all* $\mathbf{a} \in \mathbb{Z}_p^n$ *and* $x \in \{0,1\}^n$. *Let* $A$ *be a* $\Phi_{\mathsf{aff}}$*-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* $\mathbb{G}$*, an adversary* $C$ *against the cr security of* $h$*, and an adversary* $D$ *against the DL problem in* $\mathbb{G}$*, such that*

$$\mathbf{Adv}^{\mathsf{prf\text{-}rka}}_{\Phi_{\mathsf{aff}},F}(A) \;\leq\; n \cdot \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(B) + \mathbf{Adv}^{\mathsf{cr}}_{h}(C) + n \cdot \mathbf{Adv}^{\mathsf{dl}}_{\mathbb{G}}(D) + \frac{Q_A^2}{2p} \;.$$

*The running time of* $B$ *is that of* $A$ *plus the time required to compute* $O(Q_A \cdot (n+1) \cdot 2^n)$ *exponentiations in* $\mathbb{G}$*. The running times of* $C$ *and* $D$ *are the same as that of* $A$*.*

## 5 Further Generalisation of the Bellare-Cash Framework

We introduce a new type of PRF, called an $(S, \Phi)$-Unique-Input-RKA-PRF. We then use this notion as a tool in a further extension of the Bellare-Cash framework that can be applied to *non*-key-malleable PRFs and *non*-claw-free classes of RKD functions. This new framework provides in particular a route to proving that the variant of the Naor-Reingold PRF introduced in Section 3 is actually $\Phi_d$-RKA-secure.

**$(S, \Phi)$-Unique-Input-RKA-PRF.** Let $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a family of functions. Let $S$ be a subset of $\mathcal{D}$ and $\Phi$ be a class of RKD functions. We consider the class of adversaries $A$ in Fig. 3 such that all queries $(\phi, x)$ with $x \in S$ made by $A$ to its oracle are for distinct values of $x$. That is, for any sequence of $A$'s queries $(\phi_1, x_1), \ldots, (\phi_k, x_k)$ with $x_i \in S$ for all $i = 1, \ldots, k$, we require all the $x_i$ to be distinct (no such restriction is made for queries $(\phi_i, x_i)$ with $x_i \notin S$). We denote the advantage of such an adversary $A$ by $\mathbf{Adv}^{\mathsf{ui\text{-}prf\text{-}rka}}_{\Phi,S,M}(B)$. We then say that $M$ is an $(S, \Phi)$-unique-input-RKA-secure PRF if the advantage of any such $\Phi$-restricted, efficient adversary $A$ in attacking $(S, \Phi)$-unique-input-prf-rka security is negligible.

| | |
|---|---|
| **proc Initialize** | **proc RKFn**$(\phi, x)$ |
| $K \xleftarrow{\$} \mathcal{K}$ ; $b \xleftarrow{\$} \{0,1\}$ | If $x \in S$ then |
| **proc Finalize**$(b')$ | If $b = 0$ then $y \leftarrow M(\phi(K), x)$ |
| Return $b' = b$ | Else $y \xleftarrow{\$} \mathcal{R}$ |
| | Else $y \leftarrow M(\phi(K), x)$ |
| | Return $y$ |

**Fig. 3.** Game defining the $(S, \Phi)$-unique-input-prf-rka security of a PRF $M$

The following theorem is an analogue of Theorem 1 in which the roles of key malleability and hash function compatibility are replaced by our new notion, $(S, \Phi)$-unique-input-prf-rka security.

**Theorem 8.** *Let* $M\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ *be a family of functions and* $\Phi$ *be a class of* RKD *functions. Let* $\mathbf{w} \in \mathcal{D}^m$ *be a strong key fingerprint for* $M$*. Let* $\overline{\mathcal{D}} = \mathcal{D} \times \mathcal{R}^m$

*and let* $H\colon \overline{\mathcal{D}} \to S$ *be a hash function, where* $S \subseteq \mathcal{D}\backslash\{\mathbf{w}[1], \ldots, \mathbf{w}[m]\}$. *Define* $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ *by*

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

*for all* $K \in \mathcal{K}$ *and* $x \in \mathcal{D}$. *Let* $A$ *be a* $\Phi$*-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* $(S, \Phi)$*-unique-input-prf-rka security of* $M$*, an adversary* $C$ *against the cr security of* $H$*, an adversary* $D$ *against the* $\Phi$*-kc security of* $M$ *and an adversary* $E$ *against* $\Phi$*-skc security for* $\mathsf{Fun}(\mathcal{K} \times \mathcal{D}, \mathcal{R})$ *such that*

$$\mathbf{Adv}^{\mathsf{prf\text{-}rka}}_{\Phi,F}(A) \;\leq\; \mathbf{Adv}^{\mathsf{ui\text{-}prf\text{-}rka}}_{\Phi,S,M}(B) + \mathbf{Adv}^{\mathsf{cr}}_{H}(C) + \mathbf{Adv}^{\mathsf{kc}}_{\Phi,M}(D) + \mathbf{Adv}^{\mathsf{skc}}_{\Phi}(E)\ . \quad (2)$$

*Adversaries* $C$*,* $D$ *and* $E$ *have the same running time as* $A$*. Adversary* $B$ *makes* $(m+1) \cdot Q_A$ *oracle queries and has the same running time as* $A$*.*

**Overview of the Proof.** The proof of the above theorem is detailed in the full version [1]. Here we provide a brief overview. Since the RKD functions that we consider in our case may have claws, we start by dealing with possible collisions on the related-keys in the RKPRFReal case, using the key-collision notion. Then, we deal with possible collisions on hash values in order to ensure that the hash values $h$ used to compute the output $y$ are distinct. Then, in contrast to the proof of Theorem 1, we use the new $(S, \Phi)$-Unique-Input-RKA-PRF notion and the compatibility condition to show that it is hard to distinguish the output of $F$ from a uniformly random output. Finally, we use the statistical-key-collision security notion to deal with possible key collisions in the RKPRFRand case so that the last game matches the description of the RKPRFRand Game.

*Remark 9.* In the full version [1], we explore the relationship between key-malleable PRFs and unique-input-RKA-secure PRFs. Specifically, we show that the $(S, \Phi)$-unique-input-prf-rka security of a $\Phi$-key-malleable PRF $M$ is implied by its regular prf security if the key-transformer $\mathsf{T}$ associated with $M$ satisfies a new condition that we call $S$-uniformity. This condition demands that the usual uniformity condition for $\mathsf{T}$ should hold on the subset $S$ of $\mathcal{D}$ rather than on all of $\mathcal{D}$. Whether $S$-uniformity is implied by (regular) uniformity is an open question.

## 6 Related-Key Security for Polynomial RKD Functions

We apply Theorem 8 to the variant $\mathsf{NR}^*$ of the Naor-Reingold PRF for the class of RKD functions $\Phi_d = \{\phi\colon \mathcal{K} \to \mathcal{K} | \phi = (\phi_1, \ldots, \phi_n); \phi_i : A \mapsto \sum_{j=0}^{d} \alpha_{i,j} \cdot A^j, (\alpha_{i,1}, \ldots, \alpha_{i,d}) \neq 0^d; \forall i = 1, \ldots, n\}$. Specifically, we prove that $\mathsf{NR}^*$ can be used to build a $\Phi_d$-RKA-secure PRF, under the $d$-DDHI assumption. Remarkably, our proof provides an efficient reduction, avoiding an exponential running time like that seen in Theorem 7. The key step in establishing our result is Lemma 12. Its proof involves at its core the construction of a bespoke key-transformer to handle $\Phi_d$ and a delicate analysis of it using sequences of hybrid games.

In what follows, we prove the various properties needed to apply Theorem 8 to $\mathsf{NR}^*$. The proofs of Lemmas 10–12 can be found in the full version [1].

**Strong Key Fingerprint.** Let $\mathbf{w}[i] = 0^{i-1}\|1\|0^{n-i}$, for $i = 1, \ldots, n$. Then, as before, $\mathbf{w}$ is a strong key fingerprint for $\mathsf{NR}^*$.

**Hash Function.** Let $\overline{\mathcal{D}} = \{0,1\}^n \times \mathbb{G}^n$ and let $h$: $\overline{\mathcal{D}} \to \{0,1\}^{n-2}$ be a collision resistant hash function. Then, as previously, the hash function defined by $H(x, \mathbf{z}) = 11\|h(x, \mathbf{z})$ is a collision resistant hash function with range $S$ satisfying the relation $S \subseteq \{0,1\}^n \backslash (\{\mathbf{w}[1], \ldots, \mathbf{w}[n]\} \cup \{0^n\})$.

**Lemma 10.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \setminus \{0^n\}$. Let $D$ be an adversary against the $\Phi_d$-key-collision security of $\mathsf{NR}^*$ that makes $Q_D$ oracle queries. Then we can construct an adversary $C$ against the $d$-SDL problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}^{\mathsf{kc}}_{\Phi_d, \mathsf{NR}^*}(D) \leq n \cdot \mathbf{Adv}^{d\text{-}\mathsf{sdl}}_{\mathbb{G}}(C) \ .$$

*The running time of $C$ is that of $D$ plus the time required to factorize a polynomial of degree at most $d$ in $\mathbb{F}_p$ (sub-quadratic in $d$ and logarithmic in $p$) plus $O(Q_D \cdot d)$ exponentiations in $\mathbb{G}$.*

**Lemma 11.** *Let $\mathbb{G}$ be a group of prime order $p$. Let $\mathsf{Fun}(\mathbb{Z}_p^n \times \{0,1\}^n, \mathbb{G})$ be the set of functions from which the random function in the $\Phi_d$-statistical-key-collision security game is taken. Let $A$ be an adversary against the $\Phi_d$-statistical-key-collision security that makes $Q_A$ queries. Then we have*

$$\mathbf{Adv}^{\mathsf{skc}}_{\Phi_d}(A) \leq \frac{d \cdot Q_A^2}{2p} \ .$$

**Lemma 12.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \setminus \{0^n\}$. Let $S$ denote the set $\{0,1\}^n \backslash (\{0^n\} \cup \{\mathbf{w}[1], \ldots, \mathbf{w}[n]\})$. Let $A$ be an adversary against the $(S, \Phi_d)$-unique-input-prf-rka security of $\mathsf{NR}^*$ that makes $Q_A$ oracle queries. Then, assuming $nd \leq \sqrt{p}$, we can design an adversary $B$ against the $d$-DDHI problem in $\mathbb{G}$ such that*

$$\mathbf{Adv}^{\mathsf{ui\text{-}prf\text{-}rka}}_{\Phi_d, S, \mathsf{NR}^*}(B) \leq \left( n \cdot d \cdot \left( \frac{p}{p-1} \right)^2 + n \cdot (d-1) \right) \cdot \mathbf{Adv}^{d\text{-}\mathsf{ddhi}}_{\mathbb{G}}(A) \ + \frac{2n \cdot Q_A}{p} \ .$$

*The running time of $B$ is that of $A$ plus the time required to compute $O(d \cdot (n + Q_A))$ exponentiations in $\mathbb{G}$ and $O(Q_A^3 \cdot (nd + Q_A))$ operations in $\mathbb{Z}_p$.*

Finally, by combining the results in Lemmas 10–12 with Theorem 8, we can prove the following theorem.

**Theorem 13.** *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $p$ and let $\mathsf{NR}^*$ be defined via $\mathsf{NR}^*(\mathbf{a}, x) = g^{\prod_{i=1}^n \mathbf{a}[i]^{x[i]}}$, where $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n \backslash \{0^n\}$. Let $\mathcal{D} = \{0,1\}^n \times$*

*$\mathbb{G}^n$ and let $h\colon \mathcal{D} \to \{0,1\}^{n-2}$ be a hash function. Let $\mathbf{w}[i] = 0^{i-1}\|1\|0^{n-i}$, for $i = 1, \ldots, n$. Define $F\colon \mathbb{Z}_p^n \times \{0,1\}^n \to \mathbb{G}$ by*

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

*for all $\mathbf{a} \in \mathbb{Z}_p^n$ and $x \in \{0,1\}^n$. Let $A$ be a $\Phi_d$-restricted adversary against the prf-rka security of $F$ that makes $Q_A \leq |\{0,1\}^{n-2}|$ oracle queries. Then, assuming $nd \leq \sqrt{p}$, we can construct an adversary $B$ against the $d$-DDHI problem in $\mathbb{G}$, an adversary $C$ against the cr security of $h$, and an adversary $D$ against the $d$-SDL problem in $\mathbb{G}$ such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf\text{-}rka}}_{\Phi_d,F}(A) \;\leq\; & \left(n \cdot d \cdot (1-1/p)^2 + n \cdot (d-1)\right) \cdot \mathbf{Adv}^{d\text{-}\mathsf{ddhi}}_{\mathbb{G}}(B) \\ & + \mathbf{Adv}^{\mathsf{cr}}_{h}(C) + n \cdot \mathbf{Adv}^{d\text{-}\mathsf{sdl}}_{\mathbb{G}}(D) + \left(d \cdot Q_A^2 + 4n \cdot Q_A\right)/(2p)\,. \quad (3)\end{aligned}$$

*The running time of $B$ is that of $A$ plus $O(d \cdot (n + Q_A))$ exponentiations in $\mathbb{G}$ and $O(Q_A^3 \cdot (nd + Q_A))$ operations in $\mathbb{Z}_p$. $C$ has the same running time as $A$. The running time of $D$ is that of $A$ plus the time required to factorize a polynomial of degree at most $d$ in $\mathbb{F}_p$, which is sub-quadratic in $d$, logarithmic in $p$.*

**Acknowledgements.** We thank Susan Thomson for bringing the issues in the original Bellare-Cash framework to our attention, and for useful comments on the paper. Michel Abdalla, Fabrice Benhamouda, and Alain Passelègue were supported by the French ANR-10-SEGI-015 PRINCE Project, the *Direction Générale de l'Armement* (DGA), the CFM Foundation, the European Commission through the FP7-ICT-2011-EU-Brazil Program under Contract 288349 SecFuNet, and the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013 Grant Agreement 339563 – CryptoCloud). Kenneth G. Paterson was supported by an EPSRC Leadership Fellowship, EP/H005455/1.

## References

1. Abdalla, M., Benhamouda, F., Passelègue, A., Paterson, K.G.: Related-key security for pseudorandom functions beyond the linear barrier, full version of this paper available at Cryptology ePrint Archive, `http://eprint.iacr.org/`
2. Banerjee, A., Peikert, C.: New and improved key-homomorphic pseudorandom functions. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 353–370. Springer, Heidelberg (2014)
3. Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010)
4. Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. Cryptology ePrint Archive, Report 2010/397 (2010), `http://eprint.iacr.org/` (last updated October 27, 2013)
5. Bellare, M., Cash, D., Miller, R.: Cryptography secure against related-key attacks and tampering. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 486–503. Springer, Heidelberg (2011)

6. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003)
7. Bellare, M., Paterson, K.G., Thomson, S.: RKA security beyond the linear barrier: IBE, encryption and signatures. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 331–348. Springer, Heidelberg (2012)
8. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
9. Biham, E.: New types of cryptanalytic attacks using related keys. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 398–409. Springer, Heidelberg (1994)
10. Biham, E., Dunkelman, O., Keller, N.: Related-key boomerang and rectangle attacks. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005)
11. Biham, E., Dunkelman, O., Keller, N.: A unified approach to related-key attacks. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 73–96. Springer, Heidelberg (2008)
12. Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., Shamir, A.: Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 299–319. Springer, Heidelberg (2010)
13. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)
14. Biryukov, A., Khovratovich, D., Nikolić, I.: Distinguisher and related-key attack on the full AES-256. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg (2009)
15. Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 410–428. Springer, Heidelberg (2013)
16. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. In: 25th FOCS, pp. 464–479. IEEE Computer Society (October 1984)
17. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)
18. Kim, J., Hong, S., Preneel, B.: Related-key rectangle attacks on reduced AES-192 and AES-256. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 225–241. Springer, Heidelberg (2007)
19. Knudsen, L.R.: Cryptanalysis of LOKI91. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 196–208. Springer, Heidelberg (1993)
20. Lewi, K., Montgomery, H., Raghunathan, A.: Improved constructions of PRFs secure against related-key attacks. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) ACNS 2014. LNCS, vol. 8479, pp. 44–61. Springer, Heidelberg (2014)
21. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th FOCS, pp. 458–467. IEEE Computer Society (October 1997)
22. Wee, H.: Public key encryption against related key attacks. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 262–279. Springer, Heidelberg (2012)