Cryptography Secure Against Related-Key Attacks and Tampering

Mihir Bellare[1], David Cash[2], and Rachel Miller[3]

[1] Department of Computer Science & Engineering, University of California San Diego, http://www.cs.ucsd.edu/users/mihir
[2] IBM T.J. Watson Research Center, http://www.cs.ucsd.edu/users/cdcash
[3] Department of Electrical Engineering and Computer Science, MIT, http://people.csail.mit.edu/rmiller/

Abstract. We show how to leverage the RKA (Related-Key Attack) security of blockciphers to provide RKA security for a suite of high-level primitives. This motivates a more general theoretical question, namely, when is it possible to transfer RKA security from a primitive P_1 to a primitive P_2? We provide both positive and negative answers. What emerges is a broad and high level picture of the way achievability of RKA security varies across primitives, showing, in particular, that some primitives resist "more" RKAs than others. A technical challenge was to achieve RKA security even for the practical classes of related-key deriving (RKD) functions underlying fault injection attacks that fail to satisfy the "claw-freeness" assumption made in previous works. We surmount this barrier for the first time based on the construction of PRGs that are not only RKA secure but satisfy a new notion of identity-collision-resistance.

1 Introduction

By fault injection [16, 10] or other means, it is possible for an attacker to induce modifications in a hardware-stored key. When the attacker can subsequently observe the outcome of the cryptographic primitive under this modified key, we have a related-key attack (RKA) [5, 19].

The key might be a signing key of a certificate authority or SSL server, a master key for an IBE system, or someone's decryption key. Once viewed merely as a way to study the security of blockciphers [9, 27, 5], RKAs emerge as real threats in practice and of interest for primitives beyond blockciphers.

It becomes of interest, accordingly, to achieve (provable) RKA security for popular high-level primitives. How can we do this?

Practical contributions. One approach to building RKA-secure high-level primitives is to do so directly, based, say, on standard number-theoretic assumptions. This, however, is likely to yield ad hoc results providing security against classes of attacks that are tied to the scheme algebra and may not reflect attacks in practice.

We take a different approach. RKA security is broadly accepted in practice as a requirement for blockciphers; in fact, AES was designed with the explicit goal

of resisting RKAs. We currently have blockciphers whose resistance to RKAs is backed by fifteen years of cryptanalytic and design effort. We propose to leverage these efforts.

We will provide a general and systematic way to immunize any given instance of a high-level primitive against RKAs with the aid of an RKA-secure blockcipher, modeling the latter, for the purpose of proofs, as a RKA-secure PRF [5]. We will do this not only for symmetric primitives that are "close" to PRFs like symmetric encryption, but even for public-key encryption, signatures and identity-based encryption. Our methods are cheap, non-intrusive from the software perspective, and able to completely transfer all the RKA security of the blockcipher so that the high-level primitive resists attacks of the sort that arise in practice.

Theoretical contributions. The ability to transfer RKA security from PRFs to other primitives lead us to ask a broader theoretical question, namely, when is it possible to transfer RKA security from a primitive P_1 to a primitive P_2? We provide positive results across a diverse set of primitives, showing, for example, that RKA-secure IBE implies RKA-secure IND-CCA PKE. We also provide negative results showing, for example, that RKA-secure signatures do not imply RKA-secure PRFs.

All our results are expressed in a compact set-based framework. For any primitive P and class Φ of related-key deriving functions —functions the adversary is allowed to apply to the target key to get a related key— we define what it means for an instance of P to be Φ-RKA secure. We let $\mathbf{RKA}[\mathsf{P}]$ be the set of all Φ such that there exists a Φ-RKA secure instance of primitive P. A transfer of RKA security from P_1 to P_2, expressed compactly as a set containment $\mathbf{RKA}[\mathsf{P}_1] \subseteq \mathbf{RKA}[\mathsf{P}_2]$, is a construction of a Φ-RKA secure instance of P_2 given both a normal-secure instance of P_2 and a Φ-RKA secure instance of P_1. Complementing this are non-containments of the form $\mathbf{RKA}[\mathsf{P}_2] \not\subseteq \mathbf{RKA}[\mathsf{P}_1]$, which show the existence of Φ such that there exists a Φ-RKA instance of P_2 yet *no* instance of P_1 can be Φ-RKA secure, indicating, in particular, that RKA security cannot be transferred from P_2 to P_1.

As Fig. 1 shows, we pick and then focus on a collection of central and representative cryptographic primitives. We then establish these containment and non-containment relations in a comprehensive and systematic way. What emerges is a broad and high level picture of the way achievability of RKA security varies across primitives, showing, in particular, that some primitives resist "more" RKAs than others.

We view these relations between $\mathbf{RKA}[\mathsf{P}]$ sets as an analog of complexity theory, where we study relations between complexity classes in order to better understand the computational complexity of particular problems. Let us now look at all this more closely.

Background. Related-key attacks were conceived in the context of blockciphers [9, 27]. The first definitions were accordingly for PRFs [5]; for $F\colon\ \mathcal{K}\times\mathcal{D}\to\mathcal{R}$ they consider the game that picks a random challenge bit b and random target key $K\in\mathcal{K}$. For each $L\in\mathcal{K}$ the game picks a random function $G(L,\cdot)\colon\ \mathcal{D}\to\mathcal{R}$,

and next allows the adversary multiple queries to an oracle that given a pair (ϕ, x) with $\phi\colon\ \mathcal{K} \to \mathcal{K}$ and $x \in \mathcal{D}$ returns $F(\phi(K), x)$ if $b = 1$ and $G(\phi(K), x)$ if $b = 0$. They say that F is Φ-RKA secure, where Φ is a class of functions mapping $\mathcal{K}$ to $\mathcal{K}$, if the adversary has low advantage in predicting b when it is only allowed in its queries to use functions ϕ from Φ.

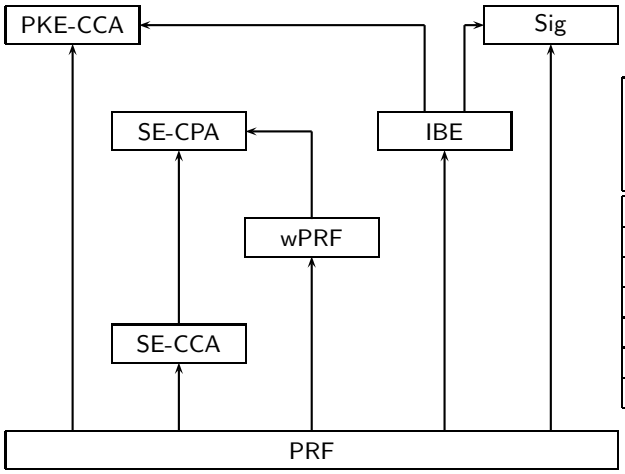
Let $\mathbf{RKA}[\mathsf{PRF}]$ be the set of all Φ for which there exists a Φ-RKA secure PRF. Which Φ are in this set? All the evidence so far is that this question has no simple answer. Bellare and Kohno [5] gave natural examples of Φ not in $\mathbf{RKA}[\mathsf{PRF}]$, showing the set is not universal. Membership of certain specific Φ in $\mathbf{RKA}[\mathsf{PRF}]$ have been shown by explicit constructions of Φ-RKA PRFs, first under novel assumptions [28] and then under standard assumptions [3]. Beyond this we must rely on cryptanalysis. Modern blockciphers including AES are designed with the stated goal of RKA security. Accordingly we are willing to assume their Φ-RKA security —meaning that $\Phi \in \mathbf{RKA}[\mathsf{PRF}]$— for whatever Φ cryptanalysts have been unable to find an attack.

Beyond PRFs. Consideration of RKAs is now expanding to primitives beyond PRFs [20, 2, 22]. This is viewed partly as a natural extension of the questions on PRFs, and partly as motivated by the view of RKAs as a class of sidechannel attacks [19]. An RKA results when the attacker alters a hardware-stored key via tampering or fault injection [16, 10] and subsequently observes the result of the evaluation of the primitive on the modified key. The concern that such attacks could be mounted on a signing key of a certificate authority or SSL server, a master key for an IBE system, or decryption keys of users makes achieving RKA security interesting for a wide range of high-level primitives.

Definitions. We focus on a small but representative set of primitives for which interesting variations in achievability of RKA security emerge. These are PRF (pseudorandom functions), Sig (Signatures), $\mathsf{PKE\text{-}CCA}$ (CCA-secure public-key encryption), $\mathsf{SE\text{-}CCA}$ (CCA-secure symmetric encryption), $\mathsf{SE\text{-}CPA}$ (CPA-secure symmetric encryption), IBE (identity-based encryption) and wPRF (weak PRFs [29]). We define what it means for an instance of P to be Φ-RKA secure for each $\mathsf{P} \in \{\mathsf{wPRF}, \mathsf{IBE}, \mathsf{Sig}, \mathsf{SE\text{-}CCA}, \mathsf{SE\text{-}CPA}, \mathsf{PKE\text{-}CCA}\}$. We follow the definitional paradigm of [5], but there are some delicate primitive-dependent choices that significantly affect the strength of the definitions and the challenge of achieving them (cf. Section 2). We let $\mathbf{RKA}[\mathsf{P}]$ be the set of all Φ for which there exists a Φ-RKA secure instance of P. These sets are all non-trivial.

Relations. We establish two kinds of relations between sets $\mathbf{RKA}[\mathsf{P}_1]$ and $\mathbf{RKA}[\mathsf{P}_2]$:

- Containment: A proof that $\mathbf{RKA}[\mathsf{P}_1] \subseteq \mathbf{RKA}[\mathsf{P}_2]$, established by constructing a Φ-RKA secure instance of P_2 from a Φ-RKA secure instance of P_1, usually under the (minimal) additional assumption that one is given a normal-secure instance of P_2. Containments yield constructions of Φ-RKA secure instances of P_2.
- Non-containment: A proof that $\mathbf{RKA}[\mathsf{P}_2] \not\subseteq \mathbf{RKA}[\mathsf{P}_1]$. Here we exhibit a particular Φ for which we (1) construct a Φ-RKA secure instance of P_1 under



	PRF	wPRF	IBE	Sig	SE-CCA	SE-CPA	PKE-CCA
PRF	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
wPRF	$\not\subseteq$	$\subseteq$	$\not\subseteq$			$\subseteq$	$\not\subseteq$
IBE	$\not\subseteq$	$\not\subseteq$	$\subseteq$	$\subseteq$	$\not\subseteq$	$\not\subseteq$	$\subseteq$
Sig	$\not\subseteq$	$\not\subseteq$		$\subseteq$	$\not\subseteq$	$\not\subseteq$	$\not\subseteq$
SE-CCA	$\not\subseteq$				$\subseteq$	$\subseteq$	
SE-CPA	$\not\subseteq$		$\not\subseteq$		$\not\subseteq$	$\subseteq$	$\not\subseteq$
PKE-CCA	$\not\subseteq$	$\not\subseteq$			$\not\subseteq$	$\not\subseteq$	$\subseteq$

Fig. 1. Relations between RKA[P] classes. A containment $\mathbf{RKA}[\mathsf{P}_1] \subseteq \mathbf{RKA}[\mathsf{P}_2]$ is represented in the picture by an arrow $\mathsf{P}_1 \rightarrow \mathsf{P}_2$ and in the table by a "$\subseteq$" in the row P_1, column P_2 entry. A non-containment $\mathbf{RKA}[\mathsf{P}_1] \not\subseteq \mathbf{RKA}[\mathsf{P}_2]$ is represented in the table by a "$\not\subseteq$" in the row P_1, column P_2 entry. The picture does not show non-containments. The picture sometimes shows a redundant containment (for example the arrow $\mathsf{PRF} \rightarrow \mathsf{Sig}$ when there is already a path $\mathsf{PRF} \rightarrow \mathsf{IBE} \rightarrow \mathsf{Sig}$) because it corresponds to an interesting direct construction. A blank entry in the table means we do not know.

some reasonable assumption, and (2) show, via attack, that *any* instance of P_2 is Φ-RKA insecure.

We show that RKA-secure PRFs are powerful enablers of RKA-security: Given a Φ-RKA PRF and a normal-secure instance of P, we construct a Φ-RKA secure instance of P for all $\mathsf{P} \in \{\mathsf{wPRF}, \mathsf{IBE}, \mathsf{Sig}, \mathsf{SE\text{-}CCA}, \mathsf{SE\text{-}CPA}, \mathsf{PKE\text{-}CCA}\}$. This is represented by the string of containments in the first row of the table in Fig. 1. On the practical side, instantiating the PRF with a blockcipher yields a cheap way to immunize the other primitives against RKAs. On the theoretical side, instantiating the PRF with the construct of [3] yields Φ-RKA secure instances of the other primitives based on standard assumptions.

The separations shown in the first column of the table of Fig. 1, however, also show that RKA-PRFs are overkill: *all* the other primitives admit Φ-RKA secure instances for a Φ for which no Φ-RKA PRF exists. This leads one to ask whether there are alternative routes to RKA-secure constructions of beyond-PRF primitives.

We show that IBE is a particularly powerful starting point. We observe that Naor's transform preserves RKA-security, allowing us to turn a Φ-RKA secure IBE scheme into a Φ-RKA secure Sig scheme. Similarly, we show that the transform of Boneh, Canetti, Halevi and Katz (BCHK) [15] turns a Φ-RKA secure

IBE scheme into a Φ-RKA secure PKE-CCA scheme. What lends these transforms well to RKA-security is that they do not change the secret key. We also show that given a Φ-RKA secure wPRF we can build a Φ-RKA secure SE-CPA scheme. (A wPRF is like a PRF except that is only required to be secure on random inputs [29].) These results motivate finding new Φ-RKA secure IBE schemes and wPRFs.

As the table of Fig. 1 indicates, we show a number of other non-containments. Sig emerges as a very "RKA-resilient" primitive in the sense that it can be secure against strictly more RKAs than most other primitives. Some of the non-containments, such as $\mathbf{RKA}[\mathsf{PKE\text{-}CCA}] \not\subseteq \mathbf{RKA}[\mathsf{SE\text{-}CPA}]$ might seem odd; doesn't PKE always imply SE? What we are saying is that the trivial transformation of a PKE scheme to an SE one does not preserve RKA-security and, moreover, there are Φ for which *no transform exists* that can do this.

Claws ok. All previous constructions of Φ-RKA secure primitives [5, 28, 3, 20, 2, 22, 23] assume Φ is claw-free (distinct functions in ϕ disagree on all inputs) because it is hard to do the proofs otherwise, but the Φ underlying practical fault injection attacks are not claw-free, making it desirable to get constructions avoiding this assumption. For the first time, we are able to do this. In Section 2 we explain the technical difficulties and sketch our solution, which is based on the construction of a Φ-RKA PRG that has a novel property we call identity-collision-resistance (ICR), a variant of the collision-resistance property from [24].

Related work. The first theoretical treatment of RKAs was by Bellare and Kohno [5]; being inspired by blockciphers, the work addressed PRFs and PRPs. They showed examples of classes not in $\mathbf{RKA}[\mathsf{PRF}]$, gave conditions on Φ for ideal ciphers to be Φ-RKA secure, and provided standard model constructs for some limited classes. Subsequently, constructions of Φ-RKA secure PRFs and PRPs for more interesting Φ were found, first under novel assumptions [28] and then under standard assumptions [3], and the results on ideal ciphers were extended in [1].

We are seeing growing interest in RKA security for primitives other than PRFs. Goldenberg and Liskov [20] study related-secret security of lower-level primitives, namely one-way functions, hardcore bits and pseudorandom generators. Applebaum, Harnik and Ishai [2] define RKA security for (randomized) symmetric encryption, gave several constructions achieving that definition for interesting Φ and then presented numerous applications. Connections with point obfuscation are made by Bitansky and Canetti [11].

Gennaro, Lysyanskaya, Malkin, Micali and Rabin [19] suggest that RKAs may arise by tampering. They show that one can achieve security when related keys are derived via arbitrary key modification, but assume an external trusted authority signs the original secret key and installs the signature on the device together with its own public key, the latter being "off limits" to the attacker. (Meaning, the related-key deriving functions may not modify them.) In our case, no such authority is assumed. The off-limit quantities are confined to pre-installed public parameters. No information that is a function of the parameters and the key is installed on the chip.

Ishai, Prabhakaran, Sahai and Wagner [25] are concerned with tampering of wires in the computation of a circuit while we are concerned with tampering with hardware-stored keys. Dziembowski, Pietrzak and Wichs [18] develop an information theoretic method for preventing tampering and show that a wide class of limited, but non-trivial, Φ can be achieved (unconditionally) for any so-called "interactive stateful system."

INDEPENDENT WORK. Interest in RKA security for higher-level primitives is evidenced by Goyal, O'Neill and Rao [22, 23], who define correlated-input (CI) hash functions, show how to construct them from the q-DHI assumption based on Boneh-Boyen signatures [13, 14] and the Dodis-Yampolskiy PRF [17], and apply this to get Φ-RKA secure signatures from q-DHI for a class Φ consisting of polynomials over a field of prime order. (They indicate their approach would also work for other primitives.) Their construction is similar to ours. Their definitions and results, unlike ours, are restricted to claw-free Φ. Also, we start from Φ-RKA-PRFs and thus get in-practice security for any class Φ for which blockciphers provide them, while they start from a number-theoretic assumption and get security for a specific class Φ, related to the scheme algebra. Their work and ours are concurrent and independent. (Ours was submitted to, and rejected from, Eurocrypt 2011, while theirs was submitted to, and accepted at, TCC 2011.)

Kalai, Kanukurthi and Sahai [26] provide encryption and signature schemes that protect against both tampering and leakage via the idea of key-updates that originated in forward-secure signatures [7]. They allow arbitrary tampering functions but only allow a bounded number of tampering queries within each time period. Their work and ours are again concurrent and independent.

2 Technical approach

Before providing formal definitions, constructions and proofs of our many positive and negative results, we would like to illustrate one technical issue, namely the challenges created by Φ that are not claw-free and how we resolve them. For concreteness, our discussion is restricted to the design of Φ-RKA signatures based on Φ-RKA PRFs.

THE CLAW-FREENESS ASSUMPTION. All known constructions of Φ-RKA-secure primitives [5, 28, 3, 20, 2, 22, 23] are restricted to Φ that are *claw-free*. This means that any two distinct functions in Φ disagree on *all* inputs. This assumption is made for technical reasons; it seems hard to do simulations and proofs without it. Yet the assumption is undesirable, for many natural and practical classes of functions are *not* claw-free. For example, fault injection might be able to set a certain bit of the key to zero, and if Φ contains the corresponding function and the identity function then it is not claw-free. Any Φ that can set the key to a constant value is also not claw-free. Accordingly it is desirable to avoid this assumption. For the first time we are able to do so, via a new technical approach.

DEFINITIONS AND ISSUES. The degree to which claw-freeness is embedded in current approaches is made manifest by the fact that the very *definition* of Φ-RKA secure signatures of [22, 23] assumes it and is unachievable without it. Let us take a closer look to see how.

The signature RKA-security game of [22, 23] picks secret signing key sk and associated public verification key vk. It gives the adversary a signing oracle SIGN that takes m and $\phi \in \Phi$, and returns the signature of message m under key $\phi(sk)$. The adversary eventually outputs m, σ. Besides validity of m, σ under vk, winning requires that m be "new," meaning not "previously signed." The delicate question is, how do we define this? The choice of [22, 23] is to disallow signing query id, m, where id is the identity function. But the adversary can easily define a function ϕ that is the identity on all but a negligible fraction of its inputs. A query ϕ, m is then valid since $\phi \neq \mathsf{id}$, but almost always returns the signature σ of m under sk, so the adversary can output m, σ and win. By assuming Φ is claw-free and contains id, [22, 23] ensure that such a ϕ is not in Φ and the attack is ruled out.

Our altered definition of m being "new" is that there was no signing query ϕ, m with $\phi(sk) = sk$. This seems, indeed, the natural requirement, ruling out nothing more than that m was signed under sk.

We now have a much more general definition that is meaningful even for the non claw-free Φ that arise in practice, but it has a subtle feature that makes achieving it a challenge. Namely, *checking whether the adversary won apparently requires knowing sk* for we have to test whether or not $\phi(sk) = sk$. In the reduction proving security, we will be designing an adversary B attempting to distinguish "real" or "random" instances of some problem given an adversary A breaking the signature scheme; B will see if A won, declaring "real" if so and "random" otherwise. But B will be simulating A and will not know sk, so the difficulty is how it can test that A won.

OVERVIEW OF SOLUTION. We start from a Φ-RKA secure PRF F: $\mathcal{K} \times \mathcal{D} \to \mathcal{R}$ that has what we call a key fingerprint for the identity function. This is a relaxation of the notion of a key fingerprint of [3]. It consists of a vector $\mathbf{w}$ over $\mathcal{D}$ such that for all K and all $\phi \in \Phi$ with $\phi(K) \neq K$ there is some i such that $F(K, \mathbf{w}[i]) \neq F(\phi(K), \mathbf{w}[i])$. This allows statistical disambiguation of the original key K from other keys. Such fingerprints exist for the Φ-RKA PRFs of [3] and for blockciphers and are thus a mild assumption.

We now turn F into a PRG (Pseudorandom Generator) $\mathcal{G}$ that has two properties. First, it is Φ-RKA secure; this means the adversary has low advantage in determining the challenge bit b in the game that picks a random target key K and random function R, and then gives the adversary an oracle GEN that on input ϕ returns $\mathcal{G}(\phi(K))$ if $b = 1$ and $R(\phi(K))$ if $b = 0$. This is of course easily obtained from a Φ-RKA PRF. We call the new second property Φ-ICR (Identity-Collision-Resistant); this means that for a hidden key K, it is hard for the adversary to find $\phi \in \Phi$ such that $\phi(K) \neq K$ yet $\mathcal{G}(\phi(K)) = \mathcal{G}(K)$. At first it might seem this follows from Φ-RKA security but Proposition 2 shows it does not. However Proposition 3 shows how to build a PRG that is both Φ-RKA

and Φ-ICR secure from a Φ-RKA PRF with an identity key fingerprint, without assuming Φ is claw-free.

We build our Φ-RKA secure signature scheme from this PRG $\mathcal{G}$ and a base (normal secure) signature scheme, as follows. The secret key of our new signature scheme is a key K for the PRG. The output of the PRG on input K, $\mathcal{G}(K)$, is used as randomness to run the key-generation algorithm $\mathcal{K}$ of the base signature scheme, yielding a public key pk which becomes the public key of our scheme, and the corresponding secret key which is discarded. (Recall the secret key of the new scheme is the PRG key K.) To sign a message m under K, run $\mathcal{G}$ on K to get coins for $\mathcal{K}$, run the latter with these coins to get pk, sk and finally sign m under sk with the base signature scheme. Verification is just as in the base signature scheme.

For the proof we must construct an adversary B breaking the Φ-RKA security of $\mathcal{G}$ given an adversary A breaking the Φ-RKA security of our signature scheme. B thinks of the key K underlying its game as the secret key for our signature scheme and then runs A. When A makes SIGN query ϕ, m, adversary B will call its GEN oracle on ϕ and use the result as coins for $\mathcal{K}$ to get a secret key under which it then signs m for A. Eventually A outputs a forgery attempt m, σ. The assumed security of the base signature scheme will make it unlikely that A's forgery is a winning one when GEN is underlain by a random function. So B would like to test if A's forgery was a winning one, outputting 1 if so and 0 otherwise, to win its game. The difficulty is that it cannot test this because, not knowing K, it cannot test whether or not A made a SIGN query ϕ, m with $\phi(K) = K$. The Φ-ICR property of $\mathcal{G}$ comes to the rescue, telling us that whether or not $\phi(K) = K$ may be determined by whether or not the outputs of $\mathcal{G}$ on these two inputs, which B *does* have, are the same.

This sketch still pushes under the rug several subtle details which are dealt with in the full proof of Theorem 5, to be found in the full version of this paper [4].

3 Preliminaries

NOTATION. For sets X, Y, Z let $\mathsf{Fun}(X, Y)$ be the set of all functions mapping X to Y, and let $\mathsf{FF}(X, Y, Z) = \mathsf{Fun}(X \times Y, Z)$. The empty string is denoted ε. 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 $a_1, \ldots, a_n$ are strings then $a_1 \,\|\, \cdots \,\|\, a_n$ denotes their concatenation. If S is a set then $|S|$ denotes its size and $s \xleftarrow{\$} S$ the operation of picking a random element of S and calling it s. We say that a real-valued function on the integers is *negligible* if it vanishes faster than the inverse of any polynomial.

ALGORITHMS. Unless otherwise indicated, an algorithm is PT (Polynomial Time) and may be randomized. An adversary is an algorithm. If A is an algorithm and $\mathbf{x}$ is a vector then $A(\mathbf{x})$ denotes the vector $(A(\mathbf{x}[1]), \ldots, A(\mathbf{x}[|\mathbf{x}|]))$. By $y \leftarrow A(x_1, x_2, \ldots; r)$ we denote the operation of running A on inputs $x_1, x_2, \ldots$

and coins $r \in \{0,1\}^*$. We denote by $y \leftarrow\!\!\$\ A(x_1, x_2, \ldots)$ the operation of picking r at random and letting $y \leftarrow A(x_1, x_2, \ldots; r)$. We denote by $[A(x_1, x_2, \ldots)]$ the set of all possible outputs of A on inputs $x_1, x_2, \ldots$. We denote by $k \in \mathbb{N}$ the security parameter and by 1^k its unary encoding. It is assumed that the length of the output of any algorithm A depends only on the lengths of its inputs. In particular we can associate to single-input algorithm A its *output length* ℓ satisfying $|A(x)| = \ell(|x|)$ for all x. If A, B are algorithms then $A \,\|\, B$ denotes the algorithm that on any input x returns $A(x) \,\|\, B(x)$.

Games. Some of our definitions and proofs are expressed via code-based games [8]. Recall that such a game consists of an Initialize procedure, procedures to respond to adversary oracle queries and a Finalize procedure. A game G is executed with an adversary A as follows. First, Initialize executes on input 1^k and its output is the input to A. Then A executes, its oracle queries being answered by the corresponding procedures of G. When A terminates, its output becomes the input to the Finalize procedure. The output of the latter, denoted G^A, is called the output of the game. We let "$\mathrm{G}^A \Rightarrow d$" denote the event that this game output takes value d. If Finalize is absent it is understood to be the identity function, so the game output is the adversary output. Boolean flags are assumed initialized to false.

4 Classes of RKDFs and RKA-PRFs

Classes of RKDFs. In [5], a class Φ of related-key deriving functions (RKDFs) is a finite set of functions, all with the same domain and range. Our more general, asymptotic treatment requires extending this, in particular to allow the functions to depend on public parameters of the scheme. For us a *class* $\Phi = (\mathcal{P}, \mathcal{Q})$ of RKDFs, also called a RKA specification, is a pair of algorithms, the second deterministic. On input 1^k, parameter generation algorithm $\mathcal{P}$ produces parameters π. On input π, a key K and a description ϕ of an RKD function, the evaluation algorithm $\mathcal{Q}$ returns either a modified key or $\perp$. We require that for all ϕ, π, either $\mathcal{Q}(\pi, K, \phi) = \perp$ for all K or for no K. We let $\Phi_{\pi,\phi}(\cdot) = \mathcal{Q}(\pi, \cdot, \phi)$. We require that Φ always includes the identity function. (Formally, there is a special symbol id such that $\Phi_{\pi,\mathsf{id}}(K) = K$ for all K, π. This is to ensure that Φ-RKA security always implies normal security.) We let ID be the class consisting of only the identity function, so that ID-RKA security will be normal security.

A scheme (regardless of the primitive) is a tuple $(\overline{\mathcal{P}}, \cdots)$ of algorithms, the first of which is a parameter generation algorithm that on input 1^k returns a string. If ℓ is the output length of $\mathcal{P}$, we say that $\Phi = (\mathcal{P}, \mathcal{Q})$ is *compatible* with the scheme if the string formed by the first $\ell(k)$ bits of the output of $\overline{\mathcal{P}}(1^k)$ is distributed identically to the output of $\mathcal{P}(1^k)$ for all $k \in \mathbb{N}$. This is done so that, in constructing one Φ-RKA primitive from another, we can extend the parameters of the constructed scheme beyond those of the original one without changing the class of RKDFs.

We say that $\Phi = (\mathcal{P}, \mathcal{Q})$ is *claw-free* if $\phi \neq \phi'$ implies $\mathcal{Q}(\pi, K, \phi) \neq \mathcal{Q}(\pi, K, \phi')$ (or both values are $\perp$) for all π, K. This property has been assumed almost

proc INITIALIZE // PRF	proc INITIALIZE // IDFP
$\pi \leftarrow\!\!\$\ \mathcal{P}(1^k)$; $K \leftarrow\!\!\$\ \mathcal{K}(\pi)$	$\pi \leftarrow\!\!\$\ \mathcal{P}(1^k)$
$b \leftarrow\!\!\$\ \{0,1\}$	$K \leftarrow\!\!\$\ \mathcal{K}(\pi)$
Return π	$\mathbf{w} \leftarrow\!\!\$\ \mathsf{IKfp}(\pi)$
	Return $\pi, \mathbf{w}$
proc FN(ϕ, x) // PRF	
$K' \leftarrow \Phi_{\pi,\phi}(K)$	proc FN(ϕ) // IDFP
If $K' = \perp$ then return $\perp$	$K' \leftarrow \Phi_{\pi,\phi}(K)$
If $b = 1$ then	If $(K' = \perp)$ then return $\perp$
$T[K', x] \leftarrow \mathcal{F}(\pi, K', x)$	If $(K' \neq K)$ then
If $b = 0$ and $T[K', x] = \perp$ then	If $(\mathcal{F}(K', \mathbf{w}) = \mathcal{F}(K, \mathbf{w}))$ then
$T[K', x] \leftarrow\!\!\$\ \mathsf{Rng}(\pi)$	WIN $\leftarrow$ true
Return $T[K', x]$	Return $\mathcal{F}(K', \mathbf{w})$
proc FINALIZE(b') // PRF	proc FINALIZE() // IDFP
Return $(b = b')$	Return WIN

Fig. 2. Games defining Φ-RKA PRF security and Φ-IDFP security of function family $\mathcal{FF} = (\mathcal{P}, \mathcal{K}, \mathcal{F})$ having range $\mathsf{Rng}(\cdot)$.

ubiquitously in previous work [5, 28, 20, 3] because of the technical difficulties created by its absence, but its assumption is in fact quite restrictive since many natural classes do not have it. We are able to remove this assumption and provide constructs secure even for non-claw-free classes via new technical approaches. We let **CF** be the set of all Φ that are claw-free.

The class $\Phi^{\mathsf{const}} = (\mathcal{P}, \mathcal{Q}^{\mathsf{const}})$ of constant functions associated to class $\Phi = (\mathcal{P}, \mathcal{Q})$ is defined by $\Phi^{\mathsf{const}}_{\pi,a}(K) = a$ for all $K, a \in \{0,1\}^*$ and all π. The union $\Phi^1 \cup \Phi^2 = (\mathcal{P}, \mathcal{Q})$ of classes $\Phi^1 = (\mathcal{P}, \mathcal{Q}^1)$ and $\Phi^2 = (\mathcal{P}, \mathcal{Q}^2)$ is defined by having $\mathcal{Q}(\pi, K, \phi)$ parse ϕ as $i \,\|\, \phi^*$ for $i \in \{1, 2\}$ and return $\mathcal{Q}^i(\pi, K, \phi^*)$.

DISCUSSION. In a non-asymptotic treatment, there is no formal line between "secure" and "insecure." This makes it unclear how to rigorously define the sets **RKA**[P]. Lead, accordingly, to pursue an asymptotic treatment, we introduce parameter dependence; this allows us to capture constructs in the literature [28, 3] where RKDFs are defined over a group that is now parameter-dependent rather than fixed. (We note that even in the non-asymptotic case, a treatment like ours is needed to capture constructs in [28] relying on a RSA group defined by random primes. This issue is glossed over in [28].) A dividend of our treatment is a separation between an RKDF and its encoding, the latter being what an adversary actually queries, another issue glossed over in previous work.

FUNCTION FAMILIES. A function family $\mathcal{FF} = (\mathcal{P}, \mathcal{K}, \mathcal{F})$ consists of a parameter generator, a key generator, and an evaluator, the last deterministic. For each $k \in \mathbb{N}$ and $\pi \in [\mathcal{P}(1^k)]$, the scheme also defines PT decidable and sampleable sets $\mathsf{Dom}(\pi)$ and $\mathsf{Rng}(\pi)$ such that $\mathcal{F}(\pi, K, \cdot)$ maps elements of $\mathsf{Dom}(\pi)$ to $\mathsf{Rng}(\pi)$. We assume there are polynomials d, l, called the input and output lengths, respectively, such that $\mathsf{Dom}(\pi) \subseteq \{0,1\}^{d(k)}$ and $\mathsf{Rng}(\pi) \subseteq \{0,1\}^{l(k)}$.

Unless otherwise indicated we assume $\mathsf{Rng}(\pi) = \{0,1\}^{l(k)}$ and $l(k) = \omega(\log(k))$ and $|\mathsf{Dom}(\pi)| \geq 2^k$ for all $\pi \in [\mathcal{P}(1^k)]$ and all $k \in \mathbb{N}$.

RKA-PRFs. Let $\mathcal{FF} = (\mathcal{P}, \mathcal{K}, \mathcal{F})$ be a function family as above. Game PRF of Fig. 2 is associated to $\mathcal{FF}$ and a RKA specification Φ that is compatible with $\mathcal{FF}$. Let $\mathbf{Adv}^{\text{prf-rka}}_{\mathcal{FF},A,\Phi}(k)$ equal $2\Pr[\text{PRF}^A \Rightarrow \mathsf{true}] - 1$ when the game has input 1^k. We say $\mathcal{FF}$ is Φ-RKA secure if this advantage function is negligible.

Identity key fingerprints. An identity key fingerprint function with vector length $v(\cdot)$ for $\mathcal{FF} = (\mathcal{P}, \mathcal{K}, \mathcal{F})$ is an algorithm IKfp that for every $\pi \in [\mathcal{P}(1^k)]$ and every $k \in \mathbb{N}$ returns, on input π, a $v(k)$-vector over $\mathsf{Dom}(\pi)$ all of whose coordinates are distinct. Game IDFP of Fig. 2 is associated to $\mathcal{FF}$ and a RKA specification $\Phi = (\mathcal{P}, \mathcal{Q})$ that is compatible with $\mathcal{FF}$. Let $\mathbf{Adv}^{\text{idfp}}_{\mathcal{FF},A,\Phi}(k)$ equal $\Pr[\text{IDFP}^A \Rightarrow \mathsf{true}]$ when the game has input 1^k. We say $\mathcal{FF}$ is Φ-IDFP secure if this advantage function is negligible.

The key fingerprint notion of [3] can be seen as allowing statistical disambiguation of any pair of keys. They showed that the Naor-Reingold PRF NR had such a fingerprint, but in general, it does not seem common. Interestingly, their own Φ-RKA PRFs, which build on NR, are not known to have such a fingerprint. Our relaxation can be seen as asking for computational disambiguation of the original key from other keys, and ends up being much easier to achieve. In particular, such fingerprints exist for the constructs of [3]. This is a consequence of something more general, namely that any Φ-RKA secure PRF with large enough range is Φ-IDFP secure if Φ is claw-free, using *any* point in the domain functioning as the fingerprint. This is formalized by Proposition 1 below, with a proof in [4]. Φ-IDFP security for the constructs of [3] follows as the Φ they use is claw-free.

Proposition 1. *Suppose Φ is claw-free and $\mathcal{FF}$ is a Φ-RKA secure PRF with associated domain $\mathsf{Dom}(\cdot)$ and super-polynomial size range $\mathsf{Rng}(\cdot)$. Let IKfp be any algorithm that on input π returns a 1-vector over $\mathsf{Dom}(\pi)$. Then $\mathcal{FF}$ is Φ-IDFP secure.*

In practice Φ-IDFP security seems like a mild assumption even when Φ is not claw-free. A vector of a few, distinct domain points ought to be a suitable fingerprint for any practical blockcipher. This does not follow from a standard assumption on it such as PRF, but is consistent with properties assumed by cryptanalysts and can be proved in the ideal cipher model.

Φ-IDFP security of given Φ-RKA PRFs, even for non-claw-free Φ, will be important in the constructions underlying our containment results, and we make it a default assumption on a Φ-RKA PRF. The above shows that this is a mild and reasonable assumption.

RKA sets. We say that an RKA specification $\Phi = (\mathcal{P}, \mathcal{Q})$ is achievable for the primitive PRF if there exists a Φ-RKA and Φ-IDFP secure PRF that is compatible with Φ. We let $\mathbf{RKA}[\mathsf{PRF}]$ be the set of all Φ that are achievable for PRF.

WHAT CAN ATTACKS MODIFY? We view the system as a whole as having the following components: algorithms (code), parameters, public keys (if any) and secret keys. Of these, our convention is that only secret keys are subject to RKAs. This is not the only possible model, nor is it necessarily the most realistic if considering tampering attacks in practice, but it is a clear and interesting one with some justification. Parameters are systemwide, meaning fixed beforehand and independent of users, and may, in an implementation, be part of the algorithm code. Public keys are accompanied by certificates under a CA public key that is in the parameters, so if parameters are safe, so are public keys. This leaves secret keys as the main target. One consequence of this is that in a public key setting the attack is only on the holder of the secret key, meaning the signer for signatures and the receiver for encryption, while in the symmetric setting, both sender and receiver are under attack, making this setting more complicated.

We could consider attacks on public keys, but these are effectively attacks on parameters. Furthermore the only way for them to succeed is to modify the CA public key in the parameters in a rather special way, replacing it by some other key under which the attack produces signatures for the modified public key. "Natural" attacks caused by fault-injection are unlikely to do this, further supporting our convention of confining attacks to secret keys.

5 ICR PRGs: A tool in our constructions

We will be using Φ-RKA PRFs to build Φ-RKA instances of many other primitives. An important technical difficulty will be to avoid assuming Φ is claw-free. A tool we introduce and use for this purpose is a Φ-RKA PRG satisfying a weak form of collision-resistance under RKA that we call Φ-ICR. In this section we define this primitive and show how to achieve it based on a Φ-RKA and Φ-IDFP secure PRF.

RKA PRGs. A PRG $\mathcal{PRG} = (\mathcal{P}, \mathcal{K}, \mathcal{G}, r)$ is specified by a parameter generation algorithm, a key generation algorithm, an evaluation algorithm and an output length $r(\cdot)$. Game PRG of Fig. 3 is associated to $\mathcal{PRG}$ and an RKA specification Φ that is compatible with $\mathcal{PRG}$. Let $\mathbf{Adv}^{\mathrm{prg}}_{\mathcal{PRG},A,\Phi}(k) = 2\Pr[\mathrm{PRG}^A \Rightarrow \mathsf{true}] - 1$ when the game has input 1^k. We say $\mathcal{PRG}$ is Φ-RKA secure if this advantage function is negligible for all A.

We clarify that unlike a normal PRG [12], we don't require a Φ-RKA PRG to be length extending, meaning that outputs need not be longer than inputs. If one does want a length extending Φ-RKA PRG (we won't) one can get it by applying a normal-secure PRG to the output of a given Φ-RKA PRG.

ICR. We define and use a weak form of collision-resistance for PRGs which requires that the adversary be unable to find ϕ so that $\Phi_{\pi,\phi}(K) \neq K$ yet $\mathcal{G}(\Phi_{\pi,\phi}(K)) = \mathcal{G}(K)$. Game ICR of Fig. 3 is associated to $\mathcal{PRG}$ and a RKA specification Φ that is compatible with $\mathcal{PRG}$. Let $\mathbf{Adv}^{\mathrm{icr}}_{\mathcal{PRG},C,\Phi}(k)$ equal $2\Pr[\mathrm{ICR}^C \Rightarrow \mathsf{true}] - 1$ when the game has input 1^k. We say $\mathcal{PRG}$ is Φ-ICR (Identity-Collision-Resistant) secure if this advantage function is negligible.

proc INITIALIZE // PRG	proc INITIALIZE // ICR
$\pi \leftarrow\!\!\$\ \mathcal{P}(1^k)$	$\pi \leftarrow\!\!\$\ \mathcal{P}(1^k)$
$K \leftarrow\!\!\$\ \mathcal{K}(\pi)\ ;\ b \leftarrow\!\!\$\ \{0,1\}$	$K \leftarrow\!\!\$\ \mathcal{K}(\pi)\ ;\ T_0 \leftarrow \mathcal{G}(\pi, K)$
Return π	Return π
proc GEN(ϕ) // PRG	proc GEN(ϕ) // ICR
$K' \leftarrow \Phi_{\pi,\phi}(K)$	$K' \leftarrow \Phi_{\pi,\phi}(K)$
If $K' = \perp$ then return $\perp$	If $K' = \perp$ then return $\perp$
If $T[K'] = \perp$ then	$S \leftarrow \mathcal{G}(\pi, K')$
If $b = 1$ then $T[K'] \leftarrow \mathcal{G}(\pi, K')$	If $((S = T_0) \wedge (K \neq K'))$ then WIN $\leftarrow$ true
Else $T[K'] \leftarrow\!\!\$\ \{0,1\}^{r(k)}$	Return S
Return $T[K']$	proc FINALIZE() // ICR
proc FINALIZE(b') // PRG	Return WIN
Return $(b = b')$	

Fig. 3. Games defining Φ-RKA security and identity-collision-resistance for PRG $\mathcal{PRG} = (\mathcal{P}, \mathcal{K}, \mathcal{G}, r)$.

DOES RKA SECURITY IMPLY ICR SECURITY? At first glance it would seem that if a PRG $\mathcal{PRG} = (\mathcal{P}, \mathcal{K}, \mathcal{G}, r)$ is Φ-RKA secure then it is also Φ-ICR secure. Indeed, suppose an adversary has ϕ such that $\Phi_{\pi,\phi}(K) \neq K$ yet $\mathcal{G}(\Phi_{\pi,\phi}(K)) = \mathcal{G}(K)$. Let it query $R_0 \leftarrow$ GEN(id) and $R_1 \leftarrow$ GEN(ϕ) and return 1 if $R_0 = R_1$ and 0 otherwise. In the real ($b = 1$) case R_0, R_1 are equal but in the random ($b = 0$) case they would appear very unlikely to be equal, so that that this strategy would appear to have high advantage in breaking the Φ-RKA security of $\mathcal{PRG}$. The catch is in our starting assumption, which made it appear that $\Phi_{\pi,\phi}(K) \neq K$ yet $\mathcal{G}(\Phi_{\pi,\phi}(K)) = \mathcal{G}(K)$ was an absolute fact, true both for $b = 0$ and $b = 1$. If $\Phi_{\pi,\phi}(K)$ and K are different in the real game but equal in the random game, the adversary sees an output collision in both cases and its advantage disappears. Can this actually happen? It can, and indeed the claim (that Φ-RKA security implies Φ-ICR security) is actually false:

Proposition 2. *Suppose there exists a normal-secure PRG $\overline{\mathcal{PRG}} = (\overline{\mathcal{P}}, \overline{\mathcal{K}}, \overline{\mathcal{G}}, r)$ with $r(\cdot) = \omega(\log(\cdot))$. Then there exists a PRG $\mathcal{PRG} = (\overline{\mathcal{P}}, \mathcal{K}, \mathcal{G}, r)$ and a class Φ such that $\mathcal{PRG}$ is Φ-RKA secure but $\mathcal{PRG}$ is not Φ-ICR secure.*

A proof is in [4]. Briefly, the constructed PRG $\mathcal{PRG}$ adds a redundant bit to the seed of $\overline{\mathcal{PRG}}$ so that seeds differing only in their first bits yield the same outputs, meaning create non-trivial collisions. But Φ is crafted so that that its members deviate from the identity function only in the real game, so that output collisions appear just as often in both cases but in the real game they are non-trivial while in the random game they are trivial.

CONSTRUCTION. We saw above that not all Φ-RKA PRGs are Φ-ICR secure. Our containments will rely crucially on ones that are. We obtain them from Φ-RKA PRFs that have key fingerprints for the identity function:

proc INITIALIZE // Sig
$\pi \leftarrow\!\!\$\ \mathcal{P}(1^k)$; $M \leftarrow \emptyset$
$(vk, sk) \leftarrow\!\!\$\ \mathcal{K}(\pi)$
Return (π, vk)

proc SIGN(ϕ, m) // Sig
$sk' \leftarrow \Phi_{\pi,\phi}(sk)$
If $sk' = \bot$ then return $\bot$
If $sk' = sk$ then $M \leftarrow M \cup \{m\}$
Return $\sigma \leftarrow\!\!\$\ \mathcal{S}(\pi, sk', m)$

proc FINALIZE(m, σ) // Sig
Return $((\mathcal{V}(\pi, vk, m, \sigma) = 1) \wedge (m \notin M))$

proc FINALIZE(b') // IBE
Return $(b = b')$

proc INITIALIZE // IBE
$\pi \leftarrow\!\!\$\ \mathcal{P}(1^k)$; $(mpk, msk) \leftarrow\!\!\$\ \mathcal{M}(\pi)$
$b \leftarrow\!\!\$\ \{0,1\}$; $id^* \leftarrow \bot$; $S \leftarrow \emptyset$
Return (π, mpk)

proc KD(ϕ, id) // IBE
$msk' \leftarrow \Phi_{\pi,\phi}(msk)$
If $msk' = \bot$ then return $\bot$
If $msk' = msk$ then $S \leftarrow S \cup \{id\}$
If $(msk' = msk) \wedge (id = id^*)$ then return $\bot$
Return $dk \leftarrow\!\!\$\ \mathcal{K}(\pi, mpk, msk', id)$

proc LR(id, m_0, m_1) // IBE
If $|m_0| \neq |m_1|$ then return $\bot$
$id^* \leftarrow id$; If $id^* \in S$ then return $\bot$
Return $C \leftarrow\!\!\$\ \mathcal{E}(\pi, mpk, id, m_b)$

proc FINALIZE(b') // IBE
Return $((b = b') \wedge (id^* \notin S))$

Fig. 4. Games defining Φ-RKA security for primitives Sig, IBE.

Proposition 3. *Let $\mathcal{FF} = (\mathcal{P}, \mathcal{K}, \mathcal{F})$ be a Φ-RKA PRF with output length l. Let IKfp be a Φ-IDFP secure identity key fingerprint function for $\mathcal{FF}$ with vector length v. Let $r = lv$ and let $\overline{\mathcal{K}}$, on input $\pi \,\|\, \mathbf{w}$, return $\mathcal{K}(\pi)$. Define PRG $\mathcal{PRG} = (\mathcal{P} \,\|\, \mathsf{IKfp}, \overline{\mathcal{K}}, \mathcal{G}, r)$ via*

$$\mathcal{G}(\pi \,\|\, \mathbf{w}, K) = \mathcal{F}(\pi, K, \mathbf{w}[1]) \,\|\, \cdots \,\|\, \mathcal{F}(\pi, K, \mathbf{w}[|\mathbf{w}|]) \ .$$

Then $\mathcal{PRG}$ is Φ-RKA secure and Φ-ICR secure.

6 Relations

We first present a containment and a non-containment related to Sig. Then we turn to IBE-related results. Other results can be found in [4].

SIGNATURES. A signature scheme $\mathcal{DS} = (\mathcal{P}, \mathcal{K}, \mathcal{S}, \mathcal{V})$ is specified as usual by its parameter generation, key generation, signing and verifying algorithms. Game Sig of Fig. 4 is associated to $\mathcal{DS}$ and an RKA specification Φ that is compatible with $\mathcal{DS}$. Let $\mathbf{Adv}^{\text{sig-rka}}_{\mathcal{DS},A,\Phi}(k) = \Pr[\text{Sig}^A \Rightarrow \mathsf{true}]$ when the game has input 1^k. We say $\mathcal{DS}$ is Φ-RKA secure if this advantage function is negligible. Normal security of a signature scheme is recovered by considering Φ that contains only the identity function. One feature of the definition worth highlighting is the way we decide which messages are not legitimate forgeries. They are the ones signed with the real key sk, which means that oracle SIGN needs to check when a related key equals the real one and record the corresponding message, which is a source of challenges in reduction-based proofs.

ATTACKS. In [4] we present an attack, adapted from [6, 19], that shows that there are some (quite simple) Φ such that *no* signature scheme is Φ-RKA secure,

meaning $\Phi \notin \mathbf{RKA}[\mathsf{Sig}]$. This indicates that the set $\mathbf{RKA}[\mathsf{Sig}]$ is non-trivial. Similar attacks can be presented for other primitives.

FROM Φ-RKA PRGS TO Φ-RKA SIGNATURES. We will prove containments of the form $\mathbf{RKA}[\mathsf{PRF}] \subseteq \mathbf{RKA}[\mathsf{P}]$ by proving $\mathbf{RKA}[\mathsf{PRG}] \subseteq \mathbf{RKA}[\mathsf{P}]$ and exploiting the fact that $\mathbf{RKA}[\mathsf{PRF}] \subseteq \mathbf{RKA}[\mathsf{PRG}]$.

We start with a Φ-RKA PRG $\mathcal{PRG} = (\mathcal{P}, \mathcal{K}, \mathcal{G}, r)$ and a normal-secure signature scheme $\overline{\mathcal{DS}} = (\overline{\mathcal{P}}, \overline{\mathcal{K}}, \overline{\mathcal{S}}, \overline{\mathcal{V}})$ such that $r(\cdot)$ is the number of coins used by $\overline{\mathcal{K}}$. We now build another signature scheme $\mathcal{DS} = (\mathcal{P} \,\|\, \overline{\mathcal{P}}, \mathcal{K}', \mathcal{S}, \mathcal{V})$ as follows:

1. Parameters: Parameters for $\mathcal{DS}$ are the concatenation $\pi \,\|\, \overline{\pi}$ of independently generated parameters for $\mathcal{PRG}$ and $\overline{\mathcal{DS}}$.
2. Keys: Pick a random seed $K \leftarrow\!\!\$\, \mathcal{K}(\pi)$ and let $(\overline{vk}, \overline{sk}) \leftarrow \overline{\mathcal{K}}(\overline{\pi}; \mathcal{G}(K))$ be the result of generating verifying and signing keys with coins $\mathcal{G}(K)$. The new signing key is K and the verifying key remains $\overline{vk}$. (Key $\overline{sk}$ is discarded.)
3. Signing: To sign message m with signing key K, recompute $(\overline{vk}, \overline{sk}) \leftarrow \overline{\mathcal{K}}(\overline{\pi}; \mathcal{G}(K))$ and then sign m under $\overline{\mathcal{S}}$ using $\overline{sk}$.
4. Verifying: Verify that σ is a base scheme signature of m under $\overline{vk}$ using $\overline{\mathcal{V}}$.

Signature scheme $\mathcal{DS}$ remains compatible with Φ since the parameters of $\mathcal{PRG}$ prefix those of $\mathcal{DS}$.

We want $\mathcal{DS}$ to inherit the Φ-RKA security of $\mathcal{PRG}$. In fact we will show more, namely that $\mathcal{DS}$ is $(\Phi \cup \Phi_{\mathrm{c}})$-RKA secure where Φ_{c} is the class of constant RKDFs associated to Φ. The intuition is deceptively simple. A signing query ϕ, m of an adversary A attacking $\mathcal{DS}$ results in a signature of m under what is effectively a fresh signing key, since it is generated using coins $\mathcal{G}(\phi(K))$ that are computationally independent of $\mathcal{G}(K)$ due to the assumed Φ-RKA security of the PRG. These can accordingly be simulated without access to K. On the other hand, signing queries in which ϕ is a constant function may be directly simulated. The first difficulty is that the adversary attacking the Φ-RKA security of $\mathcal{PRG}$ that we must build needs to know when A succeeds, and for this it needs to know when a derived seed equals the real one, and it is unclear how to do this without knowing the real seed. The second difficulty is that a queried constant might equal the key. We take an incremental approach to showing how these difficulties are resolved, beginning by assuming Φ is claw-free, which makes the first difficulty vanish:

Theorem 4. *Let signature scheme $\mathcal{DS} = (\mathcal{P} \,\|\, \overline{\mathcal{P}}, \mathcal{K}', \mathcal{S}, \mathcal{V})$ be constructed as above from Φ-RKA PRG $\mathcal{PRG} = (\mathcal{P}, \mathcal{K}, \mathcal{G}, r)$ and normal-secure signature scheme $\overline{\mathcal{DS}} = (\overline{\mathcal{P}}, \overline{\mathcal{K}}, \overline{\mathcal{S}}, \overline{\mathcal{V}})$ and assume Φ is claw-free. Then $\mathcal{DS}$ is $(\Phi \cup \Phi_{\mathrm{c}})$-RKA secure.*

A proof of Theorem 4 is in [4], and the intuition was discussed in Section 2. This result, however, is weaker than we would like, for, as we have already said, many interesting classes are not claw-free. Also, this result fails to prove $\mathbf{RKA}[\mathsf{PRF}] \subseteq \mathbf{RKA}[\mathsf{Sig}]$ since the first set may contain Φ that are not claw-free. To address this we show that the claw-freeness assumption on Φ can be replaced by the assumption that $\mathcal{PRG}$ is Φ-ICR secure:

Theorem 5. *Let signature scheme* $\mathcal{DS} = (\mathcal{P} \,\|\, \overline{\mathcal{P}}, \mathcal{K}', \mathcal{S}, \mathcal{V})$ *be constructed as above from* Φ*-RKA secure and* Φ*-ICR secure PRG* $\mathcal{PRG} = (\mathcal{P}, \mathcal{K}, \mathcal{G}, r)$ *and normal-secure signature scheme* $\overline{\mathcal{DS}} = (\overline{\mathcal{P}}, \overline{\mathcal{K}}, \overline{\mathcal{S}}, \overline{\mathcal{V}})$*. Then* $\mathcal{DS}$ *is* $(\Phi \cup \Phi_{\mathrm{c}})$*-RKA secure.*

A proof of Theorem 5 is in [4]. Proposition 3 says we can get the PRGs we want from Φ-RKA PRFs so Theorem 5 establishes the containment $\mathbf{RKA}[\mathsf{PRF}] \subseteq \mathbf{RKA}[\mathsf{Sig}]$. (Theorem 4 only established $\mathbf{RKA}[\mathsf{PRF}] \cap \mathbf{CF} \subseteq \mathbf{RKA}[\mathsf{Sig}] \cap \mathbf{CF}$.)

Our construction has the advantage that the verification process as well as the form of the signatures and public key are unchanged. This means it has minimal impact on software, making it easier to deploy than a totally new scheme. Signing in the scheme now involves evaluation of a Φ-RKA-PRG but this can be made cheap via an AES-based instantiation. However, signing also involves running the key-generation algorithm $\overline{\mathcal{K}}$ of the base scheme which might be expensive.

This construction also meets a stronger notion of Φ-RKA security where the adversary cannot even forge a signature relative to the public keys associated with the derived secret keys. We elaborate on this in [4].

Some base signature schemes lend themselves naturally and directly to immunization against RKAs via Φ-RKA PRFs. This is true for the binary-tree, one-time signature based scheme discussed in [21], where the secret key is already that of a PRF. If the latter is Φ-RKA secure we can show the signature scheme (unmodified) is too, and moreover also meets the strong version of the definition alluded to above. See [4].

Separating Φ-RKA PRFs from Φ-RKA signatures. Having just shown that $\mathbf{RKA}[\mathsf{PRF}] \subseteq \mathbf{RKA}[\mathsf{Sig}]$ it is natural to ask whether the converse is true as well, meaning whether the sets are equal. The answer is no, so $\mathbf{RKA}[\mathsf{Sig}] \not\subseteq \mathbf{RKA}[\mathsf{PRF}]$. The interpretation is that there exist Φ such that there exist Φ-RKA secure signatures, but there are *no* Φ-RKA PRFs. An example is when $\Phi = \Phi_{\mathrm{c}}$ is the set of constant functions. Theorem 4 implies that there exists a Φ_{c}-RKA secure signature scheme by setting $\Phi = \emptyset$ in the theorem, so that $\mathcal{PRG}$ need only be a normal-secure PRG. But attacks from [5] show that no PRF can be Φ_{c}-RKA secure. Thus, this separation is quite easily obtained. In [4] we present others which are more interesting. This separation motivates finding other avenues to Φ-RKA signatures. Below we will show that IBE is one such avenue.

IBE. Our specification of an IBE scheme $\mathcal{IBE} = (\mathcal{P}, \mathcal{M}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ adds a parameter generation algorithm $\mathcal{P}$ that given 1^k returns parameters π on which the masterkey generation algorithm $\mathcal{M}$ runs to produce the master public key mpk and master secret key msk. The rest is as usual except that algorithms get π as an additional input. Game IBE of Fig. 4 is associated to $\mathcal{IBE}$ and an RKA specification $\Phi = (\mathcal{P}, \mathcal{Q})$ that is compatible with $\mathcal{IBE}$. An adversary is allowed only one query to LR. Let $\mathbf{Adv}^{\text{ibe-rka}}_{\mathcal{IBE},A,\Phi}(k)$ equal $2\Pr[\text{IBE}^A \Rightarrow \mathsf{true}] - 1$ when the game has input 1^k. We say $\mathcal{IBE}$ is Φ-RKA secure if this advantage function is negligible. Here the feature of the definition worth remarking on is that the adversary loses if it ever issues a query to KD that contains the challenge identity *and* derives the same master secret key. In [4] we show (1) that the standard Naor transform pre-

serves RKA security and thus $\mathbf{RKA}[\mathsf{IBE}] \subseteq \mathbf{RKA}[\mathsf{Sig}]$, and (2) that the BCHK transform [15] preserves RKA security and thus $\mathbf{RKA}[\mathsf{IBE}] \subseteq \mathbf{RKA}[\mathsf{PKE\text{-}CCA}]$.

Other relations. The remaining results and definitions from Fig. 1 are presented in [4].

Acknowledgments

We thank Susan Thomson, Martijn Stam, Pooya Farshim and the Asiacrypt 2011 reviewers for their comments and corrections. Mihir Bellare was supported in part by NSF grants CCF-0915675 and CNS-0904380. Work done while David Cash was at UCSD, supported in part by NSF grant CCF-0915675. Rachel Miller was supported in part by a DOD NDSEG Graduate Fellowship and NSF grant CCF-1018064.

References

1. M. Albrecht, P. Farshim, K. Paterson, and G. Watson. On cipher-dependent related-key attacks in the ideal-cipher model. Cryptology ePrint Archive, Report 2011/213, 2011. `http://eprint.iacr.org/`.
2. B. Applebaum, D. Harnik, and Y. Ishai. Semantic security under related-key attacks and applications. In A. C.-C. Yao, editor, *ICS 2011*. Tsinghua University Press, 2011.
3. M. Bellare and D. Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 666–684. Springer, Aug. 2010.
4. M. Bellare, D. Cash, and R. Miller. Cryptography secure against related-key attacks. Cryptology ePrint Archive, Report 2011/252, 2011. Full version of this paper, `http://eprint.iacr.org/2011/252`.
5. 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.
6. M. Bellare and T. Kohno. Hash function balance and its impact on birthday attacks. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 401–418. Springer, May 2004.
7. M. Bellare and S. K. Miner. A forward-secure digital signature scheme. In M. J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 431–448. Springer, Aug. 1999.
8. 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.
9. 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.
10. E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 513–525. Springer, Aug. 1997.

11. N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 520–537. Springer, Aug. 2010.
12. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.
13. 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.
14. D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, Apr. 2008.
15. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):915–942, 2006.
16. 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.
17. 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.
18. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In A. C.-C. Yao, editor, *ICS 2010*. Tsinghua University Press, 2010.
19. 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.
20. 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.
21. O. Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.
22. V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, Mar. 2011.
23. V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. Cryptology ePrint Archive, Report 2011/233, 2011. Full version of [22], `http://eprint.iacr.org/`.
24. S. Halevi and H. Krawczyk. Security under key-dependent inputs. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 466–475. ACM Press, Oct. 2007.
25. Y. Ishai, M. Prabhakaran, A. Sahai, and D. Wagner. Private circuits II: Keeping secrets in tamperable circuits. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327. Springer, May / June 2006.
26. Y. T. Kalai, B. Kanukurthi, and A. Sahai. Cryptography with tamperable and leaky memory. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 373–390. Springer, Aug. 2011.
27. 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.
28. 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.
29. M. Naor and O. Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.