RKA Security beyond the Linear Barrier: IBE, Encryption and Signatures

Mihir Bellare[1], Kenneth G. Paterson[2], and Susan Thomson[3]

[1] Department of Computer Science & Engineering, University of California San Diego. mihir@eng.ucsd.edu; cseweb.ucsd.edu/~mihir/
[2] Information Security Group, Royal Holloway, University of London. kenny.paterson@rhul.ac.uk; www.isg.rhul.ac.uk/~kp
[3] Information Security Group, Royal Holloway, University of London. s.thomson@rhul.ac.uk

Abstract. We provide a framework enabling the construction of IBE schemes that are secure under related-key attacks (RKAs). Specific instantiations of the framework yield RKA-secure IBE schemes for sets of related key derivation functions that are non-linear, thus overcoming a current barrier in RKA security. In particular, we obtain IBE schemes that are RKA secure for sets consisting of all affine functions and all polynomial functions of bounded degree. Based on this we obtain the first constructions of RKA-secure schemes for the same sets for the following primitives: CCA-secure public-key encryption, CCA-secure symmetric encryption and Signatures. All our results are in the standard model and hold under reasonable hardness assumptions.

1 Introduction

Related-key attacks (RKAs) were first conceived as tools for the cryptanalysis of blockciphers [22, 9]. However, the ability of attackers to modify keys stored in memory via tampering [13, 10] raises concerns that RKAs can actually be mounted in practice. The key could be an IBE master key, a signing key of a certificate authority, or a decryption key, making RKA security important for a wide variety of primitives.

Provably achieving security against RKAs, however, has proven extremely challenging. This paper aims to advance the theory with new feasibility results showing achievability of security under richer classes of attacks than previously known across a variety of primitives.

Contributions in brief. The primitive we target in this paper is IBE. RKA security for this primitive was defined by Bellare, Cash, and Miller [4]. As per the founding theoretical treatment of RKAs by Bellare and Kohno [5], the definition is parameterized by the class Φ of functions that the adversary is allowed to apply to the target key. (With no restrictions, security is unachievable.) For future reference we define a few relevant classes of functions over the space $\mathcal{S}$ of master keys. The set $\Phi^c = \{\phi_c\}_{c\in\mathcal{S}}$ with $\phi_c(s) = c$ is the set of constant functions. If $\mathcal{S}$ is a group under an operation $*$ then $\Phi^{\text{lin}} = \{\phi_a\}_{a\in\mathcal{S}}$ with $\phi_a(s) = a * s$ is

the class of linear functions. (Here $*$ could be multiplication or addition.) If $\mathcal{S}$ is a field we let $\Phi^{\mathrm{aff}} = \{\phi_{a,b}\}_{a,b\in\mathcal{S}}$ with $\phi_{a,b}(s) = as + b$ be the class of affine functions and $\Phi^{\mathrm{poly}(d)} = \{\phi_q\}_{q\in\mathcal{S}_d[x]}$ with $\phi_q(s) = q(s)$ the class of polynomial functions, where q ranges over the set $\mathcal{S}_d[x]$ of polynomials over $\mathcal{S}$ of degree at most d. RKA security increases and is a more ambitious target as we move from Φ^{lin} to Φ^{aff} to $\Phi^{\mathrm{poly}(d)}$.

The choice of IBE as a primitive is not arbitrary. First, IBE is seeing a lot of deployment, and compromise of the master secret key would cause widespread damage, so we are well motivated to protect it against side-channel attacks. Second, IBE was shown in [4] to be an enabling primitive in the RKA domain: achieving RKA-secure IBE for any class Φ immediately yields Φ-RKA-secure CCA-PKE (CCA-secure public-key encryption) and Sig (signature) schemes. These results were obtained by noting that the CHK [12] IBE-to-CCA-PKE transform and the Naor IBE-to-Sig transform both preserve RKA security. Thus, results for IBE would immediately have wide impact.

We begin by presenting attacks showing that existing IBE schemes such as those of Boneh-Franklin [14] and Waters [25] are not RKA secure, even for Φ^{lin}. This means we must seek new designs.

We present a framework for constructing RKA-secure IBE schemes. It is an adaptation of the framework of Bellare and Cash [3] that builds RKA-secure PRFs based on key-malleable PRFs and fingerprinting. Our framework has two corresponding components. First, we require a starting IBE scheme that has a key-malleability property relative to our target class Φ of related-key deriving functions. Second, we require the IBE scheme to support what we call collision-resistant identity renaming. We provide a simple and efficient way to transform any IBE scheme with these properties into one that is Φ-RKA secure.

To exploit the framework, we must find key-malleable IBE schemes. Somewhat paradoxically, we show that the very attack strategies that broke the RKA security of existing IBE schemes can be used to show that these schemes are Φ-key-malleable, not just for $\Phi = \Phi^{\mathrm{lin}}$ but even for $\Phi = \Phi^{\mathrm{aff}}$. We additionally show that these schemes support efficient collision-resistant identity renaming. As a consequence we obtain Φ^{aff}-RKA-secure IBE schemes based on the same assumptions used to prove standard IBE security of the base IBE schemes.

From the practical perspective, the attraction of these results is that our schemes modify the known ones in a very small and local way limited only to the way identities are hashed. They thus not only preserve the efficiency of the base schemes, but implementing them would require minimal and modular software changes, so that non-trivial RKA security may be added without much increase in cost. From the theoretical perspective, the step of importance here is to be able to achieve RKA security for non-linear functions, and this without extra computational assumptions. As we will see below, linear RKAs, meaning Φ^{lin}-RKA security, has so far been a barrier for most primitives.

However, we can go further, providing a $\Phi^{\mathrm{poly}(d)}$-RKA-secure IBE scheme. Our scheme is an extension of Waters' scheme [25]. The proof is under a q-type hardness assumption that we show holds in the generic group model. The

significance of this result is to show that for IBE we can go well beyond linear RKAs, something not known for PRFs.

As indicated above, we immediately get Φ-RKA-secure CCA-PKE and Sig schemes for any class Φ for which we obtained Φ-RKA-secure IBE schemes, and under the same assumptions. When the base IBE scheme has a further malleability property, the CCA-PKE scheme so obtained can be converted into a Φ-RKA-secure CCA-SE (CCA-secure symmetric encryption) scheme. This yields the first RKA secure schemes for the primitives Sig, CCA-PKE, and CCA-SE for non-linear RKAs, meaning beyond Φ^{lin}.

Background and context. The theoretical foundations of RKA security were laid by Bellare and Kohno [5], who treated the case of PRFs and PRPs. Research then expanded to consider other primitives [20, 2, 21, 4]. In particular, Bellare, Cash and Miller [4] provide a comprehensive treatment including strong definitions for many primitives and ways to transfer Φ-RKA security from one primitive to another.

RKA-security is finding applications beyond providing protection against tampering-based sidechannel attacks [19], including instantiating random oracles in higher-level protocols and improving efficiency [2, 1].

With regard to achieving security, early efforts were able to find PRFs with proven RKA security only for limited Φ or under very strong assumptions. Eventually, using new techniques, Bellare and Cash [3] were able to present DDH-based PRFs secure against linear RKAs ($\Phi = \Phi^{\text{lin}}$). But it is not clear how to take their techniques further to handle larger RKA sets Φ.

Fig. 1 summarizes the broad position. Primitives for which efforts have now been made to achieve RKA security include CPA-SE (CPA secure symmetric encryption), CCA-SE (CCA secure symmetric encryption), CCA-PKE (CCA secure public-key encryption[4]) Sig (Signatures), and IBE (CPA secure identity-based encryption). Schemes proven secure under a variety of assumptions have been provided. But the salient fact that stands out is that prior to our work, results were all for linear RKAs with the one exception of CPA-SE where a scheme secure against polynomial (and thus affine) RKAs was provided by [21].

In more detail, Bellare, Cash and Miller [4] show how to transfer RKA security from PRF to any other primitive, assuming an existing standard-secure instance of the primitive. Combining this with [3] yields DDH-based schemes secure against linear RKAs for all the primitives, indicated by a "[4]+[3]" table entry. Applebaum, Harnik and Ishai [2] present LPN and LWE-based CPA-SE schemes secure against linear RKAs. Wee [26] presents CCA-PKE secure schemes for linear RKAs. Goyal, O'Neill and Rao [21] gave a CPA-SE scheme secure against polynomial RKAs. (We note that their result statement should be amended to exclude constant RKD functions, for no symmetric primitive can be secure under these.) Wee [26] (based on a communication of Wichs) remarks that AMD codes [18] may be used to achieve RKA security for CCA-PKE, a

[4] RKAs are interesting for symmetric encryption already in the CPA case because encryption depends on the secret key, but for public-key encryption they are only interesting for the CCA case because encryption does not depend on the secret key.

Primitive	Linear	Affine	Polynomial
IBE	[4]+[3]	✓	✓
Sig	[4]+[3]	✓	✓
CCA-PKE	[26], [4]+[3]	✓	✓
CPA-SE	[2], [4]+[3]	[21]	[21]
CCA-SE	[4]+[3]	✓	✓*
PRF	[3]	–	–

Fig. 1. Rows are indexed by primitives. Columns are indexed by the class Φ of related-key derivation functions, $\Phi^{\mathrm{lin}}, \Phi^{\mathrm{aff}}$ and $\Phi^{\mathrm{poly}(d)}$ respectively. Entries indicate work achieving Φ-RKA security for the primitive in question. Checkmarks indicate results from this paper that bring many primitives all the way to security under polynomial RKAs in one step. The table only considers achieving the strong, adaptive notions of security from [4]; non-adaptively secure signature schemes for non-linear RKAs were provided in [21]. Note that symmetric key primitives cannot be RKA secure against constant RKD functions, so affine and polynomial RKA security for the last three rows is with respect to the RKD sets $\Phi^{\mathrm{aff}} \setminus \Phi^c$ and $\Phi^{\mathrm{poly}(d)} \setminus \Phi^c$. The "*" in the CCA-SE row is because our CCA-SE construction is insecure against RKD functions where the linear coefficient is zero, so does not achieve RKA security against the full set $\Phi^{\mathrm{poly}(d)} \setminus \Phi^c$. See the full version for details.

method that extends to other primitives including IBE (but not PRF), but with current constructions of these codes [18], the results continue to be restricted to linear RKAs. We note that we are interested in the stronger, adaptive versions of the definitions as given in [4], but non-adaptively secure signature schemes for non-linear RKAs were provided in [21].

In summary, a basic theoretical question that emerges is how to go beyond linear RKAs. A concrete target here is to bring other primitives to parity with CPA-SE by achieving security for affine and polynomial RKAs. Ideally, we would like approaches that are general, meaning each primitive does not have to be treated separately. As discussed above, we are able to reach these goals with IBE as a starting point.

A CLOSER LOOK. Informally, key-malleability means that user-level private keys obtained by running the IBE scheme's key derivation algorithm $\mathcal{K}$ using a modified master secret key $\phi(s)$ (where $\phi \in \Phi$ and $s \in \mathcal{S}$, the space of master secret keys) can alternatively be computed by running $\mathcal{K}$ using the original master secret key s, followed by a suitable transformation. A collision-resistant identity renaming transform maps identities from the to-be-constructed RKA-secure IBE scheme back into identities in the starting IBE scheme in such a way as to "separate" the sets of identities coming from different values of $\phi(s)$. By modifying the starting IBE scheme to use renamed identities instead of the original ones, we obtain a means to handle otherwise difficult key extraction queries in the RKA setting.

To show that the framework is applicable to the Boneh-Franklin [14] and Waters [25] IBE schemes with $\Phi = \Phi^{\text{aff}}$ (the space of master keys here is $\mathbb{Z}_p$), we exploit specific algebraic properties of the starting IBE schemes. In the Waters case, we obtain an efficient, Φ^{aff}-RKA-secure IBE scheme in the standard model, under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. In the Boneh-Franklin case, we obtain an efficient, Φ^{aff}-RKA-secure IBE scheme under the Bilinear Diffie-Hellman (BDH) assumption with more compact public keys at the expense of working in the Random Oracle Model. Going further, we exhibit a simple modification of the Waters scheme which allows us to handle related key attacks for $\Phi^{\text{poly}(d)}$, this being the set of polynomial functions of bounded degree d. This requires the inclusion of an extra $2d - 2$ elements in the master public key, and a modified, q-type hardness assumption. We show that this assumption holds in the generic group model.

Applying the results of [4] to these IBE schemes, we obtain the first constructions of RKA-secure CCA-PKE and signature schemes for Φ^{aff} and $\Phi^{\text{poly}(d)}$. Again, our schemes are efficient and our results hold in the standard model under reasonable hardness assumptions. The CCA-PKE schemes, being derived via the CHK transform [12], just involve the addition of a one-time signature and verification key to the IBE ciphertexts and so incur little additional overhead for RKA security. As an auxiliary result that improves on the corresponding result of [4], we show in the full version [6] that the more efficient MAC-based transform of [15, 12] can be used in place of the CHK transform. The signature schemes arise from the Naor trick, wherein identities are mapped to messages, IBE user private keys are used as signatures, and a trial encryption and decryption on a random plaintext are used to verify the correctness of a signature. This generic construction can often be improved by tweaking the verification procedure, and the same is true here: for example, for the Waters-based signature scheme, we can base security on the CDH assumption instead of DBDH, and can achieve more efficient verification. We stress that our signature schemes are provably unforgeable in a fully adaptive related-key setting, in contrast to the recently proposed signatures in [21].

Note that RKA-secure PRFs for sets Φ^{aff} and $\Phi^{\text{poly}(d)}$ cannot exist, since these sets contain constant functions, and we know that no PRF can be RKA-secure in this case [5]. Thus we are able to show stronger results for IBE, CCA-PKE and Sig than are possible for PRF. Also, although Bellare, Cash and Miller [4] showed that Φ-RKA security for PRF implies Φ-RKA security for Sig and CCA-PKE, the observation just made means we cannot use this result to get Φ^{aff} or $\Phi^{\text{poly}(d)}$ RKA-secure IBE, CCA-PKE or Sig schemes. This provides further motivation for starting from RKA-secure IBE as we do, rather than from RKA-secure PRF.

Finally we note that even for linear RKAs where IBE schemes were known via [4]+[3], our schemes are significantly more efficient.

Further contributions. In the full version [6], as a combination of the results of [4] and [24], we provide definitions for RKA security in the joint security setting, where the same key pair is used for both signature and encryption func-

tions, and show that a Φ-RKA-secure IBE scheme can be used to build a Φ-RKA and jointly secure combined signature and encryption scheme. This construction can be instantiated using any of our specific IBE schemes, by which we obtain the first concrete jointly secure combined signature and encryption schemes for the RKA setting.

We also show in [6] how to adapt the KEM-DEM (or hybrid encryption) paradigm to the RKA setting, and describe a highly efficient, Φ^{aff}-RKA-secure CCA-KEM that is inspired by our IBE framework and is based on the scheme of Boyen, Mei and Waters [17]. Our CCA-KEM's security rests on the hardness of the DBDH problem for asymmetric pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$; its ciphertexts consist of 2 group elements (one in $\mathbb{G}_1$ and one in $\mathbb{G}_2$), public keys are 3 group elements (two in $\mathbb{G}_2$ and one in $\mathbb{G}_T$), encryption is pairing-free, and the decryption cost is dominated by 3 pairing operations.

The final contribution (also in [6]) is an extension of our framework that lets us build an RKA-secure CCA-SE scheme from any IBE scheme satisfying an additional master public key malleability property. Such an IBE scheme, when subjected to our transformation, meets a notion of strong Φ-RKA security [4] where the challenge encryption is also subject to RKA. Applying the CHK transform gives a strong Φ-RKA-secure CCA-PKE scheme which can be converted into a Φ-RKA-secure CCA-SE scheme in the natural way.

Paper organization. Section 2 contains preliminaries, Section 3 describes some IBE schemes and RKA attacks on them, while Section 4 presents our framework for constructing RKA-secure IBE schemes. Section 5 applies the framework to specific schemes, and sketches the CCA-PKE and signature schemes that result from applying the techniques of [4].

2 Preliminaries

Notation. For sets X, Y let $\mathsf{Fun}(X, Y)$ be the set of all functions mapping X to Y. If S is a set then $|S|$ denotes its size and $s \leftarrow\!\!\$\ S$ the operation of picking a random element of S and denoting it by s. Unless otherwise indicated, an algorithm may be randomized. An adversary is an algorithm. By $y \leftarrow\!\!\$\ A(x_1, x_2, \ldots)$ we denote the operation of running A on inputs $x_1, x_2, \ldots$ and letting y denote the outcome. We denote by $[A(x_1, x_2, \ldots, x_n)]$ the set of all possible outputs of A on inputs $x_1, x_2, \ldots, x_n$.

Games. Some of our definitions and proofs are expressed through 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 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 is called the output of the game. We let G^A denote the event that this game output takes value 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.

RKD FUNCTIONS AND CLASSES. We say that ϕ is a related-key deriving (RKD) function over a set $\mathcal{S}$ if $\phi \in \mathsf{Fun}(\mathcal{S},\mathcal{S})$. We say that Φ is a class of RKD functions over $\mathcal{S}$ if $\Phi \subseteq \mathsf{Fun}(\mathcal{S},\mathcal{S})$ and $\mathsf{id} \in \Phi$ where id is the identity function on $\mathcal{S}$. In our constructs, $\mathcal{S}$ will have an algebraic structure, such as being a group, ring or field. In the last case, for $a, b \in \mathcal{S}$ we define $\phi^+_b, \phi^*_a, \phi^{\rm aff}_{a,b} \in \mathsf{Fun}(\mathcal{S},\mathcal{S})$ via $\phi^+_b(s) = s + b$, $\phi^*_a(s) = as$, and $\phi^{\rm aff}_{a,b}(s) = as + b$ for all $s \in \mathcal{S}$. For a polynomial q over field $\mathcal{S}$, we define $\phi^{\rm poly}_q(s) = q(s)$ for all $s \in \mathcal{S}$. We let $\Phi^+ = \{\ \phi^+_b\ :\ b \in \mathcal{S}\ \}$ be the class of additive RKD functions, $\Phi^* = \{\ \phi^*_a\ :\ a \in \mathcal{S}\ \}$ be the class of multiplicative RKD functions, $\Phi^{\rm aff} = \{\ \phi^{\rm aff}_{a,b}\ :\ a, b \in \mathcal{S}\ \}$ the class of affine RKD functions, and for any fixed positive integer d, we let $\Phi^{{\rm poly}(d)} = \{\ \phi^{\rm poly}_q\ :\ \deg q \leq d\ \}$ be the set of polynomial RKD functions of bounded degree d.

If $\phi \neq \phi'$ are distinct functions in a class Φ there is of course by definition an s such that $\phi(s) \neq \phi'(s)$, but there could also be keys s on which $\phi(s) = \phi'(s)$. We say that a class Φ is claw-free if the latter does not happen, meaning for all distinct $\phi \neq \phi'$ in Φ we have $\phi(s) \neq \phi'(s)$ for *all* $s \in \mathcal{S}$. With the exception of [21], all previous constructions of Φ-RKA-secure primitives with proofs of security have been for claw-free classes [5, 23, 20, 3, 4, 26]. In particular, key fingerprints are defined in [3] in such a way that their assumption of a Φ-key fingerprint automatically implies that Φ is claw-free.

IBE SYNTAX. We specify an IBE scheme $\mathcal{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ by first specifying a non-empty set $\mathcal{S}$ called the *master-key space* from which the master secret key s is drawn at random. The master public key $\pi \leftarrow \mathcal{P}(s)$ is then produced by applying to s a deterministic master public key generation algorithm $\mathcal{P}$. A decryption key for an identity u is produced via $dk_u \leftarrow\!\!\$\ \mathcal{K}(s, u)$. A ciphertext C encrypting a message M for u is generated via $C \leftarrow\!\!\$\ \mathcal{E}(\pi, u, M)$. A ciphertext C is deterministically decrypted via $M \leftarrow \mathcal{D}(dk, C)$. Correctness requires that $\mathcal{D}(\mathcal{K}(s, u), \mathcal{E}(\pi, u, M)) = M$ with probability one for all $M \in \mathsf{MSp}$ and all $u \in \mathsf{USp}$ where $\mathsf{MSp}, \mathsf{USp}$ are, respectively, the message and identity spaces associated to $\mathcal{IBE}$.

The usual IBE syntax specifies a single parameter generation algorithm that produces s, π together, and although there is of course a space from which the master secret key is drawn, it is not explicitly named. But RKD functions will have domain the space of master keys of the IBE scheme, which is why it is convenient in our context to make it explicit in the syntax. Saying the master public key is a deterministic function of the master secret key is not strictly necessary for us, but it helps make some things a little simpler and is true in all known schemes, so we assume it.

We make an important distinction between parameters and the master public key, namely that the former may not depend on s while the latter might. Parameters will be groups, group generators, pairings and the like. They will be fixed and available to all algorithms without being named as explicit inputs.

proc INITIALIZE
$s \leftarrow\!\!\$\ \mathcal{S}; \pi \leftarrow \mathcal{P}(s)$
$b \leftarrow\!\!\$\ \{0,1\}$
$u^* \leftarrow \perp\ ;\ I \leftarrow \emptyset$
Ret π

proc FINALIZE(b')
Ret $(b = b')$

proc KD(ϕ, u)
$s' \leftarrow \phi(s)$
If $(s' = s)$ $I \leftarrow I \cup \{u\}$
If $(u^* \in I)$ Ret $\perp$
Ret $dk \leftarrow\!\!\$\ \mathcal{K}(s', u)$

proc LR(u, M_0, M_1)
If $(|M_0| \neq |M_1|)$ Ret $\perp$
$u^* \leftarrow u$
If $(u^* \in I)$ Ret $\perp$
Ret $C \leftarrow\!\!\$\ \mathcal{E}(\pi, u^*, M_b)$

Fig. 2. Game IBE defining Φ-RKA-security of IBE scheme $\mathcal{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$.

$\mathcal{P}(s)$:
$\pi \leftarrow g^s$
Ret π

$\mathcal{K}(s, u)$:
$dk \leftarrow H_1(u)^s$
Ret dk

$\mathcal{E}(\pi, u, M)$:
$t \leftarrow\!\!\$\ \mathbb{Z}_p$
$C_1 \leftarrow g^t$
$C_2 \leftarrow H_2(e(\pi, H_1(u))^t) \oplus M$
Ret (C_1, C_2)

$\mathcal{D}(dk, C)$:
$M \leftarrow C_2 \oplus H_2(e(dk, C_1))$
Ret M

$\mathcal{P}(s)$:
$\pi \leftarrow g^s$
Ret π

$\mathcal{K}(s, u)$:
$r \leftarrow\!\!\$\ \mathbb{Z}_p$
$dk_1 \leftarrow g_1^s \cdot H(u)^r$
$dk_2 \leftarrow g^r$
Ret (dk_1, dk_2)

$\mathcal{E}(\pi, u, M)$:
$t \leftarrow\!\!\$\ \mathbb{Z}_p$
$C_1 \leftarrow g^t$
$C_2 \leftarrow H(u)^t$
$C_3 \leftarrow e(\pi, g_1)^t \cdot M$
Ret (C_1, C_2, C_3)

$\mathcal{D}(dk, C)$:
$M \leftarrow C_3 \cdot \frac{e(dk_2, C_2)}{e(dk_1, C_1)}$
Ret M

Fig. 3. Boneh-Franklin IBE scheme on the left, Waters IBE scheme on the right.

RKA-SECURE IBE. We define Φ-RKA security of IBE schemes following [4]. Game IBE of Fig. 2 is associated to $\mathcal{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ and a class Φ of RKD functions over $\mathcal{S}$. An adversary is allowed only one query to LR. Let $\mathbf{Adv}^{\mathrm{ibe-rka}}_{\mathcal{IBE},\Phi}(A)$ equal $2\Pr[\mathrm{IBE}^A]-1$. A feature of the definition we draw attention to is that the key derivation oracle KD refuses to act only when the identity it is given matches the challenge one *and* the derived key equals the real one. This not only creates a strong security requirement but one that is challenging to achieve because a simulator, not knowing s, cannot check whether or not the IBE adversary succeeded. This difficulty is easily resolved if Φ is claw-free but not otherwise. We consider this particular RKA security definition as, in addition to its strength, it is the level of RKA security required of an IBE scheme so that application of the CHK and Naor transforms results in RKA-secure CCA-PKE and signature schemes.

3 Existing IBE schemes and RKA attacks on them

The algorithms of the Boneh-Franklin BasicIdent IBE scheme [14] are given in Figure 3. The parameters of the scheme are groups $\mathbb{G}_1, \mathbb{G}_T$ of prime order p, a symmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, a generator g of $\mathbb{G}_1$ and hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \mathbb{G}_T \to \{0,1\}^n$ which are modeled as random oracles in the security analysis. Formally, these are output by a pairing parameter generator

on input 1^k. This scheme is IND-CPA secure in the usual model for IBE security, under the Bilinear Diffie-Hellman (BDH) assumption.

The algorithms of the Waters IBE scheme [25] are also given in Figure 3. The parameters of the scheme are groups $\mathbb{G}_1, \mathbb{G}_T$ of prime order p, a symmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, generators g, g_1 of $\mathbb{G}_1$ and group elements $h_0, \ldots, h_n \in \mathbb{G}_1$ specifying the hash function $H(u) = h_0 \prod_{i \in u} h_i$. The Waters IBE scheme is also IND-CPA secure in the usual model for IBE security, under the DBDH assumption.

The Waters IBE scheme is not RKA secure if Φ includes a function $\phi_a^*(s) = as$. A call to the key derivation oracle with any such ϕ yields a user secret key $(dk_1, dk_2) = (g_1^{as} \cdot H(u)^r, g^r)$. Raising this to a^{-1} gives $(dk_1', dk_2') = (g_1^s \cdot H(u)^{ra^{-1}}, g^{ra^{-1}})$, so that (dk_1', dk_2') is a user secret key for identity u under the original master secret key with randomness $r' = ra^{-1}$. An RKA adversary can thus obtain the user secret key for any identity of his choosing and hence break the RKA security of the Waters scheme. A similar attack applies to the Boneh-Franklin scheme.

4 Framework for deriving RKA-secure IBE schemes

In the previous section we saw that the Boneh-Franklin and Waters schemes are not RKA secure. Here we will show how to modify these and other schemes to be RKA secure by taking advantage, in part, of the very algebra that leads to the attacks. We describe a general framework for creating RKA-secure IBE schemes and then apply it obtain several such schemes.

We target a very particular type of framework, one that allows us to reduce RKA security of a modified IBE scheme directly to the normal IBE security of a base IBE scheme. This will allow us to exploit known results on IBE in a blackbox way and avoid re-entering the often complex security proofs of the base IBE schemes.

KEY-MALLEABILITY. We say that an IBE scheme $\mathit{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is Φ-key-malleable if there is an algorithm T, called the key simulator, which, given π, an identity u, a decryption key $dk' \leftarrow\!\!\$\, \mathcal{K}(s, u)$ for u under s and an RKD function $\phi \in \Phi$, outputs a decryption key dk for u under master secret key $\phi(s)$ that is distributed identically to the output of $\mathcal{K}(\phi(s), u)$. The formalization takes a little more care for in talking about two objects being identically distributed one needs to be precise about relative to what other known information this is true. A simple and rigorous definition here can be made using games. We ask that

$$\Pr[\mathrm{KMReal}^M_{\mathit{IBE},\Phi}] \;=\; \Pr[\mathrm{KMSim}^M_{\mathit{IBE},\Phi,T}]$$

for all (not necessarily computationally bounded) adversaries M, where the games are as follows. The INITIALIZE procedure of both picks s at random from $\mathcal{S}$ and returns $\pi \leftarrow \mathcal{P}(s)$ to the adversary. In game $\mathrm{KMReal}_{\mathit{IBE},\Phi}$, oracle $\mathrm{KD}(\phi, u)$ returns $dk \leftarrow\!\!\$\, \mathcal{K}(\phi(s), u)$ but in game $\mathrm{KMSim}_{\mathit{IBE},\Phi,T}$ it lets $dk' \leftarrow\!\!\$\, \mathcal{K}(s, u)$ and

returns $T(\pi, u, dk', \phi)$. There are no other oracles, and FINALIZE(b') returns $(b' = 1)$.

USING KM. Intuitively, key-malleability allows us to simulate a Φ-RKA adversary via a normal adversary and would thus seem to be enough to prove Φ-RKA security of $\mathcal{IBE}$ based on its normal security. Let us see how this argument goes and then see the catches that motivate a transformation of the scheme via collision-resistant identity renaming. Letting $\overline{A}$ be an adversary attacking the Φ-RKA security of $\mathcal{IBE}$, we aim to build an adversary A such that

$$\mathbf{Adv}^{\mathrm{ibe-rka}}_{\mathcal{IBE},\Phi}(\overline{A}) \leq \mathbf{Adv}^{\mathrm{ibe}}_{\mathcal{IBE}}(A) \ . \tag{1}$$

On input π, adversary A runs $\overline{A}(\pi)$. When the latter makes a KD(ϕ, u) query, A lets $dk \leftarrow \mathrm{KD}(\mathsf{id}, u)$, where KD is A's own key derivation oracle. It then lets $\overline{dk} \leftarrow T(\pi, u, dk, \phi)$ and returns $\overline{dk}$ to $\overline{A}$. Key-malleability tells us that $\overline{dk}$ is distributed identically to an output of KD(ϕ, u), so the response provided by A is perfectly correct. When $\overline{A}$ makes a LR(u, M_0, M_1) query, A lets $C \leftarrow \mathrm{LR}(u, M_0, M_1)$ and returns C to A. Finally when $\overline{A}$ halts with output a bit b', adversary A does the same.

The simulation seems perfect, so we appear to have established Equation (1). What's the catch? The problem is avoiding *challenge key derivation*. Suppose $\overline{A}$ made a KD(ϕ, u) query for a ϕ such that $\phi(s) \neq s$; then made a LR(u, M_0, M_1) query; and finally, given C, correctly computed b. It would win its game, because the condition $\phi(s) \neq s$ means that identity u may legitimately be used both in a key derivation query and in the challenge LR query. But our constructed adversary A, in the simulation, would make query KD(id, u) to answer $\overline{A}$'s KD(ϕ, u) query, and then make query LR(u, M_0, M_1). A would thus have queried the challenge identity u to the key-extraction oracle and would not win.

This issue is dealt with by transforming the base scheme via what we call identity renaming, so that Φ-RKA security of the transformed scheme can be proved based on the Φ-key-malleability of the base scheme.

IDENTITY RENAMING. Renaming is a way to map identities in the new scheme back to identities of the given, base scheme. Let us now say how renaming works more precisely and then define the modified scheme.

Let $\mathcal{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ denote the given, base IBE scheme, and let USp be its identity space. A renaming scheme is a pair (SI, PI) of functions where SI: $\mathcal{S} \times \overline{\mathsf{USp}} \to \mathsf{USp}$ and PI: $[\mathcal{P}(\mathcal{S})] \times \overline{\mathsf{USp}} \times \Phi \to \mathsf{USp}$ where $\overline{\mathsf{USp}}$, implicitly specified by the renaming scheme, will be the identity space of the new scheme we will soon define. The first function SI, called the secret renaming function, uses the master secret key, while its counterpart public renaming function PI uses the master public key. We require that $\mathrm{SI}(\phi(s), \overline{u}) = \mathrm{PI}(\pi, \overline{u}, \phi)$ for all $s \in \mathcal{S}$, all $\pi \in [\mathcal{P}(s)]$, all $\overline{u} \in \overline{\mathsf{USp}}$ and all $\phi \in \Phi$. This *compatibility condition* says that the two functions arrive, in different ways, at the same outcome.

THE TRANSFORM. The above is all we need to specify our Identity Renaming Transform **IRT** that maps a base IBE scheme $\mathcal{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ to a new IBE scheme $\overline{\mathcal{IBE}} = (\mathcal{S}, \mathcal{P}, \overline{\mathcal{K}}, \overline{\mathcal{E}}, \mathcal{D})$. As the notation indicates, the master key

space, master public key generation algorithm and decryption algorithm are unchanged. The other algorithms are defined by

$$\overline{\mathcal{K}}(s, \overline{u}) = \mathcal{K}(s, \mathrm{SI}(s, \overline{u})) \qquad \text{and} \qquad \overline{\mathcal{E}}(\pi, \overline{u}, M) = \mathcal{E}(\pi, \mathrm{PI}(\pi, \overline{u}, \mathsf{id}), M) \ .$$

We clarify that algorithms of the new IBE scheme do not, and cannot, have as input the RKD functions ϕ used by the attacker. We are defining an IBE scheme, and algorithm inputs must follow the syntax of IBE schemes. When the new encryption algorithm invokes PI, it sets ϕ to the identity function id. (Looking ahead, the simulation will call the renaming functions with ϕ emanating from the adversary attacking the new IBE scheme.) The key derivation algorithm has s but not π (recall we cannot give it π because otherwise it becomes subject to the RKA) and thus uses the secret renaming function. On the other hand the encryption algorithm has π but obviously not s and thus uses the public renaming function. This explains why we need two, compatible renaming functions. The new scheme has the same message space as the old one. Its identity space is inherited from the renaming scheme, being the space $\overline{\mathsf{USp}}$ from which the renaming functions draw their identity inputs.

The above compatibility requirement implies that $\mathrm{SI}(s, \overline{u}) = \mathrm{PI}(\pi, \overline{u}, \mathsf{id})$. From this it follows that $\overline{\mathit{IBE}}$ preserves the correctness of IBE. We now go on to specifying properties of the base IBE scheme and the renaming functions that suffice to prove Φ-RKA security of the new scheme.

A trivial renaming scheme is obtained by setting $\mathrm{SI}(s, \overline{u}) = \overline{u} = \mathrm{PI}(\pi, \overline{u}, \phi)$. This satisfies the compatibility condition. However, the transformed IBE scheme $\overline{\mathit{IBE}}$ ends up identical to the base IBE and thus this trivial renaming cannot aid in getting security. We now turn to putting a non-trivial condition on the renaming scheme that we will show suffices.

Collision-resistance. The renaming scheme $(\mathrm{SI}, \mathrm{PI})$ will be required to have a collision-resistance property. In its simplest and strongest form the requirement is that

$$(\phi(s), \overline{u}_1) \neq (s, \overline{u}_2) \qquad \Rightarrow \qquad \mathrm{SI}(\phi(s), \overline{u}_1) \neq \mathrm{SI}(s, \overline{u}_2)$$

for all $s \in \mathcal{S}$, all $\overline{u}_1, \overline{u}_2 \in \overline{\mathsf{USp}}$ and all $\phi \in \Phi$. This *statistical collision-resistance* will be enough to prove that $\overline{\mathit{IBE}}$ is Φ-RKA secure if IBE is Φ-key-malleable (cf. Theorem 1). We will now see how this goes. Then we will instantiate these ideas to get concrete Φ-RKA-secure schemes for many interesting classes Φ including Φ^{aff} and $\Phi^{\mathrm{poly}(d)}$.

Theorem 1. *Let $\mathit{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a Φ-key-malleable IBE scheme with key simulator T. Let $\overline{\mathit{IBE}} = (\mathcal{S}, \mathcal{P}, \overline{\mathcal{K}}, \overline{\mathcal{E}}, \mathcal{D})$ be obtained from IBE and renaming scheme $(\mathrm{SI}, \mathrm{PI})$ via the transform* **IRT** *described above. Assume the renaming scheme is statistically collision-resistant. Let $\overline{A}$ be a Φ-RKA adversary against $\overline{\mathit{IBE}}$ that makes q key derivation queries. Then there is an adversary A making q key derivation queries such that*

$$\mathbf{Adv}^{\mathrm{ibe-rka}}_{\overline{\mathit{IBE}}, \Phi}(\overline{A}) \leq \mathbf{Adv}^{\mathrm{ibe}}_{\mathit{IBE}}(A) \ . \tag{2}$$

$\underline{\text{proc INITIALIZE}}\ /\!/ \mathrm{G}_0$
$000\ s \leftarrow\!\!\$\ \mathcal{S}\ ;\ \pi \leftarrow \mathcal{P}(s)$
$001\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ \overline{u}^* \leftarrow \bot$
$002\ \overline{I} \leftarrow \emptyset$
$003\ \text{Ret}\ \pi$

$\underline{\text{proc INITIALIZE}}\ /\!/ \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3$
$100\ s \leftarrow\!\!\$\ \mathcal{S}\ ;\ \pi \leftarrow \mathcal{P}(s)$
$101\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ u^* \leftarrow \bot$
$102\ I \leftarrow \emptyset$
$103\ \text{Ret}\ \pi$

$\underline{\text{proc KD}(\phi, \overline{u})}\ /\!/ \mathrm{G}_0$
$010\ s' \leftarrow \phi(s)$
$011\ \text{If}\ (s' = s)\ \overline{I} \leftarrow \overline{I} \cup \{\overline{u}\}$
$012\ \text{If}\ (\overline{u}^* \in \overline{I})\ \text{Ret}\ \bot$
$013\ u \leftarrow \text{SI}(s', \overline{u})$
$014\ \text{Ret}\ \overline{dk} \leftarrow\!\!\$\ \mathcal{K}(s', u)$

$\underline{\text{proc KD}(\phi, \overline{u})}\ /\!/ \mathrm{G}_1$
$110\ s' \leftarrow \phi(s)$
$111\ u \leftarrow \text{SI}(s', \overline{u})$
$112\ I \leftarrow I \cup \{u\}$
$113\ \text{If}\ (u^* \in I)\ \text{Ret}\ \bot$
$114\ \text{Ret}\ \overline{dk} \leftarrow\!\!\$\ \mathcal{K}(s', u)$

$\underline{\text{proc KD}(\phi, \overline{u})}\ /\!/ \mathrm{G}_2$
$210\ u \leftarrow \text{PI}(\pi, \overline{u}, \phi)$
$211\ I \leftarrow I \cup \{u\}$
$212\ \text{If}\ (u^* \in I)\ \text{Ret}\ \bot$
$213\ \text{Ret}\ \overline{dk} \leftarrow\!\!\$\ \mathcal{K}(\phi(s), u)$

$\underline{\text{proc KD}(\phi, \overline{u})}\ /\!/ \mathrm{G}_3$
$310\ u \leftarrow \text{PI}(\pi, \overline{u}, \phi)$
$311\ I \leftarrow I \cup \{u\}$
$312\ \text{If}\ (u^* \in I)\ \text{Ret}\ \bot$
$313\ dk \leftarrow\!\!\$\ \mathcal{K}(s, u)$
$314\ \text{Ret}\ \overline{dk} \leftarrow T(\pi, u, dk, \phi)$

$\underline{\text{proc LR}(\overline{u}, M_0, M_1)}\ /\!/ \mathrm{G}_0$
$020\ \text{If}\ (|M_0| \neq |M_1|)\ \text{Ret}\ \bot$
$021\ \overline{u}^* \leftarrow \overline{u}$
$022\ \text{If}\ (\overline{u}^* \in \overline{I})\ \text{Ret}\ \bot$
$023\ u^* \leftarrow \text{SI}(s, \overline{u}^*)$
$024\ \text{Ret}\ C \leftarrow\!\!\$\ \mathcal{E}(\pi, u^*, M_b)$

$\underline{\text{proc LR}(\overline{u}, M_0, M_1)}\ /\!/ \mathrm{G}_1$
$120\ \text{If}\ (|M_0| \neq |M_1|)\ \text{Ret}\ \bot$
$121\ u^* \leftarrow \text{SI}(s, \overline{u})$
$122\ \text{If}\ (u^* \in I)\ \text{Ret}\ \bot$
$123\ \text{Ret}\ C \leftarrow\!\!\$\ \mathcal{E}(\pi, u^*, M_b)$

$\underline{\text{proc LR}(\overline{u}, M_0, M_1)}\ /\!/ \mathrm{G}_2, \mathrm{G}_3$
$220\ \text{If}\ (|M_0| \neq |M_1|)\ \text{Ret}\ \bot$
$221\ u^* \leftarrow \text{PI}(\pi, \overline{u}, \mathsf{id})$
$222\ \text{If}\ (u^* \in I)\ \text{Ret}\ \bot$
$223\ \text{Ret}\ C \leftarrow\!\!\$\ \mathcal{E}(\pi, u^*, M_b)$

$\underline{\text{proc FINALIZE}(b')}\ /\!/ \text{All}$
$030\ \text{Ret}\ (b = b')$

Fig. 4. Games for proof of Theorem 1.

Furthermore, the running time of A is that of $\overline{A}$ plus the time for q executions of T and $q+1$ executions of PI.

Proof (Theorem 1). Consider the games of Fig. 4. Game G_0 is written to be equivalent to game $\mathrm{IBE}_{\overline{\mathit{IBE}}}$, so that

$$\mathbf{Adv}^{\mathrm{ibe-rka}}_{\overline{\mathit{IBE}},\Phi}(\overline{A}) \;=\; 2\Pr[\mathrm{G}_0^{\overline{A}}] - 1\ . \tag{3}$$

In answering a $\mathrm{KD}(\phi, \overline{u})$ query, G_0 must use the key-generation algorithm $\overline{\mathcal{K}}$ of the new scheme $\overline{\mathit{IBE}}$ but with master secret key $s' = \phi(s)$. From the definition of $\overline{\mathcal{K}}$, it follows that not only is the key-generation at line 014 done under s', but also the identity renaming at line 013. LR, correspondingly, should use $\overline{\mathcal{E}}$, and thus the public renaming function PI. The compatibility property however allows us at line 023 to use SI instead. This will be useful in exploiting statistical collision-resistance in the next step, after which we will revert back to PI.

The adversary A we aim to construct will not know s. A central difficulty in the simulation is thus lines 011, 012 of G_0 where the response provided to $\overline{A}$ depends on the result of a test involving s, a test that A cannot perform. Before we can design A we must get rid of this test. Statistical collision-resistance is what will allow us to do so. KD of game G_1 moves the identity renaming up before the list of queried identities is updated to line 111 and then, at line 112, adds the transformed identity to the list. LR is likewise modified so its test now involves the transformed (rather than original) identities. We claim this makes no difference, meaning

$$\Pr[\mathrm{G}_0^{\overline{A}}] \;=\; \Pr[\mathrm{G}_1^{\overline{A}}]\ . \tag{4}$$

Indeed, statistical collision-resistance tell us that $(s', \overline{u}) = (s, \overline{u}^*)$ iff $\mathrm{SI}(s', \overline{u}) = \mathrm{SI}(s, \overline{u}^*)$. This means that lines 011, 012 and lines 112, 113 are equivalent.

Compatibility is invoked to use PI in place of SI in both KD and in LR in G_2, so that

$$\Pr[\mathrm{G}_1^{\overline{A}}] \;=\; \Pr[\mathrm{G}_2^{\overline{A}}]\;. \tag{5}$$

Rather than use s' for key generation as at 213, G_3 uses s at 313 and then applies the key simulator T. We claim the key-malleability implies

$$\Pr[\mathrm{G}_2^{\overline{A}}] \;=\; \Pr[\mathrm{G}_3^{\overline{A}}]\;. \tag{6}$$

To justify this we show that there is an adversary M such that

$$\Pr[\mathrm{KMReal}^{M}_{\mathcal{IBE},\Phi}] \;=\; \Pr[\mathrm{G}_2^{\overline{A}}] \qquad \text{and} \qquad \Pr[\mathrm{KMSim}^{M}_{\mathcal{IBE},\Phi,T}] \;=\; \Pr[\mathrm{G}_3^{\overline{A}}]\;.$$

Adversary M, on input π, begins with the initializations $u^* \leftarrow \perp$; $I \leftarrow \emptyset$; $b \leftarrow\!\!\$\ \{0,1\}$ and then runs $\overline{A}$ on input π. When $\overline{A}$ makes a $\mathrm{KD}(\phi, \overline{u})$ query, M does the following:

$$u \leftarrow \mathrm{PI}(\pi, \overline{u}, \phi)\ ;\ I \leftarrow I \cup \{u\}\ ;\ \text{If } (u^* \in I) \text{ Ret } \perp\ ;\ \overline{dk} \leftarrow \mathrm{KD}(\phi, u).$$

If M is playing game KMReal then its KD oracle will behave as line 213 in game G_2, while if M is playing game KMSim its KD oracle will behave as lines 313,314 in game G_3. When $\overline{A}$ makes its $\mathrm{LR}(\overline{u}, M_0, M_1)$ query M sets $u^* \leftarrow \mathrm{PI}(\pi, \overline{u}, \mathsf{id})$ and checks if $u^* \in I$, returning $\perp$ if so. M then computes $C \leftarrow\!\!\$\ \mathcal{E}(\pi, u^*, M_b)$ which it returns to $\overline{A}$. When $\overline{A}$ halts with output b', M returns the result of $(b' = b)$. If M is playing game KMReal then game G_2 is perfectly simulated, while if M is playing KMSim then game G_3 is perfectly simulated, so M returns 1 with the same probability that $\overline{A}$ wins in each case and by the key-malleability of $\mathcal{IBE}$ Equation (6) holds.

Finally, we design A so that

$$\mathbf{Adv}^{\mathrm{ibe}}_{\mathcal{IBE}}(A) \;=\; 2\Pr[\mathrm{G}_3^{\overline{A}}] - 1\;. \tag{7}$$

On input π, adversary A runs $\overline{A}(\pi)$. When the latter makes a $\mathrm{KD}(\phi, \overline{u})$ query, A does the following:

$$u \leftarrow \mathrm{PI}(\pi, \overline{u}, \phi)\ ;\ dk \leftarrow \mathrm{KD}(\mathsf{id}, u)\ ;\ \overline{dk} \leftarrow T(\pi, u, dk, \phi).$$

It then returns $\overline{dk}$ to $\overline{A}$. The KD invoked in this code is A's own oracle. Compatibility tells us that $u = \mathrm{SI}(\phi(s), \overline{u})$ and thus from the definition of $\overline{\mathcal{IBE}}$, the response to $\overline{A}$'s query is distributed according to $\mathcal{K}(\phi(s), u)$. But key-malleability then tells us that $\overline{dk}$ is distributed identically to this, so the response provided by A is perfectly correct. When $\overline{A}$ makes a $\mathrm{LR}(\overline{u}, M_0, M_1)$ query, A does the following:

$$u \leftarrow \mathrm{PI}(\pi, \overline{u}, \mathsf{id})\ ;\ C \leftarrow \mathrm{LR}(u, M_0, M_1).$$

It then returns C to $\overline{A}$. The LR invoked in this code is A's own oracle. The definition of $\overline{\mathcal{IBE}}$ implies that the response provided by A is again perfectly correct. Finally when $\overline{A}$ halts with output a bit b', adversary A does the same.

5 Applying the framework

Affine RKD functions for Boneh-Franklin and Waters. We show how the framework can be instantiated with the IBE schemes of Boneh-Franklin and Waters to achieve IBE schemes secure against affine related-key attacks. First we look at key-malleability. Keys in the Boneh-Franklin IBE scheme are of the form $dk' = H_1(u)^s$, so the algorithm T is as follows:

$T(\pi, u, dk', \phi_{a,b})$: $dk \leftarrow dk'^a \cdot H_1(u)^b$; Ret dk

The output of T is a valid key for user u under master secret key $\phi_{a,b}(s)$, since: $dk'^a \cdot H_1(u)^b = H_1(u)^{sa} \cdot H_1(u)^b = H_1(u)^{as+b}$. Since the key derivation algorithm is deterministic, the keys output by T are distributed identically to the keys output by $\mathcal{K}(\phi(s), u)$, and so the Boneh-Franklin IBE scheme is key-malleable.

Keys in the Waters IBE scheme are of the form $(dk'_1, dk'_2) = (g_1^s \cdot H(u)^r, g^r)$ for some r in $\mathbb{Z}_p$, so the algorithm T is as follows:

$T(\pi, u, dk', \phi_{a,b})$:
If $(a = 0)$ then $r \leftarrow_{\$} \mathbb{Z}_p$; $dk_1 \leftarrow g_1^b \cdot H(u)^r$; $dk_2 \leftarrow g^r$
Else $dk_1 \leftarrow dk_1'^a \cdot g_1^b$; $dk_2 \leftarrow dk_2'^a$
Ret (dk_1, dk_2)

When the RKD function is a constant function, T behaves exactly as the key derivation algorithm under master secret key b, so its output is valid and correctly distributed. Otherwise, the output of T is still a valid key for user u under master secret key $\phi_{a,b}(s)$, now under randomness ra, since:

$$dk_1'^a \cdot g_1^b = (g_1^s \cdot H(u)^r)^a \cdot g_1^b = g_1^{as+b} H(u)^{ra} \qquad dk_2'^a = g^{ra} \ .$$

Since r is uniformly distributed in $\mathbb{Z}_p$, ra is also uniformly distributed in $\mathbb{Z}_p$ and so the keys output by T are distributed identically to those output by $\mathcal{K}(\phi(s), u)$. Hence the Waters IBE scheme is key-malleable.

The same identity renaming scheme can be used for both IBE schemes. Namely, $\mathrm{SI}(s, \overline{u})$ returns $\overline{u}||g^s$ and $\mathrm{PI}(\pi, \overline{u}, \phi_{a,b})$ returns $\overline{u}||\pi^a \cdot g^b$. The compatibility requirement is satisfied and the renaming scheme is clearly collision-resistant since $\overline{u}_1||g^{\phi(s)} = \overline{u}_2||g^s \Rightarrow \overline{u}_1 = \overline{u}_2 \wedge \phi(s) = s$. Thus the IBE schemes of Boneh-Franklin and Waters are key-malleable and admit a suitable identity renaming scheme, and so satisfy the requirements of Theorem 1. Notice that in the Waters case, we must increase the parameter n by the bit length of elements of $\mathbb{G}_1$ (and hence increase the size of the description of the scheme parameters) to allow identities of the form $\overline{u}||g^s$ to be used in the renaming scheme.

The following theorem is obtained by combining Theorem 1 with [14], and the running time of B below may be obtained in the same way.

Theorem 2. *Let $\overline{\mathit{IBE}} = (\mathcal{S}, \mathcal{P}, \overline{\mathcal{K}}, \overline{\mathcal{E}}, \mathcal{D})$ be the Boneh-Franklin IBE scheme shown in Fig. 3 under the above identity renaming transform. Let $\overline{A}$ be a Φ^{aff}-RKA adversary against $\overline{\mathit{IBE}}$ making q_{KD} key derivation queries and q_{H_2} queries to random oracle H_2. Then there is an algorithm B solving the Decision Bilinear Diffie-Hellman problem such that*

$$\mathbf{Adv}^{\mathrm{ibe-rka}}_{\overline{\mathit{IBE}}, \Phi^{\mathrm{aff}}}(\overline{A}) \leq \frac{e(1 + q_{KD})q_{H_2}}{2} \cdot \mathbf{Adv}^{\mathrm{dbdh}}(B) \ . \tag{8}$$

The following theorem is obtained by combining Theorem 1 with [25], and the running time of B below may be obtained in the same way. Concrete-security improvements would be obtained by using instead the analysis of Waters' scheme from [7].

Theorem 3. *Let $\overline{\mathcal{IBE}} = (\mathcal{S}, \mathcal{P}, \overline{\mathcal{K}}, \overline{\mathcal{E}}, \mathcal{D})$ be the Waters scheme shown in Fig. 3 under the above identity renaming transform. Let $\overline{A}$ be a Φ^{aff}-RKA adversary against $\overline{\mathcal{IBE}}$ making q_{KD} key derivation queries. Then there is an algorithm B solving the Decision Bilinear Diffie-Hellman problem such that*

$$\mathbf{Adv}^{\mathrm{ibe-rka}}_{\overline{\mathcal{IBE}},\Phi^{\mathrm{aff}}}(\overline{A}) \leq 32(n+1) \cdot q_{KD} \cdot \mathbf{Adv}^{\mathrm{dbdh}}(B) \; . \tag{9}$$

We recall from [4] that, given a Φ-RKA-secure IBE scheme, the CHK transform [12] yields a Φ-RKA-secure CCA-PKE scheme at the cost of adding a strongly unforgeable one-time secure signature and its verification key to the IBE ciphertexts. In the full version [6] we show that the more efficient Boneh-Katz transform [12] can also be used to the same effect. We omit the details of the Φ^{aff}-RKA-secure CCA-PKE schemes that result from applying these transforms to the above IBE schemes. We simply note that the resulting CCA-PKE schemes are as efficient as the pairing-based schemes of Wee [26], which are only Φ^{lin}-RKA-secure. Similarly, using a result of [4], we may apply the Naor transform to these IBE schemes to obtain Φ^{aff}-RKA-secure signature schemes that are closely related to (and as efficient as) the Boneh-Lynn-Shacham [16] and Waters [25] signature schemes. The verification algorithms of these signature schemes can be improved by replacing Naor's trial encryption and decryption procedure by bespoke algorithms, exactly as in [16, 25].

An IBE scheme handling RKAs for bounded degree polynomials. We show how to construct an IBE scheme that is RKA secure when the RKD function set equals $\Phi^{\mathrm{poly}(d)}$, the set of all polynomials of degree at most d, for an arbitrary d chosen at the time of master key generation. The scheme is obtained through a simple extension of the IBE scheme of Waters combined with the identity renaming transform used above. The only change we make to the Waters scheme is in the master public key, where we add the extra elements $g^{s^2}, \ldots, g^{s^d}, {g_1}^{s^2}, \ldots, {g_1}^{s^d}$ alongside g^s. These elements assist in achieving key-malleability for the set $\Phi^{\mathrm{poly}(d)}$. The master public-key generation algorithm $\mathcal{P}$ of the extended Waters scheme, on input s, returns $\pi \leftarrow (g^s, g^{s^2}, \ldots, g^{s^d}, (g_1)^{s^2}, \ldots, (g_1)^{s^d})$. The other algorithms and keys remain unchanged; in particular, key derivation does not make use of these new elements. This extended Waters IBE scheme is secure (in the usual IND-CPA sense for IBE) under the q-type extension of the standard DBDH assumption captured by the game in Fig. 5. We define the advantage of an adversary A against the problem as $\mathbf{Adv}^{q\text{-edbdh}}(A) = 2\Pr[q\text{-EDBDH}^A] - 1$.

Theorem 4. *Let $\mathcal{IBE} = (\mathcal{S}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be the extended Waters scheme. Let A be an adversary against $\mathcal{IBE}$ making q_{KD} key derivation queries. Then there is an algorithm B solving the q-Extended Decision Bilinear Diffie-Hellman problem*

proc INITIALIZE
$g \leftarrow\$ \mathbb{G}_1$; $x, y, z \leftarrow\$ \mathbb{Z}_p$; $b \leftarrow\$ \{0,1\}$
If $(b = 1)$ $T \leftarrow e(g,g)^{xyz}$
Else $T \leftarrow\$ \mathbb{G}_T$
Ret $g, g^x, g^{x^2}, \ldots, g^{x^q}, g^y, g^{(x^2)y}, g^{(x^3)y}, \ldots, g^{(x^q)y}, g^z, T$

proc FINALIZE(b')
Ret $(b = b')$

Fig. 5. q-Extended Decision Bilinear Diffie-Hellman (q-EDBDH) game.

for $q = d$ such that

$$\mathbf{Adv}^{\mathrm{ibe}}_{I\!B\!E}(A) \leq 32(n+1) \cdot q_{KD} \cdot \mathbf{Adv}^{q\text{-edbdh}}(B) \ . \tag{10}$$

To see this, observe that the original proof of security for Waters' scheme [25, 7] also goes through for the extended scheme, using the elements g, g^x, g^y, T from the q-EDBDH problem to run the simulation as in the original proof and using the additional elements from the q-EDBDH problem to set up the master public key in the extended scheme.

We give evidence for the validity of the q-EDBDH assumption by examining the difficulty of the problem in the generic group model. The problem falls within the framework of the generic group model "master theorem" of Boneh, Boyen and Goh [11]. In their notation, we have $P = \{1, x, x^2, \ldots, x^q, y, x^2y, \ldots, x^qy, z\}, Q = 1$, and $f = xyz$. It is clear by inspection that P, Q and f meet the independence requirement of the master theorem, and it gives a lower bound on an adversary's advantage of solving the q-EDBDH problem in a generic group of the form $(q+1)(q_\xi + 4q + 6)^2/p$ where q_ξ is a bound on the number of queries made by the adversary to the oracles computing the group operations in $\mathbb{G}, \mathbb{G}_T$. While a lower bound in the generic group model does not rule out an efficient algorithm when the group is instantiated, it lends heuristic support to our assumption.

The extended Waters IBE scheme is $\Phi^{\mathrm{poly}(d)}$-key malleable with algorithm T as follows:

$T(\pi, u, dk', \phi_{a_0,a_1,\ldots,a_d})$:
If $(a_0 = 0)$ then $r \leftarrow\$ \mathbb{Z}_p$; $dk_1 \leftarrow g_1^{a_0} \cdot H(u)^r \cdot \left(g_1{}^{s^2}\right)^{a_2} \cdots \left(g_1{}^{s^d}\right)^{a_d}$; $dk_2 \leftarrow g^r$
Else $dk_1 \leftarrow g_1^{a_0} \cdot dk_1'^{a_1} \cdot \left(g_1{}^{s^2}\right)^{a_2} \cdots \left(g_1{}^{s^d}\right)^{a_d}$; $dk_2 \leftarrow dk_2'^{a_1}$
Ret (dk_1, dk_2)

The identity renaming scheme is then defined via

$$\mathrm{SI}(s, \overline{u}) = \overline{u} || g^s \quad \text{and} \quad \mathrm{PI}(\pi, \overline{u}, \phi_{a_0,a_1,\ldots,a_d}) = \overline{u} || g^{a_0} \cdot \pi^{a_1} \cdot \left(g^{s^2}\right)^{a_2} \cdots \left(g^{s^d}\right)^{a_d}$$

which clearly meets the compatibility and collision-resistance requirements. Combining Theorem 1 with Theorem 4 gives the following theorem.

Theorem 5. *Let $\overline{I\!B\!E} = (\mathcal{S}, \mathcal{P}, \overline{\mathcal{K}}, \overline{\mathcal{E}}, \mathcal{D})$ be the extended Waters scheme under the above identity renaming transform. Let $\overline{A}$ be a $\Phi^{\mathrm{poly}(d)}$-RKA adversary against $\overline{I\!B\!E}$ making q_{KD} key derivation queries. Then there is an algorithm B*

solving the q-Extended Decision Bilinear Diffie-Hellman problem for $q = d$ such that

$$\mathbf{Adv}^{\mathrm{ibe-rka}}_{\overline{IBE},\Phi^{\mathrm{poly}(d)}}(\overline{A}) \leq 32(n+1) \cdot q_{KD} \cdot \mathbf{Adv}^{q\text{-edbdh}}(B) \; . \tag{11}$$

As in the affine case, we may apply results of [4] to obtain a $\Phi^{\mathrm{poly}(d)}$-RKA-secure CCA-PKE scheme and a $\Phi^{\mathrm{poly}(d)}$-RKA-secure signature scheme. We omit the detailed but obvious description of these schemes, noting merely that they are efficient and secure in the standard model under the q-EDBDH assumption.

Acknowledgments

Bellare was supported in part by NSF grants CNS-1116800, CNS 0904380 and CCF-0915675. Paterson and Thomson were supported by EPSRC Leadership Fellowship EP/H005455/1.

References

1. B. Applebaum. Garbling XOR gates "for free" in the standard model. Cryptology ePrint Archive, Report 2012/516, 2012. `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 and tampering. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 486–503. Springer, Dec. 2011.
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, K. G. Paterson, and S. Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. Cryptology ePrint Archive, Report 2012/514, 2012. Full version of this abstract, `http://eprint.iacr.org/`.
7. M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Apr. 2009.
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. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, May 2005.
12. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.
13. 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.
14. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
15. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103. Springer, Feb. 2005.
16. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Dec. 2001.
17. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In V. Atluri, C. Meadows, and A. Juels, editors, *ACM CCS 05*, pages 320–329. ACM Press, Nov. 2005.
18. R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 471–488. Springer, Apr. 2008.
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. 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.
22. 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.
23. 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.
24. K. G. Paterson, J. C. N. Schuldt, M. Stam, and S. Thomson. On the joint security of encryption and signature, revisited. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 161–178. Springer, Dec. 2011.
25. B. R. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, May 2005.
26. H. Wee. Public key encryption against related key attacks. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2012.