Key-Versatile Signatures and Applications: RKA, KDM and Joint Enc/Sig

Mihir Bellare[1], Sarah Meiklejohn[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] Department of Computer Science & Engineering, University of California San Diego. smeiklej@eng.ucsd.edu; cseweb.ucsd.edu/~smeiklejohn/
[3] Department of Computer Science, University of Bristol. susan.thomson@bristol.ac.uk; sthomson.co.uk

Abstract. This paper introduces key-versatile signatures. Key-versatile signatures allow us to sign with keys already in use for another purpose, without changing the keys and without impacting the security of the original purpose. This allows us to obtain advances across a collection of challenging domains including joint Enc/Sig, security against related-key attack (RKA) and security for key-dependent messages (KDM). Specifically we can (1) Add signing capability to existing encryption capability with zero overhead in the size of the public key (2) Obtain RKA-secure signatures from any RKA-secure one-way function, yielding new RKA-secure signature schemes (3) Add integrity to encryption while maintaining KDM-security.

1 Introduction

One of the recommended principles of sound cryptographic design is key separation, meaning that keys used for one purpose (e.g. encryption) should not be used for another purpose (e.g. signing). The reason is that, even if the individual uses are secure, the joint usage could be insecure [39]. This paper shows, to the contrary, that there are important applications where key reuse is not only desirable but crucial to maintain security, and that when done "right" it works. We offer key-versatile signatures as a general tool to enable signing with existing keys already in use for another purpose, without adding key material and while maintaining security of both the new and the old usage of the keys. Our applications include: (1) adding signing capability to existing encryption capability with zero overhead in the size of the public key (2) obtaining RKA-secure signatures from RKA-secure one-way functions (3) adding integrity to encryption while preserving KDM security.

Closer look. Key-versatility refers to the ability to take an arbitrary one-way function F and return a signature scheme where the secret signing key is a random domain point x for F and the public verification key is its image $y = F(x)$. By requiring strong simulatability and key-extractability security conditions [33] from these "F-keyed" signatures, and then defining F based

on keys already existing for another purpose, we will be able to add signing capability while maintaining existing keys and security.

The most compelling motivation comes from security against related-key attack (RKA) and security for key-dependent messages (KDM), technically challenging areas where solutions create, and depend on, very specific key structures. We would like to expand the set of primitives for which we can provide these forms of security. Rather than start from scratch, we would like to leverage the existing, hard-won advances in these areas by modular design, transforming a primitive $\mathbf{X}$ into a primitive $\mathbf{Y}$ while preserving RKA or KDM security. Since security is relative to a set of functions (either key or message deriving) on the space of keys, the transform must preserve the existing keys. Key-versatile signatures will thus allow us to create new RKA and KDM secure primitives in a modular way.

We warn that our results are theoretical feasibility ones. They demonstrate that certain practical goals can in principle be reached, but the solutions are not efficient. Below we begin with a more direct application of key versatile signatures to Joint Enc/Sig and then go on to our RKA and KDM results.

Joining signatures to encryption with zero public-key overhead. Suppose Alice has keys (sk_e, pk_e) for a public-key encryption scheme and wants to also have signing capability. Certainly, she could pick new and separate keys (sk_s, pk_s) enabling her to use her favorite signature scheme. However, it means that Alice's public key, now $pk = (pk_e, pk_s)$, has doubled in size. Practitioners ask if one can do better. We want a joint encryption and signature (JES) scheme [49, 57], where there is a single key-pair (sk, pk) used for both encryption and signing. We aim to minimize the public-key overhead, (loosely) defined as the size of pk minus the size of the public key pk_e of the underlying encryption scheme.

Haber and Pinkas [49] initiated an investigation of JES. They note that the key re-use requires defining and achieving new notions of security particular to JES: signatures should remain unforgeable even in the presence of a decryption oracle, and encryption should retain IND-CCA privacy even in the presence of a signing oracle. In the random oracle model [17], *specific* IND-CCA-secure public-key encryption schemes have been presented where signing can be added with no public-key overhead [49, 34, 53]. In the standard model, encryption schemes have been presented that allow signing with a public-key overhead lower than that of the "Cartesian product" solution of just adding a separate signing key [49, 57], with the best results, from [57], using IBE or combining encryption and signature schemes of [27, 23].

All these results, however, pertain to *specific* encryption schemes. We step back to ask a general theoretical question. Namely, suppose we are given an *arbitrary* IND-CCA-secure public-key encryption scheme. We wish to add signing capability to form a JES scheme. How low can the public-key overhead go? The (perhaps surprising) answer we provide is that we can achieve a public-key overhead of *zero*. The public key for our JES scheme remains *exactly* that of the given encryption scheme, meaning we add signing capability without changing

the public key. (Zero public-key overhead has a particular advantage besides space savings, namely that, in adding signing, no new certificates are needed. This makes key management significantly easier for the potentially large number of entities already using Alice's public key. This advantage is absent if the public key is at all modified.) We emphasize again that this is for *any* starting encryption scheme.

To do this, we let F be the function that maps the secret key of the given encryption scheme to the public key. (Not all encryption schemes will directly derive the public key as a deterministic function of the secret key, although many, including Cramer-Shoup [35], do. However, we can modify any encryption scheme to have this property, *without changing the public key*, by using the coins of the key-generation algorithm as the secret key.) The assumed security of the encryption scheme means this function is one-way. Now, we simply use an F-keyed signature scheme, with the keys remaining those of the encryption scheme. No new keys are introduced. We need however to ensure that the joint use of the keys does not result in bad interactions that make either the encryption or the signature insecure. This amounts to showing that the JES security conditions, namely that encryption remains secure even given a signing oracle and signing remains secure even given a decryption oracle, are met. This will follow from the simulatability and key-extractability requirements we impose on our F-keyed signatures. See Section 4.

New RKA-secure signatures. In a related-key attack (RKA) [52, 18, 13, 9] an adversary can modify a stored secret key and observe outcomes of the cryptographic primitive under the modified key. Such attacks may be mounted by tampering [25, 19, 44], so RKA security improves resistance to side-channel attacks. Achieving proven security against RKAs, however, is broadly recognized as very challenging. This has lead several authors [45, 9] to suggest that we "bootstrap," building higher-level Φ-RKA-secure primitives from lower-level Φ-RKA-secure primitives. (As per the framework of [13, 9], security is parameterized by the class of functions Φ that the adversary is allowed to apply to the key. Security is never possible for the class of all functions [13], so we seek results for specific Φ.) In this vein, [9] show how to build Φ-RKA signatures from Φ-RKA PRFs. Building Φ-RKA PRFs remains difficult, however, and we really have only one construction [8]. This has lead to direct (non-bootstrapping) constructions of Φ-RKA signatures for classes Φ of polynomials over certain specific pairing groups [16].

We return to bootstrapping and provide a much stronger result, building Φ-RKA signatures from Φ-RKA one-way functions rather than from Φ-RKA PRFs. (For a one-way function, the input is the "key." In attempting to recover x from $F(x)$, the adversary may also obtain $F(x')$ where x' is created by applying to x some modification function from Φ. The definition is from [45].) The difference is significant because building Φ-RKA one-way functions under standard assumptions is easy. Adapting the key-malleability technique of [8], we show that many natural one-way functions are Φ-RKA secure *assuming nothing more than their standard one-wayness*. In particular this is true for discrete exponentiation over

an arbitrary group and for the one-way functions underlying the LWE and LPN problems. In this way we obtain Φ-RKA signatures for many new and natural classes Φ.

The central challenge in our bootstrapping is to preserve the keyspace, meaning that the space of secret keys of the constructed signature scheme must be the domain of the given Φ-RKA one-way function F. (Without this, it is not even meaningful to talk of preserving Φ-RKA security, let alone to show that it happens.) This is exactly what an F-keyed signature scheme allows us to do. The proof that Φ-RKA security is preserved exploits strong features built into our definitions of simulatability and key-extractability for F-keyed signatures, in particular that these conditions hold even under secret keys selected by the adversary. See Section 5.

KDM-SECURE STORAGE. Over the last few years we have seen a large number of sophisticated schemes to address the (challenging) problem of encryption of key-dependent data (e.g., [21, 26, 5, 4, 30, 31, 20, 7, 55, 3, 28, 29, 12, 42, 51]). The most touted application is secure outsourced storage, where Alice's decryption key, or some function thereof, is in a file she is encrypting and uploading to the cloud. But in this setting integrity is just as important as privacy. To this end, we would like to add signatures, thus enabling the server, based on Alice's public key, to validate her uploads, and enabling Alice herself to validate her downloads, all *while preserving KDM security.*

What emerges is a new goal that we call KDM-secure (encrypted and authenticated) storage. In Section 6 we formalize the corresponding primitive, providing both syntax and notions of security for key-dependent messages. Briefly, Alice uses a secret key sk to turn her message M into an encrypted and authenticated "data" object that she stores on the server. The server is able to check integrity based on Alice's public key. When Alice retrieves data, she can check integrity and decrypt based on her secret key. Security requires both privacy and integrity even when M depends on sk. (As we explain in more depth below, this goal is different from signcryption [62], authenticated public-key encryption [2] and authenticated symmetric encryption [15, 58], even in the absence of KDM considerations.)

A natural approach to achieve our goal is for Alice to encrypt under a symmetric, KDM-secure scheme and sign the ciphertexts under a conventional signature scheme. But it is not clear how to prove the resulting storage scheme is KDM-secure. The difficulty is that sk would include the signing key in addition to the encryption (and decryption) key K, so that messages depend on both these keys while the KDM security of the encryption only covers messages depending on K. We could attempt to start from scratch and design a secure storage scheme meeting our notions. But key-versatile signatures offer a simpler and more modular solution. Briefly, we take a KDM-secure *public-key* encryption scheme and let F be the one-way function that maps a secret key to a public key. Alice holds (only) a secret key sk and the server holds $pk = F(sk)$. To upload M, Alice re-computes pk from sk, encrypts M under it using the KDM scheme,

and signs the ciphertext with an F-keyed signature scheme using the *same* key sk. The server verifies signatures under pk.

In Section 6 we present in full the construction outlined above, and prove that it meets our notion of KDM security. The crux, as for our RKA-secure constructions, is that adding signing capability without changing the keys puts us in a position to exploit the assumed KDM security of the underlying encryption scheme. The strong simulatability and key-extractability properties of our signatures do the rest. We note that as an added bonus, we assume only CPA KDM security of the base encryption scheme, yet our storage scheme achieves CCA KDM security.

Getting F-keyed signatures. In Section 3 we define F-keyed signature schemes and show how to construct them for arbitrary one-way F. This enables us to realize the above applications.

Our simulatability condition, adapting [33, 1, 32], asks for a trapdoor allowing the creation of simulated signatures given only the message and public key, even when the secret key underlying this public key is adversarially chosen. Our key-extractability condition, adapting [33], asks that, using the same trapdoor, one can extract from a valid signature the corresponding secret key, even when the public key is adversarially chosen. Theorem 1, showing these conditions imply not just standard but strong unforgeability, functions not just as a sanity check but as a way to introduce, in a simple form, a proof template that we will extend for our applications.

Our construction of an F-keyed signature scheme is a minor adaptation of a NIZK-based signature scheme of Dodis, Haralambiev, López-Alt and Wichs (DHLW) [40]. While DHLW [40] prove leakage-resilience of their scheme, we prove simulatability and key-extractability. The underlying SE NIZKs are a variant of simulation-sound extractable NIZKs [36, 47, 48] introduced by [40] under the name tSE NIZKs and shown by [40, 50] to be achievable for all of **NP** under standard assumptions.

Discussion and related work. F-keyed signatures can be viewed as a special case of signatures of knowledge as introduced by Chase and Lysyanskaya [33]. The main novelty of our work is in the notion of key-versatility, namely that F-keyed signatures can add signing capability without changing keys, and the ensuing applications to Joint Enc/Sig, RKA and KDM. In particular our work shows that signatures of knowledge have applications beyond those envisaged in [33].

The first NIZK-based signature scheme was that of [10]. It achieved only unforgeability. Simulatability and extractability were achieved in [33] using dense cryptosystems [38, 37] and simulation-sound NIZKs [60, 36]. The DHLW construction we use can be viewed as a simplification and strengthening made possible by the significant advances in NIZK technology since then.

F-keyed signatures, and, more generally, signatures of knowledge [33] can be seen as a signing analogue of Witness encryption [43, 11], and we might have named them Witness Signatures. GGSW [43] show how witness encryption al-

lows encryption with a flexible choice of keys, just as we show that F-keyed signatures allow signing with a flexible choice of keys.

Signcryption [62], authenticated public-key encryption [2], JES [49, 57] and our secure storage goal all have in common that both encryption and signature are involved. However, in signcryption and authenticated public-key encryption, there are two parties and thus two sets of keys, Alice encrypting under Bob's public key and signing under her own secret key. In JES and secure storage, there is one set of keys, namely Alice's. Thus for signcryption and authenticated public-key encryption, the question of using the same keys for the two purposes, which is at the core of our goals and methods, does not arise. Self-signcryption [41] is however similar to secure storage, minus the key-dependent message aspect. Authenticated symmetric encryption [15, 58] also involves both encryption and authentication, but under a shared key, while JES and secure storage involve public keys. KDM-secure authenticated symmetric encryption was studied in [12, 6].

KDM-secure signatures were studied in [56], who show limitations on the security achievable. Our secure storage scheme bypasses these limitations by signing ciphertexts rather than plaintexts and by avoiding KDM-secure signatures altogether: we use F-keyed signatures and are making no standalone claims or assumptions regarding their KDM security. Combining KDM encryption and KDM signatures would not give us KDM-secure storage because the keys for the two primitives would be different and we want joint KDM security.

Secure storage is an amalgam of symmetric and asymmetric cryptography, encryption being of the former kind and authentication of the latter. With secure storage, we are directly modeling a goal of practical interest rather than trying to create a general-purpose tool like many of the other works just mentioned. The difference between JES and secure storage is that in the former, arbitrary messages may be signed, while in the latter only ciphertexts may be signed. The difference is crucial for KDM security, which for JES would inherit the limitations of KDM-secure signatures just mentioned, but is not so limited for secure storage.

2 Notation

The empty string is denoted by ε. If x is a (binary) string then $|x|$ is its length. If S is a finite set then $|S|$ denotes its size and $s \xleftarrow{\$} S$ denotes picking an element uniformly from S and assigning it to s. We denote by $\lambda \in \mathbb{N}$ the security parameter and by 1^λ its unary representation. Algorithms are randomized unless otherwise indicated. "PT" stands for "polynomial time," whether for randomized algorithms or deterministic. By $y \leftarrow A(x_1, \ldots; R)$, we denote the operation of running algorithm A on inputs $x_1, \ldots$ and coins R and letting y denote the output. By $y \xleftarrow{\$} A(x_1, \ldots)$, we denote the operation of letting $y \leftarrow A(x_1, \ldots; R)$ for random R. We denote by $[A(x_1, \ldots)]$ the set of points that have positive probability of being output by A on inputs $x_1, \ldots$. Adversaries are algorithms.

We use games in definitions of security and in proofs. A game G (e.g. Fig. 1) has a MAIN procedure whose output (what it returns) is the output of the game. We let Pr[G] denote the probability that this output is the boolean true. The boolean flag bad, if used in a game, is assumed initialized to false.

3 Key-versatile signatures

We define F-keyed signature schemes, for F a family of functions rather than the single function F used for simplicity in Section 1. The requirement is that the secret key sk is an input for an instance fp of the family and the public key $pk = \mathsf{F.Ev}(1^\lambda, fp, sk)$ is the corresponding image under this instance, the instance fp itself specified in public parameters. We intend to use these schemes to add authenticity in a setting where keys (sk, pk) may already be in use for another purpose (such as encryption). We need to ensure that signing will neither lessen the security of the existing usage of the keys nor have its own security be lessened by it. To ensure this strong form of composability, we define simulatability and key-extractability requirements for our F-keyed schemes. The fact that the keys will already be in use for another purpose also means that we do not have the luxury of picking the family F, but must work with an arbitrary family emerging from another setting. The only assumption we will make on F is thus that it is one-way. (This is necessary, else security is clearly impossible.) With the definitions in place, we go on to indicate how to build F-keyed signature schemes for arbitrary, one-way F.

We clarify that being F-keyed under an F assumed to be one-way does not mean that security (simulatability and key-extractability) of the signature scheme is based *solely* on the assumption that F is one-way. The additional assumption is a SE-secure NIZK. (But this itself can be built under standard assumptions.) It is possible to build a signature scheme that is unforgeable assuming only that a given F is one-way [59], but this scheme will not be F-keyed relative to the same F underlying its security, and it will not be simulatable or key-extractable.

SIGNATURE SCHEMES. A signature scheme DS specifies the following PT algorithms: via $pp \leftarrow\!\!\$\ \mathsf{DS.Pg}(1^\lambda)$ one generates public parameters pp common to all users; via $(sk, pk) \leftarrow\!\!\$\ \mathsf{DS.Kg}(1^\lambda, pp)$ a user can generate a secret signing key sk and corresponding public verification key pk; via $\sigma \leftarrow\!\!\$\ \mathsf{DS.Sig}(1^\lambda, pp, sk, M)$ the signer can generate a signature σ on a message $M \in \{0,1\}^*$; via $d \leftarrow \mathsf{DS.Ver}(1^\lambda, pp, pk, M, \sigma)$ a verifier can deterministically produce a decision $d \in \{\mathsf{true}, \mathsf{false}\}$ regarding whether σ is a valid signature of M under pk. Correctness requires that $\mathsf{DS.Ver}(1^\lambda, pp, pk, M, \mathsf{DS.Sig}(1^\lambda, pp, sk, M)) = \mathsf{true}$ for all $\lambda \in \mathbb{N}$, all $pp \in [\mathsf{DS.Pg}(1^\lambda)]$, all $(sk, pk) \in [\mathsf{DS.Kg}(1^\lambda, pp)]$, and all M.

FUNCTION FAMILIES. A function family F specifies the following. Via $fp \leftarrow\!\!\$\ \mathsf{F.Pg}(1^\lambda)$ one can in PT generate a description fp of a function $\mathsf{F.Ev}(1^\lambda, fp, \cdot)$: $\mathsf{F.Dom}(1^\lambda, fp) \to \mathsf{F.Rng}(1^\lambda, fp)$. We assume that membership of x in the non-empty domain $\mathsf{F.Dom}(1^\lambda, fp)$ can be tested in time polynomial in $1^\lambda, fp, x$ and

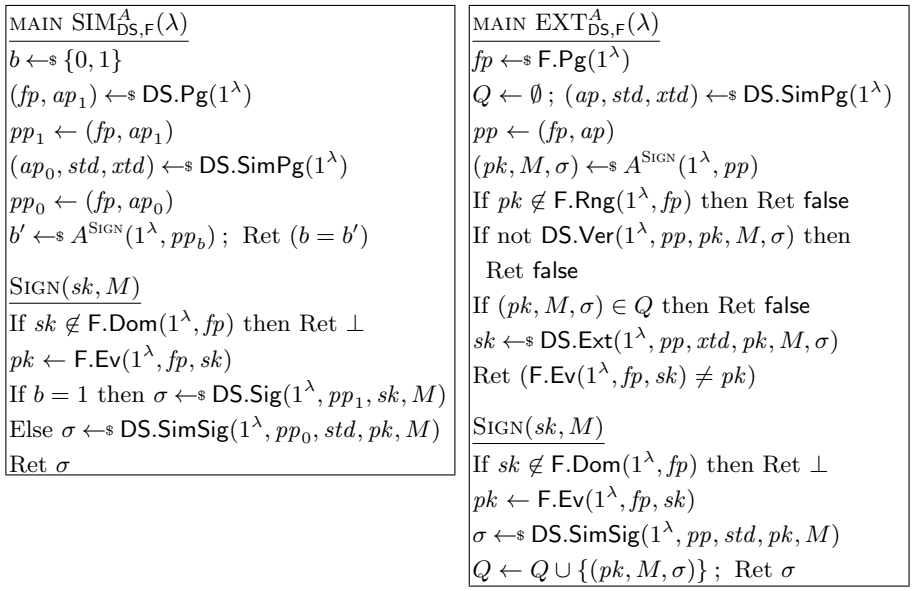

Fig. 1. Games defining security of F-keyed signature scheme DS. Left: Game defining simulatability. Right: Game defining key-extractability.

one can in time polynomial in $1^\lambda, fp$ sample a point $x \leftarrow\$ \mathsf{F.Dom}(1^\lambda, fp)$ from the domain $\mathsf{F.Dom}(1^\lambda, fp)$. The deterministic evaluation algorithm $\mathsf{F.Ev}$ is PT. The range is defined by $\mathsf{F.Rng}(1^\lambda, fp) = \{\, \mathsf{F.Ev}(1^\lambda, fp, x) \;:\; x \in \mathsf{F.Dom}(1^\lambda, fp) \,\}$. Testing membership in the range is not required to be PT. (But is in many examples.) We say that F is one-way or F is a OWF if $\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{F},I}(\cdot)$ is negligible for all PT I, where $\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{F},I}(\lambda) = \Pr[\mathsf{F.Ev}(1^\lambda, fp, x') = y]$ under the experiment $fp \leftarrow\$ \mathsf{F.Pg}(1^\lambda)$; $x \leftarrow\$ \mathsf{F.Dom}(1^\lambda, fp)$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fp, x)$; $x' \leftarrow\$ I(1^\lambda, fp, y)$.

F-KEYED SIGNATURE SCHEMES. Let F be a function family. We say that a signature scheme DS is F-*keyed* if the following are true:

- Parameter compatibility: Parameters pp for DS are a pair $pp = (fp, ap)$ consisting of parameters fp for F and auxiliary parameters ap, these independently generated. Formally, there is a PT *auxiliary parameter generation* algorithm APg such that $\mathsf{DS.Pg}(1^\lambda)$ picks $fp \leftarrow\$ \mathsf{F.Pg}(1^\lambda)$; $ap \leftarrow\$ \mathsf{APg}(1^\lambda)$ and returns (fp, ap).
- Key compatibility: The signing key sk is a random point in the domain of $\mathsf{F.Ev}$ and the verifying key pk its image under $\mathsf{F.Ev}$. Formally, $\mathsf{DS.Kg}(1^\lambda, (fp, ap))$ picks $sk \leftarrow\$ \mathsf{F.Dom}(1^\lambda, fp)$, lets $pk \leftarrow \mathsf{F.Ev}(1^\lambda, fp, sk)$ and returns (sk, pk). ($\mathsf{DS.Kg}$ ignores the auxiliary parameters ap, meaning the keys do not depend on it.)

SECURITY OF F-KEYED SIGNATURE SCHEMES. We require two (strong) security properties of an F-keyed signature scheme DS:

- Simulatable: Under simulated auxiliary parameters and an associated simulation trapdoor *std*, a simulator, given $pk = \mathsf{F.Ev}(1^\lambda, fp, sk)$ and M, can produce a signature σ indistinguishable from the real one produced under sk, when not just M, *but even the secret key* sk, is adaptively chosen by the adversary. Formally, DS is *simulatable* if it specifies additional PT algorithms DS.SimPg (the auxiliary parameter simulator) and DS.SimSig (the signature simulator) such that $\mathbf{Adv}^{\mathrm{sim}}_{\mathsf{DS},\mathsf{F},A}(\cdot)$ is negligible for every PT adversary A, where $\mathbf{Adv}^{\mathrm{sim}}_{\mathsf{DS},\mathsf{F},A}(\lambda) = 2\Pr[\mathrm{SIM}^{A}_{\mathsf{DS},\mathsf{F}}(\lambda)] - 1$ and game SIM is specified on the left-hand side of Fig. 1.
- Key-extractable: Under the same simulated auxiliary parameters and an associated extraction trapdoor *xtd*, an extractor can extract from any valid forgery relative to pk an underlying secret key sk, even when pk *is chosen by the adversary* and the adversary can adaptively obtain simulated signatures *under secret keys of its choice.* Formally, DS is *key-extractable* if it specifies another PT algorithm DS.Ext (the extractor) such that $\mathbf{Adv}^{\mathrm{ext}}_{\mathsf{DS},\mathsf{F},A}(\cdot)$ is negligible for every PT adversary A, where $\mathbf{Adv}^{\mathrm{ext}}_{\mathsf{DS},\mathsf{F},A}(\lambda) = \Pr[\mathrm{EXT}^{A}_{\mathsf{DS},\mathsf{F}}(\lambda)]$ and game EXT is specified on the right-hand side of Fig. 1.

The EXT game includes a possibly non-PT test of membership in the range of the family, but we will ensure that adversaries (who must remain PT) do not perform this test. Our definition of simulatability follows [33, 1, 32]. Those definitions were for general signatures, not F-keyed ones, and one difference is that our simulator can set only the auxiliary parameters, not the full parameters, meaning it does not set *fp*.

Sim+Ext implies unforgeability. The simulatability and key-extractability notions we have defined may seem quite unrelated to the standard unforgeability requirement for signature schemes [46]. As a warm-up towards applying these new conditions, we show that in fact they imply not just the standard unforgeability but strong unforgeability, under the minimal assumption that F is one-way. In [14] we recall the definition of strong unforgeability and formally prove the following:

Theorem 1. *Let* DS *be an* F*-keyed signature scheme that is simulatable and key-extractable. If* F *is one-way then* DS *is strongly unforgeable.*

Construction. A *key-versatile signing schema* is a transform **KvS** that given an arbitrary family of functions F returns an F-keyed signature scheme $\mathsf{DS} = \mathbf{KvS}[\mathsf{F}]$. We want the constructed signature scheme to be simulatable and key-extractable. We now show that this is possible with the aid of appropriate NIZK systems which are themselves known to be possible under standard assumptions.

Theorem 2. *Assume there exist SE NIZK systems for all of* **NP**. *Then there is a key-versatile signing schema* **KvS** *such that if* F *is any family of functions then the signature scheme* $\mathsf{DS} = \mathbf{KvS}[\mathsf{F}]$ *is simulatable and key-extractable.*

In [14] we recall the definition of a SE (Simulation Extractable) NIZK system. SE was called tSE in [40] and is a variant of NIZK-security notions from [47, 36,

60]. We then specify the construction and prove it has the claimed properties. Here we sketch the construction and its history.

The scheme is simple. We define the relation $\mathsf{R}((1^\lambda, fp, pk, M), sk)$ to return true iff $\mathsf{F.Ev}(1^\lambda, fp, sk) = pk$. A signature of M under sk is then a SE-secure NIZK proof for this relation in which the witness is sk and the instance (input) is $(1^\lambda, fp, pk, M)$. The interesting aspect of this construction is that it at first sounds blatantly insecure, since the relation R ignores the message M. Does this not mean that a signature is independent of the message, in which case an adversary could violate unforgeability by requesting a signature σ of a message M under pk and then outputting (M', σ) as a forgery for some $M' \neq M$? What prevents this is the strength of the SE notion of NIZKs. The message M is present in the instance $(1^\lambda, fp, pk, M)$, even if it is ignored by the relation; the proof in turn depends on the instance, making the signature depend on M.

A similar construction of signatures was given in [40] starting from a leakage-resilient hard relation rather than (as in our case) a relation arising from a one-way function. Our construction could be considered a special case of theirs, with the added difference that they use labeled NIZKs with the message as the label while we avoid labels and put the message in the input. The claims established about the construction are however different, with [40] establishing leakage resilience and unforgeability of the signature and our work showing simulatability and key-extractability.

4 JES with no public-key overhead

Let PKE be an arbitrary IND-CCA-secure public-key encryption scheme. Alice has already established a key-pair (sk_e, pk_e) for this scheme, allowing anyone to send her ciphertexts computed under pk_e that she can decrypt under sk_e. She wants now to add signature capability. This is easily done. She can create a key-pair (sk_s, pk_s) for her favorite signature scheme and sign an arbitrary message M under sk_s, verification being possible given pk_s. The difficulty is that her public key is now $pk = (pk_e, pk_s)$. It is not just larger but will require a new certificate. The question we ask is whether we can add signing capability in a way that is more parsimonious with regard to public key size. Technically, we seek a joint encryption and signature (JES) scheme where Alice has a single key-pair (sk, pk), with sk used to decrypt and sign, and pk used to encrypt and verify, each usage secure in the face of the other, and we want pk smaller than that of the trivial solution $pk = (pk_e, pk_s)$. Perhaps surprisingly, we show how to construct a JES scheme with pk-overhead zero, meaning pk is unchanged, remaining pk_e. Previous standard model JES schemes had been able to reduce the pk-overhead only for *specific* starting encryption schemes [49, 57] while our result says the overhead can be zero regardless of the starting encryption scheme.

JES SCHEMES. A joint encryption and signature (JES) scheme JES specifies the following PT algorithms: via $jp \leftarrow\!\$\ \mathsf{JES.Pg}(1^\lambda)$ one generates public parameters jp common to all users; via $(sk, pk) \leftarrow\!\$\ \mathsf{JES.Kg}(1^\lambda, jp)$ a user can generate a secret (signing and decryption) key sk and corresponding public (verification

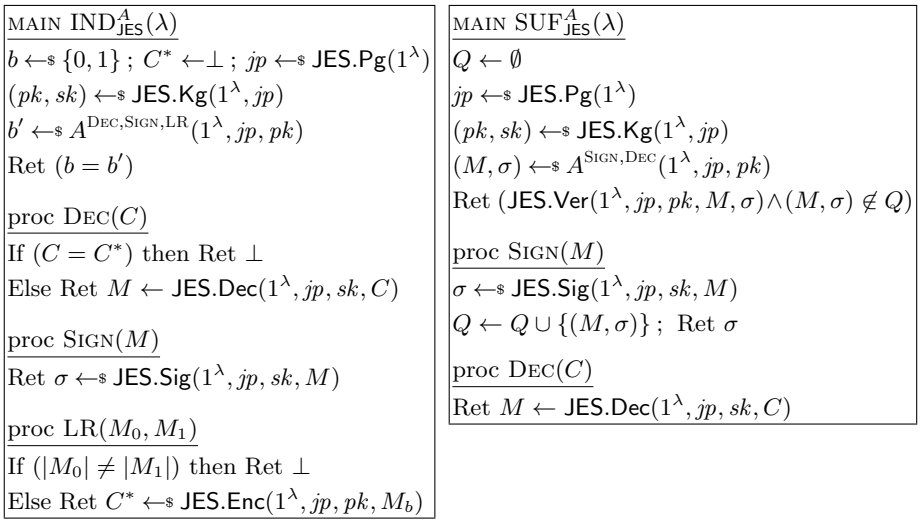

Fig. 2. Games defining security of joint encryption and signature scheme JES. Left: Game IND defining privacy against chosen-ciphertext attack in the presence of a signing oracle. Right: Game SUF defining strong unforgeability in the presence of a decryption oracle.

and encryption) key pk; via $\sigma \leftarrow\!\!\$ \mathsf{JES.Sig}(1^\lambda, jp, sk, M)$ the user can generate a signature σ on a message $M \in \{0,1\}^*$; via $d \leftarrow \mathsf{JES.Ver}(1^\lambda, jp, pk, M, \sigma)$ a verifier can deterministically produce a decision $d \in \{\mathsf{true}, \mathsf{false}\}$ regarding whether σ is a valid signature of M under pk; via $C \leftarrow\!\!\$ \mathsf{JES.Enc}(1^\lambda, jp, pk, M)$ anyone can generate a ciphertext C encrypting message M under pk; via $M \leftarrow \mathsf{JES.Dec}(1^\lambda, jp, sk, C)$ the user can deterministically decrypt ciphertext C to get a value $M \in \{0,1\}^* \cup \{\perp\}$. Correctness requires that $\mathsf{JES.Ver}(1^\lambda, jp, pk, M, \mathsf{JES.Sig}(1^\lambda, jp, sk, M)) = \mathsf{true}$ and that $\mathsf{JES.Dec}(1^\lambda, jp, sk, \mathsf{JES.Enc}(1^\lambda, jp, pk, M)) = M$ for all $\lambda \in \mathbb{N}$, all $jp \in [\mathsf{JES.Pg}(1^\lambda)]$, all $(sk, pk) \in [\mathsf{JES.Kg}(1^\lambda, jp)]$, and all $M \in \{0,1\}^*$. We say that JES is IND-secure if $\mathbf{Adv}^{\text{ind}}_{\mathsf{JES},A}(\cdot)$ is negligible for all PT adversaries A, where $\mathbf{Adv}^{\text{ind}}_{\mathsf{JES},A}(\lambda) = 2\Pr[\text{IND}^{A}_{\mathsf{JES}}(\lambda)] - 1$ and game IND is on the left-hand side of Fig. 2. Here the adversary is allowed only one query to LR. This represents privacy under chosen-ciphertext attack in the presence of a signing oracle. We say that JES is SUF-secure if $\mathbf{Adv}^{\text{suf}}_{\mathsf{JES},A}(\cdot)$ is negligible for all PT adversaries A, where $\mathbf{Adv}^{\text{suf}}_{\mathsf{JES},A}(\lambda) = \Pr[\text{SUF}^{A}_{\mathsf{JES}}(\lambda)]$ and game SUF is on the right-hand side of Fig. 2. This represents (strong) unforgeability of the signature in the presence of a decryption oracle. These definitions are from [49, 57].

THE BASE PKE SCHEME. We are given a public-key encryption scheme PKE, specifying the following PT algorithms: via $fp \leftarrow\!\!\$ \mathsf{PKE.Pg}(1^\lambda)$ one generates public parameters; via $(sk, pk) \leftarrow\!\!\$ \mathsf{PKE.Kg}(1^\lambda, fp)$ a user generates a decryption key sk and encryption key pk; via $C \leftarrow\!\!\$ \mathsf{PKE.Enc}(1^\lambda, fp, pk, M)$ anyone can

generate a ciphertext C encrypting a message M under pk; and via $M \leftarrow \mathsf{PKE.Dec}(1^\lambda, fp, sk, C)$ a user can deterministically decrypt a ciphertext C to get a value $M \in \{0,1\}^* \cup \{\perp\}$. Correctness requires that $\mathsf{PKE.Dec}(1^\lambda, fp, sk, \mathsf{PKE.Enc}(1^\lambda, fp, pk, M)) = M$ for all $\lambda \in \mathbb{N}$, all $fp \in [\mathsf{PKE.Pg}(1^\lambda)]$, all $(sk, pk) \in [\mathsf{PKE.Kg}(1^\lambda, fp)]$, and all $M \in \{0,1\}^*$. We assume that PKE meets the usual notion of IND-CCA security.

Let us say that PKE is *canonical* if the operation $(sk, pk) \leftarrow\!\!\$\ \mathsf{PKE.Kg}(1^\lambda, fp)$ picks sk at random from a finite, non-empty set we denote $\mathsf{PKE.SK}(1^\lambda, fp)$, and then applies to $(1^\lambda, fp, sk)$ a PT deterministic *public-key derivation function* we denote PKE.PK to get pk. Canonicity may seem like an extra assumption, but isn't. First, many (most) schemes are already canonical. This is true for the Cramer-Shoup scheme [35], the Kurosawa-Desmedt scheme [54] and for schemes obtained via the BCHK transform [24] applied to the identity-based encryption schemes of Boneh-Boyen [22] or Waters [61]. Second, if by chance a scheme is not canonical, we can modify it be so. Crucially (for our purposes), the modification *does not change the public key.* (But it might change the secret key.) Briefly, the modification, which is standard, is to use the random coins of the key generation algorithm as the secret key.

Construction. Given canonical PKE as above, we construct a JES scheme JES. The first step is to construct from PKE a function family F as follows: let F.Pg = PKE.Pg, so the parameters of F are the same those of PKE; let F.Dom = PKE.SK, so the domain of F is the space of secret keys of PKE; and let F.Ev = PKE.PK, so the function defined by fp maps a secret key to a corresponding public key. Now let DS be an F-keyed signature scheme that is simulatable and key-extractable. (We can obtain DS via Theorem 2.) Now we define our JES scheme JES. Let JES.Pg = DS.Pg, so parameters for JES have the form $jp = (fp, ap)$, where fp are parameters for F, which by definition of F are also parameters for PKE. Let JES.Kg = DS.Kg. (Keys are those of PKE which are also those of DS.) Let JES.Sig = DS.Sig and JES.Ver = DS.Ver, so the signing and verifying algorithms of the joint scheme JES are inherited from the signature scheme DS. Let $\mathsf{JES.Enc}(1^\lambda, (fp, ap), pk, M)$ return $\mathsf{PKE.Enc}(1^\lambda, fp, pk, M)$ and let $\mathsf{JES.Dec}(1^\lambda, (fp, ap), sk, C)$ return $\mathsf{PKE.Dec}(1^\lambda, fp, sk, C)$, so the encryption and decryption algorithms of the joint scheme JES are inherited from the PKE scheme PKE. Note that the public key of the joint scheme JES is exactly that of PKE, so there is zero public-key overhead. The following says that JES is both IND and SUF secure. The proof is in [14].

Theorem 3. *Let* PKE *be a canonical public-key encryption scheme. Let* F *be defined from it as above. Let* DS *be an* F*-keyed signature scheme, and let* JES *be the corresponding joint encryption and signature scheme constructed above. Assume* PKE *is IND-CCA secure. Assume* DS *is simulatable and key-extractable. Then (1)* JES *is IND secure, and (2)* JES *is SUF secure.*

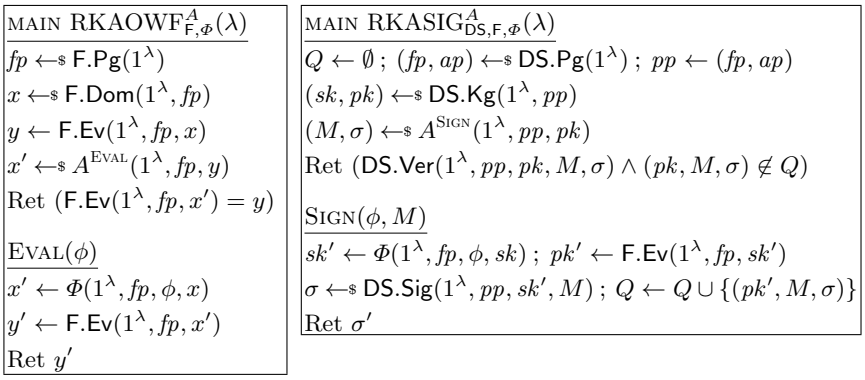

Fig. 3. Games defining Φ-RKA security of a function family F (left) and an F-keyed signature scheme DS (right).

5 RKA-secure signatures from RKA-secure OWFs

RKA security is notoriously hard to provably achieve. Recognizing this, several authors [45, 9] have suggested a bootstrapping approach in which we build higher-level RKA-secure primitives from lower-level RKA-secure primitives. In this vein, a construction of RKA-secure signatures from RKA-secure PRFs was given in [9]. We improve on this via a construction of RKA-secure signatures from RKA-secure one-way functions. The benefit is that (as we will show) many popular OWFs are already RKA secure and we immediately get new RKA-secure signatures.

RKA SECURITY. Let F be a function family. A class of RKD (related-key deriving) functions Φ for F is a PT-computable function that specifies for each $\lambda \in \mathbb{N}$, each $fp \in [\mathsf{F.Pg}(1^\lambda)]$ and each $\phi \in \{0,1\}^*$ a map $\Phi(1^\lambda, fp, \phi, \cdot) : \mathsf{F.Dom}(1^\lambda, fp) \to \mathsf{F.Dom}(1^\lambda, fp)$ called the RKD function described by ϕ. We say that F is Φ-RKA secure if $\mathbf{Adv}^{\text{rka}}_{\mathsf{F},A,\Phi}(\cdot)$ is negligible for every PT adversary A, where $\mathbf{Adv}^{\text{rka}}_{\mathsf{F},A,\Phi}(\lambda) = \Pr[\text{RKAOWF}^A_{\mathsf{F},\Phi}(\lambda)]$ and game RKAOWF is on the left-hand side of Fig. 3. The definition is from [45].

Let DS be an F-keyed signature scheme and let Φ be as above. We say that DS is Φ-RKA secure if $\mathbf{Adv}^{\text{rka}}_{\mathsf{DS},\mathsf{F},A,\Phi}(\cdot)$ is negligible for every PT adversary A, where $\mathbf{Adv}^{\text{rka}}_{\mathsf{DS},\mathsf{F},A,\Phi}(\lambda) = \Pr[\text{RKASIG}^A_{\mathsf{DS},\mathsf{F},\Phi}(\lambda)]$ and game RKASIG is on the right-hand side of Fig. 3. The definition is from [9].

CONSTRUCTION. Suppose we are given a Φ-RKA-secure OWF F and want to build a Φ-RKA-secure signature scheme. For the question to even make sense, RKD functions specified by Φ must apply to the secret signing key. Thus, the secret key needs to be an input for the OWF and the public key needs to be the image of the secret key under the OWF. The main technical difficulty is, given F, finding a signature scheme with this property. But this is exactly what a key-versatile signing schema gives us. The following says that if the signature scheme

produced by this schema is simulatable and key-extractable then it inherits the Φ-RKA security of the OWF. The proof is in [14].

Theorem 4. *Let* DS *be an* F*-keyed signature scheme that is simulatable and key-extractable. Let* Φ *be a class of RKD functions. If* F *is* Φ*-RKA secure then* DS *is also* Φ*-RKA secure.*

Finding Φ-RKA OWFs. Theorem 4 motivates finding Φ-RKA-secure function families F. The merit of our approach is that there are many such families. To enable systematically identifying them, we adapt the definition of key-malleable PRFs of [8] to OWFs. We say that a function family F is *Φ-key-malleable* if there is a PT algorithm M, called a Φ-key-simulator, such that $M(1^\lambda, fp, \phi, \mathsf{F.Ev}(1^\lambda, fp, x)) = \mathsf{F.Ev}(1^\lambda, fp, \Phi(1^\lambda, fp, \phi, x))$ for all $\lambda \in \mathbb{N}$, all $fp \in [\mathsf{F.Pg}(1^\lambda)]$, all $\phi \in \{0,1\}^*$ and all $x \in \mathsf{F.Dom}(1^\lambda, fp)$. The proof of the following is in [14].

Proposition 5. *Let* F *be a function family and* Φ *a class of RKD functions. If* F *is* Φ*-key-malleable and one-way then* F *is* Φ*-RKA secure.*

Previous uses of key-malleability [8, 16] for RKA security required additional conditions on the primitives, such as key-fingerprints in the first case and some form of collision-resistance in the second. For OWFs, it is considerably easier, key-malleability alone sufficing. In [14] we show how to leverage Proposition 5 to show Φ-RKA-security for three popular one-way functions, namely discrete exponentiation in a cyclic group, RSA, and the LWE one-way function, thence obtaining, via Theorem 4, Φ-RKA-secure signature schemes.

6 KDM-secure storage

Services like Dropbox, Google Drive and Amazon S3 offer outsourced storage. Users see obvious benefits but equally obvious security concerns. We would like to secure this storage, even when messages (files needing to be stored) depend on the keys securing them. If privacy is the only concern, existing KDM-secure encryption schemes (e.g., [21, 26, 5, 4, 30, 31, 20, 7, 55, 3, 28, 29, 12, 42, 51]) will do the job. However, integrity is just as much of a concern, and adding it without losing KDM security is challenging. This is because conventional ways of adding integrity introduce new keys and create new ways for messages to depend on keys. Key-versatile signing, by leaving the keys unchanged, will provide a solution.

In [14], we begin by formalizing our goal of encrypted and authenticated outsourced storage secure for key-dependent messages. In our syntax, the user encrypts and authenticates under her secret key, and then verifies and decrypts under the same secret key, with the public key utilized by the server for verification. Our requirement for KDM security has two components: IND for privacy and SUF for integrity. With the definitions in hand, we take a base KDM-secure encryption scheme and show how, via a key-versatile signature, to obtain storage schemes meeting our goal. Our resulting storage schemes will achieve KDM security with respect to the same class of message-deriving functions Φ as the underlying encryption scheme. Also, we will assume only CPA KDM security

of the base scheme, yet achieve CCA KDM privacy for the constructed storage scheme. Interestingly, our solution uses a *public-key* base encryption scheme, even though the privacy component of the goal is symmetric and nobody but the user will encrypt. This allows us to start with KDM privacy under keys permitting signatures through key-versatile signing. This represents a novel application for public-key KDM-secure encryption. We refer the reader to [14] for details.

Acknowledgments

Bellare was supported in part by NSF grants CNS-0904380, CCF-0915675, CNS-1116800 and CNS-1228890. Meiklejohn was supported in part by NSF grant CNS-1237264. Part of this work was done while Thomson was at Royal Holloway, University of London supported in part by EPSRC Leadership Fellowship EP/H005455/1.

References

1. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, Aug. 2010.
2. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 83–107. Springer, Apr. / May 2002.
3. B. Applebaum. Key-dependent message security: Generic amplification and completeness. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, May 2011.
4. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Aug. 2009.
5. M. Backes, M. Dürmuth, and D. Unruh. OAEP is secure under key-dependent messages. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 506–523. Springer, Dec. 2008.
6. M. Backes, B. Pfitzmann, and A. Scedrov. Key-dependent message security under active attacks - brsim/uc-soundness of dolev-yao-style encryption with key cycles. *Journal of Computer Security*, 16(5):497–530, 2008.
7. B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 423–444. Springer, May 2010.
8. 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.
9. 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.
10. M. Bellare and S. Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 194–211. Springer, Aug. 1990.

11. M. Bellare and V. T. Hoang. Adaptive witness encryption and asymmetric password-based cryptography. Cryptology ePrint Archive, Report 2013/704, 2013.
12. M. Bellare and S. Keelveedhi. Authenticated and misuse-resistant encryption of key-dependent data. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 610–629. Springer, Aug. 2011.
13. 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.
14. M. Bellare, S. Meiklejohn, and S. Thomson. Key-versatile signatures and applications: RKA, KDM and joint Enc/Sig. Cryptology ePrint Archive, Report 2013/326, 2013. Full version of this abstract.
15. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, Oct. 2008.
16. M. Bellare, K. G. Paterson, and S. Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. In *ASIACRYPT*, pages 331–348, 2012.
17. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.
18. 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.
19. 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.
20. 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.
21. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Aug. 2003.
22. D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, May 2004.
23. 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.
24. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.
25. 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.
26. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision diffie-hellman. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, Aug. 2008.
27. 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.
28. Z. Brakerski, S. Goldwasser, and Y. T. Kalai. Black-box circular-secure encryption beyond affine functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 201–218. Springer, Mar. 2011.

29. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Aug. 2011.
30. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, Apr. 2009.
31. R. Canetti, Y. T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 52–71. Springer, Feb. 2010.
32. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable signatures: Complex unary transformations and delegatable anonymous credentials. Cryptology ePrint Archive, Report 2013/179, 2013.
33. M. Chase and A. Lysyanskaya. On signatures of knowledge. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 78–96. Springer, Aug. 2006.
34. J.-S. Coron, M. Joye, D. Naccache, and P. Paillier. Universal padding schemes for RSA. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 226–241. Springer, Aug. 2002.
35. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
36. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, Aug. 2001.
37. A. De Santis, G. Di Crescenzo, and G. Persiano. Necessary and sufficient assumptions for non-interactive zero-knowledge proofs of knowledge for all np relations. In *Automata, Languages and Programming*, pages 451–462. Springer, 2000.
38. A. De Santis and G. Persiano. Zero-knowledge proofs of knowledge without interaction. In *Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on*, pages 427–436. IEEE, 1992.
39. J. P. Degabriele, A. Lehmann, K. G. Paterson, N. P. Smart, and M. Strefler. On the joint security of encryption and signature in EMV. In O. Dunkelman, editor, *CT-RSA 2012*, volume 7178 of *LNCS*, pages 116–135. Springer, Feb. / Mar. 2012.
40. Y. Dodis, K. Haralambiev, A. López-Alt, and D. Wichs. Efficient public-key cryptography in the presence of key leakage. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 613–631. Springer, Dec. 2010.
41. J. Fan, Y. Zheng, and X. Tang. A single key pair is adequate for the zheng signcryption. In U. Parampalli and P. Hawkes, editors, *ACISP 11*, volume 6812 of *LNCS*, pages 371–388. Springer, July 2011.
42. D. Galindo, J. Herranz, and J. L. Villar. Identity-based encryption with master key-dependent message security and leakage-resilience. In S. Foresti, M. Yung, and F. Martinelli, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 627–642. Springer, Sept. 2012.
43. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 467–476. ACM, 2013.
44. 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.

45. 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.
46. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988.
47. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Dec. 2006.
48. J. Groth and R. Ostrovsky. Cryptography in the multi-string model. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 323–341. Springer, Aug. 2007.
49. S. Haber and B. Pinkas. Securely combining public-key cryptosystems. In *ACM CCS 01*, pages 215–224. ACM Press, Nov. 2001.
50. K. Haralambiev. *Efficient Cryptographic Primitives for Non-Interactive Zero-Knowledge Proofs and Applications*. PhD thesis, New York University, May 2011.
51. D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 520–536. Springer, 2013. To appear.
52. 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.
53. Y. Komano and K. Ohta. Efficient universal padding techniques for multiplicative trapdoor one-way permutation. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 366–382. Springer, Aug. 2003.
54. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, Aug. 2004.
55. T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 507–526. Springer, May 2011.
56. M. G. Muñiz and R. Steinwandt. Security of signature schemes in the presence of key-dependent messages. *Tatra Mt. Math. Publ.*, 47:15–29, 2010.
57. 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.
58. P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *ACM CCS 02*, pages 98–107. ACM Press, Nov. 2002.
59. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.
60. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, Oct. 1999.
61. 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.
62. Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 165–179. Springer, Aug. 1997.