Public Key Encryption Against Related Key Attacks

Hoeteck Wee*

George Washington University
hoeteck@gwu.edu

Abstract. In this work, we present efficient public-key encryption schemes resilient against linear related key attacks (RKA) under standard assumptions and in the standard model. Specifically, we obtain encryption schemes based on hardness of factoring, BDDH and LWE that remain secure even against an adversary that may query the decryption oracle on linear shifts of the actual secret key. Moreover, the ciphertext overhead is only an additive constant number of group elements.

1 Introduction

The traditional model for security assumes that the internal states of the honest parties are completely hidden from the adversary. We often also extend the same assumption to cryptographic hardware devices such as a RSA SecurID token; here, we assume the internal states to be both completely hidden and protected from the adversary. However, recent timing, 'cold-boot' and virtual-machine attacks demonstrated that physical side-channels can leak partial information about internal states of program executions [32, 25, 40]. Similarly, given physical access to a hardware device, we can use fault injection techniques to tamper with and induce modifications to the internal state of the device [10, 8]. When an adversary tampers with the key stored in a cryptographic hardware device and subsequently observes the outcome of the cryptographic primitive under this modified key, we have a related-key attack (RKA) [21, 7]. The key here may be a signing key of a certificate authority or SSL server or a decryption key for an encryption scheme.

RKA security for public-key encryption. In this work, we study public-key encryption schemes secure against related-key attacks (RKA), under the definition given by Bellare et. al [7]. The attack is on the secret key, so we are considering a chosen-ciphertext related-key attack (CC-RKA). The decryption oracle refuses to act only when the ciphertext it is given matches the challenge ciphertext *and* the derived key equals the real one. We will also consider weak CC-RKA security, where the decryption oracle refuses to act whenever the ciphertext it is given matches the challenge ciphertext. Note that both notions imply IND-CCA security [39, 19], which correspond to the special case where the related-key attack uses the identity function.

We view the system as having the following components: algorithms (code), public parameters, public/secret key pairs. Of these, only the public and secret keys are subject to RKAs. The public parameters are system-wide, meaning fixed beforehand and independent of users. In an implementation, these parameters could be hardwired into the algorithm code and stored on tamper-proof hardware, or distributed via some public channel where tampering is infeasible or could be

* Supported by NSF CAREER Award CNS-0953626.

easily detected. In our constructions, the decryption algorithms do not use the public key and therefore we will only consider attacks on secret keys. We note that our model is the same as that considered in prior works [4, 7], though it is by no means the only possible model.

1.1 Our results

We present the first public-key encryption schemes resilient against linear related key attacks under standard assumptions and in the standard model (see Appendix A for examples of such attacks). Specifically, we obtain encryption schemes based on hardness of factoring and BDDH that remain secure even against an adversary that may query the decryption oracle on linear shifts of the actual secret key. In addition, we present schemes based on DDH and LWE that achieve the weaker notion of RKA security where the adversary is not allowed to query the decryption oracle on the challenge ciphertext.

Moreover, in all these schemes, the ciphertext overhead is only an additive constant number of group elements. Our factoring-based scheme is also the first RKA-secure primitive based on standard number-theoretic assumptions related to factoring, as well as the first from search assumptions not related to lattices. (The latter is somewhat surprising in lieu of the negative results in [22], showing that certain natural classes of constructions based on search assumptions cannot achieve RKA-pseudorandomness).

Warm-up. The starting point of our constructions are CCA-secure encryption schemes in which the decryption of a ciphertext C using a secret key $\phi(\text{SK})$ – where ϕ denotes a linear shift – equals the decryption of some other (efficiently computable) ciphertext C' using the original secret key SK. We refer to this property as key homomorphism. Roughly speaking, this enables us to reduce the CC-RKA-security of the scheme to its CCA-security. The same high-level strategy of exploiting homomorphism was also used in [4, 3] to achieve RKA security for pseudorandom functions and private-key encryption respectively.

The above strategy breaks down whenever the ciphertext C' equals challenge ciphertext in the CCA-security game. We address this problem with the following modifications:

- We work with a tag-based notion of CCA-security [34, 30], where we derive the tag using a strong one-time signature scheme and add a signature to the ciphertext. In addition, we require that the two ciphertexts above C and C' (where C' is derived from C via key homomorphism) share the same tag. We may then consider two cases: if C shares the same tag as the challenge ciphertext, then the one-time signature scheme tells us that C must equal the challenge ciphertext. On the other hand, if C has a different tag from the challenge ciphertext, then so does C' and we can decrypt C' using the decryption oracle in the CCA-security game. This suffices for weak CC-RKA security, where the RKA decryption oracle refuses to act whenever the ciphertext it is given matches the challenge ciphertext.
- In order to achieve "full fledged" CC-RKA security, we need to handle the case where the ciphertext C equals the challenge ciphertext but $\phi(\text{SK}) \neq \text{SK}$. Here, we simply stipulate that the challenge ciphertext is an invalid ciphertext under any key $\text{SK}' \neq \text{SK}$; we refer to this property as finger-printing (c.f. [4, 7]). In other words, a random valid ciphertext (along with the public parameters but not the public key) uniquely determines a consistent secret key.

At this point, it suffices to describe how we instantiate the underlying building blocks, namely a tag-based CCA-secure encryption scheme that achieves both finger-printing and key-homomorphism, as well as an efficient strong one-time signature scheme.

Achieving finger-printing. As it turns out, the Cramer-Shoup CCA-secure constructions [15, 16] do not satisfy the finger-printing; this is in some sense inherent since the smoothness requirement in hash proof systems essentially stipulate the secret key has some residual entropy given only its evaluation on a NO instance of the underlying subset membership problem (but not the public key). Instead, we turn to constructions of CCA-secure public-key encryption based on the "all-but-one extraction" paradigm, starting with [9], and further developed in [12, 11, 30, 38, 26, 1, 31, 42, 35]. In these constructions, the secret key is often only a single group element, which makes achieving finger-printing much simpler. While the Cramer-Shoup framework inherently relies on decisional assumptions e.g., the Decisional Diffie-Hellman (DDH) assumption or the quadratic residuosity assumption, the "all-but-one extraction" paradigm admits instantiations from search assumptions, such as factoring. Search assumptions encompass a larger class of intractable problems than decisional assumptions.

Achieving key homomorphism. This leads us to our final technical hurdle, namely that CCA-secure public-key encryption schemes based on search assumptions may not be key-homomorphic. Take for instance the Hofheinz-Kiltz factoring-based CCA-secure scheme [26]; it is not key-homomorphic because the underlying Blum-Blum-Shub PRG is not homomorphic. As it turns out, the "all-but-one extraction" paradigm allows us to overcome this hurdle too – informally, the trapdoor decryption algorithm allows us to recover the *seed* of the PRG (for CCA security, it suffices to recover the *output* of the PRG). For this reason, we present our schemes via the framework of adaptive trapdoor relations [42, 31], which seems particularly suited for our analysis, as it abstracts the "all-but-one" aspect for achieving CCA-security, allowing us to directly focus on the new challenges posed by CC-RKA-security. For the concrete instantiations of CC-RKA-secure encryption, we look at known instantiations of adaptive trapdoor relations given in [42, 35]; we show that the ones based on hardness of factoring and BDDH satisfy key homomorphism and finger-printing, and that the ones based on DDH and LWE satisfy key homorphism.

One-time signatures. As a result of independent interest, we present a new strong one-time signature scheme based on hardness of factoring, which is inspired by Groth's one-time signature based on hardness of discrete log [24]. In Appendix B, we also sketch a generic construction of strong one-time signatures starting from any Σ-protocol. In the application to CCA-security and our CC-RKA-secure schemes, we want to design one-time signature schemes where the total cost of key generation and signing is small. In our factoring-based scheme, the signing algorithm does not require knowing the factorization of the modulus and we may therefore use a modulus from the public parameter instead of generating RSA modulus from scratch (which requires a linear number of exponentiations).

1.2 Discussion

There is a general transformation for achieving security against linear related key attacks via algebraic manipulation detection (AMD) codes [18, 20] – in the case of encryption, this requires modifying the key generation algorithm of a CCA-secure encryption scheme, so that the stored secret key is the encoded version of the original secret key, using such a code (thereby increasing the secret key size). The encoding has the property that with high probability any linear shift of a valid codeword can be detected (and in those cases the new decryption algorithm would simply reject). Our constructions achieve several advantages over this generic approach: first, the key generation algorithm coincides with existing CCA-secure encryption schemes. This offers compatibility with existing public key set-ups. Second, we avoid the blow-up in key sizes. Finally, the existing constructions of AMD codes only work over finite fields, which are not applicable to the constructions based on hardness of factoring.

Perspective. We do not know if linear relations capture any meaningful attacks in practice, or whether security against linear relations would be useful for specific practical applications. As such, we regard our results largely as proof of concept, demonstrating that we can indeed achieve RKA-security for a non-trivial class of functions while paying only a small overhead in efficiency and without changing existing public-key set-ups. Interestingly, our constructions exploit the fact that certain encryption schemes are susceptible to linear related-key attacks (as implied by key homomorphism) to obtain encryption schemes that are secure against linear related-key attacks. Ironically, the fact that we do not know how to achieve RKA security for larger classes of relations is intimately related to the fact that we do not know of theoretical attacks for such classes!

Additional related work. The works of Lucks, Goldenberg and Liskov, and Bellare, Cash and Miller [33, 22, 7] gave constructions of RKA-secure primitives from RKA-secure building blocks, but provided no new constructions of the latter and hence of the former. Also, a number of works gave RKA-secure schemes in the standard model, notably symmetric encryption [2, 3], signatures [23] (based on q-ary assumption) in addition to PRFs [4]; these schemes all rely lattices and Diffie-Hellman type assumptions, none of these are based on number-theoretic assumptions. There are also feasibility results on RKA-secure public-key encryption based on non-standard assumptions, e.g. [28] as well as results on tamper-resilient UC-secure computation [14]. We also point out here that encryption schemes secure against linear related-key attacks have also found applications in garbled circuits used in secure computation [3, 29].

Organization. We present our main construction in Section 3. We present the instantiations from various classes of assumptions in Sections 5 through 6.

2 Preliminaries

Strong one-time signatures. For a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv.OTS}^{\mathcal{A}}(\lambda)$ to be:

$$\Pr\left[\begin{array}{l|l} & (\textsc{vksig}, \textsc{sksig}) \leftarrow \mathsf{SignKeyGen}(1^\lambda); \\ \mathsf{Verify}(\textsc{vksig}, M', \sigma') = 1 & M \leftarrow \mathcal{A}(\textsc{vksig}); \\ \text{and } (M', \sigma') \neq (M, \sigma) & \sigma \leftarrow \mathsf{Sign}(\textsc{sksig}, M); \\ & (M', \sigma') \leftarrow \mathcal{A}(\sigma) \end{array}\right]$$

A signature scheme is a *strong one-time signature* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv.OTS}^{\mathcal{A}}(\lambda)$ is a negligible function in λ.

Adaptive trapdoor relations. Informally, trapdoor functions are a family of functions $\{\mathsf{F}_{\textsc{fid}}\}$ that are easy to sample, compute and invert with trapdoor, but hard to invert without the trapdoor (we always assume that the functions are injective). In the tag-based setting, the function takes an additional input, namely the tag; also, the trapdoor is independent of the tag. A family of *adaptive trapdoor functions* [31] is one that remains one-way even if the adversary is given access to an inversion oracle, except the adversary cannot query the oracle on the same tag as that in the challenge. In a trapdoor relation, instead of requiring that $\mathsf{F}_{\textsc{fid}}$ be efficiently computable, we only require that we can efficiently sample from the distribution $(s, \mathsf{F}_{\textsc{fid}}(\textsc{tag}, s))$ for a random s given $\textsc{fid}, \textsc{tag}$.

More precisely, a family of (tag-based) *adaptive trapdoor relations* [42] is given by a family of injective functions $\{\mathsf{F}_{\textsc{fid}}\}$ that satisfies the following properties:

(TRAPDOOR GENERATION.) There is an efficient randomized algorithm TDG that outputs a random $(\textsc{fid}, \textsc{tid})$.

(PUBLIC SAMPLING.) There is an efficient randomized algorithm PSamp that on input $(\textsc{fid}, \textsc{tag})$, outputs $(s, \mathsf{F}_{\textsc{fid}}(\textsc{tag}, s))$ for a random s.

(TRAPDOOR INVERSION.) There is an efficient algorithm TdInv such that for all $(\textsc{fid}, \textsc{tid}) \leftarrow \mathsf{TDG}$ and for all $\textsc{tag}, y$, computes $\mathsf{TdInv}(\textsc{tid}, \textsc{tag}, y) = \mathsf{F}^{-1}_{\textsc{fid}}(\textsc{tag}, y)$.

(ADAPTIVE ONE-WAYNESS.) For all efficient stateful adversaries $\mathcal{A}$, the following quantity is negligible in λ:

$$\Pr\left[s = s' : \begin{array}{l} \textsc{tag}^* \leftarrow \mathcal{A}(1^\lambda); \\ (\textsc{fid}, \textsc{tid}) \leftarrow_{\mathrm{R}} \mathsf{TDG}(1^\lambda); \\ (s, y) \leftarrow_{\mathrm{R}} \mathsf{PSamp}(\textsc{fid}, \textsc{tag}^*); \\ s' \leftarrow \mathcal{A}^{\mathsf{F}^{-1}_{\textsc{fid}}(\cdot,\cdot)}(\textsc{fid}, y) \end{array}\right]$$

where $\mathcal{A}$ is allowed to query $\mathsf{F}^{-1}_{\textsc{fid}}(\cdot, \cdot)$ on any tag different from $\textsc{tag}^*$.

It is convenient to work with the following stronger notion of *adaptive pseudorandomness* [37], where the adversary has to distinguish $\mathsf{G}(s)$ from random given y and an inversion oracle, for some

pseudorandom generator G associated with the family $\{\mathsf{F}_{\text{FID}}\}$. There is indeed a generic way to obtain adaptive pseudorandomness from adaptive one-wayness via the Goldreich-Levin hard-core bit (since the proof relativizes with respect to the inversion oracle). However, for the concrete instantiations we consider here, there are more efficient ways to derive multiple hard-core bits.

(ADAPTIVE PSEUDORANDOMNESS.) For all efficient stateful adversaries $\mathcal{A}$, the following quantity is negligible in λ:

$$\Pr\left[b = b' : \begin{array}{l} \text{TAG}^* \leftarrow \mathcal{A}(1^\lambda); \\ (\text{FID}, \text{TID}) \leftarrow_{\text{R}} \mathsf{TDG}(1^\lambda); \\ (s, y) \leftarrow_{\text{R}} \mathsf{PSamp}(\text{FID}, \text{TAG}^*); \\ K_0 := \mathsf{G}(s); K_1 \leftarrow_{\text{R}} \{0,1\}^\lambda; \\ b \leftarrow_{\text{R}} \{0,1\}; \\ b' \leftarrow \mathcal{A}^{\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)}(\text{FID}, y, K_b) \end{array}\right] - \frac{1}{2}$$

where $\mathcal{A}$ is allowed to query $\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)$ on any tag different from TAG^*.

2.1 RKA Security

Related-key derivation functions. Following [5], a class of Φ of related-key deriving functions (RKDFs) is a finite set of functions, all with the same domain and range that could possibly depend on the public parameter PP. The class of functions should also admit an efficient membership test, and its functions should be efficiently computable. For our concrete instantiations, we consider the class Φ^+ of linear shifts.

CC-RKA security. We follow the definition of related-key attack (RKA) security from [7, 4]. For a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv.RKA.PKE}^{\mathcal{A},\Phi}(\lambda)$ to be:

$$\Pr\left[b = b' : \begin{array}{l} \text{PP} \leftarrow \mathsf{Setup}(1^\lambda);\ (\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(\text{PP}); \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{RKA.Dec}(\text{SK},\cdot,\cdot)}(\text{PP}, \text{PK}),\ |m_0| = |m_1|; \\ b \leftarrow_{\text{R}} \{0,1\}; \\ C^* \leftarrow \mathsf{Enc}(\text{PK}, m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{RKA.Dec}(\text{SK},\cdot,\cdot)}(C^*) \end{array}\right] - \frac{1}{2}$$

where $\mathsf{RKA.Dec}(\text{SK}, \cdot, \cdot)$ is an oracle that on input (ϕ, C): returns $\mathsf{Dec}(\phi(\text{SK}), C)$. We restrict the adversary $\mathcal{A}$ to only make queries (ϕ, C) such that $\phi \in \Phi$ and $(\phi(\text{SK}), C) \neq (\text{SK}, C^*)$. An encryption scheme is said to be Φ-*CC-RKA secure* if for all PPT $\mathcal{A}$, the advantage $\mathsf{Adv.RKA.PKE}^{\mathcal{A},\Phi}(\lambda)$ is a negligible function in λ.

Weaker CC-RKA security. We also consider weak CC-RKA security, where in the security experiment, we further restrict the adversary $\mathcal{A}$ to only make queries (ϕ, C) such that $\phi \in \Phi$ and $C \neq C^*$ where C^* is the challenge ciphertext. Previously, we also allow queries (ϕ, C^*) as long as $\phi(\text{SK}) \neq \text{SK}$.

3 Realization from Adaptive Trapdoor Relations

In this section, we present our constructions of RKA-secure encryption via adaptive trapdoor relations. We begin by introducing two additional notions for adaptive trapdoor relations.

Φ-Key homomorphism. We say that $\{\mathsf{F}_{\text{FID}}\}$ is Φ-key homomorphic if there is a PPT algorithm T such that with overwhelming probability over PP, for all $\phi \in \Phi$ and all TID, TAG, y:

$$\mathsf{TdInv}(\phi(\text{TID}), \text{TAG}, y) = \mathsf{TdInv}(\text{TID}, \text{TAG}, T(\text{PP}, \phi, \text{TAG}, y))$$

In fact, a weaker formulation that asserts an oracle PPT algorithm T that outputs $\mathsf{TdInv}(\phi(\text{TID}), \text{TAG}, y)$ given oracle access to $\mathsf{TdInv}(\text{TID}, \text{TAG}, \cdot)$ suffices for our proofs. This latter formulation is more similar to the formulation of key-malleability in [4, Section 3.1] for achieving RKA-security for pseudorandom functions. A similar notion also appears in [3] for symmetric-key encryption.

Φ-Fingerprinting. Informally, Φ-fingerprinting stipulates that any attempt to maul TID invalidates a random output of $\mathsf{F}_{\text{FID}}(\cdot)$. More formally, for a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv.FP}^{\mathcal{A},\Phi}(\lambda)$ to be:

$$\Pr\left[\begin{array}{l|l} & \text{TAG}^* \leftarrow \mathcal{A}(\text{PP}); \\ \mathsf{TdInv}(\phi(\text{TID}), \text{TAG}^*, y) \neq \perp & (\text{FID}, \text{TID}) \leftarrow \mathsf{TDG}(\text{PP}); \\ \text{and } \phi \in \Phi \text{ and } \phi(\text{TID}) \neq \text{TID} & (s, y) \leftarrow_{\text{R}} \mathsf{PSamp}(\text{FID}, \text{TAG}^*); \\ & \phi \leftarrow \mathcal{A}(\text{PP}, \text{FID}, \text{TID}, y); \end{array}\right]$$

A trapdoor relation admits a Φ-*fingerprint* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv.FP}^{\mathcal{A},\Phi}(\lambda)$ is a negligible function in λ. We stress that in the above experiment, the adversary receives TID, which it can use to compute s from y.

3.1 Our construction

We present our construction in Fig 1, which is the same as the construction of CCA-secure encryption schemes from adaptive trapdoor relations via strong one-time signatures, as given in [31, 42].

Theorem 1. *Suppose the following hold:*

1. $\{\mathsf{F}_{\text{FID}}\}$ *is a family of adaptive trapdoor relations;*
2. $\{\mathsf{F}_{\text{FID}}\}$ *is* Φ*-key homomorphic;*
3. $\{\mathsf{F}_{\text{FID}}\}$ *admits a* Φ*-fingerprinting;*
4. $(\mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is a strong one-time signature scheme.*

Then, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as given in Fig 1 is a* Φ*-CC-RKA secure public-key encryption scheme. Moreover, if all of the conditions hold apart from* Φ*-fingerprinting, then* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as given in the above construction is a* Φ*-weak-CC-RKA secure public-key encryption scheme.*

RKA PKE

$\mathsf{Gen}(\text{PP})$: Run $\mathsf{TDG}(\text{PP}) \to (\text{FID}, \text{TID})$. Output $(\text{PK}, \text{SK}) := (\text{FID}, \text{TID})$.

$\mathsf{Enc}(\text{PK}, m)$: On input PK and a message m:

1. Run $\mathsf{SignKeyGen}(\text{PP}) \to (\text{VKSIG}, \text{SKSIG})$;
2. Run $\mathsf{PSamp}(\text{PK}, \text{VKSIG}) \to (s, y)$;
3. Compute $\psi := \mathsf{G}(s) \oplus m$;
4. Run $\mathsf{Sign}(\text{SKSIG}, y \| \psi) \to \sigma$;

Output as ciphertext $\quad \text{VKSIG} \| \sigma \| y \| \psi$

$\mathsf{Dec}(\text{SK}, C)$: On input SK and a ciphertext $C = \text{VKSIG} \| \sigma \| y \| \psi$,

1. Output $\perp$ if $\mathsf{Verify}(\text{VKSIG}, y \| \psi, \sigma) = \texttt{reject}$.
2. Compute $s := \mathsf{TdInv}(\text{TID}, \text{VKSIG}, y)$. Output $\perp$ if $s = \perp$.

Otherwise, output $\quad \mathsf{G}(s) \oplus \psi$

Fig. 1. CC-RKA security from adaptive trapdoor relations

We observe that correctness of the encryption scheme follows readily from the correctness of trapdoor inversion. Φ-CC-RKA security follows from the next technical claim. After the proof, we explain how to deduce Φ-weak-CC-RKA security without relying on Φ-fingerprinting.

Lemma 1. *Let $\mathcal{A}$ be an adversary against the Φ-CC-RKA security of the above encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that makes at most Q oracle queries. Then, we can construct an adversary $\mathcal{B}_0$ against the strong one-time security of $(\mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{Verify})$, an adversary $\mathcal{B}_1$ against Φ-fingerprinting of $\{\mathsf{F}_{\text{FID}}\}$, and an adversary $\mathcal{B}_2$ against adaptive pseudorandomness of $\{\mathsf{F}_{\text{FID}}\}$ and G such that:*

$$\mathsf{Adv.RKA.PKE}^{\mathcal{A},\Phi}(\lambda) \leq \mathsf{Adv.OTS}^{\mathcal{B}_0}(\lambda) + \mathsf{Adv.FP}^{\mathcal{B}_1,\Phi}(\lambda) + \mathsf{Adv.Adaptive.PRG}^{\mathcal{B}_2}(\lambda)$$

The running times of $\mathcal{B}_0$ and $\mathcal{B}_1$ are that of $\mathcal{A}$ plus an additional polynomial overhead that grows linearly with Q. The running time of $\mathcal{B}_2$ is similar to that of $\mathcal{A}$, and $\mathcal{B}_2$ makes at most Q oracle queries.

Proof. In the following, we write $C^* = \text{VKSIG}^* \| \sigma^* \| y^* \| \psi^*$ to denote the ciphertext in the Φ-CC-RKA experiment. We proceed via a sequence of games. We start with Game 0 as in the Φ-CC-RKA experiment and end up with a game where the view of $\mathcal{A}$ is statistically independent of the challenge bit b. The sequence of games is analogous to those for obtaining CCA security from all-but-one extractable hash proofs and adaptive trapdoor functions [42, 31]; the main difference lies in handling the RKA queries in the first two games.

GAME 1: ELIMINATING TAG REUSE. We replace the decapsulation mechanism $\mathsf{RKA.Dec}$ with $\mathsf{RKA.Dec}'$ that outputs $\perp$ on ciphertexts $\text{VKSIG} \| \sigma \| y \| \psi$ such that $\text{VKSIG} = \text{VKSIG}^*$ but otherwise proceeds

like $\mathsf{RKA.Dec}$. We show that Games 0 and 1 are computationally indistinguishable, by arguing that $\mathsf{RKA.Dec}$ and $\mathsf{RKA.Dec}'$ essentially agree on all inputs $\textsc{vksig}\|\sigma\|y\|\psi$. We consider four cases depending on the input:

- Case 1: $\textsc{vksig} \neq \textsc{vksig}^*$. Here, $\mathsf{RKA.Dec}$ and $\mathsf{RKA.Dec}'$ agree by definition of $\mathsf{RKA.Dec}'$.
- Case 2: $\textsc{vksig} = \textsc{vksig}^*, (\sigma, y\|\psi) = (\sigma^*, y^*\|\psi^*)$ and $\phi(\textsc{sk}) = \textsc{sk}$. Such queries are ruled out by definition of the Φ-CC-RKA security game.
- Case 3: $\textsc{vksig} = \textsc{vksig}^*, (\sigma, y\|\psi) \neq (\sigma^*, y^*\|\psi^*)$. Here, by the security of the signature scheme, we have:

  $$\Pr[\mathsf{Verify}(\textsc{vksig}, y\|\psi, \sigma) = 1] \leq \mathsf{Adv.OTS}(\lambda)$$

  Therefore, $\mathsf{RKA.Dec}$ outputs $\perp$ except with negligible probability.
- Case 4: $\textsc{vksig} = \textsc{vksig}^*, (\sigma, y\|\psi) = (\sigma^*, y^*\|\psi^*)$ and $\phi(\textsc{sk}) \neq \textsc{sk}$. Here, by the Φ-fingerprinting property, we have:

  $$\Pr[\mathsf{TdInv}(\phi(\textsc{sk}), \textsc{vksig}^*, y) \neq \perp] \leq \mathsf{Adv.FP}(\lambda)$$

  (Here, we use the fact that the adversary in the Φ-fingerprinting experiment is given $\textsc{tid}$, which is needed to simulate the decryption oracle.) Therefore, $\mathsf{RKA.Dec}$ outputs $\perp$ except with negligible probability.

GAME 2: DECRYPTING USING $\mathsf{F}^{-1}_{\textsc{fid}}(\cdot,\cdot)$. Next, we simulate oracle access to $\mathsf{RKA.Dec}'$ using oracle access to $\mathsf{F}^{-1}_{\textsc{fid}}(\cdot,\cdot)$ as follows: on input $(\phi, \textsc{vksig}\|\sigma\|y\|\psi)$,

1. If $\textsc{vksig} = \textsc{vksig}^*$ or $\mathsf{Verify}(\textsc{vksig}, y\|\psi, \sigma) = 0$, output $\perp$.
2. Compute $s' := \mathsf{F}^{-1}_{\textsc{fid}}(\textsc{vksig}, T(\textsc{pp}, \phi, \textsc{vksig}, y))$. Output $\perp$ if $s' = \perp$.
3. Otherwise, output $\psi := \mathsf{G}(s') \oplus \psi$.

Note that we only query $\mathsf{F}^{-1}_{\textsc{fid}}(\cdot,\cdot)$ on tags different from $\textsc{vksig}^*$. Observe that

$$\begin{aligned} s' &= \mathsf{F}^{-1}_{\textsc{fid}}(\textsc{vksig}, T(\textsc{pp}, \phi, \textsc{vksig}, y)) \\ &= \mathsf{TdInv}(\textsc{tid}, \textsc{vksig}, T(\textsc{pp}, \phi, \textsc{vksig}, y)) && \text{using trapdoor inversion} \\ &= \mathsf{TdInv}(\phi(\textsc{tid}), \textsc{vksig}, y) && \text{using } \Phi\text{-key homomorphism} \end{aligned}$$

Correctness of the simulation follows readily, and thus Games 1 and 2 are identically distributed.

GAME 3: REPLACING $\mathsf{G}(\cdot)$ WITH RANDOM. In the computation of $\mathsf{Enc}(\textsc{pk}, m_b)$ in the $\mathsf{Adv.RKA.PKE}$ experiment, we replace $\psi^* := \mathsf{G}(s^*) \oplus m_b$ with $\psi^* := K \oplus m_b$ where $K \leftarrow_{\textsc{r}} \{0,1\}^\lambda$. Then, Games 2 and 3 and computationally indistinguishable by adaptive pseudorandomness using $\textsc{vksig}^*$ as the selective tag.

We conclude by observing that in Game 3, the distribution of ϕ^* is statistically independent of the challenge bit b. Hence, the probability that $b' = b$ is exactly $1/2$. □

Observe that in the above proof, we only used Φ-fingerprinting in the analysis of Game 1 Case 4. For Φ-weak-CC-RKA security, the queries for this case are ruled out by definition and therefore we do not need Φ-fingerprinting.

4 Instantiations from Hardness of Factoring

Fix a Blum integer $N = PQ$ for λ-bit primes $P, Q \equiv 3 \pmod 4$ such that $P = 2p+1$ and $Q = 2q+1$ for primes p, q. Let $\mathbb{J}_N$ denote the subgroup of $\mathbb{Z}_N^*$ with Jacobi symbol $+1$, and let $\mathbb{QR}_N$ denote the subgroup of quadratic residues. Observe that $|\mathbb{J}_N| = 2pq = 2|\mathbb{QR}_N|$. Following [27], we work over the cyclic group of signed quadratic residues, given by the quotient group $\mathbb{QR}_N^+ := \mathbb{J}_N / \pm 1$. $\mathbb{QR}_N^+$ is a cyclic group of order pq and is efficiently recognizable (by verifying that the Jacobi symbol is $+1$). Here, we use a hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$, though we will treat the output of H as a number in $\mathbb{Z}_{2^\lambda}$.

4.1 Strong one-time signature

For main construction in Section 3, we require efficient strong one-time signature schemes, where the total computational complexity for key generation and signing is small. In addition, we want short verification key and signatures. Previous factoring-based one-time signatures [41, 36] require generating an RSA modulus during key generation, which is computationally expensive. We provide a new construction that uses a public modulus. For the one-time signature, we can work with any Blum integer $N = PQ$, that is, we do not require that P, Q be safe primes.

$\mathsf{SignKeyGen}(\text{PP}), \text{PP} = (N)$:	$\mathsf{Sign}(\text{SKSIG}, m)$:	$\mathsf{Verify}(\text{VKSIG}, m, (e, w))$
$\text{SKSIG} := (s_0, s_1, x) \leftarrow_{\text{R}} \mathbb{QR}_N^+$ $(u_0, u_1, c) := (s_0^{2^\lambda}, s_1^{2^\lambda}, x^{2^\lambda})$ $\text{VKSIG} := (u_0, u_1, c)$ return $(\text{VKSIG}, \text{SKSIG})$	$e \leftarrow_{\text{R}} \mathbb{Z}_{2^\lambda}$ $w := x \cdot s_0^e \cdot s_1^{\mathsf{H}(m)+e \bmod 2^\lambda}$ return $(e, w) \in \mathbb{Z}_{2^\lambda} \times \mathbb{QR}_N^+$	check $w^{2^\lambda} = c \cdot u_0^e \cdot u_1^{\mathsf{H}(m)+e \bmod 2^\lambda}$

Fig. 2. Factoring-based strong one-time signature

Theorem 2. *Suppose factoring Blum integers is hard on average and H is collision resistant. Then, the protocol* $(\mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *described above is a strong one-time signature scheme for signing messages $m \in \{0,1\}^*$ with perfect correctness.*

Proof. Correctness is straight-forward. To establish security, we first describe two simulators $\mathsf{Sim}_0, \mathsf{Sim}_1$ that given (u_0, s_1) and (s_0, u_1) respectively, simulates the verification key and the signature on a single message.

$\mathsf{Sim}_0(N, u_0, s_1)$: Pick $\tilde{w} \leftarrow_{\text{R}} \mathbb{QR}_N^+, e \leftarrow_{\text{R}} \mathbb{Z}_{2^\lambda}$. Output

$$\text{VKSIG} := (u_0, u_1, \tilde{w}^{2^\lambda} \cdot u_0^{-e})$$

When asked to sign a message $m \in \{0,1\}^*$, output

$$\left(e,\ \tilde{w} \cdot s_1^{H(m)+e \bmod 2^\lambda}\right)$$

$\mathsf{Sim}_1(N, s_0, u_1)$: Pick $\tilde{w} \leftarrow_{\mathrm{R}} \mathbb{QR}_N^+, \tilde{e} \leftarrow_{\mathrm{R}} \mathbb{Z}_{2^\lambda}$. Output

$$\textsc{vksig} := (u_0, u_1, \tilde{w}^{2^\lambda} \cdot u_1^{-\tilde{e}})$$

When asked to sign a message $m \in \{0,1\}^*$, output

$$\left(\tilde{e} - H(m) \bmod 2^\lambda,\ \tilde{w} \cdot s_0^{\tilde{e} - H(m) \bmod 2^\lambda}\right)$$

It is straight-forward to check that the outputs of both Sim_0 and Sim_1 are identically distributed to the output of a honestly generated VKSIG and an honestly generated signature on a single message. Now, we consider several cases for a forgery (e', w') on m':

- $m' = m$, same $e' = e$: then, $w' = w$.
- $e \neq e'$: in Sim_0, the forgery will allow us to compute the 2^λ'th root of $u_0^{e-e'}$ where $|e - e'| < 2^\lambda$, i.e.:

$$(u_0^{e-e'})^{2^{-\lambda}} = \frac{w}{w'} \cdot \frac{s_1^{\mathsf{H}(m')+e'}}{s_1^{\mathsf{H}(m)+e}}$$

  Using Shamir's GCD in the exponent algorithm, this value along with u_0 allows us to recover a square root of u_0.
- $e = e'$, $\mathsf{H}(m) \neq \mathsf{H}(m')$: in Sim_1, extract a square root of u_1, analogous to the previous case.
- $e = e'$, $\mathsf{H}(m) = \mathsf{H}(m')$, but $m' \neq m$: contradict collision resistance of H.

That is, we can show that if an adversary outputs a forgery with probability ϵ, then we can compute a square root of a random challenge u with probability roughly $\epsilon/2$ as follows: we pick $b \leftarrow_{\mathrm{R}} \{0,1\}$, run Sim_b with u as u_b and choosing a random s_{1-b}. □

4.2 Adaptive trapdoor relations

$\mathsf{TDG}(\textsc{pp})$, $\textsc{pp} = (N, g)$:	$\mathsf{PSamp}(\textsc{fid}, \textsc{tag}; r)$:	$\mathsf{TdInv}(\textsc{tid}, \textsc{tag}, u\|\tau)$:
$\textsc{tid} \leftarrow_{\mathrm{R}} [(N-1)/4]$	$(s, u) := (g^{2^\ell r}, g^{2^{\lambda+\ell} r})$	check $u, \tau \in \mathbb{QR}_N^+$
$\textsc{fid} := g^{2^{\lambda+\ell} \cdot \textsc{tid}}$	$\tau := (\textsc{fid} \cdot g^{\textsc{tag}})^r$	check $\tau^{2^{\lambda+\ell}} = u^{\textsc{tag}+2^{\lambda+\ell} \cdot \textsc{tid}}$
return $(\textsc{fid}, \textsc{tid})$	return $(s, u\|\tau)$	find $a, b, c \in \mathbb{Z}$: $2^c = a\textsc{tag} + b2^{\lambda+\ell}$
$\mathsf{G}(s) := \mathsf{BBS}(s)$		return $(\tau^a \cdot u^{b - a \cdot \textsc{tid}})^{2^{\ell-c}}$

Fig. 3. An adaptive trapdoor relation based on factoring [42, 26]

The class Φ^+. The functions $\phi_\Delta : [N/4] \to \mathbb{Z}$ in this class are indexed by $\Delta \in [-N/4, N/4]$, where $\phi_\Delta(\textsc{tid}) := \textsc{tid} + \Delta$.

Φ^+*-key homomorphism.* Observe that for all $\textsc{tid}, \Delta \in \mathbb{Z}$, all $\textsc{tag}$ and all $u, \tau \in \mathbb{QR}_N^+$:

$$\mathsf{TdInv}(\textsc{tid} + \Delta, \textsc{tag}, u\|\tau) = \mathsf{TdInv}(\textsc{tid}, \textsc{tag}, u\|(\tau \cdot u^{-\Delta}))$$

The above equality follows from the fact that TdInv returns $s = u^{2^{-\lambda}}$ in both sides of the equation when the following condition holds

$$\tau^{2^{\lambda+\ell}} = u^{\mathrm{TAG}+2^{\lambda+\ell}\cdot(\mathrm{TID}+\Delta)} \iff (\tau \cdot u^{-\Delta})^{2^{\lambda+\ell}} = u^{\mathrm{TAG}+2^{\lambda+\ell}\cdot\mathrm{TID}}$$

and $\perp$ otherwise.

Φ^+-*fingerprinting.* We establish a stronger statement, namely Φ-fingerprinting for any class Φ of efficiently computable functions $\phi : [(N-1)/4] \to \{-N, \ldots, N\}$. Fix an adversary $\mathcal{A}$. Let $y = u \| \tau$ denote the challenge in the security experiment. Furthermore, suppose $\mathcal{A}$ outputs ϕ such that $\phi(\mathrm{TID}) \neq \mathrm{TID}$ and $\mathsf{TdInv}(\phi(\mathrm{TID}), \mathrm{TAG}^*, y) \neq \perp$. This means:

$$\tau^{2^{\lambda+\ell}} = u^{\mathrm{TAG}^*+2^{\lambda+\ell}\cdot\mathrm{TID}} = u^{\mathrm{TAG}^*+2^{\lambda+\ell}\cdot\phi(\mathrm{TID})}$$

and thus

$$u^{\mathrm{TID}} = u^{\phi(\mathrm{TID})}$$

With probability $1 - O(\sqrt{N})$, both g and u are generators of $\mathbb{QR}_N^+$. This means $\mathrm{TID} = \phi(\mathrm{TID}) \pmod{\phi(N)/4}$. This would allow us to factor N.

5 Instantiations from Diffie-Hellman Assumptions

5.1 Strong one-time signature from hardness of discrete log

For completeness, we present here Groth's one-time signature scheme [24, Section 5.4]; we modified the underlying algebra in order to clarify the similarity to our factoring-based scheme. Here, we use a hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q$. The scheme is secure if computing discrete log is hard on average and H is collision resistant.

$\mathsf{SignKeyGen}(\mathrm{PP})$, $\mathrm{PP} = (\mathbb{G}, q, g)$:	$\mathsf{Sign}(\mathrm{SKSIG}, m)$:	$\mathsf{Verify}(\mathrm{VKSIG}, m, (e, w))$
$\mathrm{SKSIG} := (s_0, s_1, x) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^3$	$e \leftarrow_{\mathrm{R}} \mathbb{Z}_q$	check $g^w = c \cdot u_0^e \cdot u_1^{\mathsf{H}(m)+e}$
$(u_0, u_1, c) := (g^{s_0}, g^{s_1}, g^x)$	$w := x + e \cdot s_0 + (\mathsf{H}(m) + e) \cdot s_1$	
$\mathrm{VKSIG} := (u_0, u_1, c)$	return $(e, w) \in \mathbb{Z}_q \times \mathbb{Z}_q$	
return $(\mathrm{VKSIG}, \mathrm{SKSIG})$		

Fig. 4. Discrete-log-based strong one-time signature [24]

5.2 Instantiations from BDDH

The class Φ^+. The functions $\phi_\Delta : Z_q \to \mathbb{Z}_q$ in this class are indexed by $\Delta \in \mathbb{Z}_q$, where $\phi_\Delta(\mathrm{TID}) := \mathrm{TID} + \Delta$.

$\mathsf{TDG}(\text{PP}), \text{PP} = (\mathbb{G}, q, g, g^\alpha, g^\gamma)$:	$\mathsf{PSamp}(\text{FID}, \text{TAG}; r)$:	$\mathsf{TdInv}(\text{TID}, \text{TAG}, u\|\tau)$:
$\text{TID} \leftarrow_{\text{R}} \mathbb{Z}_q$; $\text{FID} := g^{\text{TID}}$	$(s, u) := ((g^\alpha)^r, g^r)$	compute $s := (\tau \cdot u^{-\text{TID}})^{\text{TAG}^{-1}}$
return (FID, TID)	$\tau := (\text{FID} \cdot (g^\alpha)^{\text{TAG}})^r$	if $e(g, s) = e(g^\alpha, u)$:
$\mathsf{G}(s) := e(s, g^\gamma)$	return $(s, u\|\tau)$	return s, else $\perp$

Fig. 5. An adaptive trapdoor relation based on BDDH [42, 9]

Φ^+*-key homomorphism.* Observe that for all $\text{TID}, \Delta \in \mathbb{Z}_q$, all TAG and all $u, \tau \in \mathbb{G}$:

$$\mathsf{TdInv}(\text{TID} + \Delta, \text{TAG}, u\|\tau) = \mathsf{TdInv}(\text{TID}, \text{TAG}, u\|(\tau \cdot u^{-\Delta}))$$

The above equality follows from the fact that on both sides of the equation, TdInv computes s such that

$$s^{\text{TAG}} = \tau \cdot u^{-(\text{TID}+\Delta)} = (\tau \cdot u^{-\Delta}) \cdot u^{-\text{TID}}$$

Φ^+*-fingerprinting.* We establish a stronger statement, namely Φ-fingerprinting for any class Φ of functions $\phi : \mathbb{Z}_q \to \mathbb{Z}_q$. Fix an adversary $\mathcal{A}$. Let $y = u\|\tau$ denote the challenge in the security experiment. Furthermore, suppose $\mathcal{A}$ outputs ϕ such that $\mathsf{TdInv}(\phi(\text{TID}), \text{TAG}^*, y) \neq \perp$. This means:

$$(\tau \cdot u^{-\text{TID}})^{\text{TAG}^{*-1}} = (\tau \cdot u^{-\phi(\text{TID})})^{\text{TAG}^{*-1}}$$

and thus

$$u^{\text{TID}} = u^{\phi(\text{TID})}$$

Hence, $\text{TID} = \phi(\text{TID})$.

5.3 Weakly CC-RKA-secure schemes from DDH

$\mathsf{TDG}(\text{PP}), \text{PP} = (\mathbb{G}, q, g)$:	$\mathsf{PSamp}(\text{FID}, \text{TAG}; r)$:	$\mathsf{TdInv}(\text{TID}, \text{TAG}, u\|\tau_0\|\tau_1)$:
$\text{TID} := (\alpha, \beta, \gamma_0, \gamma_1) \leftarrow_{\text{R}} \mathbb{Z}_q^4$	$(s, u) := ((g^\alpha)^r, g^r)$	compute $s_0 := (\tau_0 \cdot u^{-\gamma_0})^{\text{TAG}^{-1}}$
$\text{FID} := (g^\alpha, g^\beta, g^{\gamma_0}, g^{\gamma_1})$	$\tau_0 := (g^{\gamma_0} \cdot (g^\alpha)^{\text{TAG}})^r$	compute $s_1 := (\tau_1 \cdot u^{-\gamma_1})^{\text{TAG}^{-1}}$
return (FID, TID)	$\tau_1 := (g^{\gamma_1} \cdot (g^\beta)^{\text{TAG}})^r$	if $s_0 = u^\alpha \wedge s_1 = u^\beta$:
$\mathsf{G}(s) := s$	return $(s, u\|\tau_0\|\tau_1))$	return s_0, else $\perp$

Fig. 6. An adaptive trapdoor relation based on DDH [42, 13]

The class Φ^+*.* The functions $\phi_\Delta : \mathbb{Z}_q^4 \to \mathbb{Z}_q^4$ in this class are indexed by $\Delta \in \mathbb{Z}_q^4$, where $\phi_\Delta(\text{TID}) := \text{TID} + \Delta$.

Φ^+*-key homomorphism.* Observe that for all $\text{TID}, \Delta \in \mathbb{Z}_q^4$, all TAG and all $u, \tau_0, \tau_1 \in \mathbb{G}$:

$$\mathsf{TdInv}(\text{TID} + \Delta, \text{TAG}, u\|\tau_0\|\tau_1) = \mathsf{TdInv}(\text{TID}, \text{TAG}, u\|(\tau_0 \cdot u^{-\Delta})\|(\tau_1 \cdot u^{-\Delta}))$$

6 Instantiations from LWE

We rely on a construction from [35, 1]. Here, $\mathbf{G}$ is a public matrix with special structure for which the bounded-distance decoding problem is easy.

$\mathsf{TDG}(\text{PP}), \text{PP} = (\mathbf{G}, \overline{\mathbf{A}}) \in \mathbb{Z}_q^{n \times (w+\overline{m})}$:
- $\text{TID} := \mathbf{R} \leftarrow_{\text{R}} \mathcal{D}_q^{\overline{m} \times w}$
- $\text{FID} := \mathbf{A}' := \overline{\mathbf{A}}\mathbf{R}$
- return (FID, TID)

$\mathsf{PSamp}(\text{FID}, \text{TAG})$:
- $\mathbf{u} := \overline{\mathbf{A}}^\top \mathbf{s} + \mathbf{e},\ \mathbf{s} \leftarrow_{\text{R}} \mathbb{Z}_q^n$
- $\mathbf{A}_{\text{TAG}} := \mathbf{A}' + \text{TAG} \cdot \mathbf{G}$
- $\mathbf{v} := \mathbf{A}_{\text{TAG}}^\top \mathbf{s} + \mathbf{e}'$
- return $(\mathbf{s}, \mathbf{u} \| \mathbf{v})$

$\mathsf{TdInv}(\text{TID}, \text{TAG}, \mathbf{u} \| \mathbf{v})$:
- compute $\mathbf{v}' = \mathbf{v} - \mathbf{R}^\top \mathbf{u}$
- solve $\mathbf{s}$ s.t. $\mathbf{v}' \approx \text{TAG} \cdot \mathbf{G}^\top \mathbf{s}$
- if $\|\mathbf{v}' - \text{TAG} \cdot \mathbf{G}^\top \mathbf{s}\|, \|\mathbf{u} - \overline{\mathbf{A}}^\top \mathbf{s}\|$ are small:
- return $\mathbf{s}$ else $\perp$

Fig. 7. An adaptive trapdoor relation based on LWE [35]

The class Φ^+. The functions $\phi_\Delta : \mathbb{Z}_q^{\overline{m} \times w} \to \mathbb{Z}_q^{\overline{m} \times w}$ in this class are indexed by $\Delta \in \mathbb{Z}_q^{\overline{m} \times w}$, where $\phi_\Delta(\text{TID}) := \text{TID} + \Delta$.

Φ^+*-key homomorphism.* Observe that for all $\mathbf{R}, \Delta \in \mathbb{Z}_q^{\overline{m} \times w}$, all TAG and all $\mathbf{u} \| \mathbf{v} \in \mathbb{Z}_q^{\overline{m}+w}$:

$$\mathsf{TdInv}(\mathbf{R} + \Delta, \text{TAG}, \mathbf{u} \| \mathbf{v}) = \mathsf{TdInv}(\mathbf{R}, \text{TAG}, \mathbf{u} \| (\mathbf{v} - \Delta^\top \mathbf{u}))$$

The above equality just follows from the fact that on both sides of the equation, TdInv computes

$$\mathbf{v}' = \mathbf{v} - (\mathbf{R} + \Delta)^\top \mathbf{u} = (\mathbf{v} - \Delta^\top \mathbf{u}) - \mathbf{R}^\top \mathbf{u}$$

Acknowledgments. I would like to thank David Cash, Dennis Hofheinz, Payman Mohassel and Daniel Wichs for helpful discussions and the anonymous referees for detailed and helpful feedback.

References

[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[2] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[3] B. Applebaum, Y. Ishai, and E. Kushilevitz. Semantic security under related-key attacks and applications. In *ICS*, pages 45–55, 2011.

[4] M. Bellare and D. Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, pages 666–684, 2010.

[5] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *EUROCRYPT*, pages 491–506, 2003.

[6] M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In *Public Key Cryptography*, pages 201–216, 2007.

[7] M. Bellare, D. Cash, and R. Miller. A comparitive study of achievability of security against related-key attack. In *Asiacrypt*, pages 486–503, 2011. Also Cryptology ePrint Archive, Report 2011/252.

[8] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, pages 513–525, 1997.

[9] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[10] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. In *EUROCRYPT*, pages 37–51, 1997.

[11] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS*, pages 320–329, 2005.

[12] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[13] D. Cash, E. Kiltz, and V. Shoup. The Twin Diffie-Hellman problem and applications. *J. Cryptology*, 22(4): 470–504, 2009.

[14] S. G. Choi, A. Kiayias, and T. Malkin. BiTR: Built-in tamper resilience. In *Asiacrypt*, 2011.

[15] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[16] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002. Also, Cryptology ePrint Archive, Report 2001/085.

[17] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.

[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 *EUROCRYPT*, pages 471–488, 2008.

[19] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[20] S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.

[21] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *TCC*, 2004.

[22] D. Goldenberg and M. Liskov. On related-secret pseudorandomness. In *TCC*, pages 255–272, 2010.

[23] V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In *TCC*, pages 182–200, 2011.

[24] J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, pages 444–459, 2006.

[25] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM*, 52 (5):91–98, 2009.

[26] D. Hofheinz and E. Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT*, pages 313–332, 2009.

[27] D. Hofheinz and E. Kiltz. The group of signed quadratic residues and applications. In *CRYPTO*, pages 637–653, 2009.
[28] Y. T. Kalai, B. Kanukurthi, and A. Sahai. Cryptography with tamperable and leaky memory. In *CRYPTO*, pages 373–390, 2011.
[29] J. Katz and L. Malka. Constant-round private function evaluation with linear complexity. In *Asiacrypt*, pages 556–571, 2011. Also Cryptology ePrint Archive, Report 2010/528.
[30] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.
[31] E. Kiltz, P. Mohassel, and A. O'Neil. Adaptive trapdoor functions and chosen ciphertext security. In *EUROCRYPT*, pages 673–692, 2010.
[32] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO*, pages 104–113, 1996.
[33] S. Lucks. Ciphers secure against related-key attacks. In *FSE*, pages 359–370, 2004.
[34] P. D. MacKenzie, M. K. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions, and applications. In *TCC*, pages 171–190, 2004.
[35] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012. To appear. Also, Cryptology ePrint Archive, Report 2011/501.
[36] P. Mohassel. One-time signatures and chameleon hash functions. In *Selected Areas in Cryptography*, pages 302–319, 2010.
[37] O. Pandey, R. Pass, and V. Vaikuntanathan. Adaptive one-way functions and applications. In *CRYPTO*, pages 57–74, 2008.
[38] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.
[39] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.
[40] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *ACM Conference on Computer and Communications Security*, pages 199–212, 2009.
[41] A. Shamir and Y. Tauman. Improved online/offline signature schemes. In *CRYPTO*, pages 355–367, 2001.
[42] H. Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *CRYPTO*, pages 314–332, 2010.

A Related-key Attacks on Cramer-Shoup

We point out two simple linear RKAs on the Cramer-Shoup CCA-secure encryption scheme [15] based on DDH; these attacks also highlight some of the main technical difficulities in achieving RKA security. We stress that this does not undermine the Cramer-Shoup scheme in any way, since the scheme was not designed to resist RKAs. The scheme is as follows:

$\mathsf{Gen}(\mathrm{PP}), \mathrm{PP} = (\mathbb{G}, q, g_1, g_2)$:	$\mathsf{Enc}(\mathrm{PK}, m; r)$:	$\mathsf{Dec}(\mathrm{SK}, u\|v\|w\|e)$:
$\mathrm{SK} := (x, y, a, b, a', b') \leftarrow_{\mathrm{R}} \mathbb{Z}_q^6$	$(u, v, w) := (g_1^r, g_2^r, h^r \cdot m)$	$t := \mathsf{TCR}(u\|v\|w)$
$(h, c, d) := (g_1^x g_2^y, g_1^a g_2^b, g_1^{a'} g_2^{b'})$	$t := \mathsf{TCR}(u\|v\|w)$	if $u^{a+t\cdot a'} \cdot v^{b+t\cdot b'} = e$:
$\mathrm{PK} := (h, c, d)$	$e := (cd^t)^r$	return $w/(u^x v^y)$ else $\perp$
return $(\mathrm{PK}, \mathrm{SK})$	return $u\|v\|w\|e$	

Suppose we are given a valid encryption (u, v, w, e) of some unknown message m. The following attacks allow us to recover m by making decryption queries on a related secret key. Specifically, for any $\Delta \in \mathbb{Z}_q$,

- if we change a in the secret key to $a + \Delta$, observe that $(u, v, w, e \cdot u^{\Delta})$ decrypts to m under the modified secret key.
- if we change x in the secret key to $x + \Delta$, observe that (u, v, w, e) decrypts to $m \cdot u^{-\Delta}$ under the modified secret key.

In both cases, we can easily recover the message m given the output of the decryption algorithm on the modified secret key.

B Strong One-Time Signatures from Σ Protocols

We sketch here a generic construction of one-time signatures for Σ protocols. We start with a Σ-protocol Π for any one-way relation. Applying the CDS-transform [17], we may derive another Σ-protocol that given a pair of instances (u_0, u_1), proves knowledge for one of the two witnesses. Now consider the following signature scheme: the verification key is (u_0, u_1, c_0, c_1) and a signature on a message M is a triplet (e, a_0, a_1) such that (c_0, e, a_0) and $(c_1, M \oplus e, a_1)$ are accepting transcripts for Π for the instances u_0 and u_1 respectively.

We show that this scheme is one-time unforgeable; moreover, if Π has unique responses, then the scheme is one-time strongly unforgeable. The proof of security is very simple: we generate (u_0, u_1) along with the witness for u_b, where $b \in \{0, 1\}$ is chosen at random. Using the witness, we can simulate the signature oracle for a single message. Given a forgery, we can extract a witness to one of u_0, u_1, which with probability $1/2$, is different from the one we already know.

Constructions of one-time signatures from Σ-protocols were also given in [36, 6]. However, the transformation given here as well as our factoring-based instantiation appear to be novel.