On the Impossibility of Tight Cryptographic Reductions

Christoph Bader, Tibor Jager(✉), Yong Li, and Sven Schäge(✉)

Horst Görtz Institute for IT Security, Ruhr-University Bochum, Bochum, Germany
sschaege@gmail.com

Abstract. The existence of *tight* reductions in cryptographic security proofs is an important question, motivated by the theoretical search for cryptosystems whose security guarantees are truly independent of adversarial behavior and the practical necessity of concrete security bounds for the theoretically-sound selection of cryptographic parameters. At Eurocrypt 2002, Coron described a *meta-reduction* technique that allows to prove the *impossibility* of tight reductions for certain digital signature schemes. This seminal result has found many further interesting applications. However, due to a technical subtlety in the argument, the applicability of this technique beyond digital signatures in the *single-user* setting has turned out to be rather limited. We describe a new meta-reduction technique for proving such impossibility results, which improves on known ones in several ways. It enables interesting novel applications, including a formal proof that for certain cryptographic primitives (including public-key encryption/key encapsulation mechanisms and digital signatures), the security loss incurred when the primitive is transferred from an idealized single-user setting to the more realistic multi-user setting is *impossible* to avoid, and a lower tightness bound for non-interactive key exchange protocols. Moreover, the technique allows to rule out tight reductions from a very general class of non-interactive complexity assumptions. Furthermore, the proofs and bounds are simpler than in Coron's technique and its extensions.

1 Introduction

Provable Security. In modern cryptography, new cryptosystems are usually constructed together with a *proof of security*. Usually this security proof consists of a reduction Λ (in a complexity-theoretic sense), which turns an efficient adversary $\mathcal{A}$ into a machine $\Lambda^{\mathcal{A}}$ solving a well-studied, assumed-to-be-hard computational problem. Under the assumption that this computational problem is not efficiently solvable, this implies that the cryptosystem is secure. This approach is usually called "provable security", it is inspired by the analysis of relations between computational problems in complexity theory, and allows to show that

T. Jager—Supported by DFG grant JA 2445/1-1.

S. Schäge—Supported by UbiCrypt, DFG grant GRK 1817/1.

M. Fischlin and J.-S. Coron (Eds.): EUROCRYPT 2016, Part II, LNCS 9666, pp. 273–304, 2016.
DOI: 10.1007/978-3-662-49896-5_10

breaking the security of a cryptosystem is at least as hard as solving a certain well-defined hard computational problem.

The Security Loss in Reduction-Based Security Proofs. The "quality" of a reduction can be measured by comparing the running time and success probability of $\Lambda^{\mathcal{A}}$ to the running time and success probability of attacker $\mathcal{A}$. Ideally, $\Lambda^{\mathcal{A}}$ has about the same running time and success probability as $\mathcal{A}$. However, most security proofs describe reductions where $\Lambda^{\mathcal{A}}$ has either a significantly larger running time or a significantly smaller success probability than $\mathcal{A}$ (or both). Thus, the reduction "loses" efficiency and/or efficacy.

Since provable security is inspired by classical complexity theory, security proofs have traditionally been formulated asymptotically. The running time and success probability of Turing machines are modeled as functions in a *security parameter* $k \in \mathbb{N}$. Let $t_{\Lambda^{\mathcal{A}}}(k)$ denote the running time and $\epsilon_{\Lambda^{\mathcal{A}}}(k)$ denote the success probability of $\Lambda^{\mathcal{A}}$. Likewise, let $t_{\mathcal{A}}(k)$ and $\epsilon_{\mathcal{A}}(k)$ denote the running time and success probability of $\mathcal{A}$. Then it holds that

$$t_{\Lambda^{\mathcal{A}}}(k)/\epsilon_{\Lambda^{\mathcal{A}}}(k) = \ell(k) \cdot t_{\mathcal{A}}(k)/\epsilon_{\mathcal{A}}(k)$$

for some "loss" $\ell(k)$. A reduction Λ is considered *efficient*, if its loss $\ell(k)$ is bounded by a polynomial. Note that in this approach the concrete size of polynomial ℓ (i.e., its degree and the size of its coefficients) does not matter. As common in classical complexity theory, it was considered sufficient to show that ℓ is polynomially-bounded.

Concrete Security Proofs, the Notion of Tightness, and Its Relevance. In order to deploy a cryptosystem in practice, the size of cryptographic parameters (like for instance the length of moduli or the size of underlying algebraic groups) has to be selected. However, the asymptotic approach described above does not allow to derive concrete recommendations for such parameters, as it only shows that sufficiently large parameters *exist*. This is because the size of parameters depends on the concrete value of ℓ, the loss of the reduction. A larger loss requires larger parameters.

The more recent approach, termed *concrete security*, makes the concrete security loss of a reduction explicit. This allows to derive concrete recommendations for parameters in a theoretically sound way (see e.g. [7] for a detailed treatment). Ideally, $\ell(k)$ is constant. In this case the reduction is said to be *tight*.[1] The existence of cryptosystems whose security is independent of deployment parameters is of course an interesting theoretical question in its own right. Moreover, it has a strong practical motivation, because the tightness of a reduction directly influences the selection of the size of cryptographic parameters, and thus has a direct impact to the efficiency of cryptosystems.

[1] When speaking of tight reductions in this paper, we mean tight reductions from non-interactive computational problems, like integer factorization, the discrete logarithm problem, etc., rather than (often trivial) tight reductions from interactive or contrived non-standard computational problems, which sometimes are very similar to the assumption that the cryptosystem is secure.

Coron's Result and Its Refinements. Coron [18] considered the existence of tight reductions for *unique*[2] signature schemes in the single user setting, and described a "rewinding argument" (cf. Goldwasser *et al.* [27]), which allowed to prove lower tightness bounds for such signature schemes. In particular, Coron considered "simple"[3] reductions, which convert a forger F breaking the security[4] of a unique signature scheme into a machine solving a computationally hard problem Π. He showed that any such reduction yields an algorithm $\mathcal{B}$ solving Π *directly* with probability $\epsilon_{\mathcal{B}}$, where

$$\epsilon_{\mathcal{B}} \geq \epsilon_{\Lambda} - \frac{\epsilon_{\mathsf{F}}}{\mathsf{exp}(1) \cdot n} \cdot \left(1 - \frac{n}{|\mathcal{M}|}\right)^{-1}. \tag{1}$$

Here ϵ_{Λ} is the success probability of Λ, ϵ_{F} is the success probability of the signature forger F used by Λ, n is the number of signatures queried by F in the EUF-CMA security experiment, and $|\mathcal{M}|$ is the size of the message space. Note that if $|\mathcal{M}| \gg n$, which is a reasonable for signature schemes, then the bound in (1) essentially implies that the success probability of ϵ_{Λ} of the reduction can not substantially exceed $\epsilon_{\mathsf{F}}/(\mathsf{exp}(1) \cdot n)$, unless there exists an algorithm $\mathcal{B}$ solving Π efficiently. The latter, however, contradicts the hardness assumption on Π. This result was later revisited by Kakvi and Kiltz [31], and generalized by Hofheinz *et al.* [30] to (non-unique) signature schemes with efficiently re-randomizable signatures, see also Appendix A.

Limitations of Known Meta-Reductions. Unfortunately, Coron's result has found only limited applications beyond digital signatures in the single-user setting. Most previous works [18,30,31] consider this setting, the (to our best knowledge) only exception is due to Lewko and Waters [33], which considers hierarchical identity-based encryption. Why isn't it possible to apply it to other primitives? One reason is that the bound in Eq. (1) ceases to be useful for reasonable values of ϵ_{Λ} and ϵ_{F} if $n \approx |\mathcal{M}|$. This can be easily seen by setting $n = |\mathcal{M}| - 1$. The assumption that $|\mathcal{M}| \gg n$ is a prerequisite for the arguments in [18,30,31] to work, thus, it is not possible to apply this technique to settings, where the assumption $|\mathcal{M}| \gg n$ is *not* reasonable.

Therefore Coron's technique is not applicable when $|\mathcal{M}|$ is polynomially-bounded. However, such a situation appears often when considering cryptographic primitives beyond digital signatures in the single-user setting. Consider, for instance, a security model where the adversary is provided with

[2] For a unique signature scheme there exists exactly one unique valid signature for each message. For instance, important instantiations of the famous Full-Domain Hash construction are unique signature schemes, see [31].

[3] Intuitively, a "simple" reduction is a reduction which has black-box access to the adversary, and runs the adversary only *sequentially*. Most reductions in cryptographic security proofs are of this type. A more precise definition is given in the body of the paper.

[4] In the sense of existential unforgeability under chosen-message attacks (EUF-CMA, cf. Definition 18).

$\mathcal{M} = \{pk_1, \ldots, pk_n\}$, where $pk_1, \ldots, pk_n$ is a list of public keys. The adversary may learn all but one of the corresponding secret keys, and is considered successful if it "breaks security" with respect to an uncorrupted key. This is a quite common setting, which occurs for instance in security models for signatures or public-key encryption in the multi-user setting with corruptions [3,4], all common security models for authenticated key exchange [4,9,15], and non-interactive key exchange [25] protocols. *How can we analyze the existence of inherent tightness bounds in these settings?*

Our Contributions. We develop a new meta-reduction technique, which is also applicable in settings where $|\mathcal{M}|$ is polynomially bounded. In comparison to [18,30,31], we achieve the simpler bound

$$\epsilon_{\mathcal{B}} \geq \epsilon_{\Lambda} - 1/n.$$

which is independent of $|\mathcal{M}|$.

Our new technique allows to rule out tight reductions from any non-interactive complexity assumption (cf. Definition 5). This includes also "decisional" assumptions (like decisional Diffie-Hellman). It avoids the combinatorial lemma of Coron [18, Lemma 1], which has a relatively technical proof. Our approach does not require such a combinatorial argument, but is more "direct".

This simplicity allows us to describe a generalized experiment with an abstract computable relation that captures the necessary properties for our tightness bounds. Then we explain that the standard security experiments for many cryptographic primitives are specific instances of this abstract experiment.

Technical Idea. To describe our technical idea, let us consider the example of digital signatures in the single-user settings, as considered in [18,30,31], for this introduction. As sketched above, the result will later be generalized and applied to other settings as well. We consider a weakened signature security definition, where the security experiment proceeds as follows.

1. The adversary receives as input a verification key vk along with n random but pairwise distinct messages $m_1, \ldots, m_n$.
2. The adversary selects an index j^*, and receives in response $n-1$ signatures σ_i for all messages m_i with $i \neq j^*$.
3. Finally, the adversary wins the experiment if it outputs σ^* that is a valid signature for m_{j^*} with respect to j^*.

Note that this is a very weak security definition, because the adversary is only able to observe signatures of random messages. However, note also that any lower tightness bound for such a weaker security definition implies a corresponding bound for any stronger definition. In particular, the above definition is weaker than the standard security definition *existential unforgeability under chosen message attacks* considered in [18,30,31], where messages may be adaptively chosen by the adversary.

Essentially, we argue that once a reduction has started the adversary in Step 1 of the above experiment, and thus has "committed" to a verification key

vk and messages $m_1, \ldots, m_n$, there can only be a single choice of j^* for which this reduction is able to output valid signatures σ_i for all $i \neq j^*$. Thus, for any adversary which chooses j^* uniformly at random the reduction has probability at most $1/n$ to succeed. We prove this by contradiction, by showing essentially that any reduction which is successful for two distinct choices of j^*, say j_0, j_1, can be used to construct a machine that breaks the underlying security assumption directly.

Technically, we proceed in two steps: first we describe an *inefficient* adversary against the reduction which chooses j^* uniformly random, and computes the signature σ^* for m_{j^*} by exhaustive search. Next, we show that this adversary can efficiently be simulated by our meta-reduction, if the reduction could succeed for two different choices j_0 and j_1 after committing to $(vk, m_1, \ldots, m_n)$. The meta-reduction simulates the inefficient adversary by rewinding the reduction. Essentially, if the reduction could succeed for two different values j_0, j_1, then it must also be able output the signatures for *all* n messages. Therefore we start the reduction and let it run until it reaches a "break point" where it outputs $(vk, m_1, \ldots, m_n)$. Next, we run the reduction n-times, each time starting from the break point and using a different index j, to search for two values j_0, j_1 such that $j_0 \neq j_1$ such that the reduction outputs valid signatures for all-but-one messages. If indeed there exist two such indices j_0, j_1, then we now have learned signatures for all messages $(m_1, \ldots, m_n)$ which are valid w.r.t. vk. Thus, we can run the reduction one last time from the break point, this time to the end, using index j_0 (or equivalently j_1), and we simulate the inefficient adversary using the fact that we know a valid signature for m_{j_0} (or m_{j_1}). Importantly, in the last execution of the reduction we are able to simulate the inefficient adversary perfectly, so the reduction will help us to break the non-interactive complexity assumption.

We caution that the rigorous proof of the above is more complex than the intuition provided in this introduction, and we have to put restrictions on the signature scheme, which depend on the considered application. For instance, when considering signatures in the single-user setting as above, we have to require that signatures are efficiently re-randomizable. In the generalized setting we will consider other applications, which require different but usually simple-to-check properties, like for instance that for each public key vk there exists a *unique* secret key. In this way, our result provides simple criteria to check whether a cryptographic construction can have a tight proof at all. At the same time it implicitly provides guidelines for the construction of tightly secure cryptographic schemes, since all tightly secure constructions must circumvent our result in one way or the other.

The fact that we consider a weakened security experiment has several nice features. We think that the approach and its analysis described above are much simpler than previous works, which enables more involved impossibility results. We will show that it achieves a simpler bound and yields a qualitatively stronger result, as it even rules out tight reductions for such weak security experiments. Like previous works, we only consider reductions that execute the adversary

sequentially and in a black-box fashion. We stress that most reductions in cryptography have this property.

We generalize the above idea from signature schemes in a single-user setting to abstract relations, which capture the relevant properties required for our impossibility argument to go through. We show that this abstraction allows to apply the result relatively easily to other cryptographic primitives, by describing applications to public-key encryption and signatures in the multi-user setting, and non-interactive key exchange.

Overview of Applications. A first, immediate application of our new technique are strengthened versions of the results of [18,30,31], but with significantly simpler proofs and tightness bounds even for weaker security notions (which is a stronger result). In contrast to previous works [18,30,31], the impossibility results hold also for "decisional" complexity assumptions.

Additionally, the fact that our meta-reduction does not require the combinatorial lemma of Coron enables further, novel applications in settings with polynomially-bounded spaces (where Coron's result worked only for exponential-sized spaces). As a first novel application of our generalized theorem, we analyze the tightness loss that occurs when security proofs in idealized single-user settings are transferred to the more realistic multi-user setting. Classical security models for standard cryptographic primitives often consider an idealized setting. For instance, the standard IND-CPA and IND-CCA security experiments for public-key encryption consider a setting with only one challenge public key and only a single challenge ciphertext. This is of course unrealistic for many practical applications. Public-key encryption is typically used in settings where an attacker sees many public keys and ciphertexts, and is (potentially) able to corrupt secret keys adaptively. Even though there is a reduction from breaking security in the multi-user setting to breaking security in the idealized setting, this reduction comes with a security loss which is linear in the number of users and ciphertexts. We show that under certain conditions (e.g., for schemes where there exists a *unique* secret key for each public key) this loss is impossible to avoid. This gives an insight into which properties a cryptosystem must or must not meet in order to allow a tight reduction in the multi-user setting.

Another novel application is the analysis of the existence of *non-interactive* key exchange (NIKE). In non-interactive key exchange (NIKE) two parties are able to derive a common shared secret. However, in contrast to traditional key exchange protocols, they do not need to exchange any messages. Besides the secret key of one party the key derivation algorithm only requires the availability of the public key of the communication partner. Security is defined solely by requiring indistinguishability of the derived shared secret from a random value. We show how to apply our main result to rule out tight reductions for a large class of NIKE protocols from a standard assumption in any sufficiently strong security model (such as the CKS-heavy model from [25]).

On Certified Public Keys and the Results of Kakvi and Kiltz. Several years after the publication of the paper of Coron [18] it has turned out that this paper

contains a subtle technical flaw. Essentially, it is implicitly assumed that the value output by the reduction to the adversary is a *correct* signature public key (recall that Coron considered only digital signature schemes in the single-user setting). This misses the fact that a reduction may possibly output *incorrect* keys which are computationally indistinguishable from correct ones. Indeed, such keys lead to the technical problem that a meta-reduction may *not* be able to simulate the adversary constructed in the meta-reduction of Coron correctly.

This flaw was identified and corrected by Kakvi and Kiltz [31]. Essentially, Kakvi and Kiltz enforce that the reduction outputs only public keys which can be efficiently recognized as correct, by introducing the notion of *certified* public keys. A different (but similar in spirit), slightly more general approach is due to Hofheinz *et al.* [30], who require that signatures are *efficiently re-randomizable* with respect to the public key output by from the reduction (regardless of whether this key is correct or not). Both these approaches [30,31] essentially overcome the subtle issue from Coron's paper by ensuring that the adversaries simulated by the meta-reductions are always able to output *correctly distributed* signatures.

In this paper, we introduce the notion of *efficiently re-randomizable relations* to overcome the subtle issue pointed out by Kakvi and Kiltz [31]. This notion further generalizes the approach of [30] in a way that suits our more general setting.

Relation to Tightly-Secure Constructions. There exist various constructions of tightly-secure cryptosystems, which have to avoid our impossibility results in one way or another. The signature schemes constructed in [1,10,19,29,32,36], for example, are tightly-secure in a single-user setting. They avoid our impossibility result because they do not have unique signatures or no efficient re-randomization algorithm is known. The same holds for the signature schemes derived from the IBE schemes of [11,17]. Bader *et al.* [4] constructed signature schemes with tight security even in the multi-user setting with adaptive secret-key corruptions. Again, our impossibility results are avoided here because signatures are not efficiently re-randomizable. The encryption schemes of Bellare, Boldyreva and Micali [6] are tightly-secure in a multi-user setting, but only without corruptions. We consider impossibility results for the multi-user setting with corruptions. The key encapsulation mechanism presented in [4] is tightly-secure even in a multi-user setting with corruptions. It avoids our impossibility result because it does not have unique secret keys.

More Related Work. Since their introduction by Boneh and Venkatesan in 1998 [12] meta-reductions have proven to be a versatile tool in many areas of provably security. Previous works have mainly used meta-reductions to derive impossibility results and efficiency/security bounds on signatures schemes [5,20–22,24,26,34,37], blind-signature schemes [23] and encryption systems [35]. In particular, among these results there exist several works that consider the existence of (tight) security proofs for the Schnorr signature scheme [5,24,26,34,37].

The results in [13,14] use meta-reductions to derive relationships among cryptographic one-more type problems. Lewko and Waters [33], building on [30], showed that under certain conditions it is impossible to prove security of hierarchical IBE (HIBE) schemes. To this end, Lewko and Waters extend the approach of [30] from signatures to hierarchical IBE to show that for certain HIBE schemes an *exponential* tightness loss is impossible to avoid. Finally, the inexistence of certain meta-reductions was considered in [22].

Outline. We begin with considering essentially the same setting as Coron and follow-up works [18,30,31], namely digital signatures in the single-user setting, as an instructive example. We prove a strengthened variant of the results of [18,30,31]. This allows us to explain how our new technique works in a known setting, which may be helpful for readers already familiar with these works. A generalized, much more abstract version will be presented in Sects. 4 and 5 gives many further interesting applications, which seem not achievable using the previous approach of [18,30,31].

2 The New Meta-reduction Technique

2.1 Preliminaries

Notation. We write $[n]$ to denote the set $[n] := \{1, 2, \ldots, n\}$, and for $j \in [n]$ we write $[n \backslash j]$ to denote the set $[n] \backslash \{j\}$. If A is a set then $a \xleftarrow{\$} A$ denotes the action of sampling a uniformly from A. Given a set A we denote by U_A the uniform distribution on A. If A is a Turing machine (TM) then $a \leftarrow A(x; r)$ denotes that A outputs a when run with input x and random coins r. By $A(x)$ we denote the distribution of $a \leftarrow A(x; r)$ over the uniform choice of r. If x is a binary string, then $|x|$ denotes its length. If M is a Turing machine, we denote by $\widehat{M}$ its description as a bitstring.

If $t : \mathbb{N} \to \mathbb{N}$ and there exists a constant c such that $t(k) \leq k^c$ for all but finitely many $k \in \mathbb{N}$, then we say that $t \in \mathsf{poly}(k)$. We denote by $\mathsf{poly}^{-1}(k)$ the set $\mathsf{poly}^{-1}(k) := \{\delta : \frac{1}{\delta} \in \mathsf{poly}(k)\}$. We say that $\epsilon : \mathbb{N} \to [0, 1]$ is *negligible* if for all $c \in \mathbb{N}$ it holds that $\epsilon(k) > k^{-c}$ is true only for at most finitely many $k \in \mathbb{N}$. We write $\epsilon \in \mathsf{negl}(k)$ to denote that ϵ is negligible.

Digital Signatures. A digital signature scheme $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ is a four-tuple of PPT-TMs:

Public Parameters. The public parameter generation machine $\Pi \xleftarrow{\$} \mathsf{Setup}(1^k)$ takes the security parameter k as input and returns public parameters Π.

Key Generation. The key generation machine takes as input public parameters Π and outputs a key pair, $(vk, sk) \xleftarrow{\$} \mathsf{Gen}(\Pi)$.

Signing. The signing machine takes as input a secret key sk and a message m and returns a signature $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)$.

Verification. The verification machine, on input a public key vk, a signature σ and a message m, outputs 0 or 1, $\mathsf{Vfy}(vk, m, \sigma) \in \{0, 1\}$.

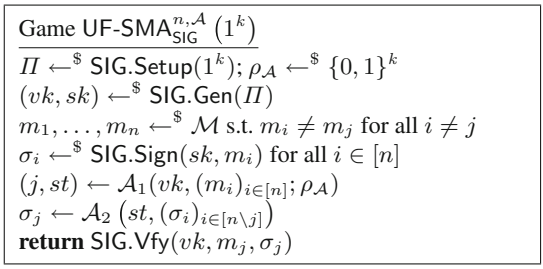

Fig. 1. The UF-SMA-security game with attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

Unique and Re-Randomizable Signatures. Let $\Sigma(vk, m) := \{\sigma : \mathsf{Vfy}(vk, m, \sigma) = 1\}$ denote the set of all valid signatures σ w.r.t. a given message m and verification key vk.

Definition 1 (Unique signatures). *We say that* SIG *is a* unique signature scheme*, if* $|\Sigma(vk, m)| = 1$ *for all* vk *and* m*.*

Definition 2 (Re-randomizable signatures). *We say that* SIG *is* $\mathsf{t}_{\mathsf{ReRand}}$*-re-randomizable, if there exists a TM* SIG.ReRand *which takes as input* (vk, m, σ) *and outputs a signature* $\sigma' \leftarrow^{\$} \mathsf{SIG.ReRand}(vk, m, \sigma)$ *with the following properties.*

1. SIG.ReRand *runs in time at most* $\mathsf{t}_{\mathsf{ReRand}}$
2. *If* $\mathsf{Vfy}(vk, m, \sigma) = 1$*, then* σ' *is distributed uniformly over* $\Sigma(vk, m)$*.*

Remark 1. Note that we do not put any bounds on $\mathsf{t}_{\mathsf{ReRand}}$. Thus, any signature scheme is $\mathsf{t}_{\mathsf{ReRand}}$-re-randomizable for sufficiently large $\mathsf{t}_{\mathsf{ReRand}}$. However, there are many examples of signature schemes which are *efficiently* re-randomizable, like the class of schemes considered in [30]. In particular, all unique signature schemes are efficiently re-randomizable by the Turing machine $\sigma \leftarrow^{\$} \mathsf{SIG.ReRand}(vk, m, \sigma)$ which simply outputs its input σ.

Unforgeability Under Static Message Attacks. The UF-SMA security experiment is depicted in Fig. 1.

Definition 3. *Let* $\mathsf{UF\text{-}SMA}^{n,\mathcal{A}}_{\mathsf{SIG}}(1^k)$ *denote the UF-SMA security experiment depicted in Fig. 1, executed with signature scheme* SIG *and attacker* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*. We say that* $\mathcal{A}$ $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$*-breaks the* UF-SMA*-security of* SIG*, if it runs in time* $\mathsf{t}_{\mathcal{A}}$ *and*

$$\Pr\left[\mathsf{UF\text{-}SMA}^{n,\mathcal{A}}_{\mathsf{SIG}}(1^k) \Rightarrow 1\right] \geq \epsilon_{\mathcal{A}}.$$

Remark 2. Observe that the messages in the UF-SMA security experiment from Fig. 1 are chosen at random (but pairwise distinct). We do this for simplicity, but stress that for our tightness bound we actually do not have to make any

assumption about the distribution of messages, apart from being pairwise distinct. For instance, the messages could alternatively be the lexicographically first n messages of the message space, for instance.

Non-interactive Complexity Assumptions. The following very general definition of non-interactive complexity assumptions is due to Abe et al. [2].

Definition 4. *A* non-interactive complexity assumption $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ *consists of three TMs. The instance generation machine* $(c, w) \leftarrow^{\$} \mathsf{T}(1^k)$ *takes the security parameter as input, and outputs a problem instance* c *and a witness* w*.* U *is a probabilistic polynomial-time machine, which takes as input* c *and outputs a candidate solution* s*. The verification TM* V *takes as input* (c, w) *and a candidate solution* s*. If* $\mathsf{V}(c, w, s) = 1$*, then we say that* s *is a correct solution to the challenge* c*.*

Intuitively, U is a probabilistic polynomial-time machine which implements a suitable "trivial" attack strategy for N. This algorithm is used to define what "breaking" N with non-trivial success probability means, cf. Definition 5 below and [2].

Consider the following experiment $\mathsf{NICA}_N^B(1^k)$.

1. The experiment runs the instance generator of N to generate a problem instance $(c, w) \leftarrow^{\$} \mathsf{T}(1^k)$. Then it samples uniformly random coins $\rho_B \leftarrow^{\$} \{0,1\}^k$ for B.
2. B is executed on input (c, ρ_B), it outputs a candidate solution s.
3. The experiment returns whatever $\mathsf{V}(c, w, s)$ returns.

Definition 5. *We say that* B (t, ϵ)*-breaks assumption* N*, if* Λ *runs in time* $t(k)$ *and it holds that*

$$\left|\Pr\left[\mathsf{NICA}_N^B\left(1^k\right) \Rightarrow 1\right] - \Pr\left[\mathsf{NICA}_N^{\mathsf{U}}\left(1^k\right) \Rightarrow 1\right]\right| \geq \epsilon(k)$$

where the probability is taken over the random coins consumed by T *and the uniformly random choices of* ρ_B *and* ρ_N *respectively.*

Simple Reductions From Non-interactive Complexity Assumptions to Breaking $\mathsf{UF\text{-}SMA}$*-Security.* A reduction from breaking the $\mathsf{UF\text{-}SMA}$-security of a signature scheme SIG to breaking the security of a non-interactive complexity assumption $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ is a TM, which turns an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ according to Definition 3 into a TM $\Lambda^{\mathcal{A}}$ according to Definition 5.

Following [18,30,31,33], we will consider a specific class of reductions in the sequel. We consider reductions having *black-box access* to the attacker, and which execute the attacker *only once* and *without rewinding*. We will generalize this later to reductions that may execute the attacker several times sequentially. Following [33], we call such reductions *simple*. At first sight we heavily constrain the class of reductions to that our result applies. However, as explained in [33], we include reductions that perform hybrid steps. Moreover, most reductions in cryptography are simple.

For preciseness and clarity, we define such a reduction as a triplet of Turing machines $\Lambda = (\Lambda_1, \Lambda_2, \Lambda_3)$. From these TMs and an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we construct a Turing machine $\Lambda^{\mathcal{A}}$ for a non-interactive complexity assumption as follows.

1. Machine $\Lambda^{\mathcal{A}}$ receives as input a challenge c of the considered non-interactive complexity assumption, as well as random coins $\rho_\Lambda \leftarrow^{\$} \{0,1\}^k$. It first runs $\Lambda_1(c, \rho_\Lambda)$, which returns the input to $\mathcal{A}_1$, consisting of a verification key vk, a sequence of messages $(m_i)_{i\in[n]}$, and random coins $\rho_{\mathcal{A}}$, as well as some state st_{Λ_2}.
2. Then $\Lambda^{\mathcal{A}}$ executes the attacker $\mathcal{A}_1$ on input $(vk, (m_i)_{i\in[n]}, \rho_{\mathcal{A}})$, which returns an index $j^* \in [n]$ and some state $st_{\mathcal{A}}$.
3. TM Λ_2 receives as input j^* and state st_{Λ_2}, and returns a list of signatures $(\sigma_i)_{i\in[n\setminus j^*]}$ and an updated state st_{Λ_3}.
4. The attacker $\mathcal{A}_2$ is executed on $(\sigma_i)_{i\in[n\setminus j^*]}$ and state $st_{\mathcal{A}}$, it returns a signature σ^*.
5. Finally, $\Lambda^{\mathcal{A}}$ runs $\Lambda_3(\sigma^*, j^*, st_{\Lambda_3})$, which produces a candidate solution s, and outputs s.

Definition 6. *We say that a Turing machine $\Lambda = (\Lambda_1, \Lambda_2, \Lambda_3)$ is a simple $(\mathsf{t}_\Lambda, n, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reduction from breaking $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ to breaking the* UF-SMA*-security of* SIG*, if for any TM $\mathcal{A}$ that $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$-breaks the* UF-SMA *security of* SIG*, TM $\Lambda^{\mathcal{A}}$ $(\mathsf{t}_\Lambda + \mathsf{t}_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N.*

Definition 7. *Let $\ell : \mathbb{N} \to \mathbb{N}$. We say that reduction Λ* loses *ℓ, if there exists an adversary $\mathcal{A}$ that $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$-breaks the* UF-SMA *security of* SIG*, such that $\Lambda^{\mathcal{A}}$ $(\mathsf{t}_\Lambda + \mathsf{t}_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N with*

$$\frac{\mathsf{t}_\Lambda(k) + \mathsf{t}_{\mathcal{A}}(k)}{\epsilon_\Lambda(k)} \geq \ell(k) \cdot \frac{\mathsf{t}_{\mathcal{A}}(k)}{\epsilon_{\mathcal{A}}(k)}.$$

Remark 3. The quotient $t_{\mathcal{A}}(k)/\epsilon_{\mathcal{A}}(k)$ of the running time $t_{\mathcal{A}}(k)$ and the success probability $\epsilon_{\mathcal{A}}(k)$ of a Turing machine $\mathcal{A}$ is called the *work factor* of $\mathcal{A}$ [8]. Thus, the factor ℓ in Definition 6 relates the work factor of attacker $\mathcal{A}$ to the work factor of TM $\Lambda^{\mathcal{A}}$, which allows us to measure the *tightness* of a cryptographic reduction. The smaller ℓ, the tighter is the reduction.

2.2 Bound for Simple Reductions Without Rewinding

For simplicity, we will consider reductions that have access to a "perfect" adversary $\mathcal{A}$, which $(\mathsf{t}_{\mathcal{A}}, \epsilon_{\mathcal{A}})$-breaks the signature scheme with $\epsilon_{\mathcal{A}} = 1$. We explain in Sect. 2.4 why the extension to adversaries with $\epsilon_{\mathcal{A}} < 1$ is straightforward.

Theorem 1. *Let $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ be a non-interactive complexity assumption, $n \in \mathsf{poly}(k)$ and let* SIG *be a signature scheme. For any simple $(\mathsf{t}_\Lambda, n, \epsilon_\Lambda, 1)$-reduction from breaking N to breaking the* UF-SMA*-security of* SIG*, there exists a Turing machine $\mathcal{B}$ that $(\mathsf{t}_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks N where*

$$\mathsf{t}_{\mathcal{B}} \leq n \cdot \mathsf{t}_\Lambda + n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + \mathsf{t}_{\mathsf{ReRand}} \qquad \textit{and} \qquad \epsilon_{\mathcal{B}} \geq \epsilon_\Lambda - 1/n.$$

Here, $\mathsf{t}_{\mathsf{ReRand}}$ *is the time required to re-randomize a signature, and* $\mathsf{t}_{\mathsf{Vfy}}$ *is the running time of the verification machine of* SIG.

Proof. Our proof structure follows the structure of [30] (also used in [33]). That is, we first describe a hypothetical, inefficient adversary, then we show how to simulate it efficiently for certain reductions.

The Hypothetical Adversary. The hypothetical adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consists of two procedures that work as follows.

$\mathcal{A}_1\left(vk, (m_i)_{i\in[n]}; \rho_{\mathcal{A}}\right)$. On input a public key vk and messages $m_1, \ldots, m_n$, $\mathcal{A}_1$ samples $j \leftarrow^{\$} [n]$ uniformly random and outputs (j, st), where $st = (vk, (m_i)_{i\in[n]}, j)$.

$\mathcal{A}_2((\sigma_i)_{i\in[n\backslash j]}, st)$. $\mathcal{A}_2$ checks whether $\mathsf{SIG.Vfy}(vk, m_i, \sigma_i) = 1$ for all $i \in [n\backslash j]$. If this holds, then it samples a uniformly random signature $\sigma_j \leftarrow^{\$} \Sigma(vk, m_j)$ for m_j. Finally, it outputs σ_j.

Note that $\mathcal{A}$ $(t_{\mathcal{A}}, 1)$-breaks the UF-SMA-security of SIG. Note also that the second step of this adversary may not be efficiently computable, which is why we call this adversary *hypothetical*.

Simulating $\mathcal{A}$. Consider the following TM $\mathcal{B}$, which runs reduction $\Lambda = (\Lambda_1, \Lambda_2, \Lambda_3)$ as a subroutine and attempts to break N. $\mathcal{B}$ receives as input $c \leftarrow^{\$} \mathsf{T}(1^k)$. It maintains an array A with n entries, which are all initialized to $\emptyset$, and proceeds as follows.

1. $\mathcal{B}$ first runs $(vk, (m_i)_{i\in[n]}, \rho_{\mathcal{A}}, st_{\Lambda_2}) \leftarrow^{\$} \Lambda_1(c; \rho_\Lambda)$ for uniformly random $\rho_\Lambda \leftarrow^{\$} \{0,1\}^k$.
2. Next, $\mathcal{B}$ runs $\Lambda_2(j, st_{\Lambda_2})$ for each $j \in [n]$. Let $((\sigma_{i,j})_{i\in[n\backslash j]}, st_{\Lambda_3,j})$ denote the output of the j-th execution of Λ_2. Whenever Λ_2 outputs $(\sigma_{i,j})_{i\in[n\backslash j]}$ such that
$$\mathsf{SIG.Vfy}(vk, m_i, \sigma_{i,j}) = 1 \text{ for all } i \in [n\backslash j]$$
then it sets $A[i] \leftarrow \sigma_{i,j}$ for all $i \in [n\backslash j]$.
3. $\mathcal{B}$ samples $j^* \leftarrow^{\$} [n]$. Then it proceeds as follows.
   - If there exists an index $i \in [n\backslash j^*]$ such that $\mathsf{SIG.Vfy}(vk, m_i, \sigma_{i,j^*}) \neq 1$, then $\mathcal{B}$ sets $\sigma^* := \bot$.
   - Otherwise, if $\mathsf{SIG.Vfy}(vk, m_i, \sigma_{i,j^*}) = 1$ for all $i \in [n\backslash j^*]$, then $\mathcal{B}$ computes
$$\sigma^* \leftarrow^{\$} \mathsf{SIG.ReRand}(vk, m_{j^*}, A[j^*]).$$
4. Finally, $\mathcal{B}$ runs $s \leftarrow \Lambda_3(\sigma^*, j^*, st_{\Lambda_3,j^*})$ and outputs s. Note that the state st_{Λ_3,j^*} used to execute Λ_3 corresponds to the state returned by Λ_2 on its j^*-th execution.

Running Time of $\mathcal{B}$. $\mathcal{B}$ essentially runs each part of Turing machine $\Lambda = (\Lambda_1, \Lambda_2, \Lambda_3)$ once, plus $n-1$ additional executions of Λ_2. Moreover, it executes

$\mathsf{SIG.Vfy}$ $n(n-1)$ times, and the re-randomization TM $\mathsf{SIG.ReRand}$ once. Thus, the total running time of $\mathcal{B}$ is at most

$$\mathsf{t}_{\mathcal{B}} \leq n \cdot \mathsf{t}_{\Lambda} + n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + \mathsf{t}_{\mathsf{ReRand}}.$$

Success Probability of $\mathcal{B}$. To analyze the success probability of $\mathcal{B}$, let us define an event bad. Intuitively, this event occurs, if j^* is the *only* (with respect to state st_{Λ_2}) value such that $\Lambda_2(st_{\Lambda_2}, j)$ outputs signatures which are all valid. More formally, for both experiments $\mathsf{NICA}_N^{\mathcal{B}}(1^k)$ and $\mathsf{NICA}_N^{\Lambda^{\mathcal{A}}}(1^k)$, let st_{Λ_2} denote the (in both experiments unique) value computed by $\Lambda_1(c;\rho_{\Lambda})$, and let j^* denote the (in both experiments unique) value given as input to $\Lambda_3(\sigma^*, j^*, st_{\Lambda_3,j^*})$. We say that bad occurs (in either $\mathsf{NICA}_N^{\mathcal{B}}(1^k)$ or $\mathsf{NICA}_N^{\Lambda^{\mathcal{A}}}(1^k)$), if $\mathsf{pred}(st_{\Lambda_2}, j^*) = 1 \wedge \mathsf{pred}(st_{\Lambda_2}, j) = 0 \ \forall\ j \in [n \backslash j^*]$, where predicate pred is defined as

$$\begin{aligned} &\mathsf{pred}(st_{\Lambda_2}, j) = 1 \\ \iff &\bigwedge_{i \in [n \backslash j]} \mathsf{SIG.Vfy}(vk, m_i, \sigma_i) = 1, \text{ where } ((\sigma_i)_{i \in [n \backslash j]}, st_{\Lambda_3}) \leftarrow \Lambda_2(st_{\Lambda_2}, j). \end{aligned}$$

Note that pred is well-defined, because Λ_2 is a deterministic TM.

Let us write $\mathsf{S}(\mathcal{F})$ shorthand for the event $\mathsf{NICA}_N^{\mathcal{F}}(1^k) \Rightarrow 1$ to abbreviate our notation. Then, it holds that

$$\left|\Pr[\mathsf{S}(\mathcal{B})] - \Pr[\mathsf{S}(\Lambda^{\mathcal{A}})]\right| \leq \left|\Pr[\mathsf{S}(\mathcal{B}) \cap \neg\mathsf{bad}] - \Pr[\mathsf{S}(\Lambda^{\mathcal{A}}) \cap \neg\mathsf{bad}]\right| + \Pr[\mathsf{bad}]. \tag{2}$$

Bounding $\Pr[\mathsf{bad}]$. Recall that event bad occurs only if

$$\mathsf{pred}(st_{\Lambda_2}, j^*) = 1 \wedge \mathsf{pred}(st_{\Lambda_2}, j) = 0 \ \forall\ j \in [n \backslash j^*] \tag{3}$$

where st_{Λ_2} is the value computed by $\Lambda_1(c;\rho_{\Lambda})$, and j^* is the value given as input to $\Lambda_3(\sigma^*, j^*, st_{\Lambda_3,j^*})$. Suppose that indeed st_{Λ_2} is such that there exist at least one $j^* \in [n]$ such that (3) holds. We claim that even then we have

$$\Pr[\mathsf{bad}] \leq 1/n. \tag{4}$$

To see this, note first that for each st_{Λ_2} there can be at most one value j^* that satisfies (3). Moreover, both the hypothetical adversary $\mathcal{A}$ and the adversary simulated by $\mathcal{B}$ choose $j^* \leftarrow^{\$} [n]$ independently and uniformly random, which yields (4).

Proving $\Pr[\mathsf{S}(\mathcal{B}) \cap \neg\mathsf{bad}] = \Pr[\mathsf{S}(\Lambda^{\mathcal{A}}) \cap \neg\mathsf{bad}]$. Note that $\mathcal{B}$ executes in particular

1. $(vk, (m_i)_{i \in [n]}, st_{\Lambda_2}) \leftarrow^{\$} \Lambda_1(c;\rho_{\Lambda})$
2. $((\sigma_{i,j^*})_{i \in [n \backslash j^*]}, st_{\Lambda_3}) \leftarrow^{\$} \Lambda_2(j^*, st_{\Lambda_2})$
3. $s \leftarrow \Lambda_3(\sigma^*, j^*, st_{\Lambda_3})$.

We show that if $\neg\mathsf{bad}$ occurs, then $\mathcal{B}$ simulates the hypothetical adversary $\mathcal{A}$ perfectly. To this end, consider the distribution of σ^* computed by $\mathcal{B}$ in following two cases.

1. Machine $\Lambda_2(j^*, st_{\Lambda_2})$ outputs $((\sigma_{i,j^*})_{i\in[n\setminus j^*]}, st_{\Lambda_3,j^*})$ such that there exists an index $i \in [n\setminus j^*]$ with $\mathsf{SIG.Vfy}(vk, m_i, \sigma_{i,j^*}) \neq 1$.
   In this case, $\mathcal{A}$ would compute $\sigma^* := \bot$. $\mathcal{B}$ also sets $\sigma^* := \bot$ in this case.
2. TM $\Lambda_2(j^*, st_{\Lambda_2})$ outputs $((\sigma_{i,j^*})_{i\in[n\setminus j^*]}, st_{\Lambda_3,j^*})$ such that for all $i \in [n\setminus j^*]$ it holds that
$$\mathsf{SIG.Vfy}(vk, m_i, \sigma_{i,j^*}) = 1.$$
   In this case, $\mathcal{A}$ would output a uniformly random signature $\sigma^* \leftarrow^{\$} \Sigma(vk, m_{j^*})$. Note that in this case $\mathcal{B}$ outputs a re-randomized signature $\sigma^* \leftarrow^{\$} \mathsf{SIG.ReRand}(vk\ , m_{j^*}, A[j^*])$, which is a uniformly distributed valid signature for m_{j^*} provided that $A[j^*] \neq \emptyset$. The latter happens whenever bad does not occur.

Thus, $\mathcal{B}$ simulates $\mathcal{A}$ perfectly in either case, provided that $\neg\mathsf{bad}$. This implies $\mathsf{S}(\mathcal{B}) \cap \neg\mathsf{bad} \iff \mathsf{S}(\Lambda^{\mathcal{A}}) \cap \neg\mathsf{bad}$, which yields

$$\Pr[\mathsf{S}(\mathcal{B}) \cap \neg\mathsf{bad}] = \Pr[\mathsf{S}(\Lambda^{\mathcal{A}}) \cap \neg\mathsf{bad}]. \tag{5}$$

Finishing the Proof of Theorem 1. By plugging (4) and (5) into Inequality (2), we obtain

$$\left|\Pr[\mathsf{S}(\mathcal{B})] - \Pr[\mathsf{S}(\Lambda^{\mathcal{A}})]\right| \leq 1/n$$

which implies

$$\epsilon_{\mathcal{B}} = |\Pr[\mathsf{S}(\mathcal{B})] - \Pr[\mathsf{S}(\mathsf{U})]| \geq |\Pr[\mathsf{S}(\Lambda)] - \Pr[\mathsf{S}(\mathsf{U})]| - 1/n = \epsilon_{\Lambda} - 1/n.$$

2.3 Interpretation

Assuming that no adversary $\mathcal{B}$ is able to $(\mathsf{t_N}, \epsilon_{\mathsf{N}})$-break the security of NICA with $\mathsf{t_N} = \mathsf{t}_{\mathcal{B}} = n \cdot \mathsf{t}_{\Lambda} + n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + \mathsf{t}_{\mathsf{ReRand}}$, we must have $\epsilon_{\mathcal{B}} \leq \epsilon_{\mathsf{N}}$. By Theorem 1, we thus must have

$$\epsilon_{\Lambda} \leq \epsilon_{\mathcal{B}} + 1/n \leq \epsilon_{\mathsf{N}} + 1/n$$

for all reductions Λ. In particular, the hypothetical adversary $\mathcal{A}$ constructed in the proof of Theorem 1 is an example of an adversary such that

$$\frac{\mathsf{t}_{\Lambda} + \mathsf{t}_{\mathcal{A}}}{\epsilon_{\Lambda}} \geq \frac{\mathsf{t}_{\mathcal{A}}}{\epsilon_{\mathsf{N}} + 1/n} = (\epsilon_{\mathsf{N}} + 1/n)^{-1} \cdot \frac{\mathsf{t}_{\mathcal{A}}}{1} = (\epsilon_{\mathsf{N}} + 1/n)^{-1} \cdot \frac{\mathsf{t}_{\mathcal{A}}}{\epsilon_{\mathcal{A}}}.$$

Thus, any reduction Λ from breaking the security of NICA N to breaking the $\mathsf{UF\text{-}SMA}$-security of signature scheme SIG loses (in the sense of Definition 7) at least a factor of $\ell \geq 1/(\epsilon_{\mathsf{N}} + 1/n)$. In particular, note that $\ell \approx n$ if ϵ_{N} is very small. This yields the following informal theorem.

Theorem 2 (Informal). *Any simple reduction from breaking the security of* NICA N *to breaking the* $\mathsf{UF\text{-}SMA}$*-security (or any stronger security notion, like* $\mathsf{EUF\text{-}CMA}$*-security, cf. Definition 19) of signature scheme* SIG *that provides* efficient signature re-randomization *loses a factor that is at least linear in the number* n *of sign queries issued by the attacker, or* N *is easy to solve.*

Remark 4. Since a unique signature scheme is trivially efficiently re-randomizable, Theorem 2 applies also to unique signature schemes.

TM $r\text{-}\Lambda^{\mathcal{A}}(c;\rho_\Lambda)$
$st_{\Lambda_{1,1}} \leftarrow \Lambda_0\,(c,\rho_\Lambda)$
for $1 \le l \le r$ **do**:
  $(vk^l,(m_i^l)_{i\in[n]},\rho_{\mathcal{A}},st_{\Lambda_{l,2}}) \leftarrow \Lambda_{l,1}(st_{\Lambda_{l,1}})$
  $(j^{l*},st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(vk^l,(m_i^l)_{i\in[n]};\rho_{\mathcal{A}})$
  $((\sigma_i^l)_{i\in[n\setminus j^{l*}]},st_{\Lambda_{l,3}}) \leftarrow \Lambda_{l,2}(j^{l*},st_{\Lambda_{l,2}})$
  $\sigma^l_{j^{l*}} \leftarrow \mathcal{A}_2((\sigma_i^l)_{i\in[n\setminus j^{l*}]},st_{\mathcal{A}})$
  $st_{\Lambda_{l+1,1}} \leftarrow \Lambda_{l,3}\left(\sigma^l_{j^{l*}},j^{l*},st_{\Lambda_{l,3}}\right)$
$s \leftarrow \Lambda_3\left(st_{\Lambda_{r+1,1}}\right)$
return s

Fig. 2. TM $r\text{-}\Lambda^{\mathcal{A}}$ that solves a non-interactive complexity assumption according to Definition 5, constructed from a r-simple reduction $r\text{-}\Lambda = \left(\Lambda_0,(\Lambda_{l,1},\Lambda_{l,2},\Lambda_{l,3})_{l\in[r]},\Lambda_3\right)$ and an attacker $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$.

2.4 Extension to "Non-perfect" Adversaries

Note that the proof of Theorem 1 trivially generalizes to $(\mathsf{t}_\Lambda, n, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reductions with $\epsilon_{\mathcal{A}} < 1$, that is, reductions that have access to an adversary which has success probability $\epsilon_{\mathcal{A}} < 1$. To this end, we first would have to describe a hypothetical adversary, which has success probability $\epsilon_{\mathcal{A}}$. This is simple, because we can simply let the hypothetical adversary constructed above toss a biased coin χ with $\Pr[\chi = 1] = \epsilon_{\mathcal{A}}$, such that $\mathcal{A}$ outputs σ^* only if $\chi = 1$. Note that in the proof of Theorem 1 we are even able to simulate a perfect adversary $\mathcal{A}$. Therefore we would also be able to simulate the non-perfect adversary sketched above, by tossing a biased coin χ and outputting σ^* only if $\chi = 1$. This yields the following theorem.

Theorem 3. *Let $N = (\mathsf{T},\mathsf{V},\mathsf{U})$ be a non-interactive complexity assumption, $n \in \mathsf{poly}(k)$ and let SIG be a signature scheme. For any simple $(\mathsf{t}_\Lambda, n, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reduction from breaking the $\mathsf{UF\text{-}SMA}$-security of SIG to breaking N, there exists a Turing machine $\mathcal{B}$ that $(\mathsf{t}_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks N where*

$$\mathsf{t}_{\mathcal{B}} \le n \cdot \mathsf{t}_\Lambda + n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + \mathsf{t}_{\mathsf{ReRand}} \qquad \text{and} \qquad \epsilon_{\mathcal{B}} \ge \epsilon_\Lambda - 1/n.$$

Here, $\mathsf{t}_{\mathsf{ReRand}}$ is the time to re-randomize a given valid signature over a message and $\mathsf{t}_{\mathsf{Vfy}}$ is the time needed to execute the verification machine of SIG.

3 Bound for Reductions with Sequential Rewinding

Theorem 1 applies only to reductions that run the forger only once. Here we show that under assumptions similar to that in Theorem 1 the work factor of any reduction that is allowed to run or rewind the adversary r times *sequentially* cannot decrease significantly below $\frac{n}{r}$ if N is hard.

Let r be an upper bound on the number of times that the adversary can be rewound by the reduction. We then consider a reduction r-Λ as a $3 \cdot r + 2$-tuple of Turing machines $r\text{-}\Lambda = \left(\Lambda_0, (\Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3})_{l\in[r]}, \Lambda_3\right)$. Let now $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an attacker against the UF-SMA-security of SIG. From these TMs we construct a Turing machine $r\text{-}\Lambda^{\mathcal{A}}$ that solves a NICA N as depicted in Fig. 2. We shortly explain Fig. 2 here.

Λ_0. r-Λ inputs a challenge c of the considered non-interactive complexity assumption and random coins ρ_Λ. It processes these inputs by running Λ_0 which outputs a state st_Λ.

$\Lambda_l = (\Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3})$. Now, for each $l \in [r]$, we have a triplet of TMs $\Lambda_l = (\Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3})$ that has black box access to attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. Note that the state st_Λ may be passed over from $\Lambda_{l,3}$ to $\Lambda_{l+1,1}$ (and Λ_3) while the state $st_{\mathcal{A}}$ of $\mathcal{A}_2$ may not be passed over to the next execution of $\mathcal{A}_1$.

- **$\Lambda_{l,1}$.** $\Lambda_{l,1}$ inputs the current state $st_{\Lambda_{l,1}}$ and outputs a public key vk^l, distinct messages $m_i^l, i \in [n]$, a random tape $\rho_{\mathcal{A}}$ for $\mathcal{A}_1$ and a state $st_{\Lambda_{l,2}}$. Next, $\mathcal{A}_1$ is run on input $\left(vk^l, (m_i)_{i\in[n]}\right); \rho_{\mathcal{A}})$ and returns a state $st_{\mathcal{A}}$ and an index j^l.
- **$\Lambda_{l,2}$.** On input index j^l and state $st_{\Lambda_{l,2}}$, $\Lambda_{l,2}$ returns signatures $\left(\sigma_i^l\right)_{i\in[n\setminus j]}$ and state $st_{\Lambda_{l,2}}$. Now, $\mathcal{A}_2$ is run on $\left(\left(\sigma_i^l\right)_{i\in[n\setminus j^l]}, st_{\mathcal{A}}\right)$ and returns $\sigma_{j^l}^l$.
- **$\Lambda_{l,3}$.** $\Lambda_{l,3}$ inputs the signature output by $\mathcal{A}_{l,2}$ and the current state $st_{\Lambda_{l,2}}$. It returns the state $st_{\Lambda_{l+1,1}}$.

Λ_3. Finally, Λ_3 inputs the current state of r-Λ and returns s. r-Λ is considered successful if $\mathsf{V}(c, w, s) = 1$.

Definition 8. *We say that a Turing machine $r\text{-}\Lambda = \left(\Lambda_0, (\Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3})_{l\in[r]}, \Lambda_3\right)$ is an r-simple $(t_\Lambda, n, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reduction from breaking $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ to breaking the UF-SMA-security of SIG, if for any TM $\mathcal{A}$ that $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$-breaks the UF-SMA security of SIG, TM $r\text{-}\Lambda^{\mathcal{A}}$ (as constructed above) $(t_\Lambda + r \cdot t_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N.*

Definition 9. *Let $\ell : \mathbb{N} \to \mathbb{N}$. We say that an r-simple reduction Λ from breaking a non-interactive complexity assumption N to breaking the UF-SMA security of a signature scheme SIG loses ℓ if there exists an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$-breaks such that $\Lambda^{\mathcal{A}}$ $(t_\Lambda + r \cdot t_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N where*

$$\frac{t_\Lambda(k) + r \cdot t_{\mathcal{A}}(k)}{\epsilon_\Lambda} \geq \ell(k) \cdot \frac{t_{\mathcal{A}}(k)}{\epsilon_{\mathcal{A}}(k)}.$$

Theorem 4. *Let $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ be a non-interactive complexity assumption, $n, r \in \mathsf{poly}(k)$ and let SIG be a signature scheme. Then for any r-simple $(t_\Lambda, n, \epsilon_\Lambda, 1)$-reduction Λ from breaking N to breaking the UF-SMA-security of SIG there exists a TM $\mathcal{B}$ that $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks N where*

$$\begin{aligned} t_{\mathcal{B}} &\leq r \cdot n \cdot \mathsf{t}_\Lambda + r \cdot n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + r \cdot \mathsf{t}_{\mathsf{ReRand}} \\ \epsilon_{\mathcal{B}} &\geq \epsilon_\Lambda - \frac{r}{n}. \end{aligned}$$

Here, $\mathsf{t}_{\mathsf{ReRand}}$ *is the time to re-randomize a given valid signature over a message and* $\mathsf{t}_{\mathsf{Vfy}}$ *is the time needed to run the verification machine of* SIG.

The proof of this theorem is structured as the proof of Theorem 1. We again first consider a hypothetical attacker $\mathcal{A}$ (cf. Page 11) that breaks the UF-SMA-security of SIG. Next, when we show how to simulate $\mathcal{A}$, we basically apply the technique from the proof of Theorem 1 r times. A detailed proof can be found in the full version of this paper.

3.1 Interpretation

Assuming that no adversary $\mathcal{B}$ is able to $(\mathsf{t}_{\mathsf{N}}, \epsilon_{\mathsf{N}})$-break the security of NICA with $\mathsf{t}_{\mathsf{N}} = \mathsf{t}_{\mathcal{B}} = r \cdot n \cdot \mathsf{t}_{\Lambda} + r \cdot n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + r \cdot \mathsf{t}_{\mathsf{ReRand}}$, we must have $\epsilon_{\mathcal{B}} \leq \epsilon_{\mathsf{N}}$. By Theorem 4, we thus must have

$$\epsilon_{\Lambda} \leq \epsilon_{\mathcal{B}} + r/n \leq \epsilon_{\mathsf{N}} + r/n$$

for all reductions Λ. In particular, the hypothetical adversary $\mathcal{A}$ constructed in the proof of Theorem 1 is an example of an adversary such that

$$\frac{\mathsf{t}_{\Lambda} + r \cdot \mathsf{t}_{\mathcal{A}}}{\epsilon_{\Lambda}} \geq \frac{r \cdot \mathsf{t}_{\mathcal{A}}}{\epsilon_{\mathsf{N}} + r/n} = (\epsilon_{\mathsf{N}} + r/n)^{-1} \cdot r \cdot \frac{\mathsf{t}_{\mathcal{A}}}{1} = (\epsilon_{\mathsf{N}} + r/n)^{-1} \cdot r \cdot \frac{\mathsf{t}_{\mathcal{A}}}{\epsilon_{\mathcal{A}}}.$$

Thus, any reduction Λ from breaking the security of NICA N to breaking the UF-SMA-security of signature scheme SIG loses (in the sense of Definition 7) at least a factor of $\ell \geq r/(\epsilon_{\mathsf{N}} + r/n)$. In particular, note that $\ell \approx n$ if ϵ_{N} is very small.

4 A Generalized Meta-reduction

In this section we state and prove our main result, which generalizes the results from Sect. 2. Essentially, we observe that for the proof to work we do not need all structural elements a signature scheme possesses. In particular we do not require dedicated parameter generation-, key generation- and sign-algorithms. Instead, we consider an abstract security experiment with the following properties:

1. The values that are publicly available "induce a relation" $R(x, y)$ that is efficiently verifiable for the adversary during the security experiment.
2. The adversary is provided with statements $y_1, \ldots, y_n$ at the beginning of the security experiment and has access to an oracle that when queried y_i returns x_i such that $R(x_i, y_i), i \in [n]$.
3. If the adversary is able to output x_j such that $R(x_j, y_j)$ and it did not query its oracle on y_j, this is sufficient to win the security game.

Remark 5. To show the usefulness of such an abstract experiment, we note that for instance the security experiments for public key encryption or key encapsulation mechanisms in the multi-user setting with corruptions [4], or digital

signature schemes in the multi-user (MU) setting with corruptions [3,4], naturally satisfy these properties as follows. Essentially, we define a relation $R(sk, pk)$ over pairs of public keys and secret keys such that $R(sk, pk) = 1$ whenever sk "matches" pk. The adversary is provided with public keys at the beginning of the experiment, and is able to obtain secret keys corresponding to public keys of its choice. Finally, if the adversary is able to output an uncorrupted secret key, it is clearly able to compute a signature over a message that was not signed before (i.e., winning the signature security game) or decrypt the challenge ciphertext (i.e., winning the PKE/KEM security game). Thus, all three requirements are satisfied. Details on how to apply the result to, e.g., digital signatures and PKE/KEMs in the multi user setting with corruptions we refer to Sect. 5.

4.1 Definitions

Re-randomizable Relations. Let $R \subseteq X \times Y$ be a relation. For (x, y) with $R(x, y) = 1$ we call x the *witness* and y the *statement*. We use $X(R, y)$ to denote the set

$$X(R, y) := \{x : R(x, y) = 1\}$$

of all witnesses x for statement y with respect to R. We denote by $L(R) := \{y : \exists\, x \,\text{s.t.}\, R(x, y) = 1\} \subseteq Y$ the language consisting of statements in R.

In the sequel we will consider *computable* relations. We will therefore identify a relation R with a machine $\widehat{R}$ that computes R. We say that a relation R is t_{Vfy}-computable, if there is a deterministic Turing machine $\widehat{R}$ that runs in time at most $t_{\mathsf{Vfy}}(|x| + |y|)$ such that $\widehat{R}(x, y) = R(x, y)$.

Definition 10. *Let $\mathcal{R} := \{R_i\}_{i \in I}$ be a family of computable relations. We say that $\mathcal{R}$ is $\mathsf{t}_{\mathsf{ReRand}}$-re-randomizable if there is a probabilistic Turing machine $\mathcal{R}.\mathsf{ReRand}$ that inputs $(\widehat{R}_i, y, x)$, runs in time at most $\mathsf{t}_{\mathsf{ReRand}}$, and outputs x' which is uniformly distributed over $X(R, y_i)$ whenever $R_i(x, y) = 1$, with probability 1.*

Example 1. Digital signatures in the single user setting, as considered in Sect. 2, may be described in terms of families of relations. We set $R_{\Pi,vk}$ to the relation over signatures and messages that is defined by a verification key vk. In this case, we have that $X(R, y) = \Sigma(vk, y)$ is the set of all valid signatures over message y with respect to public key vk. Note that the family of relations $(R_{\Pi,vk})_{\Pi,vk}$ is $\mathsf{t}_{\mathsf{ReRand}}$-re-randomizable, if the signature scheme is $\mathsf{t}_{\mathsf{ReRand}}$-re-randomizable (cf. Definition 2).

Witness Unforgeability Under Static Statement Attacks. We will consider a weak security experiment for computable relations, which is inspired by the UF-SMA-security experiment considered in Sect. 2, but abstract and general enough to be applicable in other useful settings. Jumping slightly ahead, we will show in Sect. 5 that this includes applications to signatures, public-key encryption, key encapsulation mechanisms in the multi-user setting, and non-interactive key exchange.

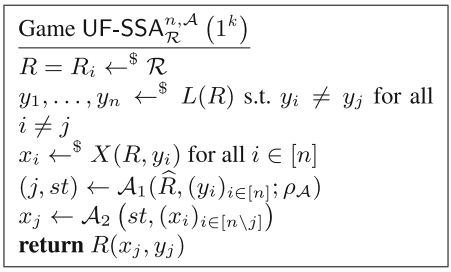

Fig. 3. The UF-SSA-security game with attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

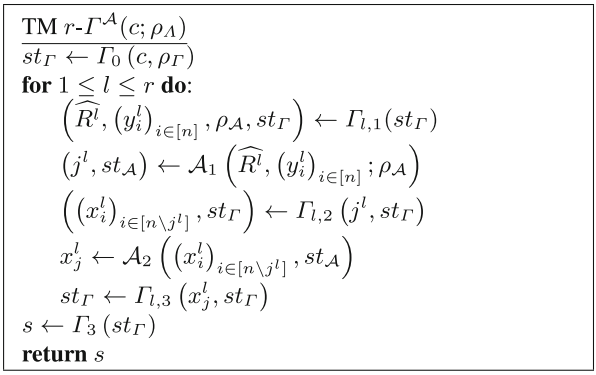

Fig. 4. TM $r\text{-}\Gamma^{\mathcal{A}}$ that solves a non-interactive complexity assumption according to Definition 5, constructed from a r-simple reduction $r\text{-}\Gamma = \left(\Gamma_0, (\Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3})_{l\in[r]}, \Gamma_3\right)$ and an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

The security experiment is described in Fig. 3. It is parametrized by a family $\mathcal{R}$ of computable relations, $\mathcal{R} = \{R_i\}_{i\in I}$, and the number n of statements the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is provided with. These statements need to be pairwise distinct. $\mathcal{A}$ may *non-adaptively* ask for witnesses for *all but one* statement, and is considered successful if it manages to output a "valid" witness for the remaining statement.

Definition 11. *Let $\mathcal{R} = \{R_i\}_{i\in I}$ be a family of computable relations. We say that an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ (t, n, ϵ)-breaks the* witness unforgeability under static statement attacks *of $\mathcal{R}$ if it runs in time t and*

$$\Pr\left[\mathsf{UF\text{-}SSA}^n_{\mathcal{R}}(\mathcal{A}) \Rightarrow 1\right] \geq \epsilon$$

where $\mathsf{UF\text{-}SSA}^n_{\mathcal{R}}(\mathcal{A})$ is the security game depicted in Fig. 3.

Simple Reductions From Non-interactive Complexity Assumptions to Breaking UF-SSA*-Security.* Informally, a reduction from breaking the UF-SSA-security of

a family of relations $\mathcal{R}$ to breaking the security of a non-interactive complexity assumption $N = (\mathsf{T}, \mathsf{U}, \mathsf{V})$ is a Turing machine Γ, which turns an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $\mathcal{R}$ according to Definition 11 into a TM $\Gamma^{\mathcal{A}}$ that breaks N according to Definition 5. As in Sect. 2, we will only consider simple reductions, i.e., reductions that have *black-box* access to the attacker and that may run the attacker at most r times *sequentially*.

We define a reduction from breaking the security of $\mathcal{R}$ to breaking N as an $(3r+2)$-tuple of TMs $\Gamma = \left(\Gamma_0, (\Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3})_{l\in[r]}, \Gamma_3\right)$, which turn a TM $\mathcal{A}$ breaking the security of $\mathcal{R}$ into a TM $\Gamma^{\mathcal{A}}$ breaking N, as described in Fig. 4. Note that this Turing machine works almost identical to that considered in Sect. 3, except that we consider a more general class of relations.

Definition 12. *We say that a TM* r-$\Gamma = \left(\Gamma_0, (\Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3})_{l\in[r]}, \Gamma_3\right)$ *is an* r*-simple* $(t_\Gamma, n, \epsilon_\Gamma, \epsilon_{\mathcal{A}})$*-reduction from breaking* $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ *to breaking the* UF-SSA*-security of a family of relations* $\mathcal{R}$*, if for any TM* $\mathcal{A}$ *that* $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$*-breaks the* UF-SSA *security of* $\mathcal{R}$*, TM* r-$\Gamma^{\mathcal{A}}$ *(cf. Fig. 4)* $(t_\Lambda + r \cdot t_{\mathcal{A}}, \epsilon_\Lambda)$*-breaks* N*.*

We define the loss of an r-simple reduction r-Γ from breaking N to breaking the UF-SSA-security of a family of computable relations $\mathcal{R}$ similar to Definition 9.

4.2 Main Result

In this Section we establish the following result that generalizes Theorem 4.

Theorem 5. *Let* $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ *be a non-interactive complexity assumption,* $n, r \in \mathsf{poly}(k)$ *and let* $\mathcal{R}$ *be a family of computable relations. Then for any* r*-simple* $(t_\Gamma, n, \epsilon_\Gamma, 1)$*-reduction* Γ *from breaking* N *to breaking the* UF-SSA*-security of* $\mathcal{R}$ *there exists a TM* $\mathcal{B}$ *that* $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$*-breaks* N *where*

$$t_{\mathcal{B}} \leq r \cdot n \cdot t_\Gamma + r \cdot n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + r \cdot \mathsf{t}_{\mathsf{ReRand}}$$
$$\epsilon_{\mathcal{B}} \geq \epsilon_\Gamma - \frac{r}{n}.$$

Here, $\mathsf{t}_{\mathsf{ReRand}}$ *is the time to re-randomize a given valid witness and* $\mathsf{t}_{\mathsf{Vfy}}$ *is the maximum time needed to compute* $R \in \mathcal{R}$*.*

The proof of Theorem 5 is nearly identical to the proof of Theorem 4, and therefore omitted. Also the interpretation of Theorem 5 is nearly identical to the interpretation described in Sect. 2.3. Assuming that no adversary $\mathcal{B}$ is able to $(\mathsf{t}_{\mathsf{N}}, \epsilon_{\mathsf{N}})$-break the security of NICA with $\mathsf{t}_{\mathsf{N}} = \mathsf{t}_{\mathcal{B}} = r \cdot n \cdot \mathsf{t}_\Lambda + r \cdot n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + r \cdot \mathsf{t}_{\mathsf{ReRand}}$, we must have $\epsilon_{\mathcal{B}} \leq \epsilon_{\mathsf{N}}$. Thus, if $\mathcal{R}$ is efficiently computable and re-randomizable, the loss of any simple reduction from breaking N to breaking the UF-SSA-security of $\mathcal{R}$ is at least linear in n.

5 New Applications

5.1 Signatures in the Multi-user Setting

Definitions. The syntax of digital signature schemes is defined in Sect. 2. Here, we define additional properties of signature schemes that are required to establish our result. Let $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ be a signature scheme. In the sequel we require *perfect correctness*, i.e., that for all $k \in \mathbb{N}$, all $\Pi \leftarrow^{\$} \mathsf{Setup}(1^k)$, all $(vk, sk) \leftarrow^{\$} \mathsf{Gen}(\Pi)$ and all m it holds that:

$$\Pr\left[\mathsf{SIG.Vfy}(vk, m, \sigma) = 1 : \sigma \leftarrow^{\$} \mathsf{SIG.Sign}(sk, m)\right] = 1.$$

Moreover, let $\Pi \leftarrow^{\$} \mathsf{Setup}(1^k)$ and let us recall that Π is contained in vk. We require an additional deterministic TM $\mathsf{SKCheck}_\Pi$ that takes as input strings sk and pk and outputs 0 or 1 such that:

$$\mathsf{SKCheck}_\Pi(pk, sk) = 1$$
$$\iff$$
$$\Pr\left[\mathsf{Vfy}(pk, m, \sigma) = 1 : m \leftarrow^{\$} |\mathcal{M}| \wedge \sigma \leftarrow^{\$} \mathsf{Sign}(sk, m)\right] = 1.$$

That is, $\mathsf{SKCheck}$ takes inputs sk and pk and returns 1 if and only if pk is a valid public key and sk is a corresponding secret key. Since we require perfect correctness for signature schemes, we have $\mathsf{SKCheck}(vk, sk) = 1$ whenever $(vk, sk) \leftarrow^{\$} \mathsf{Gen}(\Pi)$.

Definition 13. (Key re-randomization). *We say that a signature encryption scheme* SIG *is* t_{ReRand}*-key re-randomizable if there exists a Turing machine* $\mathsf{SIG.ReRand}$ *that runs in time at most* t_{ReRand}*, takes as input* $\Pi(vk, sk)$ *and returns* sk *uniformly distributed over* $\{sk : \mathsf{SKCheck}_\Pi(vk, sk) = 1\}$ *whenever* $\mathsf{SKCheck}_\Pi(vk, sk) = 1$.

Example 2. If we consider, for example, the Waters signature scheme [38], a public key consists among others of elements $g, g_1, g_2 \in \mathcal{G}$ where $g_1 = g^\alpha$. The key generation algorithm outputs a corresponding secret key as $sk = g_2^\alpha$. However, there may be other secret keys that might be accepted by $\mathsf{SKCheck}$.

To investigate this issue we shortly recall the signing and verification algorithms of [38]. The signing algorithm, when given as input a secret key and a message returns $\sigma = (\sigma_1, \sigma_2) = (g^r, sk \cdot (H(m))^r)$ where r is uniformly random chosen from $\mathbb{Z}_p$. Verification returns $e(g_1, g_2) \stackrel{?}{=} e(g, \sigma_2) \cdot e(\sigma_1, H(m))^{-1} = e(g, sk) \cdot e(g, H(m))^r \cdot e(g, H(m))^{-r}$.

We observe that by definition of $\mathsf{SKCheck}$ we must have $\mathsf{SKCheck}(vk, sk) = 1 \Leftrightarrow e(g_1, g_2) = e(g, sk)$. Thus there is an efficient $\mathsf{SKCheck}$ procedure. Moreover, since there is only one value that satisfies this equation in prime order groups we have an efficient secret key re-randomization algorithm, namely, the identity map. This is all that is to verify before applying our result.

Game $\mathsf{MU\text{-}EUF\text{-}CMA\text{-}C}^{n,\mu}_{\mathsf{SIG}}(\mathcal{A})$

$\Pi \xleftarrow{\$} \mathsf{SIG.Setup}(1^k)$	$\underline{\mathcal{O}.\mathsf{Sign}(m,i)}$
$(vk_i, sk_i) \xleftarrow{\$} \mathsf{SIG.Gen}(\Pi)$	**if** $\lvert Q_i \rvert \geq \mu$ **return** $\bot$
$\rho_{\mathcal{A}} \xleftarrow{\$} \{0,1\}^k$	$Q_i \leftarrow Q_i \cup \{m\}$
$Q^{\mathsf{Corrupt}} = Q_1 = \ldots = Q_n \leftarrow \emptyset$	**return** $\sigma \xleftarrow{\$} \mathsf{SIG.Sign}(sk_i, m)$
$(^*i, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}.\mathsf{Sign}(\cdot,\cdot),\mathcal{O}.\mathsf{Corrupt}(\cdot)}\left((vk_i)_{i\in[n]}; \rho_{\mathcal{A}}\right)$	$\underline{\mathcal{O}.\mathsf{Corrupt}(i)}$
return $vk_{i^*} \notin Q^{\mathsf{Corrupt}} \wedge m^* \notin Q_{i^*} \wedge \mathsf{SIG.Vfy}(vk_{i^*}, m^*, \sigma^*)$	$Q^{\mathsf{Corrupt}} \leftarrow Q^{\mathsf{Corrupt}} \cup \{vk_i\}$
	return sk_i

Fig. 5. MU-EUF-CMA-C-security game. The attacker has access to a signing oracle $\mathcal{O}$.Sign and a corrupt oracle $\mathcal{O}$.Corrupt.

Security Definition. The MU-EUF-CMA-C-security game is depicted in Fig. 5. Here the adversary $\mathcal{A}$ is provided with public keys $vk_1, \ldots, vk_n$ of the signature scheme. It may now adaptively issue *sign* and *corrupt*-queries. To issue a sign query it specifies a message m and a public key $vk_i, i \in [n]$ and obtains a valid signature σ over m that is valid with respect to vk_i. In order to issue a corrupt query, $\mathcal{A}$ specifies an index $i \in [n]$ and obtains a secret key sk_i that "matches" vk_i. Finally, $\mathcal{A}$ outputs a triplet (i, m, σ) and is considered successful if it did neither issue a corrupt query for i nor a sign query for (m, vk_i) and at the same time σ is valid over m with respect to vk_i.

Definition 14 (MU-EUF-CMA-C-security). *We say that an adversary (t, n, μ, ϵ)-breaks the* MU-EUF-CMA-C*-security of a signature scheme* SIG *if it runs in time t and*

$$\Pr\left[\mathsf{MU\text{-}EUF\text{-}CMA\text{-}C}^{n,\mu}_{\mathsf{SIG}}(\mathcal{A}) \Rightarrow 1\right] \geq \epsilon.$$

Definition 15. *We say that a Turing machine r-Γ is an r-simple $(\mathsf{t}_\Lambda, n, \mu, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reduction from breaking $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ to breaking the* MU-EUF-CMA-C*-security of* SIG*, if for any TM $\mathcal{A}$ that $(\mathsf{t}_{\mathcal{A}}, n, \mu, \epsilon_{\mathcal{A}})$-breaks the* MU-EUF-CMA-C *security of* SIG*, TM $\Lambda^{\mathcal{A}}$ $(\mathsf{t}_\Lambda + r \cdot \mathsf{t}_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N.*

The loss of an r-simple reduction Γ from breaking N to breaking the MU-EUF-CMA-C-security of SIG is defined similar to Definition 7.

Defining a Suitable Relation. Let $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ be a signature scheme and let I be the range of Setup. We set $\mathcal{R}_{\mathsf{SIG}} = \{R_\Pi\}_{\Pi \in I}$ where $R_\Pi(x, y) := \mathsf{SKCheck}_\Pi(y, x)$. Now, if SIG is t_{ReRand}-key re-randomizable then $\mathcal{R}_{\mathsf{SIG}}$ is t_{ReRand} re-randomizable.

UF-SSA *Security for $\mathcal{R}_{\mathsf{SIG}}$ is Weaker Than* MU-EUF-CMA-C*-Security for* SIG. Let now SIG be a perfectly correct signature scheme and let $\mathcal{R}_{\mathsf{SIG}}$ be derived from SIG as described in Sect. 5.1.

Claim. If there is an attacker $\mathcal{A}$ that (t, n, e)-breaks the UF-SSA-security for $\mathcal{R}_{\mathsf{SIG}}$ then there is an attacker $\mathcal{B}$ that $(t', n, 0, \epsilon')$-breaks the MU-EUF-CMA-C-security of SIG with $t' = \mathcal{O}(t)$ and $\epsilon' \geq \epsilon$.

Proof. We construct $\mathcal{B}$ that $(t', n, 0, \epsilon')$-breaks the MU-EUF-CMA-C-security of SIG, given black box access to $\mathcal{A}$ as follows:

1. $\mathcal{B}$ is called on input a set of public key $(vk)_{i\in[n]}$ and random tape ρ. Recall that Π are contained in vk. First, $\mathcal{B}$ samples and $\rho_{\mathcal{A}}$, the random coins of $\mathcal{A}$. After that, it runs $(j, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1\left(\Pi, (vk)_{i\in[n]}, \rho_{\mathcal{A}}\right)$.
2. $\mathcal{B}$ will issue a corrupt-query to oracle $\mathcal{O}$.Corrupt for all $i \in [n\backslash j]$. It will obtain sk_i such that $\mathsf{SKCheck}_{\Pi}(vk_i, sk_i)$. Next, $\mathcal{B}$ runs $sk_j \leftarrow^{\$} \mathcal{A}_2\left((sk_i)_{i\in[n\backslash j]}, st_{\mathcal{A}}\right)$. Note that $\mathsf{SKCheck}_{\Pi}(vk_j, sk_j) = 1$ with probability ϵ.
3. $\mathcal{B}$ samples $m \leftarrow^{\$} \mathcal{M}$ and computes $\sigma \leftarrow^{\$} \mathsf{SIG.Sign}(sk_j, m)$ and outputs (j, m, σ). Note that $vk_j \notin Q^{\mathsf{Corrupt}}$ and $m \notin Q_j$. Moreover, by the property of SKCheck we have $\mathsf{SIG.Vfy}(vk_j, m, \sigma) = 1$.

Tightness Bound

Theorem 6 (informal). *Any simple reduction from breaking the security of a* NICA N *to breaking the* MU-EUF-CMA-C*-security of a perfectly correct signature scheme* SIG *(cf. Definition 15) that provides* efficient key re-randomization *and that supports an efficient* SKCheck *loses a factor that is linear in the number of public keys the attacker is provided with and that it may corrupt, or N is easy to solve.*

We prove the Theorem via the following technical Theorem, which follows immediately from Theorem 5.

Theorem 7. *Let $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ be a non-interactive complexity assumption, $n, r \in \mathsf{poly}(k)$ and let $\mathcal{R}_{\mathsf{SIG}}$ be a family of computable relations as described above. Then for any r-simple $(t_\Gamma, n, \epsilon_\Gamma, 1)$-reduction Γ from breaking N to breaking the* UF-SSA*-security of $\mathcal{R}_{\mathsf{SIG}}$ there exists a TM $\mathcal{B}$ that $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks N where*

$$t_{\mathcal{B}} \leq r \cdot n \cdot \mathsf{t}_\Gamma + r \cdot n \cdot (n-1) \cdot \mathsf{t}_{\mathsf{Vfy}} + r \cdot \mathsf{t}_{\mathsf{ReRand}}$$
$$\epsilon_{\mathcal{B}} \geq \epsilon_\Gamma - \frac{r}{n}.$$

Here, $\mathsf{t}_{\mathsf{ReRand}}$ is the time to re-randomize a given valid witness and $\mathsf{t}_{\mathsf{Vfy}}$ is the maximum time needed to compute $R \in \mathcal{R}_{\mathsf{SIG}}$.

5.2 Public-Key Encryption in the Multi-user Setting

Our main result also applies to public key encryption in the multi-user setting with corruptions (and a similar result for key encapsulation mechanisms is straightforward). In the following, we only sketch the main steps to establishing our result. The full version contains a detailed, formal treatment. We start off by first defining MU-IND-CPA-C-security (Fig. 6), a security definition for public key encryption schemes $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ in the multi-user setting with corruptions. To apply our main result, we again have to formally define a

Game MU-IND-CPA-$\mathsf{C}^{n,\mu}_{\mathsf{PKE}}(\mathcal{A})$

$\Pi \leftarrow^{\$} \mathsf{PKE.Setup}(1^k)$	$\mathcal{O}.\mathsf{Encrypt}(m_0, m_1, i^*)$
$(pk_i, sk_i) \leftarrow^{\$} \mathsf{PKE.Gen}(\Pi)$	**if** $\lvert m_0\rvert \neq \lvert m_1\rvert$ **return** $\bot$
$\rho_{\mathcal{A}} \leftarrow^{\$} \{0,1\}^k$	$b \leftarrow^{\$} \{0,1\}$
$Q^{\mathsf{Corrupt}} \leftarrow \emptyset$	**return** $c \leftarrow^{\$} \mathsf{Enc}(pk_i, m_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}.\mathsf{Encrypt}(\cdot,\cdot,\cdot),\mathcal{O}.\mathsf{Corrupt}(\cdot)}\left((pk_i)_{i\in[n]}; \rho_{\mathcal{A}}\right)$	$\mathcal{O}.\mathsf{Corrupt}(i)$
return $b = b' \wedge pk_{i^*} \notin Q^{\mathsf{Corrupt}}$	$Q^{\mathsf{Corrupt}} \leftarrow Q^{\mathsf{Corrupt}} \cup \{pk_i\}$
	return sk_i

Fig. 6. MU-IND-CPA-C-security game. The attacker has access to an encryption oracle $\mathcal{O}.\mathsf{Encrypt}$ which may be queried only once and a corrupt oracle $\mathcal{O}.\mathsf{Corrupt}$.

family $\mathcal{R}_{\mathsf{PKE}}$ of suitable computable relations. To this end (and similar to the case of digital signatures in the multi user setting), we require the existence of an additional TM $\mathsf{SKCheck}_\Pi$ for $\Pi \leftarrow^{\$} \mathsf{Setup}(1^k)$ such that

$$\mathsf{SKCheck}_\Pi(pk, sk) = 1 \iff \Pr\left[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m : m \leftarrow^{\$} \mathcal{M}\right] = 1.$$

That is, $\mathsf{SKCheck}$ takes inputs sk and pk and returns 1 if and only if pk is a PKE public key and sk is a secret key corresponding to public key pk. To define our suitable relation, we set $\mathcal{R}_{\mathsf{PKE}} = \{R_\Pi\}_{\Pi\in I}$ where $R_\Pi(x, y) := \mathsf{SKCheck}_\Pi(y, x)$ and I is the set of all public parameters that can be output by Setup. Finally, we show that MU-IND-CPA-C-security for PKE is stronger than UF-SSA-security for $\mathcal{R}_{\mathsf{PKE}}$. Via our main result, this immediately proves that any security reduction must have a security loss that is (at least) linear in the number of public keys considered in the MU-IND-CPA-C-security experiment.

5.3 Non-interactive Key Exchange

In this section we will show how to apply our main result to non-interactive key exchange (NIKE) [25]. This case differs from the cases considered before in that we will have to define a relation $R(x, y)$, which is not efficiently verifiable, given just x and y. Instead, we will need additional information, which will be available in the NIKE security experiment. Formally, we consider again UF-SSA-security for some relation R but model $\mathcal{A}_2$ as an oracle machine. The responses of the oracle may depend on the output of $\mathcal{A}_1$. We explain that this makes it possible to extend the range of covered cryptographic primitives to NIKE.

Definitions. Following [16,25], a NIKE protocol consists of three PPT-TMs with the following syntax:

Public Parameters. On input 1^k, the public parameter generation machine $\Pi \leftarrow^{\$} \mathsf{NIKE.Setup}(1^k)$ outputs a set Π of system parameters.

Key Generation. The key generation machine takes as input Π and outputs a random key pair (sk_i, pk_i) for party i, i.e. $(sk_i, pk_i) \leftarrow^{\$} \mathsf{NIKE.Gen}(\Pi)$. We assume that pk contains Π and 1^k.

Shared Key Generation. The deterministic shared key machine $\mathsf{SharedKey}$ takes as input (sk_i, pk_j) and outputs a shared key $\mathsf{K}_{i,j}$ in time $\mathsf{t}_{\mathsf{Vfy}}$, where $\mathsf{K}_{i,j} = \perp$ if $i = j$.

We require perfect correctness, that is,

$$\Pr\left[\mathsf{SharedKey}(sk_i, pk_j) = \mathsf{SharedKey}(sk_j, pk_i)\right] = 1$$

for all $\Pi \leftarrow^{\$} \mathsf{NIKE.Setup}(1^k)$ and $(pk_i, sk_i), (pk_j, sk_j) \leftarrow^{\$} \mathsf{NIKE.Gen}(\Pi)$.

We require an additional Turing machine $\mathsf{PKCheck}$ that inputs strings Π and pk and evaluates to true if pk is in the range of $\mathsf{NIKE.Gen}(\Pi)$. Moreover, whenever two public keys pk and pk' are accepted by $\mathsf{PKCheck}$, we require that the respective shared key is uniquely determined, given only pk and pk'. In the sequel we will denote this key by $K(pk, pk')$ and call NIKE *unique*. The pairing-based NIKE scheme from [25] satisfies uniqueness.

NIKE Security. There exists several different, but polynomial-time equivalent [25] security models for NIKE. Of course the tightness of a reduction depends on the choice of the security model. Indeed, the weakest security model considered in [25] is the *CKS-light* model. However, this model is strongly idealized. The reduction from breaking security in a stronger and more realistic security model (called the CKS model in [25]) to breaking security in this idealized model loses a factor of n^2, where n is the number of users. We show that this loss is inherent for NIKE schemes with the properties defined above.

CKS-*Security for NIKE.* The *CKS-security* experiment is depicted in Fig. 7.

Game $\mathsf{CKS}^{n,\mathcal{A}}_{\mathsf{NIKE}}(1^k)$

$\Pi \leftarrow^{\$} \mathsf{Setup}(1^k)$	$\mathcal{O}.\mathsf{Corrupt}(i)$
$(pk_i, sk_i) \leftarrow^{\$} \mathsf{NIKE.Gen}(\Pi)$	$Q^{\mathsf{Corrupt}} \leftarrow Q^{\mathsf{Corrupt}} \cup \{pk_i\}$
$\rho_{\mathcal{A}} \leftarrow^{\$} \{0,1\}^k$	**return** sk_i
$Q^{\mathsf{Corrupt}} = Q^{\mathsf{Reveal}} \leftarrow \emptyset$	
$b' \leftarrow^{\$} \mathcal{A}^{\mathcal{O}.\mathsf{Corrupt}(\cdot),\mathcal{O}.\mathsf{Reveal}(\cdot,\cdot),\mathcal{O}.\mathsf{Test}(\cdot,\cdot)}\left(\Pi, (pk_i)_{i\in[n]}; \rho_{\mathcal{A}}\right)$	$\mathcal{O}.\mathsf{Reveal}(i,j)$
return $b' = b \wedge pk_{i^*}, pk_{j^*} \notin Q^{\mathsf{Corrupt}} \wedge (i^*, j^*) \notin Q^{\mathsf{Reveal}}$	$Q^{\mathsf{Reveal}} \leftarrow Q^{\mathsf{Reveal}} \cup \{(i,j)\}$
	return $\mathsf{SharedKey}(sk_i, pk_j)$
$\mathcal{O}.\mathsf{Test}(i^*, j^*)$	
$K_0 \leftarrow \mathsf{SharedKey}(sk_{i^*}, pk_{j^*})$; $K_1 \leftarrow^{\$} \mathsf{SharedKey}(\cdot,\cdot)$	
$b \leftarrow^{\$} \{0,1\}$	
return K_b	

Fig. 7. CKS-Security game for NIKE. Oracle $\mathcal{O}.\mathsf{Test}$ may be queried only once. K_1 is sampled uniform from the range of $\mathsf{SharedKey}$.

Definition 16. *We say that an adversary $\mathcal{A}$ (t, n, ϵ)-breaks the* CKS*-security of a non-interactive key exchange protocol* NIKE *if it runs in time at most t and*

$$\Pr\left[\mathsf{CKS}^{n,\mathcal{A}}_{\mathsf{NIKE}}(1^k) \Rightarrow 1\right] \geq \epsilon.$$

Definition 17. *We say that a Turing machine r-Γ is an r-simple $(\mathsf{t}_\Lambda, n, \epsilon_\Lambda, \epsilon_\mathcal{A})$-reduction from breaking $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ to breaking the* CKS*-security of* NIKE*, if for any TM $\mathcal{A}$ that $(\mathsf{t}_\mathcal{A}, n, \epsilon_\mathcal{A})$-breaks the* CKS *security of* NIKE*, TM $\Lambda^\mathcal{A}$ $(\mathsf{t}_\Lambda + r \cdot \mathsf{t}_\mathcal{A}, \epsilon_\Lambda)$-breaks N.*

The loss of an r-simple reduction Γ from breaking the security of N to breaking the CKS-security of NIKE is defined similar to Definition 7.

Defining a Suitable Relation. Let $\mathsf{NIKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{SharedKey})$ be a unique NIKE scheme and let I be the range of Setup. We set $\mathcal{R}_\mathsf{NIKE} = \{R_\Pi\}_{\Pi \in I}$ where

$$R_\Pi(x, (y_1, y_2)) = 1 \Leftrightarrow x = K(y_1, y_2).$$

Let us fix Π for the moment. Note that the attacker is provided with $\tilde{n} = (n-1) \cdot n$ R_Π statements if it is provided with n NIKE-public keys.

Let now $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ denote an attacker against the UF-SSA-security of $\mathcal{R}_\mathsf{NIKE}$. Because R may not be efficiently verifiable, we let $\mathcal{A}_2$ have oracle access to Oracle $\mathsf{Corrupt}_{i^*,j^*}$ that returns secret key sk_i when queried on input $i \in [n \backslash \{i^*, j^*\}]$. Here $K(pk_{i^*}, pk_{j^*})$ is the shared key that $\mathcal{A}$ needs to compute to break the UF-SSA security of $\mathcal{R}$ and n is the number of public keys that $\mathcal{A}$ is provided with (note that this leads to $\tilde{n}$ NIKE shared keys).

UF-SSA-*Security for* $\mathcal{R}_\mathsf{NIKE}$ *is Weaker Than* CKS-*Security for* NIKE. Next, we show that any adversary that breaks the UF-SSA-security of $\mathcal{R}_\mathsf{NIKE}$ then there is an attacker that breaks the CKS-security of NIKE.

Claim. If there is an attacker $\mathcal{A}$ that $(t, \tilde{n}, \epsilon)$-breaks the UF-SSA-security of $\mathcal{R}_\mathsf{NIKE}$ then there is an attacker $\mathcal{B}$ that (t', n, ϵ')-breaks the CKS-security of NIKE with $t' = \mathcal{O}(t)$ and $\epsilon' \geq \epsilon$.

Proof. We construct $\mathcal{B}$ that (t', n, ϵ')-breaks the CKS-security of NIKE, given black box access to $\mathcal{A}$ as follows:

1. $\mathcal{B}$ is called on input a set of public keys $(pk)_{i \in [n]}$ and random tape ρ. Recall that Π is contained in pk. First, $\mathcal{B}$ samples and $\rho_\mathcal{A}$, the random coins of $\mathcal{A}$. Next, it runs $((i^*, j^*), st_\mathcal{A}) \leftarrow \mathcal{A}_1\left(\Pi, (pk)_{i \in [n]}, \rho_\mathcal{A}\right)$. Note that n public keys define $n \cdot (n-1)$ statements for R_Π. The one that $\mathcal{A}$ will compute is determined by i^* and j^*.
2. $\mathcal{B}$ will issue a reveal-query to oracle $\mathcal{O}.\mathsf{Reveal}$ for all $(i, j) \in [n]^2 \backslash \{(i^*, j^*)\}, i \neq j$. It will obtain $K_{i,j} = \mathsf{SharedKey}(sk_i, pk_j)$. Next, $\mathcal{B}$ runs

$$K^* \leftarrow^\$ \mathcal{A}_2^{\mathcal{O}.\mathsf{Corrupt}_{i^*,j^*}(\cdot)}\left((K_{i,j})_{(i,j) \in [n]^2 \backslash \{i^*, j^*\}, i \neq j}, st_\mathcal{A}\right).$$

   $\mathcal{B}$ provides $\mathcal{A}$ with oracle $\mathsf{Corrupt}_{i^*,j^*}$ by forwarding all queries to oracle $\mathcal{O}.\mathsf{Corrupt}()$ and forwarding the response back to $\mathcal{A}$. Note that, using sk_i, $\mathcal{A}$ may efficiently check whether $K_{i,j} = \mathsf{SharedKey}(sk_i, pk_j)$ for all $j \in [n]$. By assumption it holds that $K^* = \mathsf{SharedKey}(sk_{i^*}, pk^{j^*})$ with probability at least ϵ.

3. Next, $\mathcal{B}$ issues (i^*, j^*) to oracle $\mathcal{O}.\mathsf{Test}()$ which will respond with K. $\mathcal{B}$ returns 0 if $K = K^*$ and 1 otherwise. Note that by construction of oracle $\mathsf{Corrupt}_{i^*,j^*}$ it holds that i^*, $j^* \notin Q^{\mathsf{Corrupt}}$. Moreover, by the perfect correctness of NIKE and the uniqueness of shared keys $\mathcal{B}$ is successful whenever $\mathcal{A}$ is successful.

Tightness Bounds

Theorem 8 (informal). *Any simple reduction from breaking the security of a* NICA N *to breaking the* CKS*-security of a perfectly correct, unique NIKE scheme* NIKE *(cf. Definition 16) that supports an efficient* $\mathsf{PKCheck}$ *loses a factor that is* quadratic *in the number of public keys the attacker is provided with and that it may corrupt, or* N *is easy to solve.*

We prove the Theorem via the following technical Theorem.

Theorem 9. *Let* $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ *be a non-interactive complexity assumption,* $\tilde{n}, r \in \mathsf{poly}(k)$ *and let* $\mathcal{R}_{\mathsf{NIKE}}$ *be a family of computable relations as described above. Then for any* r*-simple* $(t_\Gamma, \tilde{n}, \epsilon_\Gamma, 1)$*-reduction* Γ *from breaking* N *to breaking the* $\mathsf{UF\text{-}SSA}$*-security of* $\mathcal{R}_{\mathsf{NIKE}}$ *there exists a TM* $\mathcal{B}$ *that* $(t_\mathcal{B}, \epsilon_\mathcal{B})$*-breaks* N *where*

$$t_\mathcal{B} \leq r \cdot \tilde{n} \cdot \mathsf{t}_\Gamma + r \cdot \tilde{n} \cdot (\tilde{n} - 1) \cdot \mathsf{t}_{\mathsf{Vfy}} \text{ and } \epsilon_\mathcal{B} \geq \quad \epsilon_\Gamma - \frac{r}{\tilde{n}}.$$

Here, $\mathsf{t}_{\mathsf{Vfy}}$ *is the maximum time needed to compute* $R \in \mathcal{R}_{\mathsf{NIKE}}$ *with access to* $\mathsf{Corrupt}_{i^*,j^*}$.

Interpretation. As mentioned before, if the attacker is provided with $\tilde{n}$ statements, it is provided only with $\approx \sqrt{\tilde{n}}$ public keys. Thus, the loss of any r-simple reduction is *quadratic* in the number of public keys if the underlying problem is assumed to be hard.

Our lower bound for NIKE can easily be generalized to systems where keys are derived from $\ell = O(\log(k))$ parties for security parameter k. Syntactically, the difference is that $\mathsf{SharedKey}$ now takes as input $\ell - 1$ public keys and a single secret key. Now, the attacker obtains $\tilde{n}$ statements and $\approx \tilde{n}^{1/\ell}$ public keys. Thus, the loss of any r-simple reduction grows with an exponent of ℓ in the number of public keys.

Extending the Result to Interactive Key Exchange. On the one hand, our NIKE bounds do not carry over directly to arbitrary interactive key exchange protocols, because these do not necessarily meet the properties of NIKE schemes that we need to put up. In particular, we have to require that any pair of NIKE public keys uniquely determines the corresponding shared key (which limits the generality of the result, but appears very reasonable for natural (and possibly all) NIKE constructions, in particular it holds for the NIKE schemes of [25]). This requirement does not hold for interactive AKE protocols, where the shared key may additionally depend on ephemeral random values (nonces or Diffie-Hellman shares, for example) exchanged between parties.

On the other hand, our tightness bounds for signatures and public-key encryption (with unique/re-randomizable secret keys, in the multi-user setting with corruptions) directly imply tightness bounds for AKE protocols that use these primitives, and where the attacker is able to adaptively corrupt the secret keys of these signature/PKE schemes. Note that this includes the vast majority of all known AKE constructions. The tightly-secure key exchange protocol of [4] overcomes this hurdle by using a signature scheme that does not have unique/re-randomizable secret keys, and this is used in a crucial way (cf. the "Naor-Yung trick for signatures" in [4]).

A Summary of Coron's Meta-reduction and Its Generalizations

EUF-CMA-security is commonly considered the standard security definition for digital signature schemes [28]. The security game is depicted in Fig. 8.

Game $\mathsf{EUF\text{-}CMA}^{n,\mathcal{A}}_{\mathsf{SIG}}(1^k)$

$\Pi \leftarrow^{\$} \mathsf{Setup}(1^k)$	$\underline{\mathcal{O}.\mathsf{Sign}(m)}$
$(vk, sk) \leftarrow^{\$} \mathsf{Gen}(\Pi)$	**if** $\lvert Q\rvert \geq n$ **return** $\bot$
$\rho_{\mathcal{A}} \leftarrow^{\$} P_{\mathcal{A}}$	$Q \leftarrow Q \cup \{m\}$
$Q \leftarrow \emptyset$	**return** $\sigma \leftarrow^{\$} \mathsf{SIG.Sign}(sk, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}.\mathsf{Sign}(\cdot)}(vk; \rho_{\mathcal{A}})$	
return $m^* \notin Q \wedge \mathsf{SIG.Vfy}(vk, m^*, \sigma^*)$	

Fig. 8. EUF-CMA-Security game. When called, the attacker has access to a signing oracle $\mathcal{O}.\mathsf{Sign}$.

Definition 18. (EUF-CMA-security). *We say that an attacker (t, n, ϵ)-breaks the* EUF-CMA*-security of a signature scheme* SIG *if it runs in time t and*

$$\Pr\left[\mathsf{EUF\text{-}CMA}^{n,\mathcal{A}}_{\mathsf{SIG}}(1^k) \Rightarrow 1\right] \geq \epsilon.$$

Definition 19. *We say that a Turing machine r-Γ is an r-simple $(\mathsf{t}_\Lambda, n, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reduction from breaking $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ to breaking the* EUF-CMA*-security of* SIG*, if for any TM $\mathcal{A}$ that $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$-breaks the* EUF-CMA *security of* SIG*, TM $\Lambda^{\mathcal{A}}$ $(\mathsf{t}_\Lambda + r \cdot \mathsf{t}_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N.*

Definition 20. *Let $\ell : \mathbb{N} \to \mathbb{N}$. We say that an r-simple reduction Γ from breaking N to breaking the* EUF-CMA*-security of* SIG loses ℓ*, if there exists an adversary $\mathcal{A}$ that $(\mathsf{t}_{\mathcal{A}}, n, \epsilon_{\mathcal{A}})$-breaks the* EUF-CMA *security of* SIG*, such that $\Lambda^{\mathcal{A}}$ $(\mathsf{t}_\Lambda + \mathsf{t}_{\mathcal{A}}, \epsilon_\Lambda)$-breaks N with*

$$\frac{\mathsf{t}_\Lambda(k) + \mathsf{t}_{\mathcal{A}}(k)}{\epsilon_\Lambda(k)} \geq \ell(k) \cdot \frac{\mathsf{t}_{\mathcal{A}}(k)}{\epsilon_{\mathcal{A}}(k)}.$$

The following lemma is due to Hofheinz *et al.* [30] and generalizes a result from Coron [18].

Lemma 1 ([18,30]). *Let N be a $(t_N, \epsilon_{\mathsf{N}})$-secure non-interactive complexity assumption where $\epsilon_{\mathsf{N}} \in \mathsf{negl}(k)$ and let SIG be a* unique *signature scheme with message space of size 2^l. If Γ is a $(t_\Gamma, n, \epsilon_\Gamma)$-reduction from breaking N to breaking the $\mathsf{EUF\text{-}CMA}$-security of SIG and $\mathsf{t_N} \geq 2 \cdot \mathsf{t_\Gamma} + \mathsf{t_{ReRand}}$ then*

$$\epsilon_\Gamma \leq \epsilon_{\mathcal{A}} \cdot \frac{\mathsf{exp}(-1)}{n} \cdot \left(1 - \frac{n}{2^l}\right)^{-1} + \mathsf{negl}(k). \qquad \square$$

Coron [18] and Hofheinz *et al.* [30] conclude that we have $\epsilon_\Lambda = \mathcal{O}\left(\frac{\epsilon_{\mathcal{A}}}{n}\right)$. The conclusion builds on the fact that $2^l \gg n$. This is reasonable for most digital signatures schemes.

B UF-SMA-Security Is Strictly Weaker Than EUF-CMA-Security

We show that any attacker $\mathcal{A}$ that breaks the $\mathsf{UF\text{-}SMA}$-security of a signature scheme SIG implies an attacker $\mathcal{A}'$ that breaks the $\mathsf{EUF\text{-}CMA}$-security (depicted in Fig. 8) of SIG in roughly the same running time and with the same probability of success. Moreover $\mathsf{UF\text{-}SMA}$-security and $\mathsf{EUF\text{-}CMA}$-security are *not* polynomially equivalent.

Claim. Let SIG be a signature scheme. If there is an attacker $\mathcal{A}$ that (t, n, ϵ)-breaks the $\mathsf{UF\text{-}SMA}$-security of a signature scheme SIG then there is an attacker $\mathcal{B}$ that (t', n, ϵ')-breaks the $\mathsf{EUF\text{-}CMA}$-security of SIG where $t' = \mathcal{O}(t)$ and $\epsilon' \geq \epsilon$.

Proof. We construct $\mathcal{B}$ that (t', n, ϵ')-breaks the $\mathsf{EUF\text{-}CMA}$-security of SIG, given black box access to $\mathcal{A}$ as follows:

1. $\mathcal{B}$ is called on input a public key vk and random tape ρ. First, $\mathcal{B}$ samples n *distinct* messages $m_1, \ldots, m_n$ from the message space and $\rho_{\mathcal{A}}$, the random coins of $\mathcal{A}$. After that, it runs $(j, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1\left(vk, (m_i)_{i \in [n]}, \rho_{\mathcal{A}}\right)$.
2. $\mathcal{B}$ will issue a sign-query to oracle Sign for all messages $m_i, i \in [n \backslash j]$. It will obtain $\sigma_i \leftarrow^{\$} \mathsf{SIG.Sign}(sk, m_i)$. Note that σ_i is a valid signature over m_i with respect to vk. Next, $\mathcal{B}$ runs $\sigma_j \leftarrow^{\$} \mathcal{A}_2\left((\sigma_i)_{i \in [n \backslash j]}, st_{\mathcal{A}}\right)$ which is valid with probability ϵ.
3. $\mathcal{B}$ outputs (m_j, σ_j). Note that due to the fact that $m_i \neq m_j$ for all $i \neq j$, this is a valid forgery which is valid with probability at least ϵ.

Let SIG be a signature scheme with exponential message space $\mathcal{M}$. Let $m \leftarrow^{\$} \mathcal{M}$. Then we define a signature scheme $\mathsf{SIG}'(m)$ that works exactly like SIG except the $\mathsf{SIG}'(m)$-verification machine will accept 0 as a valid signature over m.

Claim. Suppose that no adversary (t, n, ϵ)-breaks the $\mathsf{EUF\text{-}CMA}$-security of SIG. Then the following holds: 1. There is no adversary that (t, n, ϵ')-breaks the $\mathsf{UF\text{-}SMA}$-security of $\mathsf{SIG}'(m)$ with $\epsilon' \geq \epsilon + \frac{n}{|\mathcal{M}|}$. 2. There exists a trivial attack strategy that $(\mathcal{O}(1), 0, 1)$-breaks the $\mathsf{EUF\text{-}CMA}$-security of $\mathsf{SIG}'(m)$.

Proof. 1. Recall that at the beginning of the UF-SMA security experiment, $\mathcal{A}$ is called on input a verification key and n distinct messages that are sampled uniformly from $\mathcal{M}$. Now, the probability that $m_i = m$ for $i \in [n]$ is upper bounded by $\frac{n}{|\mathcal{M}|}$. However, if for all $i \in [n]$ we have $m_i \neq m$ then we can apply the previous claim. When called on vk, $\mathcal{A}$ simply outputs $(m, 0)$ which is a valid forgery.

References

1. Abdalla, M., Fouque, P.-A., Lyubashevsky, V., Tibouchi, M.: Tightly-secure signatures from lossy identification schemes. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 572–590. Springer, Heidelberg (2012)
2. Abe, M., Groth, J., Ohkubo, M.: Separating short structure-preserving signatures from non-interactive assumptions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 628–646. Springer, Heidelberg (2011)
3. Bader, C.: Efficient signatures with tight real world security in the random-oracle model. In: Gritzalis, D., Kiayias, A., Askoxylakis, I. (eds.) CANS 2014. LNCS, vol. 8813, pp. 370–383. Springer, Heidelberg (2014)
4. Bader, C., Hofheinz, D., Jager, T., Kiltz, E., Li, Y.: Tightly-secure authenticated key exchange. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 629–658. Springer, Heidelberg (2015)
5. Baldimtsi, F., Lysyanskaya, A.: On the security of one-witness blind signature schemes. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 82–99. Springer, Heidelberg (2013)
6. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)
7. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: simplified proof and improved concrete security for waters' IBE scheme. Cryptology ePrint Archive, Report 2009/084 (2009). http://eprint.iacr.org/2009/084
8. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: simplified proof and improved concrete security for waters' ibe scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009)
9. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
10. Blazy, O., Kakvi, S.A., Kiltz, E., Pan, J.: Tightly-secure signatures from chameleon hash functions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 256–279. Springer, Heidelberg (2015)
11. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 408–425. Springer, Heidelberg (2014)
12. Boneh, D., Venkatesan, R.: Breaking RSA may not be equivalent to factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 59–71. Springer, Heidelberg (1998)
13. Bresson, E., Monnerat, J., Vergnaud, D.: Separation results on the "one-more" computational problems. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 71–87. Springer, Heidelberg (2008)

14. Brown, D.R.L.: Irreducibility to the one-more evaluation problems: more may be less. Cryptology ePrint Archive, Report 2007/435 (2007). http://eprint.iacr.org/
15. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer, Heidelberg (2001)
16. Cash, D.M., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
17. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013)
18. Coron, J.-S.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002)
19. Cramer, R., Damgård, I.B.: New generation of secure and practical RSA-based signatures. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 173–185. Springer, Heidelberg (1996)
20. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 449–466. Springer, Heidelberg (2005)
21. Dodis, Y., Reyzin, L.: On the power of claw-free permutations. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 55–73. Springer, Heidelberg (2003)
22. Fischlin, M., Fleischhacker, N.: Limitations of the meta-reduction technique: the case of Schnorr signatures. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 444–460. Springer, Heidelberg (2013)
23. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Heidelberg (2010)
24. Fleischhacker, N., Jager, T., Schröder, D.: On tight security proofs for Schnorr signatures. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 512–531. Springer, Heidelberg (2014)
25. Freire, E.S.V., Hofheinz, D., Kiltz, E., Paterson, K.G.: Non-interactive key exchange. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 254–271. Springer, Heidelberg (2013)
26. Garg, S., Bhaskar, R., Lokam, S.V.: Improved bounds on security reductions for discrete log based signatures. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 93–107. Springer, Heidelberg (2008)
27. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
28. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988)
29. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 590–607. Springer, Heidelberg (2012)
30. Hofheinz, D., Jager, T., Knapp, E.: Waters signatures with optimal security reduction. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 66–83. Springer, Heidelberg (2012)
31. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 537–553. Springer, Heidelberg (2012)

32. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003, pp. 155–164. ACM Press, October 2003
33. Lewko, A., Waters, B.: Why proving HIBE systems secure is difficult. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 58–76. Springer, Heidelberg (2014)
34. Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 1–20. Springer, Heidelberg (2005)
35. Paillier, P., Villar, J.L.: Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer, Heidelberg (2006)
36. Schäge, S.: Tight proofs for signature schemes without random oracles. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 189–206. Springer, Heidelberg (2011)
37. Seurin, Y.: On the exact security of schnorr-type signatures in the random oracle model. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 554–571. Springer, Heidelberg (2012)
38. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)