Completely Non-malleable Schemes

(Extended Abstract)

Marc Fischlin*

Institute for Theoretical Computer Science, ETH Zürich, Switzerland
marc.fischlin@inf.ethz.ch
http://www.fischlin.de/

Abstract. An encryption scheme is non-malleable if the adversary cannot transform a ciphertext into one of a related message under the given public key. Although providing a very strong security property, some application scenarios like the recently proposed key-substitution attacks yet show the limitations of this notion. In such settings the adversary may have the power to transform the ciphertext *and* the given public key, possibly without knowing the corresponding secret key of her own public key. In this paper we therefore introduce the notion of completely non-malleable cryptographic schemes withstanding such attacks. We show that classical schemes like the well-known Cramer-Shoup DDH encryption scheme become indeed insecure against this stronger kind of attack, implying that the notion is a strict extension of chosen-ciphertext security. We also prove that, unless one puts further restrictions on the adversary's success goals, completely non-malleable schemes are hard to construct (as in the case of encryption) or even impossible (as in the case of signatures). Identifying the appropriate restrictions we then show how to modify well-known constructions like RSA-OAEP and Fiat-Shamir signatures yielding practical solutions for the problem in the random oracle model.

1 Introduction

According to the seminal paper by Dolev et al. [7] an encryption scheme is called non-malleable if giving a ciphertext to an adversary does not significantly help this adversary to produce a ciphertext of a related message under the same public key. Analogous requirements can be formulated for other cryptographic primitives like signatures or commitments. While this definition of non-malleability is already quite strong and suffices in most settings it yet leaves open if there are cases where refined notions are needed and, if so, whether they can be achieved at all.

* This work was supported by the Emmy Noether Programme Fi 940/1-1 of the German Research Foundation (DFG). Part of this work done while visiting University of California, San Diego, USA.

L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, pp. 779–790, 2005.

Motivation. A possible stronger definition of non-malleability, introduced here as complete non-malleability, basically allows the adversary to transform the public key as well. That is, in case of encryption the adversary may output a ciphertext of a related message *under an adversarial chosen public key.* For this, the adversary does not even need to know the matching secret key to the chosen public key.

Our initial interest in completely non-malleable schemes stems from the area of (regular) non-malleable commitments. Previous constructions of such non-malleable commitments usually require a common reference string [4, 8, 5, 9], or are rather theoretical in terms of efficiency [7, 1]. Coming up with an efficient non-malleable commitment protocol in the plain model is still an open problem.

Early in cryptography it has been observed that efficient commitment schemes can be derived from encryption schemes. To commit, the sender creates a key pair and sends a ciphertext of the message together with the public key. To decommit, the sender transmits the message with the random bits used to create the ciphertext, or simply sends the secret key (if appropriate). Now, if the encryption scheme was *completely* non-malleable then the resulting commitment scheme in this basic construction would be non-malleable in the ordinary sense. And the derived commitment scheme would be non-interactive and would not rely on public parameters either.

In addition to the application to commitment schemes, it turns out that, recently, the problem of complete non-malleability also appeared in similar flavors in related areas like signatures or hash functions [3, 13, 12]. For example, Blake-Wilson and Menezes [3] show how to deploy unknown key-share attacks to show weaknesses in the station-to-station key agreement protocol. In their case, the adversary is given a signature s for message m under some public verification key vk and her task is to find a different key pair (sk^*, vk^*) such that s is also a valid signature for m under vk^*.

Our Results. In this work we discuss the issue of complete non-malleability for public-key encryption and signatures. We first show that most of the well-known encryption and signature schemes fall prey to complete non-malleability attacks. Specifically, we propose attacks against the Cramer-Shoup DDH encryption scheme, RSA-OAEP and signatures of the Fiat-Shamir type like Schnorr signatures (of which only the first one appears in this version). This shows that the security notion of complete non-malleability is not covered by chosen-ciphertext security and by unforgeability against chosen-message attacks, respectively.

Then we give a formal framework for complete non-malleability of public-key encryption and signatures. There are two major differences to the basic definition of non-malleability. First, the adversary's goal in the definition of [7] for encryption is to relate the original secret message m to a chosen message m^* via a relation $\mathsf{R}(m, m^*)$. Here we extend the relation to include the given public key pk. For message-only relations it remains for example unclear if it is easy to modify a ciphertext of some message m under some RSA-based non-malleable encryption with random RSA-exponent e into a ciphertext of the related message $m^* = m+e$ under the same public key. We answer this in the affirmative, showing

that this is indeed easy for general schemes. Namely, we present a scheme which is non-malleable for relations over messages, but for which the adversary can easily produce a ciphertext c^* of a message m^* under pk such that a specific relation $\mathsf{R}(pk, m, m^*)$ is satisfied. We stress that the adversary does not even take advantage of the possibility to select her own public key for this attack.

Our separating example for relations $\mathsf{R}(m, m^*)$ over messages shows that (regular) non-malleable commitments constructed by means of encryption schemes in the common reference string model (as in [5]) may not provide adequate security for the classical Internet auction example. In the auction case the users' bids are encrypted with a public key published in the reference string. Now, an adversarial bidder may be able to transform such a sealed bid of an honest user into one which is related via this public key, and may thus overbid this user easily with a reasonably small amount (e.g., by $m^* = m + e$).

The second, and more significant extension of the [7] framework for encryption is that the adversary now has the power to tamper the public key. Consequently, the relations now also range over the given public key pk, the adversarial chosen public key pk^* and, for sake of generality, also over adversary's ciphertext. Similarly, for signatures we let the relation include the given verification key vk, the adversarial key vk^*, message m^* and signature s^*.

Concerning constructions of completely non-malleable schemes, the bad news is that schemes *for general relations* are hard to derive or even impossible. We show that there are relations where complete non-malleability cannot be proven via black-box proofs for both encryption and signatures. Even worse, for more complex relations we prove that completely non-malleable signature schemes do not exist at all.

On the positive side, we can show that practical schemes like RSA-OAEP and Fiat-Shamir signatures can be made completely non-malleable *in the random oracle model* (while the basic versions do not achieve this goal, not even in the random oracle model). Security holds for a broad class of relations which, roughly, excludes only such relations for which we are able to show our unconditional impossibility results. Also, our solutions are essentially as efficient as the original schemes, thus giving us complete non-malleability almost for free.

However, we remark that the completely non-malleable versions of the schemes above are proven secure in the random oracle model only. A closer look reveals why this model provides a useful countermeasure: Random oracles are by nature highly non-malleable constructs, because outputs of related inputs are completely uncorrelated and because all users in the system use the *same* hash function oracle as a common anchor. The advantage of giving security of these schemes in terms of complete non-malleability, even in the random oracle model, is that security now follows for a vast number of attacks, including for example so-called key-substitution and strong-unforgeability attacks. That is, any attack where the adversary's goal can be cast through such relations provably fails; extra security proofs become obsolete. An interesting open question is whether there are secure schemes in the plain model for interesting relations or not.

Organization. To provide some intuition about the power of complete non-malleability attack we start with the attack on the Cramer-Shoup encryption scheme in Section 2. Then we define completely non-malleable schemes formally in Section 3. Because of the complexity of the topic we mainly focus on the definitions. Our impossibility and positive results are outlined in Section 4, further details appear in the full version.

2 Attack on Cramer-Shoup Encryption Scheme

The Cramer-Shoup encryption scheme [6] is semantically secure against adaptive chosen-ciphertext attacks under the decisional Diffie-Hellman assumption. It is thus also non-malleable (in the classical sense) with respect to such attacks.

Key Generation: The public key is given by the description of a group $\mathcal{G}_q$ of prime order q for which the decisional Diffie-Hellman problem is believed to be intractable, two random generators g_1, g_2 of this group as well as c, d and h where

$$c = g_1^{x_1} g_2^{x_2}, \qquad d = g_1^{y_1} g_2^{y_2}, \qquad h = g_1^{z_1} g_2^{z_2}$$

for random values $x_1, x_2, y_1, y_2, z_1, z_2 \leftarrow \mathbb{Z}_q$. The public key also contains a collision-intractable hash function H. The secret key is $(x_1, x_2, y_1, y_2, z_1, z_2)$.

Encryption: To encrypt a message $m \in \mathcal{G}_q$ pick a random $r \leftarrow \mathbb{Z}_q$ and compute

$$u_1 = g_1^r, \quad u_2 = g_2^r, \quad e = h^r m, \quad \alpha = H(u_1, u_2, e), \quad v = c^r d^{r\alpha}$$

The ciphertext is given by (u_1, u_2, e, v).

Decryption: To decrypt a ciphertext (u_1, u_2, e, v) compute $\alpha = H(u_1, u_2, e)$ and verify that $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$. If so, then output $m = e/u_1^{z_1} u_2^{z_2}$.

The attack showing that the scheme fails to provide complete non-malleability now proceeds as follows. Given a public key $(\mathcal{G}, g_1, g_2, H, c, d, h)$ and a ciphertext (u_1, u_2, e, v) first recompute $\alpha = H(u_1, u_2, e)$. With overwhelming probability $\alpha \neq 0 \mod q$ and we can invert α in $\mathbb{Z}_q^*$; else, if a random ciphertext maps to 0 with noticeable probability, collisions for H could be found easily. Next compute

$$u_1^* = u_1^2, \quad u_2^* = u_2^2, \quad e^* = e^2, \quad \alpha^* = H(u_1^*, u_2^*, e^*), \quad v^* = v^{2\alpha^*/\alpha}$$

and finally prepare the public key as

$$c^* = c^{\alpha^*/\alpha}, \qquad d^* = d, \qquad h^* = h.$$

A simple calculation shows that

$$v^* = v^{2\alpha^*/\alpha} = c^{2r\alpha^*/\alpha} d^{2r\alpha\alpha^*/\alpha} = (c^*)^{2r} (d^*)^{2r\alpha^*}$$

Hence, the tuple (u_1^*, u_2^*, e^*, v^*) is a valid ciphertext of $m^* = m^2 \in \mathcal{G}$ under randomness $r^* = 2r \bmod q$ and public key $(\mathcal{G}, g_1, g_2, H, c^*, d^*, h^*)$.[1]

The attack shows that the encryption scheme cannot be used as a non-malleable commitment scheme, as explained in the introduction. With this attack the adversary would be able to open her commitment correctly with $(2r, m^2)$ after seeing the decommitment (m, r) of the original sender. Analogously, if the adversary is given the original secret key $(x_1, x_2, y_1, y_2, z_1, z_2)$ she can modify it to $(x_1\alpha^*/\alpha, x_2\alpha^*/\alpha, y_1, y_2, z_1, z_2)$. We stress that the scheme still satisfies its designated security goal of chosen-ciphertext security.

3 Definitions

In this section we define completely non-malleable public-key encryption and signature schemes. Our approach follows the line of Dolev at el. [7] and also investigates the non-malleability question of an encryption or signature scheme merely with respect to itself. Achieving non-malleability between different schemes is in general impossible, even in the basic case.

3.1 Encryption

An obvious problem with defining completely non-malleable encryption schemes lies in the adversary's possibility to choose her own public key and the uniqueness of ciphertexts. With a fake, yet valid-looking public key the adversary might be able to produce ciphertexts which can be decrypted ambiguously. We consider this to be a characteristic of the encryption scheme, and not an issue of complete non-malleability. Specifically, we allow the adversary to produce such phony keys if the scheme supports it, i.e., if one cannot distinguish good keys from fake ones. We note that, for the application to non-malleable commitments as explained in the introduction, verifying the validity of keys is for example necessary.

Relations. As mentioned in the introduction, regular non-malleability says that it is hard to transform a given ciphertext of message m into one of a related message m^* under the same key. There, related messages are designated according to an efficiently computable (probabilistic) algorithm R which basically takes the messages m and m^* as input.[2] But here we are interested in more general attacks where, as in the examples of non-malleable commitments or key-substitution attacks on signatures, finding a related public key pk^* or ciphertext c^* to the given

[1] At first glance it seems that replacing h by $h^* = h^a$ (or similar substitutions), and leaving the other ciphertext components untouched, would work as well. But then the adversary would encrypt a message $m^* = e/(h^*)^r = mh^{r(1-a)}$. This, however, would be a random message (over the choice of r) and would be thus unlikely to be related to m in a reasonable way.

[2] The definition in [7] lets the relations include another string chosen by the adversary, mainly to deal with the case of symmetric encryption schemes. All our positive and negative results for public-key encryption and signatures remain valid for this extension.

key pk may be considered a success. Hence, we let the relations in general also depend on pk and pk^*, c^*.

Our approach of allowing the relation to depend on other parameters than the messages introduces an interesting issue for non-malleable encryption schemes in the "ordinary" sense. In the original definition of [7] the relation $\mathsf{R}(m, m^*)$ does not range over the user's public key pk. Hence, it remains unclear if it is infeasible to find a ciphertext of a message m^* to a given ciphertext of some unknown m such that m^* is related to m via the public key pk for such schemes. We discuss this in more detail in the final version, presenting an example which is malleable if the relation includes the public key, but which is provably non-malleable if the relations are defined over messages only.

To capture both the original definition of relations over messages only and the more general approach including public keys, we look at classes $\mathcal{R}$ of relations and define complete non-malleability with respect to such classes. The class for the basic definition then spans over relations $\mathsf{R}(pk, m, pk^*, m^*, c^*) = \mathsf{R}_0(m, m^*) \wedge pk = pk^*$, for example.

Message Distributions. We assume that the distribution of the user's message is determined according to some efficiently computable probabilistic algorithm M from some class $\mathcal{M}$. The message distribution M may depend on the given public key. Dolev at al. [7] let the adversary and the simulator determine the message distribution after seeing the public key and having queried the decryption oracle in a preprocessing phase. This can be subsumed in our model by letting these two algorithms output some parameter μ before the ciphertext is created. Unless stated differently all our results, positive and negative ones, remain valid in the setting where the adversary and simulator select such values; yet, we usually do not include them here for sake of simplicity.

Attack Model. In the first stage of the actual attack the adversary $\mathcal{A}$ is given a public key pk and access to a decryption oracle $\text{DEC}(sk, \cdot)$, where $(sk, pk) \leftarrow \text{KGEN}(1^k)$ have been produced by the key generator. The adversary also gets a description of the relation R and the message distribution M. A message m is sampled according to the distribution $\mathsf{M}(pk) \in \mathcal{M}$ and encrypted under pk to ciphertext $c \leftarrow \text{ENC}(pk, m; r)$. The adversary starts the attack on the ciphertext c, the decryption oracle and some information about the message m in form of the value $h \leftarrow \mathsf{hist}(m)$ of an efficiently computable probabilistic function hist. This function can be formally regarded of part of the distribution M. The adversary finally outputs a public key pk^*, possibly for a different yet polynomially related security parameter, and a ciphertext c^*.

Let $\pi_{\text{enc}}(\mathcal{A}, \mathsf{M}, \mathsf{R})$ be the probability that $(pk, c) \neq (pk^*, c^*)$ and that there exists some m^*, r^* such that $c^* = \text{ENC}(pk^*, m^*; r^*)$ and $\mathsf{R}(pk, m, pk^*, m^*, c^*)$ for the relation R. We call this a related-ciphertext attack. Here, as usual for non-malleability definitions, R may implicitly depend on the encryption scheme itself and some security parameter. However, we do not demand that $m \neq m^*$; it suffices to produce a different key/ciphertext pair.

As explained in the introduction, the usage of the encryption scheme as a commitment may result in different attacks and success goals, e.g., the adver-

	$\mathcal{A}$ gets pk, c, oracle $\text{DEC}(sk, \cdot)$ and ...	$\mathcal{S}$ gets pk [and possibly oracle $\text{DEC}(sk, \cdot)$] and...
$\pi_{\text{enc}}^{(\prime)}$	$\mathcal{A}$ outputs pk^*, c^*	$\mathcal{S}$ outputs pk', c', m', r'
$\pi_{\text{open}}^{(\prime)}$	$\mathcal{A}$ outputs pk^*, c^*, then m^*, r^* after m, r	$\mathcal{S}$ outputs pk', c', m', r'
$\pi_{\text{sk-open}}^{(\prime)}$	$\mathcal{A}$ outputs pk^*, c^*, then sk^* after sk	$\mathcal{S}$ outputs pk', c', m', r', sk'

Fig. 1. Overview of Attack and Simulation Modes for Encryption

sary may be obliged to actually open her ciphertext after seeing the opening of the original ciphertext. Therefore, let $\pi_{\text{open}}(\mathcal{A}, \mathsf{M}, \mathsf{R})$ denote the probability that $\mathcal{A}$ after the first stage, on input α^* and values m, r, also returns m^*, r^* such that $c^* = \text{ENC}(pk^*, m^*; r^*)$ and $\mathsf{R}(pk, m, pk^*, m^*, c^*) = 1$. This is called a related-opening attack. Write $\pi_{\text{sk-open}}(\mathcal{A}, \mathsf{M}, \mathsf{R})$ for the probability that $\mathcal{A}$ for input α^* and the secret key sk returns sk^* such that $\text{DEC}(sk^*, c^*) = m^*$ and $\mathsf{R}(pk, m, pk^*, m^*, c^*) = 1$ in a so-called related-key-opening attack. The three cases are described informally in the middle column in Figure 1.

Simulation Model. To capture the idea of the user's ciphertext not helping to produce a ciphertext of a related message we define a simulator $\mathcal{S}$ which is supposed to be as successful as the adversary but without seeing the ciphertext. $\mathcal{S}$ gets as input a public key pk and descriptions of the relation and the message distribution, but does not get access to a decryption oracle. Then, a message m is sampled according to $\mathsf{M}(pk)$ and algorithm $\mathcal{S}$ receives $h \leftarrow \mathsf{hist}(m)$ as input.

Depending on the adversary's attack mode, the simulator's task becomes increasingly challenging such that a successful simulator for a security level automatically constitutes a simulator for a lower level. Precisely, the simulator is supposed to output a key pk', a ciphertext c', a message m' and randomness r' (if the adversary runs a related-ciphertext or a related-opening attack),[3] and a key pk', a ciphertext c', a message m', a random string r' and a secret key sk' (if the adversary runs a related-key-opening attack). Again, see Figure 1 for an overview.

Concerning the auxiliary power of the simulator there are two possibilities. One version is to give the simulator, like the adversary, additional access to the decryption oracle. We call this an *assisted* simulator. This reflects the approach that the simulator should have comparable power as the adversary. The other possibility is to deny the simulator access to DEC. We call such simulators *standalone* simulators. This approach follows the definition of [7].

Although the definition with assisted simulators appears to be more intuitive at first, it is not clear that giving the simulator access to DEC captures the "right flavor" of complete non-malleability. The additional power may for

[3] For some of our negative results we use a milder requirement and let the simulator only output pk', c'. This even strengthens these results.

example allow to prove schemes to be secure which are completely malleable in a natural sense. While this question has somewhat been settled for chosen-ciphertext security, where this additional power is acceptable, our separation of complete non-malleability from chosen-ciphertext security means that these arguments cannot be transfered without precautions. Instead, a conservative approach for designing schemes is therefore to rely on stand-alone simulators, as it suffices for our solutions in the random oracle model for example. We note that our impossibility results hold for both cases, although in a slightly weaker sense for assisted simulators.

Let both $\pi'_{\text{enc}}(\mathcal{S}, \mathsf{M}, \mathsf{R})$ and $\pi'_{\text{open}}(\mathcal{S}, \mathsf{M}, \mathsf{R})$ denote the probability that $c' = \text{ENC}(pk', m'; r')$ and that $\mathsf{R}(pk, m, pk', m', c') = 1$ in the first and second simulation experiment, respectively. Similarly, $\pi'_{\text{sk-open}}(\mathcal{S}, \mathsf{M}, \mathsf{R})$ stands for the probability that $c' = \text{ENC}(pk', m'; r')$, $m' = \text{DEC}(sk', c')$ and $\mathsf{R}(pk, m, pk', m', c') = 1$ in the third simulation experiment.

Definition 1. *A public-key encryption scheme is completely non-malleable (for stand-alone or assisted simulator) with respect to kind* $\in \{enc, open, sk\text{-}open\}$*, distribution class* $\mathcal{M}$ *and relation class* $\mathcal{R}$*, if for any adversary* $\mathcal{A}$ *there exists a (stand-alone or assisted) simulator* $\mathcal{S}$ *such that for any distribution* $\mathsf{M} \in \mathcal{M}$ *and any relation* $\mathsf{R} \in \mathcal{R}$ *the absolute difference* $|\pi_{kind}(\mathcal{A}, \mathsf{M}, \mathsf{R}) - \pi'_{kind}(\mathcal{S}, \mathsf{M}, \mathsf{R})|$ *is negligible.*

In the sequel, when speaking of completely non-malleable encryption schemes we refer to related-ciphertext attacks and $\pi_{\text{enc}}(\mathcal{A}, \mathsf{M}, \mathsf{R})$, $\pi'_{\text{enc}}(\mathcal{S}, \mathsf{M}, \mathsf{R})$. The definitions for completely non-malleable encryption (and signatures in the next section) can be extended in a straightforward way to the random oracle model.

3.2 Signatures

The attack scenario for completely non-malleable signature schemes resembles the setting of adaptive chosen-message attacks known from regular signature schemes.

Discussion. Defining the attack model for completely non-malleable signature schemed as outlined above, it seems that the adversary can always generate a new signature under a new public key, i.e., the adversary can naturally generate a new key pair and sign some message with the self-generated secret key. As explained, this attack can be confined as in the example of unknown-key attacks [3] where the adversary is supposed to find a matching key pair for a given message and a given signature. Here we do not restrict the adversary's goal in such a way. First, we do not want to give up generality and exclude certain application scenarios, e.g., signatures encrypted together with the message under a malleable encryption scheme, where the message is not known but the signature may still be transformable by permeating the malleable ciphertext. Second, if the adversary can trivially output a signature, i.e., without relying on the original signature, then this does not violate the idea of (complete) non-malleability and we should therefore be able to prove this formally as well.

	$\mathcal{A}$ gets vk, oracle $\text{SIG}(sk, \cdot)$ and ...	$\mathcal{S}$ gets vk and ...
$\pi_{\text{sig}}^{(\prime)}$	$\mathcal{A}$ outputs vk^*, m^*, s^*	$\mathcal{S}$ outputs vk', m', s'

Fig. 2. Overview of Attack and Simulation Mode for Signatures

Attack and Simulation Model. At the outset of the complete non-malleability attack the adversary $\mathcal{A}$ gets as input the description of the relation R and a verification key vk, generated together with the secret signing key sk by $\text{KGEN}(1^k)$. The adversary is then allowed to query a signature oracle $\text{SIG}(sk, \cdot)$ about messages of her choice. For definitional reasons we let the signature oracle prepend the verification key vk and the message m to each signature reply s for such a query. The adversary finally outputs some verification key vk^*, a message m^* and some signature s^*. Define $\pi_{\text{sig}}(\mathcal{A}, \mathsf{R})$ as the probability that s^* is a valid signature for m^* under vk^*, i.e., $\text{VF}(vk^*, m^*, s^*) = 1$, that (vk^*, m^*, s^*) is different from any previously given answer (vk, m, s) of the signature oracle, and that $\mathsf{R}(vk, vk^*, m^*, s^*)$ holds for relation R from the class $\mathcal{R}$.

The simulator only gets vk and the relation as input and is supposed to output a triple (vk', m', s') without having oracle access to $\text{SIG}(sk, \cdot)$. Let $\pi'_{\text{sig}}(\mathcal{S}, \mathsf{R})$ be the probability that s' is a valid signature for m' under vk' and that $\mathsf{R}(vk, vk', m', s')$ is satisfied. The attack and simulation model is outlined in Figure 2.

Similar to the encryption case one could also distinguish between stand-alone simulators (as defined here) and assisted simulators (which additionally get access to the signature oracle). In the latter case one would have to unorthodoxly extend the model to allow the adversary to ask for a "challenge signature" which the simulator is denied. We do not follow this approach here as our negative results would hold for this case as well, and our constructions in the random oracle already work for stand-alone simulators.

Security Definition. The idea is now to say that for any adversary there is a simulator such that the success probabilities differ only insignificantly. But with this definition a signature scheme could be completely non-malleable and yet be insecure in the sense of unforgeability, e.g., if it is easy to derive the secret key from the verification key. Therefore, we also throw in the mild assumption that the signature scheme must be unforgeable under key-only attacks, i.e., it must be infeasible on input vk (but no signature oracle) to find some message together with a valid signature under vk.

Definition 2. *A signature scheme is completely non-malleable for relation class $\mathcal{R}$ if it is existentially unforgeable under key-only attacks and if for any adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$ such that for any relation $\mathsf{R} \in \mathcal{R}$ the absolute difference $|\pi_{sig}(\mathcal{A}, \mathsf{R}) - \pi'_{sig}(\mathcal{S}, \mathsf{R})|$ is negligible.*

We briefly discuss some consequences of the definition, showing that the definition is powerful to reflect the notions of strong unforgeability (i.e., where the adversary is also considered victorious if she finds a new signature under the

original verification key to a message previously signed by the signature oracle) or key-substitution attacks (where the adversary tries to find another key vk^* to a valid triple vk, m, s), both under adaptive chosen-message attacks. For this, let $\mathsf{R}_{\text{str-unf}}(vk, vk^*, m^*, s^*)$ be the relation such that $\mathsf{R}_{\text{str-unf}}(vk, vk^*, m^*, s^*) = 1$ iff $vk = vk^*$; let $\mathsf{R}_{\text{key-sub}}$ be the relation such that $\mathsf{R}_{\text{key-sub}}(vk, vk^*, m^*, s^*) = 1$ iff $\text{VF}(vk, m^*, s^*) = 1$. The proof is omitted.

Proposition 1. *Let* (KGEN, SIG, VF) *be a signature scheme which is completely non-malleable with respect to* $\mathcal{R} \ni \mathsf{R}_{str\text{-}unf}$ *resp.* $\mathcal{R} \ni \mathsf{R}_{key\text{-}sub}$*. Then the scheme is strongly unforgeable under adaptive chosen-message attacks resp. secure against key-substitution attacks.*

4 Summary of Results

In this section we summarize our (positive and negative) results. For better comprehensibility the results are stated in an informal way. The formal results and technical details can be found in the the full version.

Regular Non-Malleability and Relations over Messages Only. We show that extending the relations in the definition of [7] for regular non-malleability, i.e., where the adversary does not tamper the public key, to include the given public key pk (in addition to the messages m, m^*) can be fatal to security:

Theorem 1 (informal). *There is an encryption scheme which is non-malleable with respect to* $\mathcal{R}_{msg} = \{R(m, m^*)\}$ *but which is malleable with respect to some relation* $\mathsf{R}_{pk}(pk, m, m^*)$*.*

Hardness of Constructions for General Relations. Here we discuss our negative results for constructions of completely non-malleable schemes where, in contrast to the previous case, the adversary is allowed to output another key pk^*. We show that there are relations for which completely non-malleable schemes are hard to construct. Although we prove this result for a specific set of "bad" relations, we note that the implication carries over to any class where such relations can be "somehow embedded" in relations of the class.

Theorem 2 (informal). *Public-key encryption schemes which are completely non-malleable according to black-box stand-alone simulators and general relations, do not exist.*

Note that the previous theorem assumes that the simulator is stand-alone. For assisted simulators, which are granted access to DEC, we can show the same result for relations which are efficiently computable relative to an oracle. We note that the black-box simulator does not have access to this oracle directly, but only through the relation. This corresponds to the case that the simulator can efficiently compute the relation (via black-box access) but is denied the description of the relation.

Theorem 3 (informal). *Public-key encryption schemes which are completely non-malleable according to black-box assisted simulators and general relations (relative to an oracle), do not exist.*

The results about encryption easily transfers to signatures:

Proposition 2 (informal). *Signature schemes which are completely non-malleable according to black-box simulations for general relations, do not exist.*

Yet, for signatures we can show that completely non-malleable systems for general relations are impossible at all, even when allowing non-black-box constructions or if the simulator depends on the relation.

Theorem 4 (informal). *There do not exist completely non-malleable signature schemes with respect to general relations.*

Constructions in the Random Oracle Model. On the positive side, solutions in the random oracle for completely non-malleable schemes exist. And while OAEP encryption [2] and Fiat-Shamir signatures [11] provably do not have this property, slight variations of these schemes work. The basic idea to simply include the public encryption or signature key, respectively, to each hash function evaluation. We append the term "with public-key hashing" to such modified schemes:

Proposition 3 (informal). *RSA-OAEP with public-key hashing is completely non-malleable with respect to stand-alone simulators and any relations, in the random oracle model.*

A similar result holds for Fiat-Shamir signatures:

Proposition 4 (informal). *Fiat-Shamir signatures with public-key hashing are completely non-malleable with respect to general relations (except for essentially those relations, for which the unconditional impossibility results of Theorem 4 holds), in the random oracle model.*

In both cases the proofs rely on the original results [2, 10, 14] about the security against regular chosen-ciphertext attacks and chosen-message attacks.

Acknowledgments

We would like to thank Yevgeniy Dodis, Alejandro Hevia, Bogdan Warinschi and the reviewers for helpful input.

References

1. Boaz Barak. *Constant-Round Coin-Tossing With a Man in the Middle or Realizing the Shared Random String Model.* Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2002. IEEE Computer Society Press, 2002.
2. Mihir Bellare and Phillip Rogaway. *Optimal Asymmetric Encryption — How to Encrypt with RSA.* Advances in Cryptology — Eurocrypt'94, Volume 950 of Lecture Notes in Computer Science, pages 92–111. Springer-Verlag, 1995.
3. Simon Blake-Wilson and Alfred Menezes. *Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol.* Public-Key Cryptography (PKC)'99, Volume 1560 of Lecture Notes in Computer Science, pages 154–170. Springer-Verlag, 1999.

4. G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. *Non-interactive and Non-Malleable Commitment.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1998, pages 141–150. ACM Press, 1998.
5. G. Di Crescenzo, J. Katz, R. Ostrovsky, and A. Smith. *Efficient And Non-Interactive Non-Malleable Commitment.* Advances in Cryptology — Eurocrypt 2001, Volume 2045 of Lecture Notes in Computer Science, pages 40–59. Springer-Verlag, 2001.
6. Ronald Cramer and Victor Shoup. *A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attacks.* Advances in Cryptology — Crypto'98, Volume 1462 of Lecture Notes in Computer Science, pages 13–25. Springer-Verlag, 1998.
7. D. Dolev, C. Dwork, and M. Naor. *Non-Malleable Cryptography. SIAM Journal on Computing*, 30(2):391–437, 2000.
8. Marc Fischlin and Roger Fischlin. *Efficient Non-Malleable Commitment Schemes.* Advances in Cryptology — Crypto 2000, Volume 1880 of Lecture Notes in Computer Science, pages 414–432. Springer-Verlag, 2000.
9. Marc Fischlin and Roger Fischlin. *The Representation Problem Based on Factoring.* Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2002, Volume 2271 of Lecture Notes in Computer Science, pages 96–113. Springer-Verlag, 2002.
10. E. Fujisaki, T. Okamoto, David Pointcheval, and Jacques Stern. *RSA-OAEP is Secure Under the RSA Assumption.* Advances in Cryptology — Crypto 2001, Volume 2139 of Lecture Notes in Computer Science. Springer-Verlag, 2001.
11. A. Fiat and A. Shamir. *How to Prove Yourself: Practical Solutions to Identification and Signature Schemes.* Advances in Cryptology — Crypto'86, Volume 263 of Lecture Notes in Computer Science, pages 186–194. Springer-Verlag, 1986.
12. Burton Kaliski. *On Hash Function Firewalls in Signature Schemes.* Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2002, Volume 2271 of Lecture Notes in Computer Science, pages 1–16. Springer-Verlag, 2002.
13. Alfred Menezes and Nigel Smart. *Security of Signature Schemes in a Multi-User Setting.* Designs, Codes and Cryptography, Volume 33, pages 261–274. Springer-Verlag, 2004.
14. David Pointcheval and Jacques Stern. *Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology*, 13(3):361–396, 2000.