On the Joint Security of Signature and Encryption Schemes under Randomness Reuse: Efficiency and Security Amplification

Afonso Arriaga[1], Manuel Barbosa[1], and Pooya Farshim[2]

[1] HASLab/INESC TEC, Universidade do Minho, Braga, Portugal
[2] Department of Computer Science, Darmstadt University of Technology, Germany
{arriaga,mbb}@di.uminho.pt, farshim@cased.de

Abstract. We extend the work of Bellare, Boldyreva and Staddon on the systematic analysis of randomness reuse to construct multi-recipient encryption schemes to the case where randomness is reused across different cryptographic primitives. We find that through the additional binding introduced through randomness reuse, one can actually obtain a *security amplification* with respect to the standard black-box compositions, and achieve a stronger level of security. We introduce stronger notions of security for encryption and signatures, where challenge messages can depend in a restricted way on the random coins used in encryption, and show that two variants of the KEM/DEM paradigm give rise to encryption schemes that meet this enhanced notion of security. We obtain the most efficient signcryption scheme to date that is secure against insider attackers without random oracles.

Keywords: Signcryption, Insider Security, Randomness Reuse.

1 Introduction

Signcryption is a cryptographic primitive that aims to simultaneously provide the guarantees of public-key encryption and signature schemes [17], i.e., confidentiality, integrity, authentication and possibly non-repudiation, whilst offering efficiency gains. One trivial way to obtain the signcryption functionality—if one is not interested in saving computational power or bandwidth—is to use a black-box combination of the two primitives. This approach was systematically studied by An, Dodis and Rabin [2], by looking at Encrypt-then-Sign (EtS), Sign-then-Encrypt (StE) and Encrypt-and-Sign (EaS) compositions. The former two constructions are natural sequential compositions of the two primitives, whereas EaS is a parallel composition using a commitment scheme to enforce the necessary binding. A well-known, albeit surprising, result in [2] is that the interaction between the signature and encryption primitives can work *against* the security of the composition, making it impossible to achieve the strongest levels of security, even when the underlying encryption and signature schemes are themselves strongly secure. For example, in an StE construction an attacker knowing the secret key of the receiver is always able to forge valid signcryptions simply by decrypting and re-encrypting the contents of a legitimately generated ciphertext, regardless of the security guarantees provided by the underlying signature scheme: this translates into a trivial break of unforgeability against insider attackers in the signcryption setting.

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 206–223, 2012.

One general approach to obtaining efficiency gains in cryptography is to reuse randomness across instantiations of various cryptographic algorithms. This technique can allow for significant savings in processing load and bandwidth, as partial results (and even ciphertext elements) can be shared between multiple instances of cryptographic algorithms. For this reason, randomness reuse is frequently used in the context of batch encryption operations where (possibly different) messages are encrypted to multiple recipients, as recognized by Kurosawa [11] in the construction of multi-recipient encryption schemes. Furthermore, randomness reuse is also used as an optimization technique, in an ad-hoc way, in the construction of signcryption schemes [17,16]. Nevertheless, this avenue must be pursued with caution, since randomness reuse may, of course, hinder the security of the resulting cryptographic schemes.

Bellare et al. [3], building on the work of Kurosawa [11], systematically study the problem of reusing randomness in multi-recipient encryption. The authors consider the particular case of constructing such schemes by running multiple instances of a public-key encryption (PKE) scheme, whilst sharing randomness across them. An interesting result in this work is a general method for identifying PKE schemes that are secure when used in this scenario. Schemes which satisfy the so-called *reproducibility test* permit establishing the security for multiple recipients with randomness reuse through a variant of the hybrid argument.

OUR CONTRIBUTIONS. In this paper we extend the work of Bellare et al. [3] to the case where randomness is reused across *different* cryptographic primitives, and analyze the security of signcryption schemes constructed by composing encryption and signature schemes under randomness reuse. More in detail, our contributions are the following:

- We define a compatibility notion that establishes classes of signature and encryption schemes that can be composed under randomness reuse to obtain correct signcryption schemes. We then identify security properties that are sufficient for the EtS and StE compositions with randomness reuse to result in secure signcryption schemes. In particular, we introduce the notion of *randomness-dependent security* for both signatures and encryption schemes. Intuitively, security must be preserved when the messages chosen by attackers are allowed to depend (in a restricted way) on the implicit randomness input to the underlying cryptographic algorithms. We believe these security notions may be of independent interest in the study of the role of randomness in cryptographic security and, particularly, in the generic analysis of randomness reuse optimizations for scenarios where multiple (possibly heterogenous) cryptographic operations are carried out in a batch procedure (e.g., optimizing the overall performance of a server continuously carrying out key agreement, signature and encryption operations).
- We find that through the additional binding that is established via the reuse of randomness, it is possible to achieve *full* insider security. Our results hold in the dynamic multi-user setting, although in some cases we require adversaries to register the full key pairs of all users created for the attack. This is usually called the *registered key model* [13] and it captures natural restrictions in many PKI settings. This is a *security amplification* with respect to the equivalent compositions without randomness reuse, in which it is *not* possible to achieve this level of security, even starting from underlying schemes providing the same security guarantees we

require for our results. In other words, our results depend in an essential way on reusing randomness, and it is *not* the case that a standard composition of randomness-dependent secure signature and encryption schemes trivially yields a comparable result. In this respect, our work generalizes independent work in the same direction presented in [13], and that we contextualize in Section 2.

- We identify a set of simple and natural properties of KEMs and DEMs that suffice to ensure that PKE schemes constructed from *both* variants of the KEM/DEM composition paradigm proposed in [9,10] fall within our framework. As a particular case, when the Kurosawa–Desmedt [12] encryption scheme is composed with the Boneh–Boyen signature scheme [6] in the StE construction, we obtain the most efficient signcryption scheme to be proven insider secure in the standard model. One caveat is that our results hold only in the registered key model. In compensation, our scheme offers non-repudiation, inherited from the StE construction, and a combination of computational and communication (bandwidth) efficiency that outperforms previous solutions.

STRUCTURE OF THE PAPER. In the next section we review the related work in more detail (a more extensive discussion can be found in the full version of the paper). Then, in Section 3 we settle notation by introducing the standard syntax, correctness and security definitions for signature, encryption and signcryption schemes. In Section 4 we describe the properties that are sufficient for the EtS and StE compositions to yield secure signcryption schemes under randomness reuse, and prove the corresponding composition theorems. Finally, in Section 5 we describe the potential instantiations of our framework and compare our scheme with existing results.

2 Related Work

Matsuda et al. [13] and Chiba et al. [8] systematically study the construction of signcryption schemes using compositions of standard cryptographic primitives, aiming to obtain levels of efficiency and security that are comparable to the best concrete schemes in the literature via generic constructions. Independently of our work, Matsuda et al. [13] show how to perform compositions of tag-based KEMs and signature schemes to obtain efficiency gains in an StE-like construction via randomness reuse. They also describe a series of schemes that can be used to instantiate these constructions. The resulting compositions are efficient and achieve full insider security, with the caveat that strong unforgeability can only be proven in a slightly weaker model, where the adversary must register the secret keys for the public keys it chooses to query to the signcryption oracle. Our results have the same limitation.

The main differences between our work and the approach in [13] are the following. Our results are more general in that they consider the composition of encryption schemes and signature schemes under randomness reuse, rather than lower level primitives. On one hand, this sets our results as natural extensions to the work by An et al. [2] on signature and encryption compositions, and also of the work of Bellare et al. [3], allowing us to establish a connection between the two results. On the other hand, our results capture the ones included in [13] on randomness reuse for the construction of signcryption schemes as particular cases, and cover a broader class of constructions. More precisely, our compatibility framework and security results apply to general encryption schemes, rather than those specifically constructed from tag-based KEMs. This allows us to capture not only schemes constructed using a specific flavor of tag-KEMs [13], but also encryption schemes constructed from other known variants of the KEM/DEM paradigm [9,10], and even schemes that do not follow this paradigm.

Chiba et al. [8] propose the first fully secure signcryption schemes in the standard model by using a variant of the StE construction that relies on a chosen-ciphertext-secure tag-based KEM, a chosen-ciphertext-secure DEM that has a "one-to-one" property, and a strongly unforgeable signature scheme. Such schemes are less efficient than the one we propose, but are proven secure without the key registration requirement.

3 Preliminaries

NOTATION. We write $a \leftarrow b$ to denote the algorithmic action of assigning the value of b to the variable a. We use $\perp \notin \{0,1\}^\star$ to denote special failure symbol. If S is a set, we write $a \leftarrow_\$ \mathsf{S}$ for sampling a from S uniformly at random. If $\mathcal{A}$ is a probabilistic algorithm we write $a \leftarrow_\$ \mathcal{A}(i_1, i_2, \ldots, i_n)$ for the action of running $\mathcal{A}$ on inputs $i_1, i_2, \ldots, i_n$ with random coins, and assigning the result to a. Sometimes we run $\mathcal{A}$ on specific coins r and write $a \leftarrow \mathcal{A}(i_1, i_2, \ldots, i_n; \mathsf{r})$.

GAMES. In this paper we use the code-based game-playing language [4]. Each game has an **Initialize** and a **Finalize** procedure. It also has specifications of procedures to respond to an adversary's various queries. A game is run with an adversary $\mathcal{A}$ as follows. First **Initialize** runs and its outputs are passed to $\mathcal{A}$. Then $\mathcal{A}$ runs and its oracle queries are answered by the procedures of the game. When $\mathcal{A}$ terminates, its output is passed to **Finalize** which returns the outcome of the game. In each game, we restrict attention to legitimate adversaries, which is defined specifically for each game. We use lists as data structures to keep relevant state in the games. The empty list is represented by square brackets [ ]. We denote by $\mathsf{List} \leftarrow a : \mathsf{List}$ the action of appending element a to the head of a list List.

PUBLIC-KEY ENCRYPTION. A public-key encryption scheme $\mathcal{E} = (\mathsf{EGen}, \mathsf{Enc}, \mathsf{Dec})$ is specified by three polynomial-time algorithms (in the length of their inputs) associated with a message space $\mathcal{M}$ and a randomness space $\mathcal{R}$.

- $\mathsf{EGen}(1^\lambda)$ is the probabilistic key-generation algorithm, taking as input the security parameter and returning a secret key sk and a public key pk.
- $\mathsf{Enc}(\mathsf{m}, \mathsf{pk}; \mathsf{r})$ is the probabilistic encryption algorithm. On input a message $\mathsf{m} \in \mathcal{M}$, a public key pk, and possibly some random coins $\mathsf{r} \in \mathcal{R}$, this algorithm outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{c}, \mathsf{sk})$ is the deterministic decryption algorithm. On input of a ciphertext c and a key sk, this algorithm outputs a message m or failure symbol $\perp$.

The correctness of a public-key encryption scheme requires that for any $\lambda \in \mathbb{N}$, any $(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{EGen}(1^\lambda)$, any $\mathsf{m} \in \mathcal{M}$, and any random coins $\mathsf{r} \in \mathcal{R}$, we have that $\mathsf{Dec}(\mathsf{Enc}(\mathsf{m}, \mathsf{pk}; \mathsf{r}), \mathsf{sk}) = \mathsf{m}$.

The standard notion of security for a public-key encryption scheme considered here is indistinguishability under chosen-ciphertext attacks ($\mathsf{IND\text{-}CCA}$). We refer the interested reader to the full version of the paper for a formal definition.

DIGITAL SIGNATURE. A signature scheme $\mathcal{S} = (\mathsf{SGen}, \mathsf{Sign}, \mathsf{Verify})$ is specified by three polynomial-time algorithms with a randomness space $\mathcal{R}$ and a message space $\mathcal{M}$.

- $\mathsf{SGen}(1^\lambda)$ is the probabilistic key-generation algorithm which takes as input the security parameter and returns a secret key sk and a public key pk.
- $\mathsf{Sign}(\mathsf{m}, \mathsf{sk}; \mathsf{r})$ is the probabilistic signature generation algorithm. On input a message m, a secret key sk, and possibly some random coins $\mathsf{r} \in \mathcal{R}$, this algorithm outputs a signature σ.
- $\mathsf{Verify}(\mathsf{m}, \sigma, \mathsf{pk})$ is the deterministic signature verification algorithm. On input of a signature σ, a message m and a public key pk, this algorithm outputs a boolean value T or F.

The correctness of a signature scheme requires that for any $\lambda \in \mathbb{N}$, any $\mathsf{m} \in \{0,1\}^\star$, any $(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{SGen}(1^\lambda)$, and any $\mathsf{r} \in \mathcal{R}$, we have that $\mathsf{Verify}(\mathsf{Sign}(\mathsf{m}, \mathsf{sk}; \mathsf{r}), \mathsf{m}, \mathsf{pk}) = \mathsf{T}$.

The standard notion of security for a digital signature scheme considered in this paper is strong existential unforgeability under chosen-message attacks (sUF-CMA). We refer the interested reader to the full version of the paper for a formal definition.

SIGNCRYPTION. A signcryption scheme $\mathcal{SC} = (\mathsf{Gen}, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ is specified by three polynomial-time algorithms associated with a message space $\mathcal{M}$ and a randomness space $\mathcal{R}$.

- $\mathsf{Gen}(1^\lambda)$ is the probabilistic key-generation algorithm which takes as input the security parameter and returns a secret key sk and a matching public key pk. Unless one wishes to signcrypt a message to oneself, two key pairs are required to signcrypt and unsigncrypt.
- $\mathsf{Signcrypt}(\mathsf{m}, \mathsf{sk}_S, \mathsf{pk}_R; \mathsf{r})$ is the probabilistic signcryption algorithm. On input a message $\mathsf{m} \in \mathcal{M}$, the sender's secret key sk_S, the receiver's public key pk_R, and possibly some random coins $\mathsf{r} \in \mathcal{R}$, this algorithm outputs a signcryption c.
- $\mathsf{Unsigncrypt}(\mathsf{c}, \mathsf{pk}_S, \mathsf{sk}_R)$ is the deterministic unsigncryption algorithm. On input a signcryption c, the sender's public key pk_S, and the receiver's secret key sk_R, this algorithm outputs a message m or failure symbol $\perp$.

The correctness of a signcryption scheme requires that for any $\mathsf{m} \in \mathcal{M}$, any $\lambda \in \mathbb{N}$, any $(\mathsf{sk}_S, \mathsf{pk}_S) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$, any $(\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$, and any random coins $\mathsf{r} \in \mathcal{R}$, we have $\mathsf{Unsigncrypt}(\mathsf{Signcrypt}(\mathsf{m}, \mathsf{sk}_S, \mathsf{pk}_R; \mathsf{r}), \mathsf{pk}_S, \mathsf{sk}_R) = \mathsf{m}$. We consider here the strong notion of confidentiality, introduced by [16], in which the adversary is allowed to choose without restrictions pk_S to query to the $\mathbf{Unsigncrypt}$ oracle. The adversary may also choose the challenge key pair $(\mathsf{sk}_S, \mathsf{pk}_S)$, but the key pair is required to be valid. Analogously to IND-CCA for encryption, $\mathbf{LoR}$ oracle can only be called once. We refer to this model as dynamic multi-user indistinguishability against insider chosen-ciphertext attacks (IND-iCCA).

Definition 1. *A signcryption scheme is* IND-iCCA *secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{IND\text{-}iCCA}}_{\mathcal{SC},\mathcal{A}}(\lambda) := 2 \cdot \Pr[\mathsf{IND\text{-}iCCA}_{\mathcal{SC},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}] - 1\,,$$

where game $\mathsf{IND\text{-}iCCA}_{\mathcal{SC},\mathcal{A}}$ *described in Figure 1.*

```
procedure Initialize($1^\lambda$):
$(\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$\mathsf{List} \leftarrow [\,]$
Return $(\mathsf{pk}_R)$

procedure Finalize($b'$):
Return $(b = b')$

procedure LoR($\mathsf{m}_0, \mathsf{m}_1, (\mathsf{sk}_S, \mathsf{pk}_S)$):
$\mathsf{c} \leftarrow_\$ \mathsf{Signcrypt}(\mathsf{m}_b, \mathsf{sk}_S, \mathsf{pk}_R)$
$\mathsf{List} \leftarrow (\mathsf{c}, \mathsf{pk}_S) : \mathsf{List}$
Return c

procedure Unsigncrypt($\mathsf{c}, \mathsf{pk}_S$):
$\mathsf{m} \leftarrow \mathsf{Unsigncrypt}(\mathsf{c}, \mathsf{pk}_S, \mathsf{sk}_R)$
Return m
```

Fig. 1. Game IND-iCCA for a signcryption $\mathcal{SC} = (\mathsf{Gen}, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$. An adversary $\mathcal{A}$ is legitimate if: 1) it calls **LoR** once, with $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ and $|\mathsf{m}_0| = |\mathsf{m}_1|$, and a valid key pair $(\mathsf{sk}_S, \mathsf{pk}_S)$; and 2) it does not query **Unsigncrypt** with $(\mathsf{c}, \mathsf{pk}_S) \in \mathsf{List}$.

We also define dynamic multi-user strong existential unforgeability against insider chosen message attacks for authenticity, but in a slightly weaker model that obliges the adversary to register a key pair $(\mathsf{sk}_R, \mathsf{pk}_R)$ before querying the **Signcrypt** oracle or **Finalize** with pk_R. For this purpose, a **Key-Reg** oracle is also available. This model is called sUF-iCMA, for short.

Definition 2. *A signcryption scheme is* sUF-iCMA *secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{sUF\text{-}iCMA}}_{\mathcal{SC},\mathcal{A}}(\lambda) := \Pr[\mathsf{sUF\text{-}iCMA}_{\mathcal{SC},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}]\,,$$

where game $\mathsf{sUF\text{-}iCMA}_{\mathcal{SC},\mathcal{A}}$ *described in Figure 2.*

```
procedure Initialize($1^\lambda$):
$(\mathsf{sk}_S, \mathsf{pk}_S) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$
$\mathsf{List} \leftarrow [\,]$
$\mathsf{List}' \leftarrow [\,]$
Return $\mathsf{pk}_S$

procedure Signcrypt($\mathsf{m}, \mathsf{pk}_R$):
If $(\star, \mathsf{pk}_R) \in \mathsf{List}'$
    $\mathsf{c} \leftarrow_\$ \mathsf{Signcrypt}(\mathsf{m}, \mathsf{sk}_S, \mathsf{pk}_R)$
    $\mathsf{List} \leftarrow (\mathsf{c}, \mathsf{pk}_R) : \mathsf{List}$
    Return c
Else Return $\perp$

procedure Key-Reg($\mathsf{sk}, \mathsf{pk}$):
If $\mathsf{isValid}(\mathsf{sk}, \mathsf{pk})$
    $\mathsf{List}' \leftarrow (\mathsf{sk}, \mathsf{pk}) : \mathsf{List}'$
    Return T
Else Return F

procedure Finalize($\mathsf{c}, \mathsf{pk}_R$):
If $(\mathsf{c}, \mathsf{pk}_R) \in \mathsf{List}$ Return F
If $(\mathsf{sk}_R, \mathsf{pk}_R) \in \mathsf{List}'$
    $\mathsf{m} \leftarrow \mathsf{Unsigncrypt}(\mathsf{c}, \mathsf{pk}_S, \mathsf{sk}_R)$
    If $\mathsf{m} \neq \perp$ Return T
Return F
```

Fig. 2. Game sUF-iCMA for a signcryption $\mathcal{SC} = (\mathsf{Gen}, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$

Remark. We assume one can confirm the validity of a key pair using an efficient algorithm isValid, which is usually the case for practical schemes. Under this assumption, one could omit the key pair validity restriction in the adversary legitimacy definition in the IND-iCCA security model, and require the signcryption algorithm to internally check for sender key pair validity. This does not apply to the key registration oracle in the unforgeability game, as this is conditioning the adversary to provide valid key pairs for the *receivers* (note that this check cannot be done internally by the signcryption algorithm). However, one could remove the validity check restriction in **Finalize**, and require that the unsigncryption algorithm does check for receiver key pair validity.

4 Compositions with Randomness Reuse

In this section we look at black-box compositions of signature and encryption under randomness reuse. We describe properties that are sufficient for the encrypt-then-sign and sign-then-encrypt constructions with shared randomness to yield secure signcryption schemes, and prove the corresponding composition theorems. Our proposed framework gives rise to signcryption schemes that attain full insider security in dynamic multi-user models. We defer a discussion on instantiability to Section 5.

4.1 Composition-Enabling Properties

PARTITIONED SCHEMES, COMPATIBILITY, AND CONDITIONAL INJECTIVITY. The notion of joint signature and encryption in the public-key setting with randomness reuse implies that the signature and encryption algorithms share the same randomness space. In order to clarify the concept and simplify the security proofs, we will restrict our attention to *partitioned* schemes [7]. Furthermore, to enable composition under randomness reuse, we also require the signature and encryption schemes to be *compatible*. We formalize these notions next.

Definition 3 (Partitioned schemes). *We say a signature scheme is* partitioned*, if its signature space is composed of pairs* (σ, R)*, where the signature generation algorithm calculates* R *independently of the input message and keys. More precisely, we require that experiment* $\mathbf{Indep}_{\mathcal{S}}$ *in Figure 3 returns* T *with probability* 1 *for all messages* m_0 *and* m_1 *in the appropriate space. Similarly, an encryption scheme is* partitioned*, if its ciphertext space is composed of pairs* (c, R) *and experiment* $\mathbf{Indep}_{\mathcal{E}}$ *in Figure 3 returns* T *with probability* 1 *for all messages* m_0 *and* m_1 *in the appropriate space.*

Definition 4 (Compatibility). *A signature scheme* $\mathcal{S}$ *and an encryption scheme* $\mathcal{E}$ *are compatible if they are partitioned, share the same random space* $\mathcal{R}$*, and the experiment* $\mathbf{Compatibility}$ *in Figure 3 returns* T *with probability* 1 *for any messages* m_0 *and* m_1 *in the appropriate spaces.*

```
test $\mathbf{Indep}_{\mathcal{S}}(\mathsf{m}_0, \mathsf{m}_1)$:
$(\mathsf{sk}_0, \mathsf{pk}_0) \leftarrow_{\$} \mathsf{SGen}(1^\lambda)$
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow_{\$} \mathsf{SGen}(1^\lambda)$
$\mathsf{r} \leftarrow_{\$} \mathcal{R}$
$(\sigma_0, \mathsf{R}_0) \leftarrow \mathsf{Sign}(\mathsf{m}_0, \mathsf{sk}_0; \mathsf{r})$
$(\sigma_1, \mathsf{R}_1) \leftarrow \mathsf{Sign}(\mathsf{m}_1, \mathsf{sk}_1; \mathsf{r})$
Return $(\mathsf{R}_0 = \mathsf{R}_1)$
```

```
test $\mathbf{Indep}_{\mathcal{E}}(\mathsf{m}_0, \mathsf{m}_1)$:
$(\mathsf{sk}_0, \mathsf{pk}_0) \leftarrow_{\$} \mathsf{EGen}(1^\lambda)$
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow_{\$} \mathsf{EGen}(1^\lambda)$
$\mathsf{r} \leftarrow_{\$} \mathcal{R}$
$(\mathsf{c}_0, \mathsf{R}_0) \leftarrow \mathsf{Enc}(\mathsf{m}_0, \mathsf{pk}_0; \mathsf{r})$
$(\mathsf{c}_1, \mathsf{R}_1) \leftarrow \mathsf{Enc}(\mathsf{m}_1, \mathsf{pk}_1; \mathsf{r})$
Return $(\mathsf{R}_0 = \mathsf{R}_1)$
```

```
test $\mathbf{Compatibility}_{\mathcal{S},\mathcal{E}}(\mathsf{m}_0, \mathsf{m}_1)$:
$(\mathsf{sk}_0, \mathsf{pk}_0) \leftarrow_{\$} \mathsf{SGen}(1^\lambda)$
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow_{\$} \mathsf{EGen}(1^\lambda)$
$\mathsf{r} \leftarrow_{\$} \mathcal{R}$
$(\sigma, \mathsf{R}_0) \leftarrow \mathsf{Sign}(\mathsf{m}_0, \mathsf{sk}_0; \mathsf{r})$
$(\mathsf{c}, \mathsf{R}_1) \leftarrow \mathsf{Enc}(\mathsf{m}_1, \mathsf{pk}_1; \mathsf{r})$
Return $(\mathsf{R}_0 = \mathsf{R}_1)$
```

Fig. 3. Partitioning and compatibility tests for a partitioned signature $\mathcal{S} = (\mathsf{SGen}, \mathsf{Sign}, \mathsf{Verify})$, and a partitioned public-key encryption $\mathcal{E} = (\mathsf{EGen}, \mathsf{Enc}, \mathsf{Dec})$

Finally, we also require the following injectivity properties in partitioned schemes, which essentially state that, once the randomness dependent component R is fixed, and for any fixed key pair, the signature generation and encryption algorithms become injective mappings from the message space onto the signature and ciphertext spaces, respectively. We observe that these properties can be relaxed to computational hardness assumptions, and all our results still go through.

Definition 5 (Conditional injectivity). *We say a partitioned signature scheme is* conditionally injective *if for all key pairs* $(\mathsf{sk}, \mathsf{pk})$*, all messages* m *and signatures* (σ, R) *in the appropriate spaces, it holds that:*

$$\mathsf{Sign}(\mathsf{m}, \mathsf{sk}) = (\sigma, \mathsf{R}) \wedge \sigma \neq \sigma' \Rightarrow \mathsf{Verify}(\mathsf{m}, (\sigma', \mathsf{R}), \mathsf{pk}) = \ \mathsf{F}.$$

We say a partitioned encryption scheme is conditionally injective *if for all key pairs* $(\mathsf{sk}, \mathsf{pk})$*, messages* m *and ciphertexts* (c, R) *in the appropriate spaces, it holds that:*

$$\mathsf{Enc}(\mathsf{m}, \mathsf{pk}) = (\mathsf{c}, \mathsf{R}) \wedge \mathsf{c} \neq \mathsf{c}' \Rightarrow \mathsf{Dec}((\mathsf{c}', \mathsf{R}), \mathsf{sk}) \neq \mathsf{m}.$$

REPRODUCIBILITY. Following the approach of Bellare et al. [3], we introduce new notions of *reproducibility* that allow identifying encryption and signature schemes for which it is possible to prove that randomness reuse does not hurt the security of compositions.

Definition 6 (Reproducibility). *We say that a signature scheme is* reproducible *if there exists a deterministic polynomial-time reproduction algorithm* $\mathsf{Rep}_{\mathcal{S}}$ *(resp.* $\mathsf{Rep}_{\mathcal{E}}$*) taking a message, a secret key, and a value* R *such that experiment* $\mathbf{Rep}_{\mathcal{S}}$ *(resp.* $\mathbf{Rep}_{\mathcal{E}}$*) in Figure 4 returns* T *with overwhelming probability for all messages* m *in the appropriate space.*

```
test $\mathbf{Rep}_{\mathcal{S}}(\mathsf{m})$:
$(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{SGen}(1^\lambda)$
$\mathsf{r} \leftarrow_\$ \mathcal{R}$
$(\sigma, \mathsf{R}) \leftarrow \mathsf{Sign}(\mathsf{m}, \mathsf{sk}; \mathsf{r})$
$\sigma' \leftarrow_\$ \mathsf{Rep}_{\mathcal{S}}(\mathsf{m}, \mathsf{sk}, \mathsf{R})$
Return $(\sigma = \sigma')$
```

```
test $\mathbf{Rep}_{\mathcal{E}}(\mathsf{m})$:
$(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{EGen}(1^\lambda)$
$\mathsf{r} \leftarrow_\$ \mathcal{R}$
$(\mathsf{c}, \mathsf{R}) \leftarrow \mathsf{Enc}(\mathsf{m}, \mathsf{pk}; \mathsf{r})$
$\mathsf{c}' \leftarrow_\$ \mathsf{Rep}_{\mathcal{E}}(\mathsf{m}, \mathsf{sk}, \mathsf{R})$
Return $(\mathsf{c} = \mathsf{c}')$
```

Fig. 4. Reproducibility test for a partitioned signature $\mathcal{S} = (\mathsf{SGen}, \mathsf{Sign}, \mathsf{Verify})$ with reproducibility algorithm $\mathsf{Rep}_{\mathcal{S}}$, and a partitioned public-key encryption $\mathcal{E} = (\mathsf{EGen}, \mathsf{Enc}, \mathsf{Dec})$ with reproducibility algorithm $\mathsf{Rep}_{\mathcal{E}}$

Intuitively, the schemes are reproducible if it is possible to reconstruct a valid signature (resp. ciphertext) without having explicit access to the random coins, but instead having access to the secret key. We note that this property seems natural for encryption schemes, where knowledge of the secret key may "compensate" for the lack of knowledge of the implicit randomness. As for signature schemes, this property may seem less natural, as the reproducibility algorithm should be able to produce valid signatures, while having access to apparently less information than the signature generation algorithm itself. However, one can easily see that if $\mathsf{R} = \mathsf{r}$, then a signature scheme is trivially reproducible. Furthermore, Matsuda et al. [13] present various (standard model) signature schemes that, not having this characteristic, are shown to be reproducible. We note that our formalization defines reproducibility as a property of a single scheme, and not as a property of a pair of schemes. We see this as an important definitional choice in ensuring that our framework can be extended to reason about randomness reuse between other cryptographic primitives.

4.2 Security under Randomness-Dependent Attacks

We introduce two new attack models, one for encryption and one for digital signatures. In a nutshell, these models allow messages queried by the adversaries to the relevant oracles to depend on the randomness component R, so this is provided to the adversary in advance. These models are specific for partitioned schemes and aimed at proving security under randomness reuse. We defer considerations on the feasibility of achieving this level of security to the following section.

SECURITY OF ENCRYPTION UNDER RANDOMNESS-DEPENDENT ATTACKS. We define a new security model for encryption, which we call "indistinguishability under randomness-dependent chosen-ciphertext attacks" (IND-RDA). This new model is similar to IND-CCA except that the adversary receives the R component for the challenge in the beginning of the game. To capture this notion of security, rather than partitioning the encryption algorithm, we simply encrypt the fixed all-zeros message at the beginning of the game, in order to obtain a pair (r, R). Note that, since R is guaranteed not to depend on the message, we have that reusing r to produce the challenge ciphertext will yield a consistent security game definition.

Definition 7. *A public-key encryption scheme is* IND-RDA *secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{IND\text{-}RDA}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \Pr[\mathsf{IND\text{-}RDA}_{\mathcal{E},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}] - 1\,,$$

where game $\mathsf{IND\text{-}RDA}_{\mathcal{E},\mathcal{A}}$ *described in Figure 5.*

```
procedure Initialize(1^λ):
    $b \leftarrow_\$ \{0,1\}$
    $\mathsf{List} \leftarrow [\,]$
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{EGen}(1^\lambda)$
    $\mathsf{r} \leftarrow_\$ \mathcal{R}$
    $(\mathsf{c}, \mathsf{R}) \leftarrow \mathsf{Enc}(0, \mathsf{pk}; \mathsf{r})$
    Return $(\mathsf{pk}, \mathsf{R})$

procedure LoR(m_0, m_1):
    $(\mathsf{c}, \mathsf{R}) \leftarrow \mathsf{Enc}(\mathsf{m}_b, \mathsf{pk}; \mathsf{r})$
    $\mathsf{List} \leftarrow (\mathsf{c}, \mathsf{R}) : \mathsf{List}$
    Return $(\mathsf{c}, \mathsf{R})$

procedure Dec(c, R):
    $\mathsf{m} \leftarrow \mathsf{Dec}((\mathsf{c}, \mathsf{R}), \mathsf{sk})$
    Return $\mathsf{m}$

procedure Finalize(b'):
    Return $(b = b')$
```

Fig. 5. Game IND-RDA for a partitioned public-key encryption $\mathcal{E} = (\mathsf{EGen}, \mathsf{Enc}, \mathsf{Dec})$. An adversary $\mathcal{A}$ is legitimate if: 1) it calls **LoR** once, with $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ and $|\mathsf{m}_0| = |\mathsf{m}_1|$; and 2) it never calls **Dec** on $(\mathsf{c}, \mathsf{R}) \in \mathsf{List}$.

SECURITY OF SIGNATURES UNDER RANDOMNESS-DEPENDENT ATTACKS. We also introduce a new security notion for partitioned signature schemes, which we call "strong unforgeability under randomness-dependent chosen message attacks" (sUF-RDA). This new model is similar to sUF-CMA, with the caveat that calls to the **Sign** oracle are done in two steps. On a first interaction, the adversary obtains the randomness component for the signature scheme, and in the next step it provides the message on which the full signature is generated.

Definition 8. *A digital signature scheme is* sUF-RDA *secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{sUF\text{-}RDA}}_{\mathcal{S},\mathcal{A}}(\lambda) := \Pr[\mathsf{sUF\text{-}RDA}_{\mathcal{S},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}]\,,$$

where game $\mathsf{sUF\text{-}RDA}_{\mathcal{S},\mathcal{A}}$ *described in Figure 6.*

```
procedure Initialize($1^\lambda$):
$(\mathsf{sk},\mathsf{pk}) \leftarrow_\$ \mathsf{SGen}(1^\lambda)$
$\mathsf{List} \leftarrow [\,]$
$\mathsf{flag} \leftarrow \mathsf{F}$
Return $(\mathsf{pk}_S)$

procedure Sign($\mathsf{m}$):
If $\mathsf{flag} = \mathsf{T}$
    $(\sigma, \mathsf{R}) \leftarrow \mathsf{Sign}(\mathsf{m}, \mathsf{sk}; \mathsf{r})$
    $\mathsf{List} \leftarrow (\mathsf{m}, (\sigma, \mathsf{R})) : \mathsf{List}$
    $\mathsf{flag} \leftarrow \mathsf{F}$
    Return $(\sigma, \mathsf{R})$
Else
    $\mathsf{r} \leftarrow_\$ \mathcal{R}$
    $(\sigma, \mathsf{R}) \leftarrow \mathsf{Sign}(0, \mathsf{sk}; \mathsf{r})$
    $\mathsf{flag} \leftarrow \mathsf{T}$
    Return $(\perp, \mathsf{R})$

procedure Finalize($\mathsf{m}, (\sigma, \mathsf{R})$):
If $(\mathsf{m}, (\sigma, \mathsf{R})) \notin \mathsf{List} \wedge \mathsf{Verify}(\mathsf{m}, (\sigma, \mathsf{R}), \mathsf{pk})$
    Return $\mathsf{T}$
Else Return $\mathsf{F}$
```

Fig. 6. Game sUF-RDA for a partitioned signature $\mathcal{S} = (\mathsf{SGen}, \mathsf{Sign}, \mathsf{Verify})$

It is clear that the security notion IND-RDA implies IND-CCA, and that sUF-RDA implies sUF-CMA. On the other hand, it is easy to find counterexamples showing that IND-CCA does not imply IND-RDA, nor does sUF-CMA imply sUF-RDA: simply construct a scheme based on an encryption/signature algorithm that returns the secret key when the input message is a fixed function of (e.g., equal to) the randomness component. We note that such counterexamples can be constructed even if the underlying schemes are reproducible, which shows reproducibility is not sufficient to imply randomness-dependent security.

4.3 Secure Compositions under Randomness Reuse

Let a digital signature $\mathcal{S}$ and a public-key encryption $\mathcal{E}$ be two *compatible* schemes, with randomness space $\mathcal{R}$. Our first construction, denoted EtS and described in Figure 7, produces a signcryption scheme from $\mathcal{E}$ and $\mathcal{S}$ in an encrypt-then-sign composition with randomness reuse. Conversely, in the StE construction, $\mathcal{E}$ and $\mathcal{S}$ are used in a sign-then-encrypt composition as shown in Figure 8, also with randomness reuse. We observe that we adopt the strategy proposed by An et al. [2] to achieve security in the multi-user model, by always including the receiver's public key in the signed data, and always including the sender's public key in the encrypted payload, so that it can be checked for consistency upon decryption[1].

```
Gen($1^\lambda$):
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow_\$ \mathsf{SGen}(1^\lambda)$
$(\mathsf{sk}_2, \mathsf{pk}_2) \leftarrow_\$ \mathsf{EGen}(1^\lambda)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow ((\mathsf{sk}_1, \mathsf{sk}_2), (\mathsf{pk}_1, \mathsf{pk}_2))$
Return $(\mathsf{sk}, \mathsf{pk})$

Signcrypt($\mathsf{m}, \mathsf{sk}_S, \mathsf{pk}_S, \mathsf{pk}_R$):
$(\mathsf{sk}_1, \mathsf{sk}_2) \leftarrow \mathsf{sk}_S$
$(\mathsf{pk}_1, \mathsf{pk}_2) \leftarrow \mathsf{pk}_R$
$\mathsf{r} \leftarrow_\$ \mathcal{R}$
$(\mathsf{c}, \mathsf{R}) \leftarrow \mathsf{Enc}((\mathsf{m}, \mathsf{pk}_S), \mathsf{pk}_2; \mathsf{r})$
$(\sigma, \mathsf{R}) \leftarrow \mathsf{Sign}((\mathsf{c}, \mathsf{R}, \mathsf{pk}_R), \mathsf{sk}_1; \mathsf{r})$
$\hat{\mathsf{c}} \leftarrow (\mathsf{c}, \sigma, \mathsf{R})$
Return $\hat{\mathsf{c}}$

Unsigncrypt($\hat{\mathsf{c}}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{pk}_R$):
$(\mathsf{pk}_1, \mathsf{pk}_2) \leftarrow \mathsf{pk}_S$
$(\mathsf{sk}_1, \mathsf{sk}_2) \leftarrow \mathsf{sk}_R$
$(\mathsf{c}, \sigma, \mathsf{R}) \leftarrow \hat{\mathsf{c}}$
$(\mathsf{m}, \mathsf{pk}'_S) \leftarrow \mathsf{Dec}((\mathsf{c}, \mathsf{R}), \mathsf{sk}_2)$
If $\mathsf{Verify}((\mathsf{c}, \mathsf{R}, \mathsf{pk}_R), (\sigma, \mathsf{R}), \mathsf{pk}_1) \wedge$
    $\mathsf{pk}_S = \mathsf{pk}'_S$ Return $\mathsf{m}$
Return $\perp$
```

Fig. 7. EtS construction with randomness reuse

The following theorems state the security guarantees provided by these constructions. The proofs can be found in the full version of this paper.

[1] The overhead of encrypting the public key can be greatly reduced by encrypting its image under a collision-resistant hash function, or using an efficient tag-based PKE as proposed in [13].

```
$\mathbf{Gen}(1^\lambda)$:
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow_\$ \mathsf{SGen}(1^\lambda)$
$(\mathsf{sk}_2, \mathsf{pk}_2) \leftarrow_\$ \mathsf{EGen}(1^\lambda)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow ((\mathsf{sk}_1, \mathsf{sk}_2), (\mathsf{pk}_1, \mathsf{pk}_2))$
Return $(\mathsf{sk}, \mathsf{pk})$
```

```
$\mathbf{Signcrypt}(\mathsf{m}, \mathsf{sk}_S, \mathsf{pk}_S, \mathsf{pk}_R)$:
$(\mathsf{sk}_1, \mathsf{sk}_2) \leftarrow \mathsf{sk}_S$
$(\mathsf{pk}_1, \mathsf{pk}_2) \leftarrow \mathsf{pk}_R$
$\mathsf{r} \leftarrow_\$ \mathcal{R}$
$(\sigma, \mathsf{R}) \leftarrow \mathsf{Sign}((\mathsf{m}, \mathsf{pk}_R), \mathsf{sk}_1; \mathsf{r})$
Return $\mathsf{Enc}((\mathsf{m}, \sigma, \mathsf{pk}_S), \mathsf{pk}_2; \mathsf{r})$
```

```
$\mathbf{Unsigncrypt}(\hat{\mathsf{c}}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{pk}_R)$:
$(\mathsf{pk}_1, \mathsf{pk}_2) \leftarrow \mathsf{pk}_S$
$(\mathsf{sk}_1, \mathsf{sk}_2) \leftarrow \mathsf{sk}_R$
$(\mathsf{c}, \mathsf{R}) \leftarrow \hat{\mathsf{c}}$
$(\mathsf{m}, \sigma, \mathsf{pk}'_S) \leftarrow \mathsf{Dec}((\mathsf{c}, \mathsf{R}), \mathsf{sk}_2)$
If $\mathsf{pk}_S = \mathsf{pk}'_S \wedge$
  $\mathsf{Verify}(\mathsf{m}, (\sigma, \mathsf{R}), \mathsf{pk}_1)$
    Return $\mathsf{m}$
Else Return $\perp$
```

Fig. 8. StE construction with randomness reuse

Theorem 1 (Security of the EtS construction). *Suppose signature scheme $\mathcal{S}$ and encryption scheme $\mathcal{E}$ are compatible and that $\mathcal{S}$ is conditionally injective. Then the following hold:*
1) If $\mathcal{E}$ is reproducible and $\mathcal{S}$ is sUF-RDA *secure, then the resulting* EtS *construction is* sUF-iCMA *secure.*
2) If $\mathcal{S}$ is reproducible and $\mathcal{E}$ is IND-CCA *secure, then the resulting* EtS *construction is* IND-iCCA *secure.*

Theorem 2 (Security of the StE construction). *Suppose signature scheme $\mathcal{S}$ and encryption scheme $\mathcal{E}$ are compatible and that $\mathcal{E}$ is conditionally injective. Then the following hold:*
1) If $\mathcal{S}$ is reproducible and $\mathcal{E}$ is IND-RDA *secure, then the resulting* StE *construction is* IND-iCCA *secure.*
2) If $\mathcal{E}$ is reproducible, and $\mathcal{S}$ is sUF-CMA *secure, then the resulting* StE *construction is* sUF-iCMA *secure.*

We note that we obtain chosen-ciphertext security and strong unforgeability, both against insider attackers, even though this could not be achieved simultaneously by plain sequential composition without randomness reuse. The intuition behind the proofs of both theorems is the following. All proofs require simulating challenge signcryptions with shared randomness across encryption and signatures, and such signcryptions must embed a challenge from a signature or encryption security game. If one tried to reduce directly to the standard notions of security for signature and encryption, this proof strategy would fail, as one needs to commit to challenge messages before having access to the randomness associated with the challenge. For example, this means that one would not be able to request a signature on a ciphertext which shares the same randomness, as this randomness is totally hidden from us. The randomness-dependent attack models fix this problem by allowing adversaries to have access to the randomness components R in challenge encryptions and signatures before committing to a challenge message. Having access to R, one can simulate signatures and encryptions using the reproducibility properties of the schemes. For example, when proving that the StE construction is IND-iCCA secure by reducing to the IND-RDA property of the encryption scheme, one constructs the challenge signcryption as follows. When the adversary provides two challenge messages $(\mathsf{m}_0, \mathsf{m}_1)$, one first obtains R and then uses the reproducibility property of $\mathcal{S}$ to simulate signatures σ_0 and σ_1 on the challenge messages. One then queries the **LoR** oracle on the resulting message/signature pairs $(\mathsf{m}_0||\sigma_0, \mathsf{m}_1||\sigma_1)$. The challenge will then be guaranteed to be correctly simulated with randomness reuse. We note that key registration is required for the unforgeability proofs, as the secret keys required to run the reproducibility algorithms must be provided by the adversary. This is not an issue in the chosen-ciphertext security proofs, since the sender's secret key must always be provided to the **LoR** oracle (i.e., this is the case even in the standard dynamic multi-user model for signcryption).

REMARK. The proofs of the theorems actually establish a slightly stronger result than that stated in the theorems. Indeed, the results would still go through if the randomness-dependent security models are modified in line with the weaker notions of generalized chosen-ciphertext security [2] and existential unforgeability. We have chosen not to include the details in the presentation for the sake of clarity.

REMARK. Our constructions aim to minimize the overhead of the resulting signcryption scheme. For this reason, StE construction does *not* include the full signature inside the ciphertext — notice that R is not included inside the ciphertext. We remark that by including the full signature we could relax the security requirements of the signature to *weak* unforgeability, whilst still achieving *strong* unforgeability for the resulting composed signcryption scheme. This security amplification is accomplished by combining the extra binding provided by the randomness sharing with the conditional injectivity of the algorithms.

REMARK. The combination of randomness-dependent security and reproducibility for encryption schemes may be of independent interest in the design of multi-recipient encryption schemes with randomness reuse. Indeed, it is straightforward to show that for schemes displaying both properties the techniques proposed by Bellare et al. [3] can be adapted to prove security under a stronger model than that originally adopted. Recall that in [3] the adversary can place parallel challenge queries of n message pairs to the challenge oracle, and this will return n ciphertexts under n different public keys. The returned ciphertexts share the same encryption randomness. Applying our techniques, one can give extra power to the adversary in that it need not be restricted to making parallel challenge queries, but may choose challenge messages adaptively after seeing ciphertexts that share the same randomness. Note that security can even be proven in an analogue of the key registration model, in which the adversary can choose some keys maliciously.

5 Instantiating the Constructions

5.1 Security under Randomness-Dependent Attacks

One interesting aspect of our results is the requirement for a stronger security guarantee from the underlying signature and encryption components, in order to obtain security under randomness reuse. Concretely, this translates into the randomness-dependent attack models we have introduced in the previous section and raises the obvious question of how likely it is that off-the-shelf public-key encryption or signature schemes meet this level of security. Although we have no positive results for signature schemes in the standard model, we will show in this section that the class of encryption schemes achieving randomness-dependent security is potentially large, simply by looking at

KEM/DEM paradigms for constructing PKEs. In fact, the Kurosawa–Desmedt [12] appears as a notably efficient example that falls within our general framework. This observation allows to go beyond the efficiency levels both in terms of computational load and bandwidth of the previously most efficient standard model constructions.

RDA-SECURE SIGNATURE SCHEMES. For signature schemes, and restricting our attention to constructions whose security does not rely on random oracles, we found that current signature schemes do not meet this level of security. The typical problem, which occurs for example in the Boneh–Boyen signature scheme, is that the security proof critically relies on the ability to postpone the release of the randomness-dependent signature component until after the adversary has provided the message to be signed. This is a possible explanation for the lack of EtS-like constructions with randomness reuse in the standard model. If we admit random oracles, then one can consider any deterministic signature scheme, and randomness reuse no longer makes much sense as an optimization. Luckily, a RDA-secure signature is only required for the EtS construction. We therefore concentrate our attention on StE compositions, where the randomness-dependent security requirement applies only to the underlying encryption scheme.

RDA-SECURE ENCRYPTION FROM THE KEM/DEM PARADIGM. The first formalization of a KEM/DEM composition theorem was presented by Cramer and Shoup in their seminal paper on chosen-ciphertext-secure public-key encryption [9]. To simplify our discussion, we will restrict our attention to KEMs where the ciphertext is public-key independent, i.e., where the user-specific components of the public key passed to encapsulation are not used to calculate the ciphertext c, but only in the calculation of the secret key[2] k. We observe that PKE constructed from KEM/DEM schemes such as the ones we consider are naturally partitioned, and that the KEM ciphertext can be seen as the R component of the PKE ciphertext.

The KEM/DEM composition theorem in [9] roughly goes as follows. One performs a single game hop, modifying the IND-CCA game so that, rather than using the secret key output by the KEM in the challenge ciphertext generation, one uses a random secret key as input to the DEM. The definition of the decryption oracle is also modified consistently with this change. The transition between the two games can then be reduced to the KEM security assumption. The adversary's advantage in the second game can finally be reduced to the security of the DEM, as the secret key being used for data encapsulation is totally unrelated to that output by the KEM.

The same proof strategy can easily be adapted to show that KEM/DEM composition yields a RDA-secure PKE. To see this, observe that the KEM adversary constructed in the proof outlined above is able to obtain the challenge ciphertext (i.e., the R component in the PKE ciphertext) right at the beginning of the game, independently of the PKE adversary's actions. Furthermore, the DEM attacker constructed in the final step of the proof can generate the KEM ciphertext for the challenge right at the beginning. We can therefore conclude that the KEM/DEM construction initially proposed by Cramer and Shoup achieves randomness dependent chosen-ciphertext security without any modifi-

[2] Such schemes are common, and include those originally proposed by Cramer and Shoup [9]. Our results could be generalized to schemes that do not meet this constraint, by introducing a notion of partitioned KEM schemes.

cation. This result shows how our framework generalizes the results published in [13], in which the authors define reproducibility over KEMs, and then prove security of a signcryption scheme constructed from a KEM, a DEM and a signature scheme, in a StE construction with randomness reuse across the KEM and the signature schemes.

REMARK. The authors in [13] actually present their results based on a notion of a tag-based KEM that allows them to bind the sender's public key to the KEM ciphertext, rather than encrypting it together with the payload, but the KEM/DEM composition theorem they rely on does not take advantage of this binding and is a particular case of the one we describe above. Indeed, it is interesting that the tag-KEM/DEM composition paradigm proposed in [1] does *not* immediately yield RDA-secure schemes. The problem here is that the tag-KEM ciphertext can only be obtained after the tag has been defined, and this depends on the encrypted message in the hybrid construction of [1].

RDA-SECURE ENCRYPTION FROM WEAKENED KEY ENCAPSULATION. Hofheinz and Kiltz [10] propose an alternative KEM/DEM composition framework in which the security of the KEM can be weakened, as long as the DEM scheme satisfies a stronger notion of security known as one-time authenticated encryption. Such schemes can be constructed using the encrypt-then-mac approach, but no length-preserving solutions exist [10]. Interestingly this hybrid encryption paradigm preserves the independence between KEM and DEM components that allowed our extension to randomness-dependent attacks to go through. Indeed, the proof for the composition theorem in [10] follows a similar structure as that described above. This means that restricting our attention to (weak) KEM schemes where ciphertexts are public key independent, we immediately obtain partitioned and randomness-dependent chosen-ciphertext secure PKEs that can be used to instantiate our signcryption constructions. Notably, the weak KEM that is used in the very efficient Kurosawa–Desmedt encryption scheme [12] has this property.

5.2 Compatibility, Reproducibility, and Conditional Injectivity

Matsuda et al. [13] presented an extensive description of schemes that meet compatibility, reproducibility and conditional injectivity properties as required by the generic constructions using a tag-based KEM, a signature and a DEM with randomness reuse. Although the presentation is slightly different, all the schemes used to instantiate their constructions can be used to instantiate our own. However, the Boneh–Boyen signature scheme [6] was not considered by [13] as a candidate for signcryption schemes constructed under randomness reuse. We present a modified version of this signature scheme in the following subsection that displays the necessary properties, which enables us to use it in the instantiation of our construction. Additionally, the KEM/DEM compositions we have described above, when using a public-key independent KEM and a deterministic DEM which is one-to-one over the messages, also give suitable encryption schemes for instantiation.

5.3 An Efficient Instantiation

In this section we present a concrete instantiation of our results that, to the best of our knowledge, is the most efficient signcryption providing full insider security without random oracles. The scheme instantiates our StE construction with randomness reuse with the Kurosawa–Desmedt encryption scheme [12] and the Boneh–Boyen signature scheme [6]. On the negative side, the scheme's strong unforgeability is only proven under the key registration restriction. On the other hand, the scheme offers non-repudiation for free, which is inherited from the StE construction: the receiver obtains a valid signature on the recovered message.

THE KUROSAWA–DESMEDT ENCRYPTION SCHEME. We recall the encryption scheme in [12]. Here, $\mathbb{G}$ is a cyclic group of prime order q in which the DDH assumption holds, and $g_1, g_2 \in \mathbb{G}$ are two random distinct generators. Also, SKE is a one-time authenticated symmetric-key encryption scheme. As referred in the previous section, SKE cannot be assumed to be length-preserving, so we will assume a minimum overhead of size $|\mathsf{MAC}|$, corresponding to a MAC tag. The scheme also requires two hash functions $H_1 : \mathbb{G} \to \{0,1\}^k$ and $H_2 : \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q$, where the former must be a secure key-derivation function (i.e., entropy smoothing), and the latter must be target collision resistant. We have shown in the previous section that the Kurosawa–Desmedt encryption scheme is partitioned, that it is randomness-dependent chosen-ciphertext-secure and that, when instantiated with a deterministic and one-to-one DEM it is conditionally injective.

To be used in our constructions, we further require the scheme to be reproducible. It is straightforward to show that the scheme satisfies this property. Given a ciphertext (c, R) under an arbitrary public key, a secret key sk and a message m, the reproducibility algorithm produces a randomness reusing encryption of m as follows. It first takes the $\mathsf{R} = (\mathsf{R}_1, \mathsf{R}_2)$ and calculates a secret key k precisely as this is done in the decryption algorithm using sk. It then encrypts the m under the DEM using k to obtain the required ciphertext (c, R).

```
algorithm Gen:
  $w \leftarrow_{\$} \mathbb{Z}_q, x \leftarrow_{\$} \mathbb{Z}_q$
  $y \leftarrow_{\$} \mathbb{Z}_q, z \leftarrow_{\$} \mathbb{Z}_q$
  $a \leftarrow g_1^w g_2^x, b \leftarrow g_1^y g_2^z$
  $\mathsf{sk} \leftarrow (w, x, y, z)$
  $\mathsf{pk} \leftarrow (a, b)$
  Return $(\mathsf{sk}, \mathsf{pk})$
```

```
algorithm Enc(m, pk):
  $(a, b) \leftarrow \mathsf{pk}$
  $\mathsf{r} \leftarrow_{\$} \mathbb{Z}_q$
  $\mathsf{R}_1 \leftarrow g_1^{\mathsf{r}}, \mathsf{R}_2 \leftarrow g_2^{\mathsf{r}}$
  $s \leftarrow H_2(\mathsf{R}_1, \mathsf{R}_2)$
  $\mathsf{K} \leftarrow H_1(a^{\mathsf{r}} b^{s\mathsf{r}})$
  $\mathsf{c} \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, \mathsf{m})$
  $\mathsf{R} \leftarrow (\mathsf{R}_1, \mathsf{R}_2)$
  Return $(\mathsf{c}, \mathsf{R})$
```

```
algorithm Dec((c, R), sk):
  $(w, x, y, z) \leftarrow \mathsf{sk}$
  $(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathsf{R}$
  $s \leftarrow H_2(\mathsf{R}_1, \mathsf{R}_2)$
  $\mathsf{K} \leftarrow H_1(\mathsf{R}_1^{w+ys} \cdot \mathsf{R}_2^{x+zs})$
  $\mathsf{m} \leftarrow \mathsf{SKE.Dec}(\mathsf{K}, \mathsf{c})$
  Return m
```

Fig. 9. The Kurosawa–Desmedt encryption scheme [12]

THE BONEH–BOYEN SIGNATURE SCHEME. The Boneh–Boyen signature scheme [6] is strongly unforgeable in the standard model. It relies on bilinear groups, and so we briefly recall this notion below.

Definition 9. *A bilinear group description Γ is a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_3, g_4)$ where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are groups of order p with efficiently computable group laws; g_3 and g_4 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively; and e is a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ satisfying the usual properties of bilinearity and non-degeneracy.*

We present the signature scheme of [6] in Figure 10. Observe that we slightly modified the signature and verification algorithms to make the scheme compatible with

Kurosawa–Desmedt encryption [12], i.e., so that signatures present the same R component. Intuitively, we replace the randomness generation operation in the signature algorithm so that, rather than sampling s directly, we obtain it from the R component in a Kurosawa–Desmedt ciphertext. We therefore consider a group $\mathbb{G}$ of order q as described by the Kurosawa–Desmedt encryption scheme, with two distinct generators $g_1, g_2 \in \mathbb{G}$.

We require an encoding function Map that takes a random element in group $\mathbb{G}$ onto an element in the randomness space of the Boneh–Boyen signature scheme.[3] This encoding function is fed with the first element in the Kurosawa–Desmedt R component g_1^{r}. The second element g_2^{r} is simply included as part of the signed message; we use the standard approach of extending the Boneh–Boyen signature scheme to messages of arbitrary length, introducing a collision-resistant hash function $H : \{0,1\}^\star \rightarrow \mathbb{Z}_p$. We note that the apparent loss in efficiency in the signature scheme disappears when one uses this version of the scheme in our StE construction. Also note that the signature scheme is reproducible. The reproduction algorithm proceeds identically to the signature algorithm, except it skips the steps where $\mathsf{r} \leftarrow_{\$} \mathbb{Z}_q$ and R are computed.

```
algorithm Gen:
$x \leftarrow_{\$} \mathbb{Z}_p, y \leftarrow_{\$} \mathbb{Z}_p$
$u \leftarrow g_4^x, v \leftarrow g_4^y$
$z \leftarrow e(g_3, g_4)$
$\mathsf{sk} \leftarrow (x, y)$
$\mathsf{pk} \leftarrow (u, v, z)$
Return $(\mathsf{sk}, \mathsf{pk})$
```

```
algorithm Sign$(\mathsf{m}, \mathsf{sk})$:
$(x, y) \leftarrow \mathsf{sk}$
$\mathsf{r} \leftarrow_{\$} \mathbb{Z}_q$
$\mathsf{R}_1 \leftarrow g_1^{\mathsf{r}}, \mathsf{R}_2 \leftarrow g_2^{\mathsf{r}}$
$s \leftarrow \mathsf{Map}(\mathsf{R}_1)$
$h \leftarrow H(\mathsf{m}, \mathsf{R}_2)$
$a \leftarrow 1/(x + h + ys) \bmod p$
$\sigma \leftarrow g_3^a$
$\mathsf{R} \leftarrow (\mathsf{R}_1, \mathsf{R}_2)$
Return $(\sigma, \mathsf{R})$
```

```
algorithm Verify$(\mathsf{m}, (\sigma, \mathsf{R}), \mathsf{pk})$:
$(u, v, z) \leftarrow \mathsf{pk}$
$(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathsf{R}$
$h \leftarrow H(\mathsf{m}, \mathsf{R}_2)$
$s \leftarrow \mathsf{Map}(\mathsf{R}_1)$
If $e(\sigma, u \cdot g_4^h \cdot v^s) = z$
    Return T
Else Return F
```

Fig. 10. The Boneh–Boyen signature scheme [6] modified to be compatible with the Kurosawa–Desmedt encryption scheme

We now discuss the security of the modified Boneh–Boyen signature scheme. It is straightforward to show that this scheme remains strongly unforgeable provided that the DDH problem is hard in group $\mathbb{G}$ and that Map is a one-to-one and efficiently invertible mapping from $\mathbb{G}$ to $\mathbb{Z}_p$ (the inversion algorithm is only used in the proof of security). A closer look at the proof reveals that even weaker properties on Map suffice. Indeed, the function only needs to be injective, efficiently invertible, and map $\mathbb{G}$ to a sufficiently large fraction of $\mathbb{Z}_p$ elements. To meet these requirements, we may instantiate $\mathbb{G}$ as the group of points on an elliptic curve, where the DDH problem is assumed to be hard. Standard point compression techniques [5] allow us to instantiate Map with an injective encoding whose image corresponds to a sufficiently large fraction of $\mathbb{Z}_p$ values. More precisely, for carefully chosen elliptic curves, there exist injective and efficiently invertible mappings from curve points into bit strings of length l, where l is approximately the logarithm of the order of the group. Such encodings will have the property we require when p is chosen to be sufficiently close to 2^l.

[3] As in the original scheme, in the unlikely event that $s = -(x + h)/y$, we simply sample a new randomness. We omit this in Figure 10 for readability.

The security proof of the modified Boneh–Boyen signature scheme can be found in the full version of this paper. Intuitively, to reduce the security of the modified scheme to the original version, one simulates signature queries by repeatedly querying the signature oracle, until one obtains a signature where the randomness value can be inverted back into $\mathbb{G}$. Furthermore, a valid forgery on the modified scheme will still constitute a valid forgery on the original scheme.

COMPARISON. We present in Table 1 a comparison of our StE construction with randomness reuse, when instantiated with the Kurosawa–Desmedt encryption scheme and the Boneh–Boyen signature scheme, with previous signcryption constructions in various relevant parameters. We consider only signcryption schemes offering full insider security in dynamic multi-user models, and not relying on random oracles. We present results for the 80-bit security level. In addition to efficiency considerations, we also present the underlying computational assumptions, whether key registration is required, and whether the scheme offers non-repudiation by providing receiver's with valid signatures on the recovered messages.

For computational efficiency, we compare the number of exponentiations, multi-exponentiations, and pairing computations (in this order), both in the signcryption and unsigncryption operations. Clearly the new scheme matches the previously computationally more efficient solution from [14]. We also include the size of the random coins required for the signcryption operation. Here, our scheme displays a saving of 50% over previous solutions, due to the randomness reuse optimization. Finally, in terms of overhead (i.e., the difference between ciphertext length and message length), our scheme compares favorably with other solutions. The 160-bit overhead with respect to the solutions in [8,13] can be explained by including a digest of the sender's public key in the payload, which must be calculated using a collision-resistant hash function. This might be avoided by considering a tag-based variant of the encryption scheme as in [8,13], although we have not considered this possibility.

Table 1. Comparison with signcryption schemes in the literature. We consider [14,15] also instantiated with the BB signature scheme. We take $|\mathbb{G}| = |\mathbb{Z}_p| = |\mathsf{H}| = 160$ and $|\mathsf{MAC}| = 80$ bits.

Scheme	Assumptions	Key Reg.	Non-Rep.	Computations sc.	 usc.	Randomness (bits)	Overhead (bits)
[8]	DBDH, q-SDH	No	Yes	$[4,0,0]$	$[1,1,2]$	320	640
[8]	DBDH, q-SDH	No	Yes	$[3,1,0]$	$[1,1,2]$	320	720
[13]	DBDH, co-CDH	Yes	Yes	$[4,1,0]$	$[1,1,3]$	320	640
[16]	DBDH, q-SDH	Yes	No	$[3,2,0]$	$[3,1,4]$	480	800
[14]	DDH, q-SDH	No	No	$[3,1,0]$	$[0,2,1]$	320	720
[15]	DDH, q-SDH	No	No	$[4,1,0]$	$[1,2,1]$	320	800
New scheme	DDH, q-SDH	Yes	Yes	$[3,1,0]$	$[0,2,1]$	160	720

Acknowledgements. The authors would like to thank Nigel Smart and the anonymous ACNS'12 reviewers for helping to improve the quality of the paper. A. Arriaga and M. Barbosa were supported by project SMART (ref. 120224), financed by National Funds through the Portuguese Foundation for Science and Technology and by ENIAC Joint Undertaking.

References

1. Abe, M., Gennaro, R., Kurosawa, K.: Tag-KEM/DEM: A new framework for hybrid encryption. Journal of Cryptology 21, 97–130 (2008)
2. An, J.H., Dodis, Y., Rabin, T.: On the Security of Joint Signature and Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
3. Bellare, M., Boldyreva, A., Staddon, J.: Randomness Re-use in Multi-recipient Encryption Schemeas. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 85–99. Springer, Heidelberg (2002)
4. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
5. Blake, I., Seroussi, G., Smart, N.: Elliptic Curves in Cryptography. London Mathematical Society Lecture Note Series, vol. 265. Cambridge University Press (1999)
6. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology 21, 149–177 (2008)
7. Boneh, D., Shen, E., Waters, B.: Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 229–240. Springer, Heidelberg (2006)
8. Chiba, D., Matsuda, T., Schuldt, J.C.N., Matsuura, K.: Efficient Generic Constructions of Signcryption with Insider Security in the Multi-user Setting. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 220–237. Springer, Heidelberg (2011)
9. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
10. Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
11. Kurosawa, K.: Multi-recipient Public-Key Encryption with Shortened Ciphertext. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 7–38. Springer, Heidelberg (2002)
12. Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
13. Matsuda, T., Matsuura, K., Schuldt, J.C.N.: Efficient Constructions of Signcryption Schemes and Signcryption Composability. In: Roy, B., Sendrier, N. (eds.) INDOCRYPT 2009. LNCS, vol. 5922, pp. 321–342. Springer, Heidelberg (2009)
14. Tan, C.H.: Insider-secure hybrid signcryption scheme without random oracles. In: Availability, Reliability and Security – ARES 2007, pp. 1148–1154 (2007)
15. Tan, C.H.: Insider-secure signcryption KEM/Tag-KEM schemes without random oracles. In: Availability, Reliability and Security – ARES 2008, pp. 1275–1281 (2008)
16. Tan, C.H.: Signcryption Scheme in Multi-user Setting without Random Oracles. In: Matsuura, K., Fujisaki, E. (eds.) IWSEC 2008. LNCS, vol. 5312, pp. 64–82. Springer, Heidelberg (2008)
17. Zheng, Y.: Digital Signcryption or How to Achieve Cost (Signature & Encryption) $<<$ Cost(Signature) + Cost(Encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)