On the Robustness of RSA-OAEP Encryption and RSA-PSS Signatures Against (Malicious) Randomness Failures

Jacob C. N. Schuldt
AIST, Japan
jacob.schuldt@aist.go.jp

Kazumasa Shinagawa
AIST and University of Tsukuba, Japan
shinagawa@cipher.risk.tsukuba.ac.jp

ABSTRACT

It has recently become apparent that both accidental and maliciously caused randomness failures pose a real and serious threat to the security of cryptographic primitives, and in response, researchers have begone the development of primitives that provide robustness against these. In this paper, however, we focus on standardized, widely available primitives. Specifically, we analyze the RSA-OAEP encryption scheme and RSA-PSS signature schemes, specified in PKCS#1, using the related randomness security notion introduced by Paterson et al. (PKC 2014) and its extension to signature schemes. We show that, under the RSA and Φ-hiding assumptions, RSA-OAEP encryption is related randomness secure for a large class of related randomness functions in the random oracle model, as long as the recipient is honest, and remains secure even when additionally considering malicious recipients, as long as the related randomness functions does not allow the malicious recipients to efficiently compute the randomness used for the honest recipient. We furthermore show that, under the RSA assumption, the RSA-PSS signature scheme is secure for any class of related randomness functions, although with a non-tight security reduction. However, under additional, albeit somewhat restrictive assumptions on the related randomness functions and the adversary, a tight reduction can be recovered. Our results provides some reassurance regarding the use of RSA-OAEP and RSA-PSS in environments where randomness failures might be a concern. Lastly, we note that, unlike RSA-OAEP and RSA-PSS, several other schemes, including RSA-KEM, part of ISO 18033-2, and DHIES, part of IEEE P1363a, are not secure under simple repeated randomness attacks.

1. INTRODUCTION

Modern cryptographic primitives are designed to meet strong notions of security, such as IND-CCA security in the case of encryption or UF-CMA security in the case of signatures, and the design of most concrete schemes are supported by a proof of security that reduce the security of the concrete scheme to appropriate computational assumptions. However, the security analysis often assumes that access to a source providing perfect uniformly distributed randomness is provided. Unfortunately, this assumption might not hold in practice. Random number generators (RNGs), used to generate randomness in practical systems, are notoriously hard to implement and test, which is evident by the history of RNG failures [19, 20, 24, 25, 32]. The perhaps best known example of this, is the Debian Linux vulnerability that causes only 15 bits of entropy to be used in the RNG [18]. However, implementation errors are not the only source of randomness failures. In particular, a determined adversary might attempt to subvert the used RNG, as suggested by the Dual EC controversy [14]. Furthermore, the deployment of virtual machine servers (e.g. using Amazon Web Services, Microsoft Azure, or Google Cloud), creates another randomness attack vector; as observed in [36], an attacker capable of provoking a virtual machine reset can cause the virtual machine to use repeated and correlated randomness values, as the entire state of the virtual machine, including the state of the RNG, will be restored to a previous state.

The consequences of randomness failures can be fatal to security; the examples of real-world security incidents due to poor randomness are many (e.g. [11, 12, 26, 31, 36]). Note that the widely used DSA signature scheme (and the elliptic curve variant ECDSA) standardized by NIST in FIPS 186-4 [33], are particularly vulnerable to randomness failures as the signing key can be recovered from two signatures using the same randomness. This property has lead to the compromise of the Playstation 3 [11], the recovery of TLS signing keys from virtualized servers [36], and allowed the theft of Bitcoins [13]. This clearly illustrates the need for primitives robust against randomness failures.

As the security risks caused by randomness failures are becoming more evident, cryptographers have begun defining various security notions capturing different kinds of randomness failures, and design schemes that provide security against these to the extend that this is possible. In the symmetric key encryption setting, Kamara and Katz [30] considered chosen randomness attacks in which the adversary can freely choose the randomness, except for in the challenge queries. In the public key encryption setting, Bellare et al. [6] introduced hedged encryption, which ensures that security is maintained as long as the message and randomness combined has sufficient entropy, and that a level of security corresponding to deterministic encryption [4] is

ASIA CCS '17, April 02 - 06, 2017, Abu Dhabi, United Arab Emirates

ISBN 978-1-4503-4944-4/17/04...$15.00

DOI: http://dx.doi.org/10.1145/3052973.3053040

achieved when neither randomness nor messages have entropy. Bellare and Tackmann [10] introduced the notion of nonce-based public key encryption which protects against randomness failures, but also requires a stateful scheme. Yilek introduced reset attacks in which repeated randomness values might occur for ordinary public key encryption. Lastly, Paterson et al. [34] introduced related randomness attacks, which allow the adversary a large degree of control over the randomness used for encryption, and which captures reset attacks as a special case.

While this line of research provides new schemes that are robust against various kinds of randomness failures, the more immediate question of how robust existing schemes are against randomness failures has not addressed in the previous results. Specifically, given the typical time frame for the development, evaluation, standardization, and deployment of new cryptographic primitives, the following question is highly relevant for assessing the security of existing systems and making design choices for systems currently in development:

To what extent are existing standardized and widely supported primitives secure against randomness failures?

1.1 Our Contribution

In this paper, we focus on the above question in the case of signatures and encryption. For our analysis, we adopt (a variation of) the related randomness model of Paterson et al. [34] and its extension to signature schemes i.e. we consider security notions which allow the adversary to manipulate the used randomness via related randomness functions ϕ. Specifically, in the case of signatures, the adversary will be able to obtain signatures created using randomness $\phi(r)$ where ϕ is a maliciously chosen function belonging to a function class Φ, and r is a fixed randomness value chosen uniformly by the security experiment[1]. In the case of encryption, the adversary will be able to obtain encryptions for maliciously chosen public keys and messages using randomness $\psi(r)$, and is challenged to distinguish between the encryptions of two maliciously chosen messages under a challenge public key using randomness $\phi(r)$, where ψ and ϕ are maliciously chosen functions belonging to function classes Ψ and Φ. Note that in contrast to the original model from [34] in which $\Phi = \Psi$, we consider separate function families Ψ and Φ, which allows the related randomness used in the encryption for malicious recipients for which the adversary might known the private key, to be distinguished from the related randomness used for the honest challenge user (note that the adversary can use his challenge encryptions as an encryption oracle for the honest challenge user by choosing identical challenge messages). This in turn allows a more detailed statement regarding the related randomness security of the analyzed schemes. See Section 4 for the details of our security model.

Firstly, we focus our attention on the widely used RSA-OAEP encryption scheme [8] included as part of PKCS#1 v2.2 [37] and furthermore adopted in IEEE P1363 [27]. Specifically, we show that, under the RSA and Φ-hiding assumptions, RSA-OAEP encryption is related randomness secure for any function families Φ and Ψ satisfying that Φ is collision resistant, and that Φ is *hard-to-compute* with respect to Ψ. The latter requirement means that, for a randomly chosen input, given the output of functions in Ψ, the output of functions in Φ remains hard to compute for the same input. This implies that RSA-OAEP is secure for a large and general class of related randomness functions which, for example, captures the special case of repeated randomness attacks, when the recipient is honest. Furthermore, even if encryption for malicious recipients is additionally considered, RSA-OAEP remains secure as long the randomness used for the honest recipient cannot be efficiently computed from the randomness used for the malicious recipients. However, we note that since RSA-OAEP encryption is randomness recovering, security is not guaranteed under randomness relations that allow malicious recipients to infer the randomness used for a honest user. This holds for any randomness recovering scheme (see discussion in Section 4).

Secondly, we focus on the RSA-PSS signature scheme [9], which is also part of PKCS#1 v2.2 [37]. Specifically, we show that the RSA-PSS signature scheme is related randomness secure for any related randomness function family Φ. While this shows robustness against any type of randomness failure, the obtained security reduction is not tight like the original security reduction for RSA-PSS, which was one of the motivating factors behind the design of the scheme. We do show, however, that if related randomness functions are not repeated in signature queries, and Φ is *continuously hard-to-compute*, a tight security reduction can be obtained. The latter requirement means that given the output of a subset of functions in Φ on a randomly chosen input, the output of the remaining functions in Φ is hard to compute. We emphasize that these assumptions can be seen as somewhat restrictive, and that, for example, repeated use of the same random value is not captured by these. However, the restrictions can potentially capture a RNG which is in a state where no new entropy is added, but is evolved for each signature generation.

1.2 Technical Challenges

The RSA-OAEP encryption scheme makes use of a padding scheme reminiscent of a Feistel network. Specifically, using hash functions G and H, a message m is encrypted by firstly picking randomness r, and essentially setting $s \leftarrow m \oplus G(r)$ and $t \leftarrow r \oplus H(s)$. Finally, a ciphertext is obtained by computing $(s||t)^e \mod N$ where e and N are the RSA encryption exponent and modulus, respectively. The standard proof of IND-CCA security of RSA-OAEP [21] crucially depend on r being fresh and unpredictable for the challenge encryption. However, this is not the case in the related randomness setting. For example, consider the case in which the same randomness r is used for the encryption of two different messages m and m'. Here, the values $s = m \oplus G(r)$ and $s' = m' \oplus G(r)$ are correlated. In particular, $s \oplus s' = m \oplus m'$, which is known to the adversary. Hence, $s||t$ and the corresponding $s'||t'$ are not independent, and the approach from original security proof, which relies on replacing $(s||t)^e \mod N$ with a random element of $\mathbb{Z}_N^*$, breaks down. However, by relying on the Φ-hiding assumption and the algebraic properties implied by this[2], we show

[1]While our security notions consider a single value r, as shown in [34], this is equivalent to considering an experiment with multiple r values.

[2]In particular, we make use of the result by Smith and Zhang [40] that essentially shows that, under the Φ-hiding assumption, an arithmetic progression in $\mathbb{Z}_N$ and an uniformly chosen value in $\mathbb{Z}_N$ are indistinguishable when ap-

that a security reduction to the RSA problem can still be obtained in this case (even when relations arising from any collision resistant function family Φ are considered).

The RSA-PSS signature scheme makes use of a different type of padding scheme. More precisely, using hash functions G_1, G_2, and H, a signature on a message m is obtained by firstly picking randomness r and computing $w \leftarrow H(m||r)$ and $y \leftarrow 0||w||(r \oplus G_1(w))||G_2(w)$. Finally, the signature is obtained by computing $\sigma \leftarrow y^d \mod N$, where d and N are the RSA decryption exponent and modulus, respectively. The security proof of RSA-PSS [9], which is tight, relies on the property that a random r will not collide with values queried by the adversary or used in previously generated signatures. This assumption obviously does not hold in the related randomness setting. Note also that since signatures reveal the used randomness, if $\phi'(r)$ is computable from $\phi(r)$, the adversary will be able to compute the randomness which will be used in a future signature query. Additionally, for signature schemes, we might even consider constant functions $\phi(\cdot) = c$ which will make signatures deterministic. In this case, the results by Coron [16] imply that a tight security reduction cannot be obtained. However, adopting the ideas used by Coron [15] to prove security of the full domain hash signature scheme, we show a reduction from the related randomness security of RSA-PSS to the RSA problem for any function family Φ, with a security loss proportional to the number of signing queries. Additionally, we show that when the adversary cannot compute $\phi'(r)$ from $\phi(r)$ (i.e. the function family Φ is continuously hard-to-compute and the adversary does not repeatably query the same function ϕ), a tight reduction can be recovered.

1.3 On the Related Randomness Security of Other Schemes

We will briefly make some simple observation regarding the related randomness security of other encryption and signature schemes. For this purpose, we consider a very weak type of a related randomness attack, repeated randomness, in which the attacker obtains two encryptions (for a challenge public key) or two signatures using the same randomness.

In the case of encryption, the problems that might arise from repeated randomness are well-known from the literature on randomness re-use for the purpose of optimization (e.g. see [3, 5, 35]). For example, it is straightforward to see that the use of repeated randomness will render the ElGamal [23] and Cramer-Shoup [17] encryption schemes insecure, as the structure of the ciphertexts for these schemes allows an attacker to compute the ratio of the encrypted messages when sent to the same recipient[3]. While DHIES [1], standardized in IEEE 1363a [28], is based on ElGamal, a similar attack is not possible due to the hybrid structure of DHIES, in which a key for a symmetric encryption scheme is derived and used to encrypt the message. However, the use of repeated randomness will cause the symmetric encryption component to use the same key and initialization vector (IV), and it is well known that common implementations of symmetric encryption, such as a block cipher used in counter (CTR) or cipher-block-chaining (CBC) mode, becomes insecure in this case. A similar observation holds for encryption based on RSA-KEM [39], standardized in ISO 18033-2 [29], which will also make use of a symmetric encryption component for the encryption of the message.

In the case of signatures, as already mentioned above, the DSA signature scheme and the elliptic curve variant ECDSA, standardized by NIST in FIPS 186-4 [33], becomes insecure if randomness values are repeated, as this allows the signing key to be recovered from the resulting signatures. This is likewise true for the Schnorr signatures scheme [38]. In contrast, the full domain hash signature scheme FDH [7], also specified as part of PKCS#1 v2.2, is deterministic and therefore remain secure for any related randomness attack.

Our results show that, unlike the above mentioned schemes (with the exception of FDH), RSA-OAEP encryption and RSA-PSS signatures provide some protection against randomness failures, and are hence preferable in environments where randomness failures might be a concern.

2. PRELIMINARIES

2.1 Notation

Throughout the paper, we will use the following notation. We let $\mathbb{N}$ denote the set of natural numbers. $\lambda \in \mathbb{N}$ denotes the security parameter, which will sometimes be written in its unary representation, 1^λ, and $\emptyset$ denotes the empty set. We let $x||y$ denote the concatenation of (the binary representation of) x and y. $x \leftarrow y$ denotes the assignment of y to x. $\mathbb{Z}_N$ denotes the residue ring $\mathbb{Z}/N\mathbb{Z}$ and $\mathbb{Z}_N^*$ denotes the multiplicative group of integers modulo N. If S is a set, then $x \leftarrow_\$ S$ denotes the selection of an element x uniformly at random from S. If x is a ℓ-bit string and $\ell \geq n$, $[x]^n$ denotes the n most significant bits of x and $[x]_n$ denotes the n least significant bits of x. If $\mathcal{A}$ is a probabilistic algorithm, then $y \leftarrow \mathcal{A}(x_1, x_2, \cdots)$ denotes that $\mathcal{A}$ takes $x_1, x_2, \cdots$ as inputs and outputs y, and $\mathcal{A}^{\mathcal{O}}$ denotes $\mathcal{A}$ has oracle access to the oracle $\mathcal{O}$. If X and Y are random variables, then $SD(X, Y)$ denotes the statistical distance between X and Y, i.e., $SD(X, Y) = \frac{1}{2}\Sigma_z |\Pr[X = z] - \Pr[Y = z]|$. A function $\epsilon : \mathbb{N} \rightarrow [0, 1]$ is said to be negligible in k if $\epsilon(k) < 1/p(k)$ for any positive polynomial $p(k)$ and all sufficiently large $k \in \mathbb{N}$.

2.2 Collision Resistance

In our analysis of the related randomness security of RSA-OAEP and RSA-PSS (with tight reduction), we will consider a class of related randomness functions Φ which is collision resistant. Following [34], we define this property as follows:

Definition 1 (Collision resistance). *Let $\Phi = \{\phi : \{0,1\}^k \rightarrow \{0,1\}^k\}$ be a family of functions, where $k(\lambda)$ is a function in λ. We say that Φ is* collision-resistant *if $\mathrm{CR}^\Phi(\lambda)$ is negligible in λ, where*

$$\mathrm{CR}^\Phi(\lambda) = \max_{\phi_1, \phi_2 \in \Phi, \phi_1 \neq \phi_2} \Pr[x \leftarrow_\$ \{0,1\}^k : \phi_1(x) = \phi_2(x)].$$

2.3 RSA and Φ-hiding Assumptions

In our security proofs, we will make use of both the RSA assumption, and in the case of RSA-OAEP, also the Φ-hiding assumption. We define these as follows.

plying the RSA function to these.

[3]Let c and c' be ElGamal encryptions of messages m and m' under public key y and randomness r. We then have that $c = (c_1, c_2) = (g^r, y^r \cdot m)$ and $c' = (c'_1, c'_2) = (g^r, y^r \cdot m')$, and can hence compute $m/m' = c_2/c'_2$.

2.3.1 RSA Assumption

Let Primes_λ denote the uniform distribution of λ-bit primes. For a constant $c \in (0,1)$, let $\mathtt{RSAGen}(1^\lambda, c)$ denote the key generation algorithm for RSA that chooses $p, q \leftarrow_\$ \mathsf{Primes}_{\lambda/2}$ and $e \leftarrow_\$ \mathsf{Primes}_{c\lambda}$ and outputs (N, e, d) where $N = p \cdot q$ and $(x^e)^d = x \bmod N$ for all $x \in \mathbb{Z}_N^*$.

Definition 2 (RSA assumption). *Let $c(\lambda) \in (0,1)$ be a function. For any probabilistic polynomial time adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ for the RSA problem is defined as*

$$\mathtt{Adv}^{RSA}_{c,\mathcal{A}}(\lambda) = \Pr\left[x^e = y \bmod N : \begin{array}{l}(N,e,d) \leftarrow \mathtt{RSAGen}(1^\lambda, c);\\ y \leftarrow_\$ \mathbb{Z}_N^*; x \leftarrow \mathcal{A}(N,e,y)\end{array}\right].$$

We say that the RSA assumption *holds for c if $\mathtt{Adv}^{RSA}_{c,\mathcal{A}}(\lambda)$ is negligible in λ for any probabilistic polynomial time adversary $\mathcal{A}$.*

In shorthand, we use $\mathtt{RSAGen}(1^\lambda)$ to denote $\mathtt{RSAGen}(1^\lambda, c)$ for some constant c satisfying RSA assumption. We say that the RSA assumption holds for $\epsilon(\lambda)$ if there exists a constant $c \in (0,1)$ such that the RSA assumption holds for $(c, \epsilon(\lambda))$.

Instead of proving the security of RSA-OAEP directly based on the RSA assumption, we will use the following equivalent assumption regarding partial-domain one-wayness:

Definition 3 (Partial-domain one-wayness). *Let f be a permutation $f : \{0,1\}^{n+k_1} \times \{0,1\}^{k_0} \to \{0,1\}^k$, where $k = n + k_0 + k_1$. The advantage for a probabilistic polynomial time algorithm $\mathcal{A}$ outputting a set of ℓ elements is defined as*

$$\mathtt{Adv}^{\text{S-PD-OW}}_{\ell,\mathcal{A}}(\lambda) = \Pr[s \in \mathcal{A}(f(s,t)) : s \leftarrow_\$ \{0,1\}^{n+k_1}, t \leftarrow_\$ \{0,1\}^{k_0}].$$

*We say that f is ℓ-*set partial-domain one-way *if for any probabilistic polynomial time algorithm $\mathcal{A}$ outputting a set of ℓ elements, $\mathtt{Adv}^{\text{S-PD-OW}}_{\ell,\mathcal{A}}(\lambda)$ is negligible in λ.*

The following lemma establishes the equivalence between the RSA assumption and partial-domain one-wayness.

Lemma 1. *(Lemma 4 in [22]) Let $2^{k-1} < N < 2^k$ and let $k > 2k_0$. Let $\mathcal{A}$ be a probabilistic polynomial time algorithm that given $x \in \mathbb{Z}_N$, outputs a set of l strings containing the $k - k_0$ most significant bits of the e-th root of x with probability ϵ (i.e. $\mathcal{A}$ breaks the l-set partial-domain one-wayness of $f : x \to x^e \bmod N$). Then there exists a polynomial time algorithm $\mathcal{B}$ that solves the RSA problem (N, e) with success probability ϵ', where*

$$\epsilon' \geq \epsilon \cdot (\epsilon - 2^{2k_0 - k + 6}).$$

2.3.2 Φ-hiding Assumption

The Φ-hiding assumption informally states that given a RSA modulus N, it is not possible to distinguish primes e' which do not divide $\Phi(N)$ from primes e that do, where $\Phi(N) = (p-1)(q-1)$ is Euler's totient function. Note that in the latter case, the RSA function $f : x \to x^e \bmod N$ is no longer a permutation, but becomes a lossy function. Following [40], we define the Φ-hiding assumption as follows.

Let $\mathsf{Primes}_\lambda[\cdots]$ denote the uniform distribution of λ-bit primes satisfying the condition in brackets. Let $\mathsf{RSA}^{inj}_{c,\theta}$ and $\mathsf{RSA}^{loss}_{c,\theta}$ be algorithms that output the public key (pq, e) and a lossy public key (pq, e) satisfying $p = 1 \bmod e$, respectively (Figure 1).

Definition 4 (Φ-hiding assumption). *Let $c(\lambda), \theta(\lambda)$ be functions such that $c \in (0,1)$ and θ is an even integer satisfying $0 < \theta < \lambda$. For any probabilistic polynomial time distinguisher $\mathcal{A}$, the advantage of $\mathcal{A}$ for breaking the* Φ-hiding assumption *is defined as*

$$\mathtt{Adv}^{\Phi\mathsf{A}}_{c,\theta,\mathcal{A}}(\lambda) = \left|\Pr[\mathcal{A}(\mathsf{RSA}^{inj}_{c,\theta}(\lambda)) = 1] - \Pr[\mathcal{A}(\mathsf{RSA}^{loss}_{c,\theta}(\lambda)) = 1]\right|.$$

We say that the Φ-hiding assumption holds for (c, θ, ϵ) if for any probabilistic polynomial time distinguisher $\mathcal{A}$, $\mathtt{Adv}^{\Phi\mathsf{A}}_{c,\theta,\mathcal{A}}(\lambda)$ is negligible in λ.

Smith and Zhang [40] showed that, under a lossy key (N, e), the distributions of $y^e \bmod N$ and $z^e \bmod N$ are statistically close, where y is chosen from an arithmetic progression and z is chosen uniformly at random. We will make use of this lemma to prove the security of IND-RR-CCA security of RSA-OAEP (Lemma 12).

Lemma 2. *(Theorem 2 in [40]) Let $N = pq$ (p, q are primes) and assume that $q > p$ and that σ, N are co-prime. Let $P_K = \{\sigma i + \tau : i = 0, 1, \cdots, K-1\}$ and assume that $K > q$. Let e be such that $p = 1 \bmod e$ and $e, q-1$ are co-prime. Then,*

$$SD(y^e \bmod n, z^e \bmod N) \leq \frac{3q}{K} + \frac{2p}{q-1} + \frac{2}{p-1} + \sqrt{\frac{N}{eK}},$$

where $y \leftarrow_\$ P_K$ and $z \leftarrow_\$ \mathbb{Z}_N^*$.

2.4 Signatures

A signature scheme $\mathtt{S}$ is defined by three algorithms with the following functionality:

$\mathtt{S.KeyGen}(1^\lambda)$ This is the key generations algorithm, which on input the security parameters 1^λ, returns a key pair (vk, sk) consisting of a verification key vk and a signing key sk.

$\mathtt{S.Sign}(sk, m)$ This is the signing algorithm, which on input a signing key sk, and a message m, returns a signature σ on m.

$\mathtt{S.Verify}(vk, m, \sigma)$ This is the verification algorithm, which on input a verification key vk, a message m, and a signature σ, returns either a symbol $\top$ indicating that σ is accepted as a valid signature on m under vk, or the rejection symbol $\bot$.

We require a signature scheme to satisfy *perfect correctness*, that is, for all λ, all $(vk, sk) \leftarrow \mathtt{S.KeyGen}(1^\lambda)$, and all messages m, it holds that $\mathtt{S.Verify}(vk, m, \mathtt{S.Sign}(sk, m)) = \top$.

2.5 Public Key Encryption

A public key encryption scheme $\mathtt{PKE}$ is defined by three algorithms with the following functionality:

$\mathtt{PKE.KeyGen}(1^\lambda)$ This is the key generations algorithm, which on input the security parameter 1^λ, returns a public/private key pair (pk, sk). The public key pk defines a supported message $\mathcal{M}(pk)$.

$\mathtt{PKE.Enc}(pk, m)$ This is the encryption algorithm, which on input a public key pk and a message m, returns an encryption c of m under pk.

$\underline{\mathsf{RSA}^{inj}_{c,\theta}(\lambda)}$:
$(N, e, d) \leftarrow_{\$} \mathtt{RSAGen}(1^\lambda, c)$
return (N, e)

$\underline{\mathsf{RSA}^{loss}_{c,\theta}(\lambda)}$:
$e \leftarrow_{\$} \mathsf{Primes}_{c\lambda}$
$p \leftarrow_{\$} \mathsf{Primes}_{\frac{\lambda}{2}-\frac{\theta}{2}}[p = 1 \bmod e]$
$q \leftarrow_{\$} \mathsf{Primes}_{\frac{\lambda}{2}+\frac{\theta}{2}}$
return (pq, e)

Figure 1: Algorithms for defining the Φ-hiding assumption

$\mathtt{PKE.Dec}(par, sk, c)$ This is the decryption algorithm, which on input a private key sk and a ciphertext c, returns either a message m or the error symbol $\perp$.

We require that an encryption scheme satisfies *perfect correctness*, that is, for all λ, all $(pk, sk) \leftarrow \mathtt{PKE.KeyGen}(1^\lambda)$, and all $m \in \mathcal{M}(pk)$, it holds that $\mathtt{PKE.Dec}(sk, \mathtt{PKE.Enc}(pk, m)) = m$.

3. PKCS#1

The Public Key Cryptography Standards (PKCS) are a group of cryptographic standards originally published by RSA Securities Inc. The PKCS#1 standard specifies RSA-based public-key encryption and signatures, and the current version, PKCS#1 v2.2 [37], includes the specification of RSA-OAEP encryption and RSA-PSS signatures, which we will recall below. Besides these, PKCS#1 v2.2 includes a signature scheme originating from the earlier PKCS#1 v1.5 standard, which is similar to the FDH signature scheme (unlike FDH, a simple padding scheme is used), as well as an encryption scheme which also originates from PKCS#1 v1.5. However, as the PKCS#1 v1.5 encryption scheme does not provide an appropriate level of security (e.g. see [2]), we will not discuss this further.

3.1 RSA-OAEP Encryption

The RSA-OAEP encryption scheme was originally proposed by Bellare and Rogaway [8], and has been shown IND-CCA secure under the RSA assumption in the random oracle model [21, 22].

In our description of the scheme we make use of $\mathtt{RSAGen}$, and the scheme is parameterized by k_0 and k_1 which are values satisfying $k = n + k_0 + k_1$, where $k(\lambda)$ is the bit length of the modulus N generated by $\mathtt{RSAGen}(1^\lambda)$ and n is the plaintext length. The scheme makes use of two hash functions, $G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0}$ and $H : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0}$.

$\mathtt{KeyGen}(1^\lambda)$ Run $(N, e, d) \leftarrow \mathtt{RSAGen}(1^\lambda)$, and set $pk \leftarrow (N, e)$ and $sk \leftarrow (N, d)$. Return (pk, sk).

$\mathtt{PKE.Enc}(pk, m)$ Pick $r \leftarrow_{\$} \{0,1\}^{k_0}$ and compute $s = (m||0^{k_1}) \oplus G(r)$ and $t = r \oplus H(s)$. Then sets $c \leftarrow (s||t)^e \bmod N$ and return c.

$\mathtt{PKE.Dec}(sk, c)$ Compute $s||t = c^d \bmod N$, $r = t \oplus H(s)$ and $M = r \oplus G(r)$. If $[M]_{k_1} = 0^{k_1}$, returns $[M]^n$. Otherwise, return $\perp$.

3.2 RSA-PSS Signatures

The RSA-PSS signature scheme makes use of $\mathtt{RSAGen}$ and is parameterized by k_0 and k_1 which are values satisfying $k_0 + k_1 \leq k - 1$, where $k = k(\lambda)$ is the bit length of the modulus N generated by $\mathtt{RSAGen}(1^\lambda)$. The scheme furthermore makes use of two hash functions, $H : \{0,1\}^* \to \{0,1\}^{k_1}$ and $G : \{0,1\}^{k_1} \to \{0,1\}^{k-k_1-1}$. For convenience, we will denote by $G_1(\cdot)$ the k_1 most significant bits of the output of G, and by $G_2(\cdot)$ the remaining $k - k_0 - k_1 - 1$ bits.

$\mathtt{PSS.KeyGen}(1^\lambda)$ Run $(N, e, d) \leftarrow \mathtt{RSAGen}(1^\lambda)$, and set $vk \leftarrow (N, e)$ and $sk \leftarrow (N, d)$. Return (vk, sk).

$\mathtt{PSS.Sign}(sk, m)$ Pick random $r \leftarrow_{\$} \{0,1\}^{k_0}$ and compute $w \leftarrow H(m||r)$. Then set $y \leftarrow 0||w||(r \oplus G_1(w))||G_2(w)$, and return the signature $\sigma \leftarrow y^d \bmod N$.

$\mathtt{PSS.Verify}(vk, m, \sigma)$ Firstly compute $y \leftarrow \sigma^d$ and parse $y \to b||w||r'||\gamma$. Furthermore, set $r \leftarrow r' \oplus G_1(w)$. If $b = 0$, $H(m||r) = w$, and $G_2(w) = \gamma$, return $\top$. Otherwise, return $\perp$.

4. SECURITY MODELS

Related randomness security for encryption was introduced by Paterson et al. [34], and captures a broad rage of randomness failures. In this section, we define (a slight variation of) the security notion from [34] which we will use to analyze RSA-OAEP encryption. We furthermore define the natural adaptation of this security notion to signature schemes which we will use in our analysis of RSA-PSS signatures. We note that Yuen et al. [41] considered an adaptation of the [34] model to signature schemes which additionally takes into account related key attacks. However, the notion we define here is only concerned with randomness failures and will not take into account an adversary with the ability to manipulate the private key material of the signer.

4.1 Related Randomness Secure Encryption

The related randomness security notion defined in [34] allows the adversary to control the randomness used in encryption via related randomness functions. Specifically, the security experiment initially picks a uniformly distributed value r, and the adversary is allowed to request encryptions $\mathtt{Enc}(pk, m; \psi(r))$ for public keys pk, messages m, and related randomness functions ψ of his choice. Note that pk might be a maliciously generated public key, and that the adversary potentially knows the corresponding private key. This captures that encryptions might be done for malicious users. The adversary is challenged to distinguish between encryptions $\mathtt{Enc}(pk^*, m_0; \phi(r))$ and $\mathtt{Enc}(pk^*, m_1; \phi(r))$ for a challenge public key pk^* honestly generated by the experiment, and messages m_0, m_1 and related randomness function ϕ of his choice. Note that in this model, the adversary cannot influence the randomness used to generate pk^*. The adversary is allowed to make multiple challenge queries, as in the related randomness setting, multi-challenge security is not implied by single-challenge security as for ordinary IND-CCA security[4]. Furthermore, note that the challenge

[4]Note that the challenge queries are no longer independent due to the use of related randomness, and hence cannot be treated separately in a reduction to single-challenge security.

$\underline{\text{IND-RR-CCA}^{\text{PKE}}_{\mathcal{A}}(\lambda)}$:
$(pk^*, sk^*) \leftarrow \texttt{PKE.KeyGen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$; $r \leftarrow_\$ \mathcal{R}$
$\mathcal{S} \leftarrow \emptyset$
$b' \leftarrow \mathcal{A}^{\text{LR,Enc,Dec}}(par, pk^*)$
return $(b = b')$

proc. Enc(pk, m, ψ):
$c \leftarrow \texttt{PKE.Enc}(pk, m; \psi(r))$
return c

proc. Dec(c):
if $c \in \mathcal{S}$, then return $\perp$
else return $\texttt{PKE.Dec}(sk^*, c)$

proc. LR(m_0, m_1, ϕ):
$c \leftarrow \texttt{PKE.Enc}(pk^*, m_b; \phi(r))$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{c\}$
return c

Figure 2: Game defining indistinguishability under related randomness and chosen ciphertext attacks (IND-RR-CCA)

queries provide the adversary with an encryption oracle for the honestly generated pk^* and functions ϕ; by submitting (m, m, ϕ), the adversary obtains $\texttt{Enc}(pk^*, m; \phi(r))$. Lastly, as we consider CCA security, the adversary is allowed to request decryptions under sk^* corresponding to pk^* for ciphertexts of his choice, but as decryption is assumed to be deterministic, this is similar to ordinary IND-CCA security. We refer to an adversary who is restricted to picking related randomness functions ϕ and ψ from functions families Φ and Ψ, as (Φ, Ψ)-*restricted*. The full security game is shown in Figure 2.

It should be noted that, in the above description, we consider separate functions families Φ and Ψ, whereas the notion defined in [34] only considers a single function family i.e. $\Phi = \Psi$. This allows us to distinguish between the related randomness functions used for malicious users for which the adversary might know the private key, and the related randomness functions used for the honest challenge user. This in turn allows a more detailed description of the security properties provided by RSA-OAEP encryption.

As should be apparent from the above description, the considered adversary in the related randomness setting is very powerful, and some restrictions must be applied to obtain a meaningful notion of security. For example, it is easy to see that an adversary submitting challenge queries $(m_0, m_1, \texttt{id})$ and $(m_0, m_2, \texttt{id})$ can easily detect whether the first or second message is being encrypted, simply by checking whether the same ciphertext is returned in response to these queries. This is similar to the notions of deterministic encryption and related key attack security where adversary restrictions are likewise needed to ensure a meaningful security definition. In the following, we will introduce a restricted class of adversaries which we denote equality-respecting adversaries. Our notion is slightly weaker than the corresponding notion in [34] as the restriction does not take into account encryption queries. This is possible as we can rule out the trivial attacks not captured by the following definition, by placing restrictions on the function families Ψ and Φ that cannot be captured using a single function family as in [34].

Definition 5 (LR-equality-respecting Adversary). *Consider a (Φ, Ψ)-restricted adversary $\mathcal{A}$ playing the* IND-RR-CCA *security game. Let $(m_0^{\phi,1}, m_1^{\phi,1}), \ldots, (m_0^{\phi,q_\phi}, m_1^{\phi,q_\phi})$ denote the messages $\mathcal{A}$ submits to the* LR *oracle for function ϕ. Then $\mathcal{A}$ is said to be* LR-equality-respecting *if, for all $\phi \in \Phi$ and for all $i, j \in [q_\phi]$ s.t. $i \neq j$,*

$$m_0^{\phi,i} = m_0^{\phi,j} \Leftrightarrow m_1^{\phi,i} = m_1^{\phi,j}$$

With the above restriction in place, we can define the notion of related randomness security.

$\underline{\text{HTC}^{\Phi,\Psi}_{\mathcal{A}}(\lambda)}$:
$x \leftarrow_\$ D$
$\mathcal{S} \leftarrow \emptyset$
$y \leftarrow \mathcal{A}^{\text{Get,Set}}(1^\lambda)$
if $y \in \mathcal{S}$
  return 1
else return 0

proc. Get(i):
return $\psi_i(x)$

proc. Set(j):
$\mathcal{S} \leftarrow \mathcal{S} \cup \{\phi_j(x)\}$

Figure 3: Game defining adaptively hard-to-compute function families

Definition 6 (IND-RR-CCA Security). *Let the advantage of an adversary $\mathcal{A}$ playing the* IND-RR-CCA *game with respect to a public key encryption scheme $\texttt{PKE} = (\texttt{Setup}, \texttt{KeyGen}, \texttt{Enc}, \texttt{Dec})$, be defined as:*

$$\texttt{Adv}^{\text{IND-RR-CCA}}_{\texttt{PKE},\mathcal{A}}(\lambda) = 2\left|\Pr[\text{IND-RR-CCA}^{\mathcal{A}}_{\texttt{PKE}}(\lambda) \Rightarrow 1] - \frac{1}{2}\right|$$

A scheme PKE *is said to be* (Φ, Ψ)-IND-RR-CCA *secure, if for all polynomial time (Φ, Ψ)-restricted and LR-equality-respecting adversaries $\mathcal{A}$, $\texttt{Adv}^{\text{IND-RR-CCA}}_{\texttt{PKE},\mathcal{A}}(\lambda)$ is negligible in the security parameter λ.*

From the definition, it is clear that security cannot be achieved for any function families Ψ and Φ. In [34], it was shown that security is only achievable for collision resistant and unpredictable functions. We furthermore note that, for any randomness recovering scheme, such as RSA-OAEP, security cannot be achieved if $\Psi \cap \Phi \neq \emptyset$. Specifically, if there exist $\psi' \in \Psi \cap \Phi$, the adversary can simply submit (m_0, m_1, ψ') to his LR oracle and (pk, m, ψ') to his Enc oracle using a public key pk for which he knows the private key. From the ciphertext obtained in the latter query, the adversary can recover $\psi'(r)$ which will also be used as randomness in his challenge query. Hence, the adversary can trivially determine the challenge bit b by re-encrypting m_0 and m_1 using $\psi'(r)$.

In this paper, we will be concerned with function families Φ which are collision resistant (see Section 2), and which are furthermore (adaptively) *hard-to-compute* with respect to the function family Ψ. The latter intuitively means that, for a randomly chosen r, the values $\phi(r)$ for $\phi \in \Phi$ are hard to compute, even when given values $\psi(r)$ for $\psi \in \Psi$. We formalize this in the following definition. Note that if Ψ is hard-to-compute with respect to Ψ, it follows that Φ is unpredictable and that $\Psi \cap \Phi = \emptyset$.

Definition 7. (Adaptively hard-to-compute function families). *A function family Φ is said to be* adaptively hard-to-compute *with respect to a function family Ψ if all polynomial time adversaries $\mathcal{A}$ have advantage $\texttt{Adv}^{\text{HTC}}_{\Phi,\Psi,\mathcal{A}}(\lambda)$*

$\underline{\text{UF-RR-CMA}^{\mathtt{S}}_{\mathcal{A}}(\lambda)}$:
$(vk, sk) \leftarrow \mathtt{S.KeyGen}(1^\lambda)$
$r \leftarrow \mathcal{R}$; $\mathcal{M} \leftarrow \emptyset$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\text{SIGN}}(vk)$
if $\mathtt{S.Verify}(vk, m^*, \sigma^*)$
$\quad \wedge m^* \notin \mathcal{M}$
$\quad$ return 1
else return 0

$\underline{\text{proc. SIGN}(m, \phi)}$:
$\sigma \leftarrow \mathtt{S.Sign}(sk, m; \phi(r))$
$\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
return σ

Figure 4: Game defining existential unforgeability under a related randomness and chosen message attack (UF-RR-CMA)

$\underline{\text{CHTC}^{\Phi}_{\mathcal{A}}(\lambda)}$:
$x \leftarrow_{\$} D$
$\mathcal{S} \leftarrow \emptyset$
$(j, y) \leftarrow \mathcal{A}^{\text{GET}}(1^\lambda)$
if $y = \phi_j(x) \wedge j \notin \mathcal{S}$
$\quad$ return 1
else return 0

$\underline{\text{proc. GET}(i)}$:
$\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$
return $\phi_i(x)$

Figure 5: Game defining continuously hard-to-compute function families

is negligible in λ, where

$$\mathtt{Adv}^{\text{HTC}}_{\Phi,\Psi,\mathcal{A}}(\lambda) = \Pr[\text{HTC}^{\Phi,\Psi}_{\mathcal{A}}(1^\lambda) \Rightarrow 1]$$

4.2 Related Randomness Secure Signatures

Adapting the related randomness security notion from [34] is relatively simple. We consider a security experiment which initially chooses a value r, and then allow the adversary to obtain signatures $\sigma = \mathtt{Sign}(sk, m; \phi(r))$ for his choice of message m and related randomness function ϕ. As in the case of encryption, we refer to an adversary as Φ-restricted, if his is restricted to querying functions $\phi \in \Phi$. The full security experiment is shown in Figure 4.

DEFINITION 8 (UF-RR-CMA SECURITY). *Let the advantage of an adversary $\mathcal{A}$ playing the* UF-RR-CMA *game with respect to a signature scheme $\mathtt{S} = (\mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Verify})$, be defined as:*

$$\mathtt{Adv}^{\text{UF-RR-CMA}}_{\mathtt{S},\mathcal{A}}(\lambda) = \Pr[\text{UF-RR-CMA}^{\mathtt{S}}_{\mathcal{A}}(\lambda) \Rightarrow 1]$$

A scheme PKE *is said to be* Φ-UF-RR-CMA *secure, if for all polynomial time Φ-restricted adversaries $\mathcal{A}$, $\mathtt{Adv}^{\text{UF-RR-CMA}}_{\mathtt{S},\mathcal{A}}(\lambda)$ is negligible in the security parameter λ.*

Note that for signatures, no restrictions are placed on the adversary. Furthermore, security is achievable for any function class Φ. In fact, any UF-CMA secure signature scheme can be converted to a scheme achieving UF-RR-CMA security for any function class Φ by de-randomizing the scheme. More concretely, using the folklore technique for de-randomization, a key k for a pseudorandom function $\mathtt{prf}$ could be added to the signing key, and whenever a signature on message m is created, the randomness $r \leftarrow \mathtt{prf}(k, m)$ is used. The UF-CMA security of the scheme is easily seen to be maintained, and as the scheme no longer uses randomness, it is secure against any type of randomness failure. However, as clearly illustrated by the incidents involving DSA signatures, randomness failures might have fatal consequences for schemes that are not de-randomized or deterministic by design. For these, considering the above notion for various function classes Φ is relevant.

In our analysis of RSA-PSS signatures, we will consider a function class Φ which is *continuously hard-to-compute*. We formally define this property as follows.

DEFINITION 9. (CONTINUOUSLY HARD-TO-COMPUTE FUNCTION FAMILIES) *A function family Φ is said to be* continuously ϵ-hard-to-compute *if all polynomial time adversaries $\mathcal{A}$ have advantage $\mathtt{Adv}^{\text{CHTC}}_{\Phi,\mathcal{A}} < \epsilon$, where*

$$Adv^{\text{CHTC}}_{\Phi,\mathcal{A}} = \Pr[\text{CHTC}^{\Phi}_{\mathcal{A}}(\lambda) \Rightarrow 1]$$

If $\epsilon = \epsilon(\lambda)$ is a negligible function, we simply say that Φ is continuously hard-to-compute.

5. RELATED RANDOMNESS SECURITY OF RSA-OAEP ENCRYPTION

In this section, we will prove that, under the RSA and Φ-hiding assumptions, RSA-OAEP is IND-RR-CCA secure in the random oracle model for function families (Φ, Ψ) where Φ is collision resistant and adaptively hard-to-compute with respect to Ψ.

As highlighted in the introduction, the key obstacle to obtaining a proof is that the IND-RR-CCA game permits adversaries to query LR oracle multiple times, and hence, the adversary may obtain two challenge ciphertexts $y_1 = (s_1||t_1)^e$ and $y_2 = (s_2||t_2)^e$ such that $s_1 \oplus s_2 = \Delta$ for known Δ. In this situation, there is no obvious way to create a reduction to the partial-domain one-wayness as in the original proof for RSA-OAEP.

To overcome this difficulty, we use the Φ-hiding assumption, which states that the public key e and a lossy key e' are statistically indistinguishable. More specifically, under a lossy key e', we show that an adversary cannot exploit this knowledge to distinguish challenge ciphertexts from random elements, and hence obtain a reduction to the partial-domain one-wayness. In the proof, we make use of the results by Smith and Zhang [40] regarding the properties of the RSA function under a lossy key (see Section 2, Lemma 2).

THEOREM 1. *Assume that Φ is collision resistant and adaptively hard-to-compute with respect to Ψ, that the RSA function satisfies set partial-domain one-wayness, and that the Φ-hiding assumption holds. Then, RSA-OAEP is* (Φ, Ψ)-IND-RR-CCA *secure. Specifically, for any polynomial time (Φ, Ψ)-restricted and LR-equality-respecting adversary $\mathcal{A}$, the following inequality holds.*

$$\begin{aligned}\mathtt{Adv}^{\text{IND-RR-CCA}}_{\mathtt{OAEP},\mathcal{A}}(\lambda) \leq\ & \frac{q_D}{2^{k_1}} + \frac{q_D q_G}{2^{k_0}} + q_{LR}^2 \cdot \text{CR}^{\Phi}(\lambda) \\ & + q_G \cdot \mathtt{Adv}^{\text{HTC}}_{\Phi,\Psi,\mathcal{A}} + 2q_{LR} \cdot \mathtt{Adv}^{\text{S-PD-OW}}_{c,q_H,\mathcal{A}}(\lambda) + \mathtt{Adv}^{\mathtt{\Phi A}}_{c,\theta,\mathcal{A}}(\lambda) \\ & + 3q_{LR} \cdot (\epsilon + 2^{-k/2+2}),\end{aligned}$$

where q_{LR}, q_G, q_H and q_D are the number of queries to the LR, G, H *and* DEC *oracles, and the parameters satisfy that $k_0 \geq \log N - \log e + 2\log\frac{1}{\epsilon} + 4$, $\theta \geq 4 + \log\frac{1}{\epsilon}$ and $3\theta < k$.*

We first define a sequence of games. Without loss of generality, we assume that $\mathcal{A}$ never repeats an oracle query. We will furthermore use the subscript i to denote the values submitted or computed in the i-th query.

Game_0. This is just the IND-RR-CCA game.

Game_1. We modify the above game by changing the DEC oracle so as to reject all ciphertext y for which the corresponding $t \oplus H(s)$ has not been queried to the G oracle previously.

Game_2. We modify the above game by changing the DEC oracle so as to reject all ciphertexts y for which the corresponding s has not been queried to the H oracle previously.

Game_3. In this game, the response to DEC oracle queries is computed without the challenge private key d. This is possible because the DEC oracle needs to answer only the queries where r and s have been previously queried, and the values $G(r)$ and $H(s)$ are sufficient to decrypt the corresponding ciphertext[5].

Game_4. We modify the above game by changing the challenge public key to a lossy key i.e. $(N, e) \leftarrow \mathsf{RSA}_{c,\theta}^{loss}(\lambda)$.

Game_5. We modify the above game by changing the LR oracle so as to use a uniformly random value g_i^+ instead of $G(\phi_i(r))$.

Game_6. We modify the above game by changing the LR oracle so as to use a uniformly random value $h_i^+ \leftarrow_\$ \{0,1\}^{k_0}$ instead of $H(s_i)$.

Game_7. We modify the above game by changing the LR oracle so as to compute all challenge ciphertexts y_i as $(x_i)^e \bmod N$ where $x_i \xleftarrow{\$} \mathbb{Z}_N$.

We denote by S_i the event $b' = b$ in Game_i. We denote by $\mathcal{O}_i^{\mathrm{H}}$ the set of all x which are queried to the H oracle by the adversary in Game_i, and by $\mathcal{O}_i^{\mathrm{LR,H}}$ the set of all s which are queried to H by Game_i in the response to LR oracle queries. We likewise define $\mathcal{O}_i^{\mathrm{G}}$ and $\mathcal{O}_i^{\mathrm{LR,G}}$ for oracle G. We denote by AskH_i the event $\mathcal{O}_i^{\mathrm{H}} \cap \mathcal{O}_i^{\mathrm{LR,H}} \neq \emptyset$, and by AskG_i the event $\mathcal{O}_i^{\mathrm{G}} \cap \mathcal{O}_i^{\mathrm{LR,G}} \neq \emptyset$.

LEMMA 3. $|\Pr[S_1] - \Pr[S_0]| \leq \frac{q_D}{2^{k_1}}$.

PROOF. The two games Game_1 and Game_2 may differ if there is a DEC query y which is a valid ciphertext while the corresponding $\phi(r)$ has not been queried to G. Since $G(\phi(r))$ is uniformly distributed, equality $[s \oplus G(\phi(r))]_{k_1} = 0^{k_1}$ happens with probability $1/2^{k_1}$. Summing up for all DEC queries, it holds that $|\Pr[S_1] - \Pr[S_0]| \leq \frac{q_D}{2^{k_1}}$. □

LEMMA 4. $|\Pr[S_2] - \Pr[S_1]| \leq \frac{q_D q_G}{2^{k_0}}$.

PROOF. The two games Game_1 and Game_2 may differ if there is a DEC query $y = (s||t)^e \bmod N$ which is a valid ciphertext and the corresponding $H(s) \oplus t$ value has been queried to G, while corresponding s has not been queried to H. Since $H(s)$ is uniformly distributed, $H(s) \oplus t$ has been queried to G with probability less than $q_G/2^{k_0}$. Summing up for all DEC queries, it holds that $|\Pr[S_3] - \Pr[S_2]| \leq \frac{q_D q_G}{2^{k_0}}$. □

LEMMA 5. $\Pr[S_3] = \Pr[S_2]$.

PROOF. The two games are the same from the adversary's point of view. □

[5]Note that the DEC oracle can test whether values r and s correspond to a ciphertext c by setting $t \leftarrow r \oplus H(s)$ and checking whether $c = (s||t)^e \bmod N$.

LEMMA 6. $|\Pr[S_4] - \Pr[S_3]| \leq \mathtt{Adv}_{c,\theta,\mathcal{A}}^{\Phi\mathsf{A}}$.

PROOF. The only difference between Game_3 and Game_4 is that the former uses the honestly generated public key but the latter uses a *lossy key*. We can create a polynomial time algorithm $\mathcal{D}$ such that the statements hold. On receiving (N, e) which is either an injective or a lossy key, $\mathcal{D}$ plays the security game interacting with $\mathcal{A}$. Since the DEC oracle was modified so that it does not need the secret key, $\mathcal{D}$ can answer all DEC queries. Finally, $\mathcal{D}$ outputs the same bit as $\mathcal{A}$. If the key is injective, $\mathcal{A}$ is in Game_4, otherwise, in Game_5. Thus, $|\Pr[S_5] - \Pr[S_4]| \leq \mathtt{Adv}_{c,\theta,\mathcal{A}}^{\Phi\mathsf{A}}$. □

LEMMA 7. $|\Pr[S_5] - \Pr[S_4]| \leq \Pr[\mathsf{AskG}_5] + q_{LR}^2 \cdot \mathrm{CR}^{\Phi}$.

PROOF. Let Coll be the event that, for the set of functions $\{\phi_1, \phi_2, \cdots, \phi_{q_{LR}}\}$ queried by $\mathcal{A}$, there is at least one collision $\phi_i(r^*) = \phi_j(r^*)$ where $\phi_i \neq \phi_j$.

We claim that $\Pr[S_5|\neg(\mathsf{Coll} \vee \mathsf{AskG}_5)] = \Pr[S_4|\neg(\mathsf{Coll} \vee \mathsf{AskG}_5)]$. Because if $\phi_1(r^*), \phi_2(r^*), \cdots, \phi_{q_{LR}}(r^*)$ are all distinct (unless $\phi_i = \phi_j$), and they are not queried to the G oracle, then $G(\phi_i(r^*))$ and g_i^+ are distributed identically.

From the well-known *difference lemma*, we get

$$|\Pr[S_5] - \Pr[S_4]| \leq \Pr[\mathsf{Coll} \vee \mathsf{AskG}_5] \leq \Pr[\mathsf{Coll}] + \Pr[\mathsf{AskG}_5].$$

We claim that $\Pr[\mathsf{Coll}] \leq q_{LR}^2 \cdot \mathrm{CR}^{\Phi}$. For fixed $\phi_i, \phi_j \in \Phi$, the probability that $\phi_i(r^*) = \phi_j(r^*)$ for a uniformly random r^* is at most CR^{Φ}. Summing up for all pairs (ϕ_i, ϕ_j), we get $\Pr[\mathsf{Coll}] \leq \frac{q_{LR}(q_{LR}-1)}{2} \cdot \mathrm{CR}^{\Phi} \leq q_{LR}^2 \cdot \mathrm{CR}^{\Phi}$. This completes the proof. □

LEMMA 8. *The following equations hold.*

$$|\Pr[S_6] - \Pr[S_5]| \leq \Pr[\mathsf{AskH}_6].$$
$$|\Pr[\mathsf{AskG}_6] - \Pr[\mathsf{AskG}_5]| \leq \Pr[\mathsf{AskH}_6].$$

PROOF. The two games Game_5 and Game_6 are the same whenever $\mathcal{O}_5^{\mathrm{H}} \cap \mathcal{O}_5^{\mathrm{LR,H}} = \mathcal{O}_6^{\mathrm{H}} \cap \mathcal{O}_6^{\mathrm{LR,H}} = \emptyset$. Thus, we get

$$\Pr[S_6|\neg\mathsf{AskH}_6] = \Pr[S_5|\neg\mathsf{AskH}_5] \text{ and}$$
$$\Pr[\mathsf{AskG}_6|\neg\mathsf{AskH}_6] = \Pr[\mathsf{AskG}_5|\neg\mathsf{AskH}_5].$$

We complete the proof by using the difference lemma. □

LEMMA 9. $\Pr[\mathsf{AskG}_6] \leq q_G \cdot \mathtt{Adv}_{\Phi,\Psi,\mathcal{A}}^{\mathrm{HTC}}$.

PROOF. We will show that if there is an adversary $\mathcal{A}$ which can cause the event AskG_6 with probability ϵ, then we can create a reduction algorithm $\mathcal{B}$ that can break the adaptively hard-to-compute property of Φ with success probability ϵ/q_G. $\mathcal{B}$ plays the role of Game_6 without knowing the randomness r^*. $\mathcal{B}$ can answer all ENC queries since $\mathcal{B}$ has access to the GET oracle, which returns $\psi(r^*)$. (Note that, in Game_6, r^* is not required to respond to LR oracle queries.) Finally, $\mathcal{B}$ outputs $x \leftarrow_\$ \mathcal{O}_6^{\mathrm{G}}$. By assumption, the event $\mathcal{O}_6^{\mathrm{G}} \cap \mathcal{O}_6^{\mathrm{LR,G}} \neq \emptyset$ occurs with probability ϵ. Thus, the probability that $x \in \mathcal{O}_6^{\mathrm{LR,G}}$ occurs, which is equivalent to $\mathtt{Adv}_{\Phi,\Psi,\mathcal{A}}^{\mathrm{HTC}}$, is at least ϵ/q_G. This completes the proof. □

LEMMA 10. *The following equations hold.*

$$|\Pr[S_7] - \Pr[S_6]| \leq q_{LR} \cdot (\epsilon + 2^{-k/2+2}).$$
$$|\Pr[\mathsf{AskH}_7] - \Pr[\mathsf{AskH}_6]| \leq q_{LR} \cdot (\epsilon + 2^{-k/2+2}).$$

PROOF. Game_7 is identical to Game_6 except that the challenge ciphertexts $y_i = (s_i||t_i)^e$ is replaced with x^e where

$x \leftarrow \mathbb{Z}_N^*$ is a uniformly random element. In Game_6, we can regard the plaintext $(s_i||t_i)$ as a sample from an arithmetic progression with length 2^{k_0} since t_i is uniformly distributed. More specifically, for $i \in \{1, \cdots, q_{LR}\}$, let Y_{s_i} be a set of the arithmetic progression $Y_{s_i} = \{(s_i||t_i) : t_i \leftarrow \{0,1\}^{k_0}\}$. From Lemmas 11 and 12 stated in the following, we have

$$SD(y^e \bmod N, z^e \bmod N) \leq \epsilon + 2^{-k/2+2},$$

where $y \leftarrow_\$ Y_{s_i}$ and $z \leftarrow_\$ \mathbb{Z}_N$. Summing up for all LR queries, we can conclude that $|\Pr[S_7] - \Pr[S_6]| \leq q_{LR} \cdot (\epsilon + 2^{-k/2+2})$ and $|\Pr[\mathsf{AskH}_7] - \Pr[\mathsf{AskH}_6]| \leq q_{LR} \cdot (\epsilon + 2^{-k/2+2})$. □

The following two lemmas completes the proof of Lemma 8. The latter lemma (Lemma 12) was originally stated by Smith and Zhang in [40, Theorem 8].[6]

LEMMA 11. *$SD(x, x') \leq 2^{-k/2+2}$, where $x \leftarrow_\$ \mathbb{Z}_N$ and $z \leftarrow_\$ \mathbb{Z}_N^*$.*

PROOF. Since $|\mathbb{Z}_N^*| = \phi(N) = N - p - q + 1$, there must be at least $2^{k-1} - p - q \geq 2^{k-1} - 2^{k/2+1}$ elements of $\mathbb{Z}_N^*$ which are less than 2^{k-1} (recall that p and q are $k/2$-bit primes). Thus, the statistical distance between the two distributions is at most $2^{k/2+1}/2^{k-1} = 2^{-k/2+2}$. □

LEMMA 12. *Let $(N, e) \leftarrow \mathsf{RSA}_{c,\theta}^{loss}(\lambda)$, where $3\theta \leq k$ for the modulus length k. Let $P_K = \{\sigma i + \tau : i = 0, 1, \cdots, K-1\}$, where σ, n are co-prime. Assume that $\log K \geq \log N - \log e + 2\log\frac{1}{\epsilon} + 4$ and $\theta \geq 4 + \log\frac{1}{\epsilon}$ for some $0 < \epsilon < 1$. Then,*

$$SD(y^e \bmod N, z^e \bmod N) \leq \epsilon,$$

where $y \leftarrow_\$ P_K$ and $z \leftarrow_\$ \mathbb{Z}_N^*$.

PROOF. From the definition of $\mathsf{RSA}_{c,\theta}^{loss}$, e and $q-1$ are co-prime, and $q > p$ and $p = 1 \bmod e$ are satisfied. The condition $K > q$ is also satisfied from the following inequality:

$$\log K \geq \log N - \log e + 2\log\frac{1}{\epsilon} + 4 \geq \log q.$$

Therefore, all conditions in Lemma 2 are satisfied. We will show that each of the four terms is less than $\frac{\epsilon}{4}$. We will use (A) and (B) to denote the conditions $\log K \geq \log N - \log e + 2\log\frac{1}{\epsilon} + 4$ and $\theta \geq 4 + \log\frac{1}{\epsilon}$, respectively.

$$\begin{aligned}
\log\frac{3q}{K} &= \log 3 + \log q - \log K \\
&\leq \log 3 + \log q - \left(\log N - \log e + 2\log\frac{1}{\epsilon} + 4\right) \\
&\leq \log 3 + \log q - \left(\log q + \log\frac{1}{\epsilon} + 4\right) = \log\frac{\epsilon}{4}.
\end{aligned}$$

$$\begin{aligned}
\log\frac{2p}{q-1} &= 1 + \log p - \log(q-1) \leq 1 + \log p - \log q \\
&= 1 - \theta \leq 2 - \left(4 + \log\frac{1}{\epsilon}\right) = \log\frac{\epsilon}{4}.
\end{aligned}$$

$$\begin{aligned}
\log\frac{2}{p-1} &= 1 - \log(p-1) \leq 1 - \log p = 1 - \frac{k-\theta}{2} \\
&\leq 1 - \theta = 1 - \left(4 + \log\frac{1}{\epsilon}\right) = \log\frac{\epsilon}{4}.
\end{aligned}$$

$$\log\sqrt{\frac{N}{eK}} = \frac{1}{2}(\log N - \log e - \log K) \leq \log\epsilon - 2 = \log\frac{\epsilon}{4}.$$

This completes the proof. □

LEMMA 13. *$\Pr[S_7] = \frac{1}{2}$.*

PROOF. We observe that the input to the adversary follows a distribution that does not depend on the bit b. Accordingly, $\Pr[S_7] = \frac{1}{2}$. □

LEMMA 14. *$\Pr[\mathsf{AskH}_7] \leq q_{LR} \cdot \mathbf{Adv}_{c,q_H,\mathcal{A}}^{\text{S-PD-OW}}$.*

PROOF. Let f be the RSA function $f(x) = x^e \bmod n$. Given $y = f(s||t)$ for random $(s||t) \xleftarrow{\$} \mathbb{Z}_n^*$, the reduction algorithm $\mathcal{B}$ plays the role of Game_7 interacting with $\mathcal{A}$. At the beginning of the game, $\mathcal{B}$ randomly chooses $i \in \{1, \cdots, q_{LR}\}$. For the i-th LR query, it returns y as the response of the oracle. Finally, $\mathcal{B}$ simply outputs all elements in the $\mathcal{O}_6^{\mathrm{H}}$. The probability that $\mathcal{B}$ breaks the q_H-set partial-domain one-wayness is greater than $\frac{1}{q_{LR}}\Pr[\mathsf{AskH}_7]$. □

The above Lemmas yield Theorem 1.

6. RELATED RANDOMNESS SECURITY OF RSA-PSS SIGNATURES

We will now turn our attention to the RSA-PSS signature scheme. Firstly, we consider related randomness security for any function family Φ. As already highlighted in the introduction, the original proof [9] will no longer work in this case, as the randomness used when signing is no longer unpredictable to the adversary. Furthermore, as an adversary is allowed to used constant functions, which will essentially make the signature scheme deterministic, the impossibility results by Coron [16] implies that a reduction with a security loss less than q_s is not achievable. However, we show a reduction that essentially meets this bound. The proof of the following lemma builds upon the techniques from [15] used to analyze the FDH signature scheme.

THEOREM 2. *Assume the RSA problem is hard with respect to* RSAGen. *Then the* PSS *signature scheme is* Φ-UF-RR-CMA *secure for any function family Φ in the random oracle model. Specifically, for any polynomial time adversary $\mathcal{A}$ against* PSS*, there exist a polynomial time algorithm $\mathcal{B}$ such that*

$$\begin{aligned}
&\mathbf{Adv}_{\mathtt{PSS},\mathcal{A}}^{\text{UF-RR-CMA}}(\lambda) \approx \\
&\quad e \cdot q_s \cdot \mathbf{Adv}_{\mathtt{RSAGen},\mathcal{B}}^{\text{RSA}}(\lambda) + (q_s + q_h) \cdot 2^{-k/2+2} \\
&\quad + e \cdot \cdot q_s \cdot (q_s + q_h) \cdot (\frac{1}{2} + 2^{-k/2+2})^{k_0} \\
&\quad + e \cdot \cdot q_s \cdot (q_s + q_h)^2 \cdot (2^{-k_1} + 2^{-k/2+2})
\end{aligned}$$

for large values of q_s, where q_s and q_h denotes the number of sign and hash queries made by $\mathcal{A}$, respectively, and e is the base of the natural logarithm.

PROOF. Given an adversary $\mathcal{A}$ that succeeds in breaking the UF-RR-CMA security of PSS with probability ϵ', we construct an algorithm $\mathcal{S}$ that solves the RSA problem with respect to RSAGen with probability ϵ as given in the theorem. $\mathcal{S}$ is constructed as follows:

Firstly, $\mathcal{S}$ receives as input values (N, e) and a challenge y; the goal of $\mathcal{S}$ is to compute x such that $y = x^e \bmod N$. $\mathcal{S}$ sets $pk \leftarrow (N, e)$, picks randomness $r^* \leftarrow_\$ \{0,1\}^{k_0}$, and runs $\mathcal{A}$ with input pk. While $\mathcal{A}$ is running, $\mathcal{S}$ will respond to

[6] The details of the proof is not given in [40]. We prove the lemma under the additional condition $3\theta \leq k$.

SIGN, H, and G oracle queries as described below. Without loss of generality, we assume that $\mathcal{A}$ never repeats an oracle query. We will furthermore use the subscript i to denote the values submitted or computed in the i-th query.

H queries On input $m_i||r_i$, if there exists $j < i$ such that $m_i||r_i = m_j||r_j$, $\mathcal{S}$ returns the previous oracle answer h_j. Otherwise, $\mathcal{S}$ picks a random $x_i \leftarrow_\$ \mathbb{Z}_N^*$. Then, with probability p (where p will be determined later), $\mathcal{S}$ proceeds as follows: $\mathcal{S}$ computes $y_i \leftarrow y(x_i)^e$ and parses y_i as $b||w_i||r_i'||\gamma_i$. If $b \neq 0$, $\mathcal{S}$ will sample a new y_i (by picking a new x_i) until a y_i value with $b = 0$ is obtained. To ensure that $\mathcal{S}$ remains a polynomial time algorithm, $\mathcal{S}$ will abort if a suitable y_i value is not obtained after k_0 trials. Then, for the obtained $y_i = 0||w_i||r_i'||\gamma_i$, if there exists a $j < i$ such that $w_i = w_j$, $\mathcal{S}$ aborts. Finally, $\mathcal{S}$ sets $h_i \leftarrow w_i$ and $g_i \leftarrow (r_i' \oplus r_i)||\gamma_i$ (this will set $G(w_i) = g_i$), and returns h_i. We will refer to queries handled in this way as type I queries.

On the other hand, with probability $1 - p$, $\mathcal{S}$ proceeds as follows: $\mathcal{S}$ computes $y_i \leftarrow x_i^e$ and parses y_i as $b||w_i||r_i'||\gamma_i$. As above, if $b \neq 0$, $\mathcal{S}$ picks a new y_i until this is the case (but aborts after k_0 trials), and furthermore aborts if $w_i = w_j$ for a $j < i$ for the obtained y_i value. Finally $\mathcal{S}$ sets $h_i \leftarrow w_i$ and $g_i \leftarrow (r_i' \oplus r_i)||\gamma_i$ (this will set $G(w_i) = g_i$), and returns h_i. We refer to queries handled in this way as a type II queries.

G queries On input w_i, if there exists $j < i$ such that $w_i = w_j$, $\mathcal{S}$ returns the corresponding g_j value. Otherwise, $\mathcal{S}$ picks random $g_i \leftarrow_\$ \{0,1\}^{k-k_1-1}$, and returns g_i.

Sign queries On input (m_i, ϕ_i), $\mathcal{S}$ first computes $r_i \leftarrow \phi_i(r^*)$, and makes the query $H(m_i||r_i)$ if $\mathcal{A}$ has not already done so. Let the H query corresponding to $m_i||r_i$ be the j-th query. If this is a type I query, $\mathcal{S}$ aborts. Otherwise, $\mathcal{S}$ simply returns $\sigma \leftarrow x_j$. Note that due to the way $\mathcal{S}$ responds to H type II queries, σ is a valid signature on m_i using randomness $\phi_i(r^*)$. Specifically, $\sigma^e = x_j^e = 0||w_j||r_j'||\gamma_j$ where $\phi_i(r^*) = r_j' \oplus G_1(w_j)$, $H(m_i||\phi_i(r^*)) = w_j$, and $G_2(w_j) = \gamma_j$.

Assume that $\mathcal{S}$ does not abort and that $\mathcal{A}$ produces a valid forgery σ^* on a message m^*. Let $(\sigma^*)^e = 0||w^*||r'||\gamma^*$ (note that the most significant bit must be 0 for the forgery to be valid) and let $r^* \leftarrow r' \oplus G_1(w^*)$. Without loss of generality, we assume that $m^*||r^*$ was queried to H. If the query is a type II query, $\mathcal{S}$ aborts. Otherwise, $H(m^*||r^*)$ must have been a type I query. Let this query be the j^*-th query. Then we must have that $\sigma^* = (0||w^*||r'||\gamma^*)^d = (y \cdot (x_{j^*})^e)^d = y^d \cdot x_{j^*}$, and $\mathcal{S}$ can compute $y^d = \sigma^*/x_{j^*}$ which is the solution to the given RSA problem. This completes the description of $\mathcal{S}$.

It remains to estimate the success probability of $\mathcal{S}$. Let $\texttt{Forge}$ denote the event that $\mathcal{A}$ produces a valid forgery, let $\mathtt{A}_1$ denote the event that $\mathcal{S}$ aborts due to k_0 y_i values being sampled in a H or a SIGN query, all having the most significant bit set to 1 (i.e. $b = 1$), let $\mathtt{A}_2$ denote the event that $\mathcal{S}$ aborts due $w_i = w_j$ in a H or SIGN query for a $j < i$, and finally let $\mathtt{A}_3$ denote the event that $\mathcal{S}$ aborts due to a SIGN query or the forgery corresponding to a wrong query type. From the above, it follows that

$$\begin{aligned}\mathtt{Adv}^{\mathrm{RSA}}_{\mathtt{RSAGen},\mathcal{S}} &= \Pr[\texttt{Forge} \wedge \overline{\mathtt{A}_1} \wedge \overline{\mathtt{A}_2} \wedge \overline{\mathtt{A}_3}] \\ &\geq \Pr[\texttt{Forge}|\overline{\mathtt{A}_1} \wedge \overline{\mathtt{A}_2} \wedge \overline{\mathtt{A}_3}] \cdot \Pr[\overline{\mathtt{A}_3}] - \Pr[\mathtt{A}_1] - \Pr[\mathtt{A}_2].\end{aligned}$$

We proceed by showing the following lemmas.

LEMMA 15. $\Pr[\texttt{Forge}|\overline{\mathtt{A}_1} \wedge \overline{\mathtt{A}_2} \wedge \overline{\mathtt{A}_3}] \geq \epsilon - (q_s + q_h) \cdot 2^{-k/2+2}$.

PROOF. Note that the view of $\mathcal{A}$ in the simulation provided by $\mathcal{S}$ would have been identical to the view of $\mathcal{A}$ in the UF-RR-CMA game had the responses to the H and G queries (i.e. the h_i and g_i values) been uniformly distributed. In this case, $\mathcal{A}$ is guaranteed to produce a forgery with probability ϵ. We will now argue that the responses are statistically close to uniform.

Firstly, note that had the y_i values been sampled from $\mathbb{Z}_{2^{k-1}}$, the h_i and g_i values would have been uniformly distributed. Secondly, note that in the simulation, the y_i values correspond to uniformly chosen elements of $\mathbb{Z}_N^*$ less than 2^{k-1}, as the x_i values are sampled from $\mathbb{Z}_N^*$, $y \in \mathbb{Z}_N^*$, and the RSA function is a permutation over $\mathbb{Z}_N^*$. Finally, due to Lemma 11 and the fact that $\mathcal{S}$ use $q_s + q_h$ samples, $\mathcal{A}$ will produce a forgery with probability at least $\epsilon - (q_s + q_h) \cdot 2^{-k/2+2}$. □

LEMMA 16. $\Pr[\mathtt{A}_1] \leq (q_s + q_h)(\frac{1}{2} + 2^{-k/2+2})^{k_0}$.

PROOF. Note that the y_i elements sampled by $\mathcal{S}$ are uniformly distributed in $\mathbb{Z}_N^*$. Also note that had the values been sampled from $\mathbb{Z}_N$, the probability that the most significant bit is 1 would have been less than $1/2$. However, as the statistical difference between sampling from $\mathbb{Z}_N$ and $\mathbb{Z}_N^*$ is $(N - \phi(N))/N \leq 2^{k/2+1}/2^{k-1} = 2^{-k/2+2}$, we have that $\mathcal{S}$ aborts in a single query with probability at most $(1/2 + 2^{-k/2+2})^{k_0}$. As there are $q_s + q_h$ queries, the lemma follows. □

LEMMA 17. $\Pr[\mathtt{A}_2] \leq (q_s + q_h)^2(2^{-k_1} + 2^{-k/2+2})$.

PROOF. Using similar observations as in Lemma 15 above, we can see that the statistical difference between sampling w_i from uniform and as done by $\mathcal{S}$, is at most $2^{-k/2+2}$. Hence, the chance that w_i collides with any of the previous values, is at most $(q_s + q_h)(2^{-k_1} + 2^{-k/2+2})$. Hence, considering all queries, we have $\Pr[\mathtt{A}_2] \leq (q_s + q_h)^2(2^{-k_1} + 2^{-k/2+2})$. □

LEMMA 18. $\Pr[\overline{\mathtt{A}_3}] \approx (e \cdot q_s)^{-1}$ *for the optimal choice of p and large values of q_s, where e is the base of the natural logarithm.*

PROOF. Note that for $\mathcal{S}$ not to abort based on the type of query, all SIGN queries has to correspond to type II queries, and the forgery has to correspond to a type I query. From the above description, it is easily seen that this will happen with probability $p^{q_s}(1 - p)$. This expression is maximized for $p = 1 - 1/(q_s + 1)$. Inserting this value in the former expression yields that $\Pr[\overline{\mathtt{A}_3}] \approx e \cdot q_s$ for large values of q_s. □

Combining the above expression for $\mathtt{Adv}^{\mathrm{RSA}}_{\mathtt{RSAGen},\mathcal{S}}$ with the above lemmas yields the theorem.

□

We will now show that by restricting the function class Φ and the adversary, a tight reduction can be obtained. Specifically, we will consider a function class Φ which is continuously hard-to-compute, and an adversary that will query a new related randomness function ϕ in each signature query.

We will refer to this type of adversary as a *unique randomness query respecting* adversary. Note that the combination of these assumptions will imply that the adversary cannot predict the randomness value using in a signature query, which allows us to obtain a tight proof, assuming the collision resistance and continuously hard-to-compute properties of Φ are sufficiently strong. However, we stress that these assumptions are relatively strong, and that, for example, repeated randomness attacks are not captured when making these. Nevertheless, our result shows that tight security is achieved, even for maliciously biased randomness values, as long as these are not predictable by the adversary.

THEOREM 3. *Assume that Φ is collision resistant and continuously hard-to-compute, and that the RSA problem is hard with respect to* `RSAGen`. *Then* `PSS` *is* UF-RR-CMA *secure for all unique randomness query respecting adversaries, and the reduction to the RSA problem with respect to* `RSAGen` *is tight. Specifically, for all polynomial time and unique randomness query respecting adversaries $\mathcal{A}$ against* `PSS`, *there exists algorithms $\mathcal{B}_1, \mathcal{B}_2$ such that*

$$\begin{aligned}\mathtt{Adv}^{\text{UF-RR-CMA}}_{\mathtt{PSS},\mathcal{A}}(\lambda) \leq \mathtt{Adv}^{\text{RSA}}_{\mathtt{RSAGen},\mathcal{B}_1}(\lambda) + (q_s+q_h)\cdot 2^{-k/2+2} \\ + (q_s+q_h)\cdot(\frac{1}{2}+2^{-k/2+2})^{k_0} \\ + (q_s+q_h)^2\cdot(2^{-k_1}+2^{-k/2+2}) \\ + q_s\cdot q_h\cdot \mathtt{Adv}^{\text{CHTC}}_{\Phi,\mathcal{B}_2}(\lambda) + q_s^2\cdot \text{CR}^{\Phi}(\lambda)\end{aligned}$$

Note that for the reduction to be tight, the continuously hard-to-compute property and the collision resistance of Φ needs to be sufficiently strong, i.e. $q_s \cdot q_h \cdot \mathtt{Adv}^{\text{CHTC}}_{\Phi,\mathcal{B}_2}$ and $q_s^2 \cdot \text{CR}^{\Phi}$ has to be negligible in the security parameter. Due to space restriction, the proof of the above theorem is not included here.

7. CONCLUSION

In this paper, we have provided a detailed analysis of the robustness of the RSA-OAEP encryption scheme and the RSA-PSS signature scheme, against related randomness attacks. Specifically, we have shown that under the RSA and Φ-hiding assumptions, RSA-OAEP encryption remains secure against related randomness attacks for function families (Φ, Ψ) where Φ is collision resistant and hard-to-compute with respect to Ψ. This implies that RSA-OAEP is secure for a large class of related randomness attacks if the recipient is honest, and remains secure even if malicious recipients are additionally considered, as long the randomness used for the honest recipient cannot be efficiently computed from the randomness used for the malicious recipients. However, we highlight that, since RSA-OAEP is randomness recovering, security is not guaranteed if highly correlated randomness is used for encryption for both malicious and honest recipients. Furthermore, we have shown that under the RSA assumption, the RSA-PSS signature scheme remains related randomness secure for any function family Φ, albeit with a non-tight security reduction, but if Φ is additionally assumed to be continuously hard-to-compute and the attack is not capable of forcing the use of repeated randomness, a tight reduction is possible. Our results show that, compared to other widely available and standardized schemes, RSA-OAEP and RSA-PSS provides better protection when used in environments where (potentially maliciously caused) randomness failures might occur.

8. REFERENCES

[1] ABDALLA, M., BELLARE, M., AND ROGAWAY, P. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *CT-RSA* (2001), vol. 2020 of *Lecture Notes in Computer Science*, Springer, pp. 143–158.

[2] BAUER, A., CORON, J., NACCACHE, D., TIBOUCHI, M., AND VERGNAUD, D. On the broadcast and validity-checking security of pkcs#1 v1.5 encryption. In *ACNS* (2010), pp. 1–18.

[3] BELLARE, M., BOLDYREVA, A., KUROSAWA, K., AND STADDON, J. Multirecipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security. *IEEE Trans. Information Theory 53*, 11 (2007), 3927–3943.

[4] BELLARE, M., BOLDYREVA, A., AND O'NEILL, A. Deterministic and Efficiently Searchable Encryption. In *CRYPTO* (2007), vol. 4622 of *Lecture Notes in Computer Science*, Springer, pp. 535–552.

[5] BELLARE, M., BOLDYREVA, A., AND STADDON, J. Randomness Re-use in Multi-recipient Encryption Schemeas. In *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings* (2003), Y. Desmedt, Ed., vol. 2567 of *Lecture Notes in Computer Science*, Springer, pp. 85–99.

[6] BELLARE, M., BRAKERSKI, Z., NAOR, M., RISTENPART, T., SEGEV, G., SHACHAM, H., AND YILEK, S. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT* (2009), pp. 232–249.

[7] BELLARE, M., AND ROGAWAY, P. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM CCS* (1993), ACM, pp. 62–73.

[8] BELLARE, M., AND ROGAWAY, P. Optimal Asymmetric Encryption. In *EUROCRYPT* (1994), vol. 950 of *Lecture Notes in Computer Science*, Springer, pp. 92–111.

[9] BELLARE, M., AND ROGAWAY, P. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *EUROCRYPT* (1996), vol. 1070 of *Lecture Notes in Computer Science*, Springer, pp. 399–416.

[10] BELLARE, M., AND TACKMANN, B. Nonce-based cryptography: Retaining security when randomness fails. In *EUROCRYPT* (2016), pp. 729–757.

[11] BENDEL, M. Hackers describe PS3 security as epic fail, gain unrestricted access, 2011. http://www.exophase.com/20540/hackers-describe-ps3-security-as-epic-fail-gain-unrestricted-access/.

[12] BERNSTEIN, D. J., CHANG, Y.-A., CHENG, C.-M., CHOU, L.-P., HENINGER, N., LANGE, T., AND VAN SOMEREN, N. Factoring RSA keys from certified smart cards: Coppersmith in the wild. Cryptology ePrint Archive, Report 2013/599, 2013. http://eprint.iacr.org/.

[13] BITCOIN.ORG. Android security vulnerability, 2013. http://bitcoin.org/en/alert/2013-08-11-android.

[14] CHECKOWAY, S., NIEDERHAGEN, R., EVERSPAUGH, A., GREEN, M., LANGE, T., RISTENPART, T., BERNSTEIN, D. J., MASKIEWICZ, J., SHACHAM, H., AND FREDRIKSON, M. On the Practical Exploitability of Dual EC in TLS Implementations. In *USENIX*

Security Symposium (2014), USENIX Association, pp. 319–335.

[15] Coron, J.-S. On the Exact Security of Full Domain Hash. In *CRYPTO* (2000), vol. 1880 of *Lecture Notes in Computer Science*, Springer, pp. 229–235.

[16] Coron, J.-S. Optimal Security Proofs for PSS and Other Signature Schemes. In *EUROCRYPT* (2002), vol. 2332 of *Lecture Notes in Computer Science*, Springer, pp. 272–287.

[17] Cramer, R., and Shoup, V. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO* (1998), vol. 1462 of *Lecture Notes in Computer Science*, Springer, pp. 13–25.

[18] Debian. Debian Security Advisory DSA-1571-1: OpenSSL – predictable random number generator, 2008. http://www.debian.org/security/2008/dsa-1571.

[19] Dodis, Y., Pointcheval, D., Ruhault, S., Vergnaud, D., and Wichs, D. Security analysis of pseudo-random number generators with input: /dev/random is not robust. In *ACM CCS* (2013), ACM, pp. 647–658.

[20] Dorrendorf, L., Gutterman, Z., and Pinkas, B. Cryptanalysis of the random number generator of the Windows operating system. *ACM Trans. Inf. Syst. Secur. 13*, 1 (2009).

[21] Fujisaki, E., Okamoto, T., Pointcheval, D., and Stern, J. RSA-OAEP Is Secure under the RSA Assumption. In *CRYPTO* (2001), vol. 2139 of *Lecture Notes in Computer Science*, Springer, pp. 260–274.

[22] Fujisaki, E., Okamoto, T., Pointcheval, D., and Stern, J. RSA-OAEP Is Secure under the RSA Assumption. *J. Cryptology 17*, 2 (2004), 81–104.

[23] Gamal, T. E. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *CRYPTO* (1984), vol. 196 of *Lecture Notes in Computer Science*, Springer, pp. 10–18.

[24] Gutterman, Z., and Malkhi, D. Hold Your Sessions: An Attack on Java Session-Id Generation. In *CT-RSA* (2005), pp. 44–57.

[25] Gutterman, Z., Pinkas, B., and Reinman, T. Analysis of the Linux Random Number Generator. In *IEEE Symposium on Security and Privacy* (2006), pp. 371–385.

[26] Heninger, N., Durumeric, Z., Wustrow, E., and Halderman, J. A. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *USENIX Security Symposium* (Aug. 2012).

[27] IEEE. IEEE p1363 – standard specifications for public-key cryptography. http://grouper.ieee.org/groups/1363/.

[28] IEEE. IEEE p1363a – standard specifications for public-key cryptography: Additional techniques. http://grouper.ieee.org/groups/1363/.

[29] ISO/IEC 18033-2. Information technology – security techniques – encryption algorithms – part 2: Asymmetric ciphers.

[30] Kamara, S., and Katz, J. How to encrypt with a malicious random number generator. In *FSE* (2008), pp. 303–315.

[31] Lenstra, A. K., Hughes, J. P., Augier, M., Bos, J. W., Kleinjung, T., and Wachter, C. Public keys. In *CRYPTO* (2012), pp. 626–642.

[32] Michaelis, K., Meyer, C., and Schwenk, J. Randomly Failed! The State of Randomness in Current Java Implementations. In *CT-RSA* (2013), pp. 129–144.

[33] National Institute of Standards and Technology (NIST). FIPS PUB 186-4: Digital signature standard (DSS), 2013. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

[34] Paterson, K. G., Schuldt, J. C. N., and Sibborn, D. L. Related randomness attacks for public key encryption. In *PKC* (2014), pp. 465–482.

[35] Pinto, A., Poettering, B., and Schuldt, J. C. N. Multi-recipient encryption, revisited. In *9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '14, Kyoto, Japan - June 03 - 06, 2014* (2014), S. Moriai, T. Jaeger, and K. Sakurai, Eds., ACM, pp. 229–238.

[36] Ristenpart, T., and Yilek, S. When Good Randomness Goes Bad: Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography. In *NDSS* (2010).

[37] RSA Laboratories. PKCS#1 v2.2: RSA cryptography standard, 2012. http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm.

[38] Schnorr, C.-P. Efficient Identification and Signatures for Smart Cards. In *CRYPTO* (1989), vol. 435 of *Lecture Notes in Computer Science*, Springer, pp. 239–252.

[39] Shoup, V. A proposal for an ISO standard for public key encryption. *IACR Cryptology ePrint Archive, Report 2001/112* (2001). http://eprint.iacr.org/2001/112.

[40] Smith, A. D., and Zhang, Y. On the regularity of lossy RSA - improved bounds and applications to padding-based encryption. In *TCC, Part I* (2015), pp. 609–628.

[41] Yuen, T. H., Zhang, C., Chow, S. S. M., and Yiu, S. Related randomness attacks for public key cryptosystems. In *ACM ASIACCS* (2015), pp. 215–223.