# Optimal Security Proofs for Full Domain Hash, Revisited

Saqib A. Kakvi and Eike Kiltz

Faculty of Mathematics, Horst Görtz Institute for IT Security
Ruhr University Bochum
{saqib.kakvi,eike.kiltz}@rub.de

**Abstract.** RSA Full Domain Hash (RSA-FDH) is a digital signature scheme, secure again chosen message attacks in the random oracle model. The best known security reduction from the RSA assumption is non-tight, i.e., it loses a factor of $q_s$, where $q_s$ is the number of signature queries made by the adversary. It was furthermore proved by Coron (EUROCRYPT 2002) that a security loss of $q_s$ is optimal and cannot possibly be improved. In this work we uncover a subtle flaw in Coron's impossibility result. Concretely, we show that it only holds if the underlying trapdoor permutation is *certified*. Since it is well known that the RSA trapdoor permutation is (for all practical parameters) not certified, this renders Coron's impossibility result moot for RSA-FDH. Motivated by this, we revisit the question whether there is a tight security proof for RSA-FDH. Concretely, we give a new tight security reduction from a stronger assumption, the Phi-Hiding assumption introduced by Cachin et al (EUROCRYPT 1999). This justifies the choice of smaller parameters in RSA-FDH, as it is commonly used in practice. All of our results (positive and negative) extend to the probabilistic signature scheme PSS.

## 1 Introduction

Among all digital signatures schemes based on the RSA problem, arguably among the most important ones is RSA Full Domain Hash (RSA-FDH) by Bellare and Rogaway [3]. It is extensively used in a wide variety of applications, and serves as the basis of several existing standards such as PKCS #1 [26]. It has been demonstrated by means of a security reduction that, in the random oracle model [2], breaking the security of RSA-FDH (in the sense of existential unforgeability against chosen message attacks) is asymptotically at least as hard as inverting the RSA function.

The seminal work by Bellare and Rogaway introduced the concept of concrete security [3] and highlights the importance of considering the tightness of a security reduction. A security reduction is *tight* if an adversary breaking the scheme yields another adversary breaking the underlying hardness assumption with roughly the same success probability and running time. The current state of RSA-FDH is as follows. Coron's reduction [11] (which improves on earlier results by Bellare and Rogaway [3]) bounds the probability $\varepsilon$ of breaking RSA-FDH in time $t$ by $\varepsilon' \cdot q_s$, where $\varepsilon'$ is the probability of inverting RSA in time

D. Pointcheval and T. Johansson (Eds.): EUROCRYPT 2012, LNCS 7237, pp. 537–553, 2012.

$t' \approx t$ and $q_s$ is the number of signature queries by the forger. In other words, the security reduction for RSA-FDH is loose (it loses a factor of $q_s$), which can have great negative impact on the practical parameter choices of the scheme. As a numerical example, for 80 bits of security and assuming that an adversary can make up to $q_s = 2^{30}$ signature queries [3], one should use a large enough RSA modulus $N$ such that inverting the RSA function cannot be done in fewer than $2^{110} = 2^{30} \cdot 2^{80}$ operations. Concretely, using the recommended key sizes from [28], this leads to a modulus $N$ of about 2432 bits, compared to 1248 bits if RSA-FDH had a tight reduction. We further refer to [9] for a recent discussion on the practical impact of non-tight security reductions in cryptography.

It is an interesting question of great practical impact whether or not there is a tight security reduction for general FDH signatures (based on any trapdoor permutation TDP) and, in particular, for RSA-FDH. Unfortunately, this question was already answered to the negative exactly 10 years ago by Coron [12,13] who showed that the above non-tight security reduction is essentially *optimal*. That is, every security reduction from inverting the TDP (i.e., RSA in the case of RSA-FDH) to breaking FDH signatures will inevitably lose a $q_s$ factor. Consequently, for RSA-FDH a large RSA modulus seems unavoidable to obtain a meaningful security proof.

### 1.1 An Overview of Our Results

Revisiting Coron's Impossibility Result. We uncover a gap in Coron's result about the impossibility of a tight security reduction for FDH signatures [13]. As acknowledged by the author of [13], his impossibility result only holds if the underlying trapdoor permutation (i.e., RSA in the case of RSA-FDH) is a *certified trapdoor permutation*. A trapdoor permutation is certified [5,22] if one can publicly verify that it actually defines a permutation. Unfortunately, the RSA trapdoor permutation is not known to be certified (unless the public exponent $e$ is prime and larger than the modulus $N$) and therefore the impossibility result does not apply any longer to the case of RSA-FDH.

A tight Security Reduction for FDH signatures. In light of the above, we revisit the question whether there exists a tight security reduction for FDH signatures. Unfortunately, we are not able to give such a tight security reduction from the assumption that the TDP is one-way, but from a stronger (yet still non-interactive) assumption, namely that the TDP is lossy (in the sense of Peikert and Waters [25]). Our main result (Theorem 8) shows that there is a *tight* security reduction from the lossiness of the TDP to breaking security of FDH, in the random oracle model.

Applications to RSA-FDH. Recently, Kiltz et al. [20] showed that the RSA trapdoor permutation is lossy under the under the $\Phi$-Hiding Assumption. The $\Phi$-Hiding Assumption was introduced by Cachin, Micali, and Stadler in 1999 [8] and it states that, roughly, $(N, e)$ with $\gcd(\varphi(N), e) = 1$ and $e < N^{1/4}$ is computationally indistinguishable from $(N', e')$ with $e' \mid \varphi(N')$. (Here $\varphi(N)$ is Euler's totient function.) This give a tight security reduction for RSA-FDH from the $\Phi$-Hiding Assumption. We remark that the $\Phi$-Hiding Assumption (or, more generally, the assumption that RSA is lossy) is a stronger assumption than the assumption that RSA is one-way. However, it dates back to 1999 [8] and has ever since been used in a number of cryptographic applications (e.g., [20,7,14,16,23,18]). It has been cryptalanyzed (e.g. [8,7,27]) and for the parameters of interest there is no known algorithm that breaks it without first factoring the modulus $N = pq$. The common interpretation is that the $\Phi$-Hiding Assumption can in practice be viewed as *as hard as factoring* and hence gives a theoretical justification as to why RSA-FDH with a small modulus $N$ is secure in practice.

### 1.2 Full Domain Hash and Coron's Impossibility Result

Recall that FDH signatures on a message $m$ is $\sigma = f^{-1}(H(m))$, where $f$ is the public description of the TDP and $H$ is a hash function modelled as a random oracle. A reduction $\mathcal{R}$ that reduces inverting the TDP to breaking FDH inputs a challenge instance $(f, y = f(x))$ of the TDP and generates a public-key for FDH that is passed to a forger $\mathcal{F}$ attacking FDH signatures. Next, $\mathcal{F}$ makes a number of signature queries (which are answered by $\mathcal{R}$) and finally outputs a forgery. Finally, $\mathcal{R}$ uses the gathered information to invert the TDP, i.e., to compute $x = f^{-1}(y)$. Reduction $\mathcal{R}$ is *tight* if the success probability of $\mathcal{R}$ is roughly the same as the one of $\mathcal{F}$.

Coron's impossibility result shows that any reduction $\mathcal{R}$ from inverting the TDP $f$ to breaking FDH which is tight (i.e., does not lose more than a factor $q_s$) can be turned into an efficient inverting algorithm $\mathcal{I}$ for the TDP $f$ (that works without forger $\mathcal{F}$). In a nutshell, the argument is as follows. Given an instance of the TDP, the inverter $\mathcal{I}$ runs reduction $\mathcal{R}$ providing it with a simulated forger $\mathcal{F}$ by making a number of hash queries and then signature queries to $\mathcal{R}$. Next, $\mathcal{I}$ rewinds reduction $\mathcal{R}$ to an earlier state (after the hash queries) and uses one of the signed messages/signature pairs (say $(m^*, \sigma^*)$) obtained before the rewind as its forgery. To $\mathcal{R}$, this counts as a valid forgery since after the rewind, $\mathcal{I}$ did not make a signing query on $m^*$. The central argument is as follows: consider a real forger that is provided with the view as the simulated forger who outputs a forgery $\sigma'$ on the same message $m^*$. FDH has *unique signatures*[1] and hence we can argue that $\sigma^*$ (provided by $\mathcal{R}$ before the rewind) equals $\sigma'$ (provided by a real forger). Hence $\mathcal{R}$ is convinced it interacts with a real forger and outputs a solution to the TDP instance. Consequently, from $\mathcal{R}$ we were able to construct an algorithm $\mathcal{I}$ that inverts the TDP without using any forger. It is shown by a combinatorial argument that the success probability of $\mathcal{I}$ is non-negative as long as the reduction $\mathcal{R}$ does not loose more than a factor of $q_s$, the number of signature queries.

The Gap in the Proof. During the proof of [12, Th. 5] it is silently assumed that the public-key *pk* generated by reduction $\mathcal{R}$ is a *real public-key*, honestly

[1] A signature scheme has unique signatures if for each message there exists exactly one signature that verifies w.r.t. a given (honestly generated) public-key.

generated by the key-generation algorithm of FDH, i.e., it contains $f$ which described a permutation.[2] However, that does not necessarily hold, the public-key generated by $\mathcal{R}$ could be anything. In fact, it is possible that the public-key generated by the reduction $\mathcal{R}$ is *fake* in the sense that the FDH signatures are no longer unique relative to this fake *pk*. Once signatures are no longer unique (with respect to the fake *pk*), it is possible that a *real forger* outputs a forgery $\sigma'$ on $m^*$ which is different from $\sigma^*$, the one provided by reduction $\mathcal{R}$ before the rewind. In fact, it could be possible that $\sigma^* \neq \sigma'$ is no longer useful for $\mathcal{R}$ in order to solve the RSA instance after the rewind and hence the impossibility result breaks down. In Section 3 we restate (and prove) a corrected version of Coron's impossibility result. Fortunately, it turns out that Coron's argument can be salvaged by requiring the trapdoor permutation in FDH to be certified. Note that in case of a certified trapdoor permutation it is not longer possible for the reduction $\mathcal{R}$ to generate a fake public-key and hence signatures are guaranteed to be unique.

### 1.3 A Tight Security Reduction for FDH Signatures

It is precisely the non-uniqueness of FDH signatures with respect to a fake public-key that will allow us to prove a tight security from the lossiness from the lossiness of the TDP (i.e., the $\Phi$-Hiding Assumption in the case of RSA-FDH). Our proof is surprisingly simple and is sketched as follows. In a first step we substitute the trapdoor permutation in public key with a lossy one. We use the programmability of the random oracle to show that this remains unnoticed by the adversary assuming lossiness of the TDP. Note that once the TDP is lossy, FDH signatures (i.e., $\sigma$ with $f(\sigma) = H(m)$) are not longer unique since, by the definition of lossiness, each $H(m)$ has many pre-images under a lossy $f$. In the second step we show that any successful forger will be able to find a collision in the TDP, i.e., two values $x \neq \hat{x}$ with $f(x) = f(\hat{x})$, which is again hard assuming lossiness. The full proof is given in Section 3.

For the important case of RSA-FDH this gives a tight security reduction from the $\Phi$-Hiding Assumption, in the random oracle model. The $\Phi$-Hiding Assumption is believed to be true for sufficiently small public RSA exponents $e < N^{1/4-\varepsilon}$ [8]. This in particular includes the important low-exponent cases of $e = 3$ and $e = 2^{16} + 1$ since they allow efficient verification of RSA-FDH signatures.[3]

It is interesting to remark, that, at a conceptual level FDH is the first signature scheme with *unique signatures* and a *tight* security reduction (from a non-interactive assumption).[4] Previously, only tight security reductions for *randomized* signatures were known (e.g., [3,17,6,15]).

[2] Such restricted reductions were called *key-preserving reductions* in [24].

[3] We stress that our tight proof technically does not give a counter-example to Coron's impossibility result since our reduction is from the $\Phi$-Hiding Assumption, not the RSA Assumption. However, as corollary the impossibility result would exclude any (even non-tight) equivalence between the $\Phi$-Hiding and the RSA assumption.

[4] Here we do not count tight security proofs from "tautological assumptions" which are essentially assuming that the signature scheme is secure.

### 1.4 Extensions

Our observations can also be applied to the probabilistic signature scheme (PSS) [3] which is contained in IEEE P1363a [19], ISO/IEC 9796-2, and PKCS#1 v2.1 [26]. Coron proved that, if $\log_2(q_s)$ bits of random salt is used in PSS, then there is a tight security reduction from the one-wayness of the TDP [12,13]. Furthermore, Coron also proved that $\log_2(q_s)$ bits of random salt are essentially optimal for a tight security reduction. Our results for PSS are similar to the ones for FDH. We note that Coron's impossibility proof for PSS contains the same gap as the one in FDH, i.e., it is only correct if the underlying trapdoor permutation is certified. However, since PSS (with random salt of arbitrary length) is at least as secure as FDH, we obtain as a corollary from Section 3 a tight security proof from lossiness to the security of PSS, with random salt of arbitrary (possibly zero) length.

### 1.5 Related Work

There is a lot of work on FDH and tightly secure signature schemes, we try to summarize part of it relevant to this work.

Tight security reduction for RSA-FDH from an Interactive Assumption. Kobiltz and Menezes [21, Sec. 3] show a tight reduction from an *interactive assumption* they call the *RSA1 assumption* (which is related to the one-more-RSA assumption RSA-CTI [1]): Given $N$, $e$, and a set of $q_s + q_h$ values $y_i$ chosen uniformly from $\mathbb{Z}_N$, the adversary is permitted adaptively to select up to $q_s$ of those $y_i$ for which he is given solutions $x_i$ to $x_i^e = y_i \bmod N$. The adversary wins if he produces a solution $x_i^e = y_i \bmod N$ for one of the remaining $y_i$. Even though the RSA1 assumption looks plausible, it is an interactive assumption and almost a tautology for expressing that RSA-FDH signatures are secure in the random oracle model. In fact, our tight security proof for RSA-FDH also serves to show a tight reduction from $\Phi$-Hiding to RSA1.

Non-unique Signatures with Tight Reductions. There exists several previous works that build digital signature schemes with a tight security reduction. We stress that all of them have, in contrast to FDH, a randomized signing algorithm, i.e., signatures are not unique. Goh et al. [17] show that adding one single bit of random salt to the hash function of FDH allows to prove a tight security reduction from the RSA assumption. Bernstein [6] shows a tight security reduction for (a certain randomized variant of) Rabin-Williams signature scheme from the factoring assumption. More generally, Gentry et al. [15] introduce the concept of preimage samplable trapdoor functions which are non-injective trapdoor functions with an efficient pre-image sampling algorithm. They further propose a probabilistic variant of FDH and prove it tightly secure. In fact, their proof technique is reminiscent to the second step in our proof of FDH from the lossiness but FDH can not be viewed as an instance of their probabilistic FDH variant.

RSA-OAEP. Recently, [20] used the $\Phi$-Hiding Assumption to show that the RSA function is lossy and used this fact to prove positive instantiability results of RSA-OAEP in the standard model.

### 1.6 Open Problems

On the one hand the *Φ*-Hiding Assumption is believed to be true for public exponents $e \leq N^{1/4-\varepsilon}$ and hence for these values we get a tight security reduction for RSA-FDH. On the other hand, Coron's impossibility results holds for prime $e$ with $e > N$. This leaves the interesting open problem whether for public exponents $N^{1/4} \leq e \leq N$ there exists a tight security reduction for RSA-FDH (under a reasonable assumption).

## 2 Definitions

### 2.1 Notations and Conventions

We denote our security parameter as $k$. For all $n \in \mathbb{N}$, we denote by $1^n$ the $n$-bit string of all ones. For any element $x$ in a set $S$, we use $x \in_R S$ to indicate that we choose $x$ uniformly random in $S$. All algorithms may be randomized. For any algorithm $A$, we define $x \leftarrow_\$ A(a_1, \ldots, a_n)$ as the execution of $A$ with inputs $a_1, \ldots, a_n$ and fresh randomness and then assigning the output to $x$. We denote the set of prime numbers by $\mathbb{P}$ and we denote the subset of $k$-bit primes as $\mathbb{P}_k$. Similarly, we have the integers denoted by $\mathbb{Z}$ and $\mathbb{Z}_k$. We denote by $\mathbb{Z}_N^*$ the multiplicative group modulo $N \in \mathbb{Z}$.

### 2.2 Games

A game (such as in Figure 2) is defined as a collection of procedures, as per the model of [4]. There is an **Initialize** procedure and a **Finalize** procedure, as well a procedure for each separate oracle. Executing a game $\mathsf{G}$ with and adversary $\mathcal{A}$ means running the adversary and using the procedures to answer any oracle queries. The adversary must first make one query to **Initialize**. Then it may query the oracles as many times as allowed by the definition of the game. After this, the adversary must then make 1 query to **Finalize**, which is the final procedure call of the game. The output of **Finalize** is denoted by $\mathsf{G}^{\mathcal{A}}$. Where the Finalize procedure simply returns the output of the adversary, we omit the Finalize procedure. We use a strongly typed pseudo-code with implicit initialization. Which means all variables maintain their type throughout the execution of the games and they are all implicitly declared and initialized. Boolean flags are initialized to false, numerical types are initialized to 0, sets are initialized to $\emptyset$. We use the notation $y \leftarrow_\$ \mathcal{A}(a_1, \ldots, a_n)$ to denote invoking the probabilistic algorithm $\mathcal{A}$ with inputs $a_1, \ldots, a_n$ and fresh randomness and assigning the output to $y$.

### 2.3 Signature Schemes

A digital signature is a message-dependant bit string $\sigma$, which can only be generated by the signer, using a secret signing key *sk* and is transmitted with the message. The signature can then be verified by the receiver using a public verification key *pk*. A digital signature scheme is defined as a triple of probabilistic algorithms $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, which we describe below:

1. $\mathsf{KeyGen}$ takes as an input the unary representation of our security parameter $(1^k)$ and outputs a signing key $sk$ and verification key $pk$.
2. $\mathsf{Sign}$ takes as input a signing key $sk$, message $m$ and outputs a signature $\sigma$.
3. $\mathsf{Verify}$ is a deterministic algorithm, which on input of a public key and a message-signature pair $(m, \sigma)$ outputs 1 (accept) or 0 (reject).

We say that $\mathsf{SIG}$ is correct if for all public key and secret key pairs generated by $\mathsf{KeyGen}$, we have:

$$\Pr[\mathsf{Verify}(pk, m, \mathsf{Sign}(sk, m)) = 1] = 1.$$

We now define $\mathsf{UF\text{-}CMA}$ (unforgeability under chosen message attacks) assuming the signature scheme $\mathsf{SIG}$ contains a hash function $h : \{0,1\}^* \rightarrow \mathsf{Dom}$ which is modeled as a random oracle.

```
procedure Initialize
$(pk, sk) \leftarrow_{\$} \mathsf{KeyGen}(1^k)$
return $pk$

procedure Hash($m$)
if $(m, \cdot) \in \mathcal{H}$ then fetch $(m, y) \in \mathcal{H}$; return $y$
else $y \in_R \mathsf{Dom}$; $\mathcal{H} \leftarrow \mathcal{H} \cup (m, y)$; return $y$

procedure Sign($m$)        Game UF-CMA
$\mathcal{M} \leftarrow \mathcal{M} \cup (m)$
return $\sigma \leftarrow_{\$} \mathsf{Sign}(sk, y)$

procedure Finalize($m^*, \sigma^*$)
if $\mathsf{Verify}(pk, m^*, \sigma^*) = 1 \wedge m^* \notin \mathcal{M}$
then return 1
else return 0
```

**Fig. 1.** Game defining $\mathsf{UF\text{-}CMA}$ security in the random oracle model

We say a signature scheme $\mathsf{SIG}$ is $(t, \varepsilon, q_h, q_s)$-$\mathsf{UF\text{-}CMA}$ secure in the random oracle model, if for all adversaries $\mathcal{A}$ running in time upto $t$, making at most $q_h$ hashing and $q_s$ signing oracle queries, they have an advantage of at most $\epsilon$, where the advantage of $\mathcal{A}$ is defined as:

$$\mathbf{Adv}_{\mathsf{SIG}}^{\mathsf{UF\text{-}CMA}}(\mathcal{A}) = \Pr\left[\mathsf{UF\text{-}CMA}^{\mathcal{A}} \Rightarrow 1\right].$$

### 2.4 Trapdoor Permutations

We recall the definition of trapdoor permutation families.

**Definition 1.** A family of trapdoor permutations $\mathsf{TDP} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ consists of following three polynomial-time algorithms.

1. The probabilistic algorithm $\mathsf{Gen}$, which on input $1^k$ outputs a public description $pub$ (which includes an efficiently sampleable domain $\mathsf{Dom}_{pub}$) and a trapdoor $td$.
2. The deterministic algorithm $\mathsf{Eval}$, which on input $pub$ and $x \in \mathsf{Dom}_{pub}$, outputs $y \in \mathsf{Dom}_{pub}$. We write $f(x) = \mathsf{Eval}(pub, x)$.
3. The deterministic algorithm $\mathsf{Invert}$, which on input $td$ and $y \in \mathsf{Dom}_{pub}$, outputs $x \in \mathsf{Dom}_{pub}$. We write $f^{-1}(y) = \mathsf{Invert}(pub, y)$.

We require that for all $k \in \mathbb{N}$ and all $(pub, td)$ output by $\mathsf{Gen}(1^k)$, $f(\cdot) = \mathsf{Eval}(pub, \cdot)$ defines a permutation over $\mathsf{Dom}_{pub}$ and that for all $x \in \mathsf{Dom}_{pub}$, $\mathsf{Invert}(td, \mathsf{Eval}(pub, x)) = x$.

We want to point out that $f_{pub}(\cdot) = \mathsf{Eval}(pub, \cdot)$ is only required to be a permutation for correctly generated $pub$ but not every bit-string $pub$ necessarily yields a permutation. A family of trapdoor permutations $\mathsf{TDP}$ is said to be *certified* [5] if the fact that it is a permutation can be verified in polynomial time given $pub$.

**Definition 2.** A family of trapdoor permutations $\mathsf{TDP}$ is called certified if there exists a deterministic polynomial-time algorithm $\mathsf{Certify}$ that, on input of $1^k$ and an arbitrary (polynomially bounded) bit-string $pub$ (potentially not generated by $\mathsf{Gen}$), returns 1 iff $f(\cdot) = \mathsf{Eval}(pub, \cdot)$ defines a permutation over $\mathsf{Dom}_{pub}$.

We now recall security notion for trapdoor permutations. A trapdoor permutation $\mathsf{TDP}$ is hard to invert (one-way) if given $pub$ and $f_{pub}(x)$ for uniform $x \in \mathsf{Dom}_{pub}$, it is hard to compute $x$. More formally, it is $(t, \varepsilon)$-hard to invert if for all adversaries running in time $t$, $\Pr[\mathcal{A}(pub, \mathsf{Eval}(pub, x)) = x] \leq \varepsilon$, where the probability is taken over $(pub, td) \leftarrow \mathsf{Gen}(1^k)$, $x \in_R \mathsf{Dom}_{pub}$ and the random coin tosses of $\mathcal{A}$. The following security notion, lossiness [25], is a stronger requirement than one-wayness.

**Definition 3.** Let $l \geq 2$. A trapdoor permutation $\mathsf{TDP}$ is a $(l, t, \varepsilon)$ lossy trapdoor permutation if the following two conditions hold.[5]

1. There exists a probabilistic polynomial-time algorithm $\mathsf{LossyGen}$, which on input $1^k$ outputs $pub'$ such that the range of $f_{pub'}(\cdot) := \mathsf{Eval}(pub', \cdot)$ under $\mathsf{Dom}_{pub'}$ is at least a factor of $l$ smaller than the domain $\mathsf{Dom}_{pub'}$: $|\mathsf{Dom}_{pub'}|/|f_{pub'}(\mathsf{Dom}_{pub'})| \geq l$. (Note that we measure the lossiness in its absolute value $l$, i.e., the function has $\lceil \log_2 l \rceil$ *bits* of lossiness.)
2. All distinguishers $\mathcal{D}$ running in time at most $t$ have an advantage $\mathbf{Adv}^{\mathsf{L}}_{\mathsf{TDP}}(\mathcal{D})$ of at most $\varepsilon$, where

$$\mathbf{Adv}^{\mathsf{L}}_{\mathsf{TDP}}(\mathcal{D}) = \Pr[\mathsf{L}_1^{\mathcal{D}} \Rightarrow 1] - \Pr[\mathsf{L}_0^{\mathcal{D}} \Rightarrow 1].$$

| **procedure Initialize** Game $\mathrm{L}_0$ | **procedure Initialize** Game $\mathrm{L}_1$ |
|---|---|
| $(pub, td) \leftarrow_{\$} \mathsf{Gen}(1^k)$ | $(pub', \bot) \leftarrow_{\$} \mathsf{LossyGen}(1^k)$ |
| return $pub$ | return $pub'$ |

**Fig. 2.** The Lossy Trapdoor Permutation Games

We say $\mathsf{TDP}$ is *regular* $(l, t, \varepsilon)$ *lossy* if $\mathsf{TDP}$ is $(l, t, \varepsilon)$ lossy and all functions $f_{pub'}(\cdot) = \mathsf{Eval}(pub', \cdot)$ generated by $\mathsf{LossyGen}$ are $l$-to-1 on $\mathsf{Dom}_{pub'}$.

[5] We deviate in two ways from the original definition of lossy trapdoor functions Peikert and Waters [25]. First, we define the permutation over arbitrary domains $\mathsf{Dom}$, rather than $\{0,1\}^k$; second, we measure the absolute lossiness $l$, rather than the bits of lossiness $\ell = \log_2(l)$.

### 2.5 The RSA Trapdoor Permutation

We define the RSA trapdoor permutation $\mathsf{RSA} = (\mathsf{RSAGen}, \mathsf{RSAEval}, \mathsf{RSAInv})$ as follows. The RSA instance generator $\mathsf{RSAGen}(1^k)$ outputs $pub = (N, e)$ and $td = d$, where $N = pq$ is the product of two $k/2$-bit primes, $\gcd(e, \varphi(N)) = 1$, and $d = e^{-1} \bmod \varphi(N)$. The domain is $\mathsf{Dom}_{pub} = \mathbb{Z}_N^*$. The evaluation algorithm $\mathsf{RSAEval}(pub, x)$ returns $f_{pub}(x) = x^e \bmod N$, the inversion algorithm $\mathsf{RSAInv}(td, y)$ returns $f_{pub}^{-1}(y) = y^d \bmod N$. The standard assumption is that $\mathsf{RSA}$ is hard to invert. We will review the (regular) lossiness of $\mathsf{RSA}$ in Section 4.

## 3 Full Domain Hash Signatures

### 3.1 The Scheme

For a familiy of trapdoor permutations $\mathsf{TDP} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ we define the Full Domain Hash ($\mathsf{TDP\text{-}FDH}$) signature scheme [3] in Figure 3.

```
procedure KeyGen                                                  TDP-FDH
$(pub, td) \leftarrow_{\$} \mathsf{Gen}(1^k)$
Pick a hash function $h : \{0,1\}^* \to \mathsf{Dom}_{pub}$
return $(pk = (h, pub), sk = td)$

procedure Sign$(sk, m)$
return $\sigma = \mathsf{Invert}(td, h(m))$                // $\sigma = f_{pub}^{-1}(h(m))$

procedure Verify$(pk, m, \sigma)$
if $\mathsf{Eval}(pub, \sigma) = h(m)$ then return 1           // $f_{pub}(\sigma) \stackrel{?}{=} h(m)$
else return 0
```

**Fig. 3.** The Full Domain Hash Signature Scheme $\mathsf{TDP\text{-}FDH}$

### 3.2 Classical Security Results of TDP-FDH

The original reduction by Bellare and Rogaway from one-wayness of $\mathsf{TDP}$ loses a factor of $(q_h + q_s)$ [3], which was later improved by Coron to a factor of $q_s$ [11] for the case of the $\mathsf{RSA}$ trapdoor permutation.

**Theorem 4.** *Assume the trapdoor permutation* $\mathsf{RSA}$ *is* $(t', \varepsilon')$*-hard to invert. Then for any* $(q_h, q_s)$, $\mathsf{RSA\text{-}FDH}$ *is* $(t, \varepsilon, q_h, q_s)$-$\mathsf{UF\text{-}CMA}$ *secure in the Random Oracle Model, where*

$$\varepsilon' = \frac{\varepsilon}{q_s} \cdot \left(1 - \frac{1}{q_s + 1}\right)^{q_s + 1} \approx \frac{\varepsilon}{q_s} \cdot \exp(-1)$$

$$t' = t + (q_h + q_s + 1) \cdot \mathcal{O}(k^3).$$

### 3.3 A Corrected Version of Coron's Optimality Result

Coron showed that a security loss of a factor $q_s$ (times some constant) is essentially optimal for $\mathsf{TDP\text{-}FDH}$ [12,13]. To state a corrected version of Coron's impossibility result, we first recall the following definitions [12].

**Definition 5.** We say a reduction $\mathcal{R}$ $(t_{\mathcal{F}}, t_{\mathcal{R}}, q_h, q_s, \varepsilon_{\mathcal{F}}, \varepsilon_{\mathcal{R}})$-reduces solving a hard problem to breaking $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ if after running a forger $\mathcal{F}$ that $(t_{\mathcal{F}}, q_h, q_s, \varepsilon_{\mathcal{F}})$-breaks $\mathsf{SIG}$, the reduction outputs a solution of the problem with probability at least $\varepsilon_{\mathcal{R}}$, with running time at most $t_{\mathcal{R}}$.

**Definition 6.** A signature scheme $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is said to be a unique signature scheme if for every public key $pk$ output by $\mathsf{KeyGen}$, for every message $m$ there exists exactly one bit-string $\sigma \in \{0,1\}^*$ such that $\mathsf{Verify}(pk, m, \sigma) = 1$.

We now state the corrected version of Coron's impossibility result which we prove in the full version of this paper.

**Theorem 7.** *Suppose* $\mathsf{TDP}$ *is a certified trapdoor permutation. Let* $\mathcal{R}$ *be a reduction that* $(t_{\mathcal{F}}, t_{\mathcal{R}}, q_h, q_s, \varepsilon_{\mathcal{F}}, \varepsilon_{\mathcal{R}})$*-reduces breaking one-wayness of* $\mathsf{TDP}$ *to breaking* $\mathsf{UF\text{-}CMA}$ *security of* $\mathsf{TDP\text{-}FDH}$*. If* $\mathcal{R}$ *runs the forger only once, then we can build an inverter* $\mathcal{I}$ *which* $(t_{\mathcal{I}}, \varepsilon_{\mathcal{I}})$*-breaks one-wayness of* $\mathsf{TDP}$ *with:*

$$t_{\mathcal{I}} \leq 2 \cdot t_{\mathcal{R}}$$

$$\varepsilon_{\mathcal{I}} \geq \varepsilon_{\mathcal{R}} - \varepsilon_{\mathcal{F}} \cdot \frac{exp(-1)}{q_s} \cdot \left(1 - \frac{q_s}{q_h}\right)^{-1} .$$

Hence, from a security reduction from one-wayness to the security of $\mathsf{TDP\text{-}FDH}$ which loses less than a factor of $q_s$, one obtains an efficient inverter $\mathcal{I}$ for $\mathsf{TDP}$.

### 3.4 A Tight Security Proof for TDP-FDH

The impossibility result of Theorem 7 only holds for $\mathsf{TDP\text{-}FDH}$ if $\mathsf{TDP}$ is certified trapdoor permutation. However if $\mathsf{TDP}$ is not certified, this leaves room for a tight proof for $\mathsf{TDP\text{-}FDH}$. We now state our main result, namely that $\mathsf{TDP\text{-}FDH}$ is tightly secure assuming $\mathsf{TDP}$ is regular lossy.

**Theorem 8.** *Assume* $\mathsf{TDP} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ *is a regular* $(l, t', \varepsilon')$*-lossy trapdoor permutation for* $l \geq 2$*. Then, for any* $(q_h, q_s)$*,* $\mathsf{TDP\text{-}FDH}$ *is* $(t, \varepsilon, q_h, q_s)$*-*$\mathsf{UF\text{-}CMA}$ *secure in the Random Oracle Model, where*

$$\varepsilon = \left(\frac{2l-1}{l-1}\right) \cdot \varepsilon'$$

$$t = t' - q_h \cdot T_{\mathsf{TDP}}$$

*and* $T_{\mathsf{TDP}}$ *is the time to evaluate* $\mathsf{TDP}$.

**Table 1.** Games for the proof of Theorem 8

| | |
|---|---|
| **procedure Initialize** Game $\mathsf{G}_0$ = $(\mathsf{UF\text{-}CMA})$<br>$(pub, td) \leftarrow_\$ \mathsf{Gen}(1^k)$<br>Return $pk = pub$ | **procedure Initialize** Games $\mathsf{G}_1$-$\mathsf{G}_4$<br>$(pub, td) \leftarrow_\$ \mathsf{Gen}(1^k)$ $//\mathsf{G}_1, \mathsf{G}_4$<br>$(pub, \perp) \leftarrow_\$ \mathsf{LossyGen}(1^k)$ $//\mathsf{G}_2, \mathsf{G}_3$<br>Return $pk = pub$ |
| **procedure Hash**$(m)$<br>if $(m, \cdot) \in \mathcal{H}$ then fetch $(m, y_m)$; return $y_m$<br>else<br>$y_m \in_R \mathsf{Dom}_{pub}$<br>$\mathcal{H} \leftarrow \mathcal{H} \cup (m, y_m)$; return $y_m$ | **procedure Hash**$(m)$<br>if $m \in \mathcal{H}$ lookup $(m, y_m, \sigma_m) \in \mathcal{H}$<br>return $y_m$<br>else<br>$\sigma_m \in_R \mathsf{Dom}_{pub}$<br>$y_m = \mathsf{Eval}(pub, \sigma_m)$<br>$\mathcal{H} \leftarrow \mathcal{H} \cup (m, y_m, \sigma_m)$; return $y_m$ |
| **procedure Sign**$(m)$<br>$\mathcal{M} \leftarrow \mathcal{M} \cup (m)$<br>return $\sigma_m = \mathsf{Invert}(td, h(m))$ | **procedure Sign**$(m)$<br>$\mathcal{M} \leftarrow \mathcal{M} \cup (m)$<br>lookup $(m, y_m, \sigma_m) \in \mathcal{H}$, return $\sigma_m$ |
| **Procedure Finalize**$(m^*, \sigma^*)$<br>if $(\mathsf{Verify}(pub, m^*, \sigma^*) = 1) \wedge (m^* \notin \mathcal{M})$<br>return 1<br>else return 0 | **Procedure Finalize**$(m^*, \sigma^*)$<br>lookup $(m^*, y_{m^*}, \sigma_{m^*}) \in \mathcal{H}$ $//\mathsf{G}_3, \mathsf{G}_4$<br>if $\sigma_{m^*} = \sigma^*$ then BAD = true<br>return 0 $//\mathsf{G}_3, \mathsf{G}_4$<br>if $\mathsf{Verify}(pub, m^*, \sigma^*) = 1 \wedge (m^* \notin \mathcal{M})$<br>return 1<br>else return 0 |

*Proof.* Let $\mathcal{A}$ be an adversary that runs in time $t$ against $\mathsf{TDP\text{-}FDH}$ executed in the UF-CMA experiment described in $\mathsf{G}_0$ in Figure 1 with $\varepsilon = \Pr[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1]$. Here we assume wlog that $\mathcal{A}$ always makes a query to $\mathsf{Hash}(m)$ before calling $\mathsf{Sign}(m)$ or $\mathsf{Finalize}(m, \cdot)$.

**Lemma 9.** $\Pr[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1]$.

*Proof.* In $\mathsf{G}_0$, we modelled the hash function as a random oracle. In $\mathsf{G}_1$ we modify the random oracle and the signing queries. On any $m$ the random oracle now works by evaluating the permutation on a random element $\sigma_m \in \mathsf{Dom}_{pub}$. We then modify the signing oracle to return this element $\sigma_m$. Note that signing no longer requires the trapdoor $td$. It can be seen that all our signatures will verify due to the fact that $\mathsf{Eval}(pub, \sigma_m) = y_m$ for all $m$. Thus our simulation of the signatures is correct. Since $\mathsf{TDP}$ is a permutation, the distribution of our hash queries in $\mathsf{G}_1$ is identical to the distribution in $\mathsf{G}_0$. Thus we have $\Pr[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1]$.

**Lemma 10.** *There exists a distinguisher $\mathcal{D}_1$ against the lossines of* $\mathsf{TDP}$*, which runs in time $t = t_{\mathcal{A}} + q_h \cdot T_{\mathsf{TDP}}$ and that* $\Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1] = \boldsymbol{Adv}_{\mathsf{TDP}}^{\mathsf{L}}(\mathcal{D}_1)$.

*Proof.* From $\mathsf{G}_1$ to $\mathsf{G}_2$, we change the key generation from a normal permutation to a lossy permutation, however the oracles are identical in both games. We now build a distinguisher $\mathcal{D}_1$ against the lossiness of $\mathsf{TDP}$, using these games. The distinguisher will run $\mathcal{A}$ and simulates the oracles $\mathsf{Sign}(\cdot), \mathsf{Hash}(\cdot)$ as described in games $\mathsf{G}_1$&$\mathsf{G}_2$, for which it requires time $q_h \cdot T_{\mathsf{TDP}}$. Note that $\mathcal{D}_1$ does not require the trapdoor $td$ to simulate the oracles. After $\mathcal{A}$ calls $\mathsf{Finalize}$, $\mathcal{D}_1$ returns the inverse of $\mathsf{Finalize}$. Thus we can see that $\Pr[\mathrm{L}_0^{\mathcal{D}_1} \Rightarrow 1] = 1 - \Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1]$. Similarly, we have $\Pr[\mathrm{L}_1^{\mathcal{D}_1} \Rightarrow 1] = 1 - \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1]$. Hence we have $\Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1] = (1 - \Pr[\mathrm{L}_0^{\mathcal{D}_1} \Rightarrow 1]) - (1 - \Pr[\mathrm{L}_1^{\mathcal{D}_1} \Rightarrow 1]) = \Pr[\mathrm{L}_1^{\mathcal{D}_1} \Rightarrow 1] - \Pr[\mathrm{L}_0^{\mathcal{D}_1} \Rightarrow 1] = \mathbf{Adv}_{\mathsf{TDP}}^{\mathsf{L}}(\mathcal{D}_1)$.

**Lemma 11.** $\Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1] = \left(\frac{l-1}{l}\right) \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1]$.

*Proof.* In $\mathsf{G}_3$, we introduce a new rule, which sets BAD to true if the forgery $\sigma^*$ provided by $\mathcal{A}$ is the same as the simulated signature $\sigma_{m^*}$ for the target message $m^*$. If this is the case, the adversary loses the game, i.e., $\mathsf{G}_3$ outputs 0. $\sigma_{m^*}$ is independent of $\mathcal{A}$'s view and is uniformly distributed in set of pre-images of $y_{m^*}$. Due to the $l$ regular lossiness of $\mathsf{TDP}$, the probability of a collision is equal to exactly $1/l$. Thus we see that the BAD rule reduces the probability of the adversary winning the game by $1/l$, hence $\Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1] = (1 - \frac{1}{l}) \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1] = \left(\frac{l-1}{l}\right) \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1]$.

**Lemma 12.** *There exists a distinguisher* $\mathcal{D}_2$ *against the lossiness of* $\mathsf{TDP}$, *which runs in time* $t = t_{\mathcal{A}} + q_h \cdot T_{\mathsf{TDP}}$ *and that* $\Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1] = \boldsymbol{Adv}_{\mathsf{TDP}}^{\mathsf{L}}(\mathcal{D}_2)$.

*Proof.* From $\mathsf{G}_3$ to $\mathsf{G}_4$, we change the key generation from a lossy permutation to a normal permutation, however the oracles are identical in both games. We now build a distinguisher $\mathcal{D}_2$ against the lossiness of $\mathsf{TDP}$, using these games. The distinguisher will act as the challenger to $\mathcal{A}$. It will simulate the oracles as described in games $\mathsf{G}_3$&$\mathsf{G}_4$, for which it requires time $q_h \cdot T_{\mathsf{TDP}}$. After $\mathcal{A}$ calls $\mathsf{Finalize}$, $\mathcal{D}_2$ returns the output of $\mathsf{Finalize}$. We can see that $\Pr[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathrm{L}_0^{\mathcal{D}_2} \Rightarrow 1]$. Similarly, we have $\Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathrm{L}_1^{\mathcal{D}_2} \Rightarrow 1]$. Hence we have $\Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathrm{L}_1^{\mathcal{D}_2} \Rightarrow 1] - \Pr[\mathrm{L}_0^{\mathcal{D}_2} \Rightarrow 1] = \mathbf{Adv}_{\mathsf{TDP}}^{\mathsf{L}}(\mathcal{D}_2)$.

**Lemma 13.** $\Pr[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1] = 0$.

*Proof.* In $\mathsf{G}_4$ we again use the original $\mathsf{KeyGen}$ such that $\mathsf{Eval}(pub, \cdot)$ defines a permutation. This means that our signing function is now a permutation, thus any forgery implies a collision. Therefore whenever the adversary is able to make a forgery, the game outputs 0 due to the BAD rule. Whenever they are unable to make a forgery, the game outputs 0. Thus we can see that in all cases, the game will output 0, hence $\Pr[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1] = 0$.

We combine Lemmas 9 to 13 to get:

$$\Pr[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1] = \mathbf{Adv}_{\mathsf{TDP}}^{\mathsf{L}}(\mathcal{D}_1) + \left(\tfrac{l}{l-1}\right)\mathbf{Adv}_{\mathsf{TDP}}^{\mathsf{L}}(\mathcal{D}_2).$$

where $l$ is the lossiness of $\mathsf{TDP}$. Because the distinguishers run in the same time, we know that both distinguishers can have at most an advantage of $\varepsilon'$, giving us:

$$\varepsilon \leq \frac{2l-1}{l-1} \cdot \varepsilon'.$$

This completes the proof.

# 4 Lossiness of RSA from the *Φ*-Hiding Assumption

## 4.1 Lossiness of RSA

The lossiness of RSA for a number of specific instance generators $\mathsf{RSAGen}$ was first considered in [20]. We now recall (and extend) some of the results from [20].

First, we recall some definitions from [20]. We denote by $\mathcal{RSA}_k := \{(N,p,q) \mid N = pq, p, q \in \mathbb{P}_{k/2}\}$ the set of all the tuples $(N,p,q)$ such that $N = pq$ is the product of two distinct $k/2$-bit primes. Such an $N$ is called an RSA modulus. By $(N,p,q) \in_{_R} \mathcal{RSA}_k$ we mean the $(N,p,q)$ is sampled according to the uniform distribution on $\mathcal{RSA}_k$. Let $R$ be some relation on $p$ and $q$. By $\mathcal{RSA}_k[R]$, we denote the subset of $\mathcal{RSA}_k$ such that the relation $R$ holds on $p$ and $q$. For example, let $e$ be a prime. Then $\mathcal{RSA}_k[p = 1 \bmod e]$ is the set of all $(N,p,q)$, where where $N = pq$ is the product of two distinct $k/2$-bit primes $p$, $q$ and $p = 1 \bmod e$. That is, the relation $R(p,q)$ is true if $p = 1 \bmod e$ and $q$ is arbitrary. By $(N,p,q) \in_{_R} \mathcal{RSA}_k[R]$ we mean that $(N,p,q)$ is sampled according to the uniform distribution on $\mathcal{RSA}_k[R]$.

$\alpha$-*Φ*-Hiding Assumption. We recall a variant of the *Φ*-Hiding Assumption introduced by Cachin, Micali and Stadler [8], where we build on a formalization by Kiltz, O'Neil and Smith [20]. The main statement of the assumption is that given an $k$-bit RSA modulus $N = pq$ and a random $\alpha \cdot k$-bit prime $e$ (where $0 < \alpha < \frac{1}{4}$ is a public constant), it is difficult to decide if $e \mid \varphi(N)$ or if $gcd(e, \varphi(N)) = 1$. We note that if $e \mid \varphi(N)$ with $e \geq N^{1/4}$, then $N$ can be factored using Coppersmith's attacks [10], see [8] for details. Hence for the *Φ*-Hiding Assumption to hold, the bit-length of $e$ must not exceed one-fourth of the bit length of $N$.

Consider a distinguisher $\mathcal{D}$ which plays one of the games $\mathsf{P}_0$ or $\mathsf{P}_1$ defined in Table 2, The advantage of $\mathcal{D}$ is defined as:

$$\mathbf{Adv}^{\Phi\mathrm{H}}(\mathcal{D}) = \Pr[\mathsf{P}_1^{\mathcal{D}} \Rightarrow 1] - \Pr[\mathsf{P}_0^{\mathcal{D}} \Rightarrow 1].$$

We say that the $\alpha$-*Φ*-Hiding Problem is $(t, \epsilon)$-hard if for all distinguishers $\mathcal{D}$ running in time at most $t$ have and advantage of at most $\epsilon$.

Define an RSA instance generator $\mathsf{RSAGen}$ as an algorithm that returns $(N, e, p, q)$ sampled as $e \in_{_R} \mathbb{P}_{\alpha k}$ and $(N,p,q) \in_{_R} \mathcal{RSA}_k[gcd(e, \varphi(N) = 1]$. (See [20] for details on the sampling algorithm.)

**Lemma 14.** *If the $\alpha$-Φ-Hiding Problem is $(t, \epsilon)$-hard, then the* $\mathsf{RSA} = (\mathsf{RSAGen}, \mathsf{RSAEval}, \mathsf{RSAInv})$ *defines a regular $(2^\alpha, t, \epsilon)$-lossy trapdoor permutation.*

**Table 2.** The $\alpha$-$\Phi$-Hiding Assumption Games

| **procedure Initialize** Game $\mathsf{P}_0$ | **procedure Initialize** Game $\mathsf{P}_1$ |
|---|---|
| $e \in_R \mathbb{P}_{\alpha k}$ | $e \in_R \mathbb{P}_{\alpha k}$ |
| $(N, p, q) \in_R \mathcal{RSA}_k[gcd(e, \varphi(N) = 1]$ | $(N, p, q) \in_R \mathcal{RSA}_k[p = 1 \mod e]$ |
| return $(N, e)$ | return $(N, e)$ |

*Proof.* If $(N, e)$ is sampled using $\mathsf{RSAGen}$, then $\gcd(e, \varphi(N) = 1)$ and $(N, e)$ defines a permutation $\mathsf{RSA}(x) = x^e \mod N$ over $\mathbb{Z}_N^*$. We define $\mathsf{LossyGen}$ to be an algorithm that returns $(N, e)$ sampled as $e \in_R \mathbb{P}_{\alpha k}$ and $(N, p, q) \in_R \mathcal{RSA}_k[p = 1 \mod e]$. If $(N, e)$ is sampled using $\mathsf{LossyGen}$ then $e \mid \varphi(N)$ and hence the RSA function is $e$-to-1 on the domain $\mathsf{Dom}_{pub} = \mathbb{Z}_N^*$. By definition, the outputs of $\mathsf{RSAGen}$ and $\mathsf{LossyGen}$ are indistinguishable if the $\alpha$-$\Phi$-Hiding Problem is hard.

Fixed-Prime $\Phi$-Hiding Assumption. In practice, $e$ is chosen to be small and is generally fixed to some specific numbers, such as $e = 3$ or $e = 2^{16} + 1$, which allows for fast exponentiation. We now show a minor variant of the $\alpha$-$\Phi$-Hiding Assumption for *fixed primes* $e$, where our formalization relies on discussions from [8] and [20, Footnote 9].

First, we discuss the special case of $e = 3$. We define our RSA instance $\mathsf{RSAGen}_3$ generator as an algorithm that samples $(N, p, q)$ uniformly from $\mathcal{RSA}_k$ $[p = 2 \mod 3, q = 2 \mod 3]$, which is equivalent to $\mathcal{RSA}_k[gcd(3, \varphi(N)) = 1]$. We note that $N \mod 3$ is always 1. This means that for the lossy case, we must also ensure the $N \mod 3 = 1$, otherwise there would be a simple distinguisher. To ensure this is to have 3 divide both $p - 1$ and $q - 1$. Thus, our lossy keys are sampled from the $\mathcal{RSA}_k[p = 1 \mod 3, q = 1 \mod 3]$.

**Table 3.** The Fixed-Prime $\Phi$-Hiding Assumption Games

| **procedure Initialize** Game $\mathsf{3F}_0$ | **procedure Initialize** Game $\mathsf{3F}_1$ |
|---|---|
| $(N, p, q) \in_R \mathcal{RSA}_k[gcd(3, \varphi(N)) = 1]$ | $(N, p, q) \in_R \mathcal{RSA}_k[p = 1 \mod 3, q = 1 \mod 3]$ |
| return $(N, e = 3)$ | return $(N, e = 3)$ |

Consider a distinguisher $\mathcal{D}$ which plays one of the games in Table 3. The advantage of $\mathcal{D}$ is defined as

$$\mathbf{Adv}^{\mathrm{F}\Phi\mathrm{H}}(\mathcal{D}) = \Pr[\mathsf{3F}_1^{\mathcal{D}} \Rightarrow 1] - \Pr[\mathsf{3F}_0^{\mathcal{D}} \Rightarrow 1].$$

We say that the Fixed-Prime $\Phi$-Hiding Problem, with $e = 3$, is $(t, \epsilon)$-hard if all distinguishers running in time at most $t$ have an advantage of at most $\epsilon$.

**Lemma 15.** *If the Fixed-Prime $\Phi$-Hiding Problem, with $e = 3$, is $(t, \epsilon)$-hard, then the* $\mathsf{RSA}_3 = (\mathsf{RSAGen}_3, \mathsf{RSAEval}, \mathsf{RSAInv})$ *defines a regular $(9, t, \epsilon)$-lossy trapdoor permutation.*

*Proof.* If $(N, p, q) \in \mathcal{RSA}_k[gcd(3, \varphi(N)) = 1]$ then $(N, 3)$ clearly makes the RSA function a permutation. If $(N, p, q) \in \mathcal{RSA}_k[p = 1 \mod 3, q = 1 \mod 3]$ then $3 \mid \varphi(N)$ and hence the RSA function is 9-to-1 on the domain $\mathsf{Dom}_{pub} = \mathbb{Z}_N^*$.

We now consider the general case of fixed $e > 3$. For this case, we define our RSA instance generator $\mathsf{RSAGen}_e$ as an algorithm that samples $(N, p, q)$ from $\mathcal{RSA}_k[gcd(e, \varphi(N)) = 1]$. We note that $N \mod e$ will be some value between 1 and $e - 1$. This means that for the lossy case, we require $e$ to divide $p - 1$ and not $q - 1$, otherwise we would have a simple distinguisher. Our lossy keys are sampled from $\mathcal{RSA}_k[p = 1 \mod e, q \neq 1 \mod e]$. Consider a distinguisher $\mathcal{D}$ which plays one of the games in Table 4. The advantage of $\mathcal{D}$ is defined as

**Table 4.** The Fixed-Prime $\Phi$-Hiding Assumption Games

| **procedure Initialize** Game $\mathsf{F}_0$ | **procedure Initialize** Game $\mathsf{F}_1$ |
|---|---|
| $(N, p, q) \in_R \mathcal{RSA}_k[gcd(e, \varphi(N)) = 1]$ | $(N, p, q) \in_R \mathcal{RSA}_k[p = 1 \mod e, q \neq 1 \mod e]$ |
| return $(N, e)$ | return $(N, e)$ |

$$\mathbf{Adv}^{\mathrm{F\Phi H}}(\mathcal{D}) = \Pr[\mathsf{F}_1^{\mathcal{D}} \Rightarrow 1] - \Pr[\mathsf{F}_0^{\mathcal{D}} \Rightarrow 1].$$

We say that the Fixed-Prime $\Phi$-Hiding Problem, with $e > 3$, is $(t, \epsilon)$-hard if for all distinguishers running in time at most $t$ have an advantage of at most $\epsilon$.

**Lemma 16.** *If the Fixed-Prime $\Phi$-Hiding Problem, with $e > 3$, is $(t, \epsilon)$-hard, then* $\mathsf{RSA}_e = (\mathsf{RSAGen}_e, \mathsf{RSAEval}, \mathsf{RSAInv})$ *defines a regular $(e, t, \epsilon)$-lossy trapdoor permutation.*

*Proof.* If $(N, p, q) \in \mathcal{RSA}_k[gcd(e, \varphi(N)) = 1]$ then $(N, e)$ clearly defines a permutation. If $(N, p, q) \in \mathcal{RSA}_k[p = 1 \mod e, q \neq 1 \mod e]$ then $e \mid \varphi(N)$ and hence the RSA function is $e$-to-1 on the domain $\mathsf{Dom}_{pub} = \mathbb{Z}_N^*$.

**Acknowledgements.** We thank Mihir Bellare and Dennis Hofheinz for valuable comments on an earlier draft.

## References

1. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology 16(3), 185–215 (2003)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993: 1st Conference on Computer and Communications Security, pp. 62–73. ACM Press (November 1993)
3. Bellare, M., Rogaway, P.: The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996)

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. Bellare, M., Yung, M.: Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. Journal of Cryptology 9(3), 149–166 (1996)
6. Bernstein, D.J.: Proving Tight Security for Rabin-Williams Signatures. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 70–87. Springer, Heidelberg (2008)
7. Cachin, C.: Efficient private bidding and auctions with an oblivious third party. In: ACM CCS 1999: 6th Conference on Computer and Communications Security, pp. 120–127. ACM Press (November 1999)
8. Cachin, C., Micali, S., Stadler, M.A.: Computationally Private Information Retrieval with Polylogarithmic Communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
9. Chatterjee, S., Menezes, A., Sarkar, P.: Another look at tightness. Cryptology ePrint Archive, Report 2011/442 (2011), http://eprint.iacr.org/
10. Coppersmith, D.: Finding a Small Root of a Univariate Modular Equation. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (1996)
11. Coron, J.-S.: On the Exact Security of Full Domain Hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000)
12. Coron, J.-S.: Optimal security proofs for pss and other signature schemes. Cryptology ePrint Archive, Report 2001/062 (2001), http://eprint.iacr.org/
13. Coron, J.-S.: Optimal Security Proofs for PSS and Other Signature Schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002)
14. Gentry, C., Mackenzie, P.D., Ramzan, Z.: Password authenticated key exchange using hidden smooth subgroups. In: Atluri, V., Meadows, C., Juels, A. (eds.) ACM CCS 2005: 12th Conference on Computer and Communications Security, pp. 299–309. ACM Press (November 2005)
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th Annual ACM Symposium on Theory of Computing, pp. 197–206. ACM Press (May 2008)
16. Gentry, C., Ramzan, Z.: Single-Database Private Information Retrieval with Constant Communication Rate. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 803–815. Springer, Heidelberg (2005)
17. Goh, E.-J., Jarecki, S., Katz, J., Wang, N.: Efficient signature schemes with tight reductions to the Diffie-Hellman problems. Journal of Cryptology 20(4), 493–514 (2007)
18. Hemenway, B., Ostrovsky, R.: Public-Key Locally-Decodable Codes. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 126–143. Springer, Heidelberg (2008)
19. IEEE P1363a Committee. IEEE P1363a / D9 — standard specifications for public key cryptography: Additional techniques, Draft Version 9 (June 2001), http://grouper.ieee.org/groups/1363/index.html/
20. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under Chosen-Plaintext Attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (2010)
21. Koblitz, N., Menezes, A.J.: Another look at "provable security". Journal of Cryptology 20(1), 3–37 (2007)

22. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential Aggregate Signatures from Trapdoor Permutations. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 74–90. Springer, Heidelberg (2004)
23. Micali, S.: Computationally sound proofs. SIAM J. Comput. 30, 1253–1298 (2000)
24. Paillier, P., Villar, J.L.: Trading One-Wayness Against Chosen-Ciphertext Security in Factoring-Based Encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer, Heidelberg (2006)
25. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th Annual ACM Symposium on Theory of Computing, pp. 187–196. ACM Press (May 2008)
26. PKCS #1: RSA cryptography standard. RSA Data Security, Inc., Version 2.0 (September 1998)
27. Schridde, C., Freisleben, B.: On the Validity of the $\Phi$-Hiding Assumption in Cryptographic Protocols. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 344–354. Springer, Heidelberg (2008)
28. Smart, N.: Ecrypt ii yearly report on algorithms and keysizes (2009-2010). Framework, p. 116 (March 2010)