On the Generic Insecurity of the Full Domain Hash

Yevgeniy Dodis*[1], Roberto Oliveira[2], and Krzysztof Pietrzak**[3]

[1] New York University, dodis@cs.nyu.edu
[2] IBM T.J. Watson Research Center, riolivei@us.ibm.com
[3] ETH Zürich, pietrzak@inf.ethz.ch

Abstract. The *Full-Domain Hash* (FDH) signature scheme [3] forms one the most basic usages of random oracles. It works with a family $\mathcal{F}$ of trapdoor permutations (TDP), where the signature of m is computed as $f^{-1}(h(m))$ (here $f \in_{\mathcal{R}} \mathcal{F}$ and h is modelled as a random oracle). It is known to be existentially unforgeable for any TDP family $\mathcal{F}$ [3], although a much tighter security reduction is known for a restrictive class of TDP's [10, 14] — namely, those induced by a family of claw-free permutations (CFP) pairs. The latter result was shown [11] to match the best *possible* "black-box" security reduction in the random oracle model, irrespective of the TDP family $\mathcal{F}$ (e.g., RSA) one might use.
In this work we investigate the question if it is possible to instantiate the random oracle h with a "real" family of hash functions $\mathcal{H}$ such that the corresponding schemes can be proven secure *in the standard model*, under some natural assumption on the family $\mathcal{F}$. Our main result *rules out* the existence of such instantiations for *any* assumption on $\mathcal{F}$ which (1) is satisfied by a family of random permutations; and (2) does not allow the attacker to invert $f \in_{\mathcal{R}} \mathcal{F}$ on an a-priori unbounded number of points. Moreover, this holds even if the choice of $\mathcal{H}$ can arbitrarily depend on f. As an immediate corollary, we rule out instantiating FDH based on general claw-free permutations, which shows that in order to prove the security of FDH in the standard model one must utilize significantly more structure on $\mathcal{F}$ than what is sufficient for the best proof of security in the random oracle model.

1 Introduction

FULL DOMAIN HASH. Dating back to Diffie-Hellman [13], the simplest classical suggestion for the design of digital signature schemes is to set the signature of the message m to be $\sigma = f^{-1}(m)$, where f comes from a family of trapdoor permutations (TDP) $\mathcal{F}$ such RSA. Unfortunately, this simple scheme is existentially forgeable (even under no message attack), since any σ happens to be the signature of $m = f(\sigma)$. A folklore suggestion to fix this problem, which is the basis of several existing standards such as PKCS #1 [1], is to hash the message before

* Supported by NSF CAREER Award CCR-0133806 and TC Grant No. CCR-0311095.
** Supported by the Swiss National Science Foundation, project No. 200020-103847/1.

inverting f: namely, to set $\sigma = f^{-1}(h(m))$ for a carefully chosen hash function h. This invalidates the trivial existential forgery above and seems to work well in practice for a "crazy" enough h, such as SHA-1. This signature scheme is commonly called *Full Domain Hash* (FDH), and yields one of the simplest and most practical signature schemes known.

From a theoretical point of view, however, one can wonder if *it is possible to formally prove the security of FDH for some TDP f and hash function h*?

RANDOM ORACLE MODEL. Partially motivated by this question, in their seminal paper Bellare and Rogaway [3] introduced the *random oracle (RO) model* as a "paradigm for designing efficient protocols". It mathematically models h as a truly random function, which is freely available to all the parties including the adversary. In particular, under this idealized assumption Bellare and Rogaway formally confirmed the intuition of practitioners that the FDH signature scheme is existentially unforgeable in the RO model, for *any* TDP family $\mathcal{F}$. In fact, this was one of the first applications of the so called "random oracle methodology". Namely, one first formally analyzes and proves the security of a scheme like FDH in the RO model, and then practically *instantiates* this abstract scheme by replacing the ideal hash function h by some "real" implementation (such as SHA-1, or, more abstractly, some family of "real" functions $\mathcal{H}$), heuristically hoping that no security flaws will suddenly appear in the standard model. Therefore, it is clearly of fundamental importance to understand under which conditions one can *provably* instantiate the random oracles in the standard model. In particular, in this work we will concentrate on the FDH signature scheme, which, as we said, is one of the most basic and important applications of random oracles. Before addressing it in more detail, however, let us summarize what is known about this scheme in the RO model.

FDH IN RO MODEL. As we mentioned, Bellare and Rogaway showed that FDH is existentially unforgeable in the RO model, for any TDP family $\mathcal{F}$. On the other hand, a much tighter security reduction in the random oracle model was subsequently found by [10, 14] for a special class of TDP's: namely, those induced by a family of *claw-free permutation (CFP) pairs*[1] which luckily includes all popular families such as RSA. Moreover, Coron [11] subsequently showed that the above tighter reduction from CFP-induced TDP's is *optimal*, as long as the reduction treats the adversary as a "black-box" and irrespective of which particular TDP family $\mathcal{F}$ is used (e.g., even with RSA one cannot find a better black-box reduction in the RO model).

OUR GOAL. As we see, in the RO model very weak assumptions on the function family $\mathcal{F}$ are sufficient to prove the security of FDH: in fact, a single (although "ideal") hash function h simultaneously works for all such $\mathcal{F}$. Unfortunately, it is not hard to see that previously studied "realizable" properties of random oracles, such as collision-resistance, pseudorandomness (even verifiable; see [5]) or perfect one-wayness [9] are not sufficient *in general* to implement the random

[1] Such families consist of pairs of functions (f, g) for which is it infeasible to find a "claw" (x, y) satisfying $f(x) = g(y)$. One get an induced TDP family by taking f and "ignoring" g.

oracle h, even for specific function families $\mathcal{F}$ (i.e., one can come up with an artificial counter-example family $\mathcal{H}$ which nevertheless satisfies the given property but for which the FDH scheme is insecure with $\mathcal{F}$). On the other hand, many practitioners strongly believe that for most "real" TDP families $\mathcal{F}$ there should probably exist "good enough" hash functions like SHA-1 which would make FDH with $\mathcal{F}$ secure. Therefore, our main question is to examine *for which TDP families $\mathcal{F}$ can we provably instantiate FDH in the standard model?*

Initial Attempt. Let us make this question more precise. Given a function family $\mathcal{F}$, we are trying to design a hash family $\mathcal{H}$, such that a random h sampled from $\mathcal{H}$ will make FDH secure. Clearly, $\mathcal{H}$ should be allowed to depend on $\mathcal{F}$ (since assuming otherwise seems to place unfair restrictions on the signature designer). In fact, we also want to allow $\mathcal{H}$ to depend on a specific function f sampled from $\mathcal{F}$ (and whatever public information is associated with such f). For example, if $\mathcal{F}$ is induced by a family of claw-free permutation pairs (which, as we know, is very beneficial in the RO model), a random member f from $\mathcal{F}$ is sampled by choosing a random pair (f, g) from the CFP family, and then "ignoring" g. In this case it seems natural that the signature designer might want to use both f and g in designing the hash function h. For example, setting $h = g$ results in a signature scheme $f^{-1}(g(m))$ which is provably unforgeable under *no message attack*. Although the latter task can be easily achieved by other means (e.g., making h to be a random constant), this shows a potential utility one might get by using g in a less obvious manner.

Thus, the ambitious question would be to characterize the TDP families $\mathcal{F}$ for which one can choose an efficient $\mathcal{H}$ (depending of f) which would make FDH secure. Unfortunately, this seems to be an extremely difficult question given our current state-of-the-art knowledge. In particular, even for specific families such as RSA we do not seem to be able to say anything more meaningful than making a tautological assumption of the form "SHA-1 makes a good RSA-based FDH signature scheme".

Our Approach. Instead, we will ask a slightly more general question: which *security assumptions on $\mathcal{F}$* are sufficient to instantiate FDH in the standard model. For example, can we match the RO result stating that any TDP family can be instantiated? And, if not, maybe more restricted CFP-induced families can? Or maybe some other elegant assumption on $\mathcal{F}$ will be sufficient?

While a positive answer to these kind of questions would be even harder and more remarkable than the ambitious question asked about specific families $\mathcal{F}$ like RSA, the extra generality will allow us to get a *meaningful negative result*, which we believe is still very important. In particular, it will allow us to further realize the differences between the standard and the random oracle model. For example, we will see that being induced by a family of CFP's *by itself* is insufficient to instantiate FDH, in contrast to the RO model, where *nothing beyond this property is likely to be of any extra advantage*! Additionally, looking at the current general proofs of security of FDH in the RO model, it seems reasonable to hope that even in the standard model some natural and relatively general assumption on $\mathcal{F}$ might be sufficient for the proof to "go through" (with an appropriately chosen $\mathcal{H}$). In this regard, our approach allows us to further understand which security assumptions on $\mathcal{F}$ will certainly be *insufficient* (by themselves) to try instantiating FDH. In particular, if a given set of properties of $\mathcal{F}$ will be consistent with some assumption that we formally rule out, then more properties are needed. For example, it easily follows from our result below that one cannot instantiate FDH even if we assume $\mathcal{F}$ to be one-way against *any* distribution of "super-logarithmic" entropy. This is an *extremely* strong assumption that might appear quite useful for FDH upon the first look: for example, a similar assumption was recently utilized by Wee [28] to *successfully* obfuscate "equality" queries in the standard model, which was previously known only in the random oracle model [24]. Yet, we show that this assumption is insufficient for FDH.

Our Modeling. A bit more formally, we will study the question if there exists a *black-box reduction* (see [21]) from a given security assumption on $\mathcal{F}$ (such as being one-way or induced by CFP family, etc.) to the security of FDH. This means that all the relevant parties — adversary and "challenger" for the assumption (see below), potential forger for FDH, *as well as the designer of the hash family* $\mathcal{H}$[2] — should work given oracle access to f (and possibly even f^{-1}; see below). While seemingly restrictive, we believe this captures the essence of what it means to instantiate FDH given *any* $\mathcal{F}$ satisfying a given security assumption. Indeed, allowing non-black-box access to $\mathcal{F}$ essentially maps us back to the original "beyond-the-reach" question, where the designer of $\mathcal{H}$ can use some "extra" properties of $\mathcal{F}$ which do not follow from the security assumption alone. For example, we do not know how to show the insecurity of RSA-based FDH when the designer of the scheme chooses h to be SHA-1. In fact, most practitioners actually hope that the resulting scheme is secure!

In our modeling, a given security assumption is formalized by a "game" G between the adversary A and the *challenger* C. At the start of the game, a random f is chosen from $\mathcal{F}$ (possibly with some other public information), after which A and C engage in some protocol using *oracle access to* f, and the end of which C output 1 if the adversary has won and 0 otherwise. For example, in the one-wayness game defining plain TDP's the challenger simply asks A to invert $f(x)$, for a random x of its choice. Similarly, in the "claw-free" game defining $\mathcal{F}$ induced by some CFP family, C simply waits for A to provide a claw (x, y), where A can have oracle access to both f and its "twin" permutation g. Many other assumptions can be put in this framework as well.

Given such an abstract game G, we can look at the corresponding class of black-box permutation families $\mathcal{F}$ for which no polynomial time adversary can win with non-negligible probability. To argue a separation result for a given game G, we must essentially (see below) show that there exist a black-box family $\mathcal{F}$ such that (1) $\mathcal{F}$ is "black-box" secure with respect to G, but (2) $\mathcal{F}$ cannot be instantiated for the use in FDH, for any polynomial size circuit family $\mathcal{H}$ (which is allowed to depend on $\mathcal{F}$, but in a "black-box" manner).

[2] As we stated, it seems very restrictive not to allow such a dependency.

Our Main Result. Our main result is pretty general: we show than no game G between A and C can lead to an "instantiable" security assumption on $\mathcal{F}$, provided that a family of *truly random permutations* satisfies the security of G. Intuitively, it rules out all the assumptions involving "inverting" f on more or less arbitrary inputs (since random permutations are very hard to invert), or finding some inputs to f whose images satisfy some non-trivial relation (e.g., x and y such that $f(x) = f(y) \oplus 1$), etc. In fact, our main results extends even to games where the challenger is allowed to invert f to the attacker, as long as this is done for an a-priori bounded number of times.[3] To state this result differently, *any assumption on $\mathcal{F}$ which (1) is satisfied by a family of random permutations; and where (2) the challenger does not invert f on an a-priori unbounded number of points, is insufficient to instantiate FDH in the standard model.*

Thus, to generically instantiate FDH one must assume a property on $\mathcal{F}$ which is *not* satisfied by random permutations, such as being "homomorphic" or "self-reducible".

Other Results. As special cases, we rule out such instantiations based on plain TDP's, as well the sub-class of TDP's induced by CFP's, since both of those are easily seen to be satisfied by random permutations. In particular, this shows that more assumptions on $\mathcal{F}$ are needed in the standard model than what is sufficient for the best reduction in the RO model, giving yet another separation between the standard the the RO model (see related work below). As another interesting corollary, we notice that many cryptographic primitives such as collision-resistant hash functions, trapdoor commitments and even general signature schemes follow — in a black-box manner — from the existence of CFP families. Our separation result therefore shows that even assuming the existence of all these powerful primitives is not sufficient to build an "FDH-like" signature scheme (in a black-box manner), despite the fact that general, "non-FDH-like" signature schemes can be built! For example, there seem to be a "price to pay" for insisting on *inverting a trapdoor permutation* on the hash of the message, as opposed to applying to it any secure signature scheme on short messages: the latter is *provably secure* as long as the hash is collision-resistant (this is the famous "hash-then-sign" paradigm), while we show that much stronger assumptions seem to be required for the former.

We remark that our main impossibility result uses the full power of the chosen message attack, since our FDH breaker is allowed to ask more signing queries than the description of the hash function h. If we restrict our attention to the class of general TDP's (as opposed to *all* hard games satisfied by random permutations), we also strengthen our separation and show that there is no black-box reduction from the security of TDP's to the security of FDH even as

[3] Essentially, for a number of times slightly smaller than the number of signing queries the FDH forger is allowed to make. Without this restriction, one can define games modeling tautological assumptions of the form "SHA-1 makes FDH secure for a given $\mathcal{F}$" (which are trivially instantiable by setting h equal to SHA-1).

a *one-time* signature scheme, as long as the message space is super-polynomial in the security parameter.[4]

OUR TECHNIQUES. In both of our results we use an elegant "two-oracle" observation of Hsiao and Reyzin [21] for showing general black-box separation results. Applied to our setting, they show that it is sufficient to design an oracle F for $\mathcal{F}$ and another "breaking" oracle G, such that G does not help the attacker to win the game G with F, but always helps the forger to break the security of FDH (even if $\mathcal{H}$ can depend of F but not on G). In both of our results we use a family of random permutations to model the oracle F for our TDP family $\mathcal{F}$. The oracles G, however, are very different.

For our general separation result we use a novel oracle G which takes a description of the hash function h, and will forge the FDH-like signature of a new message *only if* the attacker can "prove" that he has oracle access to the FDH signing oracle. Remarkably, the oracle G is designed in such a careful way that its addition is literally of "no use" to the attacker in *any* game G! So if G was secure with random permutations, addition of G will not change this fact. Yet, it clearly breaks any FDH instantiation $\mathcal{H}$, since the forger has a "real" access to the signing oracle, and thus can successfully utilize G.

On the other hand, the oracle G for our TDP-specific separation is very different and is based on to the corresponding oracle by Simon [27] used to separate collision-resistant hash functions from one-way permutations.[5] In essence, this oracle returns collisions for any length-decreasing function h (which could depend of f), but in a careful way which does not allow the attacker to invert f. On the other hand, any collision clearly makes FDH insecure against one-message attack, as both of the colliding messages have the same signature. The main technical difficulty we have to resolve here is the fact that Simon's oracle only covers length-decreasing function families $\mathcal{H}$ (in fact, it is completely useless for most length-increasing hash families). Therefore, we have to non-trivially extend it to allow one break FDH for arbitrary function families $\mathcal{H}$, and yet without suddenly helping the adversary to invert f at a random point.

RELATED WORK. Our work is related to several important results [7, 26, 20, 8, 2] showing that various schemes provably secure in the random oracle model cannot be securely instantiated in the standard model. Canetti, Goldreich and Halevi [7, 8] gave concrete (although somewhat artificial) examples of general signature and encryption scheme with this property. Nielsen [26] considered the question of designing so called "non-committing encryption schemes" [6] capable of encrypting arbitrary number of messages, and showed that one cannot build such scheme *at all* in the standard model, although simple solutions in the random oracle model exist. Goldwasser and Tauman [20] concentrated on the soundness of the Fiat-Shamir heuristics [15], and showed a secure (although artificial) 3-round identification scheme which does not result in a secure signature scheme

[4] Otherwise, one can of course instantiate FDH by "hardwiring" an independent random challenge y_m to be the hash of m.

[5] For example, such oracle cannot be extended to cover CFP-induced TDP's, since it is known how to build collision-resistance hash functions from CFP's.

in the standard model, no matter how one implements the hash family. Finally, Bellare, Boldyreva and Palacio [2] showed a natural ElGamal-based key encapsulation mechanism for hybrid encryption which is secure in the random oracle model (for any symmetric-key component), but where for every real hash family one can come up with (artificial) symmetric-key encryption scheme making the overall hybrid scheme insecure.

We notice that an attractive feature of all these results as compared to our result, is that their separations are not black-box. However, our setting appears to be significantly more constrained as well. Intuitively, in all of the above results the syntax of the question allowed one enough freedom to adapt the scheme *after* the hash function h was chosen. While such adaptation was pretty non-trivial in each of the above works, our setting appears to be more restrictive. Namely, we must "commit" to a "real" TDP family $\mathcal{F}$ (possibly satisfying even more constraints), and then, given an arbitrary non-black-box function h *depending of* f, find some point m where we can invert $h(m)$! Without "reverse-engineering" such an h, the latter task seems quite hopeless to do (even using the signing oracle since it is hard to predict on which points it will invert f). Indeed, our black-box assumption essentially allows us to get a weak, but luckily sufficient "handle" to determine how h actually depends on f.

From a different perspective, our work naturally relates to a rich body of work on various black-box separations [22, 27, 18, 23, 17, 16, 11, 14, 21]. For example, we already pointed out how our breaking oracle for the case of general TDP's relates to the oracle of Simon [27], and how we use the simplified framework of Hsiao and Reyzin [21] to get our black-box separations. To the best of our knowledge, however, our work is the first to show a black-box separation result with respect to *instantiating random oracles in the standard model*, as opposed to separating different cryptographic assumptions from each other [22, 27, 18, 21] or showing lower bounds on the efficiency or exact security of various "black-box" reductions [23, 17, 16, 11, 14].

Finally, we already mentioned a complimentary recent work of Boldyreva and Fischlin [5], who considered the question of instantiating random oracles in various scenarios, including FDH, by popular families of "realizable" hash functions, such as verifiable pseudorandom functions [25] (VRFs). In particular, they showed that such VRFs cannot generically instantiate FDH, no matter which TDP family $\mathcal{F}$ is used.

2 Preliminaries

BASIC DEFINITIONS AND NOTATION. For a set $\mathcal{X}$ we denote by $x \in_R \mathcal{X}$ a value chosen uniformly at random from $\mathcal{X}$. A function $\mu : \mathbb{N} \to [0, 1]$ is negligible if for any $c > 0$ there is an n_0 such that $\mu(n) \leq 1/n^c$ for all $n \geq n_0$. We write $\mathsf{negl}(\cdot)$ as a shorthand for a *negligible* function.

TM is a shorthand for Turing-machine. We use the standard definition of probabilistic polynomial-time TMs (pptTM for short) and pptTMs with oracle access (opptTM for short). We say that something can be efficiently computed (relative to an oracle O) if it can be computed by a pptTM (by a opptTM with oracle access to O).

TRAPDOOR PERMUTATIONS. A *trapdoor permutation family* (TDP) is a pair of efficient algorithms $(KeyGen, F)$. $KeyGen$ is probabilistic and on input 1^n generates a key/trapdoor pair $KeyGen(1^n) \to (pk, td)$ where $F(pk, \cdot)$ implements a permutation $f_{pk}(.)$ over $\{0,1\}^n$ and $F(td, \cdot)$ implements its inverse $f_{pk}^{-1}(.)$.

SECURITY OF TDPS. The standard security property for TDPs is *one-wayness* which says that inverting is hard *without* the trapdoor, i.e. for any pptTM A

$$\mathbf{Pr}_{KeyGen(1^n)\to(pk,td),x\in_R\{0,1\}^n}[A(f_{pk}(x), pk) = x] = \mathsf{negl}(n).$$

A stronger security property is *claw-freeness* which says that given two independently sampled permutations it is hard to find a collision, i.e. for any pptTM A

$$\mathbf{Pr}_{i\in\{1,2\}:KeyGen(1^n)\to(pk_i,td_i)}[A(pk_1, pk_2) = (x_1, x_2) \text{ where } f_{pk_1}(x_1) = f_{pk_2}(x_2)] = \mathsf{negl}(n).$$

A TDP with this property is not a standard assumption, but it implies the following popular primitive.

CLAW-FREE PAIRS OF TRAPDOOR PERMUTATIONS. A *family of claw-free pairs of trapdoor permutations* (CFP) is a triple of efficient algorithms $(KeyGen, F, G)$ where $KeyGen$ is probabilistic and on input 1^n generates a key/trapdoor pair $KeyGen(1^n) \to (pk, td)$ for which $F(pk, \cdot)$ and $G(pk, \cdot)$ implement permutations $f_{pk}(.)$ and $g_{pk}(.)$ over $\{0,1\}^n$ respectively. $F(td, \cdot)$ and $G(td, \cdot)$ implement the inverses $f_{pk}^{-1}(.)$ and $g_{pk}^{-1}(.)$. The security property for CFPs requires that for any pptTM A

$$\mathbf{Pr}_{KeyGen(1^n)\to(pk,td)}[A(pk) = (x_1, x_2) \text{ where } f_{pk}(x_1) = g_{pk}(x_2)] = \mathsf{negl}(n).$$

HASH-FUNCTION. A family of hash-functions is a pair of efficient algorithms $(Index, H)$. $Index$ is probabilistic and on input 1^n generates an index $i \in \mathcal{I}_n$. For each $i \in \mathcal{I}_n$, $H(i, .)$ implements a function $h_i : \{0,1\}^* \to \{0,1\}^n$. A family of hash-functions is collision resistant if

$$\mathbf{Pr}_{Index(1^n)\to i}[A(1^n, i) = (x_1, x_2) \text{ where } x_1 \neq x_2 \text{ and } h_i(x_1) = h_i(x_2)] = \mathsf{negl}(n).$$

FULL-DOMAIN HASH (FDH). The FDH signature-scheme based on a trapdoor permutation family $(KeyGen_F, F)$ and a family of hash functions $(Index, H)$ is defined as a triple of functions $(KeyGen_{FDH}, sign, verify)$ where for security parameter n

- $KeyGen_{FDH}(1^n)$ first runs $KeyGen_F(1^n) \to (pk, td)$ and $Index(1^n) \to i$.[6] It outputs the triple (pk, td, i). The public-key of the signature scheme is (pk, i) and the secret key is (td, i).

[6] One would probably choose the randomness for $Index$ and for $KeyGen_F$ independent here, but we make no such assumption. In particular, (pk, td) and i can be arbitrarily correlated.

- $sign(m, td, i)$, the signature of a message $m \in \{0,1\}^*$ is $f_{pk}^{-1} h_i(m)$ (i.e. computed as $F(td, H(i,m))$).
- $verify(\sigma, m, pk, i)$, the verification function evaluates to 1 (with the meaning that the signature is valid) iff $f_{pk}(\sigma) = h_i(m)$ and to 0 otherwise.

SECURITY OF FDH. A FDH signature scheme as above is secure against an existential forgery in a chosen message attack if for any opptTM A

$$\mathbf{Pr}_{KeyGen_{FDH}(1^n)\to(pk,td,i)}[A^{sign(.,td,i)}(pk,i) \to (m,\sigma) \text{ where } verify(\sigma,m,pk,i) = 1 \text{ and } A \text{ did not make the oracle query } m] = \mathsf{negl}(n). \quad (1)$$

This means that A cannot come up with a valid signature/message pair for a message that he had not already signed by the signing oracle.

GAME. A game is defined by two opptTMs, a prover A and a challenger C, which have a common communication tape over which they can exchange messages. The challenger finally output a decision bit. We say that A wins the game if this bit is 1 and denote this event by $\langle A, C\rangle \to 1$.

HARD GAME. An opptTM C as above defines a *hard* game if no opptTM A can win the game when the oracle is instantiated with t (where $t = t(n)$ is implicitly defined by C and can be polynomial in n) uniform random permutations $\pi_1, \dots, \pi_t$ over $\{0,1\}^n$. I.e. C defines a hard game if for all opptTM A

$$\mathbf{Pr}[\langle A^{\pi_1(.),\dots,\pi_t(.)}(1^n), C^{\pi_1(.),\dots,\pi_t(.)}(1^n)\rangle \to 1] = \mathsf{negl}(n). \quad (2)$$

A TDP $(KeyGen, F)$ is secure for the hard game C if (2) is satisfied even if the random permutations are replaced with this TDP, i.e. for all pptTM A

$$\mathbf{Pr}_{\forall i=1\dots t: KeyGen(n)\to(pk_i,td_i)}[\langle A(pk_1,\dots,pk_t), C^{F(pk_1,\cdot),\dots,F(pk_t,\cdot)}(1^n)\rangle \to 1] = \mathsf{negl}(n).$$

Hard games capture many natural security properties, in particular

- one-wayness: $C^{f(\cdot)}(1^n)$ samples $x \in \{0,1\}^n$ uniformly at random and sends $f(x)$ to A. It outputs 1 iff it receives as the next message x.
- claw-freeness: $C^{f_1(\cdot),f_2(\cdot)}(1^n)$ just expects $x_1, x_2 \in \{0,1\}^n$ and outputs 1 iff $f_1(x_1) = f_2(x_2)$.

In the next section we will show that a TDP which is secure for *all* hard games cannot be black-box reduces the security of FDH. There does not exist a TDP which is secure for all hard games in the standard model,[7] but we show an impossibility result, and showing impossibility from such a hypothetic TDP implies

[7] Consider for example a game where C expects as input a circuit and then checks if the circuit computes the same value as its oracle on a few (n is enough) random inputs and outputs 1 only if this is the case. In the standard model A can always win this game by sending a circuit which computes C's oracle $F(pk,.)$. But this is a hard game as if the oracle is a random permutation, it will with high probability disagree with every polynomial size circuit on most inputs, and C will reject almost certainly.

impossibility for any assumption it implies. Then in section 4 we will extend the notion of hard games and give the challenger also access to inversion oracles $\pi_i^{-1}(.)$ which he may query at most polynomially many times (for some arbitrary but a priori fixed polynomial). With such games we can cover additional natural assumptions for TDPs, which will therefore be also insufficient to get a reduction to an FDH signature scheme.

3 No Reduction from any Hard Game

Theorem 1. *There is no black-box reduction from* a trapdoor permutations family which is secure for all hard games *to a FDH signature scheme secure against chosen-message attacks.*

More precisely, given a TDP $(KeyGen, F)$ which is secure for all hard games and any hash function family $(Index, H)$, the security of the signature scheme $sign(m) = f_{pk}^{-1}h(m)$ (where $KeyGen(1^n) \to (pk, sk)$ and $Index(1^n) \to i$) cannot be black-box reduced from the security of the TDP. Here the hash function can use the TDP as a black-box and the randomness used for $KeyGen$ and $Index$ can be arbitrarily correlated. Moreover, if we let $s(n) = \max\{|h_i| : i \in Range(Index(1^n))\}$ denote an upper bound on the size of a description of the hash function used, then the theorem even holds if we restrict the number of chosen message queries to $s(n)$ and the size of the message-space of the signature scheme to $s(n) + 1$.

As corollaries we get that any assumption on TDPs which can be formulated as a hard game will not be enough to get a reduction to FDH, e.g.

Corollary 1. *There is no black-box reduction from* claw-free pairs of trapdoor permutations *to a FDH signature scheme secure against chosen-message attacks.*

Proof (of Theorem 1). Following [21] (Lemma 1), as to rule out black-box reductions, it is enough to prove that there are two oracles F and G such that the following holds:

1. There is an opptTM D such that D^{F} implements[8] TDP.
2. There is an opptTM A such that $A^{\mathsf{F},\mathsf{G}}$ finds a forgery for any signature scheme of the form $sign(m) = f^{-1}(h^{\mathsf{F}}(m))$ in a chosen message attack, where f is the TDP implemented by D^{F} and h is any oracle circuit.
3. There is no opptTM B where $B^{\mathsf{F},\mathsf{G}}$ breaks the security of TDP implemented by D^{F}. This means that $B^{\mathsf{F},\mathsf{G}}$ cannot win any hard game C instantiated with this TDP with non-negligible probability.

Points 2 and 3 will follow from Lemmas 1 and 2 below. The first point is satisfied by the definition of the oracle F we will give, which implements TDP. This F *alone* is trivially a *secure* implementation of TDP. We then define a *breaking oracle* G

[8] Here implement has a purely functional meaning and does not imply any security assumptions.

for which we will show that it can be used to break any FDH scheme based on the TDP implemented by F but not the security of the TDP itself. The oracle G will simply provide a forgery (for the message $m = 0$) to any signature scheme of the form $sign(m) = f^{-1}(h^{\mathsf{F}}(m))$ (where $f \in \mathsf{F}$ and h is any oracle circuit), *but only if* it can be sure that the requesting party can compute those signatures herself (e.g. because she has access to the signing oracle $sign(m) = f^{-1}(h^{\mathsf{F}}(m))$ or knows the trapdoor for f). For this our G expects as input the values $f^{-1}(h^{\mathsf{F}}(m))$ for $m = 1 \ldots \ell$, where $\ell = |h|$. This choice of ℓ should make it impossible for an adversary to hardwire the outputs of h on all the inputs requested to values where she can invert f. However, there would still be at least two ways in which an adversary could abuse the oracle G to break the security of TDP implemented by F.

1. She could define an h such that the output of h^{F} collides on (some of) the requested inputs. Say $h^{\mathsf{F}}(i) = y$ for all $1 \leq i \leq \ell$ (where she knows $f^{-1}(y)$) and $h^{\mathsf{F}}(0) = z$ (where z could be a challenge in the one-wayness game). As she can provide the requested signatures $f^{-1}(h^{\mathsf{F}}(i))$ to G, G will output a forgery $w = f^{-1}(h^{\mathsf{F}}(0)) = f^{-1}(z)$ and she wins the game! To avoid this our G will check if there is such a collision before providing the forgery. This will not affect the usability of G to provide forgeries, as having a collision for h^{F} one can compute a forgery without the help of G anyway.
2. She could use f in the definition of h^{F} in a clever way, for example by choosing an h where $h^{\mathsf{F}}(m) = f(m)$ for $m \neq 0$ (then $f^{-1}(h^{\mathsf{F}}(m)) = m$) and $h(0) = z$ (where z could be a challenge in the one-wayness game). Our G will prevent this by checking whether in the computation of h^{F} on any of the requested inputs, the oracle for f is queried on an input x where $f(x) = h^{\mathsf{F}}(i)$ for $i, 1 \leq i \leq \ell$. Again, if this check fails we have a forgery as $x = f^{-1}(h^{\mathsf{F}}(i))$.

We will show that the two above checks are not only necessary, but already sufficient to guarantee that G cannot be used to break the security of TDP implemented by F.

Definition of F *(TDP secure for every hard game).* The definition of the oracle F is straight forward. For any $n \in \mathbb{N}$ choose $2^n + 1$ permutations $f_{0,n}, \ldots, f_{2^n-1,n}$ and t_n at random. Now F is defined as[9]

- $\mathsf{F}(td2pk, n, td) \rightarrow t_n(td)$
- $\mathsf{F}(eval, n, pk, x) \rightarrow f_{pk,n}(x)$
- $\mathsf{F}(invert, n, td, y) \rightarrow f^{-1}_{pk,n}(y)$

[9] With this F a TDP $(KeyGen, F)$ can be implemented as follows. $KeyGen(1^n)$ first samples a random trapdoor $td \in_R \{0,1\}^n$, then computes the corresponding public-key $\mathsf{F}(td2pk, n, td) \rightarrow pk$ and outputs (pk, td). $F(pk, .)$ and $F(td, .)$ are computed by $\mathsf{F}(eval, n, pk, .)$ and $\mathsf{F}(invert, n, td, .)$ respectively. Informally the reason that this TDP is secure for every hard game follows from the fact that a permutation, chosen at random from a set of 2^n randomly chosen permutations, is computationally indistinguishable from a truly random permutation. But if there was a hard game that this TDP could win, we could turn it into a distinguisher.

Definition of G *(Breaking Oracle).* The oracle G takes as input $(n \in \mathbb{N}, k \in \{0,1\}^n, h \in \{0,1\}^*, V)$ where h is (the description of) an oracle circuit.[10] This can be seen as a request for an existential forgery for the signature scheme $sign(m) = f_{pk,n}^{-1}(h^{\mathsf{F}}(m))$. The vector $V = [v_1, \ldots, v_{|h|}]$ is a "proof" that the requesting party can compute those signatures herself. We say that G accepts the input if the input has the correct form (as above) and

1. $f_{pk,n}^{-1}(h^{\mathsf{F}}(i)) = v_i$ for all $i = 1, \ldots, |h|$.
2. $v_i \neq v_j$ for all $1 \leq i < j \leq |h|$.
3. $\{h^{\mathsf{F}}(1), \ldots, h^{\mathsf{F}}(|h|)\} \cap Y_{\mathsf{F}}^h = \emptyset$ where Y_{F}^h is defined as

$$Y_{\mathsf{F}}^h = \{f_{pk,n}(x) | \, \exists i, 1 \leq i \leq |h|, h^{\mathsf{F}}(i) \text{ makes the query } \mathsf{F}(eval, n, pk, x)\} \quad (3)$$

If G accepts the input it outputs a forgery $f_{pk,n}^{-1}(h^{\mathsf{F}}(0))$ and $\perp$ otherwise.

G *Breaks any FDH Signature Scheme.* Now we will show that G breaks any FDH signature scheme based on F.

Lemma 1. *There is an* opptTM A *which outputs a forgery for any signature scheme* $sign(m) = f_{pk,n}^{-1}(h^{\mathsf{F}}(m))$ *with probability 1, i.e.*

$$\begin{aligned}\mathbf{Pr}[A^{\mathsf{F},\mathsf{G},sign(.)}(n, pk, h) \rightarrow (m, s) \quad &\textit{where} \quad s = f_{pk,n}^{-1}(h^{\mathsf{F}}(m)) \\ &\textit{and} \quad sign(.) \textit{ was not queried on input } m \,] = 1\end{aligned}$$

Proof (of Lemma). A must only check if h satisfies conditions 2 and 3. If one of them is not satisfied, this directly gives a forgery, otherwise A can use G to get a forgery. More formally A does the following:

- Compute $h^{\mathsf{F}}(1), \ldots, h^{\mathsf{F}}(|h|)$, doing this also compute Y_{F}^h as in (3).
  - If any of the $h^{\mathsf{F}}(1), \ldots, h^{\mathsf{F}}(|h|)$ collide we have a forgery: If say $h^{\mathsf{F}}(i) = h^{\mathsf{F}}(j)$, then query $sign(i)$ and output the forgery $(j, sign(i))$.
  - If $\{h^{\mathsf{F}}(1), \ldots, h^{\mathsf{F}}(|h|)\} \cap Y_{\mathsf{F}}^h \neq \emptyset$, then we have found an x and an i satisfying $f_{pk,n}(x) = h^{\mathsf{F}}(i)$ and thus have a forgery as $sign(i) = f_{pk,n}^{-1}(h^{\mathsf{F}}(i)) = x$.
- If none of the above is the case, then call the oracle $sign$ on inputs $1, \ldots, |h|$ and let $V = [sign(1), \ldots, sign(|h|)]$. Now query G on input (n, pk, h, V) to get a forgery for the message $m = 0$. ◊

G *does not break the security of* F. In this section we will prove that F is a *secure* implementation of a family of claw-free trapdoor permutations, even when given access to G, i.e.

[10] Usually the hash function h is given as a TM and not as a circuit, but a TM can be simulated by a circuit whose size is only polynomial in the running time of the TM. In particular for every efficient h there is an $m \in \mathbb{N}$ and a circuit h_c such that $\forall i \in \{0,1\}^m : h_c(i) = h(i)$ and $|h_c| < 2^m$, moreover such h_c can be efficiently computed and is sufficient here.

Lemma 2. *With probability 1 (over the choice of* F*) for any* opptTM *B and any hard game C (with $t = t(n)$ implicitly defined by C)*

$$\mathbf{Pr}_{\forall i=1\ldots t:KeyGen(n)\to(pk_i,td_i)}[(B^{\mathsf{F},\mathsf{G}}(pk_1,\ldots,pk_t), C^{f_{pk_1,n},\ldots,f_{pk_t,n}}(1^n))\to 1] \\ = \mathsf{negl}(n). \quad (4)$$

Proof (Proof Sketch of Lemma). If the oracle G was not there, then (4) would follow from the fact that for a random pk, $f_{pk,n}$ is computationally indistinguishable from a random permutation and that the randomly chosen permutation t_n is one-way (thus one cannot get the trapdoor $t_n^{-1}(pk)$).

Now we must argue that the presence of the oracle G will not help to win any hard game. This is not so obvious, after all G provides forgeries $f_{pk,n}^{-1}(h^{\mathsf{F}}(0))$ for an h of our choice. But to learn such a forgery we must find an accepting input (see the definition of G) for G. From Lemma 3 below it now follows that B cannot find such an accepting input for a random pk and thus will not learn anything about the $f_{pk_i,n}$'s that he could not compute on its own.[11] $\Diamond$

Lemma 3. *Let f be a random permutation on $\{0,1\}^n$ and $c \geq 1$ be a constant. For any oracle* TM *A which makes at most n^c oracle calls, we have (the probability is over the random permutation f)*

$$\mathbf{Pr}[A^f \to (h, x_1, \ldots, x_{|h|})] = \mathsf{negl}(n)$$

where $h, |h| \leq n^c$ is an oracle circuit and the output satisfies the conditions

1. $f^{-1}(h^f(i)) = x_i$ *for all* $i = 1, \ldots, |h|$.
2. $x_i \neq x_j$ *for all* $1 \leq i < j \leq |h|$.
3. $\{h^f(1), \ldots, h^f(|h|)\} \cap Y_f^h = \emptyset$ *where* Y_f^h *is defined as*

$$Y_f^h = \{f(x) | \exists i, 1 \leq i \leq |h|, h^f(i) \text{ makes the oracle query } x\}$$

Proof (of Lemma). Consider any oracle TM A where A^f makes n^c oracle queries. After having used up all his oracle queries A^f must come up with an output $(h, x_1, \ldots, x_{|h|})$ where h satisfies conditions 2 and 3. Below we prove that with overwhelming probability there does not even exist an h which satisfies conditions 2 and 3 *and* where A^f has made all the queries $x_1, \ldots, x_{|h|}$ satisfying condition 1. But in this case, even when choosing an h which satisfies conditions 2 and 3, A^f would still have to guess at least one x_i (i.e. $f^{-1}(h^f(i))$). The probability that it will guess correctly (i.e. this x_i will satisfy condition 1) is

[11] To make the proof and the statement of Lemma 3 simple (i.e. purely information theoretic), we will consider a computationally unbounded TM with oracle access to a truly random permutation which it can access a polynomial number of times, whereas Lemma 2 is about a opptTM and permutations chosen randomly from some family of exponential size. But as already mentioned, considering a random permutation is fine as a opptTM cannot distinguish a random permutation from $f_{pk,n}$ where $pk \in_R \{0,1\}^n$ anyway. And considering any computationally unbounded oracle TM (instead of only opptTMs) makes the lemma only stronger.

negligible.[12] We must now prove the above statement, i.e. that an h satisfying conditions 2 and 3 and where A^f made all the queries $x_1, \ldots, x_{|h|}$ satisfying condition 1 exists only with negligible probability. Let $X_f^A, |X_f^A| = n^c$ denote all oracle queries made by A^f, i.e.

$$X_f^A := \{x |\, A^f \text{ makes the oracle query } x\}.$$

Now consider any fixed oracle circuit $h, |h| \leq n^c$ which satisfies the conditions 2 and 3. Let $X_f^h = \{f^{-1}(y) | y \in Y_f^h\}$, i.e.

$$X_f^h := \{x |\, \exists i, 1 \leq i \leq |h|, h^f(i) \text{ makes the oracle query } x\}$$

and let

$$H := \{f^{-1}(h^f(1)), \ldots, f^{-1}(h^f(|h|))\}.$$

Condition 3 states that $f(H) \cap f(X_f^h) = \emptyset$, and as f is a permutation this is equivalent to

$$H \cap X_f^h = \emptyset.$$

Given X_f^h and conditioned on h^f satisfying condition 3, the set H is a random subset of $\{0,1\}^n \setminus X_f^h$. If condition 2 is satisfied then $|H| = |h|$ moreover $|X_f^h| \leq |h|^2 \leq n^{2c}$. Now the probability that $H \subseteq X_f^A$ can be upper bounded as (here the probability is over the random permutation f and for a fixed h conditioned on h^f satisfying conditions 2 and 3)

$$\mathbf{Pr}[H \subseteq X_f^A] = \prod_{i=0}^{|H|-1} \frac{|X_f^A| - |X_f^A \cap X_f^h| - i}{2^n - i - |X_f^h|}$$

$$\leq \left(\frac{|X_f^A|}{2^n - n^c - |X_f^h|} \right)^{|H|} = \left(\frac{n^c}{2^n - 2n^{2c}} \right)^{|h|}.$$

By taking the union bound over all oracle circuits $h, |h| \leq n^c$ we can now upper bound the probability that there exists an h satisfying conditions 2 and 3 and where A^f knows all x_i satisfying condition 1 as

$$\sum_{|h|=1}^{n^c} 2^{|h|} \left(\frac{n^c}{2^n - 2n^{2c}} \right)^{|h|} \leq \left(\frac{2n^c}{2^n - 2n^{2c}} \right) = \mathsf{negl}(n)$$

where in the first step we assumed that the sum takes it maximum for $|h| = 1$ which holds for all sufficiently large n. $\Diamond$ $\Box$

[12] It can easily be upper bounded by $1/(2^n - n^c - n^{2c})$: given the n^c oracle queries (not containing the query x_i) made by A^f and additionally the $\leq n^{2c}$ oracle queries made by h^f on inputs $1, \ldots, |h|$ (which will not contain the query x_i because of condition 3), x_i is a random variable with the uniform distribution over a set of size $\geq 2^n - n^c - n^{2c}$.

4 Hard Games with Inversions

In the last section we have seen that a TDP which is secure for all hard games (and thus has the one-wayness and claw-freeness security property) cannot be black-box reduced to a FDH signature scheme. In this section we will see that even a stronger notion of hard games does not allow for such a reduction. We extend the definition of a hard-game and allow (a limited number of) inversion queries.[13] To motivate this let us define one more security property for TDPs which can be modelled as such a game.

– A TDP has the *one-way with q(.)-inversions* security property if it is one-way, even with an oracle for f_{pk}^{-1} that can be used at most $q(n)$ times on any input except the challenge $f_{pk}(x)$, i.e.[14]

$$\mathbf{Pr}_{KeyGen(1^n)\to(pk,td),x\in_R\{0,1\}^n}[A^{f_{pk}^{-1}(.)}(f_{pk}(x),pk)=x]=\mathsf{negl}(n).$$

Hard game with $q(.)$ inversions. An opptTM C defines a hard game with $q(.)$ inversions if for a random permutation π and all opptTM A

$$\mathbf{Pr}[\langle A^{\pi(.)}(1^n),C^{\pi(.),\pi^{-1}(.)}(1^n)\rangle\to 1]=\mathsf{negl}(n) \tag{5}$$

where C may query the $\pi^{-1}(.)$ oracle at most $q(n)$ times. A TDP $(KeyGen,F)$ is secure for a hard game C with $q(.)$ inversions if

$$\mathbf{Pr}_{KeyGen(n)\to(pk,td)}[\langle A(pk),C^{F(pk,.),F(td,.)}(1^n)\rangle\to 1]=\mathsf{negl}(n).$$

The one-way with $q(.)$ inversions property is captured by such a game as follows:

– $C^{f(.),f^{-1}(.)}(1^n)$ samples $x\in_R\{0,1\}^n$ and sends $f(x)$ to A. Now C answers at most $q(n)$ queries $z\in\{0,1\}^n$ where $z\neq f(x)$ with $f^{-1}(z)$. C accepts and outputs 1 if it receives x as the $(q(n)+1)$'th message.

Lemma 2 is easily seen *not* to extend to hard games with $q(.)$ inversions already for $q(n)=O(n)$.[15] But if in the definition of the breaking oracle G we increase the number of requested signatures from $|h|$ to $|h|+q(n)$, then it is again impossible to find an accepting input for G and Lemma 2 can be shown to hold even for hard games with $q(.)$ inversions (using a similar strengthening of Lemma 3).

---

[13] For clarity of exposition we will consider the case where C expects only one permutation oracle, i.e. $t=1$.

[14] This property directly implies some others like security for the *known-target inversion problem* introduced in [4]. Here one gets $q(n)+1$ random challenges to invert and may use an inversion oracle on arbitrary inputs $q(n)$ times, i.e. once less than the number of challenges.

[15] For example A could win the one-way with cn inversions game (for some constant c) as follows. On challenge $y=f_{pk}(x)$ let $h(x)=x\oplus y$ (the c must satisfy $|h|\leq cn$). Now use the cn inversion queries to C to find an accepting input for the breaking oracle G, which will then provide the forgery $s=f_{pk}^{-1}(h(0)=y)$. Send $s=x$ to C and win the game.

Theorem 2. *For any polynomially bounded function $q(.)$, there is no black-box reduction from* a TDP family which is secure for all hard games with $q(.)$ inversions *to a FDH signature scheme secure against chosen-message attacks.*[16]

As corollaries we get that any assumption on TDPs which can be formulated as such a game will not be enough to get a reduction to FDH, e.g.

Corollary 2. *For any polynomially bounded function $q(.)$, there is no black-box reduction from a TDP satisfying the* one-way with $q(.)$ inversions security property *to a FDH signature scheme secure against chosen-message attacks.*

Finally, let us remark that in Theorem 2 it is necessary to have $q(.)$ bounded by some fixed polynomial. As if one allows a superpolynomial $q(n) \in n^{\omega(1)}$ then a TDP which is secure for all hard games with $q(.)$ inversions *can* be black-box reduced to a secure FDH signature scheme (note that this has a priori no practical consequences as such TDPs do not exist in the standard model). The main observation here is that the "existential forgery in a chosen message attack" (1) can be seen as a game where the challenger plays the role of the signing oracle $sign(m, td, i) \rightarrow f_{pk}^{-1}(h_i(m))$ and finally accepts if it receives a forgery from the prover A. We have not yet defined which FDH signature scheme to use in the above game. This scheme can not be arbitrary as we must make sure that this game is actually a *hard* game (i.e. no efficient A can win it when the oracles are instantiated with random permutations), but it is not difficult to construct a secure FDH scheme from random permutations $\pi_1, \pi_2, \ldots$ with only the signer having access to inversion oracles. For example, for a message space restricted to $\{0,1\}^n$, $sign(m) = \pi_1^{-1}(\pi_2(m))$ will already do it.

5 No Reduction from Trapdoor Permutations

We conclude the paper by observing that "the plain TDP assumption" implies an extreme black-box security limitation for FDH: not even security against a *one-chosen-message attack* can be achieved.[17]

Theorem 3. *There is no black-box reduction from* trapdoor permutation families *to a full-domain hash scheme secure against one-chosen-message attacks.*

For space reasons, we leave the proof of this theorem to a full version of the paper, here only discussing the key choice in the proof: that of the oracle G that breaks FDH but not TDP (cf. the proof of Theorem 1; F is the same as before).

G is partly based on the collision-finding oracle of Simon [27]. However, his "collision-finding" oracle only works for length-decreasing hash functions. To deal

[16] Moreover, if we let $s(n) = \max\{|h_i| : i \in Range(Index(1^n))\}$ denote an upper bound on the size of a description of the hash function used, then the theorem even holds if we restrict the number of chosen message queries to $s(n) + q(n)$ and the size of the message-space of the signature scheme to $s(n) + q(n) + 1$.

[17] A one-chosen-message attack is precisely an attack where at most one query to the signing oracle is allowed.

with arbitrary (potentially length-increasing) hash functions, we extend Simon's oracle to forge the FDH signature of a special input when no "good collision" to h was found: But we have to make sure that the inversion of $f_{pk,n}$ resulting from this forgery will not allow the attacker to invert $f_{pk,n}$ on its own challenge.

More specifically, G takes inputs of the form $(1^L, 1^t, \langle h \rangle, pk)$, where $\langle h \rangle$ is the description of a *deterministic* oracle TM. Such a query can be seen as a request for a forgery to signature scheme $sign(m) = f_{pk,n}^{-1}(h(m))$, here $n = |pk|$. G first checks if the running time of $h^{\mathsf{F}_0}(x)$ is $> \lfloor t/2 \rfloor$ for some $x \in \{0,1\}^L$ and all potential choices $\mathsf{F} = \mathsf{F}_0$ for the oracle F; if so, it outputs $\perp$ and stops. Otherwise, $u \in_R \{0,1\}^L$ is chosen and $w \equiv h^{\mathsf{F}}(u)$ is computed; then $v \in_R \{0,1\}^L$ is sampled *conditioned on* $h^{\mathsf{F}}(v) = w$. If $u = v$, $|w| = n$ *and* $L \geq \mu(n)$, where $\mu(n) = \log^2(n)$, F outputs $(u, v, y, f_{pk,n}^{-1}(w), s, \mathsf{inversion})$, where s describes the computations $h^{\mathsf{F}}(u)$ and $h^{\mathsf{F}}(v)$ (including all F-queries). Else, the output of G is $(u, v, w, s, \mathsf{collision})$, with s as above.

It is easy to see that with this oracle G one can forge $sign(m) = f_{pk,n}^{-1}(h(m))$ for any efficient h: Just query $\mathsf{G}(1^L, 1^t, \langle h \rangle, pk)$ (for appropriate L, t) to obtain u and v with $h(u) = h(v)$. If $u \neq v$, we can forge a signature for v by asking the signing oracle to sign u, which will also give a signature of v. If $u = v$, then G also outputs $f_{pk,n}^{-1}(h(u))$, which is a direct forgery (with no queries to its signing oracle).

More subtleties arise when showing that G does not help the adversary to invert F. In particular, they motivate the need for t and the "μ-test" in G. The former avoids that an adversary $A = A^{\mathsf{F},\mathsf{G}}$ receives the result of more oracle queries than she would have time to compute. As for the μ-test, it avoids that the TDP is *inverted on specific inputs*, for it makes negligible the probability that A could use G to invert some specific y of interest (e.g., the challenge in the one-wayness game). Indeed, for this to happen (1) a random u should map to y; *and* (2) a random preimage v of y ($v \in h^{-1}(y)$) should be u again. Now, it is easy to see that the probability of this happening is negligible indeed:

$$\mathbf{Pr}[u \in h^{-1}(y)]\mathbf{Pr}[v = u \mid v \in h^{-1}(y)] = \frac{|h^{-1}(y)|}{2^L} \frac{1}{|h^{-1}(y)|} = 2^{-L} \leq 2^{-\mu(n)}. \tag{6}$$

This simple fact turns out to be ultimately responsible for G not breaking the TDP property. More details will be given in the full version.

References

1. PKCS #1 v2.1, *RSA Cryptography Standard (draft)*, document available at `www.rsa security.com/rsalabs/pkcs`.
2. Mihir Bellare, Alexandra Boldyreva and Adriana Palacio. An Uninstantiable Random-Oracle-Model Scheme for a Hybrid-Encryption Problem. *EUROCRYPT 04*, pp. 171–188.
3. Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. *ACM CCS 93*, pp. 62–73.

4. Mihir Bellare, Chanathip Namprempre, David Pointcheval and Michael Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *J. of Cryptology*, **16** (3), pp. 185–215, 2003.
5. Alexandra Boldyreva and Marc Fischlin. Analysis of Random Oracle Instantiation Scenarios for OAEP and Other Practical Schemes. *CRYPTO 05*.
6. Ran Canetti, Uri Feige, Oded Goldreich and Moni Naor. Adaptively Secure Multi-Party Computation. *STOC 96*, pp. 22–24.
7. Ran Canetti, Oded Goldreich and Shai Halevi. The Random Oracle Methodology, Revisited. *STOC 98*, pp. 209–218.
8. Ran Canetti, Oded Goldreich and Shai Halevi. On the Random Oracle Methodology as Applied to Length-Restricted Signature Schemes. *TCC 04*, pp. 40–57.
9. Ran Canetti, Daniele Micciancio and Omer Reingold. Perfectly One-Way Probabilistic Hash Functions. *STOC 98*, pp. 131–140.
10. Jean-Sébastian Coron. On the Exact Security of Full Domain Hash. *CRYPTO 00*, pp. 229–235.
11. Jean-Sébastian Coron. Optimal Security Proofs for PSS and other Signature Schemes. *EUROCRYPT 02*, pp. 272–287.
12. Ivan Damgård. Collision-Free Hash Functions and Public-Key Signature Schemes. *EUROCRYPT 87*, pp. 203-216.
13. Whitfield Diffie and Martin Hellman. New directions in cryptography. IEEE Transactions on Information Theory 22 (1976), pp. 644–654.
14. Yevgeniy Dodis and Leonid Reyzin. On the Power of Claw-Free Permutations. *SCN 02*, pp. 55–73.
15. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. *CRYPTO 86*, pp. 186–194.
16. Rosario Gennaro, Yael Gertner and Jonathan Katz. Lower Bounds on the Efficiency of Encryption and Digital Signature Schemes. *STOC 03*, pp. 417–425.
17. Rosario Gennaro and Luca Trevisan. Lower Bounds on the Efficiency of Generic Cryptographic Constructions. *FOCS 00*, pp. 305–313.
18. Yael Gertner, Tal Malkin, and Omer Reingold. On the Impossibility of Basing Trapdoor Functions on Trapdoor Predicates. *FOCS 01*, pp. 126–135.
19. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold and Mahesh Viswanathan. The Relationship Between Public-Key Encryption and Oblivious Transfer. *FOCS 00*, pp. 325–335.
20. Shafi Goldwasser and Yael Tauman. On the (In)security of the Fiat-Shamir Paradigm. *FOCS 03*, pp. 102–114.
21. Chun-Yuan Hsiao and Leonid Reyzin. Finding Collisions on a Public Road, or do Secure Hash Functions Need Secret Coins? *CRYPTO 04*, pp. 92–105.
22. Russell Impagliazzo and Steven Rudich. Limits on the Provable Consequences of One-Way Permutations. *STOC 89*, pp. 44–61.
23. Jeong Han Kim, Daniel R. Simon and Prasad Tetali. Limits on the Efficiency of One-Way Permutation-Based Hash Functions. *FOCS 99*, pp. 535–542.
24. Ben Lynn, Manoj Prabhakaran and Amit Sahai. Positive Results and Techniques for Obfuscation. *EUROCRYPT 04*, pp. 20–39.
25. Silvio Micali, Michael Rabin and Salil Vadhan. Verifiable Random Functions. *FOCS 99*, pp. 120–130.
26. Jesper Buus Nielsen. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-Committing Encryption Case. *CRYPTO 02*, pp. 111–126.
27. Daniel Simon. Finding Collisions on a One-Way Street: Can Secure Hash Functions be Based on General Assumptions? *EUROCRYPT 98*, pp. 334–345.
28. Hoeteck Wee. On Obfuscating Point Functions. *STOC 05*, pp. 523–532.