How Risky is the Random-Oracle Model?

Gaëtan Leurent[1] and Phong Q. Nguyen[2]

[1] DGA and ENS, France. `http://www.eleves.ens.fr/leurent/`
[2] INRIA and ENS, France. `http://www.di.ens.fr/~pnguyen/`

Abstract. RSA-FDH and many other schemes secure in the Random-Oracle Model (ROM) require a hash function with output size larger than standard sizes. We show that the random-oracle instantiations proposed in the literature for such cases are weaker than a random oracle, including the proposals by Bellare and Rogaway from 1993 and 1996, and the ones implicit in IEEE P1363 and PKCS standards: for instance, we obtain a 2^{67} preimage attack on BR93 for 1024-bit digests. Next, we study the security impact of hash function defects for ROM signatures. As an extreme case, we note that any hash collision would suffice to disclose the master key in the ID-based cryptosystem by Boneh *et al.* from FOCS '07, and the secret key in the Rabin-Williams signature for which Bernstein proved tight security at EUROCRYPT '08. Interestingly, for both of these schemes, a slight modification can prevent these attacks, while preserving the ROM security result. We give evidence that in the case of RSA and Rabin/Rabin-Williams, an appropriate PSS padding is more robust than all other paddings known.

1 Introduction

The Random-Oracle Model (ROM), in which hash functions are viewed as random oracles, goes back to at least Fiat and Shamir [17]. Popularized by Bellare and Rogaway [1], it has been the subject of much debate. On the one hand, it is widespread in research papers and standards, because ROM schemes are usually more efficient, their security proofs can be based on well-studied assumptions, and can sometimes be tight. In fact, many standardized public-key schemes are, at best, only proven secure in the ROM, *e.g.* RSA-OAEP [2] and RSA-PSS [3].

On the other hand, the ROM is not an assumption (on the hash function): there is no random-oracle definition which a public function can hope to satisfy. A security proof in the standard model (SM) precisely gives assumptions which are sufficient to ensure security properties. By contrast, a ROM security proof only shows that an attack which does not break the proof assumptions must exploit a property not satisfied by the random-oracle simulation of the security proof. This allowed Canetti *et al.* [10] to build signature and encryption schemes which are secure in the ROM, but insecure for any (efficient) implementation of the random oracle, because such implementations can be simulated by a Universal Turing machine, and therefore be distinguished [33] from a random oracle. However, the constructions [10, 33, 20, 4] showing the limitations of the ROM are arguably "unnatural" and significantly differ from real-world constructions. Still, one should be careful with idealized security models: like all Merkle-Damgård/Davies-Meyer hash functions, MD5 was provably collision-resistant [50, 7] (up to the birthday bound) in the ideal cipher model (with respect to the block cipher underlying the compression function); yet, computing MD5 collisions only costs a few seconds now [43].

This stresses the importance of studying the actual security of schemes proven in the ROM. Unfortunately, very few ROM schemes have also been proven secure in the SM [42, 8]; and for several cases, there is evidence that a security proof in the SM is unlikely [34, 39, 15, 28]. Recent breakthroughs in the cryptanalysis of hash functions [48, 47, 45] have shown that standard hash functions like MD5 or SHA-1 are far from behaving like random oracles. However, the impact on the public-key world has been limited so far, with the exception of [45], which constructs two colliding X.509 certificates for different identities and public keys, and has recently been extended in [43] to construct a rogue CA certificate.

But to study the actual security, one needs to know how the random oracle will be instantiated in practice, should the scheme ever be used. Often, the random-oracle output size matches that of standard hash functions (like 160 bits for SHA-1) or the upcoming SHA-3. In this case, standard hash functions are most likely to be used, despite well-known properties of MD-iterated hash functions (such as the derivation of $h(m_1||m_2)$ from $h(m_1)$ and m_2) which make them easily differentiable from random oracles. But RSA-FDH [3] and many other ROM schemes (such as [24, 12, 13, 27, 6, 19, 9]) actually require a "non-standard" hash function. First, the output may not be a uniformally distributed bitstring: it could be residue classes, or elliptic curve points, *etc.*, fortunately it is known how to deal with such situations given an instantiation with arbitrary output $\{0,1\}^n$. But if the output bit-length is larger than standard sizes (*e.g.* RSA-FDH which needs at least 1024 bits), it is unclear how the oracle will be instantiated. To the best of our knowledge, the only proposals of random-oracle instantiations supporting arbitrary outbit bit-length are the following: two historical instantiations proposed by Bellare and Rogaway in their seminal papers [1] (on the ROM) and [3] (on FDH and PSS), recent constructions by Coron *et al.* in the full version of [14], and the instantiations implicit in PKCS#1 v2.1 [41] and IEEE P1363 [23] standards. It seems that none of these instantiations have been analyzed in the literature, except [14] in the indifferentiability framework of Maurer *et al.* [33].

This also raises the question of the impact of potential defects in random-oracle instantiations. When an article provides a ROM security proof, it usually does not say how to instantiate the random oracle, neither what might happen if the hash function does not behave like a random oracle. Assume that Alice implements a scheme ROM-secure under a well-known computational assumption. Several years later, the assumption still stands, but Alice learns that her random-oracle implementation is not as perfect as she thought: Should Alice worry? Are the risks confined to chosen-message existential forgery and ciphertext distinguishability? If Alice had the choice between two (equally efficient) schemes secure in the ROM under the same assumption, maybe Alice would rather choose the least risky one, in terms of robustness to hash function defects: the scheme with less stringent security requirements on the hash function, and the least affected if ever the requirements are not satisfied.

Our Results. We analyze the main proposals [1, 3, 41, 23] of random-oracle instantiations supporting arbitrary output bit-length. While none of these proposals made it clear what was exactly the expected security guarantee, one might argue that an instantiation with output bit-length n should offer $2^{n/2}$ resistance to collisions, and 2^n resistance to preimages, as required for SHA-3. We show that the proposals fall short of those bounds: for instance, for 1024-bit digests, we give a 2^{67} preimage attack on BR93 [1] and a 2^{106} collision attack on BR96 [3]. We note that the instantiations implicit in PKCS [41] and IEEE [23] standards are not collision-resistant: independently of the output size, collisions follow directly from SHA-1 collisions, which cost only 2^{61} [26, 47]. And we show that when applied to the compression functions of MD5 or SHA-1, the theoretical constructions of Coron *et al.* [14] are no more collision-resistant than MD5 or SHA-1 themselves. This highlights the difficulty of instantiating/simulating a random oracle, and motivates the study of the impact of hash defects on schemes secure in the ROM.

As a second contribution, we show that, while the ROM is useful to detect structural flaws, it can hide very different security requirements on the hash function, and very different damages in case of hash defects, independently of the computational assumption and the tightness of the security reduction. We illustrate this phenomenon with a well-known class of ROM schemes: padding-based signatures from trapdoor one-way functions, such as RSA, Rabin, Rabin-Williams and ESIGN. While it is often believed that a hash collision may at worst give rise to an existential forgery, we show that for several secure signatures proposed in the literature [3, 24, 9, 6, 19], collisions or slight hash function defects can have much more dramatic consequences, namely key-recovery attacks. However, this does not contradict the ROM security proofs, and does not mean that such signatures cannot be secure with a proper hash function. Our most interesting examples are related to Rabin and Rabin-Williams signatures, but the issues are not restricted to factoring-based schemes: in the full version, we show similar problems for a recent lattice-based signature scheme [19]. For instance, we remark that any hash collision discloses the master key in the ID-based cryptosystem of Boneh *et al.* [9], and the secret key in the Rabin-Williams signature scheme for which Bernstein [6] recently proved tight security, which was not mentioned in either [9, 6]. Interestingly, we show that a slight modification of the signing process can prevent our collision-based key-recovery attacks, while preserving the ROM security result. We give evidence that in the case of RSA and Rabin/Rabin-Williams, an appropriate PSS padding (with large salt) is more robust than all other paddings known, especially deterministic ones and randomized paddings with small randomness, including Katz-Wang [27] which achieve tightness.

To put things into perspective, consider the Rabin-Williams signature included in the IEEE P1363 standard [23]: it can be optionally deterministic or randomized, but the deterministic version has no tight reduction. Since tightness was a key factor in standardizing RSA-PSS over RSA-FDH, one might be tempted to replace this deterministic Rabin-Williams by its "tight" variant analyzed by Bernstein [6], but doing so would have led to a chosen-message key-recovery attack from any SHA-1 collision because such collisions provide collisions on the RO-instantiation of the P1363 standard. Alternatively, there would have also been a chosen-message key-recovery attack if IEEE P1363 had simply used the BR93 RO-instantiation [1], due to the preimage attack.

Some of the problems we discuss are related to the issue of how to derandomize a signature scheme. As a side-remark, we show that the derandomization technique proposed by Granboulan [22] to fix ESIGN during the NESSIE European project is not completely sound: when applied to ESIGN or DSA, it may leak the secret key. Finally, while this paper focuses on the ROM, we believe that in general, not just in the ROM, it is also interesting to identify necessary (possibly minimal) assumptions for security, and to assess the security impact when assumptions do not hold. This is useful when comparing cryptographic schemes, and can complement provable security results.

Road map. We assume the reader is familiar with hash functions, the ROM [1] and provable security for signatures [21]. In Sect. 2, we recall and analyze random-oracle instantiations for large output size. Next, we study the implications of hash function defects for ROM signatures based on a trapdoor one-way function and a padding: we recall such schemes in Sect. 3. In Sect. 4, we study the robustness of derandomized Rabin/Rabin-Williams signatures, which are used in [6] and the ID-based cryptosystem of [9]. In Sect. 5, we compare the robustness of RSA and Rabin/Rabin-Williams.

2 Random-Oracle Instantiations for Large Output

In this section, we describe and analyze random-oracle instantiations supporting arbitrary output bit-length that have been proposed in the literature, namely [1, 3, 14] and the instantiations implicit in the PKCS#1 v2.1 [41] and IEEE P1363 [23] standards. For completeness, we also briefly discuss collision-resistant hash functions such as [11, 32]. While some of the instantiations make use of MD5, that alone is insufficient to discard them. Indeed, though the collision-resistance of MD5 is seriously broken [48, 29], many usages of MD5 are not threatened yet: for instance, there is still no practical attack on HMAC-MD5.

2.1 The 1993 Instantiation by Bellare and Rogaway

Description. In their seminal paper on the ROM [1], Bellare and Rogaway gave guidelines to instantiate a random oracle (see [1, Sect. 6]), but the only explicit construction is the following one, which we call BR93:

- Let $h_4 : \{0,1\}^{512} \rightarrow \{0,1\}^{128}$ be the first compression function of MD5, that is the compression function evaluated with the initial IV of MD5.

- Let $h' : \{0,1\}^{256} \to \{0,1\}^{64}$ defined by $h'(x)$ being the first 64 bits of $h_4((xx) \oplus C)$, for a randomly chosen 512-bit constant C. The function h' defines a pseudo-random number generator $h''(x) : \{0,1\}^{192} \to \{0,1\}^*$ by counter as follows[3]: $h''(x) = h'(x\langle 0\rangle)||h'(x\langle 1\rangle)||h'(x\langle 2\rangle)\dots$ where $\langle i\rangle$ is the encoding of i into 64 bits.
- Finally, the BR93 instantiation of the random oracle is the truncation (prefix) of $h(x) : \{0,1\}^* \to \{0,1\}^*$ defined as follows. First, one applies a padding to x by adding a bit 1 and enough bits 0 to obtain a bitstring x' whose bit-length is a multiple of 128. Then, if we divide x' into 128-bit blocks as $x' = x'_0 \dots x'_{n-1}$, then $h(x) = h''(x'_0\langle 0\rangle) \oplus h''(x'_1\langle 1\rangle) \oplus \dots \oplus h''(x'_{n-1}\langle n-1\rangle)$, that is, $h(x)$ is the XOR of the n streams produced by each of the x'_i.

Weaknesses. We claim that BR93 is much weaker than a random oracle with respect to collision and preimage resistance, independently of the choice of the underlying compression function (MD5 here): the main idea is to adapt Wagner's generalized birthday algorithm [46]. Concretely, we give:

- a collision attack on $(k+2)r$ bits with complexity $2^k \cdot r2^r$ using messages of 2^k blocks. For instance, a collision attack on 1024 bits with messages of 2^{30} blocks costs $2^{30} \cdot 32 \cdot 2^{32} = 2^{67}$ elementary operations. If we limit the message size to 2^{14} blocks, the complexity is $2^{14} \cdot 64 \cdot 2^{64} = 2^{84}$. The complexity does not depend on the size of the underlying compression function.
- a preimage attack on $(k+1)r$ bits with complexity of $2^k \cdot r2^r$ using messages of 2^k blocks. For instance, a preimage attack on 1024 bits with messages of 2^{31} blocks costs $2^{31} \cdot 32 \cdot 2^{32} = 2^{68}$ elementary operations.

Generalized birthday. Recall Wagner's generalized birthday algorithm [46]. The basic operation is the general join $\bowtie_j$: $L \bowtie_j L'$ consists of all elements of $L \times L'$ such that their j least significant bits match:

$$L \bowtie_j L' = \left\{ l \oplus l' : (l,l') \in L \times L' \;\middle|\; (l \oplus l')^{[0..j-1]} = 0^j \right\}.$$

Assume that we are given several lists $L_0, L_1, \dots$, each of size 2^r. Our goal is to find $l_0 \in L_0, l_1 \in L_1, \dots$ such that $\bigoplus l_i = 0$. The idea is to join the lists using a binary tree. We build the first level with $L_{01} = L_0 \bowtie_r L_1$, $L_{23} = L_2 \bowtie_r L_3$, and so on. By the birthday paradox, these new lists should still contain about 2^r elements. On the next level, we build $L_{0123} = L_{01} \bowtie_{2r} L_{23}$. Since the elements of L_{01} and L_{23} already agree on their r lower bits, we are only doing a birthday paradox on the bits r to $2r$, so we still expect to find 2^r elements. If we start with 2^k lists, on the last level we end up with one list of 2^r elements which all begin with kr zeros. In this list, we expect to find two elements that agree on $2r$ extra bits, so that we have a collision on $(k+2)r$ bits.

Collisions. We now use Wagner's algorithm to find collisions on BR93. For each message block x'_i, we will consider 2^r possible values, and build a list L_i with the resulting $h''(x'_i\langle i\rangle)$. Then we can use Wagner's attack on these lists. A collision attack on $(k+2)r$ bits will have a complexity of $2^k \cdot r2^r$ using messages of 2^k

[3] The paper [1] actually says 224 instead of 192 for the input size of h', but that would be incompatible with the definition of h' as $224 + 64 = 288 > 256$.

blocks. For instance, a collision attack on 1024 bits with messages of 2^{30} blocks costs $2^{30} \cdot 32 \cdot 2^{32} = 2^{67}$ elementary operations. If we limit the message size to 2^{14} blocks, the complexity is $2^{14} \cdot 64 \cdot 2^{64} = 2^{84}$. Note that this complexity does not depend on the size of the underlying hash function.

Preimages. We can also use Wagner's algorithm to find preimages on BR93. If we want a preimage of H, we first replace L_0 by $L_0 \oplus H$. On the last level of the tree, we will still have one list of 2^r elements which all begins with kr zeros, but instead of looking for a collision on $2r$ extra bits, we look for an element with r extra zeroes. This element corresponds to a message x such that $H \oplus h(x) = H \oplus h''(x'_0\langle 0\rangle) \oplus h''(x'_1\langle 1\rangle) \oplus ...h''(x'_{2^k-1}\langle 2^k-1\rangle) =_{(k+1)r} 0$, *i.e.* the $(k+1)r$ first bits of $h(x)$ agree with H. A preimage attack on $(k+1)r$ bits will have a complexity of $2^k \cdot r2^r$ using messages of 2^k blocks. A preimage attack on 1024 bits with messages of 2^{31} blocks costs $2^{31} \cdot 32 \cdot 2^{32} = 2^{68}$ elementary operations.

2.2 The 1996 Instantiation by Bellare and Rogaway

Description. In their paper [3] on PSS, Bellare and Rogaway proposed another instantiation [3, App. A], which we call BR96: let H = MD5 or SHA-1, and define $h_{BR96}(x)$ as the appropriate truncation (prefix) of:

$$H(\texttt{const}\langle 0\rangle x)||H(\texttt{const}\langle 1\rangle x)||H(\texttt{const}\langle 2\rangle x)||\ldots$$

where the constant `const` should be unique to h. If another instantiation is needed, one should change `const`. BR96 is very close to a previous construction described in the OAEP paper [2] where H above is replaced by the 80-bit truncation of SHA-1, but that construction was not explicitly recommended to instantiate a random oracle, though it was used as a building block in an implementation of RSA-OAEP.

Weaknesses. We note that since H is a MD function, h_{BR96} can be viewed as the concatenation of MD functions, where each $H_i : x \mapsto H(\texttt{const}\langle i\rangle x)$ is a distinct iterative hash function, which implies:

– h_{BR96} can be distinguished from a random oracle. More precisely, BR96 suffers from the same extension problems as any MD function: if the output size of h_{BR96} is an exact multiple of that of H, and if m_1 has appropriate size, then $h_{BR96}(m_1||m_2)$ can be computed from $h_{BR96}(m_1)$ and m_2.
– h_{BR96} is weaker than a random oracle with respect to collision and preimage resistance, independently of the choice of the underlying hash function (except its output size), thanks to Joux's multicollision technique [25].

Recall that Joux [25] showed that the concatenation of two or more MD-iterated hash functions has roughly the same security as a single hash function. More precisely, if one concatenates k iterated hash functions with an internal size of n bits each, [25] finds a collision in the concatenation for a workload of $n^{k-1} \times 2^{n/2}$, and preimages for an imprecise workload of $\mathsf{poly}(n^k)2^n$.

A closer analysis of the collision attack shows that the cost $n^{k-1} \times 2^{n/2}$ can actually be reduced to $[(n/2)^{k-1} + (n/2)^{k-2} + \cdots + 1] \times 2^{n/2} \leq (n/2)^{k-1} \times$

$(n/2)/(n/2-1) \times 2^{n/2} \approx (n/2)^{k-1} \times 2^{n/2}$. And there seems to be a more efficient preimage attack by generalizing the basic preimage attack against two hash functions as follows. First, build a $2^{n^{k-1}/2^{k-2}}$-multicollision on the first hash function F_1, and look for an extra block that maps this multicollision to the target value of F_1. Then build a multicollision in F_2 using the messages of the first multicollision: each collision in F_2 requires a set of $2^{n/2}$ messages, which will be built from $n/2$ colliding pairs in F_1. Thus we should get a $2^{n^{k-2}}/2^{k-3}$-multicollision in F_1. We will also use the last n colliding pairs for a preimage search on F_1. This gives us a $2^{n^{k-2}}/2^{k-3}$-multicollision in $F_1||F_2$ which is also a preimage. We apply the technique iteratively to build a 2^n-multicollision for $F_1||F_2||...F_{k-1}$ which is also a preimage. If we compute F_k on the set of 2^n colliding messages, we expect to find one preimage against the full concatenation. The most expensive steps of this attack are the preimage search, because the collision finding steps all have complexity $O(n^k \times 2^{n/2})$. The preimage step on F_i requires to compute F_i on 2^n messages, which are made of n block pairs of length $n^{i-2}/2^{i-2}$ and one block of length $n^{i-2}/2i-3$. If we do an amortized analysis, each computation requires to hash 2 blocks from message pairs, and the final block, which gives a cost of $n^{i-2}/2^{i-4} \times 2^n$. The cost of the full preimage search is roughly equivalent to the cost of the last preimage search, which is $n^{k-2}/2^{k-4} \times 2^n$.

We now apply this to BR96. For instance, if H is MD5, we can find collisions in 1024 bits of the output with a workload of essentially $64^7 \cdot 2^{64} = 2^{106}$, where the colliding messages will be of length $64^7 = 2^{42}$ blocks; and we can find preimages of 1024 bits of the output with a workload of $128^6/2^4 \cdot 2^{128} = 2^{166}$. These complexities are clearly impractical and do not threaten [3], but they are much lower than the theoretical security of a 1024-bit random oracle.

For the same reason, BR96 is also malleable. For instance, we can create pairs of messages x_0, x_1 such that $H(\texttt{const}\langle i\rangle x_0) = H(\texttt{const}\langle i\rangle x_1)$ for all i's except the last one. We will build a multicollision set of $2^{n/4}$ such messages, and we expect to find one quadruplet such that $H(x_0) \oplus H(x_1) \oplus H(x_2) \oplus H(x_3) = 0$. In the full version, we show how this kind of malleability can be exploited to attack GPV [19].

As another example, consider the previous instantiation of BR96 for 1024-bit digests, using MD5. We can find two near-collisions where the most significant 384 bits are equal, with of a workload of essentially $64^2 \cdot 2^{64} = 2^{76}$. Such near-collisions give existential forgeries for the historical version [37] of ESIGN.

2.3 Recent Instantiations by Coron *et al.* (CDMP)

Description. Coron *et al.* [14] (CDMP) proposed several variations of Merkle-Damgård to build a random oracle from an (ideal) compression function or an (ideal) block-cipher using the Davies-Meyer mode. They proposed four variants of MD for *input* domain extensions (namely, *Prefix-Free Encoding*, *Dropping Some Output Bits*, *Using NMAC*, and *Using HMAC*) and one scheme (only in the full version of [14]) for *output* domain extension. The output extension scheme is similar to BR96, but the counter is included after the message (which is reminiscent of the MGF1 pseudo-random number generator used in several standards [23, 41]):

$$h_{CDMP}(x) = H(x\langle 0\rangle)||H(x\langle 1\rangle)||H(x\langle 2\rangle)|| \ldots$$

where H is one of the four input extension schemes. This choice is due to efficiency considerations, but we will see that it has a strong security impact. The main advantage of [14] is its security proof: all the constructions are proved indifferentiable from a random oracle (in the sense of Maurer *et al.* [33]), if the underlying compression function is a random oracle, or if it uses the Davies-Meyer mode with an ideal block cipher. However, no recommendation is given in [14] for the choice of the underlying compression function (or the underlying block cipher for Davies-Meyer). So strictly speaking, unlike [1, 3], there was no fully concrete proposal of a random-oracle instantiation: still, one may want to apply the constructions to usual compression functions.

Weaknesses. One should be careful not to overestimate the significance of indifferentiability security proofs: in practice, there is no ideal compression function. It was shown by [5] that none of the CDMP constructions necessarily preserve collision-resistance: they give (theoretical) examples of collision-resistant compression functions for which the resulting hash function is not collision-resistant.

While [14] was presented as a fix to the MD construction, we show that if one applies these fixes to MD5 or SHA-1, one can still find collisions in the new hash function (independently of the chosen output length) with the same cost as the original MD5 or SHA-1. This means that [14] does not address collision attacks on MD5 and SHA-1. To see this, we first show that the four input extensions are not collision resistant if applied to the compression functions of MD5 or SHA-1. This is trivial for *Dropping Some Output Bits*, *Using NMAC*, and *Using HMAC*, because these constructions are nested: an inner collision becomes an outer collision. So the only potentially tricky case is *Prefix-Free Encoding*, for which [14] proposed only two instantiations:

– prepend the message size as the first block. It turns out that MD5/SHA-1 collision attacks [48, 47] can be extended to this case, because the number of blocks of colliding messages produced is equal and already known in advance, and it is well-known that existing MD5/SHA-1 collision attacks can be extended to any given IV.
– use the first bit of each message block as a flag to distinguish the last message block. Since the number of blocks in MD5/SHA-1 colliding messages is very small, and the first bit of each block is random looking, we can simply produce random collisions until one has the required form.

Now, because of the iterated structure of the four input extensions, these collisions give rise to collisions in the output extension h_{CDMP}. More generally, while h_{CDMP} is indifferentiable from a random oracle if H is also indifferentiable, any collision in H becomes a collision in h_{CDMP} if H has an iterative structure like MD or the four input extensions: namely, $H(x_0) = H(x_1)$ implies $H(x_0\langle i\rangle) = H(x_1\langle i\rangle)$ and therefore $h_{CDMP}(x_0) = h_{CDMP}(x_1)$.

Hence, we have shown that if the CDMP constructions are applied to the compression functions of MD5 or SHA-1 for an arbitrary output size, the cost of producing collisions remains essentially the same as for MD5 or SHA-1 [26]. Of course, one could try to use different compression functions, but no concrete recommendation is given in [14].

2.4 Instantiations in PKCS and IEEE Standards

Description. No cryptographic standard currently specifies a random-oracle instantiation for arbitrary size. However, several instantiations are implicit in PKCS #1 v2.1 [41] and IEEE P1363 [23], because RSA-OAEP [2] and RSA-PSS [3] are standardized:

- RSA-OAEP requires two random oracles G and H with small input size (less than the RSA modulus), which are both instantiated in PKCS by the MGF1 pseudo-random number generator [41]. Recall that MGF1 is simply a hash function in counter mode like h_{CDMP}, except that the counter is over four bytes: $\text{MGF1}(x) = h(x\langle 0\rangle)||h(x\langle 1\rangle)||h(x\langle 2\rangle)||\ldots$, where h is either SHA-1 or a SHA-2.
- RSA-PSS also requires two random oracles G and H, but while G still has small input size, H has a small output size but possibly large inputs. In PKCS, H is instantiated by SHA-1 or SHA-2, and G is instantiated by MGF1.

Thus, none of the oracles required by RSA-OAEP and RSA-PSS have both a large input and output as would be required by RSA-FDH. Still, MGF1 is a potential random-oracle instantiation, because it supports arbitrarily large input and output.

There is another implicit instantiation in IEEE P1363 [23]. Indeed, it includes a Rabin-Williams signature using a variant of the PSS encoding [3] (as described in [24]) called EMSA-PSS in [41] and EMSA4 in [23]: the main difference between EMSA-PSS and PSS [3] (described in Sect. 3.2) is that the message is first hashed before going through the PSS encoding. But it is specified in [23] that the salt can optionally be set to zero, in which case "*the signature scheme is deterministic, similar to Full-Domain Hashing*". Thus, one can view EMSA-PSS with zero salt as an instantiation of a FDH: since the padding constants are zero, this amounts to essentially hash the message twice in a row, then apply MGF1; concatenate the output and the input of MGF1, and append the "BC" byte.

Weaknesses. The case of MGF1 has already been analyzed with the CDMP case in the previous subsection: using SHA-1 or any MD-iterated hash function, the cost of producing collisions in MGF1 remains as low as for the underlying hash function. And EMSA-PSS with zero salt is clearly no more collision-resistant than the underlying hash function. Note also that the "BC" byte makes it differentiable from a random oracle. Hence, independently of the output size chosen, finding collisions on the PKCS/IEEE instantiations costs as low as for the underlying hash function MD5 or SHA-1.

2.5 Provably collision-resistant hash functions

To conclude this section, we briefly discuss the case of hash functions which are provably collision-resistant under appropriate computational assumptions. Though not designed nor recommended to instantiate random oracles, they might be potential candidates since they usually support large output size. But it is folklore that none should be viewed nor used as a random oracle, because they have special properties which are not satisfied by a random oracle, typically malleability. Consider for instance two recent collision-resistant hash functions:

- VSH [11], which is collision-resistant provided that a certain problem related to factorization is hard. The output set is $\mathbb{Z}_N^{\times}$, where N is hard to factor.
- SWIFFT [32], which is (asymptotically) collision-resistant and one-way, provided that certain lattice approximation problems are hard. The smallest output size is 528 bits, but larger sizes are possible.

These functions are malleable in the following sense. In [32], it is noted that for any two inputs x_1 and x_2 such that $x_1 + x_2$ is a valid input, $\text{SWIFFT}(x_1) + \text{SWIFFT}(x_2) = \text{SWIFFT}(x_1 + x_2)$. By definition of VSH [11], it is easy to generate $M_0 \neq M_1$ such that $4\text{VSH}(M_0) \equiv \text{VSH}(M_1) \pmod N$ where N is the public modulus. More generally, for any product $s > 1$ of very small distinct primes (chosen among the primes used by the VSH compression function), it is easy to generate $M_0 \neq M_1$ such that $s^2\text{VSH}(M_0) \equiv \text{VSH}(M_1) \pmod N$.

We will see that such malleability relationships can be exploited to attack certain signature schemes. The malleability of SWIFFT can be exploited to attack the GPV signature [19] (see the full version), and the malleability of VSH can be exploited to attack Rabin and Rabin-Williams signatures. But we stress that neither VSH or SWIFFT were recommended to be used with these signatures.

3 Padding-based Signatures in the Random-Oracle Model

We study the impact of hash function defects for the class of ROM-secure signatures obtained by combining a trapdoor one-way function/permutation and a padding. More precisely, we consider secure versions of RSA, Rabin, Rabin-Williams and ESIGN with appropriate paddings: FDH [1], PSS [3], PFDH [12] and KW [27]. This section briefly recalls these components.

3.1 Signatures from Trapdoor One-Way Functions

Let σ denote the raw signature algorithm which, given as input a message $m \in \mathcal{M}$ outputs a signature $\sigma(m) \in \mathcal{S}$: the algorithm can be either deterministic or probabilistic. We consider signatures based on a trapdoor one-way function $f : \mathcal{S} \to \mathcal{M}$:

- If f is 1-to-1, we obtain a deterministic scheme with $\sigma(m) = f^{-1}(m)$.
- If f is many-to-1, we obtain a probabilistic scheme with $\sigma(m)$ selected uniformly at random in the set $f^{-}(m)$ of preimages: we require that the trapdoor enables such preimage sampling.

Verification checks that a given signature s belongs to $\mathcal{S}$ and that $f(s) = m$.
RSA. Let (N, e, d) be the usual RSA keys where the exponent d is secret. We have $\mathcal{M} = \mathcal{S} = \mathbb{Z}_N$, and take the RSA trapdoor permutation $f(x) = x^e$ defined over $\mathcal{M}$, whose inverse is $f^{-1}(x) = x^d$.
Rabin [40]. Let $N = pq$ be a usual RSA modulus. Then we take the squaring function $f(x) = x^2$, which is a 4-to-1 mapping from $\mathcal{S} = \mathbb{Z}_N^\times$ to the subgroup $\mathcal{M}$ of quadratic residues mod N. Inverting f is equivalent to factoring N.
Rabin-Williams [49]. This is a variation of Rabin signatures based on tweaks, using a modulus $N = pq$ such that $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$. This has two notable features: one can take $\mathcal{M} = \mathbb{Z}_N$ (rather than having to deal with quadratic residues), and one can obtain mappings onto $\mathcal{M}$ which are either 4-to-1 or 1-to-1, and whose inversion is equivalent to factoring. For any $m \in \mathcal{M}$, there are exactly four triplets $(e, f, s) \in \mathcal{S} = \{-1, 1\} \times \{1, 2\} \times \{0, \ldots, N-1\}$ such that $m \equiv efs^2$, and these so-called tweaked square roots can all be efficiently computed using p and q. Furthermore, the *principal* tweaked square root is the only square root such that e is 1 if m is a square modulo q, otherwise -1; f is 1 if em is a square modulo p, otherwise 2; and s is a square modulo $N = pq$. We thus obtain two trapdoor one-way functions, depending on the choice of $\mathcal{S}$:

- By taking the 4-to-1 mapping, we obtain the probabilistic signature scheme PRW.
- By taking the 1-to-1 mapping where $\mathcal{S}$ is confined to principal tweaked square roots, we obtain the deterministic signature scheme DRW. Ignoring technical details, this is essentially the Rabin-Williams used in IEEE P1363 [23].

ESIGN [37, 36]. We only give an informal description. ESIGN uses an RSA modulus of the form $N = p^2 q$ such that p, q have bit-length k, and N has bit-length $3k$. There is a small public exponent $e \geq 8$ not necessarily coprime with $\phi(N)$. The one-way function is the truncation of the RSA permutation $f(x) = x^e$ to its k most significant bits. This is a many-to-one mapping from $\mathcal{S} = \mathbb{Z}_N$ to $\mathcal{M} = \{0, \ldots, \lfloor N/2^{2k} \rfloor\}$, whose inversion problem is called AER [36].

3.2 Paddings

A padding Π specifies how to sign arbitrary messages $m \in \{0,1\}^*$: it may be deterministic or randomized. Then the signature is $\sigma(\Pi(m))$, with additional data in the case of PFDH. With the exception of PSS, all the following paddings use a full-domain hash h from $\{0,1\}^*$ to $\mathcal{M}$.
FDH [1]. This deterministic padding is simply $\Pi(m) = h(m)$.
PFDH [12]. One selects a salt $r \leftarrow_{\mathcal{R}} \{0,1\}^k$, and let $\Pi(m) = h(m||r)$ where h is a random oracle from $\{0,1\}^*$ to $\mathcal{M}$. The signature must include the salt r.
KW [27]. If m has never been signed, select a one-bit salt $r \leftarrow_{\mathcal{R}} \{0,1\}$, and let $\Pi(m) = h(r||m)$.
PSS [3]. Assume to simplify that $\mathcal{M} = \{0,1\}^k$ where $k \geq k_0 + k_1$ for some integers k_0 and k_1. Two random oracles $h : \{0,1\}^* \to \{0,1\}^{k_1}$ and $g : \{0,1\}^{k_1} \to \{0,1\}^{k-k_1}$ are used. We let g_1 be the function which on input $w \in \{0,1\}^{k_1}$ returns the first k_0 bits of $g(w)$, and let g_2 be the function which on input $w \in \{0,1\}^{k_1}$ returns the remaining $k - k_0 - k_1$ bits of $g(w)$. For any message $m \in \{0,1\}^*$,

one selects $r \in_R \{0,1\}^{k_0}$ and let $w = h(m\|r)$ and $r^* = g_1(w) \oplus r$. Finally, $\Pi(m) = w\|r^*\|g_2(w)$.
Standards. The PKCS [41] and IEEE P1363 [23] standards actually implement a slightly different version of PSS, called EMSA-PSS [24]: the main difference is that the message is first hashed before going through the PSS encoding. As mentioned earlier in Sect. 2.4, Rabin-Williams in IEEE P1363 is implemented as essentially DRW-PSS, but the salt can optionally be zero, in which case it becomes DRW-FDH with a specific RO-instantiation.

3.3 Derandomization

Several schemes [27, 6, 9] crucially require to derandomize the signature or the padding:

- The main scheme analyzed by Bernstein [6] is derandomized PRW-FDH with the requirement that if ever the same message is submitted twice, the same signature should be output.
- The ID-based cryptosystem of Boneh *et al.* [9] uses derandomized Rabin-FDH for mapping identities to secret keys.
- To implement RSA-KW, Katz and Wang [27] suggested to select the one-bit salt deterministically from a secret key and the message.

We will see that how the derandomization is performed has a big impact on the security. Bernstein [6] did not specify how this derandomization should be performed: he only mentioned that if ever the same message is submitted twice, the same signature should be output, otherwise an attacker would be able to compute two random tweaked square roots of the same element (by signing twice the same message), which discloses the secret key. But Katz and Wang discussed [27, Sect. 4.1] that issue in the context of RSA signatures, and proposed the following methods:

- KW1: select the nonce as $r = h'(K||m)$, where h' is an independent random oracle, K is an additional secret key and m is the message.
- KW2: select $r = h'(K||h(m))$ with the same notations as KW1.
- KW3: select $r = F_K(m)$ where F is a PRF and K is an additional secret key. This is the method used in the ID-based cryptosystem [9], which is shown to preserve the ROM security proof.

In order to derandomize ESIGN to fix its security proof (see [44, 38]), Granboulan [22] earlier proposed a slightly different method: select $r = \phi(h(m)||K||c)$, where ϕ is a one-way function with uniformly distributed output, K is an additional secret key, and c is an optional counter. The counter is necessary in ESIGN, because the signature process may actually fail for certain nonces. This derandomization technique was later adopted with a specific choice of ϕ in the revised submission [18] of ESIGN to the NESSIE European project.

3.4 Security Results

It is well-known that for any trapdoor permutation f, the signatures FDH, PFDH and PSS are all provably secure in the ROM under the hardness of inverting the permutation, but the security proof is loose [1, 16]. In the particular case of RSA, all these security proofs can be improved, by exploiting the multiplicativity of RSA: RSA-FDH remains loose [12, 1, 3], but RSA-PFDH and RSA-PSS have a tight reduction provided that the salt is sufficiently large (see [12, 16]). Surprisingly, RSA-KW also has a tight reduction [27]. These security results also apply to DRW (under the factoring assumption), because DRW uses a 1-to-1 mapping which is homomorphic (see [24]).

The picture is a bit different with Rabin and PRW, since they use 4-to-1 mappings, but squaring is homomorphic. The derandomized versions of Rabin-FDH and PRW-FDH have a tight reduction (see [9] for Rabin-FDH and [6] for PRW-FDH). Rabin-PSS has a tight reduction (see [3]). For a complete picture of the ROM security of all Rabin-Williams variants, see [6].

ESIGN-FDH [36] and ESIGN-PSS [30] both have a loose security proof under the AER assumption, but the proof of ESIGN-FDH requires a more restricted security model than usual (see [38, 44]). There is no tightness because the one-way function is not multiplicative.

4 Robustness of Derandomized Signatures

We now study the robustness of derandomized signatures described in Sect. 3, for the derandomization methods proposed in [22, 27, 9], which we described in Sect. 3.3. This derandomization is crucial for the tightness of certain security proofs. We focus on derandomized Rabin-FDH and PRW-FDH.

4.1 Soundness

First of all, one should make sure that the derandomization technique does not affect the security proof of the randomized scheme. For KW3, this was proved in [9, App. B] using of course the PRF assumption. And a similar argument can be proved for KW1 and KW2, but both require another random oracle, which is debatable if the goal is to understand what are the minimal assumptions.

For the fourth randomization method however, Granboulan [22] only gave an informal argument. In fact, we note that his method is not completely sound. More precisely, if the informal argument was correct, it would also apply to the choice $r = \phi(m||K||c)$. Now, assume that we take for ϕ an MD-iterated function for which it is easy to find collisions, but still, the function is one-way with uniformly distributed output: one potential example is SHA-1. Then an adversary could create a collision (m, m') where m and m' have the same length, which would imply that $\phi(m||K||c) = \phi(m'||K||c)$ for all K and c. This means that by querying the signature of m and m', the adversary would obtain two pairs message-signature which both used the same nonce r: in the case of ESIGN, this discloses the secret factorization of the modulus, and in the case of DSA, this clearly discloses the secret key. This means that [22] did not give the right assumption: instead of a one-way function with uniform output, one should consider a PRF such as in KW3.

4.2 Robustness to Collisions

We now look at the security if the full-domain hash h has defects, namely collisions. We show that any hash collision suffices to disclose the master key in the ID-based cryptosystem of Boneh *et al.* [9], and the secret key in the Rabin-Williams signature scheme for which Bernstein proved tight security [6], because of the way derandomization is performed. But we also show that these attacks can be prevented by slightly modifying the derandomization.

The idea is, of course, to obtain two random preimages of a known element $m \in \mathcal{M}$. Doing so for the trapdoor one-way function f of Rabin and PRW discloses the secret factorization of the modulus with probability $1/2$. Let $h : \{0,1\}^* \rightarrow \mathcal{M}$ be the random-oracle instantiation, to be paired with derandomized versions of Rabin or PRW. We have the following chosen-message key-recovery attack:

– Assume that the attacker is able to generate a collision (M_0, M_1) on h. Then $H(M_0) = H(M_1)$ with $M_0 \neq M_1$.
– The attacker queries the signing oracle on M_0 and M_1, and obtains the signature s_0 and s_1.
– Depending on how the derandomization is performed, we claim that s_0 and s_1 will be two random preimages of the same element $h(M_0) = h(M_1) \in \mathbb{Z}_N$, in which case it is easy to obtain the factorization of N with probability $1/2$. This is true in either of the following cases:
  - If one follows the informal method of Bernstein [6]: since M_0 and M_1 are different, the signer is not required to output the same preimage, so each preimage will be chosen at random.
  - If one follows KW1 [27], because $h(M_0) = h(M_1)$ is independent from $h'(K\|M_0) = h'(K\|M_1)$ where K denotes the secret key and h' is another random oracle.
  - If one follows KW3 [27] as in the ID-based cryptosystem [9], because $h(M_0) = h(M_1)$ is independent from $F_K(M_0) = F_K(M_1)$.

  But if one follows KW2 [27], the same preimage will be output, and the attack will fail. An alternative method to prevent the attack is to use the following variant of KW3: $r = F_K(h(m))$ so that collisions on h gives collisions on r. For both variants, the previous key-recovery attacks no longer work, but that does not mean that the schemes are immune to collisions: any hash collision gives rise to an existential forgery. The interest of KW2 and the KW3 variant is that they both decrease (but not remove) the security impact of collisions.

Independently of the random-oracle instantiation and the choice of the derandomization, these weaknesses also appear if one considers fault attacks. Indeed, a similar attack works if the adversary is able to submit twice the same message, and perturbate the calculation of the nonce, using fault attacks.

By contrast, the deterministic version DRW-FDH (and the one implemented in IEEE P1363 [23]) is immune to such attacks. It might help to see a concrete example. Assume that we plug the compression function of MD5 into the CDMP random-oracle construction [14] (see Sect. 2.4), and that we use this instantiation as a full-domain hash for DRW-FDH and derandomized PRW-FDH. Then, because of the indifferentiability framework [33], both signature schemes become provably secure under the factoring assumption, in the ideal cipher model (with respect to the MD5 block cipher) or in the random oracle model (with respect to the MD5 compression function). But in practice, there is an instant chosen-message key-recovery attack on the PRW scheme, which fails on the DRW one.

4.3 Robustness to malleability

The previous key-recovery attack can be adapted to malleability variants of collisions on the hash function. To simplify, consider first the case of derandomized PRW-FDH: a similar attack works for derandomized Rabin-FDH. Assume that the attacker is able to generate a pair (M_0, M_1) of distinct messages such that:

$$4h(M_0) \equiv h(M_1) \pmod{N}. \tag{1}$$

From Sect. 2, we know that this is easy if ever h is VSH [11] using the same modulus N, even though it might be hard to find collisions on VSH, but we stress that it was never suggested to use VSH for Rabin/Rabin-Williams that way: we give this example to show that solving (1) is not necessarily harder than finding collisions. Solving (1) is also possible (with more effort) if h is BR93 [1]: select any M_0 then apply the preimage attack to find M_1 satisfying (1). Again, the attacker queries the signing oracle on M_0 and M_1, which gives rise to tweaked square roots $(e_i, f_i, s_i) \in \{-1, 1\} \times \{1, 2\} \times \{0, \ldots, N-1\}$ of $h(M_i)$. Note though that there is a one-to-one correspondance between the four tweaked square roots of $h(M_0)$ and the four tweaked square roots of $h(M_1)$, thanks to (1). More precisely, if (e, f, s) is a tweaked square root of $h(M_0)$, then $(e, f, 2s \bmod N)$ is a tweaked square root of $h(M_1)$. This implies that $(e_0, f_0, 2s_0 \bmod N)$ and (e_1, f_1, s_1) are two "independent" random tweaked square roots of $h(M_1)$, which means that one can factor N with probability $1/2$.

Obviously, this attack is independent of the way derandomization is performed, and can be adapted to other malleability properties. For instance, similar attacks apply if one is able to find a pair (M_0, M_1) of distinct messages such that $h(M_0) \equiv -h(M_1) \pmod{N}$, or $k^2 h(M_0) \equiv h(M_1) \pmod{N}$ for some known $k \in \mathbb{Z}_N^\times$.

Furthermore, as opposed to collision attacks, the previous attack can be adapted to DRW-FDH. Starting again from (1), let $(e_i, f_i, s_i) \in \{-1, 1\} \times \{1, 2\} \times \{0, \ldots, N-1\}$ be the principal tweaked square root of $h(M_i)$. Because 4 is a square mod p and q, (1) implies that $e_0 = e_1$ and $f_0 = f_1$. Since $(e_0, f_0, 2s_0 \bmod N)$ is a tweaked square root of $h(M_1)$, we have $4s_0^2 \equiv s_1^2 \pmod{N}$, and therefore $s_1 \times (2s_0)^{-1} \bmod N$ is a square root of 1 mod N. But it must be a non-trivial square root because it has different Legendre symbols mod p and q: indeed, both s_0 and s_1 are squares mod N, while $\left(\frac{2}{p}\right) = 1$ and $\left(\frac{2}{q}\right) = -1$. Hence, this discloses the factorization of N. This attack can be generalized if congruence (1) is replaced by $k^2 h(M_0) \equiv h(M_1) \pmod{N}$ for any known $k \in \mathbb{Z}_N^\times$ such that $\left(\frac{k}{p}\right) \neq \left(\frac{k}{q}\right)$, which is slightly more restrictive than the PRW case.

There are similar attacks for the Rabin case.

4.4 Robustness to preimages

The previous attacks also show that both derandomized PRW-FDH and DRW-FDH become strongly insecure if the full-domain hash function h is not one-way, like BR93 [1]. Alternatively, one can simply select $(e, f, s) \in \{-1, 1\} \times \{1, 2\} \times \{0, \ldots, N-1\}$ uniformly at random, and compute $m = efs^2 \pmod N$. By inverting h, one obtains a message M such that $m = h(M)$. Finally, by signing the message M with either DRW-FDH or derandomized PRW-FDH, one will obtain another tweaked square root of m (principal or not), which will disclose the factorization of N with probability at least $1/2$ because (e, f, s) is a random tweaked square root. A similar attack works for the Rabin case.

5 Compared Robustness of Signatures

5.1 RSA Signatures

The PKCS#1 v2.1 standard [41] uses RSA-PSS since Sept. 1999 (or more precisely, the variant RSA-EMSA-PSS [24] of RSA-PSS), and it has been reported that one of the main reasons why RSA-PSS was selected over RSA-FDH was the tightness of the security proof. If tightness was the main factor, one might now be tempted to select RSA-KW over RSA-PSS, because the salt in RSA-KW is reduced to one bit (which can be deterministically derived from the secret key and the message). However, by comparing the robustness of RSA signatures with respect to potential defects in the random-oracle instantiation, a different picture emerges.

Robustness to collisions. Because RSA-FDH and RSA-EMSA-PSS are hash-and-sign schemes, they do not tolerate collisions: any collision obviously leads to a chosen-message existential forgery. Similarly, any collision leads to a chosen-message existential forgery on RSA-KW, with probability 1/2 because of the one-bit salt. One may think that the probabilistic schemes RSA-PFDH and RSA-PSS are more robust. In this direction, Numayama *et al.* [35] showed that RSA-PFDH tolerates collisions in a weakened ROM, but their model does not take into account MD-iterated hash functions. We observe that if h is a MD-iterated hash function, then any collision in h with the same number of blocks gives rise to a chosen-message existential forgery on RSA-PFDH and RSA-PSS. This is because RSA-PFDH and RSA-PSS both use $h(m||r)$. And if $h(m_1) = h(m_2)$ where m_1 and m_2 have the same number of blocks, then $h(m_1||r) = h(m_2||r)$ for any r. This implies that for both RSA-PFDH and RSA-PSS, any signature of m_1 is also valid for m_2. It can be noted that if RSA-PFDH and RSA-PSS had used $h(r||m)$ instead of $h(m||r)$, then the ROM security proofs would remain valid, but the previous attack would fail.

Robustness to preimages. It is easy to prove that if the full-domain hash is not one-way, then there are chosen-message universal forgery attacks on RSA-FDH, RSA-PFDH and RSA-KW. On the other hand, preimages in h do not seem to provide stronger attacks than chosen-message existential forgeries on RSA-PSS.

Conclusion. While RSA-KW has a much better security reduction than RSA-FDH, there are essentially the same attacks on both RSA-KW and RSA-FDH as soon as there are defects in the full-domain hash. On the other hand, RSA-PSS with a large salt seems more robust than all other paddings, especially if one uses $h(r||m)$ instead of $h(m||r)$. This differs from the conclusion of [31], where it was argued that RSA-FDH was the best method known to sign with RSA.

5.2 Rabin and Rabin-Williams

Based on Sect. 4, the advantages of Rabin over Rabin-Williams are unclear from a security point of view. There are two benefits with Rabin-Williams: one can select $\mathcal{M} = \mathbb{Z}_N$, and one can use a 1-to-1 mapping instead of a 4-to-1 mapping. This 1-to-1 mapping avoids collision attacks or the fault attacks on derandomization: This suggests that DRW is preferable to both PRW and Rabin. And for the same reason as for RSA, a PSS padding with large salt seems more robust than all other paddings. Furthermore, when using Rabin or PRW, the size of the salt is extremely important to avoid key-recovery replay attacks. By submitting many times the same message, an adversary would obtain two random preimages of the same element, hence the factorization. We illustrate this phenomenon in the full version, with a converse to the security proof of [3].

Acknowledgements. We thank Dan Boneh, Charles Bouillaguet, Louis Granboulan, Eike Kiltz, Kaoru Kurosawa, Alfred Menezes, Martijn Stam and the anonymous reviewers for their comments. Part of this work was supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 ECRYPT II.

References

1. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: CCS '93, ACM Press (1993) 62–73
2. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: EUROCRYPT '94. LNCS 950, Springer (1995)
3. Bellare, M., Rogaway, P.: The exact security of digital signatures - how to sign with RSA and Rabin. In: EUROCRYPT '96. LNCS 1070, Springer (1996)
4. Bellare, M., Boldyreva, A., Palacio, A.: An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In: EUROCRYPT '04. LNCS 3027, Springer (2004)
5. Bellare, M., Ristenpart, T.: Multi-property-preserving hash domain extension and the EMD transform. In: ASIACRYPT '06. Volume 4284 of LNCS., Springer (2006)
6. Bernstein, D.J.: Proving tight security for Rabin-Williams signatures. In: EUROCRYPT '08. LNCS, Springer (2008)
7. Black, J., Rogaway, P., Shrimpton, T.: Black-box analysis of the block-cipher-based hash-function constructions from PGV. In: CRYPTO '02. LNCS 2442, Springer (2002)
8. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: EUROCRYPT '04. LNCS 3027, Springer (2004)

9. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS '07, IEEE Computer Society (2007) 647–657
10. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM **51**(4) (2004) 557–594 (electronic) Preliminary version at STOC '98.
11. Contini, S., Lenstra, A.K., Steinfeld, R.: VSH, an efficient and provable collision-resistant hash function. In: EUROCRYPT '06. LNCS 4004, Springer (2006)
12. Coron, J.S.: Optimal security proofs for PSS and other signature schemes. In: EUROCRYPT '02. Volume 2332 of LNCS., Springer (2002) 272–287
13. Coron, J.S.: Security proof for partial-domain hash signature schemes. In: CRYPTO '02. Volume 2442 of LNCS., Springer (2002) 613–626
14. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: CRYPTO '05. LNCS 3621, Springer (2005) Full version available on Dodis' webpage.
15. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: CRYPTO '05. Volume 3621 of LNCS., Springer (2005) 449–466
16. Dodis, Y., Reyzin, L.: On the power of claw-free permutations. In: SCN '02. Volume 2576 of LNCS., Springer (2003) 55–73
17. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO '86. LNCS 263, Springer (1987)
18. Fujisaki, E., Kobayashi, T., Morita, H., Oguro, H., Okamoto, T., Okazaki, S.: ESIGN-D specification. Submission to the NESSIE European Project, available on the NESSIE webpage (2002)
19. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC '08, ACM (2008)
20. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: FOCS '03, IEEE Computer Society (2003)
21. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2) (1988) 281–308
22. Granboulan, L.: How to repair ESIGN. In: SCN '02. Volume 2576 of LNCS., Springer (2003) 234–240
23. IEEE: P1363: Standard specifications for public-key cryptography. Available at `http://grouper.ieee.org/groups/1363/`.
24. Jonsson, J.: Security proofs for the RSA-PSS signature scheme and its variants. Report 2001/053 of the Cryptology ePrint Archive (2001)
25. Joux, A.: Multicollisions in iterated hash functions. application to cascaded constructions. In: CRYPTO '04. Volume 3152 of LNCS., Springer (2004) 306–316
26. Joux, A., Peyrin, T.: Hash functions and the (amplified) boomerang attack. In: CRYPTO '07. Volume 4622 of LNCS., Springer (2007) 244–263
27. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: CCS '03, ACM (2003)
28. Kiltz, E., Pietrzak, K.: On the security of padding-based encryption schemes (or: Why we cannot prove OAEP secure in the standard model). In: EUROCRYPT '09. LNCS, Springer (2009)
29. Klima, V.: Tunnels in Hash Functions: MD5 Collisions Within a Minute. Cryptology ePrint Archive, Report 2006/105 (2006)
30. Kobayashi, T., Fujisaki, E.: Security of ESIGN-PSS. IEICE Transactions **90-A**(7) (2007) 1395–1405
31. Koblitz, N., Menezes, A.J.: Another look at "provable security". J. Cryptology **20**(1) (2007) 3–37
32. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A modest proposal for FFT hashing. In: FSE '08. Volume 5086 of LNCS., Springer (2008)

33. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: TCC '04. Volume 2951 of LNCS., Springer (2004) 21–39
34. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: CRYPTO '02. LNCS 2442, Springer (2002)
35. Numayama, A., Isshiki, T., Tanaka, K.: Security of digital signature schemes in weakened random oracle models. In: PKC '08. LNCS 4939, Springer (2008)
36. Okamoto, T., Fujisaki, E., Morita, H.: TSH-ESIGN: Efficient digital signature scheme using trisection size hash. Submission to IEEE P1363a (1998)
37. Okamoto, T.: A fast signature scheme based on congruential polynomial operations. IEEE Transactions on Information Theory **36**(1) (1990) 47–53
38. Okamoto, T., Stern, J.: Almost uniform density of power residues and the provable security of ESIGN. In: ASIACRYPT '03. LNCS 2894, Springer (2003)
39. Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In: ASIACRYPT '05. Volume 3788 of LNCS., Springer (2005) 1–20
40. Rabin, M.: Digital signatures and public key functions as intractable as factorization. Technical report, MIT Laboratory for Computer Science (1979) TR-212.
41. RSA Laboratories: PKCS #1 v2.1: RSA cryptography standard. June 14, 2002
42. Shoup, V.: Using hash functions as a hedge against chosen ciphertext attack. In: EUROCRYPT '00. LNCS, Springer (2000) 275–288
43. Sotirov, A., Stevens, M., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D.A., de Weger, B.: Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In: CRYPTO '09. LNCS, Springer (2009)
44. Stern, J., Pointcheval, D., Malone-Lee, J., Smart, N.P.: Flaws in applying proof methodologies to signature schemes. In: CRYPTO '02. LNCS 2442, Springer (2002)
45. Stevens, M., Lenstra, A.K., de Weger, B.: Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities. In: EUROCRYPT '07. Volume 4515 of LNCS., Springer (2007)
46. Wagner, D.: A generalized birthday problem. In: CRYPTO '02. Volume 2442 of LNCS., Springer (2002) 288–303
47. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: CRYPTO '05. LNCS 3621. Springer (2005)
48. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: EUROCRYPT '05. LNCS 3494, Springer (2005)
49. Williams, H.C.: A modification of the RSA public-key encryption procedure. IEEE Trans. Inform. Theory **26**(6) (1980) 726–729
50. Winternitz, R.S.: A secure one-way hash function built from DES. In: Proc. IEEE Symposium on Security and Privacy. (1984) 88–90