ECRYPT Hash Workshop 2007 (May 24–25 2007, Barcelona, Spain)
V. Rijmen, chair. Workshop Proceedings, pages 4–19.

Revisiting Security Relations Between Signature Schemes and their Inner Hash Functions

Emmanuel Bresson[3] and Benoît Chevallier-Mames[1] and Christophe Clavier[1] and Blandine Debraize[1] and Pierre-Alain Fouque[4] and Louis Goubin[1] and Aline Gouget[1] and Gaetan Leurent[4] and Phong Nguyen[4] and Pascal Paillier[1] and Thomas Peyrin[2] and Sébastien Zimmer[4]

[1] Cryptography and Innovation, Gemalto Security Labs
[2] France Telecom Division R&D
[3] DCSSI
[4] Ecole Normale Supérieure

Abstract. After years of almost full confidence in the security of common hash functions such as MD5 and SHA-1, the cryptographic community is now facing the unprecedented threat of seeing practical security applications succumb to concrete attacks. A way to cope with this crisis is to fasten the development of new hash functions, but another crucial task is to assess the implications these attacks on hash functions may have on cryptographic systems. This paper reports a thorough investigation on how recent attacks on hash functions impact the security of signature schemes. We suggest the notion of probabilistic hash-and-sign signatures and further classify signature schemes into various related categories which allow us to identify completely the nature of security relations between signature schemes and their inner hash functions. We also determine how using iterated hash functions a la Merkle-Damgård impacts the security of deterministic (resp. probabilistic) hash-and-sign signatures. We confirm that the security gain inherent to using the probabilistic hash-and-sign paradigm may be lost completely if instantiated with a Merkle-Damgård hash function and unwise operating mode.

1 Introduction

Undoubtedly, hash functions constitute an essential component of all sorts of systems and constructions in cryptography, should these stand in the public-key or secret-key paradigm. From basic primitives (encryption, signatures, commitments, etc) to advanced protocols (fair exchange, ecash, multiparty computations and so forth), they have spread all over the place, appreciated for the many and very different properties one expects these functions to fulfill. In [32] more than 50

Since the early days of modern cryptography however, the design of hash functions has remained mostly heuristical and for a large part based on ideas and concepts arising from the design of block ciphers. Even if most hash functions were conceived with little more than a paper and a pencil, a large confidence in the security of some hash functions such as SHA-1 has long been shared in the cryptographic community. The very few concrete attacks on trusted hash functions that came to public attention for more than a decade definitely played a role in reinforcing this belief.

Recently however, initiated by the works of Chabaud and Joux [10] and later Biham and Chen [4], powerful attack techniques were suddenly discovered and reported by Wang [38]. Applicable to a wide range of hash functions and leading sometimes to unexpectedly low workfactors, new attacks are now progressively emerging which collect and assemble techniques of independent nature [39,40,15,25,28]. It is now clear that hash functions designs cannot be trusted without adequate public scrutiny and the confidence is low that commonly used hash functions such as the SHS standard SHA-1 can withstand cryptanalytic efforts for ever [16].

How broken hash functions impact cryptosystems. Because of the systematic appearance of hash functions in cryptographic constructions, the security of a large variety of systems is now at risk [21,37]. However the role played by hash functions in the overall security of a system is, in many constructions, so badly understood that it is not obvious at all to see how that system suffers from being based on broken hash functions. Does the security of the OAEP padding (used in conjunction with a trapdoor permutation to yield random-oracle secure encryption) vanish when a collision on one of the inner hash functions is discovered?

The goal of our work is to explore the interplay between these two fundamental concerns: the security of a cryptographic system $\mathcal{S} = \mathcal{S}[H_1, \ldots, H_n]$ based on hash functions $H_1, \ldots, H_n$ and the security of $H_1, \ldots, H_n$.

Assuming for instance that a scheme $\mathcal{S}$ involves a unique hash function H, we want to determine how the security of H relates to the one of $\mathcal{S}$. This amounts to computationally connect security notions for $\mathcal{S}$ with security notions for H. Our approach is to exhaust all these connections (at least all the ones we can see) which we categorize into the four following types:

1. an attack, which we define as a polynomial reduction $\mathsf{Break}(H) \geq \mathsf{Break}(\mathcal{S})$ for well-defined security notions[1]. In this case, our reduction makes explicit how an attack of a given type on the hash function is enough to break the scheme in a prescribed way;
2. a security proof that is, a polynomial reduction $\mathsf{Break}(H) \leq \mathsf{Break}(\mathcal{S})$. Here, the reduction tells there is no attack of a certain type on $\mathcal{S}$ unless one finds a particular weakness in H;
3. an impossibility of attack, namely a means to show that there is no reduction $\mathsf{Break}(H) \geq \mathsf{Break}(\mathcal{S})$. This is usually done based on a meta-reduction [12]: if $\mathsf{Break}(H) \geq_{\mathcal{R}} \mathsf{Break}(\mathcal{S})$ then $\mathcal{R} \geq_{\mathcal{M}} P$ where P is an auxiliary computational problem (hence this technique requires an extra assumption);
4. an impossibility of security proof *i.e.* evidence that a polynomial reduction $\mathsf{Break}(H) \leq \mathsf{Break}(\mathcal{S})$ does not exist (using the meta-reduction technique as well).

We view 2. and 3. as *positive* security results and 1. and 4. as *negative* security results.

Our contributions. This paper only deals with signature schemes. The overall goal of our work is to report as clearly as possible the implications of the current crisis in the field of hash functions and identify the new threats signature schemes are facing in practice. Applying the approach described above, we provide a number of (reductionist) relations between the security profile[2] of a signature scheme $\mathcal{S} = \mathcal{S}[H]$ and the security profile of the hash function H (we also suggest convenient security notions for hash function families on our way). We find that these relations heavily depend on the way $\mathcal{S}$ makes use of H. We therefore classify signature schemes into different categories: deterministic hash-and-sign signatures and probabilistic hash-and-sign signatures. We also introduce the properties of primitiveness and injectivity which enlarge the scope of certain security relations. Our classification remains as general as possible while capturing the case of signature schemes of common practice such as RSA-PSS or ECDSA. The security relations we expose in this paper confirm that all signature schemes are not implicated on the same level by the recent attacks on hash functions.

We also determine how using iterated hash functions a la Merkle-Damgård impacts the security of deterministic (resp. probabilistic) hash-and-sign signatures. Our motivation here is to identify more specific results in the case of functions such as MD5 and SHA-1. We confirm that the security gain inherent to using the probabilistic hash-and-sign paradigm may be lost completely if instantiated with a Merkle-Damgård hash function and unwise operating mode. We finally give concrete attack workloads for attacking popular operating modes used in practical implementations of signature schemes.

Roadmap. The next section provides a number of preliminary facts on security reductions. We remind definitions for hash functions and adopt well-defined security notions in Section 3. In Section 4, we define the deterministic versus probabilistic hash-and-sign paradigms and the notions of primitive and injective schemes. We then review and classify a number of signature schemes based on these four basic concepts. Sections 5 and 6 present the security relations between a hash-and-sign signature scheme and its inner hash function in the deterministic and probabilistic cases respectively. Section 7 explores the particular case of Merkle-Damgård hash functions as more specific results can be stated in this context. We finally conclude with recommendations on the use of hash functions in signature schemes.

2 Preliminary Notions of Provable Security

2.1 Black-Box Reductions

We adopt the concrete-security setting [], as opposed to the asymptotic one. Given a computational problem P, a probabilistic algorithm is said to (τ, ε)-solve[3] P when it halts after at most τ elementary steps and outputs a

[1] More details on reductions are found in the next section.
[2] The hierarchy of all the security levels related to a scheme.
[3] We may say (τ, ε)-break P when P is a security related problem.

solution of P with success probability at least ε. The execution time τ relates to some fixed model of computation. $\mathsf{Succ}(P,\tau)$ denotes the maximal success probability taken over all probabilistic algorithms solving P in no more than τ elementary steps. A concrete black-box[4] reduction $\mathcal{R}$ between two computational problems P_1 and P_2 is a probabilistic algorithm $\mathcal{R}$ which (τ_1,ε_1)-solves P_1 given black-box access to an oracle (τ_2,ε_2)-solving P_2. We write $P_1 \leq_{\mathcal{R}} P_2$ when $\mathcal{R}$ is known to reduce P_1 to P_2 with polynomial reduction cost, meaning that $\tau_1 = \mathsf{poly}(\tau_2)$ and $\varepsilon_1 = \mathsf{poly}(\varepsilon_2)$ where the polynomials may depend on additional reduction parameters. Note that $\mathcal{R}$ can be polynomial even when no algorithm (τ_2,ε_2)-solving P_2 is known to exist. $P_1 \leq P_2$ states that a polynomial $\mathcal{R}$ exists such that $P_1 \leq_{\mathcal{R}} P_2$ and $P_1 \equiv P_2$ means $P_1 \leq P_2$ and $P_1 \geq P_2$. We write $P_1 \leq_{\mathcal{R}} P_2 \wedge P_3 \wedge \cdots \wedge P_n$ when $\mathcal{R}$ solves P_1 given oracle access to solvers for $P_2,\ldots,P_n$ and $\tau_1 = \mathsf{poly}(\tau_2,\ldots,\tau_n)$, $\varepsilon_1 = \mathsf{poly}(\varepsilon_2,\ldots,\varepsilon_n)$. Relation $\leq$ is a transitive ordering among computational problems. Given a function f with t inputs, $\mathsf{Time}(f,\ell_1,\ldots,\ell_t)$ stands for the minimum, taken over all algorithms $\mathcal{A}_f$ computing f, of the worst case running time of $\mathcal{A}_f$ for inputs of respective bitlength $\ell_1,\ldots,\ell_t$.

2.2 Constructive Security Reductions

Let P_1, P_2 be two computational problems. We defined earlier the predicate $P_1 \leq P_2$ as $\left(\exists\mathcal{R} : P_1 \leq_{\mathcal{R}} P_2\right)$ where we recall that $P_1 \leq_{\mathcal{R}} P_2$ implies that $\mathcal{R}$ is polynomial. A provable-security statement $P_1 \leq P_2$ (where P_1 and P_2 are now security-related problems) therefore indicates that if an efficient solver for P_2 exists then an efficient solver for P_1 exists as well. Such a statement is meaningful when there are reasons to believe that P_1 cannot be solved efficiently. When efficient solvers for P_1 are known to exist, however, the security statement vacuously holds. So is the statement $P_1 \leq P_2$ of any interest in this case?

Well, $P_1 \leq P_2$ is still meaningful if *a)* such an efficient solver for P_1 exists but is *unknown* and seems hard to construct, *b)* the proof of $P_1 \leq P_2$ is *constructive* meaning that an explicit black-box reduction $\mathcal{R}$ is given such that $P_1 \leq_{\mathcal{R}} P_2$. The statement can then be reinterpreted by saying that if one had access to an effective solver for P_2 then an effective solver for P_1 would be found. P_2 then remains unsolvable under the assumption that an effective solver for P_1 is hard to construct under current human knowledge. Note that this formulation is stronger than the previous, existential formulation. It implies for instance that $\mathcal{R}$ can be used as a *compiler* which, given the source code of a program solving P_2, can be used to build a program solving P_1. We refer to [33] for more detail. From now on, $P_1 \Leftarrow P_2$ states that we know a constructive proof for $P_1 \leq P_2$, meaning $\left(\exists\mathcal{R} : P_1 \leq_{\mathcal{R}} P_2\right)$ and such an $\mathcal{R}$ is explicitly given by its pseudo-code in the proof. Constructive polynomial equivalence is noted $P_1 \Leftrightarrow P_2$.

2.3 The Random Oracle Model

The random oracle (RO) model was implicitly introduced by Fiat and Shamir [18] and later formalized by Bellare and Rogaway [1]. It has been extensively used since. By opposition to using a fully detailed specification of a cryptosystem (its standard-model description), the RO model abstracts away the description of certain components such as hash functions or pseudo-random generators by making them external to the cryptosystem under the form of oracles. Generating or verifying a signature (and possibly generating keys as well) then requires oracle calls. It is striking to observe that (with a few exceptions) the intensive use of the RO model has rendered innocuous the use of hash functions in cryptographic designs. In the RO model, studying the security relations between a construction and its inner hash functions is completely vacuous. We therefore stick to the standard model in our investigations.

3 Hash Functions and Related Security Notions

We first give a number of definitions related to hash functions. We clarify that hash functions (as per our definitions) only handle bitstrings and must be composed with other specific formatting functions to form integer arrays, group elements, etc. The following formal notions of a hash function, a compression function and an iterated hash function are similar to those given by Black, Rogaway and Shrimpton in [5].

Definition 1 (Hash function). *A function H is a hash function if it maps $\{0,1\}^*$ (the set of finite bitstrings) to $\{0,1\}^{\mathsf{m}}$ (the set of m-bit strings) for some integer $\mathsf{m} > 0$ called the output size of H.*

[4] All reductions considered in this paper are concrete and fully black-box.

Definition 2 (Compression function). *A compression function is a function* $f : \{0,1\}^{\mathsf{m}} \times \{0,1\}^{\mathsf{b}} \to \{0,1\}^{\mathsf{m}}$ *where* m, b *are integers such that* $\mathsf{m} > 0$ *and* $\mathsf{b} > 0$.

Definition 3 (Iterated hash function). *Let* $f : \{0,1\}^{\mathsf{m}} \times \{0,1\}^{\mathsf{b}} \to \{0,1\}^{\mathsf{m}}$ *be a compression function. Given an arbitrary function* $g : \{0,1\}^* \to (\{0,1\}^{\mathsf{b}})^*$ *and* $IV_0 \in \{0,1\}^{\mathsf{m}}$*, the iterated hash function* H *constructed from* (f, g, IV_0) *is the hash function defined for* $M \in \{0,1\}^*$ *by* $H(M) = h_t$ *where* $X_1||...||X_t = g(M)$*,* $h_0 = IV_0$ *and* $h_i = f(h_{i-1}, X_i)$ *for* $i \in [1..t]$*. We call* b *the block size of* H.

3.1 Security Notions for Hash Functions

The definitions for (second) preimage-resistance and for collision-resistance were given by Brown in [9] and appear under various forms in [29]. These definitions are somewhat simplified here because we only deal with fixed hash functions rather than families of hash functions. Subsequently, the related security assumptions, most particularly the assumption about collision-resistance, are stronger than the usual assumptions for hash functions families.

Definition 4 (Preimage-resistance $\mathsf{PRE}^n[H]$). *Let* $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ *be a hash function. The preimage problem* $\mathsf{PRE}^n[H]$ *consists, given a random* $m \leftarrow \{0,1\}^{\mathsf{m}}$ *in finding an* n*-bit string* M *such that* $H(M) = m$*. Formally,*

$$\mathsf{Succ}\left(\mathsf{PRE}^n[H], \tau\right) = \max_{\mathcal{A}_H} \Pr\left[m \leftarrow \{0,1\}^{\mathsf{m}}; M \leftarrow \mathcal{A}_H(m) : \left(M \in \{0,1\}^n\right) \wedge \left(H(M) = h\right)\right]$$

where the probability is taken over the random choices of m *and* $\mathcal{A}_H$ *and the maximum is taken over all* τ*-time probabilistic algorithms* $\mathcal{A}_H$*.* H *is said to be* (τ, ε)*-preimage-resistant on* $\{0,1\}^n$ *if* $\mathsf{Succ}\left(\mathsf{PRE}^n[H], \tau\right) < \varepsilon$.

Definition 5 (Second-preimage-resistance $\mathsf{SEC}^{n_2}_{n_1}[H]$). *The second preimage problem* $\mathsf{SEC}^{n_2}_{n_1}[H]$ *consists, given a random* $M_1 \leftarrow \{0,1\}^{n_1}$ *in finding* $M_2 \in \{0,1\}^{n_2}$ *such that* $H(M_2) = H(M_1)$ *and* $M_2 \neq M_1$*. Let us pose*

$$\mathsf{Succ}\left(\mathsf{SEC}^{n_2}_{n_1}[H], \tau\right) = \max_{\mathcal{A}_H} \Pr\left[M_1 \leftarrow \{0,1\}^{n_1}; M_2 \leftarrow \mathcal{A}_H(M_1) : \left(M_2 \in \{0,1\}^{n_2} \setminus \{M_1\}\right) \wedge \left(H(M_2) = H(M_1)\right)\right]$$

where the probability is taken over the random choices of M_1 *and* $\mathcal{A}_H$ *and the maximum is taken over all* τ*-time adversaries* $\mathcal{A}_H$*.* H *is said to be* (τ, ε)*-second-preimage-resistant with respect to* (n_1, n_2) *if* $\mathsf{Succ}\left(\mathsf{SEC}^{n_2}_{n_1}[H], \tau\right) < \varepsilon$.

For clarity, all security problems parameterized by the bitsize of their inputs or outputs are written with input sizes as subscripts and outputs sizes as superscripts.

Definition 6 (Collision-resistance $\mathsf{COL}^{n_1,n_2}[H]$). *Solving the collision-finding problem* $\mathsf{COL}^{n_1,n_2}[H]$ *consists in finding* $M_1 \in \{0,1\}^{n_1}$ *and* $M_2 \in \{0,1\}^{n_2}$ *such that* $M_1 \neq M_2$ *and* $H(M_1) = H(M_2)$*. We then call* (M_1, M_2) *an* (n_1, n_2)*-collision. For any probabilistic algorithm* $\mathcal{A}_H$*, let*

$$\mathsf{Succ}^{\mathsf{COL}^{n_1,n_2}[H]}(\mathcal{A}_H) = \Pr\left[(M_1, M_2) \leftarrow \mathcal{A}_H() : \left(M_1 \in \{0,1\}^{n_1}\right) \wedge \left(M_2 \in \{0,1\}^{n_2} \setminus \{M_1\}\right) \wedge \left(H(M_2) = H(M_1)\right)\right]$$

where the probability is taken over the random choices of $\mathcal{A}_H$*. We say that* $\mathcal{A}_H$ (τ, ε)*-solves* $\mathsf{COL}^{n_1,n_2}[H]$ *if* $\mathcal{A}_H$ *runs in time at most* τ *and* $\mathsf{Succ}^{\mathsf{COL}^{n_1,n_2}[H]}(\mathcal{A}_H) \geq \varepsilon$.

It is well-known that defining (τ, ε)-collision-resistance for (unkeyed) hash functions using an existential formulation *i.e.* by saying that

$$\mathsf{Succ}\left(\mathsf{COL}^{n_1,n_2}[H], \tau\right) = \max_{\mathcal{A}_H} \mathsf{Succ}^{\mathsf{COL}^{n_1,n_2}[H]}(\mathcal{A}_H) < \varepsilon \tag{1}$$

is futile [33]. Indeed either H admits no (n_1, n_2)-collision in which case $\mathsf{Succ}^{\mathsf{COL}^{n_1,n_2}[H]}(\mathcal{A}_H) = 0$ for any $\mathcal{A}_H$ (and $\mathsf{COL}^{n_1,n_2}[H]$ is trivially unsolvable), or there always exists $\mathcal{A}_H$ which simply outputs a fixed "hardwired" (n_1, n_2)-collision (M_1, M_2). So in this case $\mathsf{Succ}\left(\mathsf{COL}^{n_1,n_2}[H], \tau\right)$ as defined above equals 1 as soon as τ exceeds the time needed to print these two strings. We therefore adopt the *human ignorance* (or *explicit reduction*) approach of [33] which amounts to state security reductions in terms of knowledge. We elaborate on this in Section 2.2.

H is said (τ, ε)-collision-resistant with respect to (n_1, n_2)-collisions if no explicit τ-time adversary $\mathcal{A}_H$ is known such that $\mathsf{Succ}^{\mathsf{COL}^{n_1,n_2}[H]}(\mathcal{A}_H) \geq \varepsilon$. Alternately, one can rely on (1) by restricting the maximum to adversaries *known to man*. Note that by the birthday paradox, we know a collision-finding algorithm $\mathcal{A}_H$ with success probability $\varepsilon = 1/2$ and running time $\tau \simeq 2^{\mathsf{m}/2}$. So no hash function $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ is $(2^{\mathsf{m}/2}, 2^{-1})$-collision-resistant for general n_1, n_2's.

Lemma 7. *Let $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ be a hash function. Then for any $n_1, n_2 > 0$,*

$$\mathsf{COL}^{n_1,n_2}[H] \Leftarrow \mathsf{SEC}^{n_2}_{n_1}[H], \mathsf{SEC}^{n_1}_{n_2}[H] \quad \text{and} \quad \mathsf{SEC}^{n_2}_{n_1}[H] \Leftarrow \mathsf{PRE}^{n_2}[H] \ .$$

Proof. $\left(\mathsf{COL}^{n_1,n_2}[H] \Leftarrow \mathsf{SEC}^{n_2}_{n_1}\right)$. Assuming a (τ_2, ε_2)-solver $\mathcal{A}_2$ for $\mathsf{SEC}^{n_2}_{n_1}[H]$, one builds a reduction algorithm $\mathcal{A}_1$ which (τ_1, ε_1)-solves $\mathsf{COL}^{n_1,n_2}[H]$ with $\varepsilon_1 = \varepsilon_2$ and $\tau_1 = \tau_2 + \mathsf{Time}(\mathtt{rand}, n_1)$ as follows. On no input, $\mathcal{A}_1$ uniformly selects $M_1 \leftarrow \{0,1\}^{n_1}$, runs $M_2 \leftarrow \mathcal{A}_2(M_1)$ and returns (M_1, M_2). A proof that $\mathsf{SEC}^{n_2}_{n_1}[H] \Leftarrow \mathsf{PRE}^{n_2}[H]$ is given for instance in [34]. To yield an efficient reduction, however, it is required that $n_1, n_2 \gg \mathsf{m}$. □

3.2 Hash Function Families and Related Security Notions

We recall the definition of a hash function family.

Definition 8 (Hash function family). *A hash function family F is a function $F : \{0,1\}^* \times \{0,1\}^{\mathsf{r}} \to \{0,1\}^{\mathsf{m}}$ for integers $\mathsf{m}, \mathsf{r} > 0$ such that for any $r \in \{0,1\}^{\mathsf{r}}$, $F(\cdot, r)$ is a hash function of output size m.*

The second argument r in $F(M, r)$ is called the *index* (or *key*) of F. We define a collection of security notions with respect to a hash function family F with index space $\{0,1\}^{\mathsf{r}}$. As for unkeyed hash functions, one essentially faces notions of preimage, second preimage and collision resistance. These security notions are defined in the concrete setting using (τ, ε)-adversaries.

Definition 9 (Existential Preimage Resistance $\mathsf{E\text{-}PRE}^n[F]$). *Breaking $\mathsf{E\text{-}PRE}^n[F]$ consists, given a random $m \leftarrow \{0,1\}^{\mathsf{m}}$ in finding a pair $(M, r) \in \{0,1\}^n \times \{0,1\}^{\mathsf{r}}$ such that $F(M, r) = m$. Formally,*

$$\mathsf{Succ}\left(\mathsf{E\text{-}PRE}^n[F], \tau\right) = \max_{\mathcal{A}_F} \Pr\left[m \leftarrow \{0,1\}^{\mathsf{m}}, (M, r) \leftarrow \mathcal{A}_F(m) : \left(|M| = n\right) \wedge \left(|r| = \mathsf{r}\right) \wedge \left(F(M, r) = m\right)\right]$$

where the probability is taken over the random choices of m and $\mathcal{A}_F$ and the maximum is taken over all τ-time probabilistic algorithms $\mathcal{A}_F$. F is said to be (τ, ε)-existentially preimage-resistant on $\{0,1\}^n$ if $\mathsf{Succ}\left(\mathsf{E\text{-}PRE}^n[F], \tau\right) < \varepsilon$.

Definition 10 (Universal Preimage Resistance $\mathsf{U\text{-}PRE}^n[F]$). *Breaking $\mathsf{U\text{-}PRE}^n[F]$ consists, given random $m \leftarrow \{0,1\}^{\mathsf{m}}$ and $r \leftarrow \{0,1\}^{\mathsf{r}}$, in finding an n-bit string M such that $F(M, r) = m$. We define*

$$\mathsf{Succ}\left(\mathsf{U\text{-}PRE}^n[F], \tau\right) = \max_{\mathcal{A}_F} \Pr\left[m \leftarrow \{0,1\}^{\mathsf{m}}, r \leftarrow \{0,1\}^{\mathsf{r}}, M \leftarrow \mathcal{A}_F(m, r) : \left(|M| = n\right) \wedge \left(F(M, r) = m\right)\right] \ ,$$

the probability being taken over the random choices of m, r and $\mathcal{A}_F$ and the maximum over τ-time probabilistic algorithms $\mathcal{A}_F$. F is said to be (τ, ε)-universally preimage-resistant on $\{0,1\}^n$ if $\mathsf{Succ}\left(\mathsf{U\text{-}PRE}^n[F], \tau\right) < \varepsilon$.

Definition 11 (Existential Second Preimage Resistance $\mathsf{E\text{-}SEC}^{n_2}_{n_1}[F]$). *Breaking $\mathsf{E\text{-}SEC}^{n_2}_{n_1}[F]$ consists, given a random $M_1 \leftarrow \{0,1\}^{n_1}$ in finding $(M_2, r) \in \{0,1\}^{n_2} \times \{0,1\}^{\mathsf{r}}$ such that $F(M_2, r) = F(M_1, r)$ and $M_2 \neq M_1$. Let*

$$\mathsf{Succ}\left(\mathsf{E\text{-}SEC}^{n_2}_{n_1}[F], \tau\right) = \max_{\mathcal{A}_F} \Pr\left[\begin{array}{c} M_1 \leftarrow \{0,1\}^{n_1} \\ (M_2, r) \leftarrow \mathcal{A}_F(M_1) \end{array} : \begin{array}{c} \left(|M_2| = n_2\right) \wedge \left(|r| = \mathsf{r}\right) \wedge \left(M_2 \neq M_1\right) \\ \wedge \left(F(M_2, r) = F(M_1, r)\right) \end{array}\right] \ ,$$

the probability being taken over the random choices of M_1 and $\mathcal{A}_F$ and the maximum over all τ-time $\mathcal{A}_F$'s. F is said to be (τ, ε)-existentially-second-preimage-resistant with respect to (n_1, n_2) if $\mathsf{Succ}\left(\mathsf{E\text{-}SEC}^{n_2}_{n_1}[F], \tau\right) < \varepsilon$.

Definition 12 (Universal Second Preimage Resistance $\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F]$). *Given a random pair $(M_1, r) \leftarrow \{0,1\}^{n_1} \times \{0,1\}^{\mathsf{r}}$, find $M_2 \in \{0,1\}^{n_2} \setminus \{M_1\}$ such that $F(M_2, r) = F(M_1, r)$ and $M_2 \neq M_1$:*

$$\mathsf{Succ}\left(\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F], \tau\right) = \max_{\mathcal{A}_F} \Pr\left[\begin{array}{c} M_1 \leftarrow \{0,1\}^{n_1} \\ r \leftarrow \{0,1\}^{\mathsf{r}} \\ M_2 \leftarrow \mathcal{A}_F(M_1, r) \end{array} : \begin{array}{c} \left(|M_2| = n_2\right) \wedge \left(M_2 \neq M_1\right) \\ \wedge \left(F(M_2, r) = F(M_1, r)\right) \end{array}\right] \ ,$$

the probability being taken over the random choices of (M_1, r) and $\mathcal{A}_F$ and the maximum over all τ-time $\mathcal{A}_F$'s. F is said to be (τ, ε)-universally-second-preimage-resistant with respect to (n_1, n_2) if $\mathsf{Succ}\left(\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F], \tau\right) < \varepsilon$.

Definition 13 (Absolute Second Preimage Resistance $\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F]$). *Given a random $M_1 \leftarrow \{0,1\}^{n_1}$, find $M_2 \in \{0,1\}^{n_2}$ such that $F(M_2,r) = F(M_1,r)$ for <u>any</u> $r \in \{0,1\}^{\mathsf{r}}$:*

$$\mathsf{Succ}\left(\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F],\tau\right) = \max_{\mathcal{A}_F} \Pr\left[\begin{array}{c} M_1 \leftarrow \{0,1\}^{n_1} \\ M_2 \leftarrow \mathcal{A}_F(M_1) \end{array} : \begin{array}{c} \left(|M_2| = n_2\right) \wedge \left(M_2 \neq M_1\right) \\ \wedge \left(F(M_2,\cdot) \triangleq F(M_1,\cdot)\right) \end{array}\right],$$

the probability being taken over the random choices of M_1 and $\mathcal{A}_F$ and the maximum over all τ-time $\mathcal{A}_F$'s. F is said to be (τ,ε)-absolutely-second-preimage-resistant with respect to (n_1,n_2) if $\mathsf{Succ}\left(\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F],\tau\right) < \varepsilon$.

Existential, universal and absolute collision-resistance are defined along the same lines:

Definition 14 (Existential Collision-resistance $\mathsf{E\text{-}COL}^{n_1,n_2}[F]$). *Solving $\mathsf{E\text{-}COL}^{n_1,n_2}[F]$ consists in finding $M_1 \in \{0,1\}^{n_1}$, $M_2 \in \{0,1\}^{n_2}$ and $r \in \{0,1\}^{\mathsf{r}}$ such that $M_1 \neq M_2$ and $F(M_2,r) = F(M_1,r)$. We then call (M_1,M_2,r) an (n_1,n_2)-collision on F. For any probabilistic algorithm $\mathcal{A}_F$, let*

$$\mathsf{Succ}\left(\mathsf{E\text{-}COL}^{n_1,n_2}[F],\tau\right) = \max_{\mathcal{A}_F} \Pr\left[(M_1,M_2,r) \leftarrow \mathcal{A}_F() : \begin{array}{c} \left(|M_1| = n_1\right) \wedge \left(|M_2| = n_2\right) \wedge \left(M_2 \neq M_1\right) \\ \wedge \left(|r| = \mathsf{r}\right) \wedge \left(F(M_2,r) = F(M_1,r)\right) \end{array}\right],$$

where the probability is taken over the random choices of $\mathcal{A}_F$ and the maximum over all <u>known</u> τ-time $\mathcal{A}_F$'s. F is said to be (τ,ε)-existentially-collision-resistant with respect to (n_1,n_2) if $\mathsf{Succ}\left(\mathsf{E\text{-}COL}^{n_1,n_2}[F],\tau\right) < \varepsilon$.

Definition 15 (Universal Collision-resistance $\mathsf{U\text{-}COL}^{n_1,n_2}[F]$). *Breaking $\mathsf{U\text{-}COL}^{n_1,n_2}[F]$ consists, given a random $r \leftarrow \{0,1\}^{\mathsf{r}}$, in finding $M_1 \in \{0,1\}^{n_1}$ and $M_2 \in \{0,1\}^{n_2}$ such that $M_1 \neq M_2$ and $F(M_2,r) = F(M_1,r)$. For any probabilistic algorithm $\mathcal{A}_F$, let*

$$\mathsf{Succ}\left(\mathsf{U\text{-}COL}^{n_1,n_2}[F],\tau\right) = \max_{\mathcal{A}_F} \Pr\left[\begin{array}{c} r \leftarrow \{0,1\}^{\mathsf{r}} \\ (M_1,M_2) \leftarrow \mathcal{A}_F(r) \end{array} : \begin{array}{c} \left(|M_1| = n_1\right) \wedge \left(|M_2| = n_2\right) \wedge \left(M_2 \neq M_1\right) \\ \wedge \left(F(M_2,r) = F(M_1,r)\right) \end{array}\right],$$

where the probability is taken over the random choices of $\mathcal{A}_F$ and the maximum over all τ-time $\mathcal{A}_F$'s. F is said to be (τ,ε)-universally-collision-resistant with respect to (n_1,n_2) if $\mathsf{Succ}\left(\mathsf{U\text{-}COL}^{n_1,n_2}[F],\tau\right) < \varepsilon$.

Definition 16 (Absolute Collision-resistance $\mathsf{A\text{-}COL}^{n_1,n_2}[F]$). *Solving $\mathsf{A\text{-}COL}^{n_1,n_2}[F]$ consists in finding $M_1 \in \{0,1\}^{n_1}$ and $M_2 \in \{0,1\}^{n_2}$ such that $M_1 \neq M_2$ and $F(M_2,r) = F(M_1,r)$ for any $r \in \{0,1\}^{\mathsf{r}}$. We then call (M_1,M_2) an (n_1,n_2)-absolute collision on F. For any probabilistic algorithm $\mathcal{A}_F$, let us define*

$$\mathsf{Succ}\left(\mathsf{A\text{-}COL}^{n_1,n_2}[F],\tau\right) = \max_{\mathcal{A}_F} \Pr\left[(M_1,M_2) \leftarrow \mathcal{A}_F() : \begin{array}{c} \left(|M_1| = n_1\right) \wedge \left(|M_2| = n_2\right) \wedge \left(M_2 \neq M_1\right) \\ \wedge \left(F(M_2,\cdot) \triangleq F(M_1,\cdot)\right) \end{array}\right],$$

where the probability is taken over the random choices of $\mathcal{A}_F$ and the maximum over all <u>known</u> τ-time $\mathcal{A}_F$'s. F is said to be (τ,ε)-absolutely-collision-resistant with respect to (n_1,n_2) if $\mathsf{Succ}\left(\mathsf{A\text{-}COL}^{n_1,n_2}[F],\tau\right) < \varepsilon$.

Theorem 17. *These security notions are connected via constructive black-box reductions as show on Fig. 1.*

$$\begin{array}{ccccc}
\mathsf{E\text{-}PRE}^{n_2}[F] & \Leftarrow & \mathsf{U\text{-}PRE}^{n_2}[F] & & \\
\Downarrow & & \Downarrow & & \\
\mathsf{E\text{-}SEC}^{n_2}_{n_1}[F] & \Leftarrow & \mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] & \Leftarrow & \mathsf{A\text{-}SEC}^{n_2}_{n_1}[F] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\mathsf{E\text{-}COL}^{n_1,n_2}[F] & \Leftarrow & \mathsf{U\text{-}COL}^{n_1,n_2}[F] & \Leftarrow & \mathsf{A\text{-}COL}^{n_1,n_2}[F]
\end{array}$$

Fig. 1. Relations among security notions for hash function families

We finally note that U-PRE, U-SEC and U-COL are identical to the notions Pre, Sec and Coll of [34].

4 Hash-and-Sign Signatures and Related Security Notions

4.1 Definitions

This section gives a number of definitions for signature schemes. In particular, we define the notion of *probabilistic hash-and-sign* signatures which can be seen as a generalization of well-known hash-and-sign signatures (we will adopt the term of *deterministic* hash-and-sign signatures to designate those). It is crucial to note here that the notions of probabilistic (resp. deterministic) hash-and-sign signatures have nothing to do with the notions of probabilistic (resp. deterministic) signatures. The signature schemes we consider are most of the time probabilistic unless otherwise stated. What we care about is in fact the probabilistic or deterministic nature of their hash-and-sign mechanism. To avoid any confusion, we make use of distinct notations $H(M)$ or $F(M,r)$ when referring to the hash function involved in a hash-and-sign signature scheme.

Definition 18 (Signature scheme). *A signature scheme $\mathcal{S}$ with message space $\mathcal{M} \subseteq \{0,1\}^*$ is identified to a tuple $\mathcal{S} \triangleq (\mathcal{S}.\mathsf{Gen}, \mathcal{S}.\mathsf{Sign}, \mathcal{S}.\mathsf{Ver})$ of algorithms. $\mathcal{S}.\mathsf{Gen}()$ is a probabilistic algorithm which takes no input and outputs a pair of strings $(\mathsf{pk}, \mathsf{sk})$. A signature on message $M \in \mathcal{M}$ is an s-bit string $\sigma = \mathcal{S}.\mathsf{Sign}(\mathsf{sk}, M, u)$ where $u \leftarrow \{0,1\}^{\mathsf{u}}$. $\mathcal{S}.\mathsf{Ver}(\mathsf{pk}, M, \sigma)$ outputs 1 if $\sigma = \mathcal{S}.\mathsf{Sign}(\mathsf{sk}, M, u)$ for some $u \in \{0,1\}^{\mathsf{u}}$, 0 otherwise. The message space is either $\mathcal{M} = \{0,1\}^{\mathsf{m}}$ for some integer $\mathsf{m} > 0$ or $\mathcal{M} = \{0,1\}^*$ when messages can be of arbitrary length.*

Definition 19 (Deterministic hash-and-sign signatures). *A deterministic hash-and-sign signature scheme is a pair $\mathcal{S} = \langle H, \Sigma \rangle$ where $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ is a hash function and Σ is a fixed-length signature scheme signing m-bit messages under u bits of randomness. $\mathcal{S}.\mathsf{Gen} \triangleq \Sigma.\mathsf{Gen}$ and a signature on $M \in \{0,1\}^*$ with respect to $\mathcal{S}$ is computed as $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, m, u)$ where $m = H(M)$ and $u \leftarrow \{0,1\}^{\mathsf{u}}$. $\mathcal{S}.\mathsf{Ver}(\mathsf{pk}, M, \sigma)$ outputs $\Sigma.\mathsf{Ver}(\mathsf{pk}, H(M), \sigma)$. Therefore $\mathcal{S}$ supports arbitrary-length messages.*

We view the deterministic hash-and-sign paradigm as a scheme compiler which, taking a hash function H and a fixed-length, message-space-compatible signature scheme Σ as inputs, builds the domain-extended scheme $\mathcal{S} = \langle H, \Sigma \rangle$. To properly capture probabilistic hash-and-sign signatures in the same fashion, we first need to take a closer look at the inner computations of a signature scheme.

Definition 20 (Two-step signatures). *A signature scheme $\mathcal{S}$ with message space $\mathcal{M} \subseteq \{0,1\}^*$ is said to be two-step if*

i) there exist two deterministic algorithms S_1, S_2 such that for any pair $(M,u) \in \mathcal{M} \times \{0,1\}^{\mathsf{u}}$, $\mathcal{S}.\mathsf{Sign}(\mathsf{sk}, M, u) = S_2(\mathsf{sk}, M, r, \mathsf{aux})$ where $(r, \mathsf{aux}) = S_1(\mathsf{sk}, u)$ and $r \in \{0,1\}^{\mathsf{r}}$. We impose that r is uniform over $\{0,1\}^{\mathsf{r}}$ if u is uniform over $\{0,1\}^{\mathsf{u}}$;

ii) there exist two deterministic algorithms V_1, V_2 such that $\mathcal{S}.\mathsf{Ver}(\mathsf{pk}, M, \sigma) = V_2(\mathsf{pk}, M, \sigma, \hat{r})$ where $\hat{r} = V_1(\mathsf{pk}, \sigma)$;

iii) if there exists $u \in \{0,1\}^{\mathsf{u}}$ such that $\sigma = \mathcal{S}.\mathsf{Sign}(\mathsf{sk}, M, u)$ then the strings $r, \hat{r}$ such that $(r, \mathsf{aux}) = S_1(\mathsf{sk}, u)$ for some aux and $\hat{r} = V_1(\mathsf{pk}, \sigma)$ are equal. We define the type of $\mathcal{S}$ as $(\mathsf{m}, \mathsf{u}, \mathsf{r}, \mathsf{s})$ if $\mathcal{M} = \{0,1\}^{\mathsf{m}}$ and $(*, \mathsf{u}, \mathsf{r}, \mathsf{s})$ if $\mathcal{M} = \{0,1\}^*$.

The property of being two-step seems totally unrestrictive as we do not know any example of a *non*-two-step signature scheme. Definition 20 merely serves at introducing notations.

Definition 21 (Probabilistic hash-and-sign signatures). *A probabilistic hash-and-sign signature scheme is a pair $\mathcal{S} = \langle F, \Sigma \rangle$ where $F : \{0,1\}^* \times \{0,1\}^{\mathsf{r}} \to \{0,1\}^{\mathsf{m}}$ is a hash function family with index space $\{0,1\}^{\mathsf{r}}$ and Σ is a two-step signature scheme of type $(\mathsf{m}, \mathsf{u}, \mathsf{r}, \mathsf{s})$ for integers $\mathsf{u}, \mathsf{s} > 0$. The key generation of $\mathcal{S}$ is identical to the one of Σ. Let $\Sigma_1, \Sigma_2, \Upsilon_1, \Upsilon_2$ be the inner functions of Σ. A signature on $M \in \{0,1\}^*$ is computed as $\sigma = \Sigma_2(\mathsf{sk}, m, r, \mathsf{aux})$ where $m = F(M,r)$, $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$ and $u \leftarrow \{0,1\}^{\mathsf{u}}$. $\mathcal{S}.\mathsf{Ver}(\mathsf{pk}, M, \sigma)$ first computes $\hat{r} = \Upsilon_1(\mathsf{pk}, \sigma)$, then $\hat{m} = F(M, \hat{r})$ and outputs $\Upsilon_2(\mathsf{pk}, \hat{m}, \sigma, \hat{r})$.*

It is easily checked that $\mathcal{S} = \langle F, \Sigma \rangle$ is a two-step signature scheme of type $(*, \mathsf{u}, \mathsf{r}, \mathsf{s})$ with inner functions $S_1 \triangleq \Sigma_1$, $S_2(\mathsf{sk}, M, r, \mathsf{aux}) = \Sigma_2(\mathsf{sk}, F(M,r), r, \mathsf{aux})$, $V_1 \triangleq \Upsilon_1$ and $V_2(\mathsf{pk}, M, \sigma, \hat{r}) = \Upsilon_2(\mathsf{pk}, F(M, \hat{r}), \sigma, \hat{r})$. Also, deterministic hash-and-sign signatures can be seen as a particular case of probabilistic hash-and-sign signatures where the index space of F is empty *i.e.* when $F(M,r) = H(M)$. Definition 21 is then easily reformulated in the terms of Definition 19 by posing $\mathsf{aux} \triangleq u$, $\mathsf{r} = 0$ and $r = \emptyset$ (thus letting $\Sigma_1(\mathsf{sk}, u) = (\emptyset, u)$) and setting $\Sigma_2 \triangleq \Sigma.\mathsf{Sign}$, $\Upsilon_1 \triangleq \emptyset$ and $\Upsilon_2 \triangleq \Sigma.\mathsf{Ver}$. We now define two additional properties that will be essential in further sections.

Definition 22 (Primitiveness). *Let $\mathcal{S} = \langle F, \Sigma \rangle$ be a probabilistic hash-and-sign signature scheme and let Σ_1, Σ_2, Υ_1, Υ_2 be the inner functions of Σ. $\mathcal{S}$ is said to be τ-primitive when we know a τ-time probabilistic algorithm $\mathcal{S}.\mathsf{Prim}$ which, for all pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$, takes pk as input and outputs a random pair $(m, \sigma = \mathcal{S}.\mathsf{Sign}(\mathsf{sk}, m, u))$ such that a) m is uniformly distributed over $\{0,1\}^{\mathsf{m}}$; b) u is uniformly distributed over $\{0,1\}^{\mathsf{r}}$.*

Note that $\mathcal{S}$ being primitive implies that Σ is existentially forgeable (EF)under a key-only attack (KOA). In essence, if $\mathcal{S} = \langle F, \Sigma \rangle$ is primitive then replacing F by the identity function *structurally* destroys the $\mathsf{EF\text{-}KOA}$ security of the modified scheme. This notion captures the intuition that F somehow plays an important role in the $\mathsf{EF\text{-}KOA}$ security of $\mathcal{S}$.

Definition 23 (Injectivity). *Let $\mathcal{S} = \langle F, \Sigma \rangle$ be a probabilistic hash-and-sign signature scheme and Σ_1, Σ_2, Υ_1, Υ_2 as above. $\mathcal{S}$ is said to be injective when for any key pair $(\mathsf{pk}, \mathsf{sk})$ and any $\sigma \in \{0,1\}^{\mathsf{s}}$, there exists at most one pair $(m, r) \in \{0,1\}^{\mathsf{m}} \times \{0,1\}^{\mathsf{r}}$ such that $\sigma = \Sigma_2(\mathsf{sk}, m, r, \mathsf{aux})$ and $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$ for some u, aux.*

It appears that most signature schemes used in practice, beside being probabilistic hash-and-sign schemes, are also primitive and injective. This is at least the case with Schnorr, FDH and PSS signatures. Other practical schemes such as DSA or GHR are also of interest but do not seem to be primitive. This observation motivates us to categorize signatures schemes according to the four definitions above and study these separately.

4.2 Classifying Common Signature Schemes

Signature schemes obtained by applying the Fiat-Shamir transform to 3-move identification protocols are a typical example of probabilistic hash-and-sign signatures that are both primitive and injective. Some variants of Fiat-Shamir-converted schemes such as KCDSA also fall in this category. Finally, all classical hash-then-invert signature schemes based on a trapdoor one-way permutation such as PSS are complete in the sense of Definitions 22 and 23 as well. Other schemes are not. To illustrate this variety, we review and characterize the signature schemes most commonly found in good cryptographic practice. We report our classification results on Table 1.

Signature Scheme	*Deterministic Hash-and-Sign*	*Probabilistic Hash-and-Sign*	*Primitive*	*Injective*
Schnorr		×	×	×
FDH	×		×	×
PFDH		×	×	×
PSS		×	×	×
EMSA-PSS	×		×	×
BLS	×		×	×
Generic DSA	×			×
GHR	×			×
CS	×			

Table 1. A classification of common signature schemes

We refer to Appendix A for a description of these signature schemes and evidence of their classification.

4.3 Security Notions for Signatures

We recall the security notions related to signature schemes that will be of interest in this paper. Security notions combine an adversarial goal with an attack model. The attacker is seen as a probabilistic polynomial time algorithm attempting to fulfill its goal while being given a number of computational resources. The attacker may interact with the scheme in different ways.

Adversarial goals. A signature scheme is said to be *universally forgeable* (UF) when there exists an adversary that returns a valid signature on a randomly chosen message $M \leftarrow \mathcal{M}$ given as input. When $\mathcal{M} = \{0,1\}^*$, M is uniformly selected from $\{0,1\}^n$ for finite $n > 0$, thus defining a collection of goals $\{\mathsf{UF}_n\}_{n>0}$. The notion of *existential forgery* EF (resp. $\{\mathsf{EF}^n\}_{n>0}$) is similar to UF but allows the adversary to choose freely the value of the signed message.

We also consider *non-repudiation*, a scenario where the attacker is the signer itself and therefore knows the secret key sk (See [27] for a general consideration on the subject). A *universal repudiation* (UR) occurs when, given a random message $M_1 \leftarrow \mathcal{M}$, the adversary outputs a different message $M_2 \in \mathcal{M} \setminus \{M_1\}$ and a signature σ such that σ is a valid signature on both M_1 and M_2. When $\mathcal{M} = \{0,1\}^*$, we draw M_1 from $\{0,1\}^{n_1}$ uniformly at random, impose $M_2 \in \{0,1\}^{n_2}$ and define a collection of goals $\{\mathsf{UR}^{n_2}_{n_1}\}_{n_1,n_2>0}$. An *existential repudiation* ER (respectively $\{\mathsf{ER}^{n_1,n_2}\}_{n_1,n_2>0}$) is a tuple (M_1, M_2, σ) where the adversary chooses $M_1, M_2 \in \mathcal{M}$ (respectively $M_1 \in \{0,1\}^{n_1}, M_2 \in \{0,1\}^{n_2}$) and σ freely, with the obvious restriction $M_1 \neq M_2$.

Attack models. We consider several attack scenarios. In a *key only attack* (KOA), the adversary is given nothing else than a public key as input. A *known message attack* (KMA) consists in giving as input to the attacker a list $(M_1, \sigma_1), \ldots, (M_\ell, \sigma_\ell)$ of random and pairwise distinct message-signature pairs. When $\mathcal{M} = \{0,1\}^*$, we draw M_i from $\{0,1\}^n$ for $i \in [1..\ell]$, thereby defining a collection of attacks $\{\mathsf{KMA}_n\}_{n>0}$. In a *chosen message attack* (CMA), the adversary is given adaptive access to a signing oracle.

Relations among security levels. We view security notions as computational problems. For instance UF-KMA $[\mathcal{S}]$ is the problem of computing a universal forgery under known message attack. This notation allows to relate security notions using reductions. In the case of KMA or CMA, we denote by ℓ-GOAL-ATK $[\mathcal{S}]$ the problem of breaking GOAL in no more than ℓ calls to the resource defined by ATK. Thus, breaking ℓ-EF-CMA $[\mathcal{S}]$ authorizes at most ℓ calls to the signing oracle to break EF. Fig. 2 displays the well-known black-box (and constructive) reductions among security levels. Indices n_1, n_2 reflecting the case $\mathcal{M} = \{0,1\}^*$ are to be removed if $\mathcal{M} = \{0,1\}^{\mathsf{m}}$.

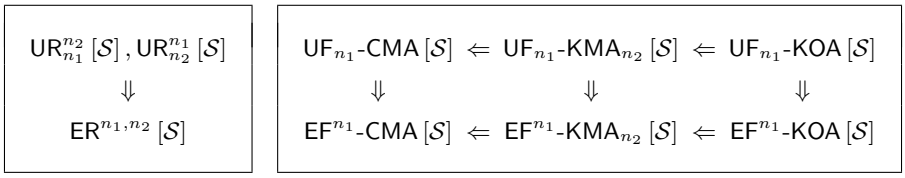

Fig. 2. Relations among security notions for signature schemes

5 Security Relations for Deterministic Hash-and-Sign Signatures

Our goal is to exhaust the security reductions standing between a deterministic hash-and-sign signature scheme $\mathcal{S} = \langle H, \Sigma \rangle$ and its inner hash function H. The signature scheme Σ plays the role of a parameter here. These reductions may have three different flavors. First, we show how breaking certain security properties of H allows to break the signature scheme (attacks). Then we show an equivalence between performing certain attacks on $\mathcal{S}$ and finding security failures in the underlying hash function H (security proofs). We finally show that certain security notions for $\mathcal{S}$ and H are computationally independent, meaning that they cannot be compared using black-box reductions. This independence shows the non-existence of certain attacks and security proofs. We start by listing attacks.

5.1 Attacking $\mathcal{S} = \langle H, \Sigma \rangle$ by Attacking H

Lemma 24. *Let $\mathcal{S} = \langle H, \Sigma \rangle$ be a deterministic hash-and-sign signature scheme as per Definition 19. Then for any integers $n_1, n_2 > 0$,*

$$\mathsf{COL}^{n_1,n_2}[H] \Rightarrow \text{1-}\mathsf{EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}], \text{1-}\mathsf{EF}^{n_2}\text{-}\mathsf{CMA}[\mathcal{S}] \tag{2}$$

$$\mathsf{COL}^{n_1,n_2}[H] \Rightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \tag{3}$$

$$\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \text{1-}\mathsf{UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}] \tag{4}$$

$$\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \text{1-}\mathsf{EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}] \tag{5}$$

$$\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}] \tag{6}$$

Proof ($\mathsf{COL}^{n_1,n_2}[H] \Rightarrow \text{1-}\mathsf{EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}]$*).* Given a (τ_H, ε_H)-solver $\mathcal{A}_H$ for $\mathsf{COL}^{n_1,n_2}[H]$ we build an EF^{n_1}-CMA attacker $\mathcal{A}_{\mathcal{S}}$ breaking $\mathcal{S}$ with probability $\varepsilon_{\mathcal{S}} = \varepsilon_H$ and $\tau_{\mathcal{S}} = \tau_H + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_2)$ in exactly one call to the signing oracle. Given a random public key $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$, our attacker $\mathcal{A}_{\mathcal{S}}$ runs $\mathcal{A}_H$ to produce an (n_1, n_2)-collision (M_1, M_2) such that $M_1 \neq M_2$ and $H(M_1) = H(M_2)$. Then $\mathcal{A}_{\mathcal{S}}$ requests a signature $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, H(M_2), u)$ on M_2 and produces σ as a valid signature on $M_1 \neq M_2$. The same works with 1-EF^{n_2}-CMA as well by symmetry. □

Proof ($\mathsf{COL}^{n_1,n_2}[H] \Rightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}]$*).* Given a (τ_H, ε_H)-collision-finder $\mathcal{A}_H$ we build a repudiator $\mathcal{A}_{\mathcal{S}}$ breaking $\mathsf{ER}^{n_1,n_2}[\mathcal{S}]$ with probability $\varepsilon_{\mathcal{S}} = \varepsilon_H$ and time $\tau_{\mathcal{S}} = \tau_H + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_1)$. Given a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$, $\mathcal{A}_{\mathcal{S}}$ runs $\mathcal{A}_H$ to build an (n_1, n_2)-collision. $\mathcal{A}_{\mathcal{S}}$ then signs M_1 under randomness $u \in \{0,1\}^{\mathsf{u}}$ to get $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, H(M_1), u)$ and outputs the tuple (M_1, M_2, σ). □

Proof ($\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \text{1-}\mathsf{UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}]$*).* Assume $\mathcal{A}_H$ (τ_H, ε_H)-solves $\mathsf{SEC}^{n_2}_{n_1}[H]$. We build a UF_{n_1}-CMA attacker $\mathcal{A}_{\mathcal{S}}$ which $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-breaks $\mathcal{S}$ in one call to the signing oracle with $\varepsilon_{\mathcal{S}} = \varepsilon_H$ and $\tau_{\mathcal{S}} = \tau_H + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_2)$. Given a public key $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$ and $M_1 \leftarrow \{0,1\}^{n_1}$, $\mathcal{A}_{\mathcal{S}}$ must produce a valid signature on M_1. $\mathcal{A}_{\mathcal{S}}$ runs $\mathcal{A}_H(M_1)$ to build a second preimage $M_2 \neq M_1$ with $|M_2| = n_2$ and $H(M_2) = H(M_1)$. $\mathcal{A}_{\mathcal{S}}$ then requests a signature on M_2 and is given $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, H(M_2), u)$ for some $u \in \{0,1\}^{\mathsf{u}}$. $\mathcal{A}_{\mathcal{S}}$ then returns (M_1, σ). □

Proof ($\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \text{1-}\mathsf{EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}]$*).* Assuming $\mathcal{A}_H$ (τ_H, ε_H)-solves $\mathsf{SEC}^{n_2}_{n_1}[H]$, we build an attacker $\mathcal{A}_{\mathcal{S}}$ which $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-solves 1-$\mathsf{EF}^{n_2}$-$\mathsf{KMA}_{n_1}[\mathcal{S}]$ with $\varepsilon_{\mathcal{S}} = \varepsilon_H$ and $\tau_{\mathcal{S}} = \tau_H$. $\mathcal{A}_{\mathcal{S}}$ is given a random $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$ and a single random message-signature pair (M_1, σ) where $M_1 \leftarrow \{0,1\}^{n_1}$ and $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, H(M_1), u)$ for $u \leftarrow \{0,1\}^{\mathsf{u}}$. $\mathcal{A}_{\mathcal{S}}$ runs $\mathcal{A}_H(M_1)$ to construct a second preimage $M_2 \neq M_1$ with $|M_2| = n_2$ and $H(M_2) = H(M_1)$. $\mathcal{A}_{\mathcal{S}}$ then returns the existential forgery (M_2, σ). □

Proof ($\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}]$*).* Assuming $\mathcal{A}_H$ (τ_H, ε_H)-solves $\mathsf{SEC}^{n_2}_{n_1}[H]$, we build a $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-repudiator $\mathcal{A}_{\mathcal{S}}$ which on input a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$ and a random message $M_1 \in \{0,1\}^{n_1}$, returns a message $M_2 \neq M_1$, $|M_2| = n_2$ and σ such that σ is a valid signature on both M_1 and M_2. Here again, $\varepsilon_{\mathcal{S}} = \varepsilon_H$ and $\tau_{\mathcal{S}} = \tau_H + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_2)$. $\mathcal{A}_{\mathcal{S}}$ runs $\mathcal{A}_H$ on M_1 to produce a second message $M_2 \in \{0,1\}^{n_2}$ with $H(M_1) = H(M_2)$. $\mathcal{A}_{\mathcal{S}}$ then produces a signature σ on M_2 and outputs (M_2, σ). □

Lemma 25 (The case of primitive signatures). *Let $\mathcal{S} = \langle H, \Sigma \rangle$ be a deterministic hash-and-sign signature scheme and assume that $\mathcal{S}$ is primitive. Then*

$$\forall n > 0, \quad \mathsf{PRE}^n[H] \Rightarrow \mathsf{EF}^n\text{-}\mathsf{KOA}[\mathcal{S}] \ .$$

Proof ($\mathsf{PRE}^n[H] \Rightarrow \mathsf{EF}^n\text{-}\mathsf{KOA}[\mathcal{S}]$*).* Assuming black-box access to $\mathcal{A}_H$ which (τ_H, ε_H)-breaks $\mathsf{PRE}^n[H]$, we build an EF^n-KOA adversary $\mathcal{A}_{\mathcal{S}}$ which $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-breaks $\mathcal{S}$ where $\varepsilon_{\mathcal{S}} = \varepsilon_H$ and $\tau_{\mathcal{S}} = \tau_H + \mathsf{Time}(\mathcal{S}.\mathsf{Prim})$. $\mathcal{A}_{\mathcal{S}}$ is given a random key $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$. Since $\mathcal{S}$ is primitive, $\mathcal{A}_{\mathcal{S}}$ can generate a random pair $(m, \sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, m, u))$ by running $\mathcal{S}.\mathsf{Prim}(\mathsf{pk})$. Note that m is uniformly distributed over $\{0,1\}^{\mathsf{m}}$. $\mathcal{A}_{\mathcal{S}}$ then runs $\mathcal{A}_H(m)$ to produce $M \in \{0,1\}^n$ such that $H(M) = m$. $\mathcal{A}_{\mathcal{S}}$ outputs (M, σ). □

5.2 Proving $\mathcal{S} = \langle H, \Sigma \rangle$ Secure Assuming H Secure

We now turn to the positive security relations between $\mathcal{S}$ and H. As confirmed later in the paper, there seems to be no security proof relating the unforgeability(-ies) of $\mathcal{S}$ to H. Nevertheless, we evidence that the levels of repudiation of $\mathcal{S}$ can be guaranteed under security assumptions on H. This assumes, however, that $\mathcal{S}$ be injective.

Lemma 26 (The case of injective signatures). *Let $\mathcal{S}$ be a deterministic hash-and-sign signature scheme and assume $\mathcal{S}$ is injective. Then for any $n_1, n_2 > 0$,*

$$\mathsf{COL}^{n_1,n_2}[H] \Leftarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \tag{7}$$

$$\mathsf{SEC}^{n_2}_{n_1}[H] \Leftarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}] \tag{8}$$

and therefore $\mathsf{SEC}^{n_2}_{n_1}[H] \Leftrightarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}]$ and $\mathsf{COL}^{n_1,n_2}[H] \Leftrightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}]$ in virtue of (3) and (6).

Proof ($\mathsf{COL}^{n_1,n_2}[H] \Leftarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}]$*).* Let us assume an adversary $\mathcal{A}_{\mathcal{S}}$ which $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-breaks $\mathsf{ER}^{n_1,n_2}[\mathcal{S}]$. We build a collision-finder $\mathcal{A}_H$ which (τ_H, ε_H)-breaks $\mathsf{COL}^{n_1,n_2}[H]$ with $\varepsilon_H = \varepsilon_{\mathcal{S}}$ and $\tau_H = \tau_{\mathcal{S}} + \mathsf{Time}(\mathcal{S}.\mathsf{Gen})$. $\mathcal{A}_H$ generates a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$ and runs $\mathcal{A}_{\mathcal{S}}(\mathsf{pk}, \mathsf{sk})$. If $\mathcal{A}_{\mathcal{S}}$ outputs (M_1, M_2, σ) where $M_1 \in \{0,1\}^{n_1}$, $M_2 \in \{0,1\}^{n_2}$, $M_2 \neq M_1$ and σ is a valid signature on M_1 and M_2, then $\mathcal{A}_H$ discards σ and outputs (M_1, M_2). If σ is a signature on M_1 and M_2 simultaneously then $\sigma = \Sigma_2(\mathsf{sk}, H(M_1), r_1, \mathsf{aux}_1) = \Sigma_2(\mathsf{sk}, H(M_2), r_2, \mathsf{aux}_2)$ for some $\mathsf{aux}_1, \mathsf{aux}_2$ and $r_1, r_2 \in \{0,1\}^{\mathsf{r}}$. Since $\mathcal{S}$ is injective, one must have $r_1 = r_2$ and $H(M_1) = H(M_2)$ so that (M_1, M_2) is an (n_1, n_2)-collision. □

Proof ($\mathsf{SEC}^{n_2}_{n_1}[H] \Leftarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}]$*).* Assuming a $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-universal repudiator $\mathcal{A}_{\mathcal{S}}$, we build an algorithm $\mathcal{A}_H$ which (τ_H, ε_H)-breaks $\mathsf{SEC}^{n_2}_{n_1}[H]$ with $\varepsilon_H = \varepsilon_{\mathcal{S}}$ and $\tau_H = \tau_{\mathcal{S}} + \mathsf{Time}(\mathcal{S}.\mathsf{Gen})$. Given a random $M_1 \leftarrow \{0,1\}^{n_1}$, $\mathcal{A}_H$ generates a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$ and runs $\mathcal{A}_{\mathcal{S}}(\mathsf{pk}, \mathsf{sk}, M_1)$ to construct a pair (M_2, σ) where $M_2 \in \{0,1\}^{n_2}$, $M_2 \neq M_1$ and σ is a valid signature on M_1 and M_2. The latter condition implies $\sigma = \Sigma_2(\mathsf{sk}, H(M_1), r_1, \mathsf{aux}_1) = \Sigma_2(\mathsf{sk}, H(M_2), r_2, \mathsf{aux}_2)$ for some $\mathsf{aux}_1, \mathsf{aux}_2$ and $r_1, r_2 \in \{0,1\}^{\mathsf{r}}$. Since $\mathcal{S}$ is injective, one must have $r_1 = r_2$ and $H(M_1) = H(M_2)$ so that M_2 yields an n_2-bit second preimage of $H(M_1)$. $\mathcal{A}_H$ then outputs M_2. □

5.3 Impossible Reductions Between Security Notions for $\mathcal{S} = \langle H, \Sigma \rangle$ and H

Let $\mathcal{S} = \langle H, \Sigma \rangle$ be a deterministic hash-and-sign signature scheme. The results above show that breaking H in the strongest sense, namely breaking $\mathsf{PRE}[H]$, is enough to break $\mathsf{EF\text{-}ATK}[\mathcal{S}]$ for $\mathsf{ATK} \in \{\mathsf{KOA}, \mathsf{KMA}, \mathsf{CMA}\}$ as well as $\mathsf{UF\text{-}CMA}[\mathcal{S}]$ but seems insufficient to break either $\mathsf{UF\text{-}KOA}[\mathcal{S}]$ or $\mathsf{UF\text{-}KMA}[\mathcal{S}]$. We now want to ascertain that it is actually impossible for these two security levels to fall even when one assumes breaking $\mathsf{PRE}[H]$ is easy.

What does it take to break $\mathsf{UF\text{-}KOA}[\mathcal{S}]$? We consider $\mathsf{UF\text{-}KOA}[\mathcal{S}]$ and show that this security level does not fall when $\mathsf{PRE}[H]$ is broken. This impossibility is based on the following simple observation:

Lemma 27. *For any $n > 0$, $\mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}] \Leftarrow \mathsf{UF\text{-}KOA}[\Sigma] \Leftarrow \mathsf{PRE}^n[H] \wedge \mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}]$.*

Proof. (Left $\Leftarrow$). Given a $(\tau_\Sigma, \varepsilon_\Sigma)$-universal forger $\mathcal{A}_\Sigma$ for Σ, we build as follows a reduction algorithm $\mathcal{A}_{\mathcal{S}}$ which $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-breaking $\mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}]$ for any $n > 0$ with $\varepsilon_{\mathcal{S}} = \varepsilon_\Sigma$ and $\tau_{\mathcal{S}} = \tau_\Sigma + \mathsf{Time}(H, n)$. Given a random $M \leftarrow \{0,1\}^n$, $\mathcal{A}_{\mathcal{S}}$ computes $m = H(M)$, runs $\mathcal{A}_\Sigma(m)$ and outputs $\mathcal{A}_\Sigma$'s output. (Right $\Leftarrow$). Assume we are given two probabilistic algorithms $\mathcal{A}_H$ and $\mathcal{A}_{\mathcal{S}}$ such that $\mathcal{A}_H$ (τ_H, ε_H)-breaks $\mathsf{PRE}^n[H]$ and $\mathcal{A}_{\mathcal{S}}$ $(\tau_{\mathcal{S}}, \varepsilon_{\mathcal{S}})$-breaks $\mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}]$. We construct a reduction algorithm $\mathcal{A}_\Sigma$ which $(\tau_\Sigma, \varepsilon_\Sigma)$-breaks $\mathsf{UF\text{-}KOA}[\Sigma]$ with $\tau_\Sigma = \varepsilon_H \varepsilon_{\mathcal{S}}$ and $\tau_\Sigma = \tau_H + \tau_{\mathcal{S}}$. Given a random $\mathsf{pk} \leftarrow \Sigma.\mathsf{Gen}$ and a random $m \leftarrow \{0,1\}^{\mathsf{m}}$, $\mathcal{A}_\Sigma$ runs $\mathcal{A}_H(m)$ to get $M \in \{0,1\}^n$ such that $H(M) = m$ and runs $\mathcal{A}_{\mathcal{S}}(M)$ to get $\sigma = \Sigma(H(M), r) = \Sigma(m, r)$ for some $r \in \{0,1\}^{\mathsf{r}}$.

We now see that a polynomial reduction $\mathcal{R}$ such that $\mathsf{PRE}^n[H] \geq_{\mathcal{R}} \mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}]$ is very unlikely. Indeed, assuming a reduction $\mathcal{R}_1$ converting an algorithm solving $\mathsf{PRE}^n[H]$ into an attacker $\mathcal{A}_{\mathcal{S}}$ against $\mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}]$, we would build a second reduction $\mathcal{R}_2$ which converts $\mathcal{A}_H$ into an algorithm breaking $\mathsf{UF\text{-}KOA}[\Sigma]$ with a similar efficiency. Since H and Σ are two independent ingredients of $\mathcal{S}$ and because Σ is a scheme parameter here, such a hierarchy does not exist. If it existed, then Σ and H would admit some form of "morphological interaction" such as sharing identical components. This cannot be the case for general Σ's. In fact, the only effect breaking $\mathsf{PRE}[H]$ has on $\mathcal{S}$ is that for any $n > 0$, $\mathsf{UF}_n\text{-}\mathsf{KOA}[\mathcal{S}] \Leftrightarrow \mathsf{UF\text{-}KOA}[\Sigma]$, thus rendering $\mathcal{S}$ exactly as secure as Σ in the sense of $\mathsf{UF\text{-}KOA}$ security.

Can $\mathsf{PRE}[H]$ break $\mathsf{UF\text{-}KMA}[\mathcal{S}]$ then? It is easily seen that the reductions of Lemma 27 cannot be readily extended to security levels $\{\mathsf{UF}_n\text{-}\mathsf{KMA}\}_{n>0}$. If the reduction $\ell\text{-}\mathsf{UF}_n\text{-}\mathsf{KMA}[\mathcal{S}] \Leftarrow \ell\text{-}\mathsf{UF\text{-}KMA}[\Sigma]$ stands for any $n > 0$ and $\ell \geq 0$, the only security reduction $\ell\text{-}\mathsf{UF\text{-}KMA}[\Sigma] \Leftarrow \mathsf{PRE}^n[H] \wedge \ell\text{-}\mathsf{UF}_n\text{-}\mathsf{KMA}[\mathcal{S}]$ which seems to work consists in converting the ℓ message-signature pairs given to the $\ell\text{-}\mathsf{UF\text{-}KMA}[\Sigma]$ attacker into a list of ℓ message-signature pairs for $\mathcal{S}$. This requires breaking $\mathsf{PRE}^n[H]$ exactly ℓ times, thereby introducing a ε_H^ℓ term in the reduction cost. Such a reduction is therefore loose as ℓ grows and cannot be seen as efficient for large ℓ's. We leave this question fully open.

6 Security Relations for Probabilistic Hash-and-Sign Signatures

We now move on to the case of a probabilistic hash-and-sign signature scheme $\mathcal{S} = \langle F, \Sigma \rangle$ where similarly as in the deterministic case, our goal is to relate the security of $\mathcal{S}$ to the one of its hash component F, the underlying signature scheme Σ being seen as a fixed parameter. Σ_1, Σ_2, Υ_1 and Υ_2 denote the inner functions of Σ.

6.1 Attacking $\mathcal{S} = \langle F, \Sigma \rangle$ by Attacking F

Lemma 28. *Let $\mathcal{S} = \langle F, \Sigma \rangle$ be a probabilistic hash-and-sign signature scheme as per Definition 21. Then for any integers $n_1, n_2 > 0$,*

$$\mathsf{A\text{-}COL}^{n_1,n_2}[F] \Rightarrow 1\text{-}\mathsf{EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}], 1\text{-}\mathsf{EF}^{n_2}\text{-}\mathsf{CMA}[\mathcal{S}] \tag{9}$$

$$\mathsf{U\text{-}COL}^{n_1,n_2}[F] \Rightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \tag{10}$$

$$\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow 1\text{-}\mathsf{UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}] \tag{11}$$

$$\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow 1\text{-}\mathsf{EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}] \tag{12}$$

$$\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}] \tag{13}$$

Proof ($\mathsf{A\text{-}COL}^{n_1,n_2}[F] \Rightarrow 1\text{-}\mathsf{EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}]$*).* Given a (τ_F, ε_F)-solver $\mathcal{A}_F$ for $\mathsf{A\text{-}COL}^{n_1,n_2}[F]$ we build an $\mathsf{EF}^{n_1}\text{-}\mathsf{CMA}$ attacker $\mathcal{A}_\mathcal{S}$ breaking $\mathcal{S}$ with probability $\varepsilon_\mathcal{S} = \varepsilon_F$ and $\tau_\mathcal{S} = \tau_F + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_2)$ in exactly one call to the signing oracle. Given a random public key $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$, our attacker $\mathcal{A}_\mathcal{S}$ runs $\mathcal{A}_F$ to produce an (n_1, n_2)-absolute-collision (M_1, M_2), namely a pair $(M_1, M_2) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$ such that $M_1 \neq M_2$ and $F(M_1, r) = F(M_2, r)$ for any $r \in \{0,1\}^{\mathsf{r}}$. Then $\mathcal{A}_\mathcal{S}$ requests a signature $\sigma = \Sigma_2(\mathsf{sk}, F(M_2, r), r, \mathsf{aux})$ on M_2 (where $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$ for some $u \in \{0,1\}^{\mathsf{u}}$) and produces σ as a valid signature on $M_1 \neq M_2$. The same works with $1\text{-}\mathsf{EF}^{n_2}\text{-}\mathsf{CMA}$ as well by symmetry. □

Proof ($\mathsf{U\text{-}COL}^{n_1,n_2}[F] \Rightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}]$*).* Given a (τ_F, ε_F)-existential-collision-finder $\mathcal{A}_F$ we build a repudiator $\mathcal{A}_\mathcal{S}$ breaking $\mathsf{ER}^{n_1,n_2}[\mathcal{S}]$ with probability $\varepsilon_\mathcal{S} = \varepsilon_F$ and time $\tau_\mathcal{S} = \tau_F + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_1)$. Given a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$, $\mathcal{A}_\mathcal{S}$ randomly picks $u \leftarrow \{0,1\}^{\mathsf{u}}$ and computes $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$. Since u is taken uniformly at random, r is uniformly distributed over $\{0,1\}^{\mathsf{r}}$. Now $\mathcal{A}_\mathcal{S}$ runs $\mathcal{A}_F(r)$ to build an (n_1, n_2)-collision for r, *i.e.* $(M_1, M_2) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$ such that $F(M_2, r) = F(M_1, r)$. Then $\mathcal{A}_\mathcal{S}$ finishes to sign M_1 by computing $\sigma = \Sigma_2(\mathsf{sk}, F(M_1, r), r, \mathsf{aux})$ and outputs the tuple (M_1, M_2, σ). □

Proof ($\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow 1\text{-}\mathsf{UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}]$*).* Assume that $\mathcal{A}_F$ (τ_F, ε_F)-solves $\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F]$. We build a $\mathsf{UF}_{n_1}\text{-}\mathsf{CMA}$ attacker $\mathcal{A}_\mathcal{S}$ which $(\tau_\mathcal{S}, \varepsilon_\mathcal{S})$-breaks $\mathcal{S}$ in one call to the signing oracle with $\varepsilon_\mathcal{S} = \varepsilon_H$ and $\tau_\mathcal{S} = \tau_H + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_2)$. Given a public key $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$ and $M_1 \leftarrow \{0,1\}^{n_1}$, $\mathcal{A}_\mathcal{S}$ must produce a valid signature on M_1. $\mathcal{A}_\mathcal{S}$ runs $\mathcal{A}_F(M_1)$ to build an (n_1, n_2)-absolute-collision (M_1, M_2). $\mathcal{A}_\mathcal{S}$ then requests a signature on M_2 and is given $\sigma = \Sigma_2(\mathsf{sk}, F(M_2, r), r, \mathsf{aux})$ where $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$ for some $u \in \{0,1\}^{\mathsf{u}}$. $\mathcal{A}_\mathcal{S}$ then returns (M_1, σ). □

Proof ($\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow 1\text{-}\mathsf{EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}]$*).* Assuming $\mathcal{A}_F$ (τ_F, ε_F)-solves $\mathsf{SEC}^{n_2}_{n_1}[F]$, we build an attacker $\mathcal{A}_\mathcal{S}$ which $(\tau_\mathcal{S}, \varepsilon_\mathcal{S})$-solves $1\text{-}\mathsf{EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}]$ with $\varepsilon_\mathcal{S} = \varepsilon_F$ and $\tau_\mathcal{S} = \tau_F$. $\mathcal{A}_\mathcal{S}$ is given a random $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$ and a single random message-signature pair (M_1, σ) where $M_1 \leftarrow \{0,1\}^{n_1}$ and $\sigma = \Sigma_2(\mathsf{sk}, F(M_1, r), r, \mathsf{aux})$ with $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$ for $u \leftarrow \{0,1\}^{\mathsf{u}}$. Since u is uniform, note that r is uniform too. $\mathcal{A}_\mathcal{S}$ runs $\mathcal{A}_F(M_1, r)$ to construct an n_2-bit string $M_2 \neq M_1$ with $F(M_2, r) = F(M_1, r)$. $\mathcal{A}_\mathcal{S}$ then returns the existential forgery (M_2, σ). □

Proof ($\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}]$*).* Assuming $\mathcal{A}_F$ (τ_F, ε_F)-solves $\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F]$, we build a $(\tau_\mathcal{S}, \varepsilon_\mathcal{S})$-repudiator $\mathcal{A}_\mathcal{S}$ which on input a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$ and a random message $M_1 \in \{0,1\}^{n_1}$, returns a message $M_2 \neq M_1$, $|M_2| = n_2$ and σ such that σ is a valid signature on both M_1 and M_2. Here again, $\varepsilon_\mathcal{S} = \varepsilon_F$ and $\tau_\mathcal{S} = \tau_F + \mathsf{Time}(\mathcal{S}.\mathsf{Sign}, n_2)$. $\mathcal{A}_\mathcal{S}$ randomly picks $u \leftarrow \{0,1\}^{\mathsf{u}}$ and computes $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$. Since u is taken uniformly at random, r is uniformly distributed over $\{0,1\}^{\mathsf{r}}$. Then $\mathcal{A}_\mathcal{S}$ runs $\mathcal{A}_F(M_1, r)$ to produce a string $M_2 \in \{0,1\}^{n_2}$, $M_2 \neq M_1$, such that $F(M_2, r) = F(M_1, r)$. $\mathcal{A}_\mathcal{S}$ then computes $\sigma = \Sigma_2(\mathsf{sk}, F(M_1, r), r, \mathsf{aux})$ and outputs (M_2, σ). □

Lemma 29 (The case of primitive signatures). *Let $\mathcal{S} = \langle F, \Sigma \rangle$ be a probabilistic hash-and-sign signature scheme and assume that $\mathcal{S}$ is primitive. Then*

$$\forall n > 0, \quad \mathsf{U\text{-}PRE}^{n}[F] \Rightarrow \mathsf{EF}^{n}\text{-}\mathsf{KOA}[\mathcal{S}]$$

Proof ($\mathsf{U\text{-}PRE}^n\,[F] \Rightarrow \mathsf{EF}^n\text{-}\mathsf{KOA}\,[\mathcal{S}]$*).* Assuming black-box access to $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{U\text{-}PRE}^n\,[F]$, we build an EF^n-KOA adversary $\mathcal{A}_\mathcal{S}$ which $(\tau_\mathcal{S}, \varepsilon_\mathcal{S})$-breaks $\mathcal{S}$ where $\varepsilon_\mathcal{S} = \varepsilon_F$ and $\tau_\mathcal{S} = \tau_F + \mathsf{Time}\,(\mathcal{S}.\mathsf{Prim}) + \mathsf{Time}\,(\mathcal{S}.\mathsf{Ver})$. $\mathcal{A}_\mathcal{S}$ is given a random key $\mathsf{pk} \leftarrow \mathcal{S}.\mathsf{Gen}()$. Since $\mathcal{S}$ is primitive, $\mathcal{A}_\mathcal{S}$ can generate a random pair $(m, \sigma = \mathcal{S}.\mathsf{Sign}(\mathsf{sk}, m, u))$ by running $\mathcal{S}.\mathsf{Prim}(\mathsf{pk})$. Note that m is uniformly distributed over $\{0,1\}^\mathsf{m}$ and u is uniform over $\{0,1\}^\mathsf{u}$, thereby making $r = \Upsilon_1(\mathsf{pk}, \sigma)$ uniform over $\{0,1\}^\mathsf{r}$. Now $\mathcal{A}_\mathcal{S}$ runs $\mathcal{A}_F(m, r)$ to construct $M \in \{0,1\}^n$ such that $F(M, r) = m$. $\mathcal{A}_\mathcal{S}$ then outputs (M, σ). □

6.2 Proving $\mathcal{S} = \langle F, \Sigma \rangle$ Secure Assuming F Secure

We are now looking for positive security relations between $\mathcal{S}$ and F. As in the deterministic case, it seems unlikely (although we do not disprove it) that security proof exist wich relate the unforgeability of $\mathcal{S}$ to F. Assuming $\mathcal{S}$ injective, however, is enough to show that non-repudiation can be guaranteed under security assumptions on F.

Lemma 30 (The case of injective signatures). *Let $\mathcal{S}$ be a probabilistic hash-and-sign signature scheme and assume $\mathcal{S}$ is injective. Then for any $n_1, n_2 > 0$,*

$$\mathsf{E\text{-}COL}^{n_1,n_2}\,[F] \Leftarrow \mathsf{ER}^{n_1,n_2}\,[\mathcal{S}] \tag{14}$$

$$\mathsf{E\text{-}SEC}^{n_2}_{n_1}\,[F] \Leftarrow \mathsf{UR}^{n_2}_{n_1}\,[\mathcal{S}]\ . \tag{15}$$

Proof ($\mathsf{E\text{-}COL}^{n_1,n_2}\,[F] \Leftarrow \mathsf{ER}^{n_1,n_2}\,[\mathcal{S}]$*).* Let us assume an adversary $\mathcal{A}_\mathcal{S}$ which $(\tau_\mathcal{S}, \varepsilon_\mathcal{S})$-breaks $\mathsf{ER}^{n_1,n_2}\,[\mathcal{S}]$. We build an existential collision-finder $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{E\text{-}COL}^{n_1,n_2}\,[F]$ with $\varepsilon_F = \varepsilon_\mathcal{S}$ and $\tau_F = \tau_\mathcal{S} + \mathsf{Time}\,(\mathcal{S}.\mathsf{Gen}) + +\mathsf{Time}\,(\mathcal{S}.\mathsf{Ver})$. $\mathcal{A}_F$ generates a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$ and runs $\mathcal{A}_\mathcal{S}(\mathsf{pk}, \mathsf{sk})$. If $\mathcal{A}_\mathcal{S}$ outputs (M_1, M_2, σ) where $M_1 \in \{0,1\}^{n_1}$, $M_2 \in \{0,1\}^{n_2}$, $M_2 \neq M_1$ and σ is a valid signature on M_1 and M_2, then $\mathcal{A}_F$ computes $r = \Upsilon_1(\mathsf{pk}, \sigma)$ and outputs (M_1, M_2, r). If σ is a signature on M_1 and M_2 simultaneously then $\sigma = \Sigma_2(\mathsf{sk}, F(M_1, r_1), r_1, \mathsf{aux}_1) = \Sigma_2(\mathsf{sk}, F(M_2, r_2), r_2, \mathsf{aux}_2)$ for some $\mathsf{aux}_1, \mathsf{aux}_2$ and $r_1, r_2 \in \{0,1\}^\mathsf{r}$. Since $\mathcal{S}$ is injective, one must have $r_1 = r_2 = r$ and $F(M_1, r) = F(M_2, r)$ so that (M_1, M_2, r) is an (n_1, n_2)-existential collision. □

Proof ($\mathsf{E\text{-}SEC}^{n_2}_{n_1}\,[F] \Leftarrow \mathsf{UR}^{n_2}_{n_1}\,[\mathcal{S}]$*).* Assuming a $(\tau_\mathcal{S}, \varepsilon_\mathcal{S})$-universal repudiator $\mathcal{A}_\mathcal{S}$, we build an algorithm $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{E\text{-}SEC}^{n_2}_{n_1}\,[F]$ with $\varepsilon_F = \varepsilon_\mathcal{S}$ and $\tau_F = \tau_\mathcal{S} + \mathsf{Time}\,(\mathcal{S}.\mathsf{Gen}) + \mathsf{Time}\,(\mathcal{S}.\mathsf{Ver})$. Given a random $M_1 \leftarrow \{0,1\}^{n_1}$, $\mathcal{A}_F$ generates a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{Gen}()$ and runs $\mathcal{A}_\mathcal{S}(\mathsf{pk}, \mathsf{sk}, M_1)$ to construct a pair (M_2, σ) where $M_2 \in \{0,1\}^{n_2}$, $M_2 \neq M_1$ and σ is a valid signature on M_1 and M_2. In this case, $\mathcal{A}_F$ computes $r = \Upsilon_1(\mathsf{pk}, \sigma)$ and outputs (M_2, r). σ being a signature on both M_1 and M_2 implies $\sigma = \Sigma_2(\mathsf{sk}, F(M_1, r_1), r_1, \mathsf{aux}_1) = \Sigma_2(\mathsf{sk}, F(M_2, r_2), r_2, \mathsf{aux}_2)$ for some $\mathsf{aux}_1, \mathsf{aux}_2$ and $r_1, r_2 \in \{0,1\}^\mathsf{r}$. Since $\mathcal{S}$ is injective, one must have $r_1 = r_2 = r$ and $F(M_1, r) = F(M_2, r)$ so that (M_2, r) yields an n_2-bit second preimage of $F(M_1, r)$. □

Discussion. As opposed to deterministic hash-an-sign signatures, the probabilistic hash-and-sign paradigm inherently offers better security guarantees. For instance, breaking $\mathcal{S}$ in the $\mathsf{EF\text{-}CMA}$ sense is easy if H is not collision-resistant in the deterministic case $\mathcal{S} = \langle H, \Sigma \rangle$. Breaking the same security level for $\mathcal{S} = \langle F, \Sigma \rangle$, however, seems to require the generation of absolute collisions for F, a much stronger result. Overall, given a fixed-size signature scheme Σ, it seems largely preferable to domain-extend it in the probabilistic way for which security is obtained under much weaker assumptions on the security of the inner hash component.

7 Merkle-Damgård-based Instantiations of F

In previous sections, we introduced and made use of new security notions for a hash function family $F : \{0,1\}^* \times \{0,1\}^\mathsf{r} \rightarrow \{0,1\}^\mathsf{m}$. To exemplify our investigation on the impacts of F on a signature scheme $\mathcal{S} = \langle F, \Sigma \rangle$, we analyze three practical instantiations of F using a Merkle-Damgård (MD for short) hash function. This is motivated by the fact that SHA-1, MD5 and most of other hash functions we use in practice are obtained by applying some variant of the Merkle-Damgård construction to an underlying compression function $f : \{0,1\}^\mathsf{m} \times \{0,1\}^\mathsf{b} \rightarrow \{0,1\}^\mathsf{m}$. First, we relate attackers against F and attackers against the Merkle-Damgård hash function. This allows to reformulate the security results of the previous sections which involve F into more specific security statements involving the MD hash function directly. Based on these reductions, we display the effective workload of the best known attacks against F in every instantiation.

7.1 Instantiating F with any hash function H

Let H be a hash function. There are many ways one may attempt to construct a hash function family F using H. We consider later on the two most "natural" constructions, namely $F(M,r) = H(M||r)$ and $F(M,r) = H(r||M)$. For these two operating modes (we actually consider a wider class $F(M,r) = H(\llbracket M,r \rrbracket)$ of operating modes), we relate the security of F – as defined in Section 3.2, to the one of H – as defined in Section 3.1. Combining these new security relations with the results of Section 6, it is easy to reformulate attacks or security assumptions on hash-and-sign signatures in terms of attacks or security assumptions on H.

Lemma 31. *Let $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ be a hash function and let $\llbracket M,r \rrbracket$ denote an $(|M| + \mathsf{r})$-bit (one-to-one) encoding of the pair $(M,r) \in \{0,1\}^* \times \{0,1\}^{\mathsf{r}}$. Let now F be the hash function family defined on $M \in \{0,1\}^*$ and $r \in \{0,1\}^{\mathsf{r}}$ (r being arbitrary) as $F(M,r) = H(\llbracket M,r \rrbracket)$. Then for any $n_1, n_2 > 0$,*

$$\begin{array}{ccccccc}
\mathsf{PRE}^{n_2+\mathsf{r}}[H] & \underline{\Leftrightarrow} & \mathsf{E\text{-}PRE}^{n_2}[F] & \Leftarrow & \mathsf{U\text{-}PRE}^{n_2}[F] & & \\
 & & & & \Downarrow & & \\
 & & \mathsf{SEC}^{n_2+\mathsf{r}}_{n_1+\mathsf{r}}[H] & \underline{\Leftarrow} & \mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] & \Leftarrow & \mathsf{A\text{-}SEC}^{n_2}_{n_1}[F] \\
 & & & & \Downarrow & & \Downarrow \\
\mathsf{COL}^{n_1+\mathsf{r},n_2+\mathsf{r}}[H] & \underline{\Leftarrow} & \mathsf{E\text{-}COL}^{n_1,n_2}[F] & \Leftarrow & \mathsf{U\text{-}COL}^{n_1,n_2}[F] & \Leftarrow & \mathsf{A\text{-}COL}^{n_1,n_2}[F]
\end{array}$$

where non-underlined relations are already known by Theorem 17.

Proof ($\mathsf{COL}^{n_1+\mathsf{r},n_2+\mathsf{r}}[H] \Leftarrow \mathsf{E\text{-}COL}^{n_1,n_2}[F]$). Given an adversary $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{E\text{-}COL}^{n_1,n_2}[F]$, we construct a reduction $\mathcal{A}_H$ which (τ_H, ε_H)-breaks $\mathsf{COL}^{n_1+\mathsf{r},n_2+\mathsf{r}}[H]$ with $\tau_H = \tau_F$ and $\varepsilon_H = \varepsilon_F + \mathsf{Time}(\text{encode}\,\llbracket \cdot,\cdot \rrbracket, n_1, \mathsf{r}) + \mathsf{Time}(\text{encode}\,\llbracket \cdot,\cdot \rrbracket, n_2, \mathsf{r})$. $\mathcal{A}_H$ runs $\mathcal{A}_F$ to construct a tuple (M_1, M_2, r) such that $F(M_1,r) = F(M_2,r)$, that is, $H(\llbracket M_1,r \rrbracket) = H(\llbracket M_2,r \rrbracket)$. $\mathcal{A}_H$ then outputs the $(n_1+\mathsf{r}, n_2+\mathsf{r})$-collision $(\llbracket M_1,r \rrbracket, \llbracket M_2,r \rrbracket)$ on H.

Proof ($\mathsf{SEC}^{n_2+\mathsf{r}}_{n_1+\mathsf{r}}[H] \Leftarrow \mathsf{U\text{-}SEC}^{n_2}_{n_1}[F]$). Given an adversary $\mathcal{A}_F$ (τ_F, ε_F)-breaking $\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F]$, one builds a reduction $\mathcal{A}_H$ which (τ_H, ε_H)-breaks $\mathsf{SEC}^{n_2+\mathsf{r}}_{n_1+\mathsf{r}}[H]$ with $\tau_H = \tau_F + \mathsf{Time}(\text{parse}\,\llbracket \cdot,\cdot \rrbracket, n_1, \mathsf{r}) + \mathsf{Time}(\text{encode}\,\llbracket \cdot,\cdot \rrbracket, n_2, \mathsf{r})$ and $\varepsilon_H = \varepsilon_F$. Given a random $\bar{M}_1 \leftarrow \{0,1\}^{n_1+\mathsf{r}}$, $\mathcal{A}_H$ parses $\bar{M}_1$ as $\bar{M}_1 = \llbracket M_1,r \rrbracket$ where $|M_1| = n_1$ and $|r| = \mathsf{r}$. Then $\mathcal{A}_H$ runs $\mathcal{A}_F(M_1,r)$ to obtain an n_2-bit string M_2 such that $F(M_2,r) = F(M_1,r)$ *i.e.* $H(\bar{M}_2) = H(\bar{M}_1)$. $\mathcal{A}_H$ then outputs $\bar{M}_2 = \llbracket M_2,r \rrbracket$. □

Proof ($\mathsf{PRE}^{n_2+\mathsf{r}}[H] \Leftarrow \mathsf{E\text{-}PRE}^{n_2}[F]$). Given an adversary $\mathcal{A}_F$ (τ_F, ε_F)-breaking $\mathsf{E\text{-}PRE}^{n_2}[F]$, one builds a reduction $\mathcal{A}_H$ which (τ_H, ε_H)-breaks $\mathsf{PRE}^{n_2+\mathsf{r}}[H]$ with $\varepsilon_H = \varepsilon_F$ and $\tau_H = \tau_F + \mathsf{Time}(\text{encode}\,\llbracket \cdot,\cdot \rrbracket, n_2, \mathsf{r})$. Given a random $m \leftarrow \{0,1\}^{\mathsf{m}}$, $\mathcal{A}_H$ runs $\mathcal{A}_F(m)$ to construct a pair $(M,r) \in \{0,1\}^{n_2} \times \{0,1\}^{\mathsf{r}}$ such that $F(M,r) = m$. Then $\mathcal{A}_H$ outputs $\bar{M} = \llbracket M,r \rrbracket$. Obviously $H(\bar{M}) = m$. □

Proof ($\mathsf{PRE}^{n_2+\mathsf{r}}[H] \Rightarrow \mathsf{E\text{-}PRE}^{n_2}[F]$). Given an adversary $\mathcal{A}_H$ (τ_H, ε_H)-breaking $\mathsf{PRE}^{n_2+\mathsf{r}}[H]$, one builds a reduction $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{E\text{-}PRE}^{n_2}[F]$ with $\varepsilon_F = \varepsilon_H$ and $\tau_F = \tau_H + \mathsf{Time}(\text{parse}\,\llbracket \cdot,\cdot \rrbracket, n_2, \mathsf{r})$. Given a random $m \leftarrow \{0,1\}^{\mathsf{m}}$, $\mathcal{A}_F$ runs $\mathcal{A}_H(m)$ to construct an $(n_2+\mathsf{r})$-bit string $\bar{M}$ such that $H(\bar{M}) = m$. Then $\mathcal{A}_F$ parses $\bar{M}$ as $\bar{M} = \llbracket M,r \rrbracket$ where $|M| = n_2$ and $|r| = \mathsf{r}$ and outputs (M,r). Obviously $F(M,r) = m$. □

7.2 Merkle-Damgård hash functions

Let $f : \{0,1\}^{\mathsf{m}} \times \{0,1\}^{\mathsf{b}} \to \{0,1\}^{\mathsf{m}}$ be a compression function, $g : \{0,1\}^* \to (\{0,1\}^{\mathsf{b}})^*$ an arbitrary function (called message padding or padding) and $IV_0 \in \{0,1\}^{\mathsf{m}}$. We recall that a hash function H can be obtained from these components by using iterated hashing as described in Definition 3. We write $H = \mathsf{ITER}[f, g, IV_0]$.

Definition 32 (Collision-propagating paddings). *Let $k_1, k_2 > 0$. A padding function $g : \{0,1\}^* \to (\{0,1\}^{\mathsf{b}})^*$ is said to be (k_1,k_2)-collision-propagating when for any compression function $f : \{0,1\}^{\mathsf{m}} \times \{0,1\}^{\mathsf{b}} \to \{0,1\}^{\mathsf{m}}$ and any $IV_0 \in \{0,1\}^{\mathsf{m}}$, the hash function $H = \mathsf{ITER}[f,g,IV_0]$ is such that for any k_1-block string $M_1 \in \{0,1\}^{k_1 \cdot \mathsf{b}}$ and k_2-block string $M_2 \in \{0,1\}^{k_2 \cdot \mathsf{b}}$, if $H(M_1) = H(M_2)$ then $H(M_1 \parallel M) = H(M_2 \parallel M)$ for any $M \in \{0,1\}^*$.*

The basic and strengthened Merkle-Damgård constructions are a specific case of iterated hashing using specific functions $g = g_0$ or $g = g_s$ defined as follows. For any $M \in \{0,1\}^*$, $g_0(M)$ is obtained by appending to M as many 0-bits as necessary to yield a $\lceil |M|/\mathsf{b} \rceil$-block string. Let $\mathsf{a} \leq \mathsf{b}$ be a size parameter. Given $M \in \{0,1\}^*$, $g_s(M)$ is formed by appending a single 1-bit to M and as many 0-bits as required so that the length of the so-obtained string is congruent to $\mathsf{b} - \mathsf{a}$ modulo b. Then the bitlength of M, seen as an a-bit string, is appended:

$$\begin{array}{ll} g_0(M) = M \parallel 0^t & \text{with} \quad t = -|M| \bmod \mathsf{b}\ , \\ g_s(M) = M \parallel 1 \parallel 0^t \parallel [|M|]_{\mathsf{a}} & \text{with} \quad t = -(|M| + \mathsf{a} + 1) \bmod \mathsf{b}\ . \end{array}$$

Appending a logical length-block prior to hashing is called MD-*strengthening* [7, 8, 17]. This technique aims at preventing collision and pseudo-collision attacks which find colliding messages of different length, including trivial collisions for random IV's, long message attacks and fixed point attacks [22, 26]. The protection it seems to offer justifies the practical use of MD-strengthening in common hash functions.

Let us now fix $f : \{0,1\}^{\mathsf{m}} \times \{0,1\}^{\mathsf{b}} \to \{0,1\}^{\mathsf{m}}$ and $IV_0 \in \{0,1\}^{\mathsf{m}}$. The related Merkle-Damgård hash function without MD-strengthening is the hash function $H_0 = \mathsf{ITER}\,[f, g_0, IV_0]$. Its counterpart with MD-strengthening is the hash function $H_s = \mathsf{ITER}\,[f, g_s, IV_0]$. We may sometimes use $H_{x,IV} = \mathsf{ITER}\,[f, g_x, IV]$ for $x = 0$ or s.

Proposition 33. *a) g_0 is a (k_1, k_2)-collision-propagating padding for any $k_1, k_2 > 0$. b) g_s is a (k, k)-collision-propagating padding for any $k > 0$.*

Hence, it is rather obvious that MD-strengthening is not enough to thwart attacks based on propagating collisions in a way or another. In fact, all known second preimage or collision-finding attacks against Merkle-Damgård hash functions with MD-strengthening generate messages with the same block length so that the padding g_s, which only depends on the length of the input message, is the same for generated messages and therefore does not interfere with the attack. In fact, it is easily seen that for any $n = k \cdot \mathsf{b} > 0$, $\mathsf{PRE}^n\,[H_0] \Leftarrow \mathsf{PRE}^n\,[H_s]$, $\mathsf{SEC}^n_n\,[H_s] \Leftarrow \mathsf{SEC}^n_n\,[H_0]$ and $\mathsf{COL}^{n,n}\,[H_s] \Leftarrow \mathsf{COL}^{n,n}\,[H_0]$. We therefore restrict ourselves to instantiations of F based on $H_s = \mathsf{ITER}\,[f, g_s, IV_0]$, knowing in advance that all recent SEC and COL attacks against H_0, which retrieve same-length colliding messages, equally apply to the real-life setting $H_s = \mathsf{ITER}\,[f, g_s, IV_0]$.

7.3 MD-based hash function families

Lemma 34. *Let $H_s = \mathsf{ITER}\,[f, g_s, IV_0]$ be an* MD *hash function with* MD *strengthening and F defined on $\{0,1\}^* \times \{0,1\}^{\mathsf{r}}$ as $F(M, r) = H_s(M \| r)$. Then for any $n = k \cdot \mathsf{b} > 0$,*

$$\begin{array}{ccccc} \mathsf{A\text{-}SEC}^n_n\,[F] & \Leftarrow & \mathsf{SEC}^n_n\,[H_s] & \Leftarrow & \mathsf{SEC}^n_n\,[H_0] \\ \Downarrow & & \Downarrow & & \Downarrow \\ \mathsf{A\text{-}COL}^{n,n}\,[F] & \Leftarrow & \mathsf{COL}^{n,n}\,[H_s] & \Leftarrow & \mathsf{COL}^{n,n}\,[H_0] \end{array}$$

Proof ($\mathsf{A\text{-}COL}^{n_1,n_2}\,[F] \Leftarrow \mathsf{COL}^{n_1,n_2}\,[H_s]$ *for any $n_1, n_2 > 0$).* Given an adversary $\mathcal{A}_{H_s}$ which $(\tau_{H_s}, \varepsilon_{H_s})$-breaks $\mathsf{COL}^{n_1,n_2}\,[H_s]$, one builds a reduction $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{A\text{-}COL}^{n_1,n_2}\,[F]$ with $\varepsilon_F = \varepsilon_{H_s}$ and $\tau_F = \tau_{H_s}$. $\mathcal{A}_F$ runs $\mathcal{A}_{H_s}$ to build an (n_1, n_2)-collision (M_1, M_2) on H and outputs it. Since g is collision-propagating, one has $H(M_1 \| r) = H(M_2 \| r)$ for any $r \in \{0,1\}^{\mathsf{r}} \subseteq \{0,1\}^*$, meaning that (M_1, M_2) is an (n_1, n_2)-absolute collision on F. □

Proof ($\mathsf{A\text{-}SEC}^{n_2}_{n_1}\,[F] \Leftarrow \mathsf{SEC}^{n_2}_{n_1}\,[H_s]$ *for any $n_1, n_2 > 0$).* Given an adversary $\mathcal{A}_{H_s}$ $(\tau_{H_s}, \varepsilon_{H_s})$-breaking $\mathsf{SEC}^{n_2}_{n_1}\,[H_s]$, one builds a reduction $\mathcal{A}_F$ which (τ_F, ε_F)-breaks $\mathsf{A\text{-}SEC}^{n_2}_{n_1}\,[F]$ with $\varepsilon_F = \varepsilon_{H_s}$ and $\tau_F = \tau_{H_s}$. Given a random $M_1 \leftarrow \{0,1\}^{n_1}$, $\mathcal{A}_F$ runs $\mathcal{A}_{H_s}(M_1)$ to build an (n_1, n_2)-collision (M_1, M_2) on H and outputs M_2. Since g is collision-propagating, one has $H(M_1 \| r) = H(M_2 \| r)$ for any $r \in \{0,1\}^{\mathsf{r}} \subseteq \{0,1\}^*$, meaning that (M_1, M_2) is an (n_1, n_2)-absolute collision on F. □

Lemma 35. *For $H_s = \mathsf{ITER}\,[f, g_s, IV_0]$ as above, let now consider F defined as $F(M, r) = H_s(r \| M)$. Then for any $n = k \cdot \mathsf{b} > 0$, $\mathsf{A\text{-}SEC}^n_n\,[F]$ and $\mathsf{A\text{-}COL}^{n,n}\,[F]$ are perfectly secure unless f and IV_0 are voluntarily constructed otherwise.*

This is even true when $F(M, r) = H_0(r \| M)$. To see this, assume for instance that r is a multiple of the block size b and pose $S = \{IV \mid IV = H_0(r), r \in \{0,1\}^{\mathsf{r}}\}$. Denote $H_{0,IV} = \mathsf{ITER}\,[f, g_0, IV]$. Then finding an (n, n) or (n_1, n_2)-absolute collision for F implies the existence of M_1, M_2 such that $\forall IV \in S, H_{0,IV}(M_1) = H_{0,IV}(M_2)$. As the set S has to be large (otherwise finding collisions on $H_0 = H_{0,IV_0}$ is easy), this property is statistically unlikely for a practical hash function.

7.4 Comparing operating modes $F(M,r) = H_s(M\|r)$ and $F(M,r) = H_s(r\|M)$

It is easily seen that a probabilistic signature scheme $\mathcal{S} = \langle F, \Sigma \rangle$ based on the operating mode $F(M,r) = H_s(M\|r)$ yields in reality deterministic hash-and-sign signatures on message subspaces $M \in \{0,1\}^{k \cdot \mathsf{b}}$. Indeed posing $H_{s,IV} = \mathsf{ITER}[f, g_s, IV]$, it is obvious that if $|M| = k \cdot \mathsf{b}$ then $H_s(M\|r) = H_{s,m}(r)$ where $m = H_{0,IV_0}(M)$. So on these subspaces, $\sigma = \mathcal{S}.\mathsf{Sign}(\mathsf{sk}, M, u) = \Sigma_2(\mathsf{sk}, F(M,r), r, \mathsf{aux})$ where $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$ can be reformulated as $\sigma = \Sigma_2(\mathsf{sk}, H_{s,m}(r), r, \mathsf{aux}) = \Sigma_2'(\mathsf{sk}, m, u)$ where $m = H_0(M)$. This drives us to the following observation.

Lemma 36. *Let $\mathcal{S} = \langle F, \Sigma \rangle$. If $F(M,r) = H_s(M\|r)$ then $\mathcal{S} = \langle H_0, \Sigma' \rangle$ on message subspaces $\{0,1\}^{k \cdot \mathsf{b}}$ for $k > 0$ for some signature scheme Σ'.*

So in this case $\mathcal{S}$ can be viewed as a deterministic signature scheme. The security benefits inherent to using the probabilistic hash-and-sign paradigm are then completely lost:

Corollary 37 (Operating mode $F(M,r) = H_s(M\|r)$). *For any $n = k \cdot \mathsf{b} > 0$, all security levels EF^n-$\mathsf{CMA}[\mathcal{S}]$, UF_n-$\mathsf{CMA}[\mathcal{S}]$, EF^n-$\mathsf{KMA}[\mathcal{S}]$, $\mathsf{ER}^n[\mathcal{S}]$ and $\mathsf{UR}_n[\mathcal{S}]$ fall as soon as $\mathsf{COL}^{n,n}[H_0]$ or $\mathsf{SEC}_n^n[H_0]$ are broken (Lemma 24). In the primitive signature case, even EF^n-$\mathsf{KOA}[\mathcal{S}]$ is easy to break if one finds an efficient preimage attack against $\mathsf{PRE}^n[H_0]$.*

By opposition, the operating mode $F(M,r) = H_s(r\|M)$ provides (unless the compression function f is "unnatural") perfect security for A-$\mathsf{COL}^{n,n}[F]$ and A-$\mathsf{SEC}_n^n[F]$, thus leaving no visible breach in the security of $\mathcal{S}$:

Corollary 38 (Operating mode $F(M,r) = H_s(r\|M)$). *Combining the security reductions of Lemmas 35 and 31 and all security statements of Section 6, there is no known way to break $\mathcal{S}$ in any sense even if $\mathsf{COL}^{n,n}$, SEC_n^n and PRE^n are totally broken for H_0.*

Clearly, this strongly suggests to prefer the second operating mode over the first one in practical signature implementations.

7.5 Concrete security figures for three instantiations of $F(M,r)$

We display concrete security workloads for effectively attacking $F(m,r)$ under current knowledge for the three instantiations $F(m,r) = H_s(M||r)$, $F(m,r) = H_s(r||M)$ and $F(m,r) = H_{s,r}(M)$ and H_s =MD4, MD5, SHA-0 and SHA-1. What we plot is the expected running time τ_{H_s} of the attack algorithm when imposing a success probability ε_{H_s} heuristically close to one.

	$H_s(M\|\|r)$				$H_s(r\|\|M)$				$H_{s,r}(M)$			
	MD4	MD5	SHA-0	SHA-1	MD4	MD5	SHA-0	SHA-1	MD4	MD5	SHA-0	SHA-1
U-PRE $[F]$												
E-PRE $[F]$												
A-SEC $[F]$					∞	∞	∞	∞	∞	∞	∞	∞
U-SEC $[F]$												
E-SEC $[F]$	?	2^{52}			2^{58}				2^{58}			
A-COL $[F]$	2^1	2^{30}	2^{39}	2^{63}	∞	∞	∞	∞	∞	∞	∞	∞
U-COL $[F]$	2^1	2^{30}	2^{39}	2^{63}	2^1	2^{30}	2^{39}	2^{63}	2^1	2^{30}	2^{39}	2^{63}
E-COL $[F]$	2^1	2^{30}	2^{39}	2^{63}	2^1	2^{30}	2^{39}	2^{63}	2^1	2^{30}	2^{39}	2^{63}

Fig. 3. Security of F for common hash functions H_s and operating modes

Breaking E-$\mathsf{COL}[F]$ is just finding collisions for the Merkle-Damgård hash function H_0 in the three cases since the value of r is unconstrained. The best attacks known are Sasaki's new message difference for MD4 [35], Klima's tunnels for MD5 [23], Wang's attack on SHA-0 [40], and the latest improvements over Wang's SHA-1 attack [39]. When the input r is treated after all message blocks, these are also A-COL attacks as previously discussed. Conversely if r is processed in the beginning of the hash function, we assume that absolute collisions do not exist. Concerning preimages, there are some results that can be used. Against MD4, Yu *et al.* gave a differential path that allow to find preimages for a class of weak messages; in some cases r can be used to randomize the IV or the message until we have a weak message:

– with $F(M,r) = H_{0,r}(M)$, we just try as many IV's as needed;
– with $F(M,r) = H_{0,IV_0}(r||M)$ we can randomize the message if $58 < |r| \leq 128$ (if r is too short we will not find a weak one, and if it is too long the two messages will be the same).

The target collision attack against MD5 by Stevens, Lenstra and de Weger [37] can be used to generate solutions for $\mathsf{E\text{-}SEC}[F]$ if r is long enough (at least 4192 bits). A similar attack can certainly be done on MD4. ∞ indicates perfect security. Empty cells denote that we are not aware of an attack more efficient than generic attacks.

8 Conclusion

In this paper, we have investigated the impact of recent attacks on hash functions on signature schemes using them. We have suggested categories of signature schemes to tell them apart in terms of deterministic or probabilistic hash-and-sign mechanism, injectivity and primitiveness. For each of these categories (and notably for signature schemes such as FDH, PSS and Schnorr), we have shown how their security relates to the one of their inner hash component.

We also focused on the case of hash functions based on the Merkle-Damgård paradigm (which is the case of MDx and SHAx functions) by enlightening the security properties of popular operating modes, the ones one would try first to construct a hash function family. We have identified $F(M,r) = H_s(M\|r)$ as the less secure one by far. A concrete conclusion of our work is the suggestion of using $F(M,r) = H_s(r\|M)$ or more intricate operating modes. No security breach on probabilistic hash-and-sign signatures using F is known in this case, even if H_0 is totally broken.

We leave as an open problem to study the impact of hash functions on encryption schemes.

References

1. M. Bellare and Ph. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proc. of ACM CCS '93*, pages 62–73. ACM Press, November 1993.
2. M. Bellare and Ph. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In U. M. Maurer, editor, *Proc. of Eurocrypt '96*, volume 1070 of *LNCS*, pages 399–416. Springer-Verlag, Berlin, May 1996.
3. M. Bellare and Ph. Rogaway. PSS: Provably secure encoding method for digital signatures. Submission to IEEE P1363a, August 1998. http://grouper.ieee.org/groups/1363/.
4. E. Biham and R. Chen. Near collision for SHA-0. In M. Franklin, editor, *Proc. of Crypto '04*, volume 3152 of *LNCS*, pages 290–305. Springer-Verlag, Berlin, August 2004.
5. J. Black, Ph. Rogaway, and T. Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In M. Yung, editor, *Proc. of Crypto '02*, volume 2442 of *LNCS*, pages 320–335. Springer-Verlag, Berlin, August 2002.
6. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Proc. of Asiacrypt '01*, volume 2248 of *LNCS*, pages 514–532. Springer-Verlag, Berlin, December 2001.
7. B. den Boer and A. Bosselaers. An attacks on the last two rounds of MD4. In J. Feigenbaum, editor, *Proc. of Crypto '91*, volume 576 of *LNCS*. Springer-Verlag, Berlin, August 1991.
8. B. den Boer and A. Bosselaers. Collisions for the compression function of MD5. In T. Helleseth, editor, *Proc. of Eurocrypt '93*, volume 765 of *LNCS*, pages 293–304. Springer-Verlag, Berlin, May 1993.
9. D. R. L. Brown. Generic groups, collision resistance, and ECDSA. http://eprint.iacr.org/2002/026/, February 2002.
10. F. Chabaud and A. Joux. Attack on sha-0. email, August 1998.
11. J.-S. Coron. On the exact security of full-domain-hash. In M. Bellare, editor, *Proc. of Crypto '00*, volume 1880 of *LNCS*, pages 229–235. Springer-Verlag, Berlin, August 2000.
12. J.-S. Coron. Optimal security proofs for PSS and other signature schemes. In L. R. Knudsen, editor, *Proc. of Eurocrypt '02*, volume 2332 of *LNCS*, pages 272–287. Springer-Verlag, Berlin, April–May 2002.
13. J.-S. Coron and D. Naccache. Security analysis of the Gennaro-Halevi-Rabin signature scheme. In B. Preneel, editor, *Proc. of Eurocrypt '00*, volume 1807 of *LNCS*, pages 91–101. Springer-Verlag, Berlin, May 2000.
14. C. de Cannière and C. Rechberger. SHA-1 collisions: Partial meaningful at no extra cost? *Rump Session* of Crypto '06, August 2006.
15. C. de Cannière and C. Rechberger. Finding SHA-1 Characteristics: General Results and Applications. In *Proc. of Asiacrypt '06*, volume 4284 of *LNCS*, pages 1–20. Springer-Verlag, Berlin, December, 2006.
16. C. De Canniere, F. Mendel and C. Rechberger . A Collision for 70-step SHA-1 in a Minute. *Rump Session* of FSE '07.
17. H. Dobbertin. The status of MD5 after a recent attack. *CryptoBytes (RSA Laboratories)*, 2(2):1–6, Summer 1996.
18. A. Fiat and A. Shamir. How to prove yourself : Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Proc. of Crypto '86*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, Berlin, August 1986.

19. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signature without the random oracle. In J. Stern, editor, *Proc. of Eurocrypt '99*, volume 1592 of *LNCS*, pages 123–139. Springer-Verlag, Berlin, May 1999.
20. American National Standards Institute. Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm. ANSI X9.62-1998, January 1999.
21. P. Gauravaram, A. McCullagh and E. Dawson. Collision Attacks on MD5 and SHA-1: Is this Sword of Damocles for E-commerce? In *Proc. of AusCERT '06*, pages 73–88, May 2006.
22. J. Kelsey and B. Schneier. Second preimages on n-bit hash functions for much less than 2^n work. In R. Cramer, editor, *Proc.of Eurocrypt '05*, volume 3494 of *LNCS*, pages 474–490. Springer-Verlag, Berlin, May 2005.
23. V. Klìma. Tunnels in hash functions: MD5 collisions within a minute. http://eprint.iacr.org/2006/105/, April 2006.
24. H. Krawczyk and T. Rabin. Chameleon signatures. In *Proc. of NDSS 2000*, pages 143–154. Internet Society, February 2000.
25. G. Leurent. Message Freedom in MD4 and MD5 Collisions: Application to APOP. In A. Biryukov, editor, *Proc. of FSE '07*, LNCS. Springer-Verlag, Berlin, March 2007.
26. S. Lucks. A failure-friendly design principle for hash functions. In B. Roy, editor, *Proc. of Asiacrypt '05*, volume 3788 of *LNCS*, pages 474–494. Springer-Verlag, Berlin, December 2005.
27. J. Stern, D. Pointcheval, J. Malone-Lee and N. Smart. Flaws in Applying Proof Methodologies to Signature Schemes. In M. Yung, editor, *Proc.of Crypto '02*, volume 2442 of *LNCS*, pages 93–110. Springer-Verlag, Berlin, August 2002.
28. K. Matusiewicz, T. Peyrin, O. Billet, S. Contini and J.Pieprzyk. Cryptanalysis of FORK-256. In A. Biryukov, editor, *Proc. of FSE '07*, LNCS. Springer-Verlag, Berlin, March 2007.
29. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, October 1997. http://cacr.math.uwaterloo.ca/hac/.
30. D. Naccache, D. Pointcheval, and J. Stern. Twin signatures: an alternative to the hash-and-sign paradigm. In P. Samarati, editor, *Proc. of ACM CCS '01*, pages 20–27. ACM Press, November 2001.
31. National Institute of Standards and Technology. Digital Signature Standard (DSS). Federal Information Processing Standards — Publication 186, May 1994.
32. B. Preneel. Analysis and design of cryptographic hash functions. PhD thesis, 1993.
33. Ph. Rogaway. Formalizing human ignorance. In P. Q. Nguyen, editor, *Proc. of Vietcrypt '06*, volume 4341 of *LNCS*, pages 211–218. Springer-Verlag, Berlin, 2006.
34. Ph. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In B. K. Roy and W. Meier, editors, *Proc. of FSE '04*, volume 3017 of *LNCS*, pages 371–388. Springer-Verlag, Berlin, February 2004.
35. Y. Sasaki, L. Wang, Y. Ohta, and N. Kunihiro. New message difference for MD4. In A. Biryukov, editor, *Proc. of FSE '07*, LNCS. Springer-Verlag, Berlin, March 2007.
36. C. P. Schnorr. Efficient signatures generation by smart cards. *J. of Cryptology*, 4(3):161–174, 1991.
37. M. Stevens, A. K. Lenstra, and B. de Weger. Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities. In M. Naor, editor, *Proc. of Eurocrypt '07*, LNCS. Springer-Verlag, Berlin, May 2007.
38. X. Y. Wang, D. Feng, X. J. Lai, and H. B. Yu. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, http://eprint.iacr.org/2004/199, August 2004. Presented at the *Rump Session* of Crypto '04.
39. X. Y. Wang, Y. L. Yin, and H. B. Yu. Finding collisions in the full sha-1. In V. Shoup, editor, *Proc. of Crypto '05*, volume 3621 of *LNCS*, pages 17–36. Springer-Verlag, Berlin, August 2005.
40. X. Y. Wang, H. B. Yu, and Y. L. Yin. Efficient collision search attacks on SHA-0. In V. Shoup, editor, *Proc. of Crypto '05*, volume 3621 of *LNCS*, pages 1–16. Springer-Verlag, Berlin, August 2005.

A Short Definition of Some Classic Signature Schemes

Schnorr. This discrete log-based signature scheme was introduced in [36]. Let $\mathcal{G}$ be an abelian group of prime order q and generator g.

Key generation	uniformly select the secret key $x \leftarrow \mathbb{Z}_q$ and deduce public key $y = g^x$.
Signature	given $M \in \{0,1\}^*$,
(Σ_1)	select $k \leftarrow \mathbb{Z}_q$, set $r = g^k$ and output (r, k)
(F)	compute $m = F(M, r)$
(Σ_2)	given m and (r, k), compute $s = k + mx \bmod q$ and return $\sigma = (s, r)$.
Verification	compute $m = F(M, r)$ and check that $g^s y^{-m} = r$.

The signature scheme PrimSchnorr consists in simply defining the signature on m as $\sigma = (s, r)$ where $r = g^k$ and $s = k + mx \bmod q$ for a randomly selected $k \leftarrow \mathbb{Z}_q$. PrimSchnorr is existentially (even universally) forgeable since for any pair $(m, s) \in \mathbb{Z}_q^2$, $\sigma = (s, r)$ where $r = g^s y^{-m}$ is a valid signature on m. It is easily seen that Schnorr signatures are also injective.

Full Domain Hash (FDH). This signature scheme was proposed by Bellare and Rogaway in [2]. The scheme relies on a hash function $H : \{0,1\}^* \to \{0,1\}^{\mathsf{k}}$ and an RSA key generator $\mathsf{Gen}(1^{\mathsf{k}})$ returning random k-bit RSA keys.

Key generation	randomly select $(n, e, d) \leftarrow \mathsf{Gen}(1^{\mathsf{k}})$ and deduce public key (n, e).
Signature	given $M \in \{0,1\}^*$,
(H)	compute $m = H(M)$
(Σ)	given m, compute $s = m^d \bmod n$ and return $\sigma = s$.
Verification	compute $m = H(M)$ and check that $s^e = m \bmod n$.

FDH is obviously a deterministic hash-and-sign signature scheme. Since RSA is a trapdoor permutation, FDH is also primitive and injective.

Probabilistic Full Domain Hash (PFDH). PFDH is a probabilistic extension of FDH suggested in [12]. We make use of a family of hash functions $F : \{0,1\}^* \times \{0,1\}^{\mathsf{r}} \to \{0,1\}^{\mathsf{k}}$ and an RSA key generator $\mathsf{Gen}(1^{\mathsf{k}})$ returning random k-bit RSA keys.

Key generation	randomly select $(n, e, d) \leftarrow \mathsf{Gen}(1^{\mathsf{k}})$ and deduce public key (n, e).
Signature	given $M \in \{0,1\}^*$,
(Σ_1)	select $r \leftarrow \{0,1\}^{\mathsf{r}}$ and return $(r, \emptyset)$
(F)	compute $m = F(M, r)$
(Σ_2)	given m and $(r, \emptyset)$, compute $s = m^d \bmod n$ and return $\sigma = (s, r)$.
Verification	compute $m = F(M, r)$ and check that $s^e = m \bmod n$.

PFDH is easily seen to be a probabilistic hash-and-sign signature scheme. Given $m \in \{0,1\}^{\mathsf{k}}$, PrimPFDH simply consists in generating $\sigma = (s, r)$ with $s = m^d \bmod n$ and $r \leftarrow \{0,1\}^{\mathsf{r}}$ and is therefore existentially forgeable. It is also the case that PFDH is injective.

(Simplified) PSS. PSS was originally conceived by Bellare and Rogaway and proposed to the IEEE P1363 working group [3]. We focus on a simplified version of PSS here. Let Gen be an RSA key generator as above and $F : \{0,1\}^* \times \{0,1\}^{\mathsf{r}} \to \{0,1\}^{\mathsf{m}}$ and $G : \{0,1\}^{\mathsf{m}} \to \{0,1\}^{\mathsf{r}}$ be two functions such that $\mathsf{m} + \mathsf{r} + \mathsf{t} = \mathsf{k}$ where t is a small constant.

Key generation	randomly select $(n, e, d) \leftarrow \mathsf{Gen}(1^{\mathsf{k}})$ and deduce public key (n, e).
Signature	given $M \in \{0,1\}^*$,
(Σ_1)	select $r \leftarrow \{0,1\}^{\mathsf{r}}$ and return $(r, \emptyset)$
(F)	compute $m = F(M, r)$
(Σ_2)	given m and $(r, \emptyset)$, compute $\mathsf{pad}(m, r) = 0^{\mathsf{t}} \| m \| (r \oplus G(m))$, $s = \mathsf{pad}(m, r)^d \bmod n$ and return $\sigma = s$.
Verification	letting $\bar{m} = s^e \bmod n$, parse $\bar{m}$ as $0^{\mathsf{t}} \| m \| (r \oplus G(m))$ to recover (m, r). If this fails return 0. Otherwise check that $m = F(M, r)$.

Given $m \in \{0,1\}^{\mathsf{m}}$, a signature on m with respect to PrimPSS is $s = \mathsf{pad}(m, r)^d \bmod n$ for some randomly selected $r \leftarrow \{0,1\}^{\mathsf{r}}$. To build an existential forgery, randomly select $s \leftarrow \mathbb{Z}_n$ such that $s^e \bmod n$ can be parsed as $0^{\mathsf{t}} \| m \| \rho$. This remains polynomial in k since t is a small constant. Now $\rho = r \oplus G(m)$ for some $r \in \{0,1\}^{\mathsf{r}}$ meaning that s is a valid signature on m. It is straightforward that any value of s admits at most one pair (m, r) such that $s = \mathsf{pad}(m, r)^d \bmod n$, so PSS is injective.

EMSA-PSS. PSS was originally conceived by Bellare and Rogaway and proposed to the IEEE P1363 working group [3]. We focus on a simplified version of PSS here. Let Gen be an RSA key generator as above and $F : \{0,1\}^* \times \{0,1\}^{\mathsf{r}} \to \{0,1\}^{\mathsf{m}}$ and $G : \{0,1\}^{\mathsf{m}} \to \{0,1\}^{\mathsf{r}}$ be two functions such that $\mathsf{m} + \mathsf{r} + \mathsf{t} = \mathsf{k}$ where t is a small constant.

Key generation randomly select $(n, e, d) \leftarrow \mathsf{Gen}(1^{\mathsf{k}})$ and deduce public key (n, e).
Signature given $M \in \{0,1\}^*$,
(Σ_1) select $r \leftarrow \{0,1\}^{\mathsf{r}}$ and return $(r, \emptyset)$
(F) compute $m = F(M, r)$
(Σ_2) given m and $(r, \emptyset)$, compute $\mathsf{pad}(m, r) = 0^{\mathsf{t}} \| m \| (r \oplus G(m))$,
$s = \mathsf{pad}(m, r)^d \bmod n$ and return $\sigma = s$.
Verification letting $\bar{m} = s^e \bmod n$, parse $\bar{m}$ as $0^{\mathsf{t}} \| m \| (r \oplus G(m))$ to recover (m, r).
If this fails return 0. Otherwise check that $m = F(M, r)$.

Given $m \in \{0,1\}^{\mathsf{m}}$, a signature on m with respect to PrimPSS is $s = \mathsf{pad}(m, r)^d \bmod n$ for some randomly selected $r \leftarrow \{0,1\}^{\mathsf{r}}$. To build an existential forgery, randomly select $s \leftarrow \mathbb{Z}_n$ such that $s^e \bmod n$ can be parsed as $0^{\mathsf{t}} \| m \| \rho$. This remains polynomial in k since t is a small constant. Now $\rho = r \oplus G(m)$ for some $r \in \{0,1\}^{\mathsf{r}}$ meaning that s is a valid signature on m. It is straightforward that any value of s admits at most one pair (m, r) such that $s = \mathsf{pad}(m, r)^d \bmod n$, so PSS is injective.

Boneh-Lynn-Shacham (BLS). The concept of this scheme was instantiated in [6] using bilinear maps. Let $\mathcal{G}$ be a pairing-friendly group of prime order q and generator g, and let $\langle \cdot, \cdot \rangle : \mathcal{G}^2 \to \mathcal{G}_t$ denote a cryptographic bilinear map over $\mathcal{G}$. The scheme relies on a hash function $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ and a reversible function $T : \{0,1\}^{\mathsf{m}} \to \mathcal{G}$ mapping m-bit strings to group elements.

Key generation uniformly select the secret key $x \leftarrow \mathbb{Z}_q$ and deduce public key $y = g^x$.
Signature given $M \in \{0,1\}^*$,
(H) compute $m = H(M)$
(Σ) given m, compute $s = T(m)^x$ and return $\sigma = s$.
Verification compute $m = H(M)$ and check that $\langle s, g \rangle = \langle T(m), y \rangle$.

A signature on m with respect to PrimBLS is $s = T(m)^x$. Therefore one can raise $y = g^x$ to a random power r until $g^r = T(m)$ can be reversed to yield $m \in \{0,1\}^{\mathsf{m}}$. Then $s = y^r = T(m)^x$ is a proper signature meaning that PrimBLS is existentially forgeable. Since T is reversible, BLS signatures are injective.

Generic DSA. This case captures DSA [31], ECDSA [20] and numerous variants. Let $\mathcal{G}$ be a finite abelian group of prime order q and generator g. Let H be a hash function and $T : \mathcal{G} \to \mathbb{Z}_q$ be an arbitrary function.

Key generation uniformly select the secret key $x \leftarrow \mathbb{Z}_q$ and deduce public key $y = g^x$.
Signature given $M \in \{0,1\}^*$,
(H) compute $m = H(M)$
(Σ) given m, randomly select $k \leftarrow \mathbb{Z}_q$, set $r = T(g^k)$,
compute $s = (m + rx)k^{-1} \bmod q$ and return $\sigma = (r, s)$.
Verification compute $m = H(M)$ and check that $T(g^{\frac{m}{s}} y^{\frac{r}{s}}) = r$.

Generic DSA signatures are deterministic hash-and-sign signatures. It is easily seen that they are also injective. However, they do not seem to be primitive in general.

Gennaro-Halevi-Rabin (GHR). GHR was suggested in [19]. Its security is based on the strong RSA assumption and the existence of division-intractable hash functions. We refer the reader to [19] for security proofs and definitions in further detail. Following the ideas of [13], a simple way to realize this property consists in mapping bitstrings to prime numbers in a collision-resistant fashion. Let $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}} \cap \mathsf{Primes}$ be such a function where $\mathsf{m} + \mathsf{t} = \mathsf{k}$ and $\mathsf{t} \geq 2$ is a small constant.

Key generation randomly select a k-bit safe RSA modulus $n = (2p' + 1)(2q' + 1)$ as well as $u \leftarrow \mathbb{Z}_n^*$ of maximal order $2p'q'$. Deduce public key (n, u).
Signature given $M \in \{0,1\}^*$,
(H) compute $m = H(M)$
(Σ) given m, compute $s = u^{m^{-1} \bmod 2p'q'} \bmod n$ and define the signature as $\sigma = s$.
Verification compute $m = H(M)$ and check that $s^m = u \bmod n$.

GHR is a deterministic hash-and-sign scheme and is also injective since for any given $s \in \mathbb{Z}_n$, there must exist at most one $m \in \{0,1\}^{\mathsf{m}} \cap \mathsf{Primes}$ such that $s^m = u \bmod n$. Indeed, if $s^{m_1} = s^{m_2} = u \bmod n$ with $m_1 < m_2$ then $2p'q'$ must divide $m_2 - m_1 < 2^{\mathsf{k}-\mathsf{t}}$. This implies $2p'q' < 2^{\mathsf{k}-\mathsf{t}}$ which contradicts $\mathsf{t} \geq 2$. However, GHR is not primitive.

Tight-GHR signature scheme. The security reduction of the above scheme to the flexible RSA problem is quite bad [19, 11], and so the GHR signature scheme has next been modified by using a chameleon hash function [24], thus obtaining a tight scheme.

More precisely, an attacker against the obtained Tight-GHR can be used to solved either the discrete logarithm or the flexible RSA problems, with roughly the same success probability and running time. Let P and Q be large prime numbers so that Q divides $P-1$, and let $\langle g \rangle$ denote the cyclic subgroup generated by an element $g \in \mathbb{Z}_p^*$ of order Q. We then define $H : \langle g \rangle \to \{0,1\}^{\ell_h}$.

Key generation randomly compute a safe RSA modulus $n = (2p'+1)(2q'+1)$, a random element u in $\mathbb{Z}_n^*$ and a random element y in $\langle g \rangle \subseteq \mathbb{Z}_P^*$. Publish (n, u, g, y, P).

Signature given $M \leftarrow \mathbb{Z}_q$,
select $r \leftarrow \mathbb{Z}_q$, compute $m = H(g^M y^r \bmod P)$,
compute $s = u^{m^{-1} \bmod 2p'q'}$ and define the signature as $\sigma = (r, s)$.

Verification compute $m = H(g^M y^r \bmod P)$ and check that $s^m \equiv u \bmod n$.

Twin-GHR signature scheme. The twinning paradigm was introduced by Naccache, Pointcheval and Stern in [30]. It consists in signing two related messages with a potentially weaker signature scheme. It particulary fits to GHR signatures. The security of the resulting signature scheme can be reduced to the flexible RSA problem. Let $\mathtt{P}$ be an injective function that maps the set $\{0,1\}^{2\ell_M}$ into the prime numbers.

Key generation picks two safe RSA moduli $n = (2p'+1)(2q'+1)$ and $N = (2P'+1)(2Q'+1)$, select two random elements $u_1 \in \mathbb{Z}_n^*$ and $u_2 \in \mathbb{Z}_N^*$. Publish (n, N, u_1, u_2).

Signature given $M \in \{0,1\}^{\ell_M}$,
select $\mu_1 \leftarrow \{0,1\}^{2\ell_M}$ at random, compute $\mu_2 = (M \| M) \oplus \mu_1$,
compute $s_1 = {u_1}^{\mathtt{P}(\mu_1)^{-1} \bmod 2p'q'} \bmod n$, compute $s_2 = {u_2}^{\mathtt{P}(\mu_2)^{-1} \bmod 2P'Q'} \bmod N$,
define $\sigma = (\mu_1, s_1, s_2)$ as the signature on M.

Verification check that $s_1^{\mathtt{P}(\mu_1)} = u_1 \bmod n$ and $s_2^{\mathtt{P}((M\|M)\oplus\mu_1)} = u_2 \bmod N$.

Cramer-Shoup signature scheme. The Cramer-Shoup signature scheme (CS) is another hash-and-sign signature scheme secure under the strong RSA assumption. One of the advantages of the Cramer-Shoup scheme, compared to the GHR scheme, is that the hash function only needs to be collision-resistant. Its disadvantage is that the reduction is loose. Let $H : \{0,1\}^* \to \{0,1\}^{\ell_h}$ be a collision-resistant hash function.

Key generation randomly compute a safe RSA modulus $n = (2p'+1)(2q'+1)$, e an (ℓ_h+1)-bit prime and x, h two random elements in $QR(n)$. Publish (n, e, x, h).

Signature given $M \in \{0,1\}^{\ell_M}$,
(H) compute $m = H(M)$
(Σ) given m, choose c as a random (ℓ_h+1)-bit prime, choose u at random in $QR(n)$,
compute $w = u^e h^{-m} \bmod n$, compute $v = (x\, h^{H(w)})^{c^{-1} \bmod p'q'} \bmod n$,
define $\sigma = (c, u, v)$ as the signature on M.

Verification check that c is an odd (ℓ_h+1)-bit integer and $v^c\, h^{-H(w')} \equiv x \bmod n$
with $w' = u^e\, h^{-H(m)} \bmod n$.