How to use Merkle-Damgård — On the Security Relations between Signature Schemes and their Inner Hash Functions

Emmanuel Bresson[1], Benoît Chevallier-Mames[1], Christophe Clavier[2], Aline Gouget[3], Pascal Paillier[3], and Thomas Peyrin[4]

[1] DCSSI, 51 bd. De la Tour Maubourg, 75700 Paris Cedex 07, France.
[2] Gemalto, 6 rue de la Verrerie, 92190 Meudon, France.
[3] Gemalto, Avenue du Jujubier, 13705 La Ciotat, France.
[4] Orange Labs, 38-40 rue du Général Leclerc, 92794 Issy-les-Moulineaux, France.

Abstract. This paper reports a thorough standard-model investigation on how attacks on hash functions impact the security of *hash-and-sign* signature schemes. We identify two important properties that appear to be crucial in analyzing the nature of security relations between signature schemes and their inner hash functions: primitiveness and injectivity. We then investigate the security relations in the general case of *hash-and-sign* signatures and in the particular case of *first-hash-then-sign* signatures, showing a gap of security guarantees between the two paradigms. We subsequently apply our results on two operating modes to construct a hash function family from a hash function based on the well-known *Merkle-Damgård* construction (such as MD5 and SHA-1). For completeness, we give concrete attack workloads for attacking operating modes used in practical implementations of signature schemes.

1 Introduction

Undoubtedly, hash functions constitute an essential component of all sorts of systems and constructions in cryptography. Recently, a number of attacks on practical hash functions such as MD5 or SHA-1 have been reported [4,24,27]. 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 or weakened hash functions.

The goal of our work is to explore the interplay between 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. These connections between $\mathcal{S}$ and H can be categorized into four types: *attack*, *security proof*, *impossibility of attack* and *impossibility of security proof*. In this paper, we focus on the first two links.

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

In Section 2, we give several definitions on provable security and on hash function and signature security notions. In Section 3, we analyze the security link between signature schemes and hash functions, for two types of signature schemes we define. In Section 4, we refine our study on the special case of Merkle-Damgård construction. As expressed in the conclusion, the security relations we expose in this paper confirm the intuition that all signature schemes are *not* implicated on the same level by the recent attacks on hash functions.

2 Preliminaries

2.1 Provable Security Statements

We adopt the concrete-security setting [20], as opposed to the asymptotic one. A concrete black-box reduction $\mathcal{R}$ between two computational problems P_1 and P_2 is a probabilistic algorithm $\mathcal{R}$ which solves P_1 given black-box access to an oracle which solves P_2. We write $P_1 \Leftarrow_{\mathcal{R}} P_2$ when $\mathcal{R}$ is known to reduce P_1 to P_2 with efficient (or tight) reduction cost. $P_1 \Leftarrow P_2$ states that an *efficient* $\mathcal{R}$ exists such that $P_1 \Leftarrow_{\mathcal{R}} P_2$ and $P_1 \Leftrightarrow P_2$ means $P_1 \Leftarrow P_2$ and $P_1 \Rightarrow P_2$. All reductions considered in this paper are concrete and fully black-box.

A provable-security statement $P_1 \Leftarrow P_2$, where P_1, P_2 are two computational problems, indicates that if an efficient solver for P_2 exists then an efficient solver for P_1 exists as well, which is meaningful when P_1 is believed to be hard. When efficient solvers for P_1 are known to exist, by contrast, we claim that $P_1 \Leftarrow P_2$ is still meaningful if *(a)* an efficient solver for P_1 exists but is *unknown*[5], *(b)* the proof of $P_1 \Leftarrow P_2$ is *constructive* meaning that an explicit black-box reduction $\mathcal{R}$ is given.

2.2 Hash Functions and Related Security Notions

In the following, $\{0,1\}^*$ denotes the set of finite bitstrings, and $\{0,1\}^{\mathsf{m}}$ denotes the set of m-bit strings. A function H is a *hash function* if it maps $\{0,1\}^*$ to $\{0,1\}^{\mathsf{m}}$ for some integer $\mathsf{m} > 0$ called the output size of H. For $n > 0$, let U_n be the uniform distribution over $\{0,1\}^n$ and for any $0 \le t \le 2^n$, let us pose $S_{n,t} = \{m \in \{0,1\}^{\mathsf{m}} : |H^{-1}(m) \cap \{0,1\}^n| = t\}$. We define the bias of H, denoted by $\delta_n(H)$, as the statistical L_1-distance between $H(U_n)$ and U_{m}, i.e. $\delta_n(H) = \|H(U_n) - U_{\mathsf{m}}\|_1 = \sum_{t=0}^{2^n} |S_{n,t}| \left|\frac{t}{2^n} - \frac{1}{2^{\mathsf{m}}}\right|$.

A *hash function family* 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.

[5] This relates to Rogaway's human ignorance paradigm, see [20].

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

Let $n > 0$, we define two statistical bias for F: $\delta_n(F) = \|F(U_n, U_\mathsf{r}) - U_\mathsf{m}\|_1$ and $\delta_n^+(F) = \max_{r\in\{0,1\}^\mathsf{r}} \|F(U_n, r) - U_\mathsf{m}\|_1$ (hence $\delta_n(F) \leq \delta_n^+(F)$).

We adopt the convention that security problems are parameterized by the bitsize of their inputs (written as subscripts) and/or their outputs (written as superscripts). The classical security notions for hash function are preimage resistance (PRE), 2nd-preimage resistance (SEC) and Collision resistance (COL).

	Given	Find	Such that
$\mathsf{PRE}^n[H]$	m	M	$H(M) = m$
$\mathsf{SEC}_{n_1}^{n_2}[H]$	M_1	$M_2 (\neq M_1)$	$H(M_2) = H(M_1)$
$\mathsf{COL}^{n_1,n_2}[H]$	$-$	M_1, M_2	$M_1 \neq M_2$ and $H(M_1) = H(M_2)$
where $m \in \{0,1\}^\mathsf{m}, M \in \{0,1\}^n, M_i \in \{0,1\}^{n_i}$			

When extending the security notions PRE, SEC and COL to hash function families, three definitional variants appear which we call existential (E-$\cdots$), universal (U-$\cdots$) and absolute (A-$\cdots$), as indicated below.

	Given	Find	Such that
$\mathsf{E\text{-}PRE}^n[F]$	m	M, r	$F(M,r) = m$
$\mathsf{U\text{-}PRE}^n[F]$	m, r	M	$F(M,r) = m$
$\mathsf{E\text{-}SEC}_{n_1}^{n_2}[F]$	M_1	M_2, r	$F(M_2,r) = F(M_1,r)$
$\mathsf{U\text{-}SEC}_{n_1}^{n_2}[F]$	M_1, r	M_2	$F(M_2,r) = F(M_1,r)$
$\mathsf{A\text{-}SEC}_{n_1}^{n_2}[F]$	M_1	M_2	$\forall r, F(M_2,r) = F(M_1,r)$
$\mathsf{E\text{-}COL}^{n_1,n_2}[F]$	$-$	M_1, M_2, r	$F(M_2,r) = F(M_1,r)$
$\mathsf{U\text{-}COL}^{n_1,n_2}[F]$	r	M_1, M_2	$F(M_2,r) = F(M_1,r)$
$\mathsf{A\text{-}COL}^{n_1,n_2}[F]$	$-$	M_1, M_2	$\forall r, F(M_2,r) = F(M_1,r)$
where $m \in \{0,1\}^\mathsf{m}, M \in \{0,1\}^n, M_i \in \{0,1\}^{n_i}, r \in \{0,1\}^\mathsf{r}, M_1 \neq M_2$			

The relations between the security notions for a hash function (resp. a hash function family) are summarized in Theorem 1.

Theorem 1. *Let H be a hash function. Then, for any $n_1, n_2 > 0$, we have:*

$$\mathsf{COL}^{n_1,n_2}[H] \Leftarrow \mathsf{SEC}_{n_1}^{n_2}[H] \Leftarrow^\star \mathsf{PRE}^{n_2}[H] \ ,$$

where the $\Leftarrow^\star$ reduction is tight only if H has sufficiently small bias[6] $\delta_{n_1}(H)$. *Let F be a hash function family. Then, for any $n_1, n_2 > 0$, we have:*

$$\begin{array}{ccccc}
\mathsf{E\text{-}PRE}^{n_2}[F] & \Leftarrow & \mathsf{U\text{-}PRE}^{n_2}[F] & & \\
 & & \Downarrow {}^\star\textit{if } \delta_{n_1}^+(F) \textit{ is small enough} & & \\
\mathsf{E\text{-}SEC}_{n_1}^{n_2}[F] & \Leftarrow & \mathsf{U\text{-}SEC}_{n_1}^{n_2}[F] & \Leftarrow & \mathsf{A\text{-}SEC}_{n_1}^{n_2}[F] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\mathsf{E\text{-}COL}^{n_1,n_2}[F] & \Leftarrow & \mathsf{U\text{-}COL}^{n_1,n_2}[F] & \Leftarrow & \mathsf{A\text{-}COL}^{n_1,n_2}[F]
\end{array}$$

[6] Well-known counterexamples can be constructed when this condition is not fulfilled, see [17].

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

Most of the relations given in Theorem 1 are rather intuitive and can be proved using very standard reduction techniques.

2.3 Signature Schemes and Related Security Notions

A signature scheme Σ with message space $\mathcal{M} \subseteq \{0,1\}^*$ is defined as $\Sigma \triangleq (\Sigma.\mathsf{Gen}, \Sigma.\mathsf{Sign}, \Sigma.\mathsf{Ver})$ such that $\Sigma.\mathsf{Gen}()$ is a probabilistic algorithm that outputs a pair of strings $(\mathsf{pk}, \mathsf{sk})$, a signature on message $M \in \mathcal{M}$ is an s-bit string $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, M, u)$ where $u \leftarrow \{0,1\}^{\mathsf{u}}$, and $\Sigma.\mathsf{Ver}(\mathsf{pk}, M, \sigma)$ outputs 1 if $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, M, u)$ for some $u \in \{0,1\}^{\mathsf{u}}$, 0 otherwise.

We now take a closer look at the inner computations of a signature scheme by defining the notion of *two-step signature* which seems totally unrestrictive as we do not know any example of a *non*-two-step signature scheme.

Definition 2 (Two-step signature scheme). *A signature scheme $\Sigma \triangleq (\Sigma.\mathsf{Gen}, \Sigma.\mathsf{Sign}, \Sigma.\mathsf{Ver})$ with message space $\mathcal{M} \subseteq \{0,1\}^*$ is said to be* two-step *if there exist four deterministic algorithms Σ_1, Σ_2, Υ_1 and Υ_2 such that:*

(i) *For any pair $(M, u) \in \mathcal{M} \times \{0,1\}^{\mathsf{u}}$, $\Sigma.\mathsf{Sign}(\mathsf{sk}, M, u) = \Sigma_2(\mathsf{sk}, M, r, \mathsf{aux})$ where $(r, \mathsf{aux}) = \Sigma_1(\mathsf{sk}, u)$, $\mathsf{aux} \in \{0,1\}^*$ and $r \in \{0,1\}^{\mathsf{r}}$.*
(ii) *$\Sigma.\mathsf{Ver}(\mathsf{pk}, M, \sigma) = \Upsilon_2(\mathsf{pk}, M, \sigma, \hat{r})$ where $\hat{r} = \Upsilon_1(\mathsf{pk}, \sigma)$;*
(iii) *If there exists $u \in \{0,1\}^{\mathsf{u}}$ such that $\sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, M, u)$ then $r = \hat{r}$.*

The type *of Σ is $(\mathsf{m}, \mathsf{u}, \mathsf{r}, \mathsf{s})$ if $\mathcal{M} = \{0,1\}^{\mathsf{m}}$ and $(*, \mathsf{u}, \mathsf{r}, \mathsf{s})$ if $\mathcal{M} = \{0,1\}^*$.*

We now properly define the notion of *hash-and-sign* signature that captures virtually all known signature schemes.

Definition 3 (Hash-and-Sign). *A* hash-and-sign signature scheme *is a pair $\mathcal{S} = \langle F, \Sigma \rangle$ where $F : \{0,1\}^* \times \{0,1\}^{\mathsf{r}} \rightarrow \{0,1\}^{\mathsf{m}}$ is a hash function family 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 $\Upsilon_2(\mathsf{pk}, \hat{m}, \sigma, \hat{r})$.*

We finally consider the notion of *first-hash-then-sign* signatures as a particular case of hash-and-sign signatures. Both paradigms are depicted on Fig. 1. A first-hash-then-sign signature is defined to be a hash-and-sign signature where the index space of F is not used *i.e.* $F(M, r) = H(M)$. Hence the hash function family collapses to a single hash function. The notion of first-hash-then-sign signature captures several well-known signature designs including Full Domain hash (FDH) and Boneh-Lynn-Shacham (BLS).

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

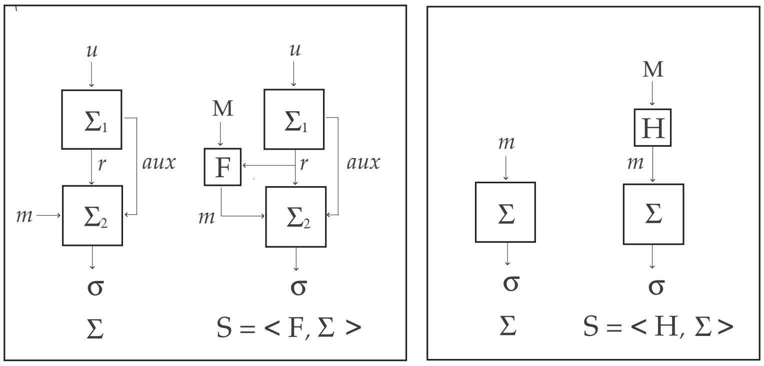

Fig. 1. The hash-and-sign (left) and first-hash-then-sign (right) paradigms.

We extend the common security notions of unforgeability, elaborating on the seminal work of Goldwasser *et al.* [10]. We consider the three usual adversarial resources: *Key Only Attack* (KOA) where the adversary is given nothing but a public key, *Known Message Attack* (KMA), where the adversary is given a list of random and pairwise distinct message-signature pairs $(M_1, \sigma_1), \ldots, (M_\ell, \sigma_\ell)$, and *Chosen Message Attack* (CMA) where the adversary is given adaptive access to a signing oracle. KMA_n denotes the restriction of KMA to messages $M_1, \ldots, M_\ell$ uniformly distributed over $\{0,1\}^n$ (we define CMA_n similarly). We distinguish four adversarial goals: *Universal Forgery* (UF), *Existential Forgery* (EF), *Universal Repudiation* (UR) and lastly *Existential Repudiation* (ER) defined as indicated below.

	Given	Find	Such that
UF UF_n	$M \in \mathcal{M}$ $M \in \{0,1\}^n$	σ σ	σ is a valid signature on the message M
EF EF_n	– –	$M \in \mathcal{M}, \sigma$ $M \in \{0,1\}^n, \sigma$	σ is a valid signature on the message M
UR $\mathsf{UR}_{n_1}^{n_2}$	$M_1 \in \mathcal{M}$ $M_1 \in \{0,1\}^{n_1}$	$M_2 \in \mathcal{M}, \sigma$ $M_2 \in \{0,1\}^{n_2}, \sigma$	$M_2 \neq M_1$ and σ is valid on both M_1 and M_2
ER ER^{n_1,n_2}	– –	$M_1 \in \mathcal{M}, M_2 \in \mathcal{M}, \sigma$ $M_1 \in \{0,1\}^{n_1}, M_2 \in \{0,1\}^{n_2}, \sigma$	$M_2 \neq M_1$ and σ is valid on both M_1 and M_2
where $n, n_1, n_2 > 0$			

We view security notions as computational problems; this allows to relate these notions using reductions. For instance $\mathsf{UF}\text{-}\mathsf{KMA}\,[\mathcal{S}]$ is the problem of computing a universal forgery under known message attack. In the case of KMA or CMA, we denote by ℓ-$\mathsf{GOAL}\text{-}\mathsf{ATK}\,[\mathcal{S}]$ the problem of breaking GOAL in no more than ℓ calls to the resource defined by ATK. In the case of UR and ER, no adversarial resource is defined, as the adversary can choose the secret/public key pair. 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 $\mathcal{M} = \{0,1\}^{\mathsf{m}}$.

$$\boxed{\begin{array}{c} \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}], \mathsf{UR}^{n_1}_{n_2}[\mathcal{S}] \\ \Downarrow \\ \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \end{array}} \quad \boxed{\begin{array}{ccccc} \mathsf{UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}] & \Leftarrow & \mathsf{UF}_{n_1}\text{-}\mathsf{KMA}_{n_2}[\mathcal{S}] & \Leftarrow & \mathsf{UF}_{n_1}\text{-}\mathsf{KOA}[\mathcal{S}] \\ \Downarrow & & \Downarrow & & \Downarrow \\ \mathsf{EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}] & \Leftarrow & \mathsf{EF}^{n_1}\text{-}\mathsf{KMA}_{n_2}[\mathcal{S}] & \Leftarrow & \mathsf{EF}^{n_1}\text{-}\mathsf{KOA}[\mathcal{S}] \end{array}}$$

Fig. 2. Black-box relations among security notions for signature schemes.

3 Analyzing Security Relations for Hash-and-Sign Signatures

In this section, we first identify two core properties that will be crucial in analyzing security relations between a signature scheme and its inner hash function. We then explicit how some attacks on hash functions can be turned into attacks on signature schemes, and on the positive side, how some attacks on signatures are impossible unless some attacks on hash functions can be made efficient.

3.1 Identified properties

It appears that most signature schemes used in practice, beside being hash-and-sign schemes, also fulfill two other properties that we call *primitiveness* and *injectivity*.

Definition 4 (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 primitive when a probabilistic algorithm $\mathcal{S}.\mathsf{Prim}$ is known which, for any key pair $(\mathsf{pk}, \mathsf{sk})$, takes pk as input and outputs a random pair $(m, \sigma = \Sigma.\mathsf{Sign}(\mathsf{sk}, m, u))$ such that (a) m is uniformly distributed over $\{0,1\}^{\mathsf{m}}$; (b) u is uniformly distributed over $\{0,1\}^{\mathsf{u}}$.*

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

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

EF-KOA security of $\mathcal{S}$. Furthermore, primitiveness is often the property that allows simulatability and makes security proofs possible in the random oracle model.

Definition 5 (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.*

We review and characterize some of the most common signature schemes. Our classification results are summarized on Table 1.

Hash-and-Sign Signature Scheme	First-Hash-then-Sign	Primitive	Injective
Schnorr [22]		X	X
FDH [2]	X	X	X
PFDH [6]		X	X
PSS [3]		X	X
EMSA-PSS [3]	X	X	X
BLS [5]	X	X	X
Generic DSA [1,19]	X		X
ElGamal Type I [17]	X	X	X
ElGamal Type II [17]		X	X
GQ [11]		X	X
GHR [9]	X		X
CS [7]	X		

Table 1. A classification of common signature schemes.

3.2 Attacks and Positive Security Relations for Hash-and-Sign Signatures

Our goal is to relate the security of $\mathcal{S} = \langle F, \Sigma \rangle$ to the one of its hash component F, the underlying signature scheme Σ being seen as a fixed parameter. The first type of connection between security notions for $\mathcal{S}$ with security notions for H is referred to as an *attack*. An attack is defined as an efficient reduction $\mathsf{Break}(H) \Rightarrow \mathsf{Break}(\mathcal{S})$ that makes explicit how an attack of a given type on a hash function H is enough to break the scheme $\mathcal{S}$ in a prescribed way. The attacks we found are summarized in Lemma 6.

Lemma 6 (Black-box attacks). *Let $\mathcal{S} = \langle F, \Sigma \rangle$ be a hash-and-sign signature scheme as per Definition 3. Then for any integers $n_1, n_2 > 0$,*

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

$$\mathsf{A\text{-}COL}^{n_1,n_2}[F] \Rightarrow \mathsf{1\text{-}EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}], \mathsf{1\text{-}EF}^{n_2}\text{-}\mathsf{CMA}[\mathcal{S}] \ , \quad (1)$$
$$\mathsf{U\text{-}COL}^{n_1,n_2}[F] \Rightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \ , \quad (2)$$
$$\mathsf{A\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow \mathsf{1\text{-}UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}] \ , \quad (3)$$
$$\mathsf{U\text{-}SEC}^{n_2}_{n_1}[F] \Rightarrow \mathsf{1\text{-}EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}], \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}] \ . \quad (4)$$

Lemma 7 (The case of primitive signatures). *Let $\mathcal{S} = \langle F, \Sigma \rangle$ be a hash-and-sign signature scheme and assume that $\mathcal{S}$ is primitive. Then in addition to the above we have, for any $n > 0$,*

$$\mathsf{U\text{-}PRE}^{n}[F] \Rightarrow \mathsf{EF}^{n}\text{-}\mathsf{KOA}[\mathcal{S}] \ .$$

We now state positive security relations between $\mathcal{S}$ and F. A *security proof* is defined as an efficient reduction $\mathsf{Break}(H) \Leftarrow \mathsf{Break}(\mathcal{S})$ which means that there is no attack of a certain type on $\mathcal{S}$ unless one finds a particular weakness in H. It seems unlikely (although we do not disprove it) that security proofs exist which relate the unforgeability of $\mathcal{S}$ to F in the general case. However, assuming that $\mathcal{S}$ is injective is enough to show that non-repudiation can be guaranteed under security assumptions on F.

Lemma 8 (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 $n_1, n_2 > 0$,*

$$\mathsf{E\text{-}COL}^{n_1,n_2}[F] \Leftarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \quad \textit{and} \quad \mathsf{E\text{-}SEC}^{n_2}_{n_1}[F] \Leftarrow \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}] \ .$$

The proofs of Lemmas 7 and 8 for hash-and-sign signature schemes can be found in Appendix A.

3.3 Attacks and Security Proof for First-Hash-then-Sign Signatures

We now move on to the particular case of first-hash-then-sign signatures. Our goal is now to exhaust the security reductions standing between a first-hash-then-sign scheme $\mathcal{S} = \langle H, \Sigma \rangle$ and its inner hash function H. Again, the signature scheme Σ plays the role of a parameter here.

The reductions stated in Lemmas 6 and 7, which show how breaking certain security properties of F allows to break $\mathcal{S}$, can be simplified in the particular case of a first-hash-then-sign scheme $\mathcal{S} = \langle H, \Sigma \rangle$. In this case indeed, in addition to the above, it holds that for any integers $n_1, n_2 > 0$:

$$\mathsf{COL}^{n_1,n_2}[H] \Rightarrow \mathsf{1\text{-}EF}^{n_1}\text{-}\mathsf{CMA}[\mathcal{S}], \mathsf{1\text{-}EF}^{n_2}\text{-}\mathsf{CMA}[\mathcal{S}], \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \quad (5)$$
$$\mathsf{SEC}^{n_2}_{n_1}[H] \Rightarrow \mathsf{1\text{-}UF}_{n_1}\text{-}\mathsf{CMA}[\mathcal{S}], \mathsf{1\text{-}EF}^{n_2}\text{-}\mathsf{KMA}_{n_1}[\mathcal{S}], \mathsf{UR}^{n_2}_{n_1}[\mathcal{S}] \quad (6)$$

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

In the same way, the positive security relations between $\mathcal{S}$ and F presented in Lemma 8 can be adapted to the particular case of first-hash-then-sign signatures. A direct corollary of these results and the relations of (5) and (6) is a security proof for injective signatures meaning that the levels of repudiation of $\mathcal{S}$ can be guaranteed under security assumptions on H.

Lemma 9 (Security proofs for injective signatures). *Let $\mathcal{S}$ be a first-hash-then-sign signature scheme and assume $\mathcal{S}$ is injective. Then for any $n_1, n_2 > 0$,*

$$\mathsf{SEC}_{n_1}^{n_2}[H] \Leftrightarrow \mathsf{UR}_{n_1}^{n_2}[\mathcal{S}] \quad \textit{and} \quad \mathsf{COL}^{n_1,n_2}[H] \Leftrightarrow \mathsf{ER}^{n_1,n_2}[\mathcal{S}] \ .$$

In some sense, the general hash-and-sign paradigm inherently offers better security guarantees than the particular case of first-hash-then-sign signatures. For instance, breaking $\mathcal{S}$ in the $\mathsf{EF\text{-}CMA}$ sense is easy if H is not collision-resistant in the 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 weakness. Overall, given a fixed-size signature scheme Σ, it seems largely preferable to domain-extend it in the general hash-and-sign paradigm for which security is obtained under much weaker assumptions on the security of the inner hash component than in the first-hash-then-sign paradigm.

4 Merkle-Damgård-based Hash Function Families

This section specifically considers Merkle-Damgård (MD) embodiments of F; this is motivated by the fact that most practical hash functions rely on some variant of the MD construction. We then apply the results presented in Section 3 to analyze two practical instantiations of F using a MD-hash function. Based on these reductions, we display the effective workload of the best known attacks against F in every instantiation.

4.1 Hash Function Families based on Merkle-Damgård

Before considering Merkle-Damgård hash functions, we first relate the security of a hash function family F, constructed from an arbitrary hash function H to the one of H. There are many ways one may attempt to construct a hash function family F using a hash function H. The two most "natural" constructions are $F(M, r) = H(M \| r)$ and $F(M, r) = H(r \| M)$.

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

Construction of a Hash Function Family. Let $H : \{0,1\}^* \to \{0,1\}^{\mathsf{m}}$ be a hash function. We construct a wide class of hash function families as follows. Let F be the hash function family defined on $M \in \{0,1\}^*$ and $r \in \{0,1\}^{\mathsf{r}}$ (the index bitsize r being arbitrary) as $F(M,r) = H([\![M,r]\!])$, where $[\![M,r]\!]$ is an $(|M| + \mathsf{r})$-bit (one-to-one) encoding of the pair $(M,r) \in \{0,1\}^* \times \{0,1\}^{\mathsf{r}}$. Then, for any $n_1, n_2 > 0$, we have :

$$\mathsf{PRE}^{n_2+\mathsf{r}}[H] \quad \Leftrightarrow \quad \mathsf{E\text{-}PRE}^{n_2}[F]$$
$$\mathsf{SEC}^{n_2+\mathsf{r}}_{n_1+\mathsf{r}}[H] \quad \Leftarrow \quad \mathsf{U\text{-}SEC}^{n_2}_{n_1}[F]$$
$$\mathsf{COL}^{n_1+\mathsf{r},n_2+\mathsf{r}}[H] \Leftarrow \mathsf{E\text{-}COL}^{n_1,n_2}[F]$$

Iterative Hash Functions. Most hash functions used in practice are *iterative hash functions* built from *compression functions*. 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 $g(M) = X_1|| \dots ||X_t$, $h_0 = IV_0$ and $h_i = f(h_{i-1}, X_i)$ for $i \in [1..t]$. We call b the block size of H and we write $H = \mathsf{ITER}[f, g, IV_0]$.

MD-based Hash Functions. The basic and strengthened Merkle-Damgård constructions [8,18] 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 $\mathsf{a} < \mathsf{b}$ 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.

Appending a logical length-block prior to hashing, called MD-*strengthening*, aims at preventing collision and pseudo-collision attacks which find colliding messages of different length (e.g. see [12,14]). 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 $H_s = \mathsf{ITER}[f, g_s, IV_0]$.

Definition 10 (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 $H(g(M_1)\|M) = H(g(M_2)\|M)$ for any $M \in \{0,1\}^*$.*

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

Proposition 11. *It holds that g_0 is a (k_1, k_2)-collision-propagating padding for any $k_1, k_2 > 0$ and 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 fact, all known second preimage or collision-finding attacks against MD-hash functions with MD-strengthening generate messages with the same block length. It turns out that for any $n = k \cdot \mathsf{b} > 0$, we have $\mathsf{PRE}^{n+b}[H_0] \Leftarrow \mathsf{PRE}^n[H_s]$, $\mathsf{SEC}_n^n[H_s] \Leftarrow \mathsf{SEC}_n^n[H_0]$ and $\mathsf{COL}^{n,n}[H_s] \Leftarrow \mathsf{COL}^{n,n}[H_0]$. We 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 simplified setting $H_0 = \mathsf{ITER}[f, g_0, IV_0]$.

4.2 MD instantiation with operating mode $F(M, r) = H_s(M \| r)$

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 $n = k \cdot \mathsf{b} > 0$,

$$\begin{array}{ccccccc} \mathsf{A\text{-}SEC}_{n+\mathsf{b}}^{n+\mathsf{b}}[F] & \Leftarrow & \mathsf{A\text{-}SEC}_n^n[F] & \Leftarrow & \mathsf{SEC}_n^n[H_0] & \Rightarrow & \mathsf{SEC}_n^n[H_s] \\ \Downarrow & & & & \Downarrow & & \\ \mathsf{A\text{-}COL}^{n+\mathsf{b},n+\mathsf{b}}[F] & \Leftarrow & \mathsf{COL}^{n,n}[H_s] & \Leftarrow & \mathsf{COL}^{n,n}[H_0] & & \end{array}$$

It is easily seen that a hash-and-sign signature scheme $\mathcal{S} = \langle F, \Sigma \rangle$ based on the operating mode $F(M, r) = H_s(M\|r)$ yields in reality a first-hash-then-sign scheme 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 $m = H_{0,IV_0}(M)$. Thus, 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 $m = H_0(M)$.

Lemma 12. *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 $\{0,1\}^{k \cdot \mathsf{b}}$ with $k > 0$ for some signature scheme Σ'.*

Thus, in the case $F(M, r) = H_s(M\|r)$, the security benefits inherent to using the general hash-and-sign paradigm are completely lost. For any $n = k \cdot \mathsf{b} > 0$, one has

$$\begin{array}{rcl} \mathsf{1\text{-}EF}^n\text{-}\mathsf{CMA}[\mathcal{S}],\ \mathsf{ER}^n[\mathcal{S}] & \Leftarrow & \mathsf{COL}^{n,n}[H_0]\ , \\ \mathsf{1\text{-}UF}_n\text{-}\mathsf{CMA}[\mathcal{S}],\ \mathsf{1\text{-}EF}^n\text{-}\mathsf{KMA}[\mathcal{S}],\ \mathsf{UR}_n[\mathcal{S}] & \Leftarrow & \mathsf{SEC}_n^n[H_0]\ . \end{array}$$

4.3 MD instantiation with operating mode $F(M, r) = H_s(r \| M)$

When looking at constructions such as $F(M, r) = H_s(r\|M)$, we first examine the "absolute"-like properties, $\mathsf{A\text{-}SEC}_n^n[F]$ and $\mathsf{A\text{-}COL}^{n,n}[F]$. Following the

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

definition, finding absolute collisions means that the mappings $r \mapsto F(M_i, r)$ are the same for two messages M_1, M_2. We can provide an upper-bound for the security by noticing that there is at most 2^n such mappings (viewed as parameterized by M of length n). The total number of r-bit-to-m-bit mappings being $(2^{\mathsf{m}})^{2^{\mathsf{r}}}$, two mappings coincide with reasonable probability as soon as $n \approx \mathsf{m}2^{\mathsf{r}}$, which is beyond practical lengths.

Claim. Let F be defined as $F(M, r) = H_s(r\|M)$ and let $\mathcal{S} = \langle F, \Sigma\rangle$. Combining the security statements of Lemmas 6 and 7, we conclude that there is **no known way** to break $\mathcal{S}$ in any sense **even** if H_0 is $\mathsf{COL}^{n,n}$-, SEC^n_n- and PRE^n-broken.

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

4.4 Concrete security figures for two instantiations of $F(M, r)$

We display on Table 2 concrete security workloads for effectively attacking $F(m, r)$ under current knowledge for the two instantiations $F(m, r) = H(M\|r)$ and $F(m, r) = H(r\|M)$, where H is chosen to be MD4, MD5, SHA-0 or SHA-1. We evaluate the expected running time τ_{H_s} of the attack algorithm when imposing a success probability ε_{H_s} heuristically close to one. For instance, for $F(M, r) = \mathsf{MD5}(M\|r)$, $\mathsf{U\text{-}COL}^{n,n}[F]$ is known to be $(2^{30}, \simeq 1)$-broken for $n = 2 \cdot 512$ *i.e.* for two-block messages. The message bitsize n is fixed to $2 \cdot 512$ for all figures except for MD4 where $n = 1 \cdot 512$.

The best attacks known are Sasaki's new message difference for MD4 [21], Klima's tunnels for MD5 [13], Wang's attack on SHA-0 [26], and the latest improvements over Wang's SHA-1 attack [27,16]. Complexity 2^{60} for SHA-1 is (announced) in [16]. Complexity 2^{33} for SHA-0 is given in [15]. The target collision attack against MD5 by Stevens et al. [23] 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.

5 Conclusion

This investigation on how recent attacks on hash functions may impact on signature schemes is particularly fruitful. We have first properly defined hash-and-sign signatures and the particular case of first-hash-then-sign signatures, and identified two core properties; injectivity and primitiveness. For each of these categories (and notably for schemes such as FDH, PSS and Schnorr), we have

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

	$H_s(M\|\|r)$				$H_s(r\|\|M)$			
	MD4	MD5	SHA-0	SHA-1	MD4	MD5	SHA-0	SHA-1
$\mathsf{A\text{-}SEC}_n^n\,[F]$					∞	∞	∞	∞
$\mathsf{E\text{-}SEC}_n^n\,[F]$		2^{52}			2^{58}			
$\mathsf{A\text{-}COL}^{n,n}\,[F]$	2^1	2^{30}	2^{33}	2^{60}	∞	∞	∞	∞
$\mathsf{U\text{-}COL}^{n,n}\,[F]$	2^1	2^{30}	2^{33}	2^{60}	2^1	2^{30}	2^{33}	2^{60}
$\mathsf{E\text{-}COL}^{n,n}\,[F]$	2^1	2^{30}	2^{33}	2^{60}	2^1	2^{30}	2^{33}	2^{60}

Table 2. Expected running time of attacks.

shown how their security relates to the one of their inner hash component. We then focused on the case of hash functions based on the Merkle-Damgård domain extender (such as MDx and SHAx). Enlightening the security properties of popular operating modes, *i.e.* the ones one would try first to construct a hash function family, we identified $F(M,r) = H_s(M\|r)$ as (by far) the worst operating mode in terms of security guarantees. Thus a concrete conclusion of our work is the suggestion of using $F(M,r) = H_s(r\|M)$ or more intricate operating modes.

6 Acknowledgments

This work is part of the SAPHIR project (`http://www.crypto-hash.fr`), partly funded by the French National Research Agency (ANR). The authors would like to thank all the members of the project for fruitful discussions.

References

1. ANSI X9.62-1998. Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm.
2. M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. Eurocrypt '96, 399–416.
3. M. Bellare and P. Rogaway. PSS: Provably secure encoding method for digital signatures. Submission to IEEE P1363a, August 1998.
4. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet and W. Jalby. Collisions of SHA-0 and Reduced SHA-1. Eurocrypt '05, 36–57.
5. D. Boneh and B. Lynn and H. Shacham. Short Signatures from the Weil Pairing. Asiacrypt '01, 514–532.
6. J.-S. Coron. Optimal security proofs for PSS and other signature schemes. Eurocrypt '2002, 272–287.
7. R. Cramer and V. Shoup. Signature Schemes Based on the Strong RSA Assumption. ACM Conference on Computer and Communications Security, 1999, 46-51.

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.

8. I. Damgård. A Design Principle for Hash Functions. Crypto '89, 416–427.
9. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signature without the random oracle. Eurocrypt '99, 123–139.
10. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. of Computing*, 17(2):281–308, 1988.
11. L. Guillou and J.-J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. Crypto '88, 216–231.
12. J. Kelsey and B. Schneier. Second preimages on n-bit hash functions for much less than 2^n work. Eurocrypt '05, 474–490.
13. V. Klìma. Tunnels in Hash Functions: MD5 Collisions Within a Minute. `http://eprint.iacr.org/2006/105`.
14. S. Lucks. A failure-friendly design principle for hash functions. Asiacrypt '05, 474–494.
15. S. Manuel and T. Peyrin. Collisions on SHA-0 in one hour. FSE '08, to appear.
16. F. Mendel, C. Rechberger and V. Rijmen. Update on SHA-1. *Rump Session* of Crypto '07.
17. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
18. R. Merkle. One Way Hash Functions and DES. Crypto '89, 428–446.
19. National Institute of Standards and Technology. Digital Signature Standard (DSS). FIPS —Publication 186, May 1994.
20. P. Rogaway Formalizing Human Ignorance. Vietcrypt 2006, 211–228.
21. Y. Sasaki, L. Wang, Y. Ohta, and N. Kunihiro. New messages difference for MD4. FSE '07, 329–348.
22. C. P. Schnorr. Efficient signatures generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
23. M. Stevens, A. Lenstra and B. de Weger. Chosen-prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities. Eurocrypt '07, 1–22.
24. X. Y. Wang and H. B. Yu. How to Break MD5 and Other Hash Functions. Eurocrypt '05, 19–35.
25. X. Y. Wang, X. J. Lai, D. Feng, H. Chen, and X. Yu. Cryptanalysis for Hash Functions MD4 and RIPEMD. Eurocrypt '05, 1–18.
26. X. Y. Wang, H. B. Yu, and Y. L. Yin. Efficient Collision Search Attacks on SHA-0. Crypto '05, 1–16.
27. X. Y. Wang, Y. L. Yin, and H. B. Yu. Finding Collisions in the Full SHA-1. Crypto '05, 17–36.

A Proofs of Lemmas 7 and 8

We adopt the concrete-security setting [20], as opposed to the asymptotic one. Given a computational problem P, a probabilistic algorithm is said to (τ, ε)-solve P when it halts after at most τ elementary steps and outputs a solution of P with success probability at least ε. A concrete black-box[7] 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.

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

A.1 Proof of Lemma 7

Proof ($\mathsf{U\text{-}PRE}^n\,[F] \Rightarrow \mathsf{EF}^n\text{-}\mathsf{KOA}\,[\mathcal{S}]$). Let us assume a 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). □

A.2 Proof of Lemma 8.

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 message $M_1 \leftarrow \{0,1\}^{n_1}$, $\mathcal{A}_F$ 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 $F(M_1, r)$. □

Appeared in J. Baek et al. (Eds.): ProvSec 2008, LNCS 5324, pp. 241–253.