Hash-and-sign with Weak Hashing Made Secure

Sylvain Pasini and Serge Vaudenay

EPFL
CH-1015 Lausanne, Switzerland
http://lasecwww.epfl.ch

Abstract. Digital signatures are often proven to be secure in the random oracle model while hash functions deviate more and more from this idealization. Liskov proposed to model a weak hash function by a random oracle together with another oracle allowing to break some properties of the hash function, e.g. a preimage oracle. To avoid the need for collision-resistance, Bellare and Rogaway proposed to use target collision resistant (TCR) randomized pre-hashing. Later, Halevi and Krawczyk suggested to use enhanced TCR (eTCR) hashing to avoid signing the random seed. To avoid the increase in signature length in the TCR construction, Mironov suggested to recycle some signing coins in the message preprocessing. In this paper, we develop and apply all those techniques. In particular, we obtain a generic preprocessing which allows to build strongly secure signature schemes when hashing is weak and the internal (textbook) signature is weakly secure. We model weak hashing by a preimage-tractable random oracle.

1 Introduction

A textbook signature scheme usually does a poor job because it restricts to input messages of fixed length and is often weakly secure. In order to sign messages of arbitrary length, hash functions [17,15,19,20] and the so-called *hash-and-sign paradigm* appeared. Clearly, the hash function must be collision resistant but they are threaten species these days [23,22,24]. In this paper we wonder how to recycle signature schemes that are currently implemented and based on (now) weak hash functions. To do so, we consider *generic* transform using preprocessing based on [4,9,13].

One crucial task is to find a model which fits to the current security of hash functions. A solution is to use the Liskov [12] idea. It consists of a random oracle that are provided together with another oracle that "breaks" the hash function, e.g. a first preimage oracle. We apply the preimage-tractable random oracle model (PT-ROM) to model weak hashing in digital signatures.

A natural solution to avoid the collision-resistance assumption is to add randomness in hashing. Bellare and Rogaway [4] proposed to sign $(K, H_K(m))$ with a random salt K where H is a Target Collision Resistant (TCR) hash function (also known as universal one-way hash function). More recently, Halevi and Krawczyk [9] proposed the concept of enhanced TCR (eTCR) hash function, some eTCR construction techniques, and the RMX construction based on current hash functions. This latter scheme only adds a randomized preprocessing on

the input message and thus standard implementations can be used as-is. As an application to their eTCR constructions, they suggest to use it as preprocessing for signatures and thus the salt K needs not to be signed. Here, we prove in our PT-ROM that this construction is *strongly* secure based on any textbook signature scheme which is *weakly* secure.

The disadvantage of the methods using a random seed K is that K must be appended to the signature. To avoid the increase in signature length, Mironov [13] proposed for DSA [7,6], RSA-PSS [3], and the Cramer-Shoup [5] schemes to reuse the randomness from the signature scheme instead of adding a new one. Finally, we generalize this construction and propose a *generic* transform that applies to special signature schemes. Indeed, we define special signature scheme for which we can split the sign algorithm in two parts: first, there is a randomized algorithm independent from the input message, then there is a deterministic algorithm which outputs the signature. We call these schemes Signatures with Randomized Precomputation (SRP). This makes the preprocessing transform less generic because we must assume that the signature generates some random coins which are available before the message is processed and which are extractable from the signature.

In this paper, we start with some preliminaries and then we present the hash-and-sign paradigm with many existing hashing methods. In particular, we present the TCR-based from Bellare-Rogaway [4] and eTCR-based signature from Halevi-Krawczyk [9] constructions. In Section 4, we give a formal proof of the Halevi-Krawczyk construction using weak hashing. In the subsequent section, we generalize the technique by Mironov [13] and we give a formal security proof. Finally, we present a direct application to DSA and validate the Halevi-Krawczyk construction with RMX preprocessing.

2 Preliminaries

Given a security parameter λ, we say that $f(\lambda)$ is polynomially bounded and we write $f(\lambda) = \mathsf{poly}(\lambda)$ if there exists n such that $f(\lambda) = \mathcal{O}(\lambda^n)$ when $\lambda \to +\infty$. We say that $f(\lambda)$ is negligible and we write $f(\lambda) = \mathsf{negl}(\lambda)$ if there exists $x > 0$ such that $f(\lambda) = \mathcal{O}(x^{-\lambda})$ when $\lambda \to +\infty$. For the sake of readability, our theorems are stated in terms of asymptotic complexity although they are proven by using exact complexities in some real-life computational model. The security parameter λ is almost always hidden in notations.

2.1 Digital Signature Schemes

Let $\mathcal{M}$ be the set of possible input messages, i.e. the *domain*. We define *fixed message-length digital signature* schemes (FML-DS) any signature scheme which applies only to a restricted message space $\mathcal{M} = \{0,1\}^{r(\lambda)}$ and *arbitrary message-length digital signature* (AML-DS) schemes the schemes when $\mathcal{M} = \{0,1\}^*$.

We formalize a *digital signature scheme* S by three algorithms: The setup algorithm $(K_p, K_s) \leftarrow S.\mathsf{setup}(1^\lambda)$ generates a key pair depending on a security parameter λ. The sign algorithm $\sigma \leftarrow S.\mathsf{sign}(K_s, m)$ outputs a signature $\sigma \in \mathcal{S}$

of a message $m \in \mathcal{M}$. The verify algorithm $b \leftarrow S.\mathsf{verify}(K_p, m, \sigma)$ tells whether the pair (m, σ) is valid or not. It returns 1 if and only if the signature is valid and 0 otherwise.

An FML-DS can be transformed into AML-DS following the hash-and-sign paradigm. Here, hashing is used as a *domain extender*. For instance, DSA [7,6] is based on SHA-1 [20] while RSA [16] uses MD5 [15] in the standard PKCS #1 v1.5 that is used in X.509 [11].

Consider an adversary $\mathcal{A}$ against S. $\mathcal{A}$ plays a game against a challenger $\mathcal{C}$ who can sign messages. The goal of $\mathcal{A}$ is to yield a valid pair $(\widehat{m}, \widehat{\sigma})$ which was not produced by $\mathcal{C}$. Textbook signature schemes such as ElGamal [8] or plain RSA [16] signatures are often existentially forgeable. We consider the *strong* security model EF-CMA and the *weak* security model UF-KMA.

UF-KMA and EF-CMA Games. The signature scheme is said (T, ℓ, ε)-UF-KMA (resp. EF-CMA) resistant if any adversary $\mathcal{A}$ bounded by a complexity T and ℓ valid signatures on known (resp. chosen) messages cannot win the game of Fig. 1 (resp. Fig. 2) with probability higher than ε[1]. The scheme is said UF-KMA secure (resp. EF-CMA-secure) if for any $T = \mathsf{poly}$ and $\ell = \mathsf{poly}$ there exists $\varepsilon = \mathsf{negl}$ such that the scheme is (T, ℓ, ε)-UF-KMA (resp. EF-CMA) resistant.

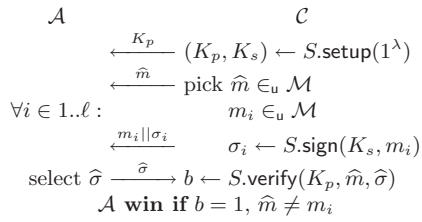

Fig. 1. UF-KMA game.

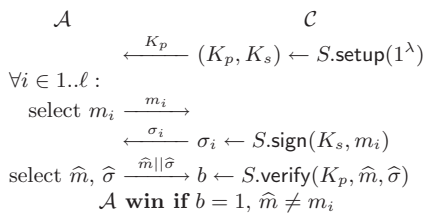

Fig. 2. EF-CMA game.

2.2 Hash Functions

Collision Resistant Hash Functions (CRHF) are hash functions in which we cannot construct two inputs x and y such that $H(x) = H(y)$ and $x \neq y$[2]. We say H depending on a security parameter λ is CRHF if any polynomially bounded adversary finds collisions with negligible probability.

Target Collision Resistant (TCR) Hash Functions was introduced by Naor and Yung [14] and then renammed in [4]. A (T, ε)-TCR is a keyed function $H : \{0,1\}^k \times \{0,1\}^* \mapsto \{0,1\}^n$ such that any adversary bounded by a complexity T cannot win the game of Fig. 3 with probability higher than ε. For $H^\lambda : \{0,1\}^{k(\lambda)} \times \{0,1\}^* \mapsto \{0,1\}^{n(\lambda)}$, we say H is TCR if any polynomially bounded adversary wins with negligible probability.

[1] Our results holds even if the winning conditions are replaced by $(\widehat{m}, \widehat{\sigma}) \neq (m_i, \sigma_i)$.
[2] Note that this definition is not so formal as discussed in Rogaway [18].

Enhanced Target Collision Resistant (eTCR) hash function was introduced by Halevi and Krawczyk [9]. A (T, ε)-eTCR is a stronger TCR function such that any adversary bounded by a complexity T cannot win the game of Fig. 4 with probability higher than ε. We say H is eTCR if any polynomially bounded adversary wins with negligible probability. A OW-eTCR hash function is an eTCR hash function for which $(\kappa, m) \mapsto H_\kappa(m)$ is also OW.

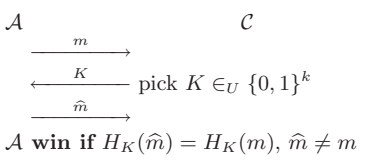

Fig. 3. TCR game.

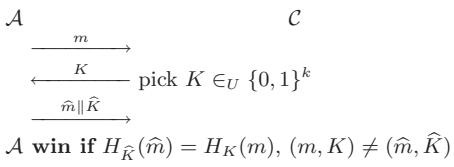

Fig. 4. eTCR game.

Random Oracle Hashing. A *Random Oracle* $\mathsf{R} : \{0,1\}^* \mapsto \{0,1\}^n$ often represents a uniformly distributed random hash function [2]. It is simulated by an oracle managing a table that is initially empty. When R receives a query with input m and there is an (m, r) entry in the table, it simply returns r. Otherwise, it picks a random value $r \in \{0,1\}^n$, returns it, and inserts a new entry (m, r) in the table.

Preimage-Tractable Random Oracle Hashing. Preimage-Tractable Random Oracles were introduced by Liskov [12]. It is used to idealize some weak hash function for which preimages are computable, i.e. the one-wayness is not guaranteed. It consists of two oracles:

- the first oracle G can be used to compute images like a random oracle, i.e. $r = \mathsf{G}(m)$,
- the second oracle $\mathsf{preimageG}$ can be used to find a preimage of a hashed value. When $\mathsf{preimageG}$ is queried with input r, it picks uniformly at random an element within the set of all its preimages, i.e. it outputs $m \in_{\mathsf{u}} \mathsf{G}^{-1}(r)$.

The simulation of G is done as for random oracle hashing with a table $\mathbb{T}$. To simulate $\mathsf{preimageG}$, upon a new query r we first compute the probability q to answer an m that is not new, i.e. $q = \Pr\left[(\mathsf{G}^{-1}(r), r) \in \mathbb{T} | \nexists_{(m',r') \in \mathbb{T}} \mathsf{G}(m') = r'\right]$. Then flip a biased coin b with $\Pr[b = 0] = q$ and if $b = 0$ we pick uniformly one (m, r) in $\mathbb{T}$ otherwise we pick uniformly one m such that $(m, r) \notin \mathbb{T}$, insert (m, r) in $\mathbb{T}$. Finally, answer by m. Note that this oracle can be used to find collisions as well.

From a theoretical viewpoint, the preimage-tractable random oracle is as powerful as the random oracle since $\mathsf{preimageG}(0\|\alpha) \oplus \mathsf{preimageG}(1\|\alpha)$ is indifferentiable from a random oracle even when $(\mathsf{G}, \mathsf{preimageG})$ is a preimage-tractable random oracle. Our motivation is to model weak hash functions which are in place without changing the algorithm implementations.

3 Domain Extension

3.1 Deterministic Hash-and-Sign

Given a hash function $H : \{0,1\}^* \to \{0,1\}^k$ and an FML-DS S_0 on domain $\{0,1\}^k$, we construct S' on domain $\{0,1\}^*$ by $S'.\mathsf{sign}(K_s, m) = S_0.\mathsf{sign}\,(K_s, H(m))$.

Theorem 1. *If H is a collision resistant hash function and S_0 is EF-CMA-secure, then S' is EF-CMA-secure.*

This folklore result is nicely treated in [18].

Theorem 2. *If H is a random oracle and S_0 is UF-KMA-secure, then S' is EF-CMA-secure.*

The proof of this folklore result is rather straightforward. Indeed, H brings *collision resistance* in domain extension as well as *unpredictability*.

3.2 Randomized Hash-and-Sign

The idea of using a TCR comes from Bellare and Rogaway [4] and was also reused recently by Mironov [13]. The constructed signature consists of the pair $(\kappa,\ S.\mathsf{sign}(K_s, \kappa \| H_\kappa(M)))$ where $H_\kappa(\cdot)$ is a TCR hash function. The following result is a straightforward generalization of Mironov [13].

Theorem 3. *Consider an FML-DS S_0 with domain $\{0,1\}^r$ and a function $G_0 : \{0,1\}^* \mapsto \{0,1\}^r$. We assume that $G_0(X)$ is indistinguishable from $Y \in_{\mathsf{u}} \{0,1\}^r$ when $X \in_{\mathsf{u}} \{0,1\}^{2r}$. Let $H : \{0,1\}^k \times \{0,1\}^* \mapsto \{0,1\}^n$ be a TCR hash function and $\mathsf{R} : \{0,1\}^{k+n} \mapsto \{0,1\}^r$ be a random oracle. We construct two AML-DS S and S' by*

$$\begin{aligned} S.\mathsf{sign}(K_s, m) &= S_0.\mathsf{sign}\,(K_s, G_0(m)) \\ S'.\mathsf{sign}(K_s, m) &= (\kappa \parallel S_0.\mathsf{sign}\,(K_s, \mathsf{R}\,(\kappa \| H_\kappa(m)))) \qquad \text{with } \kappa \in_{\mathsf{u}} \{0,1\}^k \end{aligned}$$

Assuming that S is EF-CMA-secure, then S' is also EF-CMA-secure.

This means that if there exists a domain extender G_0 that makes S secure, then S' is secure.

Proof. Consider $H : \{0,1\}^k \times \{0,1\}^* \mapsto \{0,1\}^n$ is a $(T + \mu_H, \varepsilon_H)$-TCR hash function for μ_H to be defined later, $\mathsf{R} : \{0,1\}^{k+n} \mapsto \{0,1\}^r$ is a random oracle bounded to q queries, and S_0 an FML-DS scheme with r-bit input messages. We assume that the construction S is $(T + \mu_S, \ell, \varepsilon_S)$-EF-CMA secure for μ_S to be defined later. We assume that G_0 is $(T+\mu_G, q+\ell+1, \varepsilon_d)$-PRG when restricted to $(2r)$-bit inputs. We will prove that the construction S' is $(T, \ell, \varepsilon_S + \ell\varepsilon_H + \varepsilon_c + \varepsilon_d)$-EF-CMA secure where ε_c represents a probability of collision on the outputs of the random oracle.

We consider an adversary $\mathcal{A}$ playing the EF-CMA game against S'. We assume without loss of generality that $\mathcal{A}$ queries R with $H_{\widehat{\kappa}}(\widehat{m})$ before releasing the final forgery $(\widehat{m}, \widehat{\kappa}, \widehat{\sigma})$ (so we have up to $q+1$ queries to R). By using an

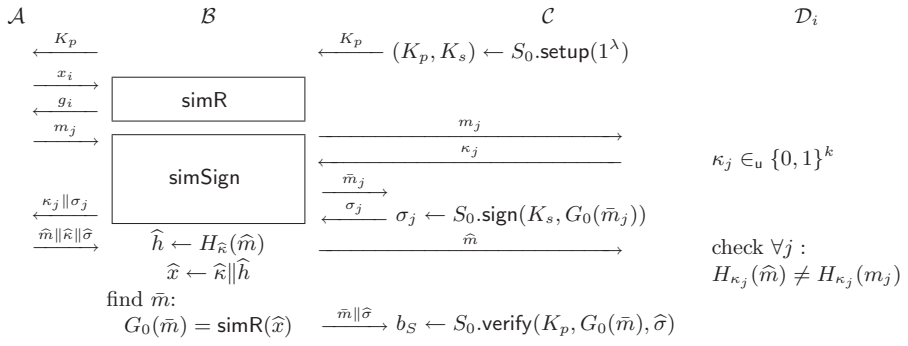

Fig. 5. Reduction to EF-CMA or TCR games (from EF-CMA).

algorithm $\mathcal{B}$, we prove that we can reduce $\mathcal{A}$ to an adversary against either the signature construction S or the TCR hash function H.

The reduction is depicted on Fig. 5. Clearly, $\mathcal{B}$ has to simulate the random oracle R and the signing oracle that we refer to simR and $\mathsf{simSign}$ respectively. The simulations work as follows:

simR**:** $\mathcal{B}$ manages a table $\mathbb{T}$ initially empty. For each R-query with input x:
- if $\mathsf{simR}(x)$ is not defined in $\mathbb{T}$, $\mathcal{B}$ picks a random $\bar{m}$ uniformly in $\{0,1\}^{2r}$ and answers $g \leftarrow G_0(\bar{m})$. Hence, a new entry $(x, g, \bar{m})$ is inserted in $\mathbb{T}$, meaning $\mathsf{simR}(x) = g = G_0(\bar{m})$. Note that the third entry $\bar{m}$ will be used by $\mathsf{simSign}$ only.
- otherwise, $\mathcal{B}$ answers $\mathsf{simR}(x)$ as defined in $\mathbb{T}$.

$\mathsf{simSign}$**:** For each sign-query with input m:
1. $\mathcal{B}$ computes $h \leftarrow H_\kappa(m)$, $x \leftarrow \kappa\|h$ where κ is returned by $\mathcal{D}_i$ on query m,
2. $\mathcal{B}$ queries $\mathsf{simR}(x)$. Let $\bar{m}$ be such that $\mathsf{simR}(x) = G_0(\bar{m})$ from $\mathbb{T}$,
3. $\mathcal{B}$ queries $\mathcal{C}$ with $\bar{m}$ to obtain its signature σ,
4. finally, $\mathcal{B}$ returns $\kappa\|\sigma$ to $\mathcal{A}$.

$\mathcal{B}$ is allowed to ℓ queries to the $S_0.\mathsf{sign}$ oracle, so $\mathcal{A}$ is also allowed to ℓ queries to $\mathsf{simSign}$. Note that the $\mathsf{simSign}$ simulation is perfect but the simR simulation is not. At the end, if $\mathcal{A}$ succeeds, he returns a forged pair $(\widehat{m}, \widehat{\kappa}, \widehat{\sigma})$ to $\mathcal{B}$. We use the proof methodology of Shoup [21]:

- Let game_0 be the EF-CMA game against S' depicted on Fig. 2.
- Let E_1 be the event that there were no collision on the output of R. Let game_1 be game_0 in which E_1 occurred.
  Clearly, when E_1 does not occur, there is a collision on the R outputs. Since there is at most $q + \ell + 1$ elements in the simR table, this probability is bounded by $\varepsilon_c \leq \frac{(q+\ell+1)^2}{2} 2^{-r}$. So, $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_0] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_1] \leq \varepsilon_c$.
- Let game_2 be game_1 where the R oracle was replaced by the simR simulator. Let $\mathcal{A}'$ simulate $\mathcal{A}$ and simR in which picking a random $\bar{m}$, computing $g \leftarrow G_0(\bar{m})$, and inserting $(x, g, \bar{m})$ in the table is replaced by getting a

random g^* from a source Σ and storing (x, g^*) in the table. We consider the two following sources: Σ_0 picks g^* with uniform distribution and Σ_1 picks $\bar{m}$ and output $g^* \leftarrow G_0(\bar{m})$. Note that using Σ_0 perfectly simulates game_1 while using Σ_1 perfectly simulates game_2. At the end, $\mathcal{A}'$ checks whether the EF-CMA game succeeded. Clearly, this is a distinguisher of some complexity $T + \mu_G$ between Σ_0 and Σ_1 by using $q + \ell + 1$ samples. So, $|\Pr[\mathcal{A} \text{ wins } \mathsf{game}_1] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_2]| \leq \varepsilon_d$.

- Let game_3 be the simulated EF-CMA game of Fig. 5. Since the simulation $\mathsf{simSign}$ of the signing oracle is perfect, we have $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_3] = \Pr[\mathcal{A} \text{ wins } \mathsf{game}_2]$.
- Let E_4 be the event that the final $\bar{m}$ was not queried to $\mathcal{C}$. Let game_4 be the game_3 in which E_4 occurred. In that case, $\mathcal{A}$ can be perfectly reduced to an EF-CMA adversary of complexity $T + \mu_s$ against $\mathcal{C}$. So, $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_4] \leq \varepsilon_S$.

  Clearly, if E_4 did not occur, $\bar{m}$ was previously queried to $\mathcal{C}$. Let $\bar{m} = \bar{m}_j$, i.e. $\bar{m}$ was queried by $\mathcal{B}$ to $\mathcal{C}$ at the j^{th} sign-query. Thus, $\mathcal{B}$ queried simR with an input x_j and obtained $(x_j, G_0(\bar{m}_j), \bar{m}_j)$. Since there were no collision on simR, $\bar{m} = \bar{m}_j$ implies that $\widehat{x} = x_j$ thus $\widehat{\kappa} = \kappa_j$ and $\widehat{h} = h_j$. We have $H_{\widehat{\kappa}}(\widehat{m}) = H_{\widehat{\kappa}}(m_j)$. $\widehat{m}$ is different from all m_i since $\mathcal{A}$ won his attack against S'. Hence, $\mathcal{A}$ can be perfectly reduced to a TCR adversary against $\mathcal{D}_j$ and $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_3] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_4] \leq \ell\varepsilon_H$.

We conclude by considering the above reductions that μ_H and μ_S are within the order of magnitude of the simulation cost which is polynomial. □

The problems of such constructions are that (1.) we do not have a full reduction to the weak security of S_0; (2.) the signature enlarges; (3.) κ must be signed; (4.) we still need a random oracle R (implicitly meaning collision-resistant hashing) so the role of R is to concentrate on unpredictability and nevertheless, R is now restricted to $\{0,1\}^{k+m}$.

Halevi and Krawczyk [9] also use a randomized hashing but avoid signing the κ salt. Indeed, they use an eTCR hash function. In [9], they proposed a construction technique for eTCR based on weak hashing and suggested to use it as preprocessing for signature schemes. The signature consists of the pair (κ, σ) where σ is $S.\mathsf{sign}(K_s, H_\kappa(m))$. One problem is that they do no provide any proof of security for the signature so far. Indeed, they only focus on the problem for constructing an eTCR hash function based on weak hashing.

4 Strong Signature Schemes With Weak Hashing

We consider a deterministic hash-and-sign signature S put together with the Halevi and Krawczyk [9] message processing. Namely, given a weakly-secure FML-DS S_0 we construct a strongly-secure AML-DS S' as follows:

$\sigma' \leftarrow S'.\mathsf{sign}(K_s, m)$:
- pick $\kappa \in_{\mathsf{u}} \{0,1\}^k$
- $s \leftarrow H_\kappa(m)$
- $h \leftarrow G(s)$
- $\sigma' \leftarrow (\kappa \| S_0.\mathsf{sign}(K_s, h))$

$b \leftarrow S'.\mathsf{verify}(K_p, m, \sigma')$: $\quad (\sigma' = \kappa \| \sigma)$
- $s \leftarrow H_\kappa(m)$
- $h \leftarrow G(s)$
- $b \leftarrow S_0.\mathsf{verify}(K_p, h, \sigma)$

where $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$ is an eTCR hash function family, $G : \{0,1\}^n \to \{0,1\}^r$ a (weak) hash function, and S_0 is an UF-KMA secure FML-DS on domain $\{0,1\}^r$. Clearly, for S defined by $S.\mathsf{sign}(K_s, m) = S_0.\mathsf{sign}(K_s, G(m))$, our construction can be seen as a regular AML-DS based on hash-and-sign with an extra randomized preprocessing $H_\kappa(\cdot)$.

Theorem 4. *Consider H is an OW-eTCR hash function family, and G is a preimage-tractable random oracle. If S_0 is an UF-KMA-secure FML-DS, then S' in the above AML-DS construction is EF-CMA-secure.*

Clearly, we can build strong signature schemes for arbitrary messages based on any weak signature scheme restricted to fixed-length input messages *without* collision-resistance and *without* a full random oracle. The remaining drawback is that the signature enlarges.

Note that the OW assumption on H is necessary since G is assumed to be preimage-tractable (otherwise, existential forgeries on S_0 would translate in existential forgeries on S'). and eTCR hash functions may be not OW. Indeed, if H is eTCR, then H' defined by

$$H'_\kappa(m) = \begin{cases} 0\|m & \text{if } \kappa = 0\ldots0 \text{ and } |m| = n-1, \\ 1\|H_\kappa(m) & \text{otherwise.} \end{cases}$$

is eTCR as well but not OW. However, when there exists a set of messages $\mathcal{M}$ such that H is a PRG when restricted to $\{0,1\}^{k\times\mathcal{M}}$, then eTCR implies OW-eTCR.

Proof. Let us assume that S_0 is $(T+\mu, \ell, \varepsilon_S)$-UF-KMA-secure, H is $(T+\mu, \varepsilon_H)$-eTCR and $(T+\mu, \varepsilon_w)$-OW, and G is a random oracle limited to $q < \ell$ queries where μ is some polynomially bounded complexity (namely, the overhead of some simulations). We will show that S' is $(T, \ell-q, \varepsilon_f + q_p\cdot\varepsilon_w + (\ell-q)\cdot\varepsilon_H + q\cdot\varepsilon_S)$-EF-CMA-secure where ε_f represents a probability of failure during the reduction.

We start by considering an EF-CMA adversary $\mathcal{A}$ against our constructed scheme S'. We assume that $\mathcal{A}$ is bounded by complexity T. By using an algorithm $\mathcal{B}$, we transform $\mathcal{A}$ into either an UF-KMA adversary against S_0 or into an eTCR adversary against H as depicted on Fig. 6. Here, $\mathcal{C}$ plays the role of the challenger in the UF-KMA game of Fig. 1 while each $\mathcal{D}_{i''}$ plays the role of the i''^{th} challenger in the eTCR game of Fig. 4.

Clearly, algorithm $\mathcal{B}$ has to simulates for $\mathcal{A}$ the signing oracle and the two oracles that model the preimage-tractable hash function that we refer by $\mathsf{simSign}$, simG, and $\mathsf{preimageG}$ respectively. To simulate G and $\mathsf{preimageG}$, we use another existing preimage-tractable random oracle G_0 and $\mathsf{preimageG}_0$ and we construct

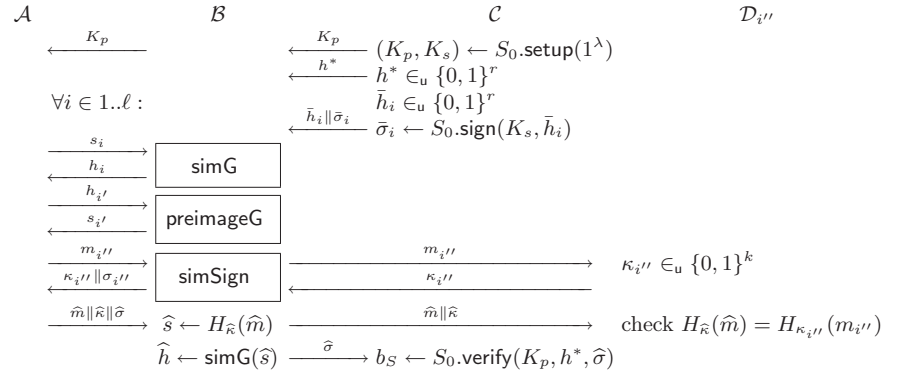

Fig. 6. Reduction to the UF-KMA or eTCR games (from EF-CMA).

a random permutation φ such that $\mathsf{G} = \varphi \circ \mathsf{G}_0$. We consider a growing pool of values of s. The pool is initially empty. A new s is put in the pool if it is queried to simG or returned by $\mathsf{preimageG}$. Without loss of generality, we assume that $\mathcal{A}$ makes no trivial queries to simG. Namely, he does not query simG with an s already in the pool. Similarly, we assume that if $\widehat{s} = H_{\widehat{\kappa}}(\widehat{m})$ is not in the pool, $\mathcal{A}$ queries $\mathsf{simG}(\widehat{s})$ before releasing $\widehat{m}\|\widehat{\kappa}\|\widehat{\sigma}$ to make sure that $\widehat{s}$ is in the pool. (So we may have $q+1$ queries to simG.) The simulations work as follows:

simG**:** At the beginning of the game, $\mathcal{B}$ picks a random $t \in_u \{1..q\}$. When $\mathcal{A}$ submits a G-query with input s:
- if $\varphi(\mathsf{G}_0(s))$ is undefined, it answers the next $\bar{h}_i$ in the sequence except that for the t^{th} query it answers h^*. Hence, there is a new entry $\varphi(G_0(s)) = h$ in the φ table.
- If $\varphi(\mathsf{G}_0(s))$ is already defined, $\mathcal{B}$ aborts.

$\mathsf{preimageG}$**:** When $\mathcal{A}$ submits a $\mathsf{preimageG}$ query with input h, if $x = \varphi^{-1}(h)$ is not defined, it picks a random x on which $\varphi(x)$ is not defined and define $\varphi(x) = h$. Then, it queries $\mathsf{preimageG}_0(x)$ and answers s.

$\mathsf{simSign}$**:** When $\mathcal{A}$ submits a sign-query with input m, $\mathcal{B}$ queries a new $\mathcal{D}_{i''}$ with input m, gets κ, and computes $s = H_\kappa(m)$. If s is in the pool, $\mathcal{B}$ abort. Otherwise, $\mathcal{B}$ runs $h \leftarrow \mathsf{simG}(s)$ without counting this query (that is, use the next $\bar{h}_i$ in the sequence and not h^*). Thus, $\mathsf{simG}(s)$ is equal to one of the $\bar{h}_i$ and $\mathcal{B}$ uses the corresponding signature $\bar{\sigma}_i$ to answer $\kappa\|\bar{\sigma}_i$.

Note that $\mathcal{B}$ has ℓ signed samples from $\mathcal{C}$, thus $\mathcal{A}$ is limited to ℓ queries to simG and $\mathsf{simSign}$. So, $q + q_s \leq \ell$. At the end, if $\mathcal{A}$ succeeds his EF-CMA game, he will send a tuple $(\widehat{m}, \widehat{\kappa}, \widehat{\sigma})$ to $\mathcal{B}$.

We use the proof methodology of Shoup [21]:

- Let game_0 be the EF-CMA game against S' of Fig. 2.
- Let game_1 be the simulated EF-CMA game against S' depicted on Fig. 6. Clearly, the simulations fails when a $\varphi(\mathsf{G}_0(s))$ is already defined while querying simG with s or when $s = H_\kappa(m)$ was already in the pool while querying

simSign. Let ε_f the bound on this failure probability. By using the difference lemma [21] we obtain $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_0] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_1] \leq \varepsilon_f$. Note that $\varepsilon_f \leq \Pr[\mathcal{B} \text{ fails on a } \mathsf{simG} \text{ query}] + \Pr[\mathcal{B} \text{ fails on a } \mathsf{simSign} \text{ query}]$. We consider $\mathcal{A}$ is bounded by q, q_p and q_s queries to simG, preimageG, and simSign respectively, and a space of 2^r elements. First, we compute the probability that $\mathcal{B}$ fails on a simG query, i.e. there were a collision of $G_0(s)$ for one s queried to simG with one $G_0(s')$ for s' in the pool. By considering the queries from $\mathcal{A}$ and from simSign, there are at most $q + q_s + 1$ queries to simG and at most $q + q_s + q_p + 1$ elements still defined in the pool. Since they are uniformly distributed, the probability that two elements collide is 2^{-r}. So, $\Pr[\mathcal{B} \text{ fails on a } \mathsf{simG} \text{ query}] \leq (q + q_s + 1)(q + q_s + q_p + 1) \cdot 2^{-r}$.
Now, we compute the probability that $\mathcal{B}$ fails on a simSign query, i.e. s was already in the pool. There are at most q_s queries to simSign and at most $q + q_s + q_p + 1$ elements s in the pool. For each query-s pair, we have the following scenario: $\mathcal{A}$ queries simSign with m, $\mathcal{B}$ queries $\mathcal{D}$ with m, gets κ, computes $H_\kappa(m)$, and looks if it is s. Clearly, this scenario can be described as game (a) of Fig.7. Let p the maximal success probability among all random coins of the adversary $\mathcal{A}$ in the game (a).

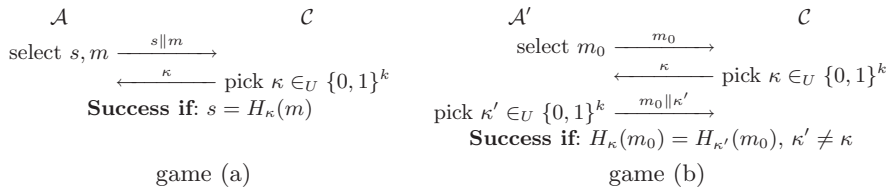

Fig. 7. Reduction to the eTCR Game

Now, consider game (b) depicted on Fig.7. Clearly, this game is harder than the eTCR game since $\mathcal{A}'$ has no control on the second message returned to $\mathcal{C}$, i.e. it is m_0. We know that ε_H is a bound on the success probability of A' in the eTCR game. Thus, we have:

$$\begin{aligned}\varepsilon_H &\geq \Pr[s_0 = H_\kappa(m_0) = H_{\kappa'}(m_0) \text{ and } \kappa' \neq \kappa] \\ &\geq \Pr[s_0 = H_\kappa(m_0) = H_{\kappa'}(m_0)] - \Pr[\kappa' = \kappa] \\ &= p^2 - 2^{-k}.\end{aligned}$$

We conclude that $p \leq \sqrt{\varepsilon_H + 2^{-k}}$ and so, $\varepsilon_f \leq (q + q_s + 1)(q + q_s + q_p + 1) \cdot 2^{-r} + q_s(q + q_s + q_p + 1) \cdot \sqrt{\varepsilon_H + 2^{-k}}$ is negligible.

– Let E_2 be the event that the forgery $\widehat{m} \| \widehat{\kappa} \| \widehat{\sigma}$ is such that $\widehat{s} \leftarrow H_{\widehat{\kappa}}(\widehat{m})$ was queried to simG. Let game_2 be game_1 in which E_2 occurred.
Since we made sure that $\widehat{s}$ is in the pool, if E_2 does not occur, the $\widehat{s}$ was returned by some preimageG(h) for the first time once. Note that when preimageG returns an unused value, it is uniformly distributed among all unused values. Clearly, $\mathcal{A}$ has to find a pair $(\widehat{m}, \widehat{\kappa})$ with $H_{\widehat{\kappa}}(\widehat{m}) = \widehat{s}$ which breaks the one-wayness of H. So, $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_1] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_2] \leq q_p \cdot \varepsilon_w$.

– Let E_3 be the event that $\widehat{s}$ is different from all $s_{i''} \leftarrow H_{\kappa_{i''}}(m_{i''})$. Let game_3 be game_2 in which E_3 occurred.
Clearly, if E_3 did not occur, $\widehat{s}$ is equal to $s_{i''}$ for a certain i''. Recall that since $\mathcal{A}$ won his game $\widehat{m}$ is different from all $m_{i''}$. So, $\mathcal{A}$ found $\widehat{m}$ and $\widehat{\kappa}$ such that $H_{\widehat{\kappa}}(\widehat{m}) = H_{\kappa_{i''}}(m_{i''})$. Here, $\mathcal{A}$ can perfectly be reduced to an eTCR adversary against all $\mathcal{D}_{i''}$. So, $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_2] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_3] \leq q_s \cdot \varepsilon_H \leq (\ell - q) \cdot \varepsilon_H$.
– Let E_4 be the event that $\widehat{h} = h^*$. In other words the forged value $\widehat{h}$ is equal to the expected value h^*. Let game_4 be game_3 in which E_4 occurred. Here, $\mathcal{A}$ can perfectly be reduced to an UF-KMA adversary against S_0. Clearly, $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_4] \leq \varepsilon_S$.
Finally $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_3] \leq q \cdot \varepsilon_S$ since E_4 occurred with probability $1/q$ and so $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_4] / \Pr[\mathcal{A} \text{ wins } \mathsf{game}_3] = 1/q$.

□

5 The Entropy Recycling Technique

To keep the same signature length, we have to avoid to append κ in the signature. The idea from [13] is to use the randomness computed in the signature scheme instead of introducing a new random parameter. Mironov [13] present specific modifications for the DSA [7,6], RSA-PSS [3], and Cramer-Shoup [5] signature schemes. In this section, we generalize the construction from Mironov. For that, we introduce a special sort of signature schemes: Signature with Randomized Precomputation.

A *Signature with Randomized Precomputation* (SRP) is any signature scheme for which the signature algorithm can be separated in two parts:

– first, a *probabilistic* precomputation algorithm generates the randomness without the message to be signed,
– then, a signature algorithm signs the message using the previous randomness.

Note that the randomness must be recoverable from the signature itself, which requires another algorithm $\mathsf{extract}$. We can formalize any SRP scheme by the following five algorithms:

$$(K_p, K_s) \leftarrow S.\mathsf{setup}(1^\lambda)$$
$$(\xi, r) \leftarrow S.\mathsf{presign}(K_s) \qquad r \leftarrow S.\mathsf{extract}(K_p, \sigma)$$
$$\sigma \leftarrow S.\mathsf{postsign}(K_s, m, \xi) \qquad b \leftarrow S.\mathsf{verify}(K_p, m, \sigma)$$

Actually, all digital signature schemes can be written this way (e.g. with r void), but we need r to have a large enough entropy. We provide the necessary quantitative definitions for that in Appendix. When talking about the entropy of a SRP scheme, we implicitly mean the entropy of r generated by $S.\mathsf{presign}(K_s)$ given a key K_s.

Theorem 5. *Consider H is an eTCR hash function with t-bit keys and S_0 is a FML-SRP. We assume that the signature construction S based on S_0 defined by*

$\sigma' \leftarrow S.\mathsf{sign}(K_s, m)$:
- *pick* $\kappa \in_{\mathsf{u}} \{0,1\}^t$
- $(\xi, r) \leftarrow S_0.\mathsf{presign}(K_s)$
- $\sigma \leftarrow S_0.\mathsf{postsign}(K_s, H_\kappa(m), \xi)$
- *output* $\kappa \| \sigma$

$b \leftarrow S.\mathsf{verify}(K_p, m, \kappa \| \sigma)$:
- $b \leftarrow S_0.\mathsf{verify}\,(K_p, H_\kappa(m), \sigma)$
- *output* b

is an EF-CMA secure AML-SRP requiring an additional randomness κ*. We assume that the SRP produces* t*-bit strings that are indistinguishable from uniformly distributed ones.*

Consider R *is a random oracle with* k*-bit output strings limited to* q *queries. The signature construction* S' *defined by*

$\sigma' \leftarrow S'.\mathsf{sign}(K_s, m)$:
- $(\xi, r) \leftarrow S_0.\mathsf{presign}(K_s)$
- $\sigma \leftarrow S_0.\mathsf{postsign}(K_s, H_{\mathsf{R}(r)}(m), \xi)$
- *output* σ

$b \leftarrow S'.\mathsf{verify}(K_p, m, \sigma')$: $\quad(\sigma' = \sigma)$
- $r \leftarrow S_0.\mathsf{extract}(K_p, \sigma)$
- $b \leftarrow S_0.\mathsf{verify}\left(K_p, H_{\mathsf{R}(r)}(m), \sigma\right)$
- *output* b

is also EF-CMA-secure even by re-using the randomness from the SRP.

Proof. Assume that the AML-SRP construction S is $(T+\mu, \ell, \varepsilon_S)$-EF-CMA secure and that r is $(T+\mu, \ell, \varepsilon_d)$-PR where μ is some polynomially bounded complexity due to the game reduction. In the following, we prove that the construction S' is $(T, \ell, \varepsilon_S + \varepsilon_c)$-EF-CMA secure where ε_c represents the probability of collision on the R outputs as defined in Lemma 3. We consider any EF-CMA adversary $\mathcal{A}$ against S'. As depicted on Fig. 8, we transform $\mathcal{A}$ into an EF-CMA adversary against the eTCR-based scheme S by using an algorithm $\mathcal{B}$ which simulates the random oracle R, the transform of $S'.\mathsf{sign}$ to $S.\mathsf{sign}$, and replaces the final $(\widehat{m}, \widehat{\sigma})$ by $(\widehat{m}, \widehat{\kappa}, \widehat{\sigma})$.

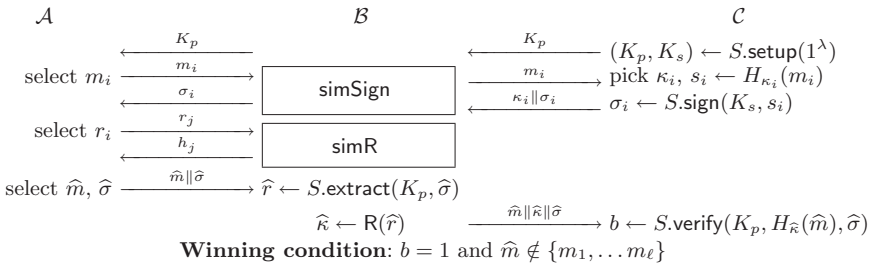

Fig. 8. Reduction to the EF-CMA game against the eTCR-based scheme S.

The simulations works as follows:

simR works as defined in Section 2.2.

simSign When $\mathcal{A}$ submits a sign-query with input m, $\mathcal{B}$ obtains (κ, σ) by querying $\mathcal{C}$ and deduces $r \leftarrow S.\mathsf{extract}(K_p, \sigma)$. If r is free in the simG table, it lets $\kappa = \mathsf{R}(r)$ and returns σ to $\mathcal{A}$, otherwise $\mathcal{B}$ fails.

$\mathcal{B}$ is allowed to ℓ queries to the $S_0.\mathsf{sign}$ oracle, so $\mathcal{A}$ is also allowed to ℓ queries to $\mathsf{simSign}$. At the end, if $\mathcal{A}$ succeeds his EF-CMA game, he will send a tuple $(\widehat{m}, \widehat{\kappa}, \widehat{\sigma})$ to $\mathcal{B}$. We use one more time the proof methodology of Shoup [21]:

- Let game_0 be the EF-CMA game against S' of Fig. 2.
- Let game_1 be the simulated EF-CMA game against S' depicted on Fig. 6. Clearly, the simulation fails if $\mathsf{simSign}$ fails, i.e. if an r_j in $\mathsf{simSign}$ is not free in the simR table. Let ε_c the bound on this probability of collision.
  Let E_1 the event that all r_j are free in the simR table. So, game_1 is game_0 in which E_1 occurred. Here, $\mathcal{A}$ can perfectly be reduced to an EF-CMA adversary against S. So $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_2] \leq \varepsilon_S$.
  We obtain $\Pr[\mathcal{A} \text{ wins } \mathsf{game}_0] - \Pr[\mathcal{A} \text{ wins } \mathsf{game}_1] \leq \varepsilon_c$ by using the difference lemma [21]. A detailed expression of ε_c is given on Lemma 3. It is clearly negligible.

□

6 Application to DSA

We apply Theorem 4 and Theorem 5 to offer a quick fix to DSA in the case that SHA-1 [20] became subject to preimage attacks. Here, standard implementations of DSA could still be used: only a "message preprocessing" would be added. First, note that DSA without hashing can be described using our SRP formalism of Section 5. We denote by m the messages of arbitrary length (input of the sign algorithm) and by h the digest in DSA, i.e. the 160-bit sting. The public parameters are q a 160-bit prime, $p = a \cdot q + 1$ a 1024-bit prime, and $g \in \mathbb{Z}_p$ a generator of order q.

The DSA construction is depicted on Fig. 9 where $f(m)$ describes some function mapping the arbitrary message length to a fixed length strings which represents the "message preprocessing".

$$
\begin{array}{ll}
(K_s, K_p) \leftarrow S.\mathsf{setup}(1^\lambda): & \text{pick } K_s \in_{\mathsf{u}} \mathbb{Z}_q \\
& K_p \leftarrow g^{K_s} \bmod p \\
\sigma \leftarrow S.\mathsf{sign}(K_s, m, k, r): & \text{pick } k \in_{\mathsf{u}} \mathbb{Z}_q^* \\
& r \leftarrow (g^k \bmod p) \bmod q \\
& h \leftarrow f(m) \\
& s \leftarrow \frac{h + K_s \cdot r}{k} \bmod p \\
& \sigma \leftarrow (r, s) \\
b \leftarrow S.\mathsf{verify}(K_p, m, \sigma): & h \leftarrow f(m) \\
& \text{check } r = (g^{\frac{h}{s} \bmod q} y^{\frac{r}{s} \bmod q} \bmod p) \bmod q
\end{array}
$$

Fig. 9. The DSA Construction

DSA uses the (original) hash-and-sign paradigm. $f(m)$ is simply

$$h \leftarrow H^*(m)$$

where H^* is a collision resistant hash function.

Consider textbook DSA is an UF-∅MA-secure FML-DS. Note that it is existentially forgeable. Theorem 4 says that the scheme of Fig. 9 where $f(m)$ is

$$h \leftarrow \mathsf{G}(H_\kappa(m)) \qquad \text{where } \kappa \in_{\mathsf{u}} \{0,1\}^k,$$

is EF-CMA-secure when G is a preimage-tractable random oracle (say SHA-1 in practice) and H is a one-way eTCR hash function. Thus, we build an EF-CMA-secure AML-DS based on DSA without collision-resistance. Assuming that $G(H_\kappa(m))$ can be instantiated by $\mathsf{SHA1}(\mathsf{RMX}(\kappa, m))$ where RMX denotes the implementation from Halevi-Krawczyk [10] of the message randomization, the Halevi-Krawczyk construction is secure. The drawback is that the signature enlarges sending κ.

Instead of picking some new randomness κ we re-use randomness from the presign algorithm if the implementation of DSA allows it, i.e. we use $\mathsf{R}(r)$ where R is a random oracle. Theorem 5 says that the scheme of Fig. 9 where $f(m)$ is

$$h \leftarrow \mathsf{G}(H_{\mathsf{R}(r)}(m))$$

is EF-CMA-secure as well.

From Theorem 4 and Theorem 5, we deduce that our construction is (T, Q, ε'_s)-EF-CMA-secure where $\varepsilon'_s \leq \varepsilon_f + q_p \cdot \varepsilon_w + (\ell - q) \cdot \varepsilon_H + q \cdot \varepsilon_S + \varepsilon_c$. Assuming an adversary bounded by a time complexity T and an online complexity $Q \leq T$, considering that ε_H, ε_s and ε_w are all equals to $T \cdot 2^{-160}$, k is 160-bit long, q, q_s, and ℓ are bounded by Q, and q_p is bounded by T, we obtain $\varepsilon_f \leq 9 \cdot Q \cdot T \cdot 2^{-160}$, $\varepsilon_c \leq Q^2 \cdot 2^{-160}$ and so

$$\varepsilon'_s \leq \left(12 \cdot Q \cdot T + Q^2\right) \cdot 2^{-160}.$$

Clearly, $Q \cdot T$ must be bounded by 2^{160}. Since Q is often near 2^{30}, we deduce that T can be close to 2^{130} which is much better than actual implementations requiring a complexity T bounded by 2^{80} to avoid collision attacks.

In summary, by using Theorem 4 and Theorem 5, we build a DSA-based EF-CMA-secure scheme for input messages of arbitrary length and with signatures as long as the original DSA scheme.

7 Conclusion

Consider any signature implementation S based on a textbook signature scheme S_0 and using the original hash-and-sign paradigm with a hash function G, i.e. $S.\mathsf{sign}(K_s, m) = S_0.\mathsf{sign}(K_s, G(m))$. Assume that S_0 is weakly secure and that some weakness on G was reported.

By using Theorem 4, we can build a strongly secure implementation by adding a preprocessing $H_\kappa(m)$ where H is an OW-eTCR hash function. Our new construction S' defined by $S'.\mathsf{sign}(K_s, m) = S.\mathsf{sign}(K_s, H_\kappa(m)) = S_0.\mathsf{sign}(K_s, G(H_\kappa(m)))$ is strongly secure and actual implementations can still be used, it simply needs to "preprocess" the input message. This assumes that G can be modeled as a preimage-tractable random oracle.

References

1. *Advances in Cryptology – CRYPTO '05*, volume 3621 of *LNCS*. Springer, 2005.
2. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93*, LNCS, pages 62–73. ACM Press, 1993.
3. M. Bellare and P. Rogaway. The exact security of digital signatures – how to sign with RSA and Rabin. In *EUROCRYPT '96*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.
4. M. Bellare and P. Rogaway. Towards Making UOWHFs Practical. In *CRYPTO '97*, volume 1294 of *LNCS*, pages 470–484. Springer, 1997.
5. R. Cramer and V. Shoup. Signature Schemes Based on the Strong RSA Assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.
6. Digital signature standard (DSS). Federal Information Processing Standard, Publication 186-2, U.S. Department of Commerce, NIST, 2000.
7. Digital signature standard (DSS). Federal Information Processing Standard, Publication 186, U.S. Department of Commerce, NIST, 1994.
8. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
9. S. Halevi and H. Krawczyk. Strengthening Digital Signatures via Randomized Hashing. In *CRYPTO '06*, volume 4117 of *LNCS*, pages 41–59. Springer, 2006.
10. S. Halevi and H. Krawczyk. The RMX Transform and Digital Signatures. http://www.ee.technion.ac.il/∼hugo/rhash/, 2006.
11. R. Housley, W. Ford, W. Polk, and D. Solo. RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. IETF RFC Publication, 1999.
12. M. Liskov. Constructing an Ideal Hash Function from Weak Ideal Compression Functions. In *SAC '06*, pages ???–???, 2006.
13. I. Mironov. Collision-Resistant No More: Hash-and-Sign Paradigm Revisited. In *PKC '06*, volume 3958 of *LNCS*, pages 140–156. Springer, 2006.
14. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *ACM Symposium on Theory of Computing*, pages 33–43, 1989.
15. R. L. Rivest. The MD5 message digest algorithm. Technical Report Internet RFC-1321,IETF, 1992.
16. R. L. Rivest, A. Shamir, and L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
17. Ronald L. Rivest. The MD4 Message Digest Algorithm. In *CRYPTO '90*, volume 537 of *LNCS*, pages 303–311. Springer, 1991.
18. P. Rogaway. Formalizing Human Ignorance: Collision-Resistant Hashing without the Keys. In *VietCrypt '06*, volume 4341 of *LNCS*, pages 221–228. Springer, 2006.
19. Secure hash standard. Federal Information Processing Standard, Publication 180, U.S. Department of Commerce, NIST, 1993.
20. Secure hash standard. Federal Information Processing Standard, Publication 180-1, U.S. Department of Commerce, NIST, 1995.
21. V. Shoup. Sequences of Games: A Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, Report 2004/332. `http://eprint.iacr.org/`.
22. X. Wang, Y. Yin, and H. Yu. Finding collisions in the full SHA1. In *CRYPTO '05* [1], pages 17–36.
23. X. Wang and H. Yu. How to break MD5 and other hash functions. In *EUROCRYPT '05*, volume 3494 of *LNCS*, pages 19–35. Springer, 2005.
24. X. Wang, X. Yu, and L. Y. Yin. Efficient collision search attacks on SHA-0. In *CRYPTO '05* [1], pages 1–16.

A Probability of Collisions

We provide the necessary quantitative definitions of the entropy of a random variable.

Definition 1. *Let X a random variable in a set $\mathcal{X}$ with distribution $\mathcal{D}$. We define:*
the min-entropy *of X by:* $H_\infty(\mathcal{D}) = -\log \max_{x\in_{\mathcal{D}}\mathcal{X}} \Pr[X = x]$
the Renyi entropy *(of order 2) of X by:* $H_2(\mathcal{D}) = -\log \sum_{x\in_{\mathcal{D}}\mathcal{X}} \Pr[X = x]^2$

Mironov [13] computed the probability of collision on the outputs of a random oracle R.

Lemma 1 ([13]). *Let $\mathcal{R}$ denotes a set of possible r_j values with cardinality q. We consider ℓ i.i.d. trials r_i with distribution $\mathcal{D}$. Let ε_c be the probability that at least one of the trials is in $\mathcal{R}$ or at least two of the trials are equal. We have*

$$\varepsilon_c \leq 2^{-2\cdot H_\infty(\mathcal{D})} \cdot \ell^2 \cdot q + 2^{-H_\infty(\mathcal{D})} \cdot \ell^2 \tag{1}$$

Note that we can use another bound for ε_c in terms of Renyi entropy as described in Lemma 2 or as pseudo-randomness as described in Lemma 3.

Lemma 2. *Let $\mathcal{R}$ denotes a set of possible r_j values with cardinality q. We consider ℓ i.i.d. trials r_i with distribution $\mathcal{D}$. Let ε_c be the probability that at least one of the trials is in $\mathcal{R}$ or at least two of the trials are equal. We have*

$$\varepsilon_c \leq \frac{\ell^2}{2} \cdot 2^{-H_2(\mathcal{D})} + \ell \cdot \sqrt{q} \cdot 2^{-\frac{H_2(\mathcal{D})}{2}} \tag{2}$$

Proof. Let $p_x = \Pr[r = x]$. We have

$$\begin{aligned}\varepsilon_c &= \Pr[\exists i, j : i \neq j, r_i = r_j \text{ or } r_i \in \mathcal{R}] \\ &\leq \frac{\ell^2}{2} \sum_x p_x^2 + \ell \sum_{x\in\mathcal{R}} p_x \leq \frac{\ell^2}{2} \sum_x p_x^2 + \ell\sqrt{q}\sqrt{\sum_x p_x^2}\end{aligned}$$

□

Lemma 3. *Let $\mathcal{R}$ denotes a set of possible r_j values with cardinality q. We consider ℓ i.i.d. trials r_i with distribution $\mathcal{D}$. Let ε_c be the probability that at least one of the trials is in $\mathcal{R}$ or at least two of the trials are equal. Assuming that $\mathcal{D}$ is (ℓ, ε)-PR in $\{0,1\}^\rho$, we have*

$$\varepsilon_c \leq q \cdot 2^{-\rho} + \frac{\ell^2}{2} \cdot 2^{-\rho} + \varepsilon \tag{3}$$