Hash Functions: From Merkle-Damgård to Shoup

Ilya Mironov*
mironov@cs.stanford.edu

Computer Science Department, Stanford University, Stanford, CA 94305

Abstract. In this paper we study two possible approaches to improving existing schemes for constructing hash functions that hash arbitrary long messages. First, we introduce a continuum of function classes that lie between universal one-way hash functions and collision-resistant functions. For some of these classes efficient (yielding short keys) composite schemes exist. Second, we prove that the schedule of the Shoup construction, which is the most efficient composition scheme for universal one-way hash functions known so far, is optimal.

1 Introduction

In the pursuit of efficient and provably secure constructions of practical cryptosystems several basic primitives have emerged as useful building blocks. Two of them are collision-resistant hash functions (CRHFs) and universal one-way hash functions (UOWHFs). In the complexity-theoretic sense UOWHF is a strictly weaker primitive than CRHF, because the latter is also the former but there is an oracle relative to which UOWHFs exist but not CRHFs [Si98]. Therefore it might be reasonable to base practical cryptosystems on a weaker primitive, which can be easier to construct. Also, since no unconditionally secure UOWHFs are known, the assumption that a particular family of functions is a UOWHF can be more plausible than the assumption of its collision-resistance.

A UOWHF is a collection of keyed compressing functions $\{h_k\}_{k\in K}$ such that winning the following game is infeasible: The adversary chooses x, then receives a key $k \in K$ picked at random and wins if he can find y such that $h_k(x) = h_k(y)$.

A CRHF is a set of keyed compressing functions $\{f_k\}_{k\in K}$ such that for a random $k \in K$ it is infeasible to find x and y that satisfy $f_k(x) = f_k(y)$.

In many applications it is convenient to have a family of UOWHFs or CRHFs, i.e., a collection of functions that map bit strings of different lengths into fixed length strings. The problem is to construct such a family given a single UOWHF or CRHF, which is typically the case when one begins with an off-the-shelf function, for instance, MD5 or SHA-1. For CRHFs a widely used, provably secure and efficient method is the Merkle-Damgård construction [D89,M89]. Surprisingly, this construction does not apply to UOWHFs ([BR97] gave a concrete example of a UOWHF on which the Merkle-Damgård construction fails). For building UOWHF families the best method known so far is due to Shoup [Sh00].

* Supported by NSF contract #CCR-9984259

In this paper we study applicability of the Merkle-Damgård construction, introducing a continuum of primitives that lie between CRHF and UOWHF. Then we give an alternative proof of the Shoup construction and prove that this construction is optimal in some restricted model of computations. The optimality result is the major contribution of the paper.

2 Motivation: Key length of different composition schemes

The first application of UOWHFs in [NY89] was to use them as a tool for constructing a signature scheme secure under the most general attack. However, most practical signature schemes that follow "hash-and-sign" paradigm use UOWHFs or CRHFs in a different way. They take a message M of an arbitrary length and hash it to obtain a constant length string, which is then fed into a signing algorithm. Many schemes use CRHF families to hash M, but as it was first noted in [BR97] a UOWHF suffices for that purpose. Indeed, if $\{h_k\}_{k \in K}$ is a UOWHF, then $(k, h_k(M))$, where k chosen at random, can be signed and still be as secure as the underlying signature algorithm. If the key length varies with the length of a message, the signing algorithm is applied to $(h_{K'}(k), h_k(M))$, where K' is part of the signer's public key. Here function $h_{K'}$ can be replaced by any second-preimage resistant function, because its input is random and chosen by the signer. Since messages can be very long, hashing speed is a crucial factor. Again, because a UOWHF is a weaker primitive than a CRHF, we may hope to find a more efficient algorithm that implements a UOWHF, thus speeding up the signature scheme.

A closer look at this approach reveals that the key k must be part of the signature so the receiver can recompute the hash. Therefore the shorter the key the better. This is our motivation for studying different composition schemes that yield hash functions with a short key.

The problem of composing a family of UOWHFs does not exist in case of CRHFs, since the Merkle-Damgård construction does not increase the key size. Ironically, if we consider two competing algorithms one implementing a CRHF and a more efficient one, which is supposedly a UOWHF, a signature scheme based on the CRHF can outperform a scheme that uses a family of UOWHFs.

Among several composition schemes for UOWHFs [BR97,Sh00] the one with the smallest key expansion is due to Shoup [Sh00]. Characteristics of the Shoup construction are the following. Suppose that the starting point is a UOWHF that has key length l and compresses n bits to m bits. The composition scheme yields a family of UOWHFs such that a function that compresses N bits to m bits is keyed by $m \cdot \log_2 \lceil N/(n-m) \rceil + l$ bits. The key length grows logarithmically with the length of a message. Schemes in [BR97,NY89] have the same asymptotics but a bigger constant factor.

In Table 1 we give a concrete example of the signature length on messages of various sizes if we couple 1024-bit modulus RSA with either a CRHF or a

UOWHF. The UOWHF as in [Sh00] results from the Shoup construction applied to a keyed SHA-1 compression function, which hashes 672 bits to 160 bits.

Table 1. Length of RSA signatures with 1024-bit modulus.

Message length	**CRHF**	**UOWHF**
$\|M\| = 1$Kb	$\|S\| = 1$Kb	$\|S\| = 1.81$Kb
1Mb	1Kb	3.22Kb
1Gb	1Kb	4.87Kb

3 Between CRHF and UOWHF

The condition imposed on the round function by the Merkle-Damgård composition theorem can be relaxed. We consider the Merkle-Damgård construction as a useful test that can be applied to function classes filling the gap between CRHFs and UOWHFs. We define these classes in the next section.

3.1 Definitions

CRHFs and UOWHFs enjoy different types of collision-resistance and their constructions base on different assumptions. This adds to the impression that these two primitives have nothing in common. In fact, the only difference between them is in the degree of freedom that the adversary has in choosing one of the colliding elements. In case of a UOWHF, the adversary commits to x before he knows the key, while to defeat a CRHF the adversary is free to choose x afterwards. This difference can be easily quantified by specifying how many bits of x the adversary commits to before he knows the key. Qualitative differences between several variations of hash functions were demonstrated in [ZMI90]. We shall see that the Merkle-Damgård construction may be extended to a class of functions that lie between CRHF and UOWHF.

Definition 1 (class $\mathrm{CR}_{\ell_i}(n_i \to m_i)$**).** Let $\{(n_i, m_i, \ell_i)\}_{i\in\mathbb{N}}$ be a sequence of non-repeating triplets of integer numbers such that $0 < m_i < n_i$ and $0 \le \ell_i \le n_i$ for any i. We say that a collection of keyed functions $h^i_k\colon \{0,1\}^{n_i} \to \{0,1\}^{m_i}$, where $k \in K_i$, belongs to **class** $\mathrm{CR}_{\ell_i}(n_i \to m_i)$ if no adversary can win the following game for infinitely many i in time $\mathrm{poly}(n_i)$ with probability at least $1/\mathrm{poly}(n_i)$:

1. The adversary selects some $x_0 \in \{0,1\}^{n_i-\ell_i}$.
2. Key k is chosen at random from K_i.
3. The adversary selects $x_1 \in \{0,1\}^{\ell_i}$ and $y \in \{0,1\}^{n_i}$ such that $h^i_k(x_1\|x_0) = h^i_k(y)$.

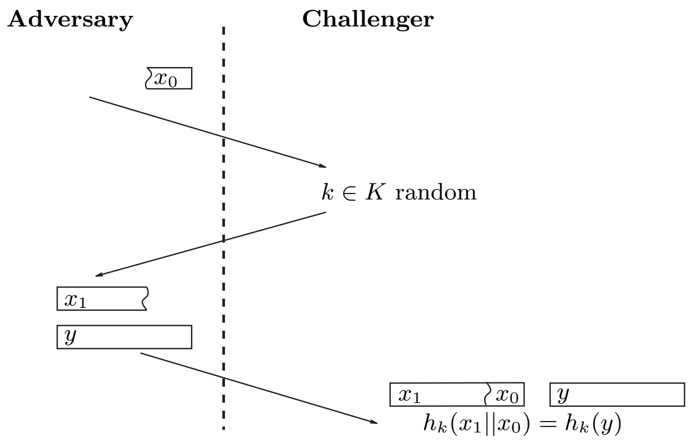

Fig. 1. Function h_k from $\mathrm{CR}_\ell(n \to m)$.

We call the ℓ_i bits that the adversary is free to choose the *flexibility* of a class.

This definition subsumes the definitions of UOWHFs and CRHFs. The class of functions with zero flexibility, $\mathrm{CR}_0(n_i \to m_i)$, is the class of UOWHFs, where the adversary must choose in its entirety one of the colliding elements before he knows the key. On the other hand, functions with full flexibility, $\mathrm{CR}_{n_i}(n_i \to m_i)$, constitute the class of CRHFs, since the adversary commits to nothing ahead of time.

We may omit the index parameter i for the sake of notation brevity. It does not imply that we consider a single triple (n, m, ℓ) (our asymptotic definition is inept in this setting), but that the subsequent arguments can be uniformly applied to the whole family of $\{(n_i, m_i, \ell_i)\}_{i \in \mathbb{N}}$. For example, we can formulate and prove the following propositions without utilizing the index variable.

Proposition 1. $\mathrm{CR}_{\ell_1}(n \to m) \subseteq \mathrm{CR}_{\ell_2}(n \to m)$ *if* $\ell_1 \geq \ell_2$.

Proof. Because higher flexibility gives more power to the adversary, any set of functions that qualifies as $\mathrm{CR}_{\ell_1}(n \to m)$ also belongs to $\mathrm{CR}_{\ell_2}(n \to m)$. $\square$

Proposition 2. *A collision for a function from* $\mathrm{CR}_\ell(n \to m)$ *can be found in* $O(2^{\max(m-\ell, m/2)})$ *evaluations of this function.*

Proof. Consider the birthday attack that applies to the flexible part of the input. $\square$

3.2 Merkle-Damgård Construction Applies to $\mathbf{CR}_{\boldsymbol{\ell}}(\boldsymbol{n} \to \boldsymbol{m})$, Where $\boldsymbol{\ell} \geq \boldsymbol{m}$

Suppose we have a family of functions $\{h_k\}_{k\in K} \in \mathrm{CR}_\ell(n \to m)$, where $\ell \geq m$. *Merkle-Damgård construction with variable IV and r rounds* (Merkle-Damgård construction for short) is an operator that takes a function h_k and transforms it into a function $\mathbf{MD^r} h_k \colon \{0,1\}^{r\cdot(n-m)+m} \mapsto \{0,1\}^m$. This function is built according to this rule:

Merkle-Damgård construction with variable IV and r rounds

1. Input x formatted as $(x_0, x_1, \ldots, x_r)$ such that $|x_0| = m$, $|x_1| = \cdots = |x_r| = n - m$.
2. Chaining variable C_0 is initialized as x_0.
3. For $i = 1$ to r let $C_i = h_k(C_{i-1}, x_i)$.
4. Output of the function $\mathbf{MD}^r h_k(x)$ is C_r.

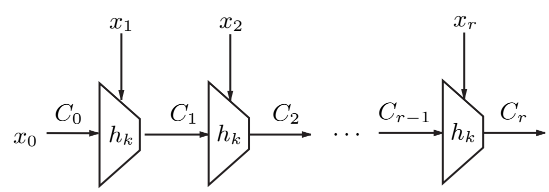

Fig. 2. r-round Merkle-Damgård construction.

This is the usual Merkle-Damgård construction except that the initializing value (IV) is also part of the input. The following theorem proves that the construction works correctly on functions from the class $\mathrm{CR}_\ell(n \to m)$, where $\ell \geq m$.

Theorem 1. *Suppose we have a family of functions $\{h_k^i\}_{i\in\mathbb{N},k\in K} \in \mathrm{CR}_{\ell_i}(n_i \to m_i)$, where $\ell_i \geq m_i$, and a sequence $\{r_i\}_{i\in\mathbb{N}}$ such that $r_i < \mathrm{poly}(n_i)$. Then $\{\mathbf{MD}^{r_i} h_k^i\}_{i\in\mathbb{N},k\in K}$ is in the class $\mathrm{CR}_{\ell_i}(r_i \cdot (n_i - m_i) + m_i \to m_i)$.*

Proof. Suppose that $\ell_i = m_i$. The case of $\ell_i > m_i$ is treated analogously. Assume for contradiction that there is an algorithm $\mathcal{A}$ that wins the game described in Definition 1 for infinitely many i. We build an algorithm $\mathcal{B}$ that contradicts the fact that $\{h_k^i\}_{i\in\mathbb{N},k\in K} \in \mathrm{CR}_{\ell_i}(n_i \to m_i)$.

Fix some i. Denote the value $\mathcal{A}$ commits to on the first step of this game by $x = (x_1, \ldots, x_r)$. Choose $0 < j \leq r_i$ at random. Commit to x_j in the game played by $\mathcal{B}$.

Key k is chosen at random. Let $\mathcal{A}$ find x_0 and $y = (y_0, y_1, \ldots, y_r)$ that make a collision. Now $\mathbf{MD}^{r_i} h_k^i(x_0||x) = \mathbf{MD}^{r_i} h_k^i(y)$. With probability at least $1/r$

we have a collision on the j^{th} application of the function h_k^i. It means that $h_k^i(C_{j-1}||x_j) = h_k^i(C'_{j-1}||y_j)$, but $C_{j-1}||x_j \neq C'_{j-1}||y_j$, where C_j and C'_j are the chaining variables. If it is the case, $\mathcal{B}$ outputs a colliding pair $C_{j-1}||x_j$ and $C'_{j-1}||y_j$. □

The fact that $\ell_i \geq m_i$ is crucial for the proof. Because the flexible part of the hashes' input is longer than their output, the adversary does not need to commit to the chaining variable if he wants to use $\mathcal{A}$ to find a collision. The value of the chaining variable depends on the key and cannot be predicted ahead of time. It is where the proof breaks if we want to apply it to the case when $\ell_i < m_i$.

Note 1. In practice we want to have a CRHF or a UOWHF that takes as input any string of some bounded length and maps it to a fixed-length string. This is stronger primitive than a collection of functions each taking a fixed-length input, because it must be collision resistant (resp. be a UOWHF) across inputs of different length. It turns out that a collection of fixed-input length functions can be strengthened to allow a variable input length. We assume that the message is padded to a length divisible by the block size ($n - m$ in case of the Merkle-Damgård construction) and the last block of the message uniquely encodes the message length before padding. This preprocessing stage was proposed in [M89] and discussed in [LM92] along with a definition of a free-start attack. Theorem 1 can be generalized to hold for this strengthened construction ([LM92] proved the theorem for families of pure CRHFs and UOWHFs).

3.3 Boundary

Because there is a complexity-theoretic jump between CRHFs and UOWHFs, we may expect to observe at least one such a jump in the sequence of classes $\mathrm{CR}_0(n \to m) \supseteq \cdots \supseteq \mathrm{CR}_n(n \to m)$. The following theorems show that this is indeed the case and there are two classes of complexity-theoretic equivalence. The boundary between them coincides with the limit of validity of the Merkle-Damgård construction (Section 3.2).

We recall that UOWHFs are one-way functions and can be built from a family of one-way functions [Ro90]. This also implies existence (via black-box constructions) of other cryptographic primitives such as secure signature schemes [NY89], pseudo-random generators [HILL99], telephone coin flipping [B82] and bit commitment protocols [N91]. [Si98] proved that there is no black-box (relativizing) construction of a CRHF based on a UOWHF. In our terminology it means that there is no unconditional construction of $\mathrm{CR}_n(n \to m)$ given access to $\mathrm{CR}_0(n \to m)$ as a black-box.

Theorem 2. $\mathrm{CR}_{m-O(\log n)}(n \to m)$ *is non-empty if and only if CRHFs exist.*

Proof. The adversary playing the game from Definition 1 may choose values for $O(\log n)$ bits randomly. His probability of success drops in this case by a factor of $2^{O(\log n)} = \mathrm{poly}(n)$. Therefore $\mathrm{CR}_{m-O(\log n)}(n \to m) = \mathrm{CR}_m(n \to m)$.

Since $\mathrm{CR}_m(n \to m) \supseteq \mathrm{CR}_n(n \to m)$ by Proposition 1, the "if" part is trivial.

Suppose we have a $\{h_k\}_{k \in K} \in \mathrm{CR}_m(n \to m)$. Define $g_k \colon \{0,1\}^{m+1} \mapsto \{0,1\}^m$ for any $k \in K$ as follows. Suppose that g_k takes two arguments—a single bit b and x, which is m-bit long. Let

$$g_k(b, x) = h_k(x || \underbrace{b \dots b}_{n-m \text{ times}}).$$

We claim that $\{g_k\}_{k \in K} \in \mathrm{CR}_{m+1}(m+1 \to m)$, i.e., it is a CRHF family.

Assume the opposite. There is an efficient algorithm $\mathcal{A}$ that for a random k finds a collision $g_k(b_0, x) = g_k(b_1, y)$. Wlog we may assume that $b_0 = 0$ with probability at least 1/2. We want to show that $\{h_k\}_{k \in K} \notin \mathrm{CR}_m(n \to m)$. In order to prove it we build algorithm $\mathcal{B}$ that wins the game from Definition 1 as follows:

step 1. Commit to 0^{n-m}.
step 2. Get $k \in K$.
step 3. Run $\mathcal{A}$ to find a collision $g_k(b_0, x) = g_k(b_1, y)$. If $b_0 = 0$ proceed to the next step, otherwise the algorithm fails.
step 4. Output $(x || 0^{n-m}, y || b_1^{n-m})$ as a collision for h_k.

The output of $\mathcal{B}$ is indeed a collision, since

$$h_k(x || 0^{n-m}) = g_k(x, 0) = g_k(x, b_0) = g_k(y, b_1) = h_k(y || b_1^{n-m})$$

and, because $b_0 || x \neq b_1 || y$, these two elements of the domain of h_k are different. The success probability of $\mathcal{B}$ is at least one half of the success probability of $\mathcal{A}$.

Notice that $g_k(0, x)$ and $g_k(1, x)$ is a pair of claw-free pseudo-injections (see [Ru95]). $\square$

Theorem 3. $\mathrm{CR}_{m - m^{\Omega(1)}}(n \to m)$ *is not empty if and only if UOWHFs exist.*

Formally, if some $\mathrm{CR}_{\ell_i}(n_i \to m_i)$ *is not empty, then UOWHFs exist. If UOWHFs exist, then for any* $\ell(m) \colon \mathbb{N} \mapsto \mathbb{N}$*, such that* $m - \ell(m) > m^c$ *for some* $0 < c < 1$*, a non-empty class* $\mathrm{CR}_{\ell(m_i)}(n_i \to m_i)$ *exists.*

Proof. Since $\mathrm{CR}_0(n \to m) \supseteq \mathrm{CR}_\ell(n \to m)$, the "only if" part is trivial.

Suppose UOWHFs exist. Take $\{h_k\}_{k \in K} \in \mathrm{CR}_0(n \to m)$. Then for any $\ell < \mathrm{poly}(n)$ we may define $g_k \colon \{0,1\}^{n+\ell} \mapsto \{0,1\}^{m+\ell}$ as

$$g_k(x, y) = y || h_k(x),$$

where $|y| = \ell$ and $|x| = n$. We claim that $\{g_k\}_{k \in K} \in \mathrm{CR}_\ell(n + \ell \to m + \ell)$.

Indeed, if there is a collision $g_k(x_0, y_0) = g_k(x_1, y_1)$, then $y_0 = y_1$ and $h_k(x_0) = h_k(x_1)$. Note that y_0 is the flexible part of the input and x_0 is the part that the adversary commits to before he knows the key. The adversary works poly-time in $n + \ell$. Since $\ell < \mathrm{poly}(n)$, the adversary's running time is also polynomial in n. Therefore the same adversary can be used to break UOWHF-ness property of $\{h_k\}_{k \in K}$.

Suppose we are given a family of UOWHFs $\{h_k^i\}_{i\in\mathbb{N},k\in K} \in \mathrm{CR}_0(n_i' \to m_i')$. For every m_i' there is some m, such that $m^c/2 < m_i' < m^c \le m - \ell(m)$. The construction above with $\ell = m - m_i' < (2m_i')^{1/c} < (2n_i')^{1/c} = \mathrm{poly}(n_i')$ yields a collection of functions from $\mathrm{CR}_\ell(n_i' + \ell \to m_i' + \ell) = \mathrm{CR}_\ell(n_i' + \ell \to m_i) \subseteq \mathrm{CR}_{\ell(m_i)}(n_i' + \ell \to m_i)$. The last inclusion is because $\ell = m - m_i' > \ell(m)$ and by Proposition 1. □

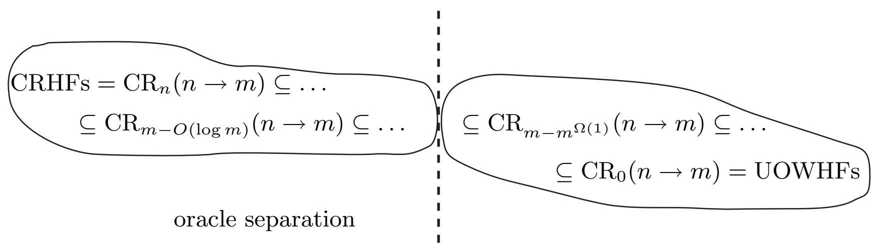

Fig. 3. Hierarchy of classes.

Theorems 2 and 3 show that there are two classes of complexity-theoretic equivalence of classes $\mathrm{CR}_\ell(n \to m)$ (Figure 3). One contains CRHFs and all $\mathrm{CR}_\ell(n \to m)$ for $\ell \ge m - O(\log n)$, the other one spans classes between UOWHFs and $\mathrm{CR}_\ell(n \to m)$ for $\ell < m - m^{\Omega(1)}$. The following note eliminates the gap between them.

Note 2. The claim of Theorem 3 can be improved if we assume that $n < \mathrm{poly}(m)$ and $\mathrm{CR}_0(n \to m)$ has "ideal" security $\Omega(2^m)$ as in Proposition 2. With these assumptions there is no gap between the two theorems and there are only two classes of equivalence.

4 Optimality of the Shoup Construction

[BR97] gave an example of a UOWHF on which the two-round Merkle-Damgård construction fails. Since we want to build UOWHFs the same way we build families of CRHFs, i.e., starting with a keyed function that has fixed-length input, other constructions have to be studied. The most efficient among different composition schemes that have appeared in the literature is the Shoup construction. We give an alternative proof of its correctness, which is technically simpler than in [Sh00] and conceptually better matches our main result, the proof of its optimality.

4.1 Shoup Construction

The Shoup construction (see Figure 4) can be viewed as an extended Merkle-Damgård construction, where the chaining variable is XORed with some mask on each iteration. Because these masks are reused, the key length grows logarithmically with the size of the message.

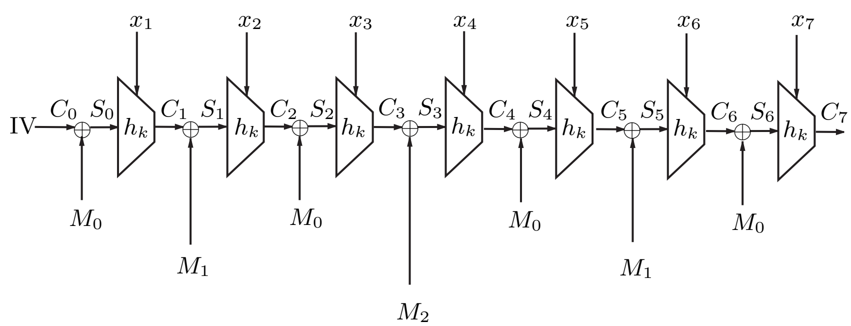

Fig. 4. 7-round Shoup construction.

Formally, *the r-round Shoup construction* is an operator that takes function $h_k\colon \{0,1\}^n \mapsto \{0,1\}^m$, bit-vector M with length $Lm > m\lfloor \log r \rfloor$, which is formatted as L masks $M = (M_0, \ldots, M_{L-1})$, and transforms it into a function $\mathbf{S}^{r,M} h_k\colon \{0,1\}^{r\cdot(n-m)} \mapsto \{0,1\}^m$. This function is built according to this rule:

r-round Shoup construction

1. Input x formatted as $(x_1, \ldots, x_r)$ such that $|x_1| = \cdots = |x_r| = n - m$.
2. Chaining variable C_0 is initialized as IV.
3. For $i = 1$ to r let $C_i = h_k(C_{i-1} \oplus M_{\nu(i)}, x_i)$, where the auxiliary function $\nu(i)$ is the highest power of 2 that divides i.
4. Output of the function $\mathbf{S}^{r,M} h_k(x)$ is C_r.

We defer the proof of correctness of this construction to Section 4.3.

4.2 Optimality of the Shoup Construction

The Shoup construction achieves its short, compared to other constructions, key length of the composite scheme by reusing the bit-masks. A legitimate question is whether the masks can be reused even more. In this section we give a negative answer to this question. We prove that the Shoup construction really reuses masks as much as possible in the strongest sense.

Definition 2. A **generalized r-round Shoup construction** is the Shoup construction as described in Section 4.1 but with function ν, which selects a mask to use on every iteration of the construction, being any function that maps $[1, \ldots, r]$ to $[0, \ldots, L-1]$. Function ν is the **schedule** of the construction. The construction is **valid** if it transforms any UOWHF family into a UOWHF family.

The Shoup construction instantiates the function $\nu(i) = \max\{j \colon 2^j | i\}$, so $L = \lfloor \log r \rfloor + 1$. A schedule is optimal if L is minimal for a fixed r. The following theorem says the Shoup scheduling is optimal for all r.

Theorem 4. *For any valid generalized r-round Shoup construction $r < 2^L$.*

Proof. The proof consists of two steps. First, we show that any schedule of a valid construction must be even-free (defined below). Second, we prove that any schedule with $r \geq 2^L$ is not even-free.

Definition 3 (even-freeness property). We say that a schedule ν is **even-free** if for any a and b, such that $1 \leq a \leq b \leq r$, there is some $0 \leq \eta < L$, such that the number of times $\nu(c)$ takes value η for $a \leq c \leq b$ is odd. In other words, there is no sub-interval that contains every mask an even number of times.

Lemma 1. *Any valid schedule ν is even-free.*

Proof. Suppose there is a non even-free schedule ν of some valid r-round generalized Shoup construction. We build a UOWHF family on which this construction fails, thus contradicting validity of the construction.

Assume that $g_k \colon \{0,1\}^n \mapsto \{0,1\}^m$ is a UOWHF, $2m + 2 < n$ and $k \in K = \{0,1\}^m$. Of course, if UOWHFs do not exist, then every construction is valid but the problem itself is moot. If we have *some* UOWHF family, by adding an additional argument to its input that gets replicated to the output we can ensure that $2m + 2 < n$. The size of the key space can be adjusted similarly. We define function $h_k \colon \{0,1\}^n \mapsto \{0,1\}^{2m+1}$ as follows:

$$h_k(y, z, b, x) = \begin{cases} g_k(y, z, b, x)||z||1 & \text{if } x \neq 0^l \text{ and } z \neq k \\ g_k(y, z, b, x)||k||1 & \text{if } x = 0^l \text{ and } z \neq k \\ 0^{2m+1} & \text{if } z = k, \end{cases}$$

where $|y| = |z| = |k| = m$, b is a bit and $|x| = l = n - 2m - 1 > 1$. As usual, we omit the index i of the family of UOWHFs, assuming that the construction of h_k and the proof below apply uniformly to all functions of the family.

We claim that h_k is a UOWHF. Indeed, a collision $h_k(y, z, x) = h_k(y', z', x')$ also yields a collision $g_k(y, z, x) = g_k(y', z', x')$ unless $z = k$ and $z' = k$. Probability that the adversary hits $z = k$ *before* he knows k is negligible.

If ν is not even-free, there is a sub-interval $[a, b]$ that contains each mask an even number of times. We exploit this property to find a collision with a previously committed value.

The composite scheme takes as its input $x = (x_1, \ldots, x_r)$, where $|x_i| = l = n - 2m - 1$ for all i. We claim that the following x and x' collide:

$$x = \underbrace{0^l||\ldots||0^l}_{a \text{ times}}||\underbrace{1^l||\ldots||1^l}_{r-a \text{ times}},$$
$$x' = \underbrace{0^l||\ldots||0^l}_{a-1 \text{ times}}||1^{l-1}0||\underbrace{1^l||\ldots||1^l}_{r-a \text{ times}}.$$

Denote the input of h_k on the i^{th} iteration of the composite scheme by (y_i, z_i, b_i, x_i) and (y'_i, z'_i, b'_i, x'_i) for the inputs x and x' respectively. Format masks $M_i = (M_i^{(1)}, M_i^{(2)}, M_i^{(3)})$, where $|M_i^{(1)}| = |M_i^{(2)}| = m$, $|M_i^{(3)}| = 1$. By definition of h_k we may compute z_a,…,z_b and z'_a,…,z'_b as follows:

$$\begin{aligned} z_a &= k &&\oplus M^{(2)}_{\nu(a)} & \qquad z'_a &= k &&\oplus M^{(2)}_{\nu(a)} \\ z_{a+1} &= z_a &&\oplus M^{(2)}_{\nu(a+1)} & z'_{a+1} &= z'_a &&\oplus M^{(2)}_{\nu(a+1)} \\ &\ldots & & & &\ldots \\ z_b &= z_{b-1} &&\oplus M^{(2)}_{\nu(b)} & z'_b &= z'_{b-1} &&\oplus M^{(2)}_{\nu(b)}. \end{aligned}$$

Therefore,

$$z_b = z'_b = k \oplus M^{(2)}_{\nu(a)} \oplus M^{(2)}_{\nu(a+1)} \cdots \oplus M^{(2)}_{\nu(b)}.$$

Since every mask appears between a and b an even number of times, all masks XOR themselves out and

$$z_b = z'_b = k.$$

If $C_i(x)$ and $C_i(x')$ are the i^{th} chaining variable of the composite scheme evaluated on x and x',

$$C_b(x) = h_k(y_b, k, b_b, x_b) = 0^{2m+1},$$
$$C_b(x') = h_k(y'_b, k, b'_b, x'_b) = 0^{2m+1}$$

by the third case of the definition of h_k.

Since x and x' agree after their a^{th} component, the output of the composite scheme will be the same on both inputs. □(Lemma 1)

Lemma 2. *If schedule is even-free, then $r < 2^L$.*

Proof. Assume the opposite. There is an even-free schedule ν with $r \geq 2^L$.

Let $\#_{a,b}(i)$ be a function that counts the number of appearances of the i^{th} mask between $\nu(a)$ to $\nu(b)$ inclusive.

Define a sequence of bit-vectors $d^i = (\#_{1,i}(0) \bmod 2, \ldots, \#_{1,i}(L-1) \bmod 2)$. Each vector has length L. Because the schedule is even-free, none of these vectors is $(0, \ldots, 0)$. Therefore, there are $r \geq 2^L$ L-bit vectors and one of 2^L possible values is not available. By the pigeonhole principle there are two equal vectors $d^a = d^b$ among them. Consider their difference.

$$\begin{aligned} d^b - d^a &= (0, \dots, 0) \\ &= ((\#_{1,b}(0) - \#_{1,a}(0)) \bmod 2, \dots, (\#_{1,b}(L-1) - \#_{1,a}(L-1)) \bmod 2) \\ &= (\#_{a+1,b}(0) \bmod 2, \dots, \#_{a+1,b}(L-1) \bmod 2). \end{aligned}$$

Because of the interval $[a+1, b]$ the schedule function is not even-free. □(Lemma 2)

From lemmas 1 and 2 the theorem follows. □

Note 3. It is instructive to see why the Shoup scheduling $\nu(i) = \max\{j : 2^j | i\}$ is even-free. In any interval $[a, b]$ there is a *unique* element c that maximizes ν. Indeed, if there were two such elements c_1 and c_2, necessarily $\nu(c_1) = \nu(c_2)$. But then the element $c = (c_1 + c_2)/2$ would be divisible by a higher power of 2. Existence of an element that appears only once, i.e., an odd number of times, in every interval is enough for even-freeness.

Note 4. What if one uses addition modulo 2^m instead of XOR to mingle a mask and a chaining variable? If this operation is commutative and has an efficiently computable inverse, then the proof goes through with minor modifications. Having an inverse is required for our proof of the Shoup construction (below, Theorem 5), but being commutative is not necessary.

4.3 Correctness of the Shoup Construction

In this section we give an alternative proof of the Shoup construction. It is different from [Sh00] in presentation of the key reconstruction algorithm.

Theorem 5. *If* $\{h_k\}_{k \in K}$ *is a UOWHF, so is* $\{\mathbf{S}^{r,M} h_k\}_{k \in K, |M| = m(\lfloor \log r \rfloor + 1)}$ *for* $r < \mathrm{poly}(n)$.

Proof. Suppose that there is an adversary $\mathcal{A}$ that finds a collision of $\mathbf{S}^{r,M} h_k$ with a non-negligible probability over the key of the composite scheme. We build an algorithm $\mathcal{B}$ that finds a collision in $\{h_k\}_{k \in K}$. Let $S_j(x) = C_{j-1} \oplus M_{\nu(j)}$—first m bits of the input of h_k on the j^{th} iteration of the scheme on input x.

Algorithm $\mathcal{B}$

1. Run $\mathcal{A}$. $\mathcal{A}$ commits to some $x = (x_1, \dots, x_r)$.
2. Choose randomly j from $\{1, \dots, r\}$ and an m-bit string $C \in \{0,1\}^m$. Commit to $C || x_j$.
3. Receive a key $k \in K$.
4. Run the key reconstruction algorithm (described below) that will output M such that $S_j(x) = C$.
5. Feed k and M to $\mathcal{A}$.

6. If $\mathcal{A}$ finds a collision $x' = (x'_1, \ldots, x'_r)$, check if $S_j(x)||x_j \neq S_j(x')||x'_j$ and $h_k(S_j(x)||x_j) = h_k(S_j(x')||x'_j)$. If so, output $S_j(x')||x'_j$ that collides with $C||x_j$.

Suppose that the output of the key reconstruction algorithm on uniformly distributed C with fixed x, j and k also has the uniform distribution. Then the probability that $\mathcal{B}$ finds a collision is $1/r$ of the success probability of $\mathcal{A}$. Indeed, if $\mathcal{A}$ finds a collision, there is at least one $i \in \{1, \ldots, r\}$ such that $S_i(x)||x_i \neq S_i(x')||x'_i$ but $h_k(S_i(x)||x_i) = h_k(S_i(x')||x'_i)$ (consider the output of each iteration of the scheme going backward). This contradicts the assumption that $\{h_k\}_{k \in K}$ is a UOWHF. From this the claim of the theorem follows.

Now all we need is to show and prove the key reconstruction algorithm.

Key reconstruction algorithm

Input: $x = (x_1, \ldots, x_r)$, $k \in K$, $C \in \{0,1\}^m$.

Output: $M = (M_0, \ldots, M_{L-1})$, such that $S_j(x) = C$.

1. Label all masks $M_0, \ldots, M_{L-1}$ as "undefined."
2. Repeat the following steps while $j > 0$. If $j = 0$, randomly define all undefined masks and quit.
3. Let $i = j - 2^{\nu(j)}$.
4. Pick D at random from $\{0,1\}^m$.
5. Randomly define all yet undefined masks from the list $M_{\nu(i+1)}, \ldots, M_{\nu(j-1)}$.
6. If $i = 0$, let $C_0 = \mathrm{IV}$, otherwise let $C_i = h_k(D, x_i)$. Compute $C_{i+1} = h_k(M_{\nu(i+1)} \oplus C_i, x_{i+1})$,…,$C_{j-1} = h_k(M_{\nu(j-1)} \oplus C_{j-2}, x_{j-1})$.
7. Let $M_{\nu(j)} = C_{j-1} \oplus C$.
8. Assign $C \leftarrow D$, $j \leftarrow i$ and go to step 2.

First, note two invariants of the algorithm.

Invariant 1. $\nu(i) > \nu(j)$.

Invariant 2. $\nu(j) > \nu(l)$ for any $i < l < j$.

Both invariants follow from the fact that $j \equiv 2^{\nu(j)} \pmod{2^{\nu(j)+1}}$.

To prove the correctness of the algorithm we need to show that a mask is never *redefined*. Masks are defined in three steps of the algorithm. In steps 2 and 5 only undefined masks are assigned random values. By Invariant 2 their numbers are less than $\nu(j)$. In step 7 mask $M_{\nu(j)}$ is defined. Because $\nu(j)$ always increases (by Invariant 1) and masks that have been defined have numbers less than $\nu(j)$, before execution of this step $M_{\nu(j)}$ was not defined.

Since $C_i, \ldots, C_{j-1}$ computed in step 6 of the algorithm are indeed the values of the corresponding chaining variables, $S_j(x) = C_{j-1} \oplus M_{\nu(j)} = C_{j-1} \oplus C_{j-1} \oplus C = C$ as required. It completes our proof of correctness of the key reconstruction algorithm.

As the last step toward the proof of the theorem we have to show that the bit-vectors M output by the algorithm have the uniform distribution. Write down all the "decisions" (strings chosen at random) done during the execution of the algorithm (including its input C). It is a list of type C, $S_{j_1}(x)$, $M_{\nu(j_2)}$, …, $M_{\nu(j_3)}$, $S_{j_4}(x)$, $M_{\nu(j_5)}$,…,$M_{\nu(j_6)}$,… that contains exactly $L+1$ m-bit string (because all strings are equally long, and every mask must be defined in steps 2, 5, or 7).

Since the algorithm never stops without yielding a result, there is a mapping from the set of these strings into the set of possible outputs, which has the same cardinality. This mapping is an injection, because a preimage of an output value can be uniquely determined (it suffices to compute $S_j(x)$ for all corresponding j given M, k and x, which is trivial). Therefore, if the "decisions" are uniformly distributed, so are the outputs of the algorithm. $\square$

Shoup proved the security of the construction above. Our proof makes the key reconstruction algorithm more explicit and one-pass, thus giving a more efficient reduction, and requires less from the function ν (we do not need Fact 2 in [Sh00]). The operation performed by the key reconstruction algorithm in step 2 brings in Theorem 5. The mask $M_{\nu(i)}$ for $i = j - 2^{\nu(j)}$ is the only mask that appears an odd number of times between $M_{\nu(j)}$ and its previous appearance (if there exists one) at $M_{\nu(j-2\cdot 2^{\nu(j)})} = M_{\nu(j)}$.

We stress that our result of optimality of the Shoup scheduling does not rule out existence of a composite scheme with a shorter key. Even more important, our result implies that there must be at least $1 + \lfloor \log r \rfloor$ *different* masks, but says nothing about their *independence*. However, the proof of validity of the Shoup construction does need full independence of masks. There is an apparent gap between these two proofs.

We may try to reduce the key length by letting the masks be the output of a pseudo-random generator initialized with a short seed. Unfortunately, once the seed is exposed we cannot suppose anything about the output of the generator unless we resort to the random-oracle model. But in this model one could assume existence of CRHFs in the first place and our construction would be of no use in this world.

5 Conclusion and Open Problem

Recent attacks on MD4, MD5 and a flaw in the first version of SHA demonstrate that practical CRHFs are hard to construct. The oracle separation result due to [Si98] backed up this empirical fact by proving that CRHFs cannot be constructed from one-way permutations. Though UOWHFs, an alternative to CRHFs, have been known for years, their deployment in practical cryptosystems was hindered by lack of efficient composite schemes. While a family of CRHFs can be based on a single compression function, similar constructions for UOWHFs can only yield families of functions with variable key length. A variable key-length hash function stands out from all cryptographic primitives we use in practice and this annoying property can propagate to higher levels of construction (see [SS00] for an example).

We may approach this problem from two directions. First, it is possible that there exists a class of functions that are weaker than CRHFs, at least as strong as UOWHFs and for which an efficient composite scheme exists. We introduce a continuum of function classes that lie between CRHFs and UOWHFs and characterized by the degree of freedom the adversary has in choosing one of the colliding elements. From the complexity-theoretic point of view the hierarchy almost collapses to two large classes. The Merkle-Damgård construction, which yields fixed length-key families of functions, applies to one class of functions and not to the other.

Another approach is to improve existing composite schemes for UOWHFs. We take the Shoup construction, which is the most efficient (key length-wise), and prove that the scheme is optimal in respect to its mask scheduling. We also give a simplified proof of the Shoup construction.

An open problem is whether there exists a lower bound on the key length of a family of UOWHFs built via a black-box construction out of one-way functions. The upper bound given by numerous schemes from [BR97,Sh00] is $O(\log n)$, where n is the length of the input to a particular function. Such a lower bound would complement the line of research of [KST99,GT00] on *efficiency* of black-box constructions for UOWHFs.

6 Acknowledgement

The author is grateful to Victor Shoup for motivational discussions on this problem and Michael Waidner for his hospitality rendered through IBM Zurich. Additional thanks go to Dan Boneh and anonymous referees for their valuable comments.

References

[B82] M. Blum, "Coin flipping by telephone," CRYPTO 81, pp. 11–15, 1981.

[BR97] M. Bellare, P. Rogaway, "Collision-resistant hashing: towards making UOWHFs practical," Proc. of CRYPTO 97, pp. 470–484, Full version of this paper is available from `http://www-cse.ucsd.edu/users/mihir/`, 1997.

[D89] I. Damgård, "A design principle for hash functions," Proc. of CRYPTO 89, pp. 416–427, 1989.

[GT00] R. Gennaro, L. Trevisan, "Lower bounds on the efficiency of generic cryptographic constructions," Proc. of FOCS'00, pp. 305–313, 2000.

[HILL99] J. Hastad, R. Impagliazzo, L. Levin, M. Luby,"A pseudo-random generator from any one-way function," SIAM J. Computing, 28(4):1364–1396, 1999.

[KST99] J.H. Kim, D. Simon, P. Tetali, "Limits on the efficiency of one-way permutation-based hash functions," Proc. of FOCS'99, pp. 535–542, 1999.

[LM92] X. Lai, J. Massey, "Hash function based on block ciphers," Proc. of EUROCRYPT 92, pp. 55–70, 1992.

[M89] R. Merkle, "One way hash functions and DES," Proc. of CRYPTO 89, pp. 428–446, 1989.

[N91] M. Naor, "Bit commitment using pseudorandomness," J. Cryptology, 4(2): 151–158, 1991.

[NY89] M. Naor, M. Yung, "Universal one-way hash functions and their cryptographic applications," Proc. of STOC'89, pp. 33–43, 1989.

[Ro90] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," Proc. of STOC'90, pp. 387–394, 1990.

[Ru95] A. Russell, "Necessary and sufficient condtions for collision-free hashing," J. of Cryptology 8(2), pp. 87–100, 1995.

[SS00] T. Schweinberger, V. Shoup, "ACE: The Advanced Cryptographic Engine," Manuscript. Available from `http://www.shoup.net`, 2000.

[Sh00] V. Shoup, "A composite theorem for universal one-way hash functions," Proc. of EUROCRYPT 2000, pp. 445–452, 2000.

[Si98] D. Simon, "Finding collisions on a one-way street: Can secure hash functions be based on general assumptions?" Proc. of EUROCRYPT 98, pp. 334-345, 1998.

[ZMI90] Y. Zheng, T. Matsumoto, H. Imai, "Structural properties of one-way hash functions," Proc. of Crypto 90, pp. 285–302, 1990.