Structural Properties of One-Way Hash Functions

Yuliang Zheng
Tsutomu Matsumoto
Hideki Imai

Division of Electrical and Computer Engineering
Yokohama National University
156 Tokiwadai, Hodogaya, Yokohama, 240 JAPAN

Abstract

We study the following two kinds of one-way hash functions: *universal one-way hash functions* (UOHs) and *collision intractable hash functions* (CIHs). The main property of the former is that *given an initial-string* x, it is computationally difficult to find a different string y that collides with x. And the main property of the latter is that it is computationally difficult to find a pair $x \neq y$ of strings such that x collides with y. Our main results are as follows. First we prove that UOHs with respect to initial-strings chosen *arbitrarily* exist if and only if UOHs with respect to initial-strings chosen *uniformly at random* exist. Then, as an application of the result, we show that UOHs with respect to initial-strings chosen arbitrarily can be constructed under a weaker assumption, the existence of one-way *quasi*-injections. Finally, we investigate relationships among various versions of one-way hash functions. We prove that some versions of one-way hash functions are strictly included in others by explicitly constructing hash functions that are one-way in the sense of the former but not in the sense of the latter.

1 Introduction

One-way hash functions are a principal primitive in cryptography. There are roughly two kinds of one-way hash functions: *universal one-way hash functions* (UOHs) and *collision intractable hash functions* (CIHs). The main property of the former is that *given an initial-string* x, it is computationally difficult to find a different string y

that collides with x. And the main property of the latter is that it is computationally difficult to find a pair $x \neq y$ of strings such that x collides with y. Naor and Yung constructed UOHs under the assumption of the existence of one-way injections (i.e., one-way one-to-one functions) [NY89], and Damgård constructed CIHs under a stronger assumption, the existence of *claw-free pairs of permutations* [Dam89]. In [NY89], Naor and Yung also presented a general method for transforming any UOH into a secure digital signature scheme. We are interested both in constructing UOHs under weaker assumptions and in relationships among various versions of one-way hash functions. Our main results are summarized as follows.

First, we prove that UOHs with respect to initial-strings chosen *uniformly at random* can be transformed into UOHs with respect to initial-strings chosen *arbitrarily*. Thus UOHs with respect to initial-strings chosen arbitrarily exist if and only if UOHs with respect to initial-strings chosen uniformly at random exist. The proof is constructive, and may significantly simplify the construction of UOHs with respect to initial-strings chosen arbitrarily, under the assumption of the existence of one-way functions. Then, as an application of the transformation result, we prove that UOHs with respect to initial-strings chosen arbitrarily can be constructed under a weaker assumption, the existence of one-way quasi-injections (whose definition is to be given in Section 5). Next, we investigate relationships among various versions of one-way hash functions. We show that some versions of one-way hash functions are strictly included in others by explicitly constructing hash functions that are one-way in the sense of the former but not in the sense of the latter. A simple method, which appears in [ZMI90], for constructing UOHs from one-way permutations whose (simultaneously) hard bits have been identified is described in Appendix.

2 Notation and Definitions

The set of all positive integers is denoted by $\mathbf{N}$. Let $\Sigma = \{0, 1\}$ be the alphabet we consider. For $n \in \mathbf{N}$, denote by Σ^n the set of all strings over Σ with length n, by Σ^* that of all finite length strings including the empty string, denoted by λ, over Σ, and by Σ^+ the set $\Sigma^* - \{\lambda\}$. The concatenation of two strings x, y is denoted by $x \diamond y$, or simply by xy if no confusion arises. The length of a string x is denoted by $|x|$, and the number of elements in a set S is denoted by $\sharp S$.

Let ℓ be a monotone increasing function from $\mathbf{N}$ to $\mathbf{N}$, and f a (total) function from D to R, where $D = \bigcup_n D_n, D_n \subseteq \Sigma^n$, and $R = \bigcup_n R_n, R_n \subseteq \Sigma^{\ell(n)}$. D is called the *domain*, and R the *range* of f. For simplicity of presentation, in this paper we always assume that $D_n = \Sigma^n$ and $R_n = \Sigma^{\ell(n)}$. Denote by f_n the restriction of f on Σ^n. We are concerned only with the case when the range of f_n is $\Sigma^{\ell(n)}$, i.e., f_n is a function from Σ^n to $\Sigma^{\ell(n)}$. f is an *injection* if each f_n is a one-to-one function, and is a *permutation* if each f_n is a one-to-one and onto function. f is (deterministic/probabilistic) *polynomial time computable* if there is a (deterministic/probabilistic) polynomial (in $|x|$) time algorithm (Turing machine) computing $f(x)$ for all $x \in D$. The composition of two functions f and g is defined as $f \circ g(x) = f(g(x))$. In particular, the i-fold composition of f is denoted by $f^{(i)}$.

A (probability) *ensemble* E with length $\ell(n)$ is a family of *probability distributions* $\{E_n | E_n : \Sigma^{\ell(n)} \to [0,1], n \in \mathbf{N}\}$. The *uniform ensemble* U with length $\ell(n)$ is the family of *uniform probability distributions* U_n, where each U_n is defined as $U_n(x) = 1/2^{\ell(n)}$ for all $x \in \Sigma^{\ell(n)}$. By $x \in_E \Sigma^{\ell(n)}$ we mean that x is randomly chosen from $\Sigma^{\ell(n)}$ according to E_n, and in particular, by $x \in_R S$ we mean that x is chosen from the set S uniformly at random. E is *samplable* if there is a (probabilistic) algorithm M that on input n outputs an $x \in_E \Sigma^{\ell(n)}$, and *polynomially samplable* if furthermore, the running time of M is polynomially bounded.

Now we introduce the notion for *one-way functions*, a topic that has received extensive research (see for examples [Yao82] [Wa88] [ILL89]).

Definition 1 *Let $f : D \to R$, where $D = \bigcup_n \Sigma^n$ and $R = \bigcup_n \Sigma^{\ell(n)}$, be a polynomial time computable function, and let E be an ensemble with length n. (1) f is* one-way with respect to *E if for each probabilistic polynomial time algorithm M, for each polynomial Q and for all sufficiently large n, $\Pr\{f_n(x) = f_n(M(f_n(x)))\} < 1/Q(n)$, when $x \in_E \Sigma^n$. (2) f is* one-way *if it is one-way with respect to the uniform ensemble U with length n.*

There are two basic computation models: Turing machines and combinational circuits (see for examples [Pip79] [KL82] [BDG88]). The above definition for one-way functions is with respect to the Turing machine model. A stronger version of one-way functions that is with respect to the circuit model can be obtained by changing algorithms M in the above definition to families $M = \{M_n \mid n \in \mathbf{N}\}$ of polynomial size circuits.

3 Universal One-Way Hash Functions

The central concept treated in this paper is *one-way hash functions.* Two kinds of one-way hash functions have been considered in the literature: *universal one-way hash functions* and *collision-intractable hash functions* (or shortly UOHs and CIHs, respectively). In [Mer89] the former is called *weakly* and the latter *strongly*, one-way hash functions respectively. Naor and Yung gave a formal definition for UOH [NY89], and Damgård gave for CIH [Dam89]. In this section, a formal definition for UOH that is more general than that of [NY89] is given. We feel our formulation more reasonable. This will be explained after the formulation is introduced. CIH will be treated in later sections.

Let ℓ be a polynomial with $\ell(n) > n$, H be a family of functions defined by $H = \bigcup_n H_n$ where H_n is a (possibly multi-)set of functions from $\Sigma^{\ell(n)}$ to Σ^n. Call H a *hash function* compressing $\ell(n)$-bit input into n-bit output strings. For two strings $x, y \in \Sigma^{\ell(n)}$ with $x \neq y$, we say that x and y collide with each other under $h \in H_n$, or (x, y) is a collision pair for h, if $h(x) = h(y)$.

H is *polynomial time computable* if there is a polynomial (in n) time algorithm computing all $h \in H$, and *accessible* if there is a probabilistic polynomial time algorithm that on input $n \in \mathbf{N}$ outputs uniformly at random a description of $h \in H_n$. It is assumed that all hash functions considered in this paper are both polynomial time computable and accessible.

Let H be a hash function compressing $\ell(n)$-bit input into n-bit output strings, and E an ensemble with length $\ell(n)$. The definition for UOH is best described as a three-party game. The three parties are S (an *initial-string supplier*), G (a *hash function instance generator*) and F (a *collision-string finder*). S is an oracle whose power is un-limited, and both G and F are probabilistic polynomial time algorithms. The first move is taken by S, who outputs an *initial-string* $x \in_E \Sigma^{\ell(n)}$ and sends it to both G and F. The second move is taken by G, who chooses, independently of x, an $h \in_R H_n$ and sends it to F. The third and also final (null) move is taken by F, who on input $x \in \Sigma^{\ell(n)}$ and $h \in H_n$ outputs either "?" (I don't know) or a string $y \in \Sigma^{\ell(n)}$ such that $x \neq y$ and $h(x) = h(y)$. F wins a game iff his/her output is *not* equal to "?". Informally, H is a universal one-way hash function with respect to E if for any collision-string finder F, the probability that F wins a game is negligible. More precisely:

Definition 2 *Let H be a hash function compressing $\ell(n)$-bit input into n-bit output strings, P a collection of ensembles with length $\ell(n)$, and F a collision-string finder. H is a* universal one-way hash function with respect to P, *denoted by UOH/P, if for each $E \in P$, for each F, for each polynomial Q, and for all sufficiently large n,* $\Pr\{F(x, h) \neq ?\} < 1/Q(n)$, *where x and h are independently chosen from $\Sigma^{\ell(n)}$ and H_n according to E_n and to the uniform distribution over H_n respectively, and the probability* $\Pr\{F(x, h) \neq ?\}$ *is computed over $\Sigma^{\ell(n)}$, H_n and the sample space of all finite strings of coin flips that F could have tossed.*

If P consists of a single ensemble E (i.e., $P = \{E\}$), UOH/E is synonymous with UOH/P. Of particular interest are the following versions of UOH: (1) UOH/$EN[\ell]$, where $EN[\ell]$ is the collection of all ensembles with length $\ell(n)$. (2) UOH/$PSE[\ell]$, where $PSE[\ell]$ is the collection of all polynomially samplable ensembles with length $\ell(n)$. (3) UOH/U, where U is the uniform ensemble with length $\ell(n)$.

In [NY89], Naor and Yung gave a definition for UOH. They did not separate initial-string ensembles from collision-string finders. Instead, they introduced a probabilistic *polynomial* time algorithm $A(\cdot, \cdot)$, called a *collision adversary* that works in two stages: At the first stage, the algorithm A, on input (λ, λ) where λ denotes the empty string, outputs an *initial value* (corresponding to our *initial-string*) $x = A(\lambda, \lambda) \in \Sigma^{\ell(n)}$. At the second stage, it, when given an $h \in H_n$, attempts to find a string $y = A(x, h) \in \Sigma^{\ell(n)}$ such that $x \neq y$ and $h(x) = h(y)$.

Thus Naor and Yung defined, in our terms, *universal one-way hash function with respect to polynomially samplable ensembles with length* $\ell(n)$, i.e., UOH/$PSE[\ell]$. Naor and Yung constructed one-way hash functions in the sense of UOH/$PSE[\ell]$ under the assumption of the existence of one-way injections [NY89]. Note that they actually obtained a construction for one-way hash functions in the sense of UOH/$EN[\ell]$. In [ZMI90] we construct, in a different approach, one-way hash functions in the sense of UOH/$EN[\ell]$ under the assumption of the existence of one-way permutations. See Appendix for the description of the construction.

Separating initial-string ensembles from collision-string finders is conceptually much clearer, and enables us to reduce the problem of constructing one-way hash functions in the sense of UOH/$EN[\ell]$ (the "strongest" UOHs) to that of constructing one-way hash functions in the sense of UOH/U (the "weakest" UOHs). This topic is treated in Section 4.

The above definition for UOH is with respect to the Turing machine model. As a natural counterpart of UOH/P, where P is a set of ensembles with length $\ell(n)$, we have UOH$_C$/P, whose definition is obtained simply by changing probabilistic polynomial time algorithms F in Definition 2 to families $F = \{F_n \mid n \in \mathbf{N}\}$ of polynomial size circuits.

The definition for UOH can also be generalized in another direction: In addition to $x \in \Sigma^{\ell(n)}$ and $h \in H_n$, a collision-string finder F is allowed to receive an extra *advice* string a. As before, the output of F is either "?" or a string $y \in \Sigma^{\ell(n)}$ such that $x \neq y$ and $h(x) = h(y)$.

Definition 3 *Let H be a hash function compressing $\ell(n)$-bit input into n-bit output strings. H is a* universal one-way hash function with respect to polynomial length advice*, denoted by UOH/EN[poly], if for each pair (Q_1, Q_2) of polynomials with $Q_1(n) \geq \ell(n)$, for each ensemble E with length $Q_1(n)$, for each collision-string finder F, and for all sufficiently large n, $\Pr\{F(x, a, h) \neq ?\} < 1/Q_2(n)$, where $x \in \Sigma^{\ell(n)}$, $a \in \Sigma^{Q_1(n)-\ell(n)}$, $x \diamond a$ and h are independently chosen from $\Sigma^{Q_1(n)}$ and H_n according to E_n and to the uniform distribution over H_n respectively, and the probability $\Pr\{F(x, a, h) \neq ?\}$ is computed over $\Sigma^{Q_1(n)}$, H_n and the sample space of all finite strings of coin flips that F could have tossed.*

Notice the difference between *Turing machines taking advice* discussed in [Pip79] [KL82] and collision-string finders in our Definition 3. In the former case, advice strings are uniquely determined for each $n \in \mathbf{N}$. While in the latter case, they are generated probabilistically. In Section 7, we will discuss relationships among various versions of one-way hash functions including UOH/U, UOH/$PSE[\ell]$, UOH/$EN[\ell]$, UOH$_C$/$EN[\ell]$, and UOH/$EN[poly]$.

4 Transforming UOH/U into UOH/$EN[\ell]$

Let P_1, P_2 be collections of ensembles with length $\ell(n)$. We say that UOH/P_1 is *transformable* into UOH/P_2 iff given a one-way hash function H in the sense of UOH/P_1, we can construct from H a one-way hash function H' in the sense of UOH/P_2. The main result of this section is Theorem 1 to be proved below, which states that UOH/U is transformable into UOH/$EN[\ell]$. Thus constructing one-way hash functions in the sense of UOH/$EN[\ell]$ under certain assumptions can be fulfilled in two steps: At the first step, we construct one-way hash functions in the sense of UOH/U. This would be easier, since a uniform ensemble would be easier to handle than arbitrary ones. Then at the second step, we apply the proof technique for Theorem 1 to obtain one-way hash functions in the sense of UOH/$EN[\ell]$.

To prove Theorem 1, we require a function family called *an invertible uniformizer*. Let T_n be a set of permutations over $\Sigma^{\ell(n)}$, and let $T = \bigcup_n T_n$. T is a *uniformizer* with length $\ell(n)$ if it has the following properties 1, 2 and 3. Furthermore, F is *invertible* if it also has the following property 4.

1. For each n, for each pair of strings $x, y \in \Sigma^{\ell(n)}$, there are exactly $\sharp T_n / 2^{\ell(n)}$ permutations in T_n that map x to y.

2. There is a probabilistic polynomial time algorithm that on input n outputs a $t \in_R T_n$.

3. There is a polynomial time algorithm that computes all $t \in T$.

4. There is a polynomial time algorithm that computes t^{-1} for all $t \in T$.

The first property implies that for any $n \in \mathbf{N}$ and any $x \in \Sigma^{\ell(n)}$, when t is chosen randomly and uniformly from T_n, the probability that $t(x)$ coincides with a particular $y \in \Sigma^{\ell(n)}$ is $(\sharp T_n / 2^{\ell(n)}) / \sharp T_n = 1/2^{\ell(n)}$, i.e., $t(x)$ is distributed randomly and uniformly over $\Sigma^{\ell(n)}$.

Now we give a concrete invertible uniformizer with length $\ell(n)$. Note that there is a natural one-to-one correspondence between strings of $\Sigma^{\ell(n)}$ and elements of $GF(2^{\ell(n)})$. So we will not distinguish $GF(2^{\ell(n)})$ from $\Sigma^{\ell(n)}$. Let a and b be elements of $GF(2^{\ell(n)})$ with $a \neq 0$. Then the affine transformation t defined by $t(x) = a \cdot x + b$ is a permutation over $GF(2^{\ell(n)})$, where $\cdot$ and $+$ are multiplication and addition over $GF(2^{\ell(n)})$ respectively. Denote by T_n the set of all the affine transformations on $GF(2^{\ell(n)})$ defined as above. Clearly, $\sharp T_n = 2^{\ell(n)}(2^{\ell(n)} - 1)$, and for any elements $x, y \in GF(2^{\ell(n)})$, there are exactly $(2^{\ell(n)} - 1) = \sharp T_n / 2^{\ell(n)}$ affine transformations in T_n that map x to

y. In addition, generating $t \in_R T_n$ is easy, and for all $t \in T$, computing t and t^{-1} are simple tasks. Thus $T = \bigcup_n T_n$ is an invertible uniformizer with length $\ell(n)$. In section 5, T will once again play a crucial role in constructing one-way hash functions in the sense of UOH/$EN[\ell]$ from one-way quasi-injections. Now we are ready to prove the following:

Theorem 1 *UOH/U is transformable into UOH/$EN[\ell]$.* [1]

Proof : Assume that H is a one-way hash function in the sense of UOH/U, where U is the uniform ensemble with length $\ell(n)$. We show how to construct from H a hash function H' that is one-way in the sense of UOH/$EN[\ell]$.

Let $T = \bigcup_n T_n$ be an invertible uniformizer with length $\ell(n)$. Given H and $T = \bigcup_n T_n$, we construct H' as follows: $H' = \bigcup_n H'_n$, where $H'_n = \{h' \mid h' = h \circ t, h \in H_n, t \in T_n\}$. We claim that H' is one-way in the sense of UOH/$EN[\ell]$.

Assume for contradiction that H' is not one-way in the sense of UOH/$EN[\ell]$. Then there are a polynomial Q, an infinite subset $\mathbf{N}' \subseteq \mathbf{N}$, an ensemble E' with length $\ell(n)$ and a probabilistic polynomial time algorithm F' such that for all $n \in \mathbf{N}'$, the algorithm F', on input $x' \in_{E'} \Sigma^{\ell(n)}$ and $h' \in_R H'_n$, finds with probability $1/Q(n)$ a string $y' \in \Sigma^{\ell(n)}$ with $x' \neq y'$ and $h'(x') = h'(y')$. Now we show how to derive from F' a collision-string finder F that for all $n \in \mathbf{N}'$, on input $x \in_R \Sigma^{\ell(n)}$ and $h \in_R H_n$ where x is produced in a particular way to be described below, outputs with the same probability $1/Q(n)$ a string $y \in \Sigma^{\ell(n)}$ with $x \neq y$ and $h(x) = h(y)$.

Let M be a probabilistic Turing machine *with an oracle O that on input n outputs an $x' \in_{E'} \Sigma^{\ell(n)}$*. M produces $x \in_R \Sigma^{\ell(n)}$ in the following particular way:

1. Query the oracle O with n. Denote by x' the string answered by O. (Note that the oracle O is indispensable, as E' may be not samplable.)

2. Generate an $s \in_R T_n$ using its random tape.

3. Output $x = s(x')$.

From the first property of the uniformizer $T = \bigcup_n T_n$, we know that the ensemble E_M defined by the output of M is the uniform ensemble with length $\ell(n)$.

Let F be a probabilistic Turing machine. F uses the *same* random tape as M's and its read-only head for the random tape is in the same position as M's at the outset. On input $x \in_{E_M} \Sigma^{\ell(n)}$ and $h \in_R H_n$, (important note: since E_M is the uniform ensemble with length $\ell(n)$, $x \in_{E_M} \Sigma^{\ell(n)}$ is equivalent to $x \in_R \Sigma^{\ell(n)}$), F works as follows:

1. Generate a $t \in_R T_n$ using the random tape and in the same way as M does. Since M shares the random tape with F, we have $t = s$.

[1] De Santis and Yung obtained, independently, this theorem too [DY90].

2. Calculate $z = t^{-1}(x)$. Since $t = s$, we have $z = x' \in_{E'} \Sigma^{\ell(n)}$.

3. Call F' with input (z, h'), where $h' = h \circ t$. Note that $h' \in_R H'_n$, since $h \in_R H_n$ and $t \in_R T_n$.

4. Let $y' = F'(z, h')$. Output $y = y'$ whenever $y' = ?$, and $y = t(y')$ otherwise.

Since F' is polynomial time bounded, F is also polynomial time bounded. Furthermore, since t is a permutation over $\Sigma^{\ell(n)}$, we have $y \neq ?$ (i.e. $x \neq y$ and $h(x) = h(y)$) iff $y' \neq ?$ (i.e. $x' \neq y'$ and $h'(x') = h'(y')$). Thus for all $n \in \mathbf{N}'$, F outputs, with the same probability $1/Q(n)$, a string y such that $x \neq y$ and $h(x) = h(y)$, which implies that H is *not* a one-way hash function in the sense of UOH/U, a contradiction.

From the above discussions we know that H' is indeed a one-way hash function in the sense of UOH/$EN[\ell]$. This completes the proof. □

A significant corollary of Theorem 1 is:

Corollary 1 *One-way hash functions in the sense of UOH/$EN[\ell]$ exist iff those in the sense of UOH/U exist.*

5 UOHs Based on a Weakened Assumption

As an application of Theorem 1, in this section we construct one-way hash functions in the sense of UOH/$EN[\ell]$ under a weaker assumption — the existence of one-way *quasi*-injections. Main ingredients of our construction include (1) one-way quasi-injections, (2) universal hash functions with the collision accessibility property, (3) pair-wise independent uniformizers and, (4) invertible uniformizers. Our construction is partially inspired by [NY89].

5.1 Preliminaries

Assume that f is a one-way function from $\bigcup_n \Sigma^n$ to $\bigcup_n \Sigma^{\ell(n)}$. A string $x \in \Sigma^n$ is said to *have a brother* if there is a string $y \in \Sigma^n$ such that $f_n(x) = f_n(y)$.

Definition 4 *A one-way function f is a one-way* quasi-*injection iff for any polynomial Q and for all sufficiently large $n \in \mathbf{N}$, $\sharp B_n / 2^n < 1/Q(n)$ where B_n is the collection of all strings in Σ^n that have brothers.*

Let ℓ be a polynomial with $\ell(n) > n$, $S = \bigcup_n S_n$ be a hash function compressing $\ell(n)$-bit input into n-bit output strings. S is a *strongly universal*$_2$ hash function [CW79] [WC81] if for each n, for each pairs (x_1, x_2) and (y_1, y_2) with $x_1 \neq x_2$, $x_1, x_2 \in \Sigma^{\ell(n)}$ and $y_1, y_2 \in \Sigma^n$, there are $\sharp S_n / (\sharp \Sigma^n)^2$ functions in S_n that map x_1 to y_1 and x_2 to y_2. S is said to have the *collision accessibility property* [NY89] if given a pair

(x, y) of strings in $\Sigma^{\ell(n)}$ with $x \neq y$ and a requirement that $s(x) = s(y)$, it is possible to generate in polynomial time a function $s \in S_n$ such that $s(x) = s(y)$ with equal probability over all functions in S_n which obey the requirement. Note that strongly universal$_2$ hash functions with collision accessibility property are available without any assumption [NY89].

Let V_n be a set of permutations over $\Sigma^{\ell(n)}$, and $V = \bigcup_n V_n$. V is a *pair-wise independent uniformizer* with length $\ell(n)$ if it has the following three properties.

1. For each n, for any pairs of strings (x_1, x_2) and (y_1, y_2), there are exactly $\sharp V_n/[2^{\ell(n)}(2^{\ell(n)} - 1)]$ permutations in V_n that map x_1 to y_1 and x_2 to y_2, where $x_1, x_2, y_1, y_2 \in \Sigma^{\ell(n)}$, $x_1 \neq x_2$, $y_1 \neq y_2$, and $2^{\ell(n)}(2^{\ell(n)} - 1)$ is the total number of ordered pairs (x, y) with $x \neq y$ and $x, y \in \Sigma^{\ell(n)}$.

2. There is a probabilistic polynomial time algorithm that on input n outputs a $v \in_R V_n$.

3. There is a polynomial time algorithm that computes all $v \in V$.

Similar to uniformizers defined in Section 4, the first property implies that for any $n \in \mathbf{N}$ and any (x_1, x_2) with $x_1 \neq x_2$ and $x_1, x_2 \in \Sigma^{\ell(n)}$, when v is chosen randomly and uniformly from V_n, $(v(x_1), v(x_2))$ is distributed randomly and uniformly over all ordered pairs (y_1, y_2) with $y_1 \neq y_2$ and $y_1, y_2 \in \Sigma^{\ell(n)}$.

Recall the invertible uniformizer $T = \bigcup_n T_n$ constructed in Section 4. For any $x_1, x_2, y_1, y_2 \in \Sigma^{\ell(n)}$ with $x_1 \neq x_2$ and $y_1 \neq y_2$, there is exactly one permutation in T_n that maps x_1 to y_1 and x_2 to y_2. Note that $1 = 2^{\ell(n)}(2^{\ell(n)} - 1)/2^{\ell(n)}(2^{\ell(n)} - 1) = \sharp T_n/[2^{\ell(n)}(2^{\ell(n)} - 1)]$, which implies that T is a pair-wise independent uniformizer.

5.2 UOHs from One-Way Quasi-Injections

Assume that we are given a one-way quasi-injection f from D to R where $D = \bigcup_n \Sigma^n$, $R = \bigcup_n \Sigma^{m(n)}$ and m is a polynomial with $m(n) \geq n$. Let $V = \bigcup_n V_n$ be a pair-wise independent uniformizer with length $m(n)$, and $S = \bigcup_n S_n$ be a strongly universal$_2$ hash function that compresses $m(n)$-bit input into $(n-1)$-bit output strings and has the collision accessibility property.

Lemma 1 *let $H_n = \{h \mid h = s \circ v \circ f_{n+1}, s \in S_{n+1}, v \in V_{n+1}\}$, and $H = \bigcup_n H_n$. Then H is a one-way hash function in the sense of UOH/U compressing $(n+1)$-bit input into n-bit output strings, under the assumption that f is a one-way quasi-injection.*

Proof : Assume for contradiction that H is not one-way in the sense of UOH/U. Then there are a polynomial Q_1, an infinite subset $\mathbf{N}' \subseteq \mathbf{N}$ and a collision-string finder F such that for all $n \in \mathbf{N}'$, the finder F, on input $x \in_R \Sigma^{n+1}$ and $h \in_R H_n$, outputs with probability at least $1/Q_1(n)$ a string $y \in \Sigma^{n+1}$ with $x \neq y$ and $h(x) = h(y)$. We show that F can be used to construct an algorithm M that for all sufficiently large $n \in \mathbf{N}'$, inverts f_{n+1} with probability greater than $1/2Q_1(n)$.

Assume that $w \in_R \Sigma^{n+1}$ and $z = f_{n+1}(w)$. On input z, the algorithm M runs as follows in trying to compute a y such that $z = f_{n+1}(y)$:

Algorithm M:

1. Generate an $x \in_R \Sigma^{n+1}$. If $z = f_{n+1}(x)$ then output $y = x$ and halt. Otherwise execute the following steps.

2. Generate a $v \in_R V_{n+1}$.

3. Let $u_1 = v \circ f_{n+1}(x)$ and $u_2 = v(z)$. Choose a random $s \in S_{n+1}$ such that $s(u_1) = s(u_2)$. This is possible according to the collision accessibility property of S.

4. Let $h = s \circ v \circ f_{n+1}$. Call F with input h and x, and output $y = F(x, h)$.

First we show that h produced by M is a random element in H_n. At Step 2, a $v \in_R V_{n+1}$ is generated. Since $f_{n+1}(x) \neq z$, from the first property of V we know that $(v \circ f_{n+1}(x), v(z))$ is distributed randomly and uniformly over all pairs (x_1, x_2) with $x_1 \neq x_2$ and $x_1, x_2 \in \Sigma^{m(n+1)}$. At Step 3, s is chosen uniformly at random from all those functions in S_{n+1} that map u_1 and u_2 to the same string. Consequently, $h = s \circ v \circ f_{n+1}$ is a random element in H_n.

The running time of M is clearly polynomial in n. Next we estimate the probability that M outputs y such that $z = f_{n+1}(y)$. Denote by $\mathrm{Inv}(z)$ the set $\{e \mid z = f_{n+1}(e), e \in \Sigma^{n+1}\}$. Then M halts at Step 1 iff $x \in \mathrm{Inv}(z)$.

First we note that

$$\Pr\{z = f_{n+1}(y)\} \geq \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has no brother}, z = f_{n+1}(y)\},$$

where $\Pr\{z = f_{n+1}(y)\}$ is computed over Σ^{n+1}, Σ^{n+1}, V_{n+1}, S_{n+1} and the sample space of all finite strings of coin flips that F could have tossed. Note that the two compound events " $x \in \Sigma^{n+1} - \mathrm{Inv}(z)$, x has no brother, $z = f_{n+1}(y)$" and " $x \in \Sigma^{n+1} - \mathrm{Inv}(z)$, x has no brother, $y \neq ?$" are in fact the same. So the probability $\Pr\{z = f_{n+1}(y)\}$ can be estimated via the probability $\Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has no brother}, y \neq ?\}$. Now we focus on the latter. By assumption, we have $\Pr\{y \neq ?\} \geq 1/Q_1(n)$ for all $n \in \mathbf{N}'$, where $\Pr\{y \neq ?\}$ is computed over Σ^{n+1}, V_{n+1}, S_{n+1} and the sample space of all finite strings of coin flips that F could have tossed. On the other hand,

$$\begin{aligned}
\Pr\{y \neq ?\} &= \Pr\{x \in \mathrm{Inv}(z), y \neq ?\} + \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), y \neq ?\} \\
&= \Pr\{x \in \mathrm{Inv}(z), y \neq ?\} + \\
&\quad \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has a brother}, y \neq ?\} + \\
&\quad \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has no brother}, y \neq ?\}.
\end{aligned}$$

Recall that f is one-way. So for all sufficiently large $n \in \mathbf{N}$, we have

$$\Pr\{x \in \mathrm{Inv}(z), y \neq ?\} \leq \Pr\{x \in \mathrm{Inv}(z)\} < 1/4Q_1(n).$$

Furthermore, for all sufficiently n we have

$$\Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has a brother}, y \neq ?\} \leq \Pr\{x \text{ has a brother}\} < 1/4Q_1(n),$$

since f is a one-way quasi-injection. Thus for all sufficiently large $n \in \mathbf{N}'$,

$$\begin{aligned}
\Pr\{z = f_{n+1}(y)\} &\geq \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has no brother}, z = f_{n+1}(y)\} \\
&= \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has no brother}, y \neq ?\} \\
&\geq 1/Q_1(n) - [\Pr\{x \in \mathrm{Inv}(z), y \neq ?\} + \\
&\quad \Pr\{x \in \Sigma^{n+1} - \mathrm{Inv}(z), x \text{ has a brother}, y \neq ?\}] \\
&\geq 1/Q_1(n) - [1/4Q_1(n) + 1/4Q_1(n)] \\
&\geq 1/2Q_1(n).
\end{aligned}$$

This contradicts our assumption that f is a one-way quasi-injection, and hence the theorem follows. □

Combining Theorem 1 and Lemma 1, we have the following result: *A one-way hash function H' in the sense of UOH/$EN[\ell']$, where ℓ' is defined by $\ell'(n) = n + 1$, can be constructed under the assumption that f is a one-way quasi-injection.* By an argument analogous to that of Theorem 3.1 of [Dam89], it can be proved that for any polynomial ℓ, we can construct from H' a one-way hash function H'' in the sense of UOH/$EN[\ell]$. Thus:

Theorem 2 *One-way hash functions in the sense of UOH/$EN[\ell]$ can be constructed assuming the existence of one-way quasi-injections.*

Similarly, we can construct one-way hash functions in the sense of UOH$_C$/$EN[\ell]$ assuming the existence of one-way quasi-injections *with respect to the circuit model.*

6 Collision Intractable Hash Functions

This section gives formal definitions for collision intractable hash functions. Let $H = \bigcup_n H_n$ be a hash function compressing $\ell(n)$-bit input into n-bit output strings. Let A, a *collision-pair finder*, be a probabilistic polynomial time algorithm that on input $h \in H_n$ outputs either "?" or a pair of strings $x, y \in \Sigma^{\ell(n)}$ with $x \neq y$ and $h(x) = h(y)$.

Definition 5 *H is called a* collision-intractable hash function (CIH) *if for each A, for each polynomial Q, and for all sufficiently large n,* $\Pr\{A(h) \neq ?\} < 1/Q(n)$*, where $h \in_R H_n$, and the probability* $\Pr\{A(h) \neq ?\}$ *is computed over H_n and the sample space of all finite strings of coin flips that A could have tossed.*

In [Dam89] (see also [Dam87]) CIH is called *collision free function family*. Damgård obtained CIHs under the assumption of the existence of claw-free pairs of permutations. In [ZMI90], we show that CIHs can be constructed from *distinction-intractable permutations*. We also propose *practical* CIHs, the fastest of which compress nearly $2n$-bit long input into n-bit long output strings by applying only *twice* a one-way function.

CIH defined above are with respect to the Turing machine model. So as in the case for UOH, we have CIH$_C$ with respect to the circuit model. The definition for CIH$_C$ is similar to Definition 5, except that probabilistic polynomial time algorithms A are replaced by families $A = \{A_n \mid n \in \mathbf{N}\}$ of polynomial size circuits.

In addition, analogous to Definition 3, we have the following generalization for CIH. Let $H = \bigcup_n H_n$ be a hash function compressing $\ell(n)$-bit input into n-bit output strings, Q_1 a polynomial, and $a \in \Sigma^{Q_1(n)}$. a is called an *advice* string of length $Q_1(n)$. Let A, a collision-pair finder, be a probabilistic polynomial time algorithm that on input $a \in \Sigma^{Q_1(n)}$ and $h \in H_n$ outputs either "?" or a pair of strings $x, y \in \Sigma^{\ell(n)}$ with $x \neq y$ and $h(x) = h(y)$.

Definition 6 *H is called a* collision intractable hash function with respect to polynomial length advice*, denoted by CIH/$EN[poly]$, if for each pair (Q_1, Q_2) of polynomials, for each ensemble E with length $Q_1(n)$, for each A, and for all sufficiently large n,* $\Pr\{A(a,h) \neq ?\} < 1/Q_2(n)$*, where a and h are independently chosen from $\Sigma^{Q_1(n)}$ and H_n according to E_n and to the uniform distribution over H_n respectively, and the probability* $\Pr\{A(a,h) \neq ?\}$ *is computed over $\Sigma^{Q_1(n)}$, H_n and the sample space of all finite strings of coin flips that A could have tossed.*

7 A Hierarchy of One-Way Hash Functions

In this section, we discuss relationships among various versions of one-way hash functions: UOH/U, UOH/$PSE[\ell]$, UOH/$EN[\ell]$, UOH$_C$/$EN[\ell]$, UOH/$EN[poly]$, CIH, CIH$_C$ and CIH/$EN[poly]$.

First we define a relation between two versions, Ver_1 and Ver_2, of one-way hash functions. We say that

1. Ver_1 is *included in* Ver_2, denoted by $Ver_1 \subseteq Ver_2$, if all one-way hash functions in the sense of Ver_1 are also one-way hash functions in the sense of Ver_2.

2. Ver_1 is *strictly* included in Ver_2, denoted by $Ver_1 \subset Ver_2$, if $Ver_1 \subseteq Ver_2$ and there is a one-way hash function in the sense of Ver_2 but not in the sense of Ver_1.

3. Ver_1 and Ver_2 are *equivalent*, denoted by $Ver_1 = Ver_2$, if $Ver_1 \subseteq Ver_2$ and $Ver_2 \subseteq Ver_1$.

Lemma 2 *The following statements hold:*

(1) $CIH_C = CIH/EN[poly]$.

(2) $UOH_C/EN[\ell] = UOH/EN[poly]$.

(3) $UOH/EN[poly] \subseteq UOH/EN[\ell] \subseteq UOH/PSE[\ell] \subseteq UOH/U$.

(4) $CIH/EN[poly] \subseteq CIH$.

(5) $CIH \subseteq UOH/PSE[\ell]$.

(6) $CIH/EN[poly] \subseteq UOH/EN[poly]$.

Proof : Proofs for (1) and (2) are analogous to that for "polynomial size circuits vs. P/poly" [Pip79]. (3),(4), (5) and (6) are obvious. Here we give a detailed description for the proof of (1). Proof for (2) is similar, and is omitted.

The "$\subseteq$" part: Assume that H is a one-way hash function in the sense of CIH$_C$. If H is not one-way in the sense of CIH/$EN[poly]$, then there are polynomials Q_1 and Q_2, an infinite subset $\mathbf{N}' \subseteq \mathbf{N}$, an ensemble E with length $Q_2(n)$, and a collision-pair finder F, such that for all $n \in \mathbf{N}'$, the finder F, on input $z \in_E \Sigma^{Q_2(n)}$ and $h \in_R H_n$, outputs a collision-pair with probability $1/Q_1(n)$. Note that for each $n \in \mathbf{N}$ and $h \in_R H_n$, the probability that F successfully outputs a collision-pair is computed over $\Sigma^{Q_2(n)}$ and the sample space of all finite strings of coin flips that F could have tossed. Let $z_{\max}$ be the first string according to the lexicographic order in $\Sigma^{Q_2(n)}$ such that for $h \in_R H_n$, F outputs a collision-pair with the maximum probability, which is certainly at least $1/Q_1(n)$. F can be converted into a family $A = \{A_n \mid n \in \mathbf{N}\}$ of probabilistic polynomial size circuits with $z_{\max}$ being "embedded in" A_n. Thus for each $n \in \mathbf{N}'$, A_n on input $h \in_R H_n$ outputs a collision-pair with probability at least $1/Q_1(n)$. In other words, H is not one-way in the sense of CIH$_C$, which is a contradiction.

The "$\supseteq$" part: Assume that H is a one-way hash function in the sense of CIH/$EN[poly]$. If H is not one-way in the sense of CIH$_C$, then there are a polynomial Q_1, an infinite subset $\mathbf{N}' \subseteq \mathbf{N}$, and a collision-pair finder $A = \{A_n \mid n \in \mathbf{N}\}$, such that for all $n \in \mathbf{N}'$, A_n outputs a collision-pair with probability $1/Q_1(n)$. Since the size of A is polynomially bounded, there is a polynomial Q_2 such that the description of A_n is not longer than $Q_2(n)$ for all $n \in \mathbf{N}$. Without loss of generality, assume that the description of A_n is exactly $Q_2(n)$ bits long. Let E be the ensemble with length $Q_2(n)$ defined by $E_n(x) = 1$ whenever x is the description of A_n, and $E_n(x) = 0$ otherwise. Note that E may be not samplable.

Recall that the (probabilistic) *circuit value problem* is (probabilistic) polynomial time computable (see [BDG88], p.110). So there is a (probabilistic) polynomial time algorithm F that on input $z \in_E \Sigma^{Q_2(n)}$ and $h \in_R H_n$, (Note: By the definition of E, we have z=the description of A_n), output a collision-pair with probability $1/Q(n)$.

This implies that H is not one-way in the sense of CIH/$EN[poly]$, which contradicts our assumption. □

Theorem 3 *The following statements hold:*

(1) *UOH/$PSE[\ell]$ $\subset$ UOH/U.*

(2) *There are one-way hash functions in the sense of UOH/$EN[poly]$ but not in the sense of CIH.*

(3) *$CIH \subset$ UOH/$PSE[\ell]$.*

(4) *CIH/$EN[poly]$ $\subset$ UOH/$EN[poly]$.*

Proof : (1) We show that given a one-way hash function H in the sense of UOH/U, we can construct from H a hash function H' that is still one-way in the sense of UOH/U but not in the sense of UOH/$PSE[\ell]$.

H' is constructed as follows: Denote by $0^{\ell(n)}$ ($1^{\ell(n)}$, respectively) the all-0 (all-1, respectively) string of length $\ell(n)$. For each $h \in H_n$, define a function $h' : \Sigma^{\ell(n)} \rightarrow \Sigma^n$ by $h'(x) = h(0^{\ell(n)})$ whenever $x = 1^{\ell(n)}$ and $h'(x) = h(x)$ otherwise. Thus the only difference between h and h' is the images of $1^{\ell(n)}$. Let H'_n be the collection of all h', and let $H' = \bigcup_n H'_n$. We claim that H' is still one-way in the sense of UOH/U but not in the sense of UOH/$PSE[\ell]$.

Let M be a polynomial time algorithm that on input n outputs $1^{\ell(n)}$. By definition, the ensemble E defined by the output of M is polynomially samplable. Let F be a collision-string finder that on input x and h' outputs the string $0^{\ell(n)}$ whenever $x = 1^{\ell(n)}$ and "?" otherwise. Clearly, for all n, $x \in_E \Sigma^{\ell(n)}$ and $h' \in H'_n$, F always finds a string y that collides with x. Therefore H' is not one-way in the sense of UOH/$PSE[\ell]$.

Now we prove that H' is one-way in the sense of UOH/U. Assume for contradiction that H' is not one-way in the sense of UOH/U. Then there are an infinite subset $\mathbf{N}' \subseteq \mathbf{N}$ and a collision-string finder F such that for some polynomial Q and for all $n \in \mathbf{N}'$, $\Pr\{F(x, h') \neq ?\} \geq 1/Q(n)$, when $x \in_R \Sigma^{\ell(n)}$ and $h' \in_R H'_n$.

Note that

$$\begin{aligned}
&\Pr\{F(x,h') \neq ?\} \\
&\quad = \Pr\{F(x,h') \neq ? \mid h'(x) = h'(0^{\ell(n)})\} \cdot \Pr\{h'(x) = h'(0^{\ell(n)})\} + \\
&\quad\quad \Pr\{F(x,h') \neq ? \mid h'(x) \neq h'(0^{\ell(n)})\} \cdot \Pr\{h'(x) \neq h'(0^{\ell(n)})\} \\
&\quad \geq 1/Q(n),
\end{aligned}$$

and that

$$\begin{aligned}
&\Pr\{F(x,h') \neq ? \mid h'(x) = h'(0^{\ell(n)})\} \cdot \Pr\{h'(x) = h'(0^{\ell(n)})\} \\
&\leq \Pr\{h'(x) = h'(0^{\ell(n)})\} \\
&\leq \Pr\{h(x) = h(0^{\ell(n)})\} + 1/2^{\ell(n)} \\
&\leq 2\Pr\{h(x) = h(0^{\ell(n)})\}.
\end{aligned}$$

Since H is one-way in the sense of UOH/U, we have $\Pr\{h(x) = h(0^{\ell(n)})\} < 1/4Q(n)$ for all sufficiently large n. Thus for all sufficiently large $n \in \mathbf{N}'$,

$$\begin{aligned}
&\Pr\{F(x,h') \neq ? \mid h'(x) \neq h'(0^{\ell(n)})\} \\
&\geq \Pr\{F(x,h') \neq ? \mid h'(x) \neq h'(0^{\ell(n)})\} \cdot \Pr\{h'(x) \neq h'(0^{\ell(n)})\} \\
&\geq 1/Q(n) - \Pr\{F(x,h') \neq ? \mid h'(x) = h'(0^{\ell(n)})\} \cdot \Pr\{h'(x) = h'(0^{\ell(n)})\} \\
&> 1/2Q(n).
\end{aligned}$$

By definition, when $h'(x) \neq h'(0^{\ell(n)})$, a string $y \in \Sigma^{\ell(n)}$ with $x \neq y$ collides with x *under* h' iff it does *under* h. Consequently, the collision-string finder F can be used to "break" H, this implies that H is not one-way in the sense of UOH/U, a contradiction.

(2) The proof is very similar to that for (1). Given H, a one-way hash function in the sense of UOH/$EN[poly]$, we construct a hash function H' that is still one-way in the sense of UOH/$EN[poly]$ but not in the sense of CIH.

Without loss of generality, assume that the length of the description of $h \in H_n$ is greater than $n/2$, and for any distinct $h_1, h_2 \in H_n$ the first $n/2$ bits of h_1 is different from that of h_2. For each $h \in H_n$, we associate with it a particular $\ell(n)$-bit string x_h that is obtained by repeatedly concatenating the first $n/2$ bits of the description of h until the length of the resulting string becomes $\ell(n)$.

For each $h \in H_n$, define a function $h' : \Sigma^{\ell(n)} \to \Sigma^n$ by $h'(x) = h(x_h)$ whenever $x = \overline{x}_h$ and $h'(x) = h(x)$ otherwise, where $\overline{x}_h$ is the complement of x_h. Thus the only difference between h and h' is the images of $\overline{x}_h$. Let H'_n be the collection of all h', and let $H' = \bigcup_n H'_n$. By analyses similar to (1), one can verify that H' is still one-way in the sense of UOH/$EN[poly]$ but not in the sense of CIH.

(3) follows from (2) and $CIH \subseteq$ UOH/$PSE[\ell]$. (4) follows from (2) and the facts that CIH/$EN[poly] \subseteq CIH$ and that CIH/$EN[poly] \subseteq$ UOH/$EN[poly]$. □

From Lemma 2 and Theorem 3, we have the following hierarchical structure for one-way hash functions (see Figure 1.)

$$\begin{array}{ccc}
 & & \mathrm{UOH}/U \\
 & & \cup \\
\mathrm{CIH} & \subset & \mathrm{UOH}/PSE[\ell] \\
 & & |\cup \\
|\cup & & \mathrm{UOH}/EN[\ell] \\
 & & |\cup \\
\mathrm{CIH}/EN[poly] & \subset & \mathrm{UOH}/EN[poly] \\
\| & & \| \\
\mathrm{CIH}_C & & \mathrm{UOH}_C/EN[\ell]
\end{array}$$

Figure 1. Hierarchical Structure of One-Way Hash Functions

By Theorem 3, there are one-way hash functions in the sense of UOH/EN[*poly*] but not in the sense of CIH. However, it is not clear whether or not CIH $\subseteq$ UOH/EN[*poly*]. So it is worth while examining such problems as whether or not CIH is *strictly* included in UOH/EN[*poly*].

8 Conclusions

We have proved that UOHs with respect to initial-strings chosen uniformly at random can be transformed into UOHs with respect to initial-strings chosen arbitrarily, and that UOHs with respect to initial-strings chosen arbitrarily can be constructed under a weaker assumption, the existence of one-way quasi-injections. We have also investigated relationships among various versions of one-way hash functions. In particular, we have shown that UOH/$PSE[\ell]$, CIH and CIH/EN[*poly*] are strictly included in UOH/U, UOH/$PSE[\ell]$ and UOH/EN[*poly*] respectively, and that there are one-way hash functions in the sense of UOH/EN[*poly*] but not in the sense of CIH.

Recently, substantial progress on the *construction* of UOHs has been made by De Santis and Yung [DY90], and especially, by Rompel [Rom90] who finally solved the problem of constructing UOHs under the sole assumption of the existence of one-way functions.

Acknowledgments We would like to thank J. Leo, M. Ogiwara, K. Ohta and K. Sakurai for their fruitful discussions.

References

[BDG88] J. Balcázar, J. Díaz and J. Gabarró: *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin, 1988.

[CW79] J. Carter and M. Wegman: "Universal classes of hash functions", *Journal of Computer and System Sciences*, Vol.18, 1979, pp.143-154.

[Dam87] I. Damgård: "Collision free hash functions and public key signature schemes", *Proceedings of EuroCrypt'87*, 1987, pp.203-216.

[Dam89] I. Damgård: "A design principle for hash functions", *Presented at Crypto'89*, 1989.

[DY90] A. De Santis and M. Yung: "On the design of provably-secure cryptographic hash functions", *Presented at EuroCrypt'90*, 1990.

[ILL89] R. Impagliazzo, L. Levin and M. Luby: "Pseudo-random generation from one-way functions", *Proceedings of the 21-th ACM Symposium on Theory of Computing*, 1989, pp.12-24.

[KL82] R. Karp and R. Lipton: "Turing machines that take advice", *L'enseigment Mathematique*, Vol.28, 1982, pp.191-209.

[Mer89] R. Merkle: "One way hash functions and DES", *Presented at Crypto'89*, 1989.

[NY89] M. Naor and M. Yung: "Universal one-way hash functions and their cryptographic applications", *Proceedings of the 21-th ACM Symposium on Theory of Computing*, 1989, pp.33-43.

[Pip79] N. Pippenger: "On simultaneous resource bounds", *Proceedings of the 20-th IEEE Symposium on the Foundations of Computer Science*, 1979, pp.307-311.

[Rom90] J. Rompel: "One-way functions are necessary and sufficient for secure signatures", *Proceedings of the 22-nd ACM Symposium on Theory of Computing*, 1990, pp.387-394.

[Wa88] O. Watanabe: "On one-way functions", Presented at *the International Symposium on Combinatorial Optimization*, Tianjin, China, 1988.

[WC81] M. Wegman and J. Carter: "New hash functions and their use in authentication and set equality", *Journal of Computer and System Sciences*, Vol.22, 1981, pp.265-279.

[Yao82] A. Yao: "Theory and applications of trapdoor functions", *Proceedings of the 23-rd IEEE Symposium on the Foundations of Computer Science*, 1982, pp.80-91.

[ZMI90] Y. Zheng, T. Matsumoto and H. Imai: "Duality between two cryptographic primitives", To be presented at *8-th International Conference on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC-8)*, Tokyo, August 1990. A preliminary version appears in *IEICE Technical Reports on Information Security*, TG ISEC89-46, March 16, 1990.

A Appendix — UOHs from One-Way Permutations

In this appendix we sketch a simple method, which appears in [ZMI90], for constructing UOHs from one-way permutations whose (simutaneously) hard bits have been identified. An interesting feature of our construction is that *it does not apply universal hash functions*, and hence is extremely compact, in comparison with most of the currently known constructions.

Assume that f is a one-way permutation on $D = \bigcup_n \Sigma^n$, and that i has been proved to be a hard bit of f. For $b \in \Sigma$, $x \in \Sigma^{n-1}$ and $y \in \Sigma^n$, define $\mathrm{ins}(x, b) = x_{n-1}x_{i-2} \cdots x_i b x_{i-1} \cdots x_2 x_1$, and denote by $\mathrm{drop}(y)$ a function dropping the i-th bit of y. Then we have the following theorem.

Theorem 4 *Let ℓ be a polynomial with $\ell(n) > n$, $\alpha \in \Sigma^{n-1}$ and $x = x_{\ell(n)} \cdots x_2 x_1$ where $x_i \in \Sigma$ for each $1 \leq i \leq \ell(n)$. Let h_α be the function from $\Sigma^{\ell(n)}$ to Σ^n defined by:*

$$\begin{array}{rcl} y_0 & = & \alpha, \\ y_1 & = & \mathrm{drop}(f_n(\mathrm{ins}(y_0, x_{\ell(n)}))), \\ & \cdots & \\ y_j & = & \mathrm{drop}(f_n(\mathrm{ins}(y_{j-1}, x_{\ell(n)-j+1}))), \\ & \cdots & \\ h_\alpha(x) & = & f_n(\mathrm{ins}(y_{\ell(n)-1}, x_1)). \end{array}$$

Let $H_n = \{h_\alpha \mid \alpha \in \Sigma^{n-1}\}$ and $H = \bigcup_n H_n$. Then under the assumption that f is a one-way permutation, H is a UOH/EN$[\ell]$ compressing $\ell(n)$-bit input into n-bit output strings.

The efficiency of the above constructed UOHs can be improved by a factor of β, for any $\beta = \mathrm{O}(\log n)$, if β simultaneously hard bits of f have been identified.