Necessary and Sufficient Conditions for Collision-Free Hashing

Alexander Russell*
Laboratory for Computer Science
545 Technology Square
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
acr@theory.lcs.mit.edu

November 15, 1995

Abstract

This paper determines an exact relationship between collision-free hash functions and other cryptographic primitives. Namely, it introduces a new concept, the pseudo-permutation, and shows that the existence of collision-free hash functions is equivalent to the existence of claw-free pairs of pseudo-permutations. We also give a simple construction of collision-free hash functions from everywhere-defined claw-free (pseudo-) permutations.

1 Introduction

Hash functions with various cryptographic properties have been studied extensively, especially with respect to signing algorithms (see [2, 3, 4, 10, 12, 14, 15]). We focus on the most natural of these functions, the *collision-free* hash functions. A function h is a collision-free hash function if $|h(x)| \leq |x| - 1$ and it is infeasible, given h and 1^k, to find a pair (x, y) so that $|x| = |y| = k$ and $h(x) = h(y)$. These functions were first carefully studied by Damgård [2] and have found several applications. In particular, they have been used to improve the efficiency of digital signature schemes by hashing messages prior to signing (which reduces the size of the object operated upon the often costly signing algorithm). They have also been applied in the construction of efficient zero-knowledge arguments [9]. Given the interest in these functions, we would like to determine necessary and sufficient conditions for their existence in terms of other, simpler, cryptographic machinery.

There has been recent attention given to the minimal complexity-theoretic requirements for other cryptographic primitives. Rompel [12], improving a construction of Naor and Yung [10], shows that the existence of secure digital signing systems (in the sense of [5]) is equivalent to the existence of one-way functions. Impagliazzo, Levin, and Luby [7] and Håstad [6] demonstrate the equivalence of the existence of pseudo-random number generators (see [1, 13]) and the existence of one-way functions.

*Supported by a NSF Graduate Fellowship, NSF grant 92-12184, AFOSR F49620-92-J-0125, and DARPA N00014-92-J-1799

Damgård [2], distilling arguments of Goldwasser, Micali, and Rivest [5], shows that the existence of another cryptographic primitive, a *claw-free* pair of permutations, is sufficient to construct collision-free hash functions. A pair of permutations (f, g) of $\mathfrak{D} \subseteq \Sigma^*$ is *claw-free* if it is infeasible, given (f, g) and 1^k, to find a pair (x, y) so that $|x| = |y| = k$ and $f(x) = g(y)$. Comparing the definitions of collision-free hash functions and claw-free pairs of permutations, there is reason to suspect that the existence of claw-free pairs of permutations is not necessary for the existence of collision-free hash functions because the hash functions have no explicit structural properties that reflect the one-to-one property of the claw-free pairs of permutations. Our paper relaxes this one-to-one property and defines a natural object the existence of which is necessary and sufficient for the existence of a family of collision-free hash functions.

We define a new concept, the *pseudo-permutation.* A function $f : \mathfrak{D} \rightarrow \mathfrak{D}$ is a pseudo-permutation if it is computationally indistinguishable from a permutation. For this "indistinguishability" we require that it be infeasible, given the function f and 1^k, to compute a quickly verifiable proof of non-injectivity, i.e. a pair (x, y) where $|x| = |y| = k, x \neq y$, and $f(x) = f(y)$. The main contribution of our paper is that the existence of a collection of claw-free pairs of pseudo-permutations is equivalent to the existence of a collection of collision-free hash functions. This fact shows that claw-freedom of some variety is essential for collision-free hashing and also weakens the assumptions necessary for the existence of collision-free hash functions.

We also consider claw-free pairs of pseudo-permutations defined on all of Σ^* which we call claw-free pairs of *simple* pseudo-permutations. We show that that the existence of claw-free pairs of simple pseudo-permutations is also equivalent to the existence of collision-free hash functions.

Collision-free hash functions are suspected to be quite different from *universal one-way hash functions* [10]. A universal one-way hash function is an element of a family of functions $\{h_\alpha : \Sigma^n \rightarrow \Sigma^*\}$ such that $|h_\alpha(x)| \leq |x| - 1$ and it is infeasible to choose an element $x \in \Sigma^n$ so that, given h_α selected at random from $\{h_\alpha\}$, it is feasible to generate an element $y \in \Sigma^n$ so that $h_\alpha(x) = h_\alpha(y)$. Although it is easy to see that any collision-free hash function is a universal one-way hash function, it is unknown if collision-free hash functions can be constructed from universal one-way hash functions. These universal one-way hash functions were introduced because their existence is equivalent to the existence of secure digital signature schemes [10]. Rompel [12] then showed that the existence of these universal one-way hash functions is equivalent to the existence of one-way functions.

In §2 we describe our notation and define some cryptographic machinery. In §3 we present our main theorem. In §4 we present some comments on the main theorem and a dual theorem for simple functions. Finally, in §5, we conclude with an open problem and the motivation for this research.

2 Notation and Definitions

We adopt the following class of *expected polynomial-time Turing machines* as our standard class of "efficient algorithms."

Definition 1 *Let $\mathcal{EA}$, our class of efficient algorithms, be the class of probabilistic Turing machines (with output) running in expected polynomial time. We consider these machines, given an input, to compute a probability distribution over Σ^*. For $M \in \mathcal{EA}$ we use the notation $M[w]$ to denote both the probability space defined by M on w over Σ^* and an element selected according to this space.*

For simplicity, let us fix a two letter alphabet $\Sigma = \{0, 1\}$. We denote the empty string of Σ^* by Λ. 1^k denotes the concatenation of k 1's. For $x \in \Sigma^*$ of length n and for $i \leq n$, x_i denotes the ith character of x. $\mathbb{Z}[x]$ denotes the set of polynomials over the integers. For a function $f : \mathfrak{D} \rightarrow \mathfrak{R}$, we write $\mathbf{dom}\, f \stackrel{\text{def}}{=} \mathfrak{D}$ and $\mathbf{im}\, f \stackrel{\text{def}}{=} \{ f(x) \mid x \in \mathfrak{D} \} \subseteq \mathfrak{R}$. Borrowing notation from [5], if S is a probability space, $x \leftarrow S$ denotes the assignment of x according to S. If $p(x_1, \ldots, x_k)$ is a predicate, then $\Pr[x_1 \leftarrow S_1, \ldots, x_k \leftarrow S_k : p(x_1, \ldots, x_k)]$ denotes the probability that p will be true after the ordered assignment of x_1 through x_k. A collection of events $\{E_k\}$ is said to occur with *non-negligible* probability if $\exists P \in \mathbb{Z}[x], \forall k_0, \exists k > k_0$,

$$\Pr[E_k] \geq \frac{1}{P(k)}$$

2.1 Claw-free Pairs of Functions

Definition 2 *A* **collection of claw-free functions** *is a collection of function tuples $\{ (f_i^0, f_i^1) \mid i \in I \}$ for some index set $I \subseteq \Sigma^*$ where $f_i^j : \mathfrak{D}_i \rightarrow \mathfrak{D}_i$ for some $\mathfrak{D}_i \subseteq \Sigma^{|i|}$ such that:*

CF1. [accessable] *there exists a generating algorithm $G \in \mathcal{EA}$ so that $G[1^n] \in \{0,1\}^n \cap I$.*

CF2. [sampleable] *there exists a sampling algorithm S so that $S[i]$ is the uniform distribution on $\mathfrak{D}_i$.*

CF3. [efficiently evaluable] *there exists an evaluating algorithm $E \in \mathcal{EA}$ so that for $i \in I, j \in \{0, 1\}$, and $x \in \mathfrak{D}_i$, $E[i, j, x] = f_i^j(x)$.*

CF4. [claw-free] *for all claw finding algorithms $A \in \mathcal{EA}$, $\forall P \in \mathbb{Z}[x]$, $\exists k_0$, $\forall k > k_0$,*

$$\Pr\left[i \leftarrow G[1^k], (x, y) \leftarrow A[i] : f_i^0(x) = f_i^1(y)\right] < \frac{1}{P(k)}$$

A collection of such functions is called **simple** *if $\forall i \in I, \mathfrak{D}_i = \Sigma^{|i|}$.*

If (f^0, f^1) is a member of a collection of claw-free pairs, then (f^0, f^1) is called a *claw-free pair* and a pair (x, y) so that $f^0(x) = f^1(y)$ is called a *claw* of (f^0, f^1).

This definition, from a cryptographic perspective, requires nothing of the function pairs involved unless they have overlapping images. One way to require that the functions have overlapping images is to require that the functions be permutations. This yields the following object, originally defined in [5] and then in this form by [2].

Definition 3 *A* **collection of claw-free permutations** *is a collection of claw-free functions $\{ (f_i^0, f_i^1) \mid i \in I \}$ where each f_i^j is a permutation.*

Although the intractability of certain number theoretic problems implies the existence of a collection of claw-free pairs of permutations[1], the existence of one-way permutations is not known to be enough.[2]

Definition 4 *A* **collection of pseudo-permutations** *is a collection of functions* $\{ f_i \mid i \in I \}$ *for some index set* $I \subseteq \Sigma^*$ *where* $f_i : \mathfrak{D}_i \to \mathfrak{D}_i$ *for some* $\mathfrak{D}_i \subseteq \Sigma^{|i|}$ *such that:*

ψP1. [accessable] *there exists a generating algorithm* $G \in \mathcal{EA}$ *so that* $G[1^n] \in \{0,1\}^n \cap I$.

ψP2. [sampleable] *there exists a sampling algorithm* S *so that* $S[i]$ *is the uniform distribution on* $\mathfrak{D}_i$.

ψP3. [efficiently evaluable] *there exists a evaluation algorithm* $E \in \mathcal{EA}$ *so that for* $i \in I$ *and* $x \in \mathfrak{D}_i$, $E[i,x] = f_i(x)$.

ψP4. [collapse-free] *for all collapse finding algorithms* $A \in \mathcal{EA}$, $\forall P \in \mathbb{Z}[x]$, $\exists k_0$, $\forall k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x,y) \leftarrow A[i] : f_i(x) = f_i(y) \wedge x \neq y\right] < \frac{1}{P(k)}$$

A collection of such functions is called **simple** *if* $\forall i \in I, \mathfrak{D}_i = \Sigma^{|i|}$.

If a function f is a member of a collection of pseudo-permutations it is called a *pseudo-permutation* and a pair (x,y) where $f(x) = f(y)$ and $x \neq y$ is called a *collapse* of f. Property *ψP4* means that it is infeasible to produce a collapse of f (which may be thought of as a quickly verifiable proof that f is not a permutation). Like the definition for claw-free functions, the above definition requires nothing cryptographically of the functions involved unless $|\mathbf{im}\, f_i| < |\mathbf{dom}\, f_i|$: if the functions in the collection are injective, then *ψP4* is vacuously true.

Pseudo-permutations are a reasonable replacement for permutations in a cryptographic setting; for example, the entire signing algorithm of Naor and Yung [10] may be implemented with one-way[3] pseudo-permutations rather than one-way permutations.

Definition 5 *A* **collection of claw-free pseudo-permutations** *is a collection of claw-free functions* $\left\{ (f_i^0, f_i^1) \mid i \in I \right\}$ *so that both* $\left\{ f_i^0 \mid i \in I \right\}$ *and* $\left\{ f_i^1 \mid i \in I \right\}$ *are collections of pseudo-permutations. A collection of such functions is called* **simple** *if* $\forall i \in I, \mathfrak{D}_i = \Sigma^{|i|}$.

Collections of claw-free pseudo-permutations gather their cryptographic strength from the tension between two otherwise weak definitions. If the pseudo-permutations lack cryptographic richness (so that they are very close to permutations) then the intersection of their images must be large and there must be many claws, imparting richness by virtue of claw-freedom. If, instead, the pair has few claws, then the images of the two functions must be nearly disjoint (and so, small) so that the functions themselves are cryptographically rich by virtue of their many collapses.

[1] In [5] the intractability of factoring is shown to be sufficient. In [2], the construction of [5] is extended and the intractability of the discrete log is also shown to be sufficient.

[2] [11] discusses algebraic forms of one way permutations sufficient for claw-free permutations.

[3] This is a collection of pseudo-permutations which are hard to invert in the sense of one-way functions.

2.2 Collision-free Hash Functions

We now formally define collision-free hash functions. We will concentrate on *one-bit contractors*: hash functions from $\Sigma^{k+1} \to \Sigma^k$. It is not hard to show that by composition these collections of hash functions can be used to construct families of collision-free hash functions $\{ h_i : i \in I \}$ where $h_i : \Sigma^{P(|i|)} \to \Sigma^{|i|}$ for any polynomial $P \in \mathbb{Z}[x]$ where $\forall x \in \mathbb{N}^+, P(x) > x$.

Definition 6 *A* **collection of collision-free hash functions** *is a collection of functions* $\{ h_i \mid i \in I \}$ *for some index set* $I \subseteq \Sigma^*$ *where* $h_i : \Sigma^{|i|+1} \to \Sigma^{|i|}$ *and:*

H1. [accessible] *there exists a generating algorithm* $G \in \mathcal{EA}$ *so that* $G[1^n] \in \{0,1\}^n \cap I$.

H2. [efficiently evaluable] *there exists a evaluation algorithm* $E \in \mathcal{EA}$ *so that for* $i \in I$, *and* $w \in \Sigma^{|i|+1}, E[i, w] = h_i(w)$.

H3. [collision-free] *for all collision generating algorithms* $A \in \mathcal{EA}, \forall P \in \mathbb{Z}[x], \exists k_0, \forall k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x, y) \leftarrow A[i] : h_i(x) = h_i(y) \wedge x \neq y\right] < \frac{1}{P(k)}$$

If h is a member of a collection of collision-free hash functions then h is called a *collision-free hash function* and a pair (x, y) where $h(x) = h(y)$ and $x \neq y$ is called a *collision* of h.

3 Main Result

The notion of a polynomial separator will be used in the following proof. For the purposes of this paper, a separator is a pair of injections from Σ^k into Σ^{k+1} so that their images have no intersection. (Because $|\Sigma| = 2$, their images cover Σ^{k+1}.)

Definition 7 *A* **collection of polynomial separators** *is a collection of function pairs* $\{ (\sigma_i^0, \sigma_i^1) \mid i \in I \}$ *for some index set* $I \subseteq \Sigma^*$ *where* $\sigma_i^j : \Sigma^{|i|} \to \Sigma^{|i|+1}$ *for* $j \in \{0, 1\}$ *and:*

PS1. [accessible] *there exists a generating algorithm* $G \in \mathcal{EA}$ *so that* $G[1^n] \in \{0,1\}^n \cap I$.

PS2. [injective] σ_i^0 *and* σ_i^1 *are injective.*

PS3. [disjoint] $\mathbf{im}\, \sigma_i^0 \cap \mathbf{im}\, \sigma_i^1 = \emptyset$

PS4. [efficiently evaluable] *there exists an evaluating algorithm* $E \in \mathcal{EA}$ *so that for* $i \in I, w \in \Sigma^{|i|}$, *and* $j \in \{0, 1\}, E[i, j, w] = \sigma_i^j(w)$.

With each such collection, we associate a collection of inverses $\{ \iota_i \mid i \in I \}$ *where* $\iota_i : \Sigma^{|i|+1} \to \Sigma^{|i|}$ *and* $\iota_i \circ \sigma_i^0 = \iota_i \circ \sigma_i^1 = \mathbf{id}_{\Sigma^{|i|}}$ *and a collection of image deciders* $\{ \delta_i \mid i \in I \}$ *where* $\delta_i : \Sigma^{|i|+1} \to \{0, 1\}$ *and* $\forall w \in \Sigma^{|i|+1}, \delta(w) = j$ *iff* $w \in \mathbf{im}\, \sigma_i^j$.

The collection is said to have a **polynomial inverse** *if the collection of inverses is so that* $\exists E^{-1} \in \mathcal{EA}, \forall w \in \Sigma^{|i|+1}, \forall i \in I, E^{-1}[i, w] = \iota_i(w)$. *If a collection is so endowed, then it is clear that the image deciders may also be efficiently evaluated.*

Construction of a family of polynomial separators with a polynomial inverse is easy: the $append_0 : x \mapsto x0$ and $append_1 : x \mapsto x1$ functions, for example. Unless explicitly stated otherwise, wherever in this paper collections of such separators are required, it will sufficient to use these functions.

Theorem 1 *The following statements are equivalent:*

1. *There exists a collection of collision-free hash functions.*
2. *There exists a collection of claw-free pairs of simple pseudo-permutations.*
3. *There exists a collection of claw-free pairs of pseudo-permutations.*

Proof: Since we are particularly interested in the construction of collision-free hash functions, we arrange this proof in order to give two different constructions: one from claw-free pairs of simple pseudo-permutations ($2 \Rightarrow 1$) and one from arbitrary claw-free pairs of pseudo-permutations ($3 \Rightarrow 1$). The construction from the simple functions is simpler and more efficient. We begin by showing that $1 \Leftrightarrow 2$:

($1 \Rightarrow 2$) Let $\{ h_i \mid i \in I \}$ be a collection of collision-free hash functions. We construct a family of claw-free pairs of simple permutations. Let $\left\{ (\sigma_i^0, \sigma_i^1) \mid i \in I \right\}$ be a collection of polynomial separators (unrelated to the hash functions, but over the same index set). Define the collection $\left\{ (f_i^0, f_i^1) \mid i \in I \right\}$ so that

$$f_i^j \stackrel{\text{def}}{=} h_i \circ \sigma_i^j \text{ for } j \in \{0, 1\}$$

We show that the collection of functions so defined is a collection of claw-free pseudo-permutations. Properties *CF1*, *CF2*, and *CF3* are immediate. Assume that property *CF4* does not hold, that is $\exists A \in \mathcal{EA}, \exists P \in \mathbb{Z}[x], \forall k_0, \exists k > k_0$,

$$\Pr\left[i \leftarrow G[1^k], (x, y) \leftarrow A[i] : f_i^0(x) = f_i^1(y)\right] \geq \frac{1}{P(k)}$$

Let (x, y) be a claw for (f_i^0, f_i^1), then $f_i^0(x) = f_i^1(y)$ implies $h_i\left(\sigma_i^0(x)\right) = h_i\left(\sigma_i^1(y)\right)$, but $\mathbf{im}\, \sigma_i^0 \cap \mathbf{im}\, \sigma_i^1 = \emptyset$ so that $\sigma_i^0(x) \neq \sigma_i^1(y)$ and a collision has been found for h_i. Then, given this claw generating algorithm A we can construct a collision generating algorithm A' succeeding with identical probability as A, violating *H3*. Therefore, *CF4* holds.

To show that $\left\{ f_i^j \mid i \in I \right\}$ for each $j \in \{0, 1\}$ are collections of pseudo-permutations, we verify properties $\psi P1 - \psi P4$ for each. $\psi P1$, $\psi P2$, and $\psi P3$ are immediate. Suppose, for contradiction, that property $\psi P4$ is not satisfied, so that $(\exists j \in \{0,1\},)$ $\exists A \in \mathcal{EA}, \exists P \in \mathbb{Z}[x], \forall k_0, \exists k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x, y) \leftarrow A[i] : f_i^j(x) = f_i^j(y)\right] \geq \frac{1}{P(k)}$$

Let (x, y) be a collapse of f_i^j, so that $f_i^j(x) = f_i^j(y)$ and $x \neq y$. Then $\sigma_i^j(x) \neq \sigma_i^j(y)$ because σ_i^j is injective, so that $\left(\sigma_i^j(x), \sigma_i^j(y)\right)$ is a nontrivial collision of h_i (because

$f_i^j = h_i \circ \sigma_i^j$). Then, given this collapse generating algorithm A we can construct a collision generating algorithm A' succeeding with identical probability as A, violating *H3*. Therefore, $\psi P3$ holds.

(2 ⇒ 1) Let $\{ (f_i^0, f_i^1) \mid i \in I \}$ be a collection of claw-free pairs of simple pseudo-permutations. We construct a collection of collision-free hash functions. Let $\{ (\sigma_i^0, \sigma_i^1) \mid i \in I \}$ be a collection of polynomial separators with inverses $\{ \iota_i \mid i \in I \}$ and image deciders $\{ \delta_i \mid i \in I \}$. Then define $\{ h_i \mid i \in I \}$ so that

$$h_i(x) \stackrel{\text{def}}{=} f_i^{\delta_i(x)}(\iota_i(x))$$

We show that $\{ h_i \mid i \in I \}$ is a collection of collision-free hash functions. Properties *H1* and *H2* are immediate. Assume, for contradiction, that property *H3* is not satisfied, that is $\exists A \in \mathcal{EA}, \exists P \in \mathbb{Z}[x], \forall k_0, \exists k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x,y) \leftarrow A[i] : h_i(x) = h_i(y) \wedge x \neq y\right] \geq \frac{1}{P(k)}$$

We encounter at least one of two possibilities:

1. $\forall k_0, \exists k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x,y) \leftarrow A[i] : h_i(x) = h_i(y) \wedge x \neq y \wedge \delta_i(x) = \delta_i(y)\right] \geq \frac{1}{2P(k)}$$

2. $\forall k_0, \exists k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x,y) \leftarrow A[i] : h_i(x) = h_i(y) \wedge x \neq y \wedge \delta_i(x) \neq \delta_i(y)\right] \geq \frac{1}{2P(k)}$$

In the event of 1 above, the algorithm A generates collisions (x,y) where $\delta_i(x) = \delta_i(y)$. In this case, for at least one $j \in \{0,1\}, \forall k_0, \exists k > k_0$

$$\Pr\left[i \leftarrow G[1^k], (x,y) \leftarrow A[i] : h_i(x) = h_i(y) \wedge x \neq y \wedge j = \delta_i(x) = \delta_i(y)\right] \geq \frac{1}{4P(k)}$$

Given a collision of this sort, $x \neq y \Rightarrow \iota_i(x) \neq \iota_i(y)$ because ι_i is injective so that $h_i(x) = h_i(y)$ implies $f_i^j(\iota_i(x)) = f_i^j(\iota_i(y))$ shows that the pair $(\iota_i(x), \iota_i(y))$ is a collapse of f_i^j. Then, given algorithm A, we may produce another algorithm A' which produces a collapse of f_i^j with non-negligible probability, violating $\psi P3$.

In the event of 2 above, the algorithm A generates collisions (x,y) where $\delta_i(y) \neq \delta_i(x)$. A collision of this sort produces a claw because $h_i(x) = h_i(y)$ implies $f_i^{\delta_i(x)}(\iota_i(x)) = f_i^{\delta_i(y)}(\iota_i(y))$. Then, with algorithm A, we may construct a claw generating algorithm A' which produces claws with non-negligible probability, violating *CF3*.

To complete the proof we show that (2 ⇒ 3) and (3 ⇒ 1):

(2 ⇒ 3) A collection of claw-free pairs of simple permutations is a collection of claw-free pairs of permutations, so this implication is clear.

(3 ⇒ 1) This proof follows [2]. Let $\{\, (f_i^0, f_i^1) \mid i \in I \,\}$ be a collection of claw-free pairs of pseudo-permutations. We construct a collection of collision-free hash functions. Given these $f_i^j : \mathfrak{D}_i \to \mathfrak{D}_i$, define

$$\begin{aligned} f_i^{[\Lambda]} &\stackrel{\text{def}}{=} \mathbf{id}_{\mathfrak{D}_i} \\ f_i^{[b\circ w]} &\stackrel{\text{def}}{=} f_i^b \circ f_i^{[w]} \text{ for } b \in \{0,1\}, w \in \{0,1\}^* \end{aligned}$$

Let $\alpha \in \mathfrak{D}_i$. Define

$$H_{i,\alpha}(w) \stackrel{\text{def}}{=} f_i^{[w]}(\alpha)$$

We show that the set $\{\, H_{i,\alpha} \mid i \in I, \alpha \in \mathfrak{D}_i \,\}$ is a family of collision-free hash functions. Properties *H1* and *H2* are clear. Assume, for contradiction, that property *H3* does not hold so that there is a collision generating algorithm A so that $\exists P \in \mathbb{Z}[x], \forall k_0, \exists k > k_0$

$$\Pr\left[(i,\alpha) \leftarrow G[1^k], (x,y) \leftarrow A[i,\alpha] : h_{i,\alpha}(x) = h_{i,\alpha}(y) \wedge x \neq y\right] \geq \frac{1}{P(k)}$$

We partition the set of collisions into three varieties. Consider (x, y), a collision for $H_{i,\alpha}$, so that $f_i^{[x]}(\alpha) = f_i^{[y]}(\alpha)$. The first variety are those which never diverge:

V1. $\forall l \in \{1, \ldots, |i|+1\}, f_i^{[x_l \cdots x_{|i|+1}]}(\alpha) = f_i^{[y_l \cdots y_{|i|+1}]}(\alpha)$

In the event that a collision does not fall into variety *V1*, we must have that

$$\exists l \in \{1, \ldots, |i|+1\}, f_i^{[x_l \cdots x_{|i|+1}]}(\alpha) \neq f_i^{[y_l \cdots y_{|i|+1}]}(\alpha)$$

In this case, define μ to be the least member of $\{1, \ldots, |i|+1\}$ so that $f_i^{[x_\mu \cdots x_{|i|+1}]}(\alpha) \neq f_i^{[y_\mu \cdots y_{|i|+1}]}(\alpha)$ (since (x, y) is a collision, $f_i^{[x_{\mu-1} \cdots x_{|i|+1}]}(\alpha) = f_i^{[y_{\mu-1} \cdots y_{|i|+1}]}(\alpha)$). The last two varieties depend on $x_\mu \stackrel{?}{=} y_\mu$:

V2. *V1* has not occured and $y_\mu \neq x_\mu$

V3. *V1* has not occured and $y_\mu = x_\mu$

If a collision falls into variety *Vi*, we write $(x, y) \in Vi$. Since these varieties cover the space of collisions, for at least one variety, *Vi*, we have that $\forall k_0, \exists k > k_0$

$$\Pr\left[(i,\alpha) \leftarrow G[1^k], (x,y) \leftarrow A[i,\alpha] : h_{i,\alpha}(x) = h_{i,\alpha}(y) \wedge x \neq y \wedge (x,y) \in Vi\right] \geq \frac{1}{3P(k)}$$

We show that, regardless of which *Vi* has this property, either the claw-freedom of (f_i^0, f_i^1) or the collapse-freedom of f_i^0 or f_i^1 is compromised:

1. Suppose A produces collisions of variety *V1* with non-negligible probability. Let (x, y) be a collision of variety $V1$ so that $\forall l \in \{1, \ldots, |i|+1\}, f_i^{[x_l \cdots x_{|i|+1}]}(\alpha) = f_i^{[y_l \cdots y_{|i|+1}]}(\alpha)$. Choose p so that $x_p \neq y_p$. Since $(x, y) \in$ *V1*, we define $z = f_i^{[x_{p+1} \ldots x_{|i|+1}]}(\alpha) = f_i^{[y_{p+1} \ldots y_{|i|+1}]}(\alpha)$ and we have that $f_i^{x_p}(z) = f_i^{y_p}(z)$ so that (since $x_p \neq y_p$) (z, z) is a claw for f_i^0 and f_i^1. A may, then, be converted into an algorithm which produces claws for (f_i^0, f_i^1) with non-negligible probability, violating *CF4*.

2. Suppose A produces collisions of variety *V2* with non-negligible probability. Let (x, y) be a collision of variety *V2* and μ as above. Then define

$$(s,t) \stackrel{\text{def}}{=} \left(f_i^{[x_\mu \dots x_{|i|+1}]}(\alpha), f_i^{[y_\mu \dots y_{|i|+1}]}(\alpha) \right)$$

We have that $f_i^{x_{\mu-1}}(s) = f_i^{y_{\mu-1}}(t)$ so that (since $x_\mu \neq y_\mu$) (s,t) is a claw for (f_i^0, f_i^1). A may, then, be converted into an algorithm which produces claws for (f_i^0, f_i^1) with non-negligible probability, violating *CF4*.

3. Suppose A produces collisions of variety *V3* with non-negligible probability. Let (x, y) be a collision of variety *V3* and μ as above. Again define

$$(s,t) \stackrel{\text{def}}{=} \left(f_i^{[x_\mu \dots x_{|i|+1}]}(\alpha), f_i^{[y_\mu \dots y_{|i|+1}]}(\alpha) \right)$$

From the definition of μ we have that $s \neq t$ and $f_i^{x_{\mu-1}}(s) = f_i^{y_{\mu-1}}(t)$ so that (since $j \stackrel{\text{def}}{=} x_\mu = y_\mu$) the pair (s,t) is a collapse for f_i^j. A may, then, be converted into an algorithm which produces collapses for f_i^j with non-negligible probability, violating ψ*P4*.

Hence *CF4* is satisfied.

□

4 Comments

The construction of collision-free hash functions given in $(2 \Rightarrow 1)$ of Theorem 1, aside from its simplicity, has two favorable properties:

1. In order to compute $h_i(x)$, one must evaluate a claw-free function at only a single value. The construction in $(3 \Rightarrow 1)$ of Theorem 1 requires $|x|$ such evaluations.

2. Due to the simple construction, hash functions built in this fashion are likely to inherit structural properties from the underlying simple claw-free functions. For example, if the simple claw-free functions are *trapdoor* functions, it is easy to see that the hash functions constructed are also trapdoor in an appropriate sense[4]. It is unclear if the functions constructed in $(3 \Rightarrow 1)$ of Theorem 1 offer inheritance of this sort.

4.1 Extensions

In the constructions and discussions above, we have restricted our attention to one-bit contractors: hash functions which shorten their input by a single bit. It is often desirable to have hash functions which, for polynomial P, contract words of length $P(k)$ to words of length k. Such functions may, naturally, be constructed by composition of $P(k) - k$ one-bit

[4] In this framework, *trapdoor* means that there is a (probabilistic) polynomial-time algorithm which, given the trapdoor information, i, and $y \in \mathbf{im}\, h_i$ can, with non-negligible probability, produce $x \in \Sigma^{i+1}$ so that $h_i(x) = y$.

contractors. The hash functions constructed in this manner require $\sum_{i=k+1}^{P(k)} i$ evaluations of the underlying claw-free functions to compute (assuming that the one-bit contractors used are those of ($3 \Rightarrow 1$) of Theorem 1). As in [2], however, one can construct these hash functions directly to obtain a more efficient construction. For example, following the construction in ($3 \Rightarrow 1$) of Theorem 1, choose $|i| = k$ and again define $h_{i,\alpha}(x) = f_i^{[x]}(\alpha)$. The proof goes through as before and evaluating this hash function requires only $P(k)$ evaluation of the claw-free pairs (rather than the $\sum_{i=k+1}^{P(k)} i$ evaluations required by the hash function constructed by composition).

In [2], Damgård shows that by using claw-free *tuples* of functions (in the ($3 \Rightarrow 1$) construction) one can reduce the number of required claw-free function evaluations by a multiplicative constant factor. This is accomplished by rewriting the input string $x \in \Sigma^*$ as a string $\hat{x} \in T_i^*$ where $T_i = \{\tau_i^j\}$ is the tuple of claw-free functions so that $|\hat{x}| = \frac{|x|}{\log_2 |T|}$. Then define $h_{i,\alpha}(x) = \tau_i^{[\hat{x}]}(\alpha)$. Evaluation then requires $|\hat{x}| = \frac{|x|}{\log_2 |T|}$ claw-free evaluations. This same procedure is applicable to the construction of ($2 \Rightarrow 1$).

4.2 A Dual Result

A pair of separators partitions Σ^{k+1} into two equal sized subsets (the images of the separators). We now couple the definition of collision-free hash functions with the definition of polynomial separators to define a class of hash functions where every collision occurs across the partition boundary: whenever $h(x) = h(y)$ we have that x and y are in the images of different separators. By adding this artificial constraint to the collision-free hash functions, one can define a class of hash functions the existence of which is equivalent to the existence of simple claw-free permutations. We call these separated collision-free hash functions:

Definition 8 *A* **collection of separated collision-free hash functions** *is a collection of function tuples* $\{\, (h_i, \sigma_i^0, \sigma_i^1) \mid i \in I \,\}$ *so that* $\{\, h_i \mid i \in I \,\}$ *forms a collection of collision-free hash functions,* $\{\, (\sigma_i^0, \sigma_i^1) \mid i \in I \,\}$ *forms a collection of polynomial separators, and*

SH. [separated] $\forall j \in \{0,1\}$, $h_i \mid_{\mathbf{im}\, \sigma_i^j}$, *the restriction of* h_i *to* $\mathbf{im}\, \sigma_i^j$, *is bijective. Equivalently,* $h_i(x) = h_i(y)$ *implies* $\delta_i(x) \neq \delta_i(y)$, *where* $\{\, \delta_i \mid i \in I \,\}$ *is the collection of image deciders for the separators.*

Theorem 2 *There exists a collection of claw-free simple permutations iff there exists a collection of separated collision-free hash functions.*

Proof:

($\Rightarrow$) Notice that the construction of ($2 \Rightarrow 1$) of Theorem 1 yields hash functions with property *SH*, proving this implication also.

($\Leftarrow$) Use the separator supplied with the separated hash functions in the construction of ($1 \Rightarrow 2$) of Theorem 1 (which calls for an arbitrary separator). Property *SH* implies that the resulting claw-free functions are (simple) permutations.

□

Itoh [8] has pointed out that in Theorems 1 and 2 above, the requirement of claw-freedom can be replaced in an appropriate way with the requirement of *distinction intractability* as defined in [14].

5 Conclusion

The motivation for this research is the following open problem: is the existence of one-way functions sufficient for the existence of collision-free hash functions? This paper shows, at least, that answering this question in the affirmative need not imply the equivalence of one-way functions and one-way permutations. We hope that this presentation of machinery the existence of which is equivalent to the existence collision-free hash functions will aid the development toward complete understanding of these functions.

6 Acknowledgements

We gratefully acknowledge the keen guidance of Silvio Micali, who originally suggested this problem. We also acknowledge Ravi Sundaram for several helpful discussions. Finally, this paper has benefitted immensely from the comments of Oded Goldreich.

References

[1] Manual Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal of Computing*, 13(4):850–864, November 1984.

[2] Ivan Damgård. Collision free hash functions and public key signature schemes. In *Proceedings of EUROCRYPT '87*, volume 304 of *Lecture Notes in Computer Science*, pages 203–216, Berlin, 1988. Springer-Verlag.

[3] Alfredo De Santis and Moti Yung. On the design of provably-secure cryptographic hash functions. In *Proceedings of EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 412 – 431, Berlin, 1990. Springer-Verlag.

[4] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[5] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attack. *SIAM Journal of Computing*, 17(2):281–308, April 1988.

[6] J. Håstad. Pseudo-random generators under uniform assumptions. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 395–404. ACM, 1990.

[7] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 12–24. ACM, 1989.

[8] Toshiya Itoh. Personal comminucation, August 1992.

[9] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the Twenty Fourth Annual ACM Symposium on Theory of Computing*, pages 723–732. ACM, 1992.

[10] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 33–43. ACM, 1989.

[11] Wakaha Ogata and Kaoru Kurosawa. On claw free families. In *Proceedings of ASIACRYPT '91*, 1991.

[12] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 387–394. ACM, 1990.

[13] Andrew Yao. Theory and applications of trapdoor functions. In *Proceedings of the Twenty Third Symposium on Foundations of Computer Science*, pages 80–91. IEEE, 1982.

[14] Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. Duality between two cryptographic primitives. In *Proceedings of the Eighth International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 508 of *Lecture Notes in Computer Science*, pages 379–390, Berlin, 1990. Springer-Verlag.

[15] Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. Structural properties of one-way hash functions. In *Proceedings of CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 285–302, Berlin, 1990. Springer-Verlag.