An abridged version, reference [2], appears as *Advances in Cryptology – CRYPTO '02*, Lecture Notes in Computer Science, Vol. ????, M. Yung, ed., Springer-Verlag. This is the full version.

Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV

J. Black [*] P. Rogaway [†] T. Shrimpton [‡]

May 31, 2002

Abstract

Preneel, Govaerts, and Vandewalle [7] considered the 64 most basic ways to construct a hash function H: $\{0,1\}^* \to \{0,1\}^n$ from a block cipher E: $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. They regarded 12 of these 64 schemes as secure, though no proofs or formal claims were given. The remaining 52 schemes were shown to be subject to various attacks. Here we provide a formal and quantitative treatment of the 64 constructions considered by PGV. We prove that, in a black-box model, the 12 schemes that PGV singled out as secure really *are* secure: we give tight upper and lower bounds on their collision resistance. Furthermore, by stepping outside of the Merkle-Damgård approach to analysis, we show that an additional 8 of the 64 schemes are just as collision resistant (up to a small constant) as the first group of schemes. Nonetheless, we are able to differentiate among the 20 collision-resistant schemes by bounding their security as one-way functions. We suggest that proving black-box bounds, of the style given here, is a feasible and useful step for understanding the security of any block-cipher-based hash-function construction.

Key words: Block ciphers, cryptographic hash functions, modes of operation, proving security.

[*] Department of Computer Science/171, University of Nevada, Reno, Nevada, 89557, USA. E-mail: `jrblack@cs.colorado.edu` WWW: `www.cs.colorado.edu/~jrblack/`

[†] Department of Computer Science, University of California, Davis, California, 95616, USA; and Department of Computer Science, Faculty of Science, Chiang Mai University, 50200 Thailand. E-mail: `rogaway@cs.ucdavis.edu` WWW: `www.cs.ucdavis.edu/~rogaway/`

[‡] Department of Electrical and Computer Engineering, University of California, Davis, California, 95616, USA. E-mail: `teshrim@ucdavis.edu` WWW: `www.ece.ucdavis.edu/~teshrim/`

Contents

1 Introduction **1**

2 Definitions **6**

3 Collision Resistance of the Group-1 Schemes **8**

4 Collision Resistance of the Group-2 Schemes **9**

5 Matching Attacks on Collision Resistance **10**

6 Security of the Schemes as OWFs **12**

References **13**

A Fatal Attacks on Five of PGV's B-Labeled Schemes **14**

B Two Notions of Inversion Resistance **14**

C Proofs **15**

C.1 Proof of Lemma 3.2 15
C.2 Proof of Claim 4.2 17
C.3 Proof of Claim 4.3 17
C.4 Proof of Lemma 5.2 19
C.5 Proof of Theorem 5.3 22
C.6 Proof of Lemma 6.2 23
C.7 Proof of Theorem 6.4 23
C.8 Proof of Lemma 6.5 23
C.9 Proof of Theorem 6.6 24
C.10 Proof of Theorem 6.7 25
C.11 Proof of Theorem 6.8 25

1 Introduction

Background. The most popular collision-resistant hash-functions (eg., MD5 and SHA-1) iterate a compression function that is constructed from scratch (i.e., one that doesn't use any lower-level cryptographic primitive). But there is another well-known approach, going back to Rabin [8], wherein one makes the compression function out of a block cipher. This approach has been less widely used, for a variety of reasons. These include export restrictions on block ciphers, a preponderance of 64-bit block lengths, problems attributable to "weak keys", and the lack of popular block ciphers with per-byte speeds comparable to that of MD5 or SHA-1. Still, the emergence of the AES has somewhat modified this landscape, and now motivates renewed interest in finding good ways to turn a block cipher into a cryptographic hash function. This paper casts some fresh light on the topic.

The PGV paper. We return to some old work by Preneel, Govaerts, and Vandewalle [7] that considered turning a block cipher $E\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ into a hash function $H\colon (\{0,1\}^n)^* \to \{0,1\}^n$ using a compression function $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ derived from E. For v a fixed n-bit constant, PGV considers all 64 compression functions f of the form $f(h_{i-1}, m_i) = E_a(b) \oplus c$ where $a, b, c \in \{h_{i-1},\ m_i,\ h_{i-1} \oplus m_i,\ v\}$. Then define the *iterated hash of* f as:

function $H\,(m_1 \cdots m_\ell)$
for $i \leftarrow 1$ **to** ℓ **do** $h_i \leftarrow f\,(h_{i-1}, m_i)$
return h_ℓ

Here h_0 is a fixed constant, say 0^n, and $|m_i| = n$ for each $i \in [1..\ell]$. Of the 64 such schemes, the authors of [7] regard 12 as secure. Another 13 schemes they classify as *backward-attackable*, which means they are subject to an identified (but not very severe) potential attack. The remaining 39 schemes are subject to damaging attacks identified by [7] and others.

Some missing results. The authors of [7] focused on attacks, not proofs. All the same, it seems to be a commonly held belief that it should be possible to produce proofs for the schemes they regarded as secure. Indeed [7] goes so far as to say that "For each of these schemes it is possible to write a 'security proof' based on a black box model of the encryption algorithm, as was done for the Davies-Meyer scheme [by Winternitz [11]]". This latter paper uses a black-box model of a block cipher—a model dating back to Shannon [9]—to show that the scheme we will later call H_5 is secure in the sense of preimage-resistance. Specifically, [11] shows that any algorithm (with E and E^{-1} oracles) that always finds a preimage under H_5 for a fixed value $y \in \{0,1\}^n$ will necessarily make at least 2^{n-1} expected oracle queries.

The model introduced by Winternitz for analyzing block-cipher-based hash functions was subsequently used by Merkle [6]. He gives black-box model arguments for H_1, and other functions, and considers questions of efficiency and concrete security. The black-box model of a block cipher has also found use in other contexts, such as [4, 5]. But, prior to the current work, we are unaware of any careful analysis in the literature, under any formalized model, for the collision-resistance of any block-cipher-based hash-function.

Summary of our results. This paper takes a more proof-centric look at the schemes from PGV [7], providing both upper and lower bounds for each. Some of our results are as expected, while others are not.

First we prove collision-resistance for the 12 schemes singled out by PGV as secure (meaning those marked "✓" or "FP" in [7]). We analyze these *group-1* schemes, $\{H_1, \ldots, H_{12}\}$, within the Merkle-Damgård paradigm. That is, we show that for each group-1 scheme $H_\imath$ its compression function $f_\imath$ is already collision resistant, and so $H_\imath$ must be collision resistant as well.

PGV's backward-attackable schemes (marked "B" in [7]) held more surprises. We find that eight of these 13 schemes are secure, in the sense of collision resistance. In fact, these eight *group-2* schemes, $\{H_{13}, \ldots, H_{20}\}$, are just as collision-resistant as the group-1 schemes.

Despite having essentially the same collision-resistance, the group-1 and group-2 schemes can be distinguished based on their security as one-way functions: we get a better bound on inversion-resistance for the group-1 schemes than we get for the group-2 schemes. Matching attacks (up to a constant) demonstrate that this difference is genuine and not an artifact of the security proof.

The remaining $44 = 64 - 20$ hash functions considered by PGV are completely insecure: for these *group-3* schemes one can find a (guaranteed) collision with two or fewer queries. This includes five of PGV's backward-attackable schemes, where [7] had suggested a (less effective) meet-in-the-middle attack (see Appendix A).

Other surprises emerged in the mechanics of carrying out our analyses. Unlike the group-1 schemes, we found that the group-2 schemes could not be analyzed within the Merkle-Damgård paradigm; in particular, these schemes are collision resistant even though their compression functions are not. We also found that, for one set of schemes, the "obvious attack" on collision resistance needed some subtle probabilistic reasoning to rigorously analyze.

The security of the 64 PGV schemes is summarized in Figures 1 and 2, which also serve to define the different hash functions $H_\imath$ and their compression functions $f_\imath$. Figure 3 gives a more readable description of $f_1, \ldots, f_{20}$. A high-level summary of our findings is given by the following chart. The model (and the meaning of q) will be described momentarily.

<table>
<tr><th>PGV Category</th><th>Our Category</th><th>Collision Bound</th><th>OWF Bound</th></tr>
<tr><td>✓ or FP (12 schemes)</td><td>group-1: $H_{1..12}$ (12 schemes)</td><td>$\Theta(q^2/2^n)$</td><td>$\Theta(q/2^n)$</td></tr>
<tr><td rowspan="2">B (13 schemes)</td><td>group-2: $H_{13..20}$ (8 schemes)</td><td>$\Theta(q^2/2^n)$</td><td>$\Theta(q^2/2^n)$</td></tr>
<tr><td rowspan="2">group-3 (44 schemes)</td><td rowspan="2">$\Theta(1)$</td><td rowspan="2">$\Theta(1)$</td></tr>
<tr><td>F, P, or D (39 schemes)</td></tr>
</table>

Black-box model. Our model is the one dating to Shannon [9] and used for works like [4, 5, 11]. Fix a key-length κ and a block length n. An adversary A is given access to oracles E and E^{-1} where E is a random block cipher E: $\{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ and E^{-1} is its inverse. That is, each key $k \in \{0,1\}^\kappa$ names a randomly-selected permutation $E_k = E(k, \cdot)$ on $\{0,1\}^n$, and the adversary is given oracles E and E^{-1}. The latter, on input (k, y), returns the point x such that $E_k(x) = y$.

For a hash function H that depends on E, the adversary's job in attacking the collision resistance of H is to find distinct M, M' such that $H(M) = H(M')$. One measures the optimal adversary's chance of doing this as a function of the number of E or E^{-1} queries it makes. Similarly, the adversary's job in inverting H is to find an inverse under H for a random range point $Y \in \{0,1\}^n$. (See Section 2 for a justification of this definition.) One measures the optimal adversary's chance of doing this as a function of the total number of E or E^{-1} queries it makes.

$\imath$	$\jmath$	$h_i =$	**CR low-bnd**	**CR up-bnd**	**IR low-bnd**	**IR up-bnd**	
	1	$E_{m_i}(m_i) \oplus v$	1	1			a
	2	$E_{h_{i-1}}(m_i) \oplus v$	1	1			b
13	3	$E_{w_i}(m_i) \oplus v$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	4	$E_v(m_i) \oplus v$	1	1			a
	5	$E_{m_i}(m_i) \oplus m_i$	1	1			a
1	6	$E_{h_{i-1}}(m_i) \oplus m_i$	$.039(q-1)(q-3)/2^n$	$q(q+1)/2^n$	$0.4q/2^n$	$2q/2^n$	d
9	7	$E_{w_i}(m_i) \oplus mi$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	8	$E_v(m_i) \oplus m_i$	1	1			a
	9	$E_{m_i}(m_i) \oplus h_{i-1}$	1	1			f
	10	$E_{h_{i-1}}(m_i) \oplus h_{i-1}$	1	1			b
11	11	$E_{w_i}(m_i) \oplus h_{i-1}$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	12	$E_v(m_i) \oplus h_{i-1}$	1	1			b
	13	$E_{m_i}(m_i) \oplus w_i$	1	1			f
3	14	$E_{h_{i-1}}(m_i) \oplus w_i$	$.039(q-1)(q-3)/2^n$	$q(q+1)/2^n$	$0.4q/2^n$	$2q/2^n$	d
14	15	$E_{w_i}(m_i) \oplus w_i$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	16	$E_v(m_i) \oplus w_i$	1	1			f
15	17	$E_{m_i}(h_{i-1}) \oplus v$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	18	$E_{h_{i-1}}(h_{i-1}) \oplus v$	1	1			a
16	19	$E_{w_i}(h_{i-1}) \oplus v$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	20	$E_v(h_{i-1}) \oplus v$	1	1			a
17	21	$E_{m_i}(h_{i-1}) \oplus m_i$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	22	$E_{h_{i-1}}(h_{i-1}) \oplus m_i$	1	1			b
12	23	$E_{w_i}(h_{i-1}) \oplus m_i$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	24	$E_v(h_{i-1}) \oplus m_i$	1	1			b
5	25	$E_{m_i}(h_{i-1}) \oplus h_{i-1}$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	26	$E_{h_{i-1}}(h_{i-1}) \oplus h_{i-1}$	1	1			a
10	27	$E_{w_i}(h_{i-1}) \oplus h_{i-1}$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	28	$E_v(h_{i-1}) \oplus h_{i-1}$	1	1			a
7	29	$E_{m_i}(h_{i-1}) \oplus w_i$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	30	$E_{h_{i-1}}(h_{i-1}) \oplus w_i$	1	1			b
18	31	$E_{w_i}(h_{i-1}) \oplus w_i$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	32	$E_v(h_{i-1}) \oplus w_i$	1	1			b

Figure 1: Summary of results. Column 1 is our number $\imath$ for the function (we write $f_\imath$ for the compression function and $H_\imath$ for its induced hash function). Column 2 is the number from [7] (we write $\hat{f}_\jmath$ and $\hat{H}_\jmath$). Column 3 defines $f_\imath(h_{i-1}, m_i)$ and $\hat{f}_\jmath(h_{i-1}, m_i)$. We write w_i for $m_i \oplus h_{i-1}$. Columns 4–7 give our collision-resistance and inversion-resistance bounds. Column 8 comments on collision-finding attacks: (a) $H(M)$ is determined by the last block only; two E queries; (b) Attack uses two E queries and one E^{-1} query; (c) Attack uses $q/2$ E queries and $q/2$ E^{-1} queries; (d) Attack given by Theorem 5.1; (e) Attack given by Theorem 5.3; (f) $H(M)$ independent of block order; two E queries; (g) Attack uses (at most) two E queries. We do not explore inversion resistance for schemes that are trivially breakable in the sense of collision resistance.

i	j	$h_i =$	**CR low-bnd**	**CR up-bnd**	**IR low-bnd**	**IR up-bnd**	
19	33	$E_{m_i}(w_i) \oplus v$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
	34	$E_{h_{i-1}}(w_i) \oplus v$	1	1			b
	35	$E_{w_i}(w_i) \oplus v$	1	1			g
	36	$E_v(w_i) \oplus v$	1	1			b
20	37	$E_{m_i}(w_i) \oplus m_i$	$.3q(q-1)/2^n$	$3q(q+1)/2^n$	$0.15q^2/2^n$	$9(q+3)^2/2^n$	c
4	38	$E_{h_{i-1}}(w_i) \oplus m_i$	$.039(q-1)(q-3)/2^n$	$q(q+1)/2^n$	$0.4q/2^n$	$2q/2^n$	d
	39	$E_{w_i}(w_i) \oplus m_i$	1	1			g
	40	$E_v(w_i) \oplus m_i$	1	1			g
8	41	$E_{m_i}(w_i) \oplus h_{i-1}$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
	42	$E_{h_{i-1}}(w_i) \oplus h_{i-1}$	1	1			b
	43	$E_{w_i}(w_i) \oplus h_{i-1}$	1	1			g
	44	$E_v(w_i) \oplus h_{i-1}$	1	1			b
6	45	$E_{m_i}(w_i) \oplus w_i$	$.3q(q-1)/2^n$	$q(q+1)/2^n$	$0.6q/2^n$	$2q/2^n$	e
2	46	$E_{h_{i-1}}(w_i) \oplus w_i$	$.039(q-1)(q-3)/2^n$	$q(q+1)/2^n$	$0.4q/2^n$	$2q/2^n$	d
	47	$E_{w_i}(w_i) \oplus w_i$	1	1			g
	48	$E_v(w_i) \oplus w_i$	1	1			g
	49	$E_{m_i}(v) \oplus v$	1	1			a
	50	$E_{h_{i-1}}(v) \oplus v$	1	1			a
	51	$E_{w_i}(v) \oplus v$	1	1			g
	52	$E_v(v) \oplus v$	1	1			a
	53	$E_{m_i}(v) \oplus m_i$	1	1			a
	54	$E_{h_{i-1}}(v) \oplus m_i$	1	1			b
	55	$E_{w_i}(v) \oplus m_i$	1	1			g
	56	$E_v(v) \oplus m_i$	1	1			a
	57	$E_{m_i}(v) \oplus h_{i-1}$	1	1			f
	58	$E_{h_{i-1}}(v) \oplus h_{i-1}$	1	1			a
	59	$E_{w_i}(v) \oplus h_{i-1}$	1	1			g
	60	$E_v(v) \oplus h_{i-1}$	1	1			a
	61	$E_{m_i}(v) \oplus w_i$	1	1			f
	62	$E_{h_{i-1}}(v) \oplus w_i$	1	1			b
	63	$E_{w_i}(v) \oplus w_i$	1	1			g
	64	$E_v(v) \oplus w_i$	1	1			b

Figure 2: Summary of results, continued. See the caption of Figure 1 for an explanation of the entries in this table.

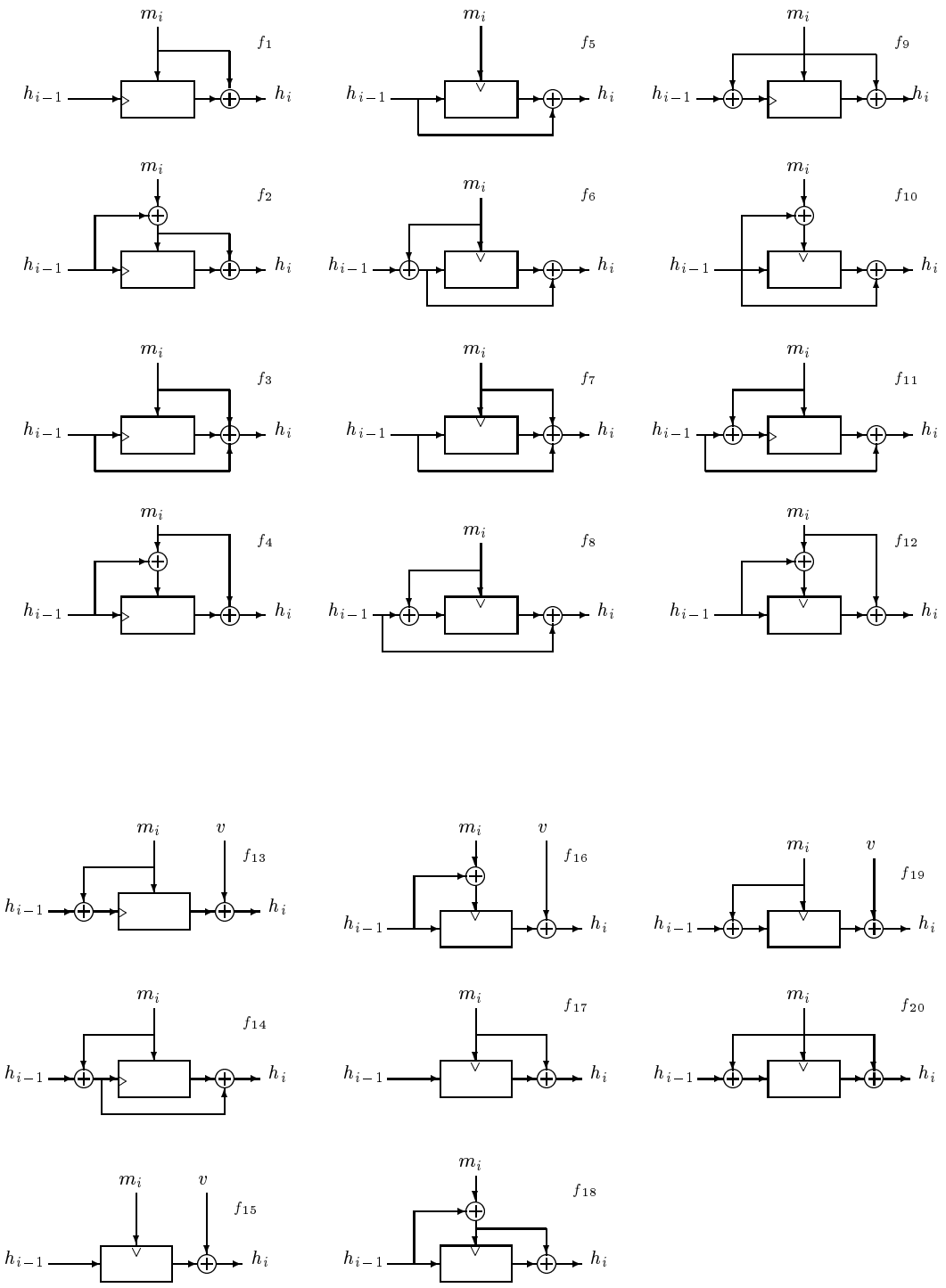

Figure 3: The compression functions $f_1, \ldots, f_{20}$ for the 20 collision-resistant hash functions $H_1, \ldots, H_{20}$. A hatch marks the location for the key.

Discussion. As with [7], we do not concern ourselves with MD-strengthening [3, 6], wherein strings are appropriately padded so that any $M \in \{0,1\}^*$ may be hashed. Simple results establish the security of the MD-strengthened hash function H^* one gets from a secure multiple-of-block-length hash-function H. All of our attacks work just as well in the presence of MD-strengthening.

It is important not to read too much or too little into black-box results. On the one hand, attacks on block-cipher-based hash-functions have usually treated the block cipher as a black box. Such attacks are doomed when one has strong results within the black-box model. On the other hand, the only structural aspect of a block cipher captured by the model is its invertibility, so one must be skeptical about what a black-box-model result suggests when using a block cipher with significant structural properties, such as weak keys. With a block cipher like AES, one hopes for better. We point out that the black-box model is considerably sharper than treating the block cipher as a random function $E\colon \{0,1\}^{\kappa+n} \to \{0,1\}^n$. Such a model should be avoided because many attacks on block-cipher-based hash-functions do use the adversary's ability to compute E_k^{-1}. Overall, we see the black-box model as an appropriate first step in understanding the security of block-cipher-based hash-functions. Of course it would be nice to make due with standard assumptions, such as the block cipher being a pseudorandom function, but that assumption is insufficient for our purposes, and no sufficient assumption has been proposed.

Future directions. Though we spoke of AES as rekindling interest in block-cipher-based hash-function designs, we do not address what we regard as the most interesting practical problem in that vein: namely, how best to use an n-bit block cipher to make a hash function with output length larger than n bits. (Many people see $n = 128$ bits as an inadequate output length for a hash function, particularly in view of [10].) The current work does not answer this question, but it does lay the groundwork for getting there.

2 Definitions

Basic notions. Let $\kappa, n \geq 1$ be numbers. A *block cipher* is a map $E\colon \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ where, for each $k \in \{0,1\}^\kappa$, the function $E_k(\cdot) = E(k,\cdot)$ is a permutation on $\{0,1\}^n$. If E is a block cipher then E^{-1} is its inverse, where $E_k^{-1}(y)$ is the string x such that $E_k(x) = y$. Let $\mathrm{Bloc}(\kappa, n)$ be the set of all block ciphers $E\colon \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. Choosing a random element of $\mathrm{Bloc}(\kappa, n)$ means that for each $k \in \{0,1\}^\kappa$ one chooses a random permutation $E_k(\cdot)$.

A (block-cipher-based) *hash function* is a map $H\colon \mathrm{Bloc}(\kappa, n) \times D \to R$ where $\kappa, n, c \geq 1$, $D \subseteq \{0,1\}^*$, and $R = \{0,1\}^c$. The function H must be given by a program that, given M, computes $H^E(M) = H(E, M)$ using an E-oracle. Hash function $f\colon \mathrm{Bloc}(\kappa, n) \times D \to R$ is a *compression function* if $D = \{0,1\}^a \times \{0,1\}^b$ for some $a, b \geq 1$ where $a + b \geq c$. Fix $h_0 \in \{0,1\}^a$. The *iterated hash* of compression function $f\colon \mathrm{Bloc}(\kappa, n) \times (\{0,1\}^a \times \{0,1\}^b) \to \{0,1\}^a$ is the hash function $H\colon \mathrm{Bloc}(\kappa, n) \times (\{0,1\}^b)^* \to \{0,1\}^a$ defined by $H^E(m_1 \cdots m_\ell) = h_\ell$ where $h_i = f^E(h_{i-1}, m_i)$. Set $H^E(\varepsilon) = h_0$. If the program for f uses a single query $E(k, x)$ to compute $f^E(m, h)$ then f (and its iterated hash H) is *rate-1*. We often omit the superscript E to f and H.

We write $x \xleftarrow{\$} S$ for the experiment of choosing a random element from the finite set S and calling it x. An *adversary* is an algorithm with access to one or more oracles. We write these as superscripts.

Collision resistance. To quantify the collision resistance of a block-cipher-based hash function H we instantiate the block cipher by a randomly chosen $E \in \mathrm{Bloc}(\kappa, n)$. An adversary A is

given oracles for $E(\cdot,\cdot)$ and $E^{-1}(\cdot,\cdot)$ and wants to find a *collision* for H^E—that is, M, M' where $M \neq M'$ but $H^E(M) = H^E(M')$. We look at the number of queries that the adversary makes and compare this with the probability of finding a collision.

Definition 2.1 (Collision resistance of a hash function) Let H be a block-cipher-based hash function, $H\colon \mathrm{Bloc}(\kappa, n) \times D \to R$, and let A be an adversary. Then the advantage of A in finding collisions in H is the real number

$$\mathbf{Adv}_H^{\mathrm{coll}}(A) \;=\; \Pr\left[E \xleftarrow{\$} \mathrm{Bloc}(\kappa, n);\ (M, M') \xleftarrow{\$} A^{E,E^{-1}} :\ M \neq M' \ \&\ H^E(M) = H^E(M')\right] \quad \diamondsuit$$

For $q \geq 1$ we write $\mathbf{Adv}_H^{\mathrm{coll}}(q) = \max_A\{\mathbf{Adv}_H^{\mathrm{coll}}(A)\}$ where the maximum is taken over all adversaries that ask at most q oracle queries (ie, E-queries + E^{-1} queries). Other advantage functions are silently extended in the same way.

We also define the advantage of an adversary in finding collisions in a compression function $f\colon \mathrm{Bloc}(\kappa, n) \times \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^c$. Naturally (h, m) and (h', m') collide under f if they are distinct and $f^E(h, m) = f^E(h', m')$, but we also give credit for finding an (h, m) such that $f^E(h, m) = h_0$, for a fixed $h_0 \in \{0,1\}^c$. If one treats the hash of the empty string as the constant h_0 then $f^E(h, m) = h_0$ amounts to having found a collision between (h, m) and the empty string.

Definition 2.2 (Collision resistance of a compression function) Let f be a block-cipher-based compression function, $f\colon \mathrm{Bloc}(\kappa, n) \times \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^c$. Fix a constant $h_0 \in \{0,1\}^c$ and an adversary A. Then the advantage of A in finding collisions in f is the real number

$$\begin{aligned}\mathbf{Adv}_f^{\mathrm{comp}}(A) \;=\; & \Pr\Big[E \xleftarrow{\$} \mathrm{Bloc}(\kappa, n);\ ((h, m), (h', m')) \xleftarrow{\$} A^{E,E^{-1}} : \\ & ((h, m) \neq (h', m') \ \wedge\ f^E(h, m) = f^E(h', m')) \ \vee\ f^E(h, m) = h_0\Big] \qquad \diamondsuit\end{aligned}$$

Inversion resistance. Though we focus on collision resistance, we are also interested in the difficulty of inverting hash functions. We use the following measure for the difficulty of inverting a hash function at a random point.

Definition 2.3 (Inverting random points) Let H be a block-cipher-based hash function, $H\colon \mathrm{Bloc}(\kappa, n) \times D \to R$, and let A be an adversary. Then the advantage of A in inverting H is the real number

$$\mathbf{Adv}_H^{\mathrm{inv}}(A) \;=\; \Pr\left[E \xleftarrow{\$} \mathrm{Bloc}(\kappa, n);\ \sigma \xleftarrow{\$} R;\ M \leftarrow A^{E,E^{-1}}(\sigma) :\ H^E(M) = \sigma\right] \qquad \diamondsuit$$

The PGV hash functions. Figure 1 and Figure 2 serve to define $f_\imath[n]\colon \mathrm{Bloc}(\kappa, n) \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ and $\hat{f}_\jmath[n]\colon \mathrm{Bloc}(\kappa, n) \times (\{0,1\}^n \times \{0,1\}^n) \to \{0,1\}^n$ for $\imath \in [1..20]$ and $\jmath \in [1..64]$. These compression functions induce hash functions $H_\imath[n]$ and $\hat{H}_\jmath[n]$. Usually we omit writing the $[n]$.

Discussion. The more customary formalization for a one-way function speaks to the difficulty of finding a preimage for the image of a random domain point (as opposed to finding a preimage of a random range point). But a random-domain-point definition becomes awkward when considering

a function H with an infinite domain: in such a case one would normally have to partition the domain into finite sets and insist that H be one-way on each of them. For each of the functions $H_1, \ldots, H_{20}$, the value $H_\imath^E(M)$ is uniform or almost uniform in $\{0,1\}^n$ when M is selected uniformly in $(\{0,1\}^n)^m$ and E is selected uniformly in $\mathrm{Bloc}(n,n)$. Thus there is no essential difference between the two notions in these cases. This observation justifies defining inversion resistance in the manner that we have. See Appendix B.

Definition 2.3 might be understood as giving the technical meaning of *preimage resistance.* However, a stronger notion of preimage resistance also makes sense, where the range value σ is a fixed point, not a random one, and one maximizes over all such points. Similarly, the usual, random-domain-point notion for a one-way function (from the prior paragraph) might be understood as a technical meaning of *2nd preimage resistance*, but a stronger notion makes sense, where the domain point M is a fixed string, not a random one, and one must maximize over all domain points of a given length. A systematic exploration of different notions of inversion resistance is beyond the scope of this paper.

Conventions. For the remainder of this paper we assume the following significant conventions. First, an adversary does not ask any oracle query in which the response is already known; namely, if A asks a query $E_k(x)$ and this returns y, then A does not ask a subsequent query of $E_k(x)$ or $E_k^{-1}(y)$; and if A asks $E_k^{-1}(y)$ and this returns x, then A does not ask a subsequent query of $E_k^{-1}(y)$ or $E_k(x)$. Second, when a (collision-finding) adversary A for H outputs M and M', adversary A has already computed $H^E(M)$ and $H^E(M')$, in the sense that A has made the necessary E or E^{-1} queries to compute $H^E(M)$ and $H^E(M')$. Similarly, we assume that a (collision-finding) adversary A for the compression function f computes $f^E(h,m)$ and $f^E(h',m')$ prior to outputting (h,m) and (h',m'). Similarly, when an (inverting adversary) A for H outputs a message M, we assume that A has already computed $H^E(M)$, in the sense that A has made the necessary E or E^{-1} queries to compute this value. These assumption are all without loss of generality, in that an adversary A not obeying these conventions can easily be modified to given an adversary A' having similar computational complexity that obeys these conventions and has the same advantage as A.

3 Collision Resistance of the Group-1 Schemes

The group-1 hash-functions $H_1, \ldots, H_{12}$ can all be analyzed using the Merkle-Damgård paradigm. Our security bound is identical for all of these schemes.

Theorem 3.1 [Collision resistance of the group-1 hash functions] Fix $n \geq 1$ and $\imath \in [1..12]$. Then $\mathbf{Adv}^{\mathrm{coll}}_{H_\imath[n]}(q) \leq q(q+1)/2^n$ for any $q \geq 1$. $\diamond$

The proof combines a lemma showing the collision-resistance of $f_1, \ldots, f_{12}$ with the classical result, stated for the black-box model, showing that a hash function is collision resistant if its compression function is. For completeness, the proof of the following is given in Appendix C.1.

Lemma 3.2 [Merkle-Damgård [3, 6] in the black-box model] Let f be a compression function $f\colon \mathrm{Bloc}(n,n) \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ and let H be the iterated hash of f. Then $\mathbf{Adv}^{\mathrm{coll}}_H(q) \leq \mathbf{Adv}^{\mathrm{comp}}_f(q)$ for all $q \geq 1$. $\diamond$

Algorithm *SimulateOracles*(A, n)
Initially, $i \leftarrow 0$ and $E_k(x) = \mathsf{undefined}$ for all $(k, x) \in \{0,1\}^n \times \{0,1\}^n$
Run $A^{?,?}$, answering oracle queries as follows:
  When A asks a query (k, x) to its left oracle:
    $i \leftarrow i + 1$; $k_i \leftarrow k$; $x_i \leftarrow x$; $y_i \xleftarrow{\$} \overline{\text{Range}}(E_k)$; $E_k(x) \leftarrow y_i$; return y_i to A
  When A asks a query (k, y) to its right oracle:
    $i \leftarrow i + 1$; $k_i \leftarrow k$; $y_i \leftarrow y$; $x_i \xleftarrow{\$} \overline{\text{Domain}}(E_k)$; $E_k(x_i) \leftarrow y$; return x_i to A
When A halts, outputting a string *out*:
  return $((x_1, k_1, y_1), ..., (x_i, k_i, y_i), out)$

Figure 4: Simulating a block-cipher oracle. $\text{Domain}(E_k)$ is the set of points x where $E_k(x)$ is no longer undefined and $\overline{\text{Domain}}(E_k) = \{0,1\}^n - \text{Range}(E_k)$. $\text{Range}(E_k)$ is the set of points where $E_k(x)$ is no longer undefined and $\overline{\text{Range}}(E_k) = \{0,1\}^n - \text{Range}(E_k)$.

Lemma 3.3 [Collision resistance of the group-1 compression functions] Fix $n \geq 1$ and $\imath \in [1..12]$. Then $\mathbf{Adv}^{\text{comp}}_{f_\imath[n]}(q) \leq q(q+1)/2^n$ for any $q \geq 1$. ◇

Proof of Lemma 3.3: Fix a constant $h_0 \in \{0,1\}^n$. We focus on $f = f_1$; assume that case. Let $A^{?,?}$ be an adversary attacking the compression function f. Assume that A asks its oracles a total of q queries. We are interested in A's behavior when its left oracle is instantiated by $E \xleftarrow{\$} \text{Bloc}(n, n)$ and its right oracle is instantiated by E^{-1}. That experiment is identical, from A's perspective, to the one defined in Figure 4. Define $((x_1, k_1, y_1), \ldots, (x_q, k_q, y_q), out)$ by running *SimulateOracles*(A, n). If A is successful it means that A outputs $(k, m), (k', m')$ such that one of the following holds: $(k, m) \neq (k', m')$ and $f(k, m) = f(k', m')$, or else $f(k, m) = h_0$. By our definition of f this means that $E_k(m) \oplus m = E_{k'}(m') \oplus m'$ for the first case, or $E_k(m) \oplus m = h_0$ for the second. By our conventions at the end of Section 2, either there are distinct $r, s \in [1..q]$ such that $(x_r, k_r, y_r) = (m, k, E_k(m))$ and $(x_s, k_s, y_s) = (m', k', E_{k'}(m'))$ and $E_{k_r}(m_r) \oplus m_r = E_{k_s}(m_s) \oplus m_s$ or else there is an $r \in [1..q]$ such that $(x_r, k_r, y_r) = (m, k, h_0)$ and $E_{k_r}(x_r) = h_0$. We show that this event is unlikely.

In the execution of *SimulateOracles*(A, n), for any $i \in [1..q]$, let C_i be the event that $y_i \oplus x_i = h_0$ or that there exists $j \in [1..i-1]$ such that either $y_i \oplus x_i = y_j \oplus x_j$. In carrying out the simulation of A's oracles, either y_i or x_i was randomly selected from a set of at least size $2^n - (i-1)$, so $\Pr[\mathsf{C}_i] \leq i/(2^n - i)$. By the contents of the previous paragraph, we thus have that $\mathbf{Adv}^{\text{comp}}_{f[n]}(A) \leq \Pr[\mathsf{C}_1 \vee \ldots \vee \mathsf{C}_q] \leq \sum_{i=1}^q \Pr[\mathsf{C}_i] \leq \sum_{i=1}^q \frac{i}{2^n-(i-1)} \leq \frac{1}{2^n-2^{n-1}} \sum_{i=1}^q i$ if $q \leq 2^{n-1}$. Continuing, our expression is at most $\frac{1}{2^{n-1}} \frac{q(q+1)}{2} = \frac{q(q+1)}{2^n}$. Since the above inequality is vacuous when $q > 2^{n-1}$, we may now drop the assumption that $q \leq 2^{n-1}$. We conclude that $\mathbf{Adv}^{\text{comp}}_{f[n]}(q) \leq q(q+1)/2^n$.

The above concludes the proof for the case of f_1. Compression functions $f_{2..12}$ are similar. ▮

4 Collision Resistance of the Group-2 Schemes

We cannot use the Merkle-Damgård paradigm for proving the security of $H_{13..20}$ because their compression functions are *not* collision-resistant. Attacks for each compression function are easy to find. For example, one can break $f_{17}(h, m) = E_m(h) \oplus m$ as a compression function by choosing any two distinct $m, m' \in \{0,1\}^n$, computing $h = E_m^{-1}(m)$ and $h' = E_{m'}^{-1}(m')$, and outputting (h, m)

and (h', m'). All the same, hash functions $H_{13..20}$ enjoy almost the same collision-resistance upper bound as $H_{1..12}$.

Theorem 4.1 [**Collision resistance of the group-2 hash functions**] Fix $n \geq 1$ and $\imath \in [13..20]$. Then $\mathbf{Adv}^{\mathrm{coll}}_{H_\imath[n]}(q) \leq 3q(q+1)/2^n$ for all $q \geq 1$. ◇

Proof of Theorem 4.1: Fix constants $h_0, v \in \{0,1\}^n$. We prove the theorem for the case of H_{13}, where $f(h,m) = f_{13}(h,m) = E_{h \oplus m}(m) \oplus v$.

We define a directed graph $G = (V_G, E_G)$ with vertex set $V_G = \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$ and an arc $(x,k,y) \to (x',k',y')$ in E_G if and only if $k' \oplus x' = y \oplus v$.

Let $A^{?,?}$ be an adversary attacking H_{13}. We analyze the behavior of A when its left oracle is instantiated by $E \xleftarrow{\$} \mathrm{Bloc}(n,n)$ and its right oracle is instantiated by E^{-1}. Assume that A asks its oracles at most q total queries. We must show that $\mathbf{Adv}^{\mathrm{coll}}_{H_{13}[n]}(A) \leq 3q(q+1)/2^n$. Run the algorithm *SimulateOracles*(A,n). As A executes with its (simulated) oracle, color the vertices of G as follows:

- Initially, each vertex of G is *uncolored*.
- When A asks an E-query (k,x) and this returns a value y, or when A asks an E^{-1}-query of (k,y) and this returns x, then: if $x \oplus k = h_0$ then vertex (x,k,y) gets colored *red*; otherwise vertex (x,k,y) gets colored *black*.

According to the conventions at the end of Section 2, every query the adversary asks results in exactly one vertex getting colored red or black, that vertex formerly being uncolored.

We give a few additional definitions. A vertex of G is colored when it gets colored red or black. A path P in G is colored if all of its vertices are colored. Vertices (x,k,y) and (x',k',y') are said to collide if $y = y'$. Distinct paths P and P' are said to collide if all of their vertices are colored and they begin with red vertices and they end with colliding vertices. Let C be the event that, as a result of the adversary's queries, there are formed in G some two colliding paths.

Claim 4.2 $\mathbf{Adv}^{\mathrm{coll}}_{H_{13}[n]}(A) \leq \Pr[\mathsf{C}]$.

Claim 4.3 $\Pr[\mathsf{C}] \leq 3q(q+1)/2^n$.

Claim 4.2 is proven in Appendix C.2 while Claim 4.3 is proven in Appendix C.3. The result for H_{13} now follows by combining the two claims. The proofs for $H_{14..20}$ can be obtained by adapting the proof we have just given (including the contents of the Appendices C.2 and C.3) with the help of Figure 5. ∎

5 Matching Attacks on Collision Resistance

In this section we show that the security bounds given in Sections 3 and 4 are tight: we devise and analyze attacks that achieve advantage close to the earlier upper bounds. Our results are as follows.

$\imath$	$h_i =$	$(x,k,y) \to (x',k',y')$ if	(x,k,y) red if	(x,k,y), (x',k',y') collide if
13	$E_{w_i}(m_i) \oplus v$	$y \oplus v = x' \oplus k'$	$x \oplus k = h_0$	$y = y'$
14	$E_{w_i}(m_i) \oplus w_i$	$k \oplus y = x' \oplus k'$	$x \oplus k = h_0$	$k \oplus y = k' \oplus y'$
15	$E_{m_i}(h_{i-1}) \oplus v$	$y \oplus v = x'$	$x = h_0$	$y \oplus v = x'$
16	$E_{w_i}(h_{i-1}) \oplus v$	$y \oplus v = x'$	$x = h_0$	$y \oplus v = x'$
17	$E_{m_i}(h_{i-1}) \oplus m_i$	$k \oplus y = x'$	$x = h_0$	$k \oplus y = k' \oplus y'$
18	$E_{w_i}(h_{i-1}) \oplus w_i$	$k \oplus y = x'$	$x = h_0$	$k \oplus y = k' \oplus y'$
19	$E_{m_i}(w_i) \oplus v$	$y \oplus v = x' \oplus k'$	$x \oplus k = h_0$	$y = y'$
20	$E_{m_i}(w_i) \oplus m_i$	$k \oplus y = x' \oplus k'$	$x \oplus k = h_0$	$k \oplus y = k' \oplus y'$

Figure 5: Rules for the existence of arcs, the coloring of a vertex red, and when vertices are said to collide. These notions are used in the proof of Theorem 4.1.

Theorem 5.1 [Finding collisions in $H_{1..4}$] Let $\imath \in [1..4]$ and $n \geq 1$. Then

$$\mathbf{Adv}^{\mathrm{coll}}_{H_\imath[n]}(q) \geq 0.039(q-1)(q-3)/2^n$$

for any even $q \in [1..2^{(n-1)/2}]$. ◇

Let $\mathrm{Perm}(n)$ be the set of all permutations on $\{0,1\}^n$. Let $\mathcal{P}_q(\{0,1\}^n)$ denote the set of all q-element subsets of $\{0,1\}^n$. The proof of Theorem 5.1 uses the following technical lemma whose proof appears in Appendix C.4.

Lemma 5.2 Fix $n \geq 1$. Then

$$\Pr\Big[\pi \xleftarrow{\$} \mathrm{Perm}(n);\, Q \xleftarrow{\$} \mathcal{P}_q(\{0,1\}^n) \;:\; \exists x, x' \in Q \text{ such that } x \neq x' \text{ and } \pi(x) \oplus x = \pi(x') \oplus x'\Big] \geq .039(q-1)(q-3)/2^n$$

for any even $q \in [1..2^{(n-1)/2}]$. ◇

Proof of Theorem 5.1: Consider the case $H^E = H^E_1$ and fix $h_0 \in \{0,1\}^n$. Let A be an adversary with the oracles E, E^{-1}. Let A select $m_1, \ldots, m_q \xleftarrow{\$} \{0,1\}^n$ and compute $y_j = E_{h_0}(m_j) \oplus m_j$, $j \in [1..q]$. If A finds $r, s \in [1..q]$ such that $r < s$ and $y_r = y_s$ then it returns (m_r, m_s); otherwise it returns (m_1, m_1) (failure). Let $\pi = E_{h_0}$. By definition π is a uniform element in $\mathrm{Perm}(n)$, so we can invoke Lemma 5.2 to see that the probability that A succeeds to find a collision among $m_1, \ldots, m_q$ under H is at least $.039(q-1)(q-3)/2^n$.

This attack and analysis extends to $H_{2..4}$ by recognizing that for each scheme and distinct one-block messages m and m' we have $H^E(m) = H^E(m')$ if and only if $\pi(x) \oplus x = \pi(x') \oplus x'$ where $\pi = E_{h_0}$ and x, x' are properly defined. For example, for H^E_2 define $x = h_0 \oplus m$ and $x' = h_0 \oplus m'$. ▮

Analysis of collision-finding attacks on $H_{5..20}$ is considerably less technical than for $H_{1..4}$. The crucial difference is that in each of $H_{5..20}$ the block cipher is keyed in the first round by either the message m, or $m \oplus h_0$, where h_0 is a fixed constant. Hence when A hashes q distinct one-block messages it always observes q random values. See Appendix C.5 for a proof of the following.

Theorem 5.3 [Finding collisions in $H_{5..20}$] Let $\imath \in [5..20]$ and $n \geq 1$. Then $\mathbf{Adv}^{\mathrm{coll}}_{H_\imath[n]}(q) \geq 0.3q(q-1)/2^n$ for any $q \in [1..2^{n/2}]$. ◇

6 Security of the Schemes as OWFs

From the perspective of collision resistance there is no reason to favor any particular scheme from $H_{1..20}$. However, in this section we show that these schemes can be separated based on their strength as one-way functions. In particular, for an n-bit block cipher, an adversary attacking a group-1 hash function requires nearly 2^n oracle queries to do well at inverting a random range point, while an adversary attacking a group-2 hash function needs roughly $2^{n/2}$ oracle queries to do the same job.

We begin with the theorem establishing good inversion-resistance for the group-1 schemes. The theorem is immediate from the two lemmas that follow it. The first result is analogous to Lemma 3.2. The second result shows that $f_{1..12}$ have good inversion-resistance. All omitted proofs can be found in the appendices.

Theorem 6.1 [OWF security of the group-1 hash functions] Fix $n \geq 1$ and $\imath \in [1..12]$. Then $\mathbf{Adv}^{\mathrm{inv}}_{H_\imath[n]}(q) \leq q/2^{n-1}$ for any $q \geq 1$. ◇

Lemma 6.2 [Merkle-Damgård for inversion resistance] Let f be a compression function having signature $f\colon \mathrm{Bloc}(n,n) \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ and let H be the iterated hash of f. Then $\mathbf{Adv}^{\mathrm{inv}}_{H}(q) \leq \mathbf{Adv}^{\mathrm{inv}}_{f}(q)$ for all $q \geq 1$. ◇

Lemma 6.3 [Inversion resistance of the group-1 compression functions] Fix $n \geq 1$ and $\imath \in [1..12]$. Then $\mathbf{Adv}^{\mathrm{inv}}_{f_\imath[n]}(q) \leq q/2^{n-1}$ for any $q \geq 1$. ◇

Proof of Lemma 6.3: Fix a constant $h_0 \in \{0,1\}^n$. We focus on compression function $f^E = f_1^E$; assume that case. Let A be an adversary with oracles E, E^{-1} and input σ. Assume that A asks its oracles q total queries.

Define $((x_1,k_1,y_1),\ldots,(x_q,k_q,y_q),out)$ by running *SimulateOracles*(A,n). By our conventions at the end of Section 2, if A outputs (h,m) such that $E(h,m) \oplus m = \sigma$ then $(m,h,E(h,m)) = (x_i,k_i,y_i)$ for some $i \in [1..q]$. Let C_i be the event that (x_i,k_i,y_i) is such that $x_i \oplus y_i = \sigma$. In carrying out the simulation of A's oracles, either x_i or y_i was randomly assigned from a set of at least size $2^n - (i-1)$, so $\Pr[\mathsf{C}_i] \leq 1/(2^n-(i-1))$. Thus $\Pr[(h,m) \leftarrow A^{E,E^{-1}}(z) : E(h,m) \oplus m = \sigma] \leq \Pr[\mathsf{C}_1 \vee \ldots \vee \mathsf{C}_q] \leq \sum_{i=1}^q \Pr[\mathsf{C}_i] \leq \sum_{i=1}^q \frac{1}{2^n-(i-1)} \leq \frac{q}{2^n-2^{n-1}}$ if $q \leq 2^{n-1}$. Continuing, our expression is at most $\frac{q}{2^{n-1}}$. Since the above inequality is vacuous when $q > 2^{n-1}$, we may now drop the assumption that $q \leq 2^{n-1}$.

The above concludes the proof for the case of f_1. Compression functions $f_{2..12}$ are similar. ▮

We cannot use Lemma 6.2 to prove the security of the group-2 schemes because the associated compression functions are *not* inversion-resistant. An attack for each is easy to find. For example, consider $f_{13}(h,m) = E(h \oplus m, m) \oplus v$. For any point σ, the adversary fixes $k = 0$, computes $m = E_0^{-1}(\sigma \oplus v)$, and returns (m,m), which is always a correct inverse to σ. Still, despite these compression functions being invertible with a single oracle query, there is a reasonable security bound for the group-2 schemes.

Theorem 6.4 [OWF security of the group-2 hash functions] Fix $n \geq 1$ and $\imath \in [13..20]$. Then $\mathbf{Adv}^{\mathrm{inv}}_{H_\imath[n]}(q) \leq 9(q+3)^2/2^n$ for any $q > 1$. ◇

The proof of Theorem 6.4 appears in Appendix C.7 and makes use of the following lemma, which guarantees that, up to a constant, for messages of length greater than n-bits, the bounds we have computed for collision resistance hold for inversion resistance as well.

Lemma 6.5 [Collision resistance $\Rightarrow$ inversion resistance] Fix $\imath \in [1..20]$ and $n \geq 1$. Let $\tilde{H} = H_\imath[n]$ restricted to domain $\mathrm{Bloc}(n,n) \times \bigcup_{i \geq 2}\{0,1\}^{in}$. Then $\mathbf{Adv}^{\mathrm{inv}}_{\tilde{H}}(q) \leq 3\mathbf{Adv}^{\mathrm{coll}}_{H}(q+2) + q/2^{n-1}$ for any $q \geq 1$. $\diamond$

For the proof of Lemma 6.5 see Appendix C.8.

Finally, we prove that the security bounds given in Theorems 6.1 and 6.4 are tight, by describing adversaries that achieve advantage very close to the upper bounds. The analysis falls into three groupings.

Theorem 6.6 [Attacking $H_{1..4}$ as OWFs] Fix $n \geq 1$ and $\imath \in [1..4]$. Then $\mathbf{Adv}^{\mathrm{inv}}_{H_\imath[n]}(q) \geq 0.4q/2^n$ for any $q \in [1..2^{n-2}]$. $\diamond$

Theorem 6.7 [Attacking $H_{5..12}$ as OWFs] Fix $n \geq 1$ and $\imath \in [5..12]$. Then $\mathbf{Adv}^{\mathrm{inv}}_{H_\imath[n]}(q) \geq 0.6q/2^n$ for any $q \in [1..2^n - 1]$. $\diamond$

Theorem 6.8 [Attacking $H_{13..20}$ as OWFs] Fix $n \geq 1$ and $\imath \in [13..20]$. Then $\mathbf{Adv}^{\mathrm{inv}}_{H_\imath[n]}(q) \geq 0.15q^2/2^n$ for any even $q \in [2..2^{n/2}]$. $\diamond$

The proofs for the above three theorems are found in Appendices C.9, C.10, and C.11, respectively.

Acknowledgments

Thanks to the anonymous reviewers for helpful comments and references.

John Black received support from NSF CAREER award CCR-0133985. This work was carried out while John was at the University of Nevada, Reno.

Phil Rogaway and his student Tom Shrimpton received support from NSF grant CCR-0085961 and a gift from CISCO Systems. Many thanks for their kind support.

References

[1] M. Bellare, J. Kilian, and P. Rogaway. The security of cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[2] J. Black, P. Rogaway, and T. Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In *Advances in Cryptology – CRYPTO '02*, volume ??? of *Lecture Notes in Computer Science*. Springer-Verlag, 2002. Proceedings version of this paper.

[3] I. Damgård. A design principle for hash functions. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[4] S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. In *Advances in Cryptology – ASIACRYPT '91*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer-Verlag, 1992.

$\jmath$	M_c
39	$h_0 \oplus c \;\|\; E_c(c) \oplus h_0$
40	$h_0 \oplus c \;\|\; E_v(c) \oplus h_0$
43	$h_0 \oplus c \;\|\; E_c(c) \oplus h_0 \oplus c$
55	$h_0 \oplus c \;\|\; E_c(v) \oplus h_0$
59	$h_0 \oplus c \;\|\; E_c(v) \oplus h_0 \oplus c$

Figure 6: Messages that collide under the five weak "backward attackable" schemes in [7]. For any $c \in \{0,1\}^n$, $\hat{H}_\jmath(M_c) = h_0$ for all five schemes listed.

[5] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. *Journal of Cryptology*, 14(1):17–35, 2001. Earlier version in CRYPTO '96.

[6] R. Merkle. One way hash functions and DES. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[7] B. Preneel, R. Govaerts, and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. In *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science, pages 368–378. Springer-Verlag, 1994.

[8] M. Rabin. Digitalized signatures. In R. DeMillo, D. Dobkin, A. Jones, and R. Lipton, editors, *Foundations of Secure Computation*, pages 155–168. Academic Press, 1978.

[9] C. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.

[10] P. van Oorschot and M. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, 1999. Earlier version in ACM CCS '94.

[11] R. Winternitz. A secure one-way hash function built from DES. In *Proceedings of the IEEE Symposium on Information Security and Privacy*, pages 88–90. IEEE Press, 1984.

A Fatal Attacks on Five of PGV's B-Labeled Schemes

In [7] there are a total of 13 schemes labeled as "backward attackable." We have already shown that eight of these, $H_{13..20}$, are collision resistant. But the remaining five schemes are completely insecure; each can be broken with two queries. Consider, for example, $H = \hat{H}_{39}$, constructed by iterating the compression function $f = \hat{f}_{39}$ defined by $f^E(h_{i-1}, m_i) = E_{m_i \oplus h_{i-1}}(m_i \oplus h_{i-1}) \oplus m_i$. For any $c \in \{0,1\}^n$ the strings $(h_0 \oplus c) \,\|\, (E_c(c) \oplus h_0)$ hashes to h_0, and so it so it takes only two queries to produce a collision. Variants of this attack, break the schemes $\hat{H}_{40}$, $\hat{H}_{43}$, $\hat{H}_{55}$ and $\hat{H}_{59}$ defined in Figure 1. See Figure 6 for the attacks.

B Two Notions of Inversion Resistance

We defined $\mathbf{Adv}_H^{\mathrm{inv}}$ by giving the adversary a random range point $\sigma \in \{0,1\}^n$ and asking the adversary to find an H-preimage for σ. The usual definition for a one-way function has one choose a random domain point M, apply H, and ask then ask the adversary to invert the result.

Definition B.1 (Conventional definition of a OWF) Let H be a block-cipher-based hash function, H: $\mathrm{Bloc}(\kappa, n) \times D \to R$, and let ℓ be a number such that $\{0,1\}^\ell \subseteq D$. Let A be an adversary. Then the advantage of A in inverting H on the distribution induced by applying H to a random ℓ-bit string is the real number

$$\begin{aligned}\mathbf{Adv}_H^{\mathrm{owf}}(A,\ell) \quad = \quad & \Pr\Big[E \xleftarrow{\$} \mathrm{Bloc}(\kappa,n);\ M \xleftarrow{\$} (\{0,1\}^n)^\ell;\ \sigma \leftarrow H^E(M); \\ & M' \leftarrow A^{E,E^{-1}}(\sigma):\ H^E(M') = \sigma\Big]\end{aligned} \qquad \diamondsuit$$

For $q \geq 0$ a number, $\mathbf{Adv}_H^{\mathrm{owf}}(q,\ell)$ is defined in the usual way, as the maximum value of $\mathbf{Adv}_H^{\mathrm{owf}}(A,\ell)$ over all adversaries A that ask at most q queries.

Though the $\mathbf{Adv}^{\mathrm{owf}}$ and $\mathbf{Adv}^{\mathrm{inv}}$ measures can, in general, be far apart, it is natural to guess that they coincide for "reasonable" hash-functions like $H_{1..20}$. In particular, one might think that the random variable $H_\imath^E(M)$ is uniformly distributed in $\{0,1\}^n$ if $M \xleftarrow{\$} \{0,1\}^{n\ell}$ and $E \xleftarrow{\$} \mathrm{Bloc}(n,n)$. Interestingly, this is not true. For example, experiments show that when $E \xleftarrow{\$} \mathrm{Bloc}(2,2)$ and $M \xleftarrow{\$} \{0,1\}^4$ the string $H_1^E(M)$ takes on the value 00 more than a quarter of the time (in fact, 31.25% of the time) while each of the remaining three possible outputs (01, 10, 11) occur less than a quarter of the time (each occurs $22.91\overline{6}$% of the time). Still, for $H_{1..20}$, the two notions are close enough that we have used Definition 2.3 as a surrogate for Definition B.1. The result is as follows.

Lemma B.2 Fix $n \geq 1$ and $\imath \in [1..20]$. Then for any $q, \ell \geq 1$,

$$\Big|\mathbf{Adv}_{H_\imath[n]}^{\mathrm{inv}}(q) - \mathbf{Adv}_{H_\imath[n]}^{\mathrm{owf}}(q,\ell)\Big| \quad \leq \quad \ell/2^{n-1} \qquad \diamondsuit$$

Proof of Lemma B.2: Consider the case for $H = H_1$. The proof is a "game argument," as in [5]. Consider Games 1 and 2, both defined in Figure 7. The probability that A outputs 1 in game 1 is exactly $\mathbf{Adv}_H^{\mathrm{inv}}(A)$, while the probability that A outputs 1 in game 2 is exactly $\mathbf{Adv}_H^{\mathrm{owf}}(A)$. Furthermore, Games 1 and 2 are identical except for the statement that follows the conditional setting of *bad* to true. Thus $|\mathbf{Adv}_H^{\mathrm{inv}}(A) - \mathbf{Adv}_H^{\mathrm{owf}}(A)|$ is at most the probability that *bad* gets set to true in Game 1. This value is at most $2\ell/2^n$. The result follows, and extends in the natural way for the other hash functions. ∎

C Proofs

C.1 Proof of Lemma 3.2

Let A be a collision-finding adversary for H that takes two oracles, E, E^{-1}. We construct from A a collision-finding adversary B for f. Adversary B also takes oracles E, E^{-1}. Let B run A. When A makes an E (resp., E^{-1}) query, B forwards it to E (resp., E^{-1}) and returns to A the result. For $i \in [1..q]$, we say that the ith triple is (x_i, k_i, y_i) if A's ith oracle query was an E-query of (k_i, x_i) and this returned y_i, or else A's ith oracle query was an E^{-1}-query (k_i, y_i) and this returned x_i. Algorithm B records the list of triples. Eventually A halts with an output $(M, M') = (m_1 \cdots m_a, m'_1 \cdots m'_b)$.

Have B compute $H(M)$ and $H(M')$. According to our conventions, all of the necessary queries for B to use in this computation are already recorded in B's list of triples, so no new oracle calls are needed to compute $H(M)$ and $H(M')$.

INITIALIZATION:

for all $(k, x) \in \{0,1\}^n \times \{0,1\}^n$ **do** $E(k, x) \leftarrow \mathsf{undefined}$

$bad \leftarrow \mathsf{false}$; $m_1, \ldots, m_{\ell-1} \xleftarrow{\$} \{0,1\}^n$

for $i \leftarrow 1$ **to** $\ell - 1$ **do**

  if $E(h_{i-1}, m_i) = \mathsf{undefined}$ **then** $E(h_{i-1}, m_i) \xleftarrow{\$} \overline{\mathrm{Range}}(E(h_{i-1}, \cdot))$

  $h_i \leftarrow E(h_{i-1}, m_i) \oplus m_i$

$\sigma \xleftarrow{\$} \{0,1\}^n$; $m_\ell \xleftarrow{\$} \{0,1\}^n$

if $E(h_{\ell-1}, m_\ell) \neq \mathsf{undefined}$ **then**

  $bad \leftarrow \mathsf{true}$ [output previously determined]

  if *Game2* **then** $\sigma \leftarrow E(h_{\ell-1}, m_\ell) \oplus m_\ell$

if there is an $i \in [1..\ell - 1]$ where $E(h_{i-1}, m_i) = \sigma \oplus m_\ell$ **then**

  $bad \leftarrow \mathsf{true}$ [output uniform over reduced set]

  if *Game2* **then** $\sigma \xleftarrow{\$} \overline{\mathrm{Range}}(E(h_{i-1}, \cdot))$

$h_\ell \leftarrow E(h_{\ell-1}, m_\ell) \leftarrow \sigma \oplus m_\ell$

ADVERSARIAL EXECUTION:

Run $A^{?,?}(\sigma)$:

- When A asks a query (k, x) to its left oracle:
  if $E(k, x) = \mathsf{undefined}$ **then** $y \xleftarrow{\$} \overline{\mathrm{Range}}(E(k, \cdot))$, $E(k, x) \leftarrow y$;
  else $y \leftarrow E(k, x)$
  return y to A.
- When A asks a query (k, y) to its right oracle:
  if $\exists\, z$ such that $E(k, z) = y$ **then** $x \leftarrow z$
  else $x \xleftarrow{\$} \overline{\mathrm{Domain}}(E(k, \cdot))$; $E(k, x) \leftarrow y$;
  return x to A.
- Eventually A halts, outputting string $M' = m'_1 m'_2 \cdots m'_{\ell'}$

TESTING:

for $i = 1$ **to** ℓ' **do** $h_i \leftarrow E(h_{i-1}, m'_i) \oplus m'_i$

if $h_{\ell'} = \sigma$ **then return** 1 **else return** 0

Figure 7: Definition of Game 1 (when *Game2* = false) and Game 2 (when *Game2* = true).

Adversary B inspects its list of triples to see if there exists distinct (x, k, y) and (x', k', y'). If so, B outputs this pair of points. Otherwise, B inspects its list of triples to see if there exists a triple (x, k, h_0). If so, B outputs (k, x), (k, x).

We claim that B succeeds whenever A succeeds. By symmetry, we can assume without loss of generality that $a \geq b$. If $H(M) = H(M')$, then $h_a = f(h_{a-1}, m_a) = f(h'_{b-1}, m'_b) = h'_b$. (Primed variables are understood to be associated to $H(M')$.) If $h_{a-1} \neq h'_{b-1}$, or $m_a \neq m'_b$, then we is done. Otherwise, $h_{a-1} = f(h_{a-2}, m_{a-1}) = f(h'_{b-2}, m'_{b-1}) = h'_{b-1}$. Again, if $h_{a-2} \neq h'_{b-2}$ or $m_{a-1} \neq m_{b-1}$, then we are done. Proceeding in this way, we must find some values $\alpha \in [1, a]$ and $\beta \in [1, b]$ such that either $h_\alpha = f(h_{\alpha-1}, m_\alpha) = f(h'_{\beta-1}, m'_\beta) = h'_\beta$, or $h_\alpha = h_0$.

C.2 Proof of Claim 4.2

Suppose that the adversary A outputs colliding messages $M = m_1 \cdots m_a$ and $M' = m'_1 \cdots m'_b$; that is, $H^E(M) = H^E(M')$ for the simulated oracle E. We show that, necessarily, there are two colliding paths.

Let $P = (x_1, k_1, y_1) \to \cdots \to (x_a, k_a, y_a)$ where, for each $i \in [1..a]$, $x_i = m_i$, $k_i = h_{i-1} \oplus x_i$, $y_i = E_{k_i}(x_i)$, and $h_i = y_i \oplus v$. (Recall that h_0 and v are fixed constants.) Similarly, let $P' = (x'_1, k'_1, y'_1) \to \cdots \to (x'_b, k'_b, y'_b)$ where, for each $i \in [1..b]$, $x'_i = m'_i$, $k'_i = h'_{i-1} \oplus x'_i$, $y'_i = E_{k_i}(x'_i)$, and $h'_i = y'_i \oplus v$, and where $h'_0 = h_0$. We claim that P and P' are colliding paths.

According to our conventions, A makes all of the queries necessary to compute $H(M)$ and $H(M')$. So, for each $i \in [1..a]$, A must have made either an E-query (k_i, x_i) or an E^{-1}-query (k_i, y_i). Similarly, for each $i \in [1..b]$, A must have made either an E-query (k'_i, x'_i) or an E^{-1}-query (k'_i, y'_i). We can conclude then that P and P' are colored. Moreover, $x_1 \oplus k_1 = h_0$ and $x'_1 \oplus k'_1 = h'_0$ so each of P and P' starts with a red node.

If $a \neq b$, then clearly P and P' are distinct. Consider $a = b$ and $M \neq M'$. (If $M = M'$, then M and M' do not collide.) There is some $i \in [1..a]$ such that $m_i \neq m'_i$, and so $(x_i, k_i, y_i) \neq (x'_i, k'_i, y'_i)$.

Finally, if M and M' collide we have $h_a = h'_b$, and hence $y_a = y'_b$. This completes the proof of the Claim.

C.3 Proof of Claim 4.3

Let C_i be the event that C occurs by the i-th query. Define C_0 be the null event. Then $\Pr[\mathsf{C}] = \sum_{i=1}^{q} \Pr[\mathsf{C}_i | \overline{\mathsf{C}}_{i-1} \wedge \ldots \wedge \overline{\mathsf{C}}_0]$. Given $\overline{\mathsf{C}}_{i-1} \wedge \ldots \wedge \overline{\mathsf{C}}_0$, the event C_i occurs in one of four ways. To make the discussion of these cases more clear, we define a little more notation. Let $\mathsf{Arc}(i,j)$ be the event that there exists in G vertices (x_i, k_i, y_i) and (x_j, k_j, y_j), and $y_i \oplus v = x_j \oplus k_j$. Let $\mathsf{Red}(i)$ be the event that there exists in G vertex (x_i, k_i, y_i) and $x_i \oplus k_i = h_0$. Let $\mathsf{Collide}(i,j)$ be the event that there exists in G vertices (x_i, k_i, y_i) and (x_j, k_j, y_j), and $y_i = y_j$.

Case 1: *A vertex (x_i, k_i, y_i) is colored on the i-th query, and there exists in G arcs $(x_r, k_r, y_r) \to (x_i, k_i, y_i)$ and $(x_i, k_i, y_i) \to (x_j, k_j, y_j)$, where (x_r, k_r, y_r) and (x_j, k_j, y_j) were colored on the r-th and j-th queries, $r, j < i$, respectively.*

This event requires $\mathsf{Arc}(r,i) \wedge \mathsf{Arc}(i,j)$ be true. If C_i occurs via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned} \Pr[\mathsf{Arc}(r,i) \wedge \mathsf{Arc}(i,j)] &= \Pr[\mathsf{Arc}(i,j) | \mathsf{Arc}(r,i)] \Pr[\mathsf{Arc}(r,i)] \\ &\leq \Pr[\mathsf{Arc}(i,j) | \mathsf{Arc}(r,i)] \leq \frac{i-1}{2^n - (i-1)} \end{aligned}$$

Alternatively, if C_i occurs in this case via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned} \Pr[\mathsf{Arc}(i,j) \wedge \mathsf{Arc}(r,i)] &= \Pr[\mathsf{Arc}(r,i) | \mathsf{Arc}(i,j)] \Pr[\mathsf{Arc}(i,j)] \\ &\leq \Pr[\mathsf{Arc}(r,i) | \mathsf{Arc}(i,j)] \leq \frac{i-1}{2^n - (i-1)} . \end{aligned}$$

Case 2: *A vertex (x_i, k_i, y_i) is colored red on the i-th query, and there exists in G an arc $(x_i, k_i, y_i) \to (x_j, k_j, y_j)$, where (x_j, k_j, y_j) was colored on the j-th query, $j < i$.*

This event requires that $\mathsf{Arc}(i,j)$ and $\mathsf{Red}(i)$. If this occurs via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned}\Pr[\mathsf{Arc}(i,j) \wedge \mathsf{Red}(i)] &= \Pr[\mathsf{Arc}(i,j)|\mathsf{Red}(i)]\Pr[\mathsf{Red}(i)] \\ &\leq \Pr[\mathsf{Arc}(i,j)|\mathsf{Red}(i)] \leq \frac{i-1}{2^n-(i-1)}\,.\end{aligned}$$

Alternatively, if C_i occurs in this case via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned}\Pr[\mathsf{Arc}(i,j) \wedge \mathsf{Red}(i)] &= \Pr[\mathsf{Red}(i)|\mathsf{Arc}(i,j)]\Pr[\mathsf{Arc}(i,j)] \\ &\leq \Pr[\mathsf{Red}(i)|\mathsf{Arc}(i,j)] \leq \frac{i-1}{2^n-(i-1)}\,.\end{aligned}$$

Case 3: *A vertex (x_i, k_i, y_i) is colored on the i-th query, and there exists in G an arc $(x_j, k_j, y_j) \to (x_i, k_i, y_i)$ and a vertex (x_r, k_r, y_r), where (x_j, k_j, y_j) and (x_r, k_r, y_r) were colored on the j-th and r-th queries, $r, j < i$, respectively, and (x_i, k_i, y_i) agrees with (x_r, k_r, y_r).*

This event requires that $\mathsf{Arc}(j,i)$ and $\mathsf{Collide}(i,r)$. If this occurs via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned}\Pr[\mathsf{Arc}(j,i) \wedge \mathsf{Collide}(i,r)] &= \Pr[\mathsf{Collide}(i,r)|\mathsf{Arc}(j,i)]\Pr[\mathsf{Arc}(j,i)] \\ &\leq \Pr[\mathsf{Collide}(i,r)|\mathsf{Arc}(j,i)] \leq \frac{i-1}{2^n-(i-1)}\,.\end{aligned}$$

Alternatively, if C_i occurs in this case via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned}\Pr[\mathsf{Arc}(j,i) \wedge \mathsf{Collide}(i,r)] &= \Pr[\mathsf{Arc}(j,i)|\mathsf{Collide}(i,r)]\Pr[\mathsf{Collide}(i,r)] \\ &\leq \Pr[\mathsf{Arc}(j,i)|\mathsf{Collide}(i,r)] \leq = \frac{i-1}{2^n-(i-1)}\,.\end{aligned}$$

Case 4: *A vertex (x_i, k_i, y_i) is colored red on the i-th query, and there exists in G an arc $(x_i, k_i, y_i) \to (x_i, k_i, y_i)$.*

This event requires that $\mathsf{Red}(i) \wedge \mathsf{Arc}(i,i)$ is true. If this occurs via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned}\Pr[\mathsf{Red}(i) \wedge \mathsf{Arc}(i,i)] &= \Pr[\mathsf{Arc}(i,i)|\mathsf{Red}(i)]\Pr[\mathsf{Red}(i)] \\ &\leq \Pr[\mathsf{Arc}(i,i)|\mathsf{Red}(i)] \leq \frac{1}{2^n-(i-1)}\,.\end{aligned}$$

Alternatively, if C_i occurs via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i-1)$. Then

$$\begin{aligned}\Pr[\mathsf{Red}(i) \wedge \mathsf{Arc}(i,i)] &= \Pr[\mathsf{Red}(i)|\mathsf{Arc}(i,i)]\Pr[\mathsf{Arc}(i,])) \\ &\leq \Pr[\mathsf{Red}(i)] \leq \frac{1}{2^n-(i-1)}\,.\end{aligned}$$

Combining all cases, we have

$$\Pr[\mathsf{C}] \leq \sum_{i=1}^{q} \frac{3(i-1)}{2^n-(i-1)} + \frac{1}{2^n-(i-1)} \leq 3\sum_{i=1}^{q} \frac{i}{2^n-(i-1)} \leq \frac{3q(q+1)}{2^n}$$

C.4 Proof of Lemma 5.2

Before beginning the proof we sketch the ideas involved. The basic idea is to lower-bound the probability that two distinct input blocks m_1 and m_2 will collide under the function $\pi(m) \oplus m$ (ie, $\pi(m_1) \oplus m_1$ will equal $\pi(m_2) \oplus m_2$) where π is a random permutation. The straightforward approach is to write out an experiment where we let the message blocks be $0, 1, 2, \ldots$ written as n-bit strings, and then incrementally fill-in the permutation π by choosing range values which observe the permutivity requirements on π (cf. *SimulateOracles*$(\cdot,\cdot)$). Then one would hope to count up the minimum number of collisions in $\pi(m_i) \oplus m_i$ for each $i \in [1..q]$. If this worked, it would provide a straightforward solution to our problem, but unfortunately there is a technical problem where the pool of consumed range points might have a nonempty intersection with the collection of points which would cause a collision, thereby lowering the chance of a collision (because some collision-causing values cannot be chosen as our next range point).

This difficulty greatly increases the complexity of the proof. So instead of the above approach we do the following: instead of choosing the one-block messages $0, 1, 2, \ldots$, we choose *random* input blocks, and then lower-bound the probability that at least half of these inputs will not cause the problem mentioned above. We then restrict our attention to these inputs and use a straightforward counting argument to obtain a lower bound for them, yielding an overall lower bound.

In order to make the following argument more digestible, we extract some of the key steps into four lemmas, which are stated and proven first. Then we present the proof of the main lemma, using the prior four.

Lemma C.1 Let r be a positive even integer and let $\mathsf{E}_1, \cdots, \mathsf{E}_r$ be r independent events where $\Pr[\mathsf{E}_i] \geq 1/2$ for all $i \in [1..r]$. Let H be the event that at least $r/2$ of the events $\mathsf{E}_1, \cdots, \mathsf{E}_r$ occur. Then $\Pr[\mathsf{H}] \geq 1/2$. $\diamond$

Lemma C.2 Let D and E be events with $\Pr[\mathsf{D} \mid \mathsf{E}] \leq p$ and $\Pr[\mathsf{E}] \geq 1/2$. Then $\Pr[\mathsf{D}] \leq (p+1)/2$. $\diamond$

Proof of Lemma C.2: Let $h = \Pr[\mathsf{E}]$. We have

$$\Pr[\mathsf{D}] = \Pr[\mathsf{D} \mid \mathsf{E}]\Pr[\mathsf{E}] + \Pr[\mathsf{D} \mid \overline{\mathsf{E}}]\Pr[\overline{\mathsf{E}}] \leq ph + (1-h) = 1 + h(p-1).$$

And since $h \geq 1/2$ and $(p-1)$ is nonpositive, we know that $h(p-1)$ is maximized for $h = 1/2$. Therefore $1 + h(p-1) \leq 1 + (p-1)/2 = (p+1)/2$. ∎

Lemma C.3 Fix positive integers N and $q \leq \sqrt{2N}$ with q even. Define a function t: $[0..q-1] \times \{0,1\} \to \mathbb{R}$ as

$$t(i,j) = \begin{cases} \frac{N-2i}{N-i} & \text{if } j = 1 \\ 1 & \text{if } j = 0 \end{cases}$$

For any binary sequence $J = (j_0, j_1, \cdots, j_{q-1})$ in $\{0,1\}^q$, define a function T: $\{0,1\}^q \to \mathbb{R}$ as $T(J) = \prod_{i=0}^{q-1} t(i, j_i)$. Then $T(J)$ is maximum for $J = (j_0, \cdots, j_{q-1})$ where $j_i = 1$ for $i \in [0..q/2-1]$ and $j_i = 0$ when $i \in [q/2..q-1]$. Furthermore $T(J) \leq 1 - 0.079q(q-2)/N$. $\diamond$

Proof of Lemma C.3: The lemma states that if we write out the product

$$\frac{N}{N} \cdot \frac{N-2}{N-1} \cdot \frac{N-4}{N-2} \cdots \frac{N-2q+2}{N-q+1}$$

and we then would like to maximize the product after eliminating exactly half of the terms listed, we should eliminate the right-most half. This follows immediately from the fact that the terms above strictly decrease from left to right. That is, $(N-2a)/(N-a) > (N-2b)/(N-b)$ whenever $a < b$ and $a, b \in [0..q-1]$. Since we must choose exactly $q/2$ terms to include in our product, choosing any but the first $q/2$ terms would not maximize the product since we could remove a smaller term and replace it with a larger term, increasing the overall product.

To obtain the upper bound stated in the last sentence of the lemma, we use the fact that for $x \in [0,1]$ we have $e^{-x} \geq 1-x$ and $(1-1/e)x \leq 1-e^{-x}$. Let $T = \prod_{i=0}^{q/2-1}(N-2i)/(N-i)$, which we have already shown to be the maximum value of $T(J)$. Using the first fact above we have that

$$T \leq \prod_{i=0}^{q/2-1} e^{-i/(N-i)} \leq \prod_{i=0}^{q/2-1} e^{-i/N} = e^{-\sum_{i=0}^{q/2-1} i/N} = e^{-q(q-2)/8N}.$$

Using the second fact above along with the fact that $q \leq \sqrt{2N}$ gives us that

$$e^{-q(q-2)/8N} \leq 1 - \left(1 - \frac{1}{e}\right) \frac{q(q-2)}{8N} \leq 1 - \frac{0.079q(q-1)}{N}.$$

This completes the proof. ▮

Lemma C.4 Fix integers $n > 0$ and $i > 0$ with $i \leq 2^{(n-1)/2}$. Let $U = [0..2^n - 1]$. Let $A = \{a_1, \cdots, a_i\}$ and $X = \{x_1, \cdots, x_i\}$ be any subsets of U each with i elements. Let $B = \{a_k \oplus x_k\colon\ k \in [1..i]\} = \{b_1, \cdots, b_m\}$, where $m \leq k$. Finally, let $C = \{a_j \oplus b_k :\ \ j \in [1..i],\ k \in [1..m]\}$. Then

$$\Pr_{x \in U \setminus X}[x \in C] \leq 1/2. \qquad \diamondsuit$$

Proof of Lemma C.4: Since each b_k is formed by $a_k \oplus x_k$ with $k \in [1..i]$, we know there are at most i^2 possible values in C. Now say b_k is formed by $a_k \oplus x_k$ with $k \in [1..i]$; notice that any value in C formed by $a_k \oplus b_k = a_k \oplus (a_k \oplus x_k) = x_k$ will not be hit by x since $x_k \notin U \setminus X$. Therefore we may bound the probability above by noticing that there are at most $i^2 - i$ values of x which might be contained in C. This gives us the bound

$$\frac{i^2 - i}{|U \setminus X|} = \frac{i^2 - i}{2^n - i} \leq \frac{2^{n-1} - 2^{(n-1)/2}}{2^n - 2^{(n-1)/2}}$$

where the last step follows from the fact that $i \leq 2^{(n-1)/2}$. Next we cancel above and below by $2^{(n-1)/2}$ and use the fact that $(a-1)/(b-1) < a/b$, whenever a and b are positive with $a < b$, to obtain

$$\frac{2^{n-1} - 2^{(n-1)/2}}{2^n - 2^{(n-1)/2}} = \frac{2^{(n-1)/2} - 1}{2^{(n+1)/2} - 1} < \frac{2^{(n-1)/2}}{2^{(n+1)/2}} = 1/2.$$

This completes the proof. ▮

Proof of Lemma 5.2: Assume that q is an even integer. We view the contents of π and Q as being incrementally determined during the process of the probabilistic experiment described in the lemma statement. We will view Q as initially empty and π as initially undetermined, and then at each step we will incrementally fill in these values until Q contains q elements and $\pi(x)$ is determined for every $x \in Q$. Let us state this more precisely. We will let Q_i represent the intermediate state of Q at the i-th step, and π_i represent the function π at the i-th step. To begin, let Q_0 be the empty set and let $\pi_0(x) = \mathsf{undefined}$ for all $x \in U$, where $\mathsf{undefined}$ is some distinguished value indicating that we have not yet determined some range value. When X is a set, we write $\pi(X)$ to mean the set of values $\{\pi(x) : x \in X\}$.

At the i-th step of our procedure, $0 < i \leq q$, we choose a random $x_i \in U \backslash Q_{i-1}$ and set $Q_i = Q_{i-1} \cup \{x_i\}$. Then we choose a random $a_i \in U \backslash \pi_{i-1}(Q_{i-1})$ and define π_i as π_{i-1} with the modification that $\pi_i(x_i) = a_i$. At the end of this procedure we have that $Q = Q_q$ is a cardinality q subset of $\mathcal{P}_q(\{0,1\}^n)$ and that π_q is a permutation on the points Q, as required.

For any $i \in [1..q]$, define $B_i = \{\pi_i(x_i) \oplus x_i : \ x_i \in Q_i\}$. So B_i is the set of hash function outputs we are concerned with here: we wish to bound the probability that no collisions will occur among the elements of B_i. However, as we shall see shortly, a difficulty arises if, during the $(i+1)$-st step, any elements in B_i and $\pi_i(Q_i)$ XOR to x_{i+1}. We avoid this difficulty by conditioning on the event that this does not happen. For notational convenience, define the Boolean function $bad(\pi_i(Q_i), B_i, x_{i+1})$ to be true if there exists a value $a \in \pi_i(Q_i)$ and $b \in B_i$ such that $a \oplus b = x_{i+1}$. Let's now bound the probability that $bad(\pi_i(Q_i), B_i, x_{i+1})$ is true when choosing x_{i+1}.

Let $\overline{\mathsf{E}_i}$ be the event that $bad(\pi_{i-1}(Q_{i-1}), B_{i-1}, x_i)$ is true, and suppose we are at step i of our procedure. But we are precisely in the setting of Lemma C.4, which tells us that $\Pr[\overline{\mathsf{E}_i}] \leq 1/2$, and therefore $\Pr[\mathsf{E}_i] \geq 1/2$. Since this is true for all $i \in [1..q]$, we have q events, each with probability $1/2$. Let event H denote the event that at least half of the E_i occur; we may invoke Lemma C.1 to see that $\Pr[\mathsf{H}] \geq 1/2$. Henceforth we condition on this event so that we may assume there exists a set of $q/2$ indices $I = \{i_1, \cdots, i_{q/2}\}$ such that $bad(\pi_{i_j-1}(Q_{i_j-1}), B_{i_j-1}, x_{i_j})$ is false for all $i_j \in I$.

We now focus on these values x_{i_j} for $i_j \in I$ for which our bad function is false. We will upper-bound the probability of a lack of collision on just these inputs to π by counting up the number of ways a choice of x_{i_j} could cause a collision in $\pi(\cdot) \oplus (\cdot)$. The probability of no collision when x_{i_j} was chosen, given none had occurred so far, is now simple to compute: there were $2^n - 2i_j$ values which would not cause a collision when x_{i_j} was chosen from a pool of $2^n - i_j$ values. And we know the intersection between these sets is empty because we have assumed that $bad(\pi_{i_j-1}(Q_{i_j-1}), B_{i_j-1}, x_{i_j})$ was false. Therefore, the probability is simply $(2^n - 2i_j)/(2^n - i_j)$.

We wish to upper-bound the overall probability that no collisions occur for Q under $\pi(\cdot) \oplus (\cdot)$. Let D_i be the event that no collision occurred in the i-th step of our experiment above. We therefore need to upper-bound $\Pr[\mathsf{D}_q]$. Let us first upper-bound $\Pr[\mathsf{D}_q \mid \mathsf{H}]$. We know

$$\Pr[\mathsf{D}_q \mid \mathsf{H}] = \Pr[\mathsf{D}_q \mid \mathsf{D}_{q-1} \wedge \mathsf{H}] \cdots \Pr[\mathsf{D}_1 \mid \mathsf{H}]$$

and the i-th term in the expansion is bounded either by $(2^n - 2i)/(2^n - i)$ or 1 depending on whether or not $i \in I$. Since we wish an upper bound, Lemma C.3 tells us we may maximize the above by taking $I = \{0, \cdots, q/2 - 1\}$. It further tells us that the maximum value attained by $\Pr[\mathsf{D}_q \mid \mathsf{H}]$ will be at most $1 - 0.079q(q-2)/2^n$. And since $\Pr[\mathsf{H}] \geq 1/2$ we may now bound $\Pr[\mathsf{D}_q]$ by invoking

Lemma C.2 to see

$$\Pr[\mathsf{D}_q] \leq \frac{(1 - .079q(q-2)/2^n) + 1}{2} \leq 1 - .039\frac{q(q-2)}{2^n}.$$

Finally, we note that the probability of having a collision is at least one minus the above quantity, thus

$$\Pr[\overline{\mathsf{D}}_q] \geq .039\frac{q(q-2)}{2^n}.$$

This completes the proof. ∎

C.5 Proof of Theorem 5.3

Consider H_5^E and fix a constant $h_0 \in \{0,1\}^n$. The total number of queries, q, is given. We construct adversary A, which has oracles E, E^{-1}, as follows. Let A compute $y_i = E_i(h_0) \oplus h_0$, $i \in [1..q]$. (Where necessary, regard i as an n-bit string.) If A finds $i, j \in [1..q]$ such that $y_i = y_j$ it returns (m_i, m_j). If A is unable to find such an i then it returns (m_1, m_1) (failure).

We now analyze A so constructed. Notice that y_i is a random value for each $i \in [1..q]$. Let p be the probability that there is at least an i and a j such that $1 \leq j < i \leq q$ and either $y_i = y_j$, or that there is an $i \in [1..q]$ such that $y_i = h_0$. We wish to upperbound p. To do this we examine the complementary quantity $1 - p$. Specifically, we define D_i, $i \in [1, \ldots, q]$ be the event that there does not exist an $1 \leq j < i$ such that $y_i = y_j$. Note that $\Pr[\mathsf{D}_1] = 1$, and that if D_i occurs then $y_1, \ldots, y_i$ are distinct. Then we have

$$\Pr[\mathsf{D}_{i+1}|\mathsf{D}_i] = \frac{2^n - i}{2^n} = 1 - \frac{i}{2^n}.$$

The probability of no collisions after the qth hash is

$$1 - p \;=\; \Pr[\mathsf{D}_q] \;=\; \prod_{i=1}^{q-1} \Pr[\mathsf{D}_{i+1}|\mathsf{D}_i] \;=\; \prod_{i=1}^{q-1}\left(1 - \frac{i}{2^n}\right)$$

We note that $i/2^n \leq 1$, and use the inequality $1 - z \leq e^{-z}$ for all $0 \leq z \leq 1$ for each term in the product, so that

$$1 - p \;\leq\; \prod_{i=1}^{q-1} e^{-i/2^n} \;=\; e^{-q(q-1)/2^{n+1}}.$$

Since $q \leq 2^{n/2}$ we know that $q(q-1)/2^{n+1} \leq 1$. Using the fact that $1 - e^{-z} \geq (1 - e^{-1})z$ for all $0 \leq z \leq 1$, we obtain after rearranging

$$p \geq \left(1 - \frac{1}{e}\right) \cdot \frac{q(q-1)}{2^{n+1}}.$$

This attack and analysis extends to the remaining schemes in $H_{5..20}$ because in each case the block cipher key in the first compression function call is either m or $m \oplus h_0$ where h_0 is a constant. For example, to prove the bound for H_6, the adversary A computes $y_i = E_i(h_0 \oplus i) \oplus h_0 \oplus i$ for $i \in [1..q]$ and returns values as described in the attack for H_5^E. The remaining schemes are handled analogously.

C.6 Proof of Lemma 6.2

Let A be an adversary for H: adversary A takes oracles E, E^{-1} and an input σ and, when successful, it outputs M such that $H^E(M) = \sigma$. We construct an adversary B for f: adversary B takes oracles E, E^{-1} and an input σ and, when successful, it outputs (h, m) such that $f^E(h, m) = \sigma$. Adversary B works as follows. It runs A on σ. When A makes an E (resp., E^{-1}) query, adversary B forwards the query to its E (resp., E^{-1}) oracle and returns to A the result. During this process, for each $i \in [1..q]$, we say that the ith triple is (x_i, k_i, y_i) if A's ith oracle query was an E-query of (k_i, x_i) and this returned y_i, or else A's ith oracle query was an E^{-1}-query (k_i, y_i) and this returned x_i. Adversary B records the list of triples. Eventually A halts with an output $M = m_1 \cdots m_a$. At that point we have B compute $H^E(M)$: for $i \leftarrow 1$ to a set $h_i \leftarrow f^E(h_{i-1}, m_i)$. According to our conventions, all of the necessary $E_k(x)$ values that B needs will already be in B's list of triples—no new oracle calls are needed to compute $H^E(M)$. Now if $h_a = f^E(h_{a-1}, m_a) = \sigma$ then B outputs (h_{a-1}, m_a) and wins its experiment; otherwise it outputs (h_0, m_1) and does not win. Clearly B succeeds whenever A succeeds.

C.7 Proof of Theorem 6.4

Consider the case of $H^E = H^E_{13}$. Fix constants $h_0, v \in \{0,1\}^n$. Let A be an adversary with oracles E, E^{-1} and input σ. Assume that A asks these oracles at most q total queries and then inverts H at σ. Suppose that the inverse provided by A is always a message M with more than two blocks. Then by Lemma 6.5 and Theorem 4.1 we'd have that $\mathbf{Adv}^{\mathrm{inv}}_H(q) \le 3\mathbf{Adv}^{\mathrm{coll}}_H(q+2) + q/2^{n-1} \le 9(q+2)(q+3)/2^n + q/2^{n-1} \le 9(q+3)^2/2^n$. So suppose instead that the inverse provided by A is always a message M having a single block. Then run $((x_1, k_1, y_1), \ldots, (x_q, k_q, y_q), out) \leftarrow \mathit{SimulateOracles}(A, n)$ and suppose that A returns an $m \in \{0,1\}^n$ such that $H(m) = \sigma$ then $E(h_0 \oplus m, m) \oplus v = \sigma$. By our convention, it must be the case that $(m, h_0 \oplus m, E(h_0 \oplus m, m)) = (x_i, k_i, y_i)$ for some $i \in [1..q]$. Let C_i be the event that (x_i, k_i, y_i) is such that $x_i \oplus k_i = h_0$ and $y_i = \sigma \oplus v$.

In carrying out the simulation of A's oracles either x_i or y_i is randomly assigned from a set of at least size $2^n - (i-1)$, and so $\Pr[\mathsf{C}_i] \le 1/(2^n - (i-1))$. Consequently, $\mathbf{Adv}^{\mathrm{inv}}_{H[n]}(A) \le \Pr[\mathsf{C}_1 \vee \ldots \vee \mathsf{C}_q] \le \sum_{i=1}^q \Pr[\mathsf{C}_i] \le \sum_{i=1}^q \frac{1}{2^n-(i-1)} \le \frac{q}{2^n - 2^{n-1}}$ if $q \le 2^{n-1}$. Continuing, our expression is at most $q/2^{n-1}$. Since the above inequality is vacuous when $q > 2^{n-1}$, we may now drop the assumption that $q \le 2^{n-1}$. This bound is inferior to that obtained when M is two or more blocks. The actual advantage of A is a weighted sum of the two advantages computed, which concludes the proof for hash function H_{13}. The remaining cases are similar.

C.8 Proof of Lemma 6.5

Let I be an adversary that attacks $\widetilde{H} = \widetilde{H}_\iota[n]$ in the inversion-resistance sense and assume that I makes q oracle queries. We construct a collision-finding adversary A for H as follows. Let A select $M \xleftarrow{\$} \{0,1\}^{2n}$ and compute $\sigma = \widetilde{H}(M)$. By our assumption that H is rate-1, this requires 2 queries to A's left oracle, E. Adversary A now runs I on σ. When I makes a query to its left (resp., right) oracle, adversary A forwards the query to its own left (resp., right) oracle, and returns to I the result. When I halts with output M', adversary A returns (M, M').

We now analyze the advantage of adversary A. There are at most $2^n - 1$ points $\sigma \in \{0,1\}^n$ that can have a unique preimage under $\widetilde{H}$. Hence there are at least $2^{2n} - 2^n + 1$ messages $M \in \{0,1\}^{2n}$

```
1. σ ←$ {0,1}^n;   win ← false;   for all x ∈ {0,1}^n do π(x) ← undefined
2. for i = 1 to q do
3.     x_i ←$ {0,1}^n; y_i ←$ {0,1}^n
4.     if π(x_i) ≠ undefined then y ← π(x_i)          [maintain permutivity of π]
5.     else if y_i ∈ Range(π) then y_i ←$ Range̅(π)    [maintain permutivity of π]
6.     π(x_i) ← y_i
7.     z_i ← y_i ⊕ x_i                                [put a ball in bin labeled z]
8.     if z_i = σ then win ← true                     [win if a ball goes in the desired bin]
```

Figure 8: Pseudocode for a game equivalent to that played by the adversary in the proof of Theorem 6.6. If *win* is set to true then the adversary has succeeded in inverting the randomly-chosen range point σ under H_1.

that hash to range points having at least two preimages. Let N_σ be the number of preimages of σ. Since A has at least a 1/2 chance of inverting $\widetilde{H}$ at σ when $N_\sigma = 1$, we have that

$$\begin{aligned}
\mathbf{Adv}_H^{\mathrm{coll}}(A) &= \Pr[M' \neq M | N_\sigma \geq 2] \cdot \Pr[N_\sigma \geq 2] \cdot \mathbf{Adv}_{\widetilde{H}}^{\mathrm{owf}}(I) \\
&\geq \frac{1}{2} \cdot \frac{2^{2n} - 2^n + 1}{2^{2n}} \cdot \mathbf{Adv}_{\widetilde{H}}^{\mathrm{owf}}(I) \\
&\geq \frac{3}{8} \mathbf{Adv}_{\widetilde{H}[n]}^{\mathrm{owf}}(I)
\end{aligned}$$

where the last inequality is true because $(2^{2n} - 2^n + 1)/2^{2n}$ is maximized when $n = 1$. Rearranging terms, we conclude that $\mathbf{Adv}_{\widetilde{H}}^{\mathrm{owf}}(q) \leq 8/3 \cdot \mathbf{Adv}_H^{\mathrm{coll}}(q+2) \leq 3\mathbf{Adv}_H^{\mathrm{coll}}(q+2)$. By Lemma B.2 we conclude that $\mathbf{Adv}_{\widetilde{H}}^{\mathrm{inv}}(q) \leq 3\mathbf{Adv}_H^{\mathrm{coll}}(q+2) + q/2^{n-1}$.

C.9 Proof of Theorem 6.6

We consider the case of $H^E = H_1^E$. Fix a constant $h_0 \in \{0,1\}^n$. Let $q > 0$ be given. We construct an adversary A with two oracles E, E^{-1}, taking input σ, which works as follows. First A chooses q random strings $m_1, \ldots, m_q \in \{0,1\}^n$. Then for each $i \in [1..q]$, the adversary computes $y_i = H^E(m_i) = E_{h_0}(m_i) \oplus m_i$. If there is some $i \in [1..q]$ such that $y_i = \sigma$ then A outputs m_i; otherwise, A outputs m_1 (failure).

We now analyze the chance that A succeeds. Consider the following game which models exactly the attack used by A against H. Let there be $N = 2^n$ bins labeled by strings in $\{0,1\}^n$. Choose a permutation π uniformly from the set of all permutations on $\{0,1\}^n$. For $i \in [1..q]$, select $x \xleftarrow{\$} \{0,1\}^n$ and place a ball in the bin labeled by $\pi(x) \oplus x$. Let $D(N,q)$ be the expected number of bins containing at least one ball at the end of this game. Then for a uniformly selected σ in $\{0,1\}^n$, the probability that a ball is in the bin labeled by σ at the end of this game is $D(N,q)/N$. Clearly, the chance that A inverts a uniformly chosen range point under H is exactly this, as well.

We claim that if $q \leq N/4$ then $D(N,q) \geq 7q/16$. To show this, we consider the pseudocode in Figure 8.

Let us examine statements 2–7 more closely. Since $q \leq N/4$, either of the **then**-clauses in statements 4 or 5 is executed in any particular loop with probability at most $1/4 + 1/4 = 1/2$, and so an expected $r \geq q/2$ of the values z_i are uniform in $\{0,1\}^n$. Notice that executing statements 4 or 5 never decreases the number of distinct z_i, and so for the purposes of lowerbounding $D(N,q)$

we can consider the simplified code **for** $j = 1$ **to** $q/2$ **do** $z_j \xleftarrow{\$} \{0,1\}^n$ which places $q/2$ uniform balls into N bins. At the time of the selection of bin z_j for ball j there are already at most $N/8$ bins full. Hence, with probability at least $7/8$ ball j goes into an empty bin (labeled z_j), and so we expect that at least $7q/16$ bins contain a ball. We conclude then that $\mathbf{Adv}_H^{\mathrm{inv}}(A) \geq (7/16)\ q/2^n$. The remaining cases $H_{2..4}$ are handled analogously.

C.10 Proof of Theorem 6.7

We consider the case of $H^E = H_5^E$. Fix a constant $h_0 \in \{0,1\}^n$. Let $q > 0$ be given. We construct an adversary $A^{E,E^{-1}}$, taking input σ, which works as follows. For each $i \in [1..q]$, the adversary computes $y_i = E_i(h_0) \oplus h_0$, where each i is taken as an n-bit string. If there is some $i \in [1..q]$ such that $y_i = \sigma$ then A outputs the string i; otherwise, A outputs the string 1 (failure).

Let D_i be the event that $y_j \neq \sigma$ for all $j \in [1..i]$. According to our block cipher model each y_i is an independently random value. Hence $\Pr[\mathsf{D}_i] = (1 - 1/2^n)^i$ for each $i \in [1..q]$. We have then that $\mathbf{Adv}_{H_5[n]}^{\mathrm{inv}}(A) = 1 - \Pr[\mathsf{D}_q] = 1 - (1 - 1/2^n)^q = 1 - (1 - 1/2^n)^{2^n q/2^n} \geq 1 - e^{-q/2^n} \geq (1 - e^{-1})(q/2^n) \geq .6q/2^n$. where we have used the facts that $(1 - 1/x)^x < 1/e$ for all $x > 0$ and that $1 - e^{-x} \geq (1 - e^{-1})x$ for all $x \in [0,1]$.

This proof approach works to prove the same bound for each of the schemes $H_{5..12}$ because in each of these the block-cipher key is distinct for each message hashed and therefore induces independent random outputs as required by the analysis above.

C.11 Proof of Theorem 6.8

We consider the case $H^E = H_{13}^E$. For simplicity, we further consider the case where $h_0 = 0^n$ (the proof can be easily adapted for other values). We prove the bound by describing and analyzing a meet-in-the-middle attack.

We construct an adversary A with two oracles E, E^{-1} and input σ. Adversary A executes as follows: let A compute $y_i = E_i(i) \oplus v$ for $i \in [1..q/2]$ (where i is encoded as an n-bit string). Let $F = \{y_1, \ldots, y_{q/2}\}$. Define a function $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $g(k, y) = E_k^{-1}(y \oplus v) \oplus k$. Let A compute the set $B = \{g(q/2+1, \sigma), \cdots, g(q, \sigma)\}$ using $q/2$ further oracle queries. If A finds a string $s \in F \cap B$, where $s = E_j(j) \oplus v = E_\ell^{-1}(\sigma \oplus v) \oplus \ell$ for some $j \in [1\,..\,q/2]$ and $\ell \in [(q/2+1)\,..\,q]$, then treating j and ℓ as strings, A outputs the two-block message $(j \parallel E_j(j) \oplus v \oplus \ell)$ otherwise it outputs the string 1 (failure). Notice that if s does exist in $F \cap B$ that $H(j \parallel E_j(j) \oplus v \oplus \ell) = \sigma$.

We now analyze the probability that A will succeed by giving a lower-bound on the probability that $F \cap B$ will be non-empty. Notice that while A runs, the keys for each invocation of E are distinct. This means that in our model the values in sets F and B are uniform independent n-bit strings. Therefore we need only lower-bound the probability that throwing $q/2$ blue balls and $q/2$ red balls randomly into 2^n bins results in a red-blue collision.

Let C denote the event that after throwing all q balls we have a bin containing at least two balls (regardless of color). Let $\mathsf{C_M}$ denote the event that there are only monochromatic collisions after throwing q balls (in other words, $\mathsf{C_M}$ denotes the event that C has occurred but for any pair of balls sharing a bin, they are either both blue or both red). Finally, let $\mathsf{C_B}$ denote a bichromatic collision (ie, the event that there is some red ball and some blue ball sharing the same bin). So C is the disjoint union of $\mathsf{C_M}$ and $\mathsf{C_B}$, or $\Pr[\mathsf{C}] = \Pr[\mathsf{C_M}] + \Pr[\mathsf{C_B}]$.

Next we claim that for any even $q > 1$ we have $\Pr[\mathsf{C_B}] > \Pr[\mathsf{C_M}]$. This claim is justified by a straightforward combinatorial argument: consider any configuration of q uncolored balls lying

in 2^n bins where two balls share a bin. If we color these two balls blue, then there are $\binom{q-2}{q/2-2}$ ways to color the remaining balls. However, if we color one ball red and the other blue, there are $\binom{q-2}{q/2-1}$ ways to color the remaining balls. The latter number is larger than the former since $(q/2-2)!(q/2)! > (q/2-1)!(q/2-1)!$. This immediately implies our claim.

Since $q \leq 2^{n/2}$ we know $\Pr[\mathsf{C}] \geq .3q^2/2^n$ (for a proof see, for example, Appendix A of [1]). Therefore $\Pr[\mathsf{C_B}] > \Pr[\mathsf{C_M}]$ implies $\Pr[\mathsf{C_B}] > .15q^2/2^n$, completing the proof.

Since the function g used in the proof above can be extracted from each scheme $H_{13..20}$ by examining the corresponding round function $f_{13..20}$, and since the above proof did not depend on the actual definition of g, we can extend the attack and analysis above to each of these schemes, obtaining the same bound.