Another Glance At Blockcipher Based Hashing

Martijn Stam

martijn.stam@epfl.ch
EPFL, Switzerland

Abstract. In this note we revisit the rate-1 blockcipher based hash functions as first studied by Preneel, Govaerts and Vandewalle (Crypto'93) and later extensively analysed by Black, Rogaway and Shrimpton (Crypto'02). We analyze a further generalization where any pre- and postprocessing is considered. By introducing a new tweak to earlier proof methods, we obtain a simpler proof that is both more general and more tight than existing results. As added benefit, this also leads to a clearer understanding of the current classification of rate-1 blockcipher based schemes as introduced by Preneel et al. and refined by Black et al.

1 Introduction

The design and analysis of hash functions has recently come under renewed interest as a consequence of the NIST competition for a new hash function standard and the events that lead NIST to launch this competition in the first place. Of the many ideas to create a hash function, one of the oldest is to base it on a blockcipher, either explicily such as in Davies-Meyer construction or implicitly such as in SHA-1. The most common approach is to use a blockcipher taking n-bit blocks and k-bit keys and construct an $n+k$-to-n bit compression function that makes only a single call to the blockcipher. Such schemes are called rate-1; in general the rate measures the number of message blocks that are hashed per call to the blockcipher. Specific examples of this approach are Davies-Meyer [4], Matyas-Meyer-Oseas [3] and Miyaguchi-Preneel [5, 6].

Oftentimes the key size equals the block size, i.e., $n = k$. Preneel et al. [6] studied the general construction $H(M,V) = E_K(X) \oplus C$ where $K, X, C \in \{0, M, V, M \oplus V\}$ (or affine offsets thereof). They concluded that of the $4^3 = 64$ possibilities all but 12 allow collision attacks on the compression function with a complexity beating the birthday bound of $2^{n/2}$. Later Black et al. [1] showed that in the ideal cipher model these 12 compression functions are indeed collision resistant up to the birthday bound. More surprisingly, they also showed that an additional 8 construction are secure when properly iterated, even though collisions can easily be found in the respective compression functions.

Neither of these articles provides a deeper understanding of what makes these 12 respectively 8 schemes special to make them secure as compression function respectively as iterated hash-function: what do they have in common that sets them apart from the other 44 schemes? Moreover, the proof given by Black et al. needs to be rechecked for each of the schemes with minor modifications. Despite these modifications being minor and fairly straightforward, this is not entirely satisfactory.

We give a new proof in the ideal cipher model for the collision resistance of rate-1 blockcipher based compression functions and their iterated hash functions, shedding new light on what it is that provides the provable security for these schemes. To do this, we consider a more general compression function, consisting of the following three simple steps (see Figure 1):

1. $(K, X) \leftarrow C_{\text{IN}}(M, V)$
2. $Y \leftarrow E_K(X)$
3. $H \leftarrow C_{\text{OUT}}(M, V, Y)$

Here E is the blockcipher (where key size and blocksize may differ) and C_{IN} and C_{OUT} can be arbitrary functions given their respective domain and codomain.

Similar to Black et al. we consider two types of schemes. Type-I schemes give rise to collision resistant compression functions whereas Type-II schemes give rise to compression functions that will turn into collision resistant hash functions when iterated. Our taxonomy is slightly different from Black et al.'s: we allow schemes to be both Type-I and Type-II, whereas Black et al. clearly separated these cases. Yet not all Type-I schemes are also of Type-II, even though a collision resistant compression function is well known to give rise to a collision resistant iterated hash function. This discrepancy stems from our definition of Type-I and Type-II schemes based on sufficient conditions that are not necessary ones. Each type is defined by a set of three conditions on C_{IN} and C_{OUT}. Both types share the first two conditions and only differ in the third.

The first condition is bijectivity of C_{IN}. The effect of this is that each forward query to E can only be used to evaluate the compression function on a single point. The second condition is that for all M, V the postprocessing $C_{\mathrm{OUT}}(M, V, \cdot)$ is bijective. This means that the output of the encryption E is maximally used for the output of the compression, ensuring unpredictability of H. For Type-I schemes, the third condition is that for all K, Y the modified postprocessing $C_{\mathrm{OUT}}(C_{\mathrm{IN}}^{-1}(K, \cdot), Y)$ is bijective. This requirement is similar in nature to the second, but then to make sure that each inverse queries corresponds to a single triple (M, V, H) where H is unpredictable. For Type-II schemes, the third condition is that for all K, $C_{\mathrm{IN}}^{-1}(K, \cdot)$ restricted to its second output V is bijective. This requirement captures that for each inverse query corresponds to a single triple (M, V, H), but this time with V being unpredictable.

We provide a proof in the ideal cipher model that the probability of finding a collision in the compression function (for Type-I) respectively in the iterated hash function (for Type-II) is upper bounded by $\frac{1}{2}q(q-1)/(2^n - q)$, where q is the number of queries allowed to the adversary and n is the block size.

The proof for Type-I schemes is fairly standard and straightforward. However, for the Type-II schemes we introduce some small tweaks to existing techniques, allowing us to get a proof that is more elegant, more general and slightly tighter than the existing one by Black et al. In particular, their proof is based upon colouring a directed graph where the vertices represent queries with all possible answers and arcs are drawn according to whether the input to one query is consistent with the output of the former, given the compression function under consideration. This leads to unwieldy graphs with a complicated notion of what consitutes a collision.

This counterintuitive use of graphs was fixed by Lucks [2], who considers a directed graph where vertices correspond to chaining values and edges are drawn (or coloured) whenever a query has been made that would allow to move from one chaining value to the next. However, Lucks' graph is not entirely satisfactory either. The fact that his graph is directed complicates his proof forcing some extra case analysis. Moreover, despite the directed nature of his graph, given an arc it is still unclear whether it was the result of a forward or an inverse query.

Our modest improvement is then simply to dispense with the direction of the arcs (that thus become edges). Although this seemingly aids the adversary (certain patterns in the graph will be deemed a success even when the underlying event on the hash function is not), this simplification surprisingly leads to a tighter bound for the Type-II schemes, mainly because we no longer need to distinguish several cases whose success probability are subsequently added.

We also investigate the ramifications of our general classification for schemes of PGV-type, so $H(M, V) = E_a(b) \oplus c$ with $a, b, c \in \{0, M, V, M \oplus V\}$. We conclude that the Type-I schemes are exactly the same. The schemes that are called Type-II by Black et al. correspond to the schemes that are Type-II but not Type-I in our classification. In particular, our work tightens their bound on collision resistance for Type-II schemes from Black et al.'s $3q(q+1)/2^n$ to $\frac{1}{2}q(q-1)/(2^n - q)$. (That is also an improvement for the Type-I schemes, but this is mainly due to our slightly tighter upper bounding of the relevant sum expressing the adversary's advantage.)

2 Notation

Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher, where the first input serves as key. We will usually write $E_K(X)$ instead of $E(K,X)$. Let $C_{\text{IN}} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n$ and $C_{\text{OUT}} : \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be arbitrary functions. Define $H^E(M,V)$ as follows (as depicted in Figure 1):

1. Perform the preprocessing $(K,X) \leftarrow C_{\text{IN}}(M,V)$
2. Call the blockcipher $Y \leftarrow E_K(X)$
3. Output $H^E(M,V) = H \leftarrow C_{\text{OUT}}(M,V,Y)$

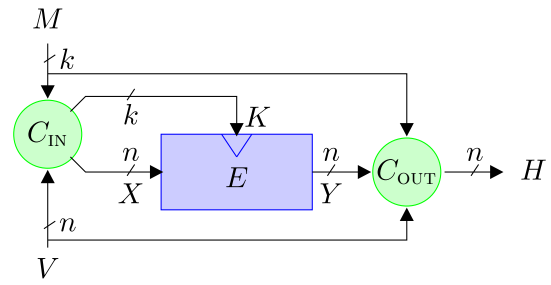

Fig. 1: The general blockcipher based construction

To deal with inverse queries to the blockcipher we introduce $C_{\text{BACK}}(K,X,Y) = C_{\text{OUT}}(C_{\text{IN}}^{-1}(K,X),Y)$. In general, this is a function mapping triplets of strings to subsets of strings, since the result of C_{IN}^{-1} can have varying cardinality. For simplicity, when C_{IN} is bijective, we understand C_{BACK} to have $\{0,1\}^n$ as its codomain. Figure 2 shows the effect of inverse queries.

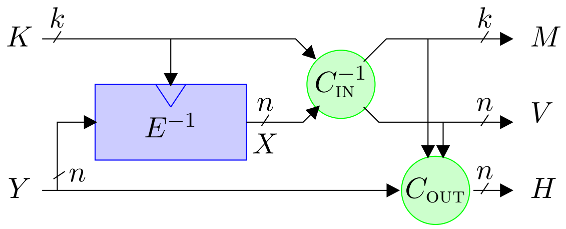

Fig. 2: The effect of inverse queries

We will consider two types of security for the compression function: those that are collision resistant in the ideal cipher model and more general those that give rise to collision resistant hash-function when iterated properly. For this we use a standard Merkle-Damgård-iteration, where we assume that there is already some injective padding from $\{0,1\}^* \to (\{0,1\}^k)^*$ in place. Given an initial vector $V_0 \in \{0,1\}^n$ define $\mathcal{H}^H : (\{0,1\}^k)^* \to \{0,1\}^n$ as follows for $\mathbf{M} = (M_1, \ldots, M_\ell)$:

1. Set $V_i \leftarrow H^E(M_i, V_{i-1})$ for $i = 1, \ldots, \ell$.
2. Output $\mathcal{H}^H(\mathbf{M}) = V_\ell$.

In particular, the hash of the empty message $\mathbf{M} =$ corresponds to $\ell = 0$, so $\mathcal{H}^H() = V_0$, the initial vector.

The task of an adversary is to find a collision. A collision in the compression function consists of two distinct pairs (M, V) and (M', V'), both elements of $\{0,1\}^k \times \{0,1\}^n$, such that $H^E(M, V) = H^E(M', V')$. A collision in the iterated hash function consists of two distinct vectors $\mathbf{M}$ and $\mathbf{M}'$, both in $(\{0,1\}^k)^*$ such that $\mathcal{H}^H(\mathbf{M}) = \mathcal{H}^H(\mathbf{M}')$. We will consider information-theoretic adversaries in the ideal cipher model. Without loss of generality we assume that an adversary has made all relevant queries when he outputs a collision.

3 Rate-1 Blockcipher Based Compression Functions

3.1 Type-I: Collision Resistant Compression Function

Definition 1 *A rate-1 blockcipher based compression function is called to be of Type-I iff the following three hold:*

1. *C_{IN} is bijective.*
2. *for all M, V the postprocessing $C_{\text{OUT}}(M, V, \cdot)$ is bijective.*
3. *for all K, Y the modified postprocessing $C_{\text{BACK}}(K, \cdot, Y)$ is bijective.*

Theorem 2 *Let H^E be a Type-I compression function. If E is an ideal cipher with block size n, then the advantage of an adversary in finding a collision in H^E after q queries is at most $\frac{1}{2}q(q-1)/(2^n - q)$.*

Proof: A collision consists of two pairs (M, V) and (M', V') satisfying $H^E(M, V) = H^E(M', V')$ yet $(M, V) \neq (M', V')$. We will maintain a list of triples (M, V, H) such that $H = H^E(M, V)$ and the adversary has made the relevant queries to E and/or E^{-1}. Since we require the adversary to have made all relevant queries when outputting a collision, we can upper bound the success probability of the adversary by bounding the probability of a collision occuring in this list. We show that any query, be it forward or inverse, will add at most one triple (M, V, H) to this list of computable compression functions, moreover the value H is almost completely out of the adversary's control.

Consider a forward query (K, X). By bijectivity of C_{IN}, there is a unique pair (M, V) corresponding to this query. Thus, each forward query will add one triple (M, V, H) to the adversary's list of computable values. The second criterion implies that the distribution of H is uniform over $\{0,1\}^n - \mathcal{Q}$, where the cardinality of $\mathcal{Q}$ equals the total number of forward and inverse queries to E using key K. Indeed, suppose that so far t queries to E have been made involving key K, resulting in t plaintext-ciphertext pairs (X_i, Y_i) with $Y_i = E_K(X_i)$ for $i = 1, \ldots, t$. The answer to a fresh query to E_K will therefore be $Y^* \neq Y_i, i = 1, \ldots, t$. Moreover, each of the $2^n - t$ answers is equally likely if E is an ideal cipher. Each possible answer Y^* will combine under C_{OUT} with the pair (M, V) consistent with the (K, X) query being made, leading to a possible compression function outcome H^*. Because C_{OUT} is bijective when (M, V) are fixed, distinct Y^* lead to distinct H^*, so there are $2^n - t$ possible outcomes H^*, all equally likely.

Similarly, consider an inverse query (K, Y). This yields a unique X and hence by bijectivity of C_{IN}, there is a unique pair (M, V) corresponding to this query once answered. Thus, each inverse query will add one triple (M, V, H) to the adversary's list of computable values. This time the third criterion implies that the distribution of H is uniform over $\{0,1\}^n - \mathcal{Q}$, where the cardinality of $\mathcal{Q}$ equals the total number of forward and inverse queries to E using key K. Indeed, suppose that so far t queries to E have been made involving key K, resulting in t plaintext-ciphertext pairs (X_i, Y_i) with $Y_i = E_K(X_i)$ for $i = 1, \ldots, t$. The answer to a fresh query to E_K^{-1} will therefore be $X^* \neq X_i, i = 1, \ldots, t$. Moreover, each of the $2^n - t$ answers is equally likely if E is an ideal cipher. Each possible answer X^* will combine under C_{IN}^{-1} and C_{OUT} with K and Y to a triple (M, V, H). Because for all K and Y the mapping from X to H is bijective, distinct X^* lead to distinct H^*, so there are $2^n - t$ possible outcomes H^*, all equally likely.

As a result, after i queries the list of computable values contains i triples (M, V, H). The $i + 1$'th query will add one triple with H uniform over a set of size at least $2^n - i$. Thus the probability that the $i + 1$'st query causes a collision with any of these triples is at most $i/(2^n - i)$. Using a union bound, the probability of a collision after q queries can then be upper bounded by $\sum_{i=1}^{q-1} i/(2^n - i) \leq \frac{1}{2}q(q-1)/(2^n - q)$. *Q.E.D.*

Note that our bound is slightly tighter than the bound $q(q+1)/2^n$ given by Black et al. There are two reasons for this. Firstly, Black et al. also consider finding a preimage to the initial vector a collision. Although this is certainly something that has to be taken into account when iterating the compression function, the concept of initial vector is somewhat alien to the compression function when considered on its own. In any case, it would have to affect of adding an extra query, which is why the numerator changes from $q(q-1)$ in our bound to $q(q+1)$ in theirs. Secondly, Black et al. use the bound $2(2^n - q) > 2^n$, at least for $q < 2^{n-1}$ (and for larger q the bound becomes vacuous anyway) to further simplify the expression.

3.2 Type-II: Collision Resistance in the Iteration

Definition 3 *A rate-1 blockcipher based compression function is called to be of Type-II iff the following three hold:*

1. *C_{IN} is bijective.*
2. *for all M, V the postprocessing $C_{\text{OUT}}(M, V, \cdot)$ is bijective.*
3. *for all K, $C_{\text{IN}}^{-1}(K, \cdot)$ restricted to V, its second output, is bijective.*

Theorem 4 *Let H^E be a Type-II compression function. If E is an ideal cipher with block size n, then the advantage of an adversary in finding a collision in the iterated hash function $\mathcal{H}^{H^E}$ after q queries is at most $\frac{1}{2}q(q+1)/(2^n - q)$.*

Proof: Consider an undirected graph with as nodes all the $N = 2^n$ possible chaining values $\{0,1\}^n$. We dynamically add edges based on the queries to E and E^{-1}. In particular, we add an edge (V, H), labelled by M, if we know a message M such that $H = H^E(M, V)$ and the relevant query to either E or E^{-1} has been made. We claim that to find a collision would require constructing a rho shape containing the (random) initial vector. Suppose that $\mathcal{H}(\mathbf{M}) = \mathcal{H}(\mathbf{M}')$ with $\mathbf{M} \neq \mathbf{M}'$. Write $\mathbf{M} = (M_1, \ldots, M_\ell)$ and $\mathbf{M}' = (M'_1, \ldots, M'_{\ell'})$ and correspondingly $V_1, \ldots, V_\ell$ respectively $V'_1, \ldots, V'_{\ell'}$ for the chaining values of the iterated hash. Note that $V'_0 = V_0$ and $V_\ell = V'_{\ell'}$. Assume $\ell \leq \ell'$. Because $M \neq M'$, there exists a t such that $M_i = M'_i$ for all $0 \leq i < t$ but $M_t \neq M'_t$ (or possibly $\ell < t \leq \ell'$). As a result, the paths $(V_0, \ldots, V_t)$ and $(V'_0, \ldots, V'_t)$ are identical, but the edges (V_t, V_{t+1}) and (V'_t, V'_{t+1}) are distinct, even when V_{t+1} happens to equal V'_{t+1} (in particular, the edges are labelled differently). Since $V_\ell = V'_{\ell'}$ at some point the paths need to come together again, completing the rho shape. Note that not every rho will lead to a collision though due to our use of an undirected graph.

Since we are dynamically adding edges to the graph, components in the graph will also grow dynamically. Let T be the set of all nodes that are in a component containing a cycle or the initial vector. The first claim is that after i queries, the set T has cardinality at most $i+1$. Indeed, the component containing the initial vector has at most $i'+1$ nodes when i' edges are used. A cyclic component based on i' edges has at most i' nodes. Thus the initial vector component is the only component in T that causes the number of nodes larger than the number of edges, by at most one. Bijectivity of C_{IN} implies that a query (either forward or inverse) will add at most one edge to the graph, so after i queries, there are at most i edges in the entire graph and at most $i+1$ nodes in T.

The second claim is that to complete a rho shape, either a cycle has to be completed within the IV-component, or the IV-component needs to be connected with a cycle. Either way, an edge has to be found of which both nodes are already part of T. For a forward query, criterion two ensures that the distribution of H is uniform over $\{0,1\}^n - \mathcal{Q}$, where the cardinality of $\mathcal{Q}$ equals the total number of forward and inverse queries to E using the current key (cf. the proof of Theorem 2). Consequently, the probability that on the i'th query a collision is found by a forward query is at most $i/(2^n-i)$. Similarly, for an inverse query the third criterion ensures that the distriburion of V is uniform over $\{0,1\}^n - \mathcal{Q}'$, where the cardinality of $\mathcal{Q}'$ equals the total number of forward and inverse queries to E using the current key. This upper bounds the probability of finding a collision on the i'th query using an inverse query by $i/(2^n-i)$.

We can now wrap up and conclude that the probability of finding a collision on the i'th query is at most $i/(2^n-i)$ and the probability after q queries is at most $\sum_{i=1}^{q} i/(2^n-i) \leq \frac{1}{2}q(q+1)/(2^n-q)$.

Q.E.D.

4 Implications to PGV-Style Schemes

In this section we investigate how our results relate to PGV-style schemes and we compare with the classification of Black et al. The PGV-style schemes are a special subclass of the schemes we consider. Firstly, the blocksize and keysize of the blockcipher are assumed equal to eachother. For C_{IN} and C_{OUT} only certain xor-based combinations are allowed. In particular, the compression function will look like $H^E(M,V) = E_K(X) \oplus C$ where $K, X, C \in S, M, V, M \oplus V$ where S is some fixed, publicly known bitstring.

We will represent the way K, X, and C are functions of M and V by elements in $\mathbb{Z}_2^2$, where a vector $\mathbf{x} \in \mathbb{Z}_2^2$ corresponds to $X = \mathbf{x} \cdot \binom{M}{V}$. We will also write $\mathbf{x} = (x_M, x_V)$. For instance $\mathbf{x} = (10)$ has $x_M = 1$ and $x_V = 0$, corresponding to $X = M$. We ignore any affine part, so $\mathbf{c} = (00)$ corresponds to the aforementioned $C = S$). Also note that we make a distinction between the linear map $\mathbf{x} \in \mathbb{Z}_2^2$ and the value $X \in \{0,1\}^n$. Since there are 4 elements in $\mathbb{Z}_2^2$ and we have to pick 3 ($\mathbf{k}, \mathbf{x}$, and $\mathbf{c}$), there are 64 constructions to consider in total. Of these 64 schemes, 12 are known to be collision resistant compression functions, another 8 are secure in the iteration.

The big question is how does this compare to our result. For this we need to cast the four conditions we have one C_{IN} and C_{OUT} in terms of $(\mathbf{k}, \mathbf{x}, \mathbf{c})$. First let's observe that $\binom{K}{X} \leftarrow C_{\text{IN}}(M,V) = \binom{\mathbf{k}}{\mathbf{x}}\binom{M}{V}$ and $H \leftarrow C_{\text{OUT}}(M,V,Y) = Y \oplus \mathbf{c}\binom{M}{V}$. Finally $H \leftarrow C_{\text{BACK}}(K,X,Y) = Y \oplus \mathbf{c}\binom{\mathbf{k}}{\mathbf{x}}^{-1}\binom{K}{X}$. For future reference, for invertible matrices $\binom{\mathbf{k}}{\mathbf{x}} \in \mathbb{Z}_2^{2\times 2}$ it holds

$$\begin{pmatrix} k_M & k_V \\ x_M & x_V \end{pmatrix}^{-1} = \begin{pmatrix} x_V & k_V \\ x_M & k_M \end{pmatrix} .$$

We are now ready to see what the requirements from Definitions 1 and 3 mean in terms of the vectors $\mathbf{k}, \mathbf{x}$ and $\mathbf{c}$. The results are listed in Table 1.

1. C_{IN} is bijective.
   This is equivalent to $\binom{\mathbf{k}}{\mathbf{x}}$ being invertible, which reduces the choice of $\mathbf{k}$ and $\mathbf{x}$ from 16 to 6.
2. for all M, V the postprocessing $C_{\text{OUT}}(M, V, \cdot)$ is bijective.
   This is always the case for the schemes at hand, as can be easily verified.

3.I for all K, Y the modified postprocessing $C_{\text{BACK}}(K, \cdot, Y)$ is bijective.
For this we require $Y \oplus \mathbf{c}\binom{\mathbf{k}}{\mathbf{x}}^{-1}\binom{K}{X}$ to be bijective as a function of X for all Y and K. This simplifies to $(c_M k_V \oplus c_V k_M)X$ being bijective, or alternatively $c_M k_V \oplus c_V k_M = \det\binom{\mathbf{k}}{\mathbf{c}} = 1$. This is equivalent to stating that $\binom{\mathbf{k}}{\mathbf{c}}$ is invertible or that $\mathbf{c}$ is not in the span of $\mathbf{k}$ (given that $\mathbf{k} \neq (00)$ due to the first criterion). For a given invertible $\binom{\mathbf{k}}{\mathbf{x}}$ this means exactly two values are possible for $\mathbf{c}$ (since (00) and $\mathbf{k}$ are ruled out), yielding $6 \cdot 2 = 12$ schemes in total. These are exactly the PGV schemes, or the Type-I schemes as defined by Black et al.

3.II for all K, $C_{\text{IN}}^{-1}(K, \cdot)$ restricted to V, its second output, is bijective.
We need for all K that $\binom{M}{V} \leftarrow \binom{\mathbf{k}}{\mathbf{x}}^{-1}\binom{K}{X}$ is bijective as a function from X to V. This is true iff k_M equals 1, so $\mathbf{k} = (11)$ or $\mathbf{k} = (10)$. Each of these two choices of $\mathbf{k}$ comes with two possible choices of $\mathbf{x}$ to make $\binom{\mathbf{k}}{\mathbf{x}}$ invertible. There are four choices for $\mathbf{c}$ for a given pair $(\mathbf{k}, \mathbf{x})$, however, two of the choices for $\mathbf{c}$ are already covered by the Type-I schemes, so we only get two new cases, namely $\mathbf{c} = (00)$ and $\mathbf{c} = \mathbf{k}$. All in all 8 schemes are exclusively Type-II. These schemes correspond to the Type-II schemes identified by Black et al.

$\binom{\mathbf{k}}{\mathbf{x}}\backslash\mathbf{c}$	(00)	(10)	(01)	(11)
$\begin{pmatrix}1&0\\0&1\end{pmatrix}$	$E_M(V)^{15}$	$E_M(V) \oplus M^{17}$	$E_M(V) \oplus V^{5}$	$E_M(V) \oplus M \oplus V^{7}$
$\begin{pmatrix}0&1\\1&0\end{pmatrix}$	insecure	$E_V(M) \oplus M^{1}$	insecure	$E_V(M) \oplus M \oplus V^{3}$
$\begin{pmatrix}1&0\\1&1\end{pmatrix}$	$E_M(M \oplus V)^{19}$	$E_M(M \oplus V) \oplus M^{20}$	$E_M(M \oplus V) \oplus V^{8}$	$E_M(M \oplus V) \oplus M \oplus V^{6}$
$\begin{pmatrix}0&1\\1&1\end{pmatrix}$	insecure	$E_V(M \oplus V) \oplus M^{4}$	insecure	$E_V(M \oplus V) \oplus M \oplus V^{2}$
$\begin{pmatrix}1&1\\1&0\end{pmatrix}$	$E_{M\oplus V}(M)^{13}$	$E_{M\oplus V}(M) \oplus M^{9}$	$E_{M\oplus V}(M) \oplus V^{11}$	$E_{M\oplus V}(M) \oplus M \oplus V^{14}$
$\begin{pmatrix}1&1\\0&1\end{pmatrix}$	$E_{M\oplus V}(V)^{16}$	$E_{M\oplus V}(V) \oplus M^{12}$	$E_{M\oplus V}(V) \oplus V^{10}$	$E_{M\oplus V}(V) \oplus M \oplus V^{18}$

Table 1: The 20 Secure PGV-style schemes. Superscripted are the $\imath$-indices from [1, Fig. 1 and 2]

This leads us to the following tightening of Black et al.'s result on the collision resistance of PGV-style Type-II schemes.

Corollary 5 *Let $\mathcal{H}^H$ be an iterated hash function with blockcipher-based compression function $H^E(M, V) = E_K(X) \oplus C$, where $K = \mathbf{k}\binom{M}{V}$, $X = \mathbf{x}\binom{M}{V}$, and $C = \mathbf{c}\binom{M}{V}$ for $\mathbf{k}, \mathbf{x}, \mathbf{c} \in \mathbb{Z}_2^2$. If $\binom{\mathbf{k}}{\mathbf{x}}$ is invertible, $\mathbf{k} \neq (01)$, then finding a collision in $\mathcal{H}^H$ for a random initial vector V_0 given q calls to E and E^{-1} succeeds with probability at most $\frac{1}{2}q(q+1)/(2^n - q)$, where n is the blocksize (and keysize) of E.*

References

1. 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 2442 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
2. S. Lucks. A collision-resistant rate-1 double-block-length hash function. In E. Biham, H. Handschuh, S. Lucks, and V. Rijmen, editors, *Symmetric Cryptography*, number 07021 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
3. S. Matyas, C. Meyer, and J. Oseas. Generating strong one-way functions with cryptographic algorithms. *IBM Technical Disclosure Bulletin*, 27(10a):5658–5659, 1985.
4. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
5. S. Miyaguchi, M. Iwata, and K. Ohta. New 128-bit hash function. In *Proceedings 4th International Joint Workshop on Computer Communications*, pages 279–288, 1989.
6. 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.