Multicollision Attacks on Generalized Hash Functions

M. Nandi[1] and D. R. Stinson[2]

[1] Applied Statistics Unit, Indian Statistical Institute, Kolkata, India
mridul_r@isical.ac.in

[2] School of Computer Science, University of Waterloo, Waterloo, Canada
dstinson@uwaterloo.ca

Abstract. In a recent paper in crypto-04, A. Joux [6] showed a multicollision attacks on the classical iterated hash function. He also showed how the multicollision attack can be used to get a collision attack on the concatenated hash function. In this paper we have shown that the multicollision attacks exist in a general class of sequential or tree based hash functions even if message blocks are used twice unlike the classical hash function.

1 Introduction

A hash function is a function from an arbitrary domain into a small fixed size domain. This has been popularly used in many public key crypto-systems like digital signature schemes, public key encryption schemes etc. Usually it is used as a preprocessor as it is much faster than the computation of the other public key cryptographic primitives. To have the security of those primitives the hash functions should satisfy some security assumptions. The most common security assumptions are collision resistance and pre-image resistance. Intuitively it is computationally hard to find two different inputs of a collision resistant hash function which give same output. For preimage resistant hash function it is computationally hard to find an inverse of a randomly given image.

A hash function $H : \{0,1\}^* \to \{0,1\}^n$ is usually designed from a compression function $f : \{0,1\}^{n+m} \to \{0,1\}^n$. There are many constructions of compression function from scratch e.g. SHA-1, SHA-256 [12] MD-family i.e. MD-4, MD-5, RIPEMD [4] [14] etc. There are several collision attacks [2] [3] [7] [17] on some of these compression functions. The most popular design of a hash function is the classical iteration or MD method [1] [11]. Here, the compression function is used sequentially and for each invocation of f a block (or a part) of the message is used for hashing. There are other methods to design a hash function from an underlying compression function which can be characterized by a directed tree [15]. These are known as tree-based hash functions. One advantage of using a tree based hash function is that it can be implemented in parallel.

Multicollision is a general concept of collision. A multicollision is a set of elements whose output values are same. The multicollision has been used earlier in many literatures [5] [8] [9] [13] [16] to find a collision attack. For a random looking function $H : \{0,1\}^* \to \{0,1\}^n$ any K-way multicollision attack (i.e. to find a set of size K whose outputs are same) requires $\Omega(2^{(K-1)n/K})$ many queries of H. But recently, A. Joux [6] found a K-way multicollision of the classical iterated hash function in time $\mathrm{O}(\log(K).2^{n/2})$. He also showed a collision attack on bigger output hash function $H||G$ where G can be any function. So concatenation of two hash functions where one of them is a classical iterated hash function is not maximally secure against collision attack. The attack on $H||G$ uses the multicollision on H and the complexity is $\mathrm{O}(\log(K).2^{n/2})$ if G also has output size n. So it is not desirable to use a hash function having multicollision for extending the size of the output. Very recently, S. Lucks [10] constructed a twin pipe hash function which is secure against multicollision attack assuming the underlying compression function is a random function.

Motivation and Our Contribution.

In the light of the above discussion it would be interesting to construct a function secure against multicollision attack (like in [10]) or give some multicollision attacks for different constructions. In this paper we have discussed some attacks on a class of generalized hash functions which includes a large number of natural extensions. We have shown that the multicollision attacks exist in generalized sequential or tree based hash functions even if message blocks are used twice unlike the classical hash function.

2 Multicollision of Classical Hash Function.

A collision on a function $g : D \to R$ is a doubleton subset $\{x, y\}$ of D such that $g(x) = g(y)$. Similarly for $r \geq 2$, a r-way collision (or multicollision) is a subset $\{x_1, \ldots, x_r\}$ (we say, *multicollision subset*) of D such that $g(x_1) = g(x_2) = \ldots = g(x_r) = z$ (say). The common output value z is known as the *collision value* for the multicollision set. Now consider the following attack :

r-way Collision Attack or multicollision attack : Find a multicollision subset $\mathcal{C}$ of size r (≥ 2), for the function g. In case of $r = 2$ it is nothing but the popularly known collision attack.

Complexity in a random oracle model. A function $g : D \to R$ is said to be a random function if for any $k > 0$ and k distinct inputs $x_1, x_2, \cdots, x_k \in D$, $g(x_i)$'s are independently and uniformly distributed on R. So, unless the value of $g(x)$ is computed, $g(x)$ will be uniformly distributed on R. We say a function g is modelled as a random oracle if the function g is assumed to be a random function. The complexity of an attack is the number of computations of the function $g(\cdot)$ to be queried. If some function is defined based on $g(\cdot)$ which is assumed to be a random function then complexity of any attack algorithm on that function is the number of computations of g to be queried. Now we state some facts regarding multicollision attack in a random oracle model.

Fact 1 *Let $g : D \to R$ be a function which is modelled as a random oracle. Then for any adversary finding r-way collisions has complexity $\Omega(|R|^{\frac{r-1}{r}})$. The complexity of the birthday attack for r-way collision is $\mathrm{O}(|R|^{\frac{r-1}{r}})$.*

So in the random oracle model the birthday attack is the best attack for finding r-way collision. In case of hash functions we usually first design a compression function $f : \{0,1\}^{n+m} \to \{0,1\}^n$ where $m > 0$ and then we design a method which extends the domain into a large arbitrary domain. For example the MD-method, where the hash function $H^f : (\{0,1\}^m)^* \to \{0,1\}^n$ based on the compression functions f is defined as in below. Here h_0 is some fixed initial value and $|m_i| = m$.

```
**Algorithm** $H^f(m_1||\dots||m_l)$
**For** $i = 1$ **to** $l$
$h_i = f(h_{i-1}, m_i)$
**Return** $h_l$
```

2.1 Joux's Multicollision attack on H^f

In a recent paper by Joux [6], it was shown that there is a 2^r-way collision attack for the above classical iterated hash function with complexity $\mathrm{O}(r2^{n/2})$ which is much less than $\Omega(2^{\frac{n(2^r-1)}{2^r}})$ (the complexity for random oracle model, see Fact 1). It also proves that H^f is not a random function. Here, by complexity we mean the number of invocations of f as H^f is defined based on f which is assumed to be a random function. The idea of the attack is to find first r successive collisions

$$f(h_{i-1}, m_i^1) = f(h_{i-1}, m_i^2) = h_i,\ 1 \le i \le r$$

So $H(m_1^{i_1}||\cdots||m_r^{i_r}) = h_r$ where, $i_1, \cdots, i_r \in \{1,2\}$. So, we have 2^r-way collision by finding only r successive 2-way collisions and hence the complexity of the attack is $\mathrm{O}(r.2^{n/2})$.

Application of Multicollision.

In the same paper by Joux [6], it was shown that how multicollision can be used. Let $H : D \to \{0,1\}^n$ be some hash function which has $2^{n/2}$-way multicollision with time complexity $q(n)$. Then for any other function $G : D \to \{0,1\}^n$, $H(\cdot)||G(\cdot)$ has collision in the time complexity $q(n)$ assuming one query is needed to compute G and $q(n) \ge 2^{n/2}$. So if $q(n) << 2^n$ (which is the complexity for birthday attack on $H||G$) then we have a better attack than the birthday attack. Basically, we will search a collision for $G(\cdot)$ from the multicollision set of $H(\cdot)$ and a collision is guaranteed in the random oracle model if the search space has desirable size (here $2^{n/2}$). When $H(\cdot)$ is the classical hash function $q(n) = n2^{n/2}$ and so the collision attack on the concatenated hash function $H||G$ has complexity $\mathrm{O}(n2^{n/2})$.

3 Multicollision Attack on Generalized Sequential Hash Function.

For each l, we can correspond a sequence viz. $< 1, 2, \ldots, l >$ with the classical algorithm. The i^{th} value of the sequence represents the message-block number involved in the i^{th} loop of the algorithm. We can generalize this idea by considering an arbitrary sequence letting the block number repeated more than once. For example, for each l consider the sequence $< 1, 2, \ldots, l, 1, 2, \ldots, l >$. If $H(IV, M)$ is the classical iterated hash function with the initial value IV then the hash function based on the sequence $< 1, 2, \ldots, l, 1, 2, \ldots, l >$ is $H(H(IV, M), M)$. At first glance, it looks secure against multicollision attack as the Joux's attack will not work here. But a variant of Joux's attack can be applied to this hash function. In fact we will prove that for a large class of sequential constructions there are multicollision attacks. First we define the generalized sequential hash function.

3.1 Generalized Sequential Hash Function.

For each positive integer l, we have a positive integer $s = s(l)$ and a sequence $\alpha^l = < \alpha^l(1), \alpha^l(2), \ldots, \alpha^l(s) >$ where, $\alpha^l(i) \in \mathrm{Z}_l := \{1, 2, \ldots, l\}$. We use α and $\alpha(i)$ (or α_i) instead of α^l and $\alpha^l(i)$ respectively when there is no confusion. We can define the hash function $H : \{0,1\}^n \times (\{0,1\}^m)^l \rightarrow \{0,1\}^n$ based on the sequence α^l. Let $M = m_1|| \ldots ||m_l$ where $|m_i| = m$ for each i.

```
**Algorithm** $H(m_1|| \ldots ||m_l)$
**For** $i = 1$ **to** $s$
$h_i = f(h_{i-1}, m_{\alpha_i})$
**Return** $h_s$
```

The sequence corresponding to the classical hash function is $< 1, 2, \ldots, l >$. We can correspond a generalized sequential hash function $H : \{0,1\}^n \times (\{0,1\}^m)^* \rightarrow \{0,1\}^m$ by an infinite sequence $< \alpha^1, \alpha^2 \cdots >$. Here an element of the sequence is a finite sequence. The l^{th} sequence represents the function which hashes the l-block messages. We will assume that each element of Z_l appears in the sequence α^l. In other words, all message blocks are used at least once to get the final hash value otherwise there is a trivial collision attack on those hash functions and hence no need to study those hash functions.

3.2 Some Terminologies on Sequence

Consider a finite sequence $\alpha = < \alpha_1, \alpha_2, \cdots, \alpha_s >$ of $\mathrm{Z}_l := \{1, 2, \cdots, l\}$. The *length* of the sequence is s and it is denoted by $|\alpha|$. The *index set* of the sequence is $[1,s] := \{1, 2, \cdots, s\}$. For any subset $I = \{i_1, \cdots, i_t\}$ of the index set $[1,s]$ we have a subsequence $\alpha(I) = < \alpha_{i_1}, \cdots, \alpha_{i_t} >$ where, $i_1 < \cdots < i_t$. I is said to be a *subinterval* if $I = [i,j] \subset [1,s]$ and a *left-end subinterval* if $i = 1$.

Definition 1. (**Independent Elements in a Subsequence**)
Distinct elements $x_1, \ldots, x_d$ from Z_l are said to be independent *in the subsequence $\alpha(I)$ if there exist d disjoint and exhaustive subintervals $I_1, \ldots, I_d$ (i.e. union gives the whole set I) such that x_i appears in the the subsequence $\alpha(I_i)$ but not in $\alpha(I_k)$ for $k \neq i$.*

We write $\mathcal{N}(\alpha(I)) := \max\{d : \exists\ d$ independent elements in the subsequence $\alpha(I)\}$. $x_1, \cdots, x_t$ will not be independent in a sequence α if there is a subsequence $< x_i, x_j, x_i >$ of α for some $1 \leq i \neq j \leq t$. Note, if there are k elements which appear only once in the sequence α then $\mathcal{N}(\alpha) \geq k$. For a sequence α of Z_l and $x \in \mathrm{Z}_l$ we write, $\mathrm{freq}(x, \alpha)$ (or simply, $\mathrm{freq}(x)$) $= |\{i : \alpha(i) = x\}|$ (*frequency of x*). It denotes the number of times x appears in the sequence α. We also write $\mathrm{freq}(\alpha)$ (*frequency of the sequence*) for the maximum frequency over all elements from the sequence. More precisely, $\mathrm{freq}(\alpha) = \max\{\mathrm{freq(x)} : x \in \mathrm{Z}_l\}$. We will show some multicollision attacks on sequential hash functions based on sequences where *frequency* is at most two. Consider the following two examples.

Example 1. Let $\vartheta^{(1)} =< 1, 2, \ldots, l >$ (the sequence for classical hash function). Note that, $\mathcal{N}(\vartheta^{(1)}) = l$. Let $\vartheta^{(2)} =< 1, 2, \ldots, l, 1, 2, \ldots, l >$. It is easy to observe that there are no two independent elements in the sequence $\vartheta^{(2)}$ and hence $\mathcal{N}(\vartheta^{(2)}) = 1$. But, $\mathcal{N}(\vartheta^{(2)}([1,l])) = \mathcal{N}(\vartheta^{(1)}) = l$.

Example 2. Let $\theta^l =< 1, 2, 1, 3, 2, 4, 3, \ldots, l-1, l-2, l, l-1, l >$. Here, $\mathcal{N}(\theta^l) = \lfloor \frac{l+1}{2} \rfloor$ as $1, 3, \cdots, l$ (if l is odd) or $1, 3, \cdots, l-1$ (if l is even) are independent elements.

3.3 Multicollision Attack on Generalized Sequential Hash Function

Now we state a multicollision attack which says that for a sequence α, $\mathcal{N}(\alpha) = r$ we have 2^r-way collision attack on the hash function based on the sequence α. The complexity of the attack is $\mathrm{O}(s2^{n/2})$ where $s = |\alpha|$. First we illustrate our attack for the hash function based on $\theta^5 =< 1, 2, 1, 3, 2, 4, 3, 5, 4, 5 >$ (also see Example 2). Here $1, 3, 5$ are independent elements in θ^5. We first fix the message blocks m_2, m_4 and m_6 by a string IV. Then find $m_1^1 \neq m_1^2$ such that

$$f(f(f(h_0, m_1^1), IV), m_1^1) = f(f(f(h_0, m_1^2), IV), m_1^2) = h_1.$$

Then similarly find $m_3^1 \neq m_3^2$ and $m_5^1 \neq m_5^2$ such that

$$f(f(f(f(h_1, m_3^1), IV), IV)m_3^1) = f(f(f(f(h_1, m_3^2), IV), IV)m_3^2) = h_2.$$
$$f(f(f(h_2, m_5^1), IV), m_5^1) = f(f(f(h_2, m_5^2), IV), m_5^2) = h_3.$$

Now it is easy to note that $\{m : m_i = m_i^1 \text{ or } m_i^2, i = 1,3,5, m_i = IV, i = 2,4,6\}$ is a multicollision set with collision value h_3.

Proposition 1. *Let H be a hash function based on a sequence $\alpha = \alpha^l$ where, $\mathcal{N}(\alpha) = r$. Then we have a 2^r-way multicollision attack on H with the complexity $\mathrm{O}(s2^{n/2})$ where $s = |\alpha|$.*

Proof. The idea of the proof is similar to that of the multicollision attack given by A.Joux [6] (also see Section 2). We define $H(h^*, [a,b], M) := h_b$ while computing $H(M)$ given that $h_a = h^*$. Note that h_a and h_b are the intermediate hash values at round a and b respectively. So, $H(h^*, [a,b], M)$ is the hash value at the round b provided we get the hash value h^* at the round a. As $\mathcal{N}(\alpha) = r$, we have r independent elements $x_1, \ldots, x_r$ and r disjoint and exhaustive intervals $I_1 = [a_1, b_1], \ldots, I_r = [a_r, b_r]$ where, $1 = a_1 \leq b_1 = a_2 \leq b_2 \cdots b_{r-1} = a_r \leq b_r = s$. Now fix all message blocks m_i by a string IV, where $i \notin \{x_1, \ldots, x_r\}$. As x_i's are independent $H(h^*, I_i, M)$ will only depend on m_{x_i} for all i. So, for simplicity, we write $H(h^*, I_i, m_{x_i})$ instead of $H(h^*, I_i, M)$. Now find r successive collisions as follows:

$$H(h_{i-1}, I_i, m_{x_i}^1) = H(h_{i-1}, I_i, m_{x_i}^2) = h_i,\ 1 \leq i \leq r.$$

Now, it is easy to check that, $\mathcal{C} = \{m_1 || \ldots || m_l; m_{x_1} = m_{x_1}^1 \text{ or } m_{x_1}^2, \ldots, m_{x_r} = m_{x_r}^1 \text{ or } m_{x_r}^2 \text{ otherwise } m_i = IV\}$ is a multicollision set of size 2^r. To get the i^{th} collision we need to query at most $|\alpha(I_i)|.2^{n/2}$. So in total we need to query at most $|\alpha| 2^{n/2}$. $\square$

In the above proposition if we take $l = 2r - 1$ then we have 2^r-way collisions on the hash function based on the sequence θ^l (See Example 2) with complexity $\mathrm{O}(r.2^{n/2})$. But we can not apply the same idea to the hash function based on the sequence $\vartheta^{(2)}$ (Example 1). Here, we have a different attack. Note that, $\mathcal{N}(\vartheta^{(2)}([1,l])) = l$ for any l. So we get multicollision up to some rounds and from that multicollision set we can again get r successive collisions.

Proposition 2. *Let H be a hash function based on α^l with $\mathrm{freq}(\alpha^l) \leq 2$. If there is a left-end subinterval I such that $\mathcal{N}(\alpha^l(I)) \geq rn/2$ then we have a 2^r-way multicollision of H with the complexity $\mathrm{O}(r^2 n.2^{n/2})$.*

Proof. Let $x_1, \cdots, x_k$ be independent elements in $\alpha(I)$ where $k = rn/2$. As in Proposition 1 we have a set $\mathcal{C} = \{M = m_1 || \ldots || m_l; m_{x_1} = m_{x_1}^1 \text{ or } m_{x_1}^2, \ldots, m_{x_k} = m_{x_k}^1\}$ of size 2^k so that $\mathcal{C}$ is a multicollision set for the hash function based on the sequence $\alpha(I)$. Let h_0' be the collision value for the multicollision set $\mathcal{C}$. Without loss of generality we assume that each x_i appears exactly once in the sequence $\alpha([a+1, s])$ in the same order as they appear in I where $I = [1, a]$ and s is the length of the sequence. Define, $\mathcal{C}_{i+1}$ for $0 \leq i \leq r-1$ as below:

$$\mathcal{C}_{i+1} = \{m_{x_{in/2+1}}^{j_1} || \cdots || m_{x_{(i+1)n/2}}^{j_{n/2}} : j_1, \cdots, j_{n/2} \in \mathrm{Z}_2\}$$

Now divide the interval $[a+1, s]$ into r disjoint and exhaustive subintervals $I'_1, I'_2, \cdots, I'_r$ so that $x_{in/2+1}, \cdots, x_{(i+1)n/2}$ appear in $I'_{i+1}, 0 \leq i \leq r-1$. To make notations simple we ignore all other message blocks as they are fixed by a string IV. We write $H(h^*, I'_{i+1}, m_{x_{in/2+1}} || \cdots || m_{x_{(i+1)n/2}})$ instead of $H(h^*, I'_{i+1}, M)$. Note, $|\mathcal{C}_i| = 2^{n/2}$. So, in the random oracle model we will get r successive collisions:

$$H(h'_{i-1}, I'_i, M_i^1) = H(h'_{i-1}, I'_i, M_i^2) = h'_i,\ 1 \leq i \leq r.$$

where, $M_i^1, M_i^2 \in \mathcal{C}_i$. Now it would be easy to observe that, $\mathcal{C}^* = \{M_1^{j_1} || \cdots || M_r^{j_r} : j_1, \cdots j_r \in \{1, 2\}\}$ is a multicollision set (of size 2^r). □

Till now we provide a multicollision attack if the underlying sequence satisfies some conditions. Now we will show that these conditions are satisfied by any sequence with frequency at most two.

Definition 2. *Given any subsequence $\alpha(I)$ of α define, $S(\alpha(I)) = |\{x \in \mathrm{Z}_l : \mathrm{freq}(x, \alpha(I)) \geq 1\}|$. Similarly, we can define $S^i(\alpha(I)) = |\{x \in \mathrm{Z}_l : \mathrm{freq}(x, \alpha(I)) = i\}|$. So, when $\mathrm{freq}(\alpha) \leq 2$ we have, $S(\alpha(i)) = S^1(\alpha(i)) + S^2(\alpha(i))$.*

Proposition 3. *Let α be a sequence of Z_l with $\mathrm{freq}(\alpha) \leq 2$ then either $\mathcal{N}(\alpha) \geq M$ or there exists a left-end subinterval I such that $\mathcal{N}(\alpha(I)) \geq N$ whenever $l \geq M.N$.*

Proof. We can assume that $S(\alpha) = l$ (the sequence represents the hash function which hashes l block messages). We will prove it by induction on l. Let $|\alpha| = s$. Note that $S^1(\alpha(I))$ increases as the interval grows. So, there will be a *left-end subinterval* $I = [1, t]$ with $S(\alpha(I)) \leq N$ such that either $\mathcal{N}(\alpha(I)) \geq S^1(\alpha(I)) = N$ or there exists one element say x_1 which appears twice in the sequence $\alpha(I)$. In the former case we are done so, assume the later. Remove all elements from α which appear in $\alpha(I)$ and call this new sequence by $\alpha_1 = \alpha(I_1)$ for some set I_1. Note, $S(\alpha_1) \geq M.N - N = (M-1)N$. By induction hypothesis either $\mathcal{N}(\alpha_1) \geq M-1$ or there exists a left-end subinterval J of the subsequence α_1 such that $\mathcal{N}(\alpha_1(J)) \geq N$. In the later case $\mathcal{N}(\alpha)([1, r]) \geq N$ where, r is the last element in the set J. So we are done. In the former case there exists $M-1$ independent elements $x_2, \ldots, x_M$ in the subsequence α_1. Also x_1 does not appear in the subsequence $\alpha[t+1, s]$ and $x_2, \ldots, x_M$ do not appear in $\alpha([1, t])$. So, $x_1, x_2, \ldots, x_M$ are independent elements in α. □

Now we have the multicollision attack for generalized sequential hash functions with frequency at most two. This is an immediate corollary of above Propositions 1, 2 and 3.

Theorem 1. *Let H be a hash function based on the sequence $< \alpha^1, \alpha^2, \cdots >$ with $\mathrm{freq}(\alpha^l) \leq 2$ for every $l \geq 1$. Then we have a 2^r-way multicollision of H with the complexity $\mathrm{O}(r^2 n.2^{n/2})$.*

4 Attacks on Generalized Tree-based Hash Functions

Similar results hold for generalized tree based hash function. First we define the generalized tree based hash function and some terminologies on tree.

4.1 Generalized Tree-based Hash Function

Here we will consider a compression function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ based on which a *(l-block) Generalized Hash Function* $H(\cdot)$ is defined.

1. Suppose that $m = m_1||m_2||\cdots||m_l$ is a l-block message with block size n i.e. $|m_i| = n$. We also have $h_1, h_2, \cdots \in \{0,1\}^n$ constants (fixed initial values which only depends on l).

2. Define a list of s ordered pairs $\{(x_j^1, x_j^2)\}_{1 \leq j \leq s}$. For $1 \leq j \leq s$, $x_j^1, x_j^2 \in \{h_1, h_2, \ldots\} \cup \{m_1, m_2, \cdots, m_l\} \cup \{z_1, \ldots, z_{j-1}\}$ and $z_j = f(x_j^1, x_j^2)$. For $j \neq s$, z_j's are known as *intermediate hash values* and z_s is known as the final hash value.

3. The final message digest for the message m is defined by the final hash value i.e. $H(m) = z_s$.

We can assume that each intermediate hash value z_i and each message block m_j are in the list and hence they are inputs of some invocations of f. So there are no message blocks and intermediate hash values which are not hashed. The above hash function also can be defined using a directed binary tree. We first define the directed binary tree and some terminologies.

Directed Binary Tree :

A directed binary tree is a directed tree so that each vertex has indeg either two or zero and outdeg exactly one except a vertex called the *root* which has zero outdeg. A *leaf* is a vertex with indeg zero. All other vertices or nodes (except the root) are known as *intermediate* nodes. So intermediate nodes have indeg 2 and outdeg 1. Now we state some terminologies on directed binary tree :

1. Let $G = (V, E)$ be a rooted directed tree with root $q \in V$ and the arc set $E \subset V \times V$. We write $v \rightarrow u$ for the arc $(v, u) \in E$ and $v \Rightarrow u$ either there is a path from the vertex v to u or $u = v$.

2. For a vertex v, define the subtree $G[v] = (V[v], E[v])$ induced by the vertices set $V[v] = \{u \in V : u \Rightarrow v\}$. We say the graph $G[v]$ is rooted at v.

3. We use the notation $L[G]$ (or simply L) for the set of leave of G and $L[v]$ for $L[G[v]]$, the set of leave of the graph $G[v]$. Note, $L[v] = L \cap V[v]$.

Generalized Tree based Hash Function :

Let $G = (V, E)$ be a rooted directed tree and $\rho : L \to [1, l] \cup \{0, 1\}^n$. If $\rho(v) \in [1, l]$ then it denotes the index of the message block. When $\rho(v) \in \{0, 1\}^n$, it denotes an initial value. Given a pair (G, ρ) and a l-block message $m = m_1|| \cdots ||m_l$ we will assign inductively a n-bit string on each vertex of G as follow:

1. For each leaf v assign an n-bit string m_i if $\rho(v) = i$ or assign h if $\rho(v) = h$.
2. For any other node v assign a n-bit string $f(z, z')$ where z and z' are assigned on the vertices u and u' and $u \to v$, $u' \to v$.

The output of the hash function $H(\cdot)$ is the value assigned on the root of the tree. Given a pair (G, ρ) there can be more than one ways of computation of final hash value. So (G, ρ) can be a characterization of several (l-block) generalized hash functions but as a function they are identical. But two different pairs always represent two different generalized hash functions. We say (G, ρ) is the *algorithm* for H. Now we will state some more terminologies which will be used in the multicollision attack.

1. For $x \in [1, l]$ we write freq(x, G) or simply freq(x) for the number of times x appears in the multi-set $\rho(L)$ (*frequency of* x). That is, freq(x) denotes the number of times the message block m_x is hashed to get the final hash. Define, freq$(G) = \max\{\text{freq}(x) : x \in L\}$.

2. We define the hash output at v (i.e. the value assigned on v while the message is m) by $H(v, m)$. Note that, a message block m_i is used to compute $H(v, m)$ if and only if i is in $\rho(L[v])$. Sometimes we also use $H(v, m_i)$ for $H(v, m)$ when $H(v, m)$ only depends on the i^{th} message block i.e. the only index appearing in $\rho(L[v])$ is i.

3. Given any vertex v define, $S(v, G)$ (or simply $S(v)$) $= |\{x \in [1, l] : \text{freq}(x, G[v]) \geq 1\}|$. Similarly, we can define $S^i(v) = |\{x \in \mathrm{Z}_l : \text{freq}(x, G[v]) = i\}|$. So $S(v)$ (or $S^i(v)$) denotes the number of message blocks which are hashed at least once (or exactly i many times respectively) to compute $H(v, m)$.

Definition 3. (independent sequence of message indices)
Given an algorithm (G, ρ), $(x_1, x_2, \ldots, x_k)$ is an independent *sequence of message indices if there exists vertices $v_1, v_2, \ldots, v_k \in V$ such that*

1. *All occurrences of x_i are in $\rho(L[v_i])$ for all i.*
2. *$x_i \notin \rho(L[v_j])$ for all $i > j$.*
3. *$v_k = q$, the root of the directed binary tree G.*

We use the notation $\mathcal{N}(v)$ to denote the maximum value of k such that there exists an independent sequence of message indices in $G[v]$ of length k. In particular, $\mathcal{N}(q)$ denotes the maximum length of an independent sequence of message indices in the graph G. We say v_i as a corresponding vertex of x_i.

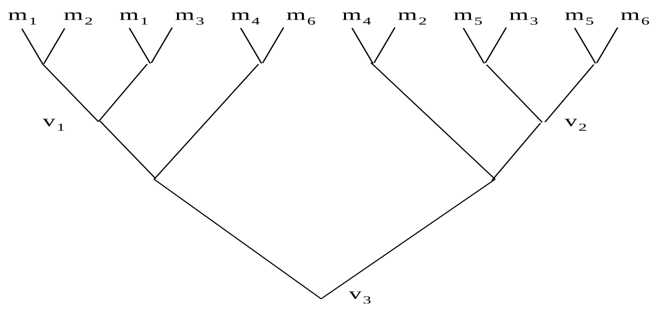
m_1 m_2 m_1 m_3 m_4 m_6 m_4 m_2 m_5 m_3 m_5 m_6
v_1
v_2
v_3

Fig. 1. An example of 6-block binary tree based hash function.

Because of condition 2 in the Definition 3, the order of independent elements are important. So $(x_2, x_1, x_3, \cdots, x_k)$ may not be independent even if $(x_1, x_2, \cdots, x_k)$ is an independent sequence.

In the Figure 1, $(1, 5, 4)$ is an *independent* sequence. Here the corresponding vertices of $1, 5$ and 4 are v_1, v_2 and v_3 respectively (shown in the figure 1). Note that $(4, 1, 5)$ is not an independent sequence as only vertex v such that all occurrence of 4 in $\rho(L[v])$ is v_3. One can also check that $(5, 4)$ is still an independent sequence in $G - G[v_1]$ and 1 does not appear in $G - G[v_1]$. In general we have the following lemma :

Lemma 1. *If* $(x_1, x_2, \ldots, x_k)$ *is an* independent *sequence in* G *then* $(x_2, \cdots, x_k)$ *is also an independent sequence in* $G - G[v_1]$ *where,* v_1 *is a corresponding vertex of* x_1. *Also we have,* $x_1 \notin \rho(L[G - G[v_1]])$.

Proof. $x_1 \notin \rho(L[G - G[v_1]])$ since all occurrences of x_1 are in $\rho(L[v_1])$ (by the condition 1 of the Definition 3). Also it is easy to check that $(x_2, \cdots, x_k)$ is an independent sequence in $G - G[v_1]$. $\square$

Now we can state one of our main theorems of the section. It says that given a pair (G, ρ) if we have r independent elements in G then there is a 2^r-way collision attack for the hash function H based on the algorithm (G, ρ). The complexity of this attack is $O((s+1).2^{n/2})$ where, s is the number of intermediate nodes in G. The idea of the attack is very much similar to that of Joux's attack that is we will try to find r pairs (not collision pairs) $(m^1_{x_1}, m^2_{x_1}), \cdots (m^1_{x_r}, m^2_{x_r})$. And then we can combine all these pairs independently to have a 2^r-way collision attack.

In the example shown in the Figure 1, we first fix the message blocks m_2, m_3 and m_6 by a n-bit string say IV. Then find $m^1_1 \neq m^2_1$ such that $H(v_1, m^1_1) = H(v_1, m^2_1) = h^*_1$ by using $3.2^{n/2}$ computations of f. Now consider the graph $G_2 = G - G[v_1]$. Similarly we will find $m^1_5 \neq m^2_5$ such that $H(v_2, m^1_5) = H(v_2, m^2_5) = h^*_2$ by using $3.2^{n/2}$ computations of f. Now consider the graph $G_3 = G_2 - G[v_3]$ and the mapping $\rho_3(v_1) = h^*_1$, $\rho_3(v_2) = h^*_2$. For this pair (G_3, ρ_3), we can find

$m_4^1 \neq m_4^2$ such that $H(v_3, m_4^1) = H(v_3, m_4^2) = h_3^*$ by using $5.2^{n/2}$ computations of f. Now it is easy to check that the following set

$$\{m : m_i = IV, i = 2, 3 \text{ and } 6, m_j = m_j^1 \text{ or } m_j^2, j = 1, 4 \text{ and } 5\}$$

is a multicollision set with the collision value h_3^*. In this example we need $\mathrm{O}(11.2^{n/2})$ computations of f where 10 is the number of intermediate nodes. Now we will prove the theorem in more detail.

Theorem 2. *If $\mathcal{N}(q) = r$ then we have 2^r-way multicollision attack of H with the complexity $\mathrm{O}((s + 1).2^{n/2})$, where s is the number of the intermediate nodes in the binary directed tree G and q is the root of the binary tree.*

Proof. We will prove that if $(x_1, \cdots, x_r)$ is an independent sequence in G then a 2^r multicollision set of the form

$$\{m : m_j = m_{x_i}^1 \text{ or } m_{x_i}^2, \text{ if } j = x_i, \text{ for some } i, \text{ otherwise } m_j = IV\}$$

can be found in the complexity $\mathrm{O}((s + 1).2^{n/2})$. We will prove this by induction on r. Let v_i be a corresponding vertex of x_i. For $r = 1$ it is just a birthday attack on H varying the message block m_{x_1} and fixing all other message blocks by a string IV. For $r > 1$, we first fix all message blocks m_i by IV where, $i \notin \{x_1, \cdots, x_r\}$. Then we will find a pair $(m_{x_1}^1, m_{x_1}^2)$ with $m_{x_1}^1 \neq m_{x_1}^2$ such that $H(v_1, m_{x_1}^1) = H(v_1, m_{x_1}^2) = h_1^*$ (say) with complexity $t.2^{n/2}$ where, $t = |V[v_1] - L[v_1]|$. Now consider the graph $G' = G - G[v_1]$ and $\rho' : L[G'] \to [1, l] \cup \{0, 1\}^n$ where, $\rho'(v_1) = h_1^*$ and $\rho'(v) = \rho(v)$ for any other leaf v in $L[G']$. By lemma 1 we know that $(x_2, \cdots, x_r)$ is an independent sequence for the algorithm (G', ρ'). So by induction hypothesis we can find a 2^{r-1}-way collision set

$$\{m : m_j = m_{x_i}^1 \text{ or } m_{x_i}^2, \text{ if } j = x_i, 2 \leq i \leq r, \text{ otherwise } m_j = IV, j \neq x_1\}$$

with the collision value h^* (say) in time complexity $\mathrm{O}(|V' - L[G']|)$. Note that there is no occurrence of index x_1 in the multi-set $\rho'(L[G'])$ and if the intermediate hash value at the vertex v_1 is h_1^* then the final hash value for (G', ρ') is same as the final hash value for (G, ρ). So,

$$\{m : m_j = m_{x_i}^1 \text{ or } m_{x_i}^2, \text{ if } j = x_i, 1 \leq i \leq r, \text{ otherwise } m_j = IV\}$$

is a 2^r-way collision set with the collision value h^* and the complexity is $\mathrm{O}((|V' - L[G']| + |V[v_1] - L[v_1]|)2^{n/2}) = \mathrm{O}(|V - L[V]|.2^{n/2}) = \mathrm{O}((s + 1)2^{n/2})$. □

Now we will prove a simple fact related to a directed binary tree which would be useful to have a multicollision attack on generalized tree based hash functions. Recall that, $S(v)$ denotes the number of indices which appears in $\rho(L[v])$.

Lemma 2. *For any pair (G, ρ) with $S(q) \geq 2N$, there will be a vertex $v \in V$ with $N \leq S(v) \leq 2N$ where q is the root of the tree $G = (V, E)$.*

Proof. Let $u_1 \rightarrow v$, $u_2 \rightarrow v$. Then it is easy to check that $S(v) \leq S(u_1) + S(u_2)$. So, if $u_1 \rightarrow q$, $u_2 \rightarrow q$ then $S(u_1) + S(u_2) \geq 2N$. There will be one vertex say u_1 with $S(u_1) \geq N$. If $S(u_1) \leq 2N$ then the result follows for $v = u_1$. If not, we can continue and we will reach a vertex v with $N \leq S(v) \leq 2N$. □

Proposition 4. *Let $l = |S(q)|$ where q is the root of the tree. If $\mathrm{freq}(G) \leq 2$ then there is a vertex v such that $\mathcal{N}(v) \geq N$ or $\mathcal{N}(q) \geq M$ whenever $l \geq 2M.N$.*

Proof. We will prove it by induction on l. For $M = 1$, the statement is trivial as $\mathcal{N}(q) \geq 1$. So assume $M > 1$. Since $S(q) \geq 2MN \geq 2N$ by Lemma 2 there is a vertex v such that $N \leq S(v) \leq 2N$. Now if $S^1(v) = S(v) \geq N$ then $\mathcal{N}(v) \geq S^1(v) \geq N$. If $S^1(v) < S(v)$ then there is an element say x_1 which appears exactly twice in $\rho(L[v])$ (note, $\mathrm{freq}(G) \leq 2$). Let $G' = G - G[v]$. After we choose an index x_1 in $\rho(L[v])$, we want to make sure that no x_i $(i > 1)$ that is is chosen later on also occurs in $\rho(L[v])$. To prevent this from happening, we take all indices of message blocks in $\rho(L[v])$ and "remove" them from any other leaves in the graph, by fixing their values, before applying the inductive hypothesis. Formally, define $\rho'(v)$ and $\rho'(u)$ by a n bit string where, $u \in \rho(L[v]) \cap \rho(L[G'])$, otherwise, $\rho'(v) = \rho(v)$. Note, $S(G') \geq 2.M.N - 2.N = 2.(M-1)N$. By induction hypothesis for the graph G' either $\mathcal{N}(q) \geq M - 1$ or there exists a vertex u such that $\mathcal{N}(u) \geq N$. In the later case $\mathcal{N}(u) \geq N$ (for the graph G). Otherwise there exists $M-1$ independent elements $x_2, \ldots, x_M$ in the graph G'. Also x_1 does not appear in $\rho(L[G'])$ and $x_2, \cdots, x_M$ do not appear in $\rho(L[v])$. So, $x_1, x_2, \ldots, x_M$ are independent elements in G. □

So whenever $l \geq 2r^2.n$ either $\mathcal{N}(q) \geq r$ or there is a vertex v such that $\mathcal{N}(v) \geq rn = k$ (say). In the former case we already have a 2^r-way collision attack. In the later case we can do the same thing what we have done in the sequential case. Let $(x_1, \cdots, x_k)$ be an independent sequence. Find r vertices $v_1, v_2, \cdots, v_r = q$ in G' $(=G - G[v])$ such that the following happen :

1. $x_{in+1}, x_{in+2}, \cdots, x_{in+n/2} \in \rho(L(G'[v_i]))$ for all i.
2. $x_{in+1}, x_{in+2}, \cdots, x_{in+n/2} \notin \rho(L(G'[v_j]))$ for all $j < i$.

So, we first find 2^k-way collision on v. Then, we will find r successive collisions from the multicollision set. The idea of the attack is very much similar with that of sequential case so we ignore the detail. So we have our main theorem as follow:

Theorem 3. *If $\mathrm{freq}(G(H)) \leq 2$ then we have a 2^r-way multicollision with the complexity $\mathrm{O}(r^2 n.2^{n/2})$.*

4.2 A Note on Multi-Preimage Attack

For the sake of completeness we briefly study about the multi-preimage attack on generalized sequential or generalized tree-based hash function. we can define the following attack for a hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$.

r-way preimage (multi-preimage) attack : Given a random $y \in \{0,1\}^n$, find a subset $\mathcal{C} = \{x_1, \cdots, x_r\}$ of size r (≥ 1) such that $H(x_1) = \cdots = H(x_r) = y$.

The complexity for r-way preimage attack for a random function is $\Omega(r2^n)$ where, for generalized tree based or sequential hash function there is a r-way preimage attack with complexity $\mathrm{O}(2^{n/2})$. It is almost same with the multicollision attack. It starts exactly same as the multicollision attack and at the end instead of finding last collision we will look for output value as given image y. The complexity for last step is $\mathrm{O}(2^n)$ which will dominate the rest complexity $(r^2 n 2^{n/2})$ of multicollision attack.

5 Future Work and Conclusion

We have found a multicollision attack on a sequential or tree based hash function where the message blocks can be used two times unlike classical definition. All these construction can be viewed by a rooted directed tree or directed acyclic graph (DAG). One can look for other directed graphs in which there can be more than one path from an intermediate vertex to the root. That is, we can use the intermediate hash values more than once. Also one can try to give some attack where the message blocks can be used more than twice.

References

1. I. B. Damgård. *A design principle for hash functions*, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Sciences, Vol. 435, Springer-Verlag, pp. 416-427, 1989.
2. H, Dobbertin.*Cryptanalysis of MD4*. Fast Software Encryption, Cambridge Workshop. Lecture Notes in Computer Science, vol 1039, D. Gollman ed. Springer-Verlag 1996.
3. H, Dobbertin.*Cryptanalysis of MD5* Rump Session of Eurocrypt 96, May. http//www.iacr.org/conferences/ec96/rump/index.html.
4. H. Dobbertin, A. Bosselaers and B. Preneel. *RIPEMD-160: A strengthened version of RIPEMD*, Fast Software Encryption. Lecture Notes in Computer Science 1039, D. Gollmann, ed., Springer-Verlag, 1996.
5. M. Hattori, S. Hirose and S. Yoshida. *Analysis of Double Block Lengh Hash Functions.* Cryptographi and Coding 2003, LNCS 2898.
6. A. Joux. *Multicollision on Iterated Hash Function.* Advances in Cryptology, CRYPTO 2004, Lecture Notes in Computer Science 3152.
7. J. Kelsey. *A long-message attack on SHAx, MDx, Tiger, N-Hash, Whirlpool and Snefru.* Draft. Unpublished Manuscritpt.
8. L. Knudsen, X. Lai and B. Preneel. Attacks on fast double block length hash functions. *J.Cryptology, vol 11 no 1, winter 1998.*
9. L. Knudsen and B. Preneel. Construction of Secure and Fast Hash Functions Using Nonbinary Error-Correcting Codes. *IEEE transactions on information theory, VOL-48, NO. 9, Sept-2002.*
10. S. Lucks. *Design principles for Iterated Hash Functions*, eprint server: http://eprint.iacr.org/2004/253.

11. R. Merkle. *One way hash functions and DES*, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Sciences, Vol. 435, Springer-Verlag, pp. 428-446, 1989.
12. NIST/NSA. *FIPS 180-2 Secure Hash Standard*, August, 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf
13. B. Preneel. *Analysis and Design of cryptographic hash.* PhD Thesis , Katholieke Universiteit Leuven. 1995.
14. R. Rivest *The MD5 message digest algorithm.* http://www.ietf.org/rfc/rfc1321.txt
15. P. Sarkar. *Domain Extender for Collision Resistant Hash Functions: Improving Upon Merkle-Damgard Iteration* http://eprint.iacr.org/2003/173/.
16. T. Satoh, M. Haga and K. Kurosawa. *Towards Secure and Fast Hash Functions.* IEICE Trans. VOL. E82-A, NO. 1 January, 1999.
17. B. Schneier. *Cryptanalysis of MD5 and SHA*. Crypto-Gram Newsletter, Sept-2004. http://www.schneier.com/crypto-gram-0409.htm#3.