An Improved “Black Box” Measure for Evaluating Collision Resistance

Qi Wu
Department of Computer Science & Technology, School of Information Technology, Jiangxi University of Finance & Economics, Nanchang, China

Abstract—Collision resistance is one of the most desired properties for a cryptographic hash function. However, in the literature, there’re some insufficient “black box” measures for evaluating collision resistance, which couldn’t even distinguish some simple hash functions. In this paper, an improved “black box” measure is proposed based on reducing the probability with which a trivial Turing machine might find collision points. It works much better than the measures in the literature.

Keywords—cryptographic hash function; collision resistance; black box; Turing machine

I. INTRODUCTION

Nowadays, design and analysis of cryptographic hash functions have attracted much attention of researchers, due to their significant roles in data integrity, digital signature, and authentication protocols [1]. There are several requirements for cryptographic hash functions, such as pre-image resistance (also known as one-wayness), second pre-image resistance, and collision resistance, in which collision resistance is the most concerned one [2-11].

Informally speaking, there’re two different approaches for evaluating collision resistance for cryptographic hash functions: “white box” [2-5] and “black box” [6-11]. The “white box” approach examines the inner structure of cryptographic hash functions in detail, and outputs collision points through differential cryptanalysis. Basically, this approach is the mainstream of collision resistance evaluation. Meanwhile, there’s some work adopting the “black box” approach, which totally neglects the internal structure of cryptographic hash functions and calculates solely based on the input and output of cryptographic hash functions. However, we find their work is far from sufficient for collision resistance evaluation. Hereby, we propose our “black box” measure for evaluating collision resistance. Albeit our measure still seems to be naive and heuristic, it may make more sense than the previous work did.

The paper is organized as follows. Section 2 reviews the “black box” approach in the literature and points out its insufficiency. Section 3 gives our “black box” measure. Section 4 concludes.

II. THE “BLACK BOX” APPROACH IN THE LITERATURE

In the literature, researchers confine both the domain and range of cryptographic hash functions to 8 bits, namely, integers ranging from 0 to 255. Let the number of points in the range owning exactly \( k \) preimages be \( n(k) \), some researchers focus on \( n(0) \), and their measure [6-8] is:

\[
L = \frac{256 - n(0)}{256}.
\]  

They deem the closer \( L \) is to 1, the less possibly collisions occur.

Some other researchers focus on \( n(1) \), and their measure [9-11] is:

\[
T = \frac{n(1)}{256}.
\]  

They deem the larger \( T \) is, the less possibly collisions occur.

Intuitively, both their measures exactly comply with common sense: the less \( n(0) \) is, the larger \( n(1) \) is, the less possibly collisions occur. However, we could find many counterexamples easily, such as:

\[
H_1(x) = \left\lfloor \frac{x}{255} \right\rfloor,
\]  

\[
H_2(x) = \left\lfloor \frac{x}{128} \right\rfloor.
\]

We could see, intuitively, finding collision points of \( H_1 \) is much easier than those of \( H_2 \), because it’s quite likely that both random inputs fall in the interval \([0,254]\) while it’s much less likely that both random inputs fall in the interval \([0,127]\) or \([128,255]\). Nevertheless, both \( n(0) \) of the two hash functions equal 254, which means \( L \) couldn’t distinguish them at all.

Suppose
\[ H_3(x) = x \mod 4, \quad (5) \]
\[ H_4(x) = \begin{cases} 0, & x \leq 252 \\ x, & x > 252 \end{cases}. \quad (6) \]

We could know, intuitively, finding collision points of \( H_3 \) is much more difficult than those of \( H_4 \), because it’s improbable that both random inputs are congruent with modulus 4, while it’s much more likely that both random inputs fall in the interval \([0,252]\). Nonetheless, both \( n(0) \) of the two hash functions equal 252, which illustrates that \( L \) couldn’t tell them apart at all.

Let
\[ H_5(x) = \left\lfloor \frac{x}{4} \right\rfloor, \quad (7) \]
\[ H_6(x) = \begin{cases} x, & x \leq 62 \\ 63, & x > 62 \end{cases}. \quad (8) \]

We could see, intuitively, finding collision points of \( H_5 \) is much more difficult than those of \( H_6 \), because it’s not likely that both random inputs fall exactly in the interval \([0,3]\) or \([4,7]\) or … or \([252,255]\), whereas it’s much more likely that both random inputs fall in the interval \([63,255]\). However, both \( n(0) \) of the two hash functions equal 192, which illustrates that \( L \) couldn’t tell them apart at all.

Assume
\[ H_7(x) = 0, \quad (9) \]
\[ H_8(x) = \left\lfloor \frac{x}{2} \right\rfloor. \quad (10) \]

We could know, intuitively, finding collision points of \( H_7 \) is much easier than those of \( H_8 \), because collisions occur everywhere in \( H_7 \) such that each pair of random inputs will be its collision points whereas it’s improbable that both random inputs fall exactly in the interval \([0,1]\) or \([2,3]\) or … or \([254,255]\). Nonetheless, both \( n(1) \) of the two hash functions equal 0, which means \( T \) can’t tell them apart at all.

Suppose
\[ H_9(x) = \begin{cases} x \mod 64, & x \leq 127 \\ x, & x > 127 \end{cases}. \quad (11) \]
\[ H_{10}(x) = \begin{cases} 128, & x \leq 127 \\ x, & x > 127 \end{cases}. \quad (12) \]

We could see, intuitively, finding collision points of \( H_9 \) is much harder than those of \( H_{10} \), because it’s unlikely that both random inputs are less than 128 and congruent with modulus 64 whereas it’s quite probable that both random inputs fall in the interval \([128,255]\). Nonetheless, both \( n(1) \) of the two hash functions equal 128, which means \( T \) can’t tell them apart at all.

Let
\[ H_{11}(x) = \begin{cases} x, & x \leq 191 \\ \frac{x}{8}, & x > 191 \end{cases}. \quad (13) \]
\[ H_{12}(x) = \begin{cases} x, & x \leq 63 \\ 64, & x > 63 \end{cases}. \quad (14) \]

We could know, intuitively, finding collision points of \( H_{11} \) is much harder than those of \( H_{12} \), because it’s unlikely that both random inputs fall exactly in the interval \([0,7]\) or \([8,15]\) or … or \([184,191]\) while it’s quite probable that both random inputs fall in the interval \([64,255]\). Nevertheless, both \( n(1) \) of the two hash functions equal 64, which means \( T \) can’t tell them apart at all.

Next, let’s give our measure for evaluating collision resistance.

III. THE PROPOSED MEASURE FOR COLLISION RESISTANCE EVALUATION

As a cryptographic hash function, it should make the success rate of every Turing machine trying to find its collision points as low as possible, of course including the trivial one. Assume there’s a Turing machine \( M \) working as follows:

First, \( M \) randomly selects a point \( x_1 \) in the domain. Then, \( M \) randomly selects a point \( x_2 \) other than \( x_1 \) in the domain. At last, \( M \) outputs \( x_1 \) and \( x_2 \) as the collision points.

Let the points in the range owning exactly \( k \) preimages form a set \( N(k) \), then the success rate of \( M \) could be
calculated as follows:

\[
Succ(M) = Pr[H(x_1) = H(x_2)]
\]

\[
= Pr[x_1, x_2 \in H^{-1}(H(x_1)) \setminus \{x_1\}]
\]

\[
= \sum_{k=2}^{n} Pr[x_1, x_2 \in H^{-1}(H(x_1)) \setminus \{x_1\} \mid H(x_1) \in N(k)]Pr[H(x_1) \in N(k)]
\]

Apparently, the term in the summation equals 0 when \(k = 0\) or \(1\), because \(H(x_1)\) couldn’t belong to \(N(0)\) (it already has a preimage \(x_1\)) and when \(H(x_1) \in N(1)\), \(H^{-1}(H(x_1)) \setminus \{x_1\} = \emptyset\), which couldn’t contain \(x_2\) at all. Then, \(Succ(M)\) could be calculated as:

\[
Succ(M) = \sum_{k=2}^{n} k^{-1}Pr[x_1, x_2 \in H^{-1}(N(k))]Pr[H(x_1) \in N(k)]
\]

\[
= \sum_{k=2}^{85} k^{-1}\frac{kn(k)}{255 \cdot 256}
\]

To be brief, \(Succ(M)\) is abbreviated as \(S\) hereafter. The smaller \(S\) is, the less collisions occur. Next, let’s see how \(S\) works on \(H_1, H_2, \ldots, H_{12}\).

For \(H_1\), in which \(n(0) = 254\), \(n(1) = n(255) = 1\) and all other \(n(k) = 0(k \not\in \{0, 1, 255\})\), we have \(S = \frac{127}{128}\). For \(H_2\), in which \(n(0) = 254\), \(n(128) = 2\), all other \(n(k) = 0(k \not\in \{0, 128\})\), we have \(S = \frac{127}{255}\). We could see, \(S\) has easily differentiated \(H_1\) and \(H_2\), pointing out that \(H_2\) is better than \(H_1\).

For \(H_3\), in which \(n(0) = 252\), \(n(64) = 4\) and all other \(n(k) = 0(k \not\in \{0, 64\})\), we have \(S = \frac{63}{255}\). For \(H_4\), in which \(n(0) = 252\), \(n(1) = 3\), \(n(253) = 1\), all other \(n(k) = 0(k \not\in \{0, 1, 253\})\), we have \(S = \frac{5313}{5440}\). We could know, \(S\) has easily distinguished \(H_3\) and \(H_4\), showing that \(H_3\) outperforms \(H_4\).

For \(H_5\), in which \(n(0) = 192\), \(n(4) = 64\) and all other \(n(k) = 0(k \not\in \{0, 4\})\), we have \(S = \frac{1}{85}\). For \(H_6\), in which \(n(0) = 192\), \(n(1) = 63\), \(n(193) = 1\), all other \(n(k) = 0(k \not\in \{0, 1, 193\})\), we have \(S = \frac{193}{340}\). We could see, \(S\) has easily differentiated \(H_5\) and \(H_6\), indicating that \(H_5\) overwhelms \(H_6\).

For \(H_7\), in which \(n(0) = 255\), \(n(256) = 1\) and all other \(n(k) = 0(k \not\in \{0, 256\})\), we have \(S = 1\). For \(H_8\), in which \(n(0) = n(2) = 128\) and all other \(n(k) = 0(k \not\in \{0, 2\})\), we have \(S = \frac{1}{255}\). Clearly, \(S\) has distinguished \(H_7\) from \(H_8\), pointing out that \(H_8\) is much better than \(H_7\).

For \(H_9\), in which \(n(0) = 64\), \(n(1) = 128\), \(n(2) = 64\), all other \(n(k) = 0(k \not\in \{0, 1, 2\})\), we have \(S = \frac{1}{510}\). For \(H_{10}\), in which \(n(0) = 127\), \(n(1) = 128\), \(n(128) = 1\), all other \(n(k) = 0(k \not\in \{0, 1, 128\})\), we have \(S = \frac{127}{510}\). Obviously, \(S\) has differentiated \(H_9\) from \(H_{10}\), pinpointing that \(H_9\) outperforms \(H_{10}\).

For \(H_{11}\), in which \(n(0) = 168\), \(n(1) = 64\), \(n(8) = 24\), all other \(n(k) = 0(k \not\in \{0, 1, 8\})\), we have \(S = \frac{7}{340}\). For \(H_{12}\), in which \(n(0) = 191\), \(n(1) = 64\), \(n(192) = 1\), all other \(n(k) = 0(k \not\in \{0, 1, 192\})\), we have \(S = \frac{191}{340}\). Apparently, \(S\) has distinguished \(H_{11}\) from \(H_{12}\), indicating that \(H_{11}\) overwhelms \(H_{12}\).

IV. CONCLUSION

In this paper, a novel “black box” measure for evaluating collision resistance is proposed. Different from those measures in the literature, the proposed measure indicates that \(n(k)(2 \leq k \leq 256)\) should be paid attention to instead of \(n(0)\) or \(n(1)\). The larger \(k\) is, the more \(n(k)\) affects the
extent of collision. Using the proposed measure, some hash functions could be told apart, indicating its excellent capability of collision resistance evaluation.

At last, we have to note that collision resistance is just one of the requirements for cryptographic hash functions, and small $S$ doesn’t necessarily make a good cryptographic hash function. For example, let $H_{13}(x) = x$, then its $S = 0$, but $H_{13}$ is never a candidate for cryptographic hash function as it totally abandons the ability of compression.

ACKNOWLEDGMENT

This work is partially supported by the Natural Science Foundation of China under Grant No. 61462033. Thanks to my supervisors Changxuan Wan & Zuowen Tan.

REFERENCES