Applications Of Surjection On Narrow-pipe Hash Functions

Xigen Yao
Wuxi Hengqi Electromechanical Device Co.Ltd.China
email : dihuo377@163.com

December 25, 2010

Abstract. Recently several reports of Cryptology ePrint Archive give some analysis and conclusions,that for a narrow-pipe hash function the entropy and codomain will reduce greatly. And even generic collision attacks on narrow-pipe hash functions are faster than birthday paradox.However,the conclusions don't apply to surjective compression functions.In practice,it is hard to design an ideal compression function and prove whether it is or not a surjection.In this paper,we give an effective way that we can design a composition of compression function C, such that: $C = g^* \circ C^*$.Where g^* is a simple epimorphic function, C^* is a normal ideal compression just as those big domain and narrow-pipe compression functions, then,we can thwart the conclusions on narrow-pipe hash function. .

keywords:hash,surjection,codomain,entropy,narrow-pipe

1 Introduction:

The Merkle-Damgaard construction[1] is the most widely used to transform a secure compression function $C : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$ into a cryptographic hash function $h_c(\cdot)$. (where,n denotes the size of the chaining value,and m denotes the block size for the compression function .)

The iterating hash functions of M-D construction try to maintain the following three properties of the cryptographic secure compression functions :

1.**pre-image resistance:**

2.**second pre-image resistance:**

3.**collision resistance:**

A number of attacks on hash functions have shown weaknesses of M-D construction,and recently hash designs are two types which called“wide-pipe ”and “narrow-pipe” hash functions. A wide-pipe iterated hash can produce large-size internal chaining value,the size of internal chaining value is more larger than the final hash value . A narrow-pipe hash can use a big domain to produce internal chaining value ,the size of internal chaining value is equal to the hash value.

There are some reports give analysis and conclusions:that for a narrow-pipe hash function the entropy and codomain will reduce greatly:

Proposition 1.

Let $\mathcal{F}_C$ be the family of all functions $C : X \to Y$ and let for every $y \in Y$, $C^{-1}(y) \subseteq X$ be the set of preimages of y i.e. $C^{-1}(y) = \{x \in X | C(x) = y\}$. For a function $C \in \mathcal{F}_C$ chosen uniformly at random and for every $y \in Y$ the probability that the set $C^{-1}(y)$ is empty is approximately e^{-1} i.e.

$$P_r\{C^{-1}(y) = \varnothing\} \approx e^{-1}.$$

Proposition 2.

Let $\mathcal{F}_W$ be the family of all functions $W : X \to Y$ where $X = \{0,1\}^{n+m}$and $Y = \{0,1\}^n$. Let for every $y \in Y$, $W^{-1}(y) \subseteq X$ be the set of preimages of y i.e.$W^{-1}(y) = \{x \in X | W(x) = y\}$.For a function $W \in \mathcal{F}_W$,chosen uniformly at random and for everyy $\in Y$,the probability that the set $W^{-1}(y)$is empty is approximately e^{-2m} i.e.

$$P_r\{C^{-1}(y) = \varnothing\} \approx e^{-2m}.$$

Proposition 3.

Let $C_1 : X \to Y$, $C_2 : X \to Y$ are two particular functions, chosen uniformly at random (where $X = Y = \{0,1\}^n$). If we define a function $C : X \to Y$ as a composition:

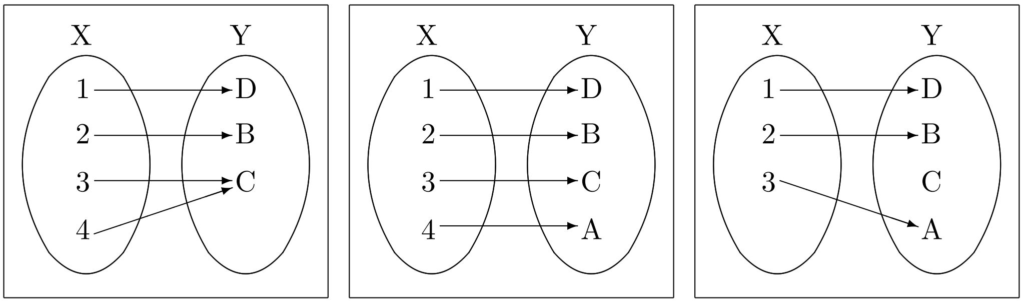
$$C = C_1 \circ C_2,$$

then for every $y \in Y$ the probability P_2that the set $C^{-1}(y)$ is empty is $P_2 = e^{-1+e^{-1}}$

However,the conclusions didn't consider any surjection.In this text,we will give a simple epimorphic function from MD5,and build a simple mode of a compression function C, such that: $C = g^* \circ C^*$,Where g^* is a simple epimorphic function, C^* is a normal ideal compression just as those big domain and narrow-pipe compression functions, then,we will try thwart the conclusions on narrow-pipe hash function.

2 Surjections

A function is said to be surjective or onto if its image is equal to its codomain. A function $f : X \to Y$is surjective if and only if for every y in the codomain Y there is at least one x in the domain X such that $f(x) = y$. A surjective function is called a surjection.If $f : X \to Y$ is surjective and B is a subset of Y, then $f(f^{-1}(B)) = B$. Thus, B can be recovered from its preimage $f^{-1}(B)$.



a. A surjective function. b.A bijective function. c.A non-surjective function.

3 MD5 Structure And It's Surjective Round

The Merkle-Damgard construction is the most common way to transform a compression function $C : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$ into a hash function $h_C(.)$,the Message Digest is n-bit value.

C denotes the compression function. M denotes the padded and Appended message , it is formatted as $16L$ $(m/16)$-bit words :$w_0, w_1, ..., w_i, ..., w_{16L-1}$ ie.,the message is made up of L m-bit blocks and each the block contains 16 $(m/16)$-bit words,for hash code H_C:

CV_i = Chaining variable ,$CV_0 = IV$(given Initial Value),M_i = the i-th block

$CV_i = C(CV_{i-1}, M_i)$

$H_C(M) = CV_L$

MD5 Algorithm [2]

The chaining variables are initialized as: $a_0 = 0x67452301; d_0 = 0x10325476; c_0 = 0x98badcfe; b_0 = 0xefcdab89$; and for the ith iteration,the chaining variables are updated by CV_{i-1},i.e., $a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}$.

The 1st Round

Step1:$\Sigma_1 = a_{i-1} + F(b_{i-1}, c_{i-1}, d_{i-1}) + m_0 + 0xd76aa478$, $a_1 = b_{i-1} + \Sigma_1 <<< 7$;
Step2:$\Sigma_2 = d_{i-1} + F(a_1, b_{i-1}, c_{i-1}) + m_1 + 0xe8c7b756$, $d_1 = a_1 + \Sigma_2 <<< 12$;
Step3:$\Sigma_3 = c_{i-1} + F(d_1, a_1, b_{i-1}) + m_2 + 0x242070db$, $c_1 = d_1 + \Sigma_3 <<< 17$;
Step4:$\Sigma_4 = b_{i-1} + F(c_1, d_1, a_1) + m_3 + 0xc1bdceee$, $b_1 = c_1 + \Sigma_4 <<< 22$;
.....
Step13:$\Sigma_{13} = a_3 + F(b_3, c_3, d_3) + m_{12} + 0x6b901122$, $a_4 = b_3 + \Sigma_{13} <<< 7$;
Step14:$\Sigma_{14} = d_3 + F(a_4, b_3, c_3) + m_{13} + 0xfd987193$, $d_4 = a_4 + \Sigma_{14} <<< 12$;
Step15:$\Sigma_{15} = c_3 + F(d_4, a_4, b_3) + m_{14} + 0xa679438e$, $c_4 = d_4 + \Sigma_{15} <<< 17$;
Step16:$\Sigma_{15} = b_3 + F(c_4, d_4, a_4) + m_{15} + 0x49b40821$, $b_4 = c_4 + \Sigma_{16} <<< 22$;

For the First Round of the ith iteration,no matter how the input values of chaining variables($a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}$) are prescribed arbitrarily, the output of the chaining variables a_4, b_4, c_4, d_4 can achieve any values prescribed arbitrarily by selecting the input values of $m_{12}, m_{13}, m_{14}$and m_{15}. If we mark the first round as a function g,then g is a surjection,$g : (0,1)^n \times (0,1)^m \to (0,1)^n$.

Namely for any given input of chaining variable CV_{i-1},the codomain is recovered completely, the mapping from the first round is a surjection, this doesn't depend on the previous chaining variable CV_{i-1}.
For $g : X \to Y$, and let for every $y \in Y$, $g^{-1}(y) \subseteq X$ be the set of preimages of y i.e. $g_1^{-1}(y) = \{x \in X | g(x) = y\}$. For a function $g \in \mathcal{F}_C$ chosen uniformly at random and for every $y \in Y$ the probability that the set $g^{-1}(y)$ is empty doesn't exist. i.e.

$$P_r\{g^{-1}(y) = \varnothing\} \approx e^{-1} \text{ doesn't exist.}$$

The rest 3 rounds of MD5 can be regarded as a function $C^* : (0,1)^n \times (0,1)^m \to (0,1)^n$,then the compression function C of MD5 is: $C = g \circ C^*$.

Since the codomain can be recovered completely in each iteration firstly, the case that the **continuous and cumulative reducing** of entropy does't exist.

There may be the probability that the set $C^{-1}(y)$ is empty is approximately e^{-1} in each iteration of the compression function C of MD5 ,i.e.

$$P_r\{C^{-1}(y) = \varnothing\} \approx e^{-1},$$

but this is discrete and independent,it's not cumulate,and the positions of empty set in each iteration are different .

It is necessary to make the difference between the input of Function C^* and of Function g,this can avoid the specificity of the mapping, we can see the sequences of the input$(m_0, m_1, ..., m_{15})$ in 4 rounds of MD5 are different.

4 Ideal Compression Function

Quote from Vlastimil Klima and Danilo Gligoroski [3]:
– hlen - the length of the chaining variable.
– mlen - the length of the message block.
– hashlen - the length of the hash function output.

If the compression function has the property, that for every value m the function $C(h, m) \equiv Ch(h)$ is an ideal random function of the variable h, we denote it as $IRF(h)$.

If the compression function has the property, that for every value h the function $C(h, m) \equiv Ch(m)$ is an ideal random function of the variable m, we denote it as $IRF(m)$.

The hash function is defined by a narrow-pipe compression function (NPCF), iff $hashlen = hlen = \frac{mlen}{2}$ and the compression function is $IRF(h)$ and $IRF(m)$.

The hash function is defined by a wide-pipe compression function (WPCF), iff $hashlen = \frac{hlen}{2} = \frac{mlen}{2}$and the compression function is $IRF(h)$ and $IRF(m)$.

Then,We can regard MD5 is a narrow-pipe $IRF(m)$ but not a $IRF(h)$.

If message input M chosen is fixed and invariant,there can't get surjection g so that the codomain can be recovered . There are continuous sets such $g^{-1}(y)$ is empty in each iteration, the entropy and the codomain will reduce greatly.We'd amend the function g again.

For a narrow-pipe $IRF(h)$, variable h is :$hashlen = hlen = \frac{mlen}{2}$.We amend the input M_i as:$M_i = g^*(h_{i-1}, M_i)$,then,the codomain is recovered by $g^*(h, M_i)$ firstly,the compression function C is amended , such that: $C = g^* \circ C^*$, i.e,We can mix CV_{i-1} into each message input M_i of a iteration while the case of fixed message input,and we can define the last block M_L^* to avoid the specific of it.

5 One Mode Of Narrow Pipe Hash

Amend a n-bit r-round compression f of MD construction into ideal compression F of LAB Form C,and to get a secure hash function construction .The compression $F[4]$:

$$CV_i = F(CV_{i-1}, \sum M_i, M_i) \ ;$$

where $\sum M_i$ is the sum (modulo addition) of the i blocks ,it can also be the xor operation of the i blocks.

Split chaining variable CV_{i-1} into 8 words:$h_{i-1,0}, h_{i-1,1}, ..., h_{i-1,7}$.

Define :For each unit $w_{i,j}$ of $M_i(w_{i,0}, w_{i,1}, ..., w_{i,15})$,

$w_{i,0} = w_{i,0} + h_{i-1,0}; w_{i,1} = w_{i,1} + h_{i-1,1}; ..., w_{i,7} = w_{i,7} + h_{i-1,7};$

$w_{i,8} = w_{i,8} + h_{i-1,0}; w_{i,9} = w_{i,9} + h_{i-1,1}, ..., w_{i,15} = w_{i,15} + h_{i-1,7}.$

And each unit $\Sigma w_{i,j}$ of Block ,define :

$\sum M_i \ (\Sigma w_{i,o}, \Sigma w_{i,1}, ..., \Sigma w_{i,15}) \ (1 \leq i \leq L \ ,0 \leq j \leq 15)$,:

$\sum M_i = M_i + \sum M_{i-1}$ i.e.:

$\Sigma w_{i,j} = w_{i,j} + \Sigma w_{i-1,j}$

We provide a detailed form of LAB Form C below.

1.Append padding bits and append length just as M-D Structure:

The message is padded with single 1-bit followed by the necessary number of 0-bits,so that its length l congruent to 896 modulo1024 [$l \equiv 896(mod1024)$],append a block of 128 bits as an unsigned 128-bit integer(most significant byte first)and contains the length of the original message.M denotes the message after padding bits and appending length. message. M is split to be L blocks:$M_1M_2...M_L$

2.Define a additive block M_0,encode the size of hash value n into M_0,just like HAIFA. Amend the last block as $M_L^* = M_{L-1}$.

3.Define an initial value IV,Set Array $A[16]$ and Array $B[16]$,for a r-round compression function F ,for i from 1 to L,do the following operations of each iteration,and get the hash value $h_F(M)$:

$CV_0 = IV$,$\sum M_0 = M_0$

$CV_i = F(h_{i-1}, \sum M_i, M_i)$

$CV_L = F(h_{L-1}, \sum M_L, M_L^*)$

The hash value is CV_L

4.Truncate the final chaining value if needed.

We provide details (e.g.):

Define :$\sum M_i = M_i + \sum M_{i-1}$ ($1 \le i \le L$,$0 \le j \le 15$)   i.e.: $\Sigma w_{i,j} = w_{i,j} + \Sigma w_{i-1,j}$

Define :$\sum M_0 = M_0$ and $A[16] = \sum M_0$.

For the i-th iteration,Update $A[16]$ and $B[16]$:

For $1 \le i \le L$,copy the i-th block M_i of 16 64-bit words into Buffer:

$B[16] \leftarrow w_{16i+j}$ ($0 \le j \le 15$), ie., $B[16] \leftarrow M_i$,and then,

$A[16] \leftarrow (M_i + \sum M_{i-1})$. ie., $A[16] \leftarrow \sum M_i$.

For r-round compression function F ,do the operations with the input of $A[16]$ and $B[16]$,and for each nonlinearity step function $s_{t,j}(m_j)$ ($1 \le t \le r, 0 \le j \le 15$) of the first round,Amend it as(the first round is function g^*):

1. Define :For each unit $w_{i,j}$ of $M_i(w_{i,0}, w_{i,1}, ..., w_{i,15})$,

   $w_{i,0} = w_{i,0} + h_{i-1,0}; w_{i,1} = w_{i-1,1} + h_{i-1,1}; ..., w_{i,7} = w_{i-1,7} + h_{i-1,7};$

   $w_{i,8} = w_{i,8} + h_{i-1,0}; w_{i,9} = w_{i,9} + h_{i-1,1}, ..., w_{i,15} = w_{i,15} + h_{i-1,7}.$

2. $S_{1,j} = s_{1,j}(\Sigma w_{i,j}) + m_{i,j}.$

3. For the rest round ,

   $S_{t,j} = s_{t,j}(m_{i,j}) + \Sigma w_{i,j}$ $(2 \le t \le r)$.

The function g^* of first round is a surjection,and the input of rest rounds are different from the input of the first round.

The computing of the last iterations is:

$CV_L = F(CV_{L-1}, \sum M_L, M_L^*)$,

where,$M_L^* = M_{L-1}$,$\sum M_L = M_L + \sum M_{L-1}$, this avoid the simpleness and specificity of the last block.

References

[1] Ralph C. Merkle, *One Way Hash Functions and DES*, Advances in Cryptology, proceedings of CRYPTO 1989, Lecture Notes in Computer Science 435, pp. 428 C 446, Springer-Verlag, 1990.

[2] Jie Liang and Xuejia Lai , *Improved Collision Attack on Hash Function MD5*Cryptology ePrint Archive2005/425

[3] Vlastimil Klima and Danilo Gligoroski,*Generic collision attacks on narrow-pipe hash functions faster than birthday paradox, applicable to MDx, SHA-1, SHA-2, and SHA-3 narrow-pipe candidates*Cryptology ePrint Archive2010/430

[4] Xigen Yao,*LAB Form for Iterated Hash Functions*,Cryptology ePrint Archive2010/269