Active Domain Expansion for Normal Narrow-pipe Hash Functions

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

May 7, 2011

Abstract.
Recently several reports of Cryptology ePrint Archive showed the discovering that for a normal iterative hash function the entropy and codomain would reduce greatly,then some conclusions were given: Narrow-pipe hash functions couldn't resist this reducing (But wide-pipe hash functions could.),and generic collision attacks on narrow-pipe hash functions would be faster than birthday paradox.The discovering and conclusions rely on the cases of active domain reducing which causes the empty set of a approximative probability e^{-1} in a iteration.However,we can thwart the conclusions by the way of Active Domain Expansion to keep or recover the entropy , by some amending for any a normal narrow-pipe hash function to realize it.And some hash mode such as LAB Mode[1]can more simply do it.In this paper,we'd introduce Active Domain Expansion which includes Surjection Round and the sum block ΣM_i.The most important is to define a sum block ΣM_i to replace the input of a normal message block M_i in compression function.ΣM_i is a sum of the foregoing i "Encoded Blocks".since the surjection round has the same purport and the form is a part of Active Domain Expansion,Surjections Round will be non-critical section in this paper.Besides,we can redefine the last block of additional bits.By these,a normal narrow-pipe hash function can resist the reducing completely. .
keywords:
narrow-pipe hash, Active Domain Expansion,Encoded Block ,entropy, recover

1 Introduction:

Most hash functions are iterative and the Merkle-Damgaard construction[3] is the most widely used to transform a secure compression function $C : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ into a cryptographic hash function $h_c(\cdot)$. (through this paper n denotes the size of the chaining value,and m denotes the block size for the

compression function.)

The construction expand the domain from $\{0,1\}^m$ of a secure compression function to the domain $\{0,1\}^{Lm}$ of the whole message string(For simplicity,we assume message string is L blocks.),and 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 normal compression function of MD construction only transmit entropy of n-bit from a m-bit block message,ie.,the construction reduces the entropy greatly in each iteration,from the domain $X = \{0,1\}^m$ to a codomain $Y = \{0,1\}^n$,and this is one of the prime vulnerabilities which cause a number of generic attacks on the hash functions , such as Multicollisions Attack,Second Preimage Attack and Herding Attack, and recently several reports of Cryptology ePrint Archive showed the discovering that for a normal iterative hash function the entropy and codomain would reduce greatly.It indicates the fact again.It's hard to avoid kinds of new attacks even the construction repairing are ingenious.It is that the wide-pipe hash can easily avoid those attacks because the compression function of a wide-pipe hash can transmit active entropy (chaining value) of $2n$-bit size just as big as of message block.

Now ,the hash designs are two types which called"wide-pipe "and "narrow-pipe" hash functions. A wide-pipe iterated hash produces a large-size internal chaining value,the size of internal chaining value is two times big of the final hash value,and for a narrow-pipe hash ,the size of internal chaining value equals to which of hash value.In this paper,we'll simply introduce a narrow-pipe mode which called "LAB Mode ". The compression function form of LAB mode is $C : CV_i = (CV_{i-1}, \Sigma M_{i-1}, M_i)$,it's a narrow-pipe hash,but actually it provides a big domain of input in each iteration,namely it transmit active entropy of a big size which is about $3n$-bit,and it is not the real big CV(chaining value) size which needs a intricate computing and large memory.

By the way,a paper is now well known*Domain Extension of Public Random Functions:Beyond the Birthday Barrier*[2] proposed by Ueli Maurer and Stefano Tessaro in 2007,who's theory on public random function can realize optimal security,it's quite strange such a significant theory has been valued until 2010 and 2011.It seems that even the latter like LAB Mode (which addressed independently on hash problems) is a small part contained by the theory,and some notions are similar,such as domain expansion,but we put emphasis on succinct narrow-pipe hash in this paper.

There is the discovering that for a normal iterative hash function the entropy and codomain will reduce greatly,and then there are some conclusions that narrow-pipe hash functions can't resist this reducing (But wide-pipe hash functions can.),and generic collision attacks on narrow-pipe hash functions are faster

than birthday paradox.The conclusions are mostly based on the case that for given message blocks the effective entropies are only dependent on the chaining value (CV)in a iteration,in which the domain is $X = (0,1)^n$,in other words,the iteration is a mapping of $X \to Y$,where $X = (0,1)^n = Y$,and in this case,the discovering based on the basic mathematical fact is :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.[4]

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

Namely there is a big reducing of the entropy and codomain in a iteration.

In this paper,we find the way of Active Domain Expansion by using the method of LAB Mode to pass and hold the entropy:firstly encode the values of CV_{i-1} (chaining values) into Block M_i,then a sum block ΣM_i is defined and added into the input ,Such that: $C : CV_i = (CV_{i-1}, \Sigma M_{i-1}, M_i)$ (where,ΣM_{i-1} is a sum of the foregoing $(i-1)$ "Encoded Blocks").

Except LAB mode,for a normal narrow-pipe hash function we can use sum block ΣM_i replacing input M_i to resist the reducing(ΣM_i can be the simple Input-Restricting Functions of [2]),and can redefine the last block of additional bits.But a narrow-pipe hash function with the simple form compression function $C : CV_i = (CV_{i-1}, \Sigma M_i)$ can't resist other generic attacks,such as Multicollisions Attack,it needs other techniques of other mode.

We can also build a composition of a compression function C, such that: $C = g^* \circ C^*$,Where g^* is a simple surjective function, C^* is a normal ideal compression just as those narrow-pipe compression functions.The codomain can be recovered completely by the surjection in each iteration firstly, the case that the **continuous and cumulative reducing** of entropy does't exist.Since normally the surjection form is a part of Active Domain Expansion,it also requires a big and active domain of input),and ΣM_i of Active Domain Expansion has provided the condition (big and active domain of input) and avoided the reducing almost, Surjections will be non-critical section in this paper.

The conclusions will be not worried for narrow-pipe hash functions since the reducing will be no more again .

2 The Probability of Empty Set

The discovering is based on the basic mathematical facts and the cases of fixed message-block bellow [4]:

Proposition 1. For finite narrow domain: Ideal random functions C map the domain of n-bit strings $X = \{0,1\}^n$ to itself i.e. to the domain $Y = \{0,1\}^n$,

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. For finite wide domain: Ideal random functions W map the domain of $(n+w)$-bit strings $X = \{0,1\}^{n+w}$ to the domain $Y = \{0,1\}^n$,

Let $\mathcal{F}_W$ be the family of all functions $W : X \to Y$ where $X = \{0,1\}^{n+w}$ 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^{-2^w} i.e.

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

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_2 that the set $C^{-1}(y)$ is empty is $P_2 = e^{-1+e^{-1}}$

If $C = C_1 \circ C_2 \circ, ..., \circ C_k$,then,$P_k = e^{-1+P_{k-1}}$

The key question is the " effective and active" size m of domain X for input in a iteration,and the size n of codomain Y is fixed.The Domain X is according to the entropy namely" the effective and active domain ",e.g:If the last block is a fixed addition namely the entropy is 0,then the entropy of the last iteration only relies on the chaining value(CV),and the domain X equals to the domain of CV.

The cases of fixed message-block quoted from Vlastimil Klima [6]:

Let us suppose that a message M is divided into two parts A and B, i.e. $M = A||B$, where the part A consist of just one message block of 512 bits, and the number of 512-bit blocks in the part B is $N = 2^{35}$ (in case of current 2TByte HDD). Let us denote by h_A the intermediate chaining value, obtained after hashing the part A of the message M and let us suppose that the content of the part B is never changing - so it consists of constant message blocks $const1, const2, ..., constN$ (note that if padding is a part of the definition, it is also a constant block). We compute the final hash with the following iterative procedure:

$$\begin{aligned}
h_1 &= C(h_A, const1)\\
h_2 &= C(h_1, const2)\\
h_3 &= C(h_2, const3)\\
&...\\
h_N &= C(h_{N-1}, constN)\\
H(M) &= h_N
\end{aligned}$$

For each of the N iterations, there's only the entropy of chaining values can be transfered,namely each the iteration is a mapping $X \to Y$,where the domain $X \leq \{0,1\}^n$ and $Y = \{0,1\}^n$.On the basis of **Proposition 3.**,there exists con-

tinuous and cumulative reducing of entropy,and the final exists a huge empty set.

3 MD5 Structure And It's Surjective Round

The Merkle-Damgard construction is the most common way to transform a compression function $C : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ 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 $16L$ 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 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 [5]:

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 a_0, b_0, c_0, d_0 are updated by CV_{i-1},i.e., $a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}$.Copy the i-th block M_i of 16 32-bit words into Buffer:

$m[16] \leftarrow w_{16i+j}$ $(0 \le j \le 15)$

The 1st Round :

Step1:$\Sigma_1 = a_0 + F(b_0, c_0, d_0) + m_0 + 0xd76aa478$, $a_1 = b_0 + \Sigma_1 <<< 7$;

Step2:$\Sigma_2 = d_0 + F(a_1, b_0, c_0) + m_1 + 0xe8c7b756$, $d_1 = a_1 + \Sigma_2 <<< 12$;

Step3:$\Sigma_3 = c_0 + F(d_1, a_1, b_0) + m_2 + 0x242070db$, $c_1 = d_1 + \Sigma_3 <<< 17$;

Step4:$\Sigma_4 = b_0 + F(c_1, d_1, a_1) + m_3 + 0xc1bdceee$, $b_1 = c_1 + \Sigma_4 <<< 22$;

.....

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(updated a_0, b_0, c_0, d_0) 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,$g : X \to Y$,i.e,$g : (0,1)^n \times (0,1)^m \to (0,1)^n$.where X=$(0,1)^m$,$Y = (0,1)^n$,$m = n + w, n = 128, w = 384$.

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 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.

$$P_r\{g^{-1}(y) = \varnothing\} \approx e^{-2^w} \text{ doesn't exist.}$$

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^{-2^w} in each iteration of the compression function C of MD5 ,i.e.

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

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

Quoted from Vlastimil Klima and Danilo Gligoroski [6]:
– 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 approximate $IRF(m)$ of surjection but not a $IRF(h)$ of surjection.

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 $C^{-1}(y)$ is empty in each iteration, the entropy and the codomain will reduce greatly.We'd

amend the function g again.

The conclusions on narrow-pipe hash functions are mostly based on the process of last additional block in iterative hash functions.One of the key questions is that processing the last block with additional bits in a normal iterative hash function,there's the entropy of CV_{L-1} only n bits,namely a n-bit domain X maps to a n-bit codomain Y,the probability of empty set is approximately e^{-1}.

There's already the way of adding CV_i into the next iteration such as :

$CV_{i+1} = CV_i + C(CV_i, M_i)$,the Davies-Meyer Mode ,which is also used in MD4, MD5, SHA-1, and SHA-2,but this is not sufficient enough.

5 Active Domain Expansion

For a narrow-pipe $IRF(m)$(where $hashlen = hlen = \frac{mlen}{2}$), firstly,we amend the input M_i as:$M_i = m^*(h_{i-1}, M_i)$*(This can take the mixing into each step function of computing and it is different from the Davies-Meyer Mode ,but this is some similar as Input-Restricting Functions of [2])*,and we can mark the new surjective function as g^*.

Then the codomain is recovered by g^* firstly,the compression function C is amended to be a approximate $IRF(m)$ and $IRF(h)$ of surjection F, such that: $F = g^* \circ F^*$, i.e,the entropy and value of CV_{i-1} can be put into each message input M_i,and the codomain can be recovered in each iteration.

The most important is that define a sum block ΣM_i and use it in each iteration.

Firstly, by the surjectiong^* ,the case that the continuous and cumulative reducing of entropy won't exist.

The second,LAB Mode increases a much big domain of input by providing additional block in each iteration,then it is in the case of **Proposition 2**,$P_r = e^{-2^w}$.(and we can define the last block M_L^*,but this step is not necessary in LAB Mode.)We can avoid the case of the other hashes of **Proposition 1**,in which the probability P_r of empty set is approximately e^{-1}.

The third,for any a normal narrow-pipe hash function,by the sum block ,there always exists the input variable $\sum M_i$ whenever the input message is or not a constant in each iteration,it is the case of **Proposition 2**,where the probability P_r of empty set is approximately e^{-2^w},for a 1024-bit block,$w \geq 1024 - 512 = 512$.

E.g,for the foregone message $M = A||B$,the iterations are:

$$h_1 = C(h_A, const1)... \longrightarrow h_1^{'} = C(h_A, const1, \sum M_1),$$

where $\sum M_1 = A + const1 + h_A$,since block A is a random message block,$\sum M_1$ is a random message block.

$h_2 = C(h_1, const2)... \longrightarrow h_2^{'} = C(h_1^{'}, const2, \sum M_2)$,$\sum M_2$ is also a random message block,and so is $\sum M_3$,...,

$$h_N = C(h_{N-1}, constN)... \longrightarrow h_i^{'} = C(h_{i-1}^{'}, constN, \sum M_i)$$

6 For HMACs

Firstly, let,'s quote to Danilo Gligoroski and Vlastimil Klima [7], and observe the problem by using the hash function SHA256 that has the compression function **CompressSHA256**().From the definition of HMAC we have that

$mac = HMAC(secret, M) = hash((secret \oplus opad) \| hash((secret \oplus ipad) \| M))$ where $\oplus$ is the operation of xor and $\|$ is the operation of string concatenation.where M is 256-bit message,and $secret$ is also 256 bits. The size of m is 512 bit,and the size n is 256.

Computing of mac will use four calls of the compression function **CompressSHA256**() in the following sequence:

1. h_1 =**CompressSHA256**$(iv_{256}, (secret \oplus ipad)) \equiv C_1(iv256)$
2. h_2 =**CompressSHA256**$(h_1, M \| CONST256) \equiv C_2(h_1)$, where

$$CONST256 = \underbrace{1000...000100000000}_{256bits}$$

3. h_3 =**CompressSHA256**$(iv_{256}, (secret \oplus opad)) \equiv C_3(iv_{256})$
4. $mac = h_4$ =**CompressSHA256**$(h_3, h_2 \| CONST256) \equiv C_4(h_3)$

Since h_1 is a fixed value,then it's the case of **Proposition 1**,in which the probability P_r of empty set is approximately e^{-1}.

The similar,h_3 is fixed,and $mac = h_4 =$ is the case of **Proposition 1**.

Let's use the technique of Active Domain Expansion(ADE)and overcome the problem of entropy reduction.

Firstly we amend the input M_i as:$M_i = m^*(h_{i-1}, M_i)$,ie,mixing of the input and CV.Secondly,there always exists the input variable $\sum M_i$ whenever the input message is or not a constant in each iteration,we can make a surjective function g^*,whenever the case of the last iteration is ,the codomain can be recovered completely. And the third,since we increase a additional variable $\sum M_i$,namely the domain is the case of **Proposition 2**,where the probability P_r of empty set is approximately e^{-2^w}:

Define a additive block M_0,Define a variable $\sum M_i$ where $\sum M_i = M_i + h_{i-1} + \sum M_{i-1}$,Define the input M_i as:$M_i = m^*(h_{i-1} \oplus M_i)$,for $i \geq 1$,we use LAB Mode:

$h_1^{'}$ =**CompressSHA256**$(iv_{256}, M_1, \sum M_0,) \equiv C_1^{'}(iv256)$,where,

$M_1 = (secret \oplus ipad) \oplus iv_{256}; \sum M_0 = M_0 + iv_{256}$

$h_2^{'}$ =**CompressSHA256**$(h_1^{'}, M_2, \sum M_1) \equiv C_2^{'}(h_1^{'})$,where,

$M_2 = M \| CONST256 \oplus h_1^{'}; \sum M_1 = M_1 + \sum M_0;$

$h_3^{'}$ =**CompressSHA256**$(h_2^{'}, M_3, \sum M_2) \equiv C_3^{'}(iv_{256})$,where,

$M_3 = (secret \oplus opad) \oplus h_2^{'}; \sum M_2 = M_2 + \sum M_1$

$mac = h_4^{'}$ =**CompressSHA256**$(h_3^{'}, M_4, \sum M_3) \equiv C_4(h_3^{'})$,where,

$M_4 = (h_2^{'} \| CONST256) \oplus h_3^{'}; \sum M_3 = M_3 + \sum M_2$.

Firstly ,The input $M_4, \sum M_3$ of $mac = h_4^{'}$ is a surjective function g^*;Secondly,the domain of the input is 2^{512} at least,then,it's the case of **Proposition 2**,where the probability P_r of empty set is approximately e^{-2^w},$w = m - n \geq 512 - 256 = 256$,ie.,the probability P_r of empty set is approximately $e^{-2^{256}}$.

Notice the different between h_3 and $h_3^{'}$,the 4 steps of ADE are continuous.

The concrete procedure of the two blocks M_i and $\sum M_i$,see LAB Mode followed.

7 One Mode of Narrow Pipe Hash

LAB Mode is succinct and efficient for narrow-pipe hash function against the main generic attacks,such as Multicollisions Attack,Second Preimage Attack and Herding Attack.It operates two blocks (one message block and a sum block)together only with a compression function.

By providing a additional variable of sum block (some similar as Input-Restricting Functions of [2])to expand active domain and substitute intricate functions design.

E.g.,for $hashlen = hlen = 512 = \frac{mlen}{2}$,$mlen = 1024$,amend a 512-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. :

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

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

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

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

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

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

LAB Form C :

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 $[l \equiv 896(mod 1024)]$,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:$M_1 M_2 ... M_L$,i.e.,M is made up of $w_0, w_1, ..., w_{16L-1}$.

2.Define a additive block M_0,encode the size of hash value n into M_0, just like HAIFA [8].

Amend the last block as $M_L^* = M_{L-1} + M_L$(But this amending is not the nec-

essary in LAB mode.).
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 $h_F(M)$:

$CV_0 = IV, \sum M_0 = M_0$

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

$CV_L = F(h_{L-1}, \sum M_{L-1}, M_L^*)$,The hash value is CV_L.

4.Truncate the final chaining value if needed.

The Details (e.g.):

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

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

Define :in the 1st iteration,$\sum M_0 = M_0$ and $A[16] \leftarrow \sum M_0$.

For the i-th iteration,

Split chaining variable CV_{i-1} into 8 words:$h_{i-1,0}, h_{i-1,1}, ..., h_{i-1,7}$,then update $B[16]$.

Copy the i-th block M_i of 16 64-bit words and add the 8 chaining variables into Buffer:

$B[16] \leftarrow (w_{16i+j} + h_{i-1,j mod 8})$ (where,$0 \leq j \leq 15$,and $M_i = m^*(CV_{i-1}, M_i)$.)

And at the ending of the i-th iteration,update $A[16]$:

$A[16] \leftarrow (B[16] + A[16])$. ie.,:

$A[16] \leftarrow \sum M_{i-1}$,

On this wise,the entropy and the values of CV are completely mixed into $\sum M_{i-1}$ and M_i ,this is different from the Davies-Meyer Mode.

Amend all the nonlinearity step functions $s_{t,j}(m_j)$ $(1 \leq t \leq r, 0 \leq j \leq 15)$.

1.For the step functions $s_{t,j}(m_j)$ of the first round,amend them ,such that:

$S_{t,j} = s_{t,j}(\Sigma m_{i,j}) + m_{i,j}$ $(t = 1$,The first round is **Surjection** g^*. $)$

2.For the rest rounds ,

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

The input mode of the rest rounds is different from which of the first round.

In this case,for each iteration of LAB Mode C:$CV_i = F(h_{i-1}, \sum M_{i-1}, M_i)$,ideal random compression function F maps the domain $X = \{0,1\}^{n+w}$ of $(n+w)$-bit strings (block $\sum M_{i-1}$ and block M_i) to the domain $Y = \{0,1\}^n$,where $n = 512, m = 2 \times 1024 = 2048, w = m - n = 2048 - 512 = 1536$,
so,the probability of empty set is(According to **Proposition 2.**) :

$P_r\{F^{-1}(y) = \varnothing\} \approx e^{-2^w} = e^{-2^{1536}}$

The probability of empty set can be ignored in each iteration even Function g^* is not a surjection namely the codomain isn't recovered completely.

The computing of the last iteration is:

$CV_L = F(CV_{L-1}, \sum M_{L-1}, M_L^*)$,
where,$M_L^* = M_{L-1} + M_L$, $\sum M_{L-1} = M_{L-1} + \sum M_{L-2}$, this avoid the simpleness and specificity of the last block.Even the last block is fixed addition,the entropy

of input $(CV_{L-1}, \sum M_{L-1}$and $M_L^*)$is much enough for the probability of empty set:$hlen = n = 512; n + w \geq 512 + 1024;$.

The probability of empty set in the last iteration is:

$$P_r\{F^{-1}(y) = \varnothing\} \approx e^{-2^w} \leq e^{-2^{1024}}$$

References

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

[2] Ueli Maurer and Stefano Tessaro *Domain Extension of Public Random Functions:Beyond the Birthday Barrier*Cryptology ePrint Archive2007/229

[3] 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.

[4] Danilo Gligoroski, *Narrow-pipe SHA-3 candidates differ significantly from ideal random functions defined over big domains*, NIST hash-forum mailing list, 7 May 2010

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

[6] 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

[7] Danilo Gligoroski and Vlastimil Klima*Practical consequences of the aberration of narrow-pipe hash designs from ideal random functions*Cryptology ePrint Archive2010/384

[8] Eli Biham, Orr Dunkelman, *A Framework for Iterative Hash Functions HAIFA*, NIST 2nd hash function workshop, Santa Barbara, August 2006.