A Proposal And Some Generic Attacks

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

September 20,2009

Abstract.This paper presents a efficient proposal for iterating hash function to avert the generic attacks which mainly includes *Multicollisions Application to Cascaded Constructions* ,*Second Preimage Attacks on Dithered Hash Functions* and *Herding Hash Functions and the Nostradamus Attack* .Based on this proposal,it's possible that a secure hash function can be built with iterating hash function while a strong compression function is necessary.

The proposal mainly contains a method called "shift whole message",it regroups the cascaded messages to be new blocks and makes the known results of the pre-computed blocks noneffective .
keywords: hash function ,iterating ,shift ,regrouped block,pre-computed

1 Introduction:

Since 2004,some of the existing hash functions MD5,SHA0,SHA1 [1][2]are broken. The existing hash functions are built by a same structure called Merkle-Damgaard Structure ,which use a iterating compression function.Many papers have shown the weakness of the structure and many generic attacks are given[3][4][5]. It seems that a iterating hash function can hardly be a secure hash function anymore.

This paper presents a efficient proposal for iterating hash function to avert those generic attacks which includes *Multicollisions Application to Cascaded Constructions* ,*Second Preimage Attacks on Dithered Hash Functions* and *Herding Hash Functions and the Nostradamus Attack*,of course, we need a stronger compression function to avert the special attacks on collision resistance .

The proposal is made up of two parallel compression functions, one acts in a normal mode,and another acts in a particular mode which contains a method called "shift whole message".

A normal pretreatment of a hash function is:after padding and appending length,a message is formatted as $16m$ 64-bit(or 32-bit) words,and the message is grouped to be m blocks ,each block contains 16 64-bit(or 32-bit)words. we add a pretreatment called "shift whole message":

an original message :$x_0, x_1, ..., x_i, ..., x_{16m-1}$

Cut s words from the head ,and link them to the end,ie.,the whole message is shifted to the left for s words.

Set a shift parameter $s = 8m + 1$,if m is a odd number;

and set $s = 8m + 7$ if m is a even number. For example,we consider m is a odd number.

The original message is:

$x_0, x_1, ..., x_{8m}, x_{8m+1} ..., x_{16m-1}$

The shifted message is:

$x_{8m+1} ..., x_{16m-1}, x_0, x_1, ..., x_{8m}$

This method is different from the previous ones[6], it makes the blocks re-grouped and makes the new blocks staggered ,then,each block is different from the original corresponding one . Once the message blocks cascade,each the block will change into a new block, the original known result of a block can't be used again,and we can avert the attacks which using known result by pre-computing blocks.

The rest of this paper,we first explain the method "shift whole message" to avert those attacks, then ,we give a more detailed proposal .

2 Shift Mode And The Generic Attacks

For a message x ,after padding and appending length ,it's length is $16m \times 64$ bits or $16m \times 32$ bits,(the succedent we assume it is 64bits) , which the hash function $H(x)$ of the M-D structure ,it's compression function is f , the message is formatted as $16m$ 64-bit words :$x_0, x_1, ..., x_i, ..., x_{16m-1}$, ie.,the message is made up of m blocks and each the block contains 16 64-bit words,for $H(x)$:

CV_i = Chaining variable ,$CV_0 = IV$(given Initial Value)

$X[i]$ = the ith block

$CV_i = f(CV_{i-1}, X[i])$

$H(x) = CV_m$

We can see,the fixed message-block $X[i]$ is computed one by one ,the attackers achieve by pre-computing the message-blocks,using known result of message-blocks, matching and yielding the chaining variables. We now use the method "shift whole message" and to see the generic attacks.

2.1 The Setting of Shift Whole Message

For a m blocks message ,we set the shift parameter $s = 8m + 1$, if m is a odd number; and set $s = 8m + 7$ if m is a even number.

For example, $m = 1$,i.e.,the message is a single block.The original message is:

$x_0, x_1, ..., x_8, x_9 ... x_{15}$

The shift parameter $s = 8m + 1 = 8 \times 1 + 1 = 9$ (m is a odd number),we cut 9($s = 9$)words $(x_0, x_1, ..., x_7, x_8)$ from the head and link them to the end.

where ,"| "denotes the cutting point:

the original message : $x_0, x_1, ..., x_8, |x_9 ... x_{15}$

And the shifted message:$|x_9 ... x_{15}, x_0, x_1, ..., x_7, x_8$

For a two-block message,we assume it contains Block A and B:

Block A :$x_0, x_1, ..., x_7, x_8, ..., x_{15}$

Block B :$y_0, y_1, ..., y_7, y_8, ..., y_{15}$

(Of curse ,if the single block A is shifted for hashing,it is:

$x_9, ..., x_{15}, x_0, x_1, ..., x_7, x_8$

the same ,if the single block B is shifted for hashing,it will be:

$y_9, ..., y_{15}, y_0, y_1, ..., y_7, y_8$)

Now, if the two single blocks are cascaded to be a two-block message,then the shift parameter $s = 8m + 7 = 8 \times 2 + 7 = 23$ ($m = 2$,it is a even number).

We cut 23($s = 23$)words from the head and link them to the end .

The original two-block message is:

$(x_0, x_1, ..., x_7, x_8, x_9 ... x_{15})(y_0, y_1, ..., y_6|y_7, ..., y_{15})$ (where | denotes the cutting point)

And the shifted two-block message is:

$(y_7, ..., y_{15}, x_0, x_1, ..., x_6)(x_7, ..., x_{15}, y_0, y_1, ..., y_6)$

We can see that the two new blocks of the cascaded message are staggered.The first new block contains elements of Block B ,and the second new block contains elements of Block A,so the two new blocks are different from either of Block A or Block B,we call the new blocks *regrouped blocks*.

Let $\widehat{BA}$ denotes the first regrouped block $(y_7, y_8, ..., y_{15}, x_0, x_1, ..., x_6)$.

($\widehat{BA}$ means the elements are from Block B and Block A.)

$\widehat{AB}$ denotes the second regrouped block $(x_7, ..., x_{15}, y_0, y_1, ..., y_6)$.

CV_{S1} denotes the output of the first block.Obviously,in shift mode:

$CV_{S1} = f(IV, \widehat{BA}) = f(IV, (y_7, ..., y_{15}, x_0, x_1, ..., x_6))$.

CV_{S2} denotes the output of the second new block:

$CV_{S2} = f(CV_{S1}, \widehat{AB})$=$f(CV_{S1}, (x_7 ... x_{15}, y_0, y_1, ..., y_6))$,

in normal mode, pre-compute CV_1 and CV_2:

$CV_1 = f(IV, A) = f(IV, (x_0, x_1, ..., x_{15}))$

$CV_2 = f(CV_1, B) = f(CV_1, (y_0, y_1, ..., y_{15})$

Obviously,$CV_{S1} \neq CV_1$ and $CV_{S2} \neq CV_2$.

So,the results of pre-computing the two single blocks $f(IV, A)$ and $f(CV_1, B)$ are of no use in this case.

Since the regrouped blocks $\widehat{BA}$ and $\widehat{AB}$ are different from the single shifted

Block A ($\overbrace{A}$ *denotes*) and shifted Block B($\overbrace{B}$ *denotes*), it's also no use that prehashing the single-shifted Block A or single-shifted Block B:

$H(A) = z_1$ ie.,z_1=$f(IV, \overbrace{A}) = f(IV, (x_9...x_{15}, x_0, x_1, ..., x_8))$

$H(B) = z_2$ ie.,z_2=$f(z_1, \overbrace{B}) = f(z_1, (y_9, ..., y_{15}, y_0, y_1, ..., y_8))$

Our proposal mainly contains the cascading of the two parallel compression functions f_1 and f_2 :

f_1 acts the original blocks , f_2 acts the shifted blocks,and the chaining values outputted are the modulo additions of theirs. f_1 ,f_2 can be similar compression functions which output chaining-value with same length (and even f_1 ,f_2 they can be a same function).

The followed is analysis of Multicollisions Application to Cascaded Constructions.

2.2 Multicollisions Application to Cascaded Constructions

For Multicollisions [3],We quote *Multicollisions in Iterated Hash Functions Application to Cascaded Constructions* :In a normal iterating function with M-D construction ,recall that a collision is a pair of different messages M and M' such that $H(M) = H(M')$. Due to the birthday paradox, there is a generic attack that find collisions after about $2^{n/2}$ evaluations of the hash function, where n is the size in bits of the hash values.If there are t collisions of t pairs of different messages,then ,there can construct multicollisions of 2^t collisions. first is how 4-collisions can be obtained,assume that two different blocks, A and A' that yield a collision, i.e. $f(IV, A) = f(IV, A')$. Let z denotes this common value and find two other blocks B and B' such that $f(z, B) = f(z, B')$. Put these two steps together to obtain the following 4-collision:

$f(f(IV, A), B) = f(f(IV, A), B') = f(f(IV, A'), B) = f(f(IV, A'), B')$

And 2^t-collision can obtain by analogy.

Now,in shift mode , we set the shift parameter $s = 8m + 1$, if m is a odd number; and set $s = 8m + 9$ if m is a even number.

Obviously ,for each the hashing computing of the single block ,$m = 1$,and the shift parameter $s = 8m + 1$.ie., $s = 8 \times 1 + 1 = 9$

For example,we show the message made up of two blocks,we also assume the previous two different single blocks A and A' that yield a collision,and B , B'yield another collision at z. $A(x_0, x_1, ..., x_6, x_7, ...x_{15})$ denotes the elements of block A,by the same,there are $B(y_0, y_1, ...y_6, y_7, ..., y_{15})$, $A'(x'_0, x'_1, ..., x'_6, x'_7, ...x'_{15})$, and $B'(y'_0, y'_1, ...y'_6, y'_7, ..., y'_{15})$.

This is a message cascaded by two blocks ,which $m = 2$(m is an even number),so the shift parameter $s = 8m + 7$.ie., $s = 8 \times 2 + 7 = 23$.

For the original message AB,(| denotes the cutting point):

$(x_0, x_1, ..., x_7, x_8, ...x_{15})(y_0, y_1, ..., y_6, |y_7..., y_{15})$

We cut the first $23(s = 8\times2+7 = 23)$ words of $x_0, x_1, ..., x_7, x_8, ...x_{15}, y_0, y_1, ...y_6$ and link them to the end
the shifted message is:

$(y_7, ..., y_{15}, x_0, x_1, ..., x_6)(x_7, ...x_{15}, y_0, y_1, ...y_6)$

We show the blocks in three situations ,the first is each the original block,the second is the each single block shifted for hashing ,and the third is each of the two cascading blocks which are regrouped by shift .For A,B:

	Block1	Block2
original :	$A(x_0, x_1, ..., x_6, x_7, ..., x_{15})$	$B(y_0, y_1, ..., y_6, y_7, ..., y_{15})$
single :	$\widehat{A}(x_9, ..., x_{15}, x_0, x_1, ..., x_8)$	$\widehat{B}(y_9, ..., y_{15}, y_0, y_1, ..., y_8)$
cascaded:	$\widehat{BA}(y_7, ..., y_{15}, x_0, x_1, ..., x_6)$	$\widehat{AB}(x_7, ..., x_{15}, y_0, y_1, ..., y_6)$

By analogy ,we link $A^{'}$,B

	Block1	Block2
original :	$A^{'}(x_0^{'}, x_1^{'}, ..., x_6^{'}, x_7^{'}, ..., x_{15}^{'})$	$B(y_0, y_1, ..., y_6, y_7, ..., y_{15})$
single :	$\widehat{A^{'}}(x_9^{'}, ..., x_{15}^{'}, x_0^{'}, x_1^{'}, ..., x_8^{'})$	$\widehat{B}(y_9, ..., y_{15}, y_0, y_1, ..., y_8)$
cascaded:	$\widehat{BA^{'}}(y_7, ..., y_{15}, x_0^{'}, x_1^{'}, ..., x_6^{'})$	$\widehat{A^{'}B}(x_7^{'}, ..., x_{15}^{'}, y_0, y_1, ..., y_6)$

We assume the two different single blocks, $\widehat{A}$ and $\widehat{A^{'}}$ that yield a collision,(Of course,we can also assume that two original blocks A and $A^{'}$ yield a collision.)

$f(IV, \widehat{A}) = f(IV, \widehat{A^{'}}) = z$

i.e., $f(IV, (x_9, ..., x_{15}, x_0, x_1, ..., x_8)) = f(IV, (x_9^{'}, ..., x_{15}^{'}, x_0^{'}, x_1^{'}, ..., x_8^{'})) = z$

For cascaded message A-B,the first regrouped block is:

$\widehat{BA}(y_7, ..., y_{15}, x_0, x_1, ..., x_6)$

For cascaded message $A^{'}$-B,the first regrouped block is:

$\widehat{BA^{'}}(y_7, ..., y_{15}, x_0^{'}, x_1^{'}, ..., x_6^{'})$

Obviously,$\widehat{BA} \neq \widehat{BA^{'}}$;Synchronously,$\widehat{BA} \neq \widehat{A}$;$\widehat{BA^{'}} \neq \widehat{A^{'}}$, we can't get $f(IV, \widehat{BA}) = f(IV, \widehat{BA^{'}}) = z$ by assuming $f(IV, \widehat{A}) = f(IV, \widehat{A^{'}}) = z$.

The first regrouped blocks are also different from the two known original blocks A or $A^{'}$.

Therefore,the first step that $f(IV, \widehat{BA}) = f(IV, \widehat{BA^{'}}) = z$ is false, and we can't use the result of known collision that $f(IV, \widehat{A}) = f(IV, \widehat{A^{'}}) = z$.

Since they don't meet the same output of chaining value z ,Synchronously in the shifted cascading, their second new blocks:

$\overbrace{AB}(x_7, ...x_{15}, y_0, y_1, ...y_6)$ and $\overbrace{A^{'}B}(x^{'}_7, ...x^{'}_{15}, y_0, y_1, ...y_6)$ are also different, we can't get the results $f(z, \overbrace{AB}) = f(z, \overbrace{A^{'}B})$

Therefore,$f(IV, AB) \neq f(IV, A^{'}B)$

By analogy,we also can't get the 4-collision :
$f(IV, AB) = f(IV, AB^{'}) = f(IV, A^{'}B) = f(IV, A^{'}B^{'})$

By the same way,that the Multicollisions Application to Cascaded Constructions, and 2^t-collision can obtain are false.

We will farther expound in the below.

2.3 Herding attacks

An attacker who can find many collisions on the hash function by brute force can first provide the hash of a message. The attacker first does a large precomputation, and then commits to a hash value h. Later, upon being challenged with a prefix P, the attacker constructs a suffix S such that hash $(P \parallel S) = h$. Kelsey and Kohno ,Their paper introduced the "diamond structure"[5], which is reminiscent of a complete binary tree. It is a 2^{ι} multi-collision in which each message in the multi-collision has a different initial chaining value, and which is constructed in the pre-computation step of the attack. The herding attack on an n-bit hash function requires approximately $2^{2n/3+1}$ work[4].

According to the previous result ,we can see that pre-computation of a block cann't be used in a cascaded message with shift mode. Now ,we consider it again by "Shift Whole Message".

For simplicity,assume the original blocks are $P(x_0, x_1, ..., x_{15})$ and $S(y_0, y_1, ..., y_{15})$

The single shifted message $\overbrace{P}$ is: $\overbrace{P}(x_9, ..., x_{15}, x_0, x_1, ..., x_8)$

The single shifted message $\overbrace{S}$ is: $\overbrace{S}(y_9, ..., y_{15}, y_0, y_1, ..., y_8)$

Assume that by pre-computing and matching ,we get $f(IV, \overbrace{P}) = z_1$ and $f(z_1, \overbrace{S}) = h$,then we'll see whether we can get that: $f(IV, P \parallel S) = h$.

$f(IV, \overbrace{P}) = f(IV, (x_9, ..., x_{15}, x_0, x_1, ..., x_8)) = z_1$;

$f(z_1, \overbrace{S}) = f(z_1, (y_9, ..., y_{15}, y_0, y_1, ..., y_8)) = h$

Now,P and S are cascaded to be a new message "$P \parallel S$",we write it as "$P - S$"or"PS".

The shift parameter s of the cascaded message PS :$s = 8m + 7 = 8 \times 2 + 7 = 23$,which $m = 2$.

the original message is:$(x_0, x_1, ..., x_6, x_7, ..., x_{15})(y_0, y_1, ..., y_6, |y_7, ..., y_{15})$

And the shifted message PS is:

$(y_7, ..., y_{15}, x_0, x_1, ..., x_6), (x_7, ..., x_{15}, y_0, y_1, ..., y_6)$

$\overbrace{SP}$ denotes the first regrouped block : $\overbrace{SP}(y_7, ..., y_{15}, x_0, x_1, ..., x_6)$

$\overbrace{PS}$ denotes the second regrouped block: $\overbrace{PS}(x_7, ..., x_{15}, y_0, y_1, ..., y_6)$

Obviously, $\overbrace{SP}$ is quite different from the single shifted block $\overbrace{P}$, so , the first steep, $f(IV, \overbrace{SP}) \neq f(IV, \overbrace{P}) = z_1$; synchronously , for the second blocks $\overbrace{PS} \neq \overbrace{S}$.

Even though $f(z_1, \overbrace{S}) = h$, We can't get $f(z_1, \overbrace{PS}) = h$, ie.,we can't get $f(IV, PS) = h$.

If the message of pre-computing is the original one that we get $f(IV, P) = z_1$, $f(z_1, S) = h$,By analogy,we'll have the same result.It means ,that to build a binary tree of multi-collision by pre-computing the original blocks or the single-shifted blocks is useless.

In fact, our proposal mainly contains the cascading of two parallel compression functions f_1 and f_2; f_1 acts the original blocks , f_2 acts the shifted blocks,the chaining values outputted are the modulo additions of theirs.

The shifted message is a shift version of the original message(by shifting it to left for $(8m+1)$or$(8m+7)$ words) , and we can also regard the original message as a shift version of the shifted message(by shifting it to right for $(8m-1)$or$(8m-7)$ words),ie.,they are the shift versions each other.

Consider a m-block message$B$:

If m is an odd number,then Block $B_{\frac{m+1}{2}}$ is the center one, where is just the cutting point.

$B(B_1, B_2, ..., B_{\frac{m+1}{2}}, B_{\frac{m+3}{2}}, ..., B_m)$

The shifted message is:

$\overbrace{B}(\overbrace{B_{\frac{m+1}{2}} B_{\frac{m+3}{2}}}, ..., \overbrace{B_{m-1} B_m}, \overbrace{B_m B_1}, ..., \overbrace{B_{\frac{m-1}{2}} B_{\frac{m+1}{2}}})$ ....Formula(1)

If m is an even number,where block$B_{\frac{m}{2}+1}$is the cutting point.

$B(B_1, B_2, ..., B_{\frac{m}{2}+1}, B_{\frac{m}{2}+2}, ..., B_m)$

$\overbrace{B}(\overbrace{B_{\frac{m}{2}+1} B_{\frac{m}{2}+2}}, ..., \overbrace{B_{m-1} B_m}, \overbrace{B_m B_1}, ..., \overbrace{B_{\frac{m}{2}} B_{\frac{m}{2}+1}})$ ....Formula(2)

The shifted message has moved about $8m$ words.ie.,each the block is shifted for about $8m$ words from it's corresponding one.

If P and S are multi-block massages,P contains m_1(Assume m_1 is an odd number) blocks,such that

$P(P_1, P_2, ..., P_{\frac{m_1+1}{2}}, ..., P_{m_1})$.

And S contains m_2(Assume m_2 is an odd number also,and $m_1 > m_2$) blocks,

$S(S_1, S_2, ..., S_{\frac{m_2+1}{2}}, ..., S_{m_2})$.

Then ,the single shifted message are(m_1,m_2 are odd numbers):
$\overbrace{P}$ $(\overbrace{P_{\frac{m_1+1}{2}}P_{\frac{m_1+3}{2}}},...,\overbrace{P_{m_1-1}P_m},\overbrace{P_{m_1}P_1},...,\overbrace{P_{\frac{m_1-1}{2}}P_{\frac{m_1+1}{2}}})$
$\overbrace{S}$ $(\overbrace{S_{\frac{m_2+1}{2}}S_{\frac{m_2+3}{2}}},...,\overbrace{S_{m_2-1}S_{m_2}},\overbrace{S_{m_2}S_1},...,\overbrace{S_{\frac{m_2-1}{2}}S_{\frac{m_2+1}{2}}})$
P,S are cascaded to be a m-block massagePS ($m = m_1 + m_2$,m is an even number)
$PS(P_1, P_2, ..., P_{\frac{m}{2}+1}, ..., P_{m_1}, S_1, S_2, ..., S_{m_2})$ (Now,$P_{\frac{m}{2}+1}$ is the cutting point) ,

And the shifted message $\overbrace{PS}$ is:
$(\overbrace{P_{\frac{m}{2}+1}P_{\frac{m}{2}+2}},...,\overbrace{P_{m_1-1}P_{m_1}},\overbrace{P_{m_1}S_1},...,\overbrace{S_{m_2-1}S_{m_2}},\overbrace{S_{m_2}P_1},\overbrace{P_1P_2},)(\overbrace{P_2P_3},...,\overbrace{P_{\frac{m}{2}}P_{\frac{m}{2}+1}})$

Obviously,the corresponding shifted massage
$(\overbrace{P_{\frac{m}{2}+1}P_{\frac{m}{2}+2}},...,\overbrace{P_{m_1-1}P_{m_1}},\overbrace{P_{m_1}S_1},...,\overbrace{S_{m_2-1}S_{m_2}},\overbrace{S_{m_2}P_1},\overbrace{P_1P_2})$ is different from $\overbrace{P}$ and $(\overbrace{P_2P_3},...,\overbrace{P_{\frac{m}{2}}P_{\frac{m}{2}+1}})$ is different from $\overbrace{S}$,therefore,we can't get $f(IV, PS) = h$.

There's another question:What about pre-computing the regrouped blocks(not the single shifted blocks and the original blocks)?
It seems that the pre-computed results can be useful if the regrouped blocks have been pre-computed . If a "diamond structure"is made up of the regrouped blocks,can the pre-computing results be used again?
If pre-compute the regrouped blocks, The original message will act as the shift version of the shifted message,and the original blocks act as the shifted ones,therefore ,just like the previous analysis ,that idea which pre-compute the regrouped blocks can't be realized.

Then,what about pre-computing the regrouped blocks and the original blocks together?
We'll explain by giving an example :
A two-block message AB,
Block A Block B
It'corresponding regrouped blocks are :
$\overbrace{BA}$ $\overbrace{AB}$
we pre-compute them by binding(e.g.,binding a pair of blocks, A and $\overbrace{BA}$).
Assume the two-block message AB(which it'$m_{AB} = 2$) is cascaded with an other message C_x(which it's m_c blocks) , there become a new message C_xAB with $m(m = m_{AB} + m_c)$blocks.then,there always $m > m_{AB} = 2$,the position of the pair of block A and $\overbrace{BA}$ has been moved,the pre-computed result of the binding blocks, a pair of A and $\overbrace{BA}$ is no use.
In fact, the blocks themself have been changed also ,there is no block $\overbrace{BA}$ anymore.
E.g.,let C_x is a sing block,ie.,$m_c = 1$.C_x is the first block,the original cascaded message is :C_xAB,ie.,Block C_x,Block B and Block$A$:

$C_x \quad A \quad B$
the corresponding regrouped blocks are :
$\overbrace{AB}, \overbrace{BC_x}, \overbrace{C_xA}$.

Now,the corresponding block of A is $\overbrace{BC_x}$, this pair of blocks Block A and $\overbrace{BC_x}$ are different from the previous binding blocks,A and $\overbrace{BA}$.

And then,there produced the new regrouped blocks :$\overbrace{BC_x}$ and $\overbrace{C_xA}$,which are different from the regrouped blocks of AB.

So we can't use the known result of Pre-computing a pair of binding blocks.

Synchronously ,if two single blocks are cascaded to be a two-block message, each block contains 16words(64bit-word),there are $64 \times 16 \times 2 = 2048$ bits,ie.,there are about 2^{2048} kinds two-block messages.

Obviously,there are about$(2^{1024})^3$ kinds of 3-block message,and$(2^{1024})^4$for 4-block message,.... therefore,that can be hardly realized to build a so called "diamond structure"for herding attack.

2.4 Second preimage attacks

For *Second Preimage Attacks on Dithered Hash Functions* of Andreeva et al[4], their basic technique relies on the diamond from the herding attack[6],if the diamond structure of herding is too hard to build,the attack is defeated.

The second preimage attack of Dean means[7]is ,to insert a block (or blocks)at so called a fixed point ith block,then make a preassigned output CV_i equal to the input CV_i.

We assume two messagesA_x and B_y. A_x contains m_1 blocks and B_y contains m_2(let's set $m_2 = 3$)blocks. $m_1 > m_2$, and both of them are odd numbers,

We will insert B_y into A_x,the point where we insert is the ith block of A_x's.Might as well set $1 \le i < \frac{m_1+m_2}{2} - 3$.

The original message A_x is:
$A_x(A_1, ..., A_i, ..., A_{\frac{m_1+1}{2}}, ..., A_{m_1})$
(*block* $A_{\frac{m_1+1}{2}}$ *is the cutting point.*)
The shift message of A_x is:
$\overbrace{A_x}(\overbrace{A_{\frac{m_1+1}{2}} A_{\frac{m_1+3}{2}}}, ..., \overbrace{A_{m_1-1}A_{m_1}}, \overbrace{A_{m_1}A_1}, ..., \overbrace{A_{\frac{m_1-1}{2}} A_{\frac{m_1+1}{2}}})$
The original message B_y is:
$B_y(B_1, B_2, B_3)$
The shift message of B_y is:
$\overbrace{B_y}(\overbrace{B_2B_3}, \overbrace{B_3B_1}, \overbrace{B_1B_2})$
Assume that: $f(CV_i, \overbrace{B_y}) = CV_i$ ie., $f(CV_i, (\overbrace{B_2B_3}, \overbrace{B_3B_1}, \overbrace{B_1B_2})) = CV_i$
Insert B_y into A_x at the fixed point A_i,$A_x - B_y$ denotes the cascaded message.

such that:

$A_x - B_y(A_1, ..., A_i, B_1, B_2, B_3, A_{i+1}, ..., A_{\frac{m_1+1}{2}}, A_{\frac{m_1+3}{2}}, A_{\frac{m_1+5}{2}}, ..., A_{m_1})$

The cascaded message $A_x - B_y$ contains m blocks,which
$m = m_1 + m_2 = m_1 + 3$,and m is an even number.

According to Formula(2): $\overbrace{B}$ $(\overbrace{B_{\frac{m}{2}+1}B_{\frac{m}{2}+2}}, ..., \overbrace{B_{m-1}B_m}, \overbrace{B_mB_1}, ..., \overbrace{B_{\frac{m}{2}}B_{\frac{m}{2}+1}})$

For message$A_x - B_y$, Block $A_{\frac{m}{2}+1}$ is the cutting point.
$(A_{\frac{m}{2}+1} = A_{\frac{m_1+m_2}{2}+1} = A_{\frac{m_1+3}{2}+1} = A_{\frac{m_1+5}{2}})$

we get the shifted massage $\overbrace{A_x - B_y}$:

$$(\overbrace{A_{\frac{m_1+5}{2}}A_{\frac{m_1+7}{2}}}, ...\overbrace{A_{m_1-1}A_{m_1}}, \overbrace{A_{m_1}A_1}, ..., \overbrace{A_{i-1}A_i}, \overbrace{A_iB_1}, \overbrace{B_1B_2}, \overbrace{B_3A_{i+1}}, ..., \overbrace{A_{\frac{m_1+3}{2}}A_{\frac{m_1+5}{2}}})$$

(*Of course ,we can write the first i regrouped blocks :*

$(\overbrace{A_{\frac{m_1+5}{2}}A_{\frac{m_1+7}{2}}}, ...\overbrace{A_{m_1-1}A_{m_1}}, \overbrace{A_{m_1}A_1}, ..., \overbrace{A_{\frac{2i-m_1+3}{2}}A_{\frac{2i-m_1+5}{2}}})$

There are t_1 blocks from $\overbrace{A_{\frac{m_1+5}{2}}A_{\frac{m_1+7}{2}}}$ to $\overbrace{A_{m_1-1}A_{m_1}}$
$(t_1 = m_1 - \frac{m_1+7}{2} + 1 = \frac{m_1-5}{2})$;
and there are t_2 blocks from $\overbrace{A_{m_1}A_1}$ to $\overbrace{A_{\frac{2i-m_1+3}{2}}A_{\frac{2i-m_1+5}{2}}}$,
$(t_2 = \frac{2i-m_1+5}{2} - 1 + 1 = i - \frac{m_1-5}{2})$
so,$t_1 + t_2 = i$

and the followed 3 regrouped blocks are:
$(\overbrace{A_{\frac{2i-m_1+5}{2}}A_{\frac{2i-m_1+7}{2}}}, \overbrace{A_{\frac{2i-m_1+7}{2}}A_{\frac{2i-m_1+9}{2}}}, \overbrace{A_{\frac{2i-m_1+9}{2}}A_{\frac{2i-m_1+11}{2}}})$

The first i regrouped blocks are different from $\overbrace{A_x}$. Compute the first i regrouped blocks such that
$f(IV, (\overbrace{A_{\frac{m_1+5}{2}}A_{\frac{m_1+7}{2}}}, ...\overbrace{A_{m_1-1}A_{m_1}}, \overbrace{A_{m_1}A_1}, ..., \overbrace{A_{\frac{2i-m_1+3}{2}}A_{\frac{2i-m_1+5}{2}}})$
we have no reason get the result is equal to CV_i.
And then,the followed 3 regrouped blocks are different from $\overbrace{B_y}$,so,there can't compute that $f(CV_i, \overbrace{B_y}) = CV_i$.

There always be $m > m_1$ and $m > m_2$,and simply it's shift parameter $s > s_1$ and $s > s_2$.So,once the message B_y is inserted into A_x,the pre-computed blocks of B_y has been shifted again,ie.,each the position of the block has been changed.thefore ,there isn't a point so called" fixed point",and there's no input CV_i for B_y.

On the other way,the pre-computed blocks of B_y have also been changed.

There'll always be the new regrouped blocks $\overbrace{A_iB_1}$ and $\overbrace{B_{m_2}A_{i+1}}$ for B_y in the cascaded message.

The new regrouped blocks $\overbrace{A_iB_1}$,$\overbrace{B_1B_2}$,and $\overbrace{B_3A_{i+1}}$ are different from their corresponding ones : $\overbrace{B_y}$ $(\overbrace{B_2B_3}, \overbrace{B_3B_1}, \overbrace{B_1B_2})$,synchronously, the chining value input CV_i, the positions of blocks and even each the blocks of A_x are all changed, we can't get a chaining value CV_i at the point *ith* block,and there's no $\overbrace{B_y}$ can be computed such that $f(CV_i, \overbrace{B_y}) = CV_i$, so,the attack is invalid.

Of course ,we can set m_2 be other values,and we'll get the same result : The attack called Second preimage attack is invalid.

2.5 Some Conditions

What about that the regrouped blocks are as same as the original blocks ?

We give an example message$A$,$m = 1$,ie.,that a single original block

A $(x_0, x_1, ..., x_6, x_7, ..., x_{15})$ is same as it's regrouped block

$\overbrace{A}$ $(x_9, x_{10}, ..., x_{15}, x_0, ..., x_8)$ .This means, their corresponding elements are equal:

$x_0 = x_9, x_1 = x_{10}, ..., x_6 = x_{15}, x_7 = x_0, ..., x_{15} = x_8$

From these 16 equations ,we can get:

$x_0 = x_1 = x_2 =, ..., = x_{15}$

It means that the message is an especial one,each of the 16 words is the same and fixed one.

By similar,we can get the similar results for a multi-blocks-message.

Even though there's this especial situation,this doesn't make different result from the previous analysis if they are cascaded.

The followed is a detailed proposal.

3 A Detailed Proposal

First,for pretreatment ,we get two messages.

For a message x,after padding and appending length,it's length is $16m \times 64$ bits, the message is formatted as $16m$ 64-bit words :$x_0, x_1, ..., x_i, ..., x_{16m-1}$, ie.,the message is made up of m blocks$(B_1, B_2, ..., B_m)$ and each the block contains 16 64-bit words.we set a shift parameter $s = 8m + 1$, if m is a odd number; and set $s = 8m + 7$ if m is a even number.

1 The original message is: $x_0, x_1, ..., x_{8m}, x_{8m+1} ..., x_{8m+6}, x_{8m+7}, ..., x_{16m-1}$

2 the shift message is:
$x_{8m+1}..., x_{16m-1}, x_0, x_1, ..., x_{8m}$(m is an odd number);
$x_{8m+7}..., x_{16m-1}, x_0, x_1, ..., x_{8m+6}$(if m is an even number)

The original message can be written as :$B_1, B_2, ...B_i, ..., B_m$
And the shift message can be written as:$B_{S1}, B_{S2}, ...B_{Si}, ..., B_{Sm}$.

Second,set two parallel compression functions f_1 and f_2 ,which are similar compression functions outputting chaining-value with same length:
f_1 outputs k(assume $k \geq 8$) 64-bit values of CV_{Oi} and f_2 outputs k 64-bit values of CV_{Si}. f_1 acts the original blocks , f_2 acts the regrouped blocks,for message x,the hash function$H(x)$, such that
for $1 \leq i \leq m$,
$CV_0 = IV$
$CV_i = CV_{Oi} + CV_{Si}$ where "+" is 2^{64} modulo addition.
$CV_{Oi} = f_1(CV_{i-1}, B_i)$; $CV_{Si} = f_2(CV_{i-1}, B_{Si})$
$H(x) = CV_m$
Where CV_{Oi} is the ith output of the original block B_i;
CV_{Si} is the ith output of the regrouped block B_{Si}.

This is a simple proposal,and we can get a stronger vision.
Based on the previous proposal,we add a step so called "Second Hash".

Make the last regrouped block B_{Si} a extra-block B_{xta}:
Add the chaining-value CV_{Om} andCV_{Sm} into the message of block B_{Si}.
B_{Si} is written as:$B_{Si}(u_0, u_1, ..., u_{15})$;
the output of f_1 is written as:$CV_{Om}(V_{O1}, V_{O2}, ..., V_{Ok})$;
the output of f_2 is written as:$CV_{Sm}(V_{S1}, V_{S2}, ..., V_{Sk})$,
then get 16 variables from the each first 8 variables of CV_{Om} and CV_{Sm}, such that:
$(V_{O1}, V_{O2}, ..., V_{O8}, V_{S1}, V_{S2}, ..., V_{S8})$ and compute:
$u_0 = u_0 + V_{O1}, u_1 = u_1 + V_{O2}, ..., u_7 = u_7 + V_{O8}$;
$u_8 = u_8 + V_{S1}, u_9 = u_9 + V_{S2}, ..., u_{15} = u_{15} + V_{S8}$

Hash the extra block B_{xta} by f_1 and f_2:
$CV_O = f_1(CV_m, B_{xta})$
$CV_S = f_2(IV, B_{xta})$
$H(x) = CV_O + CV_S$

We may select a new strong-avalanche hash function to work as the two parallel compression functions f_1 and f_2,and to resist the differential attacks on collision resistance.

4 summary

That the main weakness of a iterating hash function is it can be modularized.The attackers make use of this and develop various attacks.

Our purpose is how to break the modularization.Based on the two parallel modes which called "original"and "shift",we built the proposal,and the "Second Hash"is helpful. It seems that the iterating hash functions should be revalued,even though that the candidates of SH3 are advanced,the task of building a real secure hash function isn't finished.

5 References

[1]Xiaoyun Wang and Hongbo Yu *How to Break MD5 and Other Hash Functions* EUROCRYPT 2005, LNCS 3494, pp. 19C35, 2005

[2]Wang,X. ,Yin,Y. ,and Yu,H.*Finding Collisions in the Full SHA-1*, ,Proceedings–Crypto'05, 2005 ,Published–Spring-Verlag

[3] Antoine Joux *Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions*,In Proceedings of CRYPTO, LNCS 3152, pp. 306-316, Springer, 2004

[4] John Kelsey, Bruce Schneier *Second Preimages on n-Bit Hash Functions for Much Less than* 2^n *Work* In Proceedings of EUROCRYPT, LNCS 3494, pp. 474-490, Springer, 2005

[5]John Kelsey, Tadayoshi Kohno *Herding Hash Functions and the Nostradamus Attack* In Proceedings of EUROCRYPT, LNCS 4004, pp. 183-200, Springer, 2006

[6]Ronald L. Rivest *Abelian square-free dithering for iterated hash functions*2005

[7]Richard D. Dean *Formal Aspects of Mobile Code Security*1999