Active Domain Expansion for Narrow-pipe Hash

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

November 30, 2012

Abstract.
In this article, we give an approach to the problem of entropy and codomain reducing in a normal iterative hash function. The problem relies on the case of effective domain reducing which causes the empty set of a approximative probability e^{-1} in a iteration. We will keep, hold or recover the entropies by a way of Active Domain Expansion(ADE). ADE replaces the input message block by $\sum M_i$ in a normal iterative hash function. A sum block $\sum M_i$ is a sum of the foregoing i "Encoded Blocks". ADE makes the Sum Block $\sum M_i$ a big domain to satisfy a surjection function, it can recover the entropy and codomain.We put emphasis on succinct narrow-pipe hash in this paper and a normal narrow-pipe hash function can resist this reducing.
keywords:
domain expansion, entropy, narrow -pipe hash , surjection

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-block).

A normal compression function of MD Construction (called Narrow-pipe hash) only transmits entropy of n-bit from a m-bit block message in each iteration, it reduces entropy greatly , from the domain $X = \{0,1\}^m$ to the codmain $Y = \{0,1\}^n$, and this is one of the prime vulnerabilities which cause a number of generic attacks, such as Multicollisions Attack, Second Preimage Attack, and Herding Attack.

The entropy and codomain reduce greatly in a normal iterative hash function, there are some conclusions[2]:

A normal narrow-pipe hash function can't resist this reducing (but a wide-pipe hash functions can.), and generic collision attacks on narrow-pipe hash functions are faster than birthday paradox. For given message blocks (where the message blocks are constants and the entropies are 0) the effective entropies are only dependent on the chaining values (CVs), in other words, the iteration is a mapping of $X \to Y$, where$X = \{0,1\}^n = Y$, and in this case, for compression function C, the empty set is approximately e^{-1}, ie.,$P_r\{C^{-1}(y) = \varnothing\} \approx e^{-1}$[4].

Then , it leads to a continuous and cumulative reducing of entropy , and at the final there exists a huge empty set.

In this paper, we will try pass and hold the effective entropy by Active Domain Expansion (ADE).

We can also build a composition of a compression function C, such that: $C = g^* \circ C^*$, where g^* is a simple surjection function, C^* is a normal ideal compression function, then the codomain can be recovered completely by surjection in each iteration firstly, the case that the continuous and cumulative reducing of entropy doesn't exist.

Normally the surjection form also requires a big and active domain of input, and $\sum M_i$ of Active Domain Expansion has provide this condition (big and active domain of input).

We can do the same on MAC to avoid the reducing.

2 The Reducing And Probability of Empty Set

The reducings based on the basic mathematical facts and the cases of fixed message-block below[4]

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.

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

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 every $y \in Y$,the probability that the set $W^{-1}(y)$is empty is approximately e^{-2^w} i.e.*

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

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

$$P_2 = e^{-1+e^{-1}} \tag{3}$$

If $C = C_1 \circ C_2 \circ, ..., \circ C_k$,then,

$$P_k = e^{-1+P_{k-1}} \tag{4}$$

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(the size is n) can be transfered, namely each the iteration is a mapping $X \to Y$, where the domain $X \leq \{0,1\}^n$ and $Y = \{0,1\}^n$. According to formula (1), (3) and (4), there exists continuous and cumulative reducing of entropy, and the final exists a huge empty set.

3 Active Domain Expansion

For the i-th iteration of a narrow-pipe hash function , define a sum block $\sum M_i$:
For i from 1 to L(we assume message string is L -block),

$$\sum M_i = R(\sum M_{i-1} + CV_{i-1} + M_i) \tag{5}$$

Where, R is a linear (or nolinear) function.
And make the compression function C as :

$$C : CV_i = C(CV_{i-1}, \sum M_i) \tag{6}$$

or:

$$C : CV_i = C(CV_{i-1}, \sum M_{i-1}, M_i) \tag{7}$$

($\sum M_i$ in Formula(7) is a few different from Formula(6),see the definition in Section IV.)
For brevity, let :

$$\sum M_i = \sum M_{i-1} + CV_{i-1} + M_i \tag{8}$$

Define $\sum M_0 = M_0$.
We can take MD5 for example and to see what are the relevant conditions of empty set .
MD5[5]:

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

$$H_C(M) = CV_L$$

where M_i is made up of $m_0, m_1, ..., m_{15}$, for ith iteration , 4 working variables are :a_j, b_j, c_j, d_j,(for j from 0 to 15) , and in the ending of ith iteration,$a \parallel b \parallel c \parallel d$ become the chaining variable CV_i, the first round is:

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:$\Sigma_{15} = b_3 + F(c_4, d_4, a_4) + m_{15} + 0x49b40821$, $b_4 = c_4 + \Sigma_{16} <<< 22$;

Let M_i be a random message block, then:

No matter how the input values of chaining variables (update 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.

Namely for any given input of chaining variable CV_{i-1},the codomain is recovered completely.

The rest 3 rounds of MD5 can be regarded as a function $C^* : (0,1)^n \times (0,1)^m \rightarrow (0,1)^n$, according to Formula(2), the probability that the set is empty set is approximately $e^{-2^w} = e^{-2^{384}}$

where $m = n + w = 512$,$n = 128$,$w = 384$.

And Compression Function C of MD5 is: $C = g^* \circ C^*$.

If the last block M_L hashed is a random message block , there is no problem on entropy and codomain reducing for the final hash value $H_C(M) = CV_L$.

But if it is not a random message block, and it's in the case of Section 2 ,where the massage is $A \parallel B$, the final will exist a huge empty set.

The continuous and cumulative reducing of entropy won't exist if iterations with the surjection g^*.

For any a normal narrow-pipe hash function with the sum block $\sum M_i$, there always exists the input variable whenever the input message is or not a constant in a iteration, it is the case of Formula(2), where the probability P_r of empty set is approximately e^{-2^w}.

For a 512-bit block, $w \geq 512 - 128 = 384$.E.g.,for the foregone message $M = A||B$, the iterations are:

$$h_1 = C(h_A, const1) \longrightarrow h_1^{'} = C(h_A, \sum M_1)$$
$$h_2 = C(h_1, const2) \longrightarrow h_2^{'} = C(h_1^{'}, \sum M_2)$$
$$...$$
$$h_N = C(h_{N-1}, constN) \longrightarrow h_i^{'} = C(h_{i-1}^{'}, \sum M_N)$$

where $\sum M_1 = A + const1 + h_A$. Since block A is a random message block, $\sum M_1$ is a random message block. $\sum M_2$ is also a random message block, and so is $\sum M_3$,...,

So, the final block $\sum M_N$ is a variable of a random message block. it's in the case of foregoing random message block of MD5, which $C = g^* \circ C^*$.

$\sum M_i$ is the simplest form, it can pass and hold the effective entropies of the foregone iterations to a big domain of 2^m , it makes a big domain in each iteration for narrow pipe hash function, just like the wide pipe hash functions.

The mode is not a wide-pipe hash, in reality, it has the attribute of narrow-pipe hash. A normal wide-pipe hash must make CV of 2 size big. E.g., if we make SHA512 to be a wide-pipe hash, there must be 32 variables in stead of 16 variables, the added 16 variables must be uniformity and indistinguishability as the normal 16 variables strictly. So, it's a hard work. But in the new mode, it produce 16 variables just as a normal SHA512, it needn't the hard work of expanding 16 variables to 32 variables which of uniformity and indistinguishability.

4 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, 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 for h_2, the entropy is only dependent on $M\|CONST256$, it's the case of Formula(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 Formula(1).

Now, we use ADE Mode:

Firstly , change the input message block , define:

$\sum M_0 = M_0$, $\sum M_i = \sum M_{i-1} + M_i^*$, $M_i^* = M_i + CV_{i-1}$
and we use Formula(7):

$$C : CV_i = C(CV_{i-1}, \sum M_{i-1}, M_i^*,)$$

i.e.

$$h_i = C(h_{i-1}, \sum M_{i-1}, M_i^*,)$$

There always exists the input variable $\sum M_i$ whenever the input message is or not a constant in last iteration, we can make a surjective function g^*, whenever the case of the last iteration is, the codomain can be recovered completely.

Define a additive block M_0, for $i \geq 1$,

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

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

$$h_3' = \textbf{CompressSHA256}(h_2', M_3^*, \sum M_2) \equiv C_3'(iv_{256}),$$
$$mac = h_4' = \textbf{CompressSHA256}(h_3', M_4^*, \sum M_3) \equiv C_4(h_3'),$$

where

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

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

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

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

For $mac = h_4'$, the input (of M_4^*, $\sum M_3$) is big domains of ADE, the domain of the input is about 2^{512},then,it's the case of Formula(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 $e^{-2^{256}}$. And in concrete round of compression function, firstly the input of big domain can be a surjection round g^* to recover the domain of previous output.

Notice the different between h_3 and h_3',the 4 steps of ADE are continuous.

5 Discuss And Summarize

$\sum M_i$ is the simplest form, it can pass and hold the effectiv entropies of the foregone iterations to a big domain of 2^m, let the size $m = 2n$.

Formula(8)is:$\sum M_i = \sum M_{i-1} + CV_{i-1} + M_i$.

For a concrete of uniform distribution of bits,we can set:

let a_i and b_i denotes the two moieties of $\sum M_i$, and $a_i \parallel b_i$ denotes $\sum M_i$,then define:

if i is odd,

$$\sum M_i + CV_{i-1} = (a_i \parallel b_i) + h_{i-1} = (a_i \oplus h_{i-1}) \parallel b_i.$$

if i is even,

$$\sum M_i + CV_{i-1} = (a_i \parallel b_i) + h_{i-1} = a_i \parallel (b_i \oplus h_{i-1}).$$

Formula (7) $C : CV_i = C(CV_{i-1}, \sum M_{i-1}, M_i)$ is one mode of the succinct and efficient designs[1] for narrow-pipe hash functions 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. It provides a additional variable of sum block.

Some views are: It fails to be secure against multicollision attacks:

To see this, group calls to F by pairs calling the result G and consider messages of the restricted form (M_1 ,$-M_1$,M_2,$-M_2$,M_3,$-M_3$,...)
Then , all calls to G are of the form :

$$G(CV_i, 0, M_i) = F(F(CV_{i-1}, 0,), M_i, -M_i)$$

As a consequence, G only receives 2 arguments as in a classical Merkle-Damgard and multicollision attacks do apply.

However, $\sum M_i = \sum M_{i-1} + CV_i + M_i$, $\sum M_i$ is as a message block and contains the ***sum*** of the foregone i CVs (not CV_i itself), and one can't simultaneously select a message and control an exact value of CV_i to make the sum block $\sum M_i = 0$ in a iteration.

$\sum M_i$ of ADE is a big domain($\{0, 1\}^{2n}$), it can pass, hold and ***accumulate*** the entropies (see Appendix).

Another question is:
ADE is a mode designed against the problem of entropy and codomain reducing, it's substantive is entropy passing, holding and surjection, it roots in LAB mode [1] against those generic attacks.
Then, is ADE mode (Formula (6), only block $\sum M_i$) sufficient enough for a normal narrow-pipe hash function against those generic attacks ?
Formula (6):

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

and Formula (8):

$$\sum M_i = \sum M_{i-1} + CV_i + M_i$$

Since one can't simultaneously select a message and control an exact value in each a iteration, and $\sum M_i$ is combined tightly with each the foregone i message block and chaining values, the adversaries cannot select the messages of same hash values to build multicollision -pair messages (and so for Checksum Control Sequences[1][8]).

6 Appendix

Let's assume a message string:

The message is made up of 512 512-bit blocks, each block only contains 1 alterable bit and 511 immutable bits, e.g.:

For i from 0 to 511, the block M_i is $a_i \parallel cont$, a_i denotes the alterable bit and *cont* denotes the 511 immutable bits. Then, for a normal narrow pipe hash or a wide pipe hash:

$$CV_1 = C(CV_0, a_0 \parallel cont)$$

$$CV_2 = C(CV_1, a_1 \parallel cont)$$

$$...$$

$$CV_i = C(CV_{i-1}, a_{i-1} \parallel cont)$$

$$...$$

$$CV_{512} = C(CV_{511}, a_{511} \parallel cont)$$

There is little entropy in the first few iterations. Then, there is a huge empty set in a iteration hash function, at the final the case is Formula (1), the mapping is :

$$\{0,1\}^n \to \{0,1\}^n$$

The probability of empty set is approximately e^{-1}.

If we use $\sum M_i$ of ADE, there will be a big domain of $\{0,1\}^{2n}$, it can pass, hold and accumulates the entropies.

For a wide pipe hash, the mapping is $\{0,1\}^{2n} \to \{0,1\}^{2n}$ in a iteration,the entropies of domain $X = \{0,1\}^{2n}$ is about :

$$2^{2n}(1 - e^{-1}) \approx 2^{2n-1}$$

at the final, it's a mapping of:

$$\{0,1\}^{2n-1} \to \{0,1\}^n$$

So it's the case of Formula(2).

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.