Data-Depend Hash Algorithm

ZiJie Xu and Ke Xu
xuzijiewz@gmail.com
xukezp@gmail.com

Abstract: We study some technologys that people had developed to analyse and attack hash algorithm. We find a way that use data-depend function to resist differential attack. Then we design a hash algorithm that called Data-Depend Hash Algorit(DDHA). And DDHA is simple and strong under differential attack.

Key Word: Hash algorithm, data-depend function

1. Introduction

Hash algorithm is the algorithm that computes a fixed size message digest from arbitrary size messages. After SHA-0 was published, some technologys that analyse and attack hash algorithm are developed. The major technologys is differential attack. Papers[Wy05, Dau05] has explain the attack.

Differential attack is the best technique to attack hash function. To attack hash function, it need do the work as follow:

1. Constitute a feasible difference path that has good possibility.
2. Constitute the adequate conditions for the difference path.
3. Find some technique to raise the possibility of the difference path.

From mentioned above description of differential attack, it is easy to know that constituting a feasible difference path is the hinge. If it can make it hard to constitute a feasible difference path, it will be hard to attack the hash function. In appenix 1, we know that the data-depend circular shift has good defence feature. And we find a message expension function that make any difference path will has at least eight data-depend circular shift difference [appenix 2]. This make it hard to constitute a feasible difference path.

At the same time, we study some technologys[1,2] that used to attack hash algorithm, DDHA use some ways to resist these attack technologys.

The following operations are applied to 32-bit or 64-bit words in

DDHA:

1. ← variable assignment

2. Bitwise logical word operations: '∧' –AND , '∨' –OR, '⊕' –XOR and '¬' –Negation.

3. Addition '+' modulo 2^{32} or modulo 2^{64} .

4. The shift right operation, $SHR^n(x)$, where x is a 32-bit or 64-bit word and n is an integer with 0⩽n<32 (resp. 0⩽n<64).

5.The shift left operation, $SHL^n(x)$, where x is a 32-bit or 64-bit word and n is an integer with 0⩽n<32 (resp. 0⩽n<64).

6. The rotate right (circular right shift) operation, $ROTR^n(x)$, where x is a 32-bit or 64-bit word and n is an integer with 0 ⩽ n < 32 (resp. 0 ⩽ n < 64).

7. The rotate left (circular left shift) operation, $ROTL^n(x)$, where x is a 32-bit or 64-bit word and n is an integer with 0 ⩽ n < 32 (resp. 0 ⩽ n < 64).

2. Data-Depend Hash Algorithm (DDHA)

DDHA has two hash functions: DDHA-256(32-bitversion), DDHA-512 (64–bitversion). DDHA-256 is used for message no bigger than 2^{64}, DDHA-512 is used for message no bigger than 2^{64}, The properties as follow:

	word	Message size	Block size	Hash value size
DDHA-256	32	$< 2^{64}$	512	256
DDHA-512	64	$< 2^{64}$	1024	512

Properties of DDHA hash functions(size in bits)

In DDHA, the message will be preprocessed. After message is preprocessed, the message will prased in N message blocks, these blocks will be processed with a compression function in order.

2.1 Preprocessing

Preprocessing in DDHA include steps:

a. padding the message M, parsing the padded message into message blocks,

b. setting the initial hash value,

2.1.1 Padding and parsing

Suppose that the length of the message M is L bits. Append the bit "1" to the end of the message, followed by k zero bits, where k is the smallest, non-negative solution to the equation L+1+k ≡ 448 mod 512 (resp. L+1+k ≡ 960 mod 1024). Then append the 64-bit block that is equal to the number L expressed using a binary representation.

After message is padded, the message will be parsed into N 512-bits(resp. 1024-bits) message blocks.

2.1.2 Initial Hash Value and constants

DDHA use the same initial hash value as that of SHA-2 (given as follow):

DDHA-256	DDHA-512
$H_0^0 = 0x6a09e667,$	$H_0^0 = 0x6a09e667f3bcc908,$
$H_1^0 = 0xbb67ae85,$	$H_1^0 = 0xbb67ae8584caa73b,$
$H_2^0 = 0x3c6ef372,$	$H_2^0 = 0x3c6ef372fe94f82b,$
$H_3^0 = 0xa54ff53a,$	$H_3^0 = 0xa54ff53a5f1d36f1,$
$H_4^0 = 0x510e527f,$	$H_4^0 = 0x510e527fade682d1f,$
$H_5^0 = 0x9b05688c,$	$H_5^0 = 0x9b05688c2b3e6c1f,$
$H_6^0 = 0x1f83d9ab,$	$H_6^0 = 0x1f83d9abfb41bd6b,$
$H_7^0 = 0x5be0cd19,$	$H_7^0 = 0x5be0cd19137e2179,$

The initial hash value for DDHA

DDHA use 32 constant words, these words are separated into two parts C1 and C2 as follow:

DDHA-256	DDHA-512
$C1_0 = 0xd76aa478,$	$C1_0 = 0xd76aa478fffa3942,$
$C1_1 = 0xe8c7b756,$	$C1_1 = 0xe8c7b7568771f681,$
$C1_2 = 0x242070db,$	$C1_2 = 0x242070db699d6122,$
$C1_3 = 0xc1bdceee,$	$C1_3 = 0xc1bdceeefde5380c,$
$C1_4 = 0xf57c0faf,$	$C1_4 = 0xf57c0fafa4beea44,$
$C1_5 = 0x4787c62a,$	$C1_5 = 0x4787c62a4bdecfa9,$
$C1_6 = 0xa8304613,$	$C1_6 = 0xa8304613f6bb4b60,$
$C1_7 = 0xfd469501,$	$C1_7 = 0xfd469501bebfbc70,$
$C1_8 = 0x698098d8,$	$C1_8 = 0x698098d8289b7ec6,$

$C1_{9} = 0x8b44f7af$,	$C1_{9} = 0x8b44f7afeaa127fa$,
$C1_{10} = 0xffff5bb1$,	$C1_{10} = 0xffff5bb1d4ef3085$,
$C1_{11} = 0x895cd7be$,	$C1_{11} = 0x895cd7be04881d05$,
$C1_{12} = 0x6b901122$,	$C1_{12} = 0x6b901122d9d4d039$,
$C1_{13} = 0xfd987193$,	$C1_{13} = 0xfd987193e6db99e5$,
$C1_{14} = 0xa679438e$,	$C1_{14} = 0xa679438e1fa27cf8$,
$C1_{15} = 0x49b40821$,	$C1_{15} = 0x49b40821c4ac5665$,

Constants C1 of DDHA

C2 as follow:

DDHA-256	DDHA-512
$C2_{0} = 0xf61e2562$,	$C2_{0} = 0xf61e2562f4292244$,
$C2_{1} = 0xc040b340$,	$C2_{1} = 0xc040b340432aff97$,
$C2_{2} = 0x265e5a51$,	$C2_{2} = 0x265e5a51ab9423a7$,
$C2_{3} = 0xe9b6c7aa$,	$C2_{3} = 0xe9b6c7aafc93a039$,
$C2_{4} = 0xd62f105d$,	$C2_{4} = 0xd62f105d655b59c3$,
$C2_{5} = 0x02441453$,	$C2_{5} = 0x024414538f0ccc92$,
$C2_{6} = 0xd8a1e681$,	$C2_{6} = 0xd8a1e681ffeff47d$,
$C2_{7} = 0xe7d3fbc8$,	$C2_{7} = 0xe7d3fbc885845dd1$,
$C2_{8} = 0x21e1cde6$,	$C2_{8} = 0x21e1cde66fa87e4f$,
$C2_{9} = 0xc33707d6$,	$C2_{9} = 0xc33707d6fe2ce6e0$,
$C2_{10} = 0xf4d50d87$,	$C2_{10} = 0xf4d50d87a3014314$,
$C2_{11} = 0x455a14ed$,	$C2_{11} = 0x455a14ed4e0811a1$,
$C2_{12} = 0xa9e3e905$,	$C2_{12} = 0xa9e3e905f7537e82$,
$C2_{13} = 0xfcefa3f8$,	$C2_{13} = 0xfcefa3f8bd3af235$,
$C2_{14} = 0x676f02d9$,	$C2_{14} = 0x676f02d92ad7d2bb$,
$C2_{15} = 0x8d2a4c8a$,	$C2_{15} = 0x8d2a4c8aeb86d391$,

Constants C2 of DDHA

2.2 processing.

If there are N message blocks $M_0,...,M_{N-1}$.

The DDHA has a compression function. The input of compression function include chaining variable(8 words, $H_0^i,...,H_7^i$), message block(16 words, $m_0^i,...,m_{15}^i$), constants(32 words, $C1_0,...,C1_{15}$, $C2_0,...,C2_{15}$), and other parameters. Then the processing as foollow:

$$
\begin{array}{l}
for \quad j=0 \quad to \quad 17 \\
\quad tm_j \leftarrow 0 \\
next \quad j \\
oc_0 = 0 \\
oc_1 = 0 \\
for \quad i=0 \quad to \quad N-2 \\
\quad h^{i+1} \leftarrow compression(repeattime, m^i, h^i, tm, oc, c1, c2) \\
\quad temp = 0 \\
\quad occ = 1 \\
\quad for \quad j=0 \quad to \quad 15 \\
\quad\quad tm_j = tm_j + m_j^i + temp \\
\quad\quad if \quad (tm_j < m_j^i) \quad then \\
\quad\quad\quad temp = 1 \\
\quad\quad else \quad if \quad (tm_j > m_j^i) \quad then \\
\quad\quad\quad temp = 0 \\
\quad\quad end \quad if \\
\quad\quad if \quad m_j^i \neq 0 \quad then \quad occ = 0 \\
\quad next \quad j \\
\quad oc_0 = oc_0 + occ \\
\quad if \quad oc_1 = 0 \quad then \quad oc_1 = oc_1 + occ \\
\quad tm_{16} = tm_{16} + temp \\
\quad if \quad tm_{16} = 0 \quad then \quad tm_{17} = tm_{17} + temp \\
next \quad i \\
h^N \leftarrow compression(repeattime, m^{N-1}, \neg h^{N-1}, oc, tm, \neg c1, \neg c2) \\
return \quad h^N
\end{array}
$$

Processing of DDHA

2.3 compresion fuction

The function compression(repeattime, m^i,h, tm,oc,ct1,ct2) takes as input seven values:

* an integer value *repeattime*. User can set *repeattime* to get higher intensity. The default value of *repeattime* is 1.

* a message block $m^i = m_0^i, ..., m_{31}^i$

* a chain value $h = h_0, ..., h_7$

* an value $tm = tm_0, ..., tm_{17}$.

* an value $oc = oc_0, oc_1$.
* a Constant $ct1 = ct1_0,..., ct1_{15}$
* a Constant $ct2 = ct2_0,..., ct2_{15}$

The compression function use two functions: SR(m, h,ct1,ct2), ME(m). In DDHA, the word is carved up to sixteen parts, every part is used as parameter of data-depend circular shift once. And in function ME(m), the circular shift operation is based on part not bit.

2.3.1 SR(m,h,ct1,ct2)

The function SR(m,h,ct1,ct2) takes as input four values:
* a chain value $h = h_0,..., h_7$
* a message block $m = m_0,..., m_{32}$
* a Constant $ct1 = ct1_0,..., ct1_{15}$
* a Constant $ct2 = ct2_0,..., ct2_{15}$

And SR(m,h,ct1,ct2) as follow:

$$
\begin{array}{l}
for \quad i = 0 \quad to \quad 15 \\
\quad im \leftarrow i >> 1 + (i \bmod 2) \times 8 \\
\quad h_0 \leftarrow (h_0 + m_{im}) \\
\quad for \quad j = 1 \quad to \quad 7 \\
\quad\quad im1 \leftarrow (i >> 1 + j) \bmod 8 + (i \bmod 2) \times 8 \\
\quad\quad h_j \leftarrow ROTR^{(m_{im} >> ((j-1) \times rl)) \wedge rv} (h_j + m_{im1}) \\
\quad next \quad j \\
\quad h_4 \leftarrow ROTR^{(m_{im} >> (7 \times rl)) \wedge rv} (h_4 + h_0) \\
\quad t \leftarrow h_1 + h_2 + h_3 + h_4 \\
\quad for \quad j = 1 \quad to \quad 4 \\
\quad\quad h_j \leftarrow ROTR^{(m_{im} >> ((j+7) \times rl)) \wedge rv} (t - h_j) + ct1_i \\
\quad next \quad j \\
\quad t \leftarrow h_4 + h_5 + h_6 + h_7 \\
\quad for \quad j = 4 \quad to \quad 7 \\
\quad\quad h_j \leftarrow ROTR^{(m_{im} >> ((j+8) \times rl)) \wedge rv} (t - h_j) + ct2_i \\
\quad next \quad j \\
\quad t \leftarrow h_7 \\
\quad for \quad j = 7 \quad to \quad 1 \\
\quad\quad h_j \leftarrow h_{j-1} \\
\quad next \quad j
\end{array}
$$

$$h_0 \leftarrow t$$
$$next \quad i$$

SR function of DDHA

In DDHA-256, the word length is 32, rl is 2, rv is 3. In DDHA- 512, the word length is 64, rl is 4, rv is 15.

2.3.2 message expension function ME(m)

The message expension function ME(m) takes as input one value:

* a message block $m = m_0, ..., m_{15}$

And ME(m) as follow:

$$t = \bigoplus_{i=0}^{15}(m_i)$$
$$for \quad i = 0 \quad to \quad 15$$
$$m_i \leftarrow (t \oplus m_i)$$
$$next \quad i$$
$$for \quad i = 0 \quad to \quad 15$$
$$m_i \leftarrow ROTR^{i \times rl}(m_i)$$
$$next \quad i$$
$$t = \bigoplus_{i=0}^{15}(m_i)$$
$$for \quad i = 0 \quad to \quad 15$$
$$m_i \leftarrow (m_i \oplus t)$$
$$next \quad i$$
$$return \quad m$$

function ME of DDHA

In DDHA-256, the word length is 32, rl is 2. In DDHA- 512, the word length is 64, rl is 4.

With function SR(m,h,ct1,ct2), ME(m), the compression function as follows:

$$h1 \leftarrow h^i$$
$$c3 \leftarrow ct1$$
$$c4 \leftarrow ct2$$
$$ma \leftarrow m$$
$$mb \leftarrow ME(m)$$
$$for \quad j = 0 \quad to \quad 15$$
$$c3_j = c3_j \oplus tm_j$$

$$
\begin{aligned}
& next \quad j \\
& c4_0 = c4_0 \oplus tm_{16} \\
& c4_1 = c4_1 \oplus tm_{17} \\
& c4_2 = c4_2 \oplus oc_0 \\
& c4_3 = c4_3 \oplus oc_1 \\
& for \quad j = 1 \quad to \quad repeattime \\
& \quad SR(ma, h1, c3, c4) \\
& \quad SR(mb, h1, c4, c3) \\
& \quad if \quad repeattime > 1 \quad then \\
& \quad\quad for \quad j1 = 0 \quad to \quad 15 \\
& \quad\quad\quad c3_{j1} \leftarrow ROTR^1(c3_{j1}) \\
& \quad\quad\quad c4_{j1} \leftarrow ROTR^1(c4_{j1}) \\
& \quad\quad next \quad j1 \\
& \quad end \quad if \\
& next \quad j \\
& h1 \leftarrow h1 + h^i \\
& return \quad h1
\end{aligned}
$$

compression function of DDHA

In DDHA-256, the word bit-length is 32. In DDHA- 512, the word bit-length is 64.

3 Security of DDHA

In this section, we discuss the resistance of DDHA to Differential attack, Length extension, Multicollisions.

3.1 Differential attack

From appendix 2, we will know that if there is any difference in the message, the difference path for DDHA will has at least eight data-depend circular shift difference that $\Delta r \neq 0$, r is the parameter of data-depend circular shift.

By proposition A.1, it is known that if a data-depend circular shift difference that $(r_1 - r_2) \neq 0$, the possibility of a data-depend circular shift difference is $2^{\mathrm{gcd}-n}$, gcd is the **greatest common divisor of** $(r1 - r2)$ **and** n**.**

In DDHA, the parameter of data-depend circular shift less than 4(resp.16). then there has:

$$
\begin{array}{ll}
0 \le r_1, r_2 < 4 & 0 \le r_1, r_2 < 16 \\
-4 < (r1 - r2) < 4 & -16 < (r1 - r2) < 16 \\
\text{gcd} = GCD(r1 - r2, 32) \le 2 & \text{gcd} = GCD(r1 - r2, 64) \le 8 \\
\quad DDHA - 256 & \quad DDHA - 512
\end{array}
$$

So the possibility of a difference path for DDHA will be:

$$p \le 2^{(\text{gcd}-n)\times 8} \le 2^{16-8\times n} (resp. 2^{64-8\times n})$$

At the same time, in a difference path for DDHA if a chain value that $\Delta r = 0$ has defences, some bits in the parameter r will be fixed, this depend on the defences that the chain value has. Here we suppose attacker can find the needed defences.

3.2 Length extension

Length extension is the attaca against keyed hash of form $h = H_k(m)$ or $h = H(k \| m)$. The attack as: given $h = H_k(m)$, the padding data is p, then find m' that make $h = H(k \| m \| p \| m')$. The $(m \| p \| m')$ is the fabricated message.

Let *tm* is sum of the message blocks defore last block.

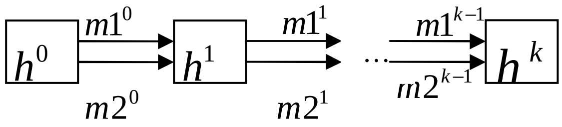
In DDHA, if $(... \| m^{N-2} \| m^{N-1})$ is padded message data, and m^{N-1} is the last message block, the final chain values is $h^N = compression(repeattime, m^{N-1}, \neg h^{N-1}, tm, oc, \neg c1, \neg c2)$. If a block m^N is extended, then the chain value between m^{N-1}, m^N will be $h^N = compression(repeattime, m^{N-1}, h^{N-1}, tm, oc, c1, c2)$ from $h^N = compression(repeattime, m^{N-1}, \neg h^{N-1}, tm, oc, \neg c1, \neg c2)$. The knowledge of $DDHA(... \| m^{N-2} \| m^{N-1})$ can not be used to compute the hash of $(... \| m^{N-2} \| m^{N-1} \| m^N)$.

3.3 Multicollisions

Many technique is developed to find Multicollisions of hash function, Joux's technique[1] and Kelsey/Schneier's technique[2] is representative technique.

3.3.1 Joux's technique

Joux [1] has proposed a technique to find a 2^k-collision for hash functions with n-bit hash values in $k \times 2^{n/2}$ as follow:



To a pair (h^i, h^{i+1}), If replace $m1^{i+1}$ with $m2^{i+1}$ will not change any parameter in follow calculation, it can apply Joux's technique.

Let *tm* is sum of the first i message blocks, and oco_i is the munber of o block before i-th chaining hash value h^i.

In DDHA, if replace $m1^{i+1}$ with $m2^{i+1}$ will change the parameter *tm* in follow calculation.

And it can alter Joux's technique to apply it on DDHA. To pair (h^i, h^{i+1}), it need find message blocks $(m3^0,...,m3^{i3-1}, m4^0,...,m4^{i3-1})$ that satisfy (3.1). let $ocou3_i$ is the munber of o block in $(m3^0,...,m3^{i-1})$, and $ocou4_i$ is the munber of o block in $(m4^0,...,m4^{i-1})$, And $ocou3_0 = ocou4_0 = 0$.

So to find Multicollisions of DDHA, to every pair chain value (h^i, h^{i+1}), it need find message blocks that satisfy (3.1). Then to 2^k-collision for DDHA, the message blocks must satify k systems that like (3.1).

$$\begin{cases} tm3 = \sum_{j=0}^{i-1} m3^j \\ tm4 = \sum_{j=0}^{i-1} m4^j \\ \sum_{j=0}^{i3-1} m3^j = \sum_{j=0}^{i3-1} m4^j \\ ocou3_{i3} = ocou4_{i3} \\ h3^1 = compression(repeattime, m3^0, h^i, tm3 + m3^0, oco_i, c1, c2) \\ h3^{ii+1} = compression(repeattime, m3^{ii}, h3^{ii}, tm3 + \sum_{j=0}^{ii} m3^j, oco_i + ocou3_{ii}, c1, c2) \quad ii = 1,...,(i3-2) \\ h^{i+1} = compression(repeattime, m3^{i3-1}, h3^{i3-1}, tm3 + \sum_{j=0}^{i3-1} m3^j, oco_i + ocou3_{i3-1}, c1, c2) \\ h4^1 = compression(repeattime, m4^0, h^i, tm4 + m4^0, oco_i, c1, c2) \\ h4^{ii+1} = compression(repeattime, m4^{ii}, h4^{ii}, tm4 + \sum_{j=0}^{ii} m4^j, oco_i + ocou4_{ii}, c1, c2) \quad ii = 1,...,(i3-2) \\ h^{i+1} = compression(repeattime, m4^{i4-1}, h4^{i4-1}, tm4 + \sum_{j=0}^{i3-1} m4^j, oco_i + ocou4_{i3-1}, c1, c2) \end{cases} \tag{3.1}$$

3.3.2 Kelsey/Schneier's technique

Kelsey/Schneier's technique bases on fixed-points of hash function. When constitute Multicollisions for a hash function, Kelsey/Schneier's technique[2] will change the order of the blocks.

In DDHA, change the order of the blocks maybe change the parameter *tm* or *oc* in some follow calculation. It is hard to apply Kelsey/Schneier's technique on DDHA. There is a simple way to resist this attack, it need use some parameter that is relate to the order of the block in blocks.

4. Improvement

In compression function, there are 512 data-depend circular shift operations, this will increase the calculation. If DDHA use a message expension function that has higher minimum hamming weight in less expand message words, it will make DDHA has same intensity with less calculation. There is a message expension function as follow:

$$
\begin{aligned}
& for \quad i=0 \quad to \quad 15 \\
& \quad em_i \leftarrow m_i \\
& next \quad i \\
& em_{16} \leftarrow \bigoplus_{i=0}^{15}(m_i) \\
& em_{17} \leftarrow 0 \\
& for \quad i=0 \quad to \quad 15 \\
& \quad em_{17} \leftarrow em_{17} \oplus ROTR^{i\times rl}(m_i) \\
& next \quad i \\
& em_{18} \leftarrow 0 \\
& for \quad i=0 \quad to \quad 15 \\
& \quad em_{18} \leftarrow em_{18} \oplus ROTR^{(16-i)\times rl}(m_i) \\
& next \quad i \\
& em_{19} \leftarrow m_7 \\
& for \quad i=0 \quad to \quad 6 \\
& \quad em_{19} \leftarrow em_{19} \oplus ROTR^{(1+i)\times rl}(m_i) \\
& next \quad i \\
& for \quad i=8 \quad to \quad 11 \\
& \quad em_{19} \leftarrow em_{19} \oplus ROTR^{(3+i)\times rl}(m_i) \\
& next \quad i \\
& for \quad i=12 \quad to \quad 15 \\
& \quad em_{19} \leftarrow em_{19} \oplus ROTR^{i\times rl}(m_i) \\
& next \quad i \\
& em_{20} \leftarrow ROTR^{7\times rl}(m_0) \oplus ROTR^{11\times rl}(m_1) \oplus ROTR^{8\times rl}(m_2) \\
& \qquad \oplus ROTR^{4\times rl}(m_3) \oplus ROTR^{9\times rl}(m_4) \\
& for \quad i=5 \quad to \quad 15 \\
& \quad em_{20} \leftarrow em_{20} \oplus m_i
\end{aligned}
$$

$$
\begin{aligned}
&next \quad i\\
&em_{16} \leftarrow em_{16} \oplus ROTR^{rl}(em_{16}) \oplus ROTR^{3\times rl}(em_{16}) \oplus ROTR^{7\times rl}(em_{16})\\
&em_{17} \leftarrow em_{17} \oplus ROTR^{rl}(em_{17}) \oplus ROTR^{3\times rl}(em_{17}) \oplus ROTR^{7\times rl}(em_{17})\\
&em_{18} \leftarrow em_{18} \oplus ROTR^{rl}(em_{18}) \oplus ROTR^{3\times rl}(em_{18}) \oplus ROTR^{7\times rl}(em_{18})\\
&em_{19} \leftarrow em_{19} \oplus ROTR^{rl}(em_{19}) \oplus ROTR^{3\times rl}(em_{19}) \oplus ROTR^{7\times rl}(em_{19})\\
&em_{20} \leftarrow em_{20} \oplus ROTR^{rl}(em_{20}) \oplus ROTR^{3\times rl}(em_{20}) \oplus ROTR^{7\times rl}(em_{20})
\end{aligned}
$$

function ME1 of DDHA

In DDHA-256, the word length is 32, rl is 2. In DDHA- 512, the word length is 64, rl is 4.

Function ME1 has a character, if a set include different expension message words, it has different minimum Hamming weight. We given the minimum Hamming weight here:

expension message words	minimum Hamming weight
$em_0,.....,em_{16}$	2
$em_0,.....,em_{17}$	4
$em_0,.....,em_{19}$	6
$em_0,.....,em_{20}$	8

Function ME1 will produce 21 expension message words which minimum Hamming weight is 8. it will reduce the calculation.

5. Conclusions

After study the technologys[wy05, Dau05] and the defence feature of data-depend function, and we find a message expension function that will make every defence path for DDHA will has at least eight data-depend circular shift defences, this make it hard to constitute a feasible difference path that has good possibility. Base on data-depende function and the message expension function, we design the hash function DDHA.

At the same time, we study other attack technologys[1,2] and length extension, and we use some measures in view of these technologys, these measures wreck the condition that applying the technologys need, this make it harder to apply these technologys on DDHA.

DDHA uses a value *repeattime* that user can set the value to change rounds to change the strength. It make it easy to raise the intensity of system.

So DDHA adopts various measures in view of the techniques that use to attack hash function, this will make DDHA will resist these attacks.

References:

[WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Cramer [Cra05], pages 19–35.

[Dau05] Magnus Daum. Cryptanalysis of Hash Functions of the MD4-Family. PhD thesis, Ruhr-Universit¨at Bochum, 2005.

[1] Antoine Joux. Multicollisions in iterated hash functions. application to cascaded construc-tions. In CRYPTO, 2004.

[2] John Kelsey and Bruce Schneier. Second preimages on n-bit hash functions for much ess than 2^n work. In EUROCRYPT, 2005.

Appendix 1: Difference of data-depend circular shift

Here we just discuss circular right shift. And we just discuss XOR differences[Dau05].

If $x \in F_2^n$, x has n-bits as:

$$x \mapsto (x_{n-1},...,x_0)$$

If y,x is n-bits word, n is 32(resp. 64), $0 \le r < 32(resp.64)$ is an integer that has 5(resp. 6) bits, then the circular right shift is:

$$y = ROTR^r(x) = ((x << (n-r)) \vee (x >> r))$$

If there is (x1,y1,r1) and (x2,y2,r2) meet:

$$y1 = ROTR^{r1}(x1) \qquad y2 = ROTR^{r2}(x2)$$

At fisrt, there is:($wbl = \log_2^n - 1$)

$$\begin{cases} r1 - r2 = \sum_{i=0}^{wbl}(\Delta^{\pm}(r1_i, r2_i) \times 2^i) \\ y1 := (x1_{r1-1},...,x1_0, x1_{n-1},...,x1_{r1}) \\ y2 := (x2_{r2-1},...,x2_0, x2_{n-1},...,x2_{r2}) \\ \Delta^{\oplus}(x1,x2) := (\Delta^{\oplus}(x1_{n-1},x2_{n-1}),...,\Delta^{\oplus}(x1_0,x2_0)) \\ \Delta^{\oplus}(y1,y2) := (\Delta^{\oplus}(x1_{r1-1},x2_{r2-1}),...,\Delta^{\oplus}(x1_{r1},x2_{r2})) \end{cases}$$

Let the **greatest common divisor of** $(r1-r2)$ **and** n is $\gcd = GCD(r1-r2,n)$. Then there exists:

1. if r1=r2, there has:

$$\begin{aligned}\Delta^{\oplus}(y1,y2) &= (\Delta^{\oplus}(x1_{r1-1},x2_{r1-1}),...,\Delta^{\oplus}(x1_0,x2_0),\Delta^{\oplus}(x1_{n-1},x2_{n-1}),...,\Delta^{\oplus}(x1_{r1},x2_{r1})) \\ &= ROTR^{r1}(\Delta^{\oplus}(x1,x2))\end{aligned}$$

So, if r1=r2, the difference of y1 and y2 will just depend on the difference of x1 and x2. of course it also depend on r1.

To given $\Delta^{\oplus}(y1,y2)$, there are 2^n $x1$. To given $(x1, \Delta^{\oplus}(y1,y2))$, there is a $x2$ that meet $x2 = x1 \oplus (LOTR^{r1}\Delta^{\oplus}(x1,x2))$. So there are 2^n pair $(x1,x2)$ has same $\Delta^{\oplus}(y1,y2)$.

2. if $r1 \ne r2$.

Divide $\Delta^{\oplus}(y1,y2)$ **and** $\Delta^{\oplus}(x1,x2)$ into $\gcd$ parts as follow:

$$\begin{cases} px_j := (\Delta^{\oplus}(x1_{(j+i\times\mathrm{gcd})\ \bmod\ n}, x2_{(j+i\times\mathrm{gcd})\ \bmod\ n}) \\ \quad | i=0,\ldots,(n/\mathrm{gcd}-1)) \qquad j=0,\ldots,\mathrm{gcd}-1 \\ py_j := (\Delta^{\oplus}(x1_{(j+r1+i\times\mathrm{gcd})\ \bmod\ n}, x2_{(r2+j+i\times\mathrm{gcd})\ \bmod\ n}) \\ \quad | i=0,\ldots,(n/\mathrm{gcd}-1)) \qquad j=0,\ldots,\mathrm{gcd}-1 \end{cases} \qquad (A.1)$$

To gven defence of patr px_j, py_j:

$$dx := (\Delta^{\oplus}(x1_{(j+i\times\mathrm{gcd})\ \bmod\ n}, x2_{(j+i\times\mathrm{gcd})\ \bmod\ n}) \\ | i=0,\ldots,(n/\mathrm{gcd}-1))$$

There are $2^{n/\mathrm{gcd}-1}$ diffence as follow:

$$dy1 := (\Delta^{\oplus}(x1_{(j+r1+i\times\mathrm{gcd})\ \bmod\ n}, x2_{(r2+j+i\times\mathrm{gcd})\ \bmod\ n}) \\ | i=0,\ldots,(n/\mathrm{gcd}-2))$$

To given pair (dx, dy1), there exists:

$$\bigoplus_{i=0}^{n/\mathrm{gcd}-1} \Delta^{\oplus}(x1_{(j+i\times\mathrm{gcd})\ \bmod\ n}, x2_{(j+i\times\mathrm{gcd})\ \bmod\ n}) \\ \oplus \Big(\bigoplus_{i=0}^{n/\mathrm{gcd}-2} \Delta^{\oplus}(x1_{(j+r1+i\times\mathrm{gcd})\ \bmod\ n}, x2_{(r2+j+i\times\mathrm{gcd})\ \bmod\ n})\Big) \\ = \Delta^{\oplus}(x1_{(j+n+r1-\mathrm{gcd})\ \bmod\ n}, x2_{(n+r2+j-\mathrm{gcd})\ \bmod\ n}) \qquad (A.2)$$

Proposition A.1: if $r1 \neq r2$, the possibility of a difference pair $\left(\Delta^{\oplus}(y1,y2), \Delta^{\oplus}(x1,x2)\right)$ is $2^{\mathrm{gcd}-n}$.

Proof:

At first, Divide $\Delta^{\oplus}(y1,y2)$ **and** $\Delta^{\oplus}(x1,x2)$ into gcd parts as (A.1), and every part satisfy (A.2). To given pair (dx, dy1), it will has the system:

$$\begin{cases} dx_{j,i} = x1_{(j+i\times\mathrm{gcd})\ \bmod\ n} \oplus x2_{(j+i\times\mathrm{gcd})\ \bmod\ n} \\ \qquad i=0,\ldots,(n/\mathrm{gcd}-1) \\ dy1_{j,i} = x1_{(j+r1+i\times\mathrm{gcd})\ \bmod\ n} \oplus x2_{(r2+j+i\times\mathrm{gcd})\ \bmod\ n} \\ \qquad i=0,\ldots,(n/\mathrm{gcd}-2) \end{cases} \qquad (A.3)$$

The system has $2\times(n/\mathrm{gcd}-1)$ variables and $2\times n/\mathrm{gcd}-3$ equations.

Apply elimination method on system (A.3), it will get (A.2). The system has two roots on GF(2).

The difference pair ($\Delta^{\oplus}(y1, y2)$, $\Delta^{\oplus}(x1, x2)$) include gcd parts that satisfy (A.2) (A.3). So there are 2^{gcd} pair $(x1, x2)$ satify these systems.

So there are 2^{gcd} pair $(x1, x2)$ have the given defference ($\Delta^{\oplus}(y1, y2)$, $\Delta^{\oplus}(x1, x2)$). Of course these pairs $(x1, x2)$ satisfy $x2 = \Delta^{\oplus}(x1, x2) \oplus x1$.

To given defference ($\Delta^{\oplus}(x1, x2)$), there are 2^n $x1$, And to given pair ($\Delta^{\oplus}(x1, x2), x1$), there is a $x2$ that satisfy $x2 = \Delta^{\oplus}(x1, x2) \oplus x1$, so there are 2^n pair $(x1, x2)$ have the given defference ($\Delta^{\oplus}(x1, x2)$).

So the possibility of a difference pair ($\Delta^{\oplus}(y1, y2)$, $\Delta^{\oplus}(x1, x2)$) is $2^{\mathrm{gcd}-n}$. □

Appendix 2: Message_expension(m)

In DDHA, the message m is expand from 16 words to 32 words. It can use a $512 \times 1024 (resp. 1024 \times 2048)$ generator matrix to describe it. It a little hard to find out the minimum deffences in expand message words with the big matrix. We will find out the minimum deffences in expand message words with other way.

At first, the follow facts is used to simplify the discussion:

1. Because the degree of the Algebraic Normal Form (ANF) that describe function ME(m) is 1. Finding out the minimum deffences in expand message words is be equal finding out the minimum Hamming weight of the expand message words when the Hamming weight of message bigger than 0.
2. The words in DDHA is carved up to sixteen parts. So it can describe a word as follow:

$$W := (w_{15}, ..., w_0)$$

Where $w_i := (b_J, ..., b_0) \quad 0 \le i < 16$, every part w_i has *J* bits. Then the message words *m* and expand message words *em* as follow:

$$\begin{cases} m := (m_{0,15}, m_{0,14}, ..., m_{0,0}, m_{1,15}, .., m_{15,0}) \\ em := (em_{0,15}, em_{0,14}, ..., em_{0,0}, em_{1,15}, ..., em_{31,0}) \\ em_{i,j} = m_{i,j} \qquad 0 \le i, j \le 15 \end{cases}$$

Then function ME(m) can be described with steps as follow, let *m1* and *m2* include 16 words.

$$\begin{cases} m1_i \leftarrow (\bigoplus_{j=0}^{15} m_j) \oplus m_i \qquad 0 \le i \le 15 \\ m2_i \leftarrow ROTR^i(m1_i) \qquad 0 \le i \le 15 \\ em_{i+16} \leftarrow (\bigoplus_{j=0}^{15} m2_j) \oplus m2_i \qquad 0 \le i \le 15 \end{cases}$$

Then there exists:

$$m2_{i,j} = m1_{i,(16-i+j) \mod 16}$$

Let HW(w) is Hamming weight of w. Then there exists:

Proposition B.1: If

$$\begin{cases} x := (x_0, ..., x_{15}) \qquad 0 \le i \le 15 \\ y := (y_0, ..., x_{15}) \qquad 0 \le i \le 15 \\ y_i = x_i \oplus (\bigoplus_{j=0}^{15} x_j) \qquad 0 \le i \le 15 \end{cases}$$

There exists:

1. If $\bigoplus_{j=0}^{15} x_j = 0$, then HW(y)=HW(x).
2. If $\bigoplus_{j=0}^{15} x_j = 1$, then HW(y)=16-HW(x).
3. If $HW(x) > 0$ and $\bigoplus_{j=0}^{15} x_j = 0$, then $HW(y) \geq 2$.
4. If $HW(x) > 0$ and $\bigoplus_{j=0}^{15} x_j = 1$, then $HW(y) \geq 1$.

proof:

There exists:

$$HW(x) = \sum_{j=0}^{15} x_j \leq 16 \qquad (B.1.1)$$

1. If $\bigoplus_{j=0}^{15} x_j = 0$ Then

$$y_i = x_i \oplus (\bigoplus_{j=0}^{15} x_j) = x_i \quad 0 \leq i \leq 15 \qquad (B.1.2)$$

Then

HW(y)=HW(x)

2. If $\bigoplus_{j=0}^{15} x_j = 1$ Then

$$y_i = x_i \oplus (\bigoplus_{j=0}^{15} x_j) = x_i \oplus 1 = \neg x_i \quad 0 \leq i \leq 15$$

Then

$$HW(y) = \sum_{i=0}^{15} y_i = \sum_{i=0}^{15} (1 - x_i) = 16 - \sum_{i=0}^{15} (x_i) = 16 - HW(x) \qquad (B.1.3)$$

3. If $HW(x) \geq 1$ and $\bigoplus_{j=0}^{15} x_j = 0$, if $HW(x) = 1$, there has $\bigoplus_{j=0}^{15} x_j = 1$. So:

$$HW(x) \geq 2$$

By (B.1.2), there exists: $HW(y) = HW(x) \geq 2$

4. If $HW(x) \geq 1$ and $\bigoplus_{j=0}^{15} x_j = 1$, and if $HW(x) = 16$, there has $\bigoplus_{j=0}^{15} x_j = 0$,

so there exists:

$$HW(x) \leq 15$$

By (B.1.3), there exists: $HW(y) = 16 - HW(x) \geq 16 - 15 = 1$ □

Proposition B.2: In message words of DDHA, if there exists $0 \leq j1 \leq 15$ make $\bigoplus_{i=0}^{15} m_{i,j1} = 1$, Then there exist $HW(em) \geq 16$.

Proof:

There has:

$$em_i = m_i \quad 0 \leq i \leq 15$$

$$\bigoplus_{i=0}^{15} em_{i,j1} = \bigoplus_{i=0}^{15} m_{i,j1} = 1$$

Suppose $I = \{i \mid m_{i,j1} = 1\}$ and $HW((em_{0,j1},...,em_{15,j1})) = H0$

There has: $m1_i = em_i \oplus (\bigoplus_{j=0}^{15} em_{i,j})$

By proposition B.1, thus

$$HW((m1_{0,j1},..., m1_{15,j1})) = 16 - H0$$
$$m1_{i,j1} = 1 \quad i \notin I$$

Let $J = \{(16 + j1 - i) \mod 16 \mid i \notin I \quad i = 0,...,15\}$, there are *16-H0* members in *J*.

Because $m2_{i,j} = m1_{i,(16-i+j) \mod 16}$

Then

$$\begin{aligned} m2_{i,j} &= m1_{i,(16-i+j) \mod 16} \quad i \notin I \quad j \in J \\ &= m1_{i,(16-i+i+j1) \mod 16} \\ &= m1_{i,j1} \\ &= 1 \end{aligned}$$

There seixst: $em_{i+16} \leftarrow (\bigoplus_{j=0}^{15} m2_j) \oplus m2_i \quad 0 \le i \le 15$

By proposition B.1, there exits:

$$HW((em_{16,j},..., m2_{31,j})) \ge 1 \quad j \in J$$
$$\begin{aligned} HW((em_{16},..., m2_{31})) &= (\sum_{j \in J} \sum_{i=16}^{31} em_{i,j}) + \sum_{j \notin J} \sum_{i=16}^{31} em_{i,j} \\ &\ge \sum_{j \in J} \sum_{i=16}^{31} em_{i,j} \\ &\ge 16 - H0 \end{aligned}$$

Then

$$\begin{aligned} HW(em) &= HW((em_0,..., em_{15})) + HW((em_{16},..., em_{31})) \\ &\ge H0 + (16 - H0) \\ &= 16 \end{aligned}$$

□

Proposition B.3: In message words of DDHA, if $HW(m) \ge 0$ there exist $HW(em) \ge 8$.

Proof:

There exists:

$$em_i = m_i \quad 0 \le i \le 15$$
$$HW((em_0,..., em_{15})) = HW((m_0,..., m_{15}))$$

1.if there exists *j1* make $\bigoplus_{i=0}^{15} em_{i,j1} = 1$, by proposition B.2, there has exists:

$$HW(em) \geq 16 > 8 \qquad (B.3.a).$$

2. if there exists

$$\bigoplus_{i=0}^{15} em_{i,j} = 0 \quad 0 \leq j \leq 15$$

Then there exists:

$$m1_{i,j} = em_{i,j} \oplus (\bigoplus_{i1=0}^{15} em_{i1,j}) = em_{i,j} \quad 0 \leq i, j \leq 15$$
$$m2_{i,j} = m1_{i,(16-i+j) \mod 16}$$

Let $I0_j = \{i \mid em_{i,j} = 1 \quad 0 \leq i \leq 15 \quad 0 \leq j \leq 15\}$, $I1_j = \{i \mid m1_{i,j} = 1 \quad 0 \leq i \leq 15 \quad 0 \leq j \leq 15\}$ then:

$$\begin{aligned} I1_j &= \{i \mid m1_{i,j} = 1 \quad 0 \leq i \leq 15 \quad 0 \leq j \leq 15\} \\ &= \{i \mid em_{i,j} = 1 \quad 0 \leq i \leq 15 \quad 0 \leq j \leq 15\} \\ &= I0_j \end{aligned}$$
$$HW(m1) = HW((em_0, ..., em_{15}))$$

Let $JB0 = \{jb_0, ...\} = \{j \mid (\sum_{i=0}^{15} I1_{i,j}) > 0 \quad 15 \geq j \geq 0\}$, and because

$$\bigoplus_{i=0}^{15} em_{i,j} = 0 \quad 0 \leq j \leq 15$$

Then by proposition B.1:

$$(\sum_{i \in I1_j} m1_{i,j}) = (\sum_{i \in I1_j} em_{i,j}) \geq 2 \qquad j \in JB0 \qquad (B.3.1)$$

Then there has:

$$m2_{i,j} = m1_{i,(16-i+j) \mod 16}$$

So there exist:

$$HW((m2_0, ..., m2_{15})) = HW((m1_0, ..., m1_{15})) = HW((em_0, ..., em_{15})) \qquad (B.3.2)$$

2.1 If there exists *i1c* make $(\bigoplus_{i=0}^{15} m2_{i,i1c}) = 1$, by proposition B.1 and (B.3.2), there exists:

$$\begin{aligned} HW((em_{16}, ..., em_{31})) &\geq HW((em_{16,i1c}, ..., em_{31,i1c})) \\ &= 16 - HW((m2_{0,i1c}, ..., m2_{15,i1c})) \\ HW((em_0, ..., em_{31})) &= HW((em_0, ..., em_{15})) + HW((em_{16}, ..., em_{31})) \\ &\geq HW((m2_0, ..., m2_{15})) + 16 - HW((m2_{0,i1c}, ..., m2_{15,i1c})) \\ &\geq HW((m2_{0,i1c}, ..., m2_{15,i1c})) + 16 - HW((m2_{0,i1c}, ..., m2_{15,i1c})) \\ &= 16 > 8 \qquad (B.3.b) \end{aligned}$$

2.2 If $(\bigoplus_{i=0}^{15} m2_i) = 0$. Then there exist:

$$em_{i,j} = m2_{i-16,j} \oplus (\bigoplus_{i1=0}^{15} m2_{i1,j}) = m2_{i-16,j} \qquad 16 \le i \le 31$$
$$HW((em_{16},...,em_{31})) = HW(m2) \qquad (B.3.3)$$

Because $HW(m) \ge 0$, there are at least one member in *JB0*. Let $jb0 = jb_0$, and by proposition B.1, there are at least two members in $I1_{jb0}$. Suppose $i1a \ne i1b \in I1_{jb0}$, then:

$$m1_{i1a,jb0} = 1 \quad and \quad m1_{i1b,jb0} = 1$$

Then there has:

$$m1_{i1a,jb0} = m2_{i1a,\ (16+jb0-i1a) \bmod 16} = 1$$
$$m1_{i1b,jb0} = m2_{i1b,\ (16+jb0-i1b) \bmod 16} = 1$$

Because there has: $0 \le jb0, i1a, i1b \le 15$ and $i1a \ne i1b$

There has: $(16 + jb0 - i1a) \bmod 16 \ne (16 + jb0 - i1b) \bmod 16$

Because $(\bigoplus_{i=0}^{15} m2_i) = 0$, Then by proposition B.1:

$$(\sum_{i=0}^{15} m2_{i,\ (16+jb0-i1a) \bmod 16}) \ge 2$$
$$(\sum_{i=0}^{15} m2_{i,\ (16+jb0-i1b) \bmod 16}) \ge 2$$
$$\begin{aligned} HW(m2) \ge\ & HW((m2_{0,\ (16+jb0-i1a) \bmod 16},....,m2_{15,\ (16+jb0-i1a) \bmod 16})) + \\ & HW((m2_{0,\ (16+jb0-i1b) \bmod 16},....,m2_{15,\ (16+jb0-i1b) \bmod 16})) \\ \ge\ & 4 \end{aligned} \qquad (B.3.4)$$

By (B.3.2) (B.3.3) (B.3.4), there exist:

$$\begin{aligned} HW((em_0,...,em_{31})) &= HW((em_0,...,em_{15})) + HW((em_{16},...,em_{31})) \\ &= 2 \times HW(m2) \\ &\ge 2 \times 4 = 8 \qquad | \qquad (B.3.c) \end{aligned}$$

So by (B.3.a) (B.3.b) (B.3.c), if $HW(m) \ge 0$ there exist $HW(em) \ge 8$. □

Because every part of every expand message words is as parameter of data-depend circular shift once. Theroem B.3 means in any difference path for DDHA, there will be at least eight data-depend circular shift defference.