Data-Depend Hash Algorithm

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

Abstract: We study some technologies 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 Algorithm (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 technologies that analyse and attack hash algorithm are developed. The major technologies 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 appendix 1, we know that the data-depend circular shift has good defence feature. And we find a message expansion function that make any difference path will has at least eight data-depend circular shift difference [appendix 2]. This make it hard to constitute a feasible difference path.

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

The following operations are applied to 32-bit or 64-bit words in
DDHA:
1. ← variable assignment
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 $0 \leq n < 32$ (resp. $0 \leq n < 64$).
5. The shift left operation, $SHL^n(x)$, where x is a 32-bit or 64-bit word and n is an integer with $0 \leq n < 32$ (resp. $0 \leq n < 64$).
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 $0 \leq n < 32$ (resp. $0 \leq n < 64$).
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 $0 \leq n < 32$ (resp. $0 \leq n < 64$).

2. Data-Depend Hash Algorithm (DDHA)

DDHA has two hash functions: DDHA-256 (32-bit version), DDHA-512 (64-bit version). 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:

<table>
<thead>
<tr>
<th></th>
<th>word</th>
<th>Message size</th>
<th>Block size</th>
<th>Hash value size</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDHA-256</td>
<td>32</td>
<td>$&lt; 2^{64}$</td>
<td>512</td>
<td>256</td>
</tr>
<tr>
<td>DDHA-512</td>
<td>64</td>
<td>$&lt; 2^{64}$</td>
<td>1024</td>
<td>512</td>
</tr>
</tbody>
</table>

Properties of DDHA hash functions(size in bits)

In DDHA, the message will be preprocessed. After message is preprocessed, the message will parsed 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 $L+1+k \equiv 448 \mod 512$ (resp. $L+1+k \equiv 960 \mod 1024$). 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):

<table>
<thead>
<tr>
<th>DDHA-256</th>
<th>DDHA-512</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_0^0 = 0x6a09e667$,</td>
<td>$H_0^0 = 0x6a09e667 f3bcc908$,</td>
</tr>
<tr>
<td>$H_1^0 = 0xbb67ae85$,</td>
<td>$H_1^0 = 0xbb67ae85c4aa73b$,</td>
</tr>
<tr>
<td>$H_2^0 = 0x3c6ef372$,</td>
<td>$H_2^0 = 0x3c6ef372 f6e894$,</td>
</tr>
<tr>
<td>$H_3^0 = 0xa54ff53a$,</td>
<td>$H_3^0 = 0xa54ff53a5f1d36f1$,</td>
</tr>
<tr>
<td>$H_4^0 = 0x510527f$,</td>
<td>$H_4^0 = 0x510527 fade682d1f$,</td>
</tr>
<tr>
<td>$H_5^0 = 0xb05688c$,</td>
<td>$H_5^0 = 0xb05688c2b3e6c1f$,</td>
</tr>
<tr>
<td>$H_6^0 = 0xf83d9ab$,</td>
<td>$H_6^0 = 0xf83d9abf41bd6b$,</td>
</tr>
<tr>
<td>$H_7^0 = 0x5be0cd19$,</td>
<td>$H_7^0 = 0x5be0cd19137e2179$,</td>
</tr>
</tbody>
</table>

The initial hash value for DDHA

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

<table>
<thead>
<tr>
<th>DDHA-256</th>
<th>DDHA-512</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{1_0} = 0xd76aa478$,</td>
<td>$C_{1_0} = 0xd76aa478 fff39a3d$,</td>
</tr>
<tr>
<td>$C_{1_1} = 0xe8c7b756$,</td>
<td>$C_{1_1} = 0xe8c7b7568771 f681$,</td>
</tr>
<tr>
<td>$C_{1_2} = 0x242070db$,</td>
<td>$C_{1_2} = 0x242070db699d122$,</td>
</tr>
<tr>
<td>$C_{1_3} = 0xc1bdc3e4$,</td>
<td>$C_{1_3} = 0xc1bdc3e4fde5380c$,</td>
</tr>
<tr>
<td>$C_{1_4} = 0xf57cafa$,</td>
<td>$C_{1_4} = 0xf57cafa4beea44$,</td>
</tr>
<tr>
<td>$C_{1_5} = 0x478762a$,</td>
<td>$C_{1_5} = 0x478762a4becf0a9$,</td>
</tr>
<tr>
<td>$C_{1_6} = 0xa8304613$,</td>
<td>$C_{1_6} = 0xa8304613 f6bba6b60$,</td>
</tr>
<tr>
<td>$C_{1_7} = 0xfd469501$,</td>
<td>$C_{1_7} = 0xfd469501becfb7c70$,</td>
</tr>
<tr>
<td>$C_{1_8} = 0x698098d8$,</td>
<td>$C_{1_8} = 0x698098d8289b7ec6$,</td>
</tr>
</tbody>
</table>
\[ C_{19} = 0x8b44f7af, \quad C_{19} = 0x8b44f7afeaa127fa, \]
\[ C_{10} = 0xffff5bb1, \quad C_{10} = 0xffff5bb1d4ef3085, \]
\[ C_{11} = 0x895cd7be, \quad C_{11} = 0x895cd7be04881d05, \]
\[ C_{12} = 0x6b901122, \quad C_{12} = 0x6b901122d9d4d039, \]
\[ C_{13} = 0xfd987193, \quad C_{13} = 0xfd987193e6db99e5, \]
\[ C_{14} = 0xa679438e, \quad C_{14} = 0xa679438efa27cf8, \]
\[ C_{15} = 0x49b40821, \quad C_{15} = 0x49b40821c4ae5665, \]

**Constants C1 of DDHA**

C2 as follow:

<table>
<thead>
<tr>
<th>DDHA-256</th>
<th>DDHA-512</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{20} = 0xf61e2562, )</td>
<td>( C_{20} = 0xf61e2562f4292244, )</td>
</tr>
<tr>
<td>( C_{21} = 0xc040b340, )</td>
<td>( C_{21} = 0xc040b340432aff97, )</td>
</tr>
<tr>
<td>( C_{22} = 0x265e5a51, )</td>
<td>( C_{22} = 0x265e5a51ab9423a7, )</td>
</tr>
<tr>
<td>( C_{23} = 0xe9b6c7aa, )</td>
<td>( C_{23} = 0xe9b6c7aafec93a039, )</td>
</tr>
<tr>
<td>( C_{24} = 0xd62f105d, )</td>
<td>( C_{24} = 0xd62f105d655b59c3, )</td>
</tr>
<tr>
<td>( C_{25} = 0x02441453, )</td>
<td>( C_{25} = 0x02441453f0ccc92, )</td>
</tr>
<tr>
<td>( C_{26} = 0xd8a1e681, )</td>
<td>( C_{26} = 0xd8a1e681ffe47df, )</td>
</tr>
<tr>
<td>( C_{27} = 0xe7d3fbc8, )</td>
<td>( C_{27} = 0xe7d3fbc885845dd1, )</td>
</tr>
<tr>
<td>( C_{28} = 0x21e1cde6, )</td>
<td>( C_{28} = 0x21e1cde66fa87e4f, )</td>
</tr>
<tr>
<td>( C_{29} = 0xc33707d6, )</td>
<td>( C_{29} = 0xc33707d6fe2ce6e0, )</td>
</tr>
<tr>
<td>( C_{30} = 0xf4d50d87, )</td>
<td>( C_{30} = 0xf4d50d87a3014314, )</td>
</tr>
<tr>
<td>( C_{31} = 0x55a14ed, )</td>
<td>( C_{31} = 0x55a14ed4e0811a1, )</td>
</tr>
<tr>
<td>( C_{32} = 0xa9e3e905, )</td>
<td>( C_{32} = 0xa9e3e905f7537e82, )</td>
</tr>
<tr>
<td>( C_{33} = 0xfcfa3f8, )</td>
<td>( C_{33} = 0xfcfa3f8bd3af235, )</td>
</tr>
<tr>
<td>( C_{34} = 0x676f02d9, )</td>
<td>( C_{34} = 0x676f02d92ad7d2bb, )</td>
</tr>
<tr>
<td>( C_{35} = 0x8d2a4c8a, )</td>
<td>( C_{35} = 0x8d2a4c8aeba86d391, )</td>
</tr>
</tbody>
</table>

**Constants C2 of DDHA**

2.2 processing.

If there are N message blocks \( M_0, \ldots, M_{N-1} \).

The DDHA has a compression function. The input of compression function include chaining variable(8 words, \( H'_0, \ldots, H'_7 \)), message block(16 words, \( m'_0, \ldots, m'_15 \)), constants(32 words, \( C_{10}, \ldots, C_{15}, C_{20}, \ldots, C_{25} \)), and other parameters. Then the processing as follow:
for $j = 0$ to 17
    $tm_j \leftarrow 0$
next $j$

$oc_0 = 0$

$oc_1 = 0$

for $i = 0$ to $N - 2$
    $h^{i+1} \leftarrow \text{compression(repeattime, } m^i, h^i, tm, oc, ct1, ct2)$
    temp = 0
    occ = 1
    for $j = 0$ to 15
        $tm_j = tm_j + m_j + \text{temp}$
        if $(tm_j < m_j)$ then
            temp = 1
        else if $(tm_j > m_j)$ then
            temp = 0
        end if
        if $m_j \neq 0$ then occ = 0
    next $j$

$oc_0 = oc_0 + occ$

if $oc_1 = 0$ then $oc_1 = oc_1 + occ$

$tm_{16} = tm_{16} + \text{temp}$

if $tm_{16} = 0$ then $tm_{17} = tm_{17} + \text{temp}$

next $i$

$h^N \leftarrow \text{compression(repeattime, } m^{N-1}, -h^{N-1}, oc, tm, -c1, -c2)$

return $h^N$

Processing of DDHA

2.3 compression function

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

* an integer value $\text{repeattime}$. User can set $\text{repeattime}$ to get higher intensity. The default value of $\text{repeattime}$ is 1.
* a message block $m^i = m^i_0, ..., m^i_{31}$
* a chain value $h = h_0, ..., h_7$
* an value $tm = tm_0, ..., tm_{17}$.
* an value \( oc = oc_0, oc_1 \).
* a Constant \( ct1 = ct1_0, \ldots, ct1_{15} \)
* a Constant \( ct2 = ct2_0, \ldots, ct2_{15} \)

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 \( h = h_0, \ldots, h_7 \)
* a message block \( m = m_0, \ldots, m_{12} \)
* a Constant \( ct1 = ct1_0, \ldots, ct1_{15} \)
* a Constant \( ct2 = ct2_0, \ldots, ct2_{15} \)

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

```plaintext
for i = 0 to 15
    im ← i >> 1 + (i mod 2)×8
    h_0 ← (h_0 + m_{im})
    for j = 1 to 7
        im1 ← (i >> 1 + j) mod 8 + (i mod 2)×8
        h_j ← ROTR^{m_{im>>(j-1)×8}}(h_j + m_{im1})
    next j
    h_4 ← ROTR^{m_{im>>(7×8)}}(h_4 + h_0)
    t ← h_1 + h_5 + h_6 + h_7
    for j = 1 to 4
        h_j ← ROTR^{m_{im>>(j+7)×8}}(t - h_j) + ct1_i
    next j
    t ← h_4 + h_5 + h_6 + h_7
    for j = 4 to 7
        h_j ← ROTR^{m_{im>>(j+8)×8}}(t - h_j) + ct2_i
    next j
    t ← h_7
    for j = 7 to 1
        h_j ← h_{j-1}
    next j
```
h₀ ← t
next 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 expansion function ME(m)
The message expansion function ME(m) takes as input one value:
* a message block \( m = m₀, \ldots, m₁₅ \)
And ME(m) as follow:

\[
t = \bigoplus_{i=0}^{15} (m_i)
\]
for \( i = 0 \) to \( 15 \)
\[m_i \leftarrow (t \oplus m_i)\]
next i
for \( i = 0 \) to \( 15 \)
\[m_i \leftarrow \text{ROTR}^{i\text{rd}}(m_i)\]
next i
\[t = \bigoplus_{i=0}^{15} (m_i)\]
for \( i = 0 \) to \( 15 \)
\[m_i \leftarrow (m_i \oplus t)\]
next i
return 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:

\[h₁ \leftarrow h'\]
\[c₃ \leftarrow ct1\]
\[c₄ \leftarrow ct2\]
\[ma \leftarrow m\]
\[mb \leftarrow ME(m)\]
for \( j = 0 \) to \( 15 \)
\[c₃ j = c₃ j \oplus tm_j\]
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 $(r_i - r_j) \neq 0$, the possibility of a data-depend circular shift difference is $2^{gcd-n}$, $gcd$ is the greatest common divisor of $(r_i-r_j)$ and $n$.

In DDHA, the parameter of data-depend circular shift less than 4(resp.16), then there has:
So the possibility of a difference path for DDHA will be:

\[ p \leq 2^{(\text{gcd} - n) \times 8} \leq 2^{16-8\alpha} (\text{resp.} 2^{64-8\alpha}) \]

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

Let \( tm \) is sum of the message blocks before last block.

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

### 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 \(2^{k/2}\)-collision for hash functions with n-bit hash values in \( k \times 2^{n/2} \) as follow:
To a pair \((h', h^{i+1})\), if replace \(m_1^{i+1}\) with \(m_2^{i+1}\) 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 number of block before \(i\)-th chaining hash value \(h'\).

In DDHA, if replace \(m_1^{i+1}\) with \(m_2^{i+1}\) will change the parameter \(tm\) in follow calculation.

And it can alter Joux's technique to apply it on DDHA. To pair \((h', h^{i+1})\), it need find message blocks \((m_3^{1}, \ldots, m_3^{i-1}, m_4^{0}, \ldots, m_4^{i-1})\) that satisfy (3.1). let \(oco_{3i}\) is the number of block in \((m_3^{1}, \ldots, m_3^{i-1})\), and \(oco_{4i}\) is the number of block in \((m_4^{0}, \ldots, m_4^{i-1})\). And \(oco_{30} = oco_{40} = 0\).

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

\[
\begin{align*}
\text{tm} & = \sum_{j=0}^{i-1} m_3^j \\
\text{tm} & = \sum_{j=0}^{i-1} m_4^j \\
\sum_{j=0}^{i-1} m_3^j & = \sum_{j=0}^{i-1} m_4^j \\
oco_{3i} & = oco_{4i} \\
h_3^i & = \text{compression(repeattime, } m_3^{0}, h', \text{tm} + m_3^{0}, oco_i, c1, c2) \\
h_3^{i+1} & = \text{compression(repeattime, } m_3^{i}, h_3^i, \text{tm} + \sum_{j=0}^{i} m_3^j, oco_i + oco_{3i}, c1, c2) \quad ii = 1, \ldots, (i3 - 2) \\
h_4^i & = \text{compression(repeattime, } m_4^{0}, h', \text{tm} + m_4^{0}, oco_i, c1, c2) \\
h_4^{i+1} & = \text{compression(repeattime, } m_4^{i}, h_4^i, \text{tm} + \sum_{j=0}^{i} m_4^j, oco_i + oco_{4i}, c1, c2) \quad ii = 1, \ldots, (i3 - 2) \\
h_2^i & = \text{compression(repeattime, } m_2^{0}, h', \text{tm} + m_2^{0}, oco_i, c1, c2) \\
h_2^{i+1} & = \text{compression(repeattime, } m_2^{i}, h_2^i, \text{tm} + \sum_{j=0}^{i} m_2^j, oco_i + oco_{2i}, c1, c2) \quad ii = 1, \ldots, (i3 - 2) \\
\end{align*}
\]

\[3.3.2\text{ Kelsey/Schneier’s technique}\]

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 expansion 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 expansion function as follow:

```
for  i = 0  to  15
     em_i <- m_i
next  i
em_{16} <- \bigoplus_{i=0}^{15}(m_i)
em_{17} <- 0
for  i = 0  to  15
     em_{17} <- em_{17} \oplus ROTR^{i\times t}(m_i)
next  i
em_{18} <- 0
for  i = 0  to  15
     em_{18} <- em_{18} \oplus ROTR^{(16-i)\times t}(m_i)
next  i
em_{19} <- m_7
for  i = 0  to  6
     em_{19} <- em_{19} \oplus ROTR^{(i+1)\times t}(m_i)
next  i
for  i = 8  to  11
     em_{19} <- em_{19} \oplus ROTR^{(3+i)\times t}(m_i)
next  i
for  i = 12  to  15
     em_{19} <- em_{19} \oplus ROTR^{i\times t}(m_i)
next  i
em_{20} <- ROTR^{7\times t}(m_0) \oplus ROTR^{1\times t}(m_1) \oplus ROTR^{8\times t}(m_2)
     \oplus ROTR^{4\times t}(m_3) \oplus ROTR^{9\times t}(m_4)
for  i = 5  to  15
     em_{20} <- em_{20} \oplus m_i
```
Next $i$

\[ em_{16} \leftarrow em_{16} \oplus \text{ROTR}^i(em_{16}) \oplus \text{ROTR}^{3\text{rd}}(em_{16}) \oplus \text{ROTR}^{7\text{rd}}(em_{16}) \]
\[ em_{17} \leftarrow em_{17} \oplus \text{ROTR}^i(em_{17}) \oplus \text{ROTR}^{3\text{rd}}(em_{17}) \oplus \text{ROTR}^{7\text{rd}}(em_{17}) \]
\[ em_{18} \leftarrow em_{18} \oplus \text{ROTR}^i(em_{18}) \oplus \text{ROTR}^{3\text{rd}}(em_{18}) \oplus \text{ROTR}^{7\text{rd}}(em_{18}) \]
\[ em_{19} \leftarrow em_{19} \oplus \text{ROTR}^i(em_{19}) \oplus \text{ROTR}^{3\text{rd}}(em_{19}) \oplus \text{ROTR}^{7\text{rd}}(em_{19}) \]
\[ em_{20} \leftarrow em_{20} \oplus \text{ROTR}^i(em_{20}) \oplus \text{ROTR}^{3\text{rd}}(em_{20}) \oplus \text{ROTR}^{7\text{rd}}(em_{20}) \]

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 expansion message words, it has different minimum Hamming weight. We given the minimum Hamming weight here:

<table>
<thead>
<tr>
<th>Expansion message words</th>
<th>Minimum Hamming weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$em_{0},...,em_{16}$</td>
<td>2</td>
</tr>
<tr>
<td>$em_{0},...,em_{17}$</td>
<td>4</td>
</tr>
<tr>
<td>$em_{0},...,em_{19}$</td>
<td>6</td>
</tr>
<tr>
<td>$em_{0},...,em_{20}$</td>
<td>8</td>
</tr>
</tbody>
</table>

Function ME1 will produce 21 expansion 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 expansion 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 expansion 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 \textit{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:

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:
\[
x \mapsto (x_{n-1}, \ldots, x_0)
\]

If \( y, x \) is \( n \)-bits word, \( n \) is 32 (resp. 64), \( 0 \leq r < 32 \) (resp. 64) is an integer that has 5 (resp. 6) bits, then the circular right shift is:
\[
y = ROTR^r(x) = ((x \ll (n - r)) \lor (x >> r))
\]

If there is \((x_1, y_1, r_1)\) and \((x_2, y_2, r_2)\) meet:
\[
y_1 = ROTR^{r_1}(x_1) \quad y_2 = ROTR^{r_2}(x_2)
\]

At first, there is:
\[
\begin{align*}
\text{let} \quad wbl &= \log_2^n - 1 \\
r_1 - r_2 &= \sum_{i=0}^{wbl} (\Delta^+(r_1, r_2) \times 2^i) \\
y_1 &= (x_{r_1-1}, \ldots, x_0, x_{n-1}, \ldots, x_{r_1}) \\
y_2 &= (x_{r_2-1}, \ldots, x_0, x_{n-1}, \ldots, x_{r_2}) \\
\Delta^\oplus(x_1, x_2) &= (\Delta^\oplus(x_{n-1}, x_{2-1}), \ldots, \Delta^\oplus(x_0, x_0)) \\
\Delta^\oplus(y_1, y_2) &= (\Delta^\oplus(x_{r_1-1}, x_{2-1}), \ldots, \Delta^\oplus(x_{r_1}, x_{r_2}))
\end{align*}
\]

Let the greatest common divisor of \((r_1 - r_2)\) and \( n \) is
\[
\text{gcd} = \text{GCD}(r_1 - r_2, n)
\]

Then there exists:

1. if \( r_1 = r_2 \), there has:
\[
\Delta^\oplus(y_1, y_2) = (\Delta^\oplus(x_{r_1-1}, x_{2-1}), \ldots, \Delta^\oplus(x_0, x_0), \Delta^\oplus(x_{n-1}, x_{2-1}), \ldots, \Delta^\oplus(x_{r_1}, x_{r_2}))
\]

So, if \( r_1 = r_2 \), the difference of \( y_1 \) and \( y_2 \) will just depend on the difference of \( x_1 \) and \( x_2 \). of course it also depend on \( r_1 \).

To given \( \Delta^\oplus(y_1, y_2) \), there are \( 2^n \times x_1 \). To given \((x_1, \Delta^\oplus(y_1, y_2))\), there is a \( x_2 \) that meet \( x_2 = x_1 \oplus (LOTR^{r_1} \Delta^\oplus(x_1, x_2)) \). So there are \( 2^n \) pair \((x_1, x_2)\) has same \( \Delta^\oplus(y_1, y_2) \).

2. if \( r_1 \neq r_2 \).

Divide \( \Delta^\oplus(y_1, y_2) \) and \( \Delta^\oplus(x_1, x_2) \) into \( \text{gcd} \) parts as follow:
\[
\begin{align*}
px_j &:= (\Delta^\oplus (x_{1(j+i\times \gcd)} \mod n, x_{2(j+i\times \gcd)} \mod n) \\
i &= 0,\ldots,(n / \gcd - 1)) & j &= 0,\ldots, \gcd - 1 \\
py_j &:= (\Delta^\oplus (x_{1(j+r1+i\times \gcd)} \mod n, x_{2(r2+j+i\times \gcd)} \mod n) \\
i &= 0,\ldots,(n / \gcd - 1)) & j &= 0,\ldots, \gcd - 1
\end{align*}
\] (A.1)

To given defence of patr \( px_j, py_j \):

\[
\begin{align*}
dx &:= (\Delta^\oplus (x_{1(j+i\times \gcd)} \mod n, x_{2(j+i\times \gcd)} \mod n) \\
i &= 0,\ldots,(n / \gcd - 1)) \\
dy1 &:= (\Delta^\oplus (x_{1(j+r1+i\times \gcd)} \mod n, x_{2(r2+j+i\times \gcd)} \mod n) \\
i &= 0,\ldots,(n / \gcd - 2))
\end{align*}
\]

There are \(2^{n / \gcd - 1}\) difference as follow:

\[
\begin{align*}
\bigoplus_{i=0}^{n / \gcd - 1} \Delta^\oplus (x_{1(j+i\times \gcd)} \mod n, x_{2(j+i\times \gcd)} \mod n) \\
+ \bigoplus_{i=0}^{n / \gcd - 2} \Delta^\oplus (x_{1(j+r1+i\times \gcd)} \mod n, x_{2(r2+j+i\times \gcd)} \mod n)
= \Delta^\oplus (x_{1(j+n+\gcd - \gcd)} \mod n, x_{2(n+r2+j-\gcd)} \mod n) \\
&= \Delta^\oplus (x_{1(j+n+r1-i\times \gcd)} \mod n, x_{2(2r2+j-i\times \gcd)} \mod n) (A.2)
\end{align*}
\]

**Proposition A.1:** if \( r1 \neq r2 \), the possibility of a difference pair \((\Delta^\oplus(y_1, y_2), \Delta^\oplus(x_1, x_2))\) is \(2^{\gcd - n}\).

**Proof:**

At first, Divide \( \Delta^\oplus(y_1, y_2) \) and \( \Delta^\oplus(x_1, x_2) \) into \( \gcd \) parts as (A.1), and every part satisfy (A.2). To given pair \((dx, dy1)\), it will has the system:

\[
\begin{align*}
dx_{j,i} &= x_{1(j+i\times \gcd)} \mod n \bigoplus x_{2(j+i\times \gcd)} \mod n \\
i &= 0,\ldots,(n / \gcd - 1) \\
dy1_{j,i} &= x_{1(j+r1+i\times \gcd)} \mod n \bigoplus x_{2(r2+j+i\times \gcd)} \mod n \\
i &= 0,\ldots,(n / \gcd - 2)
\end{align*}
\] (A.3)

The system has \(2 \times (n / \gcd - 1)\) variables and \(2 \times n / \gcd - 3\) equations.
Apply elimination method on system (A.3), it will get (A.2). The system has two roots on GF(2).

The difference pair \((\Delta^0(y_1, y_2), \Delta^0(x_1, x_2))\) include \(\text{gcd}\) parts that satisfy (A.2) (A.3). So there are \(2^{\text{gcd}}\) pair \((x_1, x_2)\) satisfy these systems.

So there are \(2^{\text{gcd}}\) pair \((x_1, x_2)\) have the given difference \((\Delta^0(y_1, y_2), \Delta^0(x_1, x_2))\). Of course these pairs \((x_1, x_2)\) satisfy \(x_2 = \Delta^0(x_1, x_2) \oplus x_1\).

To given difference \((\Delta^0(x_1, x_2))\), there are \(2^n\) \(x_1\), And to given pair \((\Delta^0(x_1, x_2), x_1)\), there is a \(x_2\) that satisfy \(x_2 = \Delta^0(x_1, x_2) \oplus x_1\), so there are \(2^n\) pair \((x_1, x_2)\) have the given difference \((\Delta^0(x_1, x_2))\).

So the possibility of a difference pair \((\Delta^0(y_1, y_2), \Delta^0(x_1, x_2))\) is \(2^{\text{gcd} - n}\). \(\square\)
Appendix 2: Message expansion(m)

In DDHA, the message \( m \) is expand from 16 words to 32 words. It can use a \( 512 \times 1024 \) \((\text{resp.} 1024 \times 2048)\) generator matrix to describe it. It a little hard to find out the minimum defences in expand message words with the big matrix. We will find out the minimum defences 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 \( \text{ME}(m) \) is 1. Finding out the minimum defences 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:

\[
W := (w_{15}, \ldots, w_0)
\]

Where \( w_i := (b_j, \ldots, b_0) \) \( 0 \leq i < 16 \), every part \( w_i \) has \( J \) bits. Then the message words \( m \) and expand message words \( em \) as follow:

\[
\begin{align*}
m & := (m_{0,15}, m_{0,14}, \ldots, m_{1,15}, \ldots, m_{15,0}) \\
em & := (em_{0,15}, em_{0,14}, \ldots, em_{0,0}, em_{4,15}, \ldots, em_{31,0}) \\
em_{i,j} & = m_{i,j} \quad 0 \leq i, j \leq 15
\end{align*}
\]

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

\[
\begin{align*}
m_{1,i} & \leftarrow (\bigoplus_{j=0}^{15} m_j) \oplus m_i \quad 0 \leq i \leq 15 \\
m_{2,i} & \leftarrow \text{ROTR}'(m_{1,i}) \quad 0 \leq i \leq 15 \\
em_{i+16} & \leftarrow (\bigoplus_{j=0}^{15} m_{2,j}) \oplus m_{2,i} \quad 0 \leq i \leq 15
\end{align*}
\]

Then there exists:

\[
m_{2,i,j} = m_{1,(i+16-j)} \mod 16
\]

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

\textbf{Proposition B.1}: If

\[
\begin{align*}
x & := (x_0, \ldots, x_{15}) \quad 0 \leq i \leq 15 \\
y & := (y_0, \ldots, y_{15}) \quad 0 \leq i \leq 15 \\
y_i & = x_i \oplus (\bigoplus_{j=0}^{15} x_j) \quad 0 \leq i \leq 15
\end{align*}
\]
There exists:
1. If \( \bigoplus_{j=0}^{15} x_j = 0 \), then \( \text{HW}(y) = \text{HW}(x) \).
2. If \( \bigoplus_{j=0}^{15} x_j = 1 \), then \( \text{HW}(y) = 16 - \text{HW}(x) \).
3. If \( \text{HW}(x) > 0 \) and \( \bigoplus_{j=0}^{15} x_j = 0 \), then \( \text{HW}(y) \geq 2 \).
4. If \( \text{HW}(x) > 0 \) and \( \bigoplus_{j=0}^{15} x_j = 1 \), then \( \text{HW}(y) \geq 1 \).

**proof:**
There exists:
\[
\text{HW}(x) = \sum_{j=0}^{15} x_j \leq 16 \quad (B.1.1)
\]
1. If \( \bigoplus_{j=0}^{15} x_j = 0 \) Then
\[
y_i = x_i \bigoplus \left( \bigoplus_{j=0}^{15} x_j \right) = x_i \quad 0 \leq i \leq 15 \quad (B.1.2)
\]
Then
\[
\text{HW}(y) = \text{HW}(x)
\]
2. If \( \bigoplus_{j=0}^{15} x_j = 1 \) Then
\[
y_i = x_i \bigoplus \left( \bigoplus_{j=0}^{15} x_j \right) = x_i \oplus 1 = \neg x_i \quad 0 \leq i \leq 15
\]
Then
\[
\text{HW}(y) = \sum_{j=0}^{15} y_i = \sum_{j=0}^{15} (1 - x_j) = 16 - \sum_{j=0}^{15} (x_j) = 16 - \text{HW}(x) \quad (B.1.3)
\]
3. If \( \text{HW}(x) \geq 1 \) and \( \bigoplus_{j=0}^{15} x_j = 0 \), if \( \text{HW}(x) = 1 \), there has \( \bigoplus_{j=0}^{15} x_j = 1 \). So:
\[
\text{HW}(x) \geq 2
\]
By (B.1.2), there exists: \( \text{HW}(y) = \text{HW}(x) \geq 2 \)

4. If \( \text{HW}(x) \geq 1 \) and \( \bigoplus_{j=0}^{15} x_j = 1 \), and if \( \text{HW}(x) = 16 \), there has \( \bigoplus_{j=0}^{15} x_j = 0 \), so there exists:
\[
\text{HW}(x) \leq 15
\]
By (B.1.3), there exists: \( \text{HW}(y) = 16 - \text{HW}(x) \geq 16 - 15 = 1 \)

**Proposition B.2:** In message words of DDHA, if there exists \( 0 \leq j_1 \leq 15 \) make \( \bigoplus_{i=0}^{15} m_{i,j_1} = 1 \), Then there exist \( \text{HW}(em) \geq 16 \).

**Proof:**
There has:
\[
em_i = m_i \quad 0 \leq i \leq 15
\]
\[
\bigoplus_{i=0}^{15} em_{i,j_1} = \bigoplus_{i=0}^{15} m_{i,j_1} = 1
\]
Suppose \( I = \{ i \mid m_{i,j_1} = 1 \} \) and \( \text{HW}((em_{0,j_1}, ..., em_{15,j_1})) = H0 \)
There has: \( m_1 = e_{m_1} \oplus (\bigoplus_{j=0}^{15} e_{m_{1,j}}) \)

By proposition B.1, thus

\[
HW((m_{1,0,j}, ..., m_{1,15,j})) = 16 - H0
\]

\( m_{1,j} = 1 \quad i \notin I \)

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

Because

Then

\[
m_{2,r,j} = m_{1,r,(16 - i + j) \mod 16}
\]

\[
m_{2,r,j} = m_{1,r,(16 - i + j) \mod 16} = m_{1,r,16 - i + j} \mod 16
\]

\[
= m_{1,16 - i + j} \mod 16
\]

\[
= 1
\]

There seixst: \( e_{m_{16}} \leftarrow (\bigoplus_{j=0}^{15} m_{2,j}) \oplus m_{2_i} \quad 0 \leq i \leq 15 \)

By proposition B.1, there exits:

\[
HW((e_{m_{16}}, ..., m_{231})) \geq 1
\]

\[
HW((e_{m_{16}}, ..., m_{231})) = (\sum_{j \in J} \sum_{i=16}^{31} e_{m_{i,j}}) + \sum_{j \in J} \sum_{i=16}^{31} e_{m_{i,j}}
\]

\[
\geq \sum_{j \in J} \sum_{i=16}^{31} e_{m_{i,j}}
\]

\[
\geq 16 - H0
\]

Then

\[
HW(e_m) = HW((e_{m_0}, ..., e_{m_{15}})) + HW((e_{m_{16}}, ..., e_{m_{31}}))
\]

\[
\geq H0 + (16 - H0)
\]

\[
= 16
\]

\[\square\]

**Proposition B.3:** In message words of DDHA, if \( HW(m) \geq 0 \) there exist \( HW(e_m) \geq 8 \).

**Proof:**

There exists:

\[
e_{m_1} = m_i \quad 0 \leq i \leq 15
\]

\[
HW((e_{m_0}, ..., e_{m_{15}})) = HW((m_0, ..., m_{15}))
\]

1. If there exists \( j \) make \( \bigoplus_{i=0}^{15} e_{m_{i,j}} = 1 \), by proposition B.2, there has
exists:

\[ HW(em) \geq 16 > 8 \] \hspace{1cm} (B.3.a)

2. if there exists

\[ \bigoplus_{i=0}^{15} em_{i,j} = 0 \hspace{0.5cm} 0 \leq j \leq 15 \]

Then there exists:

\[ m_{1,i,j} = em_{i,j} \oplus (\bigoplus_{i=0}^{15} em_{i,j}) = em_{i,j} \hspace{0.5cm} 0 \leq i, j \leq 15 \]

\[ m_{2,i,j} = m_{1,j,(16-i+j) \mod 16} \]

Let \( I_{0,j} = \{i \mid em_{i,j} = 1 \ 0 \leq i \leq 15 \ 0 \leq j \leq 15 \} \), \( I_{1,j} = \{i \mid m_{1,i,j} = 1 \ 0 \leq i \leq 15 \ 0 \leq j \leq 15 \} \) then:

\[ I_{1,j} = \{i \mid m_{1,i,j} = 1 \ 0 \leq i \leq 15 \ 0 \leq j \leq 15 \} = \{i \mid em_{i,j} = 1 \ 0 \leq i \leq 15 \ 0 \leq j \leq 15 \} = I_{0,j} \]

\[ HW(m_{1}) = HW((em_{0},...,em_{15})) \]

Let \( JB_{0} = \{j | (\sum_{i=0}^{15} I_{1,i,j}) > 0 \ 15 \geq j \geq 0 \} \), and because

\[ \bigoplus_{i=0}^{15} em_{i,j} = 0 \hspace{0.5cm} 0 \leq j \leq 15 \]

Then by proposition B.1:

\[ (\sum_{i \in I_{1,j}} m_{1,i,j}) = (\sum_{i \in I_{1,j}} em_{i,j}) \geq 2 \hspace{0.5cm} j \in JB_{0} \] \hspace{1cm} (B.3.1)

Then there has:

\[ m_{2,i,j} = m_{1,j,(16-i+j) \mod 16} \]

So there exist:

\[ HW ((m_{0,16},...,m_{2,15})) = HW ((m_{1,16},...,m_{1,15})) = HW ((em_{0},...,em_{15})) \hspace{1cm} (B.3.2) \]

2.1 If there exists \( i \in c \) make \( (\bigoplus_{i=0}^{15} m_{2,i,c}) = 1 \), by proposition B.1 and (B.3.2), there exists:

\[ HW ((em_{16},...,em_{31})) \geq HW ((em_{16,11c},...,em_{31,11c})) \]

\[ = 16 - HW ((m_{2,16,11c},...,m_{2,15,11c})) \]

\[ HW ((em_{0},...,em_{31})) = HW ((em_{0},...,em_{15})) + HW ((em_{16},...,em_{31})) \]

\[ \geq HW ((m_{2,0},...,m_{2,15})) + 16 - HW ((m_{2,0,11c},...,m_{2,15,11c})) \]

\[ \geq HW ((m_{2,0,11c},...,m_{2,15,11c})) + 16 - HW ((m_{2,0,11c},...,m_{2,15,11c})) \]

\[ = 16 > 8 \] \hspace{1cm} (B.3.b)

2.2 If \( (\bigoplus_{i=0}^{15} m_{2,i}) = 0 \). Then there exist:
\[ em_{i,j} = m^2_{i-16,j} \oplus (\bigoplus_{i=0}^{15} m^2_{i,j}) = m^2_{i-16,j} \quad 16 \leq i \leq 31 \]
\[ \text{HW}((em_{16},...,em_{31})) = \text{HW}(m^2) \quad (B.3.3) \]

Because \( \text{HW}(m) \geq 0 \), there are at least one member in \( JB_0 \). Let \( jb_0 = jb \), and by proposition B.1, there are at least two members in \( I_{1, jb_0} \).
Suppose \( ila \neq ilb \in I_{1, jb_0} \), then:
\[ m_{1, ila, jb_0} = 1 \quad \text{and} \quad m_{1, ilb, jb_0} = 1 \]

Then there has:
\[ m_{1, ila, jb_0} = m^2_{1, ila, (16 + jb_0 - ila) \mod 16} = 1 \]
\[ m_{1, ilb, jb_0} = m^2_{1, ilb, (16 + jb_0 - ilb) \mod 16} = 1 \]

Because there has: \( 0 \leq jb_0, ila, ilb \leq 15 \) and \( ila \neq ilb \)
There has: \( (16 + jb_0 - ila) \mod 16 \neq (16 + jb_0 - ilb) \mod 16 \)

Because \( (\bigoplus_{i=0}^{15} m^2_{i}) = 0 \), Then by proposition B.1:
\[ \left( \sum_{i=0}^{15} m^2_{i, (16 + jb_0 - ila) \mod 16} \right) \geq 2 \]
\[ \left( \sum_{i=0}^{15} m^2_{i, (16 + jb_0 - ilb) \mod 16} \right) \geq 2 \]
\[ \text{HW}(m) \geq \text{HW}((m^2_{0, (16 + jb_0 - ila) \mod 16},...,m^2_{15, (16 + jb_0 - ila) \mod 16}))) + \]
\[ \text{HW}((m^2_{0, (16 + jb_0 - ilb) \mod 16},...,m^2_{15, (16 + jb_0 - ilb) \mod 16}))) \]
\[ \geq 4 \quad (B.3.4) \]

By (B.3.2) (B.3.3) (B.3.4), there exist:
\[ \text{HW}((em_0,...,em_{31})) = \text{HW}((em_0,...,em_{15})) + \text{HW}((em_{16},...,em_{31})) \]
\[ = 2 \times \text{HW}(m^2) \]
\[ \geq 2 \times 4 = 8 \quad | \quad (B.3.c) \]

So by (B.3.a) (B.3.b) (B.3.c), if \( \text{HW}(m) \geq 0 \) there exist \( \text{HW}(em) \geq 8 \).

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