Attacks on Round-Reduced BLAKE

Li Ji and Xu Liangyu

Sony China Research Laboratory
{Ji.Li, Liangyu.Xu}@sony.com.cn

Abstract. BLAKE is a new hash family proposed for SHA-3. The core of compression function reuses the core function of ChaCha. A round-dependent permutation is used as message schedule. BLAKE is claimed to achieve full diffusion after 2 rounds. However, message words can be controlled on the first several founds. By exploiting properties of message permutation, we can attack 2.5 reduced rounds. The results do not threat the security claimed in the specification.

1 Description of BLAKE

The hash family of BLAKE [1] includes four instances BLAKE-28, BLAKE-32, BLAKE-48, BLAKE-64.

BLAKE-28 and BLAKE-32 operate on 32-bit words and output 224 bits and 256 bits digest. BLAKE-48 and BLAKE-64 operate 64-bit words and output 384 bits and 512 bits digest.

We give a short description of BLAKE-32 with the same notations in [1].

The compression function of BLAKE-32 takes four values as inputs:

- A previous chain value (8 words) $h^{t-1} = h_0^{t-1}, \cdots, h_7^{t-1}$
- A message block (16 words) $m = m_0, \cdots, m_{15}$
- A salt (4 words) $s = s_0, \cdots, s_3$
- A counter (2 words) $t = t_0, t_1$

The compression function is written as:

$$h^t = \mathbf{compress}(h^{t-1}, m, s, t)$$

A 16-word state $v_0, \cdots, v_{15}$ is initialized such that different inputs produce different initial states, which is represented as 4×4 matrix as follows:

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \leftarrow \begin{pmatrix} \mathrm{h}_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ s_0 \oplus c_0 & s_1 \oplus c_1 & s_2 \oplus c_2 & s_3 \oplus c_3 \\ t_0 \oplus c_4 & t_1 \oplus c_5 & t_0 \oplus c_6 & t_1 \oplus c_7 \end{pmatrix}$$

After initialized, the state v is transformed by a round function, which computes:

$$\begin{array}{llll} G_0(v_0, v_4, v_8, v_{12}) & G_1(v_1, v_5, v_9, v_{13}) & G_2(v_2, v_6, v_{10}, v_{14}) & G_3(v_3, v_7, v_{11}, v_{15}) \\ G_4(v_0, v_5, v_{10}, v_{11}) & G_5(v_1, v_6, v_{11}, v_{12}) & G_6(v_2, v_7, v_8, v_{13}) & G_7(v_3, v_4, v_9, v_{14}) \end{array}$$

Where, $G_i(a, b, c, d)$ is defined as

$$a \leftarrow a + b + (m_{\sigma_r(2i)} \oplus c_{\sigma_r(2i+1)}) \tag{1}$$
$$d \leftarrow (d \oplus a)^{\lll 16} \tag{2}$$
$$c \leftarrow c + d \tag{3}$$
$$b \leftarrow (b \oplus c)^{\lll 12} \tag{4}$$
$$a \leftarrow a + b + (m_{\sigma_r(2i+1)} \oplus c_{\sigma_r(2i)}) \tag{5}$$
$$d \leftarrow (d \oplus a)^{\lll 8} \tag{6}$$
$$c \leftarrow c + d \tag{7}$$
$$b \leftarrow (b \oplus c)^{\lll 7} \tag{8}$$

The same permutation $\sigma_r(j)$ $(0 \le j < 16)$ for message words and round constants refers to Table 1.

Table 1. Message and Constants Permutation

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
σ_0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
σ_1	14	10	4	8	9	15	13	6	1	12	0	2	11	7	5	3
σ_2	11	8	12	0	5	2	15	13	10	14	3	6	7	1	9	4
σ_3	7	9	3	1	13	12	11	14	2	6	5	10	4	0	15	8
σ_4	9	0	5	7	2	4	10	15	14	1	11	12	6	8	3	13
σ_5	2	12	6	10	0	11	8	3	4	13	7	5	15	14	1	9
σ_6	12	5	1	15	14	13	4	10	0	7	6	3	9	2	8	11
σ_7	13	11	7	14	12	1	3	9	5	0	15	4	8	6	2	10
σ_8	6	15	14	9	11	3	0	8	12	2	13	7	1	4	10	5
σ_9	10	2	8	4	7	6	1	5	15	11	9	14	3	12	13	0

BLAKE-32 is recommended to iterates 10 rounds.

After the rounds sequence, the new chain value is extracted with the new state, the salt and the feedforward of the initial chain value.

$$\begin{aligned}
h_0^t &\leftarrow h_0^{t-1} \oplus s_0 \oplus v_0 \oplus v_8 \\
h_1^t &\leftarrow h_1^{t-1} \oplus s_1 \oplus v_1 \oplus v_9 \\
h_2^t &\leftarrow h_2^{t-1} \oplus s_2 \oplus v_2 \oplus v_{10} \\
h_3^t &\leftarrow h_3^{t-1} \oplus s_3 \oplus v_3 \oplus v_{11} \\
h_4^t &\leftarrow h_4^{t-1} \oplus s_0 \oplus v_4 \oplus v_{12} \\
h_5^t &\leftarrow h_5^{t-1} \oplus s_1 \oplus v_5 \oplus v_{13} \\
h_6^t &\leftarrow h_6^{t-1} \oplus s_2 \oplus v_6 \oplus v_{14} \\
h_7^t &\leftarrow h_7^{t-1} \oplus s_3 \oplus v_7 \oplus v_{15}
\end{aligned}$$

2 Observations on the Message Permutation of BLAKE

The round function of BLAKE reuses the core function of ChaCha [2]. One BLAKE-32 round is equated to two ChaCha rounds in [1]. Each round of BLAKE is composed of eight calls to the G function. Each round of ChaCha requires four calls of G. One BLAKE-32 round is equated to two ChaCha rounds in [1]. We call four calls of G like ChaCha's one round as a half round of BLAKE.

Following, we show how to control some parts of the output hash value by exploiting the characteristics on the message permutation.

2.1 Message Modification on 1.5 rounds

On the first 1.5 rounds, we have enough freedom to control two words of the output hash value of the compression function.

Observation 1 *It is obvious that we can control one of the four output variables on the G function by modifying the value of* $m_{\sigma_r(2i+1)}$.

According to the order of message permutation, given random message words and fixed initial value as inputs, if we modify the value of m_9, we can control the value of state words v_0 after G_4 on the first one round. Similarly, if we modify the values of m_{11}, m_{13} and m_{15}, the values of state words v_{12}, v_8 and v_4 can be controlled after G_5, G_6 and G_7 on the first one round. Since all the inputs to G_0 on the first 1.5 rounds can be controlled, we always can control the output of G_0.

Table 2 lists these relations.

Table 2. Message Modification on the First 1.5 rounds

Init	v_0, v_4, v_8, v_{12}	v_1, v_5, v_9, v_{13}	v_2, v_6, v_{10}, v_{14}	v_3, v_7, v_{11}, v_{15}
R 0.5	$G_0(v_0, v_4, v_8, v_{12})$	$G_1(v_1, v_5, v_9, v_{13})$	$G_2(v_2, v_6, v_{10}, v_{14})$	$G_3(v_3, v_7, v_{11}, v_{15})$
R 1	$G_4(\hat{\mathbf{v_0}}, v_5, v_{10}, v_{11})$ $\hat{\mathbf{m_9}}$	$G_5(v_1, v_6, v_{11}, \hat{\mathbf{v_{12}}})$ $\hat{\mathbf{m_{11}}}$	$G_6(v_2, v_7, \hat{\mathbf{v_8}}, v_{13})$ $\hat{\mathbf{m_{13}}}$	$G_7(v_3, \hat{\mathbf{v_4}}, v_9, v_{14})$ $\hat{\mathbf{m_{15}}}$
R 1.5	$G_0(\hat{\mathbf{v_0}}, \hat{\mathbf{v_4}}, \hat{\mathbf{v_8}}, \hat{\mathbf{v_{12}}})$	$G_1(v_1, v_5, v_9, v_{13})$	$G_2(v_2, v_6, v_{10}, v_{14})$ $\hat{\mathbf{m_9}}$	$G_3(v_3, v_7, v_{11}, v_{15})$ $\hat{\mathbf{m_{13}}}$

That means we can control the values of h'_0 and h'_4 after the the first 1.5 rounds after the finalization.

$$h'_0 \leftarrow h_0^{t-1} \oplus s_0 \oplus \hat{v_0} \oplus \hat{v_8}$$
$$h'_4 \leftarrow h_4^{t-1} \oplus s_0 \oplus \hat{v_4} \oplus \hat{v_{12}}$$

2.2 Message Modification on 2 rounds

Observation 2 *When modifying the initial value* $h_i^{t-1}(i = 0, \cdots, 3)$*, we can keep these state words of v after the* G_i *function unaffected by modifying the message words* $m_{\sigma_0(2i)}$.

Table 3 shows how to control the hash value h'_0 by modifying the initial value word h_0 and message word m_0 together.

Table 3. Message Modification on the First 2 rounds

Init	$\hat{\mathbf{v}}_\mathbf{0}, v_4, v_8, v_{12}$ $\hat{\mathbf{h}}_\mathbf{0}^\mathbf{t-1}$	v_1, v_5, v_9, v_{13}	v_2, v_6, v_{10}, v_{14}	v_3, v_7, v_{11}, v_{15}
R 0.5	$G_0(v_0, v_4, v_8, v_{12})$ $\hat{\mathbf{m}}_\mathbf{0}$	$G_1(v_1, v_5, v_9, v_{13})$	$G_2(v_2, v_6, v_{10}, v_{14})$	$G_3(v_3, v_7, v_{11}, v_{15})$
R 1	$G_4(v_0, v_5, v_{10}, v_{11})$	$G_5(v_1, v_6, v_{11}, v_{12})$	$G_6(v_2, v_7, v_8, v_{13})$	$G_7(v_3, v_4, v_9, v_{14})$
R 1.5	$G_0(v_0, v_4, v_8, v_{12})$	$G_1(v_1, v_5, v_9, v_{13})$	$G_2(v_2, v_6, v_{10}, v_{14})$	$G_3(v_3, v_7, v_{11}, v_{15})$
R 2	$G_4(v_0, v_5, v_{10}, v_{11})$	$G_5(\hat{\mathbf{v}}_\mathbf{1}, \hat{\mathbf{v}}_\mathbf{6}, \hat{\mathbf{v}}_\mathbf{11}, \hat{\mathbf{v}}_\mathbf{12})$ $\hat{\mathbf{m}}_\mathbf{0}$	$G_6(v_2, v_7, v_8, v_{13})$	$G_7(v_3, v_4, v_9, v_{14})$

Because the values of v_0 and v_8 are unaffected, according to the finalization: $h'_0 \leftarrow \hat{h}_0^{t-1} \oplus s_0 \oplus v_0 \oplus v_8$, the output h'_0 is changed by the value of $\hat{h}_0^{t-1}$ directly. We can control the word of hash value h'_0 after 2 rounds.

2.3 Message Modification on 2.5 rounds

By using message word m_0 and m_2 together, we can control the output value of h'_0 after 2.5 rounds.

Firstly, we select the value of h_0^{t-1} with the same method in the section 2.2. Then, we can choose the value of m_2 to control the value of v_{12} unaffected on the second round according to the observation 2. The affection of changing m_2 on the first 0.5 round will be patched by modifying the value of h_1^{t-1}. Table 4 shows the details.

Table 4. Message Modification on the First 2.5 rounds

Init	$\hat{\mathbf{v}}_\mathbf{0}, v_4, v_8, v_{12}$ $\hat{\mathbf{h}}_\mathbf{0}^\mathbf{t-1}$	$\hat{\mathbf{v}}_\mathbf{1}, v_5, v_9, v_{13}$ $\hat{\mathbf{h}}_1^{t-1}$	v_2, v_6, v_{10}, v_{14}	v_3, v_7, v_{11}, v_{15}
R 0.5	$G_0(v_0, v_4, v_8, v_{12})$ $\hat{\mathbf{m}}_\mathbf{0}$	$G_1(v_1, v_5, v_9, v_{13})$ $\hat{\mathbf{m}}_\mathbf{2}$	$G_2(v_2, v_6, v_{10}, v_{14})$	$G_3(v_3, v_7, v_{11}, v_{15})$
R 1	$G_4(v_0, v_5, v_{10}, v_{11})$	$G_5(v_1, v_6, v_{11}, v_{12})$	$G_6(v_2, v_7, v_8, v_{13})$	$G_7(v_3, v_4, v_9, v_{14})$
R 1.5	$G_0(v_0, v_4, v_8, v_{12})$	$G_1(v_1, v_5, v_9, v_{13})$	$G_2(v_2, v_6, v_{10}, v_{14})$	$G_3(v_3, v_7, v_{11}, v_{15})$
R 2	$G_4(v_0, v_5, v_{10}, v_{11})$	$G_5(\hat{\mathbf{v}}_\mathbf{1}, \hat{\mathbf{v}}_\mathbf{6}, \hat{\mathbf{v}}_\mathbf{11}, v_{12})$ $\hat{\mathbf{m}}_\mathbf{0}$ $\hat{\mathbf{m}}_\mathbf{2}$	$G_6(v_2, v_7, v_8, v_{13})$	$G_7(v_3, v_4, v_9, v_{14})$
R 2.5	$G_0(v_0, v_4, v_8, v_{12})$	$G_1(\hat{\mathbf{v}}_\mathbf{1}, \hat{\mathbf{v}}_\mathbf{5}, \hat{\mathbf{v}}_\mathbf{9}, \hat{\mathbf{v}}_\mathbf{13})$ $\hat{\mathbf{m}}_\mathbf{0}$	$G_2(\hat{\mathbf{v}}_\mathbf{2}, \hat{\mathbf{v}}_\mathbf{6}, \hat{\mathbf{v}}_\mathbf{10}, \hat{\mathbf{v}}_\mathbf{14})$ $\hat{\mathbf{m}}_\mathbf{2}$	$G_3(v_3, v_7, v_{11}, v_{15})$

As the values of v_0, v_4, v_8 and v_{12} are unaffected after the second round, the output of G_0 on the 2.5 round will also keep unaffected. According to the finalization: $h'_0 \leftarrow \hat{h}_0^{t-1} \oplus s_0 \oplus v_0 \oplus v_8$, the word of h'_0 is changed by the value of $\hat{h}_0^{t-1}$ directly. We can control the word of hash value h'_0 after 2.5 rounds.

Since we can control some words of the output hash value, the round reduced variants of BLAKE can be attacked.

3 Attacks of BLAKE

3.1 Attacks on 1.5 rounds of BLAKE

Firstly, we explain how to do preimage attack and 2nd preimage attack based on the observation in section 2.1.

Given fixed initial value $h^{t-1} = h_0^{t-1}, \cdots, h_7^{t-1}$, hash value $h^t = h_0^t, \cdots, h_7^t$ and relevant salt s and counter t, we can do following calculations:

1. Set message words $m = m_0, \cdots, m_{15}$ randomly.
2. Set $v_0^{(1.5)}$ randomly (where $v_i^{(r)}$ denotes the i-th word value of state v on r-th round), calculate $v_8^{(1.5)}$ according to $v_0^{(1.5)} \oplus v_8^{(1.5)} = h_0^t \oplus h_0^{t-1} \oplus s_0$.
3. Set $v_4^{(1.5)}$ randomly , calculate $v_{12}^{(1.5)}$ according to $v_4^{1.5} \oplus v_{12}^{(1.5)} = h_4^t \oplus h_4^{t-1} \oplus s_0$.
4. Calculate the reverse function of G_0 and get
$$((v_0^{(1)}, v_4^{(1)}, v_8^{(1)}, v_{12}^{(1)})) = G_0^{-1}(v_0^{(1.5)}, v_4^{(1.5)}, v_8^{(1.5)}, v_{12}^{(1.5)}).$$
5. Calculate $v'^{(1)} = G_{0,\cdots,7}(h^{t-1}, m, s, t)$ and record all immediate values of state v in each step of $G_{4,\cdots,7}$.
6. Modify m_9, m_{11}, m_{13} and m_{15}. The right m_9 can be calculated from: $\hat{m_9} = (a+b) \oplus (v_0'^{(1)} \oplus v_0^{(1)}) \oplus c_8$, where a and b are relevant immediate values of G_4 on the state $v^{(1)}$. Similarly, the right $\hat{m_{11}}$, $\hat{m_{13}}$ and $\hat{m_{15}}$ can be calculated according to immediate values of G_5, G_6 and G_7. Then the hash value h' will be changed into $h_0' = h_0^t$ and $h_4' = h_4^t$.
7. Try above steps, until other 6 words hash value equate to the given value on h^t.

If considering of the padding scheme of BLAKE, m_{15} will be determined by the length of message. We can modify message words m_1, m_3, m_5 and m_7 on the 0.5 round instead. Then we can control v_4 by controlling the inputs of G_7. h_0' and h_4' also can be controlled.

As a result, we get the preimage of given hash value. The complexity of the preimage attack need the time complexity of $2^{6\times 32} = 2^{192}$ for BLAKE-32. The attack need a fixed size memory to store the immediate values of state, which can be ignored. We can construct 2nd preimage attack by the similar method and the same complexity.

Considering of collision attack, we can fix the two words of hash value h_0' and h_4' in advance. For the same initial values h^{t-1}, we can find messages m to output the same values on h_0' and h_4'. Then we can use memoryless collision searching to find collision on other 6 words of hash value. The collision attack requires the time complexity of $2^{3\times 32} = 2^{96}$ for BLAKE-32 and trivial memory.

3.2 Attacks on 2 rounds of BLAKE

Firstly, we explain the free-start attacks on 2 rounds of BLAKE, then we extend the free-start attacks to the situation with given initial values.

Given hash value $h^t = h_0^t, \cdots, h_7^t$, initial value h^{t-1}, relevant salt s and counter t, to find free-start preimage, we do following steps :

1. Set message words $m = m_0, \cdots, m_{15}$ randomly. (If consider the padding scheme, m_{13}, m_{14} and m_{15} should be set according to the padding scheme.)
2. Calculate the hash value of 2 rounds: $h' = \mathbf{compress}_{2\times R}(h^{t-1}, m, s, t)$.
3. Set the value of $\hat{h}_0^{t-1}$ according to $\hat{h}_0^{t-1} = h_0^{t-1} \oplus h'_0 \oplus h_0^t$.
4. Calculate the value of $\hat{m_0}$ by reversing G_0 on the first 0.5 round:

$$\begin{aligned} h_0^{t-1} + h_4^{t-1} + (m_0 \oplus c_1) &= \hat{h_0^{t-1}} + h_4^{t-1} + (\hat{m_0} \oplus c_1) \\ \Rightarrow \hat{m_0} &= (h_0^{t-1} + (m_0 \oplus c_1) - \hat{h_0^{t-1}}) \oplus c_1. \end{aligned}$$

5. Modify m_0 to $\hat{m_0}$ and calculate $h' = \mathbf{compress}_{2\times R}(h, \hat{m}, s, t)$, then we can get $h'_0 = h_0^t$ again.
6. Try above steps, until left 7 words hash value equate to the given value on h^t.

As a result, we find the free-start preimage of given hash value. The free-start preimage attack requires the time complexity $2^{7\times 32} = 2^{224}$ for reduced 2 rounds BLAKE-32 and trivial memory. Similarly, we can do free-start 2nd preimage attack and free-start collision attack on reduced 2 rounds of BLAKE-32.

Considering of the version of BLAKE without salt, the free-start (2nd) preimage attack can be extended to (2nd) preimage attack by the method in [3]. The attacks require the time complexity of 2^{241} and trivial memory.

3.3 Attacks on 2.5 rounds of BLAKE

On 2.5 rounds of BLAKE, we modify message words m_0, m_2 and initial value words h_0^{t-1} and h_2^{t-1}. Given hash value $h^t = h_0^t, \cdots, h_7^t$ and relevant salt s and counter t, do following calculations:

1. Set message words $m = m_0, \cdots, m_{15}$ randomly. (If consider the padding scheme, m_{13}, m_{14} and m_{15} should be set according to the padding scheme.)
2. Calculate the hash value of 2.5 rounds: $h' = \mathbf{compress}_{2.5\times R}(h^{t-1}, m, s, t)$ and record the immediate value of $v_{12}^{(2)}$ after G_5 on the second round.
3. Set the value of $\hat{h}_0^{t-1}$ to $\hat{h^{t-1}}_0 = h_0^{t-1} \oplus h'_0 \oplus h_0^t$.
4. Calculate the value of $\hat{m_0}$ by reversing G_0 on the first 0.5 round: $\hat{m_0} = (h_0^{t-1} + (m_0 \oplus c_1) - \hat{h}_0^{t-1}) \oplus c_1$.
5. Modify m_0 to the values of $\hat{m_0}$ (h_0^{t-1} is set to the value of $\hat{h}_0^{t-1}$), calculate G_5 on the second round again and get new value of v'_{12}.

6. Modify m_2 to $\hat{m_2}$ according to:

$$\begin{aligned} v'_{12} &= a' + b' + (m_2 \oplus c_3) \\ v_{12} &= a' + b' + (\hat{m_2} \oplus c_3) \\ \Rightarrow \hat{m_2} &= (v_{12} - v'_{12} + (m_2 \oplus c_3)) \oplus c_3 \end{aligned}$$

7. Modify the value of h_2^{t-1} to $\hat{h}_2^{t-1} = h_2^{t-1} + (m_2 \oplus c_3) - (\hat{m_2} \oplus c_3)$.
8. Try above steps, until left 7 words of h' equate to the given value on h^t.

As a result, we get the free-start preimage of given hash value. The attacks also can be extended to free-start 2nd preimage and free-start collision attack with the same complexity in the section 3.2.

3.4 Complexities of Attacks

These attacks also can be used to attack other versions of BLAKE. Table 5 lists the complexities of these attacks. Appendix A explains how to use similar method to attack one of toy versions of BLAKE [4].

Table 5. Attacks complexities on reduced round of BLAKE

Version	Rounds	free-start collision	free-start (2nd) preimage	collision	(2nd) preimage
BLAKE-28	1.5 rounds	2^{80}	2^{160}	2^{80}	2^{160}
	2 rounds	2^{96}	2^{192}	-	2^{209}
	2.5 rounds	2^{96}	2^{192}	-	2^{209}
BLAKE-32	1.5 rounds	2^{96}	2^{192}	2^{96}	2^{192}
	2 rounds	2^{112}	2^{224}	-	2^{241}
	2.5 rounds	2^{112}	2^{224}	-	2^{241}
BLAKE-48	1.5 rounds	2^{128}	2^{256}	2^{128}	2^{256}
	2 rounds	2^{160}	2^{320}	-	2^{355}
	2.5 rounds	2^{160}	2^{320}	-	2^{355}
BLAKE-64	1.5 rounds	2^{192}	2^{384}	2^{192}	2^{384}
	2 rounds	2^{224}	2^{448}	-	2^{481}
	2.5 rounds	2^{224}	2^{448}	-	2^{481}

4 Conclusion

Message permutation is a light weight message schedule and easy to implement. It has been used to design many compression functions for iterated hash.

In this paper, we presented attacks on round-reduced BLAKE by message modification techniques. We analyzed the properties of round function and message permutation. Relevant attacks on reduced rounds were proposed. These attacks show BLAKE has no enough diffusion in 2.5 rounds.

References

1. Aumasson, J.P., Henzen, L., Meier, W., Phan, R.C.W.: Sha-3 proposal blake. Submission to NIST (2008) `http://131002.net/blake/blake.pdf`.
2. Bernstein, D.J.: Chacha, a variant of salsa20 (2008) `http://cr.yp.to/chacha.html`.
3. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)
4. Aumasson, J.P., Henzen, L., Meier, W., Phan, R.C.W.: Toy versions of blake (2009) `http://131002.net/blake/toyblake.pdf`.

A Attacks on Round-Reduced FLAKE

FLAKE [4] is one the toy versions of the hash family of BLAKE. The compression function of FLAKE makes no Feedforward, so the finalization of FLAKE-32 is just:

$$\begin{aligned} h_0^t &\leftarrow v_0 \oplus v_8 \\ h_1^t &\leftarrow v_1 \oplus v_9 \\ h_2^t &\leftarrow v_2 \oplus v_{10} \\ h_3^t &\leftarrow v_3 \oplus v_{11} \\ h_4^t &\leftarrow v_4 \oplus v_{12} \\ h_5^t &\leftarrow v_5 \oplus v_{13} \\ h_6^t &\leftarrow v_6 \oplus v_{14} \\ h_7^t &\leftarrow v_7 \oplus v_{15} \end{aligned}$$

Observation 3 *For FLAKE, given initial value $h^{t-1} = h_0^{t-1}, \cdots, h_7^{t-1}$ and message $m = m_0, \cdots, m_{15}$, We can choose $2^{32} - 1$ new values of m_0 and modify m_2, h_0^{t-1} and h_2^{t-1} to keep the output word of hash value h_0^t unchanged after 2.5 rounds.*

The observation is obvious according to the finalization of FLAKE-32 and the previous 2.5 rounds message modification.

Following we show how to construct free-start preimage attack on 2.5 reduced rounds by the observation. Other attacks can be constructed with similar method.

Given hash value $h^t = h_0^t, \cdots, h_7^t$, we do calculations as follows:

1. Set h^{t-1} and message $m = m_0, \cdots, m_{15}$ randomly, try to find one (h^{t-1}, m) to make the output word $h_0' = v_0 \oplus v_8 = h_0^t$. That needs to try about 2^{32} times.
2. Searching $2^{32} - 1$ values of m_0 and modify m_2, h_0^{t-1} and h_2^{t-1}, then we find $2^{32} - 1$ new values of (h^{t-1}, m) to output the $h_0' = h_0^t$. That means we can find one (h^{t-1}, m) to get one word hash value h_0^t with the cost of $O(1)$.
3. For left 7 words of hash value, repeat above steps 2^{196} times to find 2^{224} values of (h^{t-1}, m). We can expect to find one value of (h^{t-1}, m) to output hash value h^t.

The free-start preimage attack requires 2^{224} calculations and trivial memory.

We also can try to find inputs to make h'_0, h'_3, h'_4 and h'_7 equate to given hash value words in the first step, then do message modification and find enough inputs to get preimage. That requires the same complexity.

Similarly, we can free-start 2nd preimage attack, free-start collision attack. Then free-start preimage and 2nd preimage attack can be extended to preimage and 2nd preimage attack.

The attack results depend on the message permutation. For other toy versions of BLAKE, the attacks will fail if the message permutation is changed into identity.