Preimage Attack on ARIRANG

Deukjo Hong, Woo-Hwan Kim, Bonwook Koo

The Attached Institute of ETRI, P.O.Box 1, Yuseong, Daejeon, 305-600, Korea
{hongdj,whkim5,bwkoo}@ensec.re.kr

Abstract. The hash function ARIRANG is one of the 1st round SHA-3 candidates. In this paper, we present preimage attacks on ARIRANG with step-reduced compression functions. We consider two step-reduced variants of the compression function. First one uses the same feedforward$_1$ as the original algorithm, and the other one has the feedforward$_1$ working at the output of the half steps. Our attack finds a preimage of the 33-step OFF(Original FeedForward$_1$)-variants of ARIRANG-256 and ARIRANG-512 from Step 1 to Step 33, and a preimage of the 31-step MFF(Middle FeedForward$_1$)-variants of ARIRANG-256 and ARIRANG-512 from Step 3 to Step 33.

Keywords: SHA-3 candidate, Preimage Attack, Hash Function

1 Introduction

The hash function ARIRANG is the one of the 1st round SHA-3 candidates [1]. It uses a MD-like domain extender with counters. ARIANG has versions with four different output lengths — ARIRANG-224, ARIRANG-256, ARIRANG-384, and ARIRANG-512. The output of ARIRANG-224 is just a 32-bit truncation of the output of ARIRANG-256, and the output of ARIRANG-384 is just a 128-bit truncation of the output of ARIRANG-512. Each compression function consists of 40 steps.

In this paper, we present preimage attacks on ARIRANG with step-reduced compression functions. We consider two step-reduced variants of the compression function. First one uses the same feedforward$_1$ as the original algorithm, and the other one has the feedforward$_1$ working at the output of the half steps. We call the first one, the OFF(Original FeedForward$_1$)-variant, and the other one, the MFF(Middle FeedForward$_1$)-variant. Our attacks begin with the observation that we can move message words up to 4 steps. Together with this word-moving property, we found the best selection of the neutral words W_7 and W_9 by examining all possible pairs of message words.

We follows the framework of Sasaki and Aoki's preimage attack [2–4]. Our attack finds a preimage of the 33-step OFF(Original FeedForward$_1$)-variant from Step 1 to Step 33, and a preimage of the 31-step MFF(Middle FeedForward$_1$)-variant from Step 3 to Step 33. All the attacks on ARIRANG-256 cost about 2^{241} computations of reduced compression functions. All the attacks on ARIRANG-512 cost about 2^{481} computations of reduced compression functions.

2 Hash Function ARIRANG

We describe the specification of ARIRANG briefly. It uses a MD-like domain extender with counters (Fig. 1). Each compression functions of ARIRANG-256 and ARIRANG-512 get a 512-bit block and 1024-bit block of messages, respectively. The lengths of the chaining variables of ARIRANG-256 and ARIRANG-512 are 256 bits and 512 bits, respectively.

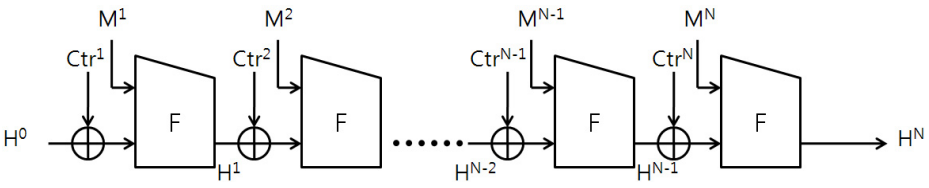

Fig. 1. The hashing structure of ARIRANG

For the ARIRANG-256 compression function, the message schedule algorithm produces 32 32-bit message words $W_0, ..., W_{31}$ from a 512-bit message block M and arrange two message words $W_{\sigma(2j)}$ and $W_{\sigma(2j+1)}$ for j-th step $(j = 0, ..., 39)$ according to the index function σ defined as the Table 1.

Table 1. The input-output table of the index function σ

j	$\sigma(2j)$	$\sigma(2j+1)$	$\sigma(2j+20)$	$\sigma(2j+21)$	$\sigma(2j+40)$	$\sigma(2j+41)$	$\sigma(2j+60)$	$\sigma(2j+61)$
0	16	17	20	21	24	25	28	29
1	0	1	3	6	12	5	7	2
2	2	3	9	12	14	7	13	8
3	4	5	15	2	0	9	3	14
4	6	7	5	8	2	11	9	4
5	18	19	22	23	26	27	30	31
6	8	9	11	14	4	13	15	10
7	10	11	1	4	6	15	5	0
8	12	13	7	10	8	1	11	6
9	14	15	13	0	10	3	1	12

The 32-bit message words $W_0, ..., W_{31}$ are produced in the following way, where $K_0, ..., K_{15}$ are 32-bit constants.

– The input message block M is divided into 16 words $W_0, ..., W_{15}$, i.e. $M = W_0 \| \cdots \| W_{15}$.

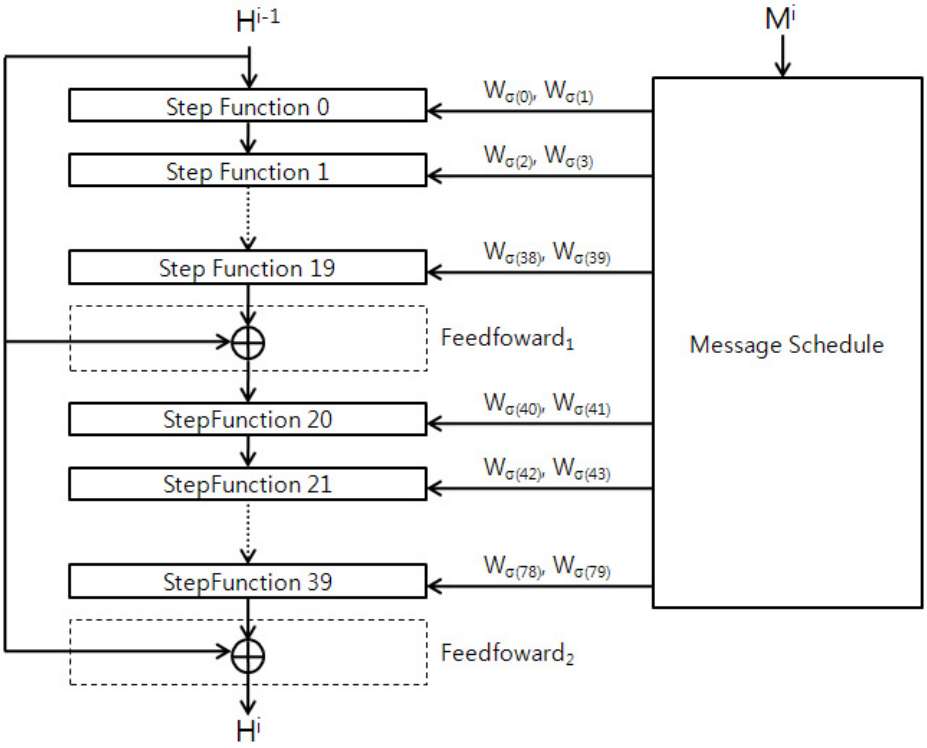

Fig. 2. The structure of the compression function of ARIRANG

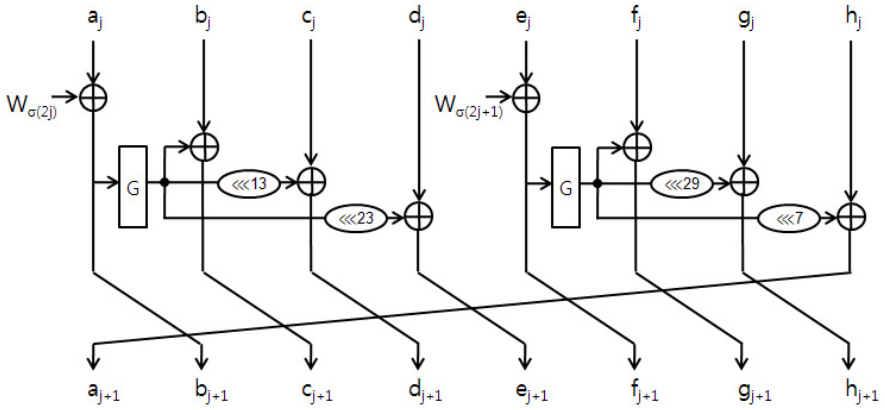

Fig. 3. The j-th step function of the compression function of ARIRANG

– The remaining 16 words are generated from $W_0, ..., W_{15}$ by

$$
\begin{aligned}
W_{16} &= (W_9 \oplus W_{11} \oplus W_{13} \oplus W_{15} \oplus K_0)^{\lll 5} \\
W_{17} &= (W_8 \oplus W_{10} \oplus W_{12} \oplus W_{14} \oplus K_1)^{\lll 11} \\
W_{18} &= (W_1 \oplus W_3 \oplus W_5 \oplus W_7 \oplus K_2)^{\lll 19} \\
W_{19} &= (W_0 \oplus W_2 \oplus W_4 \oplus W_6 \oplus K_3)^{\lll 31} \\
W_{20} &= (W_{14} \oplus W_4 \oplus W_{10} \oplus W_0 \oplus K_4)^{\lll 5} \\
W_{21} &= (W_{11} \oplus W_1 \oplus W_7 \oplus W_{13} \oplus K_5)^{\lll 11} \\
W_{22} &= (W_6 \oplus W_{12} \oplus W_2 \oplus W_8 \oplus K_6)^{\lll 19} \\
W_{23} &= (W_3 \oplus W_9 \oplus W_{15} \oplus W_5 \oplus K_7)^{\lll 31} \\
W_{24} &= (W_{13} \oplus W_{15} \oplus W_1 \oplus W_3 \oplus K_8)^{\lll 5} \\
W_{25} &= (W_4 \oplus W_6 \oplus W_8 \oplus W_{10} \oplus K_9)^{\lll 11} \\
W_{26} &= (W_5 \oplus W_7 \oplus W_9 \oplus W_{11} \oplus K_{10})^{\lll 19} \\
W_{27} &= (W_{12} \oplus W_{14} \oplus W_0 \oplus W_2 \oplus K_{11})^{\lll 31} \\
W_{28} &= (W_{10} \oplus W_0 \oplus W_6 \oplus W_{12} \oplus K_{12})^{\lll 5} \\
W_{29} &= (W_{15} \oplus W_5 \oplus W_{11} \oplus W_1 \oplus K_{13})^{\lll 11} \\
W_{30} &= (W_2 \oplus W_8 \oplus W_{14} \oplus W_4 \oplus K_{14})^{\lll 19} \\
W_{31} &= (W_7 \oplus W_{13} \oplus W_3 \oplus W_9 \oplus K_{15})^{\lll 31}
\end{aligned}
$$

The structures of the compression functions of ARIRANG-256 and ARIRANG-512 are same except that the word size of ARIRANG-256 is 32-bit but the word size of ARIRANG-512 is 64-bit. The structures of the compression function and the step function of ARIRANG are shown in Fig. 2 and Fig. 3. The function G is the composition of four parallel 8-bit S-boxes and one 4×4 MDS matrix for ARIRANG-256 like AES, and the composition of eight parallel 8-bit S-boxes and one 8×8 MDS matrix for ARIRANG-512.

3 Preimage Attack and Techniques

3.1 The Framework of Sasaki and Aoki's Preimage Attack

We follows the framework of Sasaki and Aoki's preimage attack [2–4]. First, we construct the pseudo-preimage attack procedure with the complexity of 2^x for target steps of the compression function. Then, we convert the pseudo-preimage attack to the preimage attack with the complexity of $2^{(n+x)/2+1}$, where n is the length of the hash value and $x < n$.

The pseudo-preimage attack on the target steps of the compression function is a kind of meet-in-the-middle attack. For meet-in-the-middle approach, we should divide the targeted steps of the compression functions into two independent chunks such that each chunk has at least one neutral message word which does not act on the other chunk at all. We call the chunk containing middle steps, the inner chunk, and the chunk containing the first and the final steps, the outer chunk.

Sasaki and Aoki developed the partial-matching, the partial-fixing, the local-collision techniques for improving the basic meet-in-the-middle attack. We checked that the partial-fixing and the local-collision techniques are not applicable to ARIRANG because of the step function and the G function. The partial-matching is possible for at most 6 steps.

3.2 Word-Moving Property

We found that each message word can be moved in the backward direction at most 4 steps. We call it the word-moving property (Fig. 4). With moving neutral words, we checked all possible pairs of neutral words (W_i, W_j) for $0 \leq i < j \leq 15$, and found that the pair (W_7, W_9) yields the best pair of chunks.

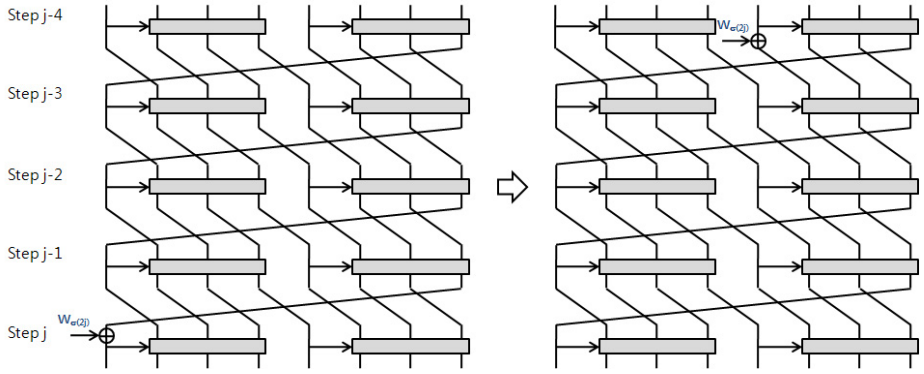

Fig. 4. The move of the message word $W_{\sigma(2j)}$ from the step j to the step $j-4$. The move of $W_{\sigma(2j+1)}$ is similar.

3.3 Feedforward$_1$ and Meet-in-the-Middle Approach

As the designers of ARIRANG claimed, the feedforward$_1$ obstructs the meet-in-the-middle approach. We summarize how the feedforward$_1$ can affect on the attack as follows.

1. If the feedforward$_1$ is contained in the matching-check steps, then the attack is possible.
2. If the feedforward$_1$ is contained in the lower steps (with the final step) of the outer chunk and the lower boarder line between the inner chunk and the outer chunk is contained in the matching-check steps, then the attack is possible.

3. If the feedforward$_1$ is contained in the upper steps (with the beginning step) of the outer chunk and the upper boarder line between the inner chunk and the outer chunk is contained in the matching-check steps, then the attack is possible.
4. If the feedforward$_1$ is contained in the inner chunk, then the attack is impossible.
5. If the feedforward$_1$ is contained in the lower steps of the outer chunk but the upper boarder line between the inner chunk and the outer chunk is contained in the matching-check steps, then the attack is impossible.
6. It the feedforward$_1$ is contained in the upper steps of the outer chunk but the lower boarder line between the inner chunk and the outer chunk is contained in the matching-check steps, then the attack is impossible.

4 Preimage Attack on 33-step OFF-Variant

In this section, we describe the preimage attack on 33-step OFF-variant from the step 1 to the step 33 by using the message words W_7 and W_9 as neutral words. As you see in Fig. 5, we determine the inner chunk (from Step 6 to Step 17) and the outer chunk (from Step 1 to Step 6 and from 22 to Step 33) after moving W_7 and W_9.

Note that the definitions of some steps are naturally modified according to Fig. 5. Now we describe how to do the partial-matching in the matching-check steps. Assume that we are given the input of the step 18, $(a_{18}, ..., h_{18})$ and the output of the step 21, $(a_{22}, ..., h_{22})$. Then we perform the partial computation from $(a_{18}, ..., h_{18})$ avoiding W_7:

$$x_I = h_{18} \oplus G(e_{18} \oplus W_{10})^{\lll 7} \oplus W_9. \tag{1}$$

On the other hand, we perform the partial computation from $(a_{22}, ..., h_{22})$ avoiding W_9:

$$x_O = G(G(b_{22}) \oplus c_{22}) \oplus d_{22} \oplus b_1. \tag{2}$$

Finally, we check whether $x_I = x_O$ (See Fig. 6).

Our attack finds a 2-block preimage. We denote a given hash value H^2. For given a hash value H^2, the following probabilistic procedure finds a pseudo-preimage.

1. Randomly choose $b_7, c_7, d_7, e_7, e_6, f_6, g_6, h_6$ and Fix them. All the message words $W_0, ..., W_{15}$ except for W_7 and W_9 are also randomly selected and fixed.
2. For each possible candidate of W_9, obtain the output of Step 17, $(a_{18}, ..., h_{18})$ by computing the step functions forwardly for Step 6, ..., Step 17, and keep it together with the candidate of W_9 and x_I in a table.
3. For each possible candidate of W_7, obtain the inputs of Step 22, $(a_{22}, ..., h_{22})$ by computing the step functions backwardly from Step 6 to Step1 and from Step 33 to Step 22, and search the values $(a_{18}, ..., h_{18}, W_9, x_I)$ in the table such that $x_I = x_O$. Then, check the matching at other positions for the partially matched pairs $(a_{18}, ..., h_{18}, W_9, a_{22}, ..., h_{22}, W_7)$.

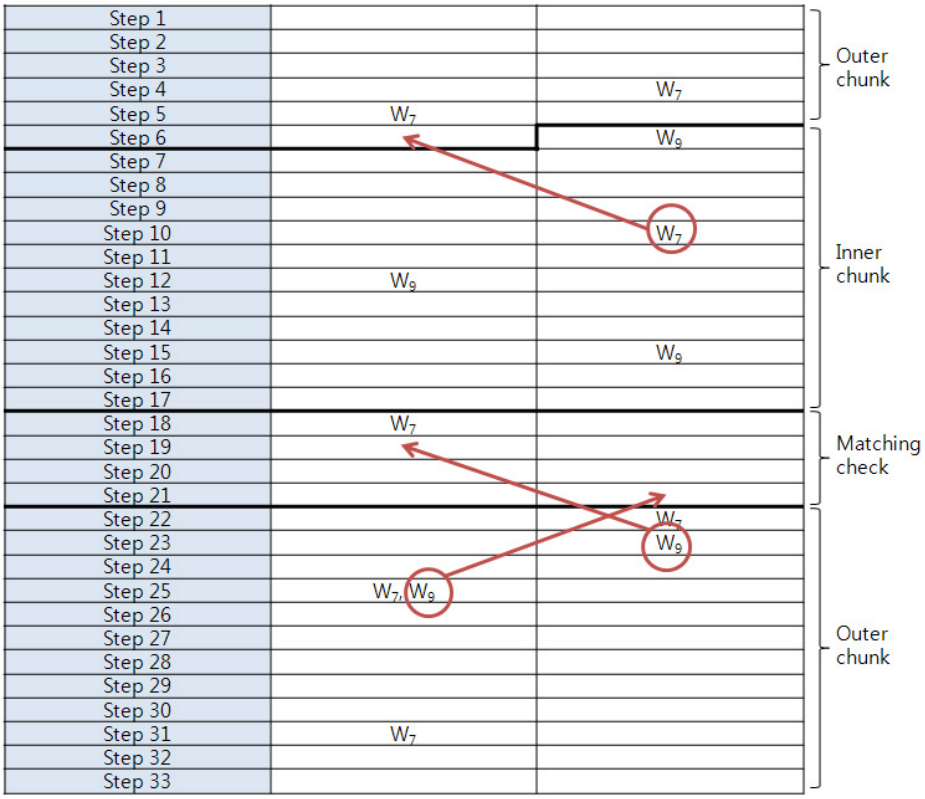

Fig. 5. The moves of the message words W_7 and W_9 in the 33 steps from the step 1 to the step 33, and the partition of the inner and outer chunks and the matching-check steps.

4. An input of Step 1 $(a_1, ..., h_1)$ corresponding to a totally matched pair is a pseudo-preimage of H^2.

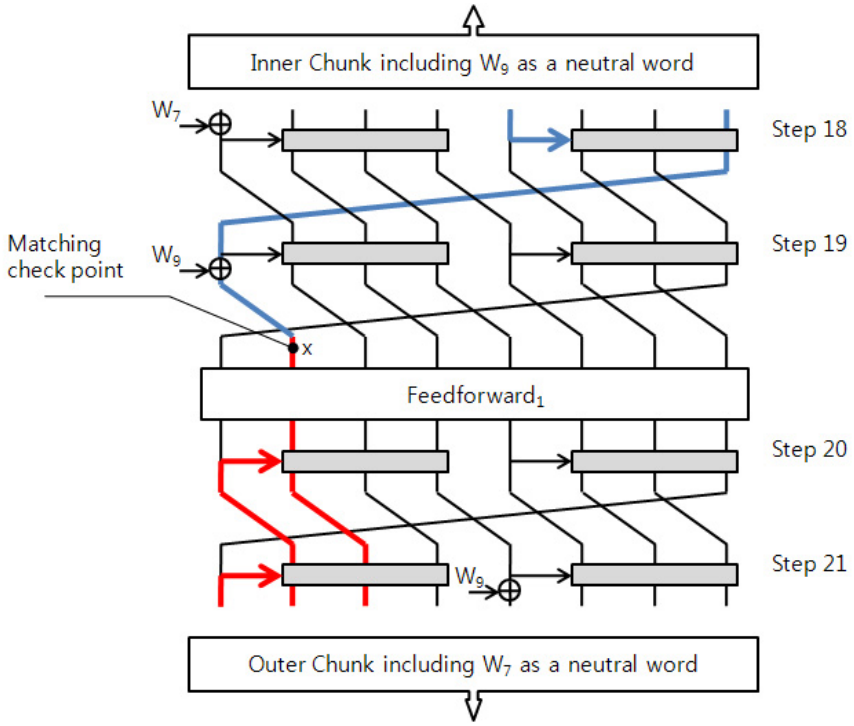

Fig. 6. Partial-matching between Step 18 and Step 21 in the attack on the 33-step OFF compression function from the step 1 to the step 33

For ARIRANG-256, the probability that the above procedure finds a pseudo-preimage of H^2 is $2^{32} \cdot 2^{32} \cdot 2^{-256} = 2^{-192}$. Therefore, if we repeat the procedure 2^{192} times, we expect one pseudo-preimage. The complexity of this pseudo-preimage finding algorithm is 2^{224} because the above procedure requires at most 2^{32} 33-step computations. According to Aoki and Sasaki, we can get a preimage with the complexity of 2^{241}. Similarly, our preimage attack on Step 1 to Step 33 for ARIRANG-512 has the complexity of 2^{481}.

5 Preimage Attack on MFF-Variants

Note that in the attack on the OFF-variant from Step 1 to Step 33, the feedforward$_1$ is located on the output of Step 19, which is contained in the matching-check steps. When we consider the MFF-variant from Step 1 to Step 33, the feedforward$_1$

is located on the output of Step 16 or Step 17, which is contained in the inner chunk, so the attack is impossible.

But, the attack on the 31-step MFF-variant from Step 3 to Step 33 is possible with the complexity of 2^{241} for ARIRANG-256 and the complexity of 2^{481} for ARIRANG-512 if the feedforward$_1$ is located on the output of Step 18. The attack on the 30-step MFF-variant from Step 4 to Step 33 is also possible for both ARIRANG-256 and ARIRANG-512.

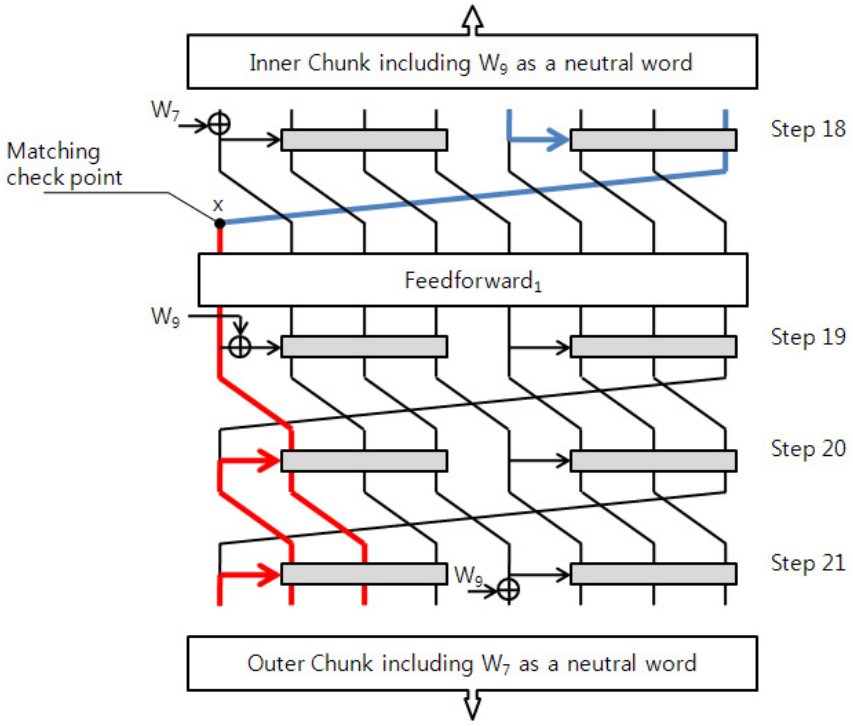

Fig. 7. Partial-matching between Step 18 and Step 21 in the attack on the 31-step OFF compression function from the step 3 to the step 33

6 Conclusion

In this paper, we presented a preimage attack on step-reduced variants of ARIRANG. Our attack finds a preimage of the 33-step OFF(Original FeedForward$_1$)-variants of ARIRANG-256 and ARIRANG-512 from Step 1 to Step 33, and a preimage of the 31-step MFF(Middle FeedForward$_1$)-variants of ARIRANG-256 and ARIRANG-512 from Step 3 to Step 33.

References

1. Donghoon Chang, Seokhie Hong, Changheon Kang, Jinkeon Kang, Jongsung Kim, Changhoon Lee, Jesang Lee, Jongtae Lee, Sangjin Lee, Yuseop Lee, Jongin Lim, Jaechul Sung, "ARIRANG: SHA-3 Proposal", available at `http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/`
2. Yu Sasaki, Kazumaro Aoki, "Preimage Attacks on Step-Reduced MD5", ACISP 2008, Springer-Verlag, LNCS 5107, pp. 282-296.
3. Yu Sasaki, Kazumaro Aoki, "Preimage Attacks on 3, 4, and 5-Pass HAVAL", ASIACRYPT 2008, Springer-Verlag, LNCS 5350, pp. 253-271.
4. Yu Sasaki, Kazumaro Aoki, "A Preimage Attack for 52-Step HAS-160", ICISC 2008, Springer-Verlag, LNCS 5461, pp. 302-317.