Low-weight Pseudo Collision Attack on Shabal and Preimage Attack on Reduced Shabal-512

Takanori Isobe and Taizo Shirai

Sony Corporation
1-7-1 Konan, Minato-ku, Tokyo 108-0075, Japan
{Takanori.Isobe,Taizo.Shirai}@jp.sony.com

Abstract. This paper studies two types of attacks on the hash function Shabal. The first attack is a low-weight pseudo collision attack on Shabal. Since a pseudo collision attack is trivial for Shabal, we focus on a low-weight pseudo collision attack. It means that only low-weight difference in a chaining value is considered. By analyzing the difference propagation in the underlying permutation, we can construct a low-weight (45-bits) pseudo collision attack on the full compression function with complexity of 2^{84}. The second attack is a preimage attack on variants of Shabal-512. We utilize a guess-and-determine technique, which is originally developed for a cryptanalysis of stream ciphers, and customize the technique for a preimage attack on Shabal-512. As a result, for the weakened variant of Shabal-512 using security parameters $(p, r) = (2, 12)$, a preimage can be found with complexity of 2^{497} and memory of 2^{400}. Moreover, for the Shabal-512 using security parameters $(p, r) = (1.5, 8)$, a preimage can be found with complexity of 2^{497} and memory of 2^{272}. To the best of our knowledge, these are best preimage attacks on Shabal variants and the second result is a first preimage attack on Shabal-512 with reduced security parameters.

Key words: Shabal, low-weight pseudo collision attack, preimage attack, guess-and-determine technique, SHA-3 competition

1 Introduction

Cryptographic hash functions play important roles in modern cryptology. Many cryptographic protocols require a secure hash function which holds several security properties classically including a preimage resistance, a second preimage resistance and a collision resistance. Due to the recent attacks on commonly used hash function (*e.g.* MD5, SHA-1) [13][14], NIST announced the SHA-3 competition to decide an advanced hash function standard [12]. In November 2008, NIST received 64 submissions, and 51 candidates proceed in the first round. In July 2009, NIST announced the 14 second round candidates.

Shabal, which is designed by E. Bresson et al. [5], is one of the second round candidates of the SHA-3 competition. Shabal employs an original mode of operation using a keyed permutation. The keyed permutation consist of an initial loop, a main loop and a final loop, and is parameterized by two security parameters (p, r):

Parameter p: The number of the main loop performed within one application of the keyed permutation. Large values of p provide better security guarantees.

Parameter r: An integer used to determine the size of one of the internal state. The minimal value is 2.

A recommended set of the parameters by the designers of Shabal is $(p, r) = (3, 12)$, which is used for the version of Shabal submitted to the SHA-3 hash competition. Shabal supports 192, 224, 256, 384 and 512-bit digest.

As for cryptanalysis of Shabal, several related-key distinguishers of the keyed permutation are reported so far [1][2][3][6][10]. These analyses focus on a building block of the Shabal's

Table 1. Summary of Our Results

Type of attack	Security parameters	Complexity	Memory	Reference
Low-weight pseudo collision on the compression function of Shabal (**difference: 45 bits** (out of 1408 bits))	(3, 12)	2^{84}	-	This paper
Preimage attack on reduced Shabal-512	(1, 12) without the final loop	2^{448}	not given	[5].
	(2, 12) without the final loop	2^{497}	2^{400}	This paper
	(1.5, 8)	2^{497}	2^{272}	This paper

compression function, and do not lead to an attack of Shabal hash function. Cryptanalysis of reduced Shabal has been reported only in a designer's self evaluation so far.

In this paper, we show new results on two kinds of attacks on Shabal hash function, they are a low-weight pseudo collision attack and a preimage attack.

A low-weight pseudo collision attack is a variant of a pseudo collision attack. In a normal setting of the pseudo collision attack, an attacker inserts a difference not only into message blocks but also into a chaining value. Finding a pseudo collision is trivial for Shabal, because the compression function of Shabal is invertible.

In this paper, we focus on a low-weight pseudo collision which considers only a low-weight difference in a chaining value. In order to construct a low-weight pseudo collision on Shabal, we use the linearization technique to analyze behaviors of differences and search an differential path effectively. As a result, we can construct a low-weight (45-bits) pseudo collision attack on the full compression function of Shabal with complexity of 2^{84}. In other word, the input chaining value has 45-bit difference and other 1363 $(= 512 + 512 + 384 - 45)$ bits do not have difference. Since a low-weight pseudo collision attack can also be seen as a near collision attack with respect to the inverse of the compression function, we consider that it reveals potential of the diffusion property of the Shabal. Although our result does not threat the claimed security of Shabal due to that this is a pseudo collision setting, this shows that the differential property of the compression function of Shabal is not ideal. The result of the attack is summarized in Table 1.

The other result is a **preimage attack** on Shabal's variants. A preimage attack on the weakened variant, Shabal-512 without the final loop, have been analyzed by designers [5]. The weakened variant using security parameters $(p, r) = (1, 12)$ does not hold preimage resistance. The preimage attack utilizes the weakness of the compression function that a part of the output of the inverse of the compression function can be controlled by an input message block. Since the same attack is not successful on the weakened variant using $(p, r) = (2, 12)$, the current record of the preimage attack is only the weakened variant using $(p, r) = (1, 12)$.

In this paper, we propose a novel approach using a guess-and-determine technique for controlling a part of the output of the inverse of the compression function. The technique enable a preimage attack on the weakened variant using $(p, r) = (2, 12)$. Moreover, we show an attack on the Shabal-512 using $(p, r) = (1.5, 8)$ including the final loop. This result is a first preimage attack on the Shabal-512 with reduced security parameters. The results of the attack are summarized in Table 1.

This paper is organized as follows. Brief descriptions of Shabal are given in Section 2. A low-weight pseudo collision attack on Shabal is shown in Section 3. The preimage attacks on two variant of Shabal-512 are shown in Section 4. Section 5 concludes this paper.

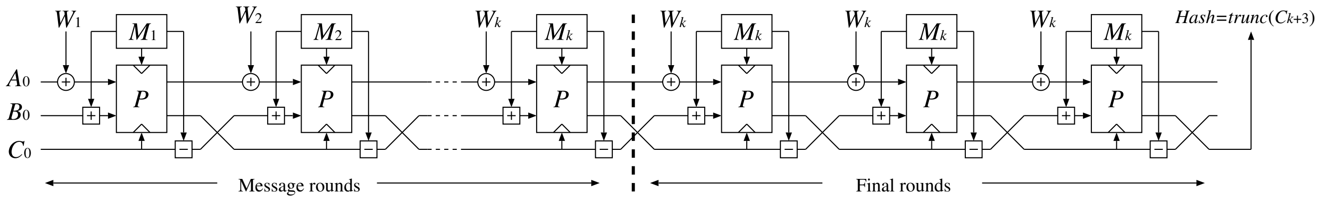

Fig. 1. The Mode of Operation of Shabal

2 Description of Shabal

In this section, we provide a brief description of the hash function Shabal [5].

2.1 Mode of Operation

Shabal employs an original mode of operation based on a keyed permutation which is called $\mathcal{P}$. The mode of operation is shown in Fig 1. In the mode of operation, the internal state consists of three parts (A, B, C) whose sizes are $32 \times r$, 512 and 512 respectively. r is one of the tunable security parameters. Let A_i, B_i, and C_i be the i-th internal states $(0 \leq i)$, and A_0, B_0 and C_0 are initialization vectors IV. M_i is the i-th message block of 512 bits $(0 < i)$.

When a message block M_{i+1} and a internal state (A_i, B_i, C_i) are given, the compression function $\mathcal{R}$ computes a new internal state $(A_{i+1}, B_{i+1}, C_{i+1})$ as follows:

$$(A_{i+1}, C_{i+1}) = \mathcal{P}_{M_{i+1},C_i}(A_i \oplus W_{i+1}, B_i + M_{i+1}),$$
$$B_{i+1} = C_i - M_{i+1}.$$

W_i is a 64-bit counter incremented at each compression function, and $\mathcal{P}_{M_{i+1},C_i}$ denotes that M_{i+1} and C_i act as the key of $\mathcal{P}$. Arithmetic operations $+$ and $-$ are wordwise ones with respected to 32-bit words, and $\oplus$ is an XOR operation. The mode of operation of Shabal repeat this calculation until the last message block. After that, three additional compression functions using the last message block are repeatedly performed without incrementing the counter. Finally a hash value is output by truncating C of the internal state.

2.2 Keyed Permutation $\mathcal{P}$

In each compression function, the keyed permutation $\mathcal{P}$ takes four inputs A, B, C and M, and outputs updated A and B. In the permutation $\mathcal{P}$, B, C and M are regarded as 16×32-bits words, and A is $r \times 32$-bit words, respectively. $A[i]$ denotes the i-th word of A. Then the definition of $\mathcal{P}$ is shown as in Table 2.

In Table 2, $\lll$, $\wedge$ and $\neg$ are a left rotation, a bitwise AND and a bitwise complement, respectively. The definition of the final loop in Table 2 is for the version of $r = 12$. If a different value of the parameter r is used, the definition of the final loop is changed. The detailed rule is described in Appendix C. The recommended set of the parameters by the designers of Shabal is $(p, r) = (3, 12)$, which is used for the version of Shabal submitted to the SHA-3 competition.

3 Low-weight Pseudo Collision Attack on Shabal

In this section, we explain a low-weight pseudo collision attack on Shabal which is one of our main contribution. To construct a low-weight pseudo collision attack, we insert a few differences

Table 2. Permutation $\mathcal{P}$

```
/* initial loop */
for $i = 0$ to 15 do
    $B[i] \leftarrow B[i] \lll 17$,
end for

/* main loop */
for $j = 0$ to $p$ - 1 do
    for $i = 0$ to 15 do
        $A[i + 16j \bmod r] \leftarrow 3 \times (A[i + 16j \bmod r]) \oplus C[8 - i \bmod 16] \oplus 5 \times (A[i - 1 + 16j \bmod r] \lll 15))$
                    $\oplus M[i] \oplus B[i + 13 \bmod 16] \oplus (B[i + 9 \bmod 16] \wedge \neg B[i + 6 \bmod 16])$,
        $B[i] \leftarrow (B[i] \lll 1) \oplus \neg A[i + 16j \bmod r]$,
    end for
end for

/* final loop */
for $j = 0$ to 35 do
    $A[j \bmod r] \leftarrow A[j \bmod r] + C[j + 3 \bmod 16]$.
end for
```

into a message and analyze propagation of the message difference through the inversion of the compression function. Let the inverse of the compression function be $\mathcal{R}^{-1}$. We need to search a differential path in which an output of $\mathcal{R}^{-1}$ is a low-weight difference to mount a low-weight pseudo collision attack. In the course of our study, we first show a near collision attack on $\mathcal{R}^{-1}$. After finding a near collision attack of $\mathcal{R}^{-1}$, it is converted into the low-weight pseudo collision attack on the compression function of Shabal.

3.1 Inverse of the Compression Function : $\mathcal{R}^{-1}$

To begin with, since we need to analyze the compression function $\mathcal{R}$ backward, we use $\mathcal{R}^{-1}$ defined as follows:

$$C_i = B_{i+1} + M_{i+1}, \tag{1}$$

$$(A'_i, B'_i) = P^{-1}_{M_{i+1},C_i}(A_{i+1}, C_{i+1}), \tag{2}$$

$$(A_i, B_i, C_i) = (A'_i \oplus W_{i+1}, B'_i - M_{i+1}, C_i), \tag{3}$$

where the inverse of the permutation $\mathcal{P}$ in Eq. (2) is defined as in Table 3, and A'_i and B'_i are intermediate values of A_i and B_i, respectively. In Table 3, $\ggg$ denotes a right rotation.

3.2 Linearization of Nonlinear Operations

In order to analyze propagation of differences, we study behaviors of differences through the nonlinear operations in $\mathcal{R}^{-1}$. Here we consider XOR differences in the analysis of $\mathcal{R}^{-1}$.

$\mathcal{R}^{-1}$ includes five nonlinear operations: a wordwise addition $+$, a wordwise subtraction $-$, a multiplication by 3^{-1}, a multiplication by 5 and a bitwise AND. To search difference propagation path in $\mathcal{R}^{-1}$ effectively, we use a linear approximation technique for these operations except AND. Differential probability of an addition was studied by Lipmaa and Moriai [11], we use their result to calculate difference probabilities.

Equations of the linear approximation and its probabilities used in our attack are as follows:

Table 3. Inverse of the Permutation $\mathcal{P}$

```
/* inverse of the final loop */
for $j$ = 35 to 0 do
    $A[j \bmod r] \leftarrow A[j \bmod r] - C[j + 3 \bmod 16]$,
end for

/* inverse of the main loop */
for $j$ = $p$ - 1 to 0 do
    for $i$ = 15 to 0 do
        $B[i] \leftarrow (B[i] \oplus \neg A[i + 16j \bmod r]) \ggg 1$,
        $A[i + 16j \bmod r] \leftarrow 3^{-1} \times (A[i + 16j \bmod r] \oplus B[i + 13 \bmod 16]$
                        $\oplus(B[i + 9 \bmod 16] \wedge \neg B[i + 6 \bmod 16]) \oplus M[i])$
                        $\oplus C[8 - i \bmod 16] \oplus A[i + 16j \bmod r] \oplus 5 \times (A[i - 1 + 16j \bmod r] \lll 15))$
    end for
end for

/* inverse of the initial loop */
for $i$ = 15 to 0 do
    $B[i] \leftarrow B[i] \ggg 17$,
end for
```

Wordwise Addition and Subtraction: With respect to differences α and β for two inputs x and y, we use the following equation of the linear approximation:

$$(x + y) \oplus ((x \oplus \alpha) + (y \oplus \beta)) = \alpha \oplus \beta. \tag{4}$$

The probability for Eq. (4) is $2^{-wt((\alpha \vee \beta) \wedge 2^{31}-1)}$, where $wt(x)$ is the number of nonzero elements of x, and $\vee$ is a bitwise OR. The linear approximation holds when there is no difference in carries. The linear approximation and the probability of subtraction are same as addition, because the subtraction can be expressed as the addition by a bitwise complement.

Multiplication by 5: With respect to difference α for an input x, we use the following equation of the linear approximation,

$$(x \times 5) \oplus ((x \oplus \alpha) \times 5) = (\alpha \ll 2) \oplus \alpha. \tag{5}$$

The probability for Eq. (5) is $2^{-wt((\alpha \ll 2) \vee \alpha) \wedge 011...100_{32})}$. The linear approximation has the same linearization form as multiplication by 9 treated by Indesteege and Preneel [8]. As well as [8], the dependency between x and $(x \ll 2)$ is ignored, because we suppose that the weight of differences α is low.

Multiplication by 3^{-1}: With respect to difference α , we use the following equation of the linear approximation,

$$(x \times 3^{-1}) \oplus ((x \oplus \alpha) \times 3^{-1}) = \gamma, \tag{6}$$

The probability for Eq. (6) is $2^{-wt((\gamma \ll 1) \vee \gamma) \wedge 011...110_{32})}$. In this equation, $\gamma = g^{-1}(\alpha)$ and $g(\gamma) = \gamma \oplus (\gamma \ll 1)$. $g^{-1}(\alpha)$ can be computed by determining from LSB to MSB of γ in each bit as follows: $\gamma_0 = \alpha_0, \gamma_1 = \alpha_1 \oplus \gamma_0, ..., \gamma_{31} = \alpha_{31} \oplus \gamma_{30}$, where γ_0 is LSB of γ, and α_0 is LSB of α, respectively.

For AND, we do not use any linear approximation. Given input differences α and β for the AND operation, the number of output difference patterns are $2^{wt(\alpha \vee \beta)}$, and each output difference pattern have the same probability, that is $2^{-wt(\alpha\vee\beta)}$. Since we suppose that α and β are low weight, this probability is high enough. Hence, we consider that AND operation is useful to construct a differential path with high probability.

3.3 Path Search Methodology

In this section, we introduce a search method to find a differential path with high probability such that the output of $\mathcal{R}^{-1}$ has low-weight differences. As previously described, we insert a few differences into the message block and analyze difference propagation in $\mathcal{R}^{-1}$. Actually, we insert only 1-bit difference into MSB of $M[0]_{i+1}$.

From Eq. (1), the 1-bit difference of $M[0]_{i+1}$ propagates to the MSB of $C_i[0]$ with probability of 1. Thus, 2-bit differences go into $\mathcal{P}^{-1}_{M,C}$. In the inverse of the final loop, difference of C propagates to A. Let $A[48], ..., A[59]$ and $B[48], ..., B[63]$ be inputs A and B of the inverse of the main loop, and $A[0], ..., A[11]$ and $B[0], ..., B[15]$ be the outputs of it. After the inverse of the final loop, MSBs of A[49] and A[53] have differences by MSB of $C[0]$ with probability of 1.

Then we start a path search from the start of the inverse of the main loop. By using the above notation of A and B, operations of the inverse of the main loop are expressed as follows:

$$\begin{aligned} A[i] &= 3^{-1} \times (A[i+12] \oplus (\neg B[i+6] \cdot B[i+9]) \oplus B[i+13] \oplus M[i]) \\ &\quad \oplus C[8-i] \oplus (5 \times (A[i+11] \lll 15)), \qquad (7) \\ B[i] &= (B[i+16] \oplus \neg A[i+12]) \ggg 1, \qquad (8) \end{aligned}$$

The path search for the inverse of the main loop is divided into two phases : an automatic search phase and a manual search phase.

In the automatic search phase, we utilize an aforementioned linear approximation technique. The loop contains three nonlinear operations such as a multiplication by 3^{-1}, a multiplication by 5, and AND. For a multiplication by 3^{-1}, we use Eq. (6). For a multiplication by 5, we use Eq. (5) and additional difference patterns such that i-th input difference changes $i, (i+1)$ and $(i+2)$-th differences, where i = 12, 13, 14. Each probability is 2^{-3}, which is calculated by [11]. For AND, we consider all possible difference patterns for each the input difference pattern. Using this method, we search the path automatically for the first 32 steps of the inverse of the main loop (in other view, it is the last 32 steps for the main loop) and find the path in which the weight of difference of A and B at the 32nd step is the smallest.

After that, for the last 16 step, we search the path manually. In this phase, we use not only linear approximations used in the automatic search phase but also other linear approximations which have even low probabilities for 3^{-1} and 5 to minimize the weight of the differences at 48th step. The reason is that when it is converted to a pseudo collision attack, probability of this phase can be increased by a technique similar to the message modification [13][14]. We will explain the detail of the technique in Sect.3.5.

3.4 Near Collision Attack on $\mathcal{R}^{-1}$

By using the above search methodology, we found a differential path which can be used for the near collision attack on $\mathcal{R}^{-1}$. Table 8 in Appendix A shows the obtained differential path. In the path, inputs A, C and M have 4 bits differences, that is, MSBs of $A[49], A[53], C[0]$ and $M[0]$. Through the inverse of the main loop, the outputs $A[0], ..., A[11]$ and $B[0], ..., B[15]$ have 15 bits and 28 bits differences, respectively. Since the inverse of the initial loop does not increase the number of differences, A'_i and B'_i also have 15 bits and 28 bits differences with probability of 2^{-156}. After the operation of Eq. (3), B_i have 29 (=28 + 1) difference with probability of

$2^{-28(=-(29-1))}$ due to B'_i and M_{i+1} with Eq. (4) and its probability. As a result, outputs of $\mathcal{R}^{-1}$ have $45(=15+29+1)$ bit differences.

This result can be seen as a near collision of $\mathcal{R}^{-1}$. Thus, $1363(=512+512-384-45)$-bit near collision of $\mathcal{R}^{-1}$ with probability of $2^{-184(=-(156+28))}$ is obtained by this analysis.

3.5 Constructing a Low-weight Pseudo Collision Attack

Next, we construct a low-weight pseudo collision attack from the near collision of the $\mathcal{R}^{-1}$. Obviously, when we insert the above 45 bits difference into A_i, B_i and C_i, a colliding message can be found with complexity of 2^{184}. Moreover, we can improve the attack by the modification technique similar to [13][14]. Table 9 in Appendix B shows sufficient conditions in the first 16 steps of the main loop for the path of Table 8. In order to improve the attack, we choose A_{i+1}, B_{i+1}, C_{i+1} and M_{i+1} which satisfy as many sufficient conditions as possible.

First, we choose the $B'_i(=B_i+M_{i+1})$ satisfying conditions of AND. For AND, 21 conditions (out of 38) holds by choosing $B'[6], ..., B'[15]$. Next, we choose M_{i+1} such that a difference act like Eq. (4) in the operation $(B'_i - M_{i+1})$. Since the weight of the difference of B'_i is only 28, the degree of freedom of M_{i+1} remain sufficiently yet. After that, we choose $A'_i(=A_i \oplus W_i)$ and C_i, and use the remaining freedom of M_{i+1} to satisfy the conditions of $\times 3$ and $\times 5$. In particular, since each word of C_i and M_{i+1} are used once in the first 16 steps, and A_i are used once in the first 12 step, we choose each word to satisfy conditions step by step. Thus, by the modification, 72 condition holds (out of 89) in the first 16 steps, and differences act like Eq. (4) in the operations of $(B'_i - M_{i+1})$.

Consequently, when we insert the obtained 45 difference into A_i, B_i and C_i, a colliding message can be found with complexity of $2^{84(=184-72-28)}$.

3.6 Discussion

We summarize two results described in this section as:

1. 1363-bit (out of 1408 bits) near collision attack on the inverse of the compression function $\mathcal{R}^{-1}$ with complexity of 2^{184}.
2. Low-weight pseudo collision attack (45 bits) on the compression function $\mathcal{R}$ with complexity of 2^{84}.

The first result shows efficiency of the diffusion property of the inverse of compression function $\mathcal{R}^{-1}$. We reveal that 1-bit message difference diffuses only 45 (out of 1408) bit through $\mathcal{R}^{-1}$ in a certain setting. We expect that this result might be useful to construct a collision attack with multi block setting as in [4]. But, we could not find such a differential path so far, thus this is an open problem. Although these result does not threat the claimed security of Shabal due to that this is a pseudo collision setting, these results show that the compression function of Shabal has the non-ideal differential property.

4 Preimage Attack on Reduced Shabal-512

In this section, we consider a preimage attack on variants of Shabal-512. In [5], preimage attacks on the weakened variants of Shabal-512 that is without the final loop in $\mathcal{P}$ have been analyzed[1]. The weakened variant of Shabal-512 using security parameter $(p, r) = (1, 12)$ does not hold the preimage resistance.

[1] The variant is called Weakinson-NoFinalUpdateA in [5]

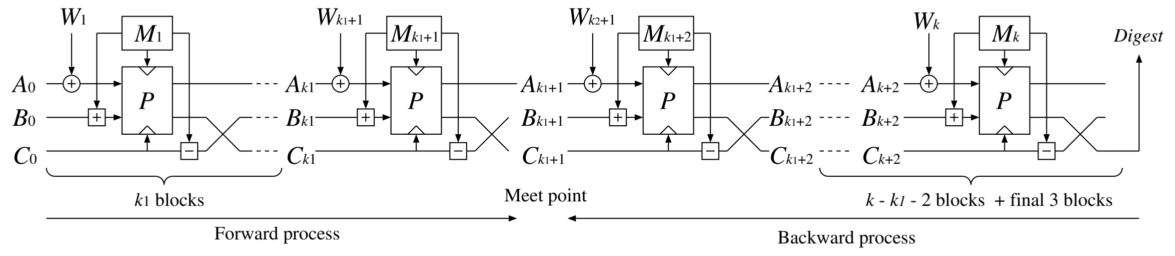

Fig. 2. Outline of the Preimage Attack

The preimage attack utilizes the weakness of the inverse of the compression function $\mathcal{R}^{-1}$: if A_{i+1}, B_{i+1} and C_{i+1} are given as inputs of $\mathcal{R}^{-1}$, the attacker is able to find a message M_{i+1} such that B_i has any predetermined value. Since the same attack is not successful on the weakened variant using $(p, r) = (2, 12)$, the current record of the preimage attack is only the weakened variant using $(p, r) = (1, 12)$.

In this section, two kinds of preimage attack are shown. First, we propose a novel preimage attack on the weakened variant of Shabal-512 using $(p, r) = (2, 12)$. Although this attack is based on the same weakness of $\mathcal{R}^{-1}$ as [5], we use an advanced technique to find M_{i+1} such that B_i has any predetermined value. Second, this technique can be applied to the Shabal-512 with $(p, r) = (1.5, 8)$ including the final loop. This is the first preimage attack on Shabal-512 with reduced security parameters.

4.1 Basic Framework of the Preimage Attack

First, we explain a basic framework for a preimage attack on the variant of Shabal-512. This framework is same as [5]. Let *digest* be a given hash value and $\mathcal{M}$ be a k-block message as a preimage of the *digest* to be calculated, where k is an arbitrary integer more than 5. The outline of the preimage attack is shown in Fig 2. We split $\mathcal{M}$ into three parts : the first k_1 blocks, the next two intermediate blocks, and the last $(k - k_1 - 2)$ blocks. For the first k_1 blocks, we randomly choose $M_1, ..., M_{k_1}$ and compute $(A_{k_1}, B_{k_1}, C_{k_1})$. For the last $(k - k_1 - 2)$ block, we set *digest* as C_{k+3} part and choose the other parts A_{k+3} and B_{k+3} randomly, where A_{k+3}, B_{k+3} and C_{k+3} are values after final three compression functions. After that, similarly, we randomly choose $M_{k_1+3}, ..., M_k$ and compute $(A_{k_1+2}, B_{k_1+2}, C_{k_1+2})$ backward. Then, for the two intermediate blocks, we use the meet-in-the-middle approach [9].

Since the size of the meet point $(A_{k_1+1}, B_{k_1+1}, C_{k_1+1})$ is too large, *e.g.* for r=12, 1408 bits, the straight meet-in-the-middle approach does not work. Thus, we utilize these two intermediate message blocks, M_{k_1+1} and M_{k_1+2}, to reduce the size of the meet point. In particular, we aim to fix B_{k_1+1} to an arbitrary value β to reduce the size of the meet point. In the forward process, it is noticed that B_{k_1+1} can be controlled easily by choosing $M_{k_1+1} = C_{k_1} - \beta$. In the backward process, if B_{k_1+1} can be also fixed to β, the size of the meet point is reduced to (A_{k_1+1}, C_{k_1+1}) from $(A_{k_1+1}, B_{k_1+1}, C_{k_1+1})$. Therefore, more effective meet-in-the-preimage attack can be executed. Our goal is to find M_{k_1+2} such that B_{k_1+1} will be β in the backward process effectively.

4.2 Guess-and-Determine Technique

We use a guess-and-determine (GD) technique to find M_{k_1+2} effectively such that B_{k_1+1} is β in the backward process. Originally the GD technique is developed for attacks of wordwise stream ciphers [7]. Since the permutation of Shabal is similar to a design of a wordwise NLFSR-type stream cipher, we customize the GD technique for finding such M_{k_1+2}.

The GD technique is a technique for obtaining unknown values by using relations between known values and unknown values. As implied by the name of the GD technique, we first guess some of unknown values and then determine the other unknown values by using the equations between these values. In the case of Shabal, the unknown value is M_{k_1+2} and the known values are A_{k_1+2}, B_{k_1+2}, C_{k_1+2} and B_{k_1+1}.

Basic attack steps are as follows:

Step 1: Construct equations describing relationships between M_{k_1+2}, A_{k_1+2}, B_{k_1+2}, C_{k_1+2} and B_{k_1+1},
Step 2: Guess some part of words in M_{k_1+2}
Step 3: Determine the other words in M_{k_1+2} by using equations obtained in the step 1.

If we find M_{k_1+2} such that B_{k_1+1} is to be β in the backward process effectively by using the GD technique, we can construct a preimage attack using the attack framework in Sect. 4.1. Next, we are going to explain details of the steps of the GD technique for Shabal-512.

4.3 Preimage Attack on the Weakened Variant of Shabal-512 Using $(p, r) = (2, 12)$

First, we show a preimage attack on the weakened variant of Shabal-512 using $(p, r) = (2, 12)$. For the variant, the number of the main loop is 32 ($= 2 \times 16$), the size of A is 384 ($= 12 \times 32$) bits, and it does not include the final loop.

The steps of the GD approach are as follows.

Step 1: Construct equations between unknown value M_{k_1+2} and known values B_{k_1+1}, A_{k_1+2}, B_{k_1+2} and C_{k_1+2}. Let $A[0], ..., A[11]$ and $B[0], ..., B[15]$ be inputs and $A[32], ..., A[43]$ and $B[32], ..., B[47]$ be outputs of the main loop of $\mathcal{P}$. In the main loop, these values satisfy Eq. (7) and (8). Since this variant does not include the final loop in $\mathcal{P}$, $A[32], ..., A[43]$ correspond to known values $A_{k_1+2}[0], ..., A_{k_1+2}[11]$, and $B[32], ..., B[47]$ correspond to known values $C_{k_1+2}[0], ..., C_{k_1+2}[15]$. In Eq. (7) and (8), other values $M[i]$ and $C[i]$ also appear. $M[0], ..., M[15]$ correspond to unknown values $M_{k_1+2}[0], ..., M_{k_1+2}[15]$. From Eq. (1), $C[i]$ is expressed by using known value B_{k_1+2} and unknown value M_{k_1+2} as follows:

$$C[i] = B_{k_1+2}[i] + M_{k_1+2}[i]. \tag{9}$$

The equation with respect to other known value B_{k_1+1} is

$$B_{k_1+1}[i] = (B[i]) \ggg 17) - M[i]. \tag{10}$$

This equation is obtained from Eq. (3) and the inverse of the initial loop.
From (7)-(10), we construct equations as shown in Table 4, which includes all equations between unknown value M_{k_1+2} and known values B_{k_1+1}, A_{k_1+2}, B_{k_1+2} and C_{k_1+2}. In Table 4, underlined values are unknown values and other values are known values, and $\beta[0], ..., \beta[15]$ correspond to known values $B_{k_1+1}[0], ..., B_{k_1+1}[15]$. From Eq. (8), $B[20], ..., B[31]$ are computed with known values $B[36], ..., B[47]$ and $A[32], ..., A[43]$. This is why that $B[20], ..., B[31]$ are known values. f_a is a function defined as follows:

$$\begin{aligned} &f_a(A[i+12], (B[i+6] \cdot B[i+9]), B[i+13], M[i], C[8-i], A[i+11]) \\ &= 3^{-1} \times (A[i+12] \oplus (\neg B[i+6] \cdot B[i+9]) \oplus B[i+13] \oplus M[i]). \\ &\quad \oplus C[8-i] \oplus (5 \times (A[i+11] \lll 15)), \end{aligned}$$

Table 4. Equations for the Weakened Shabal Using $(p, r) = (2, 12)$

$$\underline{A[31]} = f_a(A[43], (B[37] \cdot B[40]), B[44], \underline{M[15]}, \underline{C[9]}, A[42]), \tag{11}$$
$$\underline{A[30]} = f_a(A[42], (B[36] \cdot B[39]), B[43], \underline{M[14]}, \underline{C[10]}, A[41]), \tag{12}$$
$$\underline{A[29]} = f_a(A[41], (B[35] \cdot B[38]), B[42], \underline{M[13]}, \underline{C[11]}, A[40]), \tag{13}$$
$$\underline{A[28]} = f_a(A[40], (B[34] \cdot B[37]), B[41], \underline{M[12]}, \underline{C[12]}, A[39]), \tag{14}$$
$$\underline{A[27]} = f_a(A[39], (B[33] \cdot B[36]), B[40], \underline{M[11]}, \underline{C[13]}, A[38]), \tag{15}$$
$$\underline{A[26]} = f_a(A[38], (B[32] \cdot B[35]), B[39], \underline{M[10]}, \underline{C[14]}, A[37]), \tag{16}$$
$$\underline{A[25]} = f_a(A[37], (B[31] \cdot B[34]), B[38], \underline{M[9]}, \underline{C[15]}, A[36]), \tag{17}$$
$$\underline{A[24]} = f_a(A[36], (B[30] \cdot B[33]), B[37], \underline{M[8]}, \underline{C[0]}, A[35]), \tag{18}$$
$$\underline{A[23]} = f_a(A[35], (B[29] \cdot B[32]), B[36], \underline{M[7]}, \underline{C[1]}, A[34]), \tag{19}$$
$$\underline{A[22]} = f_a(A[34], (B[28] \cdot B[31]), B[35], \underline{M[6]}, \underline{C[2]}, A[33]), \tag{20}$$
$$\underline{A[21]} = f_a(A[33], (B[27] \cdot B[30]), B[34], \underline{M[5]}, \underline{C[3]}, A[32]), \tag{21}$$
$$\underline{A[20]} = f_a(A[32], (B[26] \cdot B[29]), B[33], \underline{M[4]}, \underline{C[4]}, \underline{A[31]}), \tag{22}$$
$$\underline{A[19]} = f_a(\underline{A[31]}, (B[25] \cdot B[28]), B[32], \underline{M[3]}, \underline{C[5]}, \underline{A[30]}), \tag{23}$$
$$\underline{A[18]} = f_a(\underline{A[30]}, (B[24] \cdot B[27]), B[31], \underline{M[2]}, \underline{C[6]}, \underline{A[29]}), \tag{24}$$
$$\underline{A[17]} = f_a(\underline{A[29]}, (B[23] \cdot B[26]), B[30], \underline{M[1]}, \underline{C[7]}, \underline{A[28]}), \tag{25}$$
$$\underline{A[16]} = f_a(\underline{A[28]}, (B[22] \cdot B[25]), B[29], \underline{M[0]}, \underline{C[8]}, \underline{A[27]}), \tag{26}$$
$$\underline{A[15]} = f_a(\underline{A[27]}, (B[21] \cdot B[24]), B[28], \underline{M[15]}, \underline{C[9]}, \underline{A[26]}), \tag{27}$$
$$\underline{A[14]} = f_a(\underline{A[26]}, (B[20] \cdot B[23]), B[27], \underline{M[14]}, \underline{C[10]}, \underline{A[25]}), \tag{28}$$
$$\underline{A[13]} = f_a(\underline{A[25]}, (\underline{B[19]} \cdot B[22]), B[26], \underline{M[13]}, \underline{C[11]}, \underline{A[24]}), \tag{29}$$
$$\underline{A[12]} = f_a(\underline{A[24]}, (\underline{B[18]} \cdot B[21]), B[25], \underline{M[12]}, \underline{C[12]}, \underline{A[23]}), \tag{30}$$

$$\underline{B[19]} = (B[35] \oplus \neg\underline{A[31]}) \ggg 1, \tag{31}$$
$$\underline{B[18]} = (B[34] \oplus \neg\underline{A[30]}) \ggg 1, \tag{32}$$
$$\underline{B[17]} = (B[33] \oplus \neg\underline{A[29]}) \ggg 1, \tag{33}$$
$$\underline{B[16]} = (B[32] \oplus \neg\underline{A[28]}) \ggg 1, \tag{34}$$
$$\beta[15] = (B[31] \oplus \neg\underline{A[27]}) \ggg 18 - \underline{M[15]}, \tag{35}$$
$$\beta[14] = (B[30] \oplus \neg\underline{A[26]}) \ggg 18 - \underline{M[14]}, \tag{36}$$
$$\beta[13] = (B[29] \oplus \neg\underline{A[25]}) \ggg 18 - \underline{M[13]}, \tag{37}$$
$$\beta[12] = (B[28] \oplus \neg\underline{A[24]}) \ggg 18 - \underline{M[12]}, \tag{38}$$
$$\beta[11] = (B[27] \oplus \neg\underline{A[23]}) \ggg 18 - \underline{M[11]}, \tag{39}$$
$$\beta[10] = (B[26] \oplus \neg\underline{A[22]}) \ggg 18 - \underline{M[10]}, \tag{40}$$
$$\beta[9] = (B[25] \oplus \neg\underline{A[21]}) \ggg 18 - \underline{M[9]}, \tag{41}$$
$$\beta[8] = (B[24] \oplus \neg\underline{A[20]}) \ggg 18 - \underline{M[8]}, \tag{42}$$
$$\beta[7] = (B[23] \oplus \neg\underline{A[19]}) \ggg 18 - \underline{M[7]}, \tag{43}$$
$$\beta[6] = (B[22] \oplus \neg\underline{A[18]}) \ggg 18 - \underline{M[6]}, \tag{44}$$
$$\beta[5] = (B[21] \oplus \neg\underline{A[17]}) \ggg 18 - \underline{M[5]}, \tag{45}$$
$$\beta[4] = (B[20] \oplus \neg\underline{A[16]}) \ggg 18 - \underline{M[4]}, \tag{46}$$
$$\beta[3] = (\underline{B[19]} \oplus \neg\underline{A[15]}) \ggg 18 - \underline{M[3]}, \tag{47}$$
$$\beta[2] = (\underline{B[18]} \oplus \neg\underline{A[14]}) \ggg 18 - \underline{M[2]}, \tag{48}$$
$$\beta[1] = (\underline{B[17]} \oplus \neg\underline{A[13]}) \ggg 18 - \underline{M[1]}, \tag{49}$$
$$\beta[0] = (\underline{B[16]} \oplus \neg\underline{A[12]}) \ggg 18 - \underline{M[0]}. \tag{50}$$

Step 2: Guess some part of words in M. In particular, we guess $M[12], M[13]$ and $M[14]$. Generally, when the number of words to be guessed is fewer, the GD attack becomes more effective. By checking all words pattern of M to be guessed, we found that at least three words in M need to be guessed to determine all words of M when using equations in Table 4. In other word, if the number of guessed words in M is smaller than 3, all words in M can not be determined from equations in Table 4. $M[12], M[13]$ and $M[14]$ are one of the least number of guessed words in M to determine all words in M obtained by our analysis.

Step 3: Determine other words in M by using equations in Table 4. The procedure for determining other word in M are summarized in Table 5. Table 5 indicates the relation between determined words and used equations at each step. For example, at step 1, $A[28]$, $A[26]$, $A[25]$ and $A[24]$ are determined by using Eq. (14), (36), (37) and (38). In particular, Eq. (14) contain unknown value $M[12]$ and $C[12]$. Since $M[12]$ is guessed in the step 2, $M[12]$ is known values and $C[12]$ also becomes known value from Eq. (9). Therefore, $A[28]$ can be determined by Eq. (14).

As Table 5, we determine all words in M step by step by using equations in Table 4. However, since Eq. (45), (46) and (49) are not used for determining all words in M, the obtained message M satisfies these equations with probability of 2^{-96}, because each variables in equation are 32-bit. Consequently, the complexity of finding a message M such that B_{k_1+1} is β in the backward process is 2^{96}.

For the preimage attack, we compute 2^{400} sets of $(A_{k+1}, \beta, C_{k+1})$ and make a table in the backward process. In the forward process, we compute 2^{496} sets of $(A_{k+1}, \beta, C_{k+1})$. Since the total size of A_{k+1} and C_{k+1} is 896(=384 + 512) bits, we can find a matching pair among $2^{896(=496+400)}$ pairs using the meet-in-the-middle technique. Therefore, a preimage on the weakened variant of Shabal-512 using security parameter $(p, r) = (2, 12)$ can be found with complexity of 2^{497} and memory of 2^{400}.

Table 5. Procedure of Determining all word in M for the Weakened Shabal Using $(p, r) = (2, 12)$

Step	Determined word	Equation	Step	Determined word	Equation
1	$A[28], A[26], A[25], A[24]$	(14), (36), (37), (38)	11	$A[12], M[15]$	(30), (35)
2	$M[10], B[16]$	(16), (34)	12	$M[9], M[0]$	(17), (50)
3	$A[30]$, $A[14]$, $A[22]$	(12), (28), (40)	13	$A[31]$,$M[8]$, $A[15]$, $A[21]$	(11), (18), (27), (41)
4	$B[18]$	(32)	14	$A[16]$, $B[19]$, $A[20]$	(26), (31), (42)
5	$M[2]$	(48)	15	$M[5]$, $M[4]$, $A[13]$, $M[3]$	(21), (22), (29), (47)
6	$M[6]$	(20)	16	$A[19]$	(23)
7	$A[18]$	(44)	17	$M[7]$	(43)
8	$A[29]$	(24)	18	$M[1]$	(19)
9	$M[11]$, $B[17]$	(13), (33)	19	$A[17]$	(25)
10	$A[27]$, $A[23]$	(15), (39)	-	-	-

4.4 Preimage attack on Shabal-512 Using $(p, r) = (1.5, 8)$

Next, we apply the GD technique to Shabal-512 using $(p, r) = (1.5, 8)$. For the variant, the number of the main loop is 24 ($= 1.5 \times 16$), the size of A is 256 ($= 8 \times 32$) bits. Since the value of r differ from 12, the final loop must be modified. In this attack, we define the final loop as follows:

```
for $j = 0$ to 24 do
    $A[j \bmod 8] \leftarrow A[j \bmod 8] + C[j + 3 \bmod 16]$.
end for
```

In Appendix C, the reason for the above modification is explained.

The attack method is same as the attack of the weakened variant using $(p, r) = (2, 12)$,

Step 1: Construct equations between unknown value M_{k_1+2} and known values B_{k_1+1}, A_{k_1+2}, B_{k_1+2} and C_{k_1+2} Let $A[0], ..., A[7]$ and $B[0], ..., B[15]$ be inputs and $A[24], ..., A[31]$ and $B[24], ..., B[39]$ be outputs of the main loop. $B[24], ..., B[39]$ correspond to $C_{k_1+2}[0], ..., C_{k_1+2}[15]$. In the main loop, these values satisfy following equations,

$$A[i] = f_a(A[i+8], (B[i+6] \cdot B[i+9]), B[i+13], M[i], C[8-i], A[i+7]), \tag{51}$$

$$B[i] = (B[i+16] \oplus \neg A[i+8]) \ggg 1, \tag{52}$$

where $M[0], ..., M[15]$ correspond to unknown value $M_{k_1+2}[0], ..., M_{k_1+2}[15]$, and $C[0], ..., C[15]$ can be also expressed by using known value B_{k_1+2} and unknown value M_{k_1+2} as Eq. (9). Due to the above redefined final loop, $A[24], ..., A[31]$ are $(A_{k_1+2}[0] - 2C[3] - C[11]), ..., (A_{k_1+2}[7] - 2C[10] - C[2])$. The equation with respect to B_{K_1+2} is same as Eq. (10).
In order to simplify equations, we fixed all words of C_{k_1+2} to 0 by choosing M_{k_1+3} as $M_{k_1+3} = -B_{k_1+3}$ in $(k_1 + 3)$ th compression function. From Eq. (10), (51) and (52), we construct equations as shown in Table 6. In Table 6, since $B[24], ..., B[39]$ are 0, unknown values $B[8], ..., B[23]$ are expressed as $(\neg A[16] \ggg 1), ..., (\neg A[31] \ggg 1)$ from Eq. (52).

Step 2: Guess $M[1], M[2], M[6], M[7], M[8], M[9]$ and $M[10]$, which are one of the least number of guessed words in M to determine all words in M.

Step 3: Determine other words in M from equations of Table 6. The procedure for determining M is shown as Table 7. For example, at step 1 of the procedure, $A[31]$ is determined from Eq (53) containing unknown value $C[10]$ and $C[2]$. Since $M[10]$ and $M[2]$ are guessed, $C[10]$ and $C[2]$ become known values by Eq. (9). Therefore, $A[31]$ can be determined from Eq (53). In this case, Eq. (80), (81), (88), (89), (90), (91), and (92) are not used, thus the obtained message M satisfies these equations with probability of 2^{-224}. Consequently, the complexity of finding message M such that B_{k_1+1} is β in the backward process is 2^{224}.

Table 6. Equations for the Shabal Using $(p, r) = (1.5, 8)$

$$\underline{A[31]} = A_{k_1+2}[7] - \underline{2C[10]} - \underline{C[2]}, \tag{53}$$
$$\underline{A[30]} = A_{k_1+2}[6] - \underline{2C[9]} - \underline{C[1]}, \tag{54}$$
$$\underline{A[29]} = A_{k_1+2}[5] - \underline{2C[8]} - \underline{C[0]}, \tag{55}$$
$$\underline{A[28]} = A_{k_1+2}[4] - \underline{2C[7]} - \underline{C[15]}, \tag{56}$$
$$\underline{A[27]} = A_{k_1+2}[3] - \underline{2C[6]} - \underline{C[14]}, \tag{57}$$
$$\underline{A[26]} = A_{k_1+2}[2] - \underline{2C[5]} - \underline{C[13]}, \tag{58}$$
$$\underline{A[25]} = A_{k_1+2}[1] - \underline{2C[4]} - \underline{C[12]}, \tag{59}$$
$$\underline{A[24]} = A_{k_1+2}[0] - \underline{2C[3]} - \underline{C[11]}, \tag{60}$$
$$\underline{A[23]} = f_a(\underline{A[31]}, 0, 0, \underline{M[7]}, \underline{C[1]}, \underline{A[30]}), \tag{61}$$
$$\underline{A[22]} = f_a(\underline{A[30]}, 0, 0, \underline{M[6]}, \underline{C[2]}, \underline{A[29]}), \tag{62}$$
$$\underline{A[21]} = f_a(\underline{A[29]}, 0, 0, \underline{M[5]}, \underline{C[3]}, \underline{A[28]}), \tag{63}$$
$$\underline{A[20]} = f_a(\underline{A[28]}, 0, 0, \underline{M[4]}, \underline{C[4]}, \underline{A[27]}), \tag{64}$$
$$\underline{A[19]} = f_a(\underline{A[27]}, 0, 0, \underline{M[3]}, \underline{C[5]}, \underline{A[26]}), \tag{65}$$
$$\underline{A[18]} = f_a(\underline{A[26]}, 0, 0, \underline{M[2]}, \underline{C[6]}, \underline{A[25]}), \tag{66}$$
$$\underline{A[17]} = f_a(\underline{A[25]}, 0, 0, \underline{M[1]}, \underline{C[7]}, \underline{A[24]}), \tag{67}$$
$$\underline{A[16]} = f_a(\underline{A[24]}, 0, 0, \underline{M[0]}, \underline{C[8]}, \underline{A[23]}), \tag{68}$$
$$\underline{A[15]} = f_a(\underline{A[23]}, 0, 0, \underline{M[15]}, \underline{C[9]}, \underline{A[22]}), \tag{69}$$
$$\underline{A[14]} = f_a(\underline{A[22]}, (\underline{B[20]} \cdot \underline{B[23]}), 0, \underline{M[14]}, \underline{C[10]}, \underline{A[21]}), \tag{70}$$
$$\underline{A[13]} = f_a(\underline{A[21]}, (\underline{B[19]} \cdot \underline{B[22]}), 0, \underline{M[13]}, \underline{C[11]}, \underline{A[20]}), \tag{71}$$
$$\underline{A[12]} = f_a(\underline{A[20]}, (\underline{B[18]} \cdot \underline{B[21]}), 0, \underline{M[12]}, \underline{C[12]}, \underline{A[19]}), \tag{72}$$

$$\underline{A[11]} = f_a(\underline{A[19]}, (\underline{B[17]} \cdot \underline{B[20]}), 0, \underline{M[11]}, \underline{C[13]}, \underline{A[18]}), \tag{73}$$
$$\underline{A[10]} = f_a(\underline{A[18]}, (\underline{B[16]} \cdot \underline{B[19]}), \underline{B[23]}, \underline{M[10]}, \underline{C[14]}, \underline{A[17]}), \tag{74}$$
$$\underline{A[9]} = f_a(\underline{A[17]}, (\underline{B[15]} \cdot \underline{B[18]}), \underline{B[22]}, \underline{M[9]}, \underline{C[15]}, \underline{A[16]}), \tag{75}$$
$$\underline{A[8]} = f_a(\underline{A[16]}, (\underline{B[14]} \cdot \underline{B[17]}), \underline{B[21]}, \underline{M[8]}, \underline{C[0]}, \underline{A[15]}), \tag{76}$$
$$\beta[15] = (\neg\underline{A[23]}) \ggg 18 - \underline{M[15]}, \tag{77}$$
$$\beta[14] = (\neg\underline{A[22]}) \ggg 18 - \underline{M[14]}, \tag{78}$$
$$\beta[13] = (\neg\underline{A[21]}) \ggg 18 - \underline{M[13]}, \tag{79}$$
$$\beta[12] = (\neg\underline{A[20]}) \ggg 18 - \underline{M[12]}, \tag{80}$$
$$\beta[11] = (\neg\underline{A[19]}) \ggg 18 - \underline{M[11]}, \tag{81}$$
$$\beta[10] = (\neg\underline{A[18]}) \ggg 18 - \underline{M[10]}, \tag{82}$$
$$\beta[9] = (\neg\underline{A[17]}) \ggg 18 - \underline{M[9]}, \tag{83}$$
$$\beta[8] = (\neg\underline{A[16]}) \ggg 18 - \underline{M[8]}, \tag{84}$$
$$\beta[7] = ((\neg\underline{A[31]} \ggg 1) \oplus \neg\underline{A[15]}) \ggg 18 - \underline{M[7]}, \tag{85}$$
$$\beta[6] = ((\neg\underline{A[30]} \ggg 1) \oplus \neg\underline{A[14]}) \ggg 18 - \underline{M[6]}, \tag{86}$$
$$\beta[5] = ((\neg\underline{A[29]} \ggg 1) \oplus \neg\underline{A[13]}) \ggg 18 - \underline{M[5]}, \tag{87}$$
$$\beta[4] = ((\neg\underline{A[28]} \ggg 1) \oplus \neg\underline{A[12]}) \ggg 18 - \underline{M[4]}, \tag{88}$$
$$\beta[3] = ((\neg\underline{A[27]} \ggg 1) \oplus \neg\underline{A[11]}) \ggg 18 - \underline{M[3]}, \tag{89}$$
$$\beta[2] = ((\neg\underline{A[26]} \ggg 1) \oplus \neg\underline{A[10]}) \ggg 18 - \underline{M[2]}, \tag{90}$$
$$\beta[1] = ((\neg\underline{A[25]} \ggg 1) \oplus \neg\underline{A[9]}) \ggg 18 - \underline{M[1]}, \tag{91}$$
$$\beta[0] = ((\neg\underline{A[24]} \ggg 1) \oplus \neg\underline{A[8]}) \ggg 18 - \underline{M[0]}. \tag{92}$$

Table 7. Procedure of Determining all word in M for the Shabal Using $(p, r) = (1.5, 8)$

Step	Determined word	Equation	Step	Determined word	Equation
1	$A[31], A[30], A[18],$ $A[17], A[16]$	(53), (54), (82), (83), (84)	10	$A[9]$	(75)
			11	$A[14]$	(86)
2	$A[23]$, $A[15]$	(61), (85)	12	$A[21]$	(70)
3	$M[15]$	(77)	13	$M[13]$	(79)
4	$A[28]$, $A[22]$	(56), (69)	14	$M[5]$	(58)
5	$A[29]$, $M[14]$	(62), (78)	15	$M[3]$, $A[13]$	(63), (87)
6	$M[0]$, $A[27]$	(55), (57)	16	$M[11]$, $A[19]$, $A[20]$	(60), (65) (71)
7	$A[24]$	(68)	17	$M[4]$, $A[11]$	(64), (73)
8	$A[25]$,$A[10]$	(67), (74)	18	$M[12]$	(59)
9	$A[26]$, $A[8]$	(66), (76)	19	$A[12]$	(72)

For the preimage attack, we compute 2^{272} sets of $(A_{k+1}, \beta, C_{k+1})$ and make a table in the backward process. In the forward process, we compute 2^{496} sets of $(A_{k+1}, \beta, C_{k+1})$. Since the total size of A_{k+1} and C_{k+1} is 768 (= 256 + 512) bits, we can find a matching pair among $2^{768(=496+272)}$ pairs using the meet-in-the-middle technique. Therefore, a preimage on Shabal-512 using security parameter $(p, r) = (1.5, 8)$ can be found with complexity of 2^{497} and memory of 2^{272}.

5 Conclusion

We have shown two types of attacks on Shabal, they are a low-weight pseudo collision attack and a preimage attack.

For a low-weight pseudo collision attack, we first constructed a 1363-bit near collision attack on the inverse of the compression function $\mathcal{R}^{-1}$ with complexity of 2^{184}. The attack reveals efficiency of the diffusion property of the inverse of the compression function $\mathcal{R}^{-1}$. After that, we converted it into low-weight (45-bits) pseudo collision attack on the full compression function of Shabal with complexity of 2^{84}. Although these result does not threat the claimed security of Shabal, these show that the compression function have the non-ideal differential property. In addition, these result might be useful for the further cryptanalysis.

For a preimage attack, we introduced the GD technique and applied it to two variants of Shabal-512. For the weakened variant of Shabal-512 without the final loop using the security parameters $(p, r) = (2, 12)$, a preimage can be found with complexity of 2^{497} and memory of 2^{400}. For the Shabal-512 using security parameter $(p, r) = (1.5, 8)$, a preimage can be found with complexity of 2^{497} and memory of 2^{272}. This is the first preimage attack on Shabal-512 with reduced security parameters. Since designer's recommended security parameter is (3, 12), we claim that this attack does not lead to the security problem of the full Shabal hash function directly.

References

1. G. V. Assche, "A rotational distinguisher on Shabal's keyed permutation and its impact on the security proofs." Available : `http://gva.noekeon.org/papers/ShabalRotation.pdf`, 2010.
2. J-P. Aumasson, "On the pseudorandomness of Shabal's keyed permutation." Available : `http://131002.net/data/papers/Aum09.pdf`, 2009.
3. J-P. Aumasson, A. Mashatan, and W. Meier, "More on Shabal's permutation." Available : `http://131002.net/data/papers/AMM09.pdf`, 2009.
4. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet, and W. Jalby, "Collisions of SHA-0 and reduced SHA-1." in *Proceedings of EUROCRYPT'05* (R. Cramer, eds.), LNCS, vol. 3494, pp. 36–57, Springer-Verlag, 2005.
5. E. Bresson, A. Canteaut, B. Chevallier-Mames, C. Clavier, T. Fuhr, A. Gouget, T. Icart, J. Misarsky, M. Naya-Plasencia, P. Paillier, T. Pornin, J. Reinhard, C. Thuillet, and M. Videau, "Shabal, a submission to NIST's cryptographic hash algorithm competition." Submission to NIST, 2008.
6. E. Bresson, A. Canteaut, B. Chevallier-Mames, C. Clavier, T. Fuhr, A. Gouget, T. Icart, J. Misarsky, M. Naya-Plasencia, P. Paillier, T. Pornin, J. Reinhard, C. Thuillet, and M. Videau, "Indifferentiability with distinguishers: Why Shabal does not require ideal ciphers." Cryptology ePrint Archive, Report 2009/199, 2009.
7. P. Hawkes and G. Rose, "Guess-and-determine attacks on SNOW." in *Proceedings of SAC'02* (K. Nyberg and H. Heys, eds.), LNCS, vol. 2595, pp. 37–46, Springer-Verlag, 2003.
8. S. Indesteege and B. Preneel, "Practical collisions for EnRUPT." in *Proceedings of FSE'09* (O. Dunkelman, eds.), LNCS, vol. 5665, pp. 246–259, Springer-Verlag, 2009.
9. D. Khovratovich, I. Nikolic, and R. Weinmann, "Meet-in-the-middle attacks on SHA-3 candidates." in *Proceedings of FSE'09* (O. Dunkelman, eds.), LNCS, vol. 5665, pp. 228–245, Springer-Verlag, 2009.
10. L. Knudsen, K. Matusiewicz, and S. Thomsen, "Observations on the Shabal keyed permutation." Available : `http://www.mat.dtu.dk/people/S.Thomsen/shabal/shabal.pdf`, 2009.
11. H. Lipmma and S. Moriai, "Efficient algorithms for computing differential properties of addition." in *Proceedings of FSE'01* (M. Matui, eds.), LNCS, vol. 2355, pp. 336–350, Springer-Verlag, 2002.
12. National Institute of Standards and Technology, " Announcing request for candidate algorithm nominations for a new Cryptographic hash algorithm (SHA-3) family, " Federal Register Notice, vol. 72(212), pp. 62212-62220, 2007.
13. X. Wang and H. Yu, "How to break MD5 and other hash functions." in *Proceedings of Eurocrypt'05* (R. Cramer, ed.), LNCS, vol. 3494, pp. 19–35, Springer-Verlag, 2005.
14. X. Wang, Y. L. Yin, and H. Yu, "Finding collisions in th full SHA-1." in *Proceedings of Crypto'05* (V. Shoup, ed.), LNCS, vol. 3621, pp. 17–36, Springer-Verlag, 2005.

A Differential Path in the Main Loop

We show the differential path in the main loop in Table 8. The first column is the step number of the main loop, and from the 2nd to the 9th columns indicate positions of difference bit in the words used for updating A and B at the i-th step. The 11th and the 12th columns indicate positions of difference bit in the words updated at the i-th step. The 13th column is probability of difference propagation at the i-th step.

Table 8. Differential Path with High Probability in the Main loop

i	$A[i]$	$A[i+11]$	$C[i-8]$	$M[i]$	$B[i+13]$	$B[i+9]$	$B[i+6]$	$B[i]$	$A[i+12]$	$B[i+16]$	Probability
0	30	-	-	31	29,30	14,28,29	13,14,28,29	29	-	30	2^{-5}
1	31	-	-	-	12,15	12,14	-	12,29,30	15,31	13,15,30	2^{-2}
2	16,28,29	15,31	-	-	28	-	11	14,15	14,16	14,15	2^{-9}
3	-	14,16	-	-	30	-	14,28,29	27	-	28	2^{-6}
4	13	-	-	-	13,15,30	29,30	12,14	11,13,28	12,14	29	2^{-6}
5	14,15,27	12,14	-	-	14,15	12,15	-	15,28,30	16,31	29	2^{-7}
6	16	16,31	-	-	28	28	-	13,14,28,29	14,16,29,30	-	2^{-7}
7	12,16	14,16,29,30	-	-	29	30	29,30	-	-	-	2^{-9}
8	29	-	31	-	29	13,15,30	12,15	11	12	-	2^{-6}
9	27,29	12	-	-	-	14,15	28	14,28,29	15	29,30	2^{-5}
10	-	15	-	-	-	28	30	12,14	-	13,15	2^{-4}
11	-	-	-	-	-	29	13,15,30	-	-	-	2^{-4}
12	-	-	-	-	29,30	29	14,15	-	30	30	2^{-3}
13	15,31	30	-	-	13,15	-	28	29,30	31	30	2^{-7}
14	14,16	31	-	-	-	-	29	12,15	15,16	13,15	2^{-6}
15	-	15,16	-	-	30	-	29	28	29	-	2^{-3}
16	12,14	29	-	31	30	29,30	-	30	31	-	2^{-4}
17	16,31	31	-	-	13,15	13,15	-	13,15,30	14,16,31	-	2^{-8}
18	14,15,29,30	14,16,31	-	-	-	-	-	14,15	15,16	-	2^{-6}
19	-	15,16	-	-	-	30	29,30	28	29	-	2^{-4}
20	12	29	-	-	-	30	13,15	29	30	-	2^{-9}
21	15	30	-	-	-	13,15	-	29	-	30	2^{-8}
22	-	-	-	-	-	-	30	-	-	-	2^{-1}
23	-	-	-	-	-	-	30	-	-	-	2^{-1}
24	30	-	31	-	30	-	13,15	-	-	-	2^{-3}
25	31	-	-	-	-	-	-	29,30	31	30	1
26	15,16	31	-	-	-	-	-	13,15	14,16	-	2^{-5}
27	29	14,16	-	-	-	-	-	-	-	-	2^{-1}
28	31	-	-	-	30	30	-	30	31	-	2^{-1}
29	14,16,31	31	-	-	-	-	-	30	31	-	2^{-2}
30	15,16	31	-	-	-	-	-	13,15	14,16	-	2^{-5}
31	29	14,16	-	-	-	-	30	-	-	-	2^{-2}
32	30	-	-	31	-	30	-	-	-	-	2^{-2}
33	-	-	-	-	-	-	-	-	-	-	1
34	-	-	-	-	-	-	-	-	-	-	1
35	-	-	-	-	-	-	30	-	-	-	2^{-1}
36	-	-	-	-	-	-	-	-	-	-	1
37	31	-	-	-	-	-	-	30	31	-	1
38	14,16	31	-	-	-	-	-	-	-	-	2^{-2}
39	-	-	-	-	-	-	-	-	-	-	1
40	31	-	31	-	-	-	-	-	-	-	1
41	31	-	-	-	-	-	-	30	31	-	1
42	14,16	31	-	-	-	-	-	-	-	-	2^{-2}
43	-	-	-	-	-	-	-	-	-	-	1
44	-	-	-	-	-	-	-	-	-	-	1
45	-	-	-	-	-	-	-	-	-	-	1
46	-	-	-	-	-	-	-	-	-	-	1
47	-	-	-	-	-	-	-	-	-	-	1

B Sufficient Conditions in the First 16 steps of the Main Loop

We show sufficient conditions in the first 16 steps of the main loop in the Table 9.

In Table 9, $A[i]_j$ denotes the j-th bit of $A[i]$, thus $A[i]_{31}$ is MSB of $A[i]$. $f_i(x[j])$, $g_i(x[j])$ and $X[i]$ are defined as follows:

$$\begin{aligned} f_i(x[j]) &= maj(x[j]_{i-1}, x[j]_{i-3}, f_{i-1}(x[j])), \\ g_i(x[j]) &= maj(x[j]_{i-1}, x[j]_{i-2}, g_{i-1}(x[j])), \\ X[i] &= A[i] \oplus (5 \times (A[i+11] \lll 15)) \oplus C[8-i], \end{aligned}$$

where $maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$, and $X[i]$ is the intermediate value used for $\times 3$. $f_i(x[j])$ and $g_i(x[j])$ imply carry of i-th bit in computations of $\times 5$ $(= (x \ll 2) + x)$ and $\times 3$ $(= (x \ll 1) + x)$, respectively.

Table 9. Sufficient Conditions in the First 16 steps of the Main Loop

i	5	3	AND
0	-	$X[0]_{28} = X[0]_{29}$	$B[6]_{14} = B[9]_{14}, B[6]_{28} = B[9]_{28}$ $B[6]_{29} \neq B[9]_{29}, B[9]_{13} = 0$
1	-	-	$B[7]_{12} = 0, B[7]_{14} = 1$
2	$f_{14}(A[13]) = A[13]_{12}, f_{16}(A[13]) = A[13]_{16}$ $f_{30}(A[13]) \neq A[13]_{28}$	$g_{14}(X[2]) \neq X[2]_{13}, X[2]_{16} = X[2]_{15}$ $X[2]_{26} = X[2]_{27}, X[2]_{28} \neq X[2]_{29} \neq X[2]_{30}$	$B[11]_{11} = 0$
3	$f_{29}(A[14]) = A[14]_{27}$	$X[3]_{27} = X[3]_{28} = X[3]_{30}$	$B[12]_{14} = B[12]_{28} = 0$ $B[12]_{29} = 1$
4	-	$g_{13}(X[4]) \neq X[4]_{12}, X[4]_{15} = X[4]_{14}$	$B[10]_{30} = 0, B[10]_{29} = 1$ $B[13]_{12} = B[13]_{14} = 1$
5	$f_{27}(A[16]) = A[16]_{25}, A[16]_{29} \neq A[16]_{27}$	$X[5]_{12} = X[5]_{13}, X[5]_{15} \neq X[5]_{14}$ $X[5]_{13} = X[5]_{16}$	$B[11]_{12} = 1, B[11]_{25} = 0$
6	$f_{14}(A[17]) = A[17]_{12}, A[17]_{16} \neq A[17]_{16}$	$g_{14}(X[6]) \neq X[6]_{13}, X[6]_{16} \neq X[6]_{15}$ $X[6]_{27} = X[6]_{28}, X[6]_{28} \neq X[6]_{30}$	$B[12]_{28} = 0$
7	$f_{12}(A[18]) \neq A[18]_{10}, f_{13}(A[18]) = A[18]_{13}$ $A[18]_{16} = A[18]_{14}, f_{29}(A[18]) = A[18]_{27}$ $f_{30}(A[18]) = A[18]_{30}$	$X[7]_{27} = X[7]_{28} = X[7]_{30}$	$B[13]_{30} \neq B[16]_{30}, B[16]_{29} = 0$
8	-	$X[8]_{27} = X[8]_{28}, X[8]_{28} \neq X[8]_{30}$	$B[14]_{15} \neq B[17]_{15}, B[14]_{30} = 0$ $B[14]_{13} = 1, B[17]_{12} = 1$
9	$f_{27}(A[20]) = A[20]_{25}, f_{29}(A[20]) = A[20]_{29}$	-	$B[15]_{14} = 1, B[15]_{15} = 0$ $B[18]_{28} = 0$
10	$f_{30}(A[21]) = A[21]_{28}$	$X[10]_{28} = X[10]_{29}$	$B[16]_{28} = 1, B[19]_{30} = 1$
11	-	-	$B[17]_{29} = 1, B[20]_{13} = 0$ $B[20]_{15} = 0, B[20]_{30} = 0$
12	-	-	$B[18]_{29} = 0, B[21]_{14} = 0$ $B[21]_{15} = 0$
13	$f_{13}(A[24]) \neq A[24]_{11}, A[24]_{14} \neq A[24]_{12}$ $f_{15}(A[24]) = A[24]_{15}$	$X[13]_{11} = X[13]_{12}, X[13]_{13} \neq X[13]_{14}$ $X[13]_{13} = X[13]_{15}$	$B[22]_{28} = 0$
14	$f_{14}(A[25]) \neq A[25]_{12}, A[25]_{15} \neq A[25]_{13}$ $f_{16}(A[25]) = A[25]_{16}$	$X[14]_{13} = X[14]_{14} = X[14]_{16}$	$B[23]_{29} = 0$
15	$f_{30}(A[26]) \neq A[26]_{28}$	$X[15]_{28} = X[15]_{29}$	$B[24]_{29} = 1$

C Final Loop for $r = 8$

Here we consider the final loop of $r = 8$. If a different value of r is used, the final loop is changed. In 4.2.6 of [5], the method of the modification of the the final loop are described.

Since the final loop adds C to A, it is expressed as $A + \sigma(C)$, where $\sigma(C)$ is the vector including C. $\sigma(C)$ is computed by a following simple loop of the form:
For i from 0 to s - 1,

$$\sigma(C)[i \bmod r] \leftarrow \sigma(C)[i \bmod r] + C[(-1)^e + \mathit{offset} \bmod r],$$

e, s and *offset* are chosen to satisfy the condition. For $r = 12$, $e = 0$, $s = 36$ and $\mathit{offset} = 3$ are chosen and $\sigma(C)[i]$ is as follows:

$$\sigma(C)[i] = C[i+3] + C[i+3+12] + C[i+3+24].$$

In the case of $r = 12$, each $A[i]$ depends on three words of C that are different for different i.

However, the final loop of $r = 8$ is not defined explicitly. We choose $e = 0$ and $\mathit{offset} = 3$ that are same as $r = 12$. We also choose $s = 24$ to satisfy that each $A[i]$ depends on three words of C that are all difference for the difference i. This is the same condition for $r = 12$.

Thus we define $\sigma(C)[i]$ of $r = 8$

$$\begin{aligned}\sigma(C)[i] &= C[i+3] + C[i+3+8] + C[i+3+16],\\ &= 2C[i+3] + C[i+11].\end{aligned}$$

In the final loop, the two same word of C are used. For $r = 8$, if $s \geq 24$, surely, same word are used with any e and *offset*. Thus, we consider that the final loop in this section is chosen in a natural manner.