Distinguisher for Shabal's Permutation Function

Peter Novotney
peternov@microsoft.com

July 20, 2010

Abstract

In this note we consider the Shabal permutation function $\mathcal{P}$ as a block cipher with input A_p,B_p and key C,M and describe a distinguisher with a data complexity of 2^{23} random inputs with a given difference. If the attacker can control one chosen bit of B_p, only 2^{21} inputs with a given difference are required on average. This distinguisher does not appear to lead directly to an attack on the full Shabal construction.

1 Introduction

The Shabal hash function [4] is a second round candidate in NIST's SHA-3 hash function competition. Shabal uses a iterated hash mode built around a keyed permutation function $\mathcal{P}$, which takes as input A_p,B_p and takes as a key C,M. In this note we will demonstrate that given an unknown key C,M, we can distinguish the permutation function $\mathcal{P}$ with known input differences with respect to XOR on A_p and B_p. Others have noted various distinguishers in the Shabal permutation function as well: In [2] the non-ideal behavior of Shabal's permutation function using a cube tester is described. Fixed points and key collisions of the permutation are described in [6]. A related key distinguisher is given in [3], and [1] presents a distinguisher based on rotational differences. In [5] the authors of Shabal respond to some of these papers. The distinguisher in this note seems to add its own unique features to those referenced above.

2 The Shabal Permutation Function

We use a slightly different description of the Shabal permutation function than given in [4]. The description below retains intermediate values, allowing them to be uniquely referenced in the differential description in section 3. Our description assumes the default tunable parameters $(p, r) = (3, 12)$ as defined in [4].

The Shabal permutation function takes 4 inputs A_p, B_p, C, and M, and gives as output A_c and B_c. We will consider A_p and B_p as the plaintext and M and C as the key. A_p contains 12 words and B_p, C and M each contain 16 words, where words are 32 bits. All additions and multiplications are mod 2^{32}. $\mathcal{P}$ is given as:

First we initialize the intermediate arrays with the input values:

```
$i \to 0 \ldots 11$
    $A[i] := A_p[i]$
$i \to 0 \ldots 15$
    $B[i] := B_p[i] \lll 17$
```

Main computation of the permutation:

```
$i \to 0 \ldots 47$
    $a_i := 5(A[11+i] \lll 15) \oplus A[i]$
    $k_i := 3(a_i \oplus C[8-i \bmod 16]) \oplus M[i \bmod 16]$
    $b_i := B[13+i] \oplus (B[9+i] \wedge \overline{B[6+i]})$
    $f_i := k_i \oplus b_i$
    $A[12+i] := f_i$
    $B[16+i] := \overline{f_i} \oplus (B[i] \lll 1)$
```

Perform the output whitening on A and copy result to output buffers:

```
$i \to 0 \ldots 11$
    $A_c[i] := A[i+48] + C[i+3] + C[i+15] + C[i+27]$
$i \to 0 \ldots 15$
    $B_c[i] := B[i+48]$
```

3 The Differential

The differential we analyze has a one bit difference in both A_p and B_p with respect to XOR and is given below:

$$\begin{aligned} \Delta A_p[10] &= \texttt{0x80000000h} \\ \Delta B_p[7] &= \texttt{0x00002000h} \end{aligned}$$

These differences are chosen such that they cancel each other out multiple times with high probability and remain unaffected as possible by the multiplication $\mathrm{mod} 2^{32}$, making it to round 26 of the permutation with a 1-bit difference with probability 1/8.

3.1 Following the Differential to Round 26

After the initial 17 bit rotations of the B values our differential is of the form

$$\begin{aligned} \Delta A[10] &= \texttt{0x80000000h} \\ \Delta B[7] &= \texttt{0x40000000h} \end{aligned}$$

From here we enter the main section of the permutation function. There are 48 total rounds counting from 0, so $i = 0...47$. The following rounds are those that involve the words with differences in A or B:

- **Round 1**: $b_1 = B[14] \oplus (B[10] \wedge \overline{B[7]})$, so $\Delta b_1 = 0$ when $B[10] \wedge \Delta B[7] = 0$ which occurs with probability 1/2. Note that we can set one bit in $B_p[10]$ appropriately so this condition is always met.

- **Round 7**: $B[23] := \overline{f_7} \oplus (B[7] \lll 1)$ so $\Delta B[23] =$ `0x80000000h`

- **Round 10**: we have $a_{10} := 5(A[9] \lll 15) \oplus A[10]$, so $\Delta a_{10} =$ `0x80000000h`. In the k_{10} step we multiple by three, but since the difference is in the highest order bit we have $\Delta k_{10} =$ `0x80000000h`. $b_{10} := B[23] \oplus (B[19] \wedge \overline{B[16]})$ so $\Delta b_{10} =$ `0x80000000h` and therefore the difference cancels at $f_{10} := k_{10} \oplus b_{10}$.

- **Round 14**: $b_{14} := B[27] \oplus (B[23] \wedge \overline{B[20]})$ where $\Delta B[23] =$ `0x80000000h`, so if $\Delta B[23] \wedge \overline{B[20]} = 0$ the difference cancels and we are left with $\Delta b_{14} = 0$. This occurs with probability 1/2.

- **Round 17**: $b_{17} := B[30] \oplus (B[26] \wedge \overline{B[23]})$ and again $\Delta B[23] =$ `0x80000000h` so if $B[26] \wedge \Delta B[20] = 0$ the difference cancels out and we have $\Delta b_{17} = 0$. This occurs with probability 1/2.

- **Round 23**: $B[39] := \overline{f_{23}} \oplus (B[23] \lll 1) =$ `0x00000001h`.

- **Round 26**: $b_{26} := B[39] \oplus (B[35] \wedge \overline{B[32]})$ and we end up with $\Delta A[38] =$ `0x00000001h` and $\Delta B[42] =$ `0x00000001h`.

From round 0 to round 26 the overall probability of hitting this one bit difference in $A[38]$ and $B[42]$ is 1/8 (1/4 if we are free to modify one bit of $B_p[10]$).

3.2 From Round 26

After round 26 the 1-bit differential begins to diffuse. However, due to there being only 6 rounds until the creation of the first output word $B_c[0] := B[48]$, the diffusion does not appear to be sufficient to remove biases in $\Delta B[48]$. Table 1 shows an example differential in $\Delta B[i+16]$ progressing from round $i = 26$ to $i = 32$. The bolded value shows the position of the original 1 bit differential considering the 15 bit rotation operation that occurs every round. In experimental data described below, this is the bit with the largest bias at the end of each round.

Example Differential	
Round	$\Delta B[i+16]$
i=26	0000000000000000000000000000000**1**
i=27	0000000000000111**1**000000000000000
i=28	1**1**000000000000000000000000111001
i=29	000000101111000001**1**0000000000001
i=30	001**1**0000001110110000111100001000
i=31	10000011110001110110**1**00101111010
i=32	01011**1**00000010110100010110000000

Table 1: Example differential in $\Delta B[i+16]$ for rounds $i = 26...32$

We measure the biases in $B_c[0]$ experimentally by the following procedure:

1. For $k = 1 \dots 2^{32}$:
    (a) Generate Random A_p, B_p, M, and C.
    (b) Set $A_p^{'} := A_p \oplus \texttt{0x80000000h}$
    Set $B_p^{'} := B_p \oplus \texttt{0x00002000h}$
    (c) $A_c, B_c :=$ Shabal-$\mathcal{P}(A_p, B_p, M, C)$
    (d) $A_c^{'}, B_c^{'} :=$ Shabal-$\mathcal{P}(A_p^{'}, B_p^{'}, M, C)$
    (e) Count value of each bit in $\Delta B_c[0] := B_c[0] \oplus B_c^{'}[0]$
2. Calculate bias of each bit in the 2^{32} samples of $\Delta B_c[0]$

With 2^{32} samples we can see that some bits are significantly biased. The results for some of the bits with the greatest bias are listed in Table 2.

Bit	Bias with Random Input	Bias after fixing $B_p[10]$
21	$\approx 2^{-13.9}$	$\approx 2^{-12.9}$
22	$\approx 2^{-13.8}$	$\approx 2^{-12.9}$
23	$\approx 2^{-14.7}$	$\approx 2^{-13.5}$
24	$\approx 2^{-14.0}$	$\approx 2^{-13.5}$
25	$\approx 2^{-12.9}$	$\approx 2^{-11.9}$
26	$\approx 2^{-11.2}$	$\approx 2^{-10.1}$

Table 2: Selection of Measured Bit Biases in $\Delta B_c[0]$

Given 2^{23} inputs with the given difference, we expect to be able to statistically distinguish the bias of bit 26. If we can fix the $B_p[10]$ value on the inputs we can distinguish with 2^{21} inputs.

4 Acknowledgments

Thank you to Anne Canteaut and the Shabal Team for taking the time to confirm the existence of this distinguisher on the inner permutation of Shabal. And thank you to Niels Ferguson, for the initial conversations that lead to this paper.

5 Conclusion

This distinguisher shows that one can skip large amounts of the mixing in $\mathcal{P}$ with a high probability given specific differences in the input. However, it does not seem possible to apply these biases to the full Shabal hash function since the IV is fixed, and multiple final iterations follow the last message block. While the difference given in this note was chosen to minimize the effects of the multiplication $\pmod{2^{32}}$ in the first 26 rounds, it seems possible that one could find other differences in A_p,B_p giving greater biases than seen here.

References

[1] Gilles Van Assche. A rotational distinguisher on shabal's keyed permutation and its impact on the security proofs. Available online, 2010.

[2] Jean-Philippe Aumasson. On the pseudorandomness of shabal's keyed permutation. Available online, 2009.

[3] Jean-Philippe Aumasson, Atefeh Mashatan, and Willi Meier. More on shabal's permutation. OFFICIAL COMMENT, 2009.

[4] Emmanuel Bresson, Anne Canteaut, Benot Chevallier-Mames, Christophe Clavier, Thomas Fuhr, Aline Gouget, Thomas Icart, Jean-Franois Misarsky, Mara Naya-Plasencia, Pascal Paillier, Thomas Pornin, Jean-Ren Reinhard, Cline Thuillet, and Marion Videau. Shabal, a submission to nists cryptographic hash algorithm competition. Submission to NIST, 2008.

[5] Emmanuel Bresson, Anne Canteaut, Benot Chevallier-Mames, Christophe Clavier, Thomas Fuhr, Aline Gouget, Thomas Icart, Jean-Franois Misarsky, Mara Naya-Plasencia, Pascal Paillier, Thomas Pornin, Jean-Ren Reinhard, Cline Thuillet, and Marion Videau. Indifferentiability with distinguishers: Why shabal does not require ideal ciphers. Cryptology ePrint Archive, Report 2009/199, 2009.

[6] Lars R. Knudsen, Krystian Matusiewicz, and Sren S. Thomsen. Observations on the shabal keyed permutation. OFFICIAL COMMENT, 2009.