

Differential and Invertibility Properties of BLAKE^{*}

Jean-Philippe Aumasson^{1,**,†}, Jian Guo^{2,***}, Simon Knellwolf^{3,†},
Krystian Matusiewicz^{4,‡}, and Willi Meier^{3,§}

¹ Nagravision SA, Cheseaux, Switzerland

² Nanyang Technological University, Singapore

³ FHNW, Windisch, Switzerland

⁴ Technical University of Denmark, Denmark

Abstract. BLAKE is a hash function selected by NIST as one of the 14 second round candidates for the SHA-3 Competition. In this paper, we follow a bottom-up approach to exhibit properties of BLAKE and of its building blocks: based on differential properties of the internal function G , we show that a round of BLAKE is a permutation on the message space, and present an efficient inversion algorithm. For 1.5 rounds we present an algorithm that finds preimages faster than in previous attacks. Discovered properties lead us to describe large classes of impossible differentials for two rounds of BLAKE's internal permutation, and particular impossible differentials for five and six rounds, respectively for BLAKE-32 and BLAKE-64. Then, using a linear and rotation-free model, we describe near-collisions for four rounds of the compression function.

Keywords: cryptanalysis, hash functions, SHA-3.

1 Introduction

BLAKE [2] is one of the 14 designs selected for the second round of the SHA-3 Competition organized by the U.S. National Institute of Standards and Technology. BLAKE uses HAIFA as [3] operation mode, with some simplifications. Its compression function is based on a keyed permutation that reuses internals of

* Supported in part by European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. A full version of this article appears in [1].

** Work done while this author was with FHNW, Switzerland, and supported by the Swiss National Science Foundation under project no. 113329.

*** The paper was partly done during the author's visit to Technical University of Denmark and was partly supported by a DCAMM grant there.

† Supported by Hasler Foundation <http://www.haslerfoundation.ch> under project number 08065.

‡ Supported by the grant from the Danish Research Council for Technology and Production Sciences number 274-07-0246.

§ Supported by GEBERT RÜF STIFTUNG, project no. GRS-069/07.

the stream cipher ChaCha [4]. Wordwise operations are integer addition, XOR, and rotation (AXR). Depending on the output length BLAKE works on 32-bit or 64-bit words. If necessary we refer to the specific instances by BLAKE-32 and BLAKE-64 respectively.

In a previous work, Ji and Liangyu [5] presented a preimage attack on round-reduced versions of BLAKE-32 and BLAKE-64 with up to 2.5 rounds (out of 10 and 14 respectively). In particular they described a method with complexity 2^{192} to find preimages of BLAKE-32 reduced to 1.5 rounds.

Contribution of this paper. We establish *differential properties* of the permutation used in the compression function of BLAKE and investigate *invertibility* of one and more rounds. Following a bottom-up approach, we first state differential properties of the core function G. We exploit them to show injectivity of one round of the permutation with respect to the message space. We derive explicit input-output equations for G, which yield an efficient algorithm to invert one round and an improved algorithm to find a preimage of 1.5 rounds (in 2^{128} for BLAKE-32). Then we exploit differential properties of G to find large classes of *impossible differentials* for one and two rounds, and specific impossible differentials for five and six rounds of BLAKE-32 and BLAKE-64 respectively. Using a linear and rotation-free model of G we find *near-collisions* for the compression function reduced to four rounds.

2 Preliminaries

This section describes the compression function of BLAKE and then fixes notations used in the rest of this paper. A complete specification of BLAKE can be found in [2].

2.1 The Compression Function of BLAKE

The compression function of BLAKE processes a 4×4 state of 16 words v_0, \dots, v_{15} . This state is initialized by a chaining value h_0, \dots, h_7 , a salt s_0, \dots, s_3 , a counter t_0, t_1 , and constants k_0, \dots, k_7 as depicted below:

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \leftarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ s_0 \oplus k_0 & s_1 \oplus k_1 & s_2 \oplus k_2 & s_3 \oplus k_3 \\ t_0 \oplus k_4 & t_0 \oplus k_5 & t_1 \oplus k_6 & t_1 \oplus k_7 \end{pmatrix}$$

The initial state is processed by 10 or 14 rounds for BLAKE-32 and BLAKE-64 respectively. A round is composed of a **column step**:

$$G_0(v_0, v_4, v_8, v_{12}) \quad G_1(v_1, v_5, v_9, v_{13}) \quad G_2(v_2, v_6, v_{10}, v_{14}) \quad G_3(v_3, v_7, v_{11}, v_{15})$$

followed by a **diagonal step**:

$$G_4(v_0, v_5, v_{10}, v_{15}) \quad G_5(v_1, v_6, v_{11}, v_{12}) \quad G_6(v_2, v_7, v_8, v_{13}) \quad G_7(v_3, v_4, v_9, v_{14}).$$

The G function depends on a *position index* $s \in \{0, \dots, 7\}$ (indicated as subscript), a *round index* $r \geq 0$, a *message block* m_0, \dots, m_{15} , and constants k_0, \dots, k_{15} . At round r of BLAKE-32, $G_s(a, b, c, d)$ computes

$$\begin{array}{l|l} 1 : a \leftarrow (a + b) + (m_i \oplus k_j) & 5 : a \leftarrow (a + b) + (m_j \oplus k_i) \\ 2 : d \leftarrow (d \oplus a) \ggg 16 & 6 : d \leftarrow (d \oplus a) \ggg 8 \\ 3 : c \leftarrow (c + d) & 7 : c \leftarrow (c + d) \\ 4 : b \leftarrow (b \oplus c) \ggg 12 & 8 : b \leftarrow (b \oplus c) \ggg 7 \end{array}$$

with $i = \sigma_r(2s)$ and $j = \sigma_r(2s + 1)$, where $\{\sigma_r\}$ is a family of permutations of $\{0, \dots, 15\}$. In BLAKE-64, the only differences—besides the word size—are the rotation constants, respectively set to 32, 25, 16, and 11.

For a fixed message block m , G is invertible and so a series of rounds is a permutation of the state. One may view the permutation as a block cipher with key m . After the 10 or 14 rounds the new chaining value h'_0, \dots, h'_7 is computed as

$$\begin{array}{l|l} h'_0 \leftarrow h_0 \oplus s_0 \oplus v_0 \oplus v_8 & h'_4 \leftarrow h_4 \oplus s_0 \oplus v_4 \oplus v_{12} \\ h'_1 \leftarrow h_1 \oplus s_1 \oplus v_1 \oplus v_9 & h'_5 \leftarrow h_5 \oplus s_1 \oplus v_5 \oplus v_{13} \\ h'_2 \leftarrow h_2 \oplus s_2 \oplus v_2 \oplus v_{10} & h'_6 \leftarrow h_6 \oplus s_2 \oplus v_6 \oplus v_{14} \\ h'_3 \leftarrow h_3 \oplus s_3 \oplus v_3 \oplus v_{11} & h'_7 \leftarrow h_7 \oplus s_3 \oplus v_7 \oplus v_{15} \end{array}$$

Observe that in the definition of G , we write the first line as “ $a \leftarrow (a+b)+(m_i \oplus k_j)$ ”, instead of “ $a \leftarrow a+b+(m_i \oplus k_j)$ ”. This is to avoid ordering ambiguities when computing probabilities of differential characteristics. For instance, a difference in m_i propagates through one addition in the former case, and through two additions in the latter, when interpreted as “ $a \leftarrow a + (b + (m_i \oplus k_j))$ ”, idem for the fifth line. Clearly, one can simultaneously use different characteristics in this model as being equivalent to a single characteristic in a model that does not make any assumption on the order of the operations.

2.2 Notations

The symbols \wedge and \vee denote logical AND and OR. Numbers in hexadecimal basis are written in typewriter (for example, ABCDEF01). A *difference* Δ always means a difference with respect to XOR, that is, two words m and m' have the difference Δ if $m \oplus \Delta = m'$. The Hamming weight of word m is denoted $|m|$, the Hamming weight of $(m \wedge 7FF \dots FF)$, that is, the Hamming weight of m excluding the most significant bit (MSB), is denoted $\|m\|$. A *differential characteristic* (DC) for BLAKE is the sequence of differences followed through application of addition, XOR, and rotation. In contrast a *differential* only consists in a pair of input and output differences.

When analyzing the differential behavior of the G function, we use the following notation:

- Δa : initial difference in a
- $\Delta \hat{a}$: difference in the intermediate value of a set at line 1
- $\Delta a'$: final difference in a
- Δ_i : difference in m_i

Analogous notations are used for differences in $b, c, d,$ and m_j . For instance, if $\Delta a = \Delta_i = 0$ and $\Delta b = 80 \cdots 00$, then $\Delta \hat{a} = 80 \cdots 00$.

3 Differential Properties of the G Function

This section enumerates properties of the G function. We first consider the case of differences in m_i and m_j only, and then consider the general case with input differences in the state. Finally we briefly look at the inverse of G.

3.1 Differences in the Message Words Only

All statements below assume zero input difference in the state words, that is, $\Delta a = \Delta b = \Delta c = \Delta d = 0$.

Proposition 1. *If $(\Delta_i = 0) \wedge (\Delta_j \neq 0)$, then $(\Delta a' \neq 0) \wedge (\Delta b' \neq 0) \wedge (\Delta c' \neq 0) \wedge (\Delta d' \neq 0)$.*

Proof. If there is no difference in m_i then there is no difference in $a, b, c,$ and d after the first four lines of G. Thus a difference Δ in m_j always gives a nonzero difference Δ' in a . Then, d always has a difference ($\Delta' \ggg 8$), which propagates to a nonzero difference Δ'' to c , and finally b has difference ($\Delta'' \ggg 7$). \square

Proposition 2. *If $\Delta_i \neq 0$, then*

$$\begin{array}{l|l} (\Delta a' = 0) \Rightarrow (\Delta d' \neq 0) & (\Delta c' = 0) \Rightarrow (\Delta b' \neq 0) \wedge (\Delta d' \neq 0) \\ (\Delta b' = 0) \Rightarrow (\Delta c' \neq 0) & (\Delta d' = 0) \Rightarrow (\Delta a' \neq 0) \wedge (\Delta c' \neq 0) \end{array}$$

Proof. We show that in the output, a and d cannot be both free of difference, idem for d and c , and for b and c . By a similar argument as in the proof of Proposition 1, after the first four lines of G the four state words have nonzero differences. In particular, the state has differences $(\Delta', \Delta'' \ggg 12, \Delta'', \Delta' \ggg 16)$, for some nonzero Δ' and Δ'' . Suppose that we obtain $\Delta a' = 0$. Then we must have $\Delta d' = (\Delta' \ggg 24)$. Hence a and d cannot be both free of difference. Similarly, cancelling the difference Δ'' in c requires a difference in d , thus c and d cannot be both free of difference. Finally, to cancel the difference in b, c must have a difference, thus b and c cannot be both free of difference. \square

Two corollaries immediately follow from Proposition 1 and Proposition 2.

Corollary 1. *If $(\Delta_i \vee \Delta_j) \neq 0$, then there are differences in at least two output words.*

Corollary 2. *All differentials with an output difference of one of the following forms are impossible:*

$$\begin{array}{cccc} (\Delta, 0, 0, 0) & (0, \Delta, 0, 0) & (\Delta, 0, 0, \Delta') & (\Delta, 0, \Delta', 0) \\ (0, 0, \Delta, 0) & (0, 0, 0, \Delta) & (\Delta, \Delta', 0, 0) & (0, \Delta, \Delta', 0) \end{array}$$

for some nonzero Δ and Δ' , and for any Δ_i and Δ_j .

Note that output differences of the form $(0, \Delta, 0, \Delta')$ are possible. For instance, if $\Delta_i = (\Delta_i \ggg 4)$, then the output difference obtained by linearization is $(0, \Delta_i \ggg 3, 0, \Delta_i)$. For such a Δ_i , highest probability 2^{-28} is achieved for $\Delta = 88888888$.

A consequence of Corollary 2 is that a difference in at least one word of m_7, \dots, m_{15} gives differences in at least two output words after the first round. This yields the following upper bounds on the probabilities of DCs.

Proposition 3. *A DC with input difference Δ_i, Δ_j has probability at most 2^{-1} if $(\Delta_i = 0) \wedge (\Delta_j \neq 0)$, at most 2^{-6} if $(\Delta_i \neq 0) \wedge (\Delta_j = 0)$ and at most 2^{-5} if $(\Delta_i \neq 0) \wedge (\Delta_j \neq 0)$.*

A proof is given in the full version of this article [1].

3.2 General Case

Statements below *no longer* assume zero input difference in the state words.

Proposition 4. *If $\Delta a' = \Delta b' = \Delta c' = \Delta d' = 0$, then $\Delta b = \Delta c = 0$.*

Proof. First, when $\Delta_i = \Delta_j = 0$, collisions do not exist since G is a permutation for fixed m_i and m_j . So we must have differences in m_i and/or m_j . By Proposition 6, in G^{-1} a difference in m_i and/or m_j cannot affect b and c , hence a collision for G needs no difference in b and c . □

In other words, a collision for G requires zero difference in the initial b and c . For instance, collisions can be obtained for certain differences $\Delta a, \Delta_i$, and zero differences in the other input words. Indeed at line 1 of the description of G , Δa propagates to $(a + b)$ with probability $2^{-\|\Delta a\|}$, Δ_i propagates to $(m_i \oplus k_j)$ with probability one, and finally Δa eventually cancels Δ_i . Note that a collision for G with difference 88888888 in both m_{11} and a is used in §6 to find near-collisions for a modified version of BLAKE-32 with 4 rounds.

The following result directly follows from Proposition 4.

Corollary 3. *The following classes of differentials for G are impossible:*

$$\begin{aligned} (\Delta, \Delta', \Delta'', \Delta''') &\mapsto (0, 0, 0, 0) \\ (\Delta, 0, \Delta'', \Delta''') &\mapsto (0, 0, 0, 0) \\ (\Delta, \Delta', 0, \Delta''') &\mapsto (0, 0, 0, 0) \end{aligned}$$

for nonzero Δ' and Δ'' , possibly zero Δ and Δ''' , and any Δ_i and Δ_j .

Many other classes of impossible differentials for G exist. For example, if $\Delta a' \neq 0$ and $\Delta b' = \Delta c' = \Delta d' = 0$, then $\Delta b = 0$.

Proposition 5. *The only DCs with probability one give $\Delta a' = \Delta b' = \Delta c' = \Delta d' = 0$ and have either*

- $\Delta_i = \Delta a = 800 \dots 00$ and $\Delta b = \Delta c = \Delta d = \Delta_j = 0$;
- $\Delta_j = \Delta a = \Delta d = 800 \dots 00$ and $\Delta b = \Delta c = \Delta_i = 0$;
- $\Delta_i = \Delta_j = \Delta d = 800 \dots 00$ and $\Delta a = \Delta b = \Delta c = 0$.

Proof. The difference (800...00) is the only difference whose differential probability is one. Hence probability-1 DCs must only have differences active in additions. By enumerating all combinations of MSB differences in the input, one observes that the only valid ones have either MSB difference in Δ_i and Δa , in Δ_j and Δa and Δd , or in Δ_i and Δ_j and Δd .

For constants k_i equal to zero, more probability-1 differentials can be obtained using differences with respect to integer addition.

3.3 Properties of G^{-1}

At round r , the inverse of G_s of BLAKE-32 computes

$$\begin{array}{l|l} 1 : b \leftarrow c \oplus (b \lll 7) & 5 : b \leftarrow c \oplus (b \lll 12) \\ 2 : c \leftarrow c - d & 6 : c \leftarrow c - d \\ 3 : d \leftarrow a \oplus (d \lll 8) & 7 : d \leftarrow a \oplus (d \lll 16) \\ 4 : a \leftarrow a - b - (m_j \oplus k_i) & 8 : a \leftarrow a - b - (m_i \oplus k_j) \end{array} \quad ,$$

where $i = \sigma_r(2s)$ and $j = \sigma_r(2s + 1)$. Unlike G , G^{-1} has low flow dependency: two consecutive lines can be computed simultaneously and independently, with concurrent access to one variable.

Many properties of G^{-1} can be deduced from the properties of G . For example, probability-1 DCs for G^{-1} can be directly obtained from Proposition 5. We report two particular properties of G^{-1} . The first one follows directly from the description of G^{-1} .

Proposition 6. *In G^{-1} , the final values of b and c do not depend on the message words m_i and m_j . In particular, b depends only on the initial b , c , and d .*

That is, when inverting G , *initial* b and c depend only on the choice of the image (a, b, c, d) , not on the message.

The following property follows from the observation in Proposition 3.





Proposition 7. *There exists no DC that gives collisions with probability one.*

Properties of G^{-1} are exploited in §4 to find impossible differentials.

4 Impossible Differentials

An *impossible differential (ID)* is a pair of input and output differences that cannot occur. This section studies IDs for several rounds of the permutation of BLAKE. First we exploit properties of the G function to describe IDs for one and two rounds. Then we apply a miss-in-the-middle strategy to reach up to five and six rounds.

To illustrate IDs we use the following color code:

	absence of difference
	undetermined (possibly zero) difference
	undetermined or partially determined nonzero difference
	totally determined nonzero difference

4.1 Impossible Differentials for One Round

The following statement describes many IDs for one round of BLAKE's permutation.

Proposition 8. *All differentials for one round (of any index) with no input difference in the initial state, any difference in the message block, and an output with difference in a single diagonal of one of the forms in Corollary 2, are impossible.*

Proof. We give a general proof for the central diagonal $(v_0, v_5, v_{10}, v_{15})$; the proof directly generalizes to the other diagonals of the state. We distinguish two cases:

1. *No differences are introduced in the column step:* the result directly follows from Proposition 4 and Corollary 2.
2. *Differences are introduced in the column step:* recall that if $\Delta b \neq 0$ or $\Delta c \neq 0$, then one cannot obtain a collision for G (see Proposition 4); in particular, if there is a difference in one of the two middle rows of the state before the diagonal step, then the corresponding diagonal cannot be free of difference after.

We reason ad absurdum: if a difference was introduced in the column step in the first or in the fourth column, then there must be a difference in the corresponding b or c (for output differences with $\Delta b' = \Delta c' = 0$ are impossible after the column step, see Corollary 2). That is, one diagonal distinct from the central diagonal must have differences.

We deduce that any state after one round with difference only in the central diagonal must be derived from a state with differences only in the second or in the third column. In particular, when applying G to the central diagonal, we have $\Delta a = \Delta d = 0$. From Proposition 2, we must thus have $\Delta a' \neq 0$, $\Delta c' \neq 0$, and $\Delta d' \neq 0$. In particular, the output differences in Corollary 2 cannot be reached.

We have shown that after one round of BLAKE, differences in the message block cannot lead to a state with only differences in the central diagonal, such that the difference is one of the differences in Corollary 2. The proof directly extends to any of the three other diagonals. \square

To illustrate Proposition 8, which is quite general and covers a large set of differentials, Fig. 1 presents two examples corresponding to the two cases in the proof.

Note that our finding of IDs with zero difference in the initial and in the final state is another way to prove Proposition 9.

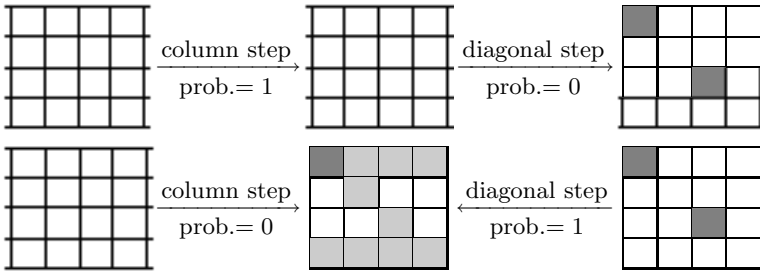


Fig. 1. Illustration of IDs after one round: when there is no difference introduced in the column step (top), and when there is one or more (bottom)

4.2 Extension to Two Rounds

We can directly extend the IDs identified above to two rounds, by prepending a probability-1 DC leading to a zero difference in the state after one round. For example, differences $800 \dots 00$ in m_0 and in v_0 always lead to zero-difference state after the first round.

By Proposition 8, a state with differences only in v_0 and v_{10} cannot be reached after one round when starting from zero-difference states. Therefore, differences $800 \dots 00$ in m_0 and v_0 cannot lead to differences only in v_0 and v_{10} after two rounds. This example is illustrated in Fig. 2.

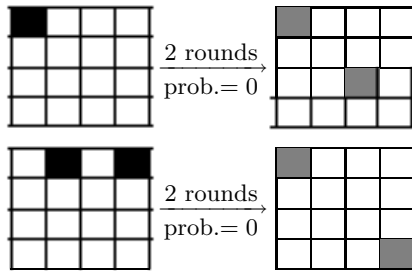


Fig. 2. Examples of IDs for two rounds: given difference $800 \dots 00$ in m_0 and v_0 (top), or in m_2, m_6, v_1, v_3 (bottom)

4.3 Miss in the Middle

The technique called *miss-in-the-middle* [6] was first applied to identify IDs in block ciphers (for instance, DEAL [7] and AES [8, 9]). Let $\Pi = \Pi_0 \circ \Pi_1$ be a permutation. A miss-in-the-middle approach consists in finding a differential $(\alpha \mapsto \beta)$ of probability one for Π_1 and a differential $(\gamma \mapsto \delta)$ of probability one for Π_0^{-1} , such that $\beta \neq \delta$. The differential $(\alpha \mapsto \delta)$ thus has probability zero and so is an ID for Π . The technique can be generalized to *truncated differentials*,

that is, to differentials β and δ that only concern a subset of the state. Below we apply such a generalized miss-in-the-middle to the permutation of BLAKE. We expose separately the application to BLAKE-32 and to BLAKE-64. The strategy is similar for both:

1. Start with a probability-1 differential with difference in the state and in the message so that difference vanish until the second round.
2. Look for bits that are changed (or not) with probability one after a few more rounds, given this difference.
3. Do same as step 2 in the backwards direction, starting from the final difference.

Good choices of differences are those that maximize the delay before the input of the first difference, more precisely, those such that the message word with the difference appears in the second position of a diagonal step forwards, and in the first position of a column step backwards. The goal is to minimize diffusion so as to maximize the chance of probability-1 truncated differentials.

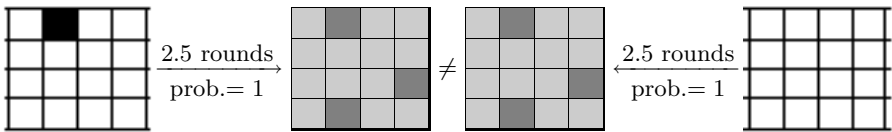


Fig. 3. Miss-in-the-middle for BLAKE-32, given the input differences 80000000 in m_2 and v_1 . The two differences in dark gray are incompatible, thus the impossibility. In the forward direction, 2.5 rounds are two rounds plus a column step; backwards, 2 inverse rounds plus an inverse diagonal step.

Application to BLAKE-32. We consider a difference 80000000 in the initial state in v_1 , and in the message block word m_2 ; we have that

- *Forwards*, differences in v_1 and m_2 cancel each other at the beginning of the column step and no difference is introduced until the diagonal step of the second round in which m_2 appears as m_j in G_5 ; after the column step of the third round (that is, after 2.5 rounds), we observe that bits¹ 35, 355, 439, and 443 are *always changed* in the state.
- *Backwards*, we start from a state free of difference, and m_2 introduces a difference at the end of the first inverse round, as it appears as m_i in the column step’s G_2 ; after 2.5 inverse rounds, we observe that bits 35, 355, 439, and 433 are *always unchanged*.

The probability-1 differentials reported above were first discovered empirically, and could be verified analytically by tracking differences, distinguishing bits with probability-1 (non-) difference, and other bits.

¹ Here, bit 35 is the fourth most significant bit of the second state word v_1 , bit 355 is the fourth most significant bit of v_{11} , etc.

We deduce from the observations above that difference 80000000 in v_1 and m_2 cannot lead to a state free of difference after five rounds. We thus identified a 5-round ID for the permutation of BLAKE-32. Fig. 3 gives a graphical description of the ID.

Application to BLAKE-64. For BLAKE-64, we follow a similar approach as for BLAKE-32, with MSB difference in m_2 and v_1 . We could detect contradictory probability-1 differentials over three instead of 2.5 rounds, both forwards and backwards. For example, we detected probability-1 inconsistencies for bits 450, 453, 457, 462, and 463 of the state. We obtain an ID for six rounds of the permutation of BLAKE-64.

Remarks

1. The probability-1 truncated differentials used above were empirically discovered, but one can easily verify them analytically. For instance, for bit 35 forward (fourth bit of v_1), we observe that the state is free of difference until the input of m_2 in the second round in G_5 , which sets a difference $\Delta = 80000000$ in v_1 , and other differences in v_6, v_{11}, v_{12} . At the next (third) round, when computing G_1 the only difference occurs in the MSB of v_1 , which gives difference $\Delta\hat{a} = \Delta$, $\Delta\hat{d} = \Delta \ggg 16$, $\Delta\hat{c}$ with no difference in the first 15 bits and a difference in the 16th, $\Delta\hat{b}$ with no difference in the first three bits and a difference in the fourth; thus we have $\Delta a'$ with no difference in the first three bits and a difference in the fourth, that is, the bit 35 of the state is always flipped after 2 rounds plus a column step. Similar verification can be realized for the backwards differentials.
2. The IDs presented in this section do not lead to IDs for the compression function. This is because a given difference in the output of the compression function can be caused by 2^{256} distinct differences in the final value of the permutation (for BLAKE-32).

5 Invertibility of a Round

Let f^r be the function $\{0, 1\}^{512} \times \{0, 1\}^{512} \rightarrow \{0, 1\}^{512}$, that for initial state v and message block m outputs the state after r rounds of the permutation of BLAKE-32. Non-integer round indices (for example $r = 1.5$) mean the application of $\lfloor r \rfloor$ rounds and the following column step. We write $f_v^r = f^r(v, \cdot)$ when considering f^r for a fixed initial state and respectively f_m^r when the message block is fixed. As noted above, f_m^r is a permutation for any message block m and any $r \geq 0$. In this section we use the differential properties of G to show that f_v^1 is also a permutation for any initial state v . Then we derive an efficient algorithm for the inverse of f_v^1 and an algorithm with complexity 2^{128} to compute a preimage of $f_v^{1.5}$ for BLAKE-32 (a similar method applies to BLAKE-64 in 2^{256}). This improves the round-reduced preimage attack presented in [5] (whose complexity was respectively 2^{192} and 2^{384} for BLAKE-32 and BLAKE-64).

5.1 A Round Is a Permutation on the Message Space

Proposition 9. *For any fixed state v , one round of BLAKE (for any index of the round) is a permutation on the message space. In particular, f_v^1 is a permutation.*

Proof. We show that if there is no difference in the state, any difference in the message block implies a difference in the state after one round of BLAKE. Suppose that there is a difference in at least one message word. We distinguish two cases:

1. *No differences are introduced in the column step:* there is thus no difference in the state after the column step. At least one of the message words used in the diagonal step has a difference; from Corollary 1, there will be differences in at least two words of the state after the diagonal step.
2. *Differences are introduced in the column step:* from Corollary 2, output differences of the form $(0, 0, 0, 0)$, $(\Delta, 0, 0, 0)$, $(0, 0, 0, \Delta)$, or $(\Delta, 0, 0, \Delta')$ are impossible. Thus, after the first column step, there will be a difference in at least one word of the two middle rows (that is, in v_4, \dots, v_{11}). These words are exactly the words used as b and c in the calls to \mathbf{G} in the diagonal step; from Proposition 4, we deduce that differences will exist in the state after the diagonal step, since $\Delta b = \Delta c = 0$ is a necessary condition to make differences vanish (see Proposition 4).

We conclude that whenever a difference is set in the message, there is a difference in the state after one round. \square

The fact that a round is a permutation with respect to the message block indicates that no information of the message is lost through a round and thus can be considered a strength of the algorithm. The same property also holds for AES-128. Note that Proposition 9 says nothing about the injectivity of f_v^r for $r \neq 1$.

5.2 Inverting One Round and More

Without loss of generality, we assume the constants equal to zero, that is, $k_i = 0$ for $i = 0, \dots, 7$ in the description of \mathbf{G} . We use explicit input-output equations of \mathbf{G} to derive our algorithms.

Input–output equations for \mathbf{G} . Consider the function \mathbf{G}_s operating at round r on a column or diagonal of the state respectively. Let (a, b, c, d) be the initial state words and (a', b', c', d') the corresponding output state words. For shorter notation let $i = \sigma_r(2s)$ and $j = \sigma_r(2s+1)$. Let $\hat{a} = a + b + m_i$ be the intermediate value of a set at line 1 of the description of \mathbf{G} . From line 2 we get $\hat{a} = (\hat{d} \lll 16) \oplus d$, where \hat{d} is the intermediate value of d set at line 2. From line 7 we get $\hat{d} = (d' \lll 8) \oplus a'$ and derive

$$a = (((d' \lll 8) \oplus a') \lll 16) \oplus d - b - m_i. \quad (1)$$

Below we use the following equations that can be derived in a similar way:

$$a = (((((((b' \lll 7) \oplus c') \lll 12) \oplus b) - c) \lll 16) \oplus d) - m_i - b \tag{2}$$

$$= a' - ((b' \lll 7) \oplus c') - m_j - b - m_i \tag{3}$$

$$b = (((b' \lll 7) \oplus c') \lll 12) \oplus (c' - d') \tag{4}$$

$$c = c' - d' - ((d' \lll 8) \oplus a') \tag{5}$$

$$= c' - d' - ((d \oplus (a + b + m_i)) \ggg 16) \tag{6}$$

$$d = (((d' \lll 8) \oplus a') \lll 16) \oplus (a' - ((b' \lll 7) \oplus c') - m_j) \tag{7}$$

$$a' = (((((((b' \lll 7) \oplus c') \lll 12) \oplus b) - c) \lll 16) \oplus d) + ((b' \lll 7) \oplus c') + m_j \tag{8}$$

$$b' = (((b \oplus (c' - d')) \ggg 12) \oplus c') \ggg 7 \tag{9}$$

$$d' = c' - c - ((d \oplus (a + b + m_i)) \ggg 16) \tag{10}$$

Observe that (1), (2) and (8) allow to determine m_i and m_j from (a, b, c, d) and (a', b', c', d') . Further, (4) and (5) imply Proposition 6.

We now apply these equations to invert f_v^1 and to find a preimage of $f_v^{1.5}(m)$ for arbitrary m and v . Denote $v^i = v_0^i, \dots, v_{15}^i$ the internal state after i rounds. Again, non-integer round indices refer to intermediate states after a column step but before the corresponding diagonal step. The state v^r is the output of $f_{v^0}^r$.

Inverting f_v^1 . Given v^0 and v^1 , the message block $m = (m_0, \dots, m_{15})$ with $f_{v^0}^1(m) = v^1$ can be determined as follows:

1. Determine $v_4^{0.5}, \dots, v_7^{0.5}$ using (4) and $v_8^{0.5}, \dots, v_{11}^{0.5}$ using (5).
2. Determine m_0, \dots, m_7 using (2), (8), and (10).
3. Determine $v_0^{0.5}, \dots, v_3^{0.5}, v_{12}^{0.5}, \dots, v_{15}^{0.5}$ using G_0, \dots, G_3 .
4. Determine m_8, \dots, m_{15} using (2), (8), and (10).

This algorithm always succeeds, as it is deterministic. Although slightly more complex than the forward computation of f_v^1 , it can be executed efficiently.

Preimage of $f_v^{1.5}(m)$. Given some v^0 , and $v^{1.5}$ in the codomain of $f_{v^0}^{1.5}$ (thus, a preimage of $v^{1.5}$ exists), a message block m with $f_{v^0}^{1.5}(m) = v^{1.5}$ can be determined as follows:

1. Guess m_8, m_{10}, m_{11} and $v_{10}^{0.5}$.
2. Determine v_4^1, \dots, v_7^1 using (4) and v_8^1, \dots, v_{11}^1 using (5), v_{12}^1, v_{13}^1 using (7).
3. Determine $v_6^{0.5}, v_7^{0.5}$ using (4), m_4 (2), v_1^1 (2), $v_{14}^{0.5}$ (6), $v_1^{0.5}$ (3), $v_{11}^{0.5}$ (5), $v_{12}^{0.5}$ (2).
4. Determine $v_2^{0.5}$ (5), m_5 (8), m_6 (2), v_{15}^1 (7), $v_{15}^{0.5}$ (6), $v_5^{0.5}$ (4), v_0^1 (5), m_9 (8), m_{14} (2).
5. Determine $v_3^{0.5}$ (5), m_7 (8), $v_0^{0.5}$ (2), $v_8^{0.5}$ (5), m_0 (1), v_2^1 (5), v_{14}^1 (2), m_{15} (8).
6. Determine $v_4^{0.5}$ (9), m_1 (8), $v_9^{0.5}$ (6), v_3^1 (8), m_{13} (2), m_2 (2), m_3 (8), $v_{13}^{0.5}$ (7), m_{12} (2).
7. If $f_{v^0}^{1.5}(m) = v^{1.5}$ output m , otherwise make a new guess.

This algorithm yields a preimage of $f_v^{1.5}(m)$ for BLAKE-32 after 2^{128} guesses in the worst case. It directly applies to find a preimage of the compression function of BLAKE reduced to 1.5 rounds and thus greatly improves the round-reduced preimage attack of [5] which has complexity 2^{192} . The method also applies to BLAKE-64, giving an algorithm of complexity 2^{256} , improving on [5]’s 2^{384} algorithm.

There are other possibilities to guess words of m and the intermediate states. But exhaustive search showed that at least four words are necessary to determine the full message block m by explicit input-output equations.

6 Near Collisions

In this section, we exploit linearization of the G function, that is, approximation of addition by XOR. This enables us to find near collisions for a variant of BLAKE-32 with four rounds.

6.1 Linearizing G

Observe that in G , the number of bits rotated are 16, 12, 8 and 7. Only 7 is not a multiple of 4.

The idea of our attack is to use differences that are invariant by rotation of 4 bits (and thus by any rotation multiple of 4), as 88888888, and try to avoid differences pass through the rotation by 7. We model the compression function in $\text{GF}(2)$, where a 1 denotes a difference in the register and 0 means no difference. We linearize the G function by replacing addition with XOR. Further we remove the rotations as the differences we choose are rotation invariant.

6.2 Differential Characteristic for Near Collisions

In our linearized model, we have 16 bits of message and 16 bits of chaining values, hence the search space is 2^{32} , which can be explored exhaustively.

We can further reduce the search space by the condition that no difference passes through rotation by 7 over four rounds of the compression function.

As the model is linear, the whole compression function can be expressed by a bit vector consisting of message and chaining value multiplied by a matrix. We used the program MAGMA to efficiently reduce the search space to 2^4 for a 4-round reduced compression function.

Linearizing a difference pattern 88888888 costs 2^7 for each addition. We aim to find those configurations which linearize the addition operation as little as possible. Note that by choosing proper chaining values and messages, we can get the first 1.5 rounds “for free”. We did the search, and the configuration with differences in m_{11} and $v_0, v_1, v_5, v_6, v_7, v_7, v_{10}, v_{11}, v_{13}, v_{15}$ and starting point at round 6 gives count 6 only. This gives complexity 2^{42} , with no memory requirements. This configuration gives after feedforward final differences in h'_1, h'_2, h'_5 , and h'_6 (assuming no difference in salts).

We thus obtain a near collision on $(256 - 40) = 216$ bits (note h'_5 contains 16 bits of difference). Figure 4 shows how differences propagate from round 6 to 9. We expect similar methods to apply to any sequence of four rounds, though with different complexities.

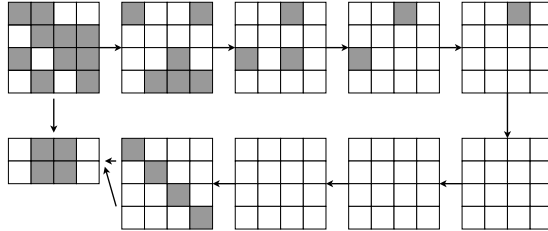


Fig. 4. Tracing the differences for near collisions on rounds 6 to 9

6.3 On the Extension to More Rounds

Consider the linearized model of G , in which we approximate addition by xor, and use the special difference 88888888 (so that differences do not propagate to the final b).

Consider a linearized round, as in §6.1. Since there are 16 chaining variables and 16 message words, hence we have 2^{16+16} different configurations. When we restrict “no difference in output b of G ”, the number of good configurations is reduced by a factor 2 when passing each G . Each round function has eight G ’s. Hence each round reduces the “good configurations” by a factor 2^8 . Thus, N rounds reduce the number of good configurations to $2^{32}/2^{8N} \geq 1$. Hence four seems to be the maximum possible number of rounds for which our method applies, which was verified by our program. This is also why we need to seek non-linear connectors to give collisions for more rounds.

The 4-round near collision applies to almost all 4-round (i.e. start with any round), however they give different complexity due to different counts for number of linearization, the round 6-9 gives lowest count 6.

7 Conclusion

We studied differential properties of the SHA-3 candidate BLAKE, and our main findings are

- *Differential properties* of BLAKE’s permutation and of its core function G .
- *Inversion algorithms* for one and 1.5 rounds of BLAKE’s round function for a fixed initial value.
- *Impossible differentials* for five (resp. six) rounds of BLAKE-32’s (resp. BLAKE-64’s) permutation.
- *Near-collisions* on four rounds of the compression function of BLAKE-32.

None of our observations seems to be a threat to the security of BLAKE.

Future work may address properties related to additive differences, instead of XOR differences. Our results may also assist cryptanalysis of the stream ciphers Salsa20 and ChaCha, on which BLAKE is based.

Acknowledgments

Jian Guo is supported by the Singapore Ministry of Education under Research Grant T206B2204.

References

1. Aumasson, J.P., Guo, J., Knellwolf, S., Meier, W., Matusiewicz, K.: Differential and invertibility properties of BLAKE (full version). Cryptology ePrint Archive, Report 2010/043 (2010)
2. Aumasson, J.P., Henzen, L., Meier, W., Phan, R.C.W.: SHA-3 proposal BLAKE. Submission to the SHA-3 Competition (2008)
3. Biham, E., Dunkelman, O.: A framework for iterative hash functions - HAIFA. Cryptology ePrint Archive, Report 2007/278 (2007)
4. Bernstein, D.J.: ChaCha, a variant of Salsa20, <http://cr.yp.to/chacha.html>
5. Ji, L., Liangyu, X.: Attacks on round-reduced BLAKE. Cryptology ePrint Archive, Report 2009/238 (2009)
6. Biham, E., Biryukov, A., Shamir, A.: Miss in the middle attacks on IDEA and Khufu. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 124–138. Springer, Heidelberg (1999)
7. Knudsen, L.R.: DEAL - a 128-bit block cipher. Technical Report 151, University of Bergen (1998); Submitted as an AES candidate
8. Jakimoski, G., Desmedt, Y.: Related-key differential cryptanalysis of 192-bit key AES variants. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 208–221. Springer, Heidelberg (2004)
9. Biham, E., Dunkelman, O., Keller, N.: Related-key impossible differential attacks on 8-round aes-192. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 21–33. Springer, Heidelberg (2006)