

On Free-Start Collisions and Collisions for TIB3

Florian Mendel and Martin Schl affer

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria.

Abstract. In this paper, we present free-start collisions for the TIB3 hash function with a complexity of about 2^{32} compression function evaluations. By using message modification techniques the complexity can be further reduced to 2^{24} . Furthermore, we show how to construct collisions for TIB3 slightly faster than brute force search using the fact that we can construct several (different) free-start collisions for the compression function. The complexity to construct collisions is about $2^{122.5}$ for TIB3-256 with similar memory requirements. The attack shows that compression function attacks have been underestimated in the design of TIB3. Although the practicality of the proposed attacks might be debatable, they nevertheless exhibit non-random properties that are not present in the SHA-2 family.

1 Introduction

A hash function maps an input of arbitrary finite length to an output of a fixed length. An important basic security requirement for a cryptographic hash function is its collision resistance – it should be computationally infeasible to find two different inputs, which hash to the same output. Recently, the collision resistance of many commonly used hash functions has been broken or doubted. Therefore, NIST has started the SHA-3 competition [7] to find a successor of the SHA-1 and SHA-2 hash functions. The cryptanalysis of the proposed SHA-3 candidates is of high importance to find a valuable hash function which is fast but still secure within the next few decades.

Many new and interesting hash functions have been proposed and some of these algorithms have a remarkable speed on certain platforms. The SHA-3 candidate TIB3 [6] is one of the fastest submissions with a speed of about 6-8 cycles/byte for all output sizes on 64-bit platforms [3]. The main design idea behind TIB3 is to use extensive parallelism by designing a “shorter” but “wider” compression function. To strengthen this short but fast compression function and to counter differential attacks, each message block is used in two subsequent compression function calls. However, in this paper we show that it is still possible to construct collisions for the hash function TIB3 below the generic complexity.

Using high-probability iterative characteristics, we can construct many practical free-start collisions for the compression function of TIB3 (Sect. 3). These free-start collisions are then used for the collision attacks on both TIB3-256 (Sect. 4) with a complexity slightly below the birthday bound. In the following section, we first give a short description of the hash function TIB3.

2 Description of TIB3

The hash function TIB3 is an iterated hash function based on the Merkle-Damg ard design principle [1,5]. The two main instances of TIB3 are called TIB3-256 and TIB3-512. TIB3-256 processes message blocks of 512 bits and produces hash values of 224 or 256 bits, while TIB3-512 processes message blocks of 1024 bits and produces hash values of 384 or 512 bits. If the message length is not a multiple of the block size, an unambiguous padding method is applied. For the description of the padding method we refer to [6]. Let $m = M_1 \| M_2 \| \dots \| M_t$ be a t -block message (after padding). Then, the hash value $h = H(m)$ is computed as follows:

$$\begin{aligned} H_0 &= IV_H, M_0 = IV_M \\ H_i &= f(H_{i-1}, M_i \| M_{i-1}) \quad \text{for } 1 \leq i \leq t \\ H_{t+1} &= f(H_t, 0 \| H_t \| M_t) = h \end{aligned}$$

where IV_H and IV_M are predefined initial values. The compression function f is used in Davies-Meyer mode [4] and consist of 2 parts: the key schedule and the state update transformation.

Key Schedule. The key schedule (or message expansion) of TIB3 takes as input the current and previous message block to compute a 4096-bit key for TIB3-256 and a 8192-bit key for TIB3-512. This key is split into 16 roundkeys k_j , where each roundkey is used in round j of the state update transformation. For a detailed description of the key schedule function we refer to [6], since we do not need it in our analysis. In the following, we describe the state update transformation for TIB3-256 and TIB3-512 in more detail.

State Update Transformation. The state update transformation of TIB3-256 starts from a (fixed) initial value IV_H of four 64-bit words and updates them in 16 rounds each. In each round one 256-bit roundkey k_j is used to update the four state variables A , C , E and G as follows:

$$\begin{aligned}
G &= G \oplus C \\
(A, C, E, G) &= (A, C, E, G) \oplus k_j \\
(A, C, E) &= Sbox(A, C, E) \\
G &= PHTX(G) \\
C &= PHTX(C) \\
A &= A \boxplus^{32} G \\
G &= E \boxplus^{32} G \\
(A, C, E, G) &= (C, E, G, A),
\end{aligned}$$

where $Sbox$ is a 3-bit S-box, $PHTX$ is a bit-mixing function and \boxplus^{32} denotes a 32-bit modular additions. For the definition of the S-boxes we refer to [6]. The function $O = PHTX(I)$ is defined as follows:

$$\begin{aligned}
T &= I + (I \ll 32) + (I \ll 47) \\
O &= T \oplus (T \gg 32) \oplus (T \gg 43)
\end{aligned}$$

After the last round of the state update transformation, the chaining values A_0, C_0, E_0, G_0 are XORed with the output values of the last round $A_{16}, C_{16}, E_{16}, F_{16}$ (feed-forward), resulting in the final value of one compression function f . For a detailed description of the hash function we refer to [6].

3 Free-Start Collisions for TIB3-256

In this section, we present a free-start collision attack on the compression function of TIB3-256 with a complexity of about 2^{24} compression function evaluations. Note that we use only differences in the chaining inputs and no differences in the message inputs are allowed. This is similar to the attack of den Boer and Bosselaers on MD5 [2]. However, in the case of TIB3 the complexity of the attack is much better due to its short compression function.

The attack is based on the fact that we can construct several 1-round iterative characteristics for the compression function of TIB3-256 with a probability between 2^{-2} and 2^{-4} , depending on the bit position of the differences. The 1-round characteristic is shown below:

$$(-, \Delta[i], \Delta[i], \Delta[i]) \rightarrow (-, \Delta[i], \Delta[i], \Delta[i]) \quad (1)$$

where $\Delta[i]$ denotes a difference at bit position i . By subsequently using this 1-round characteristic 16 times, we will get a free-start collision for the whole 16-round compression function of TIB3-256. Note that the differences of the last round in C_{16}, E_{16} and G_{16} will be canceled due to the feed-forward, *i.e.* $A_0 \oplus A_{16}, C_0 \oplus C_{16}, E_0 \oplus E_{16}$, and $G_0 \oplus G_{16}$.

Table 1. Differential probability for all non-zero input (S_i) to output (S_o) differences of the 3-bit S-box (*cf.* [6, page 15]). Probabilities are given in base 2 logarithms.

$S_i \setminus S_o$	1	2	3	4	5	6	7
1		-2	-2			-2	-2
2	-2		-2		-2		-2
3	-2	-2			-2	-2	
4				-2	-2	-2	-2
5		-2	-2	-2	-2		
6	-2		-2	-2		-2	
7	-2	-2		-2			-2

3.1 On the Probability of the Characteristic

Before we describe the probability of the 1-round characteristic in detail, we first have a look at the differential probabilities of the S-box. Table 1, shows the probabilities for all input/output differences of the 3-bit S-box of TIB3.

Table 2. Shows the differential characteristic for one round of TIB3. The output differences of the respective functions at bit position i of A, C, E and G are marked by “x”. Probabilities are in base 2 logarithms.

step	A	C	E	G	prob. for i at		
					32	64	else
$j - 1$		x	x	x			
xor		x	x				
rndkey		x	x				
sbox	x		x		-2	-2	-2
phtx							
add	x			x			-2
j		x	x	x			

Now, lets take a closer look at the probability of the characteristic for each round j which is shown in Table 2. Note that the xor of the roundkey in each round never changes the difference. In the following, we describe this 1-round characteristic in detail.

- We start with the differences $\Delta[i]$ in C , E and G . After the first xor operation, the difference in G is canceled. In order to guarantee that the characteristic holds, we need that the differences $\Delta[i]$ in C and E at the input of the S-box propagate to the differences $\Delta[i]$ in A and E after the S-box. This holds with a probability of 2^{-2} , see Table 1.
- Note that there are no differences in the *PHTX* functions.
- In the case of $i = \{32, 64\}$, no carry occurs in the four 32-bit modular additions and the differences $\Delta[i]$ in A and E propagate to $\Delta[i]$ in A , E and G with a probability of 1. In the case of $i \neq \{32, 64\}$ no carry occurs in the two additions with a probability of 2^{-2} .
- Hence, the resulting difference $\Delta[i]$ in C , E and G after one round is the same as in the input to this round.

The characteristic holds for one round with a probability of 2^{-2} for $i = \{32, 64\}$ and 2^{-4} for $i \neq \{32, 64\}$ and we get a characteristic for all 16 rounds with a probability of 2^{-32} and 2^{-64} , respectively. Thus, we can construct a free-start collision for the compression function of TIB3-256 with a complexity of about 2^{32} for $i = \{32, 64\}$ and 2^{64} for $i \neq \{32, 64\}$ instead of 2^{128} as expected for a compression function with 256 bits. An example for a free-start collision for TIB3-256 with $i = 64$ is given in Table 3.

Table 3. A free-start collision for TIB3-256 with differences at bit position 64.

H'_1		H''_1		ΔH_1	
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	80000000	00000000	80000000	00000000
00000000	00000000	80000000	00000000	80000000	00000000
00000000	00000000	80000000	00000000	80000000	00000000
M_1		M_2		$\Delta M_1, \Delta M_2$	
90BDD5C0	451CE787	E75BFF16	FACB4B84	00000000	00000000
6BB03ABE	8141141B	6D6A0C85	52A79F37	00000000	00000000
F45283B2	4019E54C	AECE5E32	A5F07508	00000000	00000000
68D47A8C	EC658400	A64F3E2B	E51D1923	00000000	00000000
20AC1B8D	5C4F42F0	E5079CCA	5CC28EBE	00000000	00000000
B239522C	8BF26045	1E7E2827	4E8C6B37	00000000	00000000
E0EC45C2	3ACE0DE7	808C0A2F	B5E1F9AA	00000000	00000000
2FB7DEBD	84DDCF10	3BBF29A5	FAB148DF	00000000	00000000
H'_2		H''_2		ΔH_2	
55F5547C	6AA5CC12	55F5547C	6AA5CC12	00000000	00000000
40831045	5CC5F776	40831045	5CC5F776	00000000	00000000
43E53C0C	4C64F862	43E53C0C	4C64F862	00000000	00000000
DD750B01	DA7AD37F	DD750B01	DA7AD37F	00000000	00000000

3.2 Improving the Attack Complexity

The complexity of the attack can be significantly improved by using message modification techniques. Message modification was introduced by Wang *et al.* in the cryptanalysis of MD5 and SHA-1 [9,10]. The idea of message modification is to use the degrees of freedom one has in the choice of the message words to fulfill conditions on the chaining variables. In the case of TIB3-256 we have 1024 bit input from two message blocks which can be used for message modification. It is easy to see from the message expansion (key schedule), that each of the message blocks can be used straight-forward to fulfill all conditions on the chaining variables in the first 4 rounds. In other words, we do not care about the probability of the characteristic in this part, since a message following the characteristic in the first 4 rounds can be found deterministically. Hence, the complexity of the attack can be reduced to 2^{24} for $i = \{32, 64\}$ and to 2^{48} for $i \neq \{32, 64\}$. We expect that the complexity can be further improved by using more sophisticated message modification techniques.

4 Collision Attack for TIB3-256

In this section, we show how the free-start collision attack on the compression function can be extended to a collision attack on the hash function. Even though the complexity of the attack is only slightly faster than a generic birthday attack, it exhibits some non-random properties that are not present in SHA-256. The attack uses the fact, that we can find several high-probability free-start collision producing characteristics for the compression function of TIB3-256.

4.1 Increasing the Number of Free-Start Collisions

In the previous section, we have constructed 64 different free-start collisions for $i = 1, \dots, 64$. To increase the number of characteristics, we can fit two high probability characteristics with bit position $i \neq j$ into the compression function:

$$(-, \Delta[i, j], \Delta[i, j], \Delta[i, j]) \rightarrow (-, \Delta[i, j], \Delta[i, j], \Delta[i, j])$$

In the case of $i \neq \{32, 64\}$, we get a total probability of 2^{128} which can be reduced to 2^{96} by message modification. Note that we can further increase the number of characteristics by allowing carries at the beginning (first rounds) and end (last rounds). Hence, we can construct at least 2^{11} different free-start collision characteristics.

4.2 From Free-Start Collisions to Collisions

In this section, we show how to use 2^x free-start collisions of the compression function to find collisions for the full hash function with a complexity of $2^{\frac{n-x}{2}}$. In the case of TIB3-256 we have constructed 2^{11} free-start collisions characteristics. Hence, the collision attack on TIB3-256 has a complexity of about $2^{122.5}$ compression function calls.

The collision attack uses 3 message blocks M_1 , M_2 and M_3 . The main idea of the attack is to find two different first message blocks M'_1 and M''_1 which result in one of the 2^{11} differences of the free-start collision in H_2 . Then, the respective free-start collision is used to get a collision in H_3 after the second compression function call. Note that we need a third message block M_3 for the message modification of the free-start collision:

$$\begin{aligned} H_1 &= f(IV_H, M_1 \| IV_M) \\ H_2 &= f(H_1, M_2 \| M_1) \\ H_3 &= f(H_2, M_3 \| M_2) \\ H_4 &= f(H_3, 0 \| H_3 \| M_3) = h \end{aligned}$$

The collision attack on TIB3-256 can then be summarized as follows:

1. Choose an arbitrary value for the message block M_2 .
2. Use a birthday attack to find a ΔH_2 (near-collision) which matches one of the 2^{11} free-start collision producing characteristics. Note that M_2 is fixed in the attack and only M_1 can be modified. This is important, since we do not allow any differences in M_2 . The birthday phase has a complexity of about $2^{\frac{256-11}{2}} = 2^{122.5}$ compression function evaluations.
3. Next, we use the respective free-start collision producing characteristic to turn the near-collision of ΔH_2 into a collision in H_3 by using the message blocks M_2 and M_3 . Note that there are no differences in these two message blocks, which is needed for the free-start collision producing characteristic to work (*cf.* Section 3). Note that M_3 can still be chosen freely in the attack and hence, used for message modification in the first 4 rounds. This step of the attack has a complexity of about 2^{96} compression function evaluations.

Alltogether, we can construct collisions in TIB3-256 with a complexity of about $2^{122.5}$ compression function evaluations and similar memory requirements. The complexity of this attack can be improved as soon as more than 2^{11} free-start collision characteristics have been constructed. One possibility to increase the number of characteristics is by allowing carries at the beginning (first rounds) and end (last rounds) of the compression function. Furthermore, it might be possible to reduce the memory requirements of our attack by using memoryless variants of the birthday attack [8] to find (specific) near-collisions.

5 Conclusion

In this paper, we have presented free-start collisions for TIB3 with a complexity of about 2^{32} compression function evaluations. By using message modification techniques the complexity can be reduced to 2^{24} . Furthermore, we can construct at least 2^{11} free-start collision producing characteristics for TIB3-256. We show how to use these free-start collisions to construct collisions slightly faster than brute force search. The attack has a complexity of about $2^{122.5}$ compression function evaluations and similar memory requirements. Memoryless variants to find (specific) near-collisions might be able to improve the memory requirements of this attack significantly.

TIB3 is one of the fastest submissions due to its parallelism but short compression function. In the design of TIB3, compression function attacks have been underestimated. In this paper, we have shown how to find high-probability free-start collisions and turn them into an attack on the hash function. Although the practicality of the proposed attacks might be debatable, they nevertheless exhibit non-random properties that are not present in the SHA-2 family. Since there is still room for improvements, this analysis can be a starting point for future attacks on TIB3.

References

1. Ivan Damgård. A Design Principle for Hash Functions. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 416–427. Springer, 1989.
2. Bert den Boer and Antoon Bosselaers. Collisions for the Compression Function of MD5. In Tor Helleseeth, editor, *EUROCRYPT*, volume 765 of *LNCS*, pages 293–304. Springer, 1993.
3. Ewan Fleischmann, Christian Forler, and Michael Gorski. Classification of the SHA-3 Candidates. Cryptology ePrint Archive, Report 2008/511, 2008. <http://eprint.iacr.org>.
4. Stephen M. Matyas, Carl H. Meyer, and Jonathan Oseas. Generating strong one-way functions with cryptographic algorithm. *IBM Technical Disclosure Bulletin*, 27(10A):5658–5659, 1985.
5. Ralph C. Merkle. One Way Hash Functions and DES. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 428–446. Springer, 1989.
6. Miguel Montes and Daniel Penazzi. The TIB3 Hash. Submission to NIST, 2008.
7. National Institute of Standards and Technology. Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family. Federal Register Notice, November 2007. Available online at: <http://csrc.nist.gov>.
8. Jean-Jacques Quisquater and Jean-Paul Delescaille. How Easy is Collision Search. New Results and Applications to DES. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 408–413. Springer, 1989.
9. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *LNCS*, pages 17–36. Springer, 2005.
10. Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 19–35. Springer, 2005.