On Pseudo-Collisions and Collisions for TIB3

Florian Mendel and Martin Schläffer

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

{florian.mendel,martin.schlaeffer}@iaik.tugraz.at

Abstract. In this paper, we present a pseudo-collision for TIB3 with a complexity of about 2^{32} compression function evaluations. By using message modification techniques the complexity can be further reduced. Furthermore, we show how to construct collisions for TIB3 slightly faster than brute force search using the fact that we can construct several (different) pseudo-collisions for the compression function. The complexity to construct collisions is about $2^{122.5}$ for TIB3-256 and $2^{244.5}$ for TIB3-512 with similar memory requirements. This attack shows that compression function attacks have been underestimated in the design of TIB3. Although the practicality of the proposed attacks might be debatable, it nevertheless exhibits non-random properties that are not present in the SHA-2 family and opens the possibility for further improved attacks.

1 Description of TIB3-256

The hash function TIB3 is an iterated hash function based on the Merkle-Damgård design principle. It processes message blocks of 512 bits and produces a hash value of 224 or 256 bits. If the message length is not a multiple of 512, an unambiguous padding method is applied. For the description of the adding method we refer to [2]. Let $m = M_1 \| M_2 \| \cdots \| M_t$ be a t-block message (after padding). The hash value $h = H(m)$ is computed as follows:

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

where H_0 and M_0 are predefined initial values. The compression function f basically consist of 2 parts: the state update transformation and the key schedule. In the following, we will describe both of them in more detail.

1.1 Key Schedule

The key schedule of TIB3-256 takes as input the current and previous message block (each 512 bits) to compute a 4096-bit key K. This key is split into 16 256-bit roundkeys k_i, where each roundkey k_i is used in one round of the state update transformation. For a detailed description of the key schedule function we refer to [2], since we do not need it for our analysis.

1.2 State Update Transformation

The state update transformation of TIB3-256 starts from a (fixed) initial value IV of four 64-bit words and updates them in 16 rounds each. In each round one 256-bit subkey k_i is used to update the four state variables A, C, E and G. One round of the TIB3-256 is shown in Figure 1.

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

where k is the current roundkey, $Sbox$ is a 3-bit S-box and $\boxplus^{32}$ two parallel 32-bit modular additions.

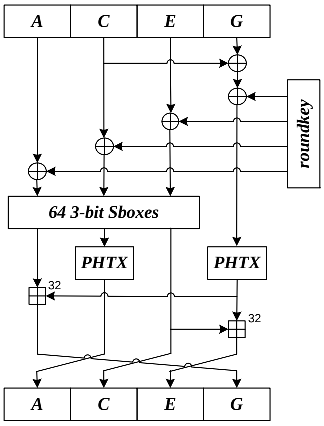

Fig. 1. One round of TIB3-256.

For the definition of the S-boxes we refer to [2]. The function $O = PHTX(I)$ is defined as follows:

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

After the last round of the state update transformation, the initial values A_0, C_0, E_0, G_0 and the output values of the last round $A_{16}, C_{16}, E_{16}, F_{16}$ are combined (xored), resulting in the final value of one iteration (feed forward). The result is the final hash value or the initial value for the next iteration. For a detailed description of the hash function we refer to [2].

2 Pseudo-Collision for TIB3-256

In this section, we show a pseudo-collision attack on the compression function of TIB3-256 with a complexity of about 2^{24} compression function evaluations. Note that we have only differences in the chaining inputs and that there are no differences in the message inputs similar to the attack of den Boer and Bosselaers on MD5 [1].

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

$$(-, \varDelta[i], \varDelta[i], \varDelta[i]) \rightarrow (-, \varDelta[i], \varDelta[i], \varDelta[i])$$

where $\varDelta[i]$ denotes a difference at bit position i. By using the 1-round characteristic 16 times, we will get a pseudo-collision for the compression function of TIB3-256 with a complexity of about $2^{2 \cdot 16} = 2^{32}$. 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.* $C_0 \oplus C_{16}$.

2.1 On the probability of the characteristic

Before describing 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/ouput differences of the 3-bit S-box of TIB3.

Table 1. Differential probability of the S-box (*cf.* [2, 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

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.

					prob. for i at		
step	A_i	C_i	E_i	G_i	32	64	else
r_0		x	x	x			
xor		x	x				
sbox	x		x		-2	-2	-2
phtx							
add	x			x			-2
r_1		x	x	x			

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

- We start with the differences $\Delta[i]$ in C_0, E_0 and G_0. After the first xor operation, the difference in G_0 is cancelled. Then, in order to guarantee that the characteristic holds we need the difference $\Delta[i]$ in C_0 and E_0 propagate to the differences $\Delta[i]$ in A_0 and E_0 after the S-box. This holds with a probability of 2^{-2}, see Table 1.
- Note that there are no diffences in the PHTX functions.
- In the case of $i = \{32, 64\}$ no carry occurs in the 4 32-bit modular additions and the differences $\Delta[i]$ in A_0 and E_0 propagate to $\Delta[i]$ in A_0, E_0 and G_0 with a probability of 1. If $i \neq \{32, 64\}$, the probability that no carry occurs in the additions is 2^{-2}.
- Hence, the resulting difference $\Delta[i]$ in C_0, E_0 and G_0 after one round is the same as 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}. Thus, we can construct a pseudo-collision for the compression function of TIB3-256 with a complexity of about 2^{32} instead of 2^{128} as expected for a compression function with 256 bits. An example for a pseudo-collision for TIB3-256 with $i = 64$ is given in Table 3.

2.2 Improving the attack complexity

The complexity of the attack can significantly improved by using message modification techniques. Message modification was introduced by Wang *et al.* in the cryptanalysis of MD5 and SHA-1 [3, 4]. 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 the message which can be used for message modification. It is easy to see from the message expansion,

Table 3. An example of a pseudo-collision for TIB3 with differences only in the MSB of C, E and G.

H_0	H_0^*	$\mathit{\Delta} H_0$
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	
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_{16}	H_{16}^*	$\mathit{\Delta} H_{16}$
55F5547C 6AA5CC12	55F5547C 6AA5CC12	00000000 00000000
40831045 5CC5F776	C0831045 5CC5F776	80000000 00000000
43E53C0C 4C64F862	C3E53C0C 4C64F862	80000000 00000000
DD750B01 DA7AD37F	5D750B01 DA7AD37F	80000000 00000000

that they can be used 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.

3 Collision Attack for TIB3-256

In this section, we show how the pseudo-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 pseudo-collision producing characteristics for the compression function of TIB3-256.

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

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

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 pseudo-collision characteristics.

We will use these 2^{11} characteristics to construct collisions for the hash function TIB3-256. The attack has a complexity of about $2^{122.5}$. It can be summarized as follows.

1. Choose an arbitrary value for M_2
2. Use a birthday attack to find a ΔH_2 (near-collision) which matches one of the 2^{11} pseudo-collision producing characteristics. The birthday phase has a complexity of about $2^{\frac{256-11}{2}} = 2^{122.5}$. 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 difference in M_2 for the next step of the attack.
3. Next we use the pseudo-collision producing characteristics to turn the near-collision for H_2 into a collision for H_3 by using an additional message block. Note that there are no differences in M_2 and M_3 which is needed for the pseudo-collision producing characteristic to work (*cf.* Section 2). Since M_3 can still be chosen freely in the attack and hence used for message modification in the first rounds, this step of the attack has a complexity of about 2^{24}.

Hence, we can construct a collision for TIB3-256 with a complexity of about $2^{122.5}$ instead of 2^{128} as expected for an hash functions with a 256 bit hash value.

4 Collision Attack for TIB3-512

The collision attack on TIB-256 can be extended to TIB3-512 as well. In TIB3-512, two instances of the TIB-256 compression function are computed in parallel. The two parallel instances are mixed by two PHTXD functions with inputs C, D and G, H. A short description of the state update of TIB3-512 is given below, for more details we refer to [2].

The state update transformation of TIB3-512 updates eight 64-bit words A, B, C, D, E, F, G and H in 16 rounds. One round of TIB3-512 is updated as follows:

$$\begin{aligned}
G &= G \oplus C \\
H &= H \oplus D \\
(A,B,C,D,E,F,G,H) &= (A,B,C,D,E,F,G,H) \oplus k \\
(A,C,E) &= Sbox(A,C,E) \\
(B,D,F) &= Sbox(B,D,F) \\
(G,H) &= PHTXD(G,H) \\
(C,D) &= PHTXD(C,D)
\end{aligned}$$

$$
\begin{aligned}
A &= A \boxplus G \\
B &= B \boxplus H \\
G &= E \boxplus G \\
H &= F \boxplus H \\
(A, B, C, D, E, F, G, H) &= (C, D, E, F, G, H, A, B)
\end{aligned}
$$

where k is the current roundkey, *Sbox* the same 3-bit S-box as in TIB3-256 and $\boxplus$ a 64-bit modular addition. The function $(O, P) = PHTXD(I, J)$ is a “double” version if PHTX and defined as follows:

$$
\begin{aligned}
P &= I \oplus J \\
P &= PHTX(P) \\
O &= I \oplus P \\
O &= PHTX(O)
\end{aligned}
$$

Since we do not have differences at the input of any PHTXD function, the pseudo-collisions of Section 2 can be used twice in parallel and we do not need to consider differences in PHTXD. Hence, we get two different independent 1-round characteristic for TIB-512 with differences in C, E, G:

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

and/or with differences in D, F, H:

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

where $\Delta[i]$ denotes a difference at bit position i. The complexity is 2^{32} for $i = 64$ and 2^{64} for $i \neq 64$ since one 64-bit addition is used instead of two 32-bit additions. TIB-512 has 16 rounds as well and we can construct a pseudo-collision for the compression function of TIB3-512 with a complexity of about $2^{2 \cdot 16} = 2^{32}$. An example for a pseudo-collision is given in Table 4.

In the case of TIB3-512, the generic complexity for a collision attack is 2^{256}. Therefore, we can easily fit up to 4 high probability characteristics next to each other (complexity 2^{256}) and reduce the costs by message modification to 2^{192}. Hence, we can construct at least 2^{23} different pseudo-collision characteristics with a complexity of less than 2^{192}. The resulting collision attack on TIB3-512 has a complexity of about $2^{\frac{512-23}{2}} = 2^{244.5}$.

5 Conclusion

In this paper, we presented a pseudo-collision for TIB3 with a complexity of about 2^{32} compression function evaluations. By using message modification techniques the

Table 4. An example of a pseudo-collision for TIB3-512 with differences only in the MSB of D, F and H.

H_0	H_0^*	ΔH_0
00000000 00000000	00000000 00000000	00000000 00000000
00000000 00000000	00000000 00000000	00000000 00000000
00000000 00000000	00000000 00000000	00000000 00000000
00000000 00000000	80000000 00000000	80000000 00000000
00000000 00000000	00000000 00000000	00000000 00000000
00000000 00000000	80000000 00000000	80000000 00000000
00000000 00000000	00000000 00000000	00000000 00000000
00000000 00000000	80000000 00000000	80000000 00000000
M_1	M_2	
246B6D96 2C90A727	240E562C 5C5D4627	00000000 00000000
6139BD35 C099E9CC	31C0A3B0 B3CC94A5	00000000 00000000
5533B6BF D6B80FB1	94E6BEBD 91BC6264	00000000 00000000
099868E2 8C9A5821	BB665DC4 B5C3E598	00000000 00000000
08ED963E A808F1E6	7AEFABF8 3DF12657	00000000 00000000
1658D8E1 94925F32	A4D3961F 2C8BFCF8	00000000 00000000
AF7DE86F 4013CAD4	626DED61 3B3BE4F7	00000000 00000000
24573C4C 867D59A2	873613B2 C1F4B14A	00000000 00000000
H_{16}	H_{16}^*	ΔH_{16}
8011137D 30451AA0	8011137D 30451AA0	00000000 00000000
5791600A B98C1C4A	5791600A B98C1C4A	00000000 00000000
60570740 31EEA496	60570740 31EEA496	00000000 00000000
31FB13D0 8A58960D	B1FB13D0 8A58960D	80000000 00000000
15C9B361 99054AB7	15C9B361 99054AB7	00000000 00000000
B6312CAB 57CF73AE	36312CAB 57CF73AE	80000000 00000000
C7055809 B6B3BB6A	C7055809 B6B3BB6A	00000000 00000000
422F8F0B 9DCCC9A4	C22F8F0B 9DCCC9A4	80000000 00000000

complexity can be significantly reduced to 2^{24}. Furthermore, we show how to construct collisions for TIB3 by using the fact that we can construct several pseudo-collision producing characteristics for the compression function. The complexity to construct collisions is slightly faster than brute force search and about $2^{122.5}$ for TIB3-256 and $2^{244.5}$ for TIB3-512 with similar memory requirements. Memoryless variants to find (specific) near-collisions might be able to improve the memory requirements.

In the design of TIB3 compression function attacks have been underestimated. Hence, we have been able to find pseudo-collisions and turn them into an attack on the hash function. Although the practicality of the proposed attacks might be debatable, it nevertheless exhibits non-random properties that are not present in the SHA-2 family. Since there is room for improvements, this analysis is a starting point for future attacks which is work in progress.

Acknowledgements

We thank the designers of TIB3 for useful discussions and pointing out a small error in a previous version of this paper.

References

1. Bert den Boer and Antoon Bosselaers. Collisions for the Compression Function of MD5. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *LNCS*, pages 293–304. Springer, 1993.
2. Miguel Montes and Daniel Penazzi. The TIB3 Hash. Submission to NIST, 2008.
3. 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.
4. 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.