NEW PSEUDO NEAR COLLISION ATTACK ON TIGER

Dibyendu Mallik and Debdeep Mukhopadhyay
Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur, India

Keywords: Hash function, Tiger, Collision attacks, Pseudo collision, Meet in the middle attack.

Abstract: Tiger is a cryptographic hash function created by Anderson and Biham in 1996 with hash value of 192 bits. Reduced round variants of Tiger have shown some weaknesses recently. Kelsey and Lucks have shown a collision attack on Tiger reduced to round 16 and 17. Mendel and Rijmen have found 1 bit pseudo near collision for full round Tiger. In this article we discover a new key schedule differential for Tiger which leads to the finding of message pairs for 1-bit pseudo near collision.

1 INTRODUCTION

Tiger is a cryptographic hash function proposed by Anderson and Biham in 1996 (Anderson and Biham, 1996). It has a block cipher like construction and operates on 512 bit messages, which serves as the key in the block cipher mode. In addition, there is a 192 bit Initial Values (IV). The final hash value is of 192 bit. There are 24 rounds in the hash function plus an additional feedforward step at the end. Recent works show well known hash function families, like MD4 (Rivest, 1990) can be subjected to collision attacks by the techniques proposed in (Yu and Wang, 2007; Dobbertin, 1998). However till now no collision attacks on full round Tiger have been discovered.

Tiger has drawn the attention of several researchers around the world and there has been some interesting works on collision attacks on reduced variants of the hash function. In FSE 2006 Kelsey and Lucks presented a collision attack on 16 and 17 rounds of Tiger, out of the total 24 rounds (Kelsey and Lucks, 2006). The complexity of the attack is 2^{44} evaluations of the compression function of Tiger. They also presented a 1 bit pseudo near collision for a variant of Tiger reduced to 20 rounds with a complexity of 2^{64}. In Indocrypt 2006 Mendel et al. presented collision attack on Tiger reduced to 19 rounds(Mendel et al., 2006). In Asiacrypt 2007, Mendel and Rijmen presented a pseudo collision attack on full round Tiger (Mendel and Rijmen, 2007). The complexity of the attack was around 2^{47} computations of the compression function of Tiger.

Contribution. In the present paper, we revisit the collision problem in Tiger Hash function and present some new results. We identify a new key schedule differential which leads to 1-bit pseudo near collision of full round Tiger. To find key schedule differential we have ensured that the message differences generated by the key scheduler, should have large number of zeros at the starting and final rounds, to reduce the number of conditions on the message bits in the first and last round. Hence the key schedule helps in easy finding of consistent colliding pairs, which are reported in the paper.

The organization of the paper is as follows:*Section 2* presents a high level description of the Tiger hash function. The attack strategy is described in *section 3*. *Section 4* details the 1-bit pseudo near collision attack on full round Tiger, while the work is concluded in *section 5*.

2 HIGH LEVEL DESCRIPTION OF TIGER HASH FUNCTION

Tiger has 2 distinct parts: a key schedule and a state update transformation. A detailed description of the function can be found in (Anderson and Biham, 1996). The notations followed in this article are: $\oplus$ (XOR), $\Delta^{\oplus}$ (XOR difference), Δ (Modular difference), + (modulo 2^{64} addition), * (modulo 2^{64} multiplication), - (modulo 2^{64} substraction), $\neg$ (bitwise NOT), I (2^{63}), I' (2^{40}), I'' (2^{81}).

Mallik D. and Mukhopadhyay D. (2010).
NEW PSEUDO NEAR COLLISION ATTACK ON TIGER.
In *Proceedings of the International Conference on Security and Cryptography*, pages 427-430
DOI: 10.5220/0002940104270430

2.1 State Update Transformation

The state update transformation of Tiger starts from an initial value of three 64 bit words and updates them in three passes of eight rounds each and after 24 rounds an additional feedforward round is added. For details of state update transformation see (Anderson and Biham, 1996)

2.2 Key Schedule

Each round of Tiger uses one message word X_i as its round key. The 512 bit message block is divided into 8 byte message $X_0,\ X_1,\ X_2,\ X_3,\ X_4,\ X_5,\ X_6,\ X_7$ which are used as the keys of the first 8 rounds. The remaining 16 rounds key are generated by an invertible operation. The *KeySchedule* operation modifies its input in two passes.

3 ATTACK STRATEGY

In this section we briefly discuss the attack strategy of Kelsey et. al. and Mendel to perform collision attacks on the Tiger hash function. However it may be noted that the final feedforward step is omitted in both cases. This attack strategy belongs to a general class of cryptanalysis as presented in (Kelsey and Lucks, 2006) and (Mendel and Rijmen, 2007). The attack can be stated as follows:

1. A differential characteristic in the key schedule which holds with high probability is found. In ideal case the probability should be 1.
2. Certain message bits are modified to obtain desired difference value in the Tiger state variables which can be canceled by the difference in message words in the subsequent rounds.

3.1 Finding a Good Differential Characteristic for Key Schedule of Tiger

The objective of this step is to find key schedules with certain desirable properties. We first observe possible differences in the state variables for the i^{th} step, which can be canceled by suitable message differences in steps i, $i+1$ and $i+2$. Such possible difference values are noted in table 1. The cancellation needs to take place at the final round of the hash function. The objective was that the message difference generated by the keyscheduler should have a large number of 0s at the starting and final rounds. To find such *good* differentials, we linearize the operations in $GF(2)$ and

Table 1: Canceling message and state variable differences.

State Variable Difference			Message Difference		
ΔA_{i-1}	ΔB_{i-1}	ΔC_{i-1}	ΔX_{i+2}	ΔX_{i+1}	ΔX_i
0	0	I	0	0	I
0	I	0	I	0	0
0	I	I	I	0	I
I	0	0	0	I	0
I	0	I	0	I	I
I	I	0	I	I	0
I	I	I	I	I	I

choose those which have a high probability of occurrence in the real hash function.

3.2 Message Modification by Meet In The Middle

For any i^{th} step of Tiger let the state variables be denoted by $A_{i-1}, B_{i-1}, C_{i-1}$ and the corresponding differential pairs by $A^*_{i-1}, B^*_{i-1}, C^*_{i-1}$. The modular difference $\Delta(A_{i-1}) = A^*_{i-1} - A_{i-1}$. Similarly, the other differences $\Delta(B_{i-1})$ and $\Delta(C_{i-1})$ are defined. We assume we are given all these values. Then the modular difference $\delta = \Delta(C_{i+1})$ (refer figure 1) can be forced to any desired value with probability $\frac{1}{2}$ using birthday attack. We try all 2^{32} possibilities of $B_{i+1}[even]$ to generate 2^{32} candidates for the $\Delta(even(B_{i+1}))$. Then we use meet in the middle approach to solve the following equation. The algorithm can be stated as follows.

1. Store the 2^{32} candidates for $\Delta odd\,(B_i)$ in a table.
2. For all 2^{32} candidates for $\Delta even\,(B_{i+1})$ test if $\Delta odd\,(B_i)$ exists with: $\Delta odd\,(B_i) = (\Delta even\,(B_{i+1}) + \delta) * mult^{-1} - \Delta\,(B_{i-1})$

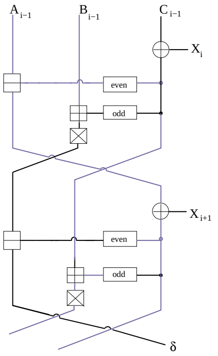

Figure 1: Meet in The Middle Attack.

This technique takes some 2^{32} evaluations of each functions even and odd and 2^{32} units of storage space. From our assumption, we know the value of C_{i-1}. Fixing the value of $X_i[even]$, the value of $C_i[even]$ gets known. Since the meet-in-the-middle

Table 2: Propagation of state variable.

step	State Variable			Message Difference
i	ΔA_i	ΔB_i	ΔC_i	$\Delta\left(X_{i+1}\right)$
-1	0	0	I	I
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	I+I'
4	*	$I+I'$	*	I
5	*	*	*	I''
6	*	*	*	0
7	*	*	*	I
8	*	*	*	0
9	*	*	*	I
10	*	*	$K^{\oplus}$	0
11	*	K^{+}	$L^{\oplus}$	I
12	0	L^{+}	I	0
13	0	I	0	0
14	I	0	0	0
15	0	0	I	I

attack produces ΔB_{i+1}(even) and ΔB_i(odd) satisfying, $X_i[odd] = C_{i-1}[odd] \oplus B_i[odd]$ and $X_{i+1}[even] = C_i[even] \oplus B_{i+1}[even]$ we are able to compute 64 local message bits.

4 1-BIT PSEUDO COLLISION ON FULL ROUND OF TIGER

In this section we present a pseudo near collision of full round Tiger including the feedforward round but for different key schedule differential as observed in (Mendel and Rijmen, 2007). Table 1 shows that message differences $(\Delta X_i, \Delta X_{i+1}, \Delta X_{i+2}) = (I, 0, 0)$ can be canceled out by introducing state variable difference $(0, 0, I)$ at $(i-1)^{th}$ step. Now introducing message difference I, 0, 0 at 16, 17, 18^{th} round and setting all the message differences of the final round of Tiger to 0, we obtain the following key schedule differential by running the inverse key schedule operation: $(I,0,0,0,I+I',I',I'',0) \rightarrow (I,0,I,0,0,0,0,0) \rightarrow (I,0,0,0,0,0,0,0)$ The key schedule differential holds with probability $\frac{1}{8}$ which is quite high and contains four and seven zeroes at the initial and final round respectively. Therefore, this differential follows the property of good differential as mentioned in section 4. In order to create collision after 23^{rd} round we have to create a collision after 16^{th} round of Tiger ie, $(\Delta A_{16}, \Delta B_{16}, \Delta C_{16}) = (0,0,0)$. Therefore the state variable difference at 15^{th} round should be $(0,0,I)$. Now if we generate state variable difference $(0,I,0)$ at 13^{th} round then the difference propagates to 15^{th} round and produces the desired difference. The state variable propagation is shown in table 2. To create $\Delta B_{13} = I$ we have to force ΔC_{12} to I and as $\Delta A_{13} = 0$, the difference $\Delta odd(B_{13}) = odd\,(B_{13}) - odd\,(B_{13} \oplus I)$ should be canceled out by creating its opposite modular difference L^{+} at B_{12}. In order to create the desired difference ΔB_{12} we force ΔC_{11} to a low weight Hamming difference $L^{\oplus}$ (Hamming weight of $L^{\oplus}$ is 10 to optimize the complexity of meet in the middle attack) by message modification technique such that the modular difference L^{+} and the XOR difference $L^{\oplus}$ are consistent. We use the same definition of consistency as mentioned in Mendel and Rijmen.(Mendel and Rijmen, 2007). Let L' be the set of all modular differences L^{+} which are consistent. For the propagation of desired difference from C_{12} to B_{13} the modular differences should be consistent. Therefore we construct the set L as follows:

$$L \quad = \quad \left\{L^{+} \in L' : L^{+} = odd\,(B_{13} \oplus I) - odd(B_{13})\right\}$$

The size of the set L is related to the Hamming weight of $L^{\oplus}$. It is observed from the round structure of Tiger that to get the desired difference $\Delta C_{13} = 0$, ΔA_{12} should be 0. Therefore from the construction of Tiger it is observed that the difference $\Delta odd(B_{12}) = odd(B_{12}) - odd(B_{12} \oplus L^{\oplus})$ should be canceled out by introducing an opposite modular difference K^{+} at B_{11} by forcing $\Delta^{\oplus} C_{10}$ to an XOR difference $K^{\oplus}$(of Hamming weight 8) which is consistent to K^{+}. We construct the set K from the set K' which is the set of all consistent modular differences K^{+} as follows:

$$K = \left\{K^{+} \in K' : K^{+} = odd(B_{12} \oplus L^{\oplus}) - odd(B_{12})\right\}$$

These sets are computed only one time in this attack. It has complexity of about 2^{33}.

4.1 Construction of Desired Differences

We have to know ΔA_8, ΔB_8 and ΔC_8 for the meet-in-the-middle attack to force ΔC_{10} to $K^{\oplus}$. For this we choose random values for B_3 and B_4 and compute $A_4 = (B_3 + odd(B_4)) * mult$, its corresponding differential A_4^{*} and $\Delta C_4 = \Delta even(B_4)$. Since there is no difference in X_3, B_3 and C_3 we get $\Delta(B_4) = I + I'$. Choosing random value for B_5 we can calculate the vales of all the state variables of step 5. Again choosing arbitrary values for X_6, X_7 and X_8 we calculate the values of A_8, B_8, C_8 and their corresponding differential values A_8^{*}, B_8^{*}, C_8^{*} are also calculated. These operations fix the message words X_5, X_6, X_7 and X_8.

As the values of ΔA_8, ΔB_8 and ΔC_8 are known now, desired XOR difference ΔC_{10}, can be constructed by using message modification technique. For all modular difference in K^{+} we do a message modification step and check if $\Delta(C_{10}) = K^{\oplus}$. This experiment holds with probability of 2^{-8}, as the Hamming weight of $K^{\oplus}$ is 8. The message modification technique holds with probability $\frac{1}{2}$ because of Birthday Paradox. Therefore, success probability of the attack is $\frac{1}{2} * (2^{-8}) * |K| = \frac{1}{2}$. This step has complexity of about 2^{41} computations. This step fixes the message words $X_{10}[even]$ and $X_9[odd]$.

Choosing some random values for $X_9[even]$(and its corresponding differential pair), ΔA_9, ΔB_9 and ΔC_9 which are needed for meet in the middle step to force the value of ΔC_{11} to $L^{\oplus}$ are calculated. We again apply message modification technique to construct the target difference. For all modular difference in L^{+} we do a message modification step and check if $\Delta(C_{11}) = L^{\oplus}$. This experiment holds with probability of 2^{-10}, as the Hamming weight of $L^{\oplus}$ is 10. The message modification technique holds with probability $\frac{1}{2}$. Therefore, success probability of the attack is $\frac{1}{2} * (2^{-10}) * |L| = \frac{1}{2}$. This step has complexity of about 2^{41} computations. This step fixes $X_{10}[odd]$ and $X_{11}[even]$.

To generate the message difference in 11^{th} round we again apply a message modification techniques. This step fixes the message word $X_{12}[even]$ and $X_{11}[odd]$. XOR difference and modular difference by I is interchangeable with probability 1 and the message modification step succeeds with $\frac{1}{2}$. Therefore this step succeeds with probability $\frac{1}{2}$ and complexity of $2^{36.5}$ evaluation of the compression function.(Mendel and Rijmen, 2007)

4.2 Constructing the Message Word

The attack fixes the words X_6 to X_{12} and $X_{13}[odd]$. To compute the values of the message word X_8 to X_{15} we choose random value for $X_{13}[even]$. From our known values we can calculate X_{14} and X_{15} from the following equations: $X_{14} = (X_6 - (X_{13} \oplus X_{12} \oplus (\neg(X_{12} + (X_{11} \oplus (\neg X_{10} \gg 23))) \gg 23))) + X_{13}$, and $X_{15} = (X_7 \oplus (X_{14} - X_{13})) - (X_{14} \oplus 0123456789ABCDEF) \gg 23))) + X_{13}$. After knowing the values of $X_8 \ldots X_{15}$ we run the inverse key schedule operation of Tiger to compute X_0 to X_7.

4.3 Constructing the Initial Values of State Variables

After knowing the values of X_0 to X_7 we can run Tiger rounds in backward direction to get the initial values. As the values of A_8, B_8, C_8 are known we can calculate the values of A_{-1}, B_{-1}, C_{-1} by backward propagation. To cancel out the message differences ΔX_0 we apply an initial value difference $\Delta C_{-1} = I$.

5 CONCLUSIONS

In this paper we have identified a new key schedule differential for the Tiger hash function. We have shown how the key schedule differentials can be applied to obtain a 1-bit pseudo near collision attack of complexity of $2^{47}$for full round Tiger. Finding of a new key schedule differential for 1-bit pseudo near collision attack shows security margins of Tiger is not as high as it was expected.

REFERENCES

Anderson, R. J. and Biham, E. (1996). Tiger: A fast new hash function. In Gollmann, D., editor, *FSE*, volume 1039 of *LNCS*, pages 89–97. Springer.

Dobbertin, H. (1998). Cryptanalysis of md4. *J. Cryptology*, 11(4):253–271.

Kelsey, J. and Lucks, S. (2006). Collisions and near-collisions for reduced-round tiger. In *FSE*, volume 4047 of *LNCS*, pages 111–125. Springer.

Mendel, F., Preneel, B., Rijmen, V., Yoshida, H., and Watanabe, D. (2006). Update on tiger. In *INDOCRYPT*, pages 63–79.

Mendel, F. and Rijmen, V. (2007). Cryptanalysis of the tiger hash function. In Kurosawa, K., editor, *ASIACRYPT*, volume 4833 of *LNCS*, pages 536–550. Springer.

Rivest, R. L. (1990). The md4 message digest algorithm. In Menezes, A. and Vanstone, S. A., editors, *CRYPTO*, volume 537 of *LNCS*, pages 303–311. Springer.

Yu, H. and Wang, X. (2007). Multi-collision attack on the compression functions of md4 and 3-pass haval. In Nam, K.-H. and Rhee, G., editors, *ICISC*, volume 4817 of *LNCS*, pages 206–226. Springer.