On the Security of the Encryption Mode of Tiger

Onur Özen, Kerem Varıcı
Institute of Applied Mathematics,
Middle East Technical University, Ankara, Turkey,
{e127740, e127761}@metu.edu.tr

Abstract—Tiger is an important type of an hash function producing 192-bit hash value from 512-bit message block and is proved to be secure so far as there is no known collision attack on the full (24 rounds) Tiger. It is designed by Biham and Anderson in 1995 to be very fast on modern computers, and in particular on the 64-bit computers, while it is still not slower than other suggested hash functions on 32-bit machines. Recently some weaknesses have been found for Tiger-hash function. First, in FSE’06 Kelsey and Lucks found a collision for 16-17 rounds of Tiger and a pseudo-near-collision for 20 rounds. Then Mendel et.al extended this attack to 19-round collision and 22-round pseudo-near-collision. Finally in 2007 Mendel and Rijmen found a pseudo-collision for the full Tiger. In this paper, we will investigate the security notion of reduced round Tiger in the encryption mode against the very well known and the efficient block cipher attacks, namely related-key boomerang and the related-key rectangle attacks. Moreover, we will present a trivial related-key boomerang and rectangle distinguishers of 20 and 22 rounds.

I. INTRODUCTION

Hash functions are one of the key elements of the cryptographic primitives which are being used for many important applications such as data integrity, authentication, digital signature etc. in everyday life. Many of the digital transactions and the e-cash applications are performed by using hash functions. Thus, the security and the effectiveness of the dedicated hash functions are of great interest nowadays.

Several cryptanalytic articles [1], [2] were published to find collisions for very well known hash functions. Especially the attacks proposed by Wang et.al [3], [4], [5] are very important attacks and many of the dedicated and widely used hash functions, such as members of MD and SHA families, were broken by the method proposed by Wang et.al.

Tiger which is another important type of an hash function and is proved to be secure so far as there is no known collision attack on the full Tiger. It is designed by Biham and Anderson in 1995 to be very fast on modern computers, and in particular on the 64-bit computers, while it is still not slower than other suggested hash functions on 32-bit machines. Recently some weaknesses have been found for Tiger-hash function. First, in FSE’06 [6] Kelsey and Lucks found a collision for 16-17 rounds of Tiger and a pseudo-near-collision for 20 rounds. Then Mendel et.al [7] extended this attack to 19-round collision and 22-round pseudo-near-collision. Finally in 2007 Mendel and Rijmen [8] found a pseudo-collision for the full Tiger.

There have been several cryptanalysis papers investigating the randomness properties of the designed hash functions under the encryption modes such as [9] by Kim et.al. In that paper, related-key boomerang and related-key rectangle attacks are performed on MD4, MD5 and HAVAL under 2, 4 related-keys or weak keys. Moreover, there have been very important attacks [10], [11], [12] on SHACAL as well which is based on the hash function SHA. In this paper, we will investigate the security notion of reduced round Tiger in the encryption mode against the very well known and the efficient block cipher attacks, namely related-key boomerang and the related-key rectangle attacks. Moreover, we will present a trivial related-key boomerang and rectangle distinguishers of 20 and 22 rounds.

The rest of the paper is structured as follows. In section two, we briefly introduce the hash function Tiger. In section three, the related-key boomerang and the related-key rectangle attacks are introduced. In section four and five, the attack on the encryption mode of the Tiger is detailed and section six concludes the paper.

II. TIGER

A. The Overview of Tiger

Tiger[13] is a cryptographic, iterative hash function which is designed for 64-bit processors by Biham and Anderson. It uses 64-bit operations such as addition, subtraction, multiplications by small constants (5, 7 and 9) and logical operations in message expansion. The main operation of Tiger is the use of even and odd functions operating on even and odd bytes of the step variables by the use of S-boxes. There exist four S-boxes in Tiger where each takes 8-bit input and produces 64-bit output. The size of the hash value and the intermediate state length are 192-bit (three 64-bit words) and the message block is 512-bit(eight 64-bit words). We will follow the notation given in Table 1.

B. The Round Function of Tiger

In Tiger, each 8-round part uses different constant values for multiplication and in each round a new expanded message word is used. Each 64-bit message words obtained from 512-bit message block are named as \(X_0, X_1, \ldots, X_7\). Four \(8 \times 64\) bit S-boxes are denoted by \(t_1, t_2, t_3\) and \(t_4\) where \(C[i]\) denotes the \(i\)th byte of \(C\) (\(0 \leq i \leq 7\)). The \(i\)th round input values are shown as \(A_i, B_i, C_i\) (3-64-bit words) where \(i \in \{0, \ldots, 24\}\), \(i\)th round message block is \(X_i\) and \(i\)th round output values are \(A_{i+1}, B_{i+1}, C_{i+1}\). In the round function \(A, B, C\) state variables are updated as:
A := A ⊟even(C)
B := (B ⊟odd(C)) × const
C := C ⊕ X

where \( const \in \{5, 7, 9\} \) and after modification part, the results are swapped and \( A, B, C \) become \( B, C, A \). The functions \( even \) and \( odd \) are defined as:

\[
\begin{align*}
*even(C) & = t_1(C[0]) \oplus t_2(C[2]) \oplus t_3(C[4]) \oplus t_4(C[6]) \\
*odd(C) & = t_1(C[7]) \oplus t_2(C[5]) \oplus t_3(C[3]) \oplus t_4(C[1])
\end{align*}
\]

Before the beginning of the second 8-round pass, intermediate values \( A, B, C \) are updated as \( C_0, A_0, B_0 \). Before the beginning of the last 8-round pass again intermediate values are updated and they are assigned to \( B_{17}, C_{17}, A_{17} \). After the last round of the state update transformation, the initial values \( A_0, B_0, C_0 \) and \( A_{24}, B_{24}, C_{24} \) are combined resulting to the hash value or the initial value of the next step

\[
\begin{align*}
A_{25} & = A_0 \oplus A_{24}, B_{25} = B_0 \oplus B_{24}, C_{25} = C_0 \oplus C_{24}
\end{align*}
\]

The block cipher mode of Tiger is straightforward. The chaining operations of the intermediate values are omitted and Tiger is treated as a block cipher encrypting 192-bit plaintext into 192-bit ciphertext using 512-bit secret key. There is no need to invert the \( odd \) and the \( even \) function since their inverses do not affect the decryption mode. In the decryption mode, we just use the inverses of the binary operations that can be defined very easily except for the division \( mod \ 2^{64} \). However, as we divide any number \( mod \ 2^{64} \), this division operation is well defined. Thus, besides the encryption function, the decryption function is well defined. Moreover, from now on, the message expansion is called the key schedule of Tiger.

C. The Key Schedule of Tiger

The non-linear key schedule of Tiger uses some logical operators together with the XOR, addition, subtraction, and shift operations. In the first 8 rounds, the original message words \( X_0, ..., X_7 \) are used and for the next 8 rounds the key scheduling is applied and the message words \( X_8, ..., X_{15} \) are formed. For the remaining 8 rounds the key scheduling is performed to the message words \( X_{16}, ..., X_{23} \). 512-bit key is expanded by the operations shown in Table 2:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A \oplus B )</td>
<td>Addition of A and B ( mod \ 2^{64} )</td>
</tr>
<tr>
<td>( A \oplus B )</td>
<td>Subtraction of A and B ( mod \ 2^{64} )</td>
</tr>
<tr>
<td>( A \otimes B )</td>
<td>Multiplication of A and B ( mod \ 2^{64} )</td>
</tr>
<tr>
<td>( A \oplus B )</td>
<td>Bitwise XOR-operation of A and B ( mod \ 2^{64} )</td>
</tr>
<tr>
<td>( \neg A )</td>
<td>Bitwise NOT-operation of A</td>
</tr>
<tr>
<td>( A &lt; &lt; n )</td>
<td>Bitwise shift of A to the left</td>
</tr>
<tr>
<td>( A &gt; &gt; n )</td>
<td>Bitwise shift of A to the right</td>
</tr>
<tr>
<td>( X_i )</td>
<td>Message word ( i )</td>
</tr>
<tr>
<td>( X_i[even] )</td>
<td>Even bytes of ( X_i )</td>
</tr>
<tr>
<td>( X_i[odd] )</td>
<td>Odd bytes of ( X_i )</td>
</tr>
</tbody>
</table>

TABLE 1: Notation

The Key Schedule of Tiger

There are some useful differentials for the message expansion of Tiger which will be given in the following sections, some of which are used to attack the hash mode of Tiger in [6], [7] and [8]. By useful differentials we mean to have probability one propagation of message differences used in the step functions.

III. RELATED-KEY BOOMERANG AND RECTANGLE ATTACKS

The related-key boomerang and the rectangle attacks are some kind of combined attacks that are introduced independently by Kim et.al [10] and Dunkelman et.al [14]. Nowadays, they are the most effective and powerful block cipher attacks that are applied to many known ciphers [15], [16], [17]. In the following subsections, we will briefly introduce these attacks.
together with their primitives, namely the pure boomerang and the rectangle attack.

A. The Boomerang and the Related-Key Boomerang Attack

The Boomerang Attack [18] may be seen as the refinement or the effective use of the pure differential cryptanalysis [19]. After the application of differential-linear cryptanalysis [20], the boomerang attack can also be called differential-differential cryptanalysis. In the boomerang process, instead of using one long-ineffective (low probability) differential, the attacker makes use of two short-high probability differentials to increase the number of rounds attacked and the probability of the differential. The disadvantage of the boomerang attack is its adaptively chosen plaintext-ciphertext nature. Besides the encryption box of the attacked cipher, it is assumed to have the decryption box.

For the sake of simplicity, we will use the same notation as in [14]. Boomerang distinguisher treats the attacked cipher \( E \) as a cascade of two sub-ciphers \( E_0 \) and \( E_1 \), i.e., \( E = E_1 \circ E_0 \). Let \( \alpha \rightarrow \beta \) with probability \( p \) be the first differential used for \( E_0 \) and \( \gamma \rightarrow \delta \) with probability \( q \) be the second differential used for \( E_1 \). Notice that, once the differential is chosen in one direction, the same differential holds for the opposite direction. Namely, the differentials \( \beta \rightarrow \alpha \) for \( E_0^{-1} \) and \( \delta \rightarrow \gamma \) for \( E_1^{-1} \) hold with probabilities \( p \) and \( q \) respectively. The boomerang distinguisher works as follows:

- Take a randomly chosen plaintext \( P_1 \) and form \( P_2 = P_1 \oplus \alpha \).
- Obtain the corresponding ciphertexts \( C_1 = E(K_1)(P_1) \) and \( C_2 = E(K_2)(P_2) \) through \( E \).
- Form the second ciphertext pair by \( C_3 = C_1 \oplus \delta \) and \( C_4 = C_2 \oplus \delta \).
- Obtain the corresponding plaintexts \( P_1 = E^{-1}(C_3) \) and \( P_4 = E^{-1}(C_4) \) through \( E^{-1} \).
- Check \( P_3 \oplus P_4 = \alpha \).

The boomerang distinguisher works with probability \( p^2q^2 \). For a random permutation, the last step of the above argument holds with probability \( 2^{-n} \) where \( n \) is the number of the bits of each plaintext \( P \). Thus, \( pq > 2^{-n/2} \) must hold for the boomerang distinguisher. The attack can be improved by using all \( \beta \) and all \( \gamma \) values at the same time. Further details are given in [14], [18], [21].

The related-key boomerang attack, on the other hand, is one of the effective combined attacks on block ciphers that can be applied to many known block ciphers. For the related-key model, attacker assumes to know the relation (difference) between the keys, but not the exact values of keys. The standard differential model tries to increase \( P(E_K(x) \oplus E_K(x + \Delta x) = \Delta y) \). The related-key model, on the other hand, tries to increase \( P(E_K(x) \oplus E_{K'\oplus\Delta K}(x + \Delta x) = \Delta y) \).

The adaptation of related-key model to the boomerang attack is straightforward. The usual related-key model is applied to the subciphers \( E_0 \) and \( E_1 \) separately and the normal procedure is applied for the boomerang distinguisher. However, some additional properties are adapted for the related-key boomerang distinguisher. Instead of one pair of related-keys, \( 4 \) (or more) [22], [21], [23] related keys can be used and the most effective one is selected for the attack according to the structure of the cipher. For Tiger, however, we are going to give details about the related-key boomerang distinguisher based on 4 related-keys as follows:

- Take a randomly chosen plaintext \( P_1 \) and form \( P_2 = \hat{P}_1 \oplus \hat{\alpha} \).
- Obtain the corresponding ciphertexts \( C_1 = E_{K_1}(P_1) \) and \( C_2 = E_{K_2}(P_2) \) through \( E \), where \( K_2 = K_1 \oplus \Delta K_{12} \).
- Form the second ciphertext pair by \( C_3 = C_1 \oplus \hat{\delta} \) and \( C_4 = C_2 \oplus \hat{\delta} \).
- Obtain the corresponding plaintexts \( P_1 = E_{K_1}^{-1}(C_1) \) and \( P_4 = E_{K_2}^{-1}(C_4) \) through \( E^{-1} \), where \( K_3 = K_1 \oplus \Delta K_{13}, \Delta K_{13} \).
- Check \( P_3 \oplus P_4 = \alpha \).

The probabilistic arguments are same as in the boomerang distinguisher but they are converted to the related-key model for the related-key boomerang distinguisher.

B. The Rectangle and the Related-Key Rectangle Attack

The rectangle attack converts the adaptively chosen nature of the boomerang attack into the chosen plaintext attack. In fact, it is the refinement of the amplified-boomerang attack [24] and used to attack many known ciphers [15], [22]. Instead of using both encryption and the decryption boxes, the rectangle attack only uses the encryption box.

In boomerang distinguisher, the \( \gamma \) difference after \( E_0 \) and before \( E_1 \) is gathered through the decryption process. However, in rectangle distinguisher, the pairs \( (P_1, P_2) \) and \( (P_3, P_4) \) conforms to the differential \( \alpha \rightarrow \beta \) and since \( (P_1, P_3) \) is taken as random, it is expected that the difference \( E_0(P_1) \oplus E_0(P_3) = \gamma \) works with probability \( 2^{-n} \). Once this is satisfied, the differential \( \gamma \rightarrow \delta \) comes to the picture. Of course, the subciphers before and after the rectangle distinguisher works as in the boomerang distinguisher. Besides the advantage of chosen plaintext nature, it also makes use of all \( \beta' \) values satisfying \( \alpha \rightarrow \beta' \) and all \( \gamma' \) values that satisfy \( \gamma' \rightarrow \delta \). For the further improvements, the details are given in [14]. Using the notations given above, one can describe the rectangle distinguisher as follows:

- Take a randomly chosen plaintext \( P_1 \) at random and obtain the corresponding ciphertext \( C_1 = E_{K_1}(P_1) \).
- Form \( P_2 = P_1 \oplus \alpha \) and obtain the corresponding ciphertext \( C_2 = E_{K_2}(P_2) \), where \( K_2 = K_1 \oplus \Delta K_{12} \).
- Pick another randomly chosen plaintext \( P_3 \) and obtain the corresponding ciphertext \( C_3 = E_{K_1}(P_3) \), where \( K_3 = K_1 \oplus \Delta K_{13} \).
- Form \( P_4 = P_3 \oplus \alpha \) and obtain the corresponding ciphertext \( C_4 = E_{K_1}(P_4) \), where \( K_4 = K_3 \oplus \Delta K_{12} \).
- Check \( C_1 \oplus C_3 = \delta \) and \( C_2 \oplus C_4 = \delta \).

The probability \( P \) of the rectangle distinguisher is given by \( P = 2^{-n}E^2q^2 \), where \( E = \sqrt{\sum_\beta P^2_{K_1,K_2}(\alpha \rightarrow \beta)} \) and \( q = \sqrt{\sum_\gamma P^2_{K_3,K_4}(\gamma \rightarrow \delta)} \). For a random cipher, the probability of the given difference is \( P' = 2^{-2n}S \) where \( S \) is the cardinality of the set of differences of all \( \delta \) values. Once \( P \geq P' \) is satisfied, the rectangle distinguisher works.
not dealing with which type of difference is used. As in [6], notice that a difference \( I \) propagates as the zero difference through the even function since it is in the odd byte and the propagates as \( I \) after the multiplication by constants.

**B. The Differentials of the Key Scheduling Algorithm**

In Tiger, the message expansion algorithm is non-linear. However, some differences propagate linearly. One of such differential is used in [6] to find collisions to reduced round Tiger. This motivates us to search for other good differentials that propagates very efficiently. What makes it good in terms of their efficiency is quite obvious in that the hamming weight of the corresponding differences should be kept small. Also, reducing carry effect by introducing the difference \( I \), we got several probability one differentials, 4 of them used in 20 and 22-round related-key rectangle and boomerang distinguishers.

In order to succeed, we need to combine some of these differentials very effectively. Observing the propagation of these differentials, we should make an extensive use of cancellations and probability one differentials. Moreover, low weight differentials and the number of rounds attacked are also very important. In the scope of this simple tricks, the following sections contain our attack on the encryption mode of Tiger.

**C. 20-Round Distinguisher**

**The Differential for \( E_0 \) (rounds 5 – 13)**

In Tiger, we can find a probability 1 related-key differential for \( E_0 \). For \( E_0 \), the related-key differential \((I, I, I) \rightarrow (0, 0, 0)\) works with probability 1 for rounds 5 – 13 under the key difference \((0, I, 0, 0, 0, I, I, I)\) shown in Table 1. In round 5, by imposing difference \( \alpha = (\Delta A_{5}, \Delta B_{5}, \Delta C_{5}) = (I, I, I) \), we cancel the subkey difference \( \Delta X_5 = I \) with \( \Delta C_5 = I \) making \( \Delta A_6, \Delta B_6, \Delta C_6 = (I, 0, I) \). In round 6, as in the previous round, we cancel the subkey difference \( \Delta X_6 = I \) with \( \Delta C_6 = I \). Finally in round 7, we have \( \Delta A_7, \Delta B_7, \Delta C_7 = (0, 0, I) \). Again, the subkey difference \( \Delta X_7 = I \) and the word \( C_7 \) difference \( \Delta C_7 = I \) cancel each other. From round 7 until round 13, we use the trivial differential which makes \( \beta = (0, 0, 0) \). Notice that, we make an extensive use of the trivial propagation of the I difference through the words \( B_i \) and even function as it does not affect the even bytes of the corresponding words.

Up to know, everything works with probability 1 and the differential probability \( p \) and \( \tilde{p} \) for the subcipher \( E_0 \) is 1. This is valid for both of the related-key rectangle and the related-key boomerang attacks.
TABLE III: The Propagation of Differences Through $E_0$

<table>
<thead>
<tr>
<th>Round</th>
<th>$\Delta A$</th>
<th>$\Delta B$</th>
<th>$\Delta C$</th>
<th>$\Delta K$</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The Differential for $E_1$ (rounds 13–22)

For the second part of our distinguisher $E_1$, the related-key differential $(0, I, 0) \to (0, 0, 0)$ works with probability 1 for rounds 13–22 under the key difference $(0, 0, 0, 0, 0, 0, I)$. Here, according to the notation given above, $\gamma = (I, 0, 0)$. Again we will use the trivial propagation of the difference $I$ through the words $B_i$. The difference $\gamma$ in round 13 propagates to the round 15 as $(\Delta A_{15}, \Delta B_{15}, \Delta C_{15}) = (0, 0, I)$ with probability 1 and cancels the subkey difference $\Delta X_{15} = I$. From the end of the round 15 till round 22, again we use the trivial differential making $\Delta A_{22}, \Delta B_{22}, \Delta C_{22} = (0, 0, 0)$. As in $E_0$, everything works with probability 1 and the differential probability $q$ and $\hat{q}$ for the subcipher $E_1$ is 1. This is valid for both of the related-key rectangle and the related-key boomerang attacks.

D. The Round Before and After the Distinguisher

We can extend the above distinguisher by adding one round before the distinguisher by imposing $\alpha'$ difference in the fifth round. Since $\Delta A_4 = I$ and $\Delta C_4 = I$ differences propagate directly to the next round, we just need to play with the difference $\Delta B_4$. Remember that we have to get $\Delta A_5 = I$. Therefore, $\Delta B_4 = I \oplus Odd(I) = \alpha'$ satisfies the desired difference $\alpha$. However, since we have a probability one differential, we just need to take one pair of plaintexts at the beginning of the fifth round. In fact, we can cancel the difference coming from the odd function by imposing the same difference to the internal variable $B_4$. That is, let $A_4 \oplus A_4 = I$ and $C_4 \oplus C_4 = I$. Now, $B_4$ can be chosen randomly but $B_4'$ is constructed as $B_4' = B_4 \oplus I \oplus Odd(C_4) \oplus Odd(C_4) \oplus I$.

TABLE IV: The Propagation of Differences Through $E_1$

<table>
<thead>
<tr>
<th>Round</th>
<th>$\Delta A$</th>
<th>$\Delta B$</th>
<th>$\Delta C$</th>
<th>$\Delta K$</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Since, we have one pair, there is no probabilistic arguments in that construction.

There is also a possibility to add a round after the distinguisher given above. We have $(\Delta A_{22}, \Delta B_{22}, \Delta C_{22}) = (0, 0, 0)$ and the subkey difference $\Delta X_{22}$ in the last round is $I$. Therefore, the propagation of this difference through the last round leads to the difference $(\Delta A_{23}, \Delta B_{23}, \Delta C_{23}) = (\delta', I, 0)$ where $A_{23} \oplus A_{23} = Odd(B_{23}) \oplus Odd(B_{23})$.

E. 22-Round Distinguisher

The Differential for $E_0$ (rounds 3–13)

Another differential for Tiger can be used to extend the distinguisher to 22 rounds. This time the other differential in Table 1 is used. In round 3, by imposing difference $\alpha = (\Delta A_3, \Delta B_3, \Delta C_3) = (0, I, 0)$, we cancel the subkey difference $\Delta X_3 = I$ with $\Delta C_3 = I$ making $(\Delta A_3, \Delta B_3, \Delta C_3) = (0, 0, 0)$. From round 6 until round 13, we use the trivial differential which makes $\beta = (0, 0, 0)$.

Again, all differential works with probability one and we make an extensive use of the propagation of $I$ difference through round operations.

The Differential for $E_1$ (rounds 13–22)

For the second part of our distinguisher $E_1$, the related-key differential $(0, I, 0) \to (0, 0, 0)$ works with probability 1 for rounds 13–22 under the key difference $(0, 0, 0, 0, 0, 0, I)$. Here, according to the notation given above, $\gamma = (0, I, 0)$. Again we will use the trivial propagation of the difference $I$ through the words $B_i$. The difference $\gamma$ in round 13 propagates to the round 15 as $(\Delta A_{15}, \Delta B_{15}, \Delta C_{15}) = (0, 0, I)$ with probability 1 and cancels the subkey difference $\Delta X_{15} = I$. From the end of the round 15 till round 22, again we use the trivial differential making $\Delta A_{22}, \Delta B_{22}, \Delta C_{22} = (0, 0, 0)$. As in $E_0$, everything works with probability 1 and the differential probability $q$ and $\hat{q}$ for the subcipher $E_1$ is 1. This is valid for both of the related-key rectangle and the related-key boomerang attacks.

TABLE V: The Propagation of Differences Through $E_0$

<table>
<thead>
<tr>
<th>Round</th>
<th>$\Delta A$</th>
<th>$\Delta B$</th>
<th>$\Delta C$</th>
<th>$\Delta K$</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
TABLE VI: The Propagation of Differences Through $E_1$

<table>
<thead>
<tr>
<th>Round</th>
<th>$\Delta A$</th>
<th>$\Delta B$</th>
<th>$\Delta C$</th>
<th>$\Delta K$</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

V. THE ATTACK

For the boomerang distinguisher, we just use the round before and after the distinguisher added to the usual related-key boomerang distinguisher that totally covers 22 rounds (20-round version is same). The related key boomerang attack to the reduced round Tiger is as follows:

- Take a randomly chosen plaintext $P_1 = (A_2, B_2, C_2)$ and form $P_2 = (A'_2, B'_2, C'_2)$ as above.
- Obtain the corresponding ciphertexts $C_1 = E_{K_1}(P_1)$ and $C_2 = E_{K_2}(P_2)$ through $E$, where $K_2 = K_1 \oplus (I, I, 0, 0, 0, I, 0, 0, 0)$.
- Take the second ciphertext pair as $C_3 = C_1 \oplus (\delta', I, 0)$ and $C_4 = C_2 \oplus (\delta', I, 0)$.
- Obtain the corresponding plaintexts $P_3 = E^{(2)}_{K_1}(C_3)$ and $P_4 = E^{(2)}_{K_1}(C_4)$ through $E^{-1}$, where $K_3 = K_1 \oplus (0, 0, 0, 0, I, 0, 0, 0, I)$.
- Check $P_3 \oplus P_4 = P_1 \oplus P_2$.
- If this is the case, identify the corresponding cipher as Tiger.

For the related-key rectangle distinguisher on the other hand, we use the round after the distinguisher added to the related-key rectangle distinguisher that totally covers the rounds $3 - 24$.

- Prepare $2^{22}$ randomly chosen plaintexts $P_1$ at random and obtain the corresponding ciphertext $C_1 = E_{K_1}(P_1)$.
- Form $P_2$ as above and obtain the corresponding ciphertext $C_2 = E_{K_2}(P_2)$, where $K_2 = K_1 \oplus (I, I, 0, 0, 0, I, 0, 0, 0)$.
- Pick another randomly chosen plaintext $P_3$ and obtain the corresponding ciphertext $C_3 = E_{K_3}(P_3)$, where $K_3 = K_1 \oplus ((0, 0, 0, I, 0, 0, 0, I, 0))$.
- Form $P_4 = P_3 \oplus \alpha$ and obtain the corresponding ciphertext $C_4 = E_{K_4}(P_4)$, where $K_4 = K_3 \oplus (I, I, 0, 0, 0, I, 0, 0, 0)$.
- Check $C_1 \oplus C_2 = C_2 \oplus C_4 = \delta$, where $\delta = (\delta', I, 0)$.
- If this is the case identify the corresponding cipher as Tiger.

VI. CONCLUSION

In this paper we applied the related-key boomerang and related-key rectangle attacks to the reduced round of Tiger.

We constructed two related-key boomerang and rectangle distinguishers of 20 and 22 rounds. In the related-key boomerang attacks, the number of required plaintext pair is equal to 2 and the time complexity of the attack is negligible. The related-key rectangle attack works with $2^{22}$ chosen plaintexts and results in a time complexity of about $2^{152.8}$. The distinguishers presented above can be easily converted to a key recovery attack due to the fact that Tiger uses a 512-bit key by guessing the key values before the distinguisher. This type of analysis also can be further applied to the hash mode of Tiger to find collisions.

REFERENCES