Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA

Julio Cesar Hernandez
Computer Security Group
Carlos III University
Leganes, 28911 Madrid, Spain
jcesar@inf.uc3m.es

Pedro Isasi
Artificial Intelligence Group
Carlos III University
Leganes, 28911 Madrid, Spain
isasi@ia.uc3m.es

Abstract- A simple way of creating new and efficient distinguishers for cryptographic primitives such as block ciphers or hash functions is introduced. This technique is then successfully applied over reduced round versions of the block cipher TEA, which is proven to be weak with less than five rounds.

1 Introduction

The construction of a distinguisher [1] (i.e. an algorithm that is able of distinguishing a random permutation or random mapping from a given cryptographic primitive such as a block cipher or hash function) is one of the objectives of a cryptanalyst. Although a distinguisher may or may not be used to recover some of the plaintext or key bits, the existence of an efficient and effective distinguisher always means the cryptographic primitive in question is weak [1,2] and must be discarded for any cryptographic usage.

1.1 The block cipher TEA

TEA stands for Tiny Encryption Algorithm. It is the name of a block cipher invented by David Wheeler and Roger Needham, members of the Computer Security Laboratory of Cambridge University. It was presented in the 1994 Fast Software Encryption Workshop [3].

TEA is a very fast block cipher that does not use predefined tables or S-boxes and does not need much initialisation time. It is a Feistel type algorithm. It works over 64 bit blocks and uses keys of 128 bits. They authors say it has a security (with 8, 16 or 32 rounds) at least comparable with DES (the Data Encryption Standard), and it is quite faster.

However, in the light of some recent results [4,5] this assertion seems to be extremely optimistic. It is very portable, simple and efficient as its compact code shows:

```
void code(long* v, long* w,long* k)
{ unsigned long y=v[0], z=v[1], sum=0,
  delta=0x9e3779b9, n=32;
  while (n-->0)
   { sum += delta ;
     y+=(z<<4)+k[0]^z+sum^(z>>5)+k[1];
     z+=(y<<4)+k[2]^y+sum^(y>>5)+k[3];
   }
  w[0]=y; w[1]=z;
}
```

1.2 Overview

Our method is based in the search for subsets of the input space that produce a high (statistically significant) deviation of the distribution of a given subset of the output produced by a given cryptographic primitive.

For this search we use a genetic-algorithm based approach in which individuals of the population codify bit masks that are used to perform a logical AND with randomly generated inputs.

In this way, we get an extremely efficient representation of those input subsets characterized by having some of the input bits fixed to a zero value. Input subsets of this form are evaluated performing chi-square tests over the output distribution of the subset under observation.

These tests vary, because additional rounds of TEA exponentially increase the dispersion of the output, and thus the difficulty of finding significant deviations.

The genetic algorithm will evolve individuals and populations towards those that, by fixing the input bits that have a greater effect over the observed output, produce a higher deviation from the expected probability distribution.

2 Results

We will briefly present the different approximations used in every case and the results obtained over them,

0-7803-7804-0 /03/$17.00 © 2003 IEEE

proposing efficient distinguishers for all these versions of the algorithm.

2.1 One round TEA

Every input subset (or bitmask) is tested by generating 2^{11} random inputs and then performing a logical AND between each of these inputs and the bitmask. Then TEA1 is applied over thus generated vectors, and the values of the least significant eight bits of the first output word of TEA, that is w[0]&255, are computed.

We have focused in this particular part of the output because there are authors, notably [6], that have shown that block ciphers using rotation as one of their round operations (as is the case of TEA) tend to exhibit bad distributions in their least significant bits of their output words. The fitness function we propose for the genetic algorithm is highly related with the chi-square statistic $\chi 2$ which in our case measures the deviation of the observed output distribution from a uniform, in this way:

$$if\ \chi^2 = 522480 \quad then \quad fitness = w^4$$
$$else\ fitness = \chi^2$$

The idea behind this fitness function is that the value of the chi square statistic cannot increase indefinitely, but has a maximum. The maximum value of the chi square statistic corresponding to a distribution with 255 degrees of freedom and 2^{11} observations is precisely 522480, which is obtained when all the possible 256 outputs collapse in a single one.

Once we find bitmasks that produce this maximum deviation, our search must continue by looking for those bitmasks that are heavier (have more 1's), and this is the reason of including the weight w in the formula above, once the maximum deviation is obtained.

Heavier bitmasks are preferred because they allow for a larger set of different inputs, so in this sense we can say they are more general, and also give more information about the input subset (the 0's in the bitmask or inactive bits) that has a stronger influence over the observed output.

Obviously, we must try to maximise this fitness value. The code we used for testing was the implementation of the genetic algorithm of William M. Spears, from the Navy Center for Applied Research. Other parameters needed for running the genetic algorithm are the size of the population, the mutation rate and the crossover probability. Those values were fixed, respectively, to 100, 0.005 and 0.85 after some trial and error testing that showed they produced good results.

The best bitmask we found after around 1000 generations and 90000 evaluations for the fitness function presented above was $\mathbf{m_1}$:

{0xFFFFEF00,0xFFFFE000,0xFFFFFE00,0xFFFFFF00,
0xFFFFFFFF,0xFFFFFFFF}

which has a weight of 153 and produces a chi square of 522240.

It was then tested with different, previously unseen input subsets of size 2^{11} and in every case it produced maximal deviations (i.e. a collapse of the output). This bitmask can be used to construct an exceptionally efficient and simple distinguisher for TEA1, which pseudocode is presented below:

INPUT: F: $Z_2^{192} \rightarrow Z_2^{64}$, a random mapping or TEA1
ALGORITHM:
Generate a random vector $v \in Z_2^{192}$
Apply the mask $\mathbf{m_1}$, getting $\mathbf{v' = v\&m_1}$ which
can take any of 2^{153} possible values
Compute F(v')=w[0]w[1]
Compute **r** = w[0]&255
OUTPUT: If **r**=185 then F is TEA1 else F is not TEA1

It is interesting to point out that this distinguisher is extremely efficient, given that with only one input text has a false positive probability of 1/256 (or around 0.4%) and a zero probability of false negatives.

2.2 Two rounds TEA

TEA with two rounds (TEA2) is much harder than TEA1. The additional round significantly increases the strength of the algorithm and no usable bitmaks producing maximal deviation (collapses) were found, so the fitness function used for TEA1 is not adequate here.

Although a fitness function consisting simply of the chi-square statistic can break TEA2, it needs some extra care with technical details (mainly a selection proportional to rank and not to fitness and different mutation and crossover probabilities) to produce good results. Another drawback of this fitness function is that it shows a very low convergence towards good solutions (those which produce results statistically better than can be expected at random).

Furthermore, this simplistic approximation is not applicable to TEA3. So, after solving the case for TEA3, we turned back to TEA2 and observed that the same fitness function would be very adequate, so we will present now a new fitness function that reflects an idea that is enough for breaking both TEA2 and TEA3. The fitness function used in this case is shown below:

$$if\ \chi^2 \geq 403.4579\ then\ fitness = 1/w^3 + w^4$$
$$else\ fitness = 1/w^3$$

The idea behind this fitness function is to divide the search for good and heavy bitmasks in two phases. In the first one, the chi-square values will be around the 0.5 percentile, and the fitness function above will simply look for low-weighted bitmasks. When the bitmasks are sufficiently low to produce high values of the chi square statistic (above the threshold of 403.4579), then the objective is to find heavier bitmasks between those which produce a very high statistical value (Table 1 shows some p-values of a chi-square distribution with 255 degrees of freedom).

In this way, we do not maximize the chi-square value but the weight of the masks that produce a statistic value over a threshold. In this case, the threshold is the corresponding value for a chi-square distribution with 255 degrees of freedom and a p-value of $5*10^{-9}$, so it clearly shows a very strong deviation from randomness.

P-value	Value
0.5	254.33
10^{-2}	310.45
10^{-3}	330.51
10^{-4}	347.65
$\mathbf{10^{-5}}$	362.98

Table 1: Some p-values for a chi-square statistic with 255 degrees of freedom

Using this approximation, we got the following bitmask $\mathbf{m_2}$ after around 800 generations and 70000 evaluations

{0xBFFFF0FA,0xFFFE7388,0xFFFFF7F8,0xFFFFF3F8, 0xFFFFEF85,0xFFFFFF8C}

which has a weight of 155.

After testing this bitmask with other previously unseen inputs, the average chi square statistic obtained was 736.05. It is then feasible to construct an efficient distinguisher for TEA2, as shown in the following pseudocode:

INPUT: F: $Z_2^{192} \rightarrow Z_2^{64}$, a random mapping or TEA2
ALGORITHM:
Generate 2^{11} random vectors $\mathbf{v}_i \in Z_2^{192}$
Apply mask $\mathbf{m_2}$ to every vector $\mathbf{v}_i$, getting $\mathbf{v}_i'=\mathbf{v}_i\&\mathbf{m_2}$ that can take any of 2^{155} possible values
Compute $F(\mathbf{v}_i')= w[0]_i\ w[1]_i$
Compute $\mathbf{r}_i = w[0]_i\ \&255$
Perform a chi-square test for checking if the observed distribution of $\mathbf{r}_i$ is consistent with the expected uniform distribution, calculating the corresponding chi-square statistic $\chi 2$
OUTPUT: If $\chi 2 > 390.0315$ then F is TEA2 else F is not TEA2

The 390.0315 is the value corresponding to a p-value of 10^{-7}, so the distinguisher described before will have a false positive probability of 10^{-7} and a false negative probability even closer to zero.

2.3 Three rounds TEA

Using essentially the same approximation described before, we get the following bitmask $\mathbf{m_3}$ for TEA with 3 rounds after around 750 generations and 63000 evaluations

{0xFFE1F040,0xFCE70446,0xFFEFF06E,0xFFE7F42A, 0xFFBF1825,0xFFFA0064}

which has a weight of 116 and produces an average chi-square statistic of 393.6 over previously unseen cases.

The distinguisher will be, then

INPUT: F: $Z_2^{192} \rightarrow Z_2^{64}$, a random mapping or TEA3
ALGORITHM:
Generate 2^{11} random vectors $\mathbf{v}_i \in Z_2^{192}$
Apply mask $\mathbf{m_3}$ to every vector $\mathbf{v}_i$, getting $\mathbf{v}_i'=\mathbf{v}_i\&\mathbf{m_3}$ that can take any of 2^{116} possible values
Compute $F(\mathbf{v}_i')= w[0]_i\ w[1]_i$
Compute $\mathbf{r}_i = w[0]_i\ \&255$
Perform a chi-square test for checking if the observed distribution of ri is consistent with the expected uniform distribution, calculating the corresponding chi-square statistic $\chi 2$
OUTPUT: If $\chi 2 > 330.5197$ then F is TEA3 else F is not TEA3

The 330.5197 is the value corresponding to a p-value of 10^{-3}, so this distinguisher will have a false positive probability of 10^{-3} and a false negative probability of around 0.5%

2.4 Four rounds TEA

This is a considerably harder case. When using the same approximation that was successful over the prior

two cases, we only managed to obtain bitmasks of relatively low weight, not more than 47.

This is interesting and can be useful for different cryptanalytic purposes, for example for starting a search of impossible differentials, but it is not enough in this case, as it does not allow for 2^{64} different inputs. If not having at least 2^{64} different elements in the input subset, we can not ask to obtain a good distribution of the output, as it must somehow reflect the low entropy of the input (although it do not need to be precisely in w[0]&255, the bits we observe).

So we need a different approximation, a subtler one, able of distinguishing from randomness behaviours that may have past undetected by other, less sensible, tests.

Our proposal is based in a test used to measure the Strict Avalanche Criterion or SAC [7]. The SAC will be measured just by flipping at random a bit that is at a position where there is an active bit (i.e. one that has a 1 in the corresponding input mask), then measuring the Hamming distance of the two outputs.

A mapping $Z_2^m \rightarrow Z_2^n$ has the SAC over a certain input subset $S \in Z_2^m$ iff the Hamming distance of the output of inputs x and x' that only differ in a position (i.e. $w(x \oplus x')=1$) and belong to the input subset S follow a Binomial distribution with parameters ½ and n.

In the case of TEA, we should have a *B(1/2,64)*. It is important to note that the satisfactibily of this criterion implies the avalanche effect (changes in the input of only one bit should produce a change of around half of the output), because the average of the distribution *B(1/2,n)* is *n/2*.

For testing if a given bitmask represents an input subset which elements meet the SAC, we propose to perform a chi-square test for the goodness of fit of the observed output distribution of the Hamming values with respect to the theoretical Binomial distribution.

In this case, we have a chi square statistic with 64 degrees of freedom. Table 2 shows some p-values of this distribution.

P-value	Value
0.5	*63.33*
10^{-2}	93.21
10^{-3}	104.71
10^{-4}	114.83
10^{-5}	124.10

Table 2: Some p-values for a chi-square statistic with 64 degrees of freedom

Using this approximation, we got the following bitmask $\mathbf{m_4}$

{0x96922A0C,0x42C06402,0x35B11001,0x97000000,0xF0000001,0xBEB00001}

with a weight of 50 (the weight is not a limit here, as far as sufficiently many input elements exist to perform the test), a fitness of 420.95 and an average chi-square value over previously unseen examples of 294.86.

In this case, we did not introduced any preference for heavier bitmasks. The corresponding distinguisher pseudocode will then be:

INPUT: F: $Z_2^{192} \rightarrow Z_2^{64}$, a random mapping or TEA4
ALGORITHM:
Generate 2^{11} *random vectors* $\mathbf{v_i} \in Z_2^{192}$
Apply mask $\mathbf{m_4}$ to every $\mathbf{v_i}$, getting $\mathbf{v_i'} = \mathbf{v_i} \& \mathbf{m_4}$ that can take 2^{50} possible values
For every $\mathbf{v_i'}$
    Select a position p at random from those that have a 1 in the bitmask (active positions)
    Generate $\mathbf{v_i''}$ by changing the value of $\mathbf{v_i'}$ at position p
End For
Compute $\mathbf{r_i} = H(F(\mathbf{v_i'}), F(\mathbf{v_i''}))$, the Hamming distance between $F(\mathbf{v_i'})$ and $F(\mathbf{v_i''})$
Perform a chi-square test for checking if the observed distribution of ri is consistent with the expected distribution, calculating the corresponding chi-square statistic $\chi 2$
OUTPUT: If $\chi 2 > 148.3564$ then F is TEA4 else F is not TEA4

where 148.3564 is the threshold value corresponding to a p-value of 10^{-8}. The distinguisher will, then, have a false positive probability of 10^{-8} and a negligible false negative probability.

3 Conclusions

We have presented a new method of constructing distinguishers for cryptographic mappings such as block ciphers or hash functions, which has been proven useful.

This method is applied over reduced round variants of the block cipher TEA, which is proved to be weak with less than five rounds.

Furthermore, the distinguishers obtained are very efficient, needing very few inputs to obtain high distinguishing probabilities. In the case of TEA1 we only need one text to obtain a high distinguishing probability, and even with TEA4 we manage to construct a distinguisher that only needs 2^{11} texts.

Moreover, these output distributions discovered with the aid of genetic algorithms and used to construct the distinguishers, could also be useful in future attacks for recovering part of the plaintext or key bits, as in [1].

Additionally, this attack could help in avoiding pitfalls while designing new cryptographic primitives. For example, if we substitute in the TEA code all operations by additions, the resulting cipher is slightly weaker than the original, as proved by the possibility of extending this attack to 5 rounds. If, on the other hand, we change all operations by xor's, the final cipher is much weaker than the original, because this attack scheme could find distinguishers even with 128 rounds.

So, in a way, and by using this attack technique, we can give reasons to support the exact combination of additions and xor's proposed by TEA designers.

Finally, these results clearly show that the block cipher TEA with less than five rounds is not robust enough as to being useful for any cryptographic purposes.

Although we acknowledge that previous works, specially [5] have shown that TEA cannot be considered a secure block cipher, and have presented stronger attacks on it, we still think the proposed approach deserves merit because it is the first published attack using AI techniques that is able of producing worthy cryptanalytic results when confronted against modern ciphers.

Acknowledgments

This research was supported by project TIC2002-04498-C05-4 of the Spanish Ministerio de Ciencia y Tecnología.

References

[1] L. Knudsen and W. Meier
Correlations in RC6 with a Reduced Number of Rounds. Proceedings of the Seventh Fast Software Encryption Workshop, Springer-Verlag, 2000

[2] T. Shimoyama, K. Takeuchi, J. Hayakawa
Correlation Attack to the block cipher RC5 and the simplified variants of RC6, Third AES Candidate Conference AES3, 2000

[3] D. Wheeler, R. Needham
TEA, A Tiny Encryption Algorithm, Proceedings of the 1995 Fast Software Encryption Workshop. pp. 97-110 Springer-Verlag. 1995

[4] David Wagner, John Kelsey, Bruce Schneier
Related-Key cryptoanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2 and TEA, Proceedings of the ICICS'97 Conference, pp. 233-246, Springer-Verlag, 1997.

[5]Dukjae Moon, Kyungdeok Hwang, Wonil Lee, Sangjin Lee and Jongim Lim.
Impossible Differential Cryptanalysis of Reduced Round XTEA and TEA. Fast Software Encryption, FSE 2002, Leuven, Belgium, February 4-6, 2002. Springer LNCS, v.2365. pp 49-60

[6] John Kelsey, Bruce Schneier, David Wagner
Mod n cryptoanalysis with applications against RC5P and M6, Proceedings of the 1999 Fast Software Encryption Workshop, pp. 139-155 Springer-Verlag, 1999.

[7] R. Forre
The Strict Avalanche Criterion: Special Properties of Boolean Functions and Extended Definition. Advances in Cryptology - CRYPTO'88, vol. 403, pp. 450-468.LNCS Springer-Verlag, 1988