New Results on the Genetic Cryptanalysis of TEA and Reduced-Round Versions of XTEA

Julio C. Hernandez
INRIA-IRISA
Campus de Beaulieu
35042 Rennes, France
Email: jcesar@irisa.fr

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

Abstract-**Recently, a simple way of creating very efficient distinguishers for cryptographic primitives such as block ciphers or hash functions, was presented by the authors. Here, this cryptanalytic attack is shown to be succesful when applied over reduced round versions of the block cipher XTEA. Additionally, a variant of this genetic attack is introduced and its results over TEA shown to be the most powerful published to date.**

I. Introduction

The construction of a distinguisher [2] (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 any 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 [2,3] and must be discarded for any cryptographic usage.

A. The block cipher XTEA

XTEA stands for eXtended Tiny Encryption Algorithm. It is the name of an improvement over TEA, a previous block cipher invented by David Wheeler and Roger Needham, members of the Computer Security Laboratory of Cambridge University.

The original TEA block cipher was presented in the 1994 Fast Software Encryption Workshop [4], but an related-key attack was proposed at [5] and the new XTEA algorithm was the answer of TEA developers's to avoid this attack, although they argued it had little to none practical implications.

XTEA, as TEA, is a very fast block cipher that does not use predefined tables or Sboxes 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 conjectured it had a security (with 8, 16 or 32 rounds) comparable with the DES (the Data Encryption Standard), being quite faster (at least with 8 rounds).

However, in the light of some recent results [5,6] this assertion seems to be extremely optimistic.

It is very portable, simple and efficient as its compact code shows:

```
void encipher(const unsigned long *const v, w, k)
{ register unsigned long y=v[0], z=v[1], sum=0,
  delta=0x9E3779B9,n=32;
  while(n-->0)
    { y += (z <<4 ^ z >>5)+z^sum+k[sum&3];
      sum += delta;
      z += (y<<4 ^ y >>5)+y^sum+k[sum>>11&3];
    }
  w[0]=y; w[1]=z; }
```

B. Overview of our methodology

Our method, already presented in [1], 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 XTEA 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.

0-7803-8515-2/04/$20.00 ©2004 IEEE

II. Results over XTEA

We will briefly present the different approximations used in every case and the results obtained over them, proposing efficient distinguishers for all these versions of the algorithm.

A. One Cycle XTEA

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 XTEA1 is applied over thus generated vectors, and the values of the least significant eight bits of the first output word of XTEA, that is *w[0]&255*, are computed.

We have focused in this particular part of the output because there are authors, notably [7], that have shown that block ciphers using rotation as one of their round operations (as is the case of XTEA) 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 ?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 730 generations and 34000 evaluations for the fitness function presented above was $\mathbf{m_1}$:

{0xFFFFFF00,0x7F7FE000,0xFFFFFF00,0xFFFEFFFF,0xFEFFFBFF,0xF7FFFFFF}

which has a weight of 157 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 XTEA1, which pseudocode is presented below:

INPUT: F: Z_2^{192} ? Z_2^{64}, a random mapping or XTEA1
ALGORITHM:
Generate a random vector $\mathbf{v} \in Z_2^{192}$
Apply the mask $\mathbf{m_1}$, getting $\mathbf{v'} = \mathbf{v}\&\mathbf{m_1}$ which can take any of 2^{157} possible values
Compute F(v')=w[0]w[1]
Compute **r** = w[0]&255
OUTPUT: If **r**=0 then F is XTEA1 else F is not XTEA1

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.

B. Two Cycles XTEA

XTEA with two cycles (XTEA2) is much harder than XTEA1. The additional cycle significantly increases the strength of the algorithm and no usable bitmaks producing maximal deviation (collapses) were found, so the fitness function used for XTEA1 is not adequate here.

Although a fitness function consisting simply of the chi-square statistic can break XTEA2, 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 XTEA3. So, after solving the case for XTEA3, we turned back to XTEA2 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 XTEA2 and XTEA3. 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 I 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.

TABLE I: SOME P-VALUES FOR A CHI-SQUARE STATISTIC WITH 255 DEGREES OF FREEDOM

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

Using this approximation, we got the following bitmask $\mathbf{m_2}$ after around 530 generations and 25000 evaluations:

{0xFFFFE8FC,0xCFFF39DC,0xFFFFF9FC,0xFFF7FFFE,0xDFFFFFEF,0xFFFFFFCE}

which has a weight of 166.

After testing this bitmask with other previously unseen inputs, the average chi square statistic obtained was 442.4. It is then feasible to construct an efficient distinguisher for XTEA2, as shown in the following pseudocode:

INPUT: F: Z_2^{192} ? Z_2^{64}, a random mapping or XTEA2
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^{166} 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 ?2
OUTPUT: If ?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.

C. Three Cycles XTEA

Using essentially the same approximation described before, we get the following bitmask $\mathbf{m_3}$ for XTEA with 3 cycles after around 1600 generations and 76000 evaluations

{0xFFC7E008,0xF9C60000,0xFFC7E018,0xFFFFF5FA,0xFFFFFF8A,0xFFF6009A}

which has a weight of 118 and produces an average chi-square statistic of 530.6 over previously unseen cases.

The distinguisher will be, then

INPUT: F: Z_2^{192} ? Z_2^{64}, a random mapping or XTEA3
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^{118} 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 ?2

OUTPUT: If ?2> 330.5197 then F is TEA3 else F is not TEA3

The 330.5197 threshold 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%.

D. Four Cycles XTEA

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.

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 [8]. 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 ? Z_2^n has the SAC over a certain input subset $S \varepsilon Z_2^m$ 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.

TABLE 2: SOME P-VALUES FOR A CHI-SQUARE STATISTIC WITH 64 DEGREES OF FREEDOM

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

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

{0xCC614648,0x22300132,0xFB00A571,0x32000011,0xF40008C1,0x17701000}

with a weight of 57 (the weight is not a limit here, as far as sufficiently many input elements exist to perform the test), a fitness of 313.06 and an average chi-square value over previously unseen examples of 278.87

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

INPUT: F: Z_2^{192} ? Z_2^{64}, a random mapping or XTEA4
ALGORITHM:
Generate 2^{11} random vectors $\mathbf{v_i} \varepsilon Z_2^{192}$
Apply mask $\mathbf{m_4}$ to every $\mathbf{v_i}$, getting $\mathbf{v_i'}= \mathbf{v_i}\&\mathbf{m_4}$ that can take 2^{57} 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 ?2
OUTPUT: If ?2>148.3564 then F is XTEA4 else F is not XTEA4

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.

III. Improved Attack over TEA

Continuing with our research in the genetic cryptanalysis of TEA, while using essentially the same approach presented in [1] and in this paper, we had also tried to find characteristics (bitmasks) that produce a significative deviation of the observed output by XORing them with the inputs, instead of ANDing them as described earlier.

This approach strongly resembles what we could call a genetic implementation of the differential attack [9], or a genetic search for impossible differentials in the vein of [6], and has lead us to a very interesting result.

We have found two characteristics c_1, c_2 in TEA that make that any two keys k, k' with the given characteristic c_i (i.e. $k \oplus k' = c_i$) produce exactly the same output, no matter how many TEA rounds are used.

These characteristics are:

$$c_1 = \{0x80000000, 0x80000000, 0x00000000, 0x00000000\}$$

and

$$c_2 = \{0x00000000, 0x00000000, 0x80000000, 0x80000000\}$$

So if we generate any key at random, say

$k=\{0xf0ea8d74, 0xeb4576c6, 0x6b14ab8e, 0x908bd353\}$

and we compute

$$k' = k \oplus c_1 = \{0x70ea8d74, 0x6b4576c6, 0x6b14ab8e, 0x908bd353\}$$

or

$$k'' = k \oplus c_2 = \{0xf0ea8d74, 0xeb4576c6, 0xeb14ab8e, 0x108bd353\}$$

then for every input block i and any number of rounds n

$$TEA_n(i,k) = TEA_n(i,k') = TEA_n(i,k'')$$

and, in particular, this happends with the full 32 rounds recommended by the authors.

This is a surprising powerful result, as normally characteristics are discovered for a certain number of rounds and then they vanish exponentially as the number of rounds increases.

IV. Conclusions

We have presented a cryptanalitic attack over reduced round versions of the block cipher XTEA based in the new method of constructing distinguishers for cryptographic mappings proposed at [1].

We have also shown that XTEA is weak with four or less cycles, that is to say, with 8 rounds or less. This is a result quite similar to those obtained for TEA applying the same genetic cryptanalysis, which forces us to conclude that, at least from the genetic cryptanalysis point of view, both TEA and XTEA have a very close security level.

This could also be interpreted as saying that the improvements introduced in XTEA for avoiding the attack presented in [5] are limited to make this related-key attack impracticable but do not improve the overall security of the cipher, which rest very similar to those of TEA.

Although we acknowledge that previous works, specially [6] have shown that both TEA and XTEA cannot be considered as secure block ciphers, and have presented stronger attacks on them, we still think the proposed approach remains valuable because it is one of the firsts published attacks using AI techniques that is able of producing worthy cryptanalytic results when confronted against modern ciphers.

Furthermore, nowadays the most powerful attack on these cipher extend, respectively, to 10 and 12 rounds and the approach proposed here could be extended to 8 rounds, which makes these results far from trivial and pretty close to the most powerful ones.

Additionally, and regarding the improved attack on TEA presented in this paper, it is, as far as we know, the most powerful attack presented to date against the block cipher. It is obviously a serious weakness which concrete implications, for example it reduces the effective key-length to 126 bits, and could being straightforwardly used to construct very efficient blackbox distinguishers. This equivalent-keys weakness is not present, at least to the best of our knowledge, in XTEA.

In fact, as one could expect, some minor changes to the TEA round function could avoid it. For example, if we substitute the original line of code (1) by (1') then the

problem remains, but when substituting any of them by, for example (2) this weakness, at least thus stated, dissappears:

```
void encipher(unsigned long *const v, w, k)
{
register unsigned long y=v[0],z=v[1],sum=0,
delta=0x9E3779B9,
a=k[0],b=k[1],c=k[2],d=k[3],n=32;
while(n-->0)
{
sum += delta;
y += (z << 4)+a ^ z+sum ^ (z >> 5)+b;   (1)
y += (z << 4)+a + z+sum ^ (z >> 5)+b;   (1')
y += (z << 4)+a * z+sum ^ (z >> 5)+b;   (2)
y += (z << 4)+a ^ b ^ z+sum ^ (z >> 5);  (3)

z += (y << 4)+c ^ y+sum ^ (y >> 5)+d;
}
w[0]=y; w[1]=z;}
```

The reason for this is the fact that in 32-bit implementations, operations are made, naturally, mod 2^{32} and for that reason, both addition mod 2^{32} and xor behave identically on the most significant bit, thus producing the indesirable effect of essentially converting (1) in (3) for the leading bit, allowing it to cancellate when simultaneously changing it in a=k[0] and b=k[1] (or, analogously, at c=k[2] and d=k[3]).

This result is similar to a previously published result [5] where the authors presented three attacks on TEA, the first of which is based in simultaneously flipping the next most significant bits (bit 30) of k[2] and k[3], the second based in changing the values of both k[1] (for $k[1] \oplus 2^{31} \oplus 2^{26}$) and v[1] (for $v[1] \oplus 2^{31}$).

Both attacks, although similar to the proposed one, failed to exhibit full round characteristics.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Hernandez, J.C. and Isasi, P. "Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA." *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pp. 341-348, IEEE Press, 2003.

[2] 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

[3] 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

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

[5] 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.

[6] Dukjae Moon, Kyungdeok H., Wonil L., et al. "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

[7] 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.

[8] 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

[9] Eli Biham, Alex Biryukov, Adi Shamir. "Cryptanalysis of Skipjack reduced to 31 rounds using Impossible Differentials." *Technion Computer Science Department Technical Report CS0947*-1998