Analysis of Local Optima in Block Ciphers

John A. Clark, Juan M.E. Tapiador

Department of Computer Science, University of York
York YO10 5DD, England, UK
{jac, jet}@cs.york.ac.uk

Abstract. We present a technique to perform key distinguishing attacks on block ciphers. The method is based on profiling the behaviour of a simple search algorithm when it is applied to recover the key under which a set of known plaintexts has been encrypted. Even though the probability of finding the correct key is negligible, it is observed that the solutions (local optima) yielded by successive searches can be highly dependent on the key, forming patterns that can be unequivocally (in a statistical sense) associated with each particular key. When a cipher suffers from such a weakness, this provides us with an effective procedure to tell apart ciphertexts generated by different and unknown keys. We illustrate the method by applying it to the TEA block cipher, for which attacks of this kind can be successfully mounted against the full version (64 rounds) with extremely simple profiling methods. The technique itself is completely black-box and admits a number of refinements. We suspect it might be applied to many other ciphers by using the same or more complex profiling schemes.
Keywords: Block ciphers; Cryptanalysis; Key distinguishability; Tiny Encryption Algorithm (TEA).

1 Introduction

Assume that $E : GF(2^{\mathcal{N}}) \times GF(2^{\mathcal{L}}) \rightarrow GF(2^{\mathcal{M}})$ is the encryption algorithm of a block cipher. For any plaintext p and key k, it is obvious that no information regarding k should be obtained by analysing the ciphertext $c = E(p, k)$. Furthermore, ciphertexts should be indistinguishable; given $c_1 = E(p, k_1)$ and $c_2 = E(p, k_2)$, an attacker should not be able to find any property in c_1 or c_2 such that it helps to indicate that it is the same plaintext encrypted under different keys, or to tell ciphertexts apart according to the key used. In practical terms, ciphertexts should be indistinguishable from the output generated by a random source.

Even when no other information can be derived (e.g. portions of the key), if an effective procedure to distinguish ciphertexts encrypted under different keys exists, this can be seen as evidence of the cipher's poor randomisation abilities. The previous observation is still valid if the attacker is allowed to know (or even to choose) the plaintexts, having at his disposal a reasonable amount of plaintext/ciphertext pairs (p/c-pairs) generated under different keys. Note,

however, that keys must remain secret – otherwise identifying the correct key is trivial.

Apart from the classical statistical tests of randomness, the approaches used to mount distinguishing attacks against ciphers vary considerably from one work to another, often making use of some properties of the cipher's internal components. In what follows, we briefly describe the key idea behind the technique proposed in this paper.

1.1 Findind the Key as an Optimisation Problem

The use of guided search techniques to find solutions to cryptographic problems has been the subject of considerable interest in recent years. It has been used successfully in component design such as Boolean functions and S-boxes with desirable properties. Almost all cryptanalysis applications of guided search have been restricted to classical substitution and transposition ciphers; there have been very few applications to modern full-strength cryptographic systems.

Formulating a problem as a search problem is fairly straightforward. First the solution space is identified. (In our case this will be the keyspace.) Next, a fitness or cost is associated with each candidate solution k. This is an evaluation of how well a candidate solution matches the desired solution. The aim is either to maximise this evaluation function or else minimise it. The terms *fitness function* and *cost function* are used for maximisation and minimisation problems respectively.

Next, a search strategy must be defined that explores the search space, using the fitness or cost of considered solutions to determine which ones to consider next. When a fitness (or cost) function is defined over each possible solution to a problem, the space of solutions is endowed with a specific *landscape* composed of the fitness value taken by each candidate solution.

The specific shape exhibited by the landscape is intimately related to the representation chosen for solutions, and the search operators and fitness function defined. It is widely recognised that an appropriate fitness landscape is critical in virtually any problem attacked by a search procedure. Factors such as the degree of smoothness/ruggedness, the number of local optima per sphere of a given radius, or its neutrality (number and spaciousness of plateaus) exert a great influence on the effectiveness of the search (see e.g. [3, 4, 15–17].)

In case of problems with relevance in cryptography (and particularly in cryptanalysis), the study of fitness landscapes is a subject that has been scarcely researched, perhaps because not many problems can be *directly* attacked by a search technique. In some circumstances, the very design of a primitive provides theoretical guarantees that make useless any form of search, whilst in others the search space is simply huge.

Suppose, for example, that E is a block cipher, $\{p_1, \ldots, p_n\}$ a set of known plaintexts, and

$$PC = \{(p_i, E_k(p_i))\}_{i=1}^{n} \tag{1}$$

the corresponding set of p/c-pairs created by using an unknown key k. Assume now that we face the problem of finding k using exclusively the set of p/c-pairs. Given a candidate solution $\tilde{k}$, the simplest way of verifying whether it is correct or not is by checking that $E_k(p_i) = E_{\tilde{k}}(p_i)$ for all p_i. Equivalently, the problem can be attacked using a cost function of the form:

$$C(\tilde{k}) = \sum_{i=1}^{n} d_H(E_k(p_i), E_{\tilde{k}}(p_i)) \tag{2}$$

d_H being the Hamming distance between the ciphertexts. This provides us with a very direct (and undoubtedly ineffective) way to reformulate the question as an optimisation problem, using expression (2) as guidance mechanism for the search. It should be clear that for any strong enough cipher, the probability of finding the correct key (i.e. one with cost zero) through the previous scheme should be negligible, regardless of what specific search technique is applied. Furthermore, given a candidate solution $\tilde{k}$, each of its neighbours (e.g. keys with a low Hamming distance to $\tilde{k}$) should have no relation with $\tilde{k}$ insofar as costs are concerned.

Assume that we have at our disposal T different sets of p/c-pairs generated under the same key k, and that for each one we search for the key as described above. The result is a set of solutions $S(k) = \{o_1, \ldots, o_T\}$ corresponding to the local optima (minima, in this case) found by the search in each problem instance. We can pose now two initial questions:

1. For a given fixed key k, is there any relation among the individual local optima $o_i \in S(k)$?
2. Given two different keys k_1 and k_2, is it possible to distinguish between sets $S(k_1)$ and $S(k_2)$?

In this work, we report positive results concerning both previous questions. Put simply, the main finding is that the set of local optima (tentative keys) obtained by the search can be, in some cases, unequivocally associated with the key under which the input has been generated. As such, this provides with the basis to mount a distinguishing attack capable of identifying under which unknown key a set of ciphertexts has been encrypted.

In the next section, we describe the specific search algorithm used in our experimentation and the subsequent analysis performed on the set local optima. Section 3 is devoted to discuss a practical application to the TEA block cipher,

for which good results have been obtained even for the full version (64 rounds.) To our knowledge, these are the best results attained so far for this algorithm as far as distinguishability is concerned. This fact makes us suspect that the attack here described could be successfully applied to other ciphers. Moreover, the analysis we have carried out on the set of local optima is extremely rudimentary. More refined techniques based on the same principle might improve the attack. Section 4 draws some conclusions regarding this aspect and identifies directions for future research.

2 Analysis of Local Optima

In this section, we provide a general description of the technique proposed in this work. We first describe the search algorithm used and then how the local optima are summed up into profiles.

2.1 Search for a Local Optimum

Given a set $\{(p_1, c_1), \ldots, (p_n, c_n)\}$ of p/c-pairs, the algorithm starts with an initial key k_{opt} set to zero and its associated cost, as defined by expression (2). The search scans each key bit from left to right. At each position b, a new candidate key $\tilde{k}_{opt}$ is generated by flipping bit b in k_{opt}. The cost $\tilde{C}$ of the new key is again obtained and, if $\tilde{C} < C$, key $\tilde{k}_{opt}$ is accepted as the best solution found so far and $\tilde{C}$ as its associated cost. Otherwise, the bit flipping is reversed and the next bit position is tried. The scanning procedure is repeated until no further single bit flip can produce an improvement (decrease) in the cost. Since the number of key bits considered for flipping is $|k|$ the search terminates after $|k| - 1$ consecutive non-improving moves. A description of the algorithm is given by Fig. 1.

2.2 Profiling Local Optima

After running T instances of the search with T different sets of p/c-pairs, each local optima in the set $S(k) = \{o_1, \ldots, o_T\}$ can be seen as a derived key that optimises the criterion defined above for each specific set of p/c-pairs. If we denote by $o_i(j)$, $j = 1, \ldots, |k|$ the value taken by bit j in local optimum o_i, a very simple way of summarizing all the information contained in $S(k)$ is by associating with key k the profile:

$$\mathcal{P}(k) = (n_1, \ldots, n_{|k|}) \qquad n_j = \sum_{i=1}^{T} o_i(j) \tag{3}$$

Input: p/c-pairs $\{(p_1, c_1), \ldots, (p_n, c_n)\}$
maximum number of consecutive non-improving moves ($MAXNIM = |k_{opt}| - 1$)
Output: local optimum k_{opt}

1. $k_{opt} \leftarrow 00 \cdots 0$
2. $C \leftarrow \sum_{i=1}^{n} d_H\left(E_{k_{opt}}(p_i), c_i\right)$
3. $nim \leftarrow 0$
4. $b \leftarrow 0$
5. **while** $(nim < MAXNIM)$ **do**
6. Obtain $\tilde{k}_{opt}$ by flipping bit b in k_{opt}
7. Compute $\tilde{C} \leftarrow \sum_{i=1}^{n} d_H\left(E_{\tilde{k}_{opt}}(p_i), c_i\right)$
8. **if** $(\tilde{C} < C)$
9. $k_{opt} \leftarrow \tilde{k}_{opt}$
10. $C \leftarrow \tilde{C}$
11. $nim \leftarrow 0$
12. **else**
13. $nim \leftarrow nim + 1$
14. **end-if**
15. $b \leftarrow (b + 1) \bmod |k_{opt}|$
16. **end-while**
17. **return** k_{opt}

Fig. 1. Local search algorithm.

Such profiles are mere histograms wherein n_j, the j-th component of $\mathcal{P}(k)$, counts how many times bit j is set to one in the derived keys obtained after T searches, each one with a different set of p/c-pairs.

Given two key profiles $\mathcal{P}(k_1) = (n_1, \ldots, n_{|k|})$ and $\mathcal{P}(k_2)(n'_1, \ldots, n'_{|k|})$, a number of similarity measures can be defined. One of the simplest is the 1-norm distance given by:

$$dist(\mathcal{P}(k_1), \mathcal{P}(k_2)) = \sum_{i=1}^{|k|} |n_i - n'_i| \tag{4}$$

The previous method is extremely simple and only the values of individual key bits in the local optima are taken into account. A slightly more sophisticated analysis may incorporate correlations between bits too. The profile is then given by a $|k| \times |k|$ symmetric matrix:

$$\mathcal{P}(k) = [c_{ij}] \qquad c_{ij} = \sum_{t=1}^{T} o_t(i) o_t(j) - \frac{n_i \cdot n_j}{T} \tag{5}$$

wherein each element c_{ij} measures the degree of correlation between bits i and j in the set of local optima attained. Values n_i and n_j correspond to the bit counts as defined by (3).

Even though more complex distances between matrices do exist, we will simply use the sum of the absolute values between cells to measure the similarity between two profiles. Formally, if $\mathcal{P}(k_1) = [c_{ij}]$ and $\mathcal{P}(k_2) = [c'_{ij}]$ are profiles, the distance between them is given by:

$$dcorr(\mathcal{P}(k_1), \mathcal{P}(k_2)) = \sum_{i=1}^{|k|} \sum_{j=i}^{|k|} |c_{ij} - c'_{ij}| \tag{6}$$

3 Key Distinguishability in TEA

The Tiny Encryption Algorithm (TEA) [18] was designed by Wheeler and Needham and rapidly gained some popularity due to its remarkably simple description –usually, a few lines of code. TEA is a Feistel network that operates on 64-bit message blocks and uses a 128-bit key. The suggested number of rounds is 64, which are often implemented in pairs termed cycles; i.e. a cycle corresponds to two rounds.

Early cryptanalysis on TEA attacked its extremely simple key schedule. Kelsey et al. showed in 1996 the existence of equivalent keys [10], demonstrating that the effective key space is 126 bits instead of the theoretical 128 bits. The same authors described in [11] related-key attacks against the cipher requiring 2^{23} chosen plaintexts with a time complexity of 2^{32}. These weaknesses led the authors to propose in [13] two variants, XTEA and Block TEA, and later XXTEA [19].

In a series of works [5–8], Hernandez et al. described several approaches aimed at discovering distinguishers for reduced round versions of TEA. It is reported that TEA with 5 or fewer cycles can be effectively distinguished from a random source with 2^{25} plaintexts. Subsequently, the authors suggest the use of genetic algorithms to evolve distinguishers. The basic idea is to find a subset of plaintexts such that it maps to a subset of ciphertexts in a detectable manner, particularly by means of a χ^2 test. The result is a distinguisher that is effective over such a class. Distinguishers for up to 8 rounds are found by using this approach.

The resistance of TEA and its variants against several differential attacks has been recently analysed. Moon et al. show in [12] an impossible differential attack against 11-rounds TEA that requires $2^{52.5}$ chosen plaintexts and a time complexity of 2^{84}. Hong et al. used in [9] truncated differentials against 17-rounds TEA with 1920 chosen plaintexts and a time complexity of $2^{123.37}$. The

same previous attacks are carried out against XTEA. Surprisingly, TEA seems to be stronger than XTEA from this point of view.

Next we present the results of our attack.

3.1 Analysis of Local Optima in TEA

The technique described in Section 2 has been applied to TEA with different number of rounds. The experiments have been carried out on a Toshiba Satellite laptop with a 1.6GHz Intel processor and 1Gb of RAM.

In the case of the local search algorithm, we used a number of 8 consecutive non-improving moves (MAXNIM). Only the first 32 bits of the key are explored, thus obtaining local optima of 32 components (the remaining 96 are set to 0.) The exploration of the full 128 bits might conduce to a more refined characterisation, though it has not proven necessary to obtain satisfactory results in case of TEA.

In order to evaluate the distinguishing abilities of the technique, we have performed the following experiment. Firstly, N_K different keys are randomly selected. The procedure described below is then done twice using this fixed set of keys:

1. For $i = 1 \ldots T$ do:
   (a) Select randomly a number N_P of plaintexts.
   (b) Encrypt the plaintexts using each key, obtaining N_K sets of p/c-pairs.
   (c) For $j = 1 \ldots N_K$ do:
      i. Search for a local optimum o_i^j for key j using the associated set of p/c-pairs.
      ii. Add o_i^j to the set S_j of local optima for key j.
2. For $j = 1 \ldots N_K$ do:
   (a) Profile S_j to obtain $\mathcal{P}_j$.

The result after the two runs is two sets of profiles, $\{\mathcal{P}_1(k_1), \ldots, \mathcal{P}_1(k_{N_K})\}$ and $\{\mathcal{P}_2(k_1), \ldots, \mathcal{P}_2(k_{N_K})\}$, corresponding to the same N_k keys. Note that, in case of the cipher being an ideal mapping, the $2N_K$ profiles should have no relationship among them.

Figure 2 depicts graphically an example of the 10 pairs of profiles obtained for 10 keys by using 16-rounds TEA. Each profile has been generated from 0.5 millions of local optima, and corresponds to the histogram given by expression 3. For visual inspection, we have performed a classical multidimensional scaling on the 32-components vectors (see e.g. [2, 14].) This is merely a principal coordinates analysis, after which only the two principal components of the transformed data (as given by the eigenvectors) are represented in a 2D map. Even though there is some loss of information by drawing only two components, it

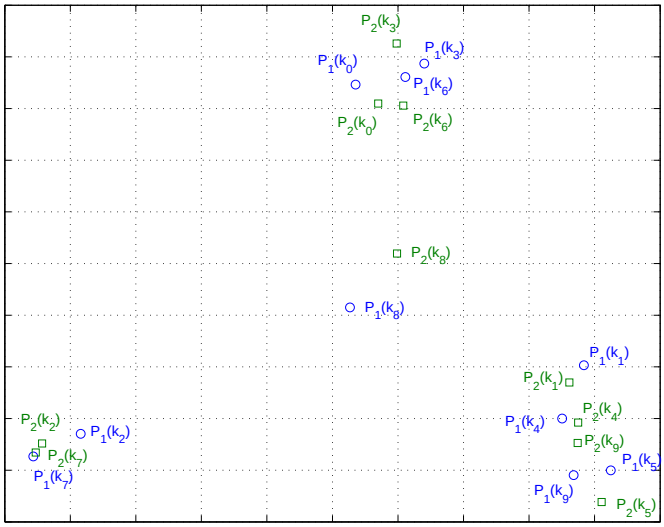

Fig. 2. Results after profiling two times the same 10 different keys for 16-rounds TEA. Each profile (32 dimensions) is reduced to its 2 principal components for its representation.

provides a rough picture of the profile distribution and, most importantly, the distance among them.

It clearly appears that pairs of profiles corresponding to the same key do have some link between them. This may support the hypothesis that the profiles are dependent upon the key, being therefore an instrument to distinguish between keys by only observing a sufficient amount of p/c-pairs.

In the next section we elaborate on this fact by presenting more detailed experimentation.

3.2 Results for 16, 32 and 64 Rounds

The similarity between each profile in the first set and any other in the second one can be measured by using one of the distances defined previously. Distances can be grouped into a $N_K \times N_K$ matrix, $D = [d_{ij}]$, where d_{ij} measures the distance between profiles $\mathcal{P}_1(k_i)$ and $\mathcal{P}_2(k_j)$. Note that perfect distinguishability is attained if $d_{ii} \leq d_{ij}\ \forall j$, i.e. whenever the nearest neighbour to a profile is that corresponding to the same key.

We can conceive, however, a less strict measure of distinguishability (or, equivalently, of classification accuracy) by considering how incorrectly classified a profile is. Given a profile $\mathcal{P}_1(k_i)$, this can be done e.g. by counting how many incorrect profiles in the second set are closer to $\mathcal{P}_1(k_i)$ than to the correct

	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
k_0	**11426.50**	28983.44	26686.26	34960.53	30565.78	28377.99	27104.85	14168.92	32055.06	28621.90
k_1	25170.47	**13100.65**	19690.10	35178.27	29396.72	15364.12	19349.01	28489.48	32805.76	29780.37
k_2	25080.36	19482.11	**11815.92**	29232.44	24166.71	20238.88	14804.42	28434.69	26915.18	24084.09
k_3	34040.78	34479.63	26563.95	**11118.53**	14695.92	34045.65	31004.96	36276.79	15509.54	16673.25
k_4	30576.49	30459.02	24916.20	15641.24	**11891.45**	31478.17	29736.91	32897.09	12122.04	11956.59
k_5	27194.61	16212.97	21018.17	34584.23	29832.58	**11746.06**	19489.04	29637.03	31277.04	29610.92
k_6	25483.05	19745.69	16439.38	34052.74	29852.42	19375.59	**11685.00**	30484.56	32620.55	29503.50
k_7	13508.91	30573.95	28392.69	37200.39	32530.52	28761.53	29048.28	**11127.97**	35364.12	32426.03
k_8	34151.54	34187.37	27946.60	17430.96	15605.39	33022.10	32238.86	35294.33	**12629.33**	15554.54
k_9	26907.18	28186.29	22239.36	17457.22	12668.74	27970.31	25084.88	29294.77	16076.31	**12427.14**

$N(k_0) = [\mathbf{k_0},\ k_7,\ k_2,\ k_6,\ k_5,\ k_9,\ k_1,\ k_4,\ k_8,\ k_3]$ $\quad \mathrm{Rank}(k_0) = 0$
$N(k_1) = [\mathbf{k_1},\ k_5,\ k_6,\ k_2,\ k_0,\ k_7,\ k_4,\ k_9,\ k_8,\ k_3]$ $\quad \mathrm{Rank}(k_1) = 0$
$N(k_2) = [\mathbf{k_2},\ k_6,\ k_1,\ k_5,\ k_9,\ k_4,\ k_0,\ k_8,\ k_7,\ k_3]$ $\quad \mathrm{Rank}(k_2) = 0$
$N(k_3) = [\mathbf{k_3},\ k_4,\ k_8,\ k_9,\ k_2,\ k_6,\ k_0,\ k_1,\ k_5,\ k_7]$ $\quad \mathrm{Rank}(k_3) = 0$
$N(k_4) = [\mathbf{k_4},\ k_9,\ k_8,\ k_3,\ k_2,\ k_6,\ k_1,\ k_0,\ k_5,\ k_7]$ $\quad \mathrm{Rank}(k_4) = 0$
$N(k_5) = [\mathbf{k_5},\ k_1,\ k_6,\ k_2,\ k_0,\ k_9,\ k_7,\ k_4,\ k_8,\ k_3]$ $\quad \mathrm{Rank}(k_5) = 0$
$N(k_6) = [\mathbf{k_6},\ k_2,\ k_5,\ k_1,\ k_0,\ k_9,\ k_4,\ k_7,\ k_8,\ k_3]$ $\quad \mathrm{Rank}(k_6) = 0$
$N(k_7) = [\mathbf{k_7},\ k_0,\ k_2,\ k_5,\ k_6,\ k_1,\ k_9,\ k_4,\ k_8,\ k_3]$ $\quad \mathrm{Rank}(k_7) = 0$
$N(k_8) = [\mathbf{k_8},\ k_9,\ k_4,\ k_3,\ k_2,\ k_6,\ k_5,\ k_0,\ k_1,\ k_7]$ $\quad \mathrm{Rank}(k_8) = 0$
$N(k_9) = [\mathbf{k_9},\ k_4,\ k_8,\ k_3,\ k_2,\ k_6,\ k_0,\ k_5,\ k_1,\ k_7]$ $\quad \mathrm{Rank}(k_9) = 0$

Table 1. Distances and ranks among key profiles for 16-rounds TEA (0.5M local optima profiled).

one. More formally:

$$Rank(\mathcal{P}_1(k_i)) = \#\{\mathcal{P}_2(k_j),\ j \neq i, \text{ such that } d_{ij} \leq d_{ii}\} \tag{7}$$

In general, it can be considered that a statistically significant distinguishability is achieved if all the ranks are less than $N_K/2$.

Tables 1, 2 and 3 show the results for 16-, 32- and 64-rounds TEA with 10 keys. In each case, the matrix of distances is first showed, pointing out in bold typeface the lowest distance in each row. The list of neighbours (in increasing order with respect to distance) and the rank of each profile are shown below. Note that we have relaxed the notation using just k_i to indicate the associated profile.

Fig. 3 depicts graphically (in colour in the electronic version) the matrix of distances for the three cases. The surface has been interpolated to facilitate its visualisation. Note, therefore, that values out of the points (i, j), with i and

	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
k_0	**9718.56**	24917.68	25601.96	20369.62	12729.52	25313.50	13881.03	21864.82	24072.76	13498.45
k_1	23364.37	**9348.07**	10802.40	14323.62	30212.35	10447.73	29164.07	15212.13	10803.70	30240.25
k_2	23473.09	**10337.57**	11312.87	14647.85	28865.43	11760.65	28298.17	16596.42	10564.97	29085.21
k_3	22964.10	15348.32	14475.25	11317.56	24681.15	16160.85	25046.65	**10467.18**	15512.38	24319.00
k_4	16743.99	30014.15	30418.82	25841.46	12047.57	31039.06	**11109.34**	26319.84	28222.44	11887.92
k_5	23120.59	11212.83	10476.68	14595.04	29117.18	10339.96	28702.53	15177.82	**9838.43**	28828.52
k_6	14891.92	29002.21	29351.05	23830.45	**10039.64**	29662.06	12769.38	25127.61	28020.75	11109.22
k_7	22829.94	15690.95	14610.79	11331.41	23718.65	16359.46	23053.11	**11029.28**	15724.36	23523.99
k_8	22685.17	10839.40	11046.13	14229.63	30367.21	**10708.39**	29529.86	15708.19	11197.12	29758.77
k_9	15251.67	29960.44	29627.52	24200.13	11133.89	30074.75	11461.93	26005.15	28075.41	**10799.47**

$$N(k_0) = [\mathbf{k_0},\ k_4,\ k_9,\ k_6,\ k_3,\ k_7,\ k_8,\ k_1,\ k_5,\ k_2] \quad \mathrm{Rank}(k_0) = 0$$
$$N(k_1) = [\mathbf{k_1},\ k_5,\ k_2,\ k_8,\ k_3,\ k_7,\ k_0,\ k_6,\ k_4,\ k_9] \quad \mathrm{Rank}(k_1) = 0$$
$$N(k_2) = [k_1,\ k_8,\ \mathbf{k_2},\ k_5,\ k_3,\ k_7,\ k_0,\ k_6,\ k_4,\ k_9] \quad \mathrm{Rank}(k_2) = 2$$
$$N(k_3) = [k_7,\ \mathbf{k_3},\ k_2,\ k_1,\ k_8,\ k_5,\ k_0,\ k_9,\ k_4,\ k_6] \quad \mathrm{Rank}(k_3) = 1$$
$$N(k_4) = [k_6,\ k_9,\ \mathbf{k_4},\ k_0,\ k_3,\ k_7,\ k_8,\ k_1,\ k_2,\ k_5] \quad \mathrm{Rank}(k_4) = 2$$
$$N(k_5) = [k_8,\ \mathbf{k_5},\ k_2,\ k_1,\ k_3,\ k_7,\ k_0,\ k_6,\ k_9,\ k_4] \quad \mathrm{Rank}(k_5) = 1$$
$$N(k_6) = [k_4,\ k_9,\ \mathbf{k_6},\ k_0,\ k_3,\ k_7,\ k_8,\ k_1,\ k_2,\ k_5] \quad \mathrm{Rank}(k_6) = 2$$
$$N(k_7) = [\mathbf{k_7},\ k_3,\ k_2,\ k_1,\ k_8,\ k_5,\ k_0,\ k_6,\ k_9,\ k_4] \quad \mathrm{Rank}(k_7) = 0$$
$$N(k_8) = [k_5,\ k_1,\ k_2,\ \mathbf{k_8},\ k_3,\ k_7,\ k_0,\ k_6,\ k_9,\ k_4] \quad \mathrm{Rank}(k_8) = 3$$
$$N(k_9) = [\mathbf{k_9},\ k_4,\ k_6,\ k_0,\ k_3,\ k_7,\ k_8,\ k_2,\ k_1,\ k_5] \quad \mathrm{Rank}(k_9) = 0$$

Table 2. Distances and ranks among key profiles for 32-rounds TEA (0.5M local optima profiled).

j integers, do not correspond to any actual result (in fact, it does not make any sense to get a value for them.) When a perfect distinguishability among N_k keys is attained, the main diagonal of the figure contains the minimum values of the corresponding row and column (in blue in the image.)

In all the cases, the results correspond to profilings carried out according to expressions (5) and (6). By using exclusively histograms, perfect distinguishability is obtained up to 32 rounds. Improvements are only reached when considering correlations among key bits as well.

We found in our experimentation that the number T of local optima profiled is a crucial parameter. As a general rule, results improve as the T increases; in particular, the more the number of rounds, the more the number of local optima required. For instance, 16-rounds TEA requires to profile around 500000 local optima to attain perfect distinguishability (see Table 1.) The same number with 32-rounds TEA (Table 2) produces satisfactory results too, though it is necessary

	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
k_0	**43287.87**	78446.92	76637.20	76495.57	78974.22	44800.44	47062.14	43882.57	77421.31	49743.48
k_1	76291.84	45245.03	**41608.05**	44280.74	47962.54	73207.33	79043.87	75134.56	43500.85	77876.26
k_2	76107.35	44025.05	43875.27	**43280.38**	48043.81	76970.40	80306.66	76486.40	46036.28	76007.87
k_3	75930.07	**42992.64**	45455.68	43645.53	43173.39	80517.86	81635.80	78649.27	45071.15	76598.40
k_4	72324.15	47993.72	**44946.54**	46125.68	47450.20	73717.86	78633.92	76179.49	47699.81	72544.78
k_5	43703.74	77990.99	73619.92	76002.94	75498.64	43806.52	45620.93	43522.37	78164.15	**43473.78**
k_6	43869.26	72566.78	72328.10	72205.36	76865.30	43456.96	**41880.32**	43590.68	77658.13	45636.19
k_7	46012.59	78663.52	77936.00	77118.60	77992.84	44143.39	42348.40	**41825.71**	81483.76	47466.31
k_8	77035.48	43881.85	43373.04	43335.58	44683.38	79560.93	82668.50	78333.77	**41734.19**	73317.93
k_9	45660.03	76732.59	71703.58	72773.38	75298.78	44192.78	43803.73	**42688.00**	74601.04	45757.45

$$
\begin{aligned}
N(k_0) &= [\mathbf{k_0},\ k_7,\ k_5,\ k_6,\ k_9,\ k_3,\ k_2,\ k_8,\ k_1,\ k_4] & \mathrm{Rank}(k_0) &= 0\\
N(k_1) &= [k_2,\ k_8,\ k_3,\ \mathbf{k_1},\ k_4,\ k_5,\ k_7,\ k_0,\ k_9,\ k_6] & \mathrm{Rank}(k_1) &= 3\\
N(k_2) &= [k_3,\ \mathbf{k_2},\ k_1,\ k_8,\ k_4,\ k_9,\ k_0,\ k_7,\ k_5,\ k_6] & \mathrm{Rank}(k_2) &= 1\\
N(k_3) &= [k_1,\ k_4,\ \mathbf{k_3},\ k_8,\ k_2,\ k_0,\ k_9,\ k_7,\ k_5,\ k_6] & \mathrm{Rank}(k_3) &= 2\\
N(k_4) &= [k_2,\ k_3,\ \mathbf{k_4},\ k_8,\ k_1,\ k_0,\ k_9,\ k_5,\ k_7,\ k_6] & \mathrm{Rank}(k_4) &= 2\\
N(k_5) &= [k_9,\ k_7,\ k_0,\ \mathbf{k_5},\ k_6,\ k_2,\ k_4,\ k_3,\ k_1,\ k_8] & \mathrm{Rank}(k_5) &= 3\\
N(k_6) &= [\mathbf{k_6},\ k_5,\ k_7,\ k_0,\ k_9,\ k_3,\ k_2,\ k_1,\ k_4,\ k_8] & \mathrm{Rank}(k_6) &= 0\\
N(k_7) &= [\mathbf{k_7},\ k_6,\ k_5,\ k_0,\ k_9,\ k_3,\ k_2,\ k_4,\ k_1,\ k_8] & \mathrm{Rank}(k_7) &= 0\\
N(k_8) &= [\mathbf{k_8},\ k_3,\ k_2,\ k_1,\ k_4,\ k_9,\ k_0,\ k_7,\ k_5,\ k_6] & \mathrm{Rank}(k_8) &= 0\\
N(k_9) &= [k_7,\ k_6,\ k_5,\ k_0,\ \mathbf{k_9},\ k_2,\ k_3,\ k_8,\ k_4,\ k_1] & \mathrm{Rank}(k_9) &= 4
\end{aligned}
$$

Table 3. Distances and ranks among key profiles for 64-rounds TEA (2M local optima profiled).

to increase T up to 10^6 to achieve a rank of 0 for all the keys (distances for this value are not shown here.) In the case of 64-rounds, $2 \cdot 10^6$ profiles are enough to obtain significant results (see Table 3).

4 Conclusions

In this paper, we have introduced a novel technique to carry out distinguishing attacks against block ciphers. Roughly, the key idea is to characterise the fitness landscape induced by a search technique when applied to recover some unknown information, e.g. the key. As presented here, the procedure used to profile the set of local optima is quite rough. Averaging the obtained local optima into a single vector results in a loss of huge amounts of information which might be certainly useful. An indication of this is the improvement reached when measuring correlations between key bits rather than simply computing distances between profiles. More precise characterisations of the local optima (or, more

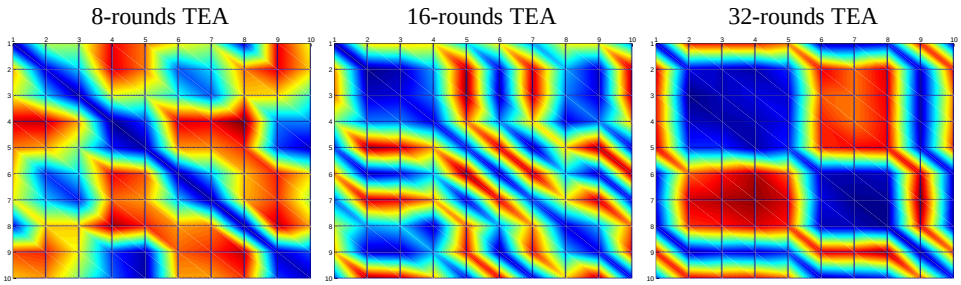

Fig. 3. Distances and ranks among key profiles for 32-rounds TEA (2M local optima profiled and 2M local optima profiled).

generally, of the fitness landscape), along with more effective procedures for measuring dissimilarity between profiles, are likely to improve considerably the accuracy of the technique.

Much more interesting, however, would be to identify any form of correlation between the local optima produced by the search and (some bits of) the specific key used. Such correlations would be a remarkable result, for it could be of help to recover portions of the key. Obtaining local optima correlated with the sought secret has been used to break zero knowledge schemes [1]. Even though the correlation there was very strong, correlations for block cipher analysis are likely to be much more subtle. In the same paper the authors also show how monitoring the trajectories taken by the search process can reveal even more information about the secret than the final optima. Again, such approaches deserve further consideration.

The most obvious improvements to our work would appear to lie in the development of more sophisticated profiling and distance measures. Ours are most rudimentary, but in a sense, this is a warning. **We know of no cipher that has been designed to be resilient against attacks of the form demonstrated in this paper.** It has generally been believed (even by cryptography researchers who use guided search regularly) that the highly discontinuous nature of modern cryptographic systems will protect against guided search attacks. These beliefs are well founded, in the sense that the search algorithms are extremely unlikely to produce the real key as an output. However, it is simply an act of faith to conclude that *the application of search will produce no use information*. As this paper demonstrates, this may simply not be the case. These results suggest that the cryptography community must seriously reconsider the potential of search based techniques for modern-day cryptanalysis tasks. The authors are currently investigating the application of similar techniqiues to DES and AES.

References

1. J.A. Clark and J.L. Jacob. “Fault Injection and a Timing Channel on an Analysis Technique.” *EUROCRYPT 2002*, LNCS Vol. 2332, pp. 181–196. Springer-Verlag, 2002.
2. T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.
3. R. Garcia-Pelayo and P.F. Stadler. “Correlation length, isotropy, and meta-stable states.” *Physica D*, **107**:240–254, 1997.
4. J. Garnier and L. Kallel. “Efficiency of local search with multiple local optima.” *SIAM Journal on Discrete Mathematics*, **15**(1):122–141, 2002.
5. J.C. Hernandez, J.M. Sierra, A. Ribagorda, B. Ramos, and J.C. Mex-Perera. “Distinguishing TEA from a random permutation: Reduced round versions of TEA do not have the SAC or do not generate random numbers.” *Proc. IMA Int. Conf. on Cryptography and Coding*, LNCS Vol. 2260, pp. 374–377. Springer-Verlag, 2001.
6. J.C. Hernandez, J.M. Sierra, P. Isasi, and A. Ribagorda. “Genetic cryptanalysis of two rounds TEA.” *ICCS 2002*, LNCS Vol. 2331, pp. 1024–1031. Springer-Verlag, 2002.
7. J.C. Hernandez, P. Isasi, and A. Ribagorda. “An aplication of genetic algorithms to the cryptanalysis of one round TEA.” *Proc. 2002 Symposium on Artificial Intelligence and its Application*, 2002.
8. J.C. Hernandez, J.M. Sierra, P. Isasi, and A. Ribagorda. “Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA.” *Proc. 2003 Congress on Evolutionary Computation*. IEEE Press, 2003.
9. S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee, and S. Lee. “Differential cryptanalysis of TEA and XTEA.” *ICISC 2003*. LNCS Vol. 2971, pp. 402–417. Springer-Verlag, 2003
10. J. Kelsey, B. Schneier, and D. Wagner. “Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES.” *CRYPTO 1996*, LNCS Vol. 1109, pp. 237–251. Springer-Verlag, 1996.
11. J. Kelsey, B. Schneier, and D. Wagner. “Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X NewDES, RC2, and TEA.” *ICICS 1997* LNCS Vol. 1334, pp. 233–246. Springer-Verlag, 1997.
12. D. Moon, K. Hwang, W. Lee, S. Lee, and J. Lim. “Impossible differential cryptanalysis of reduced round XTEA and TEA.” *FSE 2002*. LNCS Vol. 2365, pp. 49–60. Springer-Verlag, 2002.
13. R.M. Needham and D.J. Wheeler. “TEA extensions.” Technical Report, Computer Laboratory, University of Cambridge. October 1997.
14. G.A.F. Seber. *Multivariate Observations*. Wiley, 1984.
15. P.F. Stadler and W. Schnabl. “The landscape of the traveling salesman problem.” *Phys. Letters A*, **161**:337–344, 1992.
16. P.F. Stadler. “Landscapes and their correlation functions.” *J. Math. Chem.*, **20**:1–45, 1996.
17. E.D. Weinberger. “Correlated and uncorrelated fitness landscapes and how to tell the difference.” *Biological Cybernetics*, **63**:325–336, 1990.
18. D.J. Wheeler and R.M. Needham. “TEA, a tiny encryption algorithm.” *FSE 1994*. LNCS Vol. 1008, pp. 363–366. Springer-Verlag, 1994.
19. D.J. Wheeler and R.M. Needham. “Correction to XTEA.” Technical Report, Computer Laboratory, University of Cambridge. October 1998.