Characteristics of Key-Dependent S-Boxes: the Case of Twofish

Marco Macchetti

Politecnico di Milano, Milan, Italy
macchett@elet.polimi.it

Abstract. In this paper we analyze and discuss the cryptographic robustness of key-dependent substitution boxes (KDSBs); these can be found in some symmetric-key algorithms such as Khufu, Blowfish, and the AES finalist Twofish. We analyze KDSBs in the framework of composite permutations, completing the theory developed by O'Connor. Under the basic assumption that KDSBs are built choosing permutations randomly from the symmetric group S_{2^m} by means of the key, the expressions of their linear and differential characteristics are derived. These results are used as a statistical tool to show that Twofish KDSBs, although very efficient, can be easily distinguished from truly randomly built KDSBs. We also analyze the motivations that lead to this previously unknown property; it can be concluded that the efficiency of the construction and the small computational complexity of Twofish KDSBs, although very desirable, cannot be easily obtained together with the highest level of security.
Keywords: key-dependent s-boxes, linear cryptanalysis, differential cryptanalysis, composite permutations, Twofish.

1 Introduction

Block ciphers are an important and widely studied class of cryptographic algorithms; the obsolete Data Encryption Standard (DES) [1] and the established Advanced Encryption Standard (AES) [2] are two well known examples.

Practically all the proposed algorithms are constructed as Substitution Permutation Networks (SPNs) or Feistel structures; in both cases the importance of the non-linear part of the algorithm is crucial, especially considering the cryptanalytic techniques known as Differential [6] and Linear Cryptanalysis [8].

The non-linear substitution layer, corresponding to the *confusion* step as referred to by Shannon [18], is commonly realized with non-linear permutations acting at the byte (or similar) level, and simply called s-boxes. The construction of differential and linear trails for the whole block cipher strongly depends on the cryptographic characteristics of the s-boxes being used [7].

Although the majority of block ciphers contain only pre-specified s-boxes, several proposed algorithms use information derived from the key to customize the s-boxes, and make them secret. This happens in Khufu [9], in SEAL [14] where the hash function SHA [3] is used as a cryptographic primitive and again in Blowfish [15], which repeatedly calls itself to generate the required look-up tables. Obviously, such constructions heavily affect the performance level and the key-agility property, since complex calculations must be carried out to update the substitution values each time the cipher key is changed.

More efficient KDSB constructions are however possible: the KDSBs contained in the AES finalist algorithm Twofish [16] are characterized by a much lighter structure; they are built starting from simple operations (with regards to both hardware and software implementations) and contribute to the high-level performance of the cipher [17]. This is also the case for a strengthened version of DES [5], where key-dependency is added by simply XORing key material before and after the s-boxes calculation.

KDSBs can be a crucial component for a block cipher, even if it seems that the advantage of using KDSBs is dependent on the specific algorithm. In the case of Twofish, it has been shown that a version with fixed s-boxes is weaker [16]. On the other side, in [5] it is shown that randomly selected s-boxes can weaken the DES algorithm with high probability. In the former case, the high-level motivation is that since the attacker does not know the substitution values, he cannot construct characteristics through multiple rounds of the cipher without knowledge of the key. For this to be rigorously true, the KDSBs should resemble as much as possible a set of randomly picked permutations, in order to leak no information whatsoever regarding their structure to the attacker. We will hereafter refer to this concept as the *randomness hypothesis*.

In Sect. 2 a quantitative approach for the above line of reasoning is given, considering KDSBs as instances of composite permutations [13]: information derived from the key acts as the control prefix that selects the particular permutation being used. Differential characteristics of composite permutations have been partially studied in [13], while linear characteristics are analyzed here for the first time. Our contribution is to establish the exact shape of the distribution of values inside the Linear Approximation Table (LAT) and the Difference Distribution Table (DDT) of a randomly built KDSB.

In Sect. 3, we focus on the particular case of Twofish; we run a χ^2 test using random samples from the global DDT and LAT of each KDSB in the algorithm; a comparison with the expected outcomes shows that the probability that they approximate randomly selected sets of permutations is negligibly small, i.e. the test rejects the randomness hypothesis with high confidence level. This contradicts a design goal highlighted in the Twofish design rationale and shows that to verify the randomness hypothesis it is not sufficient to run tests on the cryptographic robustness of the single permutations; instead, KDSBs should be globally analyzed, and tested, as composite permutations.

In Sect. 4 we give an explanation for the deviation of Twofish KDSBs from the expected behaviour. The bits of the control prefix do not act uniformly within each KDSB; some of them operate on the KDSB structure nearer to the output of the permutation than others are. This eventually has the effect of introducing high values in the global LAT and DDT tables.

Section 5 concludes the paper.

2 Cryptographic Characteristics of KDSBs

Following [13], let $\rho : {F_2}^c \times {F_2}^m \rightarrow {F_2}^m$ be a mapping that consists of c control bits and m data bits. The ρ mapping contains 2^c m-bit permutations $\pi_i : {F_2}^m \rightarrow {F_2}^m$, $0 \leq i \leq 2^c - 1$; each different permutation is selected by a particular value of the control bits, referred to as the *control prefix*, and is used as a substitution table[1]. If all the c control bits are set to 0, we have a zero control prefix; in all other cases we say we have a non-zero control prefix.

To quantitatively evaluate the strength of KDSBs, we propose to regard them as composite permutations, where the bits directly or indirectly derived from the key constitute the control prefix. In the following, let $k \in {F_2}^c$ indicate the control prefix, $x \in {F_2}^m$ indicate the set of input bits and $y \in {F_2}^m$ indicate the set of output bits of ρ.

2.1 Linear Characteristics

The LAT of a vectorial Boolean function $f : {F_2}^m \rightarrow {F_2}^n$ is obtained by counting the number $\lambda_f(a, b)$ of solutions x of the equation

$$a \cdot x = b \cdot f(x) \qquad a \in {F_2}^m, b \in {F_2}^n \tag{1}$$

where the inner product is indicated with "$\cdot$" and gives a value in F_2. The robustness to linear cryptanalysis is then commonly measured with the maximum value $L_f = \max_{a,b \neq 0}(||\lambda_f(a, b) - 2^{m-1}||)$.

In the case of KDSBs, x and the control prefix k are the function's inputs and y is the function's output; thus, the equation becomes:

$$(a_p \cdot x) \oplus (a_k \cdot k) = b \cdot y \qquad a_p \in {F_2}^m, a_k \in {F_2}^c, b \in {F_2}^m \tag{2}$$

where sum in F_2 is indicated with $\oplus$ and the mask a to be applied to the input vector of ρ is split into the sub-mask a_p applied to the input vector x and the sub-mask a_k applied to the control prefix k. If we denote the concatenation of two vectors over F_2 with "$|$" then $a = a_k|a_p$.

The main difference between (1) and (2) is the presence of the control prefix k; to calculate the LAT of a KDSB, x and k are run over all the possible combinations of values, and the number of solutions of (2) is recorded in a specific cell, indexed by the values of a and b.

We first analyze the case $a_k \neq 0$. Under this hypothesis, $a_k \cdot k$ is a balanced function, and for 2^{c-1} values of the control prefix k, it will result $a_k \cdot k = 0$; for the remaining 2^{c-1} values we have $a_k \cdot k = 1$. The subset of values of the control prefix for which $a_k \cdot k = 0$ depends on the particular value of a_k, and is denoted with $C_0^{a_k}$; all the remaining values of k belong to $C_1^{a_k}$.

Therefore, (2) is split into the two equations:

$$(a_p \cdot x) \oplus (b \cdot y) = 0 \qquad \forall k \in C_0^{a_k} \tag{3}$$

[1] For instance, the eight DES s-boxes are examples of composite permutations with 2 control bits and 4 data bits.

$$(a_p \cdot x) \oplus (b \cdot y) = 1 \qquad \forall k \in C_1^{a_k} \tag{4}$$

Half of the permutations π_i are selected by the values of $k \in C_0^{a_k}$ and will be used to derive the values of y from those of x in (3), the other half being used to solve (4). The subset of all permutations selected by $k \in C_0^{a_k}$ is denoted with P_0^k, the remaining ones belonging to P_1^k.

Let us denote with $\Lambda_\rho(a, b)$ the value of a cell in the LAT of ρ indexed by a, b obtained by solving (2) and with $\lambda_{\pi_i}(a_p, b)$ the value in a cell of the LAT of π_i indexed by a_p, b. Then:

$$\begin{aligned} \Lambda_\rho(a, b) &= \sum_{\pi_i \in P_0^k} \lambda_{\pi_i}(a_p, b) \quad + \sum_{\pi_i \in P_1^k} (2^m - \lambda_{\pi_i}(a_p, b)) = \\ &= 2^{m+c-1} + \sum_{\pi_i \in P_0^k} \lambda_{\pi_i}(a_p, b) \quad - \sum_{\pi_i \in P_1^k} \lambda_{\pi_i}(a_p, b) \end{aligned} \tag{5}$$

The first and the second sum give respectively the total number of solutions of (3) and (4).

Since what normally matters in linear cryptanalysis are not the absolute values of the LAT cells, but the differences from the average values (2^{m-1} in case of λ_{π_i} and 2^{c+m-1} in case of Λ_ρ), (5) is re-written as

$$\widehat{\Lambda}_\rho(a, b) = \sum_{\pi_i \in P_0^k} \hat{\lambda}_{\pi_i}(a_p, b) \quad - \sum_{\pi_i \in P_1^k} \hat{\lambda}_{\pi_i}(a_p, b) \tag{6}$$

where hats denote the fact that the differences from the respective average values are taken.

In the remaining case $a_k = 0$, we always have $a_k \cdot k = 0$ and (2) becomes:

$$\widehat{\Lambda}_\rho(a, b) = \sum_{\pi_i} \hat{\lambda}_{\pi_i}(a_p, b) \tag{7}$$

Equations (6) and (7) state that the values in the LAT of a composite permutation can be obtained by simply adding the values in the LATs of the single permutations; the relations are valid regardless of the way in which the single permutations are selected.

In the case of KDSBs, let us suppose that the permutations π_i are randomly picked from the symmetric group S_{2^m}; the statistical distribution of Λ_ρ can thus be derived under the randomness hypothesis. We use the result of Youssef and Tavares [20], who give the distribution of the linear characteristics of a random permutation as:

$$\Pr[\lambda_{\pi_i}(a_p, b) = 2l] = \frac{\binom{2^{m-1}}{l}^2 (2^{m-1}!)^2}{2^m!} \tag{8}$$

which is a unimodal and symmetric distribution and is equal to zero for odd values. For reasons of symmetry, the mean values are:

$$\mathrm{E}[\lambda_{\pi_i}(a_p, b)] = 2^{m-1} \tag{9}$$

$$\mathrm{E}[\hat{\lambda}_{\pi_i}(a_p, b)] = 0 \tag{10}$$

Moreover, the variance of $\hat{\lambda}_{\pi_i}(a_p, b)$ has the following expression:

$$\sigma^2[\hat{\lambda}_{\pi_i}(a_p, b)] = \sigma^2[\lambda_{\pi_i}(a_p, b)] = \sum_{l=0}^{2^{m-1}} (2l - 2^{m-1})^2 \frac{\binom{2^{m-1}}{l}^2 (2^{m-1}!)^2}{2^m!} \tag{11}$$

Analytical simplification carried out with Mathematica [4] leads to the simpler form:

$$\sigma^2[\hat{\lambda}_{\pi_i}(a_p, b)] = \frac{2^{2m-2}}{2^m - 1} \tag{12}$$

The distribution of values in the LAT of the KDSB can be obtained from (6), (7), (10) and (12) since for all values of a_k, $\widehat{\Lambda}_\rho(a, b)$ is the sum of 2^c random variables[2] that are independent and identically distributed as $\hat{\lambda}_{\pi_i}(a_p, b)$. The Central Limit Theorem can be applied and for sufficiently large values of c, say for $c \geq 4$, the distribution of $\widehat{\Lambda}_\rho(a, b)$ is very well approximated by a Gaussian distribution with

$$\mathrm{E}[\widehat{\Lambda}_\rho(a, b)] = 0 \tag{13}$$

$$\sigma^2[\widehat{\Lambda}_\rho(a, b)] = 2^c \frac{2^{2m-2}}{2^m - 1} \tag{14}$$

The standard deviation of $\widehat{\Lambda}_\rho(a, b)$ is remarkably near to the value $2^{\frac{c+m}{2}-1}$ that is the minimum possible value of L_f for a function with $c+m$ input bits and m output bits; functions globally reaching this minimum are called Bent [10].

According to the above results, 68% of the LAT cells of a randomly built KDSB are expected to have an absolute value less than $2^{\frac{c+m}{2}-1}$, i.e. the Bent limit, while 99.99% of the LAT cells are expected to have an absolute value not greater than $2^{\frac{c+m}{2}+1}$. These quantitative results can be usefully exploited to statistically verify the randomness hypothesis, i.e. that the KDSB under test is well approximated by a set of randomly selected permutations.

2.2 Differential Characteristics

The DDT of a given function $f : {F_2}^m \rightarrow {F_2}^n$ contains the number $\delta_f(a, b)$ of solutions $x \in {F_2}^m$ of the equation

$$f(x \oplus a) \oplus f(x) = b \qquad a \in {F_2}^m, b \in {F_2}^n \tag{15}$$

where $\oplus$ denotes a bitwise XOR operation. The lower the value of the maximum entry in the table, $D_f = \max_{a \neq 0, b}(\delta_f(a, b))$, the more robust is function f versus differential cryptanalysis. Let us denote with $\Delta_\rho(a, b)$ the value of a cell in the DDT of the composite permutation ρ indexed by a, b and obtained by solving (15) with ρ in place of f.

[2] Adding or subtracting i.i.d. random variables that are symmetrically distributed leads to the same result.

The differential characteristics of composite permutations were analyzed in [13], where, under the hypothesis that the π_i are randomly selected, it was proven that $\Delta_\rho(a,b)$ can be expressed as a sum of independent identically distributed variables:

$$\Delta_\rho(a,b) = 2\sum_{k=1}^{2^{c-1}} \theta_m \qquad \left\lfloor \frac{a}{2^m} \right\rfloor > 0 \tag{16}$$

$$\Delta_\rho(a,b) = \sum_{k=1}^{2^{c}} \Theta_m \qquad \left\lfloor \frac{a}{2^m} \right\rfloor = 0 \tag{17}$$

Equation (16) is applicable in case of non-zero control prefix, while (17) covers the cases with a zero control prefix. The exact distributions of θ_m and Θ_m are characterized by:

$$\Pr[\theta_m = k] = \frac{1}{k!}\sum_{i=0}^{2^m-k} \frac{(-1)^i}{i!} \tag{18}$$

$$\Pr[\Theta_m = 2k] = \binom{2^{m-1}}{k}^2 \frac{k! 2^k \Phi(2^{m-1}-k)}{2^m!} \tag{19}$$

$$\Phi(d) = \sum_{i=0}^{d} (-1)^i \binom{d}{i}^2 2^i i!(2d-2i)! \tag{20}$$

These results were used by O'Connor to derive the maximum value and the fraction of null entries in the DDT of a composite permutation; this information is interesting from a cryptanalytic point of view, but to be able to verify the randomness hypothesis we are more interested in determining the global shape of the distribution of the DDT values. Due to the complexity of the above expressions, no simple closed form could be *provably* established for the mean and variance of Θ_m and θ_m; however, using the Mathematica program, the following equalities were verified with analytical calculations for $1 \le m \le 9$:

$$\mathrm{E}[\theta_m] = 1 \tag{21}$$

$$\sigma^2[\theta_m] = 1 \tag{22}$$

$$\mathrm{E}[\Theta_m] = \frac{2^m}{2^m-1} \simeq 1 \tag{23}$$

$$\sigma^2[\Theta_m] = \frac{2^{m+3}(2^{m-1}-1)^2}{(2^m-3)(2^m-1)^2} \simeq 2 \tag{24}$$

We conjecture these expressions to be true for all values of m. The approximate values of 1 and 2, respectively for the mean value and the variance of Θ_m, will be used in the following; for $m = 8$ the error is less than 1%.

Combining the above results, and again applying the Central Limit Theorem, it can be concluded that, for both zero and non-zero prefixes, $\Delta_\rho(a,b)$ is approximately characterized by a Gaussian distribution with

$$\mathrm{E}[\Delta_\rho(a,b)] = 2^c \tag{25}$$
$$\sigma^2[\Delta_\rho(a,b)] = 2^{c+1} \tag{26}$$

for sufficiently large values of c, say $c \geq 4$. The mean of the distribution is equal to the minimum possible value of D_f for a function with $c+m$ input bits and m output bits; functions globally reaching this minimum are called Perfect Non-Linear [11], and have been proven to be the same as Bent functions in Finite Fields of characteristic 2.

Quantitatively, 50% of the DDT cells of a KDSB are expected to contain a value less than 2^c, i.e. the Perfect Non-Linear limit, and 99.99% of the cells are expected to contain a value not greater than $2^c + 2^{\frac{c+5}{2}}$. Equations (25) and (26), together with (13) and (14), fully characterize the cryptographic robustness of random KDSBs.

3 The Case of Twofish

3.1 Review of the Twofish Encryption Algorithm

Twofish was designed in 1998 by the Twofish Team (Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson) and was one of the AES finalists. The algorithm has interesting characteristics: the high speed of computation and the flexibility to have different trade-offs between costs of implementations and performance on a wide range of platforms, are some key factors to be positively considered.

Twofish is a 128-bit block cipher with an iterated 16 rounds Feistel-like structure, a single round comprising different transformations: a layer of non-linear KDSBs, a diffusion layer realized with an MDS mapping, a pseudo-Hadamard transform and an addition with the round-key. Different operators such has modular integer additions on 32-bit words and constant multiplications in Galois Fields are mixed in order to achieve the desired level of robustness.

There are four different KDSBs in Twofish[3]; the value of m is 8, i.e. the permutations operate at byte level. The actual amount of key information injected into the KDSBs depends on the dimension of the cipher key: two bytes are used per each KDSB in the 128-bit key variant, three in the 192-bit and four in the 256-bit one; the values of c are, respectively, 16, 24 and 32.

The four KDSBs are characterized by the same structure; the main building blocks are the fixed byte-level permutations q_0 and q_1. In each KDSB, instances of q_i are alternated with XORs with the key-derived bytes; different orderings of the q_i instances are sufficient to produce different sets of permutations and eventually different round functions, as was proven by the designers [19]. The following equations describe how the four KDSBs are obtained in the 128-bit

[3] Note that each KDSB is instantiated twice within the round function.

key case:

$$s_0(x) = q_1(q_0(q_0(x) \oplus k_{0,0}) \oplus k_{1,0}) \quad (27)$$

$$s_1(x) = q_0(q_0(q_1(x) \oplus k_{0,1}) \oplus k_{1,1}) \quad (28)$$

$$s_2(x) = q_1(q_1(q_0(x) \oplus k_{0,2}) \oplus k_{1,2}) \quad (29)$$

$$s_3(x) = q_0(q_1(q_1(x) \oplus k_{0,3}) \oplus k_{1,3}) \quad (30)$$

In the above equations, x is the KDSB input, $s_i(x)$ is the KDSB output and $k_{i,j}$ are bytes derived from the cipher key bytes. The actual robustness to differential and linear cryptanalysis is guaranteed by the q_0 and q_1 fixed permutations, which are selected under constraints that $L_{q_i} = 32$ and $D_{q_i} = 10$.

3.2 Testing Twofish KDSBs

In [16] the designers of Twofish propose several statistical tests to support the evidence that the above KDSBs are indistinguishable from sets of randomly selected permutations; different cryptographic properties of the individual byte-level permutations have been calculated and good correspondence has been found for the number of fixed points, the distribution of the D_f and L_f values and the mutual correlation.

However, no data on the global DDT and LAT of the KDSBs (considered as composite permutations) has ever been published; if the randomness hypothesis is verified, the parameters $\widehat{\Lambda}_{s_i}(a,b)$ and $\Delta_{s_i}(a,b)$ would be expected to follow a normal distribution with mean and variance as given in (13), (14), (25) and (26). We think this is a crucial test for any algorithm employing KDSBs, and a significant lack of analysis with regards to the Twofish algorithm.

In order to perform the test, we collected random samples from the DDT and LAT of each KDSB in the 128-bit key case. Figure 1 shows a comparison between a random set of 2^{16} values from the DDT of s_0 (light bars) and the expected theoretical distribution from Sect. 2 (thick curve).

To enhance the readability of the graphs, the values have been grouped into bins. Each bin B_i is indexed by a number i and contains all the DDT cells whose value is such that:

$$B_i = \{\Delta_{s_0}(a,b) \quad | \quad \lfloor \frac{\Delta_{s_0}(a,b)}{2^8} \rfloor = i\}$$

The bin indexes are shown on the x-axis and the cardinality of each bin is shown on the y-axis, in logarithmic scale.

Several differences between the real and the expected distribution are visually noticeable, namely:

1. The real distribution is not unimodal; apart from a central region, which appears quite regular although not completely symmetrical, several empty bins can be found between non-empty bins.
2. The real distribution is wider than expected; one exceptional high value belonging to B_{1536}, whose existence probability should be negligibly small, has been found and displayed separately in the bottom graph.

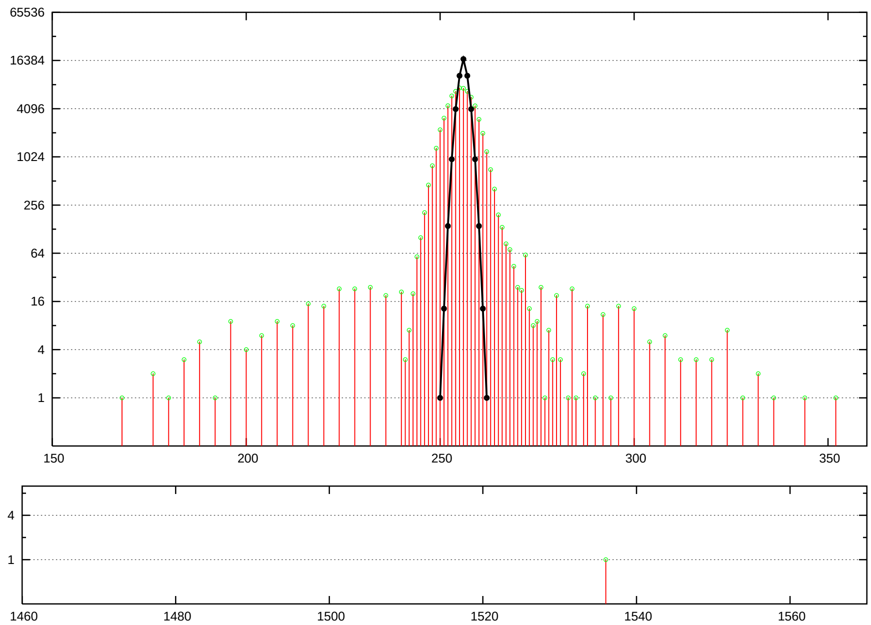

Fig. 1. A random sample of DDT values of s_0 versus the expected distribution.

3. The cardinalities of the bins are different than expected. The cardinality of bin B_{256}, i.e. the maximum of the distribution, is less than half of what is expected.

Figure 2 contains an analogous comparison between a sample of 2^{15} values from the LAT of s_0 (light bars) and the expected theoretical distribution from Sect. 2 (black curve). In this case the bins are defined as:

$$B_i = \{\widehat{\Lambda}_{s_0}(a,b) \quad | \quad \lfloor \frac{\widehat{\Lambda}_{s_0}(a,b)}{2^8} \rfloor = i\}$$

Similar qualitative considerations can be made also in this case.

We have carried out χ^2 tests using the above data to validate the randomness hypothesis; the bins have been used as data classes and the bins in the tails of the distributions have been merged to enhance the accuracy of the tests.

The results are the same for both the DDT and the LAT values: the probability that the given samples are taken from the expected distributions is certainly less than 10^{-6} or, in other words, chances that the randomness hypothesis is verified for KDSB s_0 are less than one in a million. The same quantitative results have been obtained for s_1, s_2 and s_3 although the corresponding graphs are not shown for space reasons.

Samples have not been collected for the 192-bit and 256-bit cases; however, in the next Section we will explain why the randomness hypothesis is provably not verified for all key sizes.

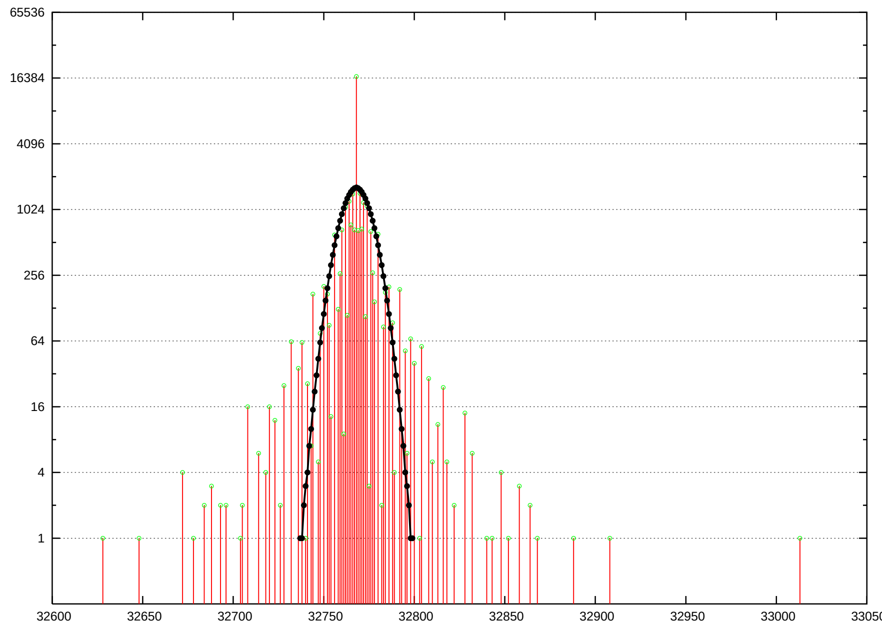

Fig. 2. A random sample of LAT values of s_0 versus the expected distribution.

It is now obvious that one of the design goals for Twofish has not been reached. For instance, the presence of high values in the DDT of the KDSBs means that it is more probable to obtain particular differences in the s-boxes output bytes starting from certain difference values in the input bytes *and* the key bytes than it is for randomly built KDSBs.

However, to be able to use this unwanted bias to mount a differential attack to the cipher, one would need to control input differences along with *key* differences. We believe it is very difficult to achieve this, apart maybe for some kind of related-key attack.

On the other hand, more concern should be put at the maximum entry values in the LATs of the KDSBs, considered as composite permutations. The reason for this is that the KDSBs actually mix key information with the input data. The designers of Twofish show that there are no high values in the LATs of the individual permutations generated by each KDSB. However, this may not be sufficient because the final goal of linear cryptanalysis is to find linear relations between data bits and key bits that may be valid with a probability significantly different from 50%; since in Twofish information from the key is actually injected into the KDSBs, high values in the LATs of the latter might lead exactly to these relations.

We do not know if the above results can be used to mount a linear attack to the cipher; anyway, it is probable that more cryptanalysis effort starting from these considerations would be needed to clearly answer the question.

4 KDSB Efficiency vs. KDSB Security

The reason why the randomness hypothesis is not verified in Twofish can be easily understood by looking at the KDSBs construction.

For all key-sizes, each KDSB is the result of a concatenation of fixed permutations and XORs with a certain amount of key material; as already said, the key information is equivalent to the control prefix in composite permutations. To verify the randomness hypothesis, it is important that every bit of the control prefix acts on the KDSB in the same way, i.e. the influence of all the key bits must be *uniform* with regards to differential and linear properties of the structure.

Let us consider the differential case; it can be seen from Fig. 1 that there are unexpected high values in some cells of the DDT of s_0. One possible explanation is that some differences in the input byte and the key bytes lead to exceptionally high frequencies for the output differences.

To show that this is true, (15) is applied directly on the structure of the 128-bit key length variant of KDSB s_0, i.e. (27); moreover, let us denote respectively with d_0, d_1, d_2 and d_3 the differences in bytes x, $k_{0,0}$, $k_{1,0}$ and $s_0(x)$ and take the particular family of differences for which $d_0 = d_1 = 0$. We therefore obtain:

$$\begin{aligned}\Delta_{s_0}(d_2, d_3) = \#\{(x, k_{0,0}, k_{1,0}) : q_1(q_0(q_0(x) \oplus k_{0,0}) \oplus k_{1,0}) \oplus \\ \oplus q_1(q_0(q_0(x) \oplus k_{0,0}) \oplus k_{1,0} \oplus d_2) = d_3\}\end{aligned} \tag{31}$$

Since the input difference is only applied to the second key byte $k_{1,0}$ and the value $f(x) = q_0(q_0(x) \oplus k_{0,0}) \oplus k_{1,0}$ is fixed for each couple of inputs with the given difference, (31) can be rewritten as:

$$\Delta_{s_0}(d_2, d_3) = \#\{(x, k_{0,0}, k_{1,0}) : q_1(f(x)) \oplus q_1(f(x) \oplus d_2) = d_3\} \tag{32}$$

and thus:

$$\Delta_{s_0}(d_2, d_3) = 2^{16}\delta_{q_1}(d_2, d_3) \tag{33}$$

The fact that $f(x)$ is always a bijective function of x implies that the differential characteristics of s_0, for the considered family of differences, are exactly equal to the differential characteristics of q_1 amplified by a factor of 2^{16}. Since q_1 has some differential characteristics equal to 10, this means that there are even some values of Δ_{s_0} belonging to bin B_{2560}; the value belonging to bin B_{1536} in Fig. 2 can be explained as above, starting from a differential characteristic of 6 for q_1.

This problem is present for all Twofish s-boxes and for all key sizes, since (33) can always be written for the last fixed s-box belonging to each KDSB. Thus, this characteristic inherently depends on a design choice of the algorithm and finally invalidates the randomness hypothesis.

It can be concluded that a good and secure construction for a KDSB contrasts with the high level of performance goal that is pursued by Twofish's designers. Other means to efficiently use key information to generate truly random s-boxes are needed, if the security of the construction is a primary objective.

5 Conclusions

In this paper we have discussed the cryptographic properties of key-dependent s-boxes (KDSBs); these can be viewed as particular instances of composite permutations, and treated accordingly to the existing theory due to O'Connor. It has been shown that the entries in the LAT of any KDSB can be calculated starting from the LATs of the single permutations; the shape of the distributions of the global LAT and DDT values have been calculated for randomly built KDSBs. These quantitative results can be used as a valid statistical test to verify if the *randomness hypothesis* is verified for a given KDSB construction. We have shown this is not the case for the AES finalist algorithm Twofish. Future work may include the proposal of an efficient and secure KDSB construction.

References

1. Data Encryption Standard (DES), NIST FIPS publication 46-3, 1999.
2. Announcing the Advanced Encryption Standard, NIST FIPS publication 197, 2001.
3. Announcing the Secure Hash Standard, NIST FIPS publication 180-2, 2002.
4. Wolfram Research, http://www.wolfram.com/
5. E. Biham, A. Biryukov, How to Strengthen DES Using Existing Hardware, Proceedings of Asiacrypt'94, 398-412, 1994.
6. E. Biham, A. Shamir, Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, 4(1):3–72, 1991.
7. J. Daemen, V. Rijmen, The Design of Rijndael: AES-The Advanced Encryption Standard, Springer-Verlag, 2002.
8. M. Matsui, Linear Cryptanalysis method for DES cipher, Proceedings of Eurocrypt'93, 386–397, 1994.
9. R.C. Merkle, Fast Software Encryption Functions, Proceedings of Crypto'90, 476-501, 1991.
10. K. Nyberg, Constructions of Bent Functions and Difference Sets, Proceedings of Eurocrypt'90, 151–160, 1991.
11. K. Nyberg, Perfect Nonlinear S-Boxes, Proceedings of Eurocrypt'91, 378–386, 1991.
12. L. O'Connor, On the Distribution of Characteristics in Bijective Mappings, Proceedings of Eurocrypt'93, 360–370, 1994.
13. L. O'Connor, On the Distribution of Characteristics in Composite Permutations, Proceedings of Crypto'93, 403–412, 1994.
14. P. Rogaway, D. Coppersmith, A Software-Optimized Encryption Algorithm, Proceedings of the 1993 Cambridge Security Workshop, 1994.
15. B. Schneier, Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish), Proceedings of the 1993 Cambridge Security Workshop, 191–204, 1994.
16. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, Twofish: a 128-bit Block Cipher, AES proposal, 1998.
17. B. Schneier, D. Whiting, A Performance Comparison of the Five AES Finalists, Third AES Candidate Conference, 2000.
18. C.E. Shannon, Communication Theory of Secrecy Systems, Bell System Technical Journal, 28:656–715, 1949.
19. D. Whiting, B. Schneier, Empirical Verification of Twofish Key Uniqueness Properties, Twofish Technical Report 2, 1998.
20. A.M. Youssef, S.E. Tavares, Resistance of balanced s-boxes to linear and differential cryptanalysis, Information Processing Letters 56, 249–252, 1995.