On insecurity of 4-round Feistel ciphers

PAVOL ZAJAC
Slovak University of Technology
Institute of Computer Science and Mathematics
Ilkovičova 3, 812 19 Bratislava
SLOVAKIA
pavol.zajac@stuba.sk

Abstract: There have been various proposals of lightweight ciphers for resource constrained devices, including proposals that use only 4 rounds of a Feistel cipher, similar to DES. In this article we show that 4-round DES-like cipher is inherently insecure with a practical attack based on impossible differentials.

Key–Words: Feistel cipher, cryptanalysis, DES

1 Introduction

Many applications in the areas of mobile computing, e-health services, wireless sensor networks, etc. require a simple way to secure communication channels. Standard cryptography provides robust and secure cryptographic primitives such as AES or its variants [4]. However, AES can be two complex/costly for some applications. The balancing issues between cipher security and implementation costs are the main object of study of the lightweight cryptography. Lightweight cryptography focuses on simple cipher designs that provides enough security with lower costs in hardware or software implementations.

The question of general lower bounds on implementation of ciphers with given security criteria is an open problem, although some results are known for specific types of Boolean functions that are important building blocks of the ciphers [5]. Thus, design of lightweight ciphers is mostly influenced by the known standard cipher designs, such as the Data Encryption Standard (DES). DES is now a historical cipher that is considered insecure due to short key and many existing attacks. There are however many promising lightweight variants of DES such as DESL [8].

In [10], an extremely lightweight version of DES is proposed with only 4 rounds of Feistel encryption. It is already known that such a design is insecure from the theoretical point of view [11], due to collision attacks. However, the complexity of these attacks is $O(2^{n/2})$, where n is the block size. Thus, it might seem that problems with 4-round DES-like cipher can be avoided by large enough block size. In this paper we show a practical key-reconstruction attack on generalized 4-round DES-like cipher. Its complexity depends on the S-box size, which cannot be increased too much (due to implementation constraints). Thus we show that 4-round DES-like ciphers are inherently insecure and should not be used in practice.

2 Definitions and Notation

Let $e : Z_2^{n_B} \times Z_2^{n_K} \to Z_2^{n_B}$ be a block cipher operating on n_B-bit blocks and having n_K bit key. We will construct function e as a composition of partial functions that denote the individual steps of the encryption algorithm.

We will call e a generalized DES cipher, if it has a Feistel structure, and its round function consists of bit expansion, key addition, S-box evaluation and bit permutation layers.

A cipher with Feistel structure works as follows:

1. Split the input string into left and right half,
2. Transform right part with a (key-dependent) round function F and XOR it into the left part,
3. Swap the two parts.

This is repeated r times, where each repetition of this process will be denoted as a round (of encryption).

Mathematically, let $x = (l|r)$ denote the input string. Then $y = (r|l \oplus F_k(r))$ is an output string of one Feistel round ($\oplus$ denotes XOR operation on bit strings).

Let $n_B = 2m$ for some positive integer m, and let $n \geq m$, and let s be an integer that divides both m, and n. Let $KS : Z_2^{n_K} \to (Z_2^n)^r$ denote a key schedule algorithm. This algorithm provides r subkeys $k^1, k^2, \ldots, k^r$ given a master key k. I.e., $KS(k) = (k^1, k^2, \ldots, k^r)$. Subkeys are used to define key dependent round functions for Feistel cipher.

Let $\sigma_i : Z_2^{n/s} \times Z_2^{m/s}$ denote any Boolean function. We define S-box layer (containing s parallel S-boxes) as a function $S : Z_2^n \times Z_2^m$, where

$$S(x_0, \ldots, x_{n/s-1}, x_{n/s}, \ldots, x_{n-1}) = \\ (\sigma_1(x_0, \ldots, x_{n/s-1}), \ldots, \\ \sigma_s(x_{n-n/s}, \ldots, x_{n-1})) .$$

I.e., we apply s S-boxes in parallel on (n/s)-bit substrings of the input, producing corresponding (m/s)-bit substrings of the output.

Let $\varepsilon : Z_n \to Z_m$, $n > m$, such that for every $y \in Z_m$, there is at least one $x \in Z_n$ such that $y = \varepsilon x$. Bit expansion function $E : Z_2^m \times Z_2^n$ is defined as

$$E(x_0, x_1, \ldots, x_{m-1}) = \left(x_{\varepsilon(0)}, x_{\varepsilon(1)}, \ldots, x_{\varepsilon(n-1)}\right) .$$

This means that each input bit is copied into output bits (in any order), and some of the input bits can occur in the output multiple times (are duplicated, triplicated, etc.).

Let $\pi : Z_m \to Z_m$ be a bijection. Bit permutation function $P : Z_2^m \times Z_2^m$ is defined as

$$E(x_0, x_1, \ldots, x_{m-1}) = \left(x_{\pi(0)}, x_{\pi(1)}, \ldots, x_{\pi(n-1)}\right) .$$

This means that each input bit is copied into output bits in the order prescribed by permutation π.

Round function $F : Z_2^m \times Z_2^n \to Z_2^m$ of a generalized DES cipher can be written as

$$F(x, k^i) = P(S(E(x) \oplus k^i)),$$

where $S : Z_2^n \times Z_2^m$ denotes the S-box layer, $E : Z_2^m \times Z_2^n$ is the bit expansion, and $P : Z_2^m \times Z_2^m$ is the bit permutation.

In non-mathematical terms, generalized DES is a Feistel cipher, with round function that first performs bit expansion (takes m input bits, and reorders/copies them to n output bits), XORs the expanded input with the round key, applies S-boxes in parallel, and finally mixes the output bits with bit permutation P.

In the following section we will show the attack on 4-round generalized DES using impossible differences.

3 Attack on 4-round generalized DES

Differential cryptanalysis was introduced by Biham and Shamir in 1991 [2]. They attack DES-like ciphers by studying the statistical distribution of differences during the encryption process. Classical differential cryptanalysis requires the knowledge of S-boxes to produce a statistical model of S-box differential response, i.e., the probability that a given change of S-box input produces a particular S-box output difference. When S-boxes are key dependent, or kept secret in other way, it is not possible to use standard techniques of differential cryptanalysis (such as computing the S-box difference table and searching for suitable differential trajectories, as described in [6]). Instead, we adapt a method of impossible differentials [3] in a way that does not require the knowledge of concrete S-boxes in generalized DES.

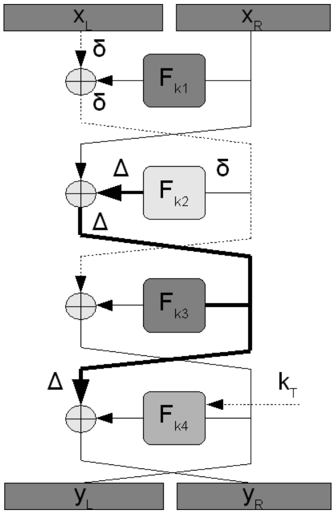

Figure 1: The propagation of difference in a 4-round Feistel scheme.

3.1 Attack overview

Let us study the response of Feistel cipher to a single bit change in the left half of input. The situation is depicted in Figure 1. First we encrypt any plaintext (x_L, x_R), getting ciphertext (y_L, y_R). The attacker chooses a single bit difference $\delta \in Z_2^m$, i.e., $w_H(\delta) = 1$. He then encrypts the plaintext $(x_L \oplus \delta, x_R)$, obtaining ciphertext

$$(y_L^*, y_R^*) = (y_L \oplus d_L, y_R \oplus d_R).$$

A good cipher should provide a strong avalanche effect, i.e., the output differences d_L and d_R should be unpredictable (with approximately one half of bits equal to zero, and one half equal to one).

If we study the encryption in more detail, we can see that in the first round the input to round function F_{k_1} is the same in both encryptions (x_R). Thus, the difference δ is unchanged, and is only swapped to the right side (and zero difference is swapped to the left side). In the second round the input to function F_{k_2} is different during the two encryptions, it differs exactly by the difference δ. If we do not know more details about the structure of F, we cannot predict how will the outputs of F_{k_2} differ. We will denote the output difference in this second round Δ. The difference Δ gets swapped to the right side, and the difference δ back to the left side. In the third round the input difference to F_{k_3} is Δ, which is unknown, thus we do not know the output difference of F_{k_3} as well. However, we know that Δ on right side is unchanged and gets swapped to the left side. In the fourth round difference Δ is further changed by the output difference of F_{k_4}.

Once the attacker obtains the ciphertexts and learns differences d_L, and d_R, he can propagate them backwards. Difference d_L is exactly the input difference of F_{k_4}, and we can see that $d_L \oplus d_3$ is the output difference of F_{k_3}. Difference d_R is the XOR sum of Δ and the output difference of F_{k_4}. Thus, if the attacker somehow knows subkey k_4, he can compute Δ in the following way:

$$\Delta = d_R \oplus F_{k_4}(y_L) \oplus F_{k_4}(y_L^*).$$

The difference Δ is an output difference of F_{k_2} provided a single bit input difference δ. If F has a DES-like structure described in Section 2, only some of differences Δ are possible. Let us denote a set of impossible differences Δ by $\mathcal{R}$, i.e.,

$$\mathcal{R} = \{\Delta; Pr_X\left(F_{k_2}(X) \oplus F_{k_2}(X \oplus \delta) = \Delta\right) = 0\}.$$

Given two P-C pairs $((x_L, x_R), (y_L, y_R))$, and $((x_L \oplus \delta, x_R), (y_L^*, y_R^*))$, attacker can use set $\mathcal{R}$ to quickly discard some of the potential subkeys used in the last round. The attacker chooses subkey value k_T, and computes

$$\Delta_T = d_R \oplus F_{k_T}(y_L) \oplus F_{k_T}(y_L^*).$$

If $k_T = k_4$, Δ_T cannot belong to set $\mathcal{R}$, otherwise there is a chance proportional to $R/2^m$ that Δ_T belong to $\mathcal{R}$. Thus, if $\Delta_T \in \mathcal{R}$, the attacker immediately knows that $k_T \neq k_4$. This allows the attacker quick computation of the last subkey, which is an efficient attack on cipher if the subkey leaks information about the key, and if the subkey is not longer than the full cipher key. However, if the cipher has DES-like cipher, we can do much better by studying the structure of function F in more details.

3.2 Characterization of impossible differentials

First, let us consider how the set $\mathcal{R}$ is constructed. The attacker chooses a single bit difference δ. Let us suppose that the bit which is changed has index i. The expansion function E propagates the change to all positions j such that $\varepsilon(j) = i$. The only S-boxes that are influenced by the change are those, where the change is propagated to. We call these S-boxes active, and other S-boxes inactive.

Suppose that $a = \min_i |\{j; \varepsilon(j) = i\}|$. The attacker will choose i in such a way that he gets at most a active S-boxes (and $s - a$ inactive S-boxes). When S-box is inactive, its inputs do not change between encryptions, thus also its outputs do not change, and its output difference contain only zero bits. The output difference bits for the active S-box can be both zero and one. We expect that the attacker do not know the S-boxes, thus we cannot (and do not need to) model the distribution of output differences from active S-boxes. Still, due to the presence of inactive S-boxes, we can be certain that the number of non-zero bits in difference is at most $a \cdot m/s$ (out of possible m bits). The zero-difference bits from the output of S-boxes are further distributed by permutation function P. A difference Δ, which has non-zero bit in a position that is an output of inactive S-box is impossible.

We can characterize the set $\mathcal{R}$ in a computation by a bit mask μ, which has 0 exactly in a position that is an output of active S-box, and 1 in a position that is an output of inactive S-box. We can quickly test whether $\Delta \in \mathcal{R}$: compute bitwise AND between Δ and μ. If it is non-zero, the difference Δ is impossible. Moreover, we can test Δ in parts: compute bitwise AND just between a selected bits of Δ and corresponding bits of μ. The non-zero result immediately tells us that Δ is impossible, regardless of the rest of the bits that were not tested.

3.3 Attack on the last subkey and S-boxes

Once we compute the mask μ, we can focus on the attack on the last round subkey. We want to find all values of k_T that do not lead to impossible differentials. We do not need to compute the whole output of F_{k_T} to discard some key, it suffices to find some part of Δ_T that can be compared with mask μ and discarded. Key k_T is XOR-ed to expanded input y_L, and y_L^*, respectively, before computing the output of S-boxes. Thus, to compute the m/s bits of Δ_T, we only need to guess n/s bits of k_T, and the contents of a single S-box. If the S-box is key-dependent, we guess the corresponding key bits that are used to generate the S-boxes. After computing the corresponding m/s bits of the difference Δ_T, we check it with the corresponding part of the mask μ. If the difference Δ_T is impossible, we know that the key guess was incorrect.

We can separate the search for a correct subkey and S-boxes: Just work with a single hypothesis for S-boxes, and try to find the correct subkey. If the S-box hypothesis is incorrect, the impossible differentials will eliminate all subkeys, otherwise a correct subkey will remain (and S-boxes are found).

For each part of the key, we test only $2^{n/s}$ hypotheses separately, for a total work of $s2^{n/s}$, instead of 2^n tests, which is an exponential speedup. E.g. for classical DES, $n = 48$, and to find the subkey using a whole Δ_T, we would need $2^{48} \approx 3 \cdot 10^{14}$ tests. If we test 4-bit blocks of Δ_T separately, we only need $8 \cdot 2^6 = 512$ tests. To prevent this attack, we would need to significantly increase the value n/s. However, the size of S-boxes is also exponential in n/s, thus this is not possible due to implementation constraints.

A single set of two P-C pairs $((x_L, x_R), (y_L, y_R))$, and $((x_L \oplus \delta, x_R), (y_L^*, y_R^*))$ provides only a partial reduction in possible key space. Suppose that we test 4-bit blocks ($m/s = 4$). If the corresponding part of mask μ is 0000, we cannot eliminate any key hypothesis. We try to avoid such blocks, or to test two or more blocks together in such a case (so that the corresponding mask does not contain only zeros). If the corresponding mask has a single bit equal to one, we can eliminate approximately half of hypotheses. If the mask has all ones, only approximately 1 out of 16 hypotheses is not eliminated. Furthermore, the attacker can provide a different sets of input P-C pairs, each of which will eliminate a fraction of remaining key hypotheses, until at most one will remain. The number of required sets of P-C pairs is logarithmic in the key space (comparable to n/s, instead of $2^{n/s}$). Thus the attack has very low complexity even for ciphers with large blocks and key sizes.

After the attack on the last round is successful, we can either reconstruct the original key (depending on the key schedule), or adapt the attack to a simpler 3-round structure.

4 Conclusion

We have shown a practical key recovery attack on generalized 4-round DES-like cipher. It can be concluded that similar designs are inherently insecure and should not be used in applications that require secure communication. Still, the 4-round construction has relatively good avalanche, thus it might be possible to use it in place where only avalanche effect and not strong security is required, e.g., in steganographic systems [7]. It is possible to strengthen the cipher by increasing the number of rounds, but it is not clear how many rounds are required to provide enough resistance against more sophisticated attacks than the one presented in this paper.

From the security point of view, we recommend to use standard ciphers, such as AES [9], instead of custom designs, and try to conserve resources in other parts of the system. Alternatively, it is possible to consider replacing block cipher with a fast and simple stream cipher [1] (see also [12] for stream cipher overview).

Acknowledgements: The research was supported by the Science Grant Agency - project VEGA 1/0173/13.

References:

[1] E. Antal, and V. Hromada, A New Stream Cipher Based on Fialka M-125, *Tatra Mountains Mathematical Publications*, 57(4), 2013, pp. 101–118

[2] E. Biham, and A. Shamir, Differential Cryptanalysis of DES-like Cryptosystems, *Journal of Cryptology*, 4(1), 1991, pp. 3–72.

[3] E. Biham, A. Biryukov, and A. Shamir, Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials, *Proceedings of EUROCRYPT'99*, LNCS 1592, 1999, pp. 12–23.

[4] O. Grosek, P. Zajac, Efficient selection of the AES-class MixColumns parameters, *WSEAS Transactions on Information Science and Applications* 4 (4), 2007, pp. 663–668.

[5] O. Grosek, P. Zajac, Graphs connected with block ciphers, *WSEAS Transactions on Information Science and Applications* 3 (2), 2006, pp. 439–443

[6] H. Heys, A tutorial on Linear and Differential Cryptanalysis, *Cryptologia*, 26(3), 2002, pp. 189–221.

[7] M. Jokay, The Design of a Steganographic System Based on the Internal MP4 File Structures, *International Journal of Computers and Communications* 5(4), 2012, pp. 207–214.

[8] G. Leander, C. Paar, A. Poschmann, and K. Schramm, New lightweight DES variants, *Fast Software Encryption*, Springer Berlin Heidelberg, 2007, pp. 196–210.

[9] National Institute of Standards and Technology, NIST FIPS PUB 197, *Advanced Encryption Standard*, U.S. Department of Commerce 2001.

[10] J. Pan, S. Li, and Z. Xu, Security mechanism for a wireless-sensor-network-based healthcare monitoring system, *IET Communications* 6 (18), 2012, pp. 3274–3280.

[11] J. Patarin, New Results on Pseudorandom Permutation Generators Based on the DES Scheme, *CRYPTO* 1991, pp. 301–312.

[12] M. Vojvoda, A Survey of Security Mechanisms in Mobile Communication Systems. *Tatra Mountains Mathematical Publications,* 25, 2002, pp. 101–117.