Differential Cryptanalysis of Feal and N-Hash

Eli Biham *Adi Shamir*

The Weizmann Institute of Science
Department of Applied Mathematics and Computer Science
Rehovot 76100, Israel

Abstract

In [1,2] we introduced the notion of differential cryptanalysis and described its application to DES[8] and several of its variants. In this paper we show the applicability of differential cryptanalysis to the Feal family of encryption algorithms and to the N-Hash hash function.

1 Introduction

Feal is a family of encryption algorithms, which are designed to have simple and efficient software implementations on 8-bit microprocessors. The original member of this family, called Feal-4[10], had four rounds. This version was broken by Den Boer[3] using a chosen plaintext attack with 100 to 10000 ciphertexts.

The designers of Feal reacted by creating a second version, called Feal-8[9,7]. This version used the same F function as Feal-4, but increased the number of rounds to eight.

Feal-8 was broken by the chosen plaintext differential cryptanalytic attack described in this paper. As a result, two new versions were added to the family: Feal-N[4] with any ev n number N of rounds, and Feal-NX[5] with an extended 128-bit key. In addition, The designers proposed a more complex eight-round version called N-Hash[6] as a cryptographically strong hash function which maps arbitrarily long inputs into 128-bit values.

The main results reported in this paper are as follows: Feal-8 is breakable under a chosen plaintext attack with 2000 ciphertexts. Feal-N can be broken faster

D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp. 1-16, 1991.

than via exhaustive search for any $N \leq 31$ rounds, and Feal-NX is just as easy to break as Feal-N for any value of N. The chosen plaintext differential cryptanalytic attacks can be transformed into known plaintext attacks, and can be applied even in the CBC mode of operation, provided we have sufficiently many known plaintext/ciphertext pairs (about 2^{38} in the case of Feal-8). Variants of N-Hash with up to 12 rounds can be broken faster than via the birthday paradox, but for technical reasons we can apply this attack only when the number of rounds is divisible by three. In the full paper we also show that Feal-4 is trivially breakable with eight chosen plaintexts or via a non-differential attack with about 100000 known plaintexts.

2 Differential Cryptanalysis of Feal

The notion of differential cryptanalysis and its application to DES-like cryptosystems are described in [1,2]. Due to space limitations, we can only give a high level description of such an attack in this extended abstract.

The basic tool of differential cryptanalytic attacks is a pair of ciphertexts whose corresponding plaintexts have a particular difference. The method analyses many pairs with the same difference and locates the most probable key. For Feal the difference is chosen as a particular XORed value of the two plaintexts.

The following notation is used in this paper:

n_x: An hexadecimal number is denoted by a subscript x (i.e., $10_x = 16$).

X^*, X': At any intermediate point during the encryption of pairs of messages, X and X^* are the corresponding intermediate values of the two executions of the algorithm, and X' is defined to be $X' = X \oplus X^*$.

P: The plaintext. P^* is the other plaintext in the pair and $P' = P \oplus P^*$ is the plaintexts XOR.

T: The ciphertexts of the corresponding plaintexts P, P^* are denoted by T and T^*. $T' = T \oplus T^*$ is the ciphertexts XOR.

(l, r): The left and right halves of the ciphertext T are denoted by l and r respectively.

$a, \ldots, h$: The 32 bit inputs of the F function in the various rounds. See figure 1.

$A, \ldots, H$: The 32 bit outputs of the F function in the various rounds. See figure 1.

ROL2(X): Rotation of the byte X by two bits to the left.

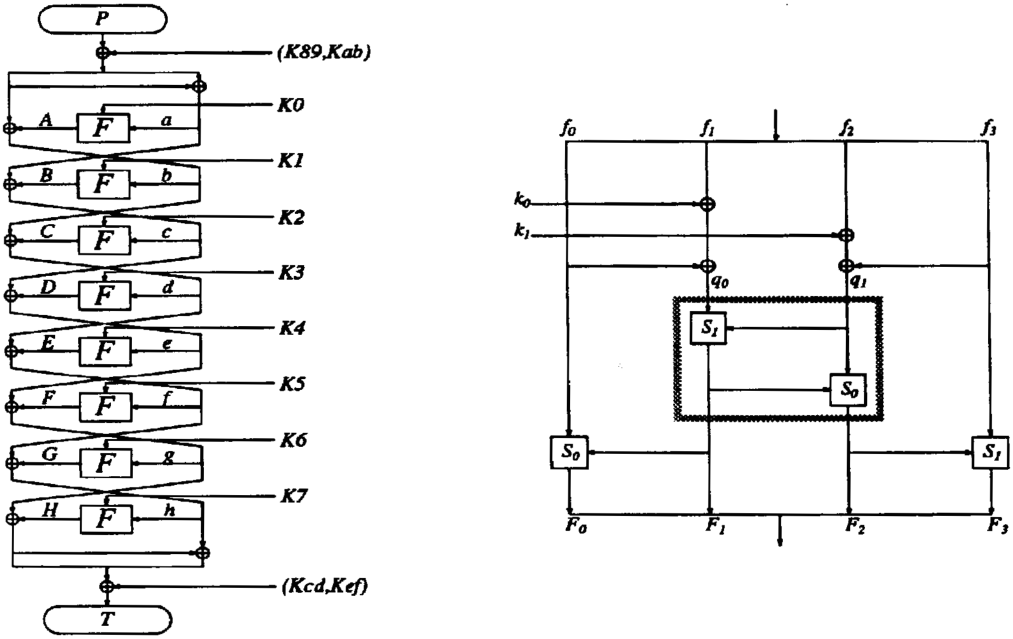


Figure 1: The structure and the F function of Feal-8

$S_i(x,y)$: The Feal S boxes: $S_i(x,y) = \mathrm{ROL2}(x+y+i \pmod{256})$.

q_i^x: The value inside the F function, with input x (one of h, g, …). Used as q_i for anonymous input and as q^x for the 16-bit value. See figure 1.

X_i: The i^{th} byte of X (for 16, 32 or 64-bit X).

$X_{i,j}$: The j^{th} bit of X_i (where 0 is the least significant bit).

$\#X$: The number of bits set to 1 in the lower seven bits of byte X.

$|$: The logical-or operator.

The structure of Feal (see figure 1) is similar to the structure of DES with a new F function and modified initial and final transformations. The F function of Feal contains two new operations: byte rotation which is XOR-linear and byte addition which is not XOR-linear. The byte addition operation is the only non linear operation in Feal and therefore the strength of Feal crucially depends on its non-linearity. At the beginning and at the end of the encryption process the right half of the data is XORed with the left half of the data and the whole data is XORed with additional subkeys, rather than permuted as in DES. Due to their linearity, these XORs pose only minor difficulty to our attacks.

The addition operations in the S boxes are not XOR-linear. However, there is still a statistical relationship between the input XORs of pairs and their output XORs. A table which shows the distribution of the input XORs and the output XORs of an S box is called the *pairs XOR distribution table* of the S box. Such a table has an entry for each combination of input XOR and output XOR, and the value of an entry is the number of possible pairs with the corresponding input XOR and output XOR. Usually several output XORs are possible for each input XOR. A special case arises when the input XOR is zero, in which case the output XOR must be zero as well. We say that X *may cause* Y (denoted by $X \rightarrow Y$) if there is a pair in which the input XOR is X and the output XOR is Y. We say that X *may cause* Y *with probability* p if for a fraction p of the pairs with input XOR X, the output XOR is Y.

Since each S box has 16 input bits and only eight output bits it is not recommended to use the pairs XOR distribution tables directly. Instead, in the first stage of the analysis we use the joint distribution table of the two middle S boxes in the F function (inside the gray rectangle in figure 1). This combination has 16 input bits and 16 output bits, and the table has many interesting entries. For example, there are two entries with probability 1 which are $00\ 00_x \rightarrow 00\ 00_x$ and $80\ 80_x \rightarrow 00\ 02_x$. About 98% of the entries are impossible (contain value 0). The average value of all the entries is 1, but the average value of the possible entries is about 50. In the full paper we describe how we can easily decide if $X \rightarrow Y$ or not for given XOR values X and Y without consulting the table.

The S boxes also have the following properties with respect to pairs: Let $Z = S_i(X,Y)$. If $X' = 80_x$ and $Y' = 80_x$ then $Z' = 00_x$ always. If $X' = 80_x$ and $Y' = 00_x$ then $Z' = 02_x$ always. For any input XORs X' and Y' of the S boxes the resultant output XOR $Z' = \mathrm{ROL2}(X' \oplus Y')$ is obtained with probability about $\frac{1}{2^{\#(X'|Y')}}$. This happens because each bit which is different in the pairs (X and X^*, or Y and Y^*) gives rise to a different carry with probability close to $\frac{1}{2}$. If all the carries happen at the same bits in the pair then the equation is satisfied.

The final XOR of the subkeys with the ciphertexts is significant when we look for the subkeys. The input of the F function in the last round is a function of the ciphertext XORed with an additional subkey of the final transformation rather than just a function of the ciphertext (as in DES). Therefore, the counting scheme finds a XOR combination of the subkey of the last round and the additional subkey, rather than the subkey of the last round itself.

Definition 1 *The actual XOR combinations of subkeys which are found by the attack are called* **actual subkeys**. *The actual subkey of round* $i+1$ *is denoted by* AKi. *The 16-bit XOR combinations* $(AKi_0 \oplus AKi_1, AKi_2 \oplus AKi_3)$ *are called* **16-bit actual subkeys**. *The actual subkey of the last round of a cryptosystem is called the* **last actual subkey**.

Example 1 *The actual subkeys of Feal-8 in the even rounds $i+1$ are the 32 bit values*

$$AKi = Kcd \oplus Kef \oplus \mathrm{am}(Ki)$$

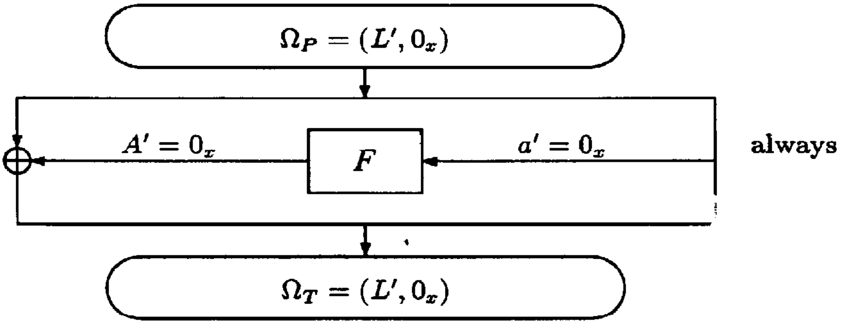
where $\mathrm{am}(Ki)$ *is the 32-bit value* $(0, Ki_0, Ki_1, 0)$. *The actual subkeys in the odd rounds are the 32 bit values*

$$AKi = Kcd \oplus \mathrm{am}(Ki).$$

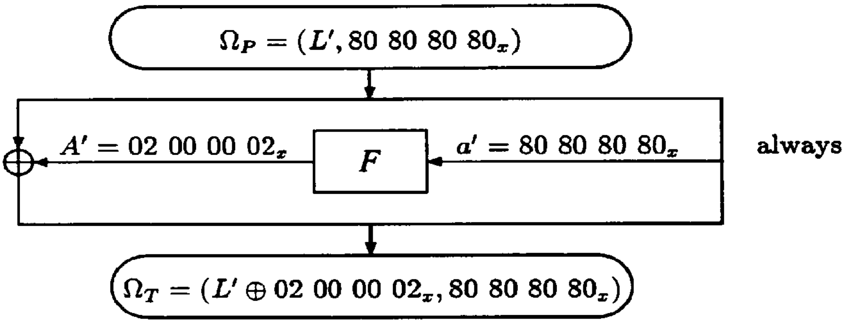
A tool which pushes the knowledge of the XORs of pairs as many rounds as possible is called a ***characteristic***. An ***n-round characteristic*** Ω starts with an input XOR value Ω_P and assigns a probability in which the data XOR after n rounds becomes Ω_T. Two characteristics Ω^1 and Ω^2 can be concatenated to form a longer characteristic whenever Ω^1_T equals the swapped value of the two halves of Ω^2_P, and the probability of Ω is the product of the probabilities of Ω^1 and Ω^2. A pair whose intermediate XORs equal the values specified by a characteristic is called a ***right pair*** with respect to the characteristic. Any other pair is called a ***wrong pair*** with respect to the characteristic. Note that in Feal, the plaintext XOR P' is different from the input XOR of the characteristic Ω_P due to the initial and final transformations.

Given a sufficiently long characteristic and a right pair we can calculate the output XOR of F function in the last round. The inputs themselves of this F function are known from the ciphertexts up to a XOR with subkeys. For any possible value of the last actual subkey, we count the number of possible pairs for which the output XOR is as expected. Every right pair suggests the right value of the actual subkey. The wrong pairs suggest random values. Since the right pairs occur with the characteristic's probability, the right value of the actual subkey should be counted more often than any other value. Therefore, it can be identified.

The simplest example of a one-round characteristic with probability 1 is:



There are three other one-round characteristics with probability 1. A typical one is:


$\Omega_P = (L', 80\ 80\ 80\ 80_x)$
$A' = 02\ 00\ 00\ 02_x$
F
$a' = 80\ 80\ 80\ 80_x$
always
$\Omega_T = (L' \oplus 02\ 00\ 00\ 02_x, 80\ 80\ 80\ 80_x)$

Three non trivial three-round characteristics with probability 1 also exist.

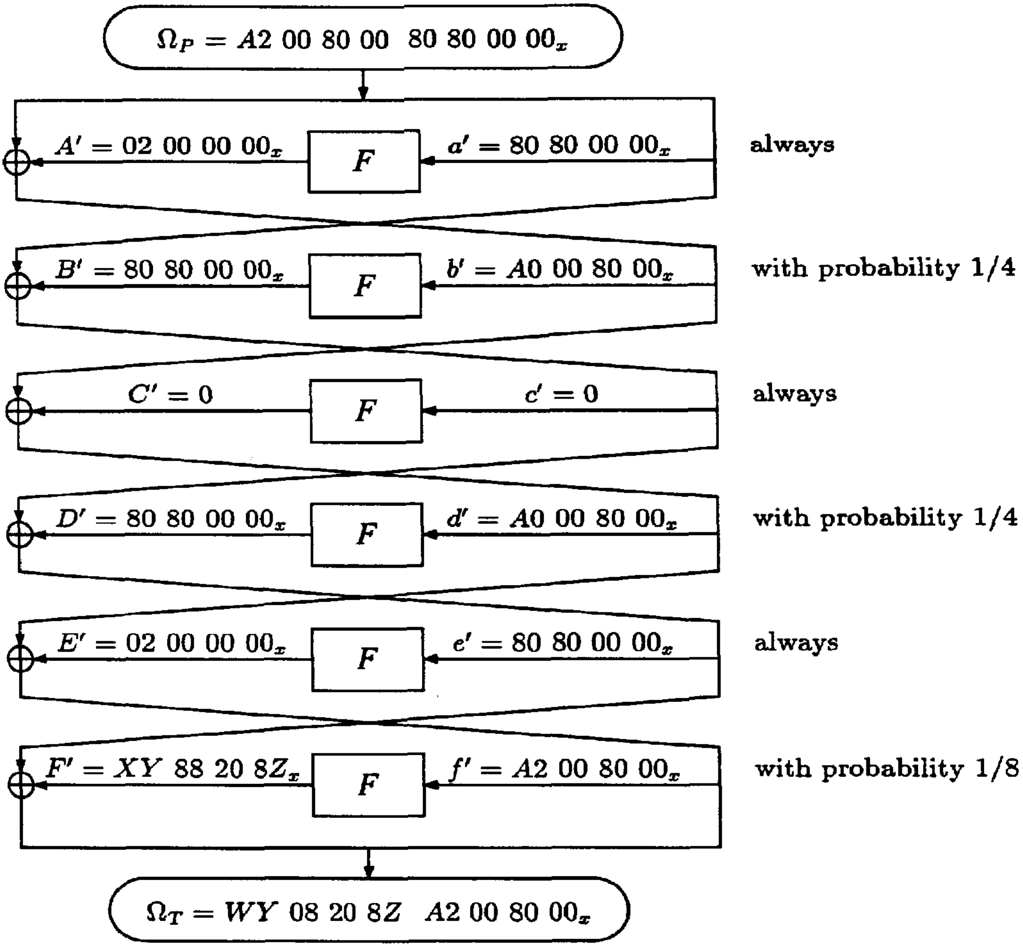
A five-round characteristic with probability $\frac{1}{16}$, a six-round characteristic with probability $\frac{1}{128}$ and an iterative characteristic with probability $\frac{1}{4}$ per round are described later in this extended abstract.

3 Cryptanalysis of Feal-8

In this section we describe a differential cryptanalytic attack on Feal-8 which uses about 1000 pairs of ciphertexts whose corresponding plaintexts are chosen at random satisfying

$$P' = A2\ 00\ 80\ 00\ 22\ 80\ 80\ 00_x.$$

The plaintext XOR is motivated by the following six round characteristic whose probability is 1/128:



where X, Y, Z and W are not fixed and can range (for different right pairs) over $X \in \{5,6,7,9,A,B,D,E,F\}$, $Y \in \{9,A,B\}$, $Z \in \{0,1,3\}$ and $W = X \oplus 8$.

In order to find the last actual subkey we do the following. Given the ciphertexts T and T^* of a right pair, we can deduce:

$$\begin{aligned} g' &= WY\ 08\ 20\ 8Z_x \\ h' &= l' \oplus r' \\ G' &= f' \oplus h' = A2\ 00\ 80\ 00_x \oplus l' \oplus r' \\ H' &= l' \oplus g' = l' \oplus WY\ 08\ 20\ 8Z_x. \end{aligned}$$

Therefore, all the bits of h' and G' and 24 bits of each of g' and H' are known.

The counting method is used to find the 16-bit last actual subkey. Filtering can be done by the knowledge of bits in the other two bytes of H' and in the seventh round. Assuming $g' \to G'$ we can reverse calculate the values of $g'_{i,0}$ from G' by

$$\begin{aligned} g'_{0,0} &= G'_{0,2} \oplus G'_{1,0} \\ g'_{3,0} &= G'_{3,2} \oplus G'_{2,0} \\ g'_{2,0} &= G'_{2,2} \oplus G'_{1,0} \oplus g'_{3,0} \\ g'_{1,0} &= G'_{1,2} \oplus g'_{0,0} \oplus g'_{2,0} \oplus g'_{3,0} \end{aligned}$$

and verify that the two known bits $g'_{1,0}$ and $g'_{2,0}$ from the characteristic are the same. About $\frac{3}{4}$ of the wrong pairs are discarded by this verification. We can also discard about $\frac{4}{5}$ of the other wrong pairs for which $g' \not\to G'$. Assuming $h' \to H'$ we can verify the four bits of $H'_{i,2}$ by

$$\begin{aligned} H'_{0,2} &= H'_{1,0} \oplus h'_{0,0} \\ H'_{1,2} &= h'_{0,0} \oplus h'_{1,0} \oplus h'_{2,0} \oplus h'_{3,0} \\ H'_{2,2} &= H'_{1,0} \oplus h'_{2,0} \oplus h'_{3,0} \\ H'_{3,2} &= H'_{2,0} \oplus h'_{3,0}. \end{aligned}$$

This verification discards about $\frac{15}{16}$ of the remaining wrong pairs.

All the right pairs must be verified correctly. Only $\frac{1}{4} \cdot \frac{1}{5} \cdot \frac{1}{16} = \frac{1}{320}$ of the wrong pairs should pass the three filters. Since the right pairs occur with the characteristic's probability of $\frac{1}{128}$, most of the remaining pairs are right pairs.

The counting scheme counts the number of pairs for which each value of the 16-bit last actual subkey is possible. Our calculations show that the right value is about 2^{15} times more likely to be counted than a random value. This ratio is so high that only eight right pairs are typically needed for the attack, and thus the total number of pairs we have to examine is about $8 \cdot 128 \approx 1000$. Note that we cannot distinguish between the right value of the actual subkey and the same value XORed with $80\,80_x$. Therefore, we find two possibilities for the 16-bit actual subkey.

Given the last 16-bit actual subkey it is possible to extend it to the full last actual subkey and then find the previous actual subkeys using similar approaches with shorter characteristics whose probabilities are much higher. Once the last actual subkey is found, we can partially decrypt the ciphertexts and proceed to find the previous five actual subkeys from which we can derive the following values:

$$\begin{gathered} K5 \oplus K7 \\ K4 \oplus K6 \\ K3 \oplus K5 \\ K2 \oplus K4 \\ K1 \oplus K3. \end{gathered}$$

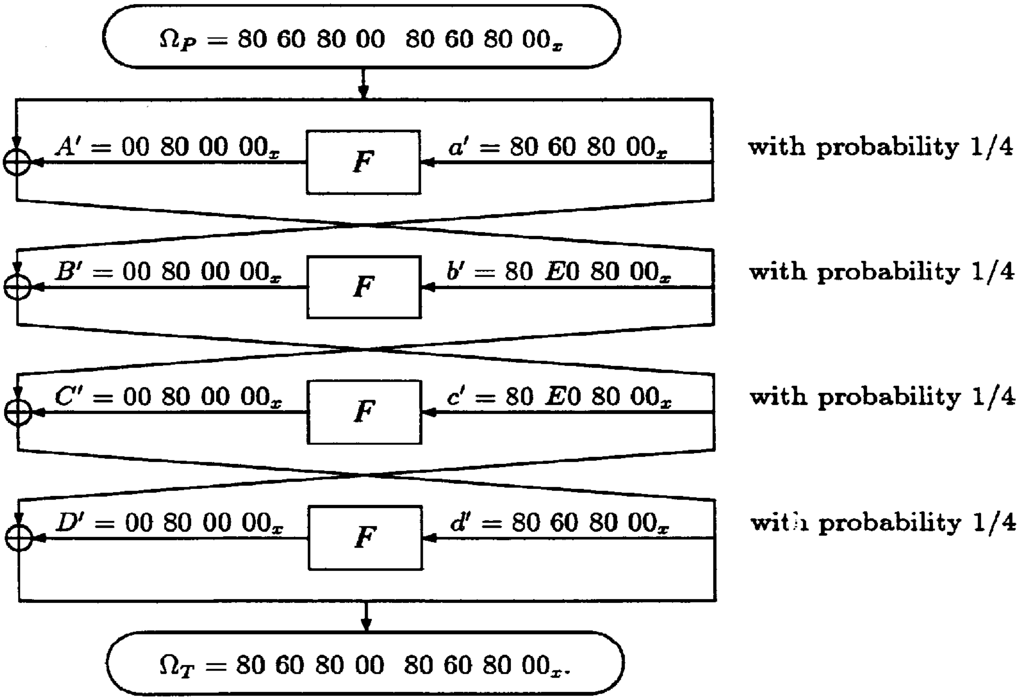
Using these values we can easily derive the value of the key itself by analyzing the structure of the key processing algorithm.

This attack was implemented on a COMPAQ personal computer. It finds the key in less than two minutes using 1000 pairs with more than 95% success rate. Using 2000 pairs it finds the key with almost 100% success rate. The program uses 280K bytes of memory.

4 Cryptanalysis of Feal-N and Feal-NX with $N \leq$ 31 rounds

Feal-N[4] was suggested as an N-round extension of Feal-8 after our attack on Feal-8 was announced. Feal-NX[5] is similar to Feal-N but uses a longer 128-bit key and a different key processing algorithm. Since our attack ignores the key processing and finds the actual subkeys, we can apply it to both Feal-N and Feal-NX with identical complexity and performance.

The attack on Feal with an arbitrary number of rounds is based on the following iterative characteristic:



The probability of each round of this characteristic is 1/4, and it can be concatenated to itself any number of times since the swapped value of the two halves of Ω_P equals Ω_T. Thus, an n-round characteristic with probability $\frac{1}{4^n} = 2^{-2n}$ can be obtained.

An attack based on a characteristic which is shorter by two rounds than the cryptosystem is called a ***2R-attack***. In this case, we know the ciphertext XOR T' and the input XOR of the last round (w.l.g. we employ the notation of an eight-round cryptosystem) h' by the ciphertext, and f' and g' by the characteristic. Thus, $G' = f' \oplus h'$ and $H' = g' \oplus l'$. Each pair is verified to have $g' \rightarrow G'$ and $h' \rightarrow H'$ and the resultant pairs are used in the process of counting the possibilities in order to find the last actual subkey. The counting is done in two steps. In the first step we find the 16-bit last actual subkey. In the second step we find the two other eight-bit values of the last actual subkey. Two of the bits of the last actual subkey are indistinguishable. Therefore, we must try the following steps in parallel for the four possibilities of these two bits. The verification of $g' \rightarrow G'$ leaves only 2^{-19} of the pairs (since for either $g' = 80\ 60\ 80\ 00_x$ or $g' = 80\ E0\ 80\ 00_x$ there are only about 2^{13} possible output XORs G'). The verification of $h' \rightarrow H'$ leaves 2^{-11} of the pairs (the fraction of the possible entries in the pairs XORs distribution table of the F function). Our calculations show that the right value of the last subkey is counted with a detectably higher probability than a random value up to $N \leq 28$ rounds, and thus we can break Feal-N with 2R-attacks for any $N \leq 28$ rounds, faster than via exhaustive search, as shown in table 1.

An attack based on a characteristic which is shorter by one round than the cryptosystem is called a ***1R-attack***. Using 1R-attacks (w.l.g. we employ the notation of an eight-round cryptosystem), we know T' and h' from the ciphertext and g' and h' from the characteristic. Also, $H' = g' \oplus l'$. We can verify that h' calculated by the ciphertext equals the h' of the characteristic, and that $h' \rightarrow H'$. The successfully filtered pairs are used in the process of counting the number of times each possible value of the last actual subkey is suggested, and finding the most popular value. Complicating factors are the small number of bits set in h' (which is a constant defined by the characteristic), and the fact that many values of H' suggest many common values of the last actual subkey. Our calculations show that the right value of the last subkey is counted with detectably higher probability than a random value up to $N \leq 31$ rounds. A summary of the 1R-attacks on Feal-N appears in table 1, and shows that the differential cryptanalysis is faster than exhaustive search up to $N \leq 31$.

Note that in both the 1R-attacks and the 2R-attacks we use eight-message octets with four characteristics (this is a special case in which an octet can have four characteristics since $\Omega_P^4 = \Omega_P^1 \oplus \Omega_P^2 \oplus \Omega_P^3$). These four characteristics are the four possible rotations of the given characteristic. Thus, each octet gives rise to 16 pairs which greatly reduces the required number of chosen plaintexts. In both kinds of attacks there are two indistinguishable bits at each of the last two subkeys. The

	2R-attack			1R-attack		
N	Prob	Pairs	Data	Prob	Pairs	Data
8	2^{-12}	2^{14}	2^{13}	2^{-14}	2^{17}	2^{16}
9	2^{-14}	2^{16}	2^{15}	2^{-16}	2^{19}	2^{18}
10	2^{-16}	2^{18}	2^{17}	2^{-18}	2^{21}	2^{20}
11	2^{-18}	2^{20}	2^{19}	2^{-20}	2^{23}	2^{22}
12	2^{-20}	2^{22}	2^{21}	2^{-22}	2^{25}	2^{24}
13	2^{-22}	2^{24}	2^{23}	2^{-24}	2^{27}	2^{26}
14	2^{-24}	2^{26}	2^{25}	2^{-26}	2^{29}	2^{28}
15	2^{-26}	2^{28}	2^{27}	2^{-28}	2^{31}	2^{30}
16	2^{-28}	2^{30}	2^{29}	2^{-30}	2^{33}	2^{32}
17	2^{-30}	2^{32}	2^{31}	2^{-32}	2^{35}	2^{34}
18	2^{-32}	2^{34}	2^{33}	2^{-34}	2^{37}	2^{36}
19	2^{-34}	2^{36}	2^{35}	2^{-36}	2^{39}	2^{38}
20	2^{-36}	2^{38}	2^{37}	2^{-38}	2^{41}	2^{40}
21	2^{-38}	2^{40}	2^{39}	2^{-40}	2^{43}	2^{42}
22	2^{-40}	2^{42}	2^{41}	2^{-42}	2^{45}	2^{44}
23	2^{-42}	2^{44}	2^{43}	2^{-44}	2^{47}	2^{46}
24	2^{-44}	2^{46}	2^{45}	2^{-46}	2^{49}	2^{48}
25	2^{-46}	2^{49}	2^{48}	2^{-48}	2^{51}	2^{50}
26	2^{-48}	2^{52}	2^{51}	2^{-50}	2^{53}	2^{52}
27	2^{-50}	2^{55}	2^{54}	2^{-52}	2^{55}	2^{54}
28	2^{-52}	2^{58}	2^{57}	2^{-54}	2^{57}	2^{56}
29	2^{-54}			2^{-56}	2^{59}	2^{58}
30	2^{-56}			2^{-58}	2^{61}	2^{60}
31	2^{-58}			2^{-60}	2^{64}	$\underline{2^{63}}$
32	2^{-60}			2^{-62}	2^{67}	$\underline{2^{66}}$

Table 1: Attacks on Feal-N

attacking program should try all the 16 possible values of these bits when analyzing the earlier subkeys.

5 Known Plaintext Differential Cryptanalytic Attacks

Differential cryptanalytic attacks are chosen plaintext attacks in which the plaintext pairs can be chosen at random as long as they satisfy the plaintext XOR condition. Unlike other chosen plaintext attacks, differential cryptanalytic attacks can be easily converted to known plaintext attacks by the following observation.

Assume that the chosen plaintext differential cryptanalytic attack needs m pairs, and that we are given $2^{32} \cdot \sqrt{2m}$ random known plaintexts and their corresponding ciphertexts. Consider all the $\frac{(2^{32} \cdot \sqrt{2m})^2}{2} = 2^{64} \cdot m$ possible pairs of plaintexts they can form. Each pair has a plaintext XOR which can be easily calculated. Since the block size is 64 bits, there are only 2^{64} possible plaintext XOR values, and thus there are about $\frac{2^{64} \cdot m}{2^{64}} = m$ pairs creating each plaintext XOR value. In particular, with high probability there are about m pairs with each one of the several plaintext XOR values needed for differential cryptanalysis.

The known plaintext attack is not limited to the electronic code book (ECB) mode of operation. In particular, the cipher block chaining (CBC) mode can also be broken by this attack since when the plaintexts and the ciphertexts are known, it is easy to calculate the real input of the encryption function.

Table 2 summarizes the resultant known plaintext differential cryptanalytic attacks on Feal and DES. For each of the listed cryptosystems with the listed number of rounds, the table describes the number of pairs of each characteristic and the total number of random plaintexts needed for the chosen plaintext attack and for the known plaintext attack. These results hold even for the variants with independent subkeys.

6 Cryptanalysis of N-Hash

N-Hash[6] is a cryptographically strong hash function which hasnes messages of arbitrary length to 128-bit values. The messages are divided into 128-bit blocks, and each block is mixed with the hashed value computed so far by a randomizing function g. The new hashed value is the XOR of the output of the g-function with the block itself and with the old hashed value. The g-function contains eight randomizing rounds, and each one of them calls the F function (similar to the one

Cryptosystem	Number of pairs of one char	Number of chosen plaintexts	Number of known plaintexts
Feal-4	4	8	$2^{33.5}$
Feal-8	1000	2000	$2^{37.5}$
Feal-16	2^{28}	2^{29}	$2^{46.5}$
Feal-24	2^{44}	2^{45}	$2^{54.5}$
Feal-30	2^{59}	2^{60}	2^{62}
Feal-31	2^{62}	2^{63}	$2^{63.5}$
DES-6	120	240	2^{36}
DES-8	25000	50000	2^{40}
DES-13	2^{43}	2^{44}	2^{54}
DES-14	2^{50}	2^{51}	$2^{57.5}$
DES-15	2^{51}	2^{52}	2^{58}
DES-16	2^{57}	2^{61}	2^{61}

Table 2: Known plaintext attacks on Feal and DES

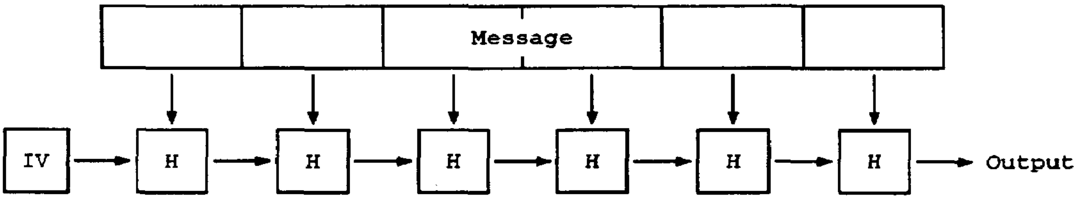


Figure 2: Outline of N-Hash

of Feal) four times. A graphic description of N-Hash is given in figures 2, 3, and 4.

Breaking a cryptographically strong hash function means finding two different messages which hash to the same value. In particular, we break N-Hash by finding two different 128-bit messages which are hashed to the same 128-bit value. Since the output of the g-function is XORed with its input in order to form the hashed value, it suffices to find a right pair for a characteristic of the g-function in which $\Omega_P = \Omega_T$. After XORing the input with the output of the g-function, the hashed value XOR becomes zero and thus the two messages have the same hashed value.

The following characteristic is a three round iterative characteristic with probability 2^{-16} (N-Hash does not swap the two halves after each round since the swap operation is part of the round itself. Therefore, the concatenation of the characteristic Ω^1 with the characteristic Ω^2 is possible whenever $\Omega_T^1 = \Omega_P^2$ without swapping). In the description of this characteristic we refer to the value 80 60 80 00_x as ψ and

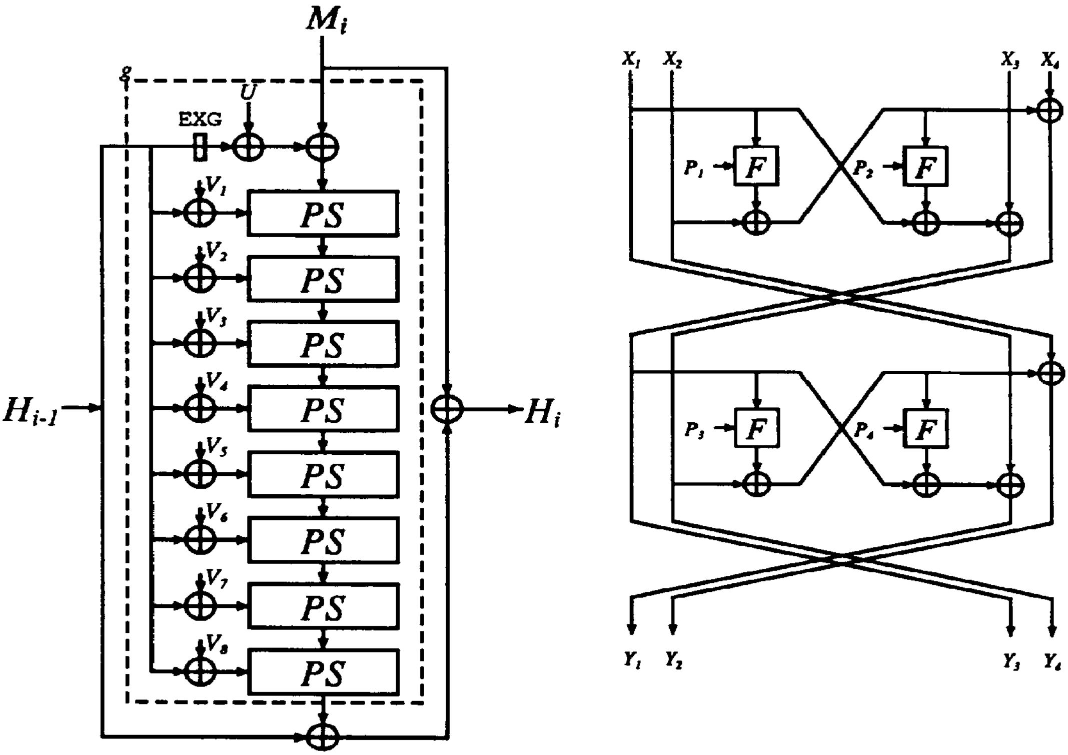


Figure 3: The function H and one round (PS) of N-Hash

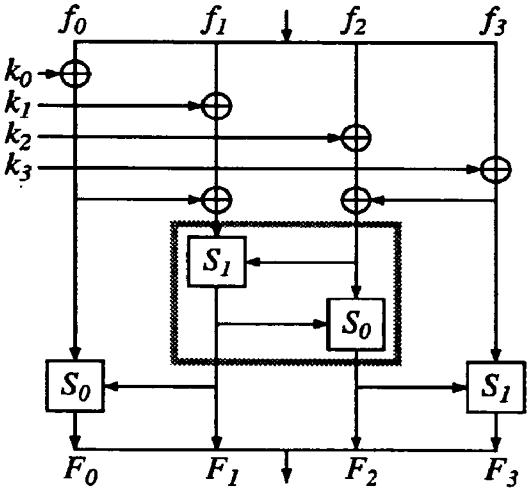


Figure 4: The F function of N-Hash

Number of Rounds	Complexity
3	2^{8}
6	2^{24}
9	2^{40}
12	2^{56}
15	2^{72}

Table 3: Results of the attack on N-Hash

to the value 80 E0 80 00_x as φ. Note that both $\psi \rightarrow (\psi \oplus \varphi)$ and $\varphi \rightarrow (\psi \oplus \varphi)$ with probability $\frac{1}{4}$ by the F function. The behavior of the XORs in the F function in this characteristic is similar to their behavior in the iterative characteristic of Feal. The characteristic itself is based on the input XOR:

$$\Omega_P = (\psi, \psi, 0, 0).$$

With probability $\frac{1}{256}$ the data XOR after the first round is

$$(0, 0, \varphi, \varphi).$$

With probability $\frac{1}{256}$ the data XOR after the second round is

$$(\psi, \psi, \varphi, \varphi).$$

The data after the third round is always

$$\Omega_T = \Omega_P = (\psi, \psi, 0, 0).$$

Therefore, the probability of the characteristic is 2^{-16}.

A pair of messages whose XOR equals Ω_P has probability $(2^{-16})^2 = 2^{-32}$ to have Ω_T as its output XOR after the sixth round of the g-function, and thus to have the same hashed value after their inputs and outputs are XORed by the six-round variant of N-Hash. Instead of trying about 2^{32} random pairs of messages we can choose only pairs from a smaller set in which the characteristic is guaranteed to be satisfied in the four F functions in the first round. The probability in this set is increased by a factor of 256, and thus only about 2^{24} such pairs have to be tested in order to find a pair of messages which hash to the same value.

This specific attack works only for variants of N-Hash whose number of rounds is divisible by three. Table 3 describes the results of this attack. We can see from the table that this attack is faster than the birthday attack (whose complexity is 2^{64}) for variants of N-Hash with up to 12 rounds.

The attack on N-Hash with six rounds was implemented on a personal computer and the following pairs of messages were found within about two hours:

- - CAECE595 127ABF3C 1ADE09C8 1F9AD8C2
  - 4A8C6595 921A3F3C 1ADE09C8 1F9AD8C2
  - Common hash value: 12B931A6 399776B7 640B9289 36C2EF1D
- - 5878BE49 F2962D67 30661E17 0C38F35E
  - D8183E49 72F6AD67 30661E17 0C38F35E
  - Common hash value: 29B0FE97 3D179E0E 5B147598 137D28CF.

References

[1] Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems (extended abstract)*, Advances in cryptology, proceedings of CRYPTO 90, 1990.

[2] Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, accepted by the Journal of Cryptology, 1990.

[3] Bert Den Boer, *Cryptanalysis of F.E.A.L.*, Advances in cryptology, proceedings of EUROCRYPT 88, 1988.

[4] Shoji Miyaguchi, *Feal-N specifications.*

[5] Shoji Miyaguchi, News on Feal cipher, talk at the RUMP session at CRYPTO 90, 1990.

[6] S. Miyaguchi, K. Ohta, M. Iwata, *128-bit hash function (N-Hash)*, proceedings of SECURICOM 90, pp. 123–137, March 1990.

[7] Shoji Miyaguchi, Akira Shiraishi, Akihiro Shimizu, *Fast Data Encryption Algorithm Feal-8*, Review of electrical communications laboratories, Vol. 36, No. 4, 1988.

[8] National Bureau of Standards, *Data Encryption Standard*, U.S. Department of Commerce, FIPS pub. 46, January 1977.

[9] Akihiro Shimizu, Shoji Miyaguchi, *Fast Data Encryption Algorithm Feal*, Advances in cryptology, proceedings of EUROCRYPT 87, pp. 267, 1987.

[10] Akihiro Shimizu, Shoji Miyaguchi, *Fast Data Encryption Algorithm Feal*, Abstracts of EUROCRYPT 87, Amsterdam, April 1987.