

Differential Attack on Message Authentication Codes

Kazuo Ohta¹ and Mitsuru Matsui²

¹ NTT Network Information Systems Laboratories
Nippon Telegraph and Telephone Corporation
1-2356, Take, Yokosuka-shi, Kanagawa-ken, 238-03 Japan

² Computer & Information Systems Laboratory
Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura-shi, Kanagawa-ken, 247 Japan

Abstract. We discuss the security of Message Authentication Code (MAC) schemes from the viewpoint of differential attack, and propose an attack that is effective against DES-MAC and FEAL-MAC. The attack derives the secret authentication key in the chosen plaintext scenario. For example, DES(8-round)-MAC can be broken with 2^{34} pairs of plaintext, while FEAL8-MAC can be broken with 2^{22} pairs. The proposed attack is applicable to any MAC scheme, even if the 32-bits are randomly selected from among the 64-bits of ciphertext generated by a cryptosystem vulnerable to differential attack in the chosen plaintext scenario.

1 Introduction

Authentication, which certifies data integrity and data origin, is becoming an important technique because the transfer of valuable information needed for electronic funds transfer, business contracts, etc. must be made across computer networks. Data integrity ensures that the data has not been modified or destroyed. Data origin authentication is the verification that the source of data received is as claimed.

There are two frameworks of authentication: digital signature schemes using public key cryptosystems and message authentication code (MAC) schemes based on secret key cryptosystems.

MAC schemes have been standardized and are being discussed by ISO/TC68/SC2 [ISO8731/2] for banking services. MAC is produced by the sender using a secret authentication key, which is known only to the sender and receiver of the authenticated message, and appended to the original message. Upon reception, the receiver uses the authentication key to check whether the received message-MAC pair is valid. The MAC of message m is the left half 32-bits of the last ciphertext block of m in CBC-mode. The Data Encryption Standard(DES) [FIPS77, ISO8731/1] and the Fast Data Encryption Algorithm(FEAL) [MSS88, MKOM90] cryptosystems are used to generate the ciphertext. Hereafter, we denote the MAC scheme based on DES by DES-MAC, and the MAC scheme based on FEAL by FEAL-MAC. We describe their specific

iteration number versions as DES(N-round)-MAC and FEAL N-MAC where N is the iteration number.

It is already known that differential attack is effective against DES [BS90, BS92], FEAL [BS91-1], and other iterated cryptosystems [BS91-2]. Differential attack requires many ciphertext pairs corresponding to plaintext pairs where the plaintext pairs are different in a significant way. Thus the differential attack is a type of chosen plaintext attack, although the value of plaintexts are not used in the attack procedure directly. Differential attack can derive the encryption/decryption key with less computational time than required by exhaustive attack using an appropriate number of ciphertext pairs in the chosen plaintext scenario. DES(8-round) can be broken with 2^{15} plaintext pairs, and FEAL8 can be broken with 2^{10} plaintext pairs. Some known plaintext attacks on these iterated cryptosystems have also been proposed [CG91, K91, MY92, M93].

In this paper, we analyze the security of MAC schemes from the viewpoint of differential attack. Since the left half 32-bits of ciphertext in CBC-mode is used as MAC, the following question is important to establishing the security of MAC schemes: Is differential attack effective against the MAC schemes? That is, how many plaintext pairs with the left-half (partial information) of their ciphertexts are necessary to derive the secret authentication key? We introduced an attack that is effective against both DES-MAC and FEAL-MAC, and estimate the number of appropriate plaintext pairs needed for the attack to derive the secret authentication key.

It is interesting that our procedure is also effective against any MAC scheme, where the 32-bits are randomly selected among the 64-bits of ciphertext generated by a cryptosystem vulnerable to differential attack in the chosen plaintext scenario. We will also discuss the influence of the bit length of ciphertext available as the MAC.

2 Related Works

2.1 Differential Attack against Ciphertext Case

Biham and Shamir proposed a differential attack against various iterated cryptosystems [BS90, BS91-1, BS91-2, BS92]. Each iteration is a function usually based on S boxes, bit permutations, arithmetic operations, and exclusive-or operations (denoted by XOR). The S boxes are known to be nonlinear. Since they are usually the only part of the cryptosystem that is not linear, the security of the cryptosystem depends on which type of S box is selected.

Differential attack depends on the following fact: Even though we cannot determine the XOR value of the S-box output from its input XOR value, some specific input XOR values yield specific output XOR value with high probability.

To find the subkeys entering each iteration function, differential attack proceeds as follows (see p.16 of [BS90]):

Step 1: Choose an appropriate plaintext XOR.

- Step 2:** Create an appropriate number of plaintext pairs with the plaintext XOR chosen at **Step 1**, encrypt them and keep only the resultant ciphertext pairs.
- Step 3:** For each pair, derive the expected output XOR of as many S boxes in the last round as possible from the plaintext XOR and the ciphertext pair.
- Step 4:** For each possible subkey value, count the number of pairs that result with the expected output XOR using this subkey value in the last round, and choose the value that is counted most often as the subkey candidate.

Note that the input pair of the last round is known, since it appears here as part of the ciphertext pair.

The pushing mechanism, with which the knowledge of the XORs of the plaintext pairs is passed through as many rounds as possible without making them zero, is termed the *statistical characteristic* of the cryptosystem in [BS90]. A pair whose intermediate XORs equal the values specified by the characteristic is called a *right pair* with respect to the characteristic. Any other pair is called a *wrong pair* with respect to the characteristic.

[BS90] showed many characteristics for several variants of DES with different round number. The two characteristics listed below will be referred to in a later section. These characteristics yield the known highest probability for the round number indicated.

$$\begin{aligned} \Omega_P^1 &= 405C0000 \ 04000000, & \Omega_T^2 &= 405C0000 \ 04000000 \\ &\text{with probability } 1/10, 486 \text{ for } 5 \text{ rounds} \\ \Omega_P^2 &= 00000000 \ 19600000, & \Omega_T^4 &= 19600000 \ 00000000 \\ &\text{with probability } (1/234)^N \approx (2^{-8})^N \text{ for } 2N \text{ rounds} \end{aligned}$$

[BS91-1] also pointed out that the following characteristics permit FEAL to be cryptanalyzed within various numbers of rounds.

$$\begin{aligned} \Sigma_P^1 &= A2008000 \ 80800000, & \Sigma_T^2 &= A2008000 \ 80800000 \\ &\text{with probability } 1/16 \text{ for } 5 \text{ rounds} \\ \Sigma_P^2 &= 80608000 \ 80608000, & \Sigma_T^3 &= 80608000 \ 80608000 \\ &\text{with probability } (1/4)^{4N} \text{ for } 4N \text{ rounds} \end{aligned}$$

Note that since differential attack requires many ciphertext pairs corresponding to plaintext pairs with particular differences, Ω_P and Σ_P , the differential attack is a type of chosen plaintext attack, although the value of plaintexts are not used in the attack procedure directly.

There are four possible types of attack, 3R-attack, 2R-attack, 1R-attack and 0R-attack, depending on the number of additional rounds in the cryptosystem that are not covered by the characteristic itself. A 3R-attack is advisable over a 2R-attack, and both are advisable over a 1R-attack, since characteristic that has higher probability requires fewer ciphertext pairs for the attack. For example, the differential attack on DES reduced to eight rounds, DES(8 round), in [BS90] uses a five-round characteristic, Ω_P^1 , with 3R-attack.

To find the right key with a counting scheme in **Step 4**, we need a high probability characteristic and enough ciphertext pairs to guarantee the existence of several right pairs. Usually we relate the number of pairs needed by a counting scheme to the number of right pairs needed. The number of right pairs needed is mainly a function of the ratio between the number of right pairs and the average count in the counting scheme, denoted by S/N . In the DES case, $S/N = \frac{2^k \cdot p}{\alpha \cdot \beta}$ holds, where k -bits of subkey are counted at **Step 4**, α is the average count per counted pair, β is the ratio of the counted to all pairs, and p is the characteristic's probability. When S/N is high enough, only a few right pairs are needed to uniquely identify the right value of the subkey bits.

Biham and Shamir found the suitable characteristic for each round number, for example, (Ω_P^1, Ω_T^1) for DES(8-round), (Ω_P^2, Ω_T^2) for DES(arbitrary round), (Σ_P^1, Σ_T^1) for FEAL8, and (Σ_P^2, Σ_T^2) for FEAL(arbitrary round), and peeled off the subkey from *the last round* using 3R-attack.

2.2 Attacks against Authentication Schemes

There are two frameworks for authentication, digital signature schemes using public key cryptosystems and message authentication code (MAC) schemes based on secret key cryptosystems. In the former case, a mixed type digital signature scheme [DP80] is practical. The signer calculates signature $s = f(h(m))$, where m is a message, h is a public hash function and f is a secret signature function known to the signer, and sends both s and m to the receiver. The receiver checks whether $f^{-1}(s) = h(m)$ holds using h and the public validation function f^{-1} . Here, hash functions are used to compress long messages into short digests to attain high efficiency, and they are not required to be secret. In the latter case, the MAC is produced by the sender using a *secret* authentication key, which is known only to the sender and receiver, and appended to the original message. Upon reception, the receiver uses the authentication key to check whether the received message-MAC pair is valid. Note that while a receiver who doesn't know the secret key can not generate a signature value in a digital signature scheme, he can calculate (forge) the MAC of any message using the authentication key in the MAC scheme. Thus the MAC scheme is applicable only to the case where the sender and receiver trust each other.

There are two kinds of threats in authentication schemes:

- (1) forgery of a digital signature: a valid signature-message pair is found using previously used pairs,

- (2) determination of secret information: the secret key of the public key cryptosystem or the secret authentication key of the MAC scheme are revealed.

The collision free property of hash functions is discussed in [D87, ZMI90] as a form of Type (1) threat. If an attacker finds a pair of messages, m and m' , satisfying $h(m) = h(m')$, where h is a public hash function, then signature value, s , of $h(m)$ is as valid as that of $h(m')$. Thus, he can replace the true message, m , with the invalid message m' ; the value, s , remains valid. It is possible to find a pair of such messages using the birthday paradox strategy in computational time, $O(2^{\ell/2})$, when the bit length of hashed value is ℓ . Other attacks of Type (1) have been discussed that addressed the weak key and semi weak key properties of cryptosystems [MOI90], and differential properties [BS91-1].

Type (2) attack means that if the attack succeeds, the authentication system is totally broken, while a Type (1) attack is a kind of ad-hoc forgery. Deriving the secret key of a public key cryptosystem is breaking the cryptosystem itself. Until now, there has not been sufficient discussion of Type (2) attacks on MAC schemes. We will point out the first attack procedure of this type from the viewpoint of differential attack.

3 How to Attack MAC Schemes

3.1 What are the Problems

In the attacks employed in the ciphertext case, it is important for the attacker to use the full length of the ciphertext. On the other hand, only the left half 32-bits of ciphertext in CBC-mode is available as the MAC value. Thus, what happens in the MAC case, where the attacker can not use the full ciphertext, but only partial information of the ciphertext?

The following questions are interesting in the differential attack against MAC schemes to derive the secret authentication key: 1) how many plaintext pairs are needed together with the left-half (partial information) of their ciphertexts, 2) what is the influence of the location of ciphertext information available to an attack, and 3) what is the influence of the bit length of ciphertext information available as the MAC.

3.2 Outline of MAC Attack

Hereafter, we will distinguish the real plaintext (message), M , to be authenticated from P treated as the plaintext in the references in [BS90, BS91-1], and the real ciphertext, C , from the ciphertext, T , where $P = IP(M)$ and $C = IP^{-1}(T)$ in DES, where IP is the initial permutation, and $(P_H, P_L) = (M_H, M_H \oplus M_L)$ and $(C_H, C_L) = (T_H, T_H \oplus T_L)$ in FEAL, where P_H is the left half of P and P_L is the right half of P .

Since the proposed procedure is a chosen plaintext attack, we assume the case, where a plaintext (message) is a single block and the initial value of CBC-mode is public.

Since the right half of the ciphertext can not be used for an attack, we modify the differential attack against MAC schemes, which are based on 0R-attack or 1R-attack introduced by [BS90], to derive the secret subkeys, generated from the authentication key, in *the first round*, as follows:

Step 1: Choose the appropriate pair of real plaintext XOR, ω_P , and real ciphertext XOR, ω_T , that has high characteristic probability and many number of subkey bits whose occurrences can be checked at **Step 4**.

Step 2: Create an appropriate number of real plaintext pairs, (M, M^*) , where $M \oplus M^* = \omega_P$, calculate their MAC values, $\gamma = MAC(K, M)$ and $\gamma^* = MAC(K, M^*)$, where K is the authentication key, and keep only the real plaintext pairs (M, M^*) which satisfy

$$\gamma \oplus \gamma^* = \text{the left half of 32-bits of } \omega_T.$$

Step 3: Derive the expected output XOR, A' , of the first round function using the differential rules.

Step 4: For each possible subkey value of $K1$, where $K1$ is used by the first round function, count the number of pairs that result with the expected output XOR, A' , using this subkey value of $K1$ in the first round, and choose the value that is counted most often as the subkey candidate of $K1$.

Remark

In the DES(r -round)-MAC case, where $r \geq 8$, ω_P is transformed to Ω_P^2 by the initial permutation (IP), that is, $IP(\omega_P) = \Omega_P^2$, and ω_T is transformed to Ω_T^2 by IP , that is, $\omega_T = IP^{-1}(\Omega_T^2)$, in the 0R-attack. ω_T is transformed to a value related to Ω_T^2 by IP , that is, $\omega_T = IP^{-1}(\varphi \oplus (\Omega_T^2)_L, (\Omega_T^2)_H)$, where φ is the output XOR of the last round iteration function with an input, $(\Omega_T^2)_H$, in the 1R-attack. $A' = 00000000$.

In the FEAL N -MAC case, where $N \geq 8$, ω_P equals $(\sigma, 00000000)$, where $\sigma = 80608000$, ω_T equals $(\sigma, 00000000)$ or a value related to $(\sigma, 00000000)$, and $A' = 00800000$.

Note

If there are several candidates for $K1$, we can adopt two strategies to attain high efficiency in order to reduce the number of candidates.

- (1) Choose another $\tilde{\omega}_P$ at **Step 1**, apply the same procedure, and choose the common value between ω_P and $\tilde{\omega}_P$ cases as the subkey candidate of $K1$.
- (2) Derive the expected output XORs, B' , of the *second round iteration* function, and count the number of pairs that result with the expected output XOR, B' , using the subkey candidate value of $K1$ and newly selected subkey value of $K2$ in the second round. If the counted number is zero, the candidate for $K1$ is discarded.

We can apply the first strategy without the increase in the number of plaintext pairs using quartets which combine the two characteristics [BS90]. The second strategy applied to the case of FEAL-MAC will be described in detail in **Section 5**.

4 Discussion

4.1 DES-MAC

The procedure described is based on 0R-attack or 1R-attack, and can be considered a differential attack against a cryptosystem changing the roles of plaintext and ciphertext. So the discussion of [BS90] is applicable with slight modification. Note that MAC values, which are the partial information of ciphertext, are sufficient in the proposed attack, since the actual plaintext values are not necessary in the original differential attack.

The possibility of subkey value can be checked on some bits of the subkey in the first round entering the S boxes with nonzero input XORs. If we use ω_P corresponding to Ω_P^2 , three S boxes, $S1, S2$, and $S3$, yield 18 bits of subkey $K1$.

Since the input XOR is constant, we can not distinguish between several subkey values. However, the number of such values is small and each can be checked later in parallel by the next part of the algorithm (see p.40 line 9 of [BS90]).

Let's consider the case of DES, which is reduced to an even round, where 0R-attack is applicable. Note that the β component of S/N should be 2^{-32} , since the output XOR of the last round iteration is $\Omega_T = (\psi, 0)$, where $\psi = 19600000$, ω_T has specific 64 bits, and the left half 32-bits ω_T are determined definitely. Thus, DES(8-round)-MAC can be broken with 2^{34} pairs, since $S/N = \frac{2^{18} \times (2^{-4})^8}{4^3 \times 2^{-32}} = 2^{12}$. DES(10-round)-MAC can be broken with 2^{42} pairs, since $S/N = \frac{2^{18} \times (2^{-4})^{10}}{4^3 \times 2^{-32}} = 2^4$. DES(12-round)-MAC can not be broken, since $S/N = \frac{2^{18} \times (2^{-4})^{12}}{4^3 \times 2^{-32}} = 2^{-4}$.

Concerning the influence of the location of ciphertext information available as MAC, since the value of the β component of S/N is 2^{-32} constantly (in 0R-attack), the security of DES(even-round)-MAC does not depend on the bit location in the ciphertext used as MAC.

We utilize the following fact in the above discussion: Biham and Shamir observed experimentally that when S/N is about 1–2, about 20–40 occurrences of right pairs are sufficient. When the S/N is much higher, even three or four right pairs are usually enough. When the S/N is much smaller than 1, the identification of the right value of subkey bits requires an unreasonably large number of pairs. (see p.23 in [BS90])

Let's consider the case of DES, which is reduced to an odd round, where 1R-attack is applicable. Note that the β component of S/N should be $2^{-20} \sim 2^{-32}$, since the output XOR of the last round iteration is $\Omega_T = (\varphi, \psi)$, where φ contains 20 bits of zeros released by $S4, \dots, S8$ at the last iteration, ω_T has 12 free bits among 64 bits, and the left half of 32-bits of ω_T might contain them. With careful check on the bit location of MAC, since 7 bits among the 12 free bits are contained in the left half of 32-bits of ω_T , $\beta = 2^{-25}$ holds. Thus, DES(7-round)-MAC can be broken with 2^{26} pairs, since $S/N = \frac{2^{18} \times (2^{-4})^6}{4^3 \times \beta} = 2^{13}$. DES(9-round)-MAC can be broken with 2^{34} pairs, since $S/N = \frac{2^{18} \times (2^{-4})^8}{4^3 \times \beta} = 2^5$. DES(11-round)-MAC can not be broken, since $S/N = \frac{2^{18} \times (2^{-4})^{10}}{4^3 \times \beta} = \frac{1}{2^3}$.

It is also clear that our attack is effective to any DES(N -round)-MAC scheme, where $N \leq 10$; the 32-bits can be randomly selected from among the 64-bits of ciphertext. It is not effective against DES(N' -round)-MAC scheme, where $N' \geq 12$. Only the security of DES(11-round)-MAC depends on the bit location in the ciphertext used as MAC, since $S/N = \frac{2^{18} \times (2^{-4})^{10}}{4^3 \times \beta} = 2^{-8} \sim 2^4$.

Concerning the influence of the bit length of ciphertext information available as the MAC, the value of the β component of S/N is $2^{-\ell}$ in DES(even-round)-MAC if the bit length of MAC is ℓ . For example, DES(8-round)-MAC(24 bit) is breakable, since $S/N = \frac{2^{18} \times (2^{-4})^8}{4^3 \times 2^{-24}} = 2^4$, while DES(8-round)-MAC(18 bit) can not be broken, since $S/N = \frac{2^{18} \times (2^{-4})^8}{4^3 \times 2^{-18}} = 2^{-2}$. The similar discussion holds in DES(odd-round)-MAC. Note that when we use the small bits of MAC, it is easy to find a pair of collision messages, while it is difficult to derive the secret authentication key.

If we use $\tilde{\omega}_P$ corresponding to $\Omega_P^3 = 00196000\ 00000000$, three S boxes, S_3, S_4 , and S_5 , give 12 more of the bits in subkey K_1 (this is in addition to the 18 bits derived with Ω_P^2). This reduces the computational time of the attack procedure.

While 2^{34} plaintext pairs are necessary for differential attack against DES(10-round) in the ciphertext case, 2^{42} pairs are necessary in the MAC case, since 2R-attack and 3R-attack are not applicable to the MAC case. It is an open problem whether 2R-attack and 3R-attack are applicable to MAC schemes.

4.2 FEAL-MAC

The attack procedure is based on 0R-attack or 1R-attack, and can be considered as a differential attack against a cryptosystem changing the roles of plaintext and ciphertext. So the discussion of [BS91-1] is applicable with slight modification.

The possibility of subkey value can be checked using some subkey bits from the first round, where the input XOR, a' , equals $\sigma = 80608000$, and the output XOR, A' , equals 00800000.

Biham and Shamir make the following statement in [BS91-1]: "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'^3 (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 (Biham and Shamir's) calculations show that the right value of the last subkey is counted with detectably higher probability than a random value up to $N \leq 31$ round." (see p.10 line 26 of [BS91-1])

We can apply the above explanation to FEAL-MAC schemes by replacing the last actual subkey with the first actual subkey: h' with a' , and H' with A' .

In estimating the sufficient number of plaintext pairs, though Biham and Shamir imply that four right pairs are sufficient if $N \leq 24$ to derive the subset

³ They employ the notation of an eight round cryptosystem.

key using a *counting method* in the ciphertext attack (see p.11 of [BS91-1]), it is not clear how many pairs are sufficient to derive the subset key in the proposed MAC attack, where the value of A' described in the above is fixed. Our experimental results, described in the next section, confirm that at most 2^6 right pairs are sufficient to derive the subkey $K1$ with a *checking method*, by simply checking whether all pairs that pass the check of **Step 2** also pass **Step 4** using A' .

Since FEAL8 has the characteristic probability, 2^{-16} , we can find 2^6 right pairs from 2^{22} pairs of plaintext with high probability in **Step 2**. On the other hand, the probability that a wrong pair is found is 2^{-10} , since the 32 bits are used in **Step 2** to filter right pairs. Thus FEAL8-MAC can be broken with 2^{22} pairs of plaintext with overwhelming probability ($= 1 - \frac{1}{2^{10}}$).

Since FEAL12 has the characteristic probability, 2^{-24} , we can find 2^6 right pairs from 2^{30} pairs of plaintext with high probability in **Step 2**. On the other hand, the probability that a wrong pair is found is 2^{-2} in **Step 2**. Since we confirmed with an experiment that even if one wrong pair is contained among 64 right pairs, we can find a correct bits among the subkey with high probability (99 %), FEAL12-MAC can also be broken with 2^{30} pairs of plaintext with high probability.

On the other hand, since FEAL16 has the characteristic probability, 2^{-32} , it is difficult to find pairs containing 2^6 right pairs and a few wrong pairs from 2^{38} pairs of plaintext at **Step 2**. Since the selected pairs at **Step 2** contain many wrong pairs with high probability, it is necessary to use the counting method instead of the checking method to break FEAL16-MAC. It is an open problem to estimate the sufficient number of pairs for the counting method.

Note that these attacks are applicable to FEAL-MAC only in the chosen plaintext attack, though some known plaintext attacks are pointed out to ciphertext case [CG91, MY92, K91]. It is an open problem to break MAC schemes in the known plaintext attack.

5 Experimental Results

The purpose of this experiment is to estimate the sufficient number of right pairs to derive the subkey $K1$. We will describe experimental results on FEAL8-MAC using the attack technique *cut off the spread of carry bit*, developed in [MY92]. Here, we adopt the second strategy described in **Section 3**: check whether all pairs that pass the check of **Step 2** also pass **Step 4** using B' of the second round function in addition to the expected output XORs, A' .

For convenience, we use the modified F-function defined by [MY92] in our experiment.

5.1 Notation

We use the following notations in this section.

M : A plaintext input to the FEAL-MAC algorithm

A_M : Output of the first round function (32 bits) corresponding to M

B_M : Output of the second round function (32 bits) corresponding to M

b_M : Input of the second round function (32 bits) corresponding to M

$B[i]$: The i -th bit of B , where $0 \leq i \leq 31$

$K1$: Subkey value used by the first round function (32 bits)

$K2$: Subkey value used by the second round function (32 bits)

$K1[i \sim j]$: The $j - i + 1$ bits data consisting of the i -th, \dots , j -th bits of $K1$

$K1[i, j]$: The XORed value of the i -th and j -th bits of $K1$

5.2 Attack Procedure against FEAL-MAC Attack

We select $\omega_P = (\sigma, 00000000)$, where $\sigma = 80608000$, $\omega_T = (\sigma, 00000000)$, and $A' = B' = 00800000$. Hereafter, we will explain how to implement Step 4 of the MAC attack procedure to reduce the number of candidates for $K1$.

The following procedure selects the 12 bits for $K1$ that influence the 8 bits, 80, of A' .

- Step 1:** Select 12 bit data of $K1$ using the bits $K1[8 \sim 13]$, $K1[16 \sim 20]$, and $K1[21, 22]$, the remaining bits are determined arbitrarily.
- Step 2:** Calculate all right pairs, (M, M^*) , using the first round function with the selected $K1$, and check whether $A_M \oplus A_{M^*} = 00800000$ holds. If all checks are correct, then let the 12 bit data be a candidate of $K1$.

The following procedure can select more 12 bits of $K1$ and 1 bit of $K2$ in addition to the 12 bits selected above.

- Step 1:** Select 17 bits of $K1$ using $K1[0 \sim 3]$, $K1[14 \sim 15]$, $K1[23 \sim 28]$, $K2[12 \sim 13]$, $K2[20 \sim 22]$, the remaining bits are fixed arbitrarily.
- Step 2:** Calculate all right pairs using the first round function and the selected $K1$, and select pairs, (M, M^*) , satisfying

$$b[4] \oplus b[12] \oplus b[20] \oplus b[28] = K2[12] \oplus K2[20]. \quad (1)$$

- Step 3:** Calculate the pairs selected in the above step using the second round function and the selected $K2$, and check whether

$$B_M[16] \oplus B_{M^*}[16] = 0, B_M[17] \oplus B_{M^*}[17] = 0, B_M[23] \oplus B_{M^*}[23] = 1. \quad (2)$$

If all checks are correct for the pairs, then let the value of the 24 bit data be a candidate of $K1$.

Since we can ignore the influence of $K2[8 \sim 11]$ and $K2[16 \sim 19]$ considering equation (1) at **Step 2**, we don't have to guess these bits in the above procedure. This technique, *cut off the spread of carry bit*, was developed by [MY92] to reduce the computational time needed to get the subkey information. Equation (2) corresponds to the fact that $B_M \oplus B_M^*$ should be $B' = 00800000$. Note that since we don't know $B_M[18 \sim 22]$ here, only three bits are checked in equation (2).

After the calculation of these 24 bits of subkey $K1$, we can determine 30 bits by repeating the previous method without the above restrictions of equation (1). Though $K1[7]$ and $K1[31]$ remain undetermined with this procedure, they can be determined by an exhaustive search.

The above procedure was implemented and tested. We have confirmed that 2^6 right pairs are sufficient to derive the subkey $K1$ with the above procedure. We could decrease the number of pairs required by checking the XOR outputs of higher round functions, C' , D' and so on.

6 Conclusion and Remarks

We have proposed a modified differential attack which is effective against MAC schemes, where only the left half 32-bits of the ciphertext is available to the attacker. The attack derives the secret authentication key in the chosen plaintext scenario. The procedure is considered as a form of differential attack, 0R-attack or 1R-attack, against a cryptosystem where the roles of plaintext and ciphertext are reversed. We have also pointed out that our procedure is also effective against any MAC scheme even if the 32-bits are randomly selected from among the 64-bits of ciphertext generated by a cryptosystem vulnerable to differential attack in the chosen plaintext scenario.

More exactly, based on the discussion of [BS90], and with slight modification of the S/N ratio, it appears that DES(12-round)-MAC is secure, while DES(8-round)-MAC can be broken with 2^{34} pairs of plaintext in the chosen plaintext scenario. It is clear that our attack is effective against any DES(N -round)-MAC scheme, where $N \leq 10$; the 32-bits can be randomly selected from among the 64-bits of ciphertext.

Concerning the influence of the bit length of ciphertext information available as the MAC, it becomes clear that, for example, DES(8-round)-MAC(24 bit) is breakable, while DES(8-round)-MAC(18 bit) can not be broken.

Based on our experiment and the discussion of [BS91-1], it is clear that FEAL8-MAC can be broken with 2^{22} pairs of plaintext with overwhelming probability and FEAL12-MAC can be broken with 2^{30} pairs with high probability in the chosen plaintext scenario.

There are several open problems:

- (1) whether MAC is broken in the known plaintext attack,
- (2) whether 2R-attack and 3R-attack are effective against MAC schemes, and
- (3) how many pairs of plaintexts are sufficient to break FEAL16-MAC using the counting method.

Acknowledgment

A part of this research was conducted while the first author was visiting the MIT Laboratory for Computer Science. He would like to acknowledge the generous support provided by MIT/LCS. The second author would like to thank Atsuhiro Yamagishi for his kind suggestion and encouragement.

References

- [BS90] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *Journal of CRYPTOLOGY*, Vol. 4, Number 1, 1991 (The extended abstract appeared at CRYPTO'90)
- [BS91-1] E. Biham and A. Shamir, "Differential Cryptanalysis of Feal and N-Hash," *EUROCRYPT'91*
- [BS91-2] E. Biham and A. Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer," *CRYPTO'91*
- [BS92] E. Biham and A. Shamir, "Differential Cryptanalysis of the full 16-round DES," *CRYPTO'92*
- [CG91] A. Tardy-Corffdir, and H. Gilbert, "A known plaintext attack of FEAL-4 and FEAL-6," *CRYPTO'91*
- [D87] I. Damgård, "Collision free hash functions and public key signature schemes," *EUROCRYPT'87*
- [DP80] D. W. Davies and W. L. Price, "The application of digital signatures based on public key cryptosystems," *Proceedings of ICC*, 1980, pp.525-530
- [FIPS77] "Data Encryption Standard." Federal Information Processing Standards Publication 46, National Bureau of Standards, U.S. Department of Commerce, 1977
- [ISO8731/1] "Banking-Approved algorithm for message authentication - Part 1: DEA-1."
- [ISO8731/2] "Banking-Approved algorithm for message authentication - Part 2: Message authentication algorithm."
- [K91] T. Kaneko, "A known plaintext cryptanalytic attack on FEAL-4," *Technical Report of the Institute of Electronics, Information and Communication Engineers*, ISEC91-25 (1991)
- [M93] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *EUROCRYPT'93*
- [MKOM90] S. Miyaguchi, S. Kurihara, K. Ohta, and H. Morita, "Expansion of FEAL Cipher," *NTT Review*, Vol. 2, No. 6, 1990
- [MOI90] S. Miyaguchi, K. Ohta and M. Iwata, "Confirmation that Some Hash Functions are not Collision Free," *EUROCRYPT'90*
- [MSS88] S. Miyaguchi, A. Shiraiishi, and A. Shimizu, "Fast data encryption algorithm FEAL-8," *Review of Electrical Communication Laboratories*, Vol. 36, No. 4, 1988
- [MY92] M. Matsui and A. Yamagishi, "A New Method for Known Plaintext Attack of FEAL Cipher," *EUROCRYPT'92*
- [ZMI90] Y. Zheng, T. Matsumoto and H. Imai, "Structural Properties of One-Way Hash Functions," *CRYPTO'90*