New attacks on the MacDES MAC Algorithm

Don Coppersmith
IBM Research
T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
copper@watson.ibm.com

Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
c.mitchell@rhbnc.ac.uk

1st July 1999

Abstract

Two new attacks are given on a CBC-MAC algorithm due to Knudsen and Preneel, [2], which is in the final stages of being standardised as MAC Algorithm 4 in ISO/IEC FDIS 9797–1. The attacks are significantly more efficient than previously known attacks, which means that the inclusion of this scheme in the standard will need to be reconsidered.

1 Introduction

CBC-MACs, i.e. Message Authentication Codes (MACs) based on a block cipher in Cipher Block Chaining (CBC) mode, have been in wide use for many years for protecting the integrity and origin of messages. A variety of minor modifications to the 'basic' CBC-MAC have been devised and adopted over the years, in response to various cryptanalytic attacks (for a survey see [3]). The latest version of the international standard for CBC-MACs, ISO/IEC 9797–1, [1], which recently reached Final Draft International Standard (FDIS) status, contains a total of six different MAC algorithms.

This paper is concerned with one of these algorithms, namely MAC Algorithm 4. This algorithm has only recently been added to the draft international standard, and was intended to offer a higher degree of security than previous schemes at a comparable computational cost. It was originally proposed by Knudsen and Preneel, [2] and, when used with the DES block cipher, was given the name 'MacDES'. The attack described here does not completely invalidate the technique, but does show that its advantages over previous schemes are less significant than was thought.

2 Preliminaries

MAC algorithm 4 uses three block cipher keys, K, K' and K'', where either K'' is derived from K', or K' and K'' are both derived from a single key. However, for the attacks below we make no assumptions about how K' and K'' are related. We assume that the block cipher uses k-bit keys. We denote the block cipher encryption operation by $Y = e_K(X)$,

where Y is the n-bit ciphertext block corresponding to the n-bit plaintext block X, and K is the k-bit key. We denote the corresponding decryption operation by $X = d_K(Y)$.

The MAC is computed on a data string by first padding the data string so that it contains an integer multiple of n bits, and then breaking it into a series of n-bit blocks. If these blocks are $D_1, D_2, \ldots, D_q$, then the MAC computation is as follows.

$$\begin{aligned} H_1 &= e_{K''}(e_K(D_1)), \\ H_i &= e_K(D_i \oplus H_{i-1}), \quad (2 \le i \le q-1), \quad \text{and} \\ M &= e_{K'}(e_K(D_q \oplus H_{q-1})), \end{aligned}$$

for some $H_1, H_2, \ldots, H_{q-1}$. Finally, M is truncated as necessary to form the MAC.

ISO/IEC FDIS 9797–1 provides three different padding methods. Padding Method 1 simply involves adding between 0 and $n-1$ zeros, as necessary, to the end of the data string. Padding Method 2 involves the addition of a single 1 bit at the end of the data string followed by between 0 and $n-1$ zeros. Padding Method 3 involves prefixing the data string with an n-bit block encoding the bit length of the data string, with the end of the data string padded as in Padding Method 1.

When using one of the six MAC algorithms from ISO/IEC FDIS 9797–1, it is necessary to choose one of the three padding methods, and the degree of truncation to be employed. We consider the case where Padding Method 1 or 2 is used, and where there is no truncation. Hence, given that the block cipher in use has an n-bit block length, the MAC has $m = n$ bits. E.g., in the case of DES we have $m = n = 64$ and $k = 56$. The attacks described here only work for Padding Methods 1 and 2 because, for these two padding methods, for every padded string there is at least one message which pads to this string. Hence the attacks can work with arbitrary padded strings, in the knowledge that the corresponding message will always exist. However, this is no longer the case for Padding Method 3.

3 Attack 1

Suppose we have $2^{n/2}$ messages and their MACs, all computed using the same key. By routine probabilistic arguments (called the 'birthday attack', see [3]), there is a good chance that two of the messages will have the same MAC. Suppose the two padded strings are $D_1, D_2, \ldots, D_q$ and $E_1, E_2, \ldots, E_r$, and suppose that the common MAC is M.

Now submit two chosen padded strings for MACing, namely the strings one obtains by deleting the last block from each of the above two messages. If we suppose that the MACs obtained are M' and M'' respectively, then we know immediately that

$$d_{K'}(M') \oplus D_q = d_K(d_{K'}(M)) = d_{K'}(M'') \oplus E_r.$$

Now run through all possibilities L for the unknown key K', and set $x(L) = d_L(M')$ and $y(L) = d_L(M'')$. For the correct guess $L = K'$ we will have $x(L) = d_{K'}(M')$ and $y(L) = d_{K'}(M'')$, and hence $D_q \oplus x(L) = E_r \oplus y(L)$. This will hold for $L = K'$ and probably not for any other value of L, given that $k < n$ (if $k \ge n$ then either a second 'collision' or a larger brute force search will probably be required).

Having recovered K', we do an exhaustive search for K using the relation $d_{K'}(M') \oplus D_q = d_K(d_{K'}(M))$ (which requires 2^k block cipher encryptions). Finally we can recover K'' by exhaustive search on any known text/MAC pair, e.g. from the set of $2^{n/2}$, which again will require 2^k block cipher encryptions.

4 Attack 2

Although attack 2 has a larger overall complexity than attack 1, it requires a large number of MAC verifications rather than a large number of MAC/message pairs. In certain practical situations this may be easier to obtain.

Suppose we have a single plaintext/MAC pair, i.e. the padded message $D_1, D_2, \ldots, D_q$ ($q \geq 2$) and its MAC M. Then, as above:

$$\begin{aligned} H_1 &= e_{K''}(e_K(D_1)), \\ H_i &= e_K(D_i \oplus H_{i-1}), \quad (2 \leq i \leq q-1), \quad \text{and} \\ M &= e_{K'}(e_K(D_q \oplus H_{q-1})), \end{aligned}$$

for some $H_1, H_2, \ldots, H_{q-1}$. Next suppose we have the MAC M' for the same plaintext but with an additional (arbitrary) block, i.e. the padded message $D_1, D_2, \ldots, D_q, D_{q+1}$. The blocks $H_1, H_2, \ldots, H_{q-1}$ are as above, and we also have:

$$\begin{aligned} H_q &= e_K(D_q \oplus H_{q-1}), \quad \text{and} \\ M' &= e_{K'}(e_K(D_{q+1} \oplus H_q)), \end{aligned}$$

for some H_q.

For each of the 2^k possible keys L compute

$$\begin{aligned} x(L) &= d_L(M), \quad \text{and} \\ y(L) &= d_L(M'). \end{aligned}$$

For each key L construct a new $(q+2)$-block (padded) data string:

$$D_1, D_2, \ldots, D_q, D_{q+1}, D_{q+1} \oplus x(L) \oplus y(L).$$

Now verify whether the MAC for each data string equals M'. We have

Lemma 1 *If $K' = L$, and if $m(L)$ is the MAC for the $(q+2)$-block (padded) data string, then $m(L) = M'$.*

Proof First observe that, by definition:

$$m(L) = e_{K'}(e_K(D_{q+1} \oplus x(L) \oplus y(L) \oplus d_{K'}(M'))).$$

Hence, if $K' = L$ then we have

$$m(L) = e_{K'}(e_K(D_{q+1} \oplus d_{K'}(M))).$$

But, from above, we know that

$$d_{K'}(M) = e_K(D_q \oplus H_{q-1}) = H_q,$$

and hence we have

$$m(L) = e_{K'}(e_K(D_{q+1} \oplus H_q)) = M'$$

and the result follows. □

If $k < n$, it is unlikely that the MAC $m(L)$ will equal M' for any incorrect key L. This gives a simple way of discovering the key K' (and hence K and K'', exactly as above).

5 Complexity of attacks

Attack 1 requires 4×2^k block cipher encipherments (2×2^{56} to find K', and a further 2^{56} each to find K and K''), $2^{n/2}$ known data string/MAC pairs, and 2 chosen data string/MAC pairs. In the case of DES (where $n = 64$, $k = 56$ and $m = 64$), and using the terminology of ISO/IEC FDIS 9797–1, this means the key recovery attack has complexity

$$[4 \times 2^{56}, 2^{32}, 2, 0].$$

Similarly, attack 2 requires 4×2^k block cipher encipherments, 1 known data string/MAC pair, 1 chosen data string/MAC pair, and 2^k on-line MAC verifications. In the case of DES, this means the key recovery attack has complexity

$$[4 \times 2^{56}, 1, 1, 2^{56}].$$

(In both cases the figure of 4×2^{56} can be reduced to 3×2^{56} if K'' is derived from K'). Both attack complexities compare very favourably with the tuple

$$[2^{89}, 0, 2^{65}, 2^{55}]$$

for the best previous known attack on this scheme, as quoted in Table 2 of Annex B of ISO/IEC FDIS 9797–1, [1]. Indeed, the pair of tuples for the two attack complexities are comparable with the tuples $[2^{57}, 2^{32}, 0, 0]$ and $[2^{56}, 1, 0, 2^{56}]$ for the two best known key recovery attacks on MAC Algorithm 3 from ISO/IEC FDIS 9797–1, the algorithm which MacDES is designed to replace.

6 Preventing the attacks

The above attacks may not be realisable in practice, even for a block cipher like DES with relatively small values of m and n. However, if the existence of the attack gives concern then there are various possible countermeasures, two of which we list below.

- Probably the most effective countermeasure is to use Padding Method 3. This would normally be recommended whichever MAC scheme from ISO/IEC 9797–1 is in use.
- As with virtually all MAC schemes, additional security against key recovery attacks is obtained if a different (random) session key is used to generate every MAC, and this session key is then sent or stored with the MAC, encrypted under some Key Encrypting Key.

7 Conclusions

Two attacks on MAC Algorithm 4 from ISO/IEC FDIS 9797–1 have been described, which apply when it is used with Padding Method 1 or 2, and where no MAC truncation is employed. In this case the attacks are significantly more efficient than the best previously known attack, as documented in [2] and Annex B, Table 2 of ISO/IEC FDIS 9797–1.

References

[1] International Organization for Standardization, Genève, Switzerland. *ISO/IEC FDIS 9797–1, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*, November 1998.

[2] L.R. Knudsen and B. Preneel. MacDES: MAC algorithm based on DES. *Electronics Letters*, **34**:871–873, 1998.

[3] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.