A new key recovery attack on the ANSI retail MAC

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

13th November 2002

Abstract

A new type of attack is introduced which takes advantage of MAC truncation to simplify key recovery attacks based on MAC verifications. One example of the attack is described which, in certain circumstances, enables a more efficient attack than was previously known to be launched against the ANSI retail MAC. The existence of this attack means that truncation for this MAC scheme should be used with greater care than was previously believed, and very short MACs should be avoided altogether.

1 Introduction

MACs, i.e. *Message Authentication Codes*, are a widely used method for protecting the integrity and guaranteeing the origin of transmitted messages and stored files. To use a MAC it is necessary for the sender and recipient of a message (or the creator and verifier of a stored file) to share a secret key K, chosen from some (large) keyspace. The data string to be protected, D say, is input to a MAC function f, along with the secret key K, and the output is the MAC. We write MAC $= f_K(D)$. The MAC is then sent or stored with the message.

1.1 The ANSI retail MAC

The ANSI retail MAC scheme [1], otherwise known as CBC-MAC-Y or ISO/IEC 9797-1 algorithm 3 [3], operates as follows. Suppose the underlying block cipher has n-bit blocks and uses a key of k bits. If X is an n-bit block then we write $e_K(X)$ (or $d_K(X)$) for the block cipher encryption (or

decryption) of X using key K. A message D is first padded and split into a sequence of q n-bit blocks: $D_1, D_2, \ldots, D_q$. The MAC scheme uses a pair of keys K, K'. The MAC computation is as follows.

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

For the purposes of this paper we assume that the padding method does not involve prefixing the data with a length block. Note that the MAC used will be truncated to the left-most m bits of the MAC value given in the above equation, where $m \le n$.

1.2 Security considerations

Following the approach used in [3], we use a four-tuple $[a, b, c, d]$ to quantify the resources needed for an attack, where a denotes the number of off-line block cipher encipherments (or decipherments), b denotes the number of known data string/MAC pairs, c denotes the number of chosen data string/MAC pairs, and d denotes the number of on-line MAC verifications. Note c and d are distinguished because, in some environments, it may be easier for an attacker to obtain MAC verifications (i.e. to submit a data string/MAC pair and learn whether the MAC is valid) than to obtain the MAC for a chosen message.

In the analysis of MAC algorithms based on a block cipher with a k-bit key, it is standard to assume that the block cipher itself is secure, and hence a key recovery attack will require at least 2^k invocations of the block cipher.

The best known key recovery attack on the ANSI retail MAC algorithm has complexity $[2^{k+1},2^{n/2},0,0]$, as described in [7]. An alternative key recovery attack, requiring only one known MAC/data string pair, but a larger number of verifications, is presented in [5]; this attack has complexity $[2^k,1,0,2^k]$. Finally, a further key recovery attack based on MAC verifications is presented in [4]; this latter attack has complexity

$$[2^{k+1}, \lceil(\max(k,n)+1)/m\rceil, 0, \lceil(k-n+m+1)/m\rceil 2^n]$$

given that $k - n + m + 1 > 0$.

2 The new attack

Suppose that an attacker has access to a device capable of performing MAC verifications (this is the scenario of the attack described in [5]). We now present an attack which, in many cases where $m < n$, requires fewer MAC verifications than the attacks of [4, 5].

2.1 Attack description

The attack operates as follows. The attacker first generates a sequence of messages by some means, and for each generated message M tests whether or not $f_{K,K'}(M) = 0^m$, where f denotes the ANSI retail MAC function, and 0^m denotes a block of m zero bits. If this equation holds then the message M is stored, and this process continues until a total of $2^{(n-m)/2}$ messages have been found for which the above equation holds. Label this collection of messages,

$$\{M_0, M_1, \ldots, M_{2^{(n-m)/2}-1}\}.$$

The attacker now chooses $\lceil n/m \rceil - 1$ arbitrary n-bit blocks

$$\{X_1, X_2, \ldots, X_{\lceil n/m \rceil - 1}\}.$$

For every i $(1 \leq i \leq \lceil n/m \rceil - 1)$ the attacker now uses as many MAC verifications as necessary to learn the MACs of the following set of $2^{(n-m)/2}$ messages:

$$\{M_0||X_i, M_1||X_i, \ldots, M_{2^{(n-m)/2}-1}||X_i\}.$$

The attacker now assembles a collection of $2^{(n-m)/2}$ ordered $(\lceil n/m \rceil - 1)$-tuples of MACs, computed on the $(\lceil n/m \rceil - 1)$-tuples of messages

$$(M_j||X_1, M_j||X_2, \ldots, M_j||X_{\lceil n/m \rceil - 1})$$

$0 \leq j \leq 2^{(n-m)/2} - 1$. If two of these tuples of MACs are equal, say the MACs are equal for the tuples of messages

$$(M_j||X_1, M_j||X_2, \ldots, M_j||X_{\lceil n/m \rceil - 1})$$

and

$$(M_{j'}||X_1, M_{j'}||X_2, \ldots, M_{j'}||X_{\lceil n/m \rceil - 1})$$

than there is a very good chance that a 'real' collision has been found, i.e. messages M_j and $M_{j'}$ whose n-bit MACs prior to truncation are equal.

Such a message pair can then be used to launch an exhaustive key search to recover the MAC key, exactly as described in [7].

2.2 Analysis of attack

First note that, given an arbitrary message M, the probability that $f_{K,K'}(M) = 0^m$ can be assumed to be 2^{-m}. Hence, the expected number of MAC verifications required to find the sequence of messages $M_0, M_1, \ldots, M_{2^{(n-m)/2}-1}$ all having a zero MAC is simply

$$2^{(n-m)/2} \times 2^m = 2^{(n+m)/2}.$$

Next observe that the expected number of MAC verifications necessary to learn a MAC is precisely 2^{m-1}. Hence, the total number of MAC verifications required for the next stage of the attack, i.e. to compute all the tuples of MACs, is

$$2^{m-1} \times 2^{(n-m)/2} \times (\lceil n/m \rceil - 1) = (\lceil n/m \rceil - 1)2^{(n+m)/2-1}.$$

This gives a total number of MAC verifications of $(\lceil n/m \rceil + 1)2^{(n+m)/2-1}$.

Now if M_j and $M_{j'}$ have the property that their n-bit MACs prior to truncation are equal, then this will also be true for $M_j||X$ and $M_{j'}||X$ for any block X. Hence, if two such messages M_j and $M_{j'}$ occur in the set, then they will be detected by the described approach. We next need to consider the probability that there will exist two such messages. Given that we chose the set of messages to have the property that the left-most m bits of the untruncated MAC are equal, then we need to consider the probability that a set of $2^{(n-m)/2}$ messages will contain a pair of messages whose untruncated n-bit MACs agree in the right-most $n-m$ bit positions. By the usual 'birthday paradox' arguments (see, for example, section 2.1.5 of [6]) there is a good chance that two such messages will exist.

We also need to consider the probability that a pair of messages will give rise to a 'false alarm', i.e. that there will exist a pair of messages whose tuples of MACs agree but whose untruncated MACs disagree. The number of such pairs will be small, and such pairs can be eliminated by using a small number of additional MAC verifications.

Finally observe that, given a 'colliding pair' of messages, the attack to recover the key, as described in [7], requires 2^{k+1} off-line encryptions (assuming that the messages are all short — in the scenario described here that is simple to arrange). Thus the total attack complexity is

$$[2^{k+1}, 0, 0, (\lceil n/m \rceil + 1)2^{(n+m)/2-1}].$$

We conclude by tabulating the complexity of the attack (in terms of MAC verifications) for some typical values of m and n.

Table 1: MAC verifications required for attack

n	m	Number of MAC verifications
64	8	2^{38}
64	16	2^{41}
64	24	2^{45}
64	32	2^{49}
64	48	2^{56}

3 Observations

As can be seen from Table 1, for DES this attack is more efficient than the attack of [5] (which requires 2^{56} MAC verifications) for all cases where $m < 48$. The rather surprising conclusion is that the use of short MACs makes launching key recovery attacks easier!

Of course, it might be argued that when MACs are truncated the trivial 'verification forgery' attack (see, for example, [2, 3]) can be used. That is, when an m-bit MAC is used, a forgery can be found using 2^{m-1} MAC verifications (on average). Although the above attack is more powerful in that it recovers the key, the number of MAC verifications required is so much larger than the number required for a single forgery that it may not have any importance in practice.

However, if the MAC key (or a simple variant of the key) is also used for encryption then key recovery is clearly a very serious issue. For example, where the block cipher is DES, closely related pairs of keys may be used for ANSI retail MAC computation and triple DES encryption. Typically the MAC key pair is derived from the encryption key pair by ex-oring it with a fixed mask. In such a case the attack described here has potentially serious consequences.

4 Conclusions

A MAC verification based key recovery attack on the ANSI retail MAC has been demonstrated that, in the case where the MAC is truncated, can require significantly fewer MAC verifications than previously known attacks. This is especially significant where a MAC key (or a simple variant of it) may also be used for encryption.

References

[1] American Bankers Association, Washington, DC. *ANSI X9.19, Financial institution retail message authentication*, August 1986.

[2] K. Brincat and C. J. Mitchell. New CBC-MAC forgery attacks. In V. Varadharajan and Y. Mu, editors, *Information Security and Privacy, ACISP 2001, Sydney, Australia, July 2001*, number 2119 in Lecture Notes in Computer Science, pages 3–14. Springer-Verlag, Berlin, 2001.

[3] International Organization for Standardization, Genève, Switzerland. *ISO/IEC 9797–1, Information technology — Security techniques — Mes-*

sage Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher, 1999.

[4] L. R. Knudsen and C. J. Mitchell. Analysis of 3gpp-MAC and two-key 3gpp-MAC. *Discrete Applied Mathematics*, to appear.

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

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

[7] B. Preneel and P.C. van Oorschot. A key recovery attack on the ANSI X9.19 retail MAC. *Electronics Letters*, **32**:1568–1569, 1996.