A key recovery attack on the ANSI X9.19 retail MAC

B. Preneel and P.C. van Oorschot

Indexing terms: Cryptography, Message Authentication

This letter presents a new divide and conquer key recovery attack on the retail MAC based on DES, which is a widely used algorithm to compute a Message Authentication Code. The attack requires $2^{32.5}$ known text-MAC pairs and $3 \cdot 2^{56}$ off-line computations to find the 112-bit key.

Introduction: Message authentication code (MAC) algorithms are commonly used to provide data integrity and data origin authentication, e.g. in banking applications [1]. They are used in a symmetric setting, where a sender and a receiver share a secret key (uppercase) K. In order to protect a message x, the sender computes $\mathrm{MAC}_K(x)$, which is a complex function of every bit of the message and the key and appends this to the message. On receipt of x, the receiver recomputes $\mathrm{MAC}_K(x)$ and verifies that it corresponds to the transmitted MAC value. In the following the key size in bits is denoted (lowercase) k, and the MAC size is denoted with m. Typical values for m are between 32 and 64 bits.

In order for a MAC to be secure, it must satisfy the following condition: without knowledge of the secret key K, it must be computationally infeasible to perform a *forgery*, i.e. to find an arbitrary message and its corresponding MAC value. Here it is assumed that the opponent is capable of performing a *chosen text attack*, i.e. he can obtain MACs corresponding to a number of messages of his choice. To be meaningful, a forgery must be for a message different than any for which a MAC was previously obtained. A *key recovery* attack is stronger than a MAC forgery: once an opponent has obtained the secret key K, he can forge message-MAC pairs at will. For an *ideal* MAC, any method to find the k-bit key is as expensive as an exhaustive search of 2^k operations. The number of text-MAC pairs required for verification of such an attack is about k/m.

CBC-MAC: By far the most common MAC is CBC-MAC, which has been standardized ubiquitously [2, 3, 4, 5]. It is widely used with DES [6] as the underlying block cipher E. CBC-MAC is defined as follows: divide the input x into t blocks of n bits each, x_1 through x_t (this might involve an unambiguous padding operation) and perform the following iterative computation:

$$H_i = E_K(H_{i-1} \oplus x_i)\,, \quad 1 \le i \le t\,.$$

Here $E_K(x)$ denotes the encryption of x using key K with an n-bit block cipher E and $H_0 = 0$. The MAC is then computed as $\mathrm{MAC}_K(x) = g(H_t)$, where g is the output transformation. The mapping g is required to preclude a simple forgery attack (see e.g., [7]).

One approach is for g to select the leftmost m bits. A widely used alternative is to replace the processing of the last block by a two-key triple encryption (with keys $K_1 = K$ and K_2); this is commonly known as the retail MAC, since it first appeared in [3] (see Figure 1):

$$g(H_t) = E_{K_1}(D_{K_2}(H_t)) = E_{K_1}(D_{K_2}(E_{K_1}(x_t \oplus H_{t-1}))) \, .$$

D denotes decryption here. This mapping has the additional advantage that it precludes an exhaustive search against the DES key, which is only 56 bits long. It is widely accepted that currently such a key does not offer sufficient protection against exhaustive key search [8]. This advantage, which requires little extra computation (only two encryptions), has resulted in a widespread use of this variant.

The currently best known attack on CBC-MAC has been presented in [7]: it is a forgery attack which requires about $2^{32.5}$ known text-MAC pairs and a single chosen text. However, this attack does not pose a problem for many environments: e.g., in the banking world, allowing an opponent to choose one single text and to obtain the corresponding MAC can already jeopardize the system, since such a text-MAC pair could be sufficient to make a substantial profit.

New attack: This section presents a divide and conquer key recovery attack on the retail MAC. The attack requires $2^{32.5}$ *known* texts (i.e. text-MAC pairs) and about $3 \cdot 2^{56}$ off-line operations when DES is used as the underlying block cipher ($n = 64$, $k = 56$). The latter figure is much smaller than what is suggested by the key size of 112 bits.

Proposition 1 *For the retail MAC [3, 5], a key recovery attack yielding both keys K_1 and K_2 requires $2^{(n+1)/2}$ known texts of at most t blocks each ($t \geq 2$) and exhaustive* *search involving at most* $(2t-1)\cdot 2^k$ *encryptions, where* $k = |K_1| = |K_2|$*,* $k \le n$*, and* $m = n$*.*

Proof: The statement is substantiated by giving the attack itself. By the birthday paradox, the set of $r = 2^{(n+1)/2}$ known texts contains a collision, i.e., a pair of texts with the same MAC value (indeed, $\binom{r}{2}/2^n \approx r^2/2^{n+1} = 1$). Since g is a permutation, this collision pair will have the same value for H_t and thus for $G = H_{t-1} \oplus x_t$. An attacker can then perform an exhaustive search for K_1 in $2(t-1)\cdot 2^k$ off-line operations. This involves computing G for each of the two colliding messages, and eliminating all trial key values not yielding a collision for G. Since $k \le n$, K_1 can be determined uniquely using one internal collision (at most 1 spurious value for K_1 is expected). Now compute $G' = E_{K_1}(G)$ and $G'' = D_{K_1}(\mathrm{MAC}(x))$ for any known text-MAC pair $(x, \mathrm{MAC}(x))$ and exhaustively check all K_2 until finding a value for which $D_{K_2}(G') = G''$. The solution (K_1, K_2) can be confirmed by testing it on one of the other known text-MAC pairs. If a spurious key K_1 arises, it can be eliminated by either exhaustively searching all values of K_2 or by increasing the number of known texts until a second MAC collision is found. ■

Proposition 1 can be generalized as follows:

i) If $k > n$, slightly more than k/n internal collisions are required to determine K_1 uniquely; similarly, about k/n of the known text-MAC pairs will be required to isolate the correct key K_2 during the exhaustive search.

ii) If $m < n$, the expected number of MAC collisions is equal to $r^2/2^{m+1} = 2^{n-m}$ (while still only a single collision for G is expected) [9]. This will increase the effort for the key search with a factor of 2^{n-m}. Alternatively, the texts which give a collision for the MAC but not for G can be eliminated using about 2^{n-m} chosen texts [7]. In addition, the recovery of K_2 requires forward computation (2^k extra encryptions in total) from G' to MAC(x) since in this case G'' cannot be recovered from MAC(x).

Conclusions: This letter shows that the ANSI X9.19 retail MAC based on an n-bit block cipher offers no increased strength against exhaustive key search if about $2^{n/2}$ known text-MAC pairs are available. The key recovery attack can be avoided by using a triple DES encryption in every iteration.

B. Preneel[1] (Katholieke Universiteit Leuven, Department Electrical Engineering-ESAT, COSIC, Kardinaal Mercierlaan 94, B–3001 Heverlee, Belgium)

P.C. van Oorschot (Bell-Northern Research/Nortel Secure Networks, P.O. Box 3511 Station C, Ottawa, K1Y 4H7, Canada)

References

[1] DAVIES, D. and PRICE, W.: 'Security for computer networks (2nd ed.)' (Wiley, 1989)

[2] ANSI X9.9 (revised): 'Financial institution message authentication (wholesale)' (American Bankers Association, April 7, 1986)

[1] N.F.W.O. postdoctoral researcher, sponsored by the National Fund for Scientific Research (Belgium).

[3] ANSI X9.19: ‘Financial institution retail message authentication’ (American Bankers Association, August 13, 1986)

[4] ISO 8731: ‘Banking – approved algorithms for message authentication, Part 1, DEA, Part 2, Message authentication algorithm (MAA)’ (ISO, 1987)

[5] ISO/IEC 9797: ‘Information technology - Data cryptographic techniques - Data integrity mechanisms using a cryptographic check function employing a block cipher algorithm’ (ISO/IEC, 1993)

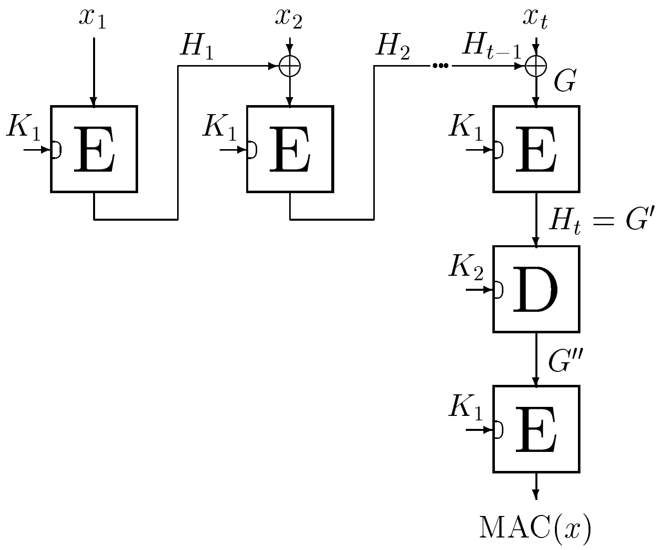
[6] FIPS 46: ‘Data encryption standard’ (NBS, U.S. Department of Commerce, January 1977)

[7] PRENEEL, B. and VAN OORSCHOT, P.C.: ‘MDx-MAC and building fast MACs from hash functions’, Advances in Cryptology, CRYPTO’95, Lect. Notes Comput. Sci. 963, (Springer-Verlag, 1995), pp. 1–4

[8] WIENER, M.J.: ‘Efficient DES key search’, Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, Canada, May 1994

[9] FELLER, W.: ‘An introduction to probability theory and its applications, vol. 1’ (Wiley, 1968)

Figure 1: Retail MAC: A strengthened version of CBC-MAC from ANSI X9.19 and ISO/IEC 9797.


x_1
H_1
x_2
H_2
H_{t-1}
x_t
G
K_1
E
K_1
E
K_1
E
$H_t = G'$
K_2
D
G''
K_1
E
$\mathrm{MAC}(x)$