On the security margin of MAC striping

Thomas Eisenbarth[1], Aaron Meyerowitz[2], and Rainer Steinwandt[2]

[1] Worcester Polytechnic Institute (WPI), 100 Institute Rd, Worcester, MA 01609
teisenbarth@wpi.edu
[2] Florida Atlantic University (FAU), 777 Glades Rd, Boca Raton, FL 33431
{meyerowi,rsteinwa}@fau.edu

Abstract. MAC striping has been suggested as a technique to authenticate encrypted payloads using short tags. For an idealized MAC scheme, the probability of a selective forgery has been estimated as $\binom{\ell+m}{m}^{-1} \cdot 2^{-m}$, when utilizing MAC striping with ℓ-bit payloads and m-bit tags. We show that this estimate is too optimistic. For $m \leq \ell$ and any payload, we achieve a selective forgery with probability $\geq \binom{\ell+m}{m}^{-1}$, and usually many orders of magnitude more than that.

Keywords: cryptanalysis, MAC, MAC striping, selective forgery

1 Introduction

When dealing with sensor networks, e. g., in mHealth applications, it may be necessary to authenticate very short payloads. A standard solution with a Message Authentication Code (MAC) can result in drastic data overhead. Mare et al. proposed *MAC striping* as a technique to alleviate this problem to some extent [1, 2]. The central idea is to place the tag bits into the payload at random positions, determined by a secret key. For a tag with m bits and a payload with ℓ bits, a security margin of $\binom{\ell+m}{m} \cdot 2^m$ was attributed to this construction, suggesting the use of remarkably short tags.

After recalling the description of MAC striping and discussing the notion of a selective forgery in this context, we show that the security margin of MAC striping has been overestimated significantly.

2 Preliminaries

Commonly, a MAC algorithm is formalized as a triple of algorithms:

Key generation: this algorithm chooses a secret key K.
Tag generation: this algorithm receives as input a payload P and the secret key K, and outputs a tag T for P.
Verification: this algorithm receives as input the secret key K, a payload P, and a tag T and outputs valid or invalid.

This formalization is tailored for a scenario where the tag is expected to be sent separately from the payload to be verified.

2.1 MAC striping

Instead of *concatenating* payload and tag, MAC striping *intermixes* them [1, 2]. Tag bit positions are chosen by means of a pseudo random generator function which depends, among other things, on a secret key.[3] For an ℓ-bit payload P and m-bit tag T, there are $\binom{\ell+m}{m}$ choices to place the tag bits in the intermixed *message*. We assume that the tag T is chosen uniformly at random in $\{0,1\}^m$ and the set S of tag bit positions is chosen uniformly at random among all $\binom{\ell+m}{m}$ size-m subsets of $\{1,\ldots,\ell+m\}$. If the same payload is processed repeatedly, each time a new tag and new tag positions are selected. This is consistent with the security analysis in [2], stating

> "to forge a message of his choice, the adversary has to guess the matching MAC bits out of 2^m possibilities, and MAC-bit locations out of $\binom{\ell+m}{m}$ possible MAC-bit locations, making the probability of success $\frac{1}{2^m\binom{\ell+m}{m}}$."

The verification algorithm does not receive payload and tag as separate inputs. Instead, it receives as input a single message M of length $m+\ell$ and the (payload-independent) secret key. The original analysis overlooks the fact that a given payload P can lead to the same message for several choices S and T. The probability $p(M|P)$ that a particular payload P results in a message M is

$$p(M|P) = \frac{c(M,P)}{2^m\binom{\ell+m}{m}}$$

where $c(M,P)$ is the number of position/tag pairs (S,T) that produce M from P. In other words, $c(M,P)$ is the number of ways to obtain P from M by deleting m bits. So $\sum\limits_{P\in\{0,1\}^\ell} c(M,P) = \binom{\ell+m}{m}$ and $\sum\limits_{M\in\{0,1\}^{\ell+m}} c(M,P) = \binom{\ell+m}{m}2^\ell$.

2.2 Selective forgeries

With a standard definition of a selective forgery, the adversary is asked to generate a valid payload/tag pair (P,T) for a payload of his choice. Following the usual convention from signature schemes, to be "of the adversary's choice" it suffices to commit to the payload prior to the attack. With this convention, restricting to the payload $P = 0^\ell$ is legitimate. We also consider the universal scenario where $P \in \{0,1\}^\ell$ is provided to the adversary before the attack.

When using MAC striping for a given payload P, the adversary does not need to select a tag T and position set S. If he is able to generate a message M which the verification algorithm classifies as valid and where the correct payload P is recovered by the recipient, the selective forgery succeeded: the chosen payload has been accepted as authentic. Note that the requirement of [2] that the payload P is an encrypted version of the intended text is immaterial here, as the verification and adversary both interact only with the payload P.

[3] In the context considered by Mare et al., a message *header* is involved as well, but it is immaterial for our analysis of MAC striping. So we omit a discussion of header information.

3 Reevaluating the security margin

Fix a payload $P \in \{0,1\}^\ell$ which the adversary will attempt to have authenticated. Our adversary will only submit messages of the correct length $\ell + m$ with $c(M,P) > 0$. The number of such messages turns out to be independent of P and is given in the following lemma. Diggavi et al. [3] attribute the first part of this result to Chvátal and Sankoff [4]. We provide a proof in the appendix.

Lemma 1. *The number of binary sequences of length $\ell + m$ containing a fixed $P \in \{0,1\}^\ell$ as a subsequence is $\sum_{k=0}^{m} \binom{\ell+m}{k}$. For $1 < m < \ell$, this number is greater than $\binom{\ell+m}{m}$ and less than $\frac{\ell}{\ell-m}\binom{\ell+m}{m}$.*

Hence, submitting (any of the) the most likely message(s) of length $\ell + m$ containing P as a subsequence results in a successful forgery with probability *at least* $\left(\sum_{k=0}^{m} \binom{\ell+m}{k}\right)^{-1}$.

For a given payload P, there are at least m messages with $c(M,P) = 1$: distribute the tag bits at the beginning and/or end of the message and make each different than the initial or terminal bit of P. Each of these messages is correct with probability $\frac{1}{2^m\binom{\ell+m}{m}}$. So one can expect that the most likely messages are accepted with probability much greater.

Example The values $\ell = 80$ and $m = 16$ have been considered in [2], and the success probability for a selective forgery is estimated to be $\frac{1}{2^{16}\binom{96}{16}} \approx 2^{-75.20}$. From Lemma 1 we see that the number of 96-bit messages containing any given 80-bit payload is $\sum_{k=0}^{16}\binom{96}{k} \approx 2^{59.51}$, and the upper bound given for this sum was $\frac{80}{80-16}\binom{96}{16} \approx 2^{59.52}$. This guarantees a success probability of $\approx 2^{-59.51}$, but the adversary can do much better. Let us look at two specific payloads which we suspect give the extremes.

$P = 0^{80}$: The message 0^{96} succeeds with probability 2^{-16} (note that the length of the payload is irrelevant here).

$P = (01)^{40}$: The message $(01)^{48}$ succeeds with probability $\frac{\binom{88}{8}}{2^{16}\binom{96}{16}} \approx 2^{-39.30}$. Where does the numerator come from? We must delete some 16 bits leaving the desired payload. When we choose a certain bit to delete we have no choice but to delete the next bit as well. Then we are free to keep or delete the following bit. So the deleted bits must be 8 adjacent pairs; any such choice will work. So, consider all the ways to line up eight 1×2 rectangular boxes and eighty 1×1 squares in a line, there are $\binom{88}{8}$. Now write $\{0,1\}^{48}$ in order with 2 bits in each rectangle and one in each square. The bits in the squares form the payload.

3.1 Selective forgery for arbitrary prescribed payload

Fix a payload $P = (P_1, \ldots, P_\ell) \in \{0,1\}^\ell$. We give a method to find a message M with $c(M,P) \geq 2^m$ assuming $m \leq \ell$. In fact, with the suggested strategy we can often do much better than this. We know that a message $M \in \{0,1\}^{\ell+m}$ will be accepted with probability $p(M|P) = \frac{c(M,P)}{2^m\binom{\ell+m}{m}}$ where $c(M,P)$ is the number of ways to delete m bits from M and recover P. Write the payload in the form $P = 0^{a_1}1^{a_2}0^{a_3}\cdots 1^{a_t}$ in terms of the runs of consecutive digits with all a_i being positive.[4] We will consider messages with the same run structure. So consider a possible message $M = 0^{a_1+b_1}1^{a_2+b_2}0^{a_3+b_3}\cdots 1^{a_t+b_t}$ where the b_i are non-negative and $\sum_{i=1}^{t} b_i = m$. The number of ways, $c(M,P)$, to choose some m bits to delete and thereby recover P, is the product $\prod_{i=1}^{t}\binom{a_i+b_i}{b_i}$. The terms with $b_i = 0$ are 1 and can be ignored. Clearly it is optimal, with this restriction on the run structure, to use $b_i > 0$ for the largest of the a_i.

Example Consider a payload of length 80 with 1 run of length 6, 2 of length 5, 5 of length 4 and 10 of length 3 and the rest of length 1 or 2. Putting one extra bit in each of the 16 longest runs gives $c(M,P) = \binom{7}{1}^1\binom{6}{1}^2\binom{5}{1}^5\binom{4}{1}^8 \approx 2^{35.59}$. There are, for this example, $\binom{10}{2} = 45$ ways to choose a message M with this success probability.

Experiments with smaller ℓ-values suggest that the easiest payloads to forge have few runs. Also that, for a large proportion of the possible payloads, messages with the same run structure as the payload are better than those with more runs. The worst case for the "maintain the run structure" construction is if all the runs are of length 1. For $m \le \ell$, in this case we still obtain $c(M,P) = 2^m$, but the strategy is not optimal: as we saw above, the message $(01)^{48}$ gives $c(M,P) = \binom{88}{8} \approx 2^{35.90}$, which is about as good as the previous example.

4 Conclusion

MAC striping is an appealing idea, but the lower bound we established with the "maintain the run structure" strategy shows that selective forgeries are possible with higher probability than expected. We emphasize that it is a lower bound only—as demonstrated, the adversary may be able to do much better. The all-zero and all-one payload are extreme examples where the success probability can be be quite high—independent of the message length. Overall, our analysis shows that MAC striping is significantly less useful than hoped for.

References

1. S. Mare, J. Sorber, M. Shin, C. Cornelius, D. Kotz, Adapt-lite: Privacy-aware, secure, and efficient mHealth sensing, in: Proceedings of the Workshop on Privacy in the Electronic Society (WPES), ACM, 2011, pp. 137–142.
2. S. Mare, J. Sorber, M. Shin, C. Cornelius, D. Kotz, Hide-n-Sense: preserving privacy efficiently in wireless mHealth, Mobile Networks and Applications 19 (3) (2014) 331–344.

[4] For simplicity, we assume that the first bit is a 0 and that the number of runs is even. There is no loss of generality in this—allow $a_1 = 0$ and/or $a_t = 0$ if desired.

3. S. N. Diggavi, M. Grossglauser, On Transmission Over Deletion Channels, in: Allerton Conference, 2001, pp. 573–582, available at `http://icapeople.epfl.ch/grossglauser/Papers/allerton01.pdf`.
4. V. Chvátal, D. Sankoff, Longest Common Subsequence of Two Random Sequences, Journal of Applied Probability 12 (2) (1975) 306–315.
5. M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, P. DeMarco, Maple 10 Programming Guide, Maplesoft, 2005.
6. P. S. Foundation, Python Programming Language – Official Website, `http://www.python.org` (2001–2014).

Appendix. Proof of Lemma 1

Fix integers $\ell, m > 0$, and let

$$\mathcal{M}_{\ell,m}(P) = \mathcal{M}(P) = \{M \in \{0,1\}^{m+\ell} \mid c(M,P) \geq 1\}.$$

The lemma we wish to prove is the following.

For any fixed $P \in \{0,1\}^\ell$, one has $\mid \mathcal{M}(P) \mid = \sum_{k=0}^{m} \binom{\ell+m}{k}$. For $1 < m < \ell$, $\binom{\ell+m}{m} < \mid \mathcal{M}(P) \mid < \frac{\ell}{\ell-m}\binom{\ell+m}{m}$.

For $M = (M_1, M_2, \cdots, M_{\ell+m}) \in \mathcal{M}(P)$ we will define the *greedy placement* of P in M to be the list $i_1 < i_2 < \cdots < i_\ell$ of positions giving the leftmost placement of P in M. It will help to also let $i_0 = 0$. Then the greedy placement is more formally defined by specifying that for each $j \geq 1$ the following hold.

– $i_j > i_{j-1}$ with $M_{i_j} = P_j$ and is the smallest index for which both of these are true.

We first establish a different formula $\mid \mathcal{M}(P) \mid = \sum_{k=\ell}^{m+\ell} \binom{k-1}{\ell-1} 2^{m+\ell-k}$ independent of P. Rather than showing directly that this sum is equal the desired one, we simply observe that for the payload $P = 0^\ell$, it is clear that $|\mathcal{M}(0^\ell)| = \sum_{k=0}^{m} \binom{\ell+m}{k}$: The set $\mathcal{M}(0^\ell)$ consists of exactly the words of length $\ell + m$ with Hamming weight $0 \leq k \leq m$. The number of words of this length with Hamming weight k is $\binom{m+\ell}{k}$.

Suppose we are given a payload P and the information that M is some word in $\mathcal{M}(P)$ such that the greedy placement of P in M has $i_\ell = k$. So $\ell \leq k = i_\ell \leq \ell + m$. Then there are $\binom{k-1}{\ell-1}$ ways to get the full greedy placement $i_0 = 0, i_1, i_2, \cdots, i_\ell$. This further information limits M to be one of $2^{m+\ell-k}$ words. This because we then know all the values M_i for $i \leq i_\ell = k$ but nothing about the later positions: If $i_{j-1} < i < i_j$ then $M_i = 1 - P_j$ since $i_j \neq i$, but $M_i = P_j$ if $i = i_j$. However M_i is unconstrained for $i > i_\ell$. Summing over the possible values, $\mid \mathcal{M}(P) \mid = \sum_{k=\ell}^{m+\ell} \binom{k-1}{\ell-1} 2^{m+\ell-k}$.

It remains only to establish the bounds given for $m < \ell$. Namely that

$$\binom{\ell+m}{m} < \sum_{j=0}^{m} \binom{\ell+m}{m-j} < \frac{\ell}{\ell-m}\binom{\ell+m}{m}.$$

Note that the lower bound is immediate and that we have changed the order of summation for convenience.

Define $r = \frac{m}{\ell}$ and note that $\binom{N}{K-1} = \frac{K}{N-K+1}\binom{N}{K}$. By repeated application of this identity,

$$\binom{\ell+m}{m-j} = \frac{m}{\ell+1}\frac{m-1}{\ell+2}\cdots\frac{m-j+1}{\ell+j}\binom{\ell+m}{m-j-1} \le r^j\binom{\ell+m}{m}.$$

Hence

$$\begin{aligned}\sum_{j=0}^{m}\binom{\ell+m}{m-j} < (1+r+r^2+\cdots+r^m)\binom{\ell+m}{m} &= \frac{1-r^{m+1}}{1-r}\binom{\ell+m}{m} \\ &< \frac{1}{1-r}\binom{\ell+m}{m}.\end{aligned}$$

Since $\frac{1}{1-r} = \frac{\ell}{\ell-m}$, we have proved what we need. □

Remark 1. We could get a tighter (and slightly more complicated) upper bound by using $r = \frac{m}{\ell+1}$ and by stating the bound as $\sum_{j=0}^{m}\binom{\ell+m}{m-j} < \frac{1-r^{m+1}}{1-r}\binom{\ell+m}{m}$. Bounds could be given for the cases where $m \approx \ell$ or $m > \ell$, but we have not done so here.