CBC-MACs
MACs of variable length

Foundations of Cryptography
Computer Science Department
Wellesley College

Fall 2016

Table of contents

Secure communication and message integrity

- Last time we discussed a paradigm for constructing secure message authentication codes based on pseudorandom functions.
- Unfortunately, the construction is only capable of dealing with *fixed length* messages and shorts ones at that.
- Here we how variable-length MACs can be constructed from fixed-length ones.

Breaking the code

Let $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ be a secure fixed length MAC for messages of length n. In each of the following three extensions, break messages m into blocks $m_1, \ldots, m_d$ of length n.

1. XOR all the the blocks together and authenticate the result, i.e., tag $t := \mathsf{Mac}'_k(\oplus_i m_i)$.
2. Authenticate each block separately, i.e., compute $t_i = \mathsf{Mac}'_k(m_i)$ and output $t = \langle t_1, \ldots, t_d \rangle$ as the tag.
3. Authenticate each block along with a sequence number, i.e., $t_i := \mathsf{Mac}'_k(i \| m_i)$ and output $t = \langle t_1, \ldots, t_d \rangle$ as the tag.

Curses, foiled again

4. The truncation attack can be thwarted by authenticating the message length along with each block.* In other words, we compute $t_i = \mathrm{Mac}'_k(\ell \parallel i \parallel m_i)$ for all i, where ℓ denotes the message length.

5. We can prevent the *mix-and-match* attack by also including a random "message identifier" along with each block that prevents blocks from different messages from being combined.

*Authenticating the message length as a separate block is not a good idea.

Constructing variable-length message authentication codes

Construction 4.7.

Let $\Pi' = (\mathrm{Gen}', \mathrm{Mac}', \mathrm{Vrfy}')$ be a fixed length MAC for messages of length n. Define a variable-length MAC as follows:

- Gen: This is identical to Gen'.
- Mac: On input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^*$ of length $\ell < 2^{\frac{n}{4}}$, parse m into d blocks $m_1, \ldots, m_d$, each of length $n/4$. Next choose a random identifier $r \leftarrow \{0,1\}^{n/4}$. For $i = 1, \ldots, d$, compute $t_i \leftarrow \mathrm{Mac}'_k(r \| \ell \| i \| m_i)$, where i and ℓ are uniquely encoded as strings of length $n/4$.
- Vrfy: On input a key $k \in \{0,1\}^n$, a message $m \in \{0,1\}^*$ of length $\ell < 2^{\frac{n}{4}}$, and a tag $t = \langle r, t_1, \ldots, t_{d'} \rangle$, parse m into d blocks $m_1, \ldots, m_d$, each of length $n/4$. Output 1 if and only if $d' = d$ and $\mathrm{Vrfy}'_k(r \| \ell \| i \| m_i, t_i) = 1$ for $1 \leq i \leq d$.

Construction 4.7 produces a secure MAC if it starts with one

Theorem 4.8. If Π' is a secure fixed-length MAC for messages of length n, then Construction 4.7 is a MAC that is existentially unforgeable under an adaptive chosen-message attack.

Secure MACs: A reminder

The message authentication experiment $\text{Mac-forge}_{\mathcal{A},\Pi}(n)$*:*

1. A random key k is generated by running $\text{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\text{Mac}_k(\cdot)$. The adversary eventually outputs a pair (m, t). Let $\mathcal{Q}$ denote the set of all queries that $\mathcal{A}$ asked to its oracle.
3. The output of the experiment is defined to be 1 if and only if (1) $\text{Vrfy}(m, t) = 1$; and (2) $m \notin \mathcal{Q}$.

Definition 4.2. A message authentication code $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ is *existentially unforgeable under an adaptive chosen-message attack* if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1] \leq \text{negl}(n).$$

Back to the proof of Theorem 4.8

Theorem 4.8. If Π' is a secure fixed-length MAC for messages of length n, then Construction 4.7 is a MAC that is existentially unforgeable under an adaptive chosen-message attack.
Proof. Let Π denote the MAC given by Construction 4.7. Let $\mathcal{A}$ be a PPT adversary. We will show that $\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1]$ is negligible.

Let Repeat denote the event that the same message identifier appears in two of the tags returned by the MAC oracle in experiment $\text{Mac-forge}_{\mathcal{A},\Pi}(n)$.

If $(m, t = \langle r, t_1, \ldots \rangle)$ denotes the final output of $\mathcal{A}$ and ℓ denotes the length of m, let NewBlock denote the event that at least one of the blocks $r \| \ell \| i \| m_i$ was never previously authenticated by the MAC oracle.

Bounding the probability of a forgery

We have

$$\begin{aligned}
\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1] &= \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{Repeat}] \\
&\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \text{NewBlock}] \\
&\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}] \\
&\le \Pr[\mathit{Repeat}] \\
&\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{NewBlock}] \\
&\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}].
\end{aligned}$$

We show that the first two terms are negligible, and the final term is 0. This implies $\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1]$ is negligible, as desired.

First claim

Claim 4.9. There is a negligible function negl with $\Pr[\text{Repeat}] \leq \text{negl}(n)$.

Proof of Claim. Let $q(n)$ be the (polynomial) number of MAC oracle queries made by $\mathcal{A}$. To answer the ith oracle query, the oracle chooses $r_i \leftarrow \{0,1\}^{n/4}$ uniformly at random. The probability of event Repeat is exactly the probability that $r_i = r_j$ for some $i \neq j$. This is the old "birthday bound*."

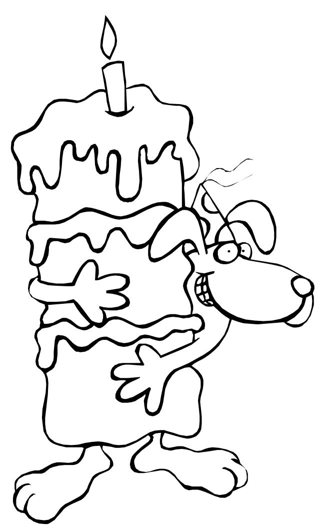
*And what, pray tell, is that?

The birthday problem

How many students do we need in a class before the probability is greater than 1/2 that two students have the same birthday?

Lemma A.16. Fix a positive integer N, and say $q \leq \sqrt{2N}$ elements $y_1, \ldots, y_q$ are chosen uniformly and independently at random from a set of size N. Then the probability that there exists distinct i, j with $y_i = y_j$ is at least $\frac{q(q-1)}{4N}$.

For the birthday problem, $N = 365$, we find the smallest q such that the probability of collision exceeds 1/2.



Here we are interested an upper bound

Lemma A.15. Let $y_1, \ldots, y_q$ be q elements chosen uniformly at random from a set of of size N. The probability that there exists distinct i, j with $y_i = y_j$ is at most $\frac{q^2}{2N}$.

Proof. Let Coll denote the event of a collision, and let $\mathsf{Coll}_{i,j}$ denote the event that $y_i = y_j$. Certainly $\Pr[\mathsf{Coll}_{i,j}] = 1/N$ for $i \neq j$. Since $\mathsf{Coll} = \bigvee_{i \neq j} \mathsf{Coll}_{i,j}$, the union bound implies

$$\begin{aligned} \Pr[\mathsf{Coll}] &= \Pr\left[\bigvee_{i \neq j} \mathsf{Coll}_{i,j}\right] \\ &\leq \sum_{i \neq j} \Pr[\mathsf{Coll}_{i,j}] \\ &= \binom{q}{2} \cdot \frac{1}{N} \leq \frac{q^2}{2N}. \end{aligned}$$

Back to the first claim

Claim 4.9. There is a negligible function negl with $\Pr[\mathsf{Repeat}] \leq \mathsf{negl}(n)$.

Proof of Claim. Let $q(n)$ be the (polynomial) number of MAC oracle queries made by $\mathcal{A}$. To answer the ith oracle query, the oracle chooses $r_i \leftarrow \{0,1\}^{n/4}$ uniformly at random. The probability of event Repeat is exactly the probability that $r_i = r_j$ for some $i \neq j$. By Lemma A.15, we have $\Pr[\mathsf{Repeat}] \leq \frac{q(n)^2}{2 \cdot 2^{n/4}}$.*

*Here we are using the fact that identifiers are chosen from a set of size $|\{0,1\}^{n/4}| = 2^{n/4}$.

On to the next party

Claim. $\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}] = 0$.

Proof. Let $q = q(n)$ denote the number of queries made by $\mathcal{A}$ and r_i denote the random identifier used to answer the ith query. If Repeat does not occur, the the values $r_1, \ldots, r_q$ are all distinct.

Let $(m, \langle r, t_1, \ldots, t_d \rangle)$ be the output of $\mathcal{A}$, with $m = m_1, \ldots$. If $r \notin \{r_1, \ldots, r_q\}$, then NewBlock clearly occurs.

If not, then $r = r_j$ for some unique j, and the blocks $r \parallel \ell \parallel 1 \parallel m_1, \ldots$ could not have been authenticated during the course of answering any query other than the jth. Let $m^{(j)}$ be the message used by $\mathcal{A}$ for its jth query, and let ℓ_j be its length.

There are two cases to consider.

The two cases

Case 1: $\ell \neq \ell_j$. The blocks authenticated when answering the jth query all have $\ell_j \neq \ell$ in the second position. So $r||\ell||1||m_1$ wsa never authenticated in the course of answering the jth query, and NewBlock occurs.

Case 2: $\ell = \ell_j$. If $\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1$, then we must have $m \neq m^{(j)}$. Let $m^{(j)} = m_1^{(j)}, \ldots$. Since m and $m^{(j)}$ have equal length, there must be at least one index i for which $m_i \neq m_i^{(j)}$. the block $r \parallel \ell \parallel i \parallel m_i$ was then never authenticated in the course of answering the jth query.

Finally we show

Claim 4.10. $\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{NewBlock}] = 0.$

The claim relies on the security of Π'. We construct an adversary $\mathcal{A}'$ who attacks the fixed-length MAC Π' and succeeds with probability

$$\Pr[\text{Mac-forge}_{\mathcal{A}',\Pi'}(n) = 1 \geq \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{NewBlock}]$$

Security of Π' implies that the left-hand side is negligible, proving the claim.

A PPT adversary $\mathcal{A}'$ attacking Π'

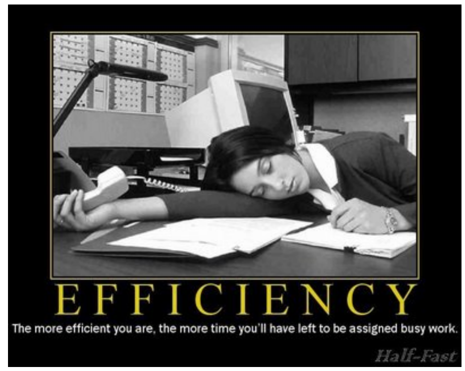
Adversary $\mathcal{A}'$:

1. $\mathcal{A}'$ runs $\mathcal{A}$ as a sub-routine, and answers the request by $\mathcal{A}$'s for a MAC tag on message m by choosing $r \leftarrow \{0,1\}^{n/4}$, parsing m appropriately, and making the appropriate queries to its own MAC oracle.
2. When $\mathcal{A}$ outputs (m, t), with $|m| = \ell$, $\mathcal{A}'$ parses m as $m_1, \ldots, m_d$ and t as $\langle r, t_1, \ldots, t_d \rangle$ and checks for a previously-unauthenticiated block $r\|\ell\|i\|m_i$, i.e, NewBlock occurs. If such a block exists, $\mathcal{A}'$ outputs $(r\|\ell\|i\|m_i, t_i)$. If not, $\mathcal{A}'$ outputs nothing.

The view of $\mathcal{A}$ when run as a sub-routine of $\mathcal{A}'$ is distributed identically to the view of $\mathcal{A}$ in $\text{Mac-forge}_{\mathcal{A},\Pi}(n)$. If Newblock occurs then $\mathcal{A}'$ outputs a block $(r\|\ell\|i\|m_i, t_i)$ that was never previously authenticated; if $\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1$ then the tag on every block is valid, and $\text{Mac-forge}_{\mathcal{A}',\Pi'}(n) = 1$.

CBC-Mac

- The previous construct works, but is rather inefficient: to compute a MAC tag on a message of length $\ell \cdot n$ requires 4ℓ application of the block cipher, and the MAC tag is $(4\ell + 1)n$ bits long.
- There is a better way: CBC-MAC construction is similar to CBC mode encryption and only requires ℓ applications of the block cipher producing a tag of length n bits long.



CBC-MAC for fixed-length messages

Construction 4.11.

Let F be a pseudorandom function, and fix a length function ℓ. The basic CBC-MAC construction is as follows.

- Gen: On input 1^n, choose $k \leftarrow \{0,1\}^n$ uniformly at random.
- Mac: On input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^*$ of length $\ell(n) \cdot n$, do the following:
  1. Parse m as $m_1, \ldots, m_\ell$, where each m_i is of length n.
  2. Set $t_0 := 0^n$. For $i = 1$ to ℓ:
     Set $t_i := F_k(t_{i-1} \oplus m_i)$.

  Output t_ℓ as the tag.
- Vrfy: On input a key $k \in \{0,1\}^n$, message $m \in \{0,1\}^*$ of length $\ell(n) \cdot n$, and a tag t of length n, output 1 if and only if $t \stackrel{?}{=} \mathrm{Mac}_k(m)$.

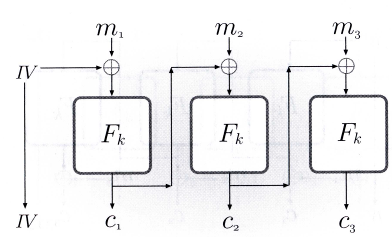
Security of CBC-MAC for fixed-length messages*

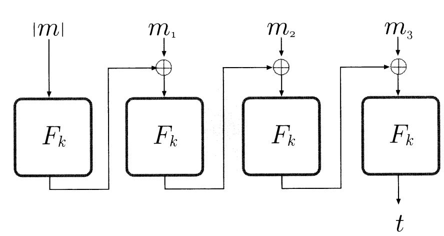
Theorem 4.12. Let ℓ be a polynomial. If F is a pseudorandom function, then Construction 4.11 is a fixed-length MAC for messages of length $\ell(n) \cdot n$ that is existentially unforgeable under an adaptive chosen-message attack.

*If an adversary is able to obtain MAC tags for messages of varying lengths, then the scheme is not longer secure.

Modification of CBC-MAC for fixed-length messages

1. CBC-mode encryption uses a *random IV* and this turned out to be crucial for its security. In contrast, CBC-MAC uses no *IV*, and this is also crucial for obtaining security.
2. In CBC-mode encryption all blocks t_i are output, whereas in CBC-MAC only the final block is output. Why not output all the blocks?





Secure CBC-MAC for variable-length messages

1. Prepend the message with its length $|m|$ and then compute the basic CBC-MAC on the resulting message.
2. Change the scheme so that key generation chooses two different keys $k_1 \leftarrow \{0,1\}^n$ and $k_2 \leftarrow \{0,1\}^n$. Then to authenticate a message m first compute the basic CBC-MAC of m using k_1 and let t be the result; output the tag $\hat{t} := F_{k_2}(t)$.