Technische Universität München (I7)
Dr. M. Luttenberger / M. Schlund

Winter term 2014/2015
Monday 8th December, 2014

Solution

Cryptography – Homework 4

Discussed on Tuesday, 3rd December, 2012.

For questions regarding the exercises, please send an email to schlund@in.tum.de or just drop by at room 03.11.055

Exercise 4.1 MAC or no MAC?

(a) Does rOFB mode yield a secure MAC?

(b) Show that if the IV in the CBC-MAC-Algorithm is not fixed (but chosen randomly and pre-pended to the CBC-output), the MAC becomes insecure.

Solution:

(a) No: Query 0^n and receive $c^{(0)}||c^{(1)}$, then compute $\texttt{tag} = c^{(0)}||\overline{c^{(1)}}$ which is a tag for the message 1^n.

(b) Query a tag for $m_0 = 0^1$ which is padded to $\lfloor 1 \rceil||0^n$, and receive $t_0 = \rho||t$. We will now forge a tag for the message $m_1 = 0^2$: First we compute the "difference" $\delta = \lfloor 1 \rceil \oplus \lfloor 2 \rceil$. Then we add this difference to ρ to get $\rho' = \rho \oplus \delta$ and produce a valid tag for m_1 by $t_1 = \rho'||t$.

Exercise 4.2 Padding done right or wrong

(a) We defined $\mathsf{pad}_{\text{cbc}} : \{0,1\}^+ \to (\{0,1\}^n)^+$ by $\mathsf{pad}_{\text{cbc}}(m) := \lfloor |m| \rceil||m||0^p$, where $|m|$ is encoded as an n-bit string in the first block, and p is minimal s.t. the output length is a multiple of n. We used this padding function for defining the CBC-MAC scheme.

Show that $\mathsf{pad}_{\text{cbc}}$ is prefix-free.

(b) We defined $\mathsf{pad}_{\text{MD}} : \{0,1\}^+ \to (\{0,1\}^n)^+$ by $\mathsf{pad}_{\text{MD}}(m) := m||0^p||\lfloor |m| \rceil$, where $|m|$ is encoded as an n-bit string in the last block, and p is minimal s.t. the output length is a multiple of n.

Show that if we replace $\mathsf{pad}_{\text{cbc}}$ in the CBC-MAC by pad_{MD}, the scheme is no longer secure (assuming a PRF of key and block length n).

Solution:

(a) We have to show that for any two different messages $m_1 \neq m_2$ from $\{0,1\}^*$ we have $\mathsf{pad}_{\text{cbc}}(m_1) \not\leq \mathsf{pad}_{\text{cbc}}(m_2)$ (where $\leq$ denotes the "is prefix"-relation on words). Observe that for any messages $\alpha, \beta \in \{0,1\}^*$ if $\alpha[i] \neq \beta[i]$ for any bit $i \leq \min(|\alpha|, |\beta|)$ then $\alpha \not\leq \beta$.

So let $m_1 \neq m_2$ be any two different messages. We distinguish two cases:

- If $|m_1| = |m_2|$ then the first n bits of $\mathsf{pad}_{\text{cbc}}(m_i)$ $(i = 1, 2)$ are equal, but as $m_1 \neq m_2$ the padding differs in at least one bit. Since both paddings have the same length in this case, one cannot be a prefix of the other.
- If $|m_1| \neq |m_2|$ then $\mathsf{pad}_{\text{cbc}}(m_i)$ $(i = 1, 2)$ differ in one of the first n bits and thus again one cannot be a prefix of the other (both messages are at least n bits long).

(b) We will construct a pair of messages which result in the same tag (a "collision"). First query 0^n and receive $t_0 = F_k(\lfloor n \rceil \oplus F_k(0^n))$. Next query 1^n and receive $t_1 = F_k(\lfloor n \rceil \oplus F_k(1^n))$. Now observe that $0^n \lfloor n \rceil t_0$ will produce the tag $F_k(\lfloor 3n \rceil \oplus 0^n)$ which is the same as the tag of the message $1^n \lfloor n \rceil t_1$. Thus we can forge a tag after three queries.

Exercise 4.3 MACs using hash-functions done wrong

Before NMAC and HMAC, several ad-hoc solutions for constructing MACs were used. For instance, given a (hash) function $H\colon \{0,1\}^* \to \{0,1\}^l$, the tag was defined to be $\mathsf{Mac}_k(m) := H(k||m)$, i.e. the outer encryption used in NMAC and HMAC is missing.

(a) Assume a PRF F with (for simplicity) $n = l_{\text{in}}(n) = l_{\text{out}}(n)$. Using the padding function $\mathsf{pad}(m) := m||10^p||\lfloor |m| \rceil$, set $\mathsf{Mac}_k(m) := H(k||m) := F_k^*(\mathsf{pad}(m))$ for $k \in \{0,1\}^n$.

Show that $\mathsf{Mac}_k(m)$ is not secure.

Hint: Recall that the outer encryption used by NMAC and HMAC is to restrict the adversary to prefix-free queries.

Solution: We first query a tag for 0^{n-1} which is padded to $0^{n-1}1\lfloor n-1 \rceil$ and we receive $k_2 = F_{k_1}(\lfloor n-1 \rceil)$ where $k_1 = F_k(0^{n-1}1)$. Now we can forge a tag for the new message $0^{n-1}1\lfloor n-1 \rceil$ (which will be padded to $0^{n-1}1\lfloor n-1 \rceil 10^{n-1}\lfloor 2n \rceil$) simply by computing $k_3 = F_{k_2}(10^{n-1})$ (we know k_2!!) and then $\mathtt{tag} = F_{k_3}(\lfloor 2n \rceil)$. It is important to remember that F_k is easily computable if we know the key k and that the *insecure* "cascading F" construction without the "sealing-off" outer encryption can leak the intermediate keys and therefore allows us to continue the computation.

Remark: The construction can be made secure by using a prefix-free padding (see lecture slides for a reference that the cascading construction is secure if no prefix-queries are allowed)!

However, for practical purposes it is often inefficient to scan the whole message first to get its length and prepend it in the padding and therefore the MD-padding is used instead!

Exercise 4.4

Let F be some secure block cipher with key and block length n (think of AES-128).

Consider the following deterministic MAC:

- Gen: as usual, in input 1^n, output $k \overset{u}{\in} \{0,1\}^n$.
- Mac: given $m \in \{0,1\}^+$ and k,

  first pad m to a multiple of n by appending a minimial number of 0,

  then break the padded message into n-bit blocks $m^{(i)}$.

  Starting with $k^{(0)} := 0^n$, compute $k^{(i)} = F_{k^{(i-1)}}(m^{(i)})$ for i from 1 to n.

  Finally, output $t := F_{k^{(n)}}(k)$.

  (Draw a picture! Note that the key is appended in this case.)
- Vrf: given m, t, and k, check that $\mathsf{Mac}_k(m) = t$.

Is this MAC secure?

Solution: This MAC is not secure: Query 0^n and receive $t_0 = F_{F_{0^n}(0^n)}(k)$. We can simply compute $F_{0^n}(0^n)$ and thus obtain $F^{-1}_{F_{0^n}(0^n)}(t_0) = k$.

Note that $F_k(x)$ is of course efficiently computable if we know the key k!