MTAT.07.003 Cryptology II
Spring 2012 / Exercise session ?? / Example Solution

Exercise (HASH + MAC = MAC). *Let $\mathcal{H}$ be a collision resistant hash function family form $\mathcal{M}$ to $\mathcal{X}$ and let $\mathsf{Mac} : \mathcal{X} \times \mathcal{K} \to \mathcal{T}$ be a secure message authentication code. Show that the following function*

$$\mathsf{HashMac}(m, k, h) = \mathsf{Mac}(h(m), k)$$

is secure message message authentication code with a signature $\mathsf{HashMac} : \mathcal{M} \times \mathcal{K} \times \mathcal{H} \to \mathcal{T}$, i.e., the usage of collision resistant functions allows us to extend the domain of a message authentication code.

Solution. Recall that according to the security definition for message authentication we must show that the probability that a t-time adversary $\mathcal{A}$ wins the following game

$$\begin{array}{l}
\mathcal{G}^{\mathcal{A}} \\
\left[\begin{array}{l}
k \leftarrow \mathcal{K} \\
h \leftarrow \mathcal{H} \\
t_0 \leftarrow \mathcal{A}(h) \\
\quad \text{For } i \in \{1, \ldots, q\} \text{ do} \\
\quad \left[\begin{array}{l} m_i \leftarrow \mathcal{A}(t_{i-1}) \\ t_i \leftarrow \mathsf{HashMac}(m_i, k, h) \end{array}\right. \\
(m, t) \leftarrow \mathcal{A}(t_q) \\
\mathsf{if}\ (m, t) \in \{(m_1, t_1), \ldots, (m_q, t_q)\}\ \textbf{return } 0 \\
\textbf{return } [t \stackrel{?}{=} \mathsf{HashMac}(m, k, h)]
\end{array}\right.
\end{array}$$

is bounded from above. By substituting the definition of $\mathsf{HashMac}$ into the game, we obtain

$$\begin{array}{l}
\mathcal{G}_0^{\mathcal{A}} \\
\left[\begin{array}{l}
k \leftarrow \mathcal{K} \\
h \leftarrow \mathcal{H} \\
t_0 \leftarrow \mathcal{A}(h) \\
\quad \text{For } i \in \{1, \ldots, q\} \text{ do} \\
\quad \left[\begin{array}{l} m_i \leftarrow \mathcal{A}(t_{i-1}) \\ x_i \leftarrow h(m_i) \\ t_i \leftarrow \mathsf{Mac}(x_i, k) \end{array}\right. \\
(m, t) \leftarrow \mathcal{A}(t_q) \\
\mathsf{if}\ (m, t) \in \{(m_1, t_1), \ldots, (m_q, t_q)\}\ \textbf{return } 0 \\
x \leftarrow h(m) \\
\textbf{return } [t \stackrel{?}{=} \mathsf{Mac}(x, k)]\ .
\end{array}\right.
\end{array}$$

Note that $\mathcal{A}$ wins the game, if $\mathcal{A}$ creates m such that $h(m) = h(m_i)$ while $m \neq m_i$. Then t_i is a known and valid message authentication tag for m. To handle this issue explicitly, we can define the following games:

$$\begin{array}{l}\mathcal{G}_1^{\mathcal{A}}\\ \left[\begin{array}{l} k \leftarrow \mathcal{K}\\ h \leftarrow \mathcal{H}\\ t_0 \leftarrow \mathcal{A}(h)\\ \quad\text{For } i \in \{1,\ldots,q\} \text{ do}\\ \quad\left[\begin{array}{l} m_i \leftarrow \mathcal{A}(t_{i-1})\\ x_i \leftarrow h(m_i)\\ t_i \leftarrow \mathsf{Mac}(x_i, k)\end{array}\right.\\ (m,t) \leftarrow \mathcal{A}(t_q)\\ \mathsf{if}\ [h(m) \notin \{h(m_1),\ldots,h(m_q)\}]\ \mathbf{return}\ 0\\ \mathsf{if}\ m \in \{m_1,\ldots,m_q\}\ \mathbf{return}\ 0\\ \mathbf{return}\ [t \stackrel{?}{=} \mathsf{Mac}(h(m),k)]\end{array}\right.\end{array} \qquad \begin{array}{l}\mathcal{G}_2^{\mathcal{A}}\\ \left[\begin{array}{l} k \leftarrow \mathcal{K}\\ h \leftarrow \mathcal{H}\\ t_0 \leftarrow \mathcal{A}(h)\\ \quad\text{For } i \in \{1,\ldots,q\} \text{ do}\\ \quad\left[\begin{array}{l} m_i \leftarrow \mathcal{A}(t_{i-1})\\ x_i \leftarrow h(m_i)\\ t_i \leftarrow \mathsf{Mac}(x_i, k)\end{array}\right.\\ (m,t) \leftarrow \mathcal{A}(t_q)\\ \mathsf{if}\ h(m) \in \{h(m_1),\ldots,h(m_q)\}\ \mathbf{return}\ 0\\ \mathbf{return}\ [t \stackrel{?}{=} \mathsf{Mac}(h(m),k)]\ .\end{array}\right.\end{array}$$

Clearly, we can split all runs of $\mathcal{G}^{\mathcal{A}}$ into two classes depending whether the event $h(m) \notin \{h(m_1),\ldots,h(m_q)\}$ holds or not. As the event $h(m) \notin \{h(m_1),\ldots,h(m_q)\}$ also implies $m \notin \{m_1,\ldots,m_q\}$, we do not have to check the condition $(m,t) \in \{(m_1,t_1),\ldots,(m_q,t_q)\}$ any more in $\mathcal{G}_2$. For the game $\mathcal{G}_1$, we still have to check that $m \notin \{m_1,\ldots,m_q\}$. Thus, by the construction of games we have established

$$\Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1\right] = \Pr\left[\mathcal{G}_1^{\mathcal{A}} = 1\right] + \Pr\left[\mathcal{G}_2^{\mathcal{A}} = 1\right] \ .$$

The game $\mathcal{G}_2$ is very close to the security game for the message authentication codes. In fact, if we define an adversary $\mathcal{B}$ such that

$$\begin{array}{l}\mathcal{B}(t_0)\\ \left[\begin{array}{l} h \xleftarrow{u} \mathcal{H}\\ m_1 \leftarrow \mathcal{A}(t_0)\\ \mathbf{return}\ h(m_1)\end{array}\right.\end{array} \qquad \begin{array}{l}\mathcal{B}(t_{i-1})\\ \left[\begin{array}{l} m_i \leftarrow \mathcal{A}(t_{i-1})\\ \mathbf{return}\ h(m_i)\end{array}\right.\end{array} \qquad \begin{array}{l}\mathcal{B}(t_q)\\ \left[\begin{array}{l} (m,t) \leftarrow \mathcal{A}(t_0)\\ \mathbf{return}\ h(m_1), t\end{array}\right.\end{array}$$

then direct substitution to the security game of message authentication code

$$\begin{array}{l}\mathcal{Q}^{\mathcal{B}}\\ \left[\begin{array}{l} k \leftarrow \mathcal{K}\\ \quad\text{For } i \in \{1,\ldots,q\} \text{ do}\\ \quad\left[\begin{array}{l} x_i \leftarrow \mathcal{B}(t_{i-1})\\ t_i \leftarrow \mathsf{Mac}(x_i, k)\end{array}\right.\\ (x,t) \leftarrow \mathcal{A}(t_q)\\ \mathsf{if}\ (x,t) \in \{(x_1,t_1),\ldots,(x_q,t_q)\}\ \mathbf{return}\ 0\\ \mathbf{return}\ [t \stackrel{?}{=} \mathsf{Mac}(x,k)]\end{array}\right.\end{array}$$

leads to the game that is equivalent to the game $\mathcal{G}_2^{\mathcal{A}}$. The only difference after the mechanical substitution is in the last check. In the game $\mathcal{G}_2^{\mathcal{A}}$, the check

$$h(m) \in \{h(m_1),\ldots,h(m_q))\}$$

is more stringent than the check $(h(m),t) \in \{(h(m_1),t_1),\ldots,(h(m_q),t_q)\}$ used in $\mathcal{Q}^{\mathcal{B}}$. Consequently,

$$\Pr\left[\mathcal{G}_2^{\mathcal{A}} = 1\right] \le \Pr\left[\mathcal{Q}^{\mathcal{B}} = 1\right] \le \mathsf{Adv}^{\mathsf{mac}}_{\mathsf{Mac}}(\mathcal{B}) \ .$$

Note that the overhead in the running time of $\mathcal{B}$ is linear in the number of queries q and thus $(t + \mathrm{O}(q), \varepsilon_1)$-secure message authentication code is sufficient for bounding the success probability in the game $\mathcal{G}_2$.

For the game $\mathcal{G}_1$, it is important to note that $\mathcal{A}$ passes first two checks only if $\mathcal{A}$ creates a hash collision: $h(m) = h(m_i)$ for $m \neq m_i$. Consequently, the following adversary

$$
\begin{array}{l}
\mathcal{C}(h)\\
\left[\begin{array}{l}
k \leftarrow \mathcal{K}\\
t_0 \leftarrow \mathcal{A}(h)\\
\ \text{For } i \in \{1, \ldots, q\} \text{ do}\\
\ \left[\begin{array}{l}
m_i \leftarrow \mathcal{A}(t_{i-1})\\
x_i \leftarrow h(m_i)\\
t_i \leftarrow \mathsf{Mac}(x_i, k)
\end{array}\right.\\
(m, t) \leftarrow \mathcal{A}(t_q)\\
\mathsf{if}\ [h(m) \notin \{h(m_1), \ldots, h(m_q)\}]\ \mathbf{return}\ 0\\
i \leftarrow \{i : h(m_i) = h(m)\}\\
\mathbf{return}\ (m, m_i)
\end{array}\right.
\end{array}
$$

can be used for the collision resistance game

$$
\begin{array}{l}
\mathcal{Q}^{\mathcal{C}}\\
\left[\begin{array}{l}
h \xleftarrow[u]{} \mathcal{H}\\
(m_0, m_1) \leftarrow \mathcal{B}(h)\\
\mathbf{return}\ [h(m_0) = h(m_1)] \wedge [m_0 \neq m_1]\ .
\end{array}\right.
\end{array}
$$

Again, the success criterion in the game $\mathcal{Q}$ is more relaxed than in the game $\mathcal{G}_1$ and thus direct substitution allows us to prove:

$$
\Pr\left[\mathcal{G}_1^{\mathcal{A}} = 1\right] \leq \Pr\left[\mathcal{Q}^{\mathcal{C}} = 1\right] \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H}}(\mathcal{C})\ .
$$

Again, the overhead in the running time of $\mathcal{C}$ is $\mathrm{O}(q)$. Thus, usage of $(t + \mathrm{O}(q), \varepsilon_2)$-collision resistant hash function family $\mathcal{H}$ is sufficient for bounding the success probability in the game $\mathcal{G}_1$.

To summarise, we have proven that $\mathsf{HashMac}$ is $(t, q, \varepsilon_1 + \varepsilon_2)$-secure message authentication code provided that $\mathcal{H}$ is a $(t + \mathrm{O}(q), \varepsilon_2)$-collision resistant hash function family and Mac is $(t + \mathrm{O}(q), \varepsilon_1)$-secure message authentication code.

On the optimality of bounds. It is easy to see that $\mathcal{A}$ can win $\mathcal{G}_1$ as soon as it produces a hash collision $h(m) = h(m_i)$, since $\mathcal{A}$ can set $t = t_i$ and pass the last check, as well. Most message authentication codes are deterministic and thus the conditions

$$
h(m) \in \{h(m_1), \ldots, h(m_q))\} \quad \text{and} \quad (h(m), t) \in \{(h(m_1), t_1), \ldots, (h(m_q), t_q)\}
$$

are equivalent. The latter implies that also the second bound is optimal.