CS 435: Introduction to Cryptography **Spring 2020**

Homework 5 Solutions

Professor Somesh Jha **Due: April 21**

1. **Exercise 4.8**

   Let F be a pseudorandom function. Show that the following MAC for messages of length $2n$ is insecure: Gen outputs a uniform $k \in \{0,1\}^n$. To authenticate a message $m_1 \| m_2$ with $|m_1| = |m_2| = n$, compute the tag $F_k(m_1) \| F_k(F_k(m_2))$.

   Solution:

   Let $\mathcal{A}$ be an adversary that queries its oracle with two messages $m = m_0||m_1$ and $m' = m'_0||m'_1$, where $m_0 \neq m'_0$ and $m_1 \neq m'_1$. Let $t = t_0||t_1$ and $t' = t'_0||t'_1$ be the respective responses from its oracle. $\mathcal{A}$ then outputs the message $\tilde{m} = m_0||m'_1$ and tag $\tilde{t} = t_0||t'_1$. By the definition of Mac, it follows that $\tilde{t}$ is a correct tag for $\tilde{m}$ and thus $\mathsf{Vrfy}_k(\tilde{m}, \tilde{t}) = 1$ always. Furthermore, since $m_0 \neq m'_0$ and $m_1 \neq m'_1$ we have that $\tilde{m} \notin \mathcal{Q}$. Thus $\mathcal{A}$ succeeds with probability 1 and the scheme is not secure.

2. **Exercise 4.1**

   Say $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ is a secure MAC, and for $k \in \{0,1\}^n$ the tag-generation algorithm Mac_k always outputs tags of length $t(n)$. Prove that t must be super-logarithmic or, equivalently, that if $t(n) = \mathcal{O}(\log n)$ then Π cannot be a secure MAC.

   Solution:

   Assume that $t(n) = c \log n$ for some constant c. Then, consider an adversary $\mathcal{A}$ who upon input 1^n just outputs an arbitrary m and a uniform $t \in \{0,1\}^{t(n)}$. Adversary $\mathcal{A}$ succeeds with probability at least $2^{-t(n)}$ since there must be *some* valid tag for m (note also that $m \notin \mathcal{Q}$ always for this $\mathcal{A}$). Since $t(n) = c \log n$ we have that $2^{-t(n)} = n^{-c}$ which is not negligible.

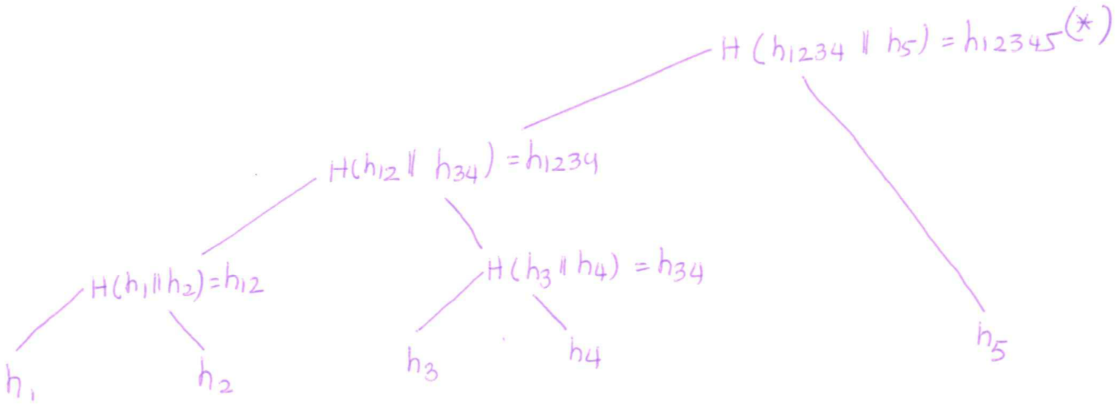
3. Alice has five files F_1, F_2, F_3, F_4, F_5 that she wants to store on Bob's computer (Bob just purchased a new server that has a gigantic hard disk). However, Alice is worried that Bob might corrupt or modify the files. Answer the following:

   (a) Show the Merkle hash tree for F_1, F_2, F_3, F_4, F_5.

   (b) What is stored on Alice's computer?

   Solution:

   (a) $h_1 = H(F_1)$, $h_2 = H(F_2)$, $h_3 = H(F_3), h_4 = H(F_4), h_5 = H(F_5)$.



(b) Alice stores the root hash (h_{12345}) on her computer.

4. Now Alice wants to retrieve file F_3 from Bob's computer.

   (a) What does Bob send to Alice? Recall that Bob needs to "prove" to Alice that the file has not been modified.

   (b) Show that it is "hard" for Bob to generate a "proof" for Alice for a file F_3' different from F_3. We of course assume that hash functions that the Merkle hash tree is constructed from is *collision resistant*.

Solution:

(a) Bob sends the file F_3' and hashes (h_4', h_{12}', h_5'). Alice computes

$$\begin{aligned} h_3' &= H(F_3') \\ h_{34}' &= H(h_3'||h_4') \\ h_{1234}' &= H(h_{12}'||h_{34}') \\ h_{12345}' &= H(h_{1234}'||h_5'), \end{aligned}$$

and then checks if $h_{12345}' = h_{12345}$. The latter is stored on Alice's computer.

(b) Suppose Alice's file was F_3. Bob gives a proof $(F_3', h_4', h_{12}', h_5')$ such that $F_3 \neq F_3'$. We prove this is not possible with high probability. Throughout, not possible means not possible with high probability.

- $h_3' = H(F_3') = h_3 = H(F_3)$. Not possible if H is collision resistant as $F_3 \neq F_3'$.
- $h_3' \neq h_3$, but $h_{34}' = H(h_3'||h_4') = h_{34} = H(h_3||h_4)$. Again, not possible because $h_3 \neq h_3'$ and H is collision resistant.
- $h_3' \neq h_3, h_{34}' \neq h_{34}$, but $h_{1234}' = h_{1234}$. Not possible reasoning as before.
- $h_3' \neq h_3, h_{34}' \neq h_{34}, h_{1234}' \neq h_{1234}$, but $h_{12345}' = h_{12345}$. Not possible.