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

Homework 5

Professor Somesh Jha **Due:** Nov 14

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 $F_k(m_1) \| F_k(F_k(m_2))$.

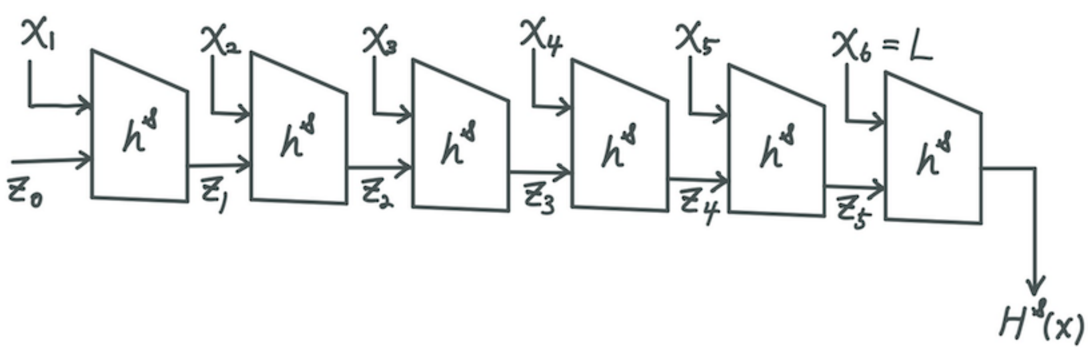
   Solution:

   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. In LectureLet 21, we did the Merkle-Damgrad construction which took a hash function h that compresses by a factor of 1/2 and constructs a hash function H by a factor of 1/3. Repeat the construction to construct a hash function H that compresses by a factor of 1/5. Also redo the proof of collision resistance that was done in the LectureLet.

   Solution:

   Let x be a string of length L and assume the compression function (Gen, h) compresses its input by half. We can construct a collision-resistant hash function (Gen, H) that maps inputs of length $5n$ to outputs of length n as follows: let $x_1, x_2, .., x_5$ be the five blocks of x, then



   Based on the assumption that h^s is collision-resistant, we now prove that the constructed H^s is also collision-resistant. Proof by contradiction, assume $H^s(x) = H^s(x')$ for $x \neq x'$ where $x = x_1x_2x_3x_4x_5L$ and $x' = x'_1x'_2x'_3x'_4x'_5L'$.

- If $L \neq L'$, $h^s(L||z_5) = h^s(L'||z_5')$ for $L||z_5 \neq L'||z_5'$ is not possible because h^s is collision-resistant.
- If $L = L'$ and $z_5 \neq z_5'$, $h^s(L||z_5) = h^s(L'||z_5')$ is not possible because $L||z_5 \neq L'||z_5'$ and h^s is collision-resistant.
- If $L = L', z_5 = z_5'$ and $x_5 \neq x_5'$, $z_5 = h^s(x_5||z_4) = h^s(x_5'||z_4') = z_5'$ is not possible because $x_5||z_4 \neq x_5'||z_4'$ and h^s is collision-resistant.
- If $L = L', z_5 = z_5', x_5 = x_5'$ and $z_4 \neq z_4'$, $z_5 = h^s(x_5||z_4) = h^s(x_5'||z_4') = z_5'$ is not possible because $x_5||z_4 \neq x_5'||z_4'$ and h^s is collision-resistant.
- ... (repeat down to $z_4, x_4, z_3, x_3, z_2, x_2, z_1$)
- If $L = L', z_5 = z_5', x_5 = x_5', z_4 = z_4', x_4 = x_4', z_3 = z_3', x_3 = x_3', z_2 = z_2', x_2 = x_2', z_1 = z_1'$ and $x_1 \neq x_1'$, $z_1 = h(x_1||z_0) = h(x_1'||z_0') = z_1'$ is not possible because $x_1||z_0 \neq x_1'||z_0'$ and h^s is collision-resistant.
- If $L = L', z_5 = z_5', x_5 = x_5', z_4 = z_4', x_4 = x_4', z_3 = z_3', x_3 = x_3', z_2 = z_2', x_2 = x_2', z_1 = z_1', x_1 = x_1'$ and $z_0 \neq z_0'$, $z_1 = h(x_1||z_0) = h(x_1'||z_0') = z_1'$ is not possible because $x_1||z_0 \neq x_1'||z_0'$ and h^s is collision-resistant.

To have no collision, we should have $L = L', x_5 = x_5', x_4 = x_4', x_3 = x_3', x_2 = x_2', x_1 = x_1'$, which contradicts the assumption $x \neq x'$. Therefore, $H^s(x) \neq H^s(x')$ for two different x and x', and H^s is collision-resistant.

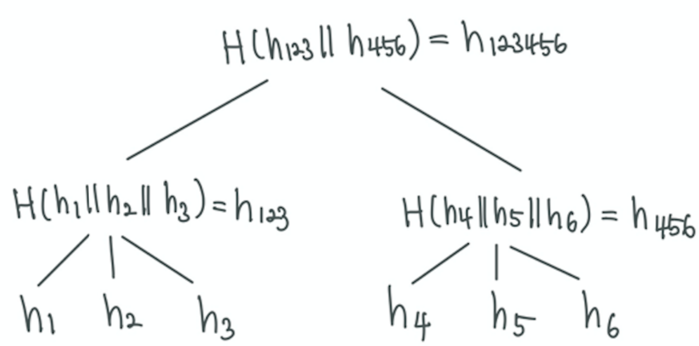
3. Alice has six files $F_1, F_2, F_3, F_4, F_5, F_6$ that she wants to store on Bob's computer (Bob just purchased a new server that has a gigantic hard disk). However, Alice is worried that Bob might corrupt or modify the files. Answer the following:

   (a) Show a Merkle hash tree for $F_1, F_2, F_3, F_4, F_5, F_6$ where the root is binary and the internal nodes are ternary. This shows that Merkle hash tree doesn't necessarily have to be binary.

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

Solution:

(a) $h_1 = H(F_1)$, $h_2 = H(F_2)$, $h_3 = H(F_3), h_4 = H(F_4), h_5 = H(F_5), h_6 = H(F_6)$.



(b) Alice stores the root hash (h_{123456}) 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_1', h_2', h_{456}'). Alice computes

$$
\begin{aligned}
h_3' &= H(F_3') \\
h_{123}' &= H(h_1'||h_2'||h_3') \\
h_{123456}' &= H(h_{123}'||h_{456}')
\end{aligned}
$$

   and then checks if $h_{123456}' = h_{123456}$, where h_{123456} was stored on Alice's computer.

   (b) Suppose Alice's file was F_3. Bob gives a proof $(F_3', h_1', h_2', h_{456}')$ 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 $h_{123}' = H(h_1'||h_2'||h_3') = h_{123} = H(h_1||h_2||h_3)$. Again, not possible because $h_3 \neq h_3'$ and H is collision resistant.
   - $h_3' \neq h_3, h_{123}' \neq h_{123}$, but $h_{123456}' = H(h_{123}'||h_{456}') = h_{123456} = H(h_{123}||h_{456})$. Not possible because $h_{123} \neq h_{123}'$ and H is collision resistant.

   Therefore, Bob cannot provide a proof to Alice for a mutated file F_3' such that $F_3' \neq F_3$ with high probability.