Solutions to some homework exercises

Homework 7

Question 2:

The security definition is pretty much the same as in the fixed length case: The adversary is granted access to an authentication oracle, this time for arbitrary length messages. She is then asked to produce a forgery, i.e. a message-tag-pair that has not been previously output by the oracle. Note that the message length of the forgery is allowed to be different from any of the messages sent to the oracle.

The attack works as follows.

- Query some m_1, $|m_1| = n$, the oracle outputs (m_1, t).
- The forgery is $(m^*, t^*) = (m_1 \| m_1 \oplus t, t)$

The attack works because $m_1 \| m_1 \oplus t \neq m_1$ and $(m_1 \| m_1 \oplus t, t)$ is a valid message-tag pair. To be more precise $\mathrm{MAC}_k(m*) = F_k(F_k(m_1) \oplus (m_1 \oplus t)) = F_k(t \oplus m_1 \oplus t) = F_k(m_1) = t$.

Question 3:

The attack works as follows.

- Query $m_1 \| m_2 \| m_3$, where all m_i are distinct and $|m_i| = n$, the oracle outputs $(m_1 \| m_2 \| m_3, t)$.
- The forgery is $(m^*, t^*) = (m_1 \| m_3 \| m_2, t)$

The attack works because $m_1 \| m_3 \| m_2 \neq m_1 \| m_2 \| m_3$, m_i are distinct, and $\mathrm{MAC}_k(m_1 \| m_3 \| m_2) = F_k(m_1) \oplus F_k(m_3) \oplus F_k(m_2) = F_k(m_1) \oplus F_k(m_2) \oplus F_k(m_3) = t$.

Question 4:

The attack works as follows.

- Query $m_1 \| m_2 \| m_3$ and $m_4 \| m_5 \| m_6$ where all m_i are distinct and $|m_i| = n$, the oracle outputs $(m_1 \| m_2 \| m_3, t_1)$ and $(m_4 \| m_5 \| m_6, t_2)$.
- Query $m_1 \| m_2 \| m_6$, the oracle outputs $(m_1 \| m_2 \| m_6, t_3)$.
- The forgery is $(m^*, t^*) = (m_4 \| m_5 \| m_3, t_1 \oplus t_2 \oplus t_3)$

The attack works because $m_4 \| m_5 \| m_3 \notin \{m_1 \| m_2 \| m_3, m_4 \| m_5 \| m_6, m_1 \| m_2 \| m_6\}$, and

$$\begin{aligned}\mathrm{MAC}_k(m^*) =& F_k(\langle 1 \rangle \| m_4) \oplus F_k(\langle 2 \rangle \| m_5) \oplus F_k(\langle 3 \rangle \| m_3) \\ =& \underbrace{F_k(\langle 1 \rangle \| m_4) \oplus F_k(\langle 2 \rangle \| m_5) \oplus F_k(\langle 3 \rangle \| m_6)}_{t_2} \oplus \underbrace{F_k(\langle 3 \rangle \| m_6) \oplus F_k(\langle 3 \rangle \| m_3)}_{t_1 \oplus t_3}.\end{aligned}$$

Question 5:

The attack works as follows.

- Query $m_1 \| m_2$ and $m_3 \| m_4$ where all m_i are distinct and $|m_i| = n$, the oracle outputs $(m_1 \| m_2, t_1 \| t_2)$ and $(m_3 \| m_4, t_3 \| t_4)$.
- The forgery is $(m^*, t^*) = (m_1 \| m_4, t_1 \| t_4)$

The attack works because $m_1 \| m_4 \notin \{m_1 \| m_2, m_3 \| m_4\}$, and $t_1 = F_k(m_1)$, $t_4 = F_k(F_k(m_4))$, so finally we have that $\mathrm{MAC}_k(m^*) = F_k(m_1) \| F_k(F_k(m_4)) = t_1 \| t_4$.

Homework 8

Question 2

If you have a collision, i.e. strings x and x' such that $H'(x) = H'(x')$, than you have a collision for H. A correct solution for this problem should be a formal reduction i.e. it starts by assuming you have an adversary $\mathcal{A}$ that finds a collision for H' and then describing what the adversary $\mathcal{A}'$ against H does that uses $\mathcal{A}$ as a subroutine.

Question 3

The adversary can query some message m to get a tag t, and then output $(m', t \oplus H(m) \oplus H(m'))$.

Question 4

Here we can use the result from Question 2 for the function f that, on input $m = m_1\ m_2 \| m_3 ...$ with $|m_i| = n$ outputs m_3 (if the input is long enough, otherwise it outputs 0^n). That way a successful adversary is the following. Send m to the oracle to get a tag t and outputs (m', t') where m' and t' are obtained from m and t by flipping the first bit.

Question 6

The adversary can flip the last bit of the challenge ciphertext and send it to the decryption oracle.

Homework 9

Question 4:

First we need to factorize $851 = 23 \cdot 37$. Using the fact that if $\gcd(a, b) = 1$, then $\phi(a \cdot b) = \phi(a) \cdot \phi(b)$, we get $\phi(851) = \phi(23 \cdot 37) = \phi(23) \cdot \phi(37)$. As factors are prime numbers we can use the fact that for p-prime $\phi(p) = p - 1$. Finally $\phi(851) = (23 - 1) \cdot (37 - 1) = 792$.

Question 5:

A generator of a group $\mathbb{G}$ is a group element g for which $\langle g \rangle = \mathbb{G}$, where $\langle g \rangle := \{g^0, g^1, g^2, \dots\}$. Note that for some power i: $g^i = 1 = g^0$, so $\langle g \rangle$ is of size at most $|\mathbb{G}|$. Exponentiation is just performing the group operation multiple times, in our case of $\mathbb{Z}_{11}^*$ the group operation is multiplication modulo 11. The subsets generated by elements of $\mathbb{Z}_{11}^*$ are

$$\begin{aligned}
\langle 1 \rangle &= \{1\} \\
\langle 2 \rangle &= \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\} \\
\langle 3 \rangle &= \{1, 3, 9, 5, 4\} \\
\langle 4 \rangle &= \{1, 4, 5, 9, 3\} \\
\langle 5 \rangle &= \{1, 5, 3, 4, 9\} \\
\langle 6 \rangle &= \{1, 6, 3, 7, 9, 10, 5, 8, 4, 2\} \\
\langle 7 \rangle &= \{1, 7, 5, 2, 3, 10, 4, 6, 9, 8\} \\
\langle 8 \rangle &= \{1, 8, 9, 6, 4, 10, 3, 2, 5, 7\} \\
\langle 9 \rangle &= \{1, 9, 4, 3, 5\} \\
\langle 10 \rangle &= \{1, 10\}.
\end{aligned}$$

The elements which generate the whole $\mathbb{Z}_{11}^*$ are then 2, 6, 7, and 8.

Homework 10

Question 1

a) This does not work, the adversary can forward the question and the answer.

b) This doesn't work, for the same reason as a)

c) This works. The transcript is different for different choices of the randomness that is necessary for any key exchange protocol, so the transcripts of different runs are different with high probability. That way, the honest parties will detect whether they participated in the same run of the key exchange protocol.

d) This is insecure. They will detect a man in the middle with high probability, in general, the hash will, however, leak some information about the key.

e) This works. It has been, for example, employed by an earlyier version of Signal for detecting possible man-in-the-middle attacks in voice calls.

Question 2

Encrypt the message with G's public key, then encrypt the ciphertext along with instructions what to do with Fs public key and send it to F.