Homework 3

Due date: Thursday , March 4nd

Question 1 (25 points)

This is a programming assignment on Gradescope. Suppose we adopt the following padding scheme to pad $k < 16$ byte message $m = m_1 \dots m_k$ to 16 bytes. $\texttt{PAD}(m) = \langle 16-k \rangle^{16-k} || m_1 \dots m_k$ where $\langle 16-k \rangle$ is the encoding of the hexadecimal digit $16-k$ as a single byte and $\langle 16-k \rangle^{16-k}$ denotes that byte repeated $16-k$ times. (In Python the command $\texttt{bytes}([i])*j$ would output j copies of the byte encoding i where $0 \le i \le 255$. Thus, to produce $\langle 16-k \rangle^{16-k}$ we would set $i = j = 16-k$ and run $\texttt{bytes}([i]) * j$. Obviously, "||" is the concatenation operation. The encryption algorithm takes as input a $k < 16$-byte message $m = m_1 \dots m_k$ and outputs $\texttt{Enc}_K(m) = (r, F_K(r \oplus \texttt{PAD}(m))$ where $F_K : \{0,1\}^{128} \to \{0,1\}^{128}$ is a PRF and the nonce $r \in \{0,1\}^{128}$ is selected uniformly at random.

In the programming assignment (Python) you will be given a ciphertext $(r, F_K(r \oplus \texttt{PAD}(m))$ encrypting an unknown message m and API access to partial decryption oracle $\texttt{PDec}_K(r, s)$ which computes $m' = r \oplus F_K^{-1}(s)$ and outputs true if m' is properly padded; otherwise, false. You will need to implement a function `RecoverMessage` which takes as input a ciphertext (r, s) and recovers the underlying message m after making oracle calls to $\texttt{PDec}_K(r, s)$. This question will be autograded with some public test cases and some secret test cases.

Your turn-in should include one .py files with name: `attack.py`. We also provide a sample copy of `oracle.py` which will be used to test your solution. You should not modify or submit this file as we will be evaluating your solution with different test cases. We provide template for the these two files and are available for your usage. Your tasks is to complete the functions that we specified. Finally, you can submit them on gradescope by uploading your `attack.py` file there.

Resource and Collaborator Statement: ...

Question 2 (20 points)

Before HMAC, it was common to define a MAC of arbitrary-length message as $\mathsf{Mac}_{\mathsf{s},\mathsf{k}}(m) = H^s(k||m)$ where H is a collision-resistant hash function. We assume s is known to the attacker, and k is kept secret.

- (5 points) Suppose that H is constructed using the Merkle-Damgård transform with the underyling hash function $h^s : \{0,1\}^{2n} \to \{0,1\}^n$. Let $m \in \{0,1\}^n$ be an arbitrary n bit message and let $t = \mathsf{Mac}_{\mathsf{s},\mathsf{k}}(m)$ be the authentication tag for m. Generate an authentication tag t' for the message $m' = m||\langle 2n \rangle$ where $\langle i \rangle$ denote an n-bit encoding of integer i.

- (15 points) Prove that this is a secure MAC if H is modeled as a random oracle.

Answer: ...

Resource and Collaborator Statement: ...

Question 3 (25 points)

One way to build a Pseudorandom Permutation from a pseudorandom function is to use a Feistel Network. In particular, if we select k different PRF keys $K_1, K_2, ..., K_k$ we can define the Pseudorandom Permutation $PRP_{K_1,K_2,...,K_k}(L_0, R_0) = (L_k, R_k)$ where for each $0 \leq i < k$ we have $L_{i+1} = R_i$ and $R_{i+1} = L_i \oplus F_{K_{i+1}}(R_i)$.

It has been shown that if F_K is a secure PRF and we use a $k = 4$ round Feistel network that the permutation PRP_{K_1,K_2,K_3,K_4} is a strong pseudorandom permutation. When $k = 3$ it is known that PRP_{K_1,K_2,K_3} is a pseudorandom permutation, but not a *strong* pseudorandom permutation. **Recall:** A strong PRP means that no PPT attacker can distinguish PRP_{K_1,K_2,K_3} from a truly random permutation f when given oracle access to *both* the permutation (either PRP_{K_1,K_2,K_3} or $f()$) AND its inverse (either $PRP^{-1}_{K_1,K_2,K_3}$ or $f^{-1}()$). In the security game for a regular PRP the distinguisher is not given oracle access to the inverse permutation.

1. (2 points) Show that when $k = 1$ the function is not a regular PRP. You should explain what the distinguisher does and show that its advantage is non-negligible.

2. (5 points) Show that when $k = 2$ the function is not a regular PRP. You should explain what the distinguisher does and show that its advantage is non-negligible.

3. (10 points) We will show that when $k = 3$ the function is not a strong PRP. Consider a distinguisher that makes two queries to the permutation g (either PRP_{K_1,K_2,K_3} or $f()$) and one query to g^{-1}. The first two queries to $g()$ are as follows $g(L_0, R_0)$ and $g(L'_0, R'_0)$ where $R_0 = R'_0$ but $L'_0 \neq L_0$. Let (L_3, R_3) and (L'_3, R'_3) denote the outputs of both queries. Finally, consider the query $g^{-1}(L'_3, R'_3 \oplus L_0 \oplus L'_0)$ and let (L''_0, R''_0) denote the output of this query. Supposing that $g = PRP_{K_1,K_2,K_3}$ is the Feistel Network defined above write down a formula for R''_0 in terms of variables known to the distinguisher. **Note:** Your formula should only use variables that are known to the distinguisher such as L_0, L'_0, R_0, R'_0 or L_3, L'_3, R_3, R'_3. By contrast, your formula should not involve the secret keys K_1, K_2, K_3 or internal values (e.g., R'_2) that would not be known to the distinguisher.

4. (5 points): Supposing that $g = f$ is a truly random permutation and letting (L''_0, R''_0) denote the output of the query $g^{-1}(L'_3, R'_3 \oplus L_0 \oplus L'_0)$ upper bound the probability that R''_0 satisfies the above formula.

5. (3 points): Using the last two observations explain why our $k = 3$ Feistel round construction PRP_{K_1,K_2,K_3} is not a strong PRP. What does the distinguisher do? (Note: it is possible to answer parts D and E without answering part C).

Answer: ...

Resource and Collaborator Statement: ...

Question 4 (15 points)

For each of the following constructions of a compression function h from a block cipher F_k, either show an attack or prove collision resistance in the ideal-cipher model:

1. $h(x, k) = F_k(x)$.
2. $h(x, k) = F_k(x) \oplus x \oplus k$.
3. $h(x, k) = F_k(x) \oplus k$.

Answer: ...

Resource and Collaborator Statement: ...

Question 5 (15 points)

In the course slides we informally introduced the notion of Authenticated Encryption with Associated Data (AEAD) when looking at Galois Counter Mode (GCM). An AEAD encryption scheme consists of three algorithms $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ where the encryption algorithm $\mathtt{Enc}_K(a, m; R)$ takes as input a message m associated data a and random coins R and outputs a ciphertext c. Similarly, the decryption algorithm $\mathtt{Dec}_K(a, c)$ takes as input a ciphertext c and associated data a and outputs a plaintext message m or $\perp$ indicating failure. For correctness we require that for any secret key K, associated data a and random coins R we have $\mathtt{Dec}_K(a, \mathtt{Enc}_K(a, m; R)) = m$. Intuitively, the scheme should resist chosen plaintext/ciphertext attacks and it should be impossible to forge a new ciphertext and/or tamper with the associated data. Your task is to provide a formal security definition that captures these intuitive properties and give a brief, but precise, explanation for why your definition captures these properties. You may choose to either define asymptotic or concrete security. (Hint: Think about what games you can define).

Answer: ...

Resource and Collaborator Statement: ...