Homework 4: Padding Oracle Attacks, MACs and Hash Functions

Submission policy. Submit your answers on Blackboard by 11:59pm Friday, **Oct. 29**, 2021. Your submission should include a .PDF file with all the answers to the theoretical problems. You should try to typeset your homeworks. Latex is especially recommended. No late submissions will be accepted. Your writeup MUST include the following information:

1. Your name and whether you take the class at **487** or **587** level.
2. List of references used (online material, course nodes, textbooks, wikipedia, etc.)

The homework will be graded by the class TA Anuj Pokhrel (apokhre@gmu.edu).

Exercise 1. Padding Oracle Attack [20 points] Suppose an adversary holds a ciphertext (c_0, c_1, c_2) encrypting a message, $m_1||m_2$ CBC-mode encryption. Let $L = 8$ denote the block length of the PRP, measured in bytes, and let $m_i^{(j)}$ denote the jth byte of the ith block of the plaintext. Similarly, let $c_i^{(j)}$ denote the jth byte of the ith block of the ciphertext. Construct an adversary that recovers the last byte of the 1st block of the message: $m_1^{(8)}$. (In class we already discussed how to recover padding and all bytes of m_2.)

Exercise 2. Insecure MACs [30 points] Let F be a fixed-length PRF. In all questions below you **cannot** do a truncation attack. The schemes are defined for fixed length messages.

1. Show that the following MAC scheme for messages m of length ℓn where $m = m_1||...||m_\ell$ (and each m_i are of size n-bits) is not secure.

$$t = F_k(m_1) \oplus \cdots \oplus F_k(m_\ell)$$

2. Show that the following MAC scheme for messages m of length $2n$ (where $m = m_1||m_2$ and each m_1, m_2 are of size n-bits) is not secure.

$$t = F_k(m_1)||F_k(F_k(m_2))$$

Exercise 3. Secure MAC [20 points] Suppose MAC is a secure MAC algorithm. Define a new algorithm:

$$MAC'(k, m) = MAC(k, m)||MAC(k, m).$$

Prove that MAC' is also a secure MAC algorithm via a security reduction. Follow these steps:

- Show how verification works.
- State the contrapositive.
- Describe your reduction.
- Write the Security Analysis for your reduction.

Exercise 4. Insecure Hash Functions [30 points] Show that the following hash functions are insecure. To do that explain how to find a specific collision pair.

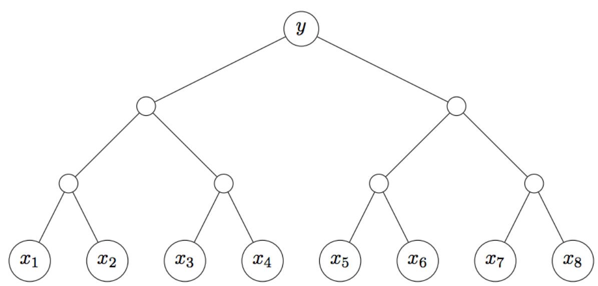
1. Consider a function $H : \{0,1\}^* \rightarrow \{0,1\}^n$. On input a message $m = x \oplus w$, the function outputs $y = H(m) = H(x) \oplus H(w)$. Show that this is NOT a collision resistance hash function (i.e. show two different messages m_1, m_2 such that they map to the same y).

2. Let $H_s^a(x_1||x_2) = H_s(x_1) \oplus H_s(x_2)$ where H is a collision resistance hash function. Show that H^a is NOT collision resistance.

Exercise 5. ONLY IF in 587 level [20 points]
One can view Merkle-trees as a "mode of operation" for hash functions. They allow to construct a hash function that takes as input a value a large value using a hash function that accepts much smaller inputs. Consider for example a CRHF $H_s : \{0,1\}^{2n} \rightarrow \{0,1\}^n$, using Merkle trees one can design a hash function $H'_s : \{0,1\}^{2n^h} \rightarrow \{0,1\}^n$ for some fixed positive integer h (i.e. the Merkle tree compresses 2^h strings of size $2n$ to an n-bit block).
The Merkle-tree construction is a binary tree of depth h constructed as follows: for every n-bit input block $x_1, x_2, \ldots, x_{2^h}$ we define the parent of two nodes to have value $H_s(a||b)$ where a and b are the values of the parent's two children, where $||$ denotes concatenation (note that order matters in "hashing up", a represents the left hand side child and b the right hand side child). Then, the root node has value $y = H'_s(x_1, x_2, \ldots, x_{2^h})$.

a. Show that if an adversary can find a collision for H' then can find a collision for H (i.e. prove that H' is collision resistant assuming H is collision resistant.)

b. Explain why the construction is insecure if h is not fixed (i.e. find two messages with different lengths that hash/match to the same value.)


y
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8