Problem Set #9

CMSC/Math 456
Instructor: Daniel Gottesman

Due on Gradscope, Thursday, Nov. 30, noon

General instructions: If you collaborate or use any outside resources, remember to cite them in your solutions.

Note: This assignment is 2 pages long.

Problem #1. Tree-Based Hash Function (70 pts.)

For this problem, we will consider an alternative construction of a hash function based on a Merkle tree. Let $h(x_1, x_2)$ be a collision-resistant compression function that takes two inputs of size s bits and has one output of size s bits. Given a message m, break it up into N blocks of size s. Assume $N < 2^s$. If the message is not a multiple of s bits long, pad it by adding sufficiently many 0s on the end to make the length equal to exactly Ns. The ith block of the message is m_i, $i = 1, \ldots, N$.

Let n be the smallest integer such that $N < 2^n$ and let $a = 2N - 2^n$ and $b = 2^n - N$. Then we will form a binary tree with a leaves at depth n and b leaves at depth $n-1$, with each leaf corresponding to a message block m_i.

For $0 < i \leq a/2$, let

$$h_i = h(m_{2i-1}, m_{2i}) \tag{1}$$

and for $a/2 < i \leq 2^{n-1}$, let

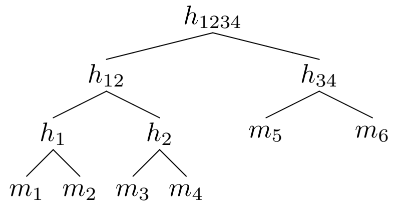
$$h_i = m_{i+a/2}. \tag{2}$$

Then let the tree entries at depth $n - d$ from the root ($1 < d \leq n$) be

$$h_{2^{d-1}(i-1)+1,\ldots,2^{d-1}i} = h(h_{2^{d-1}(i-1)+1,\ldots,2^{d-2}(2i-1)}, h_{2^{d-2}(2i-1)+1,\ldots,2^{d-1}i}) \tag{3}$$

for $1 \leq i \leq 2^{n-d}$. This gives us the Merkle tree construction. Then

$$H(m) = h_{1,\ldots,2^{n-1}}. \tag{4}$$



Above is a figure of an example with $N = 6$, which means $n = 3$, $a = 4$, $b = 2$,

$$H(m) = h_{1234} = h(h_{12}, h_{34}) = h(h(h_1, h_2), h(m_5, m_6)) = h(h(h(m_1, m_2), h(m_3, m_4)), h(m_5, m_6)) \tag{5}$$

The function $h(x_1, x_2)$ is collision resistant in the sense that it is computationally hard to find two input pairs (x_1, x_2) and (x'_1, x'_2) such that $h(x_1, x_2) = h(x'_1, x'_2)$ with $x_1 \neq x'_1$ or $x_2 \neq x'_2$ (or both $x_1 \neq x'_1$ and $x_2 \neq x'_2$). We say that $H(x)$ is collision resistant for pairs of messages both of length N if it is hard to find x, x', both of length N blocks, with $x \neq x'$ but $H(x) = H(x')$.

a) (10 points) Draw a tree and write the formula for the hash function evaluated on a message of length 4 blocks.

b) (10 points) Explain how to quickly find a collision between a message of length 4 blocks and one of length 6 blocks. You may choose both messages provided they have the given lengths.

c) (10 points) Explain how to quickly find a collision between a message of length N_1 blocks and one of length N_2 blocks with $N_1 < N_2$. N_1 and N_2 are given to you, but you may choose both messages provided they have the given lengths.

d) (10 points) Prove that if $h(x_1, x_2)$ is collision resistant, then $H(m)$ is collision resistant for pairs of messages both of length exactly $N = 2$ blocks. (That is, no padding is needed for the two messages.)

e) (10 points) Prove that if $h(x_1, x_2)$ is collision resistant and $H(m)$ is collision resistant for pairs of messages both of length N, then $H(m)$ is collision resistant for pairs of messages both of length exactly $N + 1$. (Again, no padding is needed for the two messages.)

f) (10 points) Use induction and parts b and c to prove that if $h(x_1, x_2)$ is collision resistant, $H(m)$ is collision resistant for pairs of messages both of length exactly N blocks (again with no padding), provided N is at most a polynomial function of s.

g) (10 points) Suggest an alternative padding technique to make it hard to find collisions between messages of different lengths. You do not have to prove that your technique works, but there should be no straightforward attacks.

Problem #2. CTR and CBC With a Decryption Oracle (50 pts.)

For this problem, assume that, as in the definition of CCA security, you have access to an encryption and decryption oracle, with the restriction that you can't apply the decryption oracle to the particular ciphertext c you are trying to decrypt. Suppose also that you are dealing with a ciphertext which is an encryption of a message broken up into N blocks each containing B bytes (so the total message length is $L = NB$). The message is padded using the PKCS #7 method we saw in class.(I.e., if b additional bytes are needed, fill them all with the value b. If 0 bytes are needed, fill the whole B-byte block with the value B.)

a) (25 points) Recall that CTR mode encrypts a message $(m_1, m_2, \ldots m_N)$ as

$$(IV, F_k(IV\|1) \oplus m_1, F_k(IV\|2) \oplus m_2, \ldots, F_k(IV\|N) \oplus m_N) \tag{6}$$

where IV is a random initial value of appropriate length, $F_k(x)$ is a pseudorandom function, and k is the key.

Using the encryption and decryption oracles the minimum amount needed, give a polynomial-time attack to decrypt a ciphertext c which was encrypted using CTR mode. How many times did you need each oracle?

b) (25 points) In class we saw an attack on CBC mode that used a padding oracle which only returned whether the message was correctly padded or not. That attack used at most $B + 256L$ padding oracle queries.

Now suppose you have full access to encryption and decryption oracles, but without being able to decrypt the particular ciphertext c. Find an efficient attack that decrypts c minimizing the use of the decryption oracle and therefore uses fewer decryption queries than the padding oracle attack. How many encryption and decryption oracle queries did you need?