Cryptography ENEE/CMSC/MATH 456: Homework 4

Due by beginning of class on 3/9/2020.

1. What is the effect of a dropped ciphertext block (e.g., if the transmitted ciphertext $c_1, c_2, c_3, \ldots$ is received as $c_1, c_3, \ldots$) when using the CBC, OFB, and CTR modes of operation?

2. OpenSSL is a utility that allows to perform various cryptographic operations. It should be pre-installed on your unix account. One of the cryptographic schemes implemented by OpenSSL is called AES (the Advanced Encryption Standard). AES is a symmetric key encryption scheme—a *block cipher*—which is used to encrypt Internet traffic. Later in the semester, we will study AES in depth. In this exercise, you will use the OpenSSL AES implementation to encrypt a file (see course webpage for the file), using your student id as the secret key. You will then use a cryptographic hash function (SHA1) to hash the ciphertext to a short string. The resulting short string should be submitted as the final answer to this exercise.
   As we will see below, an AES secret key is only 256 bits (32 bytes), but we will use it in CBC mode to encrypt a file of size nearly one million bytes. This is in contrast to perfectly secret schemes, where the key must be as long as the message.
   Here are some more details:
   - Read about OpenSSL here: `http://wiki.openssl.org/index.php/Enc`
   - We will be using AES-256-CBC to encrypt the file linked to on the course webpage. Make sure to explicitly set the key and the IV.
   - The AES-256 key is 256 bits and the IV is 128 bits For the IV, use a string of all 0s. For the key, use the 9 digits of your student ID appended with an appropriate number of zeros. For example, if your student id is 123456789, your secret key should be 12345678900. . .00.
   - Use OpenSSL to encrypt the file and place it in a temporary file.
   - Use the unix command gsha1sum (see documentation here: `http://manned.org/gsha1sum/392b94d5`) to output the cryptographic hash of the temporary file (we will cover cryptographic hash functions later this semester as well). Submit this final value as the answer to this exercise.
   - I will be precomputing the correct SHA1 hash for each student and will check that your final hash value matches mine.

3. Say $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ is a secure MAC, and for $k \in \{0,1\}^n$, the tag-generation algorithm Mac_k always outputs tags of length $t(n)$. Prove that t must be super-logarithmic or, equivalently, that if $t(n) = O(\log n)$ then Π cannot be a secure MAC.
   Hint: Consider the probability of randomly guessing a valid tag.

4. Consider the following MAC for messages of length $\ell(n) = 2n - 2$ using a pseudorandom function F: On input a message $m_0||m_1$ (with $|m_0| = |m_1| = n - 1$) and key $k \in \{0,1\}^n$, algorithm Mac outputs $t = F_k(0||m_0)||F_k(1||m_1)$. Algorithm Vrfy is defined in the natural way. Is $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ secure? Prove your answer.

5. Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticated fixed-length messages. (In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let $\langle i \rangle$ denote an $n/2$-bit encoding of the integer i.)

(a) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \cdots \oplus F_k(m_\ell)$.
(b) To authenticate a message $m = m_1, ..., m_\ell$, where $m_i \in \{0,1\}^{n/2}$, compute $t := F_k(\langle 1 \rangle || m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle || m_\ell)$.