COMP 537: Cryptography **Fall 2024**

Problem Set 1

Instructions: You **must** typeset your solution in LaTeX using the provided template. Please submit your problem set via Gradescope. Include your name and the names of any collaborators at the top of your submission.

Acknowledgment: Several of the problems in this problem set come from the Boneh-Shoup textbook.

Problem 1: Meet Me in the Middle [10 points]. This problem considers a broken way to design block ciphers with longer keys. Suppose we want to get security equivalent to AES256 (which uses a 256 bit key) but only want to use AES128 in our implementation. One (broken) way to do this would be to pick a key $k = (k_1, k_2)$, where $k_1, k_2 \in \{0,1\}^{128}$. Then we define 2AES128 as follows.

$$\mathsf{2AES128.Enc}((k_1, k_2), m) := \mathsf{AES128.Enc}(k_2, \mathsf{AES128.Enc}(k_1, m))$$

$$\mathsf{2AES128.Dec}((k_1, k_2), c) := \mathsf{AES128.Dec}(k_1, \mathsf{AES128.Dec}(k_2, c))$$

It turns out that this approach does not provide the same security as AES256, and it is susceptible to an attack that recovers the whole key k, but only uses 2^{129} time and space (so you're not actually getting more security than using AES128 a single time by itself). Show how an attacker who is given a plaintext/ciphertext pair (m, c) can find a key k such that $\mathsf{2AES128.Enc}(k, m) = c$.
Hint: The 2018 song "The Middle" by Zedd, Morris, and Grey can be thought of as a tribute to this problem.
Better Hint: The solution to this problem can be found in chapter 4.2 of the Boneh-Shoup textbook. Feel free to read it and explain it here in your own words.

Problem 2: Exercising the PRG definition [15 points]. Suppose G is a secure PRG that outputs bit strings in $\{0,1\}^n$. Which of the following generators derived from G are secure? For each one, either state "secure" or "insecure" followed by a short (1-2 sentence) justification. If it's insecure, your justification should state an attack on PRG security. Note: we use both the $x||y$ and (x, y) notations to denote concatenation.

(a) $G'(s) = (G(s), G(s))$

(b) $G'(s) = G(s) \oplus 1^n$

(c) $G'(s_1||s_2) = G(s_1) \oplus G(s_2)$

(d) $G'(s_1||s_2) = G(s_1) \wedge G(s_2)$ where $\wedge$ denotes bitwise AND

(e) $G'(s_1||s_2) = (s_1, G(s_2))$

Problem 3: Encryption and Compression [10 points]. Suppose two standards committees propose to save bandwidth by combining (lossless) compression with encryption. This kind of compression generally works by removing repeated sequences of bytes in a file to save space. Both committees plan on using a variable-length one-time pad for encryption.

(a) One committee proposes to compress messages before encrypting them. Explain why this is a bad idea.
Hint: Recall that compression can significantly shrink the size of some messages while having little impact on the length of other messages.

(b) The other committee proposes to compress ciphertexts after encryption. Explain why this is a bad idea.

Note: Over the years, many problems have surfaced when combining encryption and compression, including multiple attacks on widely-used cryptographic schemes.

Problem 4: Self-Referential Encryption [10 points]. Let us show that encrypting a key under itself can be dangerous. Let $\mathcal{E}$ be a (one-time) semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, where $\mathcal{K} \subseteq \mathcal{M}$, and let $k \xleftarrow{R} \mathcal{K}$. A ciphertext $c_* := \mathsf{Enc}(k, k)$ that encrypts k using k is called a *self referential encryption*.

(a) Construct a cipher $\mathcal{E}' = (\mathsf{Enc}', \mathsf{Dec}')$ derived from $\mathcal{E}$ such that $\mathcal{E}'$ is semantically secure, but becomes insecure if the adversary is given $\mathsf{Enc}'(k, k)$ at the beginning of the semantic security game. Please provide the cipher, the attack, and a 1-2 sentence explanation for why it remains semantically secure if the adversary is not given $\mathsf{Enc}'(k, k)$.

(b) Construct a cipher $\mathcal{E}' = (\mathsf{Enc}', \mathsf{Dec}')$ derived from $\mathcal{E}$ such that $\mathcal{E}'$ is semantically secure and remains semantically secure (provably) even when the adversary is given $\mathsf{Enc}'(k, k)$ at the beginning of the semantic security game. To prove that $\mathcal{E}'$ is semantically secure, you should show the following: for every adversary $\mathcal{A}$ that attacks $\mathcal{E}'$, there exists an adversary $\mathcal{B}$ that attacks $\mathcal{E}$ such that (i) the running time of $\mathcal{B}$ is about the same as that of $\mathcal{A}$, and (ii) $\mathsf{SSAdv}[\mathcal{A}, \mathcal{E}'] \leq \mathsf{SSAdv}[\mathcal{B}, \mathcal{E}] + \mathsf{negl}$.

Optional Feedback [5 points]. Please answer the following questions to help design future problem sets. You are not required to answer these questions (the points are free), and if you would prefer to answer anonymously, please use the anonymous feedback form. However, we do encourage you to provide feedback on how to improve the course experience.

(a) Roughly how long did you spend on this problem set?

(b) What was your favorite problem on this problem set?

(c) What was your least favorite problem on this problem set?

(d) Any other feedback for this problem set? Was it too easy/difficult?

(e) Any other feedback on the course so far?