CS 6222: Introduction to Cryptography **Spring 2021**

Homework 1: Symmetric Cryptography

Due: February 17, 2021 at 5pm (Submit on Gradescope) **Instructor:** David Wu

Instructions. You **must** typeset your solution in LaTeX using the provided template:

`https://www.cs.virginia.edu/dwu4/courses/sp21/static/homework.tex`

You must submit your problem set via Gradescope. Please use course code **D55GP5** to sign up.

Collaboration Policy. You may discuss your general *high-level* strategy with other students, but you may not share any written documents or code. You should not search online for solutions to these problems. If you do consult external sources, you must cite them in your submission. You must include the computing IDs of all of your collaborators with your submission. Refer to the official course policies for the full details.

Problem 1: Encryption and Compression [8 points]. You are designing a new web service and would like to take advantage of both encryption and compression when responding to client requests. You are deciding between the following two proposals: (1) encrypt the data first, then compress the result before sending; or (2) compress the data first, then encrypt the result before sending.

(a) If you are optimizing for *performance*, which option do you prefer? Briefly explain your choice.

(b) Are there any security implications for the option you selected above? Briefly explain.

Problem 2: Pseudorandom Generators [18 points]. Let $G\colon \{0,1\}^\lambda \to \{0,1\}^n$ be a secure PRG. For each of the following functions G', indicate whether it is a secure PRG or not. If it is secure, give a *formal* proof; if not, describe an explicit attack.

(a) $G'(s) := G(s) \| (G(s) \oplus 1^n)$, where 1^n denotes the all-ones string of length n.

(b) $G'(s_1 \| s_2) := G(s_1) \oplus G(s_2)$.

(c) $G'(s_1 \| s_2) := s_1 \| G(s_2)$.

Please refer to this handout for examples of how to formally show whether a construction is secure or not.

Problem 3: Encrypting Twice? [12 points]. Intuitively, encrypting a message twice should not harm security. It turns out that this is not always true. Let $(\mathsf{Encrypt}, \mathsf{Decrypt})$ be a cipher and define the "encrypt-twice" cipher $(\mathsf{Encrypt}_2, \mathsf{Decrypt}_2)$ where $\mathsf{Encrypt}_2(k,m) := \mathsf{Encrypt}(k, \mathsf{Encrypt}(k,m))$.

(a) Give an example of a cipher $(\mathsf{Encrypt}, \mathsf{Decrypt})$ that is semantically secure, but $(\mathsf{Encrypt}_2, \mathsf{Decrypt}_2)$ is not semantically secure.

(b) Suppose $(\mathsf{Encrypt}, \mathsf{Decrypt})$ is CPA-secure. Prove that $(\mathsf{Encrypt}_2, \mathsf{Decrypt}_2)$ is also CPA-secure.

Problem 4: Key Leakage in PRFs [25 points]. Let $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a secure PRF. Use F to construct a function $F'\colon \{0,1\}^{n+1} \times \{0,1\}^n \to \{0,1\}$ with the following two properties:

- F' is a secure PRF.
- If the adversary learns the last bit of the key, then F' is no longer secure.

(a) Describe your construction $F'\colon \{0,1\}^{n+1} \times \{0,1\}^n \to \{0,1\}$. **Hint:** Consider changing the value of F at a single point.

(b) Show that if F is a secure PRF, then your construction F' is a secure PRF.

(c) Show that your construction F' is insecure against an adversary that learns the last bit of the key (i.e., at the beginning of the PRF security game, the challenger gives the last bit of the PRF key to the adversary). Specifically, give a complete description of your adversary and compute its advantage. This problem shows that leaking even a *single* bit of the secret key can break PRF security.

(d) Show how to use F to construct a PRF that remains secure against an adversary who learns *any single* bit of the secret key. Your solution should not assume any property about F (e.g., F may be completely insecure against adversaries that learn one bit of the key). Prove the security of your construction. To model this security property, consider a variant of the PRF security game where the adversary starts by specifying an index i and the challenger replies with the i^{th} bit of the PRF key. The rest of the security game proceeds as in the standard PRF security game. **Hint:** Consider using a *larger* key-space.

Problem 5: Time-Memory Trade-offs [12 points]. Let $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher (i.e., a PRP). Let $G\colon \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ be the function $G(k_1 \| k_2, x) := F(k_2, F(k_1, x))$. Given a pair (x, y), your goal is to find *some* key $k = k_1 \| k_2$ such that $y = G(k_1 \| k_2, x)$. A brute force search over all possible keys $k \in \{0,1\}^{2n}$ would take time $2^{2n} \cdot \text{poly}(n)$. Show that using $2^n \cdot \text{poly}(n)$ space, there is an algorithm that recovers some key k in $2^n \cdot \text{poly}(n)$ time. Prove the correctness and running time of your algorithm. Your solution to this problem is an example of a time-memory trade-off that is often possible in cryptanalysis.

Problem 6: Time Spent [3 extra credit points]. How long did you spend on this problem set? This is for calibration purposes, and the response you provide does not affect your score.

Optional Feedback. Please answer the following *optional* questions to help us design future problem sets. You do not need to answer these questions. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) What was your favorite problem on this problem set? Why?

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

(c) Do you have any other feedback for this problem set?

(d) Do you have any other feedback on the course so far?