COMP 537: Cryptography **Fall 2025**

Problem Set 2

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: 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) $\mathsf{SSAdv}[\mathcal{A}, \mathcal{E}'] \le \mathsf{SSAdv}[\mathcal{B}, \mathcal{E}] + \mathsf{negl}$.

Problem 2: CRHF Combiners [10 points]. We want to build a CRHF H using two CRHFs H_1 and H_2 such that if at some future time one of H_1 or H_2 is broken (but not both), then H is still secure.

(a) Show that $H'(x) = H_1(H_2(x))$ may not be secure if one of H_1 or H_2 is broken. For this problem, it suffices to show that H' is broken if only one of H_1 or H_2 is broken, you don't need to consider both cases.

(b) Suppose H_1 and H_2 are defined over $(\mathcal{M}, \mathcal{T})$. Let $H(m) := (H_1(m), H_2(m))$. Prove that H is a secure CRHF if either H_1 is secure or H_2 is secure.

Problem 3: The 802.11b Insecure MAC [5 points]. Consider the following MAC (a variant of which was used for WiFi encryption in 802.11b WEP). Let $F : \mathcal{K} \times \mathcal{R} \to \mathcal{Y}$ be a PRF where $\mathcal{Y} = \{0,1\}^{32}$. Let the function $\mathsf{CRC32} : \{0,1\}^* \to \{0,1\}^{32}$ be the simple and popular error-detecting code by that name, which is designed to detect random errors. Show that this attempted MAC scheme is insecure by describing an attack on existential unforgeability that requires only a single MAC query and a very small number of additional operations, and succeeds with probability 1.

You don't need to know anything about how CRC32 works to do the attack, and the attack should work regardless of the size of the output space $\mathcal{Y}$, i.e., the attack would work even if the scheme were instantiated with a secure PRF and hash function with a 256-bit output.

$\mathsf{Sign}(k,m) := r \xleftarrow{R} \mathcal{R}$

$\quad t \leftarrow F(k,r) \oplus \mathsf{CRC32}(m)$

Output (r, t)

$\mathsf{Verify}(k, m, (r, t))$: if $t = F(k, r) \oplus \mathsf{CRC32}(m)$: output "accept"

else: output "reject"

Problem 4: AE Practice [20 points]. For this problem, assume that the cipher $(\mathsf{Enc}, \mathsf{Dec})$ provides authenticated encryption, that $(\mathsf{Enc}_{\mathsf{CPA}}, \mathsf{Dec}_{\mathsf{CPA}})$ provides CPA security, and that H is a collision resistant hash function. For each proposed cipher, state whether it provides AE, provides CPA security only, or provides neither AE nor CPA security. In each case, provide an attack on AE/CPA security and/or a proof sketch (in at most a few sentences each). Note that if you say a cipher provides CPA security, you will need to both show an attack on AE (ciphertext integrity) and a proof sketch for CPA security.

(a) $\mathsf{Enc}'(k, m) := (\mathsf{Enc}(k, m), \mathsf{Enc}(k, m))$

$\mathsf{Dec}'(k, (c_1, c_2)) := \mathsf{Dec}(k, c_1)$ if $\mathsf{Dec}(k, c_1) = \mathsf{Dec}(k, c_2)$; $\perp$ otherwise

(b) $\mathsf{Enc}'(k, m) := c \leftarrow \mathsf{Enc}(k, m)$; Output (c, c)

$\mathsf{Dec}'(k, (c_1, c_2)) := \mathsf{Dec}(k, c_1)$ if $c_1 = c_2$; $\perp$ otherwise

(c) $\mathsf{Enc}'(k, m) := (\mathsf{Enc}_{\mathsf{CPA}}(k, m), H(m))$

$\mathsf{Dec}'(k, (c_1, c_2)) := \mathsf{Dec}_{\mathsf{CPA}}(k, c_1)$ if $H(\mathsf{Dec}_{\mathsf{CPA}}(k, c_1)) = c_2$; $\perp$ otherwise

(d) $\mathsf{Enc}'(k, m) := c \leftarrow \mathsf{Enc}_{\mathsf{CPA}}(k, m)$; Output $(c, H(c))$

$\mathsf{Dec}'(k, (c_1, c_2)) := \mathsf{Dec}_{\mathsf{CPA}}(k, c_1)$ if $H(c_1) = c_2$; $\perp$ otherwise

Problem 5: An Attack on Android KeyStore [10 points]. Let (E, D) be a secure block cipher (PRP) defined over $(\mathcal{K}, \mathcal{X})$, and let $(E_{\mathsf{cbc}}, D_{\mathsf{cbc}})$ be the cipher derived from (E, D) using randomized CBC mode. Let $H : \mathcal{X}^{\leq L} \to \mathcal{X}$ be a collision resistant hash function. Consider the following attempt to build an AE-secure cipher over $(\mathcal{K}, \mathcal{X}^{\leq L}, \mathcal{X}^{\leq L+2})$:

$E'(k, m) := E_{\mathsf{cbc}}(k, (H(m), m))$

$D'(k, c) : (t, m) \leftarrow D_{\mathsf{cbc}}(k, c)$

if $t = H(m)$: output m

else: output $\perp$

Note that, under this scheme, the encryption of a one-block message $m \in \mathcal{X}$ is a three-block ciphertext: the IV, a ciphertext block corresponding to $H(m)$, and a ciphertext block corresponding to m. Show that (E', D') is not AE-secure by giving a chosen ciphertext attack on it. That is, show that it does not satisfy CCA security.

This construction was once used to protect secret keys in Android KeyStore. The chosen ciphertext attack resulted in a compromise of the key store, and this scheme is no longer in use. For this problem only, you are allowed to follow the link provided by the Boneh-Shoup textbook to the paper that presents the original attack.

Hint: The attack only needs a single decryption query.

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?