Instructions: You must typeset your solution in LaTeX using the provided template. Please submit your problem set via Gradescope.

Bugs: If it looks like there might be a mistake in the statement of a problem, please ask a clarifying question on Piazza.

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

Problem 1: DDH PRG [5 points].
Let $G$ be a cyclic group of prime order $q$ generated by $g \in G$. Consider the following PRG defined over $(\mathbb{Z}_q^2, G^3)$:

$$G(\alpha, \beta) := (g^\alpha, g^\beta, g^{\alpha \beta}).$$

Show that $G$ is a secure PRG assuming that the DDH assumption holds in $G$.

Problem 2: Non-Binding Signatures [15 points].
It turns out that secure signatures are not necessarily binding. That is, suppose the signer generates a signature $\sigma$ on a message $m$. The definition of a secure signature does not preclude the signer from producing another message $m' \neq m$ for which $\sigma$ is a valid signature. It turns out binding isn't needed for many applications, so it's left out of the definition. That said, many signature schemes we have seen are in fact binding.

(a) Please give an example of a situation where a non-binding signature may cause problems.

(b) Please give an example of a situation where a non-binding signature should not cause problems.

(c) Please give an example of a signature scheme that is not binding: for a given $(pk, sk)$, the signer can find two distinct messages $m_0$ and $m_1$ where the same signature $\sigma$ is valid for both messages under $pk$. You don't need to prove security, but please do give the intuition for why the scheme is secure as well as an example of how the attacker can produce a second message for the same signature.

Hint: think about where else we've seen the message corresponding to a signature change after the signature is produced.

Problem 3: Attacking RSA-FDH [5 points].
Consider the RSA-FDH signature scheme. The public key is a pair $(N, e)$ where $N$ is an RSA modulus, and a signature on a message $m \in \mathcal{M}$ is defined as $\sigma := H(m)^{1/e} \in \mathbb{Z}_N$, where $H: \mathcal{M} \rightarrow \mathbb{Z}_N$ is a hash function. Suppose the adversary could find three messages $m_1, m_2, m_3 \in \mathcal{M}$ such that $H(m_1) \cdot H(m_2) = H(m_3)$ in $\mathbb{Z}_N$. Show that the resulting RSA-FDH signature scheme is no longer existentially unforgeable under a chosen message attack.
Problem 4: RSA Signatures with Same Modulus [15 points]. This problem explores why every party has to be assigned a different modulus $N = pq$ in the RSA trapdoor permutation. Suppose we try to use the same modulus $N = pq$ for everyone. Every party is assigned a public exponent $e_i \in \mathbb{Z}$ and a private exponent $d_i \in \mathbb{Z}$ such that $e_i \cdot d_i = 1 \mod \varphi(N)$. At first, this appears to work fine. To sign a message, $m \in \mathcal{M}$, Alice would publish the signature $\sigma_a \leftarrow H(m)^{d_a} \in \mathbb{Z}_N$ where $H : \mathcal{M} \to \mathbb{Z}_N^*$ is a hash function. Similarly, Bob would publish the signature $\sigma_b \leftarrow H(m)^{d_b} \in \mathbb{Z}_N$. Since Alice is the only one who knows $d_a$ and Bob is the only one who knows $d_b$, this seems fine. Unfortunately, this scheme is completely insecure. Bob can use his secret key $d_b$ to sign messages on behalf of Alice.

(a) Show that Bob can use his public-private key pair $(e_b, d_b)$ to obtain a multiple of $\varphi(N)$. Denote this integer by $V$.

(b) Suppose Bob knows Alice’s public key $e_a$, and assume for now that $e_a$ is relatively prime to $V$. Show that for any message $m \in \mathcal{M}$, Bob can compute $\sigma \leftarrow H(m)^{1/e_a}$. In other words, Bob can invert Alice’s trapdoor permutation and obtain her signature on $m$.

Hint. Recall since $e_a$ and $V$ are relatively prime, Bob can find an integer $d$ such that $d \cdot e_a = 1 \mod V$, i.e., Bob can compute the inverse of $e_a \mod V$.

(c) Show how to make your solution in part (b) work even if $e_a$ is not relatively prime to $V$.

Problem 5: Double One-Way Functions [15 points]. In the lecture on identification protocols, we saw a protocol called S/key that uses an iterated one-way function. In this question we explore the security of iterated one-way functions.

(a) The iteration of one-way functions need not be one-way. Let $f : \mathcal{X} \to \mathcal{X}$ be a one-way function, where $0 \in \mathcal{X}$. Let $\hat{f} : \mathcal{X}^2 \to \mathcal{X}^2$ be defined as:

$$\hat{f}(x, y) = \begin{cases} (0, 0) & \text{if } y = 0 \\ (f(x), 0) & \text{otherwise} \end{cases}$$

Show that $\hat{f}$ is one-way, but $\hat{f}^{(2)}(x, y) := \hat{f}(\hat{f}(x, y))$ is not.

Note: while it’s possible that iterating a one-way function breaks security, that is not thought to be the case with the one-way functions used to instantiate S/Key.

(b) Show that the iteration of a one-way permutation is also one-way. In particular, suppose $f : \mathcal{X} \to \mathcal{X}$ is a one-way permutation, and show that $f^{(2)}(x) := f(f(x))$ is also one-way. As usual, prove the contrapositive.

(c) Explain why your proof from part (b) does not apply to a one-way function. Where does the proof fail?
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 Piazza's anonymous private post feature. 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?