Instructions.

- **When:** This problem set is due on **October 6, 2021** before **11pm ET**.

- **How:** You should use LaTeX to type up your solutions (you can use our LaTeX template from the course webpage). Solutions should be uploaded on Gradescope as a single pdf file.

- **Acknowledge your collaborators:** Collaboration is permitted and encouraged in small groups of at most three. You must write up your solutions *entirely on your own* and *acknowledge your collaborators*.

- **Reference your sources:** If you use material from outside the lectures, you must reference your sources (papers, websites, wikipedia, . . .).

- **When in doubt, ask questions:** Use Piazza or the TA office hours for questions about the problem set. See the course webpage for the timings.

Problem 1. Let's Encrypt *and* Authenticate!

Let $(\mathsf{Gen_{Enc}}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure encryption scheme and $(\mathsf{Gen_{MAC}}, \mathsf{MAC}, \mathsf{Verify})$ be an EUF-CMA secure scheme (defined below). Suppose Alice and Bob meet up in their secret hideout before Alice leaves to study abroad on Mars, and generate two secret keys k_1 and k_2, for encryption and authentication, respectively. That is, we define their algorithm $\mathsf{Gen}'(1^\lambda)$ to return $k_1 \leftarrow \mathsf{Gen_{Enc}}(1^\lambda)$ and $k_2 \leftarrow \mathsf{Gen_{MAC}}(1^\lambda)$.

While Alice is on Mars, they can only send each other messages via a public Earth-Mars broadcast (which is monitored by their nemesis E.V.E.), but they still want to communicate in a private and authenticated way. For Alice and Bob, this just means IND-CPA and EUF-CMA security (note that in the real world, we want much stronger guarantees.

Definition 1 (IND-CPA-security)

Let $\mathcal{E} = (\mathsf{Gen_{Enc}}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme with message space $\mathcal{M}$ and key space $\mathcal{K}$ with security parameter λ. Sample secret key $k \leftarrow \mathsf{Gen_{Enc}}(1^\lambda)$ and define encryption oracle $\mathsf{Enc}(k, \cdot)$, which on query m, outputs $\mathsf{Enc}(k, m)$. $\mathcal{E}$ is said to be IND-CPA-secure (a.k.a. computationally indistinguishable against chosen plaintext attacks) if for all PPT algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there exists a negligible function negl such that for all λ

$$\Pr \left[\begin{array}{l} k \leftarrow \mathsf{Gen}(1^\lambda); \\ (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Enc}(k,\cdot)}(1^\lambda); \\ b \xleftarrow{R} \{0,1\}; \ c \leftarrow \mathsf{Enc}(k, m_b); \\ b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}(k,\cdot)}(1^\lambda, c, \mathsf{state}) : \\ b' = b \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Definition 2 (EUF-CMA-security)
Let $\mathcal{E} = (\mathsf{Gen}_{\mathsf{MAC}}, \mathsf{MAC}, \mathsf{Verify})$ be a message authentication scheme with message space $\mathcal{M}$ and key space $\mathcal{K}$ with security parameter λ. Let $k \leftarrow \mathsf{Gen}(1^\lambda)$ and define MAC oracle $\mathsf{MAC}(k, \cdot)$, which on query m, outputs $\mathsf{MAC}(k, m)$. $\mathcal{E}$ is said to be EUF-CMA-secure (a.k.a. existentially unforgeabile against chosen message attacks) if for all PPT algorithms $\mathcal{A}$ there exists a negligible function negl such that for all λ

$$\Pr \left[\begin{array}{l} k \leftarrow \mathsf{Gen}(1^\lambda); \\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{MAC}(k, \cdot)}(1^\lambda): \\ m^* \notin Q \text{ and } \mathsf{Verify}(k, m^*, \sigma^*) = 1 \end{array} \right] \leq \mathsf{negl}(\lambda),$$

where Q is the set of messages that $\mathcal{A}$ queried to the oracle.

For each of the following:

- **Construct algorithms $\mathsf{Dec}', \mathsf{Verify}'$ such that $\mathcal{E}_1 = (\mathsf{Gen}', \mathsf{Transmit}, \mathsf{Dec}')$ is a correct encryption scheme and $\mathcal{E}_2 = (\mathsf{Gen}', \mathsf{Transmit}, \mathsf{Verify}')$ is a correct message authentication scheme (both schemes will use both keys).**
- **Either prove $\mathcal{E}_1$ is IND-CPA secure and $\mathcal{E}_2$ is EUF-CMA secure via reductions, or provide a attack on at least one.**

(a) $\mathsf{Transmit}(k_1, k_2, m) = \mathsf{Enc}(k_1, (m, \mathsf{MAC}(k_2, m)))$.

(b) $\mathsf{Transmit}(k_1, k_2, m) = (\mathsf{Enc}(k_1, m), \mathsf{MAC}(k_2, m))$.

(c) $\mathsf{Transmit}(k_1, k_2, m) = (\mathsf{Enc}(k_1, m), \mathsf{MAC}(k_2, \mathsf{Enc}(k_1, m)))$.

Problem 2. Building one-way functions

Suppose f is a length-preserving[1] one-way function. In this problem, we write

- $\oplus$ to denote bitwise XOR,
- $||$ to denote concatenation of bit-strings,
- $\bar{x}$ to denote the bitwise complement of x.

For each of the following functions f', either prove that f' is always a OWF (by a reduction to the one-wayness of f), or provide a counter example showing that f' is not always a OWF for some OWF f.

(a) $f'(x, y) = f(x) || f(x \oplus y)$, where $|x| = |y|$.

(b) $f'(x) = f(\bar{x})$

(c) $f'(x) = f(x)_{[1:|x|-1]}$

(d) $f'(x) = f(f(x))$

---

[1] For every $x \in \{0, 1\}^*$, it holds that $|f(x)| = |x|$.

Problem 3. More fun with one-way functions!

Alice comes across a function $f(x, y) = (g_1(x), g_2(y))$ based on two one-way functions g_1, g_2. She wants to try and invert this function. Let $x, y \in \{0, 1\}^\lambda$. Her friend Bob has a access to a special black-box algorithm $\mathcal{B}$ which, on input $(g_1(x), g_2(y))$, computes the inner product of x and y mod 2, denoted $\langle x, y \rangle \bmod 2$. Specifically, $\mathcal{B}(1^\lambda, g_1(x), g_2(y))$ outputs

$$\langle x, y \rangle \bmod 2 = \sum_{i=1}^{\lambda} x_i y_i \bmod 2.$$

He's willing to help Alice by giving her access to $\mathcal{B}$.

(a) **Suppose that $\mathcal{B}$ outputs the correct inner product with near-perfect probability $1 - \mathsf{negl}(\lambda)$ on <u>random</u> x and y. Prove that with Bob's help, Alice can use $\mathcal{B}$ to invert f with non-negligible probability.**

(b) **Now, suppose $\mathcal{B}$ outputs the correct inner product with probability $\frac{1}{2} + \epsilon$ for some constant $\frac{1}{4} < \epsilon < \frac{1}{2}$, again for random x and y. Prove that Alice can still use $\mathcal{B}$ to invert f with non-negligible probability. (The runtime of Alice's inverter can depend on ϵ.)**

Problem 4. Random self-reducibility

(a) Recall the Computational Diffie–Hellman (CDH) assumption from lecture.

CDH assumption: Given $(\mathbb{G}, g, g^a, g^b)$ where p is prime, $\mathbb{G}$ is a cyclic group of order p, and a, b are *random* in $\mathbb{Z}_p$, it is computationally intractable to compute g^{ab}.

Suppose that Bob has an instance of a Diffie-Hellman tuple $(\mathbb{G}, g, g^x, g^y)$ for some **worst-case** $x, y \in \mathbb{Z}_p$. Bob has no idea how to compute g^{xy} for these values of x and y, but Alice does have an algorithm $\mathcal{A}$ which on input $(\mathbb{G}, g, g^\alpha, g^\beta)$ for **random** $\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$, can output $g^{\alpha\beta}$ with non-negligible probability over the sampling of α, β.

Prove that CDH is random self-reducible in $\mathbb{Z}_p^*$. I.e., Bob can use Alice's $\mathcal{A}$ to solve his worst-case instance.

(b) Consider the following variant of the CDH assumption (sometimes referred to as the n-CDH assumption).

n-CDH assumption: Given n-CDH tuple $(\mathbb{G}, g, g^a, g^{a^2}, \ldots, g^{a^{n-1}})$ where p is prime, $\mathbb{G}$ is a cyclic group of order p, and a is *random* in $\mathbb{Z}_p$, it is computationally intractable to compute g^{a^n}.

Alice claims that n-CDH is not random self-reducible. Bob, however, believes that it is.

Who is correct? Is the n-CDH assumption random self-reducible? Prove your answer.

Problem 5. Designatable PRFs

Recall the definition of pseudorandom functions presented in lecture. We have also seen some constructions of PRFs, from other primitives like PRGs. In this problem we will consider a variant of PRFs.

We define a **designatable PRF** to be a PRF family $\mathcal{F} = \{f_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ for $n = n(\lambda)$ equipped with two special PPT algorithms $\mathsf{Designate} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ and $\mathsf{Execute} : \{0,1\}^n \times \{0,1\}^{n-m} \to \{0,1\}^n$ such that for any $k \in \{0,1\}^n, y \in \{0,1\}^m, z \in \{0,1\}^{n-m}$,

$$\mathsf{Execute}\Big(\mathsf{Designate}(k,y), z\Big) = f_k(y\|z).$$

In other words, $\mathsf{Designate}$ takes in the key k and some "prefix" y, and outputs a designated key k_y. Given the designated key k_y, $\mathsf{Execute}$ can compute $f_k(x)$ for any $x \in \{0,1\}^n$ that has the prefix y (i.e. $x = y\|z$ for some $z \in \{0,1\}^{n-m}$).

Additionally, we require that any PPT algorithm $\mathcal{A}$ given a designated key k_y for prefix y, can *only* compute $f_k(x)$ with non-negligible probability for x with the prefix y. That is, for any $y, y' \in \{0,1\}^m, z \in \{0,1\}^{n-m}$ with $y \neq y'$, we have

$$\Pr[k_y \leftarrow \mathsf{Designate}(k,y) : \mathcal{A}(k_y, y'\|z) = f_k(y'\|z)] \leq \mathsf{negl}(\lambda).$$

Prove that if PRFs exist, so do designatable PRFs.