

Midterm II

Name:

SID:

- Not all parts of a problem are weighted equally. Before you answer any question, read the problem carefully. Be precise and concise in your answers.
- You may consult at most *10 sheets of notes*. Apart from that, you may not look at books, notes, etc. Calculators, phones, computers, and other electronic devices are NOT permitted for looking up content. However, you may use an electronic device such as a tablet for writing your answers.
- You have 80 minutes: there are 3 questions on this exam worth a total of 100 points.
- You are allocated 90 minutes and the extra 10 minutes are provided for the submission of the exam on Gradescope. You must submit/upload the exam on time. Note that late submissions will not be accepted.
- **DSP Students** must submit the exam in time as per your accommodation. Thus, if you are allowed $1.5\times$ (resp., $2\times$) the exam time then you must submit it within $80 * 1.5 + 10$ (resp., $80 * 2 + 10$) mins, i.e. 130 (resp., 170) mins. **Your allotted time has already been adjusted on Gradescope, and you will not need to email the midterm to the instructors like last time. Please submit on Gradescope.**
- The exam must be submitted before 8 PM PT on March 31st, 2021. No exams later than this time will be accepted regardless of when you start the test.
- We will not be answering questions during the exam. If you feel that something is unclear please write a note in your answer.

1 One-Way Functions and Collision-Resistant Hash Functions (30 points)

- (a) Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be an efficiently computable (not necessarily one-to-one) function with a hard-code predicate $B : \{0, 1\}^n \rightarrow \{0, 1\}$. Show that f is not necessarily a one-way function. That is, describe a function f and a predicate B , and prove that i) B is a hard-code predicate for f and ii) f is not one-way (10 points).

Name: _____

- (b) Let $f_1 : \{0, 1\}^n \rightarrow \{0, 1\}^n$, $f_2 : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be one-way functions. Show that $g : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ defined as $g(x, y) = f_1(x) \oplus f_2(y)$ is not necessarily one-way. That is, describe two functions f_1, f_2 , and prove that i) each is one-way, and ii) g is not one-way. You may **either** follow the template below, **or** come up with your own construction.

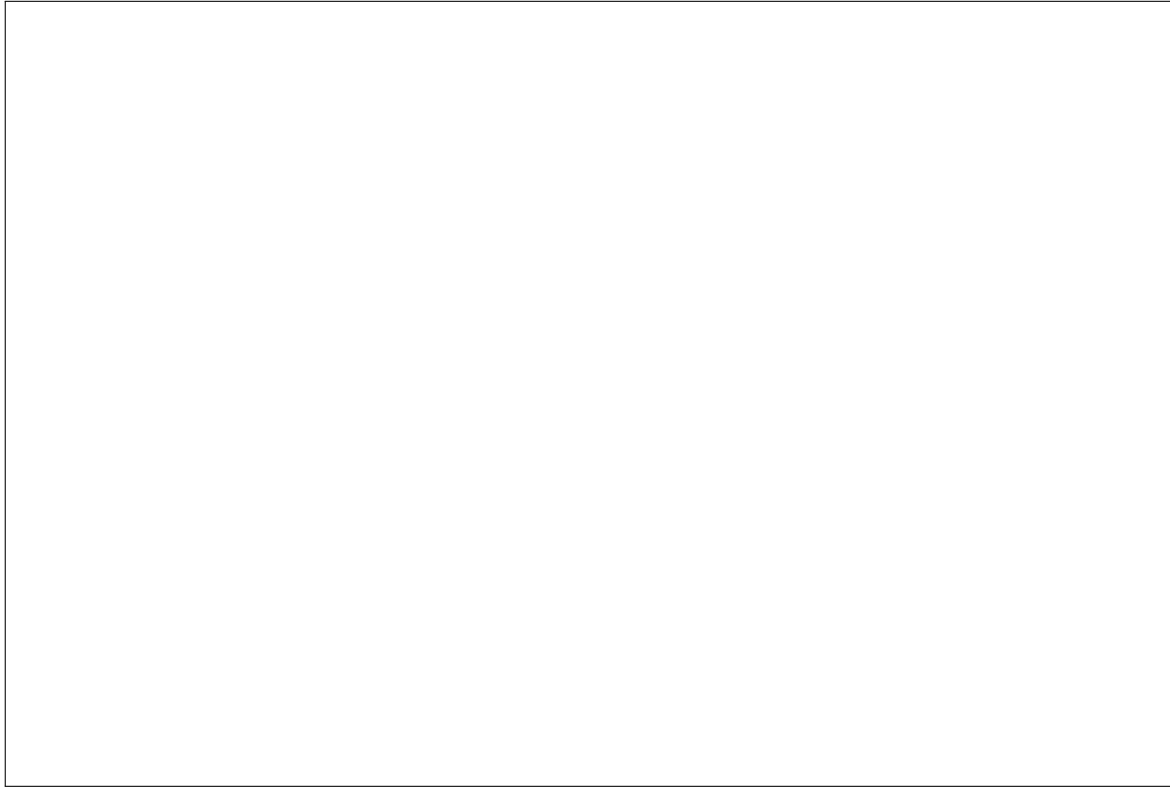
Template: Let $h' : \{0, 1\}^n \rightarrow \{0, 1\}^{n-2}$ be a one-way function, and let $h(x) = (1, h'(x), 1)$. Argue that h is one-way (**2 points**).

Now, define

$$f_1(x) = \begin{cases} \boxed{} & \text{if } x = (z, 0^{n/2}) \\ \boxed{} & \text{otherwise} \end{cases}, f_2(x) = \begin{cases} \boxed{} & \text{if } x = (z, 0^{n/2}) \\ \boxed{} & \text{otherwise} \end{cases}$$

and argue that each is one-way (**5 points**).

Finally, show that $g(x, y) = f_1(x) \oplus f_2(y)$ is not one-way (**3 points**).



Name:

Free response: Alternatively, present your own construction (**10 points**).

-
- (c) Let (Gen, H) be a collision resistant hash function, where $H^s : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$. Let (Gen', H') be defined as follows. $\text{Gen}'(1^n)$ runs $\text{Gen}(1^n)$ twice to obtain independently sampled hash keys s_1, s_2 . Then $H'^{s_1, s_2} : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ is defined as $H'^{s_1, s_2}(x) = H^{s_1}(x) \oplus H^{s_2}(x)$. Show that (Gen', H') is not necessarily collision-resistant. That is, define a hash function (Gen, H) and prove that i) (Gen, H) is collision-resistant (assuming the existence of some other collision-resistant hash function (Gen^*, H^*)), and ii) (Gen', H') is not collision-resistant (**10 points**).

Name:

2 CCA-secure MAC (45 points)

Consider a “CCA-style” extension to the definition of secure message authentication codes, where the adversary is provided with both a `Mac` and a `Vrfy` oracle. Our starting point will be the “standard” notion of MAC security, called “existential unforgeability under adaptive chosen-message attacks”, and we will consider a variant of this definition that allows for `Vrfy` oracle queries.

- (a) Provide a formal definition of CCA-secure MACs. That is, describe an experiment called `CCA – Mac – Forge \mathcal{A}, Π (n)`, and provide a security requirement stating that no adversary can win your game except with negligible probability (**10 points**).

-
- (b) Assume that Π is a standard secure *deterministic* MAC that has *canonical verification*, meaning that i) the Mac algorithm is deterministic and ii) the Vrfy algorithm, on input (m, t) , recomputes $t' := \text{Mac}_k(m)$ and accepts if $t' = t$. Prove that Π also satisfies your definition from part (a) (**15 points**).

Name:

- (c) Assume that $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ is a standard secure MAC. Construct another MAC $\Pi' = (\text{Gen}', \text{Mac}', \text{Vrfy}')$ that is standard secure, but is *not* secure under your definition from part (a). You may **either** follow the template below, **or** come up with your own construction.

Template: The idea will be to set things up so that the Vrfy' oracle can be used by the adversary to learn the key k one bit at a time. In particular, Gen' will just run Gen , and Mac' will simply run Mac and then append 3 bits, all set to 0:

$$\text{Mac}'_k(m) = (\text{Mac}_k(m), 0, 0, 0).$$

Now, define your Vrfy' function (**10 points**):

Next, show that Π' is a secure MAC (**5 points**):

Finally, show that Π' is insecure when \mathcal{A} is given a verify oracle (**5 points**).

Free response: Alternatively, present your own construction (**20 points**).

Name:

3 Feistel Network (25 points)

Consider a three-round Feistel network with block size 64 bits and mangler function $\hat{f} : \{0, 1\}^{48} \times \{0, 1\}^{32} \rightarrow \{0, 1\}^{32}$ that takes a key k of length 48 bits and maps 32-bit inputs to 32-bit outputs. Suppose that there was a flaw in the design of \hat{f} that resulted in the following behavior. With probability $1/2$ over the sampling of the key, it holds that *for all inputs*, the first bit of the output of $\hat{f}(k, \cdot)$ will be set to the first bit of its input. That is,

$$\Pr_{k \leftarrow \{0, 1\}^{48}} [\forall x \in \{0, 1\}^{32}, \hat{f}(k, x)_1 = x_1] = 1/2.$$

Show that there exists some efficient adversary \mathcal{A} and constant $\epsilon > 0$ such that \mathcal{A} has advantage ϵ in distinguishing this Feistel network from a uniformly random permutation. Your \mathcal{A} should only require a single input-output pair, and you are not required to optimize the distinguishing advantage ϵ (just show that it is greater than 0). You may make the following heuristic assumptions.

- Each round i of the network applies \hat{f} with an independent and uniformly sampled key k_i .
- For keys k that do not satisfy the condition that $\forall x \in \{0, 1\}^{32}, \hat{f}(k, x)_1 = x_1$, assume that $\hat{f}(k, \cdot)$ is a uniformly random function.