Example Answers From Midterm II

CS 171

March 2024

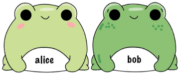
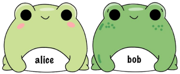
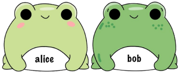


Table of Contents



Summary

- Let's go over some typical answers to the short answer questions and discuss what works and what doesn't.
- We give partial credit for stating the correct intuition for the proof.



What makes a good proof?

What makes a good proof?

- No corner cases: It should not be possible to find flaws in your argument. There should be no corner cases where your claims are false.
- Clear writing: Express your ideas clearly. Use complete sentences, precise language, etc.

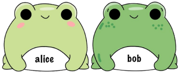
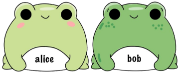


Table of Contents



Question 2: OWFs

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a OWF. Use f to construct another OWF g such that $g : \{0,1\}^n \to \{0,1\}^n$ and $g(0^n) = 0^n$. Your answer should:

1. Describe a construction of g.
2. Prove that g is a OWF.

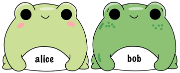
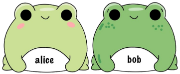


Table of Contents



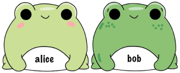
Example Answer

- $g(x) = \begin{cases} 0^n & x = 0^n \\ f(x) & \text{otherwise} \end{cases}$
- If $g(x)$ is not a secure OWF, then we should be able to find a preimage of $g(x)$ with non-negligible probability.
- This function $g(x)$ is still a OWF because the output of the function that is invertible is 0^n.
- However, 0^n only occurs with probabilty $\frac{1}{2^n}$ and every other output is $f(x)$ which is secure.
- Therefore, we can only invert $g(x)$ with negligible probability and it is a OWF.

Comments:

- The intuition is right, however a reduction to the security of f is absent.
- The answer loosely resembles a contradiction proof in the beginning, but then the "proof" is given as an *observation* instead of a reduction.

Table of Contents

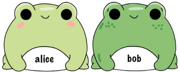


Question 3: Domain Extension

- Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a pseudorandom function.
- Let $\mathcal{H} = (\mathsf{Gen}, H)$ be a collision-resistant hash function with key space $\{0,1\}^n$ and input space $\mathcal{X}$, which may be very large. For every key $s \leftarrow \mathsf{Gen}(1^n)$, $s \in \{0,1\}^n$ and $H^s : \mathcal{X} \to \{0,1\}^n$.
- Let $G : \{0,1\}^{2n} \times \mathcal{X} \to \{0,1\}^n$ be defined as follows:

$$G((k,s),x) = F\big(k, H^s(x)\big)$$

- **Question:** Prove that G is a pseudorandom function.



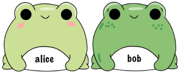
Hyb_0

Let $\underline{\mathsf{Hyb}_0(\mathcal{A}, n)}$ be the PRF security game in which the adversary $\mathcal{A}$ gets query access to G. In particular:

1. The PRF challenger samples $k \leftarrow \{0,1\}^n$ and $s \leftarrow \mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ gets query access to the following function:

$$G(\cdot) = F(k, H^s(\cdot))$$

3. The adversary outputs a bit b, which is the output of the hybrid.

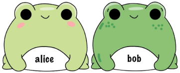


Let $\underline{\mathsf{Hyb}_1(\mathcal{A}, n)}$ be the same as $\mathsf{Hyb}_0(\mathcal{A}, n)$, except $F(k, \cdot)$ is replaced with a uniformly random function $R_1 : \{0,1\}^n \to \{0,1\}^n$:

1. The PRF challenger samples a function R_1 uniformly at random from the set of all functions mapping $\{0,1\}^n \to \{0,1\}^n$. They also sample $s \leftarrow \mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ gets query access to the following function:

$$R_1(H^s(\cdot))$$

3. The adversary outputs a bit b, which is the output of the hybrid.



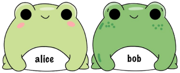
Hyb_2

Let $\underline{\mathsf{Hyb}_2(\mathcal{A}, n)}$ be the same as $\mathsf{Hyb}_0(\mathcal{A}, n)$ except $F(k, H^s(\cdot))$ is replaced with a uniformly random function $R_2 : \mathcal{X} \to \{0,1\}^n$:

1. The PRF challenger samples a function R_2 uniformly at random from the set of all functions mapping $\mathcal{X} \to \{0,1\}^n$.
2. The adversary $\mathcal{A}$ gets query access to:

$$R_2(\cdot)$$

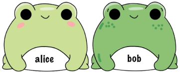
3. The adversary outputs a bit b, which is the output of the hybrid.



Lemma 3.1

Prove that for any PPT adversary $\mathcal{A}$,

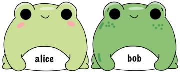
$$\left| \Pr[\mathsf{Hyb}_0(\mathcal{A}, n) \to 1] - \Pr[\mathsf{Hyb}_1(\mathcal{A}, n) \to 1] \right| \leq \mathsf{negl}(n)$$



Lemma 3.2

Prove that for any PPT adversary $\mathcal{A}$,

$$\left|\Pr[\mathsf{Hyb}_1(\mathcal{A}, n) \to 1] - \Pr[\mathsf{Hyb}_2(\mathcal{A}, n) \to 1]\right| \leq \mathsf{negl}(n)$$



Intuition for the Proof

- Our proof must use the PRF security of F and the collision-resistance of $\mathcal{H}$.
- If F is not a PRF, then G is not a PRF. Example: What if $F(x) = 0$ for all x.
- If $\mathcal{H}$ is not collision-resistant, then G is not a PRF. Example: What if $H^s(x) = H^s(\overline{x})$ for all s, x.

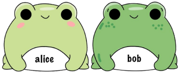
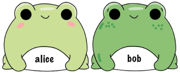


Table of Contents

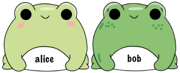


To prove lemma 3.1:

- $F(k, \cdot)$ is indistinguishable from $R_1(\cdot)$ because F is a pseudorandom function.
- We can treat $H^s(x)$ as just an input to $F(k, \cdot)$ or $R_1(\cdot)$ in hybrids 0 and 1.
- In conclusion, $F(k, H^s(\cdot))$ is indistinguishable from $R_1(H^s(\cdot))$ because $F(k, \cdot)$ is indistinguishable from $R_1(\cdot)$.

Comments:

- The intuition is right, but the argument doesn't get more concrete than intuition.
- You need to construct an adversary that will break the pseudorandomness of F.



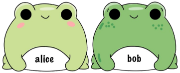
Example Answer

To prove lemma 3.2:

- R_1 and R_2 are truly random functions, so $R_1(H^s(\cdot))$ and $R_2(\cdot)$ are also uniformly random in some sense.
- Given query access to $R_1(H^s(\cdot))$ or $R_2(\cdot)$, the adversary cannot tell which of the two functions they are querying, because in either case, every query receives a uniformly random string in response.
- Therefore, Hyb_1 and Hyb_2 are indistinguishable.

Comments:

- It's possible to poke holes in this argument. What if $H^s(\cdot)$ is not collision-resistant? Then by querying the oracle on inputs that collide in H^s, you can distinguish $R_1(H^s(\cdot))$ and $R_2(\cdot)$.



Example Answer

To prove lemma 3.2:

- Since H^s is collision-resistant, then the adversary in Hyb_1 will (with overwhelming probability) query the function on inputs that do not collide.
- In response to each distinct query, the adversary will receive a uniformly random string that is independent of the other responses. This is the same distribution of responses that the adversary receives in Hyb_2. Therefore, Hyb_1 and Hyb_2 are indistinguishable.

Comments:

- The intuition is right, and the ideas are stated clearly.
- To get full credit, the answer needs to describe an algorithm that can find collisions in H^s (given an adversary that distinguishes Hyb_1 and Hyb_2).

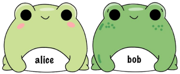
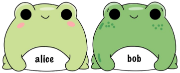
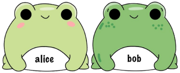


Table of Contents



Question 4: Encryption Combiner

Let us be given two public-key encryption schemes $\Pi_1 = (\text{Gen}_{1,1\ ,1}\ )$ and $\Pi_2 = (\text{Gen}_{2,2\ ,2}\ )$. Let the ciphertext space of Enc_2 be the same as the message space of Enc_1. Also, one of Π_1 or Π_2 is CPA secure, and the other one is not, but we don't know which one is secure.

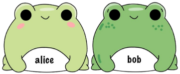


Question 4: Encryption Combiner

Define the composed scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

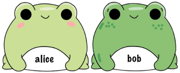
- $\mathsf{Gen}(1^n)$: Run $\mathsf{Gen}_1(1^n) \to (\mathsf{pk}_1, \mathsf{sk}_1)$ and $\mathsf{Gen}_2(1^n) \to (\mathsf{pk}_2, \mathsf{sk}_2)$. Return $((\mathsf{pk}_1, \mathsf{pk}_2), (\mathsf{sk}_1, \mathsf{sk}_2))$.
- $\mathsf{Enc}((\mathsf{pk}_1, \mathsf{pk}_2), m)$: Return $c = \mathsf{Enc}_1(\mathsf{pk}_1, \mathsf{Enc}_2(\mathsf{pk}_2, m))$.
- $\mathsf{Dec}((\mathsf{sk}_1, \mathsf{sk}_2), c)$: Return $m' = \mathsf{Dec}_2(\mathsf{sk}_2, \mathsf{Dec}_1(\mathsf{sk}_1, c))$

Question: Prove that if Π_1 is CPA-secure or Π_2 is CPA-secure, then Π is CPA-secure.



Question 4: Encryption Combiner

Use $\mathcal{A}$ to construct an adversary $\mathcal{B}_1$ for the CPA game for Π_1. $\mathcal{B}_1$ should win the CPA game for Π_1 with the same probability that $\mathcal{A}$ wins the CPA game for Π.



Question 4: Encryption Combiner

Use $\mathcal{A}$ to construct an adversary $\mathcal{B}_2$ for the CPA game for Π_2. $\mathcal{B}_2$ should win the CPA game for Π_2 with the same probability that $\mathcal{A}$ wins the CPA game for Π.

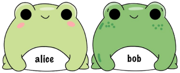
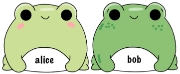


Table of Contents



Example Answer

Most people had very similar answers and errors in both parts.

To construct an adversary $\mathcal{B}_1$, do the following:

- Whenever A makes a query m to the encryption oracle, send $Enc_2(m)$ to the B_1 oracle and respond with the output $Enc_1(Enc_2(m))$.
- Get the two queries m_0, m_1 from A and send $Enc_2(m_0)$ and $Enc_2(m_1)$ to the challenger to get $Enc_1(Enc_2(m_b))$.
- Output whatever A outputs.

Comments:

- The main ideas in this proof are correct – constructing the correct responses that matches what A expects to receive and using it to break CPA security.
- There are two main issues here that need to be fixed for full credit:
    - The key generation is not described – The challenger for B_1 passes pk_1 to B_1 and B_1 must itself sample pk_2 and pass $(\mathrm{pk}_1, \mathrm{pk}_2)$ to A.
    - Encryption queries do *not* have to be simulated – since this is PKE, anyone can encrypt messages when given the public key for the scheme.