National University of Singapore

CS4236: Cryptography Theory and Practice

FINAL ASSESSMENT

Semester 1, 2024/2025

Time allowed: 2 hours **Maximum score:** 40

INSTRUCTIONS FOR STUDENTS

1. Write down your **Student Number** on the answer sheet and shade **completely** the corresponding bubbles in the grid for each digit or letter. **Do not write your name.**
2. This question paper contains **THREE (3) sections** containing multiple problems each, and comprises **SIX (6) pages** including this cover page.
3. This is a **closed book** assessment. You are allowed to bring one A4-sized double-sided cheatsheet, and nothing else. No calculators, log tables, set-squares, etc..
4. You must submit only **one answer sheet** and no other documents. All questions must be answered in the space provided on the answer sheet; no extra sheets will be accepted as answers. Please be aware of this limitation in space and manage your writing accordingly.
5. Marks may be deducted for unrecognisable handwriting and/or for not shading the student number properly. You can use pencils as long as all writing and shading is clear.
6. An excerpt of the question may be provided in the answer sheet to aid you in answering in the correct box, where applicable. It is not the exact question. You should still refer to the original question in this question booklet.
7. Whenever a problem asks you to "prove" something, a formal mathematical proof is required in support of your answer. If it only asks you to "explain" or "justify", a convincing supporting argument is sufficient.
8. When asked to show a counterexample or to construct an adversary against a scheme, you should clearly describe the counterexample or adversary (ideally with pseudocode), and provide a supporting argument for why the counterexample or adversary works.
9. Any algorithms (constructions or adversaries) in your answers should be described using **clear pseudocode**. Unclear descriptions and proofs may not be given full credit.
10. When asked to define the security properties of a cryptographic primitive, you are required to do so formally. That is, define the relevant security game, clearly stating the actions taken by the Challenger and Adversary and the winning condition for the Adversary, and then state the condition on the Adversary's probability of winning that defines the security property.

This page is intentionally left blank.

It may be used as scratch paper.

Question 1: Composing Primitives [12 marks]

In the following problems, you need not provide a complete proof, but you need to provide a sufficiently convincing argument (or proof sketch) in support of your answers. Throughout, the symbol "||" denotes concatenation, "$\oplus$" denotes bitwise XOR, and "$\wedge$" denotes bitwise AND.

A. Suppose $G : \{0,1\}^\lambda \to \{0,1\}^{4\lambda}$ is a PRG. In each of the following cases, determine whether the function described is a PRG, and justify your answer. Below, x_1, and x_2 are each of length λ. [4 marks]

(a) $G_1(x_1||x_2) = (G(x_1) \oplus G(x_2))$

(b) $G_2(x_1||x_2) = (G(x_1) \wedge G(x_2))$

Recall that a One-Way Function (OWF) is a function that can be evaluated in polynomial-time, but any polynomial-time algorithm trying to invert it has negligible success probability (this probability is over the randomly chosen input and the randomness of the algorithm). In your answers to the following questions, if needed, you may assume that for any λ and $\lambda' \geq \sqrt{\lambda}$, there exists a OWF $f : \{0,1\}^\lambda \to \{0,1\}^{\lambda'}$.

B. Suppose $g : \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a OWF. In each of the following, determine whether the function described is a OWF, and justify your answer. Below, x, x_1, and x_2 are of length λ. [4 marks]

(a) $h_1(x_1||x_2) = g(x_1)$

(b) $h_2(x) = (\text{first } \lambda/2 \text{ bits of } g(x))$

C. Consider keyless hash functions $H_1 : \{0,1\}^n \to \{0,1\}^{n/4}$ and $H_2 : \{0,1\}^n \to \{0,1\}^{n/4}$. Suppose you are guaranteed that H_1 is collision-resistant (but H_2 might not be). In each of the following cases, determine whether the constructed hash function is necessarily collision-resistant. Justify your answers.

[4 marks]

(a) $H : \{0,1\}^n \to \{0,1\}^{n/2}$ defined as: $H(x) = H_1(x)||H_2(x)$

(b) $H' : \{0,1\}^{2n} \to \{0,1\}^{n/4}$ defined as: $H'(x_1||x_2) = H_1(x_1) \oplus H_2(x_2)$

Question 2: Weak PRFs [17 marks]

Generally, we define function families using the following two algorithms:

- $\mathsf{Gen}(\lambda)$: randomized function that outputs a key k (that implicitly describes function f_k)
- $\mathsf{Eval}(\lambda, k, x)$: deterministic function that outputs y (that is defined to be $f_k(x)$)

Throughout this section, we take the functions (f_k, g_k, etc.) to have input domain and output co-domain $\{0,1\}^\lambda$, where λ is the security parameter for which the key k was generated by $\mathsf{Gen}(\lambda)$.

Recall that the security of a *Pseudo-Random Function (PRF)* family is described as follows:

An efficient adversary that is given outputs of a random function from the family, on inputs of its choice, cannot distinguish the function from a uniformly random function

A. Write the formal definition of the security of a PRF. [2 marks]

In contrast, a *weak PRF* family is only required to have a similar property when the adversary, instead of being able to query the function of inputs of its choice, is given the outputs for random inputs. Its security described as follows:

An efficient adversary that is given input-output pairs of a random function from the family, with inputs chosen uniformly at random, cannot distinguish the function from a uniformly random function

B. Write the formal definition of the security of a weak PRF. [3 marks]

For the following questions, assume you have a PRF family $F = \left\{ f_k : \{0,1\}^\lambda \to \{0,1\}^\lambda \right\}$ defined by the algorithms $(\mathsf{Gen}_1, \mathsf{Eval}_1)$.

C. Construct a family of functions $(\mathsf{Gen}_2, \mathsf{Eval}_2)$ that is a weak PRF but not a PRF. You will have to describe the functions Gen_2 and Eval_2, demonstrate an attack that shows that your construction is not a PRF family, and present a convincing argument for it being a weak PRF family. [4 marks]

In class, we constructed a CPA-secure symmetric-key encryption scheme using a PRF family. Here, you will attempt to do the same with a weak PRF family instead. Suppose you have weak PRF family $G = \left\{ g_k : \{0,1\}^\lambda \to \{0,1\}^\lambda \right\}$ defined by the algorithms $(\mathsf{Gen}_3, \mathsf{Eval}_3)$. The following encryption scheme, with security parameter λ, should be designed to support the message space $\{0,1\}^\lambda$.

- $\mathsf{Gen}_E(\lambda)$: Output $k \leftarrow \mathsf{Gen}_3(\lambda)$
- $\mathsf{Enc}(\lambda, k, m)$: ____ ; output ciphertext c
- $\mathsf{Dec}(\lambda, k, c)$: ____ ; output message m

D. Complete the Enc and Dec algorithms in the above template in such a way that the resulting encryption scheme satisfies correctness and CPA-security if $(\mathsf{Gen}_3, \mathsf{Eval}_3)$ is a weak PRF family. Explain why your construction is CPA-secure if $(\mathsf{Gen}_3, \mathsf{Eval}_3)$ is a weak PRF family. You need not provide a full security reduction. [4 marks]

Next, we will try to do the same with MAC schemes. With a weak PRF family as above, consider the following MAC scheme that supports messages that are strings of length λ.

- $\mathsf{Gen}_M(\lambda)$: Output $k \leftarrow \mathsf{Gen}_3(\lambda)$
- $\mathsf{MAC}_1(\lambda, k, m)$: Output tag $t \leftarrow \mathsf{Eval}_3(\lambda, k, m)$
- $\mathsf{Verify}_1(\lambda, k, m, t)$: Accept if $t = \mathsf{Eval}_3(\lambda, k, m)$; Reject otherwise

E. Is the MAC scheme $(\mathsf{Gen}_M, \mathsf{MAC}_1, \mathsf{Verify}_1)$ EUF-CMA secure if the family $(\mathsf{Gen}_3, \mathsf{Eval}_3)$ is a weak PRF? If yes, provide a justifying argument for this security. If no, demonstrate an instantiation of the family $(\mathsf{Gen}_3, \mathsf{Eval}_3)$ that is a weak PRF, together with an attack against the above MAC construction that breaks the EUF-CMA security. [2 marks]

Next, consider the following modified construction of a MAC scheme.

- $\mathsf{Gen}_M(\lambda)$: Output $k \leftarrow \mathsf{Gen}_3(\lambda)$
- $\mathsf{MAC}_2(\lambda, k, m)$: Sample $x \leftarrow \{0,1\}^\lambda$; Compute $t \leftarrow \mathsf{Eval}_3(\lambda, k, x)$; Output tag $(m \oplus x, t)$
- $\mathsf{Verify}_2(\lambda, k, m, (a, b))$: Accept if $b = \mathsf{Eval}_3(\lambda, k, m \oplus a)$; Reject otherwise

F. Is the MAC scheme $(\mathsf{Gen}_M, \mathsf{MAC}_2, \mathsf{Verify}_2)$ EUF-CMA secure if the family $(\mathsf{Gen}_3, \mathsf{Eval}_3)$ is a weak PRF? If yes, provide a justifying argument for this security. If no, demonstrate an instantiation of the family $(\mathsf{Gen}_3, \mathsf{Eval}_3)$ that is a weak PRF, together with an attack against the above MAC construction that breaks the EUF-CMA security. [2 marks]

Question 3: Computational Diffie-Hellman [11 marks]

Throughout this section, we will use the following sets:

- For any $\lambda \in \mathbb{N}$, $PRIMES_\lambda$ is the set of all prime numbers that can be written using λ bits
- For any prime number p, GEN_p is the set of all generators of $\mathbb{Z}_p^*$

A. State the Computational Diffie-Hellman (CDH) assumption for multiplicative groups modulo random prime numbers. (That is, the CDH assumption as stated in the lectures.) [2 marks]

For brevity, we will refer to the CDH assumption over the above groups as simply "the CDH assumption", without additionally specifying the group each time.

The ElGamal encryption scheme, when instantiated with modular multiplicative groups $\mathbb{Z}_p^*$, is described as follows, and supports messages that are also elements in $\mathbb{Z}_p^*$. (Below, note that all multiplication and inversion operations on elements of $\mathbb{Z}_p^*$ are performed modulo p.)

$\underline{\mathsf{Gen}(\lambda)}$:

- Sample $p \leftarrow PRIMES_p$, $g \leftarrow GEN_p$, and $x \leftarrow [p-1]$
- Output $pk = (p, g, g^x)$, and $sk = x$

$\underline{\mathsf{Enc}(\lambda, (p, g, g^x), m)}$:

- Sample $y \leftarrow [p-1]$
- Output $(c_1, c_2) = (g^y, (g^x)^y \cdot m)$

$\underline{\mathsf{Dec}(\lambda, x, (c_1, c_2))}$:

- Compute $z \leftarrow (c_1)^x$
- Output $m = z^{-1} \cdot c_2$

We showed in class that if the *Decisional* Diffie-Hellman (DDH) assumption was true in the group used by the ElGamal encryption scheme, then it is CPA-secure. However, as we know, the DDH assumption is not true for $\mathbb{Z}_p^*$, though the CDH assumption might be true.

B. Suppose the CDH assumption is true. Explain why it would still be insufficient to show CPA security of the above ElGamal encryption scheme. [2 marks]

There are a couple of ways in which the above scheme can be modified to make it secure under just the CDH assumption, without needing DDH. One of these is to use a random oracle. For the rest of the problem, assume that all parties have access to a random oracle $H_\lambda : \{0,1\}^\lambda \to \{0,1\}^\lambda$ for the chosen security parameter λ. Note that if p can be represented by a λ-bit string, then every element in $\mathbb{Z}_p^*$ can also be represented by a λ-bit string.

The above scheme can then be modified as follows into an encryption scheme that supports messages from $\{0,1\}^\lambda$.

$\mathsf{Gen}(\lambda)$:

- Sample $p \leftarrow PRIMES_p$, $g \leftarrow GEN_p$, and $x \leftarrow [p-1]$
- Output $pk = (p, g, g^x)$, and $sk = x$

$\mathsf{Enc}^H(\lambda, (p,g,g^x), m)$:

- Sample $y \leftarrow [p-1]$
- Output $(c_1, c_2) = (\_\_, \_\_)$

$\mathsf{Dec}^H(\lambda, (g,x), (c_1,c_2))$:

- Compute $z \leftarrow (c_1)^x$
- Output $m = \_\_$

C. Complete the Enc^H and Dec^H algorithms above in such a way that the resulting encryption scheme is correct and CPA-secure (in the random oracle model) if the CDH assumption is true. Explain why your construction is CPA-secure. [3 marks]

D. Prove rigorously that the encryption scheme $(\mathsf{Gen}, \mathsf{Enc}^H, \mathsf{Dec}^H)$ you have designed above is CPA-secure (in the random oracle model) if the CDH assumption is true. [4 marks]

— END OF QUESTION PAPER —

CS4236—Cryptography Theory and Practice

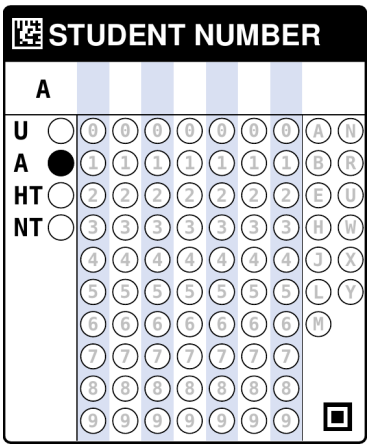
Final Assessment Answer Sheet

Semester 1, 2024/2025

Time allowed: 2 hours

Instructions (please read carefully):

1. Write down your **Student Number** on the answer sheet and shade **completely** the corresponding bubbles in the grid for each digit or letter. **Do not write your name.**
2. This answer booklet comprises **TWELVE (12) pages**, including this cover page.
3. This is a **closed-book** assessment. You are allow one A4-sized double-sided cheatsheet.
4. Weightage of questions is given in square brackets. The maximum attainable score is 40.
5. You must submit only **this answer sheet** and no other documents. All questions must be answered in the space provided on the answer sheet; no extra sheets will be accepted as answers. Please be aware of this limitation in space and manage your writing accordingly.
6. Marks may be deducted for unrecognisable handwriting and/or for not shading the student number properly.
7. An excerpt of the question may be provided to aid you in answering in the correct box, where applicable. It is not the exact question. You should still refer to the original question in the question booklet.
8. Whenever a problem asks you to "prove" something, a formal mathematical proof is required in support of your answer. If it only asks you to "explain" or "justify", a convincing supporting argument is sufficient.
9. Any algorithms (constructions or adversaries) in your answers should be described using **clear pseudocode**. Unclear descriptions and proofs may not be given full credit.



For Examiner's Use Only

Question	Marks
Q1	/ 12
Q2	/ 17
Q3	/ 11
Total	/ 40

Question 1A Composing PRG's [4 marks]

Question 1B Composing OWF's [4 marks]

Question 1C Composing CRHF's [4 marks]

Question 2A PRF Security [2 marks]

Question 2B Weak PRF Security [3 marks]

[**Common mistake 1:** Challenger also needs to return the random inputs it samples.]

[**Common mistake 2:** A random function is a deterministic function sampled uniformly at random. It is not a randomized function sampling every output uniformly at random.]

Question 2C Weak PRF but not PRF [4 marks]

Question 2D SKE from weak PRF [4 marks]

Question 2E MAC from Weak PRF 1 [2 marks]

Question 2F MAC from Weak PRF 2 [2 marks]

Question 3A CDH assumption [2 marks]

Question 3B Insecurity Explanation [2 marks]

Question 3C CPA-secure modification [3 marks]

Question 3D Proof of CPA security [4 marks]

Question 1A Composing PRG's [4 marks]

(a) G_1 is a PRG. Since, in the pseudorandomness definition for the PRG, x_1 and x_2 are taken to be uniformly random independently of each other, $G(x_1)$ is pseudorandom irrespective of the value of x_2. So $G(x_1) \oplus G(x_2)$ is also pseudorandom, as $G(x_1)$ acts as a pseudorandom mask.

[**Common mistake 1:** In the definition of PRG, the adversary is not allowed to pick the input. Instead, it is given a single output of a randomly sampled input.]

(b) G_2 is not a PRG. Consider, for example, the first bit of the output. Since G is a PRG, the first bit of its output has to be almost unbiased. But the AND of two unbiased bits is biased significantly towards 0, and so just checking whether the first bit is 0 can distinguish between the output of G_2 and a random string with constant advantage.

Question 1B Composing OWF's [4 marks]

(a) h_1 is a OWF. Any adversary that inverts h_1 also, in the process, inverts g, as the first half of the inverse for h_1 is precisely an inverse for g with the same output.

[**Common mistake 1:** Misunderstanding the definition of OWF: A is supposed to find an x' such that f(x') = f(x), x' doesn't have to be x.]

(b) h_2 might not be a OWF. Consider, for example, a OWF g such that its first $\lambda/2$ output bits are always 0 – this can be constructed by taking another OWF with a shorter output and just pre-pending 0's to the output. For such a g, the output of h_2 will be always $0^{\lambda/2}$, and this can be trivially inverted. But g is clearly a OWF, otherwise the internal OWF will be broken (contradiction).

Question 1C Composing CRHF's [4 marks]

(a) H is a collision-resistant. Any valid collision for H is also a collision for H_1. So if there is an efficient adversary that can find collisions for H, it would also find collisions for H_1, which is a contradiction.

(b) H' need not be collision-resistant. Any collision (x_2, x_2') for H_2 can be used to construct a collision $(x_1||x_2, x_1||x_2')$ for H', using any x_1. So if H_2 is not collision-resistant, H' is not either.

Question 2A PRF Security [2 marks]

PRF-IND(λ):

1. Challenger samples bit $b \leftarrow \{0,1\}$.
   - If $b = 0$, set g to be a uniformly random function from $\{0,1\}^\lambda$ to $\{0,1\}^\lambda$
   - If $b = 1$, sample $k \leftarrow \mathsf{Gen}(\lambda)$ and set $g(\cdot) \equiv \mathsf{Eval}(\lambda, k, \cdot)$
2. Repeat until Adversary stops making queries:
   - Adversary sends query $x \in \{0,1\}^\lambda$ to Challenger
   - Challenger responds with $g(x)$
3. Adversary outputs bit b'
4. Adversary wins if $b' = b$

The function family defined by $(\mathsf{Gen}, \mathsf{Eval})$ is a PRF if, for any polynomial-time adversary Adv, there is a negligible function ν such that:

$$\Pr[Adv(\lambda) \text{ wins PRF-IND}(\lambda)] \leq \frac{1}{2} + \nu(\lambda)$$

Question 2B Weak PRF Security [3 marks]

wPRF-IND(λ):

1. Challenger samples bit $b \leftarrow \{0,1\}$.
   - If $b = 0$, set g to be a uniformly random function from $\{0,1\}^\lambda$ to $\{0,1\}^\lambda$
   - If $b = 1$, sample $k \leftarrow \mathsf{Gen}(\lambda)$ and set $g(\cdot) \equiv \mathsf{Eval}(\lambda, k, \cdot)$
2. Repeat until Adversary stops making queries:
   - Adversary makes a query for an input-output pair to Challenger
   - Challenger samples $x \leftarrow \{0,1\}^\lambda$ and responds with $(x, g(x))$
3. Adversary outputs bit b'
4. Adversary wins if $b' = b$

The function family defined by $(\mathsf{Gen}, \mathsf{Eval})$ is a weak PRF if, for any polynomial-time adversary Adv, there is a negligible function ν such that:

$$\Pr[Adv(\lambda) \text{ wins wPRF-IND}(\lambda)] \leq \frac{1}{2} + \nu(\lambda)$$

[**Common mistake 1:** Challenger also needs to return the random inputs it samples.]

[**Common mistake 2:** A random function is a deterministic function sampled uniformly at random. It is not a randomized function sampling every output uniformly at random.]

Question 2C Weak PRF but not PRF [4 marks]

(Various solutions are possible. The following is one of the simpler ones.)

Define the candidate weak PRF as follows:

- $\mathsf{Gen}_2(\lambda)$: output $k \leftarrow \mathsf{Gen}_1(\lambda)$
- $\mathsf{Eval}_2(\lambda, k, x)$: if $x = 0^\lambda$, output 0^λ, else output $\mathsf{Eval}_1(\lambda, k, x)$

Consider the adversary for the PRF-IND game that makes a query for input $x = 0^\lambda$, and outputs $b' = 1$ if the result is $g(x) = 0^\lambda$, and outputs $b' = 0$ otherwise. When $b = 1$ and the function g is from $(\mathsf{Gen}_2, \mathsf{Eval}_2)$, the event $g(x) = 0^\lambda$ happens with probability 1 and so $b' = 1$ with probability 1. When $b = 0$ and g is a random function, this happens with probability $1/2^\lambda$ and otherwise $b' = 0$. So the probability that this efficient adversary wins is $(1 - 1/2^{\lambda+1})$, and so the above is not a PRF.

In the wPRF-IND game with $(\mathsf{Gen}_2, \mathsf{Eval}_2)$, the adversary is only given random $(x, g(x))$ samples. If none of the samples it receives has $x = 0^\lambda$, then the game looks identical to the adversary to the PRF-IND game with $(\mathsf{Gen}_1, \mathsf{Eval}_1)$, in which we know that no efficient adversary can win with non-negligible advantage. Since the wPRF-IND adversary gets at most $\mathrm{poly}(\lambda)$ samples, the probability that at least one of them has $x = 0^\lambda$ is at most $\mathrm{poly}(\lambda)/2^\lambda$. Thus, the advantage of any efficient wPRF-IND adversary is at most $(\mathrm{negl}(\lambda) + \mathrm{poly}(\lambda)/2^\lambda)$, which is also negligible.

Question 2D SKE from weak PRF [4 marks]

The construction we saw in class still works with weak PRFs.

- $\mathsf{Gen}_E(\lambda)$: Output $k \leftarrow \mathsf{Gen}_3(\lambda)$
- $\mathsf{Enc}(\lambda, k, m)$: sample $x \leftarrow \{0,1\}^\lambda$, and compute $y \leftarrow \mathsf{Eval}(\lambda, k, x)$; output ciphertext $c = (x, y \oplus m)$
- $\mathsf{Dec}(\lambda, k, c)$: given $c = (x, z)$, compute $y \leftarrow \mathsf{Eval}(\lambda, k, x)$; output message $m = z \oplus y$

Consider a CPA adversary that has access to a polynomial number of ciphertexts generated using some secret key of this scheme, for messages of its choice. That is, it has the tuples $(x_i, f_k(x_i) \oplus m_i)$ for m_i's of its choice and where the x_i's are uniformly random and k is randomly generated and hidden from it. Essentially the only information this reveals to the adversary is the set of tuples $(x_i, f_k(x_i))$, which are simply the evaluations of the function f_k at random inputs. According to the property of weak PRF's, this information is insufficient to distinguish such an f_k from a random function. So, except with some negligible probability with which the x_i's might be repeated, these tuples are indistinguishable from (x_i, y_i) where the y_i's are uniformly random and thus carry no information about k. With just the (x_i, y_i)'s, the adversary cannot distinguish between ciphertexts of any two messages, giving CPA security.

Question 2E MAC from Weak PRF 1 [2 marks]

This MAC construction is not secure. Consider the weak PRF construction from the solution to part **C**. If that were used in the construction here, the tag for the message $m = 0^\lambda$ is always $t = 0^\lambda$, irrespective of the key. So the valid message-tag pair $(0^\lambda, 0^\lambda)$ can always be generated efficiently.

Question 2F MAC from Weak PRF 2 [2 marks]

This MAC construction is not secure, even if the function family used is a strong PRF. Querying the tag for the message 0^k would give the adversary an (x,t) such that $t = \mathsf{Eval}_3(\lambda, k, x)$. Then, for any message m, the tag $(m \oplus x, t)$ would be a valid tag, and thus can be forged easily.

Question 3A CDH assumption [2 marks]

For any polynomial-time algorithm A, there is a negligible function ν such that for all λ:

$$\Pr_{p \leftarrow PRIMES_\lambda, g \leftarrow GEN_p, x,y \leftarrow [p-1]} [A(p,g,g^x,g^y) = g^{xy}] \leq \nu(\lambda)$$

Question 3B Insecurity Explanation [2 marks]

The fact that g^{xy} cannot be computed does not necessarily mean that $g^{xy} \cdot m_0$ and $g^{xy} \cdot m_1$ cannot be distinguished. It could be possible, for instance, to be able to learn the first bit of m from $g^{xy} \cdot m$ without contradicting CDH, and this would enable distinguishing between encrpytions of messages that differed on that bit.

Question 3C CPA-secure modification [3 marks]

- $(c_1, c_2) = (g^y, H((g^x)^y) \oplus m)$
- $m = H(z) \oplus c_2$

If H is a random oracle, then to an adversary that does not know g^{xy}, the string $H(g^{xy})$ looks completely random. By the CDH assumption, the CPA adversary that learns just the public-key (p, g, g^x) and the first part of the ciphertext g^y cannot compute g^{xy}, and so $H(g^{xy})$ looks completely random to it, and can thus function as a one-time pad, making the second half of the ciphertext $H(g^{xy}) \oplus m$ look random irrespective of m.

Question 3D Proof of CPA security [4 marks]

(Detailed proof of security needed for full marks. Partial marks awarded for correct initial set up and other important ideas being present.)