National University of Singapore

CS4236: Cryptography Theory and Practice

FINAL ASSESSMENT

Semester 1, 2023/2024

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.
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.
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" or "show" 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.

This page is intentionally left blank.

It may be used as scratch paper.

Question 1: Short Problems [22 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.

A. Suppose $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is a PRG. In each of the following cases, determine whether the function described is a PRG, and justify your answer. Below, x, x_1, and x_2 are all of length λ. [6 marks]

(a) $G_1(x) = (x||G(x))$

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

(c) $G_3(x_1||x_2) = (G(x_1)||G(x_2))$

B. Consider the following unsuccessful attempts to build a PRF from a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$. In each case, show how to break the pseudorandomness of the family constructed. [4 marks]

(a) $F_1 = \left\{ f_k : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda} \right\}$, where $f_k(x) = G(k) \oplus x$

(b) $F_2 = \left\{ f_k : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda} \right\}$, where $f_k(x) = G(k) \wedge x$

(the symbol $\oplus$ here denotes the bitwise-XOR operation, and $\wedge$ denotes bitwise-AND)

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 a 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 $\lambda' \geq \lambda$, there exists a OWF $f : \{0,1\}^\lambda \to \{0,1\}^{\lambda'}$.

C. If $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is a PRG, is it also necessarily a OWF? Justify your answer. [4 marks]

D. If $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is a OWF, is it also necessarily a PRG? Justify your answer. [4 marks]

E. 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 at least one of H_1 and H_2 is collision-resistant (but possibly not both). 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) $H' : \{0,1\}^{2n} \to \{0,1\}^{n/2}$ defined as: $H'(x_1||x_2) = H_1(x_1)||H_2(x_2)$

Question 2: Random Unforgeability [12 marks]

In this problem, we will define and analyse digital signature schemes that satisfy a weaker security guarantee than Existential Unforgeability (EUF). This property, called *Random Unforgeability (RUF)*, is described informally as follows:

An adversary, even knowing the public verification key, cannot forge a signature on a uniformly random message that is given to it

Crucially, note that the adversary does not pick the message it tries to forge a signature for. This message is chosen at random from the message domain M and given to the adversary.

A. Define formally the RUF security of a signature scheme $(Gen, Sign, Verify)$. Your definition should describe clearly the security game, and the defining condition on the adversary's probability of winning the game. [2 marks]

Random Unforgeability under Chosen Message Attacks (RUF-CMA) is similar to RUF security, but here the adversary is allowed to see signatures on messages of its choice (that is, perform a chosen-message attack). This is described informally as follows:

An adversary, even knowing the public verification key and with the ability to obtain signatures on messages of its choice, cannot forge a new signature on a uniformly random message that is given to it

Above, the adversary has the ability to the obtain signatures on messages both before and after it is given the uniformly random message that it is supposed to forge a new signature for.

B. Define formally the RUF-CMA security of a signature scheme $(Gen, Sign, Verify)$. Your definition should describe clearly the security game, and the defining condition on the adversary's probability of winning the game.

[3 marks]

Recall the *plain RSA* signature scheme described as follows, with the message domain $\mathbb{Z}_N^*$. The various symbols used below are as defined in the lectures.

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

- Sample $p, q \leftarrow PRIMES_\lambda$
- Set $N \leftarrow pq$
- Pick e such that $gcd(e, \phi(N)) = 1$
- Compute $d \leftarrow e^{-1} (mod\ N)$
- Output $vk = (N, e)$, and $sk = (N, d)$

$\underline{Sign(\lambda, (N, d), m)}$:

- Output $\sigma \leftarrow m^d (mod\ N)$

$\underline{Verify(\lambda, (N,e), m, \sigma):}$

- Accept iff $m = \sigma^e (mod\ N)$

C. State the RSA assumption. [2 marks]

D. Suppose the RSA assumption is true. Prove that the plain RSA signature scheme is RUF-secure.

[2 marks]

E. Show that the plain RSA signature scheme is *not* RUF-CMA secure. [3 marks]

Note that if your answer to part **A** (respectively part **B**) is incorrect, it might not be possible for you to get full marks for part **D** (resp. part **E**). You can still get partial marks if your approach in these latter parts is sound.

Question 3: Chosen-Ciphertext Security for PKE [6 marks]

In this problem, we will look at a stronger notion of security for Public-Key Encryption (PKE) schemes that was mentioned briefly in class, namely security against chosen ciphertext attacks. This property, called *CCA-security*, is described informally as follows:

An adversary, even knowing the public key and with access to a decryption oracle that decrypts any ciphertext of its choice, still cannot learn anything about the message encrypted in a ciphertext that it did not query the oracle with

Above, the adversary has the ability to obtain decryptions of ciphertexts both before and after it sees the special ciphertext whose message it wants to learn. We defined in class the analogous notion for symmetric-key encryption.

A. Define formally the CCA-security of a PKE scheme (Gen, Enc, Dec). Your definition should describe clearly the security game, and the defining condition on the adversary's probability of winning the game.

[3 marks]

Recall that the ElGamal Encryption scheme over a group G of order q with set of generators $GEN(G)$ is defined as follows.

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

- Sample $g \leftarrow GEN(G)$ and $x \leftarrow [q]$
- Output $pk = (g, g^x)$, and $sk = (g, x)$

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

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

<u>$Dec(\lambda, (g, x), (c_1, c_2))$:</u>

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

B. Suppose the Decisional Diffie-Hellman assumption is true in group G. Is the ElGamal Encryption scheme CCA-secure? If so, prove this. If not, construct an adversary that breaks the CCA security. [3 marks]

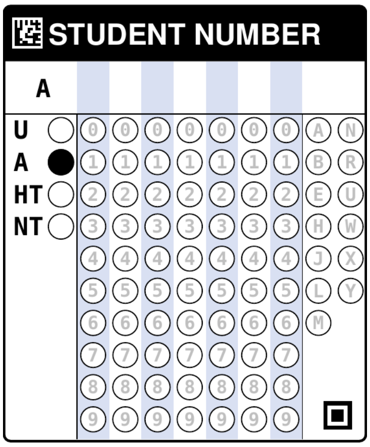
— END OF QUESTION PAPER —

CS4236—Cryptography Theory and Practice

Final Assessment Answer Sheet

Semester 1, 2023/2024

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 **FOURTEEN (14) 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" or "show" 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	/ 22
Q2	/ 12
Q3	/ 6
Total	/ 40

Question 1A Constructing PRG's [6 marks]

Question 1B Breaking PRF's [4 marks]

Question 1C Is a PRG a OWF? [4 marks]

Question 1D Is a OWF a PRG? [4 marks]

Question 1E Composing CRHF's [4 marks]

Question 2A RUF definition [2 marks]

Question 2B RUF-CMA definition [3 marks]

Question 2C RSA assumption [2 marks]

Question 2D Plain RSA is RUF-secure [2 marks]

Question 2E Plain RSA is not RUF-CMA secure [3 marks]

Question 3A CCA security definition [3 marks]

Question 3B CCA-security of ElGamal [3 marks]

This page is intentionally left blank.

It may be used as scratch paper.

— END OF ANSWER SHEET —

Question 1A Constructing PRG's [6 marks]

(a) G_1 is not a PRG. The distinguisher can evaluate $G(x)$ using the x in the PRG output and see whether the rest of the output is consistent with this.

(b) G_2 is a PRG. When x_2 is uniformly random and independent of x_1, the security of G ensures that $G(x_2)$ is pseudorandom. So for random and independent x_1 and x_2, $(x_1||G(x_2))$ is pseudorandom.

(c) G_3 is a PRG. The argument is similar to the previous part, and this can be proven formally using a hyrid argument, showing that $(G(x_1)||G(x_2))$ is indistinguishable from $(y||G(x_2))$ for a random y, which is in turn indistinguishable from $(y||z)$ for random and independent y and z.

Question 1B Breaking PRF's [4 marks]

(a) There are several attacks possible using the simple linear dependence on the input. For instance, note that $f_k(x_1) \oplus f_k(x_2) = x_1 \oplus x_2$, which only happens in a random function with exponentially small probability. The distinguisher can query the function on any two different inputs and check whether this holds.

(b) Here, $f_k(0^{2\lambda})$ is always $0^{2\lambda}$, which happens for a random function only with exponentially small probability. The distinguisher can just check whether this happens.

Question 1C Is a PRG a OWF? [4 marks]

G is also a OWF. Suppose it is not. Then, there exists an adversary A that inverts it with non-negligible probability. The output of G can then be distinguished from random as follows: given y, runs $A(y)$ to get x; if $G(x) = y$, output 1, else output 0. If y was truly random, the probability that such an x even exists is exponentially small. On the other hand, if y came from G, then A will find this x with non-negligible probability. Thus, the distinguisher outputs 1 with non-negligibly different probabilities in the two cases.

Question 1D Is a OWF a PRG? [4 marks]

G is not necessarily a PRG. For instance, consider any OWF $f : \{0,1\}^\lambda \to \{0,1\}^\lambda$, and set $G(x) = (f(x)||0^\lambda)$. G is still one-way, as inverting G amounts to inverting f. But it is clearly distinguishable from random.

Question 1E Composing CRHF's [4 marks]

(a) H is collision-resistant. Any collision for H is a collision for both H_1 and H_2. So any algorithm that finds a collision for H will also find collisions for both of these. But this contradicts the hypothesis that at least one of them is collision-resistant.

(b) H' may not be collision-resistant. Any collision (x_1, x_1') for H_1 can be used to construct the collision $(x_1, 0^\ell)$ and $(x_1', 0^\ell)$ for H'. Collisions for H_2 may also be used similarly.

Question 2A RUF definition [2 marks]

The RUF security game for signature scheme $(Gen, Sign, Verify)$ for messages in a domain M is defined as follows, with Challenger C and adversary A:

$\underline{RUF(\lambda)}$:

1. C samples $(sk, vk) \leftarrow Gen(\lambda)$, and sends vk to A
2. C samples a random $m \leftarrow M$ and sends it to A
3. A outputs signature σ
4. A wins if $Verify(\lambda, m, \sigma)$ accepts

The signature scheme is RUF-secure if:

$$\forall \text{ PPT } A \ \exists \text{ negligible function } \nu : \Pr[A \text{ wins } RUF(\lambda)] < \nu(\lambda)$$

Question 2B RUF-CMA definition [3 marks]

The RUF-CMA security game for signature scheme $(Gen, Sign, Verify)$ for messages in a domain M is defined as follows, with Challenger C and adversary A:

$\underline{RUF\text{-}CMA(\lambda)}$:

1. C samples $(sk, vk) \leftarrow Gen(\lambda)$, and sends vk to A
2. Repeat until A stops sending queries:
   - A sends message m' to C
   - C computes $\sigma' \leftarrow Sign(\lambda, sk, m')$ and sends it to A
3. C samples a random $m \leftarrow M$ and sends it to A
4. Repeat until A stops sending queries:
   - A sends message m' to C
   - C computes $\sigma' \leftarrow Sign(\lambda, sk, m')$ and sends it to A
5. A outputs signature σ
6. A wins if $Verify(\lambda, m, \sigma)$ accepts *and* (m, σ) is different from all of the (m', σ') seen in the queries by the adversary.

The signature scheme is RUF-CMA-secure if:

$$\forall \text{ PPT } A \ \exists \text{ negligible function } \nu : \Pr[A \text{ wins } RUF\text{-}CMA(\lambda)] < \nu(\lambda)$$

Question 2C RSA assumption [2 marks]

For every PPT A, there exists a negligible function ν such that: with $p, q \leftarrow PRIMES_\lambda$, $N \leftarrow pq$, for any e such that $gcd(e, \phi(N)) = 1$, and $a \leftarrow \mathbb{Z}_N^*$, we have:

$$\Pr\left[A(N, e, a^e (mod\ N)) = a\right] < \nu(\lambda)$$

where the probability is over the choices of p, q, a, and any internal randomness of A.

Question 2D Plain RSA is RUF-secure [2 marks]

In the RUF game, when instantiated with the plain RSA signature scheme, the challenger picks a random (N,e) in *Gen* as the verification key and a random message $m \leftarrow \mathbb{Z}_N^*$, and sends these to the adversary. The adversary then wins if it can generate a signature that passes verification – that is, if it can find a σ such that $m = \sigma^e (mod\ N)$. Note that this σ is unique, and is given by $m^d (mod\ N)$. So if m is a uniformly random element in $\mathbb{Z}_N^*$, then so is $\sigma = m^d (mod\ N)$. Thus, the adversary's task is, given $(N, e, \sigma^e (mod\ N))$, where (N,e) is picked as in *Gen* and σ is uniformly random over $\mathbb{Z}_N^*$, to find σ. This is exactly what the RSA assumption says is not possible to do when N. So under the RSA assumption, no polynomial-time adversary can win the RUF game with non-negligible advantage. This proves RUF-security of the plain RSA signature scheme.

Question 2E Plain RSA is not RUF-CMA secure [3 marks]

Given verification key (N,e) and a message $m \in \mathbb{Z}_N^*$ by the challenger, the adversary picks $m_1 \neq 1$ from $\mathbb{Z}_N^*$, computes $m_1^{-1}(mod\ N)$ using the Extended Euclidean algorithm, and $m_2 = m \cdot m_1^{-1}(mod\ N)$. It then queries the challenger for signatures to m_1 and m_2. These are, respectively, $m_1^d(mod\ N)$ and $m_2^d(mod\ N)$. Multiplying these then gives the valid signature $\sigma = m^d(mod\ N)$. This is also different from the queries made by the adversary so far, as m_1 and m_2 are both different from m. This attack succeeds with probability 1, and thus the plain RSA scheme is not RUF-CMA secure. (This attack was covered in class.)

Question 3A CCA security definition [3 marks]

The CCA security game for PKE scheme (Gen, Enc, Dec), with Challenger C and adversary A:

$\underline{CCA(\lambda):}$

1. C samples $(pk, sk) \leftarrow Gen(\lambda)$, and sends pk to A
2. Repeat until A stops sending queries:
   - A sends ciphertext c' to C
   - C computes $m' \leftarrow Dec(\lambda, sk, c')$ and sends it to A
3. A sends messages m_0, m_1 to C
4. C samples bit $b \leftarrow \{0,1\}$, and sends $c \leftarrow Enc(\lambda, pk, m_b)$ to A
5. Repeat until A stops sending queries:
   - A sends ciphertext c' to C such that $c' \neq c$
   - C computes $m' \leftarrow Dec(\lambda, sk, c')$ and sends it to A
6. A outputs bit b'
7. A wins if $b' = b$

The PKE scheme is CCA-secure if:

$$\forall \text{ PPT } A \ \exists \text{ negligible function } \nu : \Pr[A \text{ wins } CCA(\lambda)] < \nu(\lambda)$$

Question 3B CCA-security of ElGamal [3 marks]

The ElGamal encryption scheme is not CCA-secure, even if the DDH assumption holds. This is because it is malleable. We can construct an adversary that wins the CCA game with probability 1 as follows. Given ciphertext $(c_1, c_2) = (g^y, g^{xy} \cdot m)$, pick any $w \in G$ that is not the identity, and query the challenger (decryption oracle) with the ciphertext $(c_1, c_2 \cdot w)$. The decryption will return $((c_1)^x)^{-1} \cdot (c_2 \cdot w) = g^{-xy} \cdot g^{xy} \cdot m \cdot w = m \cdot w$. Multiplying this with w^{-1} now gives m. So the adversary can decrypt the challenge ciphertext and always guess b correctly.