TMA4160 CRYPTOGRAPHY
(WITH PARTIAL SOLUTIONS)
NTNU, SUMMER 2021

RESIT EXAM (AUGUST 2021)

Jiaxin Pan

Exercise 1:	5 points
Exercise 2:	5 points
Exercise 3:	5 points
Exercise 4:	5 points
Exercise 5:	10 points
Exercise 6:	5 points
Exercise 7:	15 points
Exercise 8:	13 points
Exercise 9:	37 points

Total: 100 points

This is only with partial solutions. In the exam, one should provide sufficient details for your arguments.

Exercise 1. Multiple Choice (One-time pad) (5 points)

AUTOMATICALLY CORRECTED BY INSPERA.

Alice uses One-Time Pad (OTP) to encrypt and notices that when her secret key is the all-zeroes string $K = 0^L$, then $E(K, m) = m$ and her message is sent in clear! To have the best security, what she should do?

(A) Nothing

(B) She can choose her secret key at random from the set of all L-bit strings except 0^L

ANSWER: *A*

Exercise 2. Multiple Choice (Negligible functions) (5 points)

SYSTEM WILL CHOOSE ONE OUT OF THE FOLLOWING 2 PROBLEMS. AUTOMATICALLY CORRECTED BY INSPERA.

(1) Which of the following functions is negligible in λ? ***There can be more than 1 correct choices. You get all 5 points if and only if you choose all the correct ones.***

(A) $\frac{1}{2^{\lambda/2}}$

(B) $\frac{1}{2^{\log(\lambda^3)}}$

(C) $\frac{1}{\lambda^2}$

(D) $\frac{f(\lambda)}{p(\lambda)}$*, where f is negligible in λ and p is a polynomial.*

ANSWER: *A, D*

(2) Which of the following functions is negligible in λ? ***There can be more than 1 correct choices. You get all 5 points if and only if you choose all the correct ones.***

(A) $\frac{1}{2^{(\log \lambda)^2}}$

Date: August 23, 2021.

(B) $\frac{1}{2\sqrt{\lambda}}$
(C) $\frac{1}{\sqrt{\lambda}}$
(D) $\frac{p(\lambda)}{f(\lambda)}$, *where* f *is negligible in* λ *and* p *is a polynomial.*

ANSWER: *A, B*

Exercise 3. Multiple Choice (Block cipher) (5 points)

System will choose one out of the following 3 problems. Automatically corrected by Inspera.

(1) *Which of the following statements about AES is true?* ***There can be more than 1 correct choices. You get all 5 points if and only if you choose all the correct ones.***
   (A) *AES is a block cipher*
   (B) *AES is an IND-CPA secure symmetric encryption scheme*
   (C) *AES is a compressed function*

   ANSWER: *A*

(2) *Which of the following statements about AES is true?* ***There can be more than 1 correct choices. You get all 5 points if and only if you choose all the correct ones.***
   (A) *AES is a block cipher*
   (B) *AES is assumed to be a pseudorandom permutation*
   (C) *AES is a public-key encryption*

   ANSWER: *A, B*

(3) *Which of the following statements about AES is true?* ***There can be more than 1 correct choices. You get all 5 points if and only if you choose all the correct ones.***
   (A) *AES can be used to construct an IND-CPA secure symmetric encryption scheme*
   (B) *AES is supposed to be computationally indistinguishable from a truly random function*
   (C) *AES can be used to construct a collision-resistant hash function*

   ANSWER: *A, B, C*

Exercise 4. Block cipher modes of operation (5 points)

Which of the Block cipher modes of operation you should not use to build a semantically secure symmetric encryption?

(A) *CBC*
(B) *CTR*
(C) *ECB*

ANSWER: *C*

Exercise 5. One-time Pad (10 points)

A ***known-plaintext attack*** *refers to a situation where an eavesdropper sees a ciphertext* $c = Enc(k, m)$ *and also learns the corresponding plaintext* m*.*

(1) *Give a known-plaintext attack on the one-time pad (OTP). (Hints: The attacker should be able to recover the key* k*.)*
(2) *Does this known-plaintext attack contradict perfect security of OTP discussed in our lecture? Explain why.*

ANSWER:

(1) The attacker has m *and* c *(where* $c = k \oplus m$*), and it can recover* $k = c \oplus m$*.*

(2) No, it doesn't. Because the security about OTP in our lecture does not consider this known-plaintext attack.

Exercise 6. Diffie-Hellman Key Exchange (5 points)

Explain what is wrong with the following argument:

"In Diffie-Hellman key exchange protocol, Alice sends $A := g^a$ and Bob sends $B := g^b$. Their shared key is g^{ab}. To break the protocol, the eavesdropper can simply compute $A \cdot B = (g^a) \cdot (g^b) = g^{ab}$."

ANSWER: *The issue is $A \cdot B = g^{a+b} \neq g^{ab}$*

Exercise 7. ElGamal Encryption (15 points)

Suppose you obtain two ElGamal ciphertexts $(R_1 = g^{r_1}, C_1 = (g^x)^{r_1} \cdot M_1)$ and $(R_2 = g^{r_2}, C_2 = (g^x)^{r_2} \cdot M_2)$ that encrypt M_1 and M_2, respectively. Suppose you also know the public key $X := g^x$ and cyclic group generator g.

(1) What information can you infer about M_1 and M_2 if $R_1 = R_2$?

(2) What information can you infer about M_1 and M_2 if $R_1 = (R_2)^2$?

(3) Is there a successful known-plaintext attack (as explained in a previous problem about OTP) on ElGamal PKE scheme? If yes, explain the attack; if not, explain why.

ANSWER:

(1) You can infer M_1/M_2 by computing $C_1/C_2 = (g^{xr_1} \cdot M_1)/(g^{xr_2} \cdot M_2)$. Since $g^{r_1} = g^{r_2}$, $g^{xr_1} = g^{xr_2}$ and $C_1/C_2 = M_1/M_2$.

(2) You can infer M_1/M_2^2 by computing $C_1/C_2^2 = \frac{g^{xr_1} \cdot M_1}{(g^{xr_2} \cdot M_2)^2}$. Since $g^{r_1} = g^{2r_2}$, $g^{xr_1} = g^{2xr_2}$ and $C_1/C_2 = M_1/M_2^2$.

(3) No, there isn't. Because given a ciphertext $(g^r, g^{xr} \cdot M)$ and the corresponding M, we can only recover g^{xr}. Since the Dlog problem is hard, we can't recover x.

Exercise 8. RSA function (13 points)

(1) Why the RSA exponent e must be odd? Or, equivalently, what kind of issue the RSA function, $f(x) = x^e \bmod N$, will have if e is even?

(2) Given an RSA modulus N and $\phi(N)$ (where ϕ is Euler's totient function), show that one can efficiently factor N to two primes p and q.

(3) Given equations $x = u \bmod r$ and $x = v \bmod s$ where x is unknown and u, v, r, s are known, there is an integer solution to x if

(A) $\gcd(r, s) = 1$

(B) $\gcd(r, s) \neq 1$

(This sub-question will be automatically correct by Inspera.)

ANSWER:

(1) If e is even, then it can be the case that $e|\phi(N)$, and there is no d such that $d = e^{-1} \bmod \phi(N)$. Thus, in this case, the RSA function cannot be inverted.

(2) One can efficiently solve the following two equations, $N = p \cdot q$ and $\phi(N) = (p-1)(q-1)$, to get the result.

(3) (A), by the Chinese Remainder Theorem.

Exercise 9. RSA signature (37 points)

Let N be an RSA modulus and e is the RSA exponent. Let $H : \mathbb{Z}_N \times \mathcal{M} \to \mathbb{Z}_e$ be a hash function. We define the following signature scheme $(Gen, Sign, Ver)$ with message space $\mathcal{M}$:

- *$Gen(1^\lambda)$: choose x from $\mathbb{Z}_N^*$ uniformly at random and compute $X := x^e \bmod N$. Return the public key $pk := X$ and secret key $sk := x$.*
- *$Sign(sk, m)$: choose r from $\mathbb{Z}_N^*$ uniformly at random and compute $R := r^e \bmod N$. Compute $h := H(R, m)$ and $s := x^h \cdot r \bmod N$. Return the signature $\sigma := (R, s)$.*

(1) (7 points) Define the verification algorithm, Ver, and show the correctness of the signature scheme.

The actual security proof of this signature scheme is not required, but the following two exercises are the important steps in the security proof.

(2) (15 points) Given two valid signatures $\sigma_1 := (R_1, s_1)$ and $\sigma_2 := (R_2, s_2)$ (namely, they both pass the verification) for messages m_1 and m_2, respectively, show that if $R_1 = R_2$ and $m_1 \neq m_2$, then one can efficiently recover the secret key x. If $R_1 = R_2$ and $m_1 = m_2$, can one still recover the secret key x? Why?

*(**Hints**: Shamir's trick.)*

(3) (15 points) By programming the random oracle H, one can efficiently simulate a signature for any message m without knowing the secret key x and the simulated signatures are distributed the same as in the real scheme. Show how to do it.

ANSWER:

(1) Trivial.

(2) If $R_1 = R_2$ and $m_1 \neq m_2$, $h_1 = H(R_1, m_1) \neq H(R_2, m_2) = h_2$ if H is collision-resistant. Then we show that $s_1^e / s_2^e = (X^{h_1} \cdot R_1)/(X^{h_2} \cdot R_2) = X^{h_1}/X^{h_2}$. Thus, we have the equation $(s_1/s_2)^e = X^{h_1 - h_2}$. Note that $\gcd(e, h_1 - h_2) = 1$ since H's outputs are in $\mathbb{Z}_e$. By Shamir's trick, we recover x such that $x^e = X \bmod N$.

The second half of the question can be an open discussion. The most direct answer is if $m_1 = m_2$ then $h_1 - h_2 = 0$ and thus the aforementioned approach cannot work. But in the actual proof of this signature scheme, the simulator will use the Forking Lemma to reprogram H such that $h_1 \neq h_2$ for the forgery. Then the aforementioned approach still works.

(3) The simulator will choose random s from $\mathbb{Z}_N$ and h from $\mathbb{Z}_e$. Compute $R := s^e / X^h$ and then program $H(R, m) := h$. Now the simulated signature (R, s) is the same as in the real scheme: R is random, since s is random and $f(s) := s^e \bmod N$ is a permutation, and s is the value that passes the verification.