1. [3.5p]

   (a) [1,5p] Explain ECB mode. Is it secure? Why? (Max. half page)

   Solution: Block ciphers encrypt some fixed number of bits. However, we usually want to encrypt messages of length different from that number of bits. There are several ways of encrypting long plaintexts with a block cipher using modes of operation. The Electronic Code Book (ECB) mode is the most straightforward way of encrypting a message. The message is divided into blocks, and each block is encrypted separately using the secret key.

   ECB mode is completely insecure. The reason is that it is deterministic. This means that identical plaintext blocks result in identical ciphertext blocks. Then, for example, an attacker can recognize if the same message has been sent twice simply by looking at the ciphertext. Also, if a ciphertext has a fixed header (for example, some metadata of a file) the header always results in the same ciphertext. Furthermore, as plaintext blocks are encrypted independently of previous blocks, an attacker can reorder the ciphertext blocks, and the resulting plaintext still is a valid plaintext that the attacker has manipulated.

   Common mistakes:

   i. Describing with OTP as encryption scheme and no the general case
   ii. Explaining and arguing about CBC instead of ECB
   iii. Using different keys for each block
   iv. Explaining just the penguin example seen in class
   v. For security, arguing only about reordering. This is indeed a weakness, but not the main one: it has to do with authenticaton and not encryption. When examining encryption schemes we are mainly worried about secrecy.

   (b) [2p] Let $h : \{0, 1, \ldots, 2^{2048} - 1\} \longrightarrow \{0, 1, \ldots 2^{256} - 1\}$ be a hash function satisfying

   $$x_1 \equiv x_2 \mod 2^{32} \Rightarrow h(x_1) = h(x_2)$$

   i. Find a second pre-image of $y = h(42)$.

   Solution: The objective is to find an $x \neq y$ such that $h(x) = h(y)$. Let $x = 42 + 2^{32}$. Clearly, $x \not\equiv 42 \mod 2^{2048}$. Still, $x \equiv 42 \mod 2^{32}$, which implies $h(x) = h(42)$. Then, x is a second pre-image of $h(42)$.

   Common mistakes:

   A. Trying to do the inverse of the digest.

   ii. Given a value $y = h(x)$, describe how you can find a pre-image of y efficiently (specifically, in time less than 2^{80}).

Solution: We want to find $x \in \{0,1\}^{2048}$ such that $h(x) = y$. It is enough to find an $\alpha \in \{0,1\}^{32}$ such that $\alpha \equiv x \mod 2^{32}$, because it implies $h(\alpha) = h(x)$, so α is a pre-image of y.
In order to find such α, we can apply brute force and do the following: for $i = 0, 1, \ldots, 2^{32} - 1$ compute $h(i)$ and check if $h(i) = y$. Since we need to check only 2^{32} values, 2^{80} is enough time to compute and check all of them.

Common mistakes:

A. Trying to use the x which we do not know.

B. Discussing about the memory and not the time

General correction criteria:

- (b.i) 0.15 for explaining Second Preimage and 0.85 for finding and arguing about the second preimage of 42.
- (b.ii) 0.5 for the idea of applying bruteforce (in some meaningfull context), 0.35 for the way it is applied and 0.15 for arguing about time.

2. [3.5p] In the following, Alice's RSA public key is $pk_A = (n, e)$, where $n = p \cdot q$ and p, q, e are prime numbers.

   (a) [1p] Bob sends both $m^e \mod n$ and $m^{e'} \mod n$ to Alice, for some prime number $e' \neq e$ that Oscar (the adversary) knows. Explain how Oscar can recover m efficiently.

   Solution: Bob sends to Alice two ciphertexts:

   $$c_1 = m^e \text{ mod } n$$
   $$c_2 = m^{e'} \text{ mod } n$$

   Because both e, e' are primes, we know there exist two coefficients k_1, k_2 such that $1 = k_1 e + k_2 e'$ (by Bezout's identity). As e, e' are known to Oscar, he can uses the EEA to find such k_1 and k_2 efficiently. Then,

   $$c_1^{k_1} c_2^{k2} = m^{k_1 e} m^{k_2 e'} = m^{k_1 e + k_2 e'} = m^1 = m$$

   Common mistakes:

   i. Trying to compute the inverse of e (or e'). This is only possible when dealing with very small numbers. For a generic n (as considered in this exercise) it is impossible to compute the secret key from (n, e).

   ii. Asking Alice for decryption of a ciphertext. If there is no explicit mentioning of Alice being able to provide the decryption of ciphertexts given by Oscar, you cannot assume that you can do so.

   (b) [1p] Is $pk_A = (77, 3)$ a valid public key? Why?

Solution: A valid public key for RSA is a pair (n, e) where $n = p \cdot q$ for two primes p, q and $\gcd(\Phi(n), e) = 1$.

- $77 = 11 \cdot 7$; because 7 and 11 are primes, n is a valid modulus.
- $\Phi(77) = (11-1)(7-1) = 10 \cdot 6 = 60$. $\gcd(60, 3) = 3 \neq 1$, then e is not a valid exponent for $n = 77$.

$(77, 3)$ is not a valid RSA public key.

Common mistakes:

i. Checking if $\gcd(n, e) = 1$ rather than $\gcd(\Phi(n), e) = 1$.

(c) [1,5p] If $pk_A = (143, 7)$,

- Prove that $8 = \text{Sign}_{pk_A}(57)$.

Solution: In order to check that 8 is the Signature under Alice's key of 57, we have to use the public key $(143, 7)$ and check if

$$8^7 \equiv 57 \mod 143.$$

For that, we will use fast exponentiation:

$$8^7 = 8^{2^2+2+1} = 8^{2^2} 8^2 8$$

$$8^2 \equiv 64 \text{ mod } 143, \ (8^{2^2}) = 64^2 = 4096 \equiv 92 \text{ mod } 143$$

Finally, $8^7 = 92 \cdot 64 \cdot 8 = 47104 \equiv 57$ mod 143, implying that $8 = Sign_{pk_A}(57)$. Notice that you could also solve this exercise after obtaining the secret key d and then checking if $57^d \equiv 8 \mod 143$.

Common mistakes:

i. Checking if $57^7 \equiv 8 \mod 143$ rather than $57^d \equiv 8 \mod 143$. That is, using the public key instead of the secret key to compute the signature.

- Compute Alice's secret key. Why wouldn't this be efficient if n was a large integer?

Solution: Because n is a small number, we can easily find its prime decomposition. In large integers, this is an intractable problem so we would not be able to find it. We know that in order to find the primes divisors of n, we have to try with all the primes that are less or equal than $143^{\frac{1}{2}}$. We see that 143 is not divisible by 2,3 or 5. If we try with 7 we find it is not as well. But, $\frac{143}{11} = 13$. Then, $143 = 11 \cdot 13$. Once we have the prime decomposition, we can calculate

$$\phi(143) = (11-1)(13-1) = 10 \cdot 12 = 120.$$

Finally, to get the private key d, we have to calculate the inverse of e modulus 120. For that, we use the EEA. We get $1 = 120 - 7 \cdot 17$, i.e., $1 = -17 \cdot 7 \mod 120$ and $-17 = 103$ is the inverse of e mod 120. Then

$$d = 103.$$

Common mistakes:

i. Stating that $\phi(143) = 142$.

- Compute $\sigma = \text{Sign}_{pk_A}(9)$. Compute $\text{Sign}_{pk_A}(9^{363})$ using σ.

Solution: We need to compute $\sigma \equiv 9^{103} \mod 143$, which can be done efficiently by fast exponentiation. Recall that to use fast exponentiation to compute x^d we start by setting an auxiliary variable (normally denoted by r) to 1 and we update it depending on the binary representation of the exponent d. We start each iteration by updating r to r^2. If the binary digit in consideration is 1 then we update r to $r \cdot x$ and move to the next digit. If the binary digit is 0 we skip this last step. In our case we have $x = 9$ and $d = (103)_{10} = (1100111)_2$. Therefore, we will have to do seven iterations of the process described above. Let $r = 1$, since the most significant binary digit of d is 1 we first compute

$$r = 1^2 \cdot 9 = 9.$$

The next digit is also 1, therefore we compute

$$r = 9^2 \cdot 9 = 729 \equiv 14 \mod 143.$$

The next digit is 0, thus we only need to compute

$$r = 14^2 = 196 \equiv 53 \mod 143.$$

If we finish the iterative process we end up with the following values of r:

$$9, 14, 53, 92, 100, 53, 113.$$

Since at the end of the process $r = 113$ we know that

$$\sigma \equiv 9^{103} \equiv 113 \mod 143.$$

Another way of thinking about fast exponentiation (using the idea explained above) is the following. By the binary representation of 103 we know that we need to compute

$$9^{2^6+2^5+2^2+2+1} \mod 143$$

We find that

$$9^2 \equiv 9^{32} \equiv 81 \mod 143 \text{ and } 9^4 \equiv 9^{64} \equiv 126 \mod 143.$$

This gives us that

$$\sigma = 9 \cdot 81^2 \cdot 126^2 \equiv 113 \mod 143.$$

For the second part, we can use σ in the following way:

$$Sign_{pk_A}(9^{363}) \equiv (9^{363})^{103} \equiv (9^{103})^{363} \equiv 113^{363} \mod 143.$$

Notice that $363 = 120 \cdot 3 + 3$ and that $\Phi(143) = 120$. Therefore, by using Euler's theorem and fast (or normal) exponentiation we obtain:

$$(113^{120})^3 113^3 \equiv 113^3 \equiv 27 \mod 143.$$

General correction criteria:

- (a) 1 point for correct use of Bezou'ts identity together with EEA and the general argument makes sense.
- (b) 0.5 (for checking if $gcd(e, \phi(n)) = 1$) + 0.5 (correct computations).
- (c1) 0.25 (for the correct approach) + 0.25 (correct computations).
- (c2) 0.35 (correct computation of the secret key) + 0.15 (stating that factorization is a computationally hard problem).
- (c3) 0.4 (for the correct approach) + 0.1 (correct computations).

3. [3p] The following interactive proof allows Peggy to prove to Victor that, for some public values $(g, h_1, h_2, y_1, y_2, z)$ in some cyclic group $\mathbb{G} \subset \mathbb{Z}_p^*$ generated by g of prime order q, she knows some secret value (x_1, x_2) such that $y_1 = g^{x_1} \mod p$, $y_2 = g^{x_2} \mod p$ and $z = h_1^{x_1} h_2^{x_2} \mod p$. To do so, the protocol goes as follows:

   1. Peggy chooses at random two values $r_1, r_2 \in \{1, 2, \ldots, q-1\}$, and computes $(R_1, R_2, Z) = (g^{r_1} \mod p, g^{r_2} \mod p, h_1^{r_1} h_2^{r_2} \mod p)$. Then she sends (R_1, R_2, Z) to Victor.
   2. Victor chooses at random a value $c \in \{0, 1, 2, \ldots, q-1\}$ and sends c to Peggy.
   3. Peggy computes $s_1 = r_1 - c \cdot x_1 \mod q$, $s_2 = r_2 - c \cdot x_2 \mod q$ and sends s_1, s_2 to Victor.
   4. Finally, Victor checks if $R_1 = g^{s_1} y_1^c \mod p$, $R_2 = g^{s_2} y_2^c \mod p$ and $Z = h_1^{s_1} h_2^{s_2} z^c \mod p$. Victor accepts if all the three equations hold and rejects otherwise.

   Answer the following questions:

   (a) [1pt] Show that if Peggy and Victor follow the protocol, Victor will accept.

   Solution: If they are both honest, then it holds that $y_1 = g^{x_1}, y_2 = g^{x_2}, z = h_1^{x_1} h_2^{x_2} \mod p$ and $s_1 = r_1 - xc, s_2 = r_2 - xc \mod q$. Then, Victor's check will be:

$$g^{s_1} y_1^c \mod p = g^{r_1 - cx_1} g^{x_1 c} \mod p = g^{r_1} \mod p = R_1$$
$$g^{s_2} y_2^c \mod p = g^{r_2 - cx_2} g^{x_2 c} \mod p = g^{r_2} \mod p = R_2$$
$$h_1^{s_1} h_2^{s_2} z^c \mod p = h_1^{r_1 - cx_1} h_2^{r_2 - cx_2} h_1^{x_1 c} h_2^{x_2 c} \mod p = h_1^{r_1} h_2^{r_2} \mod p = Z$$

   As all the equations hold, it accepts.

   Common mistakes:

   i. Most of you answered correctly, although some times there was some confusion between p and q.

   (b) [1pt] Assume Oscar does not know (x_1, x_2) and tries to impersonate Peggy, prove that for every fixed (R_1, R_2, Z), Oscar's answer in the last round can only make Victor accept for one single value of c. Does your argument use that q is a prime number?

Solution: Assume Oscar manages to answer to two different challenges c_A and c_B by sending respectively (s_1^A, s_2^A) and (s_1^B, s_2^B) such that Victor accepts. In particular, this means the first equation holds for both challenges and answers, meaning:

$$R_1 = g^{s_1^A} y_1^{c_A} \bmod p = g^{s_1^A} g^{x_1 c_A} \bmod p = g^{s_1^A + x_1 c_A} \bmod p \text{ , and also}$$
$$R_1 = g^{s_1^B} y_1^{c_B} \bmod p = g^{s_1^B} g^{x_1 c_B} \bmod p = g^{s_1^B + x_1 c_B} \bmod p$$

This implies

$$g^{s_1^A + x_1 c_A} = g^{s_1^B + x_1 c_B} \bmod p$$

what happens if and only if

$$s_1^A + x_1 c_A = s_1^B + x_1 c_B \bmod q$$

Resulting on

$$x_1 = \frac{(s_1^B - s_1^A)}{(c_A - c_B)} \bmod q$$

Because $c_A \neq c_B$ **and the fact that** q **is a prime** we have that $(c_A - c_B)$ has an inverse mod q and Oscar knows x_1, which is an absurd.

Common mistakes:

i. In the expression $R_1 = g^{s_1^A} y_1^{c_A} \bmod p = g^{s_1^B} y^{c_B} \bmod p$ replace $s_1^A = r_1 - c_A x_1$ and $s_1^B = r_1 - c_B x_1$. This does not allow you to conclude anything about x_1, because the terms with x_1 cancel.

ii. Assuming $R_1 = R_2$ and trying to solve the equation (which is then in terms of both x_1 and x_2).

iii. The problem says "for fixed (R_1, R_2, Z)", while some of you were using different values of (R_1, R_2, Z) for each challenge.

iv. Not mentioning that $c_A - c_B$ can only be inverted if q is prime.

(c) [1pt] Assume Oscar does not know (x_1, x_2) and tries to impersonate Peggy, show how he can succeed with probability $1/q$.

Solution: As Victor chooses its challenge as a number in $\{0, \ldots, q-1\}$, Oscar can guess it with probability $\frac{1}{q}$ by sampling a random number of the set.

Once Oscar makes it guess, it can choose s_1, s_2 randomly in $\mathbb{Z}_q$ and construct R_1, R_2 and Z from the public information, his guess of the challenge c and its chosen s_1, s_2 in such a way they will always satisfy the verification equations. Let

$$R_1 = g^{s_1} y_1^c \bmod p$$
$$R_2 = g^{s_2} y_2^c \bmod p$$
$$Z = h_1^{s_1} h_2^{s_2} z^c \bmod p$$

So, Oscar prepares R_1, R_2, Z, s_1, s_2 in advance for an specific challenge c. If Victor chooses the challenge c Oscar was ready for, it will accept, otherwise it looses. Then, Oscar wins with probability $\frac{1}{q}$

Common mistakes:

(a) Some of you argue that it is sufficient if the attacker guesses (s_1, s_2), but the attacker only has a chance of $1/q^2$ of doing so.

(b) Many of you do not say anything about how (R_1, R_2, Z, s_1, s_2) should be defined in terms of c, or do not give sufficient details. In particular, it should be said that s_1, s_2 are chosen before (R_1, R_2, Z).

General correction criteria:

- (a) 1 point for seeing that the 3 checks are satisfied if both prover and verifier act honestly.
- (b) 0.25 for setting up correctly the system of equations + 0.5 for finding expression in $\mathbb{Z}_q$ + 0.25 for arguing why q needs to be a prime.
- (c) 0.25 for arguing that the attacker just needs to guess c correctly for a successful attack + 0.75 for showing how this attack is done as a function of c.