

Due: Sunday, 17 February 2019, at 11:59pm

**Instructions.** This homework is due **Sunday, 17 February 2019, at 11:59pm**. No late homeworks will be accepted unless you have prior accommodations from us. This assignment must be done on your own.

Create an EECS instructional class account if you have not already. To do so, visit <https://inst.eecs.berkeley.edu/webacct/>, click “Login using your Berkeley CalNet ID,” then find the cs161 row and click “Get a new account.” Be sure to take note of the account login and password, and log in to your instructional account.

Make sure you have a Gradescope account and are joined in this course. The homework *must* be submitted electronically via Gradescope (not by any other method). Your answer for each question, when submitted on Gradescope, should be written in the space provided on this PDF form. You may either use the LaTeX form provided to fill out your responses, use Adobe Acrobat to fill in this fillable PDF, or print this paper out and handwrite your solutions, but **please make sure your responses do not overflow the box provided before submitting to ensure that you get full credit for your response.**

**Problem 1 True-or-False Questions****(45 points)**

Answer each question. You don't need to justify or explain your answer.

- (a) Select if true ☒ : The Diffie-Hellman key exchange protocol protects against eavesdroppers but is vulnerable to man-in-the-middle attacks.
- (b) Select if true ☐ : Suppose there is a transmission error in a block  $B$  of ciphertext using CBC mode. This error propagates to every block in decryption, which means that the block  $B$  and every block after  $B$  cannot be decrypted correctly.
- (c) Select if true ☐ : The IV for CBC mode must be kept secret.
- (d) Select if true ☒ : The random number  $r$  in El Gamal must be kept secret.
- (e) Select if true ☐ : The best way to be confident in the cryptography that you use is to write your own implementation.
- (f) Select if true ☐ : **(OPTIONAL)** Alice and Bob share a symmetric key  $k$ . Alice sends Bob a message encrypted with  $k$  stating, "I owe you \$100", using AES-CBC encryption. Assuming AES is secure, we can be confident that an active attacker cannot tamper with this message; its integrity is protected.
- (g) Select if true ☐ : If the daily lottery numbers are truly random, then they can be used as the entropy for a one-time-pad since a one-time-pad needs to be random.
- (h) Select if true ☒ : It is okay if multiple people perform El Gamal encryption with the same modulus  $p$ .
- (i) Select if true ☐ : **(OPTIONAL)** Alice and Bob share a secret symmetric key  $k$  which they use for calculating MACs. Alice sends the message  $M = \text{"I, Alice, owe you, Bob, \$100"}$  to Bob along with its message authentication code  $\text{MAC}_k(M)$ . Bob can present  $(M, \text{MAC}_k(M))$  to a judge as proof that Alice owes him \$100 since a MAC provides integrity.

**Problem 2 New Block Cipher Mode****(35 points)**

Nick decides to invent a new block cipher mode, called NBC. It is defined as follows:

$$C_i = E_k(C_{i-1}) \oplus P_i$$

$$C_0 = IV$$

Here  $(P_1, \dots, P_n)$  is the plaintext message,  $E_k$  is block cipher encryption with key  $k$ .

- (a) Given  $(C_0, C_1, \dots, C_n)$  and the key  $k$ , explain how to recover the original message  $(P_1, \dots, P_n)$ .

Since  $x \oplus y \oplus x = y$ , we can do...

$$C_i \oplus E_k(C_{i-1}) = E_k(C_{i-1}) \oplus P_i \oplus E_k(C_{i-1});$$

$$C_i \oplus E_k(C_{i-1}) = P_i$$

- (b) Is NBC encryption parallelizable? How about decryption? Provide a short justification for each.

Encryption is not parallelizable because each ciphertext block  $C_i$  relies on the previous ciphertext block,  $C_{i-1}$ .

Decryption is parallelizable because we are given all the ciphertext chunks  $C_0 \dots C_n$ , so we don't need to rely on a previous step.

- (c) As we saw in discussion, CBC mode is vulnerable to a chosen plaintext attack when the IV which will be used to encrypt the message is known in advance. Is NBC vulnerable to the same issue?

No. This is because the IV is encrypted before being XORed with the plaintext block, so knowing it doesn't tell you anything about the encryption of IV.

- (d) Say that Alice means to send the message  $(P_1, \dots, P_n)$  to Bob using NBC mode. By accident, Alice typos and encrypts  $(P_1 \oplus 1, \dots, P_n)$  instead (i.e., she accidentally flips the last bit of the first block).

TRUE or FALSE: after Bob decrypts the resulting ciphertext, every block after the first is incorrect. Explain your answer.

False. This is the same as just encrypting  $(P_1 \oplus 1, \dots, P_n)$  in the first place, so decrypting it will result in a correct decryption for that specific encryption.

- (e) Alice encrypts the message  $(P_1, \dots, P_5)$ . Unfortunately, the block  $C_3$  of the ciphertext is lost in transmission, so that Bob receives  $(C_0, C_1, C_2, C_4, C_5)$ . Assuming that Bob knows that he is missing  $C_3$ , which blocks of the original plaintext can Bob recover?

Select if the block is recoverable:  $P_1$  ☒  $P_2$  ☒  $P_3$  ☐  $P_4$  ☐  $P_5$  ☒

**Problem 3 Hashing Functions****(15 points)**

Recall the definition of “one-way functions” and “collision-resistance” from lecture. We say a function  $f$  is one-way if given  $f(x)$  it is hard to find  $x'$  such that  $f(x') = f(x)$ . Likewise, we say a function  $f$  is “collision-resistant” if it is hard to find two inputs  $x, y$  such that  $f(x) = f(y)$  but  $x \neq y$ . For each of the given functions  $H$  below, determine if it is one-way or not, and if it is collision-resistant or not. (State any assumptions that you make in the margin.)

- (a) Select if  $H(x) = x$  is: One-way ☐ Collision-resistant ☒
- (b) Select if  $H(x) = x \bmod 2$  is: One-way ☐ Collision-resistant ☐
- (c) Select if  $H(x) = E_k(x)$  is: One-way ☐ Collision-resistant ☒ , where  $E_k$  is a ideally secure block cipher with a known and published key  $k$ .

For part a),  $H(x)$  is not one-way because after repeated  $(x,y)$  pairs, it is fairly easy to see that  $x'$  should equal  $x$  in order for  $f(x')$  to equal  $f(x)$ . It is collision-resistant because unless  $x = y$ , there is no way for  $H(x)$  to equal  $H(y)$ .

For part b),  $H(x)$  is not one-way because  $f(x)$  can only be equal to 0 or 1, which means that after multiple  $(x,y)$  pairs we can figure out the hash function. It is not collision-resistant because there are multiple values of  $x$  which map to the same value of  $y$  (after all,  $y$  can only be equal to 0 or 1).

**Problem 4** *Finding Common Patients* (40 points)

Caltopia has two hospitals: Bear Hospital and Tree Hospital, each of which has a database of patient medical records that contain highly sensitive, confidential patient information. For both hospitals, each medical record is a tuple  $(p_i, m_i)$ , where  $p_i$  and  $m_i$  are strings that correspond to the patient’s full name and medical record respectively; assume that every person in Caltopia has a unique full name. Thus, we can think of Bear Hospital’s patient database as a list of tuples  $(x_1, m_1), (x_2, m_2), \dots, (x_n, m_n)$ , where  $m_i$  is the medical information that Bear Hospital has for patient  $x_i$ . Similarly, we can think of Tree Hospital’s database as a list  $(y_1, m'_1), (y_2, m'_2), \dots, (y_m, m'_m)$ , where  $m'_i$  is a string that encodes the medical information that Tree Hospital has for the patient named  $y_i$ . Note that for a given patient, Tree Hospital and Bear Hospital might have different medical information.

The two hospitals want to collaborate on a way to identify which Caltopia citizens are patients at both hospitals. However, due to privacy laws, the two hospitals cannot share any plaintext information about patients (including their names) unless both hospitals know *a priori* that a patient has used both hospitals.

Thus, the two hospitals decide to build a system that will allow them to identify common patients of both hospitals. They enlist the help of Lady Olenna, who provides them with a trusted, third-party server  $S$ , which they will use to discover the names of patients who use both hospitals. Specifically, Bear Hospital will take some information from its patient database and transform it into a list  $(x_1^*), (x_2^*), \dots, (x_n^*)$  (where  $(x_i^*)$  is somehow derived from  $x_i$  (the patient's full name) and upload it to  $S$ . Similarly, Tree Hospital will take information from its patient database, transform it into a list  $(y_1^*), (y_2^*), \dots, (y_m^*)$ , and upload this transformed list to  $S$ . Finally,  $S$  will compute a set of tuples  $P = (i, j) : x_i = y_j$  of all pairs  $(i, j)$  such that  $x_i^* = y_j^*$  and send  $P$  to both Bear Hospital and Tree Hospital. The two hospitals can then take their respective indices from the tuples in  $P$  to identify patients who use both hospitals.

We want to ensure three requirements with the above scheme: (1) if  $x_i = y_j$ , then  $(i, j) \in P$ , (2) if  $x_i \neq y_j$ , then it is very unlikely that  $(i, j) \in P$ , (3) even if Eve (an attacker) compromises  $S$ , she cannot learn the name of any patient at either hospital or the medical information for any patient. For this question, assume that Eve is a passive attacker who cannot conduct Chosen Plaintext Attacks; however, she does know the names of everyone in Caltopia, and there are citizens whose full names are a unique length.

Your solution can use the cryptographic hash SHA-256 and/or AES with one of the three block cipher encryption modes discussed in class; keep in mind that Eve can also compute SHA-256 hashes and use AES with any block cipher mode. You can assume that Bear Hospital and Tree Hospital share a key  $k$  that is not known to anyone else. You *cannot* use public-key cryptography or modular arithmetic.

- (a) In the collaboration scheme described above, how should Bear Hospital compute  $x_i^*$  (as a function of  $x_i$ )? How should Tree Hospital compute  $y_i^*$  (as a function of  $y_i$ )? Specifically, your solution should define a function  $F$  that Bear Hospital will use to

transform  $x_i$  into  $x_i^*$ , and if relevant, a function  $G$  that Tree Hospital will use to transform  $y_i$  into  $y_i^*$ .

Let's define our encryption scheme  $E_k(x)$  to be a deterministic function (which is admissible since all patient names are unique) under the ECB benchmark. The function should be deterministic in order to preserve randomness for the same patient name. Let  $H(x)$  be the SHA-256 hash function in order to uphold requirement 3) and prevent Eve from knowing the lengths of patients' names. Then, we have the function  $F_k(x) = G_k(x) = E_k(H(x))$ . Then, Bear Hospital can compute  $x_i^* = F_k(x_i)$  and Tree Hospital can compute  $y_i^* = G_k(y_i)$ , where  $x_i$  and  $y_i$  are the patient names.

- (b) Explain why requirement (1) is met by your solution, i.e., explain why it is guaranteed that if  $x_i = y_j$ , then  $x_i^* = y_j^*$  will hold. Explain your answer in one or two sentences.

Since the encryption scheme and the hashing scheme are deterministic, we will get the same output for the same input, which means that if  $x_i = y_j$ , then  $x_i^* = y_j^*$ .

- (c) Explain why requirement (2) is met by your solution, i.e., if  $x_i \neq y_j$ , explain why it is unlikely that  $x_i^* = y_j^*$ . Explain your answer in one or two sentences.

Since we are using SHA-256 as our hash function, which is collision-resistant, it's unlikely that there will be any collisions among different patient names. Furthermore, the ECB block cipher encryption is an invertible, bijective function which makes it difficult for collisions to occur.

- (d) Explain why requirement (3) is met by your solution, i.e., if  $S$  is compromised by Eve, then the information known to  $S$  does not let Eve learn any patient information (neither the names of patients at a particular hospital nor the medical history for any patient). Explain your answer in one or two sentences.

Since this scheme encrypts the hash, it ensures confidentiality. Also because there is no IV, if multiple messages have the same first block, then the corresponding blocks of the ciphertext will also be identical, which is fine since only the hashed values are being encrypted (unlikely that multiple hashes will be similar due to collision-resistance), so Eve can't possibly learn the names of patients or the medical history.

**Problem 5 El Gamal Encryption (30 points)**

Recall the definition of El Gamal encryption from lecture. Bob publishes a large prime  $p$ , and an integer  $g$  with  $1 < g < p - 1$ . To generate a key, Bob chooses a random value  $0 \leq b \leq p - 2$ , and computes  $B = g^b \bmod p$ . Bob's public key is  $B$ , and his private key is  $b$ . If Alice wants to send a message  $m$  to Bob, she begins by generating a random  $r$  such that  $0 \leq r \leq p - 2$ , and creates the ciphertext  $(c_1, c_2) = (g^r \bmod p, m \cdot B^r \bmod p)$ . To decrypt the ciphertext, Bob calculates  $c_1^{-b} c_2 \equiv m \pmod{p}$ .

Note: As mentioned in the notes, this simplified El Gamal scheme is actually not semantically secure.

- (a) Suppose you intercept a ciphertext  $(c_1, c_2)$  that Alice has encrypted for Bob, which is the encryption for some message  $m$ . Construct a ciphertext  $(c'_1, c'_2)$  which is the encryption of  $2m$ . **Answer Format:** (\_\_\_\_\_, \_\_\_\_\_)

$$(c'_1, c'_2) = (g^r \bmod p, 2m \cdot B^r \bmod p)$$

- (b) Suppose you intercept two ciphertexts  $(c_1, c_2)$  and  $(c'_1, c'_2)$  that Alice has encrypted for Bob. Assume they are encryptions of some unknown messages  $m_1$  and  $m_2$ . Construct a ciphertext  $(c''_1, c''_2)$  which is a valid El Gamal encryption of the message  $m_1 \cdot m_2 \bmod p$ . **Answer Format:** (\_\_\_\_\_, \_\_\_\_\_)

$$\begin{aligned} (c_1, c_2) &= (g^{r_1}, m_1 \cdot B^{r_1} \bmod p); \\ (c'_1, c'_2) &= (g^{r_2}, m_2 \cdot B^{r_2} \bmod p); \\ (c''_1, c''_2) &= (c_1 \cdot c'_1, c_2 \cdot c'_2) = (g^{(r_1 + r_2)}, m_1 \cdot m_2 \cdot B^{(r_1 + r_2)} \bmod p); \end{aligned}$$

- (c) Consider a new scheme where the value  $r$  is not generated randomly every time. Instead, Alice begins by randomly generating an initial value  $r_0$ , and then simply incrementing  $r_0$  by 1 every time she needs to encrypt another message. Is the resulting encryption scheme IND-CPA?

No. If an adversary sends a message  $m_1$ , which is encrypted with random value  $r_0$  by the challenger, then the resulting ciphertext would be  $(c_1, c_2) = (g^{r_0}, m_1 \cdot B^{r_0})$ .

Then, the adversary sends two different messages ( $m_1$  and  $m_2$ ) and the challenger randomly chooses one to encrypt, resulting in the ciphertext  $(c'_1, c'_2) = (g^{(r_0 + 1)}, m_3 \cdot B^{(r_0 + 1)})$  where  $m_3$  is either  $m_1$  or  $m_2$ .

However, this ciphertext  $(c'_1, c'_2)$  leaks information about the ciphertext  $(c_1, c_2)$ . Namely, the adversary can calculate  $(g \cdot c_1, B \cdot c_2)$  to get  $(g^{(r_0 + 1)}, m_1 \cdot B^{(r_0 + 1)})$ . If this is equal to  $(c'_1, c'_2)$ , then we can say with probability greater than  $1/2$  that  $m_3 = m_1$ . If not, then  $m_3 = m_2$ .



**Problem 6   *Feedback*****(0 points)**

Optionally, feel free to include feedback. What's the single thing we could do to make the class better? Or, what did you find most difficult or confusing from lectures or the rest of class, and what would you like to see explained better? If you have feedback, submit your comments as your answer to Q6.