Practice Problems

Question	Parts	Points
1: Instructions	1	0
2: True or False	10	20
3: Authentication schemes	6	20
4: Short Answer Questions	3	15
5: Generating Randomness	1	5
6: Pseudorandom functions	2	10
7: Protocol Analysis	2	10
8: Encryption and MAC	1	5
9: MAC1	1	5
10: Variants on Diffie-Hellman (DH)	4	20
11: Course Survey	5	5
Total:		115

Name: ____

The actual midterm will be for a total of 120 points.

Problem 1. [0 points] **Instructions** (1 part)

- This is an open book exam: you can use your notes from this class, or any material released by us this term. You cannot use the internet. Use of any material not released by us this term is *strictly* forbidden.
- Any form of collaboration is *strictly* forbidden.
- If you need assistance clarifying a question in the exam, raise your hand and a proctor will come by.
- Point totals correspond roughly to how much time we expect you to spend on each problem (part).

Problem 2. [20 points] **True or False** (10 parts)
Please circle **T** or **F** for the following. *No justification is needed (nor will be considered).*

(a) [2 points] A CPA-secure encryption scheme must hide the length of encrypted messages.

(b) [2 points] Let H be a standard collision-resistant hash function (e.g., SHA256), and let p be a random 32-bit string. An attacker who learns $H(p)$ can recover p.

(c) [2 points] Time-based one-time passwords require a client and a server to share a secret.

(d) [2 points] The shortest digital signatures commonly used in practice today are 2048 bits long.

(e) [2 points] HTTPS uses a “trust on first use” design for distribution of public keys.

(f) [2 points] Let (E, D) be an arbitrary CPA-secure symmetric encryption scheme. This scheme remains secure if an attacker learns only one bit of the secret key.

(g) [2 points] A CPA-secure encryption scheme on n-bit messages (with perfect correctness) must have ciphertexts that are more than n bits long.

(h) [2 points] Large-scale quantum computers can efficiently break all known digital-signature schemes.

(i) [2 points] In the lab 2 web of trust, Alice adds Bob as a trusted contact by signing Bob's public key. Cedric also adds Bob as a trusted contact by signing Bob's public key. Assuming lab 2 has been succesfully completed, Alice can now retreive Cedric's public key in an authenticated manner.

(j) [2 points] Is it the case that for *every possible* CPA secure encryption scheme, its ciphertext looks random?

Problem 3. [20 points] **Authentication schemes** (6 parts)

We consider the problem of a client authenticating to a database server over a network. After authenticating, the client will send a variable-length query command to the server.

(a) [3 points] Give two advantages of password-based authentication over MAC-based authentication.

(b) [3 points] Give two advantages of MAC-based authentication over password-based authentication.

(c) [3 points] Give two advantages of MAC-based authentication over signature-based authentication.

(d) [2 points] Give one advantage of signature-based authentication over MAC-based authentication.

(e) [5 points] To provide strong security, the database administrator decides to use MAC-based authentication. During account registration, each client will generate a MAC key k and send it to the server. To authenticate, the client will send the message $t = \text{MAC}(k, \texttt{user\_name})$ to the server over the open network. The server, will check that $t == \text{MAC}(k, \texttt{user\_name})$ and will authenticate the client if so.

Notice that if an attacker can observe the communications between Alice and the server, explain how the attacker can impersonate Alice without knowing her key k. Explain how to modify the authentication protocol to eliminate this replay attack (*without* using additional cryptographic tools).

(f) [4 points] The scheme from the prior part does not authenticate the client's query. Assume that the MAC takes as input 128-bit messages and the client's query can be of arbitrary length. Your friend proposes using a hash function $H\colon \{0,1\}^* \to \{0,1\}^{128}$, to hash the username and query and then feed the hashed values to the MAC. Explain why the resulting authentication scheme cannot have 128-bit security.

Problem 4. [15 points] **Short Answer Questions** (3 parts)

(a) [5 points] Consider the Diffie-Hellman key exchange protocol working modulo a large prime p with generator g. (In the rest of this problem, we leave the "$\mathrm{mod} p$" operator implicit.) In the usually Diffie-Hellman protocol, Alice chooses a independently and uniformly at random from $\{1, \ldots, p\}$ and sends g^a to Bob. Bob chooses b similarly and sends g^b to Alice.

Now assume that there exists an active adversary who can replace g^a sent by Alice with a new value of the adversary's choosing. Similarly, the attacker can replace the value that Bob sends to Alice. What might the Adversary do to learn the secret key on which Alice and Bob agree? Explain.

(b) [5 points] We are given a random oracle $H : \{0,1\}^* \to \{0,1\}^{256}$. Is $H(x) = x \parallel H(x)$ collision resistant? Is it one way? $\parallel$ denotes concatenation.

(c) [5 points] Show how to turn any one-way hash function into a hash function that is one-way but not collision-resistant (CR).

Problem 5. [5 points] **Generating Randomness** (1 part)
As we have seen in class, cryptography relies on the fact that users can generate randomness (which is needed for encryption, key exchange, etc). However, perfect randomness can be hard to generate. Suppose a user works hard to generate a perfectly random (secret) seed $K \in \{0,1\}^{256}$, but doesn't have the ability to generate more randomness. How can the user use K for all his cryptographic purposes, which require generating many random strings?

Problem 6. [10 points] **Pseudorandom functions** (2 parts)
Let $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a pseudorandom function, where the first argument is the key and the second argument is the input.

(a) [5 points] Say that, for all $x \in \{0,1\}^n$, it holds that $F(0^n, x) = 0^n$. Can F still be a secure PRF? Why or why not?

(b) [5 points] Let $R(x)$ be a random function from $\{0,1\}^n$ to $\{0,1\}^n$. At how many distinct points $x_1, \ldots, x_k$ must you evaluate R before expecting to find a collision in the values $\{R(x_1), \ldots, R(x_k)\}$ with a significant probability (say, 50%).

Problem 7. [10 points] **Protocol Analysis** (2 parts)
You are given are two protocols in which the sender's party performs the following operation:

Protocol A

$$y = e_{k_1}(x \parallel H(k_2 \parallel x))$$

where x is the message, H is a hash function such as SHA256, e is a symmetric-key encryption algorithm, $\parallel$ denotes concatenation, and k_1, k_2 are secret keys which are only known to the sender and the receiver.

Protocol B

$$y = x,\ E_{PK}(H(x))$$

where SK is a private key of the sender (not shared with the receiver), PK is a public key of the receiver, and E is a public-key encryption algorithm.

(a) [5 points] Provide a step-by-step description of what the receiver does upon receipt of y.

(b) [5 points] State whether confidentiality and integrity is achieved for each of the two protocols given in the previous problem. Briefly justify each answer.

Problem 8. [5 points] **Encryption and MAC** (1 part)

(a) [5 points] Alice wishes to provide Chosen Ciphertext Attack (CCA) security. Alice proposes the following scheme, where k_1 and k_2 are two independent randomly-chosen keys, $\|$ denotes concatenation, Enc is a Chosen Plaintext Attack (CPA)-secure encryption scheme, and Mac is a message authentication code that is existentially secure against adaptive chosen message attacks.

$$m \rightarrow Enc(k_1, m) \parallel Mac(k_2, m)$$

Show Alice that this scheme is not CCA-secure by choosing a particular Mac.

Problem 9. [5 points] **MAC1** (1 part)
Let MAC be a message authentication code for messages in $\{0,1\}^{256}$ that is secure against adaptive chosen message attacks. Let $F : \{0,1\}^{512} \to \{0,1\}^{256}$ be a one-way function (i.e., easy to compute and hard to invert). Consider the new message authentication code for messages in $\{0,1\}^{512}$:

$$\mathsf{MAC}'(K, M) = \mathsf{MAC}(K, F(M)).$$

Is MAC' secure against adaptive chosen message attacks?

Problem 10. [20 points] **Variants on Diffie-Hellman (DH)** (4 parts)
In class, we saw the Diffie-Hellman key-exchange protocol. The protocol works with the integers modulo a large prime $p \approx 2^{2048}$ and uses a carefully chosen generator $g \in \{1, \dots, p\}$.

The protocols we consider use the following form:

- Alice samples $a \leftarrow_R \{1, \dots, p\}$ at random and sends Bob $A = g^a \bmod p$.
- Bob samples $b \leftarrow_R \{1, \dots, p\}$ at random and sends Alice $B = g^b \bmod p$.

(a) [5 points] Assume that one addition or one multiplication modulo p takes time $O(\log p)$ operations. What is the asymptotic complexity of computing $g^a \bmod p$?

(b) [5 points] Say that Alice and Bob plan to use $g^{a+b} \bmod p$ as their shared secret. What is wrong with this plan?

(c) [5 points] Say that Alice and Bob plan to use $g^{(a^b)} \bmod p$ as their shared secret. What is wrong with this plan?

(d) [5 points] Say that Alice and Bob plan to use $(g^{ab} + g^a + g^b + 127) \bmod p$ as their shared secret. Why is this variant as secure against a passive eavesdropper as the original Diffie-Hellman protocol?

Problem 11. [5 points] **Course Survey** (5 parts)

(a) [1 point] What was your favorite topic in the class so far?

(b) [1 point] Are there any topics that you think we should remove next time?

(c) [1 point] Are there any topics that you were hoping to see covered in this class or lab assignments, which aren't currently covered?

(d) [1 point] What was your favorite part of the lab assignments so far?

(e) [1 point] What was the most tedious or least interesting part of the lab assignments?