Midterm

Question	Parts	Points
1: Instructions	1	0
2: True or False	10	20
3: Security overview	1	6
4: User authentication	3	9
5: Hashing long messages	3	10
6: Message Authentication Codes	2	10
7: One-time pad	5	15
8: Naming and public keys	4	20
9: Encrypted software	3	15
10: DH Key-Exchange and CPA security	1	10
11: Course Survey	5	5
Total:		120

Name: ____

You can answer the survey question at the back of the midterm before the start of the midterm!

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 answer **T** or **F** for the following. *No justification is needed (nor will be considered).*

(a) [2 points] Target collision resistance implies collision resistance.

(b) [2 points] An attacker that can run in unbounded time can find collisions in any efficiently computable hash function $H\colon \{0,1\}^* \to \{0,1\}^{256}$.

(c) [2 points] Bitcoin miners collectively perform more than 2^{160} SHA256 hashes every year.

(d) [2 points] A pseudorandom function is a type of collision-resistant hash function.

(e) [2 points] Pseudorandom functions only exist under computational hardness assumptions.

(f) [2 points] Performing a single Diffie-Hellman key-exchange operation (with the parameters commonly used in practice today) is more expensive than hashing a 1 KB message with SHA256.

(g) [2 points] In a public-key infrastructure based on certificates, different clients can choose to trust different certificate authorities (CAs).

(h) [2 points] AES-GCM is an example of the "encrypt-then-MAC" paradigm for designing an authenticated-encryption scheme.

(i) [2 points] When a message is encrypted with a CCA-secure public-key encryption scheme, the ciphertext leaks no information about the recipient's public key.

(j) [2 points] If the lab 1 server is malicious, but lab 1 has been successfully completed, we can guarantee that the new device will fully synchronize all photos with the old device.

Problem 3. [6 points] **Security overview** (1 part)

(a) [6 points] For each item in the following list, identify it as either an examples of a threat model (mark with "T"), an examples of a security goal (mark with "G"), or neither (mark with "N").

- ☐ Students in a class should be able to print a report of their lab grades.
- ☐ Only course staff should be able to access the submitted lab assignments.
- ☐ The adversary cannot monitor user keystrokes.
- ☐ Adversaries who are not members of the MIT community should not be able to access library resources.
- ☐ An adversary is assumed to not be able to factor integers that are the product of two 1024-bit primes.
- ☐ People that are not authorized to enter a building should be told to contact the card office when swiping their card at the building's card reader.

Problem 4. [9 points] **User authentication** (3 parts)

(a) [4 points] Ben Bitdiddle develops Ben's Biometric Hash, a hash function that takes a picture of a user's fingerprint and outputs a 160-bit value. The hash function is *stable*: given two pictures of the same user's fingerprint, it will return the same hash value with high probability. However, it is collision-resistant in the sense of it being difficult for an adversary to find another fingerprint that produces the same hash, and expensive to compute. Ben proposes using his hash instead of passwords: to log into a web site, a user sends a picture of their fingerprint, and the web server compares the hash of the submitted picture to the user's registered fingerprint hash.

List two significant security problem with Ben's design.

(b) [3 points] Undeterred, Ben hears about password salting, and adds salting to Ben's Biometric Hash. With salting, his hash function takes two inputs: the picture of the fingerprint and a salt value. When a user registers on a web site, a new random salt value is chosen for hashing that user's fingerprint, and the salt value is stored alongside the hash value.

What security property might a web site achieve by using a salted version of Ben's hash?

(c) [2 points] Ben finds it difficult to sell his authentication system to web sites, and instead develops Ben's Password Manager, a smartphone application that encrypts the user's passwords using a key derived from the user's fingerprint using Ben's Biometric Hash. Is this a better idea than using Ben's hash for web sites (as above)? Explain why or why not.

Problem 5. [10 points] **Hashing long messages** (3 parts)
Let $h\colon \{0,1\}^{2n} \to \{0,1\}^n$ be a collision-resistant hash function. Let $|x|$ denote the length of the bitstring x and let $\|$ denote string concatenation.

(a) [3 points] The function h operates on $2n$-bit inputs, but Alice would like a collision-resistant hash function on messages of any length between 0 and $2n$ bits. Alice defines the function $h_{\mathsf{var}}\colon \bigcup_{\ell=0}^{2n} \{0,1\}^\ell \to \{0,1\}^n$ as

$$h_{\mathsf{var}}(x) := h(x \| 0^{2n-|x|}).$$

Give an example that shows that h_{var} is not collision resistant.

(b) [3 points] Define $H(b_1, b_2) := h\big(h(b_1) \| h(b_2)\big)$, where each of the inputs to H is $2n$ bits long (so there are no attacks based on variable input lengths). Assume that you have an efficient algorithm $\mathcal{A}$ that outputs a collision in H. Use $\mathcal{A}$ to construct an efficient algorithm $\mathcal{B}$ that outputs a collision in h.

(c) [4 points] Let H be the function defined in the prior part. Show that if an attacker is given a *single* collision x, x' in h, it can efficiently compute *multiple* collisions in H.

Problem 6. [10 points] **Message Authentication Codes** (2 parts)
Let K be a random 128 bit AES key, and let $\ell = 2^{10}$. Consider the following randomized message authentication code for messages in $(\{0,1\}^{40})^{\leq \ell}$ (i.e., messages that contain at most ℓ blocks each of length 40): Given any message $M = (M_1, \ldots, M_{\ell'}) \in (\{0,1\}^{40})^{\ell'}$ for $\ell' \in \{1, \ldots, \ell\}$ choose a random nonce $r \in \{0,1\}^{88-\log \ell}$ and output

$$\mathsf{MAC}(K, M, r) = (\mathsf{AES}(K, (r, 1, M_1))||\ldots||\mathsf{AES}(K, (r, \ell', M_{\ell'})), r)$$

where each index $1, 2, \ldots, \ell'$ is encoded as an element in $\{0,1\}^{\log \ell}$.

(a) [7 points] Is this MAC secure against adaptive chosen message attacks? Why or why not?

(b) [3 points] Is this MAC secure against adaptive chosen message attacks if we restrict the message space to be $(\{0,1\}^{40})^{\ell}$ (i.e., each message consists of exactly ℓ blocks), assuming AES is a pseudorandom permutation? You just need to answer yes or no.

Problem 7. [15 points] **One-time pad** (5 parts)
Say that Alice and Bob share keys $k_1, k_2, k_3, k_4, k_5, k_6$, each of n bits long, sampled independently and uniformly at random from the set of n-bit strings. Throughout this question, assume that n is even.

(a) [3 points] Alice encrypts her one-bit message m using the one-time pad with key k_1. What is the value of ciphertext as a function of m and k_1?

(b) [3 points] Alice has two n-bit messages to send Bob: m_1 and m_2. She uses the one-time pad encryption scheme but to conserve randomness, she encrypts both messages with the same key k_2.

Show that if an eavesdropper can intercept both ciphertexts, it can learn a function of m_1 and m_2 without knowledge of either.

(c) [3 points] Alice again wants to send two n-bit messages to Bob using the one-time pad. It is public information that her first message m_1 begins with a publicly known $n/2$-bit string (e.g., `To: bob@mit.edu`) and that her second message m_2 ends with a publicly known $n/2$-bit string. To conserve randomness, Alice encrypts both messages using the same key k_3. Show how an attacker can recover the entire key and both plaintexts.

(d) [3 points] Alice now wants to send three n-bit messages m_1, m_2, m_3 to Bob. She encrypts them using the one-time pad with keys k_4, k_5, and $k_4 \oplus k_5$. Can the attacker recover any information about the plaintexts given only the ciphertexts? Give an explanation (if secure) or an attack (if insecure).

(e) [3 points] Is it possible to use the one-time pad to encrypt a sequence of $n/2$ two-bit messages with key k_6? (Note that if n is large, you might be encrypting the same plaintext multiple times.) Explain how or explain why it is not possible.

Problem 8. [20 points] **Naming and public keys** (4 parts)
On the Internet today, we use DNS to map hostnames (e.g., `mit.edu`) to IP addresses (e.g., `104.93.189.3`). We use a certificate-based public-key infrastructure to associate public keys with hostnames.

This problem explores a number of **alternate possible designs** that are not dependent on each other.

(a) [5 points] Instead of using human-readable hostnames, we could use the public key as the site identifier. So, to visit `mit.edu`, you would browse to

`https://<hash of MIT's public key>/.`

List two benefits and two drawbacks of this approach.

(b) [5 points] Instead of using the public-key certificates, the client could fetch MIT's public key from the DNS infrastructure. In particular, the client would query the DNS server for (a) the IP address for `mit.edu` and (b) the public key for `mit.edu`. Explain one security problem with this approach.

(c) [5 points] The public-key certificate for `mit.edu` is signed by an "intermediate" certificate authority, whose key is in turn signed by a "root" certificate authority. Say that your web browser has the public keys for 100 root certificate authorities CAs. How many root or intermediate CA secret keys does an attacker need to compromise to issue a fraudulent certificate for `mit.edu`? Briefly explain.

(d) [5 points] Instead of using a certificate-based public-key infrastructure, the web could take a "trust on first use" approach to public-key distribution. Give two benefits and two drawbacks of such a design.

Problem 9. [15 points] **Encrypted software** (3 parts)

A good friend of yours is working on a new cutting-edge computer game and wants to send it to you to try out. To prevent spies in the network from stealing the code, she encrypts the game's Python code using AES-GCM with a random 256-bit key and stores the ciphertext in the following Python program called `decrypt.py`:

```
def main():
  ciphertext = """<ENCRYPTED SOURCE CODE OF GAME HERE >"""
  key = get_key_from_command_prompt()
  source_plaintext = AES_GCM_decrypt(key, ciphertext)
  eval(source_plaintext)

main()
```

You and your friend have a shared 256-bit secret key. Your friend sends you `decrypt.py` over the network (i.e., an insecure channel) and instructs you to decrypt it using your shared secret.

(a) [5 points] Say you run `decrypt.py` and feed it the correct decryption key. Explain how the attacker can recover the game source code.

(b) [5 points] Say you never run `decrypt.py`. Can an attacker in the network recover the game source code? If yes, explain how.

(c) [5 points] Explain how your friend could send you the game source code without the risk of it being stolen. You and your friend share a 256-bit secret key but can only communicate over an insecure open network.

Problem 10. [10 points] **DH Key-Exchange and CPA security** (1 part)
You are given a hash function $H : \{1, \ldots, p\} \to \{0,1\}^{256}$ where p is a random 2048 bit prime. Show how to use ideas from the Diffie-Hellman key exchange protocol together with H to construct a public-key CPA secure encryption scheme for messages in $\{0,1\}^{256}$ (you may assume that H is a random oracle). Give Gen, Enc, and Dec algorithms. There is no need to prove security, and the scheme does not require authentication.

Think about a variation of the El Gamal encryption scheme presented in Lecture 11.

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?