Foundations of Computer Security
Massachusetts Institute of Technology
Henry Corrigan-Gibbs, Nickolai Zeldovich

October 25, 2023
6.1600 Fall 2023
Midterm

Midterm

Question	Parts	Points
1: True or False	7	7
2: User authentication	1	8
3: Signatures	4	8
4: RSA	3	15
5: Lab 1	1	10
6: Key exchange	4	17
7: TLS security	1	6
8: Law and technology	1	6
9: Course survey	3	3
Total:		80

Name: ____

- This is an open book exam: you can use your notes or any material released by us this term. You cannot use the internet.
- Any form of collaboration is *strictly* forbidden.
- If you find a question ambiguous, be sure to write down any assumptions you make.

This midterm exam is printed double-sided!

Problem 1. [7 points] **True or False** (7 parts)
Please answer **T** or **F** for the following. *No justification is needed (nor will be considered).*

(a) [1 point] Implementation bugs can subvert security even if the designers had the correct goal.

(b) [1 point] A one-way function must also be collision-resistant.

(c) [1 point] In practice, we use AES as a collision-resistant hash function.

(d) [1 point] There are message authentication codes (MACs) that provide λ-bit security an that have a MAC tag length of λ bits, under standard cryptographic assumptions.

(e) [1 point] We believe that Lamport signatures (instantiated with a suitable one-way function) will resist attacks by large-scale quantum computers.

(f) [1 point] Let N be an RSA modulus with public exponent $e = 3$. Define the function $F\colon \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ as $F(x) := (x + 7)^3 \bmod N$. Under the RSA assumption, the function F is a one-way permutation.

(g) [1 point] The hash-then-sign paradigm is useful because it avoids the need to use a collision-resistant signature scheme.

Problem 2. [8 points] **User authentication** (1 part)
Ben Bitdiddle runs a popular web site, in which users create accounts using their email address as their username. Ben Bitdiddle is worried about the overhead of storing a separate salt for every user's account in his web site's password database. Ben devises the following alternative plan: his web site will store a single global salt s, and for every user, the database will store the user's username (email address) and $H(s||\mathit{username}||\mathit{password})$.

Is this scheme as good as using individual salts for every password? Explain why, or describe an attack for which this scheme would give an adversary some advantage.

Problem 3. [8 points] **Signatures** (4 parts)

(a) [2 points] What is one benefit of RSA signatures over EC-DSA signatures?

(b) [2 points] What is one benefit of EC-DSA signatures over RSA signatures?

(c) [2 points] What is one benefit of Lamport signatures over EC-DSA signatures?

(d) [2 points] What is one benefit of RSA signatures over Lamport signatures?

Problem 4. [15 points] **RSA** (3 parts)
Let N be an RSA modulus with public exponent $e = 3$ and private exponent d (i.e., $ed \equiv 1 \bmod \phi(N)$). The full-domain-hash signature scheme uses a hash function $H\colon \{0,1\}^* \to \mathbb{Z}_N^*$. The scheme computes the signature on a message $m \in \{0,1\}^*$ as:

$$\sigma \leftarrow H(m)^d \bmod N.$$

(a) [3 points] Your friend (who has taken 6.042 but *not* 6.1600) proposes removing the hash function H from the full-domain-hash signature scheme. With this modified scheme, a signature on a message $m \in \mathbb{Z}_N^*$ is $\sigma \leftarrow m^d \bmod N$. Show that an attacker, given only the public key (N, e), can produce a valid forged signature σ^* on some message $m^* \in \mathbb{Z}_N^*$. Your answer should include a valid message-signature pair: (m^*, σ^*).

(b) [5 points] Consider the full-domain-hash signature scheme instantiated with some hash function H. Say that an attacker can find two messages $m_0, m_1 \in \{0,1\}^*$, such that $H(m_0) = H(m_1)^2 \in \mathbb{Z}_N^*$. Explain how the attacker can use (m_0, m_1) to win the signature security game.

Problem continues on the next page...

(c) [7 points] Model the hash function $H\colon \{0,1\}^* \to \mathbb{Z}_N^*$ as a truly random function. How many times would an attacker have to evaluate the hash function H, on average, to find messages $m_0, m_1 \in \mathbb{Z}_N^*$ such that $H(m_0) = H(m_1)^2 \in \mathbb{Z}_N^*$.

Problem 5. [10 points] **Lab 1** (1 part)
Ben Bitdiddle is hired by the 6.1600 course staff to defend the lab 1 key-value store against the attacks covered in the lab. Ben decides to change how key-value leaf nodes are hashed, by inserting a slash separator between the key and the value, as follows:

```
def H_kv(key, val):
    return H(key + "/" + val)
```

How can you modify the attack in scenario 2 (many fake key-value pairs) so that it still works against Ben's modified design?

Problem 6. [17 points] **Key exchange** (4 parts)
In his final project for his undergraduate security class, Ralph Merkle proposed a key-exchange protocol based on hash functions.

The protocol uses a hash function $H\colon \mathbb{Z}_n \to \{0,1\}^{256}$, where n is on the order of 2^{60}, and proceeds as follows:

- Alice picks $\sqrt{n}$ random numbers $a_1, \dots, a_{\sqrt{n}} \in \mathbb{Z}_n$ and sends $H(a_1), \dots, H(a_{\sqrt{n}})$ to Bob.
- Bob picks $\sqrt{n}$ random numbers $b_1, \dots, b_{\sqrt{n}} \in \mathbb{Z}_n$ and sends $H(b_1), \dots, H(b_{\sqrt{n}})$ to Alice.
- If there exist $i, j \in \{1, \dots, \sqrt{n}\}$ such that $H(a_i) = H(b_j)$:
  - Alice uses a_i as her shared secret with Bob.
  - Bob uses b_j as his shared secret with Alice.

  (If there are many such (i, j) pairs, Alice and Bob use the lexicographically first one.)

Model the hash function as a truly random function.

(a) [2 points] Explain why Alice and Bob will agree on a shared secret with constant probability.

(b) [2 points] How much time does it take Alice to generate her message to Bob? Assume that evaluating $H(\cdot)$ takes a constant amount of time.

Problem continues on the next page…

(c) [3 points] If an attacker eavesdrops on the communication between Alice and Bob, much time does it take the attacker to recover the shared secret $a_i = b_j$?

Problem continues on the next page…

(d) [10 points] Define the *goodness* of a key-exchange protocol to be the ratio:

$$\text{goodness} = \frac{\text{the attacker's running time}}{\text{Alice's running time}},$$

where the attacker's running time is the time required to recover the shared secret.

- What is the goodness of Merkle's protocol?
- What is the goodness of Diffie-Hellman key exchange in $\mathbb{Z}_p^*$ for a large prime p, assuming that it takes $O(1)$ time to multiply two integers in $\mathbb{Z}_p^*$? (In reality the time for a big-integer multiplication grows with p.)
- What is the goodness of elliptic-curve Diffie-Hellman key exchange in a group of prime order p, assuming that it takes $O(1)$ time to perform a single elliptic-curve point operation $\boxplus$?

Problem 7. [6 points] **TLS security** (1 part)
Ben Bitdiddle is designing an Android application where users can send money to each other, by username. The application relies on a central server, which authenticates requests from user devices. When the application wants to transfer some amount of money to another user, it opens a TLS connection to the server, and sends the following message:

```
USER: username
PASS: password
REQUEST: transfer
AMOUNT: amount
RECIPIENT: recipient
```

where `username` and `password` authenticate the sender, and the request asks the server to transfer `amount` to `recipient`'s account.

Explain how a network adversary may be able to redirect a transfer to their account. You can assume the adversary knows the user, the fact that the user is transferring money, how much they are transferring, and to whom, but does not know the user's password. Assume that the TLS certificates are correct (i.e., certificate authorities will not issue an incorrect certificate) and that the adversary cannot guess the user's password.

Problem 8. [6 points] **Law and technology** (1 part)

(a) [6 points] Chris Conley's guest lecture outlined several U.S. acts that regulate federal computer law. Imagine that you have mounted one of the attacks from the 6.1600 labs without permission against a victim server elsewhere on the Internet. Name a U.S. law and an attack from either lab0, lab1, or lab2 that could violate that law. Give a one-sentence explanation for why your chosen attack would violate that law.

Problem 9. [3 points] **Course survey** (3 parts)
Please answer each of these questions in one or two sentences.

(a) [1 point] What lab assignment or lecture should we get rid of next year, in your opinion?

(b) [1 point] Did you think that the lectures so far have been too fast or too slow?

(c) [1 point] What is one topic (on the theme of the course) that you would like to learn more about?