YOUR NAME: ____________________________

Collaboration policy

No collaboration is permitted on this assignment. Any cheating (e.g., submitting another person’s work as your own, or permitting your work to be copied) will automatically result in a failing grade. The Department of Computer Science code of conduct can be found at https://www.cs.utexas.edu/academics/conduct.

Late submission policy

This homework is due at the beginning of class on February 15. All late submissions will be subject to the following policy.

You start the semester with a credit of 3 late days. For the purpose of counting late days, a “day” is 24 hours starting at 11am on the assignment’s due date. Partial days are rounded up to the next full day. You are free to divide your late days among the take-home assignments (3 homeworks and 2 projects) any way you want: submit three assignments 1 day late, submit one assignment 3 days late, etc. After your 3 days are used up, no late submissions will be accepted and you will automatically receive 0 points for each late assignment.

You may submit late assignments to Oliver Jensen, either during office hours or by email. Email submissions must be typed: photographs of hand-written pages will not be accepted.

If you are submitting late, please indicate how many late days you are using.

Write the number of late days you are using: _____
Homework #1: A Trip to Molvanía (50 points)

Molvanía is a small, land-locked republic in Eastern Europe famous for its phishers, spammers, botmasters—and computer security researchers. It also produces 83% of the world’s b33tr00t. Most people get to Molvanía either by air or by accident, but in this homework, we travel there virtually.

Problem 1

For each of the following applications of cryptographic hash functions, explain whether they require one-wayness, collision-resistance, and/or weak collision-resistance.

Problem 1a (2 points)

Jerko poses to Zlad a tough math problem and claims he has solved it. Zlad would like to try it himself, but wants to be sure that Jerko is not bluffing. Therefore, Jerko writes down his solution, appends a long random number, computes its hash and tells Zlad the hash value (keeping the solution and the random number secret). This way, when Zlad comes up with the solution himself a few days later, Jerko can verify his solution but still be able to prove that he had a solution earlier.

Problem 1b (2 points)

The system administrator of a Molvanian mainframe is concerned about possible breakins. He computes the hash values of important system binaries and stores these hashes in a read-only file. A monitor program periodically recomputes the hash values of the protected files and compares them to the stored values. A malicious user who overwrites one of the protected files should not be able to do so without detection.

Problem 1c (2 points)

Cryptographic signatures are produced by computing a hash of a message, then applying a signature function to this hash value. Jerko has a list of messages $m_1, \ldots, m_n$ and their
signatures computed using Zlad’s signing key, but does not have the actual key. Assuming
that the signature function is not susceptible to attack, it should not be possible for Jerko
to present Zlad’s signature on any message other than \(m_1, \ldots, m_n\).

Problem 1d (2 points)

When not composing synthpop hits for Eurovision, Zlad works for the Molvanian Certificate
Authority. He does not have access to the special hardware that computes digital signatures,
but knows the hash function. In addition, Zlad can get messages signed (by applying the
signing hardware to the hash of any message), but every signed message automatically goes
into a log file that Zlad cannot change. Zlad should not be able to produce a certificate
signed by the certificate authority that does not appear in the log file.

Problem 2

MolvaSoft released a video game player called Zbox Half. For a monthly subscription fee,
gamers can join the Zbox Half-Alive online gaming service.

User authentication for Zbox Half-Alive works as follows. When the user first subscribes
for the service, he must establish a password. This password is stored on the Half-Alive server
together with the serial number of the user’s Zbox Half. Afterwards, whenever the user’s
Zbox connects to the server over the Internet, he is asked for his password, which is trans-
mitted in the clear together with the serial number of the Zbox. The server verifies whether
the received password matches the password in its database and whether subscription fees
have been paid for this serial number. If so, it allows the user to connect.

Problem 2a (2 points)

Fyodor a paid-up Half-Alive subscription. He is using a wireless Internet connection for his
3am gaming marathons, and the signal leaks into Jerko’s house (i.e., Jerko can passively
eavesdrop on all messages transmitted to and from Fyodor’s Zbox, but cannot modify them
or introduce new messages). How can Jerko exploit this to connect his own Zbox to the
Half-Alive server for free? Assume that Jerko can modify his Zbox.
Problem 2b (5 points)
Design a user authentication scheme for Zbox Half-Alive based on a cryptographic hash function that prevents passive attackers from exploiting eavesdropped messages between the Zbox and the Half-Alive server.

Problem 3
The sole source of electricity in Molvania is Sjerezo, Europe’s oldest nuclear reactor. Its command-and-control server must authenticate remote logins from reactor operators.

Problem 3a (4 points)
Suppose the password database at the command-and-control server stores users’ passwords as follows:

\[
\langle s_i, H(s_i, p_i, r) \rangle
\]

where \( p_i \) is the password; \( s_i \) is a 12-bit salt generated randomly for each password; \( r \) is a 128-bit secret value, the same for all passwords, kept in a global variable by the login program; \( H \) is a cryptographic hash function like SHA-256.

Does this improve the security of the password database vs. regular salting? Explain why or why not.

Problem 3b (4 points)
Molvanian is a complicated language that takes an average of 16 years to learn, thus Molvanian passwords are not easily vulnerable to dictionary and brute-force attacks.
Suppose that, instead of the above scheme, the server simply stores $x_i = H(p_i)$ and users authenticate via a challenge-response scheme. To log in, the user enters his password $p_i$ into his local machine. The server sends a long, fresh, random challenge $c$, and the user’s machine responds with $y = H(H(p_i), c)$. The server accepts the login attempt if $y = H(x_i, c)$.

Is this protocol secure against eavesdropping and server compromise? Explain.

**Problem 4**

To access his account online at the Bank of Molvanía, a user must install a client program on his Windows 98 PC. The user’s password is set up when the account is created and stored on the bank’s server.

When the user logs in, the client prompts him for his password $p$, computes HMAC of $p$ and current time $t$ rounded to a minute, and sends the result to the server. The server recomputes HMAC using $p$ and its own time. If the resulting value is equal to the value received from the client, the server allows access.

**Problem 4a (5 points)**

Unfortunately, Windows 98 PCs crash a lot and when they crash, the clock resets to midnight, October 28, 1955. Subsequently, the client’s timestamps are all wrong and authentication fails.

The Bank of Molvanía hired George Spelvin, Molvanía’s premier security expert, to fix the problem. Spelvin suggested the following clever modification to this authentication scheme. Instead of the client generating the timestamp $t$, the server sends $t$ to the client as the challenge. The client’s response is computed as before.

Jerko would like the ability to log in as Zlad at 3 am while Zlad is asleep. Is such an attack possible under this authentication system? If so, describe the attack; if not, explain why not (Hint: consider an active man-in-the-middle attacker who controls the network.)
Problem 4b (5 points)

Modify Spelvin’s scheme so that it is secure against an active man-in-the-middle attacker, but still does not require the client to generate its own timestamps. Your solution should use only timestamps, passwords, and HMAC.

Problem 5 (6 points)

Recall that web browsers’ same origin policy (SOP) for DOM access is based on the (protocol, domain, port) triple, and that cookie scope for sending cookies to websites involves domain and path, and cookies marked “secure” are sent over HTTPS only.

Molvanía’s most prominent university is the UniversiTate, generally referred to as UT. They provide all students and faculty with their own web-space.

Oldav Yensen has constructed a web-application to handle grading, which he accesses from https://www.utate.edu.mi/~oyensen/grading. In order to modify his students’ grades, he must first log in at https://www.utate.edu.mi/~oyensen/grading/login, which creates an authenticated session and sets the following cookie:

- **Name:** sessionID
- **Value:** [a fresh random session ID]
- **Domain:** www.utate.edu.mi
- **Path:** /~oyensen/grading
- **Secure:** TRUE

After logging in and modifying a student’s grade, Oldav visits one of his students’ webpage located at http://www.utate.edu.mi/~h4x0r (note that www.utate.edu.mi allows access via http or https, but Oldav has written his grading script to require https)
Problem 5a (4 points)

Can Oldav’s sessionID cookie be stolen by a passive network attacker? How about the student that controls http://www.utate.edu.mi/~h4x0r? In each case, either explain why not, or outline an attack to steal the cookie.

Problem 5b (2 points)

Oldav updates his grading application such that the cookies set httpOnly: TRUE in addition to the above-mentioned properties. How does this affect your answers to problem 5a?

Problem 6

Moola.mi is the sole financial website in Molvanía and thus a target of many phishing attacks. To help protect users from entering their credentials into a phishing site, every Molvanian browser includes a button which, if clicked, executes the following JavaScript:

```javascript
if (top.location.href.indexOf("Moola.mi") == -1) {
    top.location = "Moola.mi"
}
```

Users are trained to always click this button when they come to a website they think is Moola.mi
Problem 6a (2 points)
What is the intended purpose of this JavaScript code?

Problem 6b (9 points)
Describe at least three distinct ways for a phishing website to render this defense ineffective even if the user does click on the button.