CS 361S - Network Security and Privacy
Spring 2017

Homework #1

Due: 11am CST (in class), February 13, 2017

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 Deparment 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 13**. 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 Dillon Caryl, by email or hard copy. **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 Molvanîa (50 points)

Molvanîa is a small, land-locked republic in Eastern Europe famous for its phishers, spamlords, botmasters—and computer security researchers. It also produces 83% of the world's b33tr00t. Most people get to Molvanîa 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 Molvanîan 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 Molvanîan Certificate Authority. He does not have access to the special harware 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 (5 points)

Molvanîan PCs still run Windows 98. Therefore, passwords in Molvanîa are hashed using Microsoft's LAN Manager (LM) hash, which works as follows:

- The password is converted into upper case, null-padded (or truncated) to 14 characters, and split into two 7-character halves.
- Each half is separately converted into a DES key. This key is used to encrypt the ASCII string "KGS!@#$", producing an 8-byte value.
- The two 8-byte values are concatenated, resulting in a 16-byte hash.

Suppose the attacker obtains a file with n hashed passwords. How much work would he need to do to crack these passwords by brute-force search? Show your calculations, and explain your assumptions.

Problem 3 (5 points)

MMACs (Molvanîan Message Authentication Codes) are intended to provide authentication and integrity for email messages between Molvanîan diplomatic missions. All missions share the secret key K. Each message M sent by one mission to another is accompanied by a MMAC, which is constructed as $MH(K, M)$.

MH is a hash function with 320-bit output invented by Molvanîan cryptologists. They took SHA-1 (which is assumed to be one-way and collision-resistant—at least for the purposes of this problem) and used it as a building block to create MH. They even proved that MH, too, is one-way, collision-resistant, and hides the key.

As it turns out, MMAC is completely broken. Someone eavesdropping on a single MMAC-protected message call can gather enough information to forge valid MMACs in the future, that is, to send any message they want accompanied by a forged MMAC that will pass verification by any Molvanîan mission.

How is MH constructed? (Important: your construction must be one-way, collision-resistant, hide the key, and still make the above scheme vulnerable to forging.)

Problem 4 (5 points)

Molvanîan Telecom is selling a fancy smartphone model called jPhone. Each jPhone stores a long, randomly generated secret value. The phone service provider keeps all secrets, together with the corresponding cell phone numbers, in its database.

When a jPhone user wishes to buy a new ringtone, the jPhone transmits its phone number followed by the secret (in the clear) to the ringtone server. The server checks in its database whether the secret corresponds to the provided phone number and, if it does, downloads the ringtone to the phone and bills the account of the phone's owner.

This design is vulnerable to a **cloning** attack. Someone eavesdropping on a jPhone transmission can easily intercept a (*phone number-secret*) pair. He can then hack his own jPhone's transmission software to use the intercepted pair, enabling him to download ringtones which are billed to the victim's account.

Design an authentication scheme for jPhone based on a cryptographically secure hash function that prevents passive attackers from exploiting eavesdropped messages between the jPhone and the ringtone server.

Problem 5

To access his account online at the Bank of Molvanîa, 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[1] 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 5a (4 points)

Unfortunately, Windows 98 PCs crash a lot and when they crash, the clock resets to midnight, January 1, 1980. Subsequently, the client's timestamps are all wrong and authentication fails.

The Bank of Molvanîa hired George Spelvin, Molvanîa'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.

Does this modification have any security consequences? (Hint: consider an active man-in-the-middle attacker who controls the network.)

Problem 5b (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.

[1] This HMAC uses SHA-1, not LAN Manager hash.

Problem 6

The Bank of Molvanîa adopted the following defense against phishing. The first time a user comes to the bank's website, she enters her username and password as usual, and is given a choice between several pictures. The association between the username and the chosen picture is stored in the bank's database. In all subsequent sessions, the user types in her username and expects to be shown a picture. Unless she sees the picture she chose during her first session, she does not type in her password. This helps users avoid giving their passwords to fake websites.

Problem 6a (4 points)

Describe a man-in-the-middle attack that allows a fake website to show the user her chosen picture. (Assume that this is not the user's first session, *i.e.*, she has already chosen the picture.)

Problem 6b (4 points)

Design a cookie-based defense for this anti-phishing scheme that prevents the man-in-the-middle attack you discovered in Problem 6a.

Problem 6c (4 points)

If every user of the bank's website has a cookie identifying her to the bank, does this eliminate the need for passwords? Explain.

Problem 7

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

Distinguished professor 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' web-page 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 7a (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 7b (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 7a?