Homework 6: Potpourri

Due: Friday, April 24 @ 11:59 pm EDT

Do not include any identifying information (name, login, Banner ID, etc.) on your handin. When you’re done, please submit a PDF file to Gradescope where your answer to each problem is on a separate page.

Reminder: While we encourage everyone to discuss and work through the problems on the homeworks, please remember that you are not permitted to take away notes (hand-written or electronic) from any collaboration sessions—your final solution must be written up independently without relying on shared answers or notes from discussions. Additionally, you are permitted to consult online resources, but you must cite the resources you consulted during your problem-solving process.

Potpourri

Problem 1: Compression and Encryption

One practical consideration of networking that we’ve abstracted away in the course so far is data compression. To improve performance, clients will often use a compression algorithm to reduce the size of data prior to sending it over the network. At a high-level, compression algorithms work by identifying repeated substrings in a message and replacing equivalent substring instances with some short, unique sequence of bits. To uncompress the message, we can later replace the sequence of bits with the actual substring they originally replaced. In this technical question, we’ll examine the implications of compression on confidentiality mechanisms like SSL and TLS.

The CS166 staff has developed a grading server on which they run autograding scripts against student submissions. Given the sensitive nature of student grades, we utilize a one-time-pad scheme to provide perfect secrecy on all communication with the server.

To interact with the server, TAs invoke a script that consumes (on stdin) a student name to grade. This sends an encrypted message containing the username as well as a fixed, TA Secret to the grading server. If you could extract the TA Secret, perhaps you could take over the grading server yourself! You’ve found a copy of the script (which you can run with the binary at /course/cs1660/student/<your-login>/hw6/script):

```python
secret = "XXX-XX-XXXX" # Redacted; the X's are numbers from 0-9
while True:
    name = raw_input("") # Read from standard input
    message = "Grade %s; my TA identifier is %s. Repeat %s." % (name, secret, secret)
    compressed = zlib.compress(message)
    ciphertext = one_time_pad(compressed)
    send_to_grading_server(ciphertext)
    print(ciphertext.encode("hex")) # Prints the ciphertext as a hexadecimal string
```

Assume that the one_time_pad never runs out of pad material and is truly random.

Question a) Consider messages $A = \text{"here is a long message"}$ and $B = \text{"roberto!! or roberto!!"}$. When uncompressed, both $A$ and $B$ are the same length. Will $A$ take up more, less, or the same space as $B$ when both messages are compressed? (No explanation needed.)

Question b) Is it possible to determine the value of the TA Secret in polynomial time in the length of the secret? If yes, provide the TA Secret, the source code of any programs you used to derive it, and a justification of the polynomial runtime of your attack. If not, explain why such an attack is impossible. (Hint: Compare what happens when you use the 19-character string "my TA identifier is" as the input to the script to what happens when you send a string of 19 random characters—for example, "asdfghjklqwertyuiop".)
Problem 2: Onion-Flavored Handins

Recall Tor, a system for anonymous web browsing. Some of the CS166 TAs have established their own Tor network (depicted in Figure 1) to allow students to submit their handins “super-anonymously” on Gradescope. Students submitting a super-anonymous handin do not need to log in and no identifying information is associated with the application-level data sent to the Gradescope server. (This means TAs don’t know who handed in a given super-anonymous handin.)

Figure 1: Tor network set up by the CS166 TAs comprising 10 nodes depicted by circles with IDs 1 through 10. Depicted with squares are the client machine of a student, with ID 0, and the Gradescope server, with ID 11. The Tor node with ID $i$ ($i = 1, \ldots, 10$) has public key $PK_i$, which is known to all other Tor nodes and the client. For simplicity, we assume messages sent to Tor nodes are encrypted with public-key encryption while in reality, a combination of public-key and symmetric encryption is used.

**Question a)** The arrows in Figure 1 depict a Tor circuit between you and the Gradescope server. Assume that the format of each network packet is [next-hop-id, data]; for example, to send message "Hello" to node 1, one would transmit network packet $[1, \text{"Hello"}]$. Also, assume that the encryption of a message $M$ with public key $PK_i$ is denoted as $E(PK_i, M)$. To send a message $M$ to the Gradescope server, what is the packet your Tor browser sends out in this circuit?

**Question b)** Bernardo uses HTTP (not HTTPS) when using the CS166 Tor network. Bernardo says TLS is not necessary to preserve confidentiality (that is, no one can read $M$ other than Bernardo and the Gradescope server) because the onion packets are already encrypted. Given this setup, is the confidentiality of $M$ maintained in the presence of a powerful attacker who eavesdrops the communication between any two machines on the network? Explain.

**Question c)** Mariya actively controls the Gradescope website and a router in the CIT through which many students simultaneously connect to the Gradescope site via Tor and HTTPS (see Figure 2). Is it possible for Mariya to determine which student submitted which super-anonymous handin? Explain. (Hint: Recall that the Tor circuit used by each student is randomly chosen, and thus the “wire distance” that messages travel along the Tor network is different for each student.)

![Multiple students using the CS166 Tor network through the CIT router.](image)

**Question d)** Consider a modified version of the scenario from part (c) where you are the only user of Tor and Mariya is now a passive attacker (specifically, she cannot modify or add traffic sent from the Gradescope server). Does using the Tor network ensure that Mariya cannot identify your computer as the one who hands in a given super-anonymous handin? Explain.
CS166  Computer Systems Security  Spring 2020

Problem 3: Anonymization Ethics

In recent years, Tor’s anonymity has become a hotbed for ethical questions and issues, particularly revolving around its history of funding, its potential to be used as a tool for surveillance, and its usage as a platform to promote different areas of crime (drug trade, child pornography, etc.). Please read these articles:

- Wired article about the CIA creating its own hidden service on Tor: [https://www.wired.com/story/cia-sets-up-shop-on-tor/](https://www.wired.com/story/cia-sets-up-shop-on-tor/)

The following questions are open-ended and will be graded for thought and justification. Please reference and/or quote specific arguments from the readings in your answers.

**Question a)** From the first The Guardian article, please state three entities that funded Tor in 2013, as well as the amount funded. Do your own research to list three other entities that have funded Tor, as well as the years during which they were funding the project. Please cite your sources. 

(Note: Wikipedia is not considered a valid source.)

**Question b)** What are some positive and negative consequences of the creation of Tor? Do the positives outweigh the negatives? Feel free to do your own research on this topic (and cite your sources!).

**Question c)** Using the second The Guardian article and the Vice article, compare and contrast the FBI and the NSA’s methodology of de-anonymizing and targeting Tor users. Was it ethical for these government entities to unmask Tor users?

**Question d)** Using the Wired article, explain why the CIA created its own hidden service. What logistical challenges arise from usage of this service? What potential malicious usage could occur? Is there an alternative platform besides an onion service that would better serve the CIA’s need?

Problem 4: Houseplant Division on the Blockchain

Recall Alice and Bob, two former lovers who do not trust each other (as we saw in Homework 2). While Alice and Bob might not miss each other, they do miss their collective shares of houseplants, and thus, Alice and Bob would like to come up with a scheme in which the following happens: at the beginning of each month, a coin is flipped that determines which person gets the collective share of houseplants for the month.

However, they do not want to use the commitment scheme we derived back in the Cryptography lecture—as we saw on Homework 2 and Midterm 1, these kinds of “commitment schemes” are vulnerable to forms of cheating that cannot fully be mitigated without the help of a third party. These schemes also require Alice and Bob to actively communicate with each other, which, given their tumultuous past, they would rather avoid as much as possible!

Luckily, Alice and Bob are Bitcoin enthusiasts, so they have agreed on the following scheme that utilizes the Bitcoin blockchain to emulate a coin flip without exchanging messages: for the first block added to the blockchain after the month starts (in some agreed on timezone), the last bit of the root hash (i.e., the hash of the transactions in the block, which is part of the block header) is the outcome of the coin flip.

**Question a)** Explain why the scheme deployed by Alice and Bob reasonably emulates an unbiased coin flip.

**Question b)** Unbeknown to Bob, Alice is a closet billionaire who made her fortune trading cryptocurrencies. Show how Alice can use her wealth to influence the outcome of the monthly coin flips.
Problem 5: Mining Pools

In a Bitcoin mining pool, the pool administrator assigns to each worker in the pool a range of nonces to use in trying to solve the current puzzle. The worker who finds the solution reports the successful nonce to the pool administrator, who then shares the block reward with all the workers in the pool in proportion to the work they have performed. What prevents a worker who has found the solution to the puzzle from taking the entire block reward themselves?