Total points: 100 (+0 bonus)

Assignment 2

Due: 11:59 PM, 24th April

Written assignment

1. Bob runs a social network web server that requires users, such as Alice, to log in. He wants to use cryptography to protect Alice's account. However, he doesn't understand cryptography very well. For each of the below uses of cryptography, identify his mistake, and explain what he should do instead.

   (a) [3 points] Bob obtains a public/private encryption key pair using RSA. When Alice visits his site, Alice generates a 128-bit secret key intended for AES in CBC mode, encrypts it using Bob's RSA public key, and sends it to Bob. Bob asks Alice to use a SHA-256 HMAC based on the same 128-bit secret key to authenticate this message so as to prove that it is indeed Alice who sent the message.

   (b) [3 points] To establish trust, Bob asks a CA to sign his public 256-bit RSA key using the CA's private 256-bit ECC key. The CA's public 256-bit ECC key is in Alice's browser, so Alice can verify Bob's key automatically when she visits his site, even though she does not explicitly know who the CA is.

   (c) [3 points] For Alice's login, Bob requires Alice to hash and salt her password on the client side using SHA-512, and then send it to Bob using 128-bit AES. Then, Bob will store Alice's hashed password and the salt in his database. In future attempts, Alice can use the same hashed password and salt to login.

   (d) [3 points] To store Alice's password securely, Bob uses AES encryption with a secret key and a 128-bit IV to encrypt her password. A CRC32 checksum is used to ensure correctness against random bit flip errors.

2. [9 points] For each of the following network-based attacks in the left column, find the most fitting network defense in the right column. Explain why.

Attack	Defense
IP spoofing	Proxies
Eavesdropping	Deep Packet Inspection
Teardrop attack	Ingress/egress filtering

3. [9 points] When a client C accesses server S through Tor, she usually builds a circuit of three nodes: N_1, N_2, and N_3. A connection is established as follows:

$$C \to N_1 \to N_2 \to N_3 \to S$$

N_1 is also known as the entry node or the guard node, and N_3 is also known as the exit node. Visit `metrics.torproject.org` to answer the following questions:

   (a) [3 points] Give the total amount of advertised bandwidth of relays with the "Guard" flag (but not the "Exit" flag) and relays with the "Exit" flag (but not the "Guard" flag) on 2020-02-01. Which is more? Give one reason to explain this phenomenon.

   (b) [3 points] Give the median download rate of a file (in bits per second) for a 50 KiB file and a 5 MiB file to the `op-hk` onion server on 2020-02-01. Can you explain the difference?

   (c) [3 points] What is a disadvantage of using three nodes in a Tor circuit instead of one node? What is an advantage of doing so?

4. **Superfish** [10 points]

   In February 2015, researchers found that Lenovo computers came pre-installed with Superfish. Superfish adds itself as a root certificate authority to the computer. Every time a user visits an HTTPS site with a valid certificate, Superfish instead presents a fake certificate for the website to the user, and the browser would automatically trust such certificates. (The website's real certificate is intercepted and never presented to the user.) Using the fake certificate, Superfish identified users' browsing patterns and added or changed advertisements on web pages, creating a scandal for Lenovo. In the same month, Windows Defender started removing Superfish, and Lenovo ended their partnership with Superfish.

   (a) [2 points] Classify Superfish as a type of malware 1) by method of spread, and 2) by effect on system.

   (b) [3 points] Explain why Superfish is able to change the contents of an encrypted web page, compromising integrity. (Think about it this way: If the user is talking to the website using the website's encryption key, then Superfish certainly cannot change the contents because Superfish doesn't have the website's decryption key. So...)

   (c) [3 points] Superfish used the same signing key in every laptop computer. If an attacker wants to impersonate a website, explain how the attacker can impersonate any website against a victim who has Superfish installed.

   (d) [2 points] How can a careful user notice that their computer is infected with Superfish by looking at the browser? (We did this activity in class, albeit not for Superfish.)

Programming assignment

Breaking Cryptography [60 points]

In this assignment, we will write programs to automatically break some weak ciphers. For every part of the program, you may assume the plaintext to satisfy the following. All characters are ASCII characters with the following byte values, all ranges being inclusive of both ends:

- Symbols: 32 to 34, 39 to 41, 44 to 59, 63.
- Capital letters: 65 to 90.
- Small letters: 97 to 122.

This also implies that no file contains a newline.

Please make sure to read the submission instructions carefully.

(a) [20 points] Two files, `ctext0` and `ctext1`, have been sent to you by e-mail. Those two files were encrypted using the same one-time pad. They are exactly 400 bytes each, and they both come from English Wikipedia articles. Find the contents of both files using crib-dragging, and submit them as `ptext0` and `ptext1`; you can reverse the two files.

(b) [30 points] (+5 Bonus) Based on your experience in the previous part, create a program to automatically break two-time pads. The program can assume `ctext0` and `ctext1` are in its folder, and that they have the same file length. Your code should be called `ttp_crack.{cpp, py, java}` . (See submission instructions on how to ensure your code can be compiled correctly.) Your code should generate `de_ctext0` and `de_ctext1`, which are the corresponding plaintexts for the two ciphertexts; you can reverse the two files. **Note that these two file names are not the same as the previous part.**

Your code can assume the possible ASCII byte values are the same as before. You are allowed to submit any additional files that are necessary for your code to run. Your code will be evaluated on a large set of two-time pad ciphertexts and scored based on what percentage of each plaintext has been found; therefore, even if you are only able to find one word for a ciphertext pair, output it in the plaintext files in the correct byte range, filling in the remaining space with anything. This means that, for example, if you don't know the first four characters, but you know the 5th to 9th ones are "whose", then you can write "0000whose" in the file. It should not spend more than 10 seconds on each ciphertext pair; otherwise, the process will be killed. It cannot access the internet. It can rely on **exactly one file of up to 10 MB called "dictionary"**; any other submitted files must be code.

The grading criteria is as follows.

- If your code can solve x% of the testing ciphertexts, your grade will be at least x% of 30. Your code will be considered to solved a ciphertext if you satisfy

at least one of the following: 1) You found 25% of all bytes; 2) You found 8 consecutive words. If you achieve these objectives partially, you will get a partial portion of that ciphertext's points.

- If your code performs better than x% of submissions, your grade will be at least x% of 15 + 15.

If you cannot finish this part, write down all of your attempts and ideas, and submit what code you wrote.

(c) [10 points] To show the weakness of a short key, write a program to crack a ciphertext which has been encrypted with an 128-bit AES key but the first 108 bits have been set to 0 (so only the last 20 bits are randomized). The encryption was done by first converting the plaintext to a bitstring using ASCII, dividing it into blocks of 16, and encrypting the text using ECB mode. The plaintext has a multiple of 16 bytes. The ciphertext is called `aestext`. Your code should be called `aes_crack.{cpp, py, java}`. and should generate the corresponding plaintext `de_aestext`. A sample `aestext` file has been uploaded; it will not be the same as the file used to test your code. Your program can spend at most 30 seconds, after which the process will be killed. You can use someone else's AES implementation here, though you are welcome to implement it.

Submission instructions

All submissions should be done through the CASS system. Submit the following files:

- `a2.pdf`, containing all your written answers.
- `ptext0` and `ptext1`, for part (a) of the programming assignment.
- `ttp_crack.{cpp, py, java}`, for part (b) of the programming assignment, as well as any other code necessary to run it, plus one file of up to 10 MB called `dictionary`. This may include a Makefile. Submit your code; do not submit any compiled files.
- `aes_crack.{cpp, py, java}`, for part (c) of the programming assignment, as well as any other code necessary to run it. This may include a Makefile. Submit your code; do not submit any compiled files.

To run ttp_crack, for example, I will do the following:

C++: I will compile `./gcc ttp_crack.cpp -o ttp_crack` and then run `./ttp_crack`.

Py: I will call `python ttp_crack.py`.

Java: I will compile `javac ttp_crack.java` and then call `java ttp_crack`.

If there is a Makefile in your folder, the Makefile will override all of the above. I will call `make` to compile the code, and then I will call `make run`.

Keep in mind that plagiarism is a serious academic offense; you may discuss the assignment, but write your assignment alone and do not show anyone your answers and code.

The submission system will be closed exactly 48 hours after the due date of the assignment. If you want to use this extension, which has no penalty, you must write an e-mail to me before the due date of the assignment. You will receive no marks if there is no submission within 48 hours after the due date.