Total points: 80 (+5 bonus)

Assignment 3

Due: 11:59 PM, 22nd May (Fri)

Written assignment

1. We have four data privacy techniques:

    1. k-anonymity
    2. Differential privacy
    3. Secure multiparty computation
    4. Private information retrieval

    For each scenario below, discuss whether **each** of the four data privacy techniques would be suitable to resolve the challenge, and explain why if they are not. It is possible to have multiple suitable ones or no suitable ones. **This means the answers to a, b, c, d should have 4 parts each.**

    (a) [4 points] People in country A are anxious to know if they have been in recent contact (¡ 20m) with someone confirmed to have caught an infectious disease called CROW, but the identity and exact location of CROW patients is private and there are so many cases that the government of country A cannot manually warn all potential contacts. Two big tech companies are trying to create an app to satisfy this need in a privacy-preserving manner with Bluetooth beaconing.

    (b) [4 points] Lacking exercise due to CROW quarantine, you want to buy a new smart device to encourage a healthy lifestyle by measuring your calorie intake. Specifically, you want to be able to look up the nutritional information of any food using your device at any time. However, you are reluctant to reveal your personal eating habits to the company that owns the device. A new company making these smart devices is willing to use a data privacy technique to protect your privacy.

    (c) [4 points] With more free time, you want to start a business that can be run from home, and you would like to purchase its web domain as the company's website. However, you are aware of the practice of cybersquatting; people may purchase the domain first if they know it is in demand, and sell it to you at an elevated price. You want to know if the domain is still available, but you are worried that attempting a DNS query for the domain will lead to some DNS servers purchasing it immediately for cybersquatting. Some helpful DNS servers recognize your need and will cooperate with you if you can identify a useful data privacy technique to use with them.

(d) [4 points] A social media company knows that people are feeling lonely, so it is offering a monetary reward for a challenge: design the best algorithm that can identify which people should be recommended to talk to each other to become online friends, based on some data about their real-life habits, online usage patterns, interests, and other personal data. However, they are aware that simply releasing everyone's raw data for this challenge is a violation of their privacy.

2. In the Multi-Level Security model used by a certain company for confidentiality, there are four levels of security:

   - Management: Can read all files. Member: Ma.
   - Accounting: Can read all Accounting and Basic files. Member: Ace.
   - Finance: Can read all Finance and Basic files. Member: Finn.
   - Basic: Can read all Basic files. Member: Batu.

   We define a security level B to be "lower than" security level A if the set of files B can read is a strict subset of the set of files A can read. A security level is only equal in security to itself. Note that this means Accounting is not higher than or equal to Finance and Finance is not higher than or equal to Accounting.

   The high-water mark Bell-LaPadula model is used here, defined as follows: If a person at security level A writes to a file at security level B, then the security of the file must be raised to a level that is allowed to read both A and B; if there are multiple such levels, choose the lowest one.

   These operations are attempted in a day on a file starting at the Accounting security level, in the following order:

   1. Finn tries to read the file.
   2. Batu tries to write to the file.
   3. Finn tries to write to the file.
   4. Ace tries to write to the file.
   5. Ace tries to read the file.
   6. Ma tries to read the file.

   (a) [3 points] For each operation, describe if the operation is successful and state the new security level (if any) of the file.

   (b) [5 points] Prove that Ace can never read a file after Finn writes to it, no matter what level the file started at and no matter what operaions are done before, between or after. This preserves confidentiality. (Hint: You may want to revisit your proof here after part c to see if it holds up.)

   (c) [3 points] Let us change the Accounting level rule to: Accounting can read all Accounting, Management and Basic files. Redo the above operations on a new Accounting file, describing if the operation is successful and state the new security level (if any) of the file.

(d) [5 points (bonus)] State a necessary and sufficient condition for the security levels of a Multi-Level Security system using the High-water mark Bell-LaPadula model such that it is always true that the security of any file can only increase or stay the same, no matter what operations are done in what order; prove it.

3. Assume the following numbers describe the Bitcoin ecosystem:

   I. The size of each block is at most 1 MB (= 1,048,576 Bytes), and there is exactly 1 block per 10 minutes. The reward for a successful block hash is 6 bitcoins.

   II. Each transaction is at least 166 bytes in size. A block may contain nothing but transactions.

   III. In total, all miners produce 10^{20} hashes per second. All miners cooperate to ensure that no two miners will attempt the same hash.

   IV. It costs \$8,000 to buy a mining device to produce 10^{16} hashes per second, and \$0.002 to run it per second for every miner.

   (a) [2 points] What is the minimum price of a bitcoin such that miners avoid making a running loss using the given mining device, if there are no transaction fees?

   (b) [2 points] What is the maximum number of transactions per second?

   (c) [2 points] Now, suppose that the price of a bitcoin stays the same as your answer in (a), but the reward for a successful block hash has been halved to 3 bitcoins. What is the mean transaction fee (in \$) required for each transaction so that miners continue to avoid making a running loss? (Show your workings clearly so that an error from (a) or (b) will not affect your marks for this part.)

   (d) [2 points] Suppose the price of a bitcoin is fixed at \$2,500, there are no transaction fees, and the reward for a successful block hash is still 6 bitcoins. What is the minimum number of Bitcoin mining devices you would need to buy and run to earn at least \$1,000,000 profit in 365 days? Note that your purchases here will affect the total hashing rate above in (III). (The cost of purchase and the cost of running should both be counted as loss.)

Programming assignment

Error Correcting Codes [45 points]

Error Correcting Codes (ECCs) can be used in unsafe transmission or storage devices to improve reliability by providing redundancy. Unlike checksums, ECCs can automatically correct errors up to a certain number of bits. In this question, you will be asked to implement the general Hamming code, which can correct any 1-bit error in a general string. The specifications of the Hamming code are as follows.

1. Write the input data string as a bitstring according to ASCII.
2. Choose enough parity bits and intersperse the data bits with parity bits, so that the i-th parity bit p_i is followed by exactly $2^{i-1} - 1$ data bits. Denote the i-th bit of H as H_i. Denote the length of H as $|H|$.
3. Let $B(x)$ be the bitstring representation of the integer x, for example, $B(5) = 00101$.
4. Define each M_i to be a set of specific bit positions in H, such that M_i is the set of integers $1 \leq x \leq |H|$ where the i-th least significant bit of $B(x)$ is 1.
5. Set the parity bit p_i to be the XOR of all the bits of H in the positions indicated by M_i, except i itself:
$$p_i = \bigoplus_{x \in M_i \setminus \{i\}} H_x$$

Here is a more detailed explanation of M_i. Consider the integer $x = 10$, so $B(10)$ is 1010. Then the 1st least significant bit is 0, the 2nd is 1, the 3rd is 0, and the 4th is 1 (reverse the bitstring). So x belongs to M_2 and M_4, not to M_1 and M_3.

To work out an example:

1. The input data string is "ab", which becomes 0110000101100010 (16 data bits).
2. We need five parity bits, p_1, p_2, p_3, p_4, p_5, as follows:
$$H = p_1 p_2 0 p_3 110 p_4 0001011 p_5 00010$$
p_5 can be followed by up to 15 data bits, but we ran out of data bits, so we stopped there. $H_3 = 0$ and $H_8 = p_4$. $|H| = 21$.
3. For example, $B(20) = 10100$.
4.
$$\begin{aligned} M_1 &= \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21\} \\ M_2 &= \{2, 3, 6, 7, 10, 11, 14, 15, 18, 19\} \\ M_3 &= \{4, 5, 6, 7, 12, 13, 14, 15, 20, 21\} \\ M_4 &= \{8, 9, 10, 11, 12, 13, 14, 15\} \\ M_5 &= \{16, 17, 18, 19, 20, 21\} \end{aligned}$$

5.

$$
\begin{aligned}
p_1 &= H_3 \oplus H_5 \oplus H_7 \oplus H_9 \oplus H_{11} \oplus H_{13} \oplus H_{15} \oplus H_{17} \oplus H_{19} \oplus H_{21} &= 0 \\
p_2 &= H_3 \oplus H_6 \oplus H_7 \oplus \ldots &= 1 \\
p_3 &= H_5 \oplus H_6 \oplus H_7 \oplus \ldots &= 0 \\
p_4 &= \ldots &= 1 \\
p_5 &= \ldots &= 1
\end{aligned}
$$

(a) [15 points] Write a program that will automatically generate the correct parity bits for any input string. The input string will be the first argument of the program. Your program, called **a3a**, should write the parity bits in the correct order to `stdout` (e.g. cout or print) as **characters**, "0" or "1". Continuing the above example, if the program is called with:

```
./a3a ab
```

The program should display:

```
01011
```

(b) [20 points] Write a program that will automatically correct any one-bit error in either the parity or the data string. The parity will be the first argument of the program, and the data string will be the second argument of the program. Your program, called **a3b**, should output the *corrected* parity bits to the first line of `stdout`, and the *corrected* data string to the second line of `stdout`. If there is no correction to be made, you should still output the original parity bits and the data string. For example, if I run your program with:

```
./a3b 01011 ac
```

Your output should be:

```
01011

ab
```

The program will always be tested with the correct number of parity bits, and there will either be a one-bit error in the parity or the data string, or no error. Either type of error should be fixed. Also, note the meaning of an ASCII one-bit error: "a" (01100001) can become "i" (01101001), but "a" cannot become "b" (01100010) because that would be two bit flips.

(c) [10 points] Submit two sentences in files `sentence1` and `sentence2`, such that they have the same Hamming code (as output by part a). They should all be logical and grammatically correct sentences with no typos, and they should all include your @connect.ust.hk username. Any ASCII characters that are not letters, numbers, or basic punctuation will be removed.

Hint: It may be easier to construct these sentences if you use numbers somewhere in your sentence. Keep in mind that you should not look at anyone else's sentence, so there shouldn't be a case where two students submit similar sentences.

Submission instructions

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

- `a3.pdf`, containing all your written answers.
- Files for the programming assignment: `sentence1`, `sentence2`.
- Code for the programming assignment, detailed below:

For the programming assignment, submit your code; do not submit any compiled files.

C++: Submit a3a.cpp, a3b.cpp. I will compile them and call `./a3a <message>`.

Python: Submit a3a.py, a3b.py. I will call `python a3a.py <message>`.

Java: Submit a3a.java, a3b.java. I will compile with `javac a3a.java` and then call `java a3a <message>`.

If there is a Makefile in your folder, the Makefile will override all of the above. This implies if you are not writing in C++, Python, or Java, you must include a Makefile.

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

The submission system will be closed exactly 48 hours after the due date of the assignment. Submissions after then will not be accepted unless you have requested an extension before the due date of the assignment. You will receive no marks if there is no submission within 48 hours after the due date.