Total points: 80 (+5 bonus)

Assignment 3

Due: 11:59 PM, 7th Aug (Fri)

Written assignment

1. [5 points] A data set allows a differentially private mean query with positive one-sided Laplacian noise:

$$Q(D) = mean(D) + abs(x)$$

In the above, *mean* is the deterministic mean function that returns the exact mean, and $x \sim \mathcal{L}(b)$ is drawn from the Laplacian noise function with scaling factor b. The Laplacian noise function has pdf $f(x, b) = \frac{1}{2b} exp(\frac{-|x|}{b})$. The maximum value for any entry in this data set is one million.

Prove that $Q(D)$ does not achieve ϵ-differential privacy for any ϵ.

Hint: With a counterexample, this question can be answered in one to two sentences.

2. We have four data privacy techniques:

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

    For each scenario below, two of the four data privacy techniques will be proposed to resolve the challenge. Choose the correct one, explain why it is suitable, and why the other choice is not suitable.

    (a) [4 points] You want to buy a new smart device that encourages a healthy lifestyle by monitoring your daily exercise. The device needs to track your movement on a map to know how much calories you are actually expending (e.g. hiking and swimming is different from walking). However, you consider this to be a privacy risk: you do not want the app to know where you are at all times. A new company making these smart devices is willing to use a data privacy technique to protect your privacy. Proposals: differential privacy, PIR.

    (b) [4 points] You would like to purchase a new web domain. 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.

To assure potential customers that it is not malicious, a DNS server is willing to cooperate with you and implement a privacy-preserving algorithm. Proposals: k-anonymity, PIR.

(c) [4 points] A password manager service wants to implement zero-knowledge authentication. In this system, the user will never send the master password to anyone, even for authentication. The service will only ever see the hashed password with a server-stored salt. Nevertheless, the user needs to be able to prove that they know the master password for authentication; upon authentication, the user will be able to fetch password that were encrypted using the master password hash as key. Proposals: PIR, SMPC.

(d) [4 points] A hospital has information about millions of carcinogenic and non-carcinogenic patients in various locations. The exact location, if revealed, is a privacy breach. Researchers want to determine if there is a relationship between cancer and building materials; they already have a large database of building materials used in different areas. Researchers do not know what kind of relationship there could be yet. Proposals: k-anonymity, SMPC.

(e) [4 points] A new service has been developed to allow users to compare salaries and potentially identify unfairness or discrimination in pay. After passing a local verification process, users can submit their salaries to the service, and the service should be able to inform them if they are being underpaid or not by comparing with other users of the same profession/experience (once there is enough data). The actual salary is considered sensitive information and should not be exposed to the service. Proposals: differential privacy, SMPC.

Programming assignment

Deanonymizing anonymization

As we saw in class, data anonymization can be flawed, and implementation errors can occur. With partial knowledge of the data set, leaked information from anonymized data can be used to recover missing data points. In this assignment, you will write code to deanonymize supposedly anonymized data.

(a) (25 points + 5 bonus) Your code, a3a-guesser, will take two inputs: a partially filled data set as the first input, and an anonymized data set using k-anonymity with $k = 3$ as the second input. Sample files are included as a3a-partial.csv and a3a-anonymized.csv. Do not assume that these will always be the filenames, the filenames should be read as system arguments. Specifications are as follows:

- Both files are CSV files. The delimiter is comma (,) and quote character is double quotations ("). All characters are in ASCII.
- There is one PID column called "Age", and one sensitive attribute column called "Condition". The Age is an integer from 0 to 150 inclusive and the Condition is one word (no spaces). There are no other columns.
- In the partial data set, there will be full data points and partial data points. For Age, a full data point is an integer, and a partial data point can be:
  - A set, written as e.g. [2,3,5,8,13], which means it is one of the five possible ages.
  - A range, written as e.g. 18-21, which means 18, 19, 20, or 21. - is the minus symbol (ASCII code 45). A set will never contain a range.
  - An asterisk, written as *, which means it is unknown.
- In the partial data set, conditions will only either be a full data point (one word) or an asterisk. The number of rows in the partial data set may be fewer than or equal to the number of rows in the anonymized data set.
- In the anonymized data set, Age has been anonymized into a set or range of possible values using the script a3a-anonymizer.py on the real data set.

Based on the anonymized data set and knowledge of the anonymization algorithm, recover as much of the partial data set as possible. It is possible that some data points cannot be fully recovered, and your code should write down as much as it can recover. In other words, your output can also contain ranges and sets, but you cannot combine ranges and sets in one data point.

Output into a3a-deanonymized.csv, which must be a CSV file. Windows CRLF style line endings are acceptable. It needs to be in the same order as the original input.

Grading. There will be twelve test inputs. The first five test basic deduction. The next five test more complicated deduction such as multiple partial data points and the

ability to recognize when deduction is not possible. The final two will test particularly clever deduction attacks for bonus marks. Each test input is worth 2.5 points.

Hint: It is possible to fully recover all data points in the sample a3a-partial.csv. The sample is on the same level of difficulty as a bonus test input, though being able to solve the sample does not necessarily guarantee full marks.

(b) (30 points) Your code, a3b-guesser, will take two inputs: a partially filled data set as the first input, and a set of query answers from a query-answering service that attempts to implement differential privacy. Sample files are included as a3b-partial.csv and a3b-queries.csv. Do not assume that these will always be the filenames, the filenames should be read as system arguments.

The service supports the following queries: MIN, MAX, MEDIAN, SUM, TOPTWO. The service allows you to make each query three times; the three answers for every query are written into the second file, for example a3b-queries.csv.

- In the partial data set, there is only one column called "Salary". The Salary is an integer from 0 to 1,000,000 inclusive. It is unsorted.
- There will only be one missing data point in the partial data set, masked with an asterisk. The partial data set will always have the same number of rows as the (hidden) full data set.
- There are no ranges or sets in this question.
- a3b-queries.csv will always contain all three answers for all five queries.

Recover the missing data point and write the full data set in a3b-deanonymized.csv, which must be a CSV file. Windows CRLF style line endings are acceptable. It does not need to be sorted.

Grading. For each flaw, there will be 3 test inputs to check if you are able to use the flaw to recover information. It is not always possible to recover the exact correct value for the missing data point; in cases where exact deduction is not possible, your answer will be accepted if it is within +/- 1000 of the correct answer.

Hint: Each query has a flaw. A total of five flaws (15 inputs) will be tested.

Submission instructions

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

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

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

C++: Submit a3a-guesser.cpp and a3b-guesser.cpp. I will compile them and call (e.g.) `./a3a-guesser <partialfile> <anonymizedfile>`.

Python: Submit a3a-guesser.py and a3b-guesser.py. I will call (e.g.) `python3 a3a-guesser.py <partialfile> <anonymizedfile>`. Ensure that your code works with Python3.12.

Java: Submit a3a-guesser.java and a3b-guesser.java. I will compile with `javac a3*-guesser.java` and then call (e.g.) `java a3a-guesser <partialfile> <anonymizedfile>`.

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.