Boston University, Computer Science.
Midterm Spring 2012.

Instructor: Sharon Goldberg
March 29, 2012. 1-2:20 AM.

- One two-sided hand-written aid sheet allowed.
- Be specific and precise with your answers.
- Show your work. Answers without justification will be given little credit.
- Please clearly indicate which parts of your solution you want graded.
- You can use the back of each page as a scratch paper. We will only grade the work you do on the exam pages unless you specifically tell us to do otherwise.

Good luck!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td><strong>Bonus</strong> 5</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>24</td>
</tr>
</tbody>
</table>
1 Privacy

Problem 1. Consider the following algorithm (written in PINQ):

```csharp
var ageList = new List<int>() { 12, 13, 14, 15, 16, 17};

var perAge = data.Partition(ageList, x => x.age);

for (int i = 0; i < ageList.Length; i++)
{
    var countPerA   = perAge[i].GroupBy(x => x.personsName)
                        .NoisyCount(0.025);

    var sickCountPerA = perAge[i].Where(y => y.sick == true)
                       .GroupBy(z => z.personsName)
                       .NoisyCount(0.025);

    var percentSickPerA = sickCountPerA / countPerA * 100;

    Console.WriteLine( "Age " + ageList[i] + " percent sick: " + percentSickPerA );
}
```

1. (4 points) You are given a privacy budget of 0.1. Does the algorithm exceed your privacy budget? Make sure to justify exactly how you arrived at your answer (else no partial credit can be given!)

   Ans:

2. (2 points) Let the standard deviation of `sickCountPerA` be $\sigma$. Determine $\sigma$.

   Ans:

3. (2 points) True or false? The standard deviation of `percentSickPerA` is $2\sigma$. Justify your response.

   Ans:
Problem 2. (4 points) Here is an algorithm to compute the probability distribution function (PDF) of a dataset. (This pseudocode is not written in PINQ!).

// let 'data' be a list of integers taking on values in the set [0, 120]
// assume the fact that this data lies in the range [0, 120] is public knowledge.

var countPerI = new int[121];

for (int i = 0; i < 121; i++)
{
    countPerI[i] = 0;
    
    if(data.Contains(i))
    {
        countPerI[i] = data.Where( x => x == i)
            .Count(); //This is not PINQ! It's just a plain noiseless count!

        countPerI[i] += Laplace(1/0.1); //We add Laplace noise here!

        countPerI[i] = countPerI[i] / data.Length();
    }
}

// then some code that plots a bar graph of countPerI versus i

1. True or False? This algorithm is differentially-private.

   Ans:

2. If you answer ‘True’, determine the privacy budget used up by this algorithm.

   If you answer ‘False’, prove that the algorithm is not differentially-private.
Problem 3. The **Intersect**($A_1, A_2, f_1, f_2$) transformation takes two different datasets $A_1, A_2$, and key selection function for each dataset $f_1(), f_2()$. It returns a set of *distinct* records

$$\{ x \mid x \in A_1 \text{ and } \exists y \in A_2 \text{ where } f_1(x) = f_2(y) \}$$

1. **(1 points)** What is the output of **Intersect**($A_1, A_2, f_1, f_2$) if $f_1$ and $f_2$ are the identity function (*i.e.*, $f_1(x) = x$ and $f_2(x) = x$), $A_1 = \{1, 2, 3, 4, 5, 6\}$ and $A_2 = \{4, 6, 7, 8, 8, 8, 9\}$.

   **Ans:**

2. **(3 points)** This transformation is $c$-stable. Determine $c$, and justify your answer.

   **Ans:**

3. **(2 points)** How does the stability of this transformation change if we use the same dataset in both inputs (*i.e.*, **Intersect**($A_1, A_1, f_1, f_1$))?  

   **Ans:**
2 Basic Crypto

Problem 4. For load-balancing purposes, a large private dataset is split between two servers, $A$ and $B$. The servers need to recombine the data so that server $B$ can answer PINQ queries made by users. For each query made by a user, they do the following:

- Server $B$ forwards the query to server $A$
- Server $A$ sends the relevant portions of the dataset over to server $B$
- Server $B$ combines its own dataset with the information sent over by $A$ and produces the answer to the PINQ query
- Server $B$ sends the answer to the user.

Suppose there is an adversary that can both (a) issue PINQ queries to server $B$ and see the answers, and (b) sit on the network between $A$ and $B$ and observe and tamper with the messages that $A$ sends to $B$.

1. **(3 points)** To protect the **confidentiality** of the dataset, should you use
   - CCA secure encryption, or
   - CPA secure encryption, or
   - a secure MAC

   on the messages $A$ sends to $B$? Justify your response.

   Ans:

2. **(3 points)** Suppose there is a user $C$ who issues PINQ queries to $B$. Suppose our adversary has the additional evil goal of wanting user $C$ to get an incorrect answer to his PINQ queries. What tool should we use to prevent this?

   - CCA secure encryption, or
   - CPA secure encryption, or
   - a secure MAC.

   Justify your response.

   Ans:
Bonus privacy question. + 12.5%

Problem 5. (Bonus! +12.5%.)

Consider applying $k$-anonymity to a graph $G$, where we think of the degree\(^1\) of a node is its quasi-identifier. Our anonymization algorithm is as follows:

$k$-Anonymization. We add and delete edges from $G$ to create a modified graph $G^\perp$, ensuring that, for every node $n$ in $G^\perp$, there are at least $k - 1$ other nodes with the same degree as node $n$ in $G^\perp$. We then release $G^\perp$.

Show that this technique fails to anonymize a node’s betweenness in the graph\(^2\). To do this, draw an example of a graph $G^\perp$ that satisfies the notion of $k$-anonymity described above but still has a single node with a unique betweenness. (Hint. To avoid clutter, degree and betweenness are defined in the footnotes below!)

1. (1 points) Show an example for $k = 3$.

2. (2 points) Generalize your example so that it works for any $k$.

---

\(^1\)Recall that a node’s degree is the number of edges incident on it.

\(^2\)Recall that the betweenness of node is a measure of its centrality in the graph. The betweenness of node $n$ is given by the fraction of shortest paths in the graph that pass through node $n$, or more precisely:

1. For each pair of nodes $s, t \in V$, compute the shortest paths between $s$ and $t$. Let $\sigma_{s,t}$ be the fraction of these shortest paths that pass through node $n$.

2. The betweenness of node $n$ is $\sum_{s,t \in V} \sigma_{s,t}$.)