CSCI 5520: Foundations of Data Privacy **Lecture 11**
The Chinese University of Hong Kong, Spring 2015 24 March 2015

In the model of differential privacy we studied so far we viewed the database as private and the queries and answers as public information. In some scenarios assuming a trusted mechanism that will not reveal any unintended information about database may not be reasonable. We will distinguish between two models of privacy violations:

- The mechanism can in general be trusted to preserve the privacy of the database participants, but it may be forced to hand over its data to an untrusted party at some point in time. This may for example be the consequence of an intrusion or a request from a higher authority.
- There is no trusted mechanism; the parties must implement the data release in a distributed privacy-preserving manner.

We will look at a representative example for each of these models.

1 Local mechanisms

General cryptographic techniques for secure multiparty computation allow for the distributed implementation of any efficient mechanism in a manner that preserves *computational* differential privacy. In general, such techniques produce protocols that require interaction among the parties and may be computationally intensive even for simple mechanisms, like the Laplace mechanism for counting queries. It would be interesting to study if protocols with better efficiency, or ones that achieve statistical differential privacy (without sacrificing accuracy), can be obtained even for a single counting query.

Today we will consider the simpler setting of non-interactive mechanisms. We think of the rows of the database as being distributed among n parties, one for each row. A *local mechanism* can be described by n randomized algorithms $M_1, \dots, M_n$ taking inputs in D and an aggregation algorithm A. Given a database $x \in D^n$ as an input, party i publishes a value $M_i(x_i)$ that may depend on its input x_i and its private randomness and the parties jointly compute the value $A(M_1(x_1), \dots, M_n(x_n))$.

The usual differential privacy requirement is that the joint distribution of the outputs of $M_1, \dots, M_n$ do not differ by much on adjacent databases x, x':

$$\Pr[M_1(x_1) = y_1 \text{ and } \cdots \text{ and } M_n(x_n) = y_n] \leq e^{\varepsilon} \Pr[M_1(x'_1) = y_1 \text{ and } \cdots \text{ and } M_n(x'_n) = y_n].$$

for all $y_1, \dots, y_n$ in the support of $M_1(x_1), \dots, M_n(x_n)$ respectively. Since local mechanisms use independent private randomness, this condition is equivalent to

$$\Pr[M_1(x_1) = y_1] \cdots \Pr[M_n(x_n) = y_n] \leq e^{\varepsilon} \Pr[M_1(x'_1) = y_1] \cdots \Pr[M_n(x'_n) = y_n].$$

The probabilities cancel out except in the entry where x and x' differ, so we can derive the following equivalent definition.

Definition 1. A local mechanism is *ε-differentially private* if for every index i, every pair of inputs $x_i, x_i' \in D$, and every possible output y_i,

$$\Pr[M_i(x_i) = y_i] \leq e^{\varepsilon} \Pr[M_i(x_i') = y_i].$$

Here is a local mechanism for counting query q_P called randomized response: Each party flips its answer independently with some probability.

```
Mechanism $RR(x)$:
    Local algorithm $M_i(x_i)$ for row $i$:
        Sample a random variable $N_i \sim \{-1, 1\}_\varepsilon$.
        ($N_i$ takes values $-1$ and $1$ with probabilities $(1-\varepsilon)/2$ and $(1+\varepsilon)/2$, respectively.)
        Output $N_i \cdot (-1)^{P(x_i)}$.
    Aggregation algorithm $A$: Output $n/2 - (1/2\varepsilon)\sum_{i=1}^n M_i(x_i)$.
```

Theorem 2. *Mechanism RR is $(2\varepsilon + O(\varepsilon^2))$-differentially private.*

Proof Sketch. Since the outputs of M_i occur with probabilities $(1-\varepsilon)/2$ and $(1+\varepsilon)/2$, we merely need to verify that $(1+\varepsilon)/2 \leq e^{2\varepsilon + O(\varepsilon^2)}(1-\varepsilon)/2$. □

We now analyze the accuracy. In expectation,

$$\begin{aligned} \mathrm{E}\Big[\sum\nolimits_{i=1}^n M_i(x_i)\Big] = \sum_{i=1}^n \mathrm{E}[M_i(x_i)] &= \sum_{i=1}^n \mathrm{E}[N_i](-1)^{P(x_i)} \\ &= \varepsilon \sum_{i=1}^n (-1)^{P(x_i)} = \varepsilon \sum_{i=1}^n (1 - 2P(x_i)) = \varepsilon(n - 2q_P(x)) \end{aligned}$$

so $\mathrm{E}[RR(x)] = q_P(x)$. By independence,

$$\mathrm{Var}\Big[\sum\nolimits_{i=1}^n M_i(x_i)\Big] = \sum_{i=1}^n \mathrm{Var}[M_i(x_i)] = \sum_{i=1}^n \mathrm{Var}[N_i] = (1-\varepsilon^2)n$$

so the standard deviation of $RR(x)$ is $\sqrt{(1-\varepsilon^2)n}/2\varepsilon$. In contrast, the standard deviation of the Laplace mechanism is $1/2\varepsilon$ for privacy parameter 2ε, so the accuracy of randomized response is worse by a factor of about $\sqrt{n}$.

In Homework 3 you will show that the randomized response mechanism has optimal accuracy up to constants: Any local, ε-differentially private mechanism for counting queries has additive error $\Omega(\sqrt{n}/\varepsilon)$ with constant probability.

2 Continual observation

Suppose we want to maintain a statistic about events happening within a given time interval. For example, individuals join and leave a party at different times. We want to know the head count at any particular time. Releasing the exact count might violate the privacy of the participants.

To model this type of scenario we will represent time by discrete units from 1 to n. A database $x \in D^n$ can then be viewed as a sequence of events over time. For example x_i can be the number of people that joined minus the number of people that left the party at time i. Our objective is then to release the vector of cumulative statistics

$$q(x) = (q_1(x), \dots, q_n(x)) \qquad \text{where } q_t(x) = x_1 + \cdots + x_t.$$

Moreover, the value $q_t(x)$ should be released online at time t before observing the values of the inputs x_{t+1} up to x_n.

For simplicity, let us assume that $D = \{0, 1\}$: we will only count arrivals and there can be at most one in any given time step. The query q has high sensitivity; the value of x_1 affects all the components of $q(x)$, so the product Laplace mechanism cannot give simultaneously good privacy and accuracy.

We now describe a differentially private algorithm for this task. We'll assume n is a power of two. Let T be the full binary tree with n nodes. We label the nodes of T by intervals of the form $[s, t] = \{s, s+1, \dots, t\}$ as follows: The i-th leaf is labeled by the singleton set $\{i\} = [i, i]$ and each internal node is labeled by the union of its leaves. For example, if $n = 4$, then the root of T is labeled by the interval $[1, 4]$, its left and right children are labeled by $[1, 2]$ and $[3, 4]$, respectively, and the leaves are labeled by $\{1\}$, $\{2\}$, $\{3\}$, and $\{4\}$, respectively.

Each interval of the form $[1, t]$ can then be written as the disjoint union of intervals indexed by at most $\log n$ nodes in the tree like this: Follow the leftmost path until you reach the first interval contained in $[1, t]$, take this interval, then recurse on the subtree rooted by its sibling.

Consider the following mechanism for the query vector q:

```
Mechanism $Cum(x)$:
    For each node $I$ of $T$:
        Sample an independent Lap($\log n/\varepsilon$) random variable $N_I$.
        Let $X_I = \sum_{i \in I} x_i + N_I$.
    In time step $t$, answer query $q_t$ as follows:
        Write $[1, t]$ as the disjoint union of at most $\log n$ intervals $I_1, \dots, I_t$.
        Output $\sum_{v=1}^{k} X_{I_v}$.
```

The answer for q_t does not require knowledge of $x_{t+1}, \dots, x_n$, so this mechanism can be implemented in an online manner.

Theorem 3. *Mechanism Cum is ε-differentially private.*

Proof. Since the output of $Cum(x)$ is computed deterministically from the values X_I, it is sufficient to show that this sequence of values $(X_I)_{I \in T}$ is ε-differentially private. This sequence can be viewed as the output of the product Laplace mechanism with parameter $\log n/\varepsilon$ on the function $f(x) = (\sum_{i \in I} x_i)_{I \in T}$. The input x_i appears in exactly $\log n + 1$ outputs of f, once for each ancestor node of $\{i\}$ (including $\{i\}$ itself but not the root). Therefore f is $\log n$-Lipschitz. By Theorem 2 from Lecture 2, the sequence $(X_I)_{I \in T}$ is ε-differentially private. □

To analyze the accuracy of *Cum* we need the following large deviation bound for sums of independent discrete Laplace random variables.

Theorem 4 (Chernoff bound for Laplace random variables)**.** *If* $Y_1, \ldots, Y_m$ *are independent* $\mathrm{Lap}(b)$ *random variables, then*

$$\Pr[Y_1 + \cdots + Y_m > d \cdot b\sqrt{m}] < e^{-d^2/8}$$

for all d *such that* $0 < d < b\sqrt{m}$*.*

By this inequality, for every query q_t,

$$\Pr\Big[\sum\nolimits_{v=1}^{k} X_{I_v} - q_t(x) > d(\log n)^{3/2}/\varepsilon\Big] \leq \Pr\Big[\sum\nolimits_{v=1}^{k} N_{I_v} > d(\log n)^{3/2}/\varepsilon\Big] \leq e^{-d^2/8}$$

and by symmetry and a union bound

$$\Pr\Big[\Big|\sum\nolimits_{v=1}^{k} X_{I_v} - q_t(x)\Big| > d(\log n)^{3/2}/\varepsilon\Big] < 2e^{-d^2/8}.$$

Taking a union bound over all t from 1 to n we obtain that the mechanism has additive error $d(\log n)^{3/2}/\varepsilon$ on all queries with probability at least $1 - 2ne^{-d^2/8}$. For $d = O(\sqrt{\log n})$, we obtain the following bound on the accuracy of *Cum*.

Theorem 5. *With probability at least* $1 - 1/n$*, for all queries* t*, the answer of* $Cum(x)$ *to* $q_t(x)$ *is within error at most* $O((\log n)^2/\varepsilon)$ *of* $x_1 + \cdots + x_t$*.*

A lower bound In fact any 1-differentially private mechanism must have additive error at least $a = \Omega(\log n)$ on at least one of the queries with probability at least 1/2 for the following reason. Suppose M is a 1-differentially private mechanism that achieves additive error less than a on all queries q_t. Partition $x \in \{0,1\}^n$ into $n/(2a)$ blocks of length $2a$ each. Let x^i be the database whose rows in the i-th block all equal 1 and all the other rows are zero. Then

$$q_j(x^i) = \begin{cases} 0, & \text{for } j = 0, 2a, 4a, \ldots, 2(i-1)a \\ 2a, & \text{for } j = 2ia, 2(i+1)a, \ldots, n. \end{cases}$$

Let T_i be the event $a_k < a$ for $j = 0, 2a, 4a, \ldots, 2(i-1)a$ and $a_k > a$ for $j = 2ia, 2(i+1)a, \ldots, n$, where a_j is mechanism's answer to the j-th query. By accuracy of the mechanism, $\Pr[M(x^i) \in T_i] \geq 1/2$. Since x^i and x^1 differ in $4a$ entries, by differential privacy,

$$\Pr[M(x^1) \in T_i] \geq e^{-4a} \Pr[M(x^i) \in T_i] \geq \frac{e^{-4a}}{2}.$$

The events $T_1, \ldots, T_{n/2a}$ are disjoint so

$$\sum_{i=1}^{n/2a} \Pr[M(x^1) \in T_i] \leq 1$$

which is impossible if $a < \ln n/4 - 1$. It should be possible to extend this bound to $(1, 0.1/n)$-differentially private mechanisms.

3 Pan-private implementations

An implementation of continual data release mechanism is pan-private if the mechanism remains private even when its state is released to an adversary at some point in time. To define this concept properly, we model the mechanism implementation as an online algorithm which at time i, receives as its input its i-th row $x_i \in D$, updates its state, produces its i-th output, and updates its state again.

Definition 6. An mechanism implementation $M\colon D^n \to R^n$ is *ε-differentially pan-private for the past* if for every time $t \in [n]$, the joint distribution of the first t outputs of M and the state of the memory of M after time t is ε-differentially private.

A more general definition that also takes into account the state of the memory after time t is also possible, but we'll stick to this one for relative simplicity. This definition may be sensible in a scenario where after an intrusion is discovered, the mechanism halts and does not take in any new inputs.

Pan-private cumulative sums We now describe a pan-private mechanism implementation for cumulative sums. This mechanism also maintains noise for each node in a complete binary tree with n leaves, but the noise is used in a somewhat different way.

```
Mechanism $PPCum$:
    Set $X = N$, where $N$ is a Lap($1/\eta$) random variable where $1/\eta = (\log n + 1)/\varepsilon$.
    For each node $I$ of $T$ except the root,
        Sample an independent Lap($1/\eta$) random variable $N_I$.
    On input $x_t$ in time step $t$:
        Add $x_t$ to $X$.
        Output $X + \sum_{I\colon t \in I} N_I$.
        Erase $x_t$ and all values $N_I$ where $I \subseteq [1, t]$.
```

The accuracy analysis is similar as for the previous algorithm. Before we prove privacy, let's see an example. When $n = 4$, the following values are released by the mechanism

$$\begin{array}{ll} N + x_1 + N_{\{1\}} + N_{[1,2]} & \text{at time step 1} \\ N + x_1 + x_2 + N_{\{2\}} + N_{[1,2]} & \text{at time step 2} \\ N + x_1 + x_2 + x_3 + N_{\{3\}} + N_{[3,4]} & \text{at time step 3.} \end{array}$$

The memory of the mechanism after time step 3 contains the values

$$N + x_1 + x_2 + x_3, N_{[3,4]}, N_{\{4\}}.$$

Let x' be the data sequence obtained from x by modifying the value of x_3. Moreover, suppose that $x_3 = 0$ and $x'_3 = 1$. Then the sequence of values that are released before time step 3 and in memory after time step 3 is identical to

$$\begin{array}{ll} N' + x_1 + N_{\{1\}} + N'_{[1,2]} & \text{at time step 1} \\ N' + x_1 + x_2 + N_{\{2\}} + N'_{[1,2]} & \text{at time step 2} \\ N' + x_1 + x_2 + x'_3 + N_{\{3\}} + N_{[3,4]} & \text{at time step 3} \\ N' + x_1 + x_2 + x'_3, N_{[3,4]}, N_{\{4\}} & \text{in memory after time 3} \end{array}$$

where $N' = N - 1$ and $N'_{[1,2]} = N_{[1,2]} + 1$. We can conclude that for any such pair x, x', the probability of any view after 3 steps on input x is at most $e^{2/\eta}$ times smaller than the probability of the same view on input x'.

We can now sketch the general proof of pan-privacy.

Theorem 7. *Mechanism PPCum is ε-differentially pan-private from the past.*

Proof Sketch. Let x, x' be two adjacent data sequences that differ in entry $j \le t$. Without loss of generality, we can assume that $x_j = 0$ and $x'_j = 1$. The values of the queries released by $PPCum(x)$ up to time t are

$$N + x_1 + \cdots + x_i + \sum_{I\,:\,i \in I} N_i \quad \text{at time } i$$

for $1 \le i \le t$. The memory contents of $PPCum(x)$ after time t consist of the value $N + x_1 + \cdots + x_t$ and those N_I such that I does not intersect the interval $[1, t]$.

We represent the $[1, j-1]$ as a disjoint union of at most $\log n$ intervals $I_1, \ldots, I_k$ as in Section 2. Set

$$N' = N - 1, N'_{I_1} = N_{I_1} + 1, \ldots, N'_{I_k} = N_{I_k} + 1.$$

and $N'_I = N_I$ for all the other intervals I.

The joint view of the first t answers and the memory after time t in $PPCum(x)$ with randomness N, N_I is then identical to the corresponding joint view of $PPCum(x')$ with randomness N', N'_I. The ratio of the probabilities of these two views is at most

$$e^{\eta} \cdot e^{k\eta} = e^{(k+1)\eta} \le e^{\varepsilon}$$

by our choice of parameters, so $PPCum$ is ε-differentially pan-private from the past. □

References

These notes are based on Chapter 12 of the survey *The Algorithmic Foundations of Differential Privacy* by Cynthia Dwork and Aaron Roth.