In our discussion of differential privacy so far we did not address the computational complexity of private data release. Let's review the two mechanisms for counting (averaging) queries that we studied and try to estimate their running time. We'll fix the accuracy parameter α to some constant, say $\alpha = 0.1$.

Recall the mechanism of Blum, Ligett, and Roth. Given a database $x \in D^n$ as input, this mechanism samples and outputs a *synthetic database* $y \in D^d$ with probability proportional to $\exp((\varepsilon n/4)u(x,y))$, where ε is the differential privacy parameter and $u(x,y)$ is the maximum deviation $|\overline{q}(x) - \overline{q}(y)|$ among all averaging queries $\overline{q}$ of interest.

To sample from the distribution of interest, it appears that we need to calculate all the utilities $u(x,y)$ for all $y \in D^d$. Calculating each such utility may in general take time proportional to the number of queries $|Q|$, so it appears that the running time of this mechanism may be as large as $|Q| \cdot |D|^d = 2^{\Theta(\log|Q|\log|D|)}$. This number can be quite large even for moderately sized databases and sets of queries.

The mechanism of Hardt and Rothblum represents the database x as a probability distribution over the domain D. The main step in this mechanism is a multiplicative update on the entries of x: Given a predicate P (coming from a counting query q_P) and an estimate check that is too high or too low, scale up those entries of x that satisfy or do not satisfy P (depending on the outcome of the estimate check) and renormalize to a probability distribution. As there are $|D|$ entries in the vector x, a naive implementation of this step would take time $\Theta(|D|)$. Is there a faster implementation?

In general, the answer is a conditional but believable "no": If cryptographic one-way functions exist, then any mechanism that is guaranteed to answer, say, n^3 counting queries cannot have complexity that is polynomial in both n and $\log|D|$. We won't show this result in its entirety in this lecture (I don't know how to prove it yet), but we will introduce a proof technique that shows hardness of data release for a much less interesting setting of parameters. Hopefully in the next lecture we will complete the proof of this negative result.

The kinds of databases that exemplify the computational hardness of private data release are not particularly natural. If we impose some restrictions on the structure of the database and the types of counting queries allowed, these limitations may sometimes be surmounted.

1 Hardness of generating synthetic databases

A *synthetic database mechanism* that on input a database $x \in D^n$ produces as its output a database $y \in D^d$ for some d. The mechanism is α-*accurate* for a set of averaging queries $\overline{Q}$ if for every query $\overline{q} \in \overline{Q}$, $|\overline{q}(x) - \overline{q}(M(x))| \leq \alpha$ with probability 1 over the randomness of M.

We will show that if a synthetic database mechanism is efficient differentially private, then it cannot be α-accurate for any $\alpha < 1$ under a standard cryptographic assumption: The existence of message

authentication codes.

The role of the message authentication code will be to "force" the rows of the synthetic database to come from the original database, thereby violating privacy.

Message authentication codes Informally, a message authentication code (MAC) is a scheme for sending messages (from Alice to Bob) that ensures the *integrity* of messages: After receiving the MAC of a message, Bob is convinced that the message is indeed the one that Alice intended to send him and not some other message. There is no privacy requirement here: the objective of message authentication is not to preserve secrecy but to prevent tampering. Message authentication codes are usually implemented by appending some verification information to the message – a *tag* – that certifies the authenticity and integrity of the message.

Message authentication codes are defined in the private-key model: Alice and Bob agree on a secret key K, chosen uniformly at random from the space of all possible keys $\{0,1\}^k$. Apart from Alice and Bob, nobody else has any prior information about this secret key.

We begin by defining the functionality requirement for message authentication codes:

Definition 1. A *message authentication code* (MAC) with key length k, message length m, and tag length t is an algorithm (Tag, Ver), where $Tag\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^t$ and $Ver\colon \{0,1\}^k \times \{0,1\}^m \times \{0,1\}^t \to \{0,1\}$ so that for every key $K \in \{0,1\}^k$ and message $M \in \{0,1\}^m$,

$$Ver(K, M, Tag(K, M)) = 1.$$

We will assume that the tagging and verification algorithms are deterministic.

We now define security. The security requirement should postulate that upon seeing the tag of a message M, the adversary should not be able to produce a *forgery* $M' \neq M$ together with a tag for M'. The notion of security we will define is quite strong: The adversary can query a *tagging oracle* that can tag messages of its choice, and its task will be to come up with a forged tag of any message that it has not previously queried.

Definition 2. A MAC (Tag, Ver) is *ε-secure* if for every efficient algorithm $A^?$ with access to a "tagging oracle",

$$\Pr[A^{Tag(K,\cdot)} \text{ produces a forgery}] \leq \varepsilon$$

where a forgery is a pair (M, T) such that (1) M is different from all the queries A made to the tagging oracle and (2) $Ver(K, M, T) = 1$. Here, the probability is taken over the choice of K and the randomness of Tag and Ver.

Under reasonable cryptographic assumptions (the existence of one-way functions that are exponentially hard to invert), there exist efficiently computable message authentication codes with message length and tag length $O(k)$ that are $2^{-\Omega(k)}$-secure against chosen message attack.

A hard example Given a MAC (Tag, Ver), we now construct a distribution over databases x and give a set of counting queries Q such that if $M(x)$ is accurate for $\overline{Q}$ then it is not differentially private.

Theorem 3. *There exists a domain of size 2^{m+t} and a set of counting queries Q of size 2^k such that if $M\colon D^n \to D^d$ is an α-accurate synthetic database mechanism for Q for some $\alpha < 1$ and $10dn \leq 2^m$ then M is not $(1, 1/10n)$ differentially private or (Tag, Ver) is not $1/2d$-secure.*

Blum, Ligett, and Roth show the existence of a differentially private synthetic database for $n \geq K(\log|D|\log|Q|)$, where K is a sufficiently large constant. Theorem 3 shows that this mechanism cannot in general be made efficient.

Proof Sketch of Theorem 3. The domain D is the set $\{0,1\}^m \times \{0,1\}^t$ of all possible message, tag pairs. To construct x we choose n random messages $M_1, \ldots, M_n \sim \{0,1\}^m$, a random key $K \sim \{0,1\}^k$, and set $x_i = (M_i, Tag(K, M_i))$ as the i-th row of x.

For every possible key $K \in \{0,1\}^n$, let q_K be the query that counts the number of rows (M,T) for which T is a correct tag for message M under key K, namely such that $Ver(K,M,T) = 1$. Let Q be the set of all such 2^k queries q_K.

Now suppose M is an efficient, α-accurate synthetic database mechanism for $\overline{Q}$. By construction, all rows of x are valid (message, tag) pairs under K, so $\overline{q}_K(x) = 1$. If $\alpha < 1$, then $\overline{q}_K(M(x)) > 0$, so at least one of the rows of $M(x)$ must be a valid (message, tag) pair $(M', Tag_K(M'))$.

We now distinguish two cases. If $(M', Tag_K(M'))$ is a row of x with probability at least $1/2$, then M' is not $(1, 0.1)$ differentially private as long as $2^m \geq 10dn$ by the following theorem which you will prove in your homework:

Theorem 4. *Suppose $M\colon D^n \to D^d$ is an (ε, δ)-differentially private synthetic database mechanism. Let $\mathcal{D}$ be a probability distribution over some finite set in which no element has probability greater than 2^{-m} and $x \sim \mathcal{D}^n$ be a random database whose rows are independent samples from $\mathcal{D}$. The probability that $M(x)$ contains a row of x is at most $e^{\varepsilon}dn/2^m + n\delta$.*

If $(M', Tag_K(M'))$ is not a row of x with probability at least $1/2$, then the following efficient algorithm produces forged tags of (Tag, Ver) with probability at least $1/2d$:

1. Query the tagging oracle on messages $M_1, \ldots, M_n$ to obtain respective tags $T_1, \ldots, T_n$.
2. Construct the database x by setting its ith row to $x_i = (M_i, T_i)$.
3. Output a random row of $M(x)$.

With probability at least $1/2$, M' is different from $M_1, \ldots, M_n$, so the probability that a random row of $M(x)$ is the forgery $(M', Tag_K(M'))$ is at least $1/2d$. □

2 Sanitized data

A sanitization mechanism is a generalization of a synthetic database mechanism. The purpose of the sanitization mechanism is to publish data that, on the one hand, provides accurate answers to queries, and on the other, protects privacy of the database participants. Unlike a synthetic

database, sanitized data does not have to "look like" a database at all; we merely want to be able to extract the answers to queries from it.

A *sanitized data release mechanism* for a collection of queries Q is a mechanism M that takes as input a database $x \in D^n$ and outputs a data item y together with a conversion algorithm that turns every query $q \in Q$ about x into a query q' about y. Such a mechanism is efficient if both M and the query conversion algorithm are efficient.

We say M is α-*accurate* for a set of counting queries Q if for every $q \in Q$,

$$|q'(M(x)) - q(x)| \leq \alpha n.$$

We now give an example of a collection Q of counting queries for which no sanitized database can provide answers that are private and accurate at the same time. This example is not particularly surprising as the number of counting queries in Q is larger than the number $|D|^n$ of possible databases, so it is quite plausible that the answers to all queries in Q completely specify the database and destroy differential privacy. Nevertheless, the reasoning will presumably turn out to be helpful in ruling out more realistic scenarios, where the size of Q is much smaller.

A hard example The domain for the rows of our "hard" database is the set $[n] \times \{0,1\}$. To each strings $c = (c_1, \ldots, c_n) \in \{0,1\}^n$ we associate a counting query $q_c(x)$ that counts the number of rows of the form (i, c_i) for some $i \in [n]$.

Now consider a database x whose i-th row has the form (i, x_i), where $x_1, \ldots, x_n$ are some strings in $\{0,1\}^k$. Then $q_{x_1,\ldots,x_n}(x) = n$, while $q_{\overline{x}_1,\ldots,\overline{x}_n}(x) = 0$ (where $\overline{b}$ is the negation of bit b.)

If the sanitized data release mechanism M is α-accurate for any $\alpha < 1/2$, then it must be true that

$$q'_{x_1,\ldots,x_n}(x) > n/2 \quad \text{and} \quad q'_{\overline{x}_1,\ldots,\overline{x}_n}(x) < n/2.$$

for every x of the given form. Now consider what happens when $x_1, \ldots, x_n$ are independent uniformly random bits. We can rewrite the above two constraints as

$$\Pr[q'_{x_1,\ldots,x_n}(M(x)) > n/2] = 1 \quad \text{and} \quad \Pr[q'_{\overline{x}_1,\ldots,\overline{x}_n}(M(x)) > n/2] = 0.$$

By a hybrid argument, there must then exist an index i such that

$$\Pr[q'_{x_1,\ldots,x_{i-1},x_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(M(x)) > n/2] - \Pr[q'_{x_1,\ldots,x_{i-1},\overline{x}_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(M(x)) > n/2] \geq 1/n.$$

If T_x is the set of all outputs y of M such that

$$\Pr[q'_{x_1,\ldots,x_{i-1},x_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(y) > n/2] - \Pr[q'_{x_1,\ldots,x_{i-1},\overline{x}_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(y) > n/2] \geq 1/n.$$

then $M(x)$ is in T_x with probability 1 over the randomness of the mechanism M.

Now let x' be the database obtained by replacing the i-th row of x by (i, x'_i), where x'_i is a uniformly random bit independent of x. Then the random variables

$$q'_{x_1,\ldots,x_{i-1},x_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(M(x')) \quad \text{and} \quad q'_{x_1,\ldots,x_{i-1},\overline{x}_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(M(x'))$$

are identically distributed, so

$$\Pr[q'_{x_1,\ldots,x_{i-1},x_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(M(x')) > n/2] - \Pr[q'_{x_1,\ldots,x_{i-1},\overline{x}_i,\overline{x}_{i+1},\ldots,\overline{x}_n}(M(x')) > n/2] = 0$$

and so the probability that $M(x')$ is in T_x is zero. Therefore the differential privacy condition

$$\Pr[M(x) \in T_x] \leq e^{\varepsilon} \Pr[M(x') \in T_x] + \delta$$

cannot hold for any nontrivial setting of ε and δ.

References

These notes are based on Chapter 9 of the survey *The Algorithmic Foundations of Differential Privacy* by Cynthia Dwork and Aaron Roth.

The background on message authentication codes is from my lecture notes on cryptography. For more, see the book *Introduction to Modern Cryptography* by Jonathan Katz and Yehuda Lindell.