

2

Basic Terms

This section motivates and presents the formal definition of differential privacy, and enumerates some of its key properties.

2.1 The model of computation

We assume the existence of a trusted and trustworthy *curator* who holds the data of *individuals* in a database D , typically comprised of some number n of rows. The intuition is that each row contains the data of a single individual, and, still speaking intuitively, the privacy goal is to simultaneously protect every individual row while permitting statistical analysis of the database as a whole.

In the *non-interactive*, or *offline*, model the curator produces some kind of object, such as a “synthetic database,” collection of summary statistics, or “sanitized database” once and for all. After this *release* the curator plays no further role and the original data may be destroyed.

A *query* is a function to be applied to a database. The *interactive*, or *online*, model permits the data analyst to ask queries adaptively, deciding which query to pose next based on the observed responses to previous queries.

The trusted curator can be replaced by a protocol run by the set of individuals, using the cryptographic techniques for secure multi-party protocols, but for the most part we will not be appealing to cryptographic assumptions. Section 12 describes this and other models studied in the literature.

When all the queries are known in advance the non-interactive model should give the best accuracy, as it is able to correlate noise knowing the structure of the queries. In contrast, when no information about the queries is known in advance, the non-interactive model poses severe challenges, as it must provide answers to all possible queries. As we will see, to ensure privacy, or even to prevent privacy catastrophes, accuracy will necessarily deteriorate with the number of questions asked, and providing accurate answers to all possible questions will be infeasible.

A *privacy mechanism*, or simply a *mechanism*, is an algorithm that takes as input a database, a universe \mathcal{X} of data types (the set of all possible database rows), random bits, and, optionally, a set of queries, and produces an output string. The hope is that the output string can be decoded to produce relatively accurate answers to the queries, if the latter are present. If no queries are presented then we are in the non-interactive case, and the hope is that the output string can be interpreted to provide answers to future queries.

In some cases we may require that the output string be a *synthetic database*. This is a multiset drawn from the universe \mathcal{X} of possible database rows. The decoding method in this case is to carry out the query on the synthetic database and then to apply some sort of simple transformation, such as multiplying by a scaling factor, to obtain an approximation to the the true answer to the query.

2.2 Towards defining private data analysis

A natural approach to defining privacy in the context of data analysis is to require that the analyst knows no more about any individual in the data set after the analysis is completed than she knew before the analysis was begun. It is also natural to formalize this goal by

requiring that the adversary's prior and posterior views about an individual (i.e., before and after having access to the database) shouldn't be "too different," or that access to the database shouldn't change the adversary's views about any individual "too much." However, if the database teaches anything at all, this notion of privacy is unachievable. For example, suppose the adversary's (incorrect) prior view is that everyone has 2 left feet. Access to the statistical database teaches that almost everyone has one left foot and one right foot. The adversary now has a very different view of whether or not any given respondent has two left feet.

Part of the appeal of before/after, or "nothing is learned," approach to defining privacy is the intuition that if nothing is learned about an individual then the individual cannot be harmed by the analysis. However, the "smoking causes cancer" example shows this intuition to be flawed; the culprit is auxiliary information (Mr. X smokes).

The "nothing is learned" approach to defining privacy is reminiscent of semantic security for a cryptosystem. Roughly speaking, semantic security says that nothing is learned about the plaintext (the unencrypted message) from the ciphertext. That is, anything known about the plaintext after seeing the ciphertext was known before seeing the ciphertext. So if there is auxiliary information saying that the ciphertext is an encryption of either "dog" or "cat," then the ciphertext leaks no further information about which of "dog" or "cat" has been encrypted. Formally, this is modeled by comparing the ability of the eavesdropper to guess which of "dog" and "cat" has been encrypted to the ability of a so-called *adversary simulator*, who has the auxiliary information but does not have access to the ciphertext, to guess the same thing. If for every eavesdropping adversary, and all auxiliary information (to which both the adversary and the simulator are privy), the adversary simulator has essentially the same odds of guessing as does the eavesdropper, then the system enjoys semantic security. Of course, for the system to be useful, the legitimate receiver must be able to correctly decrypt the message; otherwise semantic security can be achieved trivially.

We know that, under standard computational assumptions, semantically secure cryptosystems exist, so why can we not build semantically

secure private database mechanisms that yield answers to queries while keeping individual rows secret?

First, the analogy is not perfect: in a semantically secure cryptosystem there are three parties: the message sender (who encrypts the plaintext message), the message receiver (who decrypts the ciphertext), and the eavesdropper (who is frustrated by her inability to learn anything about the plaintext that she did not already know before it was sent). In contrast, in the setting of private data analysis there are only two parties: the curator, who runs the privacy mechanism (analogous to the sender) and the data analyst, who receives the informative responses to queries (like the message receiver) and also tries to squeeze out privacy-compromising information about individuals (like the eavesdropper). Because the legitimate receiver is the same party as the snooping adversary, the analogy to encryption is flawed: denying all information to the adversary means denying all information to the data analyst.

Second, as with an encryption scheme, we require the privacy mechanism to be useful, which means that it teaches the analyst something she did not previously know. This teaching is unavailable to an adversary simulator; that is, no simulator can “predict” what the analyst has learned. We can therefore look at the database as a weak source of random (unpredictable) bits, from which we can extract some very high quality randomness to be used as a *random pad*. This can be used in an encryption technique in which a secret message is added to a random value (the “random pad”) in order to produce a string that information-theoretically hides the secret. Only someone knowing the random pad can learn the secret; any party that knows nothing about the pad learns nothing at all about the secret, no matter his or her computational power. Given access to the database, the analyst can learn the random pad, but the adversary simulator, not given access to the database, learns nothing at all about the pad. Thus, given as auxiliary information the encryption of a secret using the random pad, the analyst can decrypt the secret, but the adversary simulator learns nothing at all about the secret. This yields a huge disparity between the ability of the adversary/analyst to learn the secret and the ability

of the adversary simulator to do the same thing, eliminating all hope of anything remotely resembling semantic security.

The obstacle in both the smoking causes cancer example and the hope for semantic security is auxiliary information. Clearly, to be meaningful, a privacy guarantee must hold even in the context of “reasonable” auxiliary knowledge, but separating reasonable from arbitrary auxiliary knowledge is problematic. For example, the analyst using a government database might be an employee at a major search engine company. What are “reasonable” assumptions about the auxiliary knowledge information available to such a person?

2.3 Formalizing differential privacy

We will begin with the technical definition of differential privacy, and then go on to interpret it. Differential privacy will provide privacy by *process*; in particular it will introduce randomness. An early example of privacy by randomized process is *randomized response*, a technique developed in the social sciences to collect statistical information about embarrassing or illegal behavior, captured by having a property P . Study participants are told to report whether or not they have property P as follows:

1. Flip a coin.
2. If **tails**, then respond truthfully.
3. If **heads**, then flip a second coin and respond “Yes” if heads and “No” if tails.

“Privacy” comes from the plausible deniability of any outcome; in particular, if having property P corresponds to engaging in illegal behavior, even a “Yes” answer is not incriminating, since this answer occurs with probability at least $1/4$ whether or not the respondent actually has property P . Accuracy comes from an understanding of the noise generation procedure (the introduction of spurious “Yes” and “No” answers from the randomization): The expected number of “Yes” answers is $1/4$ times the number of participants who do not have property P plus $3/4$ the number having property P . Thus, if p is the true fraction of

participants having property P , the expected number of “Yes” answers is $(1/4)(1-p) + (3/4)p = (1/4) + p/2$. Thus, we can estimate p as twice the fraction answering “Yes” minus $1/2$, that is, $2((1/4) + p/2) - 1/2$.

Randomization is essential; more precisely, any *non-trivial* privacy guarantee that holds regardless of all present or even future sources of auxiliary information, including other databases, studies, Web sites, on-line communities, gossip, newspapers, government statistics, and so on, requires randomization. This follows from a simple hybrid argument, which we now sketch. Suppose, for the sake of contradiction, that we have a non-trivial deterministic algorithm. Non-triviality says that there exists a query and two databases that yield different outputs under this query. Changing one row at a time we see there exists a pair of databases differing only in the value of a single row, on which the same query yields different outputs. An adversary knowing that the database is one of these two almost identical databases learns the value of the data in the unknown row.

We will therefore need to discuss the input and output space of randomized algorithms. Throughout this monograph we work with discrete probability spaces. Sometimes we will describe our algorithms as sampling from continuous distributions, but these should always be discretized to finite precision in an appropriately careful way (see Remark 2.1 below). In general, a randomized algorithm with domain A and (discrete) range B will be associated with a mapping from A to the probability simplex over B , denoted $\Delta(B)$:

Definition 2.1 (Probability Simplex). Given a discrete set B , the *probability simplex* over B , denoted $\Delta(B)$ is defined to be:

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|B|} x_i = 1 \right\}$$

Definition 2.2 (Randomized Algorithm). A randomized algorithm \mathcal{M} with domain A and discrete range B is associated with a mapping $M : A \rightarrow \Delta(B)$. On input $a \in A$, the algorithm \mathcal{M} outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$ for each $b \in B$. The probability space is over the coin flips of the algorithm \mathcal{M} .

We will think of databases x as being collections of records from a universe \mathcal{X} . It will often be convenient to represent databases by their histograms: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of *type* $i \in \mathcal{X}$ (we abuse notation slightly, letting the symbol \mathbb{N} denote the set of all non-negative integers, including zero). In this representation, a natural measure of the distance between two databases x and y will be their ℓ_1 distance:

Definition 2.3 (Distance Between Databases). The ℓ_1 norm of a database x is denoted $\|x\|_1$ and is defined to be:

$$\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|.$$

The ℓ_1 distance between two databases x and y is $\|x - y\|_1$

Note that $\|x\|_1$ is a measure of the *size* of a database x (i.e., the number of records it contains), and $\|x - y\|_1$ is a measure of how many records *differ* between x and y .

Databases may also be represented by multisets of *rows* (elements of \mathcal{X}) or even ordered lists of rows, which is a special case of a set, where the row number becomes part of the name of the element. In this case distance between databases is typically measured by the Hamming distance, i.e., the number of rows on which they differ.

However, unless otherwise noted, we will use the histogram representation described above. (Note, however, that even when the histogram notation is more mathematically convenient, in actual implementations, the multiset representation will often be much more concise).

We are now ready to formally define *differential privacy*, which intuitively will guarantee that a randomized algorithm behaves similarly on similar input databases.

Definition 2.4 (Differential Privacy). A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta,$$

where the probability space is over the coin flips of the mechanism \mathcal{M} . If $\delta = 0$, we say that \mathcal{M} is ε -differentially private.

Typically we are interested in values of δ that are less than the inverse of any polynomial in the size of the database. In particular, values of δ on the order of $1/\|x\|_1$ are very dangerous: they permit “preserving privacy” by publishing the complete records of a small number of database participants — precisely the “just a few” philosophy discussed in Section 1.

Even when δ is negligible, however, there are theoretical distinctions between $(\varepsilon, 0)$ - and (ε, δ) -differential privacy. Chief among these is what amounts to a switch of quantification order. $(\varepsilon, 0)$ -differential privacy ensures that, for *every* run of the mechanism $\mathcal{M}(x)$, the output observed is (almost) equally likely to be observed on *every* neighboring database, simultaneously. In contrast (ε, δ) -differential privacy says that for every pair of neighboring databases x, y , it is extremely unlikely that, *ex post facto* the observed value $\mathcal{M}(x)$ will be much more or much less likely to be generated when the database is x than when the database is y . However, given an output $\xi \sim \mathcal{M}(x)$ it may be possible to find a database y such that ξ is much more likely to be produced on y than it is when the database is x . That is, the mass of ξ in the distribution $\mathcal{M}(y)$ may be substantially larger than its mass in the distribution $\mathcal{M}(x)$.

The quantity

$$\mathcal{L}_{\mathcal{M}(x)\|\mathcal{M}(y)}^{(\xi)} = \ln \left(\frac{\Pr[\mathcal{M}(x) = \xi]}{\Pr[\mathcal{M}(y) = \xi]} \right)$$

is important to us; we refer to it as the *privacy loss* incurred by observing ξ . This loss might be positive (when an event is more likely under x than under y) or it might be negative (when an event is more likely under y than under x). As we will see in Lemma 3.17, (ε, δ) -differential privacy ensures that for all adjacent x, y , the absolute value of the privacy loss will be bounded by ε with probability at least $1 - \delta$. As always, the probability space is over the coins of the mechanism \mathcal{M} .

Differential privacy is immune to *post-processing*: A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a private algorithm \mathcal{M} and make it

less differentially private. That is, if an algorithm protects an individual's privacy, then a data analyst cannot increase privacy loss — either under the formal definition or even in any intuitive sense — simply by sitting in a corner and *thinking about* the output of the algorithm. Formally, the composition of a data-independent mapping f with an (ε, δ) -differentially private algorithm \mathcal{M} is also (ε, δ) differentially private:

Proposition 2.1 (Post-Processing). Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ be a randomized algorithm that is (ε, δ) -differentially private. Let $f : R \rightarrow R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R'$ is (ε, δ) -differentially private.

Proof. We prove the proposition for a deterministic function $f : R \rightarrow R'$. The result then follows because any randomized mapping can be decomposed into a convex combination of deterministic functions, and a convex combination of differentially private mechanisms is differentially private.

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$. Let $T = \{r \in R : f(r) \in S\}$. We then have:

$$\begin{aligned} \Pr[f(\mathcal{M}(x)) \in S] &= \Pr[\mathcal{M}(x) \in T] \\ &\leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in T] + \delta \\ &= \exp(\epsilon) \Pr[f(\mathcal{M}(y)) \in S] + \delta \end{aligned}$$

which was what we wanted. \square

It follows immediately from Definition 2.4 that $(\varepsilon, 0)$ -differential privacy composes in a straightforward way: the composition of two $(\varepsilon, 0)$ -differentially private mechanisms is $(2\varepsilon, 0)$ -differentially private. More generally (Theorem 3.16), “the epsilons and the deltas add up”: the composition of k differentially private mechanisms, where the i th mechanism is $(\varepsilon_i, \delta_i)$ -differentially private, for $1 \leq i \leq k$, is $(\sum_i \varepsilon_i, \sum_i \delta_i)$ -differentially private.

Group privacy for $(\varepsilon, 0)$ -differentially private mechanisms also follows immediately from Definition 2.4, with the strength of the privacy guarantee drops linearly with the size of the group.

Theorem 2.2. Any $(\varepsilon, 0)$ -differentially private mechanism \mathcal{M} is $(k\varepsilon, 0)$ -differentially private for groups of size k . That is, for all $\|x - y\|_1 \leq k$ and all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(k\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}],$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

This addresses, for example, the question of privacy in surveys that include multiple family members.¹

More generally, composition and group privacy are not the same thing and the improved composition bounds in Section 3.5.2 (Theorem 3.20), which substantially improve upon the factor of k , do not — and cannot — yield the same gains for group privacy, even when $\delta = 0$.

2.3.1 What differential privacy promises

An Economic View. Differential privacy promises to protect individuals from any *additional* harm that they might face due to their data being in the private database x that they would not have faced had their data not been part of x . Although individuals may indeed face harm once the results $\mathcal{M}(x)$ of a differentially private mechanism \mathcal{M} have been released, differential privacy promises that the probability of harm was not significantly increased by their choice to participate. This is a very utilitarian definition of privacy, because when an individual is deciding whether or not to include her data in a database that will be used in a differentially private manner, it is exactly this difference that she is considering: the probability of harm given that she participates, as compared to the probability of harm given that she does not participate. She has no control over the remaining contents of the database. Given the promise of differential privacy, she is assured that she should

¹However, as the group gets larger, the privacy guarantee deteriorates, and this is what we want: clearly, if we replace an entire surveyed population, say, of cancer patients, with a completely different group of respondents, say, healthy teenagers, we *should* get different answers to queries about the fraction of respondents who regularly run three miles each day. Although something similar holds for (ε, δ) -differential privacy, the approximation term δ takes a big hit, and we only obtain $(k\varepsilon, ke^{(k-1)\varepsilon}\delta)$ -differential privacy for groups of size k .

be almost indifferent between participating and not, from the point of view of future harm. Given any incentive — from altruism to monetary reward — differential privacy may convince her to allow her data to be used. This intuition can be formalized in a utility-theoretic sense, which we here briefly sketch.

Consider an individual i who has arbitrary preferences over the set of all possible future events, which we denote by \mathcal{A} . These preferences are expressed by a utility function $u_i : \mathcal{A} \rightarrow \mathbb{R}_{\geq 0}$, and we say that individual i experiences utility $u_i(a)$ in the event that $a \in \mathcal{A}$ comes to pass. Suppose that $x \in \mathbb{N}^{|\mathcal{X}|}$ is a data-set containing individual i 's private data, and that \mathcal{M} is an ε -differentially private algorithm. Let y be a data-set that is identical to x except that it does not include the data of individual i (in particular, $\|x - y\|_1 = 1$), and let $f : \text{Range}(\mathcal{M}) \rightarrow \Delta(\mathcal{A})$ be the (arbitrary) function that determines the distribution over future events \mathcal{A} , conditioned on the output of mechanism \mathcal{M} . By the guarantee of differential privacy, together with the resilience to arbitrary post-processing guaranteed by Proposition 2.1, we have:

$$\begin{aligned} \mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] &= \sum_{a \in \mathcal{A}} u_i(a) \cdot \Pr_{f(\mathcal{M}(x))}[a] \\ &\leq \sum_{a \in \mathcal{A}} u_i(a) \cdot \exp(\varepsilon) \Pr_{f(\mathcal{M}(y))}[a] \\ &= \exp(\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)] \end{aligned}$$

Similarly,

$$\mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] \geq \exp(-\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)].$$

Hence, by promising a guarantee of ε -differential privacy, a data analyst can promise an individual that his expected future utility will not be harmed by more than an $\exp(\varepsilon) \approx (1 + \varepsilon)$ factor. Note that this promise holds *independently* of the individual i 's utility function u_i , and holds *simultaneously* for multiple individuals who may have completely different utility functions.

2.3.2 What differential privacy does not promise

As we saw in the Smoking Causes Cancer example, while differential privacy is an extremely strong guarantee, it does not promise unconditional freedom from harm. Nor does it create privacy where none previously exists. More generally, differential privacy does not guarantee that what one believes to be one's secrets will remain secret. It merely ensures that one's participation in a survey will not in itself be disclosed, nor will participation lead to disclosure of any specifics that one has contributed to the survey. It is very possible that conclusions drawn from the survey may reflect statistical information about an individual. A health survey intended to discover early indicators of a particular ailment may produce strong, even conclusive results; that these conclusions hold for a given individual is not evidence of a differential privacy violation; the individual may not even have participated in the survey (again, differential privacy ensures that these conclusive results would be obtained with very similar probability whether or not the individual participated in the survey). In particular, if the survey teaches us that specific *private* attributes correlate strongly with *publicly observable* attributes, this is not a violation of differential privacy, since this same correlation would be observed with almost the same probability independent of the presence or absence of any respondent.

Qualitative Properties of Differential Privacy. Having introduced and formally defined differential privacy, we recapitulate its key desirable qualities.

1. *Protection against arbitrary risks*, moving beyond protection against re-identification.
2. *Automatic neutralization of linkage attacks*, including all those attempted with all past, present, *and future* datasets and other forms and sources of auxiliary information.
3. *Quantification of privacy loss*. Differential privacy is not a binary concept, and has a measure of privacy loss. This permits comparisons among different techniques: for a fixed bound on privacy loss, which technique provides better accuracy? For a fixed accuracy, which technique provides better privacy?

4. *Composition.* Perhaps most crucially, the quantification of loss also permits the analysis and control of cumulative privacy loss over multiple computations. Understanding the behavior of differentially private mechanisms under composition enables the design and analysis of complex differentially private algorithms from simpler differentially private building blocks.
5. *Group Privacy.* Differential privacy permits the analysis and control of privacy loss incurred by groups, such as families.
6. *Closure Under Post-Processing* Differential privacy is immune to post-processing: A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a differentially private algorithm M and make it less differentially private. That is, a data analyst cannot increase privacy loss, either under the formal definition or even in any intuitive sense, simply by sitting in a corner and thinking about the output of the algorithm, *no matter what auxiliary information is available*.

These are the signal attributes of differential privacy. Can we prove a converse? That is, do these attributes, or some subset thereof, imply differential privacy? Can differential privacy be weakened in these respects and still be meaningful? These are open questions.

2.3.3 Final remarks on the definition

The Granularity of Privacy. Claims of differential privacy should be carefully scrutinized to ascertain the level of granularity at which privacy is being promised. Differential privacy promises that the behavior of an algorithm will be roughly unchanged even if a single entry in the database is modified. But what constitutes a single entry in the database? Consider for example a database that takes the form of a *graph*. Such a database might encode a social network: each individual $i \in [n]$ is represented by a vertex in the graph, and friendships between individuals are represented by edges.

We could consider differential privacy at a level of granularity corresponding to individuals: that is, we could require that differentially

private algorithms be insensitive to the addition or removal of any *vertex* from the graph. This gives a strong privacy guarantee, but might in fact be stronger than we need. the addition or removal of a single vertex could after all add or remove up to n edges in the graph. Depending on what it is we hope to learn from the graph, insensitivity to n edge removals might be an impossible constraint to meet.

We could on the other hand consider differential privacy at a level of granularity corresponding to edges, and ask our algorithms to be insensitive only to the addition or removal of single, or small numbers of, *edges* from the graph. This is of course a weaker guarantee, but might still be sufficient for some purposes. Informally speaking, if we promise ε -differential privacy at the level of a single edge, then no data analyst should be able to conclude anything about the existence of any subset of $1/\varepsilon$ edges in the graph. In some circumstances, large groups of social contacts might not be considered sensitive information: for example, an individual might not feel the need to hide the fact that the majority of his contacts are with individuals in his city or workplace, because where he lives and where he works are public information. On the other hand, there might be a small number of social contacts whose existence is highly sensitive (for example a prospective new employer, or an intimate friend). In this case, edge privacy should be sufficient to protect sensitive information, while still allowing a fuller analysis of the data than vertex privacy. Edge privacy will protect such an individual's sensitive information provided that he has fewer than $1/\varepsilon$ such friends.

As another example, a differentially private movie recommendation system can be designed to protect the data in the training set at the “event” level of single movies, hiding the viewing/rating of any single movie but not, say, hiding an individual's enthusiasm for cowboy westerns or gore, or at the “user” level of an individual's entire viewing and rating history.

All Small Epsilons Are Alike. When ε is small, $(\varepsilon, 0)$ -differential privacy asserts that for all pairs of adjacent databases x, y and all outputs o , an adversary cannot distinguish which is the true database

on the basis of observing o . When ε is small, *failing* to be $(\varepsilon, 0)$ -differentially private is not necessarily alarming — for example, the mechanism may be $(2\varepsilon, 0)$ -differentially private. The nature of the privacy guarantees with differing but small epsilons are quite similar. But what of large values for ε ? Failure to be $(15, 0)$ -differentially private merely says there exist neighboring databases and an output o for which the ratio of probabilities of observing o conditioned on the database being, respectively, x or y , is large. An output of o might be very unlikely (this is addressed by (ε, δ) -differential privacy); databases x and y might be terribly contrived and unlikely to occur in the “real world”; the adversary may not have the right auxiliary information to recognize that a revealing output has occurred; or may not know enough about the database(s) to determine the value of their symmetric difference. Thus, much as a weak cryptosystem may leak anything from only the least significant bit of a message to the complete decryption key, the failure to be $(\varepsilon, 0)$ - or (ε, δ) -differentially private may range from effectively meaningless privacy breaches to complete revelation of the entire database. A large epsilon is large after its own fashion.

A Few Additional Formalisms. Our privacy mechanism \mathcal{M} will often take some auxiliary parameters w as input, in addition to the database x . For example, w may specify a query q_w on the database x , or a collection \mathcal{Q}_w of queries. The mechanism $\mathcal{M}(w, x)$ might (respectively) respond with a differentially private approximation to $q_w(x)$ or to some or all of the queries in \mathcal{Q}_w . For all $\delta \geq 0$, we say that a mechanism $\mathcal{M}(\cdot, \cdot)$ satisfies (ε, δ) -differential privacy if for every w , $\mathcal{M}(w, \cdot)$ satisfies (ε, δ) -differential privacy.

Another example of a parameter that may be included in w is a *security parameter* κ to govern how small $\delta = \delta(\kappa)$ should be. That is, $\mathcal{M}(\kappa, \cdot)$ should be $(\varepsilon, \delta(\kappa))$ differentially private for all κ . Typically, and throughout this monograph, we require that δ be a negligible function in κ , i.e., $\delta = \kappa^{-\omega(1)}$. Thus, we think of δ as being cryptographically small, whereas ε is typically thought of as a moderately small constant.

In the case where the auxiliary parameter w specifies a collection $\mathcal{Q}_w = \{q : \mathcal{X}^n \rightarrow \mathbb{R}\}$ of queries, we call the mechanism \mathcal{M} a

synopsis generator. A synopsis generator outputs a (differentially private) synopsis \mathcal{A} which can be used to compute answers to all the queries in \mathcal{Q}_w . That is, we require that there exists a reconstruction procedure R such that for each input v specifying a query $q_v \in \mathcal{Q}_w$, the reconstruction procedure outputs $R(\mathcal{A}, v) \in \mathbb{R}$. Typically, we will require that with high probability \mathcal{M} produces a synopsis \mathcal{A} such that the reconstruction procedure, using \mathcal{A} , computes accurate answers. That is, for all or most (weighted by some distribution) of the queries $q_v \in \mathcal{Q}_w$, the error $|R(\mathcal{A}, v) - q_v(x)|$ will be bounded. We will occasionally abuse notation and refer to the reconstruction procedure taking as input the actual query q (rather than some representation v of it), and outputting $R(\mathcal{A}, q)$.

A special case of a synopsis is a *synthetic database*. As the name suggests, the rows of a synthetic database are of the same type as rows of the original database. An advantage to synthetic databases is that they may be analyzed using the same software that the analyst would use on the original database, obviating the need for a special reconstruction procedure R .

Remark 2.1. Considerable care must be taken when programming real-valued mechanisms, such as the Laplace mechanism, due to subtleties in the implementation of floating point numbers. Otherwise differential privacy can be destroyed, as outputs with non-zero probability on a database x , may, because of rounding, have zero probability on adjacent databases y . This is just one way in which the implementation of floating point requires scrutiny in the context of differential privacy, and it is not unique.

2.4 Bibliographic notes

The definition of differential privacy is due to Dwork et al. [23]; the precise formulation used here and in the literature first appears in [20] and is due to Dwork and McSherry. The term “differential privacy” was coined by Michael Schroeder. The impossibility of semantic security is due to Dwork and Naor [25]. Composition and group privacy for $(\epsilon, 0)$ -differentially private mechanisms is first addressed in [23].

Composition for (ε, δ) -differential privacy was first addressed in [21] (but see the corrected proof in Appendix B, due to Dwork and Lei [22]). The vulnerability of differential privacy to inappropriate implementations of floating point numbers was observed by Mironov, who proposed a mitigation [63].