CS 591: Formal Methods in Security and Privacy

Differential Privacy

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

From the previous classes

An example

```
OneTimePad(m : private msg) : public msg
  key :=$ Uniform({0,1}^n);
   cipher := m xor key;
   return cipher
```

Learning a ciphertext does not change any a priori knowledge about the likelihood of messages.

Probabilistic Relational Hoare Quadruples

Precondition

Program$_1$ ~ Program$_2$

Postcondition

Precondition

$$c_1 \sim c_2 : P \Rightarrow Q$$

Probabilistic Program

Probabilistic Program

Postcondition

Relational Assertions

$$c_1 \sim c_2 : P \Rightarrow Q$$

logical formula over pair of memories (i.e. relation over memories)

logical formula over ????

R-Coupling

Given two distributions $\mu_1 \in D(A)$, and $\mu_2 \in D(B)$, an R-coupling between them, for $R \subseteq A \times B$, is a joint distribution $\mu \in D(A \times B)$ such that:

1) the marginal distributions of μ are μ_1 and μ_2, respectively,
2) the support of μ is contained in R. That is, if $\mu(a,b)>0$, then $(a,b) \in R$.

Relational lifting of a predicate

We say that two subdistributions $\mu_1 \in D(A)$ and $\mu_2 \in D(B)$ are in the relational lifting of the relation $\mathtt{R} \subseteq A \times B$, denoted $\mu_1\ \mathtt{R^*}\ \mu_2$ if and only if there exist an $\mathtt{R}$-coupling between them.

Validity of Probabilistic Hoare quadruple

We say that the quadruple $c_1 \sim c_2 : P \Rightarrow Q$ is valid if and only if for every pair of memories m_1, m_2 such that $P(m_1, m_2)$ we have:

$\{c_1\}_{m1} = \mu_1$ and $\{c_2\}_{m2} = \mu_2$ implies $Q^*(\mu_1, \mu_2)$.

Differential Privacy

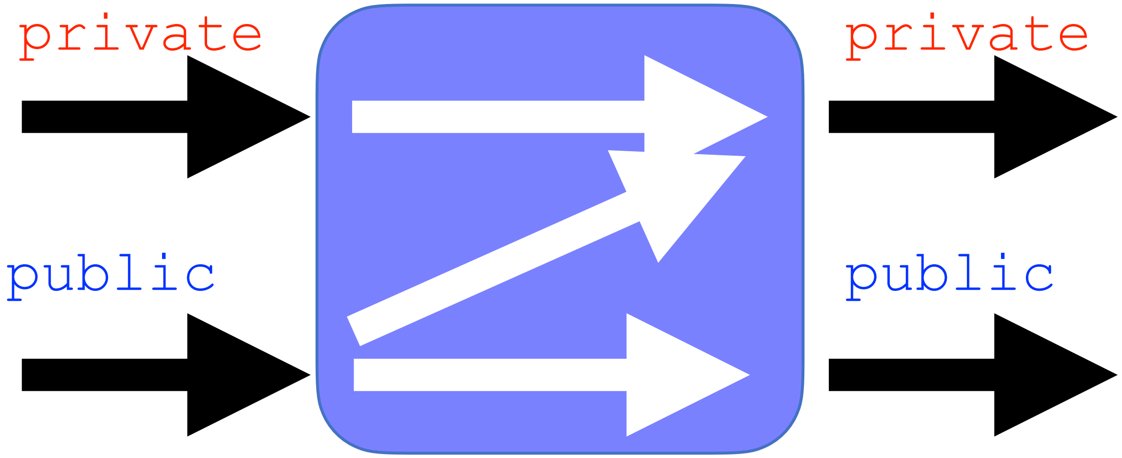
Data

Releasing the mean of Some Data

```
Mean(d : private data) : public real
   i:=0;
   s:=0;
   while (i<size(d))
      s:=s + d[i]
      i:=i+1;
   return (s/i)
```

Releasing the mean of Some Data

```
Mean(d : private data) : public real
   i:=0;
   s:=0;
   while (i<size(d))
      s:=s + d[i]
      i:=i+1;
   return (s/i)
```

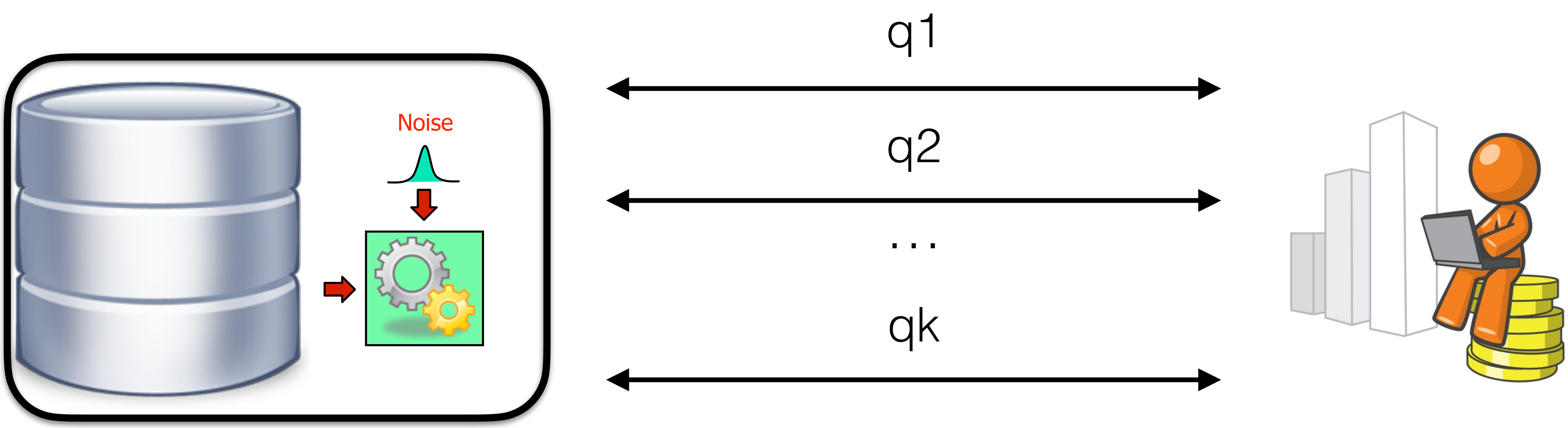


Privacy-preserving data analysis?

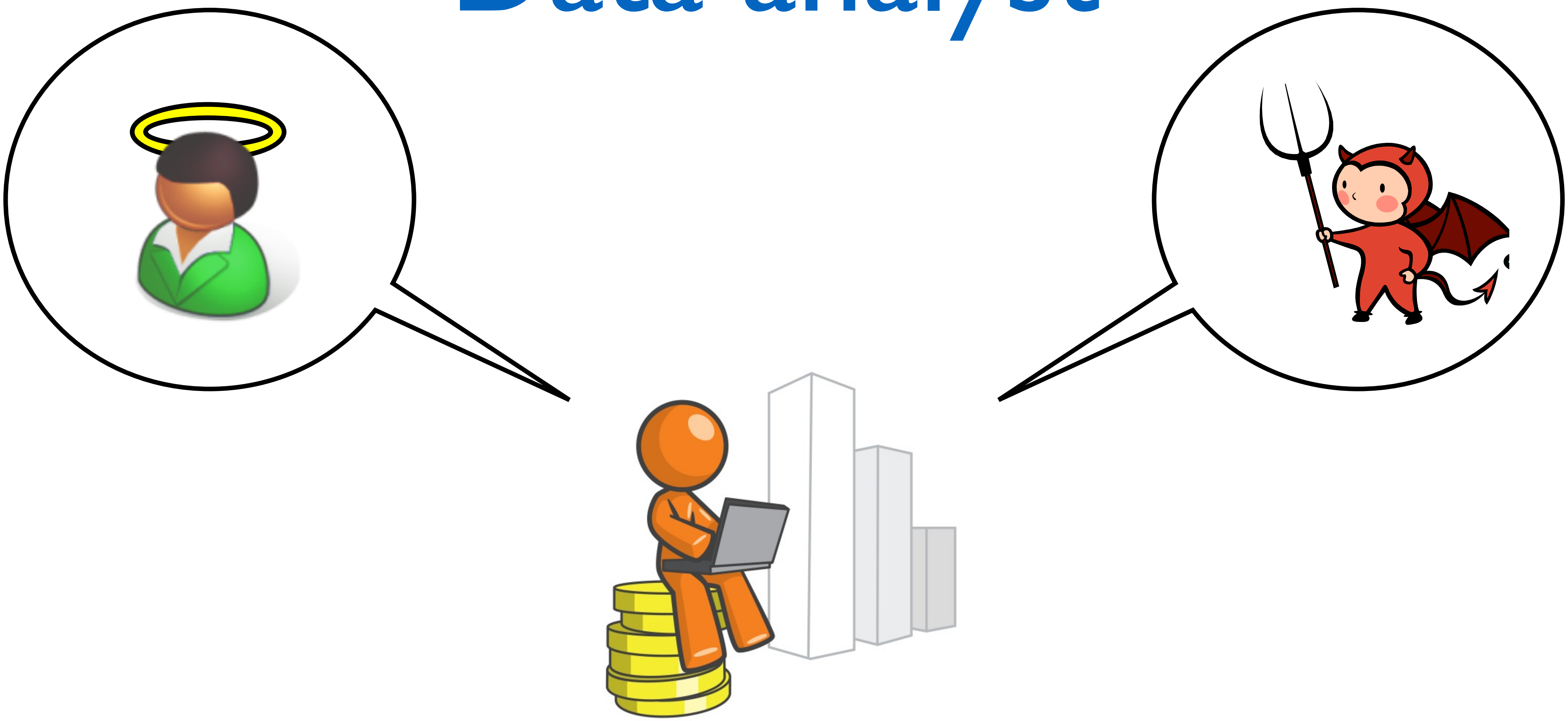
We want to release some information to a data analyst and protect the privacy of the individuals contributing their data.

Privacy-preserving data analysis?

We want to release some information to a data analyst and protect the privacy of the individuals contributing their data.



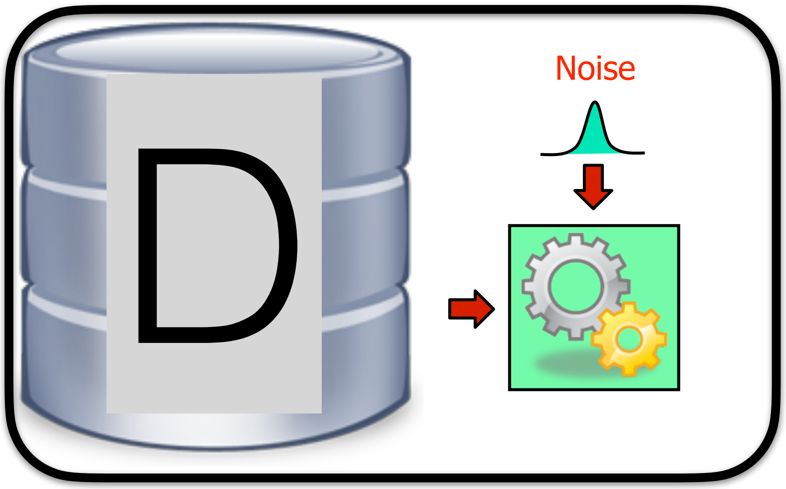
Data analyst



Fundamental Law of Information Reconstruction

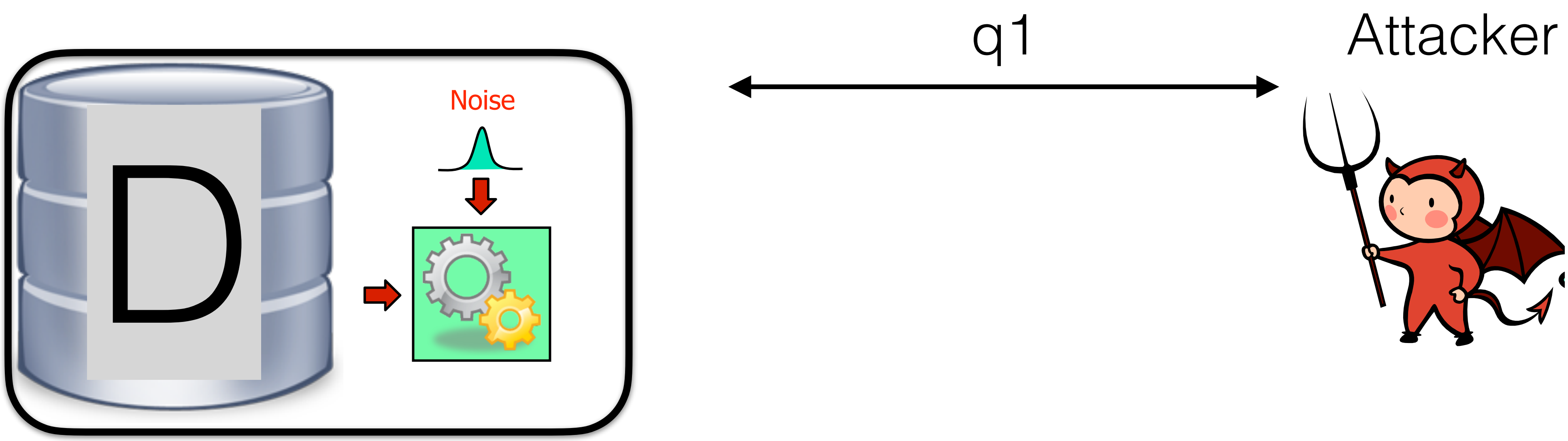
The release of too many overly accurate statistics permits reconstruction attacks.

Reconstruction attack

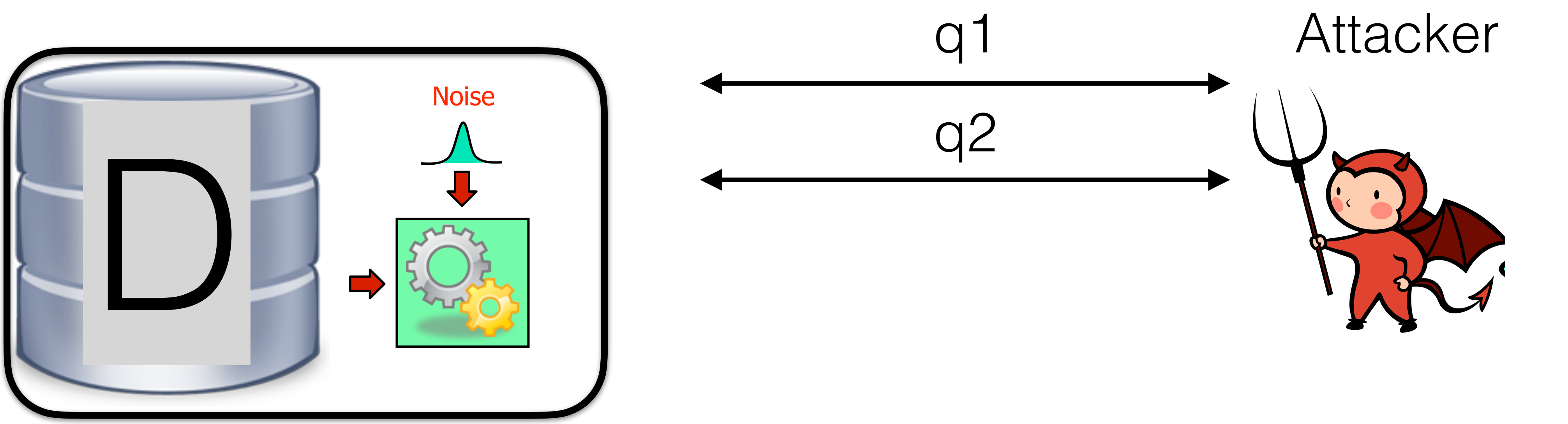


Attacker

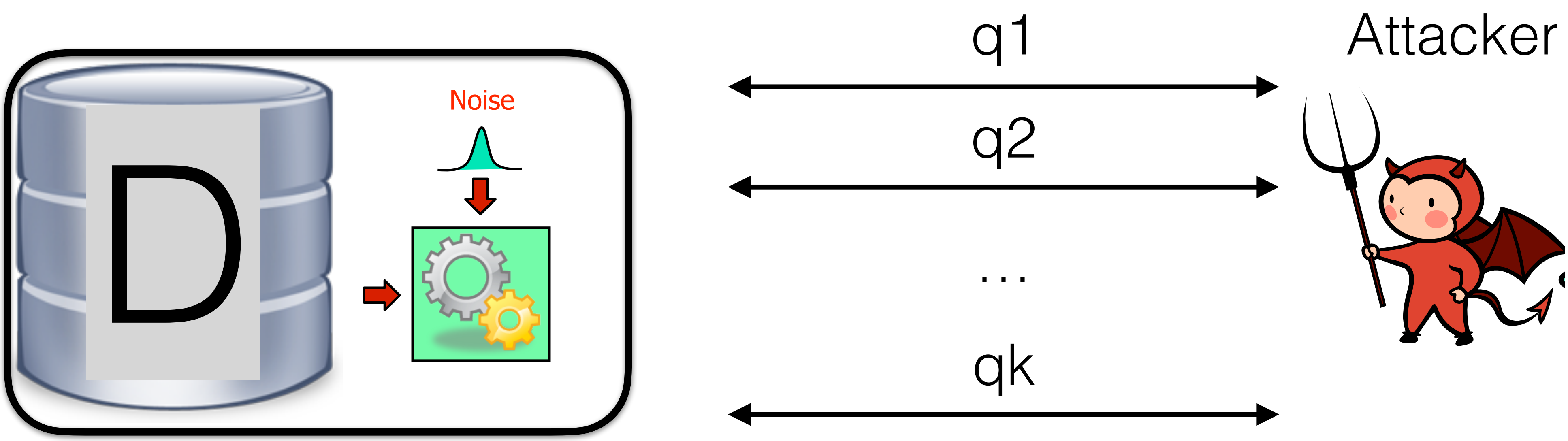
Reconstruction attack



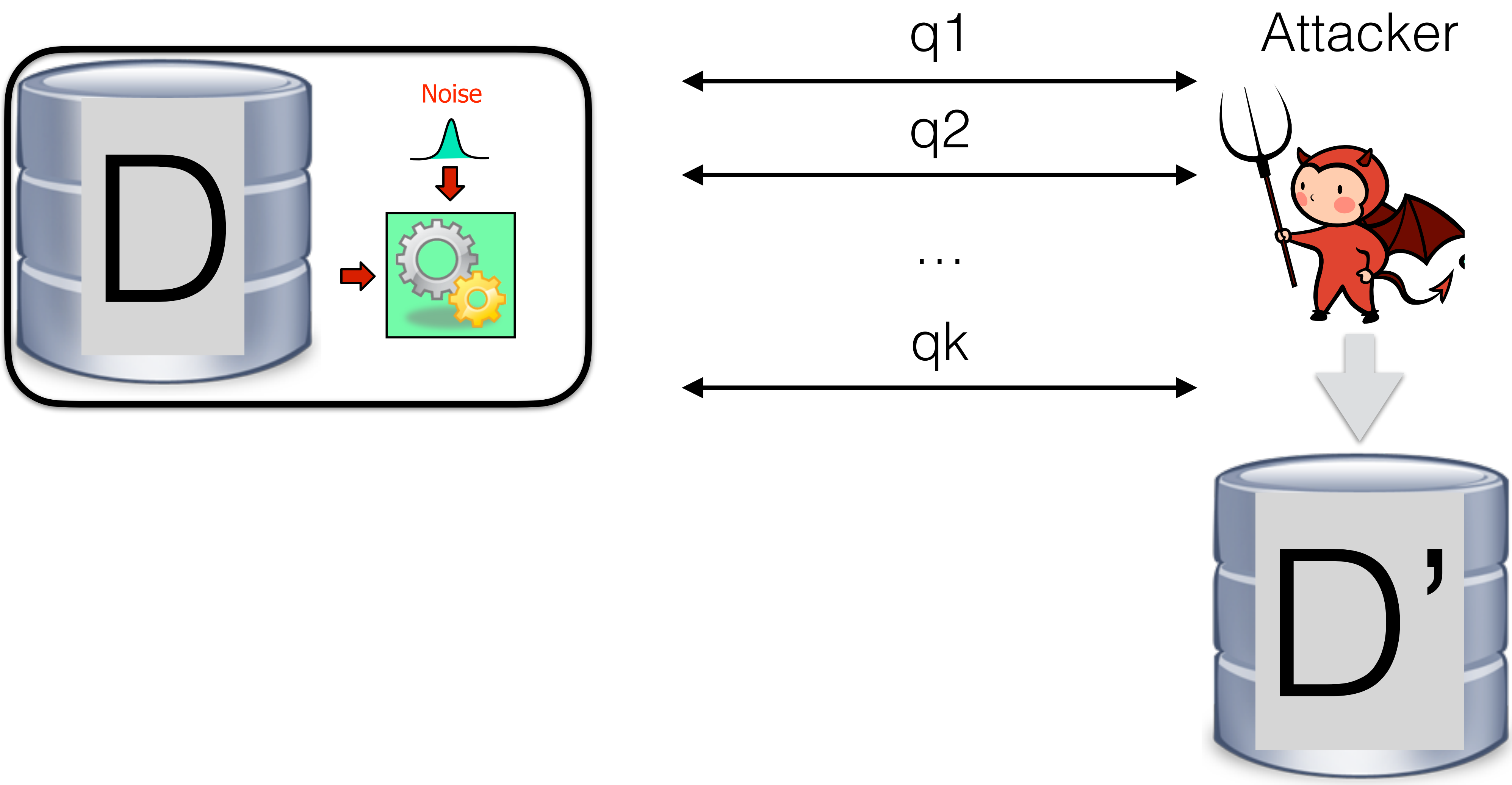
Reconstruction attack



Reconstruction attack



Reconstruction attack



Reconstruction attack

We say that the attacker wins if

$$d(D, D') \sim 0$$

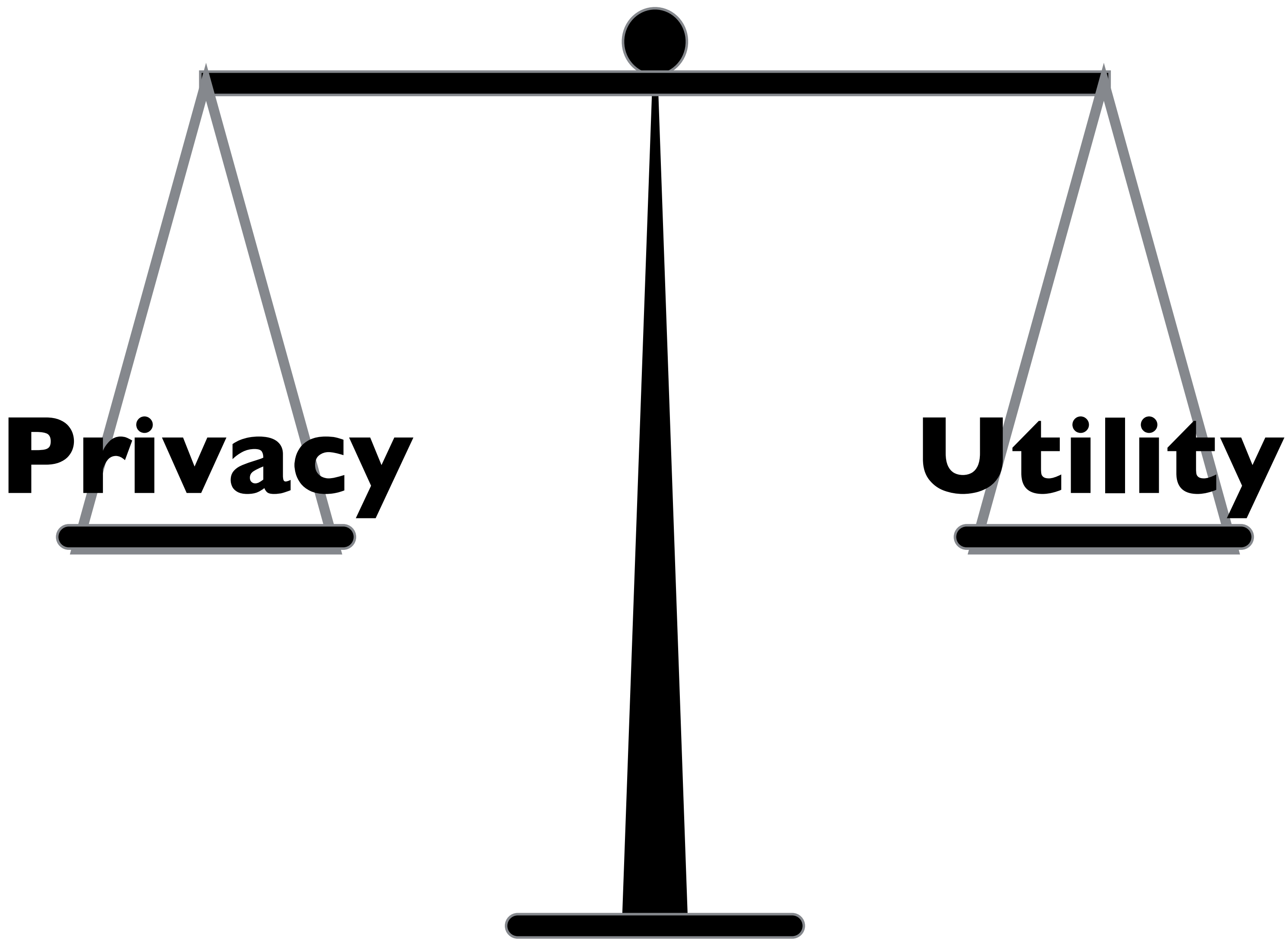
Reconstruction attack

We say that the attacker wins if

$$d(D, D') \sim 0$$

In this class case we can use Hamming distance

Privacy vs Utility



Quantitative notions of Privacy

- The impossibility results discussed above suggest a quantitative notion of privacy,
- a notion where the privacy loss depends on the number of queries that are allowed,
- and on the accuracy with which we answer them.

Differential privacy: understanding the mathematical and computational meaning of this trade-off.

[Dwork, McSherry, Nissim, Smith, TCC06]

Privacy-preserving data analysis?

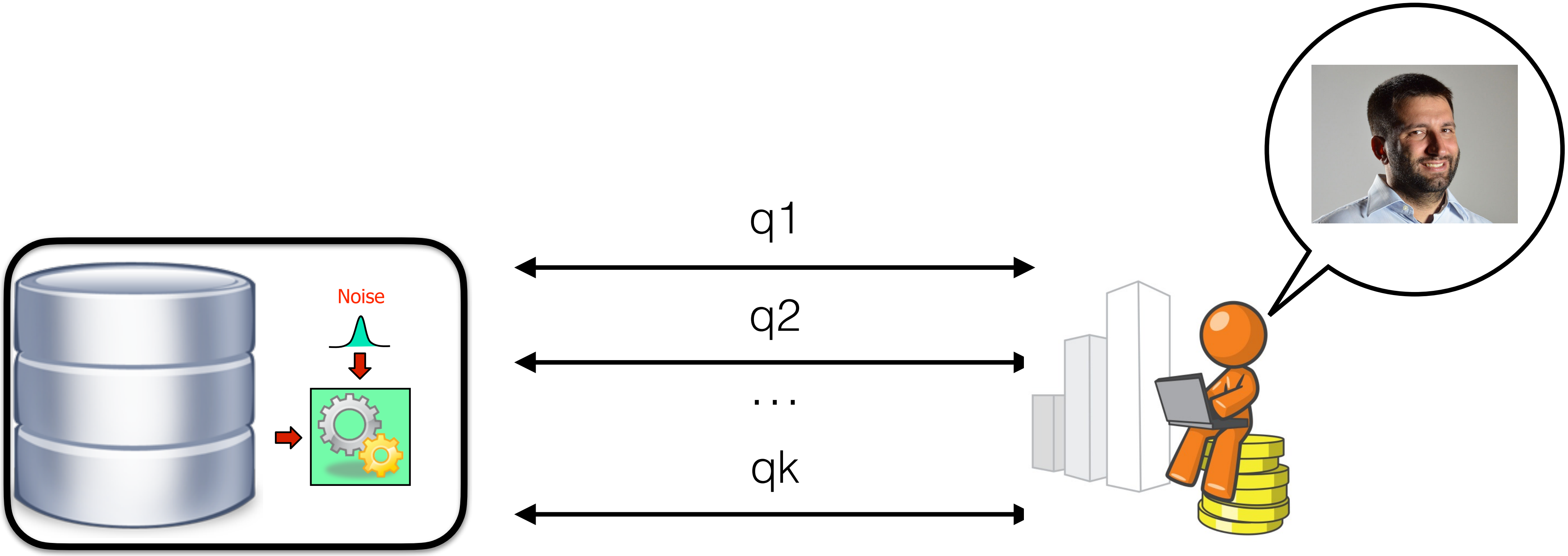
- The analyst knows no more about me after the analysis than what she knew before the analysis.

Privacy-preserving data analysis?

- The analyst knows no more about me after the analysis than what she knew before the analysis.

Privacy-preserving data analysis?

- The analyst knows no more about me after the analysis than what she knew before the analysis.



Privacy-preserving data analysis?

Prior Knowledge

~

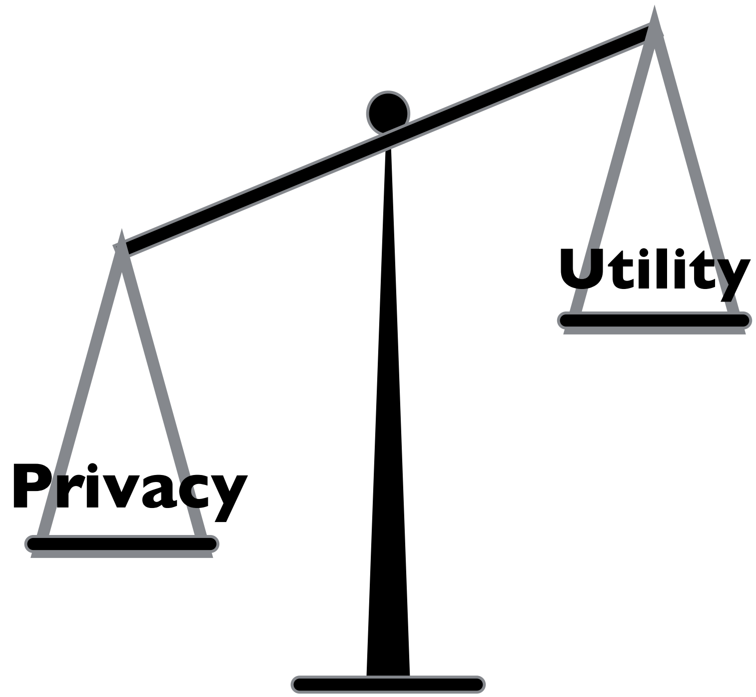
Posterior Knowledge

Privacy-preserving data analysis?

Privacy-preserving data analysis?

Question: What is the problem with this requirement?

Privacy-preserving data analysis?



If nothing can be learned about an individual, then nothing at all can be learned at all!

[DworkNaor10]

Privacy-preserving data analysis?

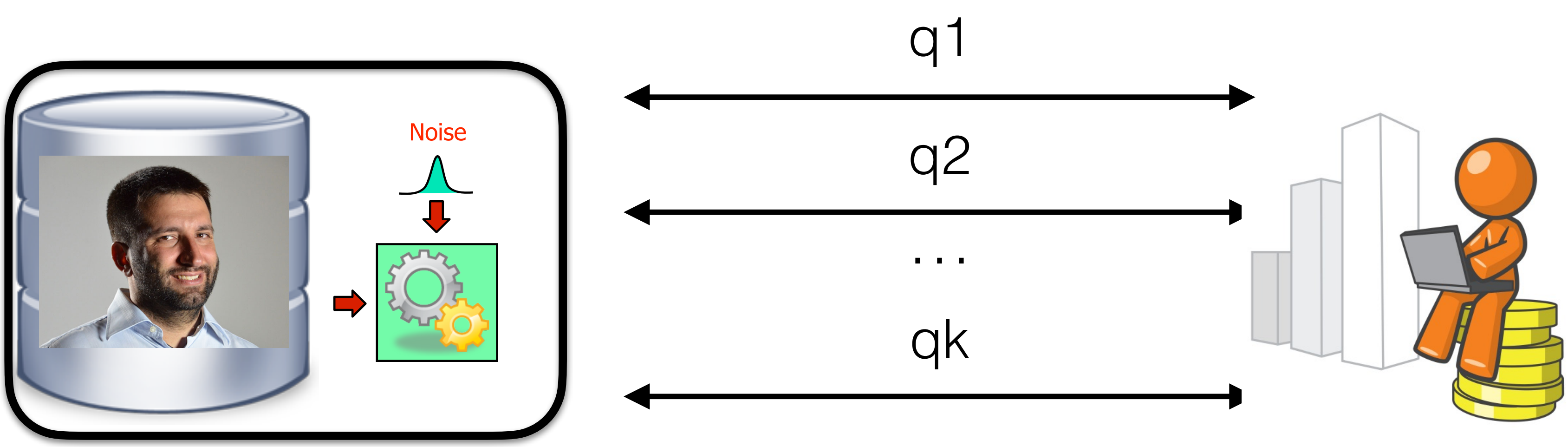
- The analyst learn almost the same about me after the analysis as what she would have learnt if I didn't contribute my data.

Privacy-preserving data analysis?

- The analyst learn almost the same about me after the analysis as what she would have learnt if I didn't contribute my data.

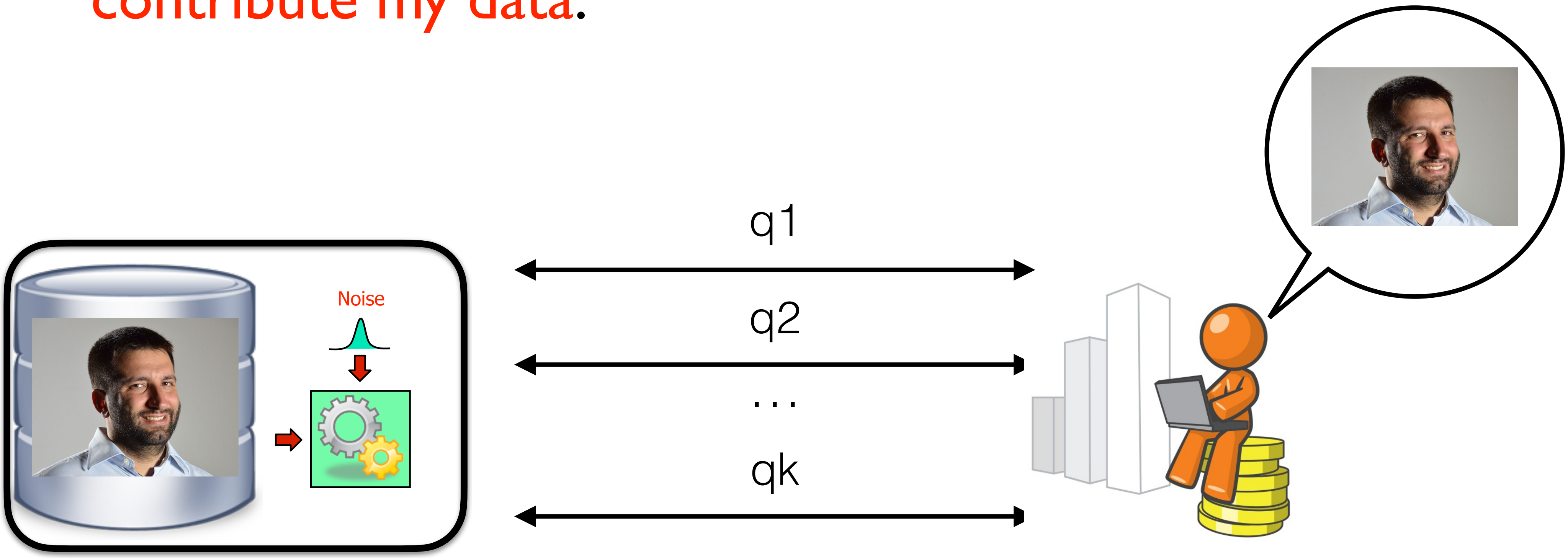
Privacy-preserving data analysis?

- The analyst learn almost the same about me after the analysis as what she would have learnt if I didn't contribute my data.



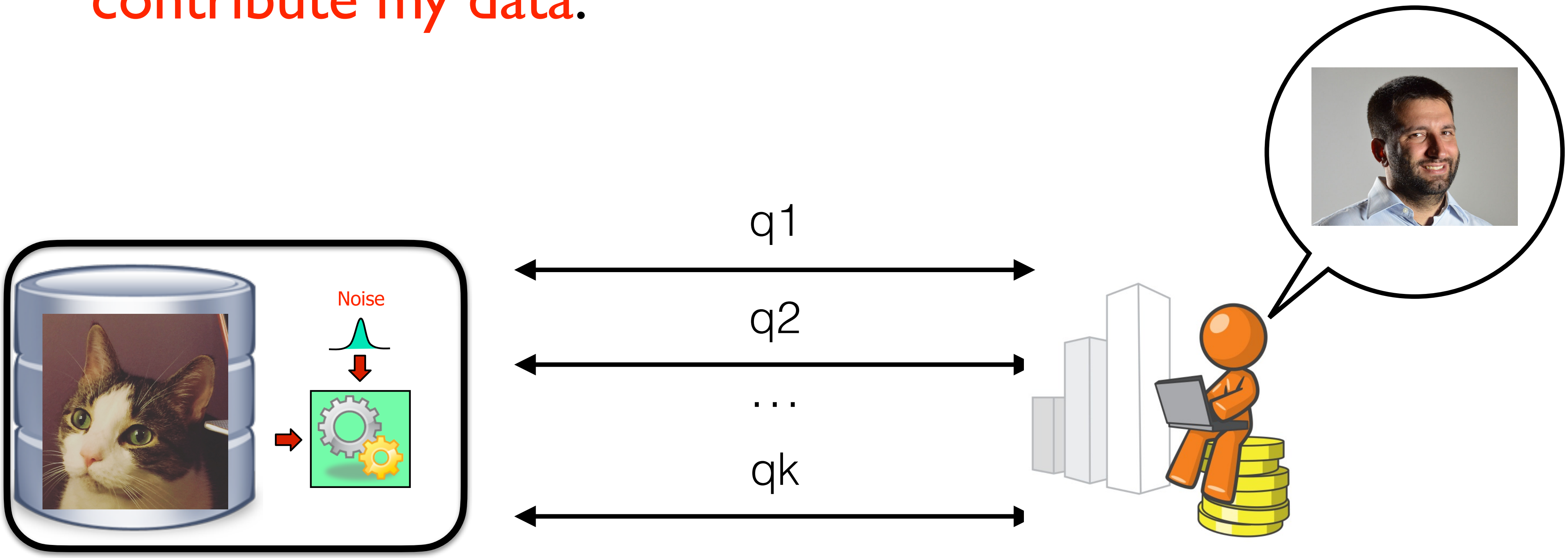
Privacy-preserving data analysis?

- The analyst learn almost the same about me after the analysis as what she would have learnt if I didn't contribute my data.



Privacy-preserving data analysis?

- The analyst learn almost the same about me after the analysis as what she would have learnt if I didn't contribute my data.



Adjacent databases

- We can formalize the concept of contributing my data or not in terms of a notion of distance between datasets.
- Given two datasets D, D'$\in$DB, their distance is defined as:

$$D\Delta D' = |\{k \leq n \mid D(k) \neq D'(k)\}|$$

- We will call two datasets adjacent when $D\Delta D'=1$ and we will write $D \sim D'$.

Privacy Loss

In general we can think about the following quantity as the privacy loss incurred by observing r on the databases b and b'.

$$L_{b,b'}(r) = \log \frac{\Pr[Q(b)=r]}{\Pr[Q(b')=r]}$$

(ε,δ)-Differential Privacy

Definition

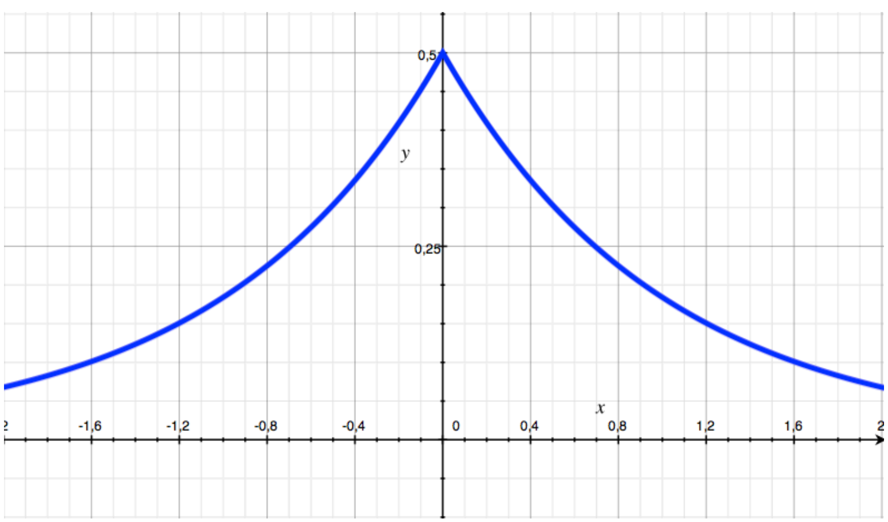
Given $\varepsilon,\delta \geq 0$, a probabilistic query $Q: X^n \rightarrow R$ is (ε,δ)-differentially private iff for all adjacent database b_1, b_2 and for every $S \subseteq R$:

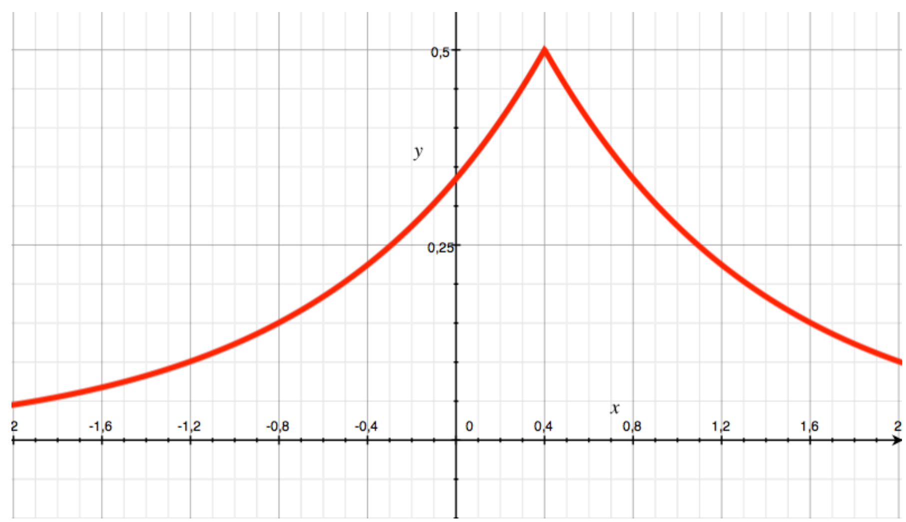
$$\Pr[Q(b_1) \in S] \leq \exp(\varepsilon) \Pr[Q(b_2) \in S] + \delta$$

Differential Privacy

Q : db => R   probabilistic

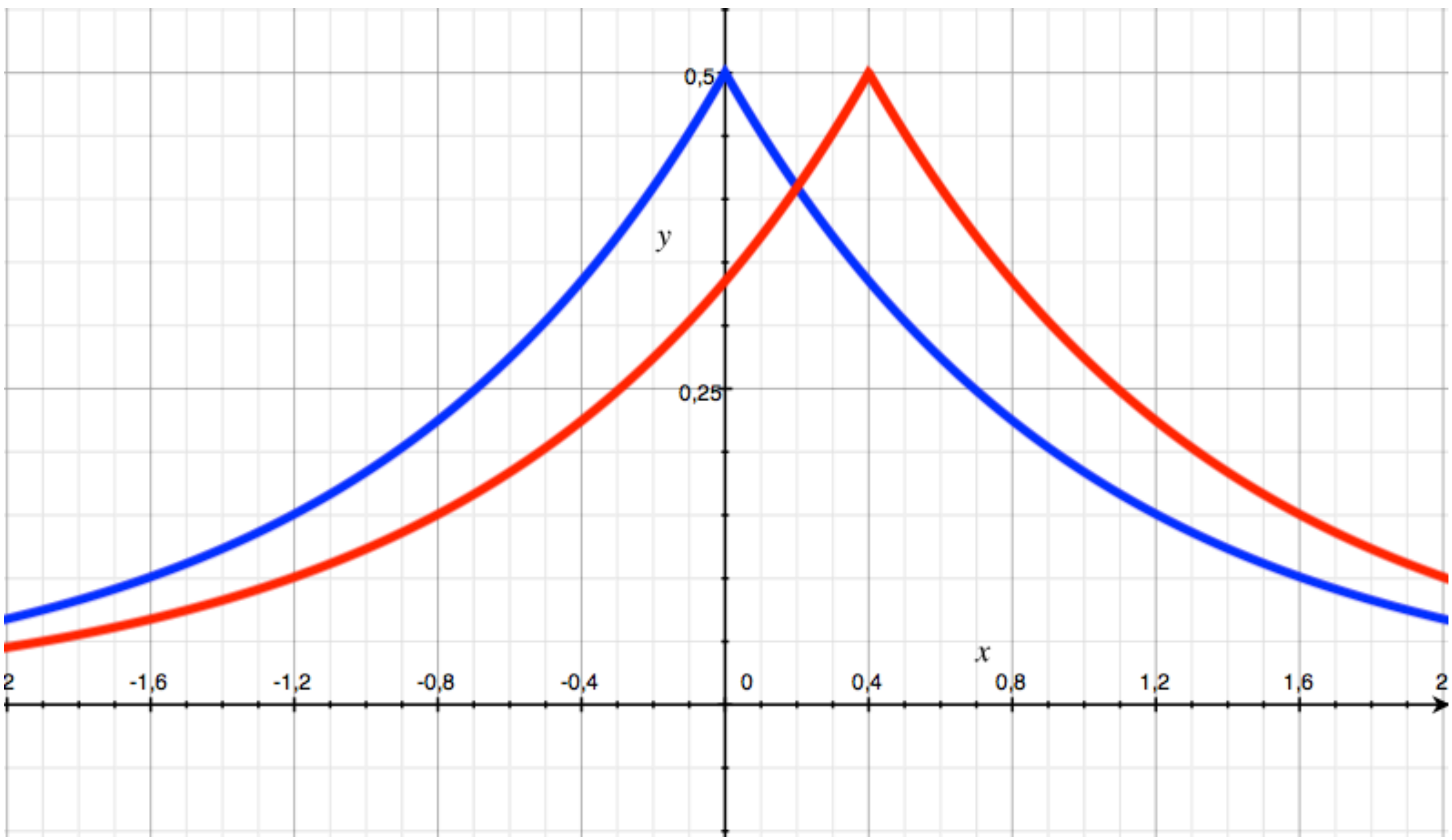
Q(b∪{x})   Q(b∪{y})





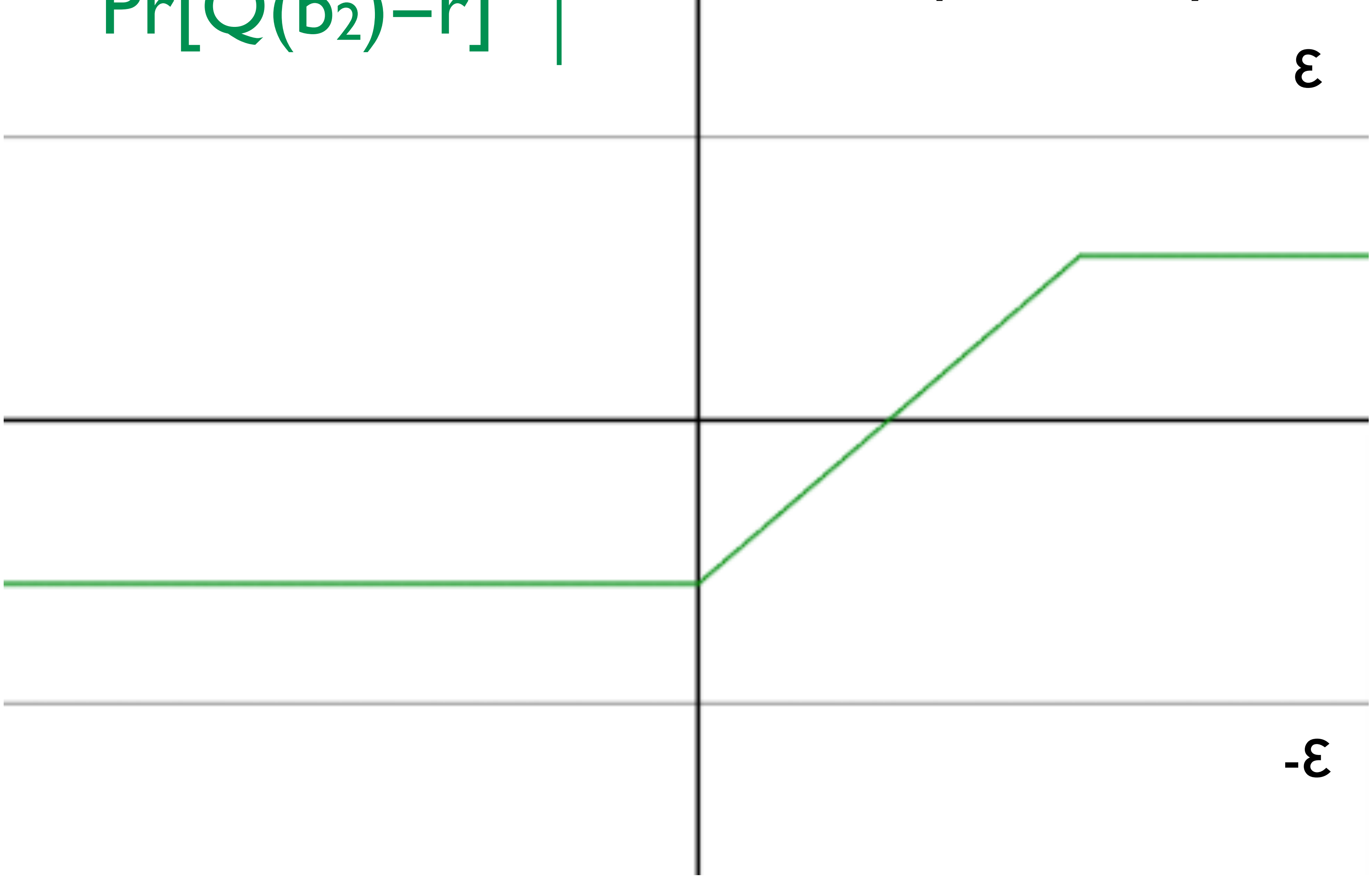
Differential Privacy

$d(Q(b \cup \{x\}), Q(b \cup \{y\})) \leq \varepsilon$ with probability $1-\delta$



(ε,δ)-Differential Privacy

$$\left| \log \frac{\Pr[Q(b_1)=r]}{\Pr[Q(b_2)=r]} \right| \leq \varepsilon$$ with probability $1-\delta$



(ε,δ)-indistinguishability

When we defined statistical distance:

$$\Delta(\mu_1,\mu_2)=\max_{E\subseteq A} \mid \mu_1(E)-\mu_2(E) \mid = \delta$$

we also used a notion of δ-indistinguishability.

We say that two distributions $\mu_1, \mu_2 \in D(A)$, are at δ-indistinguishable if:

$$\Delta(\mu 1,\mu 2)\leq \delta$$

(ε,δ)-indistinguishability

We can define a ε-skewed version of statistical distance. We call this notion ε-distance.

$$\Delta_{\varepsilon}(\mu 1,\mu 2)=\sup_{E\subseteq A} \max(\mu_1(E)-e^{\varepsilon}\mu_2(E),\ \mu_2(E)-e^{\varepsilon}\mu_1(E),0)$$

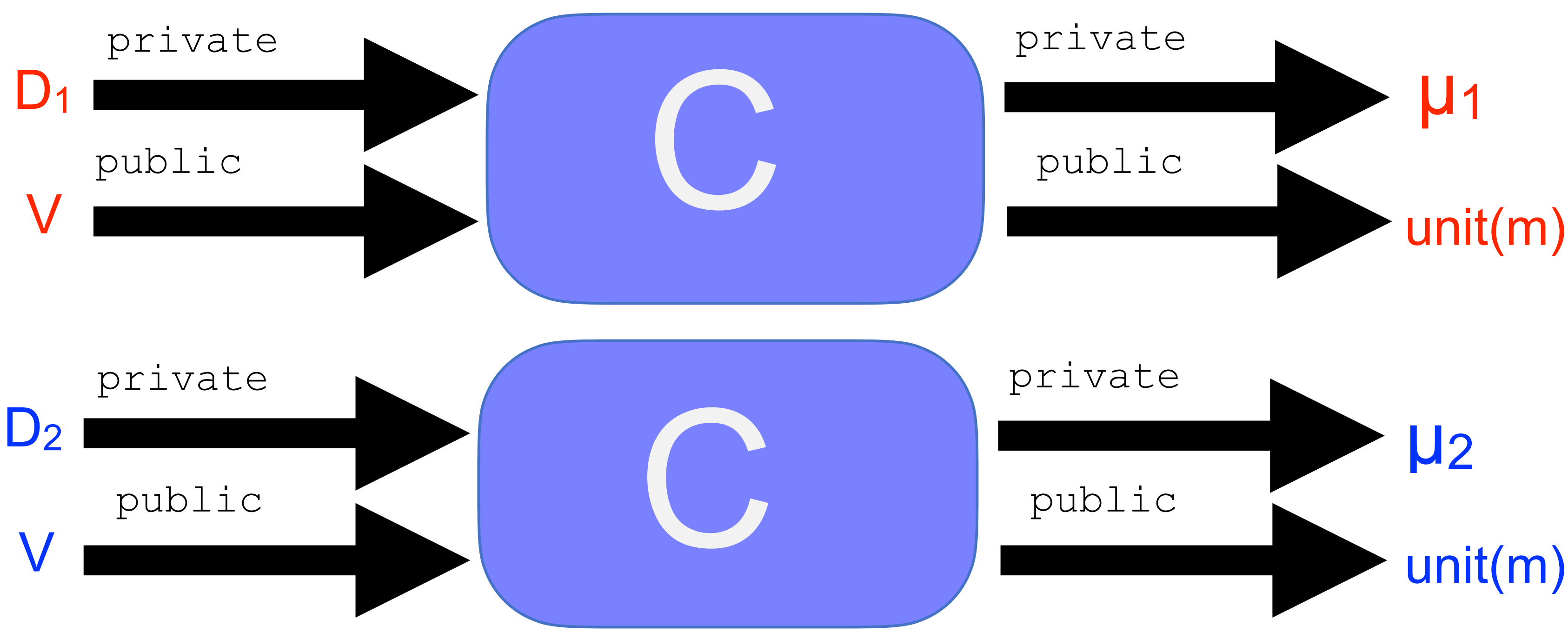
We say that two distributions $\mu_1, \mu_2 \in D(A)$, are at (ε,δ)-indistinguishable if:

$$\Delta_{\varepsilon}(\mu 1,\mu 2) \le \delta$$

Differential Privacy as a Relational Property

c is differentially private if and only if for every $m_1 \sim m_2$ (extending the notion of adjacency to memories):

$\{c\}_{m1}=\mu_1$ and $\{c\}_{m2}=\mu_2$ implies $\Delta_\varepsilon(\mu_1,\mu_2) \leq \delta$



Releasing the mean of Some Data

```
Mean(d : private data) : public real
   i:=0;
   s:=0;
   while (i<size(d))
      s:=s + d[i]
      i:=i+1;
   return (s/i)
```

Adding Noise

Question: What is a good way to add noise to the output of a statistical query to achieve (ε,0)-DP?

Adding Noise

Question: What is a good way to add noise to the output of a statistical query to achieve $(\varepsilon,0)$-DP?

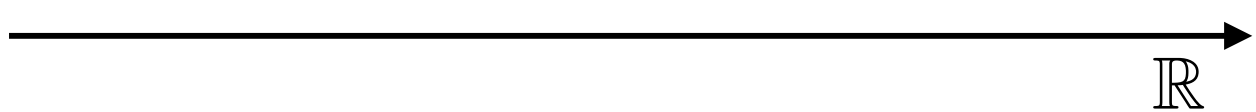
Intuitive answer: it should depend on ε or the accuracy we want to achieve, and on the scale that a change of an individual can have on the output.

Global Sensitivity

$$GS_q = \max\{\,|\,q(D) - q(D')|\ \text{s.t.}\ D \sim D'\}$$

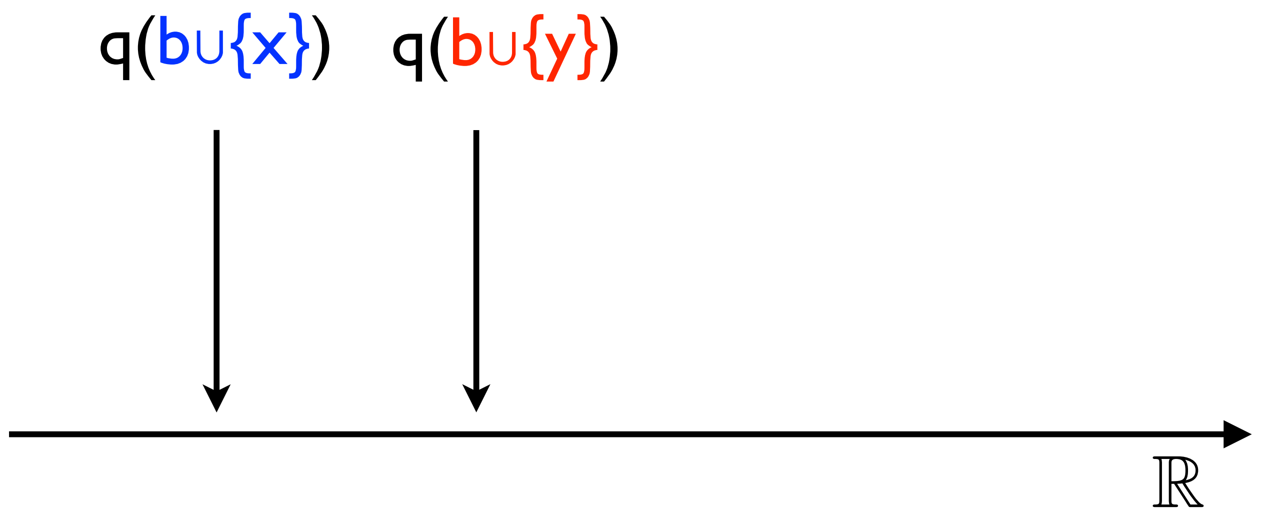
Global Sensitivity

$$GS_q = \max\{\, |q(D) - q(D')| \text{ s.t. } D \sim D' \}$$



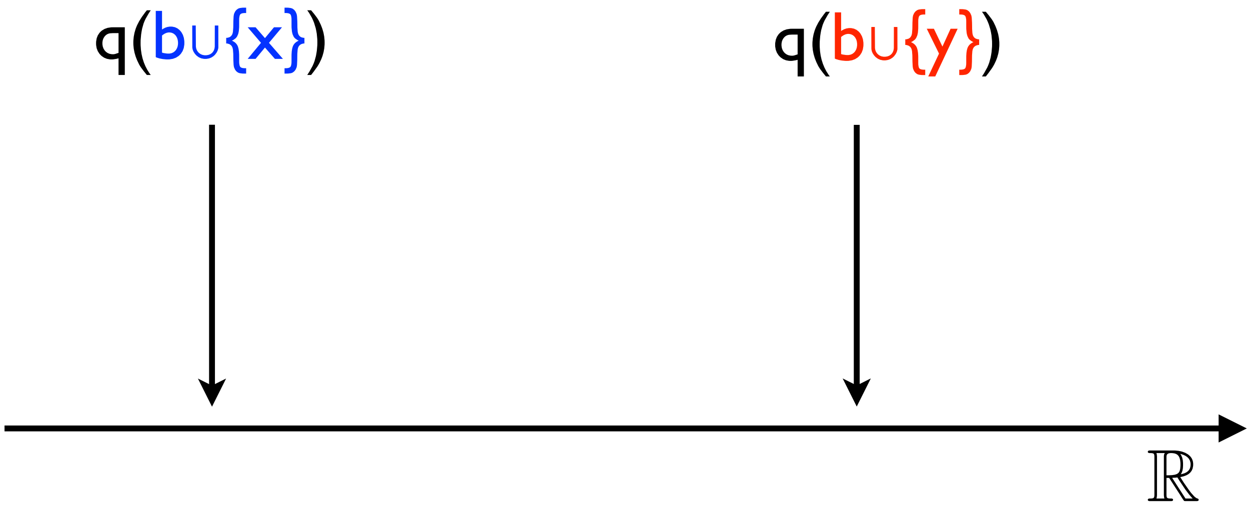
Global Sensitivity

$$GS_q = \max\{\,|q(D) - q(D')|\ \text{s.t.}\ D \sim D'\}$$



Global Sensitivity

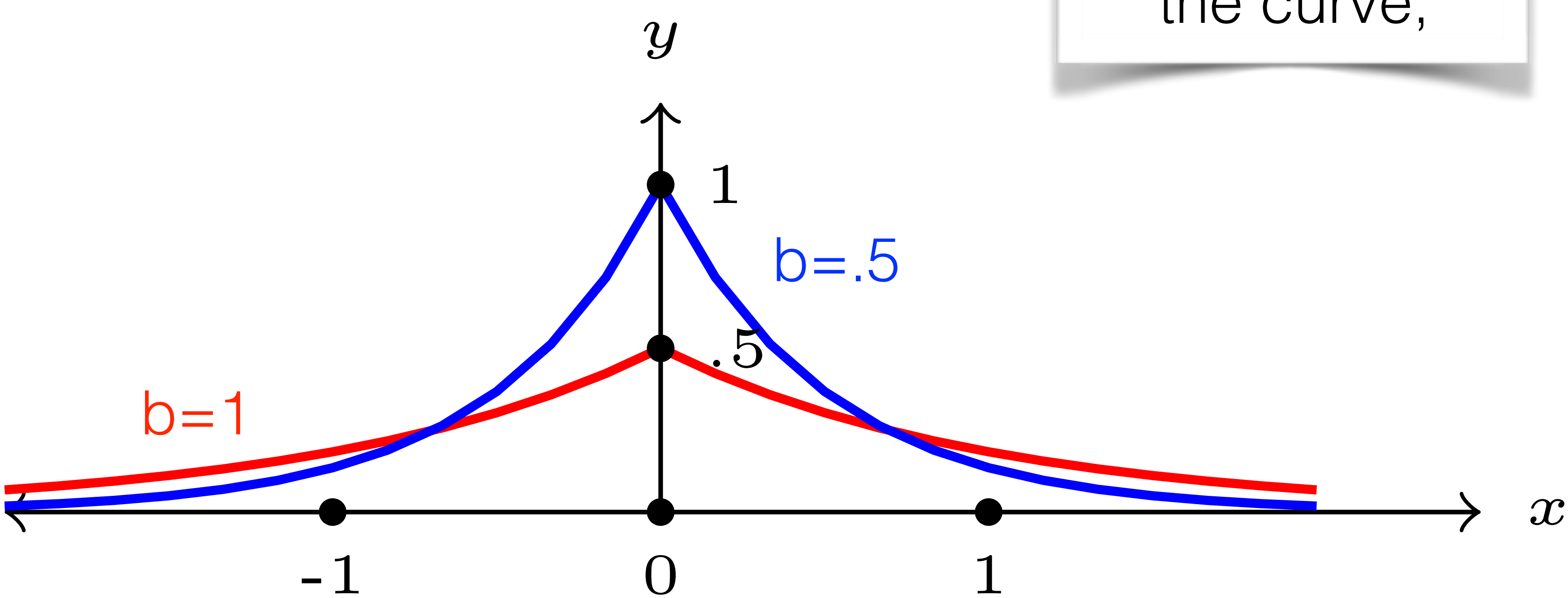
$$GS_q = \max\{\, |q(D) - q(D')| \text{ s.t. } D \sim D' \}$$



Laplace Distribution

$$\mathsf{Lap}(b,\mu)(X) = \frac{1}{2b}\exp\Big(-\frac{|\mu - X|}{b}\Big)$$

b regulates the skewness of the curve,



Releasing privately the mean of Some Data

```
Mean(d : private data) : public real
   i:=0;
   s:=0;
   while (i<size(d))
      s:=s + d[i]
      i:=i+1;
   z:=$ Laplace(sens/eps,0)
   z:= (s/i)+z
   return z
```

Laplace Mechanism

```
Lap(d : priv data)(f: data -> real)
    (e:real) : pub real
   z:=$ Laplace(GSf/e,0)
   z:= f(d)+z
   return z
```

Laplace Mechanism

```
Lap(d : priv data)(f: data -> real)
    (e:real) : pub real
   z:=$ Laplace(GSf/e,0)
   z:= f(d)+z
   return z
```

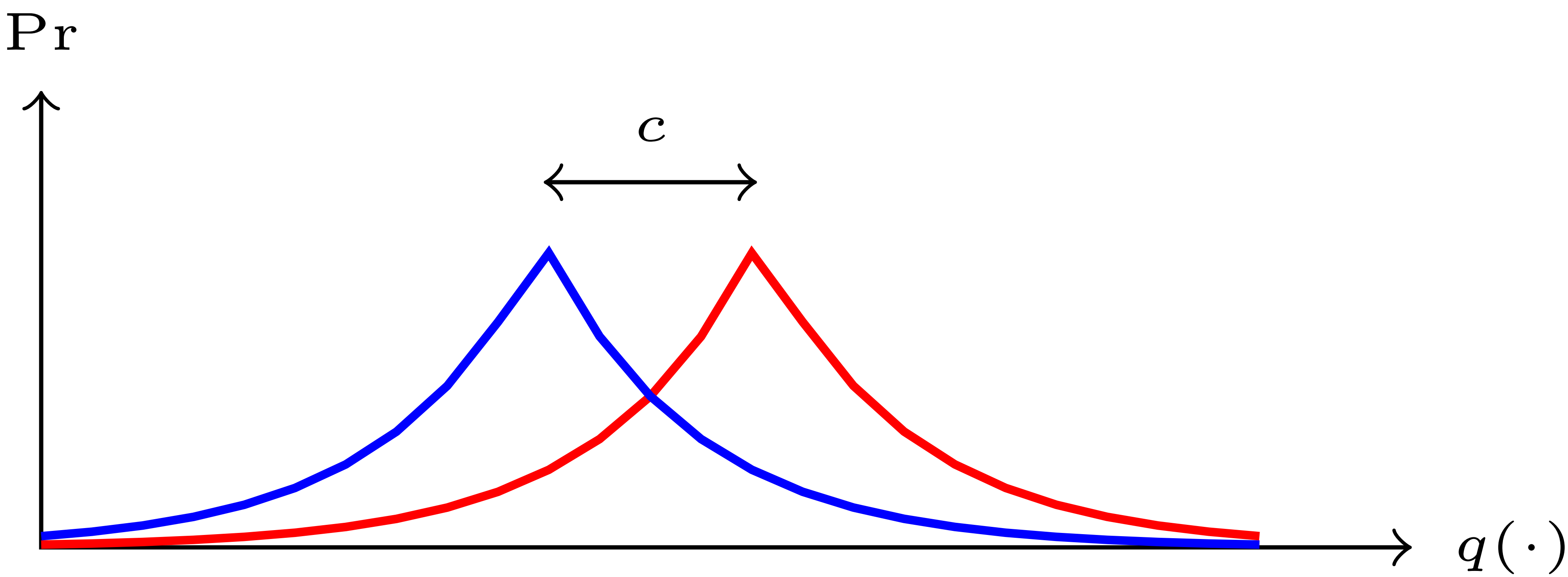
It turns out that we could also write it as:

```
Lap(d : priv data)(f: data -> real)
    (e:real) : pub real
   z:=$ Laplace(GSf/e,f(d))
   return z
```

Laplace Mechanism

Theorem (Privacy of the Laplace Mechanism)
The Laplace mechanism is (ε,0)-differentially private.

Proof: Intuitively



Laplace Mechanism

Theorem (Privacy of the Laplace Mechanism)
The Laplace mechanism is $(\varepsilon,0)$-differentially private.

Laplace Mechanism

Theorem (Privacy of the Laplace Mechanism)
The Laplace mechanism is $(\varepsilon,0)$-differentially private.

Laplace Mechanism

Question: How accurate is the answer that we get from the Laplace Mechanism?

Properties of Differential Privacy

Some important properties

- Resilience to post-processing
- Group privacy
- Composition

Some important properties

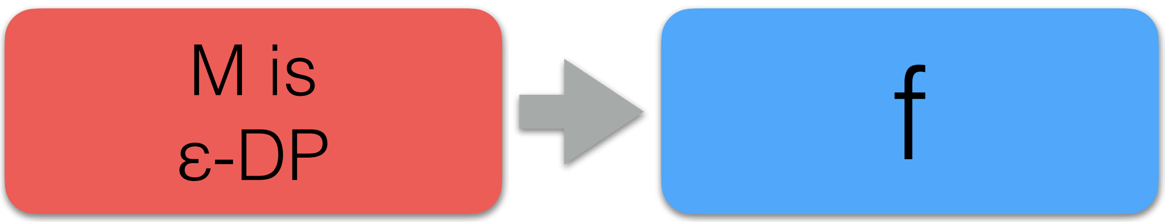
- Resilience to post-processing
- Group privacy
- Composition

We will look at them in the context of (ε,0)-differential privacy.

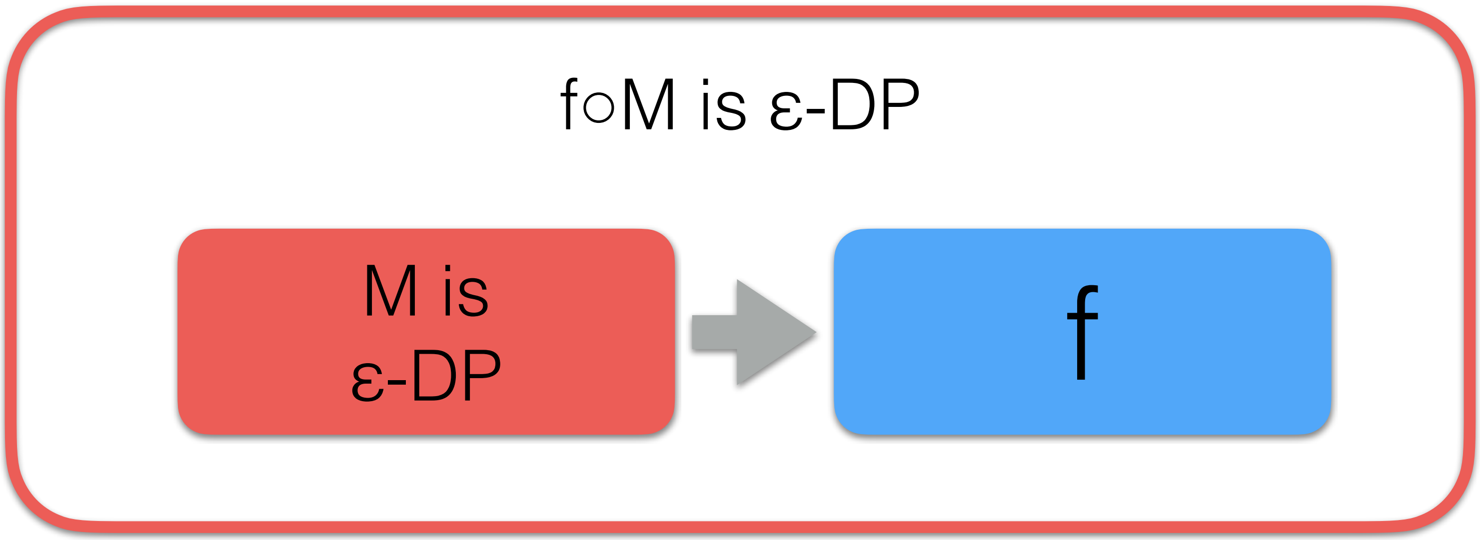
Resilience to Post-processing

M is
ε-DP

Resilience to Post-processing



Resilience to Post-processing



Resilience to Post-processing

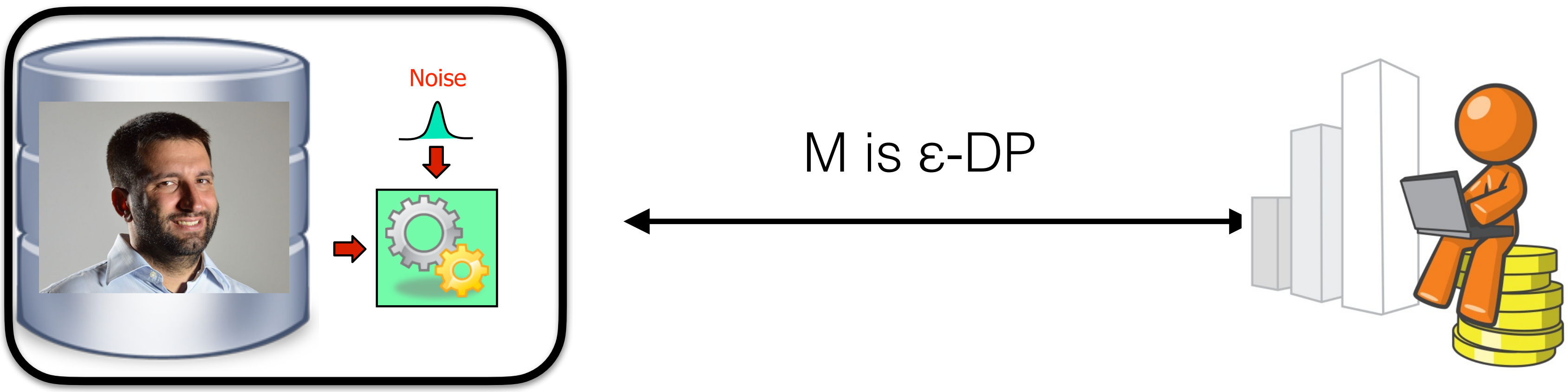
Question: Why is resilience to post-processing important?

Resilience to Post-processing

Question: Why is resilience to post-processing important?

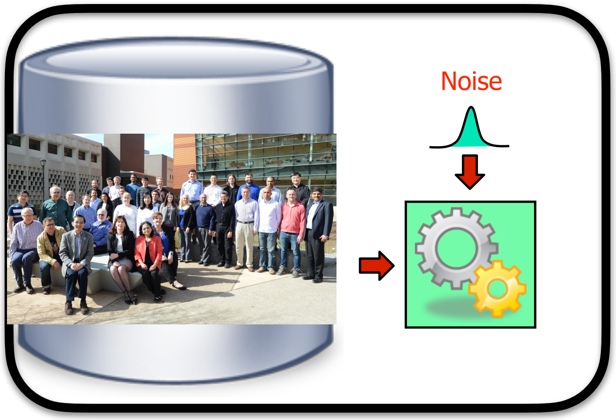
Answer: Because it is what allows us to publicly release the result of a differentially private analysis!

Group Privacy



$$\Pr[\mathcal{M}(D) = r] \leq e^{\epsilon}\,\Pr[\mathcal{M}(D') = r]$$

Group Privacy



M is kε-DP

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(k\epsilon) \Pr[\mathcal{M}(D') \in S]$$

Group Privacy

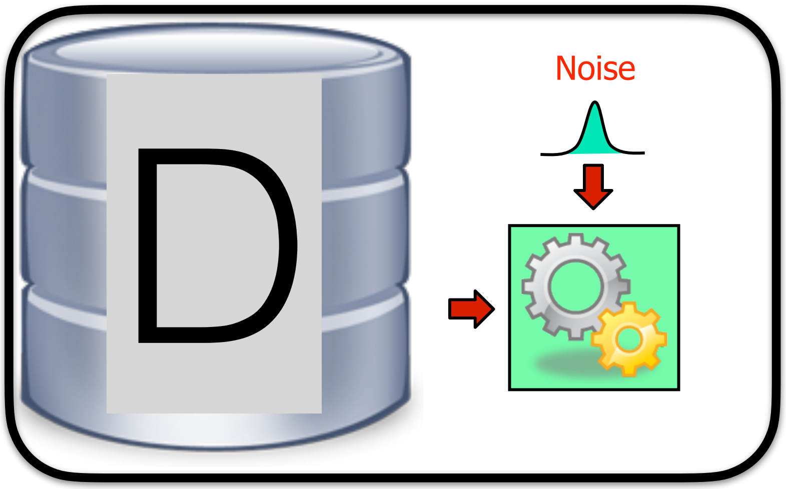
Question: Why is group privacy important?

Group Privacy

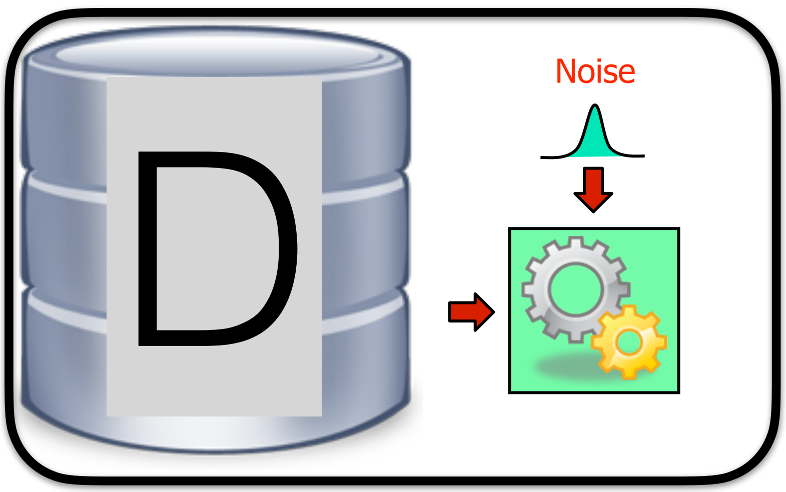
Question: Why is group privacy important?

Answer: Because it allows to reason about privacy at different level of granularities!

Composition

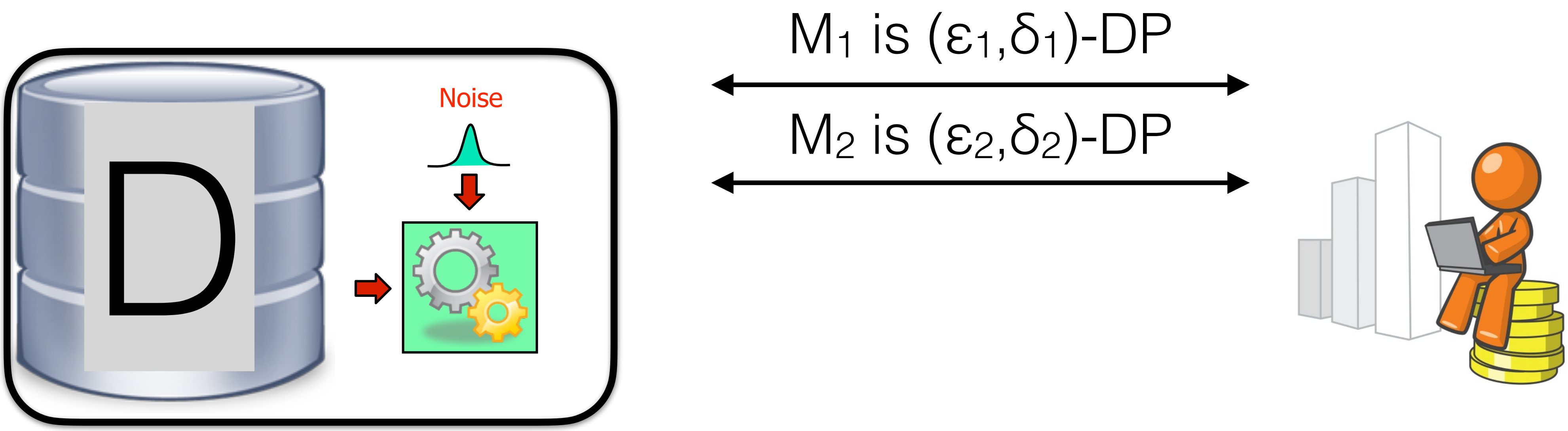


Composition

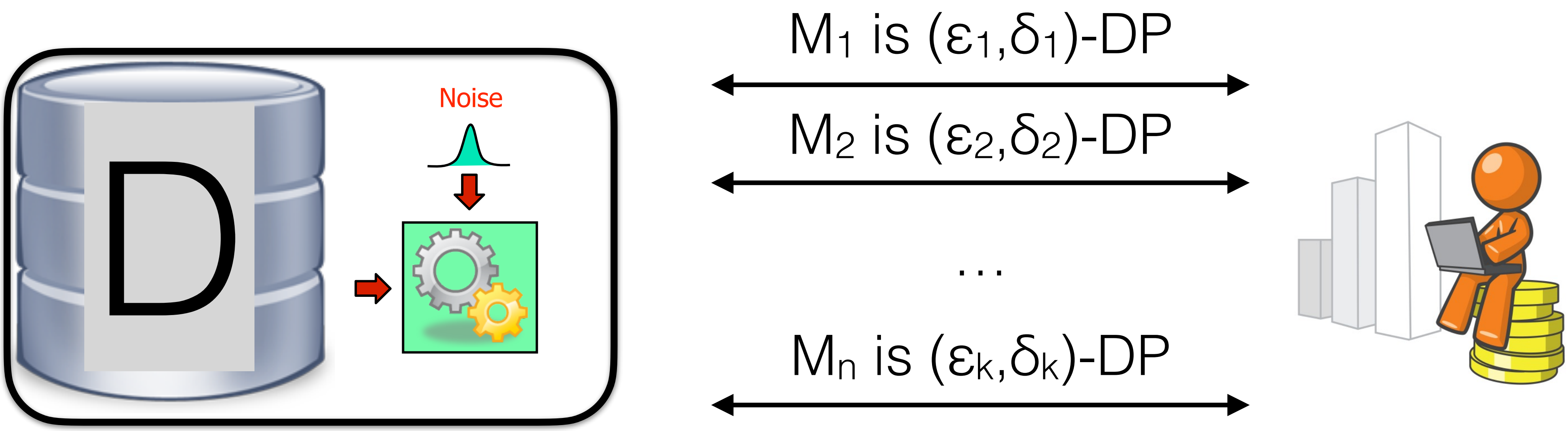


M_1 is (ε_1,δ_1)-DP

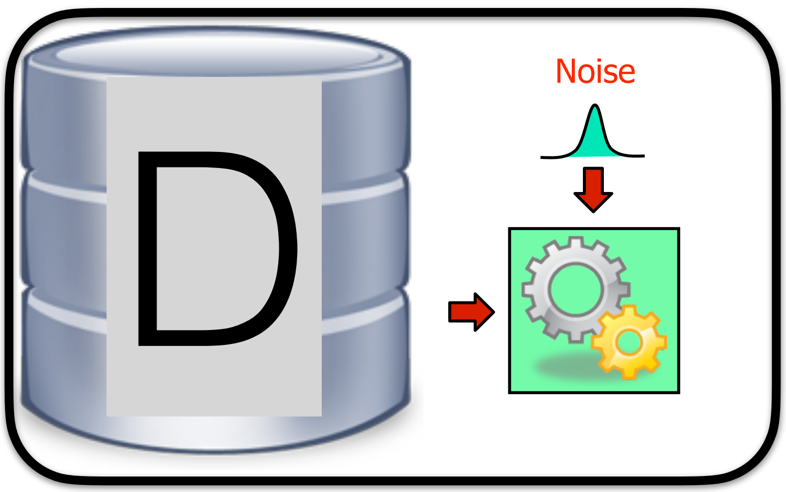
Composition



Composition



Composition



M_1 is (ε_1,δ_1)-DP

M_2 is (ε_2,δ_2)-DP

…

M_n is (ε_k,δ_k)-DP

The overall process is $(\varepsilon_1+\varepsilon_2+\ldots+\varepsilon_k,\delta_1+\delta_2+\ldots+\delta_k)$-DP

Composition

Let M_1:DB $\rightarrow R_1$ be a (ε_1,δ_1)-differentially private program and M_2:DB $\rightarrow R_2$ be a (ε_2,δ_1)-differentially private program. Then, their composition $M_{1,2}$:DB$\rightarrow R_1 x R_2$ defined as

$$M_{1,2}(D)=(M_1(D),M_2(D))$$

is $(\varepsilon_1+\varepsilon_2,\delta_1+\delta_2)$-differentially private.

Composition

Question: Why composition is important?

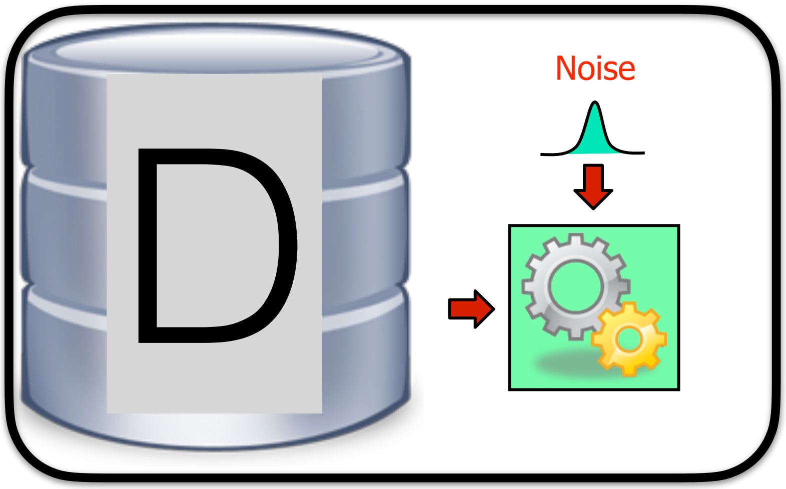
Composition

Question: Why composition is important?

Answer: Because it allows to reason about privacy as a budget!

Composition

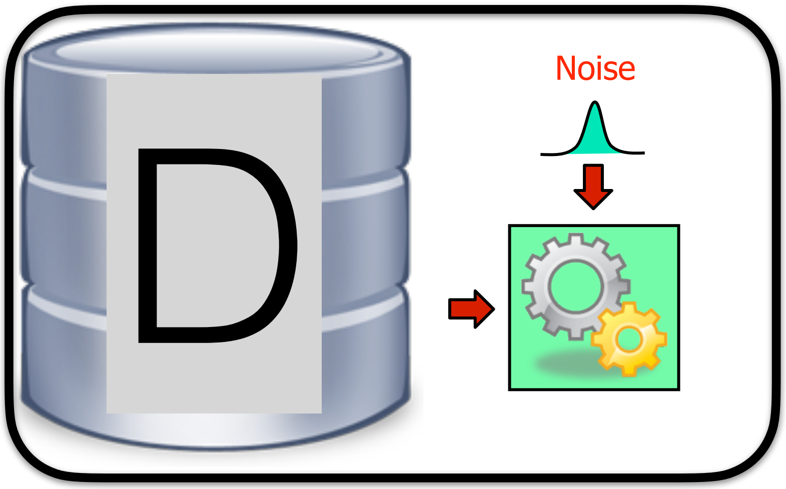
Budget=ε_{global}



Composition

Budget=ε_{global}

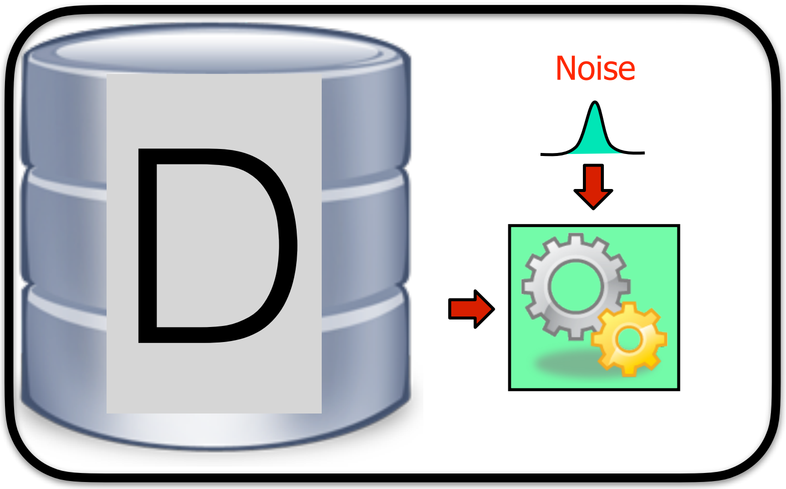
M_1 is ε_1-DP



Composition

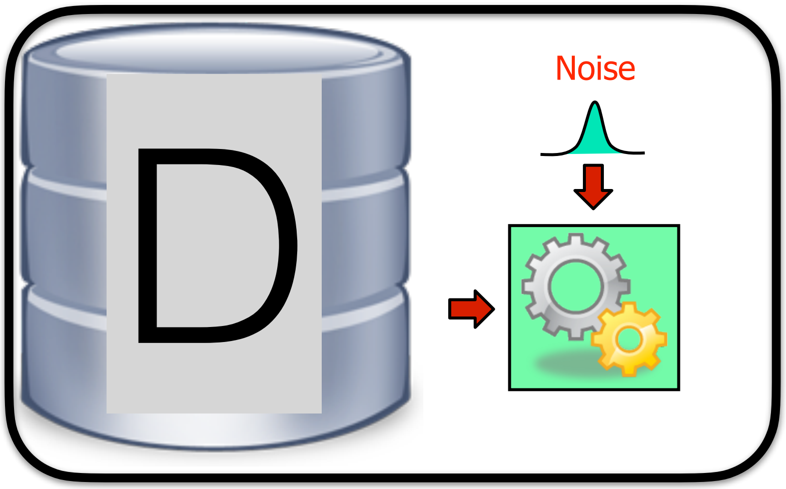
Budget=ε_{global} - ε_1

M_1 is ε_1-DP



Composition

Budget=ε_{global} - ε_1

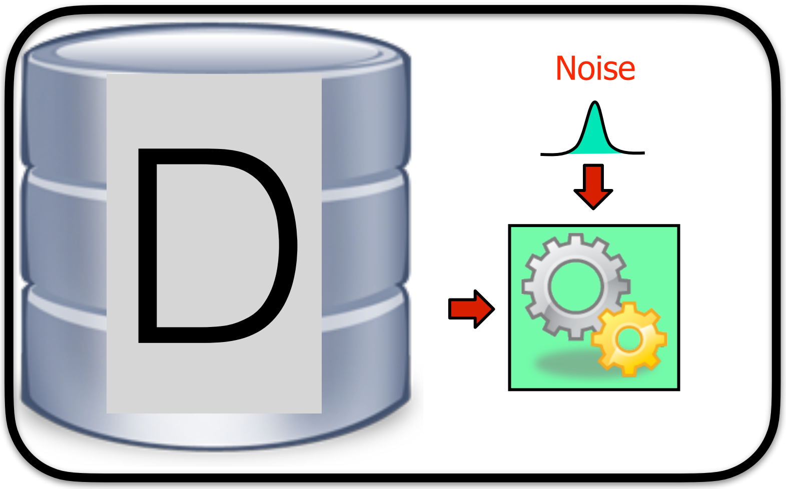


M_1 is ε_1-DP

M_2 is ε_2-DP

Composition

Budget=ε_{global} - ε_1 - ε_2

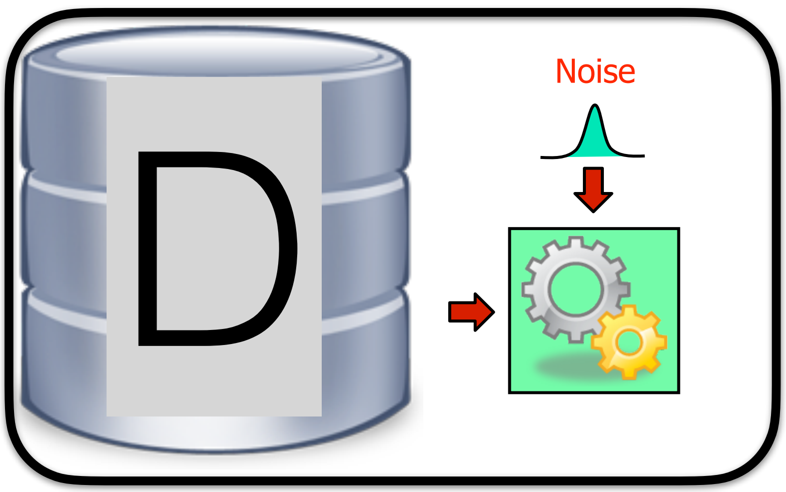


M_1 is ε_1-DP

M_2 is ε_2-DP

Composition

Budget=ε_{global} - ε_1 - ε_2 …



M_1 is ε_1-DP

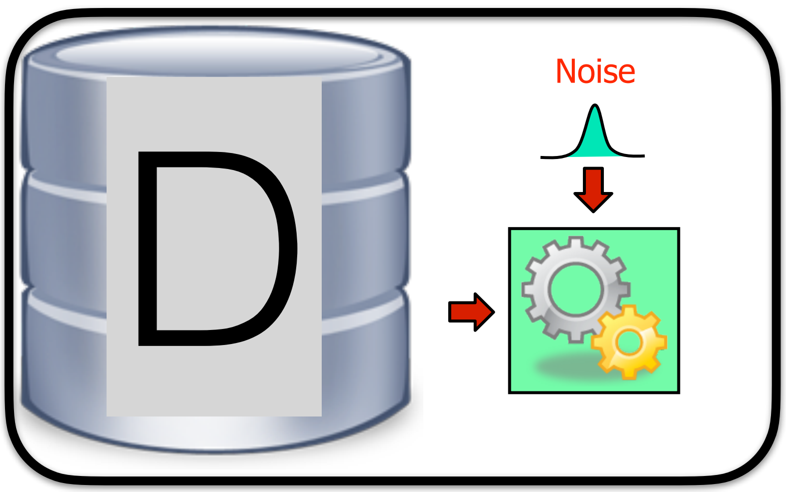
M_2 is ε_2-DP

…

M_n is ε_n-DP

Composition

$$\text{Budget} = \varepsilon_{global} \;\; - \varepsilon_1 \;\; - \varepsilon_2 \;\; \ldots \;\; - \varepsilon_n$$



M_1 is ε_1-DP

M_2 is ε_2-DP

...

M_n is ε_n-DP

CDF

Budget= ε_{global} - ε_1 - ε_2 - ε_3 - ε_4 - ε_5 - ε_6 - ε_7 - ε_8

$X=\{0,1\}^3$ ordered wrt binary encoding.

$D \in X^{10} =$

	D1	D2	D3
I1	0	0	0
I2	1	0	1
I3	0	1	0
I4	1	0	1
I5	0	0	0
I6	0	0	1
I7	1	1	0
I8	0	0	0
I9	0	1	0
I10	1	0	1

$q^*_{000}(D) = .3+L(1/\varepsilon_1)$

$q^*_{001}(D) = .4+L(1/\varepsilon_2)$

$q^*_{010}(D) = .6+L(1/\varepsilon_3)$

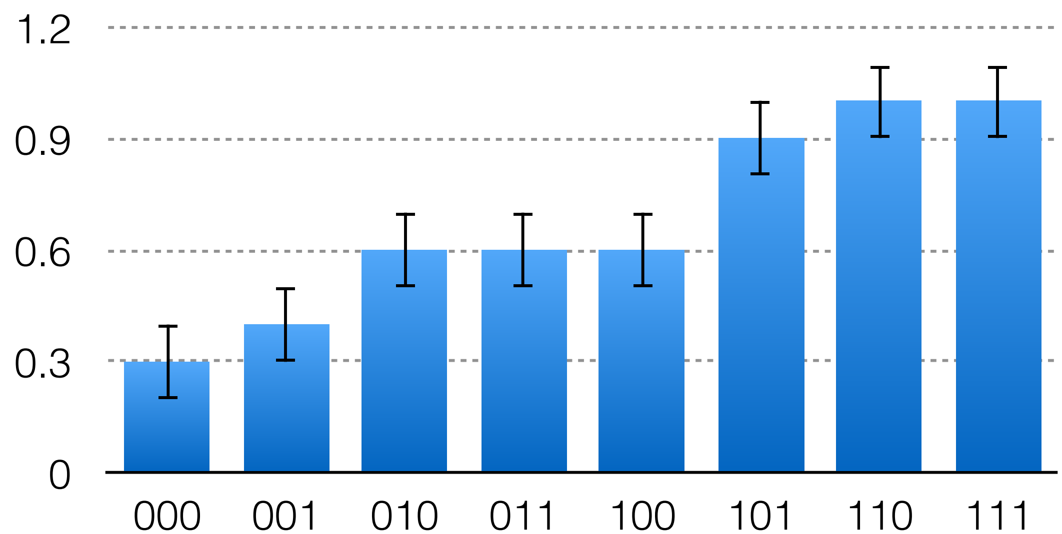
$q^*_{011}(D) = .6+L(1/\varepsilon_4)$

$q^*_{100}(D) = .6+L(1/\varepsilon_5)$

$q^*_{101}(D) = .9+L(1/\varepsilon_6)$

$q^*_{110}(D) = 1+L(1/\varepsilon_7)$

$q^*_{111}(D) = 1+L(1/\varepsilon_8)$



Marginals

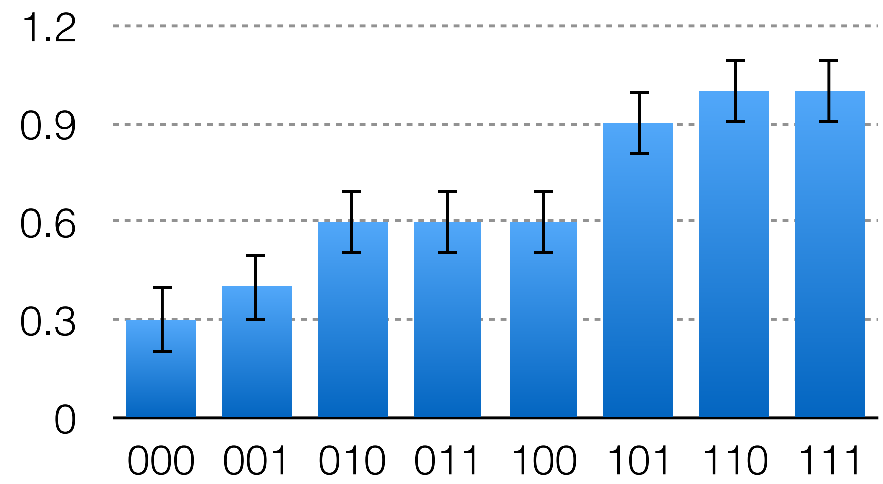
$$\text{Budget} = \varepsilon_{\text{global}} - \varepsilon_1 - \varepsilon_2 - \varepsilon_3$$

$D \in X^{10} =$

	D1	D2	D3
I1	0	0	0
I2	1	0	1
I3	0	1	0
I4	1	0	1
I5	0	0	0
I6	0	0	1
I7	1	1	0
I8	0	0	0
I9	0	1	0
I10	1	0	1
margin	$.4+Y_1$	$.3+Y_2$	$.4+Y_3$

$$q^*_1(D) = .4+L(1/(10^*\varepsilon_1))$$
$$q^*_2(D) = .3+L(1/(10^*\varepsilon_2))$$
$$q^*_3(D) = .4+L(1/(10^*\varepsilon_3))$$

$$\text{Budget}=\varepsilon_{global} - \varepsilon_1 - \varepsilon_2 - \varepsilon_3 - \varepsilon_4 - \varepsilon_5 - \varepsilon_6 - \varepsilon_7 - \varepsilon_8$$



$$\text{Budget}=\varepsilon_{global} - \varepsilon_1 - \varepsilon_2 - \varepsilon_3$$

	D1	D2	D3
I1	0	0	0
I2	1	0	1
I3	0	1	0
I4	1	0	1
I5	0	0	0
I6	0	0	1
I7	1	1	0
I8	0	0	0
I9	0	1	0
I10	1	0	1
margin	$.4+Y_1$	$.3+Y_2$	$.4+Y_3$

Privacy Budget vs Epsilon

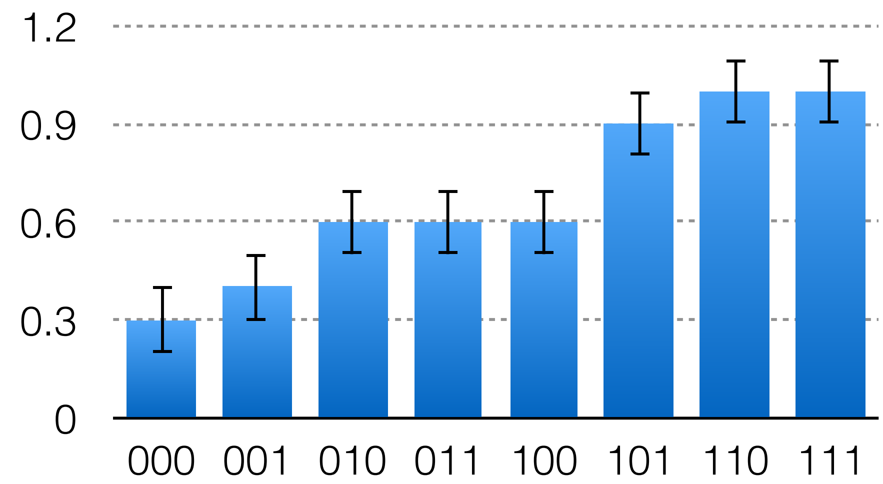
Sometimes is more convenient to think in terms of Privacy Budget: Budget=$\varepsilon_{global} - \sum \varepsilon_{local}$

Sometimes is more convenient to think in terms of epsilon: $\varepsilon_{global} = \sum \varepsilon_{local}$

Also making them uniforms is sometimes more informative.

$$\text{Budget} = \varepsilon_{global} - \varepsilon_1 - \varepsilon_2 - \varepsilon_3 - \varepsilon_4 - \varepsilon_5 - \varepsilon_6 - \varepsilon_7 - \varepsilon_8$$

$$\varepsilon_{global} = \varepsilon+\varepsilon+\varepsilon+\varepsilon+\varepsilon+\varepsilon+\varepsilon+\varepsilon = 8\varepsilon$$



$$\text{Budget} = \varepsilon_{global} - \varepsilon_1 - \varepsilon_2 - \varepsilon_3$$

$$\varepsilon_{global} = \varepsilon+\varepsilon+\varepsilon = 3\varepsilon$$

	D1	D2	D3
I1	0	0	0
I2	1	0	1
I3	0	1	0
I4	1	0	1
I5	0	0	0
I6	0	0	1
I7	1	1	0
I8	0	0	0
I9	0	1	0
I10	1	0	1
margin	$.4+Y_1$	$.3+Y_2$	$.4+Y_3$

Releasing partial sums

```
DummySum(d : {0,1} list) : real list
   i:= 0;
   s:= 0;
   r:= [];
   t:= 0;
   while (i<size d)
      s:= s + d[i]
      z:=$ Laplace(1/eps,0)
      t:= s+z;
      r:= r ++ [t];
      i:= i+1;
   return r
```

Releasing partial sums

```
DummySum(d : {0,1} list) : real list
   i:=0;
   s:=0;
   r:=[];
   t:= 0;
   while (i<size d)
      z:=$ Laplace(1/eps,0)
      t:= d[i] + z
      s:= s + t
      r:= r ++ [s];
      i:=i+1;
   return r
```

Parallel Composition

Let M_1:DB →R be a (ε_1,δ_1)-differentially private program and M_2:DB →R be a (ε_2,δ_2)-differentially private program. Suppose that we partition D in a data-independent way into two datasets D_1 and D_2. Then, the composition $M_{1,2}$:DB→R defined as

$$MP_{1,2}(D)=(M_1(D_1),M_2(D_2))$$

is $(\max(\varepsilon_1,\varepsilon_2),\max(\delta_1,\delta_2))$-differentially private.