CS 511: Formal Methods in Security and Privacy

Differential Privacy

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

(ε,δ)-Differential Privacy

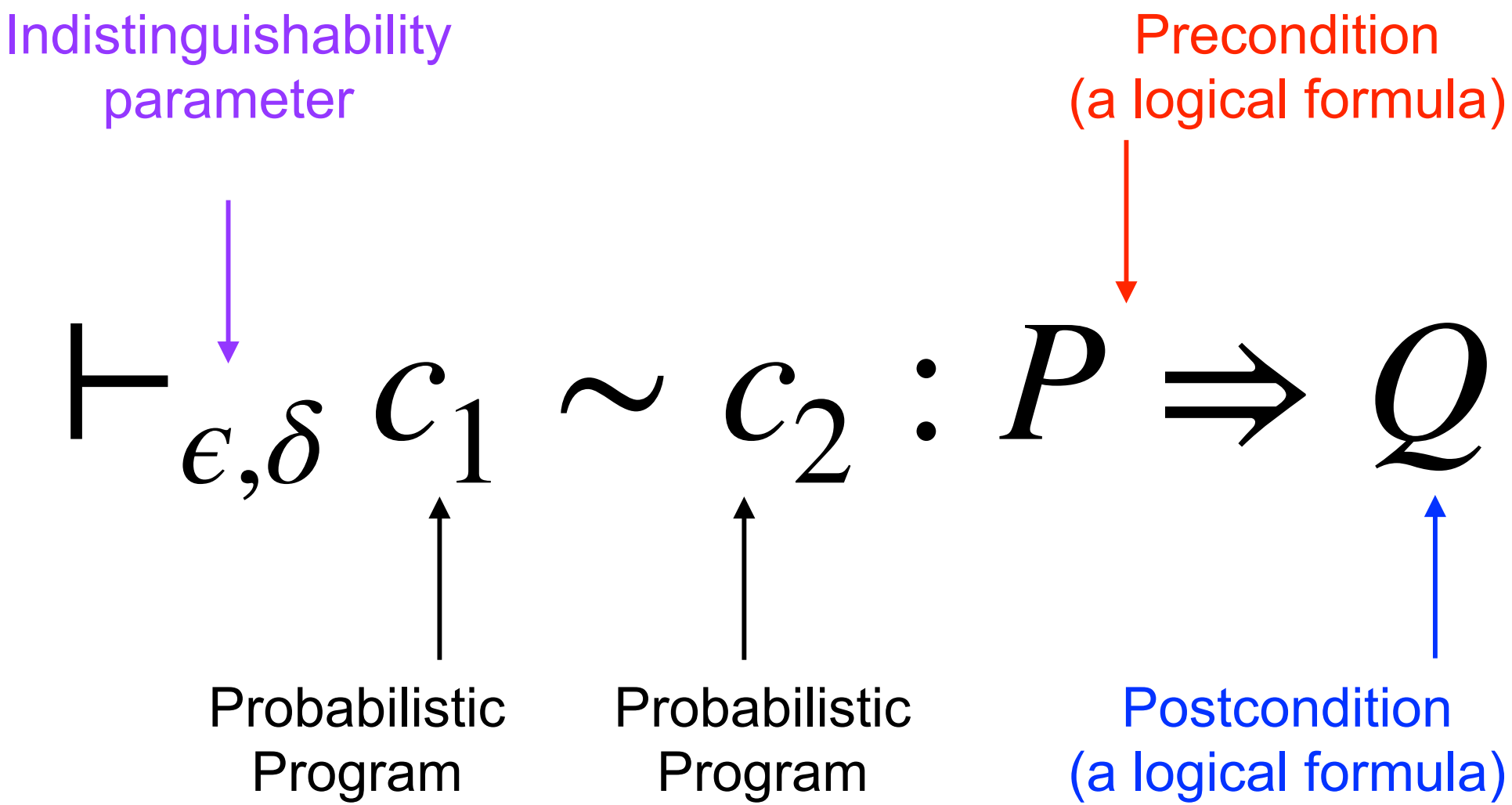
Definition

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 $S \subseteq R$:

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

apRHL


Indistinguishability parameter
Precondition (a logical formula)
$\vdash_{\epsilon,\delta} c_1 \sim c_2 : P \Rightarrow Q$
Probabilistic Program
Probabilistic Program
Postcondition (a logical formula)

Differential Privacy in apRHL

To show differential privacy it is enough to show:

$$\vdash_{\epsilon,\delta} c \sim c : \mathsf{Adj} \Rightarrow \mathsf{out}_1 = \mathsf{out}_2$$

apRHL: More general Lap rule (still restricted)

$$\vdash_{k*\varepsilon,0} \frac{}{x_1 :=\$\ \mathrm{Lap}(1/\varepsilon, y_1) \sim x_2 :=\$\ \mathrm{Lap}(1/\varepsilon, y_2) : |y_1 - y_2| \leq k \Rightarrow =}$$

Releasing privately the average - v2

```
Mean(d : priv [n] age [0,100]) : pub real
   i:=0;
   s:=0;
   while (i<n)
      s:=s + d[i]
      i:=i+1;
   z:=$ Lap(100/(n*eps), (s/n))
   return z
```

Releasing privately two statistics

```
Mean(d : priv age [0,100]) : pub real*real
   i:=0;
   s:=0;
   while (i<size(d))
      s:=s + d[i]
      i:=i+1;
   z:=$ Lap(100/eps, s)
   y:=$ Lap(1/eps, i)
   return (z,y)
```

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.

Probabilistic Relational Hoare Logic

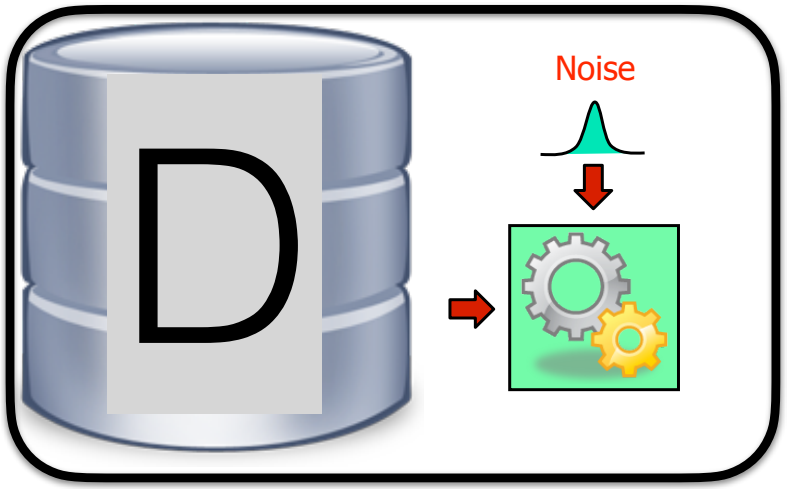
Composition

$$\frac{\vdash_{\varepsilon_1,\delta_1} c_1 \sim c_2 : P \Rightarrow R \qquad \vdash_{\varepsilon_2,\delta_2} c_1' \sim c_2' : R \Rightarrow S}{\vdash_{\varepsilon_1+\varepsilon_2,\delta_1+\delta_2} c_1; c_1' \sim c_2; c_2' : P \Rightarrow S}$$

Releasing partial sums

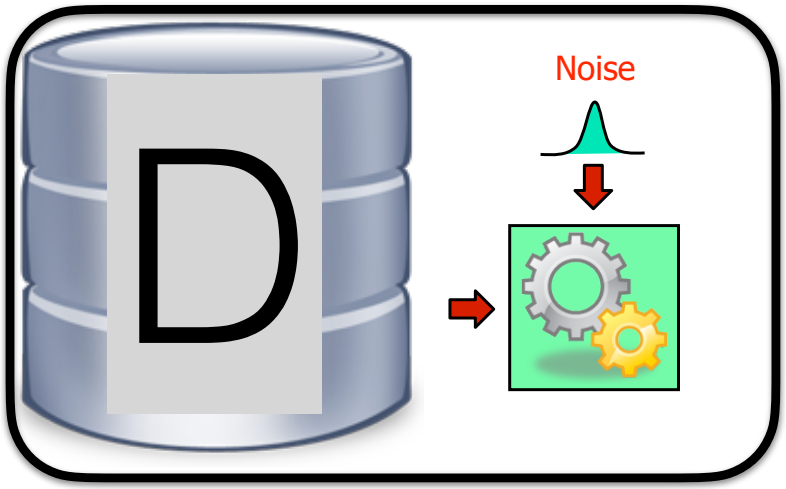
```
DummySum(d : {0,1} list) : real list
   i:= 0;
   s:= 0;
   r:= [];
   while (i<size d)
      s:= s + d[i]
      z:=$ Lap(1/eps,s)
      r:= r ++ [z];
      i:= i+1;
   return r
```

Composition





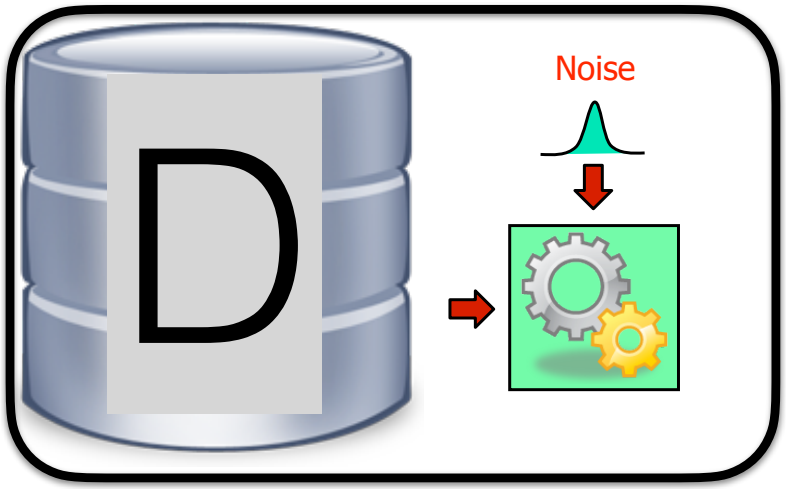
Composition



M_1 is (ε_1,δ_1)-DP



Composition

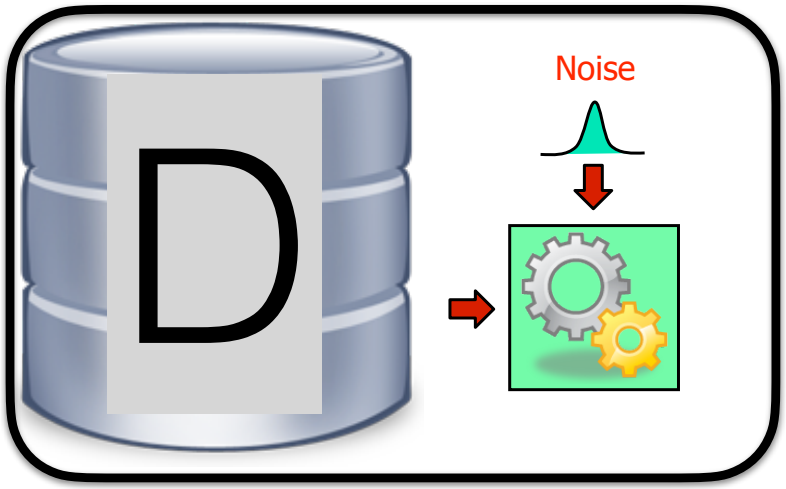


M_1 is (ε_1,δ_1)-DP

M_2 is (ε_2,δ_2)-DP



Composition



M_1 is (ε_1,δ_1)-DP

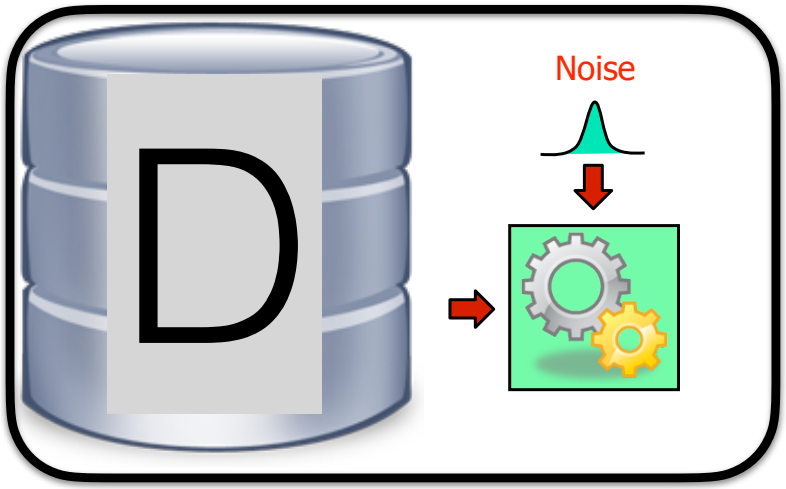
M_2 is (ε_2,δ_2)-DP

…

M_n is (ε_k,δ_k)-DP



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

Question: Why composition is important?

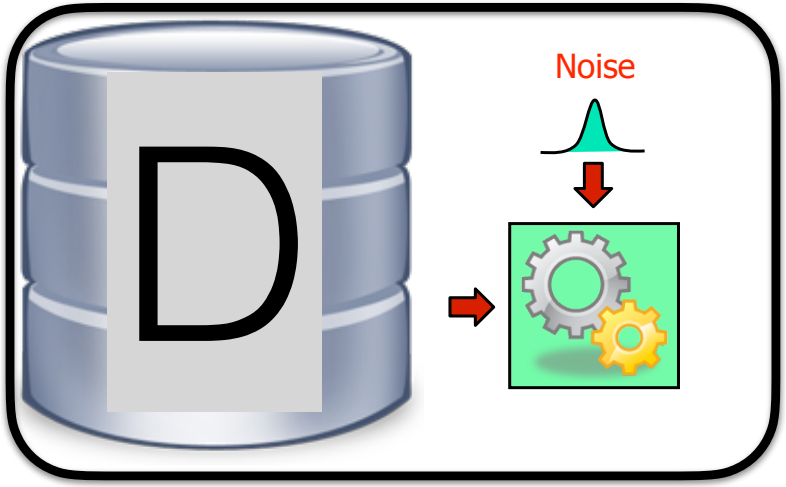
Composition

Question: Why composition is important?

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

Composition

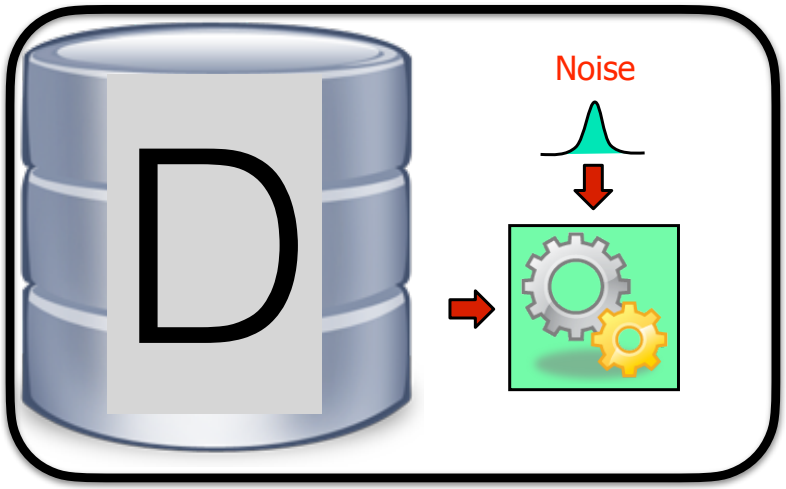
Budget=ε_{global}





Composition

Budget=ε_{global}

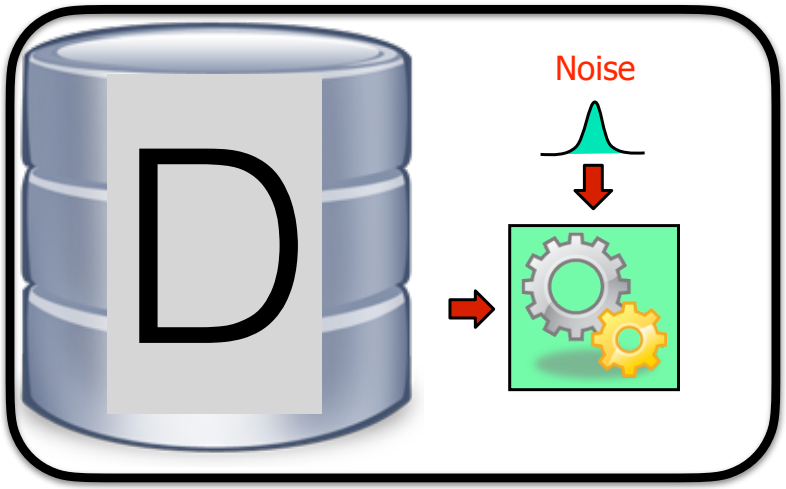


M_1 is ε_1-DP



Composition

Budget=ε_{global} - ε_1

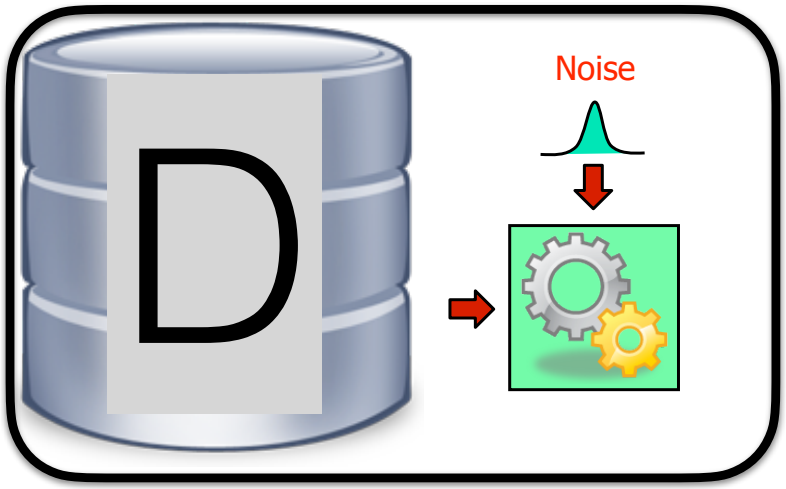


M_1 is ε_1-DP



Composition

Budget=ε_{global} - ε_1



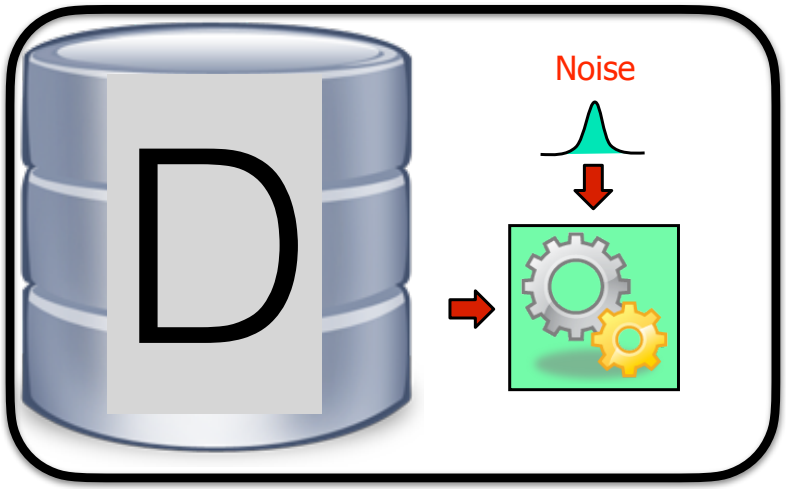
M_1 is ε_1-DP

M_2 is ε_2-DP



Composition

Budget=ε_{global} - ε_1 - ε_2



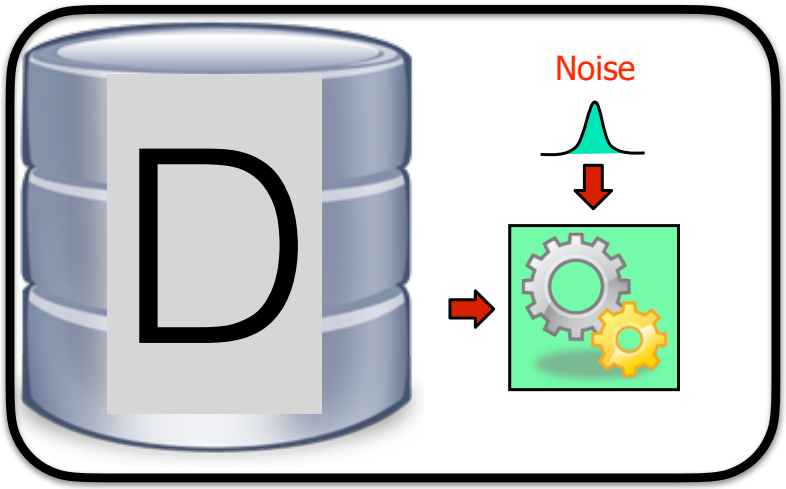
M_1 is ε_1-DP

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

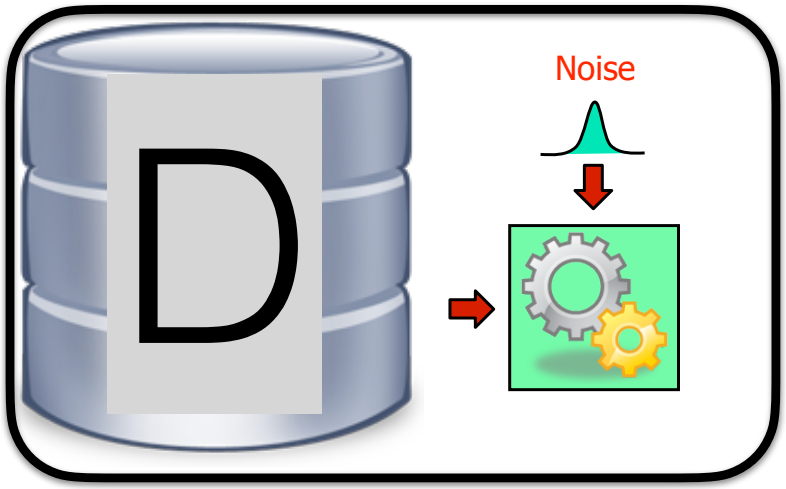
…

M_n is ε_n-DP



Composition

Budget=ε_{global} - ε_1 - ε_2 … - ε_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.

$q^*_{000}(D) = .3+L(1/10*\varepsilon_1)$

$q^*_{001}(D) = .4+L(1/10*\varepsilon_2)$

$q^*_{010}(D) = .6+L(1/10*\varepsilon_3)$

$q^*_{011}(D) = .6+L(1/10*\varepsilon_4)$

$q^*_{100}(D) = .6+L(1/10*\varepsilon_5)$

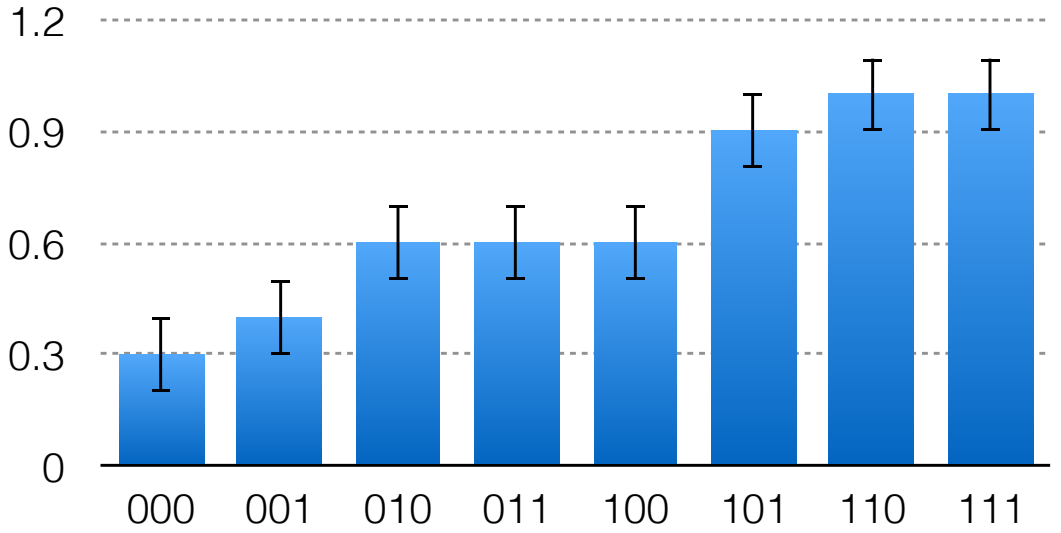
$q^*_{101}(D) = .9+L(1/10*\varepsilon_6)$

$q^*_{110}(D) = 1+L(1/10*\varepsilon_7)$

$q^*_{111}(D) = 1+L(1/10*\varepsilon_8)$

$D \in X^{10} =$

	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



Marginals

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

$D \in X^{10} =$

	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	$.4+Y_1$	$.3+Y_2$	$.4+Y_3$

$q^*_1(D) = .4+L(1/(10^*\varepsilon_1))$

$q^*_2(D) = .3+L(1/(10^*\varepsilon_2))$

$q^*_3(D) = .4+L(1/(10^*\varepsilon_3))$

Parallel Composition

Let M_1:DB $\rightarrow$R be a (ε_1,δ_1)-differentially private program and M_2:DB $\rightarrow$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$\rightarrow$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.

Probabilistic Relational Hoare Logic

Composition

$$\frac{\vdash_{\varepsilon_1,\delta_1} c_1 \sim c_2 : P \Rightarrow R \qquad \vdash_{\varepsilon_2,\delta_2} c_1' \sim c_2' : R \Rightarrow S}{\vdash_{\varepsilon_1+\varepsilon_2,\delta_1+\delta_2} c_1; c_1' \sim c_2; c_2' : P \Rightarrow S}$$

Releasing partial sums

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

apRHL
awhile

$$P/\backslash\ e\langle 1\rangle \leq 0 \Rightarrow \neg b1\langle 1\rangle$$

$$\vdash_{\varepsilon_k,\delta_k} c1 \sim c2 : P/\backslash b1\langle 1\rangle/\backslash b2\langle 2\rangle/\backslash k = e\langle 1\rangle\ /\backslash\ e\langle 1\rangle \leq n \Longrightarrow P\ /\backslash\ b1\langle 1\rangle = b2\langle 2\rangle\ /\backslash k < e\langle 1\rangle$$

$$\vdash_{\sum\varepsilon_k,\sum\delta_k} \text{while } b1 \text{ do } c1 \sim \text{while } b2 \text{ do } c2 : P/\backslash\ b1\langle 1\rangle = b2\langle 2\rangle/\backslash\ e\langle 1\rangle \leq n \Longrightarrow P\ /\backslash \neg b1\langle 1\rangle/\backslash \neg b2\langle 2\rangle$$

apRHL
Consequence

$$\frac{\vdash_{\varepsilon,\delta} c_1 \sim c_2 : S \Rightarrow R \qquad \varepsilon \leq \varepsilon' \quad \delta \leq \delta' \quad P \Rightarrow S \quad R \Rightarrow Q}{\vdash_{\varepsilon',\delta'} c_1 \sim c_2 : P \Rightarrow Q}$$

Properties of Differential Privacy

Some important properties

- Resilience to post-processing
- Group privacy

Some important properties

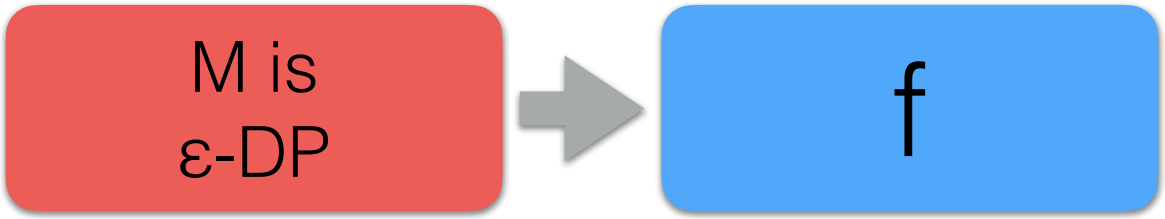
- Resilience to post-processing
- Group privacy

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

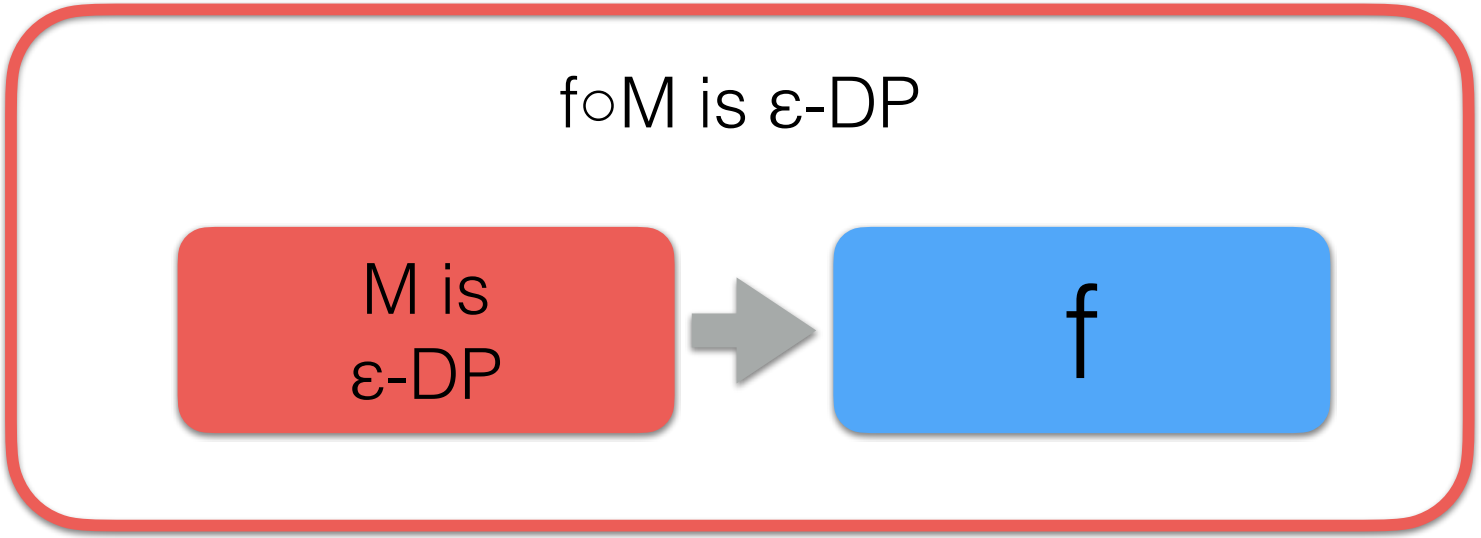
Resilience to Post-processing

M is
ε-DP

Resilience to Post-processing



Resilience to Post-processing



Resilience to Post-processing

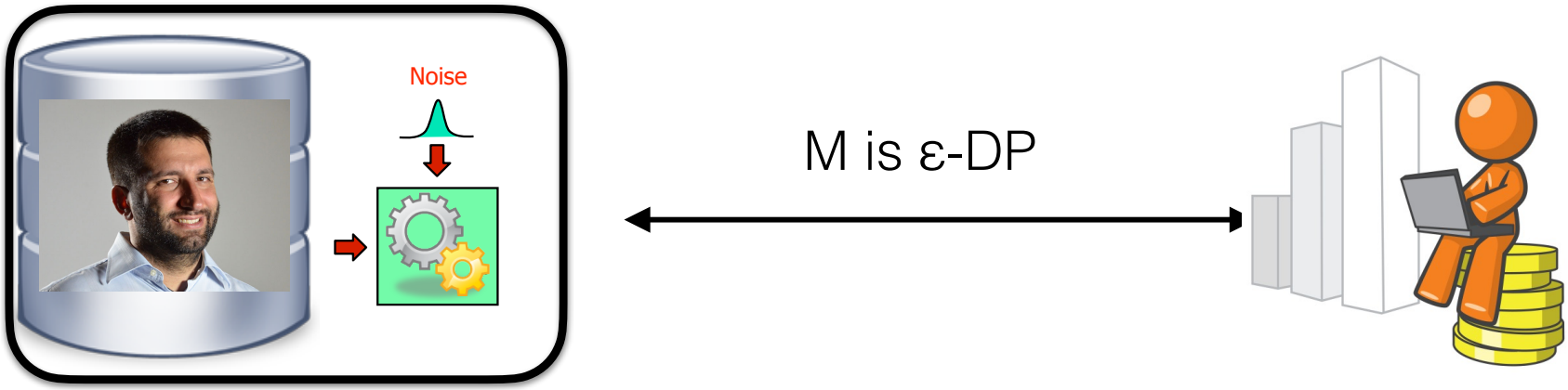
Question: Why is resilience to post-processing important?

Resilience to Post-processing

Question: Why is resilience to post-processing important?

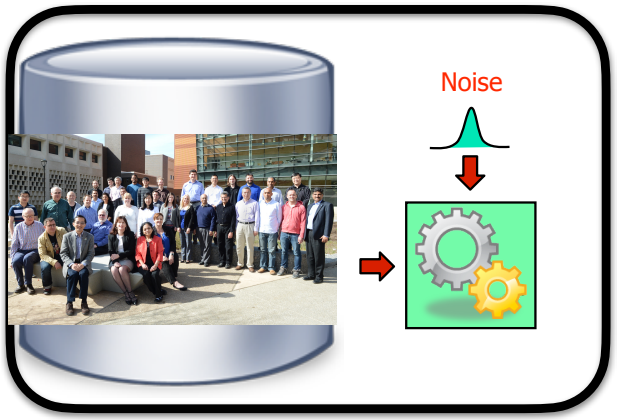
Answer: Because it is what allows us to publicly release the result of a differentially private analysis!

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!