Proving Differential Privacy via Probabilistic Couplings

Gilles Barthe, Marco Gaboardi, Benjamin Grégoire,
Justin Hsu*, Pierre-Yves Strub

IMDEA Software, University at Buffalo, Inria, University of Pennsylvania*

July 8, 2016

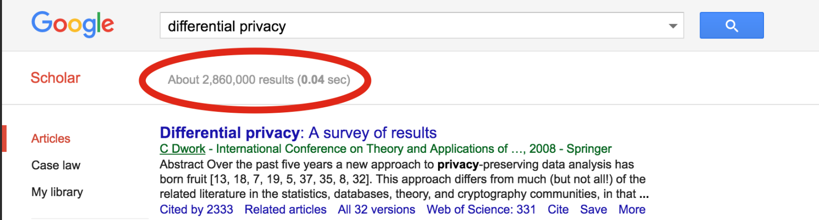
A new approach to formulating privacy goals: the risk to one's privacy, or in general, any type of risk ... should not substantially increase as a result of participating in a statistical database.

This is captured by differential privacy.

— Cynthia Dwork

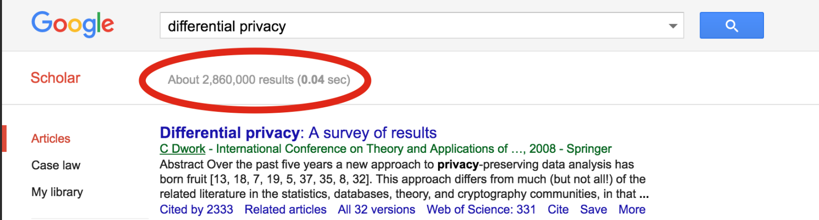
Increasing interest

In research. . .

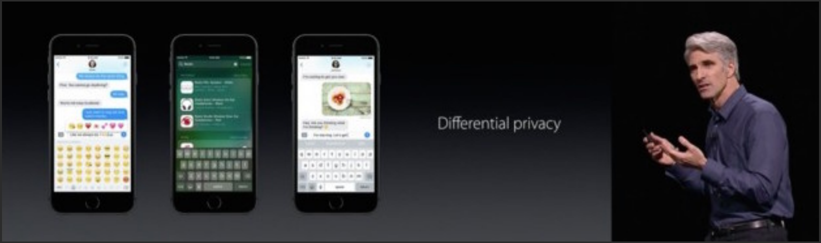


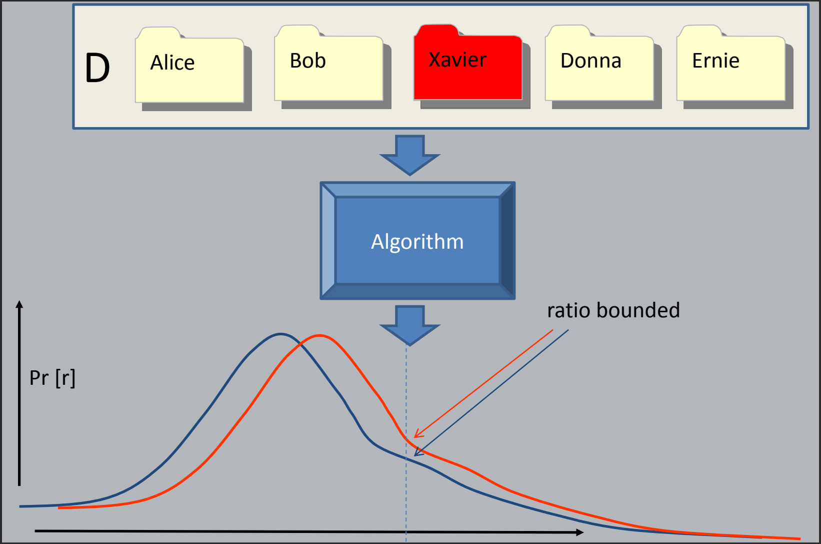
Increasing interest

In research. . .



. . . and in the “real world”



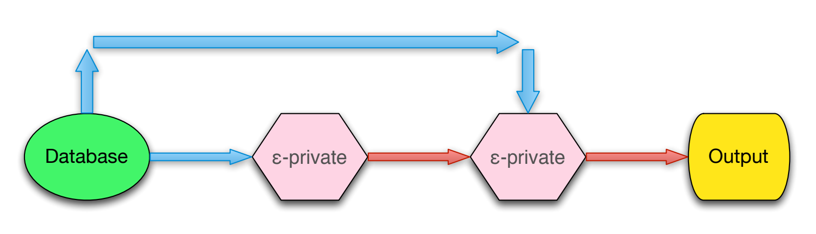

D
Alice
Bob
Xavier
Donna
Ernie
Algorithm
ratio bounded
Pr [r]

Dwork, McSherry, Nissim, and Smith

Let $\epsilon \geq 0$ be a parameter, and suppose there is a binary adjacency relation *Adj* on D. A randomized algorithm $M : D \to \mathbf{Distr}(R)$ is ϵ-differentially private if for every set of outputs $S \subseteq R$ and every pair of adjacent inputs d_1, d_2, we have

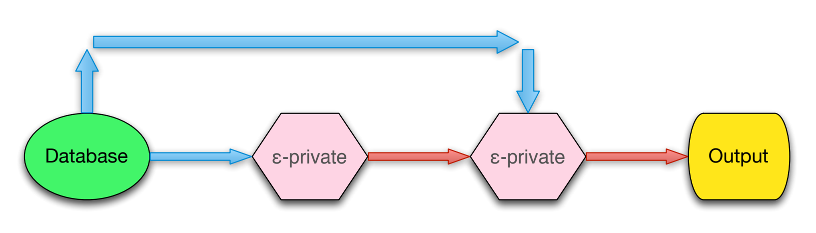
$$\Pr_{x \sim M(d_1)}[x \in S] \leq \exp(\epsilon) \cdot \Pr_{x \sim M(d_2)}[x \in S].$$

Composition properties



Whole program is 2ϵ-private

Composition properties



Whole program is 2ϵ-private

Formally ...

Consider randomized algorithms $M : D \to \mathbf{Distr}(R)$ and $M : R \to D \to \mathbf{Distr}(R')$. If M is ϵ-private and for every $r \in R$, $M'(r)$ is ϵ'-private, then the composition is $(\epsilon + \epsilon')$-private:

$$r \xleftarrow{\$} M(d); res \xleftarrow{\$} M(r, d); \mathrm{return}(res)$$

Differential privacy is a:

relational property of
probabilistic programs.

When privacy follows from composition...

When privacy follows from composition...

(Linear types, refinement types, self products, relational Hoare logics, ...)

When privacy doesn't follow from composition...

When privacy doesn't follow from composition...

How to formally verify?

Use approximate coupling view of privacy to extend the logic apRHL

Combine smaller, pointwise proofs to prove differential privacy in apRHL

Get new, much simpler proofs using coupling composition principle

A crash course: apRHL [BKOZB]

Imperative language with sampling

$$x \xleftarrow{\$} \mathcal{L}_{\epsilon}(e)$$

A crash course: apRHL [BKOZB]

Imperative language with sampling

$$x \xleftarrow{\$} \mathcal{L}_{\epsilon}(e)$$

approximate probabilistic Relational Hoare Logic

$$\vdash \{\Phi\} \; c_1 \sim_{\epsilon} c_2 \; \{\Psi\}$$

A crash course: apRHL [BKOZB]

Imperative language with sampling

$$x \xleftarrow{\$} \mathcal{L}_\epsilon(e)$$

approximate probabilistic Relational Hoare Logic

$$\vdash \{\Phi\} \; c_1 \sim_\epsilon c_2 \; \{\Psi\}$$

Non-probablistic

A crash course: apRHL [BKOZB]

Imperative language with sampling

$$x \xleftarrow{\$} \mathcal{L}_\epsilon(e)$$

approximate probabilistic Relational Hoare Logic

$$\vdash \{\Phi\} \; c_1 \sim_\epsilon c_2 \; \{\Psi\}$$

Numeric index

Approximate liftings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by the ϵ-approximate lifting of R if there exists $\mu_L, \mu_R \in \mathbf{Distr}(A \times A)$ with:

Approximate liftings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by the ϵ-approximate lifting of R if there exists $\mu_L, \mu_R \in \mathbf{Distr}(A \times A)$ with:

- support in R ;

Approximate liftings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by the ϵ-approximate lifting of R if there exists $\mu_L, \mu_R \in \mathbf{Distr}(A \times A)$ with:

- support in R ;
- $\pi_1(\mu_L) = \mu_1$ and $\pi_2(\mu_R) = \mu_2$;

Approximate liftings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon \geq 0$. Two distributions $\mu_1, \mu_2 \in \textbf{Distr}(A)$ are related by the ϵ-approximate lifting of R if there exists $\mu_L, \mu_R \in \textbf{Distr}(A \times A)$ with:

- support in R ;
- $\pi_1(\mu_L) = \mu_1$ and $\pi_2(\mu_R) = \mu_2$;
- for every $S \subseteq A \times A$,

$$\Pr_{z \sim \mu_L}[z \in S] \leq \exp(\epsilon) \cdot \Pr_{z \sim \mu_R}[z \in S]$$

Approximate liftings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by the ϵ-approximate lifting of R if there exists $\mu_L, \mu_R \in \mathbf{Distr}(A \times A)$ with:

- support in R ;
- $\pi_1(\mu_L) = \mu_1$ and $\pi_2(\mu_R) = \mu_2$;
- for every $S \subseteq A \times A$,

$$\Pr_{z \sim \mu_L}[z \in S] \leq \exp(\epsilon) \cdot \Pr_{z \sim \mu_R}[z \in S]$$

Approximate liftings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by the ϵ-approximate lifting of R if there exists $\mu_L, \mu_R \in \mathbf{Distr}(A \times A)$ with:

- support in R ;
- $\pi_1(\mu_L) = \mu_1$ and $\pi_2(\mu_R) = \mu_2$;
- for every $S \subseteq A \times A$,

$$\Pr_{z \sim \mu_L}[z \in S] \leq \exp(\epsilon) \cdot \Pr_{z \sim \mu_R}[z \in S]$$

Write: $\mu_1 \; R^{\sharp\epsilon} \; \mu_2$

Interpreting judgments

$$\vdash \{\Phi\} \;\; c_1 \sim_\epsilon c_2 \;\; \{\Psi\}$$

Interpreting judgments

$$\vdash \{\Phi\} \; c_1 \sim_\epsilon c_2 \; \{\Psi\}$$

Memories related by Φ

Interpreting judgments

$$\vdash \{\Phi\} \;\; c_1 \sim_\epsilon c_2 \;\; \{\Psi\}$$

Memories related by Φ

$$\Downarrow$$

Distributions related by $\Psi^{\sharp\epsilon}$

Differential privacy in apRHL

$$\vdash \{Adj(d_1, d_2)\} \;\; c \sim_\epsilon c \;\; \{res_1 = res_2\}$$

Differential privacy in apRHL

$$\vdash \{Adj(d_1, d_2)\} \;\; c \sim_\epsilon c \;\; \{res_1 = res_2\}$$

Exactly ϵ-differential privacy

Proof system

$$\vdash \{\Psi\{e_1\langle 1\rangle, e_2\langle 2\rangle / x_1\langle 1\rangle, x_2\langle 2\rangle\}\}\ \ x_1 \leftarrow e_1 \sim_0 x_2 \leftarrow e_2\ \ \{\Psi\}[\textsc{Assn}]$$

$$\frac{}{\vdash \{|e_1 - e_2| \leq k\}\ \ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{k\cdot\epsilon} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2)\ \ \{x_1 = x_2\}}[\textsc{Lap}]$$

$$\frac{\vdash \{\Phi\}\ \ c_1 \sim_\epsilon c_2\ \ \{\Psi'\} \qquad \vdash \{\Psi'\}\ \ c_1' \sim_{\epsilon'} c_2'\ \ \{\Psi\}}{\vdash \{\Phi\}\ \ c_1; c_1' \sim_{\epsilon+\epsilon'} c_2; c_2'\ \ \{\Psi\}}[\textsc{Seq}]$$

$$\frac{\vdash \{\Phi \wedge b_1\langle 1\rangle\}\ \ c_1 \sim_\epsilon c_2\ \ \{\Psi\} \qquad \vdash \{\Phi \wedge \neg b_1\langle 1\rangle\}\ \ d_1 \sim_\epsilon d_2\ \ \{\Psi\}}{\vdash \{\Phi \wedge b_1\langle 1\rangle = b_2\langle 2\rangle\}\ \ \text{if } b_1 \text{ then } c_1 \text{ else } d_1 \sim_\epsilon \text{if } b_2 \text{ then } c_2 \text{ else } d_2\ \ \{\Psi\}}[\textsc{Cond}]$$

$$\frac{\begin{array}{c}\Theta \wedge e\langle 1\rangle \leq 0 \Rightarrow \neg b_1\langle 1\rangle \\ \vdash \{\Theta \wedge b_1\langle 1\rangle \wedge b_2\langle 2\rangle \wedge k = e\langle 1\rangle \wedge e\langle 1\rangle \leq n\}\ \ c_1 \sim_{\epsilon_k} c_2\ \ \{\Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge k < e\langle 1\rangle\}\end{array}}{\vdash \{\Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge e\langle 1\rangle \leq n\}\ \ \text{while } b_1 \text{ do } c_1 \sim_{\sum_{k=1}^n \epsilon_k} \text{while } b_2 \text{ do } c_2\ \ \{\Theta \wedge \neg b_1\langle 1\rangle \wedge \neg b_2\langle 2\rangle\}}[\textsc{While}]$$

$$\frac{\vdash \{\Phi'\}\ \ c_1 \sim_{\epsilon'} c_2\ \ \{\Psi'\} \qquad \Phi \Rightarrow \Phi' \qquad \Psi' \Rightarrow \Psi \qquad \epsilon' \leq \epsilon \qquad \delta' \leq \delta}{\vdash \{\Phi\}\ \ c_1 \sim_\epsilon c_2\ \ \{\Psi\}}[\textsc{Conseq}]$$

Proof system

$$\vdash \{\Psi\{e_1\langle 1\rangle, e_2\langle 2\rangle / x_1\langle 1\rangle, x_2\langle 2\rangle\}\}\ \ x_1 \leftarrow e_1 \sim_0 x_2 \leftarrow e_2\ \ \{\Psi\}[\textsc{Assn}]$$

$$\frac{}{\vdash \{|e_1 - e_2| \le k\}\ \ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{k\cdot\epsilon} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2)\ \ \{x_1 = x_2\}}[\textsc{Lap}]$$

$$\frac{\vdash \{\Phi\}\ \ c_1 \sim_\epsilon c_2\ \ \{\Psi'\} \qquad \vdash \{\Psi'\}\ \ c_1' \sim_{\epsilon'} c_2'\ \ \{\Psi\}}{\vdash \{\Phi\}\ \ c_1; c_1' \sim_{\epsilon+\epsilon'} c_2; c_2'\ \ \{\Psi\}}[\textsc{Seq}]$$

$$\frac{\vdash \{\Phi \wedge b_1\langle 1\rangle\}\ \ c_1 \sim_\epsilon c_2\ \ \{\Psi\} \qquad \vdash \{\Phi \wedge \neg b_1\langle 1\rangle\}\ \ d_1 \sim_\epsilon d_2\ \ \{\Psi\}}{\vdash \{\Phi \wedge b_1\langle 1\rangle = b_2\langle 2\rangle\}\ \ \text{if } b_1 \text{ then } c_1 \text{ else } d_1 \sim_\epsilon \text{if } b_2 \text{ then } c_2 \text{ else } d_2\ \ \{\Psi\}}[\textsc{Cond}]$$

$$\frac{\begin{array}{c}\Theta \wedge e\langle 1\rangle \le 0 \Rightarrow \neg b_1\langle 1\rangle \\ \vdash \{\Theta \wedge b_1\langle 1\rangle \wedge b_2\langle 2\rangle \wedge k = e\langle 1\rangle \wedge e\langle 1\rangle \le n\}\ \ c_1 \sim_{\epsilon_k} c_2\ \ \{\Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge k < e\langle 1\rangle\}\end{array}}{\vdash \{\Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge e\langle 1\rangle \le n\}\ \ \text{while } b_1 \text{ do } c_1 \sim_{\sum_{k=1}^{n} \epsilon_k} \text{while } b_2 \text{ do } c_2\ \ \{\Theta \wedge \neg b_1\langle 1\rangle \wedge \neg b_2\langle 2\rangle\}}[\textsc{While}]$$

$$\frac{\vdash \{\Phi'\}\ \ c_1 \sim_{\epsilon'} c_2\ \ \{\Psi'\} \qquad \Phi \Rightarrow \Phi' \qquad \Psi' \Rightarrow \Psi \qquad \epsilon' \le \epsilon \qquad \delta' \le \delta}{\vdash \{\Phi\}\ \ c_1 \sim_\epsilon c_2\ \ \{\Psi\}}[\textsc{Conseq}]$$

(Laplace) Sampling rule

$$\frac{}{\vdash \{|e_1 - e_2| \leq \mathbf{k}\}\ \ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\mathbf{k}\cdot\epsilon} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2)\ \ \{x_1 = x_2\}}\ \textsc{Lap}$$

(Laplace) Sampling rule

$$\frac{}{\vdash \{|e_1 - e_2| \leq \mathbf{k}\}\ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\mathbf{k}\cdot\epsilon} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2)\ \{x_1 = x_2\}}\ \textsc{Lap}$$

"Pay" distance between centers

$\Downarrow$

Assume samples are equal

Sequence rule

$$\frac{\vdash \{\Phi\}\ c_1 \sim_{\epsilon} c_2\ \{\Theta\} \quad \vdash \{\Theta\}\ c_1' \sim_{\epsilon'} c_2'\ \{\Psi\}}{\vdash \{\Phi\}\ c_1; c_1' \sim_{\epsilon+\epsilon'} c_2; c_2'\ \{\Psi\}}\ \textsc{Seq}$$

Generalizes privacy composition

- Θ, Ψ assert equality on outputs

Sequence rule

$$\frac{\vdash \{\Phi\}\ c_1 \sim_{\epsilon} c_2\ \{\Theta\} \qquad \vdash \{\Theta\}\ c_1' \sim_{\epsilon'} c_2'\ \{\Psi\}}{\vdash \{\Phi\}\ c_1; c_1' \sim_{\epsilon+\epsilon'} c_2; c_2'\ \{\Psi\}}\ \textsc{Seq}$$

Generalizes privacy composition

- Θ, Ψ assert equality on outputs

“Costs” sum up

Assume “paid” facts in rest of program

Approximate liftings are approximate versions of probabilistic couplings

Approximate liftings are approximate versions of probabilistic couplings

New liftings $\iff$ New proof rules

New sampling rule: [LapNull]

$$\frac{x_1 \notin FV(e_1), x_2 \notin FV(e_2)}{\vdash \{\top\}\ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\mathbf{0}} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2)\ \{x_1 - x_2 = e_1 - e_2\}}$$

New sampling rule: [LapNull]

$$\frac{x_1 \notin FV(e_1), x_2 \notin FV(e_2)}{\vdash \{\top\}\ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\mathbf{0}} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2)\ \{x_1 - x_2 = e_1 - e_2\}}$$

"Pay" zero cost

$\Downarrow$

Distance between samples

=

Distance between centers

New sampling rule: [LapGen]

$$\frac{x_1 \notin FV(e_1), x_2 \notin FV(e_2)}{\vdash \{|e_1 - (e_2 + s)| \leq k\} \;\; x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\mathbf{k} \cdot \epsilon} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2) \;\; \{x_1 = x_2 + s\}}$$

New sampling rule: [LapGen]

$$\frac{x_1 \notin FV(e_1), x_2 \notin FV(e_2)}{\vdash \{|e_1 - (e_2 + s)| \leq k\} \ \ x_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\mathbf{k} \cdot \epsilon} x_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2) \ \ \{x_1 = x_2 + s\}}$$

"Pay" distance to shift centers

$\Downarrow$

Assume shifted samples

New lifting principle: combining pointwise liftings

$$\frac{\text{for all } v, \quad \vdash \{\Phi\}\ c_1 \sim_\epsilon c_2\ \{(e_1 = v) \to (e_2 = v)\}}{\vdash \{\Phi\}\ c_1 \sim_\epsilon c_2\ \{e_1 = e_2\}}\ \textsc{PW-Eq}$$

New lifting principle: combining pointwise liftings

$$\frac{\text{for all } v, \quad \vdash \{\Phi\}\ c_1 \sim_\epsilon c_2\ \{(e_1 = v) \to (e_2 = v)\}}{\vdash \{\Phi\}\ c_1 \sim_\epsilon c_2\ \{e_1 = e_2\}}\ \textsc{PW-Eq}$$

Separate proofs for each output

New lifting principle: combining pointwise liftings

$$\frac{\text{for all } v, \quad \vdash \{\Phi\}\ c_1 \sim_\epsilon c_2\ \{(e_1 = v) \to (e_2 = v)\}}{\vdash \{\Phi\}\ c_1 \sim_\epsilon c_2\ \{e_1 = e_2\}}\ \text{PW-EQ}$$

Separate proofs for each output

$\Downarrow$

Combine for differential privacy

Logical interpretation

Leibniz equality

$$(\forall v, (e_1 = v) \to (e_2 = v)) \to e_1 = e_2$$

Internalizing a universal quantifier

- Not sound in general
- Sound for certain equality predicates

Logical interpretation

$$\forall \text{ values}, \exists \text{ a lifting such that } \ldots$$

$$\Downarrow$$

$$\exists \text{ a lifting such that } \forall \text{ values}, \ldots$$

Logical interpretation

$$\forall \text{ values}, \exists \text{ a lifting such that } \ldots$$

$$\Downarrow$$

$$\exists \text{ a lifting such that } \forall \text{ values}, \ldots$$

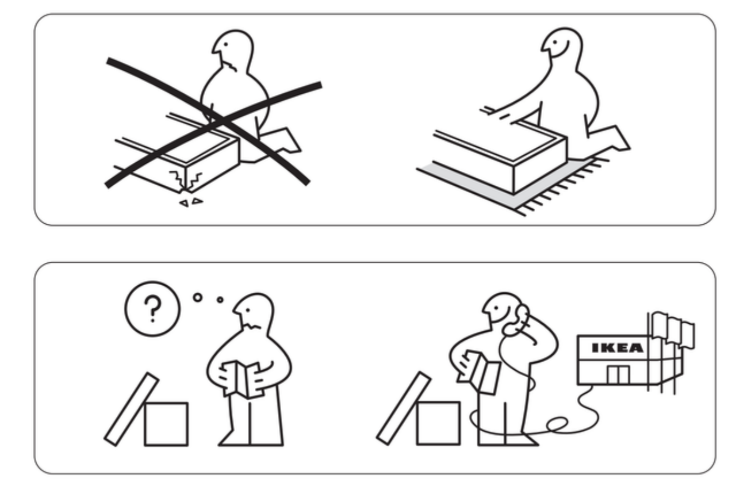
Logical interpretation

$$\forall \text{ values}, \exists \text{ a lifting such that } \ldots$$

$$\Downarrow$$

$$\exists \text{ a lifting such that } \forall \text{ values}, \ldots$$

Putting it all together



Please see the paper!

A brief preview: the Above Threshold algorithm

$$
\begin{array}{l}
AT(t, d) = \{ \\
\quad i \leftarrow 1; x \leftarrow 0; \\
\quad \tilde{t} \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(t); \\
\quad \text{while } i \leq k \text{ do} \\
\quad\quad s \xleftarrow{\$} \mathcal{L}_{\epsilon/4}(q[i](d)); \\
\quad\quad \text{if } (s \geq \tilde{t} \ \wedge x = 0) \text{ then } x \leftarrow i; \\
\quad\quad i \leftarrow i + 1; \\
\quad \text{return } x \\
\}
\end{array}
$$

A brief preview: the Above Threshold algorithm

$$
\begin{array}{l}
AT(t, d) = \{ \\
\quad i \leftarrow 1; x \leftarrow 0; \\
\quad \tilde{t} \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(t); \\
\quad \text{while } i \leq k \text{ do} \\
\quad\quad s \xleftarrow{\$} \mathcal{L}_{\epsilon/4}(q[i](d)); \\
\quad\quad \text{if } (s \geq \tilde{t} \ \wedge x = 0) \text{ then } x \leftarrow i; \\
\quad\quad i \leftarrow i + 1; \\
\quad \text{return } x \\
\}
\end{array}
$$

Standard composition: $AT(t, -)$ is $k\epsilon$-private

A brief preview: the Above Threshold algorithm

$$
\begin{array}{l}
AT(t, d) = \{ \\
\quad i \leftarrow 1; x \leftarrow 0; \\
\quad \tilde{t} \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(t); \\
\quad \text{while } i \leq k \text{ do} \\
\quad\quad s \xleftarrow{\$} \mathcal{L}_{\epsilon/4}(q[i](d)); \\
\quad\quad \text{if } (s \geq \tilde{t} \ \wedge x = 0) \text{ then } x \leftarrow i; \\
\quad\quad i \leftarrow i + 1; \\
\quad \text{return } x \\
\}
\end{array}
$$

~~Standard composition: $AT(t, -)$ is $k\epsilon$-private~~

In fact: $AT(t, -)$ is ϵ-private

Complicated privacy proof(s)

3.1 Privacy Proof for Algorithm 1

We now prove the privacy of Algorithm 1. We break the proof down into two steps, to make the proof easier to understand, and, more importantly, to point out what confusions likely caused the different non-private variants of SVT to be proposed. In the first step, we analyze the situation where the output is $\perp^\ell$, a length-ℓ vector $\langle \perp, \cdots, \perp \rangle$, indicating that all ℓ queries are tested to be below the threshold.

LEMMA 1. *Let $\mathcal{A}$ be Algorithm 1. For any neighboring datasets D and D', and any integer ℓ, we have*

$$\Pr\left[\mathcal{A}(D) = \perp^\ell\right] \le e^{\frac{\epsilon}{2}} \Pr\left[\mathcal{A}(D') = \perp^\ell\right].$$

PROOF. We have

$$\Pr\left[\mathcal{A}(D) = \perp^\ell\right] = \int_{-\infty}^{\infty} f_\perp(D, z, L)\, dz,$$

$$\text{where } f_\perp(D, z, L) = \Pr[\rho = z] \prod_{i \in L} \Pr[q_i(D) + \nu_i < T_i + z], \quad (1)$$

$$\text{and } L = \{1, 2, \cdots, \ell\}.$$

The probability of outputting $\perp^\ell$ over D is the summation (or integral) of terms $f_\perp(D, z, L)$, each of which is the product of $\Pr[\rho = z]$, the probability that the threshold noise equals z, and $\prod_{i \in L} \Pr[q_i(D) + \nu_i < T_i + z]$, the conditional probability that $\perp^\ell$ is the output on D given that the threshold noise ρ is z. (Note that given D, $\mathbf{T}$, the queries, and ρ, whether one query results in $\perp$ or not depends completely on the noise ν_i and is independent from whether any other query results in $\perp$.) If we can prove

$$f_\perp(D, z, L) \le e^{\frac{\epsilon}{2}} f_\perp(D', z + \Delta, L), \quad (2)$$

then we have

$$\begin{aligned}
\Pr\left[\mathcal{A}(D) = \perp^\ell\right] &= \int_{-\infty}^{\infty} f_\perp(D, z, L)\, dz \\
&\le \int_{-\infty}^{\infty} e^{\frac{\epsilon}{2}} f_\perp(D', z + \Delta, L)\, dz && \text{from (2)} \\
&= e^{\frac{\epsilon}{2}} \int_{-\infty}^{\infty} f_\perp(D', z', L)\, dz' && \text{let } z' = z + \Delta \\
&= e^{\frac{\epsilon}{2}} \Pr\left[\mathcal{A}(D') = \perp^\ell\right].
\end{aligned}$$

This proves the lemma. It remains to prove Eq (2). For any query q_i, because $|q_i(D) - q_i(D')| \le \Delta$ and thus $-q_i(D) \le \Delta - q_i(D')$, we have

$$\begin{aligned}
\Pr[q_i(D) + \nu_i < T_i + z] &= \Pr[\nu_i < T_i - q_i(D) + z] \\
&\le \Pr\left[\nu_i < T_i + \Delta - q_i(D') + z\right] \\
&= \Pr\left[q_i(D') + \nu_i < T_i + (z + \Delta)\right] \quad (3)
\end{aligned}$$

With (3), we prove (2) as follows:

$$\begin{aligned}
f_\perp(D, z, L) &= \Pr[\rho = z] \prod_{i \in L} \Pr[q_i(D) + \nu_i < T_i + z] \\
&\le e^{\frac{\epsilon}{2}} \Pr[\rho = z + \Delta] \prod_{i \in L} \Pr\left[q_i(D') + \nu_i < T_i + (z + \Delta)\right] \\
&= e^{\frac{\epsilon}{2}} f_\perp(D', z + \Delta, L).
\end{aligned}$$

□

That is, by using a noisy threshold, we are able to bound the probability ratio for all the negative query answers (i.e., $\perp$'s) by $e^{\frac{\epsilon}{2}}$, no matter how many negative answers there are.

We can obtain a similar result for positive query answers in the same way.

$$\text{Let } f_\top(D, z, L) = \Pr[\rho = z] \prod_{i \in L} \Pr[q_i(D) + \nu_i \ge T_i + z].$$

$$\text{We have } f_\top(D, z, L) \le e^{\frac{\epsilon}{2}} f_\top(D', z - \Delta, L), \text{ and thus}$$

$$\Pr\left[\mathcal{A}(D) = \top^\ell\right] \le e^{\frac{\epsilon}{2}} \Pr\left[\mathcal{A}(D') = \top^\ell\right].$$

This likely contributes to the misunderstandings behind Algorithms 5 and 6, which treat positive and negative answers exactly the same way. The problem is that while one is free to choose to bound positive or negative side, one cannot bound both.

We also observe that the proof of Lemma 1 will go through if no noise is added to the query answers, i.e., $\nu_i = 0$, because Eq (3) holds even when $\nu_i = 0$. It is likely because of this observation that Algorithm 5 adds no noise to query answers. However, when considering outcomes that include both positive answers ($\top$'s) and negative answers ($\perp$'s), one has to add noises to the query answers, as we show below.

THEOREM 2. *Algorithm 1 is ϵ-DP.*

PROOF. Consider any output vector $\boldsymbol{a} \in \{\perp, \top\}^\ell$. Let $\boldsymbol{a} = \langle a_1, \cdots, a_\ell \rangle$, $\mathbf{I}_\top^{\boldsymbol{a}} = \{i : a_i = \top\}$, and $\mathbf{I}_\perp^{\boldsymbol{a}} = \{i : a_i = \perp\}$. Clearly, $|\mathbf{I}_\top^{\boldsymbol{a}}| \le c$. We have

$$\Pr[\mathcal{A}(D) = \boldsymbol{a}] = \int_{-\infty}^{\infty} g(D, z)\, dz, \text{ where}$$

$$g(D, z) = \Pr[\rho = z] \prod_{i \in \mathbf{I}_\perp^{\boldsymbol{a}}} \Pr[q_i(D) + \nu_i < T_i + z] \prod_{i \in \mathbf{I}_\top^{\boldsymbol{a}}} \Pr[q_i(D) + \nu_i \ge T_i + z].$$

We want to show that $g(D, z) \le e^\epsilon g(D', z + \Delta)$. This suffices to prove that $\Pr[\mathcal{A}(D) = a] \le e^\epsilon \Pr[\mathcal{A}(D') = \boldsymbol{a}]$. Note that $g(D, z)$ can be written as:

$$g(D, z) = f_\perp(D, z, \mathbf{I}_\perp^{\boldsymbol{a}}) \prod_{i \in \mathbf{I}_\top^{\boldsymbol{a}}} \Pr[q_i(D) + \nu_i \ge T_i + z].$$

Following the proof of Lemma 1, we can show that $f_\perp(D, z, \mathbf{I}_\perp^{\boldsymbol{a}}) \le e^{\frac{\epsilon}{2}} f_\perp(D', z + \Delta, \mathbf{I}_\perp^{\boldsymbol{a}})$, and it remains to show

$$\prod_{i \in \mathbf{I}_\top^{\boldsymbol{a}}} \Pr[q_i(D) + \nu_i \ge T_i + z] \le e^{\frac{\epsilon}{2}} \prod_{i \in \mathbf{I}_\top^{\boldsymbol{a}}} \Pr[q_i(D') + \nu_i \ge T_i + z + \Delta]. \quad (4)$$

Because $\nu_i = \mathsf{Lap}\left(\frac{4c\Delta}{\epsilon}\right)$ and $|q_i(D) - q_i(D')| \le \Delta$, we have

$$\begin{aligned}
\Pr[q_i(D) + \nu_i \ge T_i + z] &= \Pr[\nu_i \ge T_i + z - q_i(D)] \\
&\le \Pr\left[\nu_i \ge T_i + z - \Delta - q_i(D')\right] \quad (5) \\
&\le e^{\frac{\epsilon}{2c}} \Pr\left[\nu_i \ge T_i + z - \Delta - q_i(D') + 2\Delta\right] \quad (6) \\
&= e^{\frac{\epsilon}{2c}} \Pr\left[q_i(D') + \nu_i \ge T_i + z + \Delta\right].
\end{aligned}$$

Eq (5) is because $-q_i(D) \ge -\Delta - q_i(D')$, and Eq (6) is from the Laplace distribution's property. This proves Eq (4). □

The basic idea of the proof is that when comparing $g(D, z)$ with $g(D', z + \Delta)$, we can bound the probability ratio for all outputs of $\perp$ to no more than $e^{\frac{\epsilon}{2}}$ by using a noisy threshold, no matter how many such outputs there are. To bound the ratio for the $\top$ outputs to no more than $e^{\frac{\epsilon}{2}}$, we need to add sufficient Laplacian noises, which should scale with c, the number of positive outputs.

Now we turn to Algorithms 3-6 to clarify what are wrong with their privacy proofs and to give their DP properties.

— Lyu, Su, Dong

Many slightly different versions

Figure 1: A Selection of SVT Variants

Input/Output shared by all SVT Algorithms
Input: A private database D, a stream of queries $Q = q_1, q_2, \cdots$ each with sensitivity no more than Δ, either a sequence of thresholds $\mathbf{T} = T_1, T_2, \cdots$ or a single threshold T (see footnote *), and c, the maximum number of queries to be answered with $\top$.
Output: A stream of answers $a_1, a_2, \cdots$, where each $a_i \in \{\top, \bot\} \cup \mathbb{R}$ and $\mathbb{R}$ denotes the set of all real numbers.

Algorithm 1 An instantiation of the SVT proposed in this paper.

```
Input: D, Q, Δ, T = T_1, T_2, ···, c.
ρ = Lap(2Δ/ε), count = 0
for each query q_i ∈ Q do
  ν_i = Lap(4cΔ/ε)
  if q_i(D) + ν_i ≥ T_i + ρ then
    Output a_i = ⊤
    count = count + 1, Abort if count ≥ c.
  else
    Output a_i = ⊥
  end if
end for
```

Algorithm 2 SVT in Dwork and Roth 2014 [8].

```
Input: D, Q, Δ, T, c.
ρ = Lap(2cΔ/ε), count = 0
for each query q_i ∈ Q do
  ν_i = Lap(4cΔ/ε)
  if q_i(D) + ν_i ≥ T + ρ then
    Output a_i = ⊤, ρ = Lap(2cΔ/ε)
    count = count + 1, Abort if count ≥ c.
  else
    Output a_i = ⊥
  end if
end for
```

Algorithm 3 SVT in Roth's 2011 Lecture Notes [15].

```
Input: D, Q, Δ, T, c.
ρ = Lap(2Δ/ε), count = 0
for each query q_i ∈ Q do
  ν_i = Lap(2cΔ/ε)
  if q_i(D) + ν_i ≥ T + ρ then
    Output a_i = q_i(D) + ν_i
    count = count + 1, Abort if count ≥ c.
  else
    Output a_i = ⊥
  end if
end for
```

Algorithm 4 SVT in Lee and Clifton 2014 [13].

```
Input: D, Q, Δ, T, c.
ρ = Lap(4Δ/ε), count = 0
for each query q_i ∈ Q do
  ν_i = Lap(4Δ/3ε)
  if q_i(D) + ν_i ≥ T + ρ then
    Output a_i = ⊤
    count = count + 1, Abort if count ≥ c.
  else
    Output a_i = ⊥
  end if
end for
```

Algorithm 5 SVT in Stoddard et al. 2014 [18].

```
Input: D, Q, Δ, T.
ρ = Lap(2Δ/ε)
for each query q_i ∈ Q do
  ν_i = 0
  if q_i(D) + ν_i ≥ T + ρ then
    Output a_i = ⊤

  else
    Output a_i = ⊥
  end if
end for
```

Algorithm 6 SVT in Chen et al. 2015 [1].

```
Input: D, Q, Δ, T = T_1, T_2, ···.
ρ = Lap(2Δ/ε)
for each query q_i ∈ Q do
  ν_i = Lap(2Δ/ε)
  if q_i(D) + ν_i ≥ T_i + ρ then
    Output a_i = ⊤

  else
    Output a_i = ⊥
  end if
end for
```

	Alg. 1	Alg. 2	Alg. 3	Alg. 4	Alg. 5	Alg. 6
Scale of threshold noise ρ	$2\Delta/\epsilon$	$2c\Delta/\epsilon$	$2\Delta/\epsilon$	$4\Delta/\epsilon$	$2\Delta/\epsilon$	$2\Delta/\epsilon$
Reset ρ after each output of $\top$	No	Yes	No	No	No	No
Scale of query noise ν_i	$4c\Delta/\epsilon$	$4c\Delta/\epsilon$	$2c\Delta/\epsilon$	$4\Delta/3\epsilon$	0	$2\Delta/\epsilon$
Outputting $q_i + \nu_i$ instead of $\top$	No	No	Yes	No	No	No
Stop after outputting c $\top$'s	Yes	Yes	Yes	Yes	No	No
Privacy Property	ϵ-DP	ϵ-DP	∞-DP **	$(\frac{1+6c}{4}\epsilon)$-DP	∞-DP	∞-DP

Figure 2: Differences among Algorithms 1-6.

— Lyu, Su, Dong

Use approximate coupling view of privacy to extend the logic apRHL

Combine smaller, pointwise proofs to prove differential privacy in apRHL

Get new, much simpler proofs using coupling composition principle

Use approximate coupling view of privacy to extend the logic apRHL

Combine smaller, pointwise proofs to prove differential privacy in apRHL

Get new, much simpler proofs using coupling composition principle

(Also, I might be looking for a job . . .)