Advanced Probabilistic Couplings for Differential Privacy

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

October 25, 2016

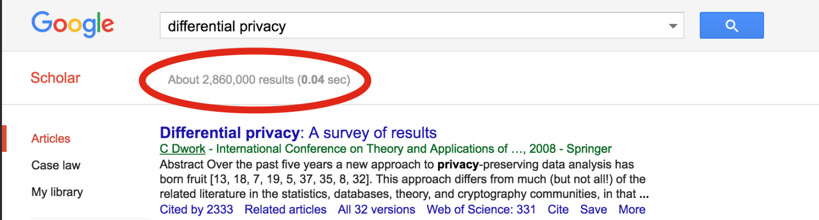
A new approach to formulating privacy goals: the risk to one's privacy, or in general, any type of risk . . . should not substantially increase as a result of participating in a statistical database.

This is captured by differential privacy.

— Cynthia Dwork

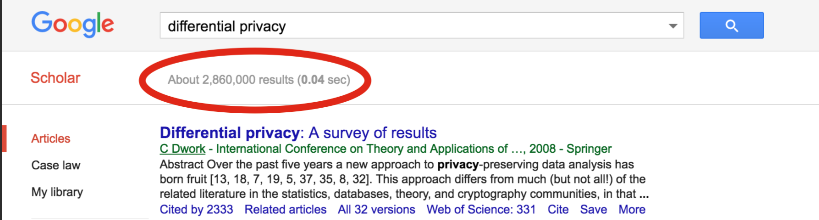
Increasing interest

In research. . .

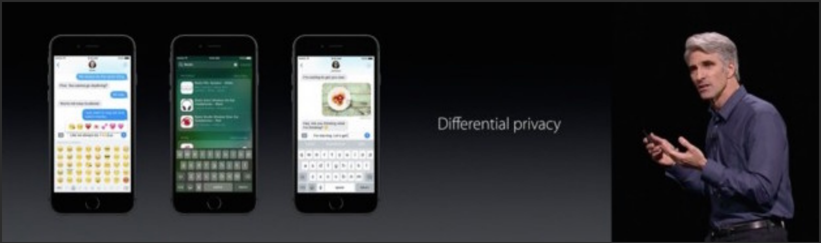


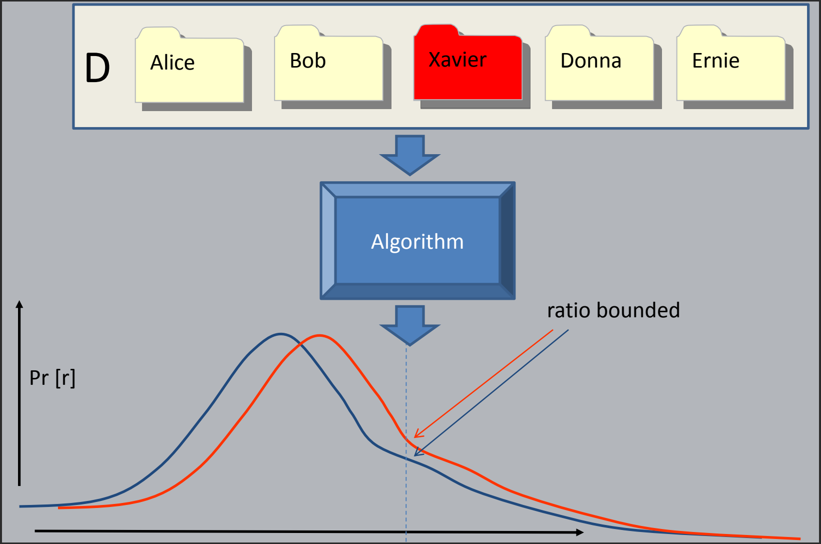
Increasing interest

In research...



...and beyond




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

Dwork, McSherry, Nissim, and Smith

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

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

Dwork, McSherry, Nissim, and Smith

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

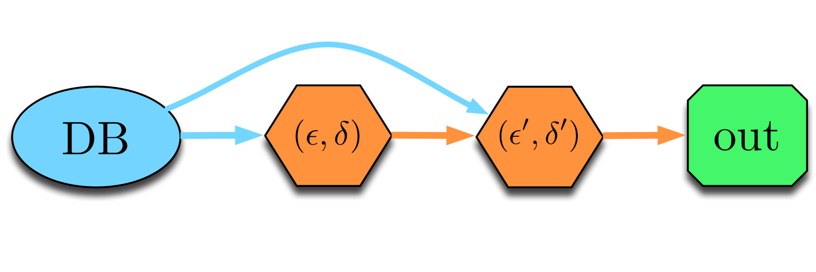
$$\Pr_{x \sim M(d_1)}[x \in S] \leq \exp(\epsilon) \cdot \Pr_{x \sim M(d_2)}[x \in S] + \delta.$$

$\boxed{\text{How to formally verify?}}$

Differential privacy is a:

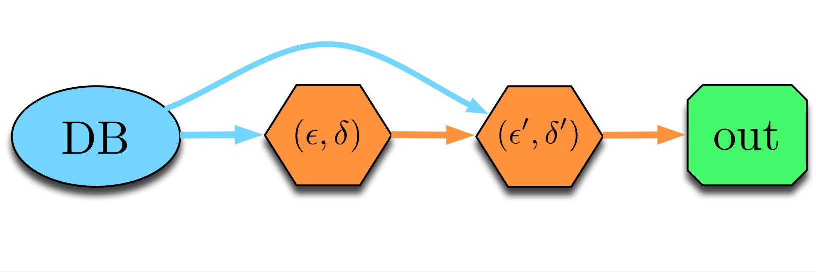
relational property of
probabilistic programs.

Composition properties



Program is $(\epsilon + \epsilon', \delta + \delta')$-private

Composition properties



Program is $(\epsilon + \epsilon', \delta + \delta')$-private

Formally

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

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

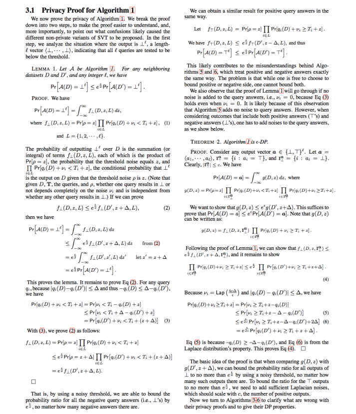
When privacy follows from composition

When privacy follows from composition

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

When privacy doesn't follow from composition

Complicated privacy proofs



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

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

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

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

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

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

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

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

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

□

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.

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

We have $f_\top(D,z,L) \le e^{\frac{\epsilon}{2}} f_\top(D',z-\Delta,L)$, 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 $\mathbf{a}\in\{\perp,\top\}^\ell$. Let $\mathbf{a}=\langle a_1,\cdots,a_\ell\rangle$, $\mathbf{I}^{\mathbf{a}}_\top=\{i : a_i=\top\}$, and $\mathbf{I}^{\mathbf{a}}_\perp=\{i : a_i=\perp\}$. Clearly, $|\mathbf{I}^{\mathbf{a}}_\top|\le c$. We have

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

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

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

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

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

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

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

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

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.

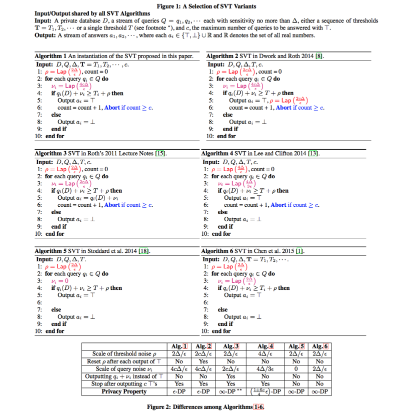


Figure 1: A Selection of SVT Variants

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 $a_i\in\{\top,\perp\}\cup\mathbb{R}$ and $\mathbb{R}$ denotes the set of all real numbers.

Algorithm 1 An instantiation of the SVT proposed in this paper.
Input: $D,Q,\Delta,\mathbf{T}=T_1,T_2,\cdots,c$.
$\rho=\mathsf{Lap}\left(\frac{2\Delta}{\epsilon}\right)$, count $=0$
for each query $q_i\in Q$ **do**
  $\nu_i=\mathsf{Lap}\left(\frac{4c\Delta}{\epsilon}\right)$
  if $q_i(D)+\nu_i\ge T_i+\rho$ **then**
    Output $a_i=\top$
    count = count + 1, **Abort** if count $\ge c$.
  else
    Output $a_i=\perp$
  end if
end for

Algorithm 2 SVT in Dwork and Roth 2014 [8].
Input: D,Q,Δ,T,c.
$\rho=\mathsf{Lap}\left(\frac{2c\Delta}{\epsilon}\right)$, count $=0$
for each query $q_i\in Q$ **do**
  $\nu_i=\mathsf{Lap}\left(\frac{4c\Delta}{\epsilon}\right)$
  if $q_i(D)+\nu_i\ge T+\rho$ **then**
    Output $a_i=\top$, $\rho=\mathsf{Lap}\left(\frac{2c\Delta}{\epsilon}\right)$
    count = count + 1, **Abort** if count $\ge c$.
  else
    Output $a_i=\perp$
  end if
end for

Algorithm 3 SVT in Roth's 2011 Lecture Notes [15].
Input: D,Q,Δ,T,c.
$\rho=\mathsf{Lap}\left(\frac{2\Delta}{\epsilon}\right)$, count $=0$
for each query $q_i\in Q$ **do**
  $\nu_i=\mathsf{Lap}\left(\frac{2c\Delta}{\epsilon}\right)$
  if $q_i(D)+\nu_i\ge T+\rho$ **then**
    Output $a_i=q_i(D)+\nu_i$
    count = count + 1, **Abort** if count $\ge c$.
  else
    Output $a_i=\perp$
  end if
end for

Algorithm 4 SVT in Lee and Clifton 2014 [13].
Input: D,Q,Δ,T,c.
$\rho=\mathsf{Lap}\left(\frac{4\Delta}{\epsilon}\right)$, count $=0$
for each query $q_i\in Q$ **do**
  $\nu_i=\mathsf{Lap}\left(\frac{4\Delta}{3\epsilon}\right)$
  if $q_i(D)+\nu_i\ge T+\rho$ **then**
    Output $a_i=\top$
    count = count + 1, **Abort** if count $\ge c$.
  else
    Output $a_i=\perp$
  end if
end for

Algorithm 5 SVT in Stoddard et al. 2014 [18].
Input: D,Q,Δ,T.
$\rho=\mathsf{Lap}\left(\frac{2\Delta}{\epsilon}\right)$
for each query $q_i\in Q$ **do**
  $\nu_i=0$
  if $q_i(D)+\nu_i\ge T+\rho$ **then**
    Output $a_i=\top$

  else
    Output $a_i=\perp$
  end if
end for

Algorithm 6 SVT in Chen et al. 2015 [1].
Input: $D,Q,\Delta,\mathbf{T}=T_1,T_2,\cdots$.
$\rho=\mathsf{Lap}\left(\frac{2\Delta}{\epsilon}\right)$
for each query $q_i\in Q$ **do**
  $\nu_i=\mathsf{Lap}\left(\frac{2\Delta}{\epsilon}\right)$
  if $q_i(D)+\nu_i\ge T_i+\rho$ **then**
    Output $a_i=\top$

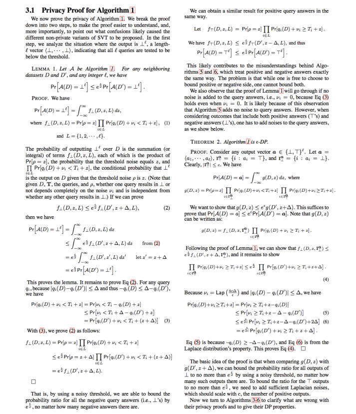
  else
    Output $a_i=\perp$
  end if
end for

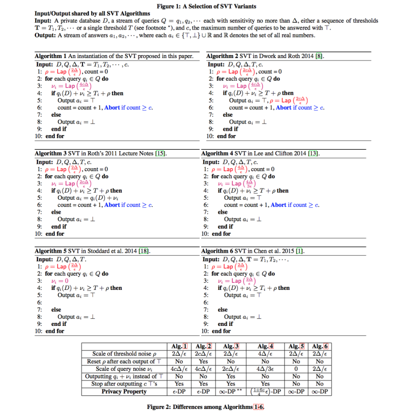
	Alg. 1	Alg. 2	Alg. 3	Alg. 4	Alg. 5	Alg. 6
Scale of threshold noise ρ	$2\Delta/\epsilon$	$2c\Delta/\epsilon$	$2\Delta/\epsilon$	$4\Delta/\epsilon$	$2\Delta/\epsilon$	$2\Delta/\epsilon$
Reset ρ after each output of $\top$	No	Yes	No	No	No	No
Scale of query noise ν_i	$4c\Delta/\epsilon$	$4c\Delta/\epsilon$	$2c\Delta/\epsilon$	$4\Delta/3\epsilon$	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 **	$\left(\frac{1+6c}{4}\epsilon\right)$-DP	∞-DP	∞-DP

Figure 2: Differences among Algorithms 1-6.

— Lyu, Su, Dong

Complicated privacy proofs





— Lyu, Su, Dong

How to verify these proofs?

Recent progress (2016)

Differential privacy $\approx$ Approximate couplings

Recent progress (2016)

Differential privacy $\approx$ Approximate couplings

Approximate couplings $\approx$ Proofs in the logic apRHL

Recent progress (2016)

Differential privacy $\approx$ Approximate couplings

Approximate couplings $\approx$ Proofs in the logic apRHL

Only proofs beyond composition for $(\epsilon, 0)$-privacy

Enhance the logic

New coupling constructions

$\Downarrow$

New proof rules

$\Downarrow$

Richer formal proofs of privacy

Our work: formal privacy proofs with:

Accuracy-dependent privacy

Advanced composition

Adaptive inputs

Our work: formal privacy proofs with:

Accuracy-dependent privacy

Advanced composition

Adaptive inputs

A crash course: the program logic apRHL [BKOZB]

Imperative language with random sampling

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

A crash course: the program logic apRHL [BKOZB]

Imperative language with random sampling

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

approximate probabilistic Relational Hoare Logic

$$\vdash \{\Phi\} \ c_1 \sim_{(\epsilon,\delta)} c_2 \ \{\Psi\}$$

A crash course: the program logic apRHL [BKOZB]

Imperative language with random sampling

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

approximate probabilistic Relational Hoare Logic

$$\vdash \{\Phi\} \quad c_1 \sim_{(\epsilon,\delta)} c_2 \quad \{\Psi\}$$

Non-probablistic, relational $(x_1 = x_2)$

A crash course: the program logic apRHL [BKOZB]

Imperative language with random sampling

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

approximate probabilistic Relational Hoare Logic

$$\vdash \{\Phi\} \; c_1 \sim_{(\epsilon,\delta)} c_2 \; \{\Psi\}$$

Numeric index

Approximate couplings [BKOZB, BO]

Definition

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

Approximate couplings [BKOZB, BO]

Definition

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

- support in R ;

Approximate couplings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon, \delta \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by an (ϵ, δ)-approximate coupling with support 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 couplings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon, \delta \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by an (ϵ, δ)-approximate coupling with support 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 $S \subseteq A \times A$,

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

Approximate couplings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon, \delta \geq 0$. Two distributions $\mu_1, \mu_2 \in \mathbf{Distr}(A)$ are related by an (ϵ, δ)-approximate coupling with support 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 $S \subseteq A \times A$,

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

Approximate couplings [BKOZB, BO]

Definition

Let $R \subseteq A \times A$ be a relation and $\epsilon, \delta \geq 0$. Two distributions $\mu_1, \mu_2 \in \textbf{Distr}(A)$ are related by an (ϵ, δ)-approximate coupling with support 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 $S \subseteq A \times A$,

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

Write: $\mu_1 \quad R^{\sharp}_{(\epsilon,\delta)} \quad \mu_2$

Interpreting judgments

$$\vdash \{\Phi\} \;\; c_1 \sim_{(\epsilon,\delta)} c_2 \;\; \{\Psi\}$$

Interpreting judgments

$$\vdash \{\Phi\} \;\; c_1 \sim_{(\epsilon,\delta)} c_2 \;\; \{\Psi\}$$

Two memories related by Φ

Interpreting judgments

$$\vdash \{\Phi\} \ \ c_1 \sim_{(\epsilon,\delta)} c_2 \ \ \{\Psi\}$$

Two memories related by Φ

$$\Downarrow$$

Two distributions related by $\Psi^{\sharp}_{(\epsilon,\delta)}$

Differential privacy in apRHL

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

Differential privacy in apRHL

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

(ϵ, δ)-differential privacy

Proof rules

$$\boxed{\text{Proof rule} \approx \text{Recipe to combine couplings}}$$

Proof rules

Proof rule ≈ Recipe to combine couplings

Sequence rule ≈ standard composition of privacy

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

Proof rules

Proof rule ≈ Recipe to combine couplings

Sequence rule ≈ standard composition of privacy

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

Our work: formal privacy proofs with:

Accuracy-dependent privacy

Advanced composition

Adaptive inputs

Accuracy-dependent privacy

Accuracy-dependent privacy

Rough intuition

- Think of δ in (ϵ, δ)-privacy as failure probability
- "Algorithm is private except with small probability δ"
- "If the noise added is not too large, then ..."

Similar to up-to-bad reasoning

- Common tool in crypto proofs
- "If bad event doesn't happen, then protocol is safe"

In apRHL: up-to-bad rule

$$\textsc{UtB}\ \frac{\begin{array}{c}\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta)} c_2\ \{\neg\Psi\langle 1\rangle \to x_1 = x_2\} \\ \models m \in \Theta \implies \Pr_{[\![c_1]\!](m_1)}[\Psi\langle 1\rangle] < \delta'\end{array}}{\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta+\delta')} c_2\ \{x_1 = x_2\}}$$

In apRHL: up-to-bad rule

$$\textsc{UtB}\ \frac{\begin{array}{c}\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta)} c_2\ \{\neg\Psi\langle 1\rangle \to x_1 = x_2\} \\ \models m \in \Theta \implies \Pr_{[\![c_1]\!](m_1)}[\Psi\langle 1\rangle] < \delta'\end{array}}{\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta+\delta')} c_2\ \{x_1 = x_2\}}$$

Notes

- $\Psi\langle 1\rangle$ is “bad event”, only mentions c_1

In apRHL: up-to-bad rule

$$\textsc{UtB}\ \frac{\begin{array}{c}\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta)} c_2\ \{\neg\Psi\langle 1\rangle \to x_1 = x_2\} \\ \models m \in \Theta \implies \Pr_{[\![c_1]\!](m_1)}[\Psi\langle 1\rangle] < \delta'\end{array}}{\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta+\delta')} c_2\ \{x_1 = x_2\}}$$

Notes

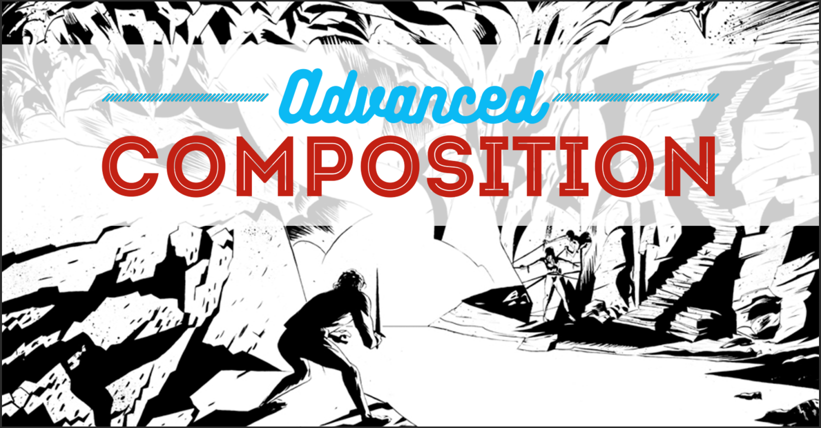
- $\Psi\langle 1\rangle$ is "bad event", only mentions c_1
- If bad event doesn't happen, have privacy

In apRHL: up-to-bad rule

$$\textsc{UtB}\ \frac{\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta)} c_2\ \{\neg\Psi\langle 1\rangle \to x_1 = x_2\} \qquad \models m \in \Theta \implies \Pr_{[\![c_1]\!](m_1)}[\Psi\langle 1\rangle] < \delta'}{\vdash \{\Phi\}\ c_1 \sim_{(\epsilon,\delta+\delta')} c_2\ \{x_1 = x_2\}}$$

Notes

- $\Psi\langle 1\rangle$ is "bad event", only mentions c_1
- If bad event doesn't happen, have privacy
- Bound probability of Ψ after c_1


Advanced
COMPOSITION

Advanced composition theorem

Compose n mechanisms, each (ϵ, δ)-private

- Standard composition: $(n \cdot \epsilon, n \cdot \delta)$-private
- Advanced composition: (ϵ^*, δ^*)-private

$$\epsilon^* \approx \sqrt{n} \cdot \epsilon \quad \text{and} \quad \delta^* \approx n \cdot \delta + \delta'$$

Advanced composition theorem

Compose n mechanisms, each (ϵ, δ)-private

- Standard composition: $(n \cdot \epsilon, n \cdot \delta)$-private
- Advanced composition: (ϵ^*, δ^*)-private

$$\epsilon^* \approx \sqrt{n} \cdot \epsilon \quad \text{and} \quad \delta^* \approx n \cdot \delta + \delta'$$

$$\boxed{\text{Trade off } \epsilon \text{ and } \delta}$$

Advanced composition theorem

Compose n mechanisms, each (ϵ, δ)-private

- Standard composition: $(n \cdot \epsilon, n \cdot \delta)$-private
- Advanced composition: (ϵ^*, δ^*)-private

$$\epsilon^* \approx \sqrt{n} \cdot \epsilon \quad \text{and} \quad \delta^* \approx n \cdot \delta + \delta'$$

$$\boxed{\text{Trade off } \epsilon \text{ and } \delta}$$

Advanced composition theorem

Compose n mechanisms, each (ϵ, δ)-private

- Standard composition: $(n \cdot \epsilon, n \cdot \delta)$-private
- Advanced composition: (ϵ^*, δ^*)-private

$$\epsilon^* \approx \sqrt{n} \cdot \epsilon \quad \text{and} \quad \delta^* \approx n \cdot \delta + \delta'$$

$$\boxed{\text{Trade off } \epsilon \text{ and } \delta}$$

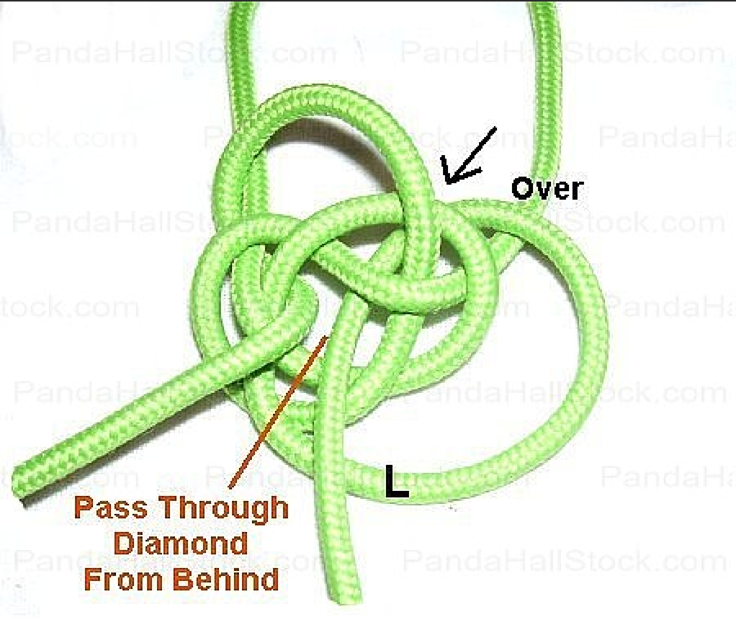
In apRHL: new while rule

$$\text{AC}\ \frac{\models \Theta \to e\langle 1\rangle = e\langle 2\rangle \qquad \vdash \{\Theta \wedge e\langle 1\rangle\}\ c_1 \sim_{(\epsilon,\delta)} c_2\ \{\Theta\} \qquad \text{while } e_1 \text{ do } c_1 \text{ exceutes at most } n \text{ iterations}}{\vdash \{\Theta\}\ \text{while } e_1 \text{ do } c_1 \sim_{(\epsilon^*,\delta^*)} \text{while } e_2 \text{ do } c_2\ \{\Theta \wedge \neg e\langle 1\rangle\}}$$

Notes

- Surprising: generalization to approximate couplings
- More surprising: privacy composition directly generalizes

Putting it all together



A brief preview: the Between Thresholds algorithm

Variant of a mechanism by Bun, Steinke, Ullman (2016)

$\mathsf{ASV}_{\mathrm{bt}}(a, b, M, N, d) :=$
$i \leftarrow 0; l \leftarrow [];$
$u \overset{\$}{\leftarrow} \mathcal{L}_{\epsilon/2}(0);$
$A \leftarrow a - u; B \leftarrow b + u;$
while $i < N \wedge |l| < M$ do
  $i' \leftarrow i; hd \leftarrow -1;$
  while $i' < N$ do
    if $(hd = -1)$
      $q \leftarrow \mathcal{A}(l);$
      $S \overset{\$}{\leftarrow} \mathcal{L}_{\epsilon'/3}(\mathsf{evalQ}(q, d));$
      if $(A \le S \le B)$ then $hd \leftarrow i;$
      $i \leftarrow i + 1;$
    $i' \leftarrow i' + 1;$
  if $(hd \neq -1)$ then $l \leftarrow hd :: l;$
return l

Formalized (ϵ, δ)-privacy in EasyCrypt

Formal proof combines many different features:

- Accuracy-dependent privacy
- Advanced composition
- Adaptively chosen inputs
- “Subset” coupling

Formal proof combines many different features:

- Accuracy-dependent privacy
- Advanced composition
- Adaptively chosen inputs
- “Subset” coupling

Formal proof combines many different features:

- Accuracy-dependent privacy
- Advanced composition
- Adaptively chosen inputs
- “Subset” coupling

Please see the paper!

Our work: formal privacy proofs with:

Accuracy-dependent privacy

Advanced composition

Adaptive inputs

Our work: formal privacy proofs with:

Accuracy-dependent privacy

Advanced composition

Adaptive inputs