Probabilistic couplings for cryptography and privacy

Gilles Barthe
IMDEA Software Institute, Madrid, Spain

September 13, 2016

Relational properties

Properties about two runs of the same program

- Assume inputs are related by Ψ
- Want to prove the outputs are related by Φ

Examples

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : out_1 \leq out_2$
- "Bigger inputs give bigger outputs"

Examples

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : out_1 \leq out_2$
- "Bigger inputs give bigger outputs"

Stability

- $\Psi : inp_1 \sim inp_2$
- $\Phi : out_1 \sim out_2$
- "If inputs are similar, then outputs are similar"

Examples

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : out_1 \leq out_2$
- "Bigger inputs give bigger outputs"

Stability

- $\Psi : inp_1 \sim inp_2$
- $\Phi : out_1 \sim out_2$
- "If inputs are similar, then outputs are similar"

Non-interference

- $\Psi : lowinp_1 = lowinp_2$
- $\Phi : lowout_1 = lowout_2$
- "If low inputs are equal, then low outputs are equal"

Probabilistic relational properties

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : \Pr[out_1 \geq k] \leq \Pr[out_2 \geq k]$

Probabilistic relational properties

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : \Pr[out_1 \geq k] \leq \Pr[out_2 \geq k]$

Stability

- $\Psi : in_1 \sim in_2$
- $\Phi : \Pr[out_1 = k] \sim \Pr[out_2 = k]$

Probabilistic relational properties

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : \Pr[out_1 \geq k] \leq \Pr[out_2 \geq k]$

Stability

- $\Psi : in_1 \sim in_2$
- $\Phi : \Pr[out_1 = k] \sim \Pr[out_2 = k]$

Non-interference

- $\Psi : lowinp_1 = lowinp_2$
- $\Phi : \Pr[lowout_1 = k] = \Pr[lowout_2 = k]$

Probabilistic relational properties

Monotonicity

- $\Psi : in_1 \leq in_2$
- $\Phi : \Pr[out_1 \geq k] \leq \Pr[out_2 \geq k]$

Stability

- $\Psi : in_1 \sim in_2$
- $\Phi : \Pr[out_1 = k] \sim \Pr[out_2 = k]$

Non-interference

- $\Psi : lowinp_1 = lowinp_2$
- $\Phi : \Pr[lowout_1 = k] = \Pr[lowout_2 = k]$

Richer properties

- Indistinguishability, differential privacy

Probabilistic couplings

- Used by mathematicians for proving relational properties
- Applications: Markov chains, probabilistic processes

Idea

- Place two processes in the same probability space
- Coordinate the sampling

Probabilistic couplings

- Used by mathematicians for proving relational properties
- Applications: Markov chains, probabilistic processes

Idea

- Place two processes in the same probability space
- Coordinate the sampling

Why is this interesting?

- Proving relational probabilistic properties reduced to proving non-relational non-probabilistic properties
- Compositional

Introducing probabilistic couplings

Basic ingredients

- Given: two distributions X_1, X_2 over set A
- Produce: joint distribution Y over $A \times A$
  - Projection over the first component is X_1
  - Projection over the second component is X_2

Introducing probabilistic couplings

Basic ingredients

- Given: two distributions X_1, X_2 over set A
- Produce: joint distribution Y over $A \times A$
  - Projection over the first component is X_1
  - Projection over the second component is X_2

Definition

Given two distributions X_1, X_2 over a set A, a coupling Y is a distribution over $A \times A$ such that $\pi_1(Y) = X_1$ and $\pi_2(Y) = X_2$

Introducing probabilistic couplings

Basic ingredients

- Given: two distributions X_1, X_2 over set A
- Produce: joint distribution Y over $A \times A$
  - Projection over the first component is X_1
  - Projection over the second component is X_2

Definition

Given two distributions X_1, X_2 over a set A, a coupling Y is a distribution over $A \times A$ such that $\pi_1(Y) = X_1$ and $\pi_2(Y) = X_2$ where

$$\pi_1(Y)(a_1) = \sum_{a_2} Y(a_1, a_2)$$

Fair coin toss

- One way to coordinate: require $x_1 = x_2$
- A different way: require $x_1 = \neg x_2$
- Yet another way: product distribution
- Choice of coupling depends on application
- Couplings always exist

Couplings vs liftings

Let $\mu_1, \mu_2 \in \mathsf{Distr}(A)$, $\mu \in \mathsf{Distr}(A \times A)$ and $R \subseteq A \times A$. Then

$$\mu \blacktriangleleft_R \langle \mu_1 \ \& \ \mu_2 \rangle \triangleq \pi_1(\mu) = \mu_1 \wedge \pi_2(\mu) = \mu_2 \wedge \Pr_{y \leftarrow \mu}[y \in R] = 1$$

Different couplings yield liftings for different relations

Convergence of random walks

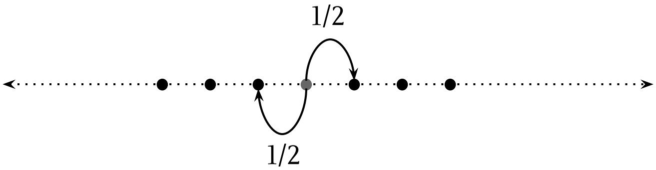
Simple random walk on integers

- Start at some position p
- Each step, flip coin $x \xleftarrow{\$} \mathit{flip}$
- Heads: $p \leftarrow p + 1$
- Tails: $p \leftarrow p - 1$

Convergence of random walks

Simple random walk on integers

- Start at some position p
- Each step, flip coin $x \xleftarrow{\$} flip$
- Heads: $p \leftarrow p + 1$
- Tails: $p \leftarrow p - 1$



Coupling the walks to meet

Case $p_1 = p_2$: Walks have met

- Arrange samplings $x_1 = x_2$
- Continue to have $p_1 = p_2$

Coupling the walks to meet

Case $p_1 = p_2$: Walks have met

- Arrange samplings $x_1 = x_2$
- Continue to have $p_1 = p_2$

Case $p_1 \neq p_2$: Walks have not met

- Arrange samplings $x_1 = \neg x_2$
- Walks make mirror moves

Coupling the walks to meet

Case $p_1 = p_2$: Walks have met

- Arrange samplings $x_1 = x_2$
- Continue to have $p_1 = p_2$

Case $p_1 \neq p_2$: Walks have not met

- Arrange samplings $x_1 = \neg x_2$
- Walks make mirror moves

Under coupling, if walks meet, they move together

Why is this interesting?

Memorylessness

Positions converge as we take more steps

Why is this interesting?

Memorylessness

Positions converge as we take more steps

Coupling bounds distance between distributions

- Once walks meet, they stay equal
- Distance is at most probability walks don't meet

Why is this interesting?

Memorylessness

Positions converge as we take more steps

Coupling bounds distance between distributions

- Once walks meet, they stay equal
- Distance is at most probability walks don't meet

Theorem

If Y is a coupling of two distributions (X_1, X_2), then

$$\|X_1 - X_2\|_{TV} \triangleq \sum_{a \in A} |X_1(a) - X_2(a)| \leq \Pr_{(y_1, y_2) \sim Y}[y_1 \neq y_2].$$

Why is this interesting?

Memorylessness

Positions converge as we take more steps

Coupling bounds distance between distributions

- Once walks meet, they stay equal
- Distance is at most probability walks don't meet

Theorem

If Y is a coupling of two distributions (X_1, X_2), then

$$\|X_1 - X_2\|_{TV} \triangleq \sum_{a \in A} |X_1(a) - X_2(a)| \leq \Pr_{(y_1, y_2) \sim Y}[y_1 \neq y_2].$$

probabilistic Relational Hoare Logic

$\vdash \{P\}c_1 \sim c_2\{Q\}$ iff there exists μ such that

$$P(m_1 \uplus m_2) \Rightarrow \mu \blacktriangleleft_Q \langle [\![c_1]\!]\ m_1\ \&\ [\![c_2]\!]\ m_2 \rangle$$

where

$$\mu \blacktriangleleft_R \langle \mu_1\ \&\ \mu_2 \rangle \triangleq \pi_1(\mu) = \mu_1 \wedge \pi_2(\mu) = \mu_2 \wedge \mathrm{supp}(\mu) \subseteq R$$

Fundamental lemma of pRHL

If $Q \triangleq E_1 \Rightarrow E_2$ then $\Pr_{([\![c_1]\!]m_1)}[E_1] \leq \Pr_{([\![c_2]\!]m_2)}[E_2]$

Core rules

$$\frac{\{\Phi\}c_1 \sim c_2\{\Theta\} \quad \{\Theta\}c_1' \sim c_2'\{\Psi\}}{\{\Phi\}c_1; c_1' \sim c_2; c_2'\{\Psi\}}$$

$$\frac{\{\Phi \wedge b_1 \wedge b_2\}c_1 \sim c_2\{\Psi\} \quad \{\Phi \wedge \neg b_1 \wedge \neg b_2\}c_1' \sim c_2'\{\Psi\}}{\{\Phi \wedge b_1 = b_2\}\text{if } b_1 \text{ then } c_1 \text{ else } c_1' \sim \text{if } b_2 \text{ then } c_2 \text{ else } c_2'\{\Psi\}}$$

$$\frac{\{\Phi \wedge b_1 \wedge b_2\}c_1 \sim c_2\{\Phi \wedge b_1 = b_2\}}{\{\Phi \wedge b_1 = b_2\}\text{while } b_1 \text{ do } c_1 \sim \text{while } b_2 \text{ do } c_2\{\Phi \wedge \neg b_1 \wedge \neg b_2\}}$$

Loops

- Benton: same number of iterations
- EasyCrypt ($\leq$ 2015): one-sided rules
- EasyCrypt (2016): asynchronous loop rule
  $\Longrightarrow$ relatively complete, subsumes 1-sided rules

$$\frac{\begin{array}{c} \Psi \implies p_0 \oplus p_1 \oplus p_2 \\ \Psi \wedge p_0 \implies e_1 \wedge e_2 \qquad \Psi \wedge p_1 \implies e_1 \qquad \Psi \wedge p_2 \implies e_2 \\ \text{while } e_1 \wedge p_1 \text{ do } c_1 \Downarrow \text{while } e_2 \wedge p_2 \text{ do } c_2 \\ \{\Psi \wedge p_1\} c_1 \sim \text{skip} \{\Psi\} \qquad \{\Psi \wedge p_2\} \text{skip} \sim c_2 \{\Psi\} \\ \{\Psi \wedge p_0\} c_1 \sim c_2 \{\Psi\} \end{array}}{\{\Psi\} \text{while } e_1 \text{ do } c_1 \sim \text{while } e_2 \text{ do } c_2 \{\Psi \wedge \neg e_1 \wedge \neg e_2\}}$$

Example

$x \leftarrow 0; i \leftarrow 0; \text{while } i \leq N \text{ do } (x \mathrel{+}= i; i{+}{+})$

$y \leftarrow 0; j \leftarrow 1; \text{while } j \leq N \text{ do } (y \mathrel{+}= j; j{+}{+})$

Rule for random assignment

$$\frac{\mu \blacktriangleleft_Q \langle \mu_1 \,\&\, \mu_2 \rangle}{\vdash \{\top\} x_1 \xleftarrow{\$} \mu_1 \sim x_2 \xleftarrow{\$} \mu_2 \{Q\}}$$

Specialized rule

$$\frac{f \in T \xrightarrow{1-1} T \qquad \forall v \in T.\, d_1(v) = d_2(f\ v)}{\vdash \{\forall v, Q[v/x_1, f\ v/x_2]\} x_1 \xleftarrow{\$} \mu_1 \sim x_2 \xleftarrow{\$} \mu_2 \{Q\}}$$

Notes

- Bijection f: specifies how to coordinate the samples
- Side condition: marginals are preserved under f
- Assume: samples coupled when proving postcondition Φ

Proofs as (products) programs: xpRHL

- Every pRHL derivation yields a product program
- Different derivations yield different programs
- Can be modelled by a proof system

$$\vdash \{\Phi\} c_1 \sim c_2 \{\Psi\} \rightsquigarrow c$$

Fundamental lemma of xpRHL

- $\vdash \{\Phi\} c_1 \sim c_2 \{\Psi \implies x_1 = x_2\} \rightsquigarrow c$
- $\{\Box\Phi\}\ c\ \{\Pr[\neg\Psi] \leq \epsilon\}$

implies

$$m_1\ \Phi\ m_2 \Rightarrow \left|\Pr_{(\llbracket c_1 \rrbracket\ m_1)}[E(x_1)] - \Pr_{(\llbracket c_2 \rrbracket\ m_2)}[E(x_2)]\right| \leq \epsilon$$

Dynkin's card trick (shift coupling)

$p \leftarrow s$; $l \leftarrow [p]$;
while $p < N$ **do**
  $n \xleftarrow{\$} [1, 10]$;
  $p \leftarrow p + n$;
  $l \leftarrow p :: l$;
return p

$p_1 \leftarrow s_1$; $p_2 \leftarrow s_2$;
$l_1 \leftarrow [p_1]$; $l_2 \leftarrow [p_2]$;
while $n_1 < N \vee n_2 < N$ **do**
  if $p_1 = p_2$ **then**
    $n \xleftarrow{\$} ([1, 10])$;
    $p_1 \leftarrow p_1 + n$; $p_2 \leftarrow p_2 + n$;
    $l_1 \leftarrow p_1 :: l_1$; $l_2 \leftarrow p_2 :: l_2$;
  else
    if $p_1 < p_2$ **then**
      $n_1 \xleftarrow{\$} [1, 10]$;
      $p_1 \leftarrow p_1 + n_1$;
      $l_1 \leftarrow p_1 :: l_1$;
    else
      $n_2 \xleftarrow{\$} [1, 10]$;
      $p_2 \leftarrow p_2 + n_2$;
      $l_2 \leftarrow p_2 :: l_2$;
return (p_1, p_2)

Convergence

If $s_1, s_2 \in [1, 10]$, and $N > 10$, then $\Delta(p_1^{\text{final}}, p_2^{\text{final}}) \leq (9/10)^{N/5-2}$

Applications to cryptography

Experiment G_1

- Cryptosystem
- Adversary $\mathcal{A}$
- Winning condition E

Experiment G_2

- Hardness assumption
- Adversary $\mathcal{B}$
- Winning condition F

For all adversary $\mathcal{A}$, there exists adversary $\mathcal{B}$ s.t. $t_{\mathcal{A}} \approx t_{\mathcal{B}}$ and

$$\Pr_{G_1}[E] \leq q \cdot \Pr_{G_2}[F] + \delta$$

Applications to cryptography

Experiment G_1

- Cryptosystem
- Adversary $\mathcal{A}$
- Winning condition E

Experiment G_2

- Hardness assumption
- Adversary $\mathcal{B}$
- Winning condition F

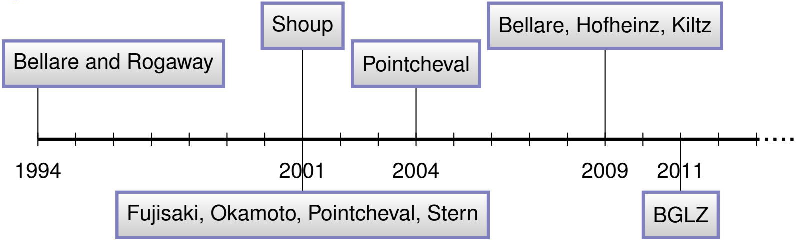
For all adversary $\mathcal{A}$, there exists adversary $\mathcal{B}$ s.t. $t_{\mathcal{A}} \approx t_{\mathcal{B}}$ and

- $\vdash \{\top\} G_1 \sim G_2 \{E \Rightarrow (F' \vee F_{bad})\}$
- $\Pr_{G_2}[F'] \leq q \cdot \Pr_{G_2}[F]$ and $\Pr_{G_2}[F_{bad}] \leq \delta$

Formalizing cryptographic proofs?

- *In our opinion, many proofs in cryptography have become essentially unverifiable. Our field may be approaching a crisis of rigor*. Bellare and Rogaway, 2004-2006
- *Do we have a problem with cryptographic proofs? Yes, we do [...] We generate more proofs than we carefully verify (and as a consequence some of our published proofs are incorrect)*. Halevi, 2005

OAEP



Provable security of OAEP

Game INDCCA($\mathcal{A}$) :	Encryption	Game sPDOW($\mathcal{I}$)
$(sk, pk) \leftarrow \mathcal{K}(\,)$;	$\mathcal{E}_{\mathrm{OAEP}(pk)}(m)$:	$(sk, pk) \leftarrow \mathcal{K}()$;
$(m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{G},\mathcal{H},\mathcal{D}}(pk)$;	$r \stackrel{\$}{\leftarrow} \{0,1\}^{k_0}$;	$y_0 \stackrel{\$}{\leftarrow} \{0,1\}^{n_0}$;
$b \stackrel{\$}{\leftarrow} \{0,1\}$;	$s \leftarrow G(r) \oplus (m \parallel 0^{k_1})$;	$y_1 \stackrel{\$}{\leftarrow} \{0,1\}^{n_1}$;
$c^\star \leftarrow \mathcal{E}_{pk}(m_b)$;	$t \leftarrow H(s) \oplus r$;	$x^\star \leftarrow \mathrm{f}_{pk}(y_0 \parallel y_1)$;
$\overline{b} \leftarrow \mathcal{A}_2^{\mathcal{G},\mathcal{H},\mathcal{D}}(c^\star)$;	return $\mathrm{f}_{pk}(s \parallel t)$	$\overline{Y} \leftarrow \mathcal{I}(x^\star)$;
return $(\overline{b} = b)$	**Decryption** ...	return $(y_0 \in \overline{Y})$

FOR ALL IND-CCA adversary $\mathcal{A}$ against $(\mathcal{K}, \mathcal{E}_{\mathrm{OAEP}}, \mathcal{D}_{\mathrm{OAEP}})$,
THERE EXISTS a sPDOW adversary $\mathcal{I}$ against $(\mathcal{K}, \mathrm{f}, \mathrm{f}^{-1})$ st

$$\left|\Pr_{\text{IND-CCA}(\mathcal{A})}\left[\overline{b} = b\right] - \frac{1}{2}\right| \leq \Pr_{\text{PDOW}(\mathcal{I})}\left[y_0 \in \overline{Y}\right] + \frac{3q_D q_G + q_D^2 + 4q_D + q_G}{2^{k_0}} + \frac{2q_D}{2^{k_1}}$$

and

$$t_{\mathcal{I}} \leq t_{\mathcal{A}} + q_D\, q_G\, q_H\, T_{\mathrm{f}}$$

The code-based game-playing approach

- Everything is a probabilistic program
- Decompose the proof in sequence of transitions
- Prove each transition using pRHL
- Bound prob. of events w/ non-relational logic

Typical couplings

- Bridging step: $\mu_1 \; =^{\#} \mu_2$, then for every event X,

$$\Pr_{z \leftarrow \mu_1}[X] = \Pr_{z \leftarrow \mu_2}[X]$$

- Failure Event: If $x \; R \; y$ iff $F(x) \Rightarrow x = y$ and $F(x) \Leftrightarrow F(y)$, then for every event X,

$$|\Pr_{z \leftarrow \mu_1}[X] - \Pr_{z \leftarrow \mu_2}[X]| \leq \max\left(\Pr_{z \leftarrow \mu_1}[\neg F], \Pr_{z \leftarrow \mu_2}[\neg F]\right)$$

- Reduction: If $x \; R \; y$ iff $F(x) \Rightarrow G(y)$, then

$$\Pr_{x \leftarrow \mu_2}[G] \leq \Pr_{y \leftarrow \mu_1}[F]$$

EasyCrypt

- Interactive proof assistant
  - backend to SMT solvers, CAS, etc.
  - encryption, signatures, hash designs, key exchange protocols, zero knowledge protocols, garbled circuits. . .
  - SHA3, e-voting
- Back-end for automated tools
- Front-end for certified compilers

approximate probabilistic Relational Hoare Logic

- Quantitative generalization of pRHL $\vdash_{\epsilon,\delta} \{P\} c_1 \sim c_2 \{Q\}$
- Valid if there exists μ_L, μ_R such that

$$P(m_1 \uplus m_2) \implies \mu_L, \mu_R \blacktriangleleft_Q^{\epsilon,\delta} \langle [\![c_1]\!]\ m_1\ \&\ [\![c_2]\!]\ m_2 \rangle$$

where

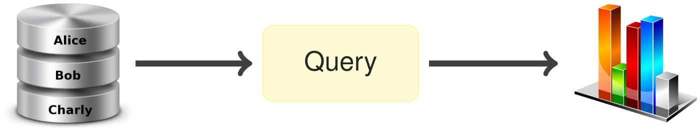
$$\mu_L, \mu_R \blacktriangleleft_Q^{\epsilon,\delta} \langle \mu_1\ \&\ \mu_2 \rangle \triangleq \begin{cases} \pi_1(\mu_L) = \mu_1 \wedge \pi_2(\mu_R) = \mu_2 \\ \text{supp}(\mu_L), \text{supp}(\mu_R) \subseteq Q \\ \Delta_\epsilon(\mu_1, \mu_2) \leq \delta \end{cases}$$

- Fundamental theorem of apRHL: if $Q \triangleq E_1 \Rightarrow E_2$ then

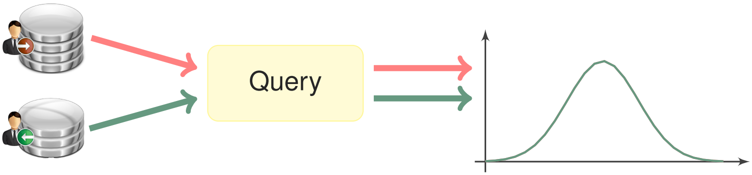
$$\Pr_{([\![c_1]\!]\ m_1)}[E_1] \leq \exp(\epsilon) \Pr_{([\![c_2]\!]\ m_2)}[E_2] + \delta$$

- Extends to f-divergences

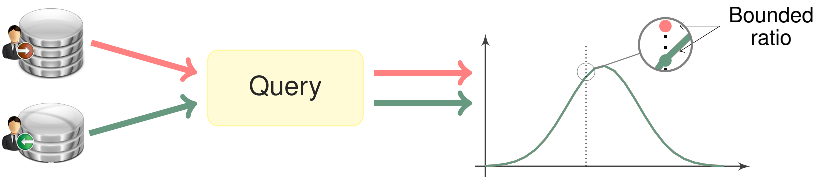
Application: differential privacy



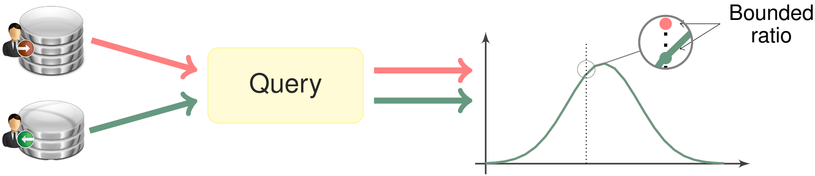
Application: differential privacy



Application: differential privacy



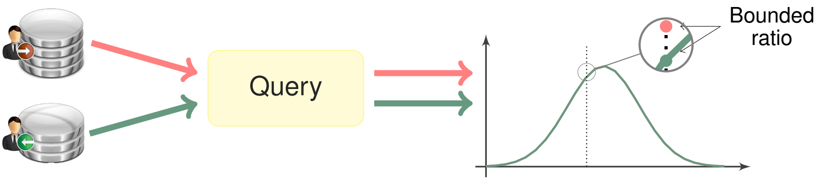
Application: differential privacy



A randomized algorithm $\mathcal{K}$ is (ϵ, δ)-differentially private w.r.t. Φ iff for all databases D_1 and D_2 s.t. $\Phi(D_1, D_2)$

$$\forall S.\ \Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{K}(D_2) \in S] + \delta$$

Application: differential privacy



A randomized algorithm $\mathcal{K}$ is (ϵ, δ)-differentially private w.r.t. Φ iff for all databases D_1 and D_2 s.t. $\Phi(D_1, D_2)$

$$\forall S.\ \Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{K}(D_2) \in S] + \delta$$

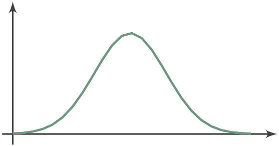
Privacy as approximate couplings

$\mathcal{K}$ is (ϵ, δ)-differentially private wrt Φ iff $\vdash_{\epsilon,\delta} \{\Phi\} \mathcal{K}_1 \sim \mathcal{K}_2 \{\equiv\}$

Differential privacy via output perturbation

Let f be k-sensitive w.r.t. Φ:

$$\Phi(a, a') \implies |f\ a - f\ a'| \leq k$$



Then $a \mapsto \mathrm{Lap}_\epsilon(f(a))$ is $(k \cdot \epsilon, 0)$-differentially private w.r.t. Φ

Proof principles for Laplace mechanism

Making different things look equal

$$\frac{\Phi \triangleq |e_1 - e_2| \leq k'}{\vdash_{k'\cdot\epsilon,0} \{\Phi\} y_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim y_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2) \{y_1 = y_2\}}$$

Making equal things look different

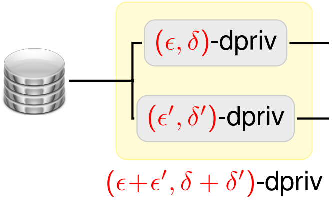
$$\frac{\Phi \triangleq e_1 = e_2}{\vdash_{k\cdot\epsilon,0} \{\Phi\} y_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim y_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2) \{y_1 + k = y_2\}}$$

Pointwise equality

$$\frac{\forall i.\ \vdash_{\epsilon,0} \{\Phi\} c_1 \sim c_2 \{x_1 = i \Rightarrow x_2 = i\}}{\vdash_{\epsilon,0} \{\Phi\} c_1 \sim c_2 \{x_1 = x_2\}}$$

Differential privacy by sequential composition

- If $\mathcal{K}$ is (ϵ, δ)-differentially private, and
- $\lambda a.\ \mathcal{K}'(a, b)$ is (ϵ', δ')-differentially private for every $b \in B$,
- then $\lambda a.\ \mathcal{K}'(a, \mathcal{K}(a))$ is $(\epsilon + \epsilon', \delta + \delta')$-differentially private



Beyond composition: Sparse Vector Technique

$\mathsf{SparseVector}_{bt}(a, b, M, N, d) :=$
$i \leftarrow 0; l \leftarrow []; u \xleftarrow{\$} \mathcal{L}_\epsilon(0); A \leftarrow a - u; B \leftarrow b + u;$
while $i < N$ do
  $i \leftarrow i + 1; q \leftarrow \mathcal{A}(l); S \xleftarrow{\$} \mathcal{L}_\epsilon(q(d));$
  if $(A \le S \le B \wedge |l| < M)$ then $l \leftarrow i :: l;$
return l

Privacy

If queries are 1-sensitive, then $(\sqrt{M}\epsilon, \delta')$-diff. private

Tools

- advanced composition
- accuracy-dependent privacy
- optimal subset coupling

Perspectives and further directions

Language-based techniques

- for provable security and differential privacy
- based on probabilistic couplings

Open questions

- semantical foundations of approximate couplings
- applications to security (complexity of attacks)