Approximate Relational Reasoning for Probabilistic Programs

PhD Candidate: Federico Olmedo
Supervisor: Gilles Barthe

IMDEA Software Institute

PhD Examination – Technical University of Madrid
January 9, 2014
Scenario: 2 new rehab. centers to be opened; 4 feasible locations.
Goal: select locations that minimize average patient commute time.
**Scenario:** 2 new rehab. centers to be opened; 4 feasible locations.

**Goal:** select locations that minimize average patient commute time.
**Scenario:** 2 new rehab. centers to be opened; 4 feasible locations.

**Goal:** select locations that minimize average patient commute time.
Scenario: 2 new rehab. centers to be opened; 4 feasible locations.
Goal: select locations that minimize average patient commute time.
**Scenario:** 2 new rehab. centers to be opened; 4 feasible locations.

**Goal:** select locations that minimize average patient commute time.

**Optimum Solution Approach:**
- Highest utility.
- Leakage of sensitive information.
The Privacy–Utility Conflict

![Balance Scale with 'privacy' and 'utility' labels]
Differential Privacy (DP)
[Dwork+, ICALP ’06]
The Privacy–Utility Conflict

**Differential Privacy (DP)**
[Dwork+, ICALP ’06]

Privacy definition

---

![Diagram showing the concept of Differential Privacy (DP) with privacy and utility scales, selection algorithm, and related icons.](image)
The Privacy–Utility Conflict

Differential Privacy (DP)  
[Dwork+, ICALP ’06]

1 Privacy definition

2 Privacy realization
- Basic mechanisms for numeric/discrete-valued computations.
- Composition theorems.
Differentially Private Location Selection

[Gupta+, SODA’10]

\[
\text{function } k\text{M}edian(C, F_0)
\]

1. \( i \leftarrow 0; \)
2. \( \text{while } i < T \text{ do} \)
3. \( (x, y) \leftarrow \text{pick–swap}(F_i \times \overline{F_i}); \)
4. \( F_{i+1} \leftarrow (F_i \setminus \{x\}) \cup \{y\}; \)
5. \( i \leftarrow i + 1 \)
6. \( \text{end;} \)
7. \( j \leftarrow \text{pick–solution}([1, \ldots, T], F); \)
8. \( \text{return } F_j \)
Verifying Differential Privacy

Dynamic verification:
- PINQ [McSherry ’09]
- AIRAVAT [Roy+ ’10]

Static verification:
- Fuzz [Reed & Pierce ’10] and DFuzz [Gaboardi+ ’13]
  - [Chaudhuri+ ’11]

Limitations of these techniques:
- Only programs that are combinations of basic mechanisms.
- Only standard differential privacy.
- Fixed set of domains and/or operations.
In this Dissertation

Our Goal

Verify differential privacy properties of probabilistic programs.

We want our technique to

- Circumvent limitations of existing techniques.
- Provide strong evidence of correctness.
- Be extensible to reason about other quantitative properties of probabilistic programs.
Outline

1. Motivation
2. Verification of Differential Privacy
3. Extensions of our Technique
4. Summary and Conclusions
Outline

1 Motivation

2 Verification of Differential Privacy

3 Extensions of our Technique

4 Summary and Conclusions
Differential Privacy – Definition

A randomized mechanism $K$ is $(\epsilon, \delta)$-differentially private if for all databases $d_1$ and $d_2$, and all events $A$,

$$\Delta(d_1, d_2) \leq 1 \Rightarrow \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A]$$
Differential Privacy – Definition

A randomized mechanism $K$ is $\epsilon$-differentially private if for all databases $d_1$ and $d_2$, and all events $A$, $\Delta(d_1, d_2) \leq 1 \Rightarrow \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A]$.
Differential Privacy – Definition

A randomized mechanism $K$ is $\epsilon$-differentially private if for all databases $d_1$ and $d_2$, and all events $A$, $\Delta(d_1, d_2) \leq 1 \Rightarrow \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A]$. 

8/26
Differential Privacy – Definition

A randomized mechanism $K$ is $\epsilon$-differentially private if for all databases $d_1$ and $d_2$, and all events $A$, $\Delta(d_1, d_2) \leq 1 \Rightarrow Pr[K(d_1) \in A] \leq e^\epsilon Pr[K(d_2) \in A]$. 

$\Delta(d_1, d_2)$ is the ratio of the change in the output probability distribution when the input changes from $d_1$ to $d_2$. $e^\epsilon$ is the privacy guarantee, indicating the relative change in the output probability distribution.
A randomized mechanism $K$ is \textit{$\epsilon$-differentially private} iff for all databases $d_1$ and $d_2$, and all events $A$,

$$\Delta(d_1, d_2) \leq 1 \implies \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A]$$
A randomized mechanism $K$ is **$\epsilon$-differentially private** iff for all databases $d_1$ and $d_2$, and all events $A$,

$$\Delta(d_1, d_2) \leq 1 \implies \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A]$$
A randomized mechanism $K$ is $(\epsilon, \delta)$-differentially private iff for all databases $d_1$ and $d_2$, and all events $A$,

$$\Delta(d_1, d_2) \leq 1 \implies \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A] + \delta$$
Differential Privacy – Fundamentals

- Basic mechanism for numeric queries.
  
  ![Diagram](https://via.placeholder.com/150)

- Composition theorem.
Verifying Differential Privacy – Our Approach

Differential privacy is a **quantitative 2-safety property**:

$$\Delta(d_1, d_2) \leq 1 \implies \forall A \cdot \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A] + \delta$$
Differential privacy is a **quantitative 2-safety property**:

\[ \Delta(d_1, d_2) \leq 1 \implies \forall A \cdot \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A] + \delta \]
Differential privacy is a quantitative 2-safety property:

\[ \Delta(d_1, d_2) \leq 1 \implies \forall A \cdot \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A] + \delta \]
Verifying Differential Privacy – Our Approach

Differential privacy is a **quantitative 2-safety property**:

\[ \Delta(d_1, d_2) \leq 1 \implies \forall A \cdot \Pr[K(d_1) \in A] \leq e^\epsilon \Pr[K(d_2) \in A] + \delta \]

We propose a **quantitative probabilistic relational Hoare logic**

\[ \{\Psi\} c_1 \sim_{\alpha, \delta} c_2 \{\Phi\} \]

such that a program \( c \) is \((\epsilon, \delta)\)-DP iff

\[ \{\equiv\} c \sim_{\epsilon, \delta} c \{\equiv\} \]

database adjacency  
equality on observable output
Relational Program Reasoning

Standard Hoare Logic

\[ \models \{ \Psi \} \, c \, \{ \Phi \} \]

\[ m \quad \Psi(m) \]

\[ \llbracket c \rrbracket \]

\[ m' \quad \Phi(m') \]
Standard Hoare Logic

\[ \models \{ \Psi \} \ c \ \{ \Phi \} \]

\[ m \xrightarrow{\llbracket c \rrbracket} m' \]
\[ \Psi(m) \quad \Phi(m') \]

Relational Hoare Logic

\[ \models \{ \Psi \} c_1 \sim c_2 \ \{ \Phi \} \]

\[ m_1 \xrightarrow{\llbracket c_1 \rrbracket} m_2 \xleftarrow{\Psi} m_1' \xrightarrow{\llbracket c_2 \rrbracket} m_2' \]
\[ m_1 \xleftarrow{\Phi} m_2' \]
Our Goal

$c$ is $(\epsilon, \delta)$-DP iff $\{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.
Characterizing Differential Privacy

Our Goal

\[ c \text{ is } (\epsilon, \delta)\text{-DP} \iff \equiv c \sim_{\epsilon, \delta} c \equiv \]

To achieve so we rely on a lifting operation and a distance measure.

\[ \mathcal{L}_\alpha^\delta (\cdot) \quad \Delta_\alpha (\cdot, \cdot) \quad \alpha \geq 1, \delta \geq 0 \]

\[ \mathcal{P} (A \times B) \rightarrow \mathcal{P} (\mathcal{D}_A \times \mathcal{D}_B) \quad \mathcal{D}_A \times \mathcal{D}_A \rightarrow \mathbb{R}^{\geq 0} \]
Characterizing Differential Privacy

**Our Goal**

\[ c \text{ is } (\epsilon, \delta)\text{-DP iff } \{\simeq\} c \sim_{\epsilon,\delta} c \{\equiv\} \]

To achieve so we rely on a **lifting operation** and a **distance measure**.

\[ \mathcal{L}_\alpha^\delta(\cdot) \quad \text{ and } \quad \Delta_\alpha(\cdot, \cdot) \quad \alpha \geq 1, \delta \geq 0 \]

\( \mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) \quad \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0} \)

Judgment \( \{\Psi\} c_1 \sim_{\alpha,\delta} c_2 \{\Phi\} \) is interpreted as

\[ m_1 \Psi m_2 \implies ([c_1] m_1) \mathcal{L}_\alpha^\delta(\Phi) ([c_2] m_2) \]
Our Goal

\( c \) is \((\epsilon, \delta)\)-DP iff \( \{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\} \)

To achieve so we rely on a lifting operation and a distance measure.

\[
\mathcal{L}_\alpha^\delta(\cdot) \quad \Delta_\alpha(\cdot, \cdot) \quad a \geq 1, \delta \geq 0
\]

\( \mathcal{P}(A \times B) \rightarrow \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) \quad \mathcal{D}_A \times \mathcal{D}_A \rightarrow \mathbb{R}^{\geq 0} \)

Judgment \( \{\Psi\} c_1 \sim_{\alpha, \delta} c_2 \{\Phi\} \) is interpreted as

\[
m_1 \Psi m_2 \implies ([c_1] m_1) \mathcal{L}_\alpha^\delta(\Phi) ([c_2] m_2)
\]

\( c \) is \((\epsilon, \delta)\)-DP iff for all memories \( m_1 \) and \( m_2 \),

\[
m_1 \simeq m_2 \implies \forall A \cdot \Pr [c(m_1) \in A] \leq e^\epsilon \Pr [c(m_2) \in A] + \delta
\]
Characterizing Differential Privacy

Our Goal

\[ c \text{ is } (\epsilon, \delta)\text{-DP} \iff \{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\} \]

To achieve so we rely on a lifting operation and a distance measure.

\[ \mathcal{L}_\alpha^\delta(\cdot) \quad \Delta_\alpha(\cdot, \cdot) \quad \alpha \geq 1, \delta \geq 0 \]

\[ \mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) \quad \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0} \]

Judgment \( \{\Psi\} c_1 \sim_{\alpha, \delta} c_2 \{\Phi\} \) is interpreted as

\[ m_1 \Psi m_2 \implies (\llbracket c_1 \rrbracket m_1) \mathcal{L}_\alpha^\delta(\Phi) (\llbracket c_2 \rrbracket m_2) \]

\[ \therefore c \text{ is } (\epsilon, \delta)\text{-DP} \iff \text{for all memories } m_1 \text{ and } m_2, \]

\[ m_1 \simeq m_2 \implies \Delta_{\epsilon^\delta} (\llbracket c \rrbracket m_1, \llbracket c \rrbracket m_2) \leq \delta \]
Characterizing Differential Privacy

Our Goal

\[ c \text{ is } (\epsilon, \delta)\text{-DP} \quad \text{iff} \quad \{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\} \]

To achieve so we rely on a **lifting operation** and a **distance measure**.

\[ \mathcal{L}_\alpha^\delta(\cdot) \quad \Delta_\alpha(\cdot, \cdot) \quad a \geq 1, \delta \geq 0 \]

Judgment \( \{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\} \) is interpreted as

\[ m_1 \simeq m_2 \implies ([c] m_1) \mathcal{L}_\epsilon^\delta(\equiv) ([c] m_2) \]

\[ c \text{ is } (\epsilon, \delta)\text{-DP} \text{ iff for all memories } m_1 \text{ and } m_2, \]

\[ m_1 \simeq m_2 \implies \Delta_\epsilon ([c] m_1, [c] m_2) \leq \delta \]
Characterizing Differential Privacy

Our Goal

\[
\begin{align*}
c \text{ is } (\epsilon, \delta)\text{-DP} & \iff \{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\}
\end{align*}
\]

To achieve so we rely on a lifting operation and a distance measure.

\[
\begin{array}{ll}
\mathcal{L}_\alpha^\delta(\cdot) & \Delta_\alpha (\cdot, \cdot) \\
\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) & \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}
\end{array}
\]

1. Judgment \{\simeq\} c \sim_{\epsilon, \delta} c \{\equiv\} is interpreted as

\[
m_1 \simeq m_2 \implies ([c] m_1) \mathcal{L}_\epsilon^\delta (\equiv) ([c] m_2)
\]

2. The lifting \(\mathcal{L}_\alpha^\delta(\equiv)\) of equality is characterized as

\[
\mu_1 \mathcal{L}_\alpha^\delta(\equiv) \mu_2 \iff \Delta_\alpha (\mu_1, \mu_2) \leq \delta
\]

3. \(c\) is \((\epsilon, \delta)\)-DP iff for all memories \(m_1\) and \(m_2\),

\[
m_1 \simeq m_2 \implies \Delta_{\epsilon^\delta} ([c] m_1, [c] m_2) \leq \delta
\]
Definition of the $\alpha$-distance is straightforward.

$$\Delta_\alpha (\mu_1, \mu_2) \overset{\Delta}{=} \max_A \Pr [\mu_1 \in A] - \alpha \Pr [\mu_2 \in A]$$

Definition of the $(\alpha, \delta)$-lifting is somewhat intricate (in the general case),

... but simpler characterization for equiv. relations.
\[
C ::= \begin{array}{ll}
\text{skip} & \text{nop} \\
\text{C; C} & \text{sequence} \\
\mathcal{V} \leftarrow \mathcal{E} & \text{assignment} \\
\mathcal{V} \leftarrow \mathcal{D} & \text{random sampling} \\
\text{if} \ \mathcal{E} \ \text{then} \ \text{C} \ \text{else} \ \text{C} & \text{conditional} \\
\text{while} \ \mathcal{E} \ \text{do} \ \text{C} & \text{while loop} \\
\mathcal{V} \leftarrow \mathcal{P}(\mathcal{E}, \ldots, \mathcal{E}) & \text{procedure call}
\end{array}
\]
Weakening

\[
\begin{align*}
\models \{\Psi'\} c_1 \sim_{\alpha', \delta'} c_2 \{\Phi'\} \\
\Psi \Rightarrow \Psi' \quad \Phi' \Rightarrow \Phi \quad \alpha' \leq \alpha \quad \delta' \leq \delta \\
\models \{\Psi\} c_1 \sim_{\alpha, \delta} c_2 \{\Phi\}
\end{align*}
\]
The Proof System

Weakening

\[ \vdash \{ \Psi' \} c_1 \sim_{\alpha', \delta'} c_2 \{ \Phi' \} \]
\[ \Psi \Rightarrow \Psi' \quad \Phi' \Rightarrow \Phi \quad \alpha' \leq \alpha \quad \delta' \leq \delta \]
\[ \vdash \{ \Psi \} c_1 \sim_{\alpha, \delta} c_2 \{ \Phi \} \]

Sequential composition

\[ \vdash \{ \Psi \} c_1 \sim_{\alpha_1, \delta_1} c_2 \{ \Phi' \} \quad \vdash \{ \Phi' \} c'_1 \sim_{\alpha_2, \delta_2} c'_2 \{ \Phi \} \]
\[ \vdash \{ \Psi \} c_1 ; c'_1 \sim_{\alpha_1 \alpha_2, \delta_1 + \delta_2} c_2 ; c'_2 \{ \Phi \} \]
The Proof System – Cont’d

Laplacian mechanism

Output perturbation makes numerical queries $\epsilon$-DP

The sensitivity of a numerical query $f : \mathcal{D} \to \mathbb{R}$ is defined as:

$$\Delta f \triangleq \max_{d_1, d_2} |f(d_1) - f(d_2)|$$

$$Lap(\lambda) \quad \lambda = 0.40$$

$$Lap(\lambda) \quad \lambda = 0.65$$
Laplacian mechanism

Output perturbation makes numerical queries $\epsilon$-DP

The sensitivity of a numerical query $f : \mathcal{D} \rightarrow \mathbb{R}$ is defined as:

$$\Delta_f \triangleq \max_{d_1 \approx d_2} |f(d_1) - f(d_2)|$$

$$\text{Lap}(\lambda)$$

$$\lambda = 0.40$$

$$\lambda = 0.65$$

$$m_1 \Psi m_2 \implies |\llbracket r \rrbracket m_1 - \llbracket r \rrbracket m_2| \leq k$$

$$\models \{ \Psi \} x \leftarrow \mathcal{L}(r, k/\epsilon) \sim_{e,0} x \leftarrow \mathcal{L}(r, k/\epsilon) \{ x(1) = x(2) \}$$
CertiPriv: framework proving interactive support for the logic built on top of the Coq proof assistant.

- Delivers machine-checked proofs of differential privacy.
- Built as an extension of CertiCrypt.
  - $\alpha$-distance + $(\alpha, \delta)$-lifting + logic soundness (+6.500 lines of Coq proof-script)
- Several case studies:
  - Laplacian, Exponential and Gaussian basic mechanisms.
  - $k$-Median, Minimum Vertex Cover, streaming algorithm.
Case Study: $k$-Median Problem

```python
function kMedian(C, F_0)
    i ← 0;
    while i < T do
        (x, y) ← pick-swap(F_i × \overline{F_i});
        F_{i+1} ← (F_i \setminus \{x\}) \cup \{y\};
        i ← i + 1
    end;
    j ← pick-solution([1, \ldots, T], F);
    return F_j
```

Differential privacy is captured by judgment

$\{\Psi\} kMedian \sim_{\alpha, 0} kMedian \{\Phi\}$

$C\langle 1 \rangle \approx C\langle 2 \rangle \land F_0\langle 1 \rangle = F_0\langle 2 \rangle$

$e^{2\epsilon \Delta(T+1)}$

$F_j\langle 1 \rangle = F_j\langle 2 \rangle$

Judgment derivation + Verification of side conditions $\approx$ 450 lines proof-script
Verifying Differential Privacy – Summary

- Program logic for reasoning about DP.
- Framework for building machine-checked proofs of DP

With G. Barthe, B. Köpf and S. Zanella Béguelin
[POPL ’12] [TOPLAS ’13]
Our Goal

Verify differential privacy properties of probabilistic programs.

We want our technique to

- Provides strong evidence of correctness. ✓
- Circumvent limitations of existing techniques. ✓
- Be extensible to reason about other quantitative properties of probabilistic programs.
1 Motivation

2 Verification of Differential Privacy

3 Extensions of our Technique

4 Summary and Conclusions
Differential privacy is a **quantitative relational property** of probabilistic programs:

\[ m_1 \Psi m_2 \implies \Delta ([c_1] m_1, [c_2] m_2) \leq \delta \]

But it is not the only one!

- Indifferentiability
- Zero Knowledge
- Pseudo-randomness
- …

Can we use our logic as it is to reason about these properties as well?

**NO.** They use distance measures different from the \( \alpha \)-distance.
Extending our Logic

Our logic is extensible to the class of $f$-divergences.

The class of $f$-divergences comprises well-known examples of distance measures and finds applications in multiple areas:

- Statistical distance
- Hellinger distance
- Relative entropy
- $\alpha$-distance
- $\chi^2$-distance
Our logic is extensible to the class of $f$-divergences.

With G. Barthe [ICALP '13]

The class of $f$-divergences comprises well-known examples of distance measures and finds applications in multiple areas:

- Statistical distance
- Hellinger distance
- Relative entropy
- $\alpha$-distance
- $\chi^2$-distance

$\text{Statistical distance} \quad \text{Hellinger distance} \quad \text{Relative entropy} \quad \text{$\alpha$-distance} \quad \text{$\chi^2$-distance} \quad \text{Cryptography} \quad \text{Information Theory} \quad \text{Pattern Recognition} \quad \text{Image Processing} \quad \text{Data Mining}
Our Goal

Verify differential privacy properties of probabilistic programs.

We want our technique to

- Provides strong evidence of correctness. ✓
- Circumvent limitations of existing technique. ✓
- Be extensible to reason about other quantitative properties of probabilistic programs. ✓
Crypto Case Study: Secure Hash Functions into Elliptic Curves [Brier+ ’10]

Security is captured by formula

\[ \forall D \cdot \Delta_{SD}(D^{H,h}, D^{RO,S}) \leq \epsilon \]

Our machine-checked proof

- Approximate observational equivalence (specialization of our Hoare logic) + adversary rule.
- Requires heavy algebraic reasoning (elliptic curves and group theory).
- 10,000+ lines of Coq proof-script.

With G. Barthe, B. Grégoire, S. Heraud, S. Zanella [POST ’12, JCS ’14]
Conclusions

Summary of contributions

- Quantitative relational Hoare logic for approximate reasoning about probabilistic programs.
- Framework for building machined-checked proofs of differential privacy (and other quantitative properties).
- Verification of several constructions from the recent literature.

Future work

- Improve automation (e.g. inference of loop invariants).
- Lipschitz continuity of probabilistic programs.
- Combination of different techniques.
The \((\alpha, \delta)\)-lifting

Witness distributions in \(\mathcal{D}_{A \times B}\)

\[
\mu_1 \mathcal{L}_\alpha^\delta(R) \mu_2 \triangleq \exists \mu_L, \mu_R \cdot \begin{cases} 
\Delta_\alpha(\mu_L, \mu_R) \leq \delta \\
\pi_1(\mu_L) = \mu_1 \land \pi_2(\mu_R) = \mu_2 \\
supp(\mu_L) \subseteq R \land supp(\mu_R) \subseteq R
\end{cases}
\]

\(\subseteq (A \times B)\)

- Admits an inductive characterization.
- For equivalence relations, it can be characterized as a closeness condition.

\[
\mu_1 \mathcal{L}_\alpha^\delta(R) \mu_2 \iff \Delta_\alpha(\mu_1/R, \mu_2/R) \leq \delta
\]

- For finite relations, it can be modeled as network-flow problem.
Generalized Data Processing Theorem

For any distribution transformer $h : D_A \to D_B$

$$\Delta_f (h(\mu_1), h(\mu_2)) \leq \Delta_f (\mu_1, \mu_2)$$
∀m₁, m₂ • m₁ Ψ m₂ ⇒ (m₁ {⟦e₁⟧ m₁/x₁}) Φ (m₂ {⟦e₂⟧ m₂/x₂}) [assn]

[∀m₁, m₂ • m₁ Ψ m₂ ⇒ Δf (⟦m₁⟧ m₁, ⟦m₂⟧ m₂) ≤ δ] [rand]

[∀m₁, m₂ • m₁ Ψ m₂ ⇒ b⟨1⟩ ≡ b′⟨2⟩] [cond]

([f₁, . . . , fₙ]) composable and monotonic

[Θ ⇔ b⟨1⟩ ≡ b′⟨2⟩] [while]

(f₁, f₂) is f₃-composable

[Ψ ⇒ Ψ′] [weak]
You need to:
- trust the type checker of Coq;
- trust the language semantics;
- make sure the security statement (a few lines in Coq) is as expected.

You don’t need to
- understand or even read the proof;
- trust program logics,
Case Study: $k$-Median Problem

Problem’s solution may leak the presence/absence of clients

Assume $k = 2$

Solution $= \{f_2, f_3\} \implies$ World 1
Solution $= \{f_1, f_2\} \implies$ World 2
Case Study: $k$-Median Problem

```plaintext
function $k$Median($C, F_0$)
1   $i \leftarrow 0$;
2   while $i < T$ do
3       $(x, y) \leftarrow \text{pick–swap}(F_i \times \overline{F_i})$;
4       $F_{i+1} \leftarrow (F_i \backslash \{x\}) \cup \{y\}$;
5       $i \leftarrow i + 1$
6   end;
7   $j \leftarrow \text{pick–solution}([1, \ldots, T], F)$;
8   return $F_j$
```

Each iteration of the loop (3-5) $\Rightarrow 2\epsilon\Delta$-DP
Selection of the solution (7) $\Rightarrow 2\epsilon\Delta$-DP

In our formalism,

$$\{\Psi\} \ k\text{Median} \sim_{\alpha, 0} \ k\text{Median} \ {\{\Phi\}}$$

$$C\langle 1 \rangle \simeq C\langle 2 \rangle \wedge F_0\langle 1 \rangle = F_0\langle 2 \rangle$$

$$e^{2\epsilon\Delta(T+1)}$$

$$F_j\langle 1 \rangle = F_j\langle 2 \rangle$$
Improving security bounds for Key-Alternating Cipher via Hellinger Distance.

\[ E_P(k, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n \]
The $f$-divergence between two distributions $\mu_1$ and $\mu_2$ over a set $A$ is defined as

$$\Delta_f (\mu_1, \mu_2) \triangleq \sum_{a \in A} \mu_2(a) f\left(\frac{\mu_1(a)}{\mu_2(a)}\right)$$

where $f : \mathbb{R}^{\geq 0} \to \mathbb{R}$ is a continuous convex function s.t. $f(1) = 0$.

Some examples

- Statistical distance ($\Delta_{SD}$) $\quad f(t) = \frac{1}{2} |t - 1|$
- Kullback-Leibler ($\Delta_{KL}$) $\quad f(t) = t \ln(t)$
- Hellinger distance ($\Delta_{HD}$) $\quad f(t) = \frac{1}{2} (\sqrt{t} - 1)^2$
- $\alpha$-distance ($\Delta_{\alpha}$) $\quad f(t) = \max\{t - \alpha, 0\}$
Indifferentiability

$F$ with access to a RO $h$ is $(t_S, q, \epsilon)$-indifferentiable from a RO $H$ if

$\exists S$ that runs in time $t_S$, $\forall D$ that makes at most $q$ queries,

$|\Pr[b \leftarrow D^{F,h} : b = 1] - \Pr[b \leftarrow D^{H,S} : b = 1]| \leq \epsilon$
Indifferentiability

$F$ with access to a RO $h$ is $(t_S, q, \epsilon)$-indifferentiable from a RO $H$ if

$\exists S$ that runs in time $t_S$, $\forall \mathcal{D}$ that makes at most $q$ queries,

$\left| \Pr[b \leftarrow \mathcal{D}^{F,h} : b = 1] - \Pr[b \leftarrow \mathcal{D}^{H,S} : b = 1] \right| \leq \epsilon$

In *any* secure cryptosystem, a random oracle $H$

*can be replaced with the construction $F$, which uses a random oracle $h$*
Indifferentiability

$F$ with access to a RO $h$ is $(t_S, q, \epsilon)$-indifferentiable from a RO $H$ if

$\exists S$ that runs in time $t_s$, $\forall D$ that makes at most $q$ queries,

$\left| \Pr \left[ b \leftarrow D^{F,h} : b = 1 \right] - \Pr \left[ b \leftarrow D^{H,S} : b = 1 \right] \right| \leq \epsilon$

In any secure cryptosystem, a random oracle $H$ into $EC(\mathbb{F}_p)$ can be replaced with the construction $F$, which uses a random oracle $h$ into $\mathbb{F}_p \times \mathbb{Z}_N$
Constructing Secure Hash Functions into Elliptic Curves (EC)
Constructing Secure **Hash Functions** into Elliptic Curves (EC)

- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
Constructing Secure **Hash Functions** into Elliptic Curves (EC)

- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
Constructing Secure Hash Functions into Elliptic Curves (EC)

Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
A Crypto Case Study

Constructing Secure **Hash Functions** into Elliptic Curves (EC)

- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
Constructing **Secure** Hash Functions into Elliptic Curves (EC)

- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
- Their output should “look like” uniformly distributed.
A Crypto Case Study

Constructing Secure Hash Functions into **Elliptic Curves** (EC)

- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
- Their output should "look like" uniformly distributed.
- Hash functions into elliptic curve allow an efficient implementation of some functionalities.
What is an elliptic curve?
Given a finite field $\mathbb{F}$ and two scalars $a, b \in \mathbb{F}$,

$$EC(\mathbb{F}) \triangleq \{(X, Y) \in \mathbb{F} \times \mathbb{F} \mid Y^2 = X^3 + aX + b\}$$
What is an elliptic curve?

Given a finite field \( \mathbb{F} \) and two scalars \( a, b \in \mathbb{F} \),

\[
EC(\mathbb{F}) \triangleq \{(X, Y) \in \mathbb{F} \times \mathbb{F} \mid Y^2 = X^3 + aX + b\} \cup \{O\}
\]

**Theorem:** the points in \( EC(\mathbb{F}) \) have a group structure.
What is an elliptic curve?
Given a finite field $\mathbb{F}$ and two scalars $a, b \in \mathbb{F},$

$$EC(\mathbb{F}) \triangleq \{(X, Y) \in \mathbb{F} \times \mathbb{F} \mid Y^2 = X^3 + aX + b\} \cup \{O\}$$

**Theorem:** the points in $EC(\mathbb{F})$ have a group structure.

How to securely hash into an elliptic curve $EC(\mathbb{F})$?
[Brier+ ’10]

$$H(m) = f(h_1(m)) \otimes g^{h_2(m)}$$

$\mathbb{F} \to EC(\mathbb{F})$  $\mathcal{M} \to \mathbb{F}$  $\mathcal{M} \to [1, \ldots, N]$
Indifferentiability from a Random Oracle

$H$ is called $\epsilon$-indifferentiable from a random oracle if and only if

$$\forall D \cdot \Delta_{SD}(D^H, D^{RO}) \leq \epsilon$$

Machine-checked version of Brier et al’s proof

- Equational theory for approximate observational equivalence (specialization of our Hoare logic) + adversary rule.
- Requires heavy algebraic reasoning (elliptic curves and group theory).
- 10,000+ lines of Coq proof-script.

With G. Barthe, B. Grégoire, S. Heraud, S. Zanella [POST ’12, JCS ’14]