DP-Sniper: Black-Box Discovery of Differential Privacy Violations using Classifiers

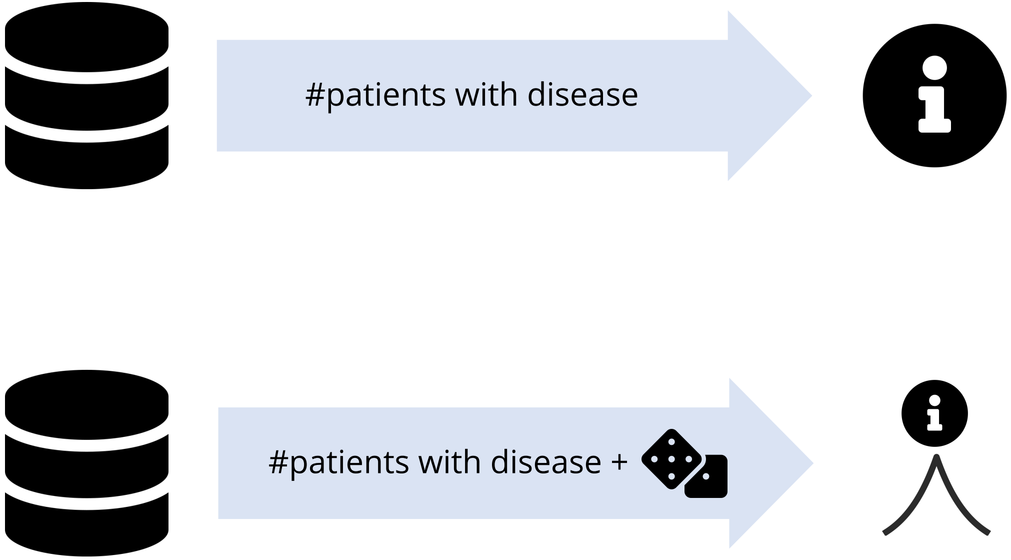
Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, Martin Vechev

Contact us via firstname.lastname@inf.ethz.ch

ETH zürich

SRILAB

Differential Privacy - Intuition



Floating-point vulnerability
Mironov, I. On significance of the least significant bits for differential privacy. CCS'12

```python
import numpy as np
epsilon = 0.1

def laplace_mechanism(n_with_disease):
  noise = np.random.laplace(scale=1/epsilon)
  return n_with_disease + noise
```

Icons: https://fontawesome.com

Detecting Floating-Point Vulnerabilities

```python
import numpy as np
epsilon = 0.1

def laplace_mechanism(n_with_disease):
  noise = np.random.laplace(scale=1/epsilon)
  return n_with_disease + noise
```

Existing DP verifiers

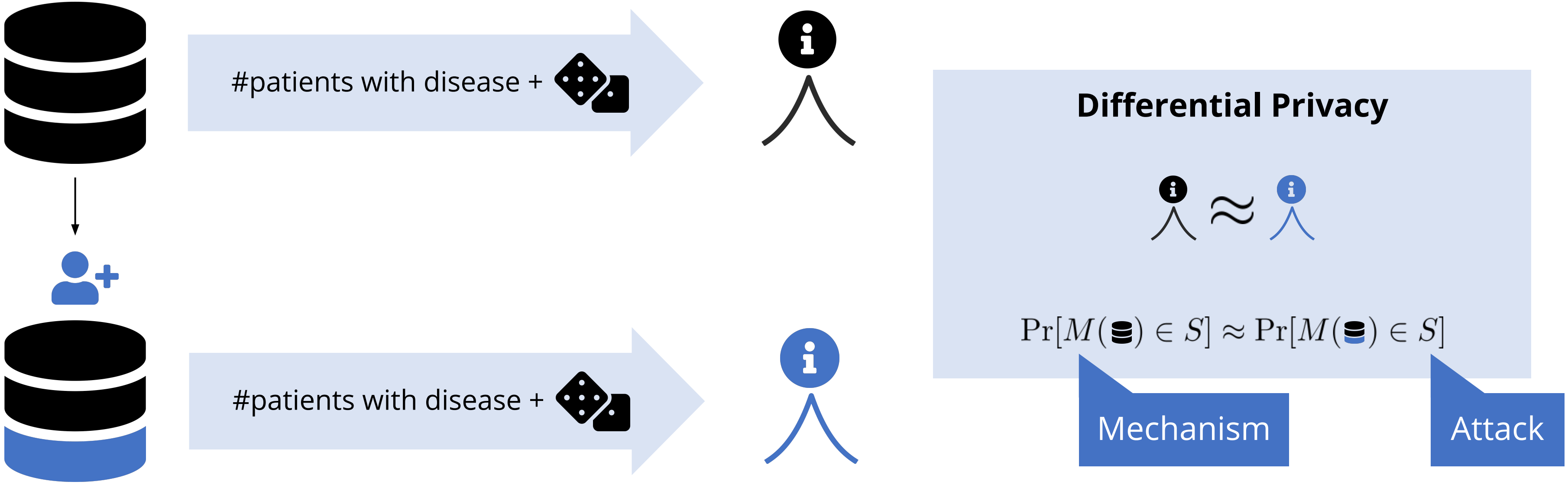
Existing DP testers

DP-Sniper (this work)

Not restricted to floating-point
Also covers other vulnerabilities

Icons: https://fontawesome.com

Differential Privacy



Icons: https://fontawesome.com

Differential Privacy

M is ε differentially private (ε-DP)

For all $(a, a') \in \mathcal{N}$ and for every attack S :

$$\ln(\Pr[M(a) \in S]) - \ln(\Pr[M(a') \in S]) \leq \epsilon$$

M is ξ differentially distinguishable (ξ-DD)

There exist $(a, a') \in \mathcal{N}$ and an attack S with:

$$\ln(\Pr[M(a) \in S]) - \ln(\Pr[M(a') \in S]) \geq \xi$$

Search Problem

Challenging

Needed for floating-point attack

$$\max_{(a,a')\in\mathcal{N}} \quad \max_{S} \quad \ln(\Pr[M(a)\in S]) - \ln(\Pr[M(a')\in S])$$

- Exhaustive
- Sampling
- Heuristics

Ding, Z., Wang, Y., Wang, G., Zhang, D. & Kifer, D. Detecting Violations of Differential Privacy. CCS'18

Icons: https://fontawesome.com

Finding an optimal attack

Target a small constant

Cannot quantify tiny probabilities accurately

$$\max_S \quad \ln(\Pr[M(a) \in S]) - \ln(\Pr[M(a') \in S])$$

$b \in S \iff \Pr[A = a \mid M(A) = b]$ is high (cp. Neyman-Pearson)

Train a classifier
Generate training data automatically

Icons: https://fontawesome.com

DP-Sniper Overview

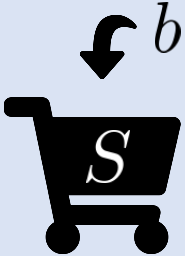
1

Train a classifier for

$$\Pr[A = a \mid M(A) = b]$$

Evaluation: Neural networks and logistic regression

2



Transform classifier to attack

$$b \in S \iff \Pr[A = a \mid M(A) = b] \geq t$$

3

Select t such that

$$\Pr[M(a') \in S] = c$$

Icons: https://fontawesome.com

Guarantees

Quantified mathematically

Theorem (informal): DP-Sniper finds an approximately optimal attack.

Assumptions

- Cannot estimate tiny probabilities
- The learned classifier is perfect

Degrades gracefully

Related Work

Tool	Black-box sufficient
T_{priv}	
StatDP	✓
DP-Finder	
DiPC	
DP-Stochastic-Tester	✓
CheckDP	
This work: DP-Sniper	✓

Only 1D outputs

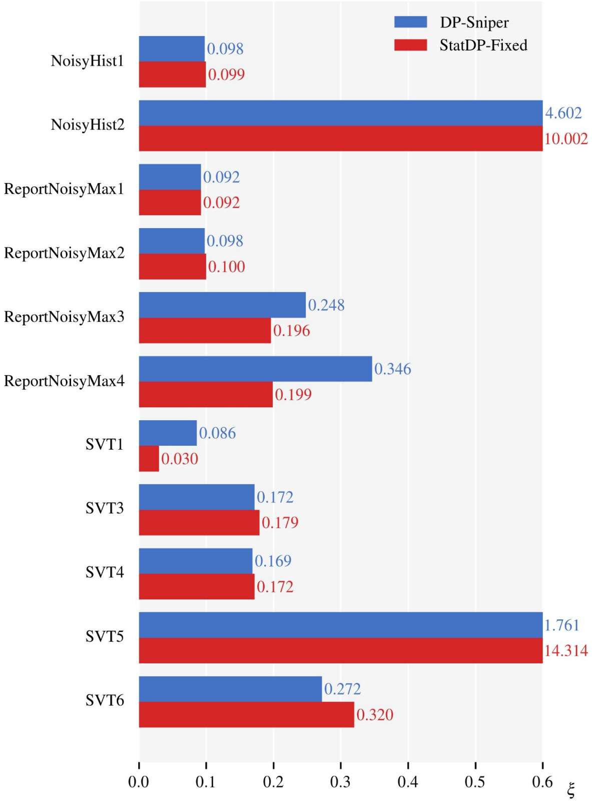
Black-box approaches are more convenient

And use floating-point arithmetic

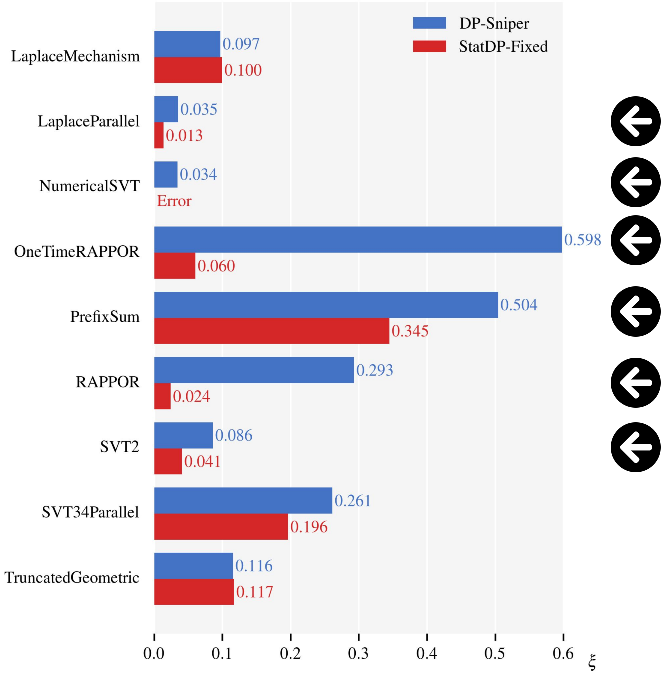
```
import numpy as np
epsilon = 0.1

def laplace_mechanism(n_with_disease):
  noise = np.random.laplace(scale=1/epsilon)
  return n_with_disease + noise
```

Icons: https://fontawesome.com

Evaluation



15x faster



Icons: https://fontawesome.com

Summary

New approach to discover DD

Code available
https://github.com/eth-sri/dp-sniper

Optimality guarantees

```
import numpy as np
epsilon = 0.1

def laplace_mechanism(n_with_disease):
  noise = np.random.laplace(scale=1/epsilon)
  return n_with_disease + noise
```

Icons: https://fontawesome.com