Estimating *g*-leakage via Machine Learning

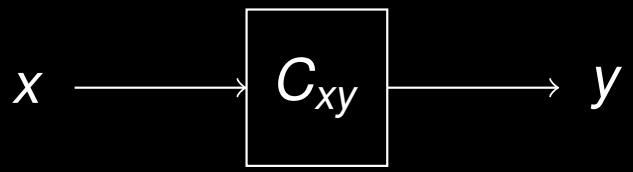
Marco Romanelli[1,2,3], Konstantinos Chatzikokolakis[4], Catuscia Palamidessi[1,2] & Pablo Piantanida[5,6,7]

[1] Inria, [2] Institut Polytechnique de Paris, [3] Università di Siena, [4] National and Kapodistrian University of Athens, [5] CentraleSupélec, [6] CNRS, [7] Université Paris Saclay

2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20).

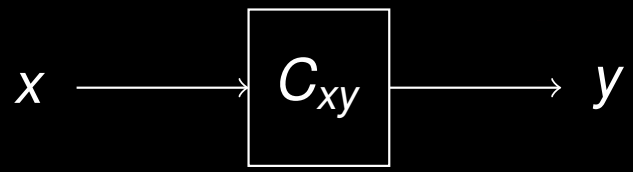
November 9-13, 2020

Motivation



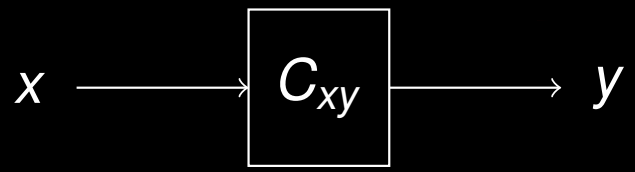
- Secret $x \in \mathcal{X}$

Motivation



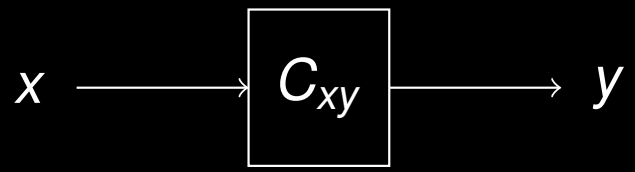
- Secret $x \in \mathcal{X}$
- Observable $y \in \mathcal{Y}$

Motivation



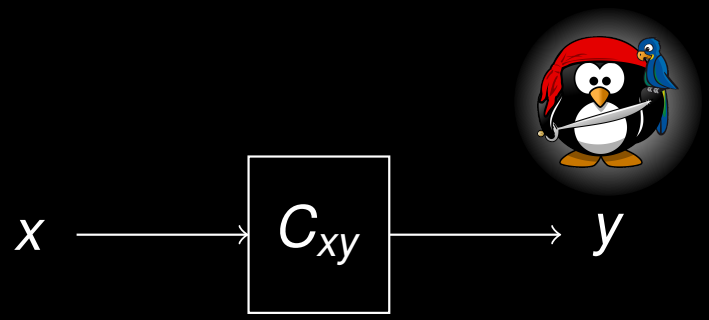
- Secret $x \in \mathcal{X}$
- Observable $y \in \mathcal{Y}$
- Channel C, such that $C_{xy} = P_{Y|X}(y|x)$

Motivation



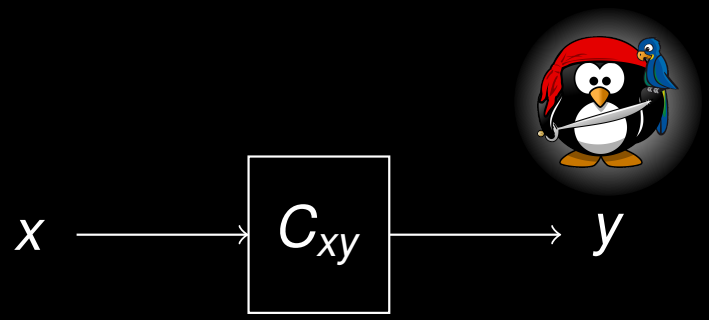
- Secret $x \in \mathcal{X}$
- Observable $y \in \mathcal{Y}$
- Channel C, such that $C_{xy} = P_{Y|X}(y|x)$
- Black-box scenario

Motivation



- Secret $x \in \mathcal{X}$
- Observable $y \in \mathcal{Y}$
- Channel C, such that $C_{xy} = P_{Y|X}(y|x)$
- Black-box scenario
- Security and privacy: information leakage is a measure of the robustness of a system to inference attacks

Motivation



- Secret $x \in \mathcal{X}$
- Observable $y \in \mathcal{Y}$
- Channel C, such that $C_{xy} = P_{Y|X}(y|x)$
- Black-box scenario
- Security and privacy: information leakage is a measure of the robustness of a system to inference attacks
- Can we use ML to estimate how much information this system leaks?

Previous approaches

F-BLEAU [Cherubin et al., 2019]: introduces the use of ML to model the attacker

Previous approaches

F-BLEAU [Cherubin et al., 2019]: introduces the use of ML to model the attacker

- 👍
  - It can deal with large alphabets for Y (also when Y follows a continuous distribution)

Previous approaches

F-BLEAU [Cherubin et al., 2019]: introduces the use of ML to model the attacker

- 👍
  - It can deal with large alphabets for Y (also when Y follows a continuous distribution)
  - ML models rely on universally consistent rules

Previous approaches

F-BLEAU [Cherubin et al., 2019]: introduces the use of ML to model the attacker

- 👍
  - It can deal with large alphabets for Y (also when Y follows a continuous distribution)
  - ML models rely on universally consistent rules
- 👎
  - It only uses k-NN to model the attacker

Previous approaches

F-BLEAU [Cherubin et al., 2019]: introduces the use of ML to model the attacker

- 👍
  - It can deal with large alphabets for Y (also when Y follows a continuous distribution)
  - ML models rely on universally consistent rules
- 👎
  - It only uses k-NN to model the attacker
  - It can only estimate Bayesian leakage

Previous approaches

F-BLEAU [Cherubin et al., 2019]: introduces the use of ML to model the attacker

- 👍
  - It can deal with large alphabets for Y (also when Y follows a continuous distribution)
  - ML models rely on universally consistent rules
- 👎
  - It only uses k-NN to model the attacker
  - It can only estimate Bayesian leakage

$\Longrightarrow$ We introduce neural network models and ***g*-leakage** estimation.

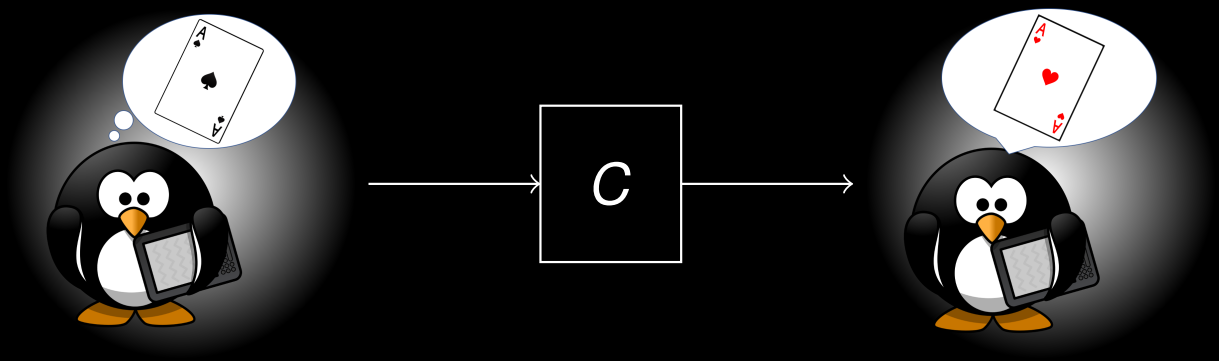
Why *g*-leakage?

set of actions (guesses)

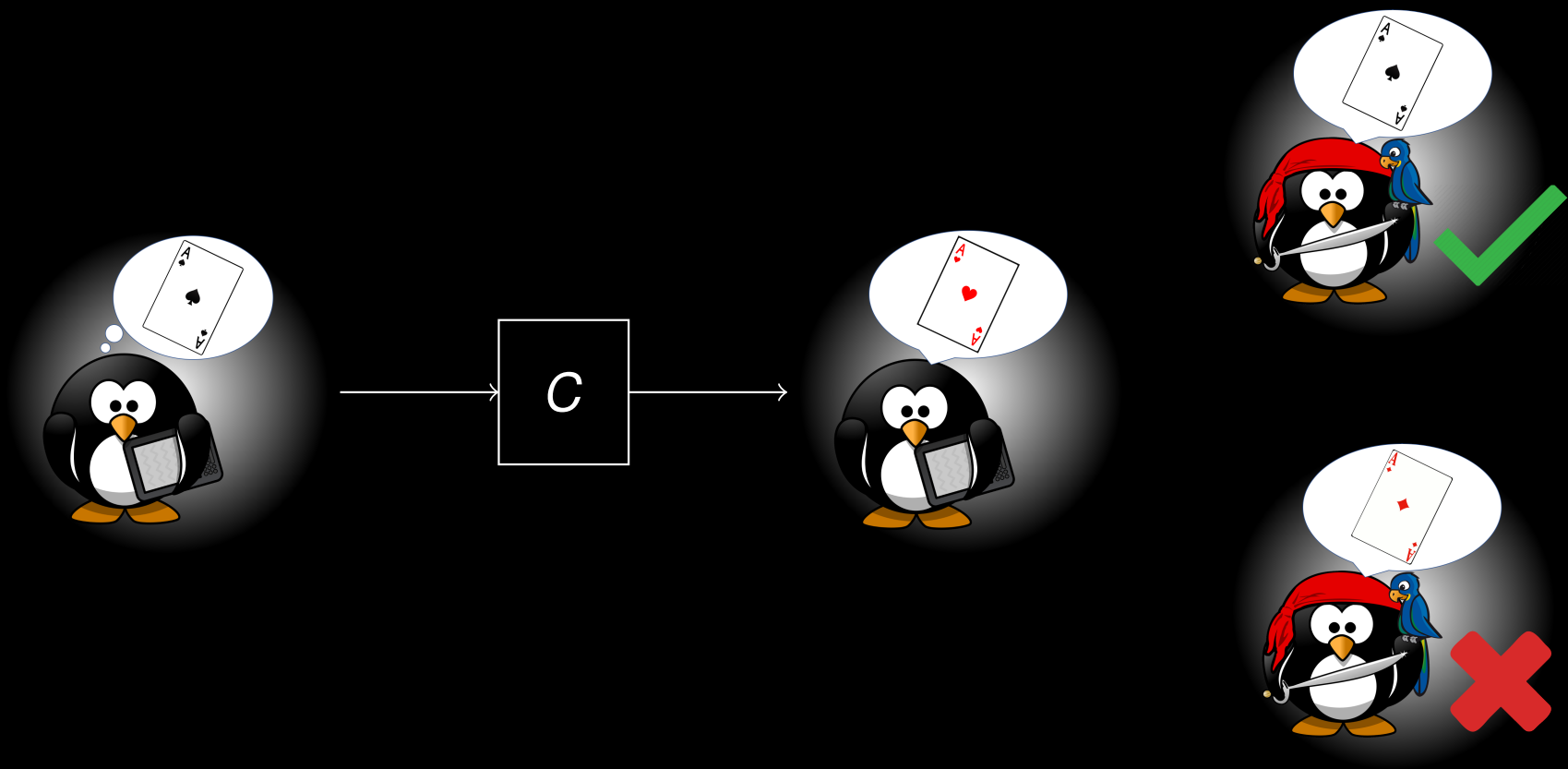
$$= \left\{ w \in \mathcal{W} \; ; \; g : \mathcal{W} \times \mathcal{X} \to [a, b] \text{ with } a, b \in \mathbb{R}^{+} \right\}$$

gain function

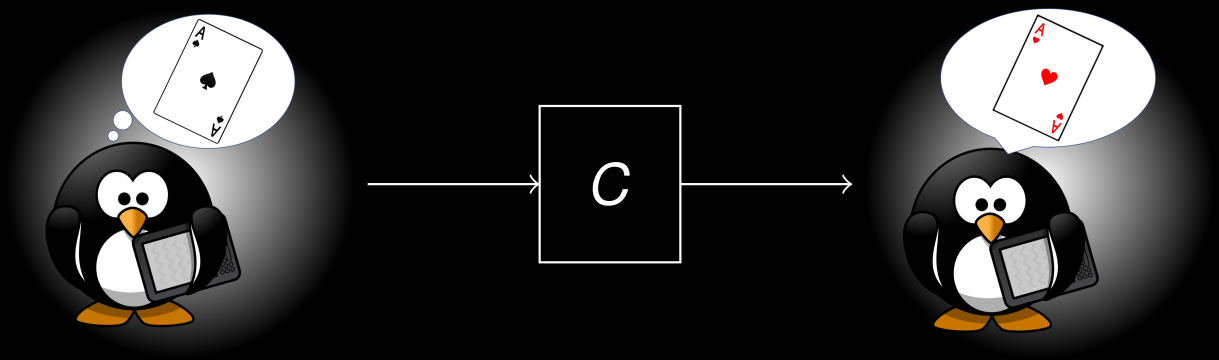
Example of attacks: Bayesian attack



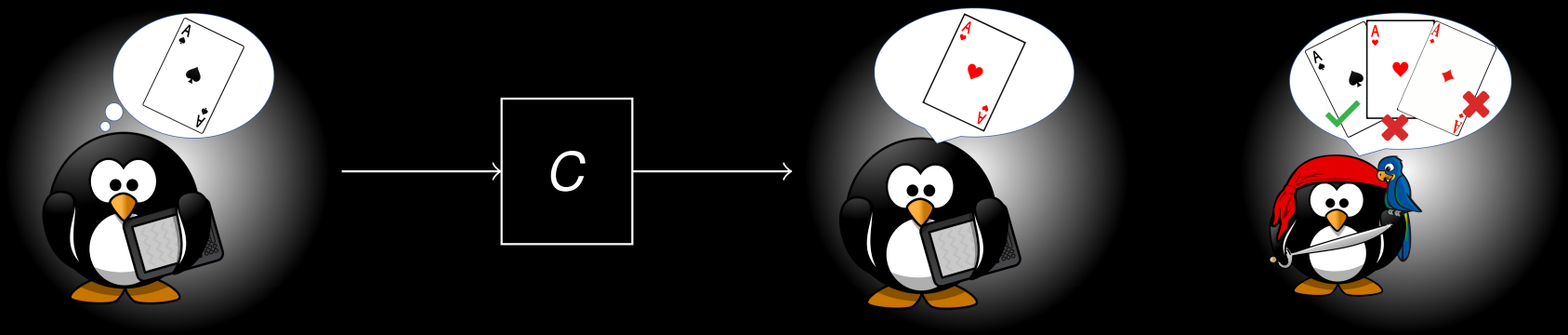
Example of attacks: Bayesian attack



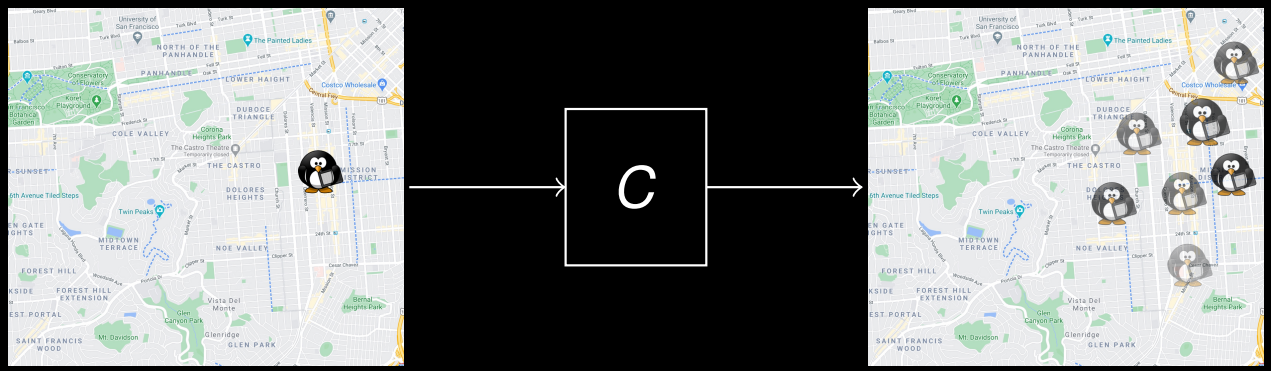
Example of attacks: multiple guesses attack



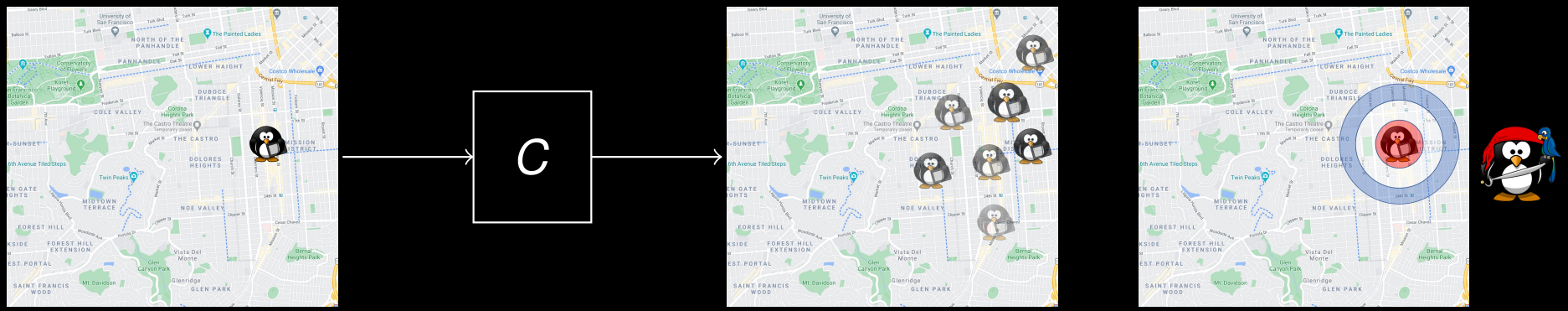
Example of attacks: multiple guesses attack



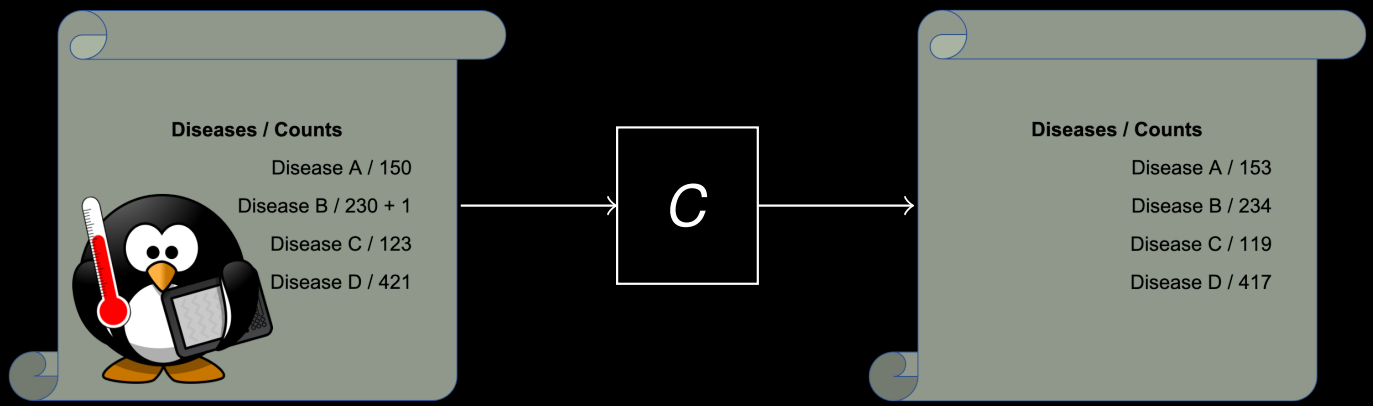
Example of attacks: location privacy attack



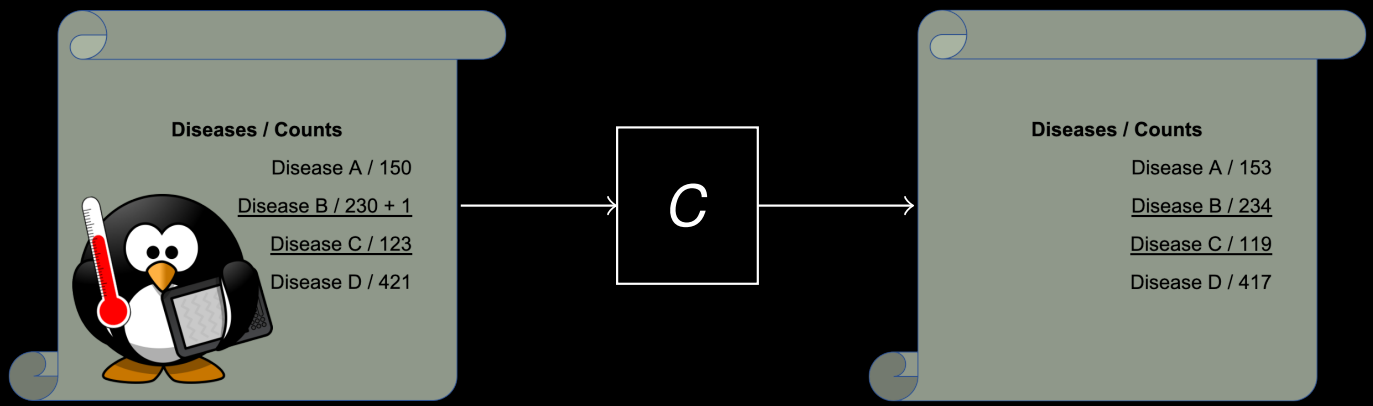
Example of attacks: location privacy attack



Example of attacks: attack against differential privacy



Example of attacks: attack against differential privacy



Estimating g-vulnerability from samples

Learnability result: We can upper-bound the expected estimation error

$$\mathbb{E}|V_g - \widehat{V}_n(f^\star_m)|$$

Estimating g-vulnerability from samples

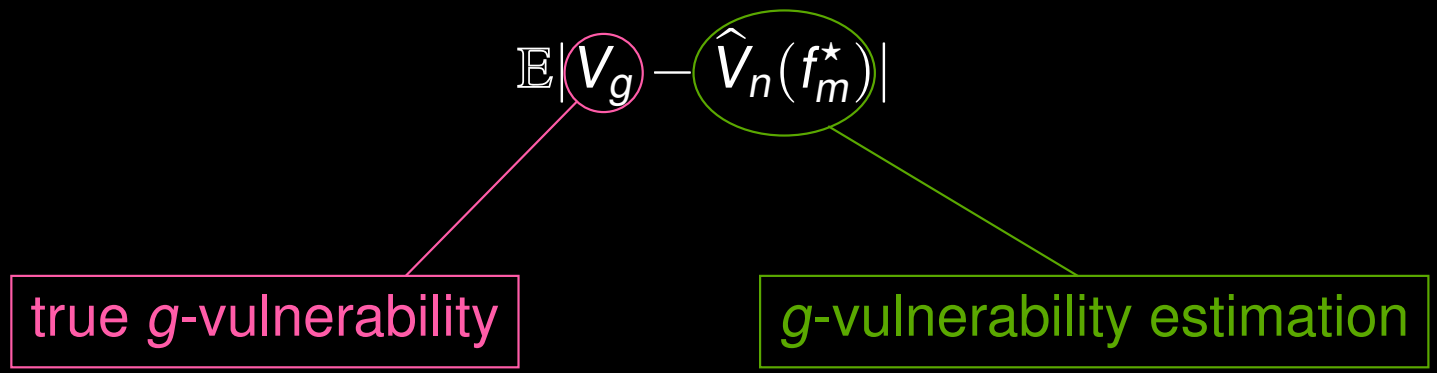
Learnability result: We can upper-bound the expected estimation error

$$\mathbb{E}\left|V_g - \widehat{V}_n(f^\star_m)\right|$$

true g-vulnerability

Estimating g-vulnerability from samples

Learnability result: We can upper-bound the expected estimation error



Estimating *g*-vulnerability from samples

strongest attacker model with m samples

Learnability result: We can upper-bound the expected estimation error

$$\mathbb{E}\left|V_g - \widehat{V}_n(f^\star_m)\right|$$

true g-vulnerability

g-vulnerability estimation

Estimating *g*-vulnerability from samples

- Attacker model

$$f : \mathcal{Y} \to \mathcal{W}$$

Estimating g-vulnerability from samples

- Attacker model

$$f : \mathcal{Y} \to \mathcal{W}$$

- We need samples (w, y)

Estimating g-vulnerability from samples

- Attacker model

$$f : \mathcal{Y} \to \mathcal{W}$$

- We need samples (w, y)
- We can collect samples (x, y) from the system

Estimating g-vulnerability from samples

- Attacker model

$$f : \mathcal{Y} \to \mathcal{W}$$

- We need samples (w, y)
- We can collect samples (x, y) from the system

How do we obtain the samples we need from the system?

Pre-processing: obtaining samples (w, y)

- Data pre-processing
- Channel pre-processing

Pre-processing: obtaining samples (w, y)

- Data pre-processing
  - No interactions with the channel needed (third party data collection)
  - Knowing $g(\cdot, \cdot)$, for each couple (x, y) in the dataset we produce a number of couples (w, y) which is proportional to $g(w, x)$

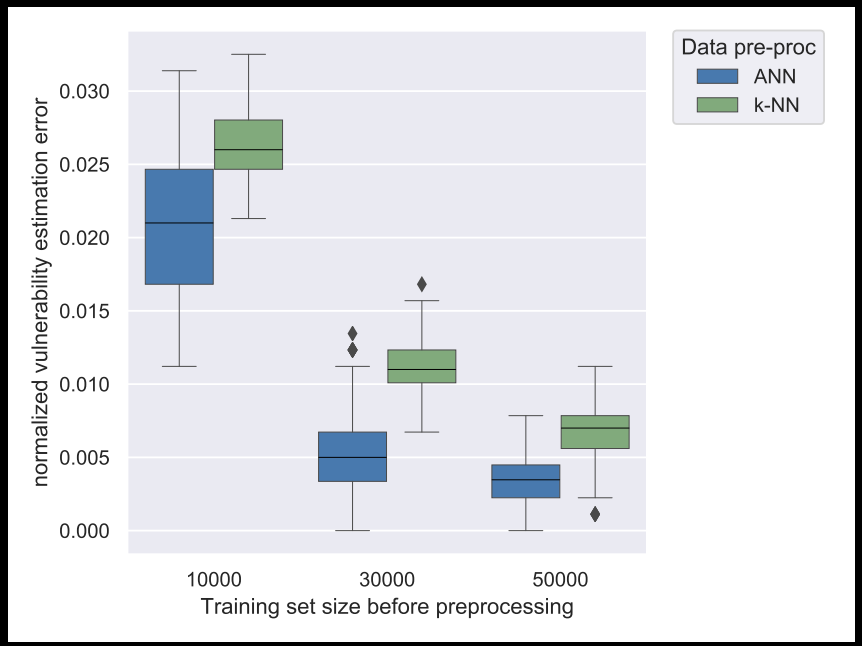
Pre-processing: obtaining samples (w, y)

- Data pre-processing
  - No interactions with the channel needed (third party data collection)
  - Knowing $g(\cdot, \cdot)$, for each couple (x, y) in the dataset we produce a number of couples (w, y) which is proportional to $g(w, x)$
- Channel pre-processing
  - Black-box interaction with the channel via queries
  - Knowing $g(\cdot, \cdot)$ and the secrets' distribution
    - $\xi : \mathbb{D}\mathcal{W}$
    - Channel $R = \mathcal{P}_{X|W}$
    - Extract samples (w, y) according to $\xi \triangleright RC$

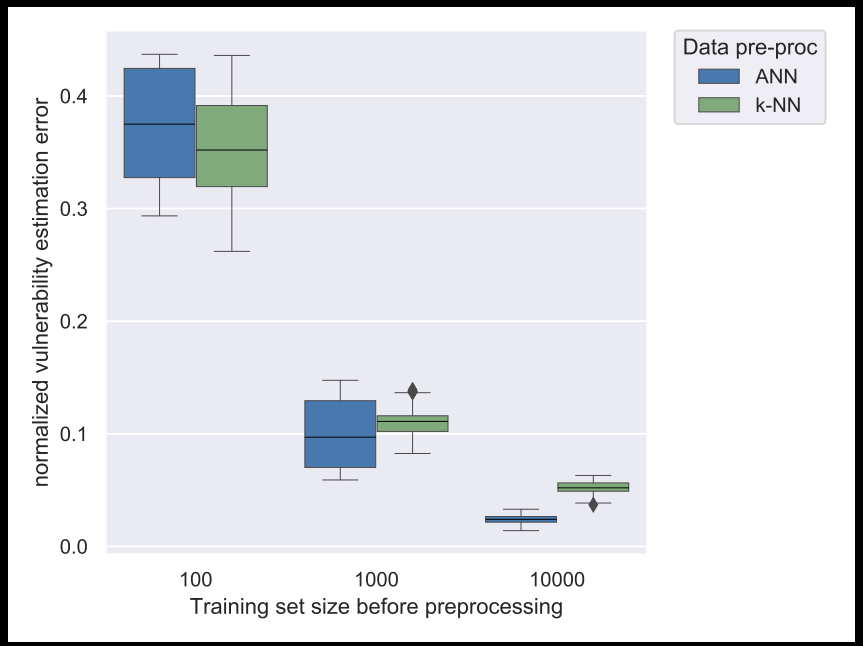
Pre-processing: obtaining samples (w, y)

- Data pre-processing
  - No interactions with the channel needed (third party data collection)
  - Knowing $g(\cdot, \cdot)$, for each couple (x, y) in the dataset we produce a number of couples (w, y) which is proportional to $g(w, x)$
- Channel pre-processing
  - Black-box interaction with the channel via queries
  - Knowing $g(\cdot, \cdot)$ and the secrets' distribution
    - $\xi : \mathbb{D}\mathcal{W}$
    - Channel $R = \mathcal{P}_{X|W}$
    - Extract samples (w, y) according to $\xi \triangleright RC$

Experimental results



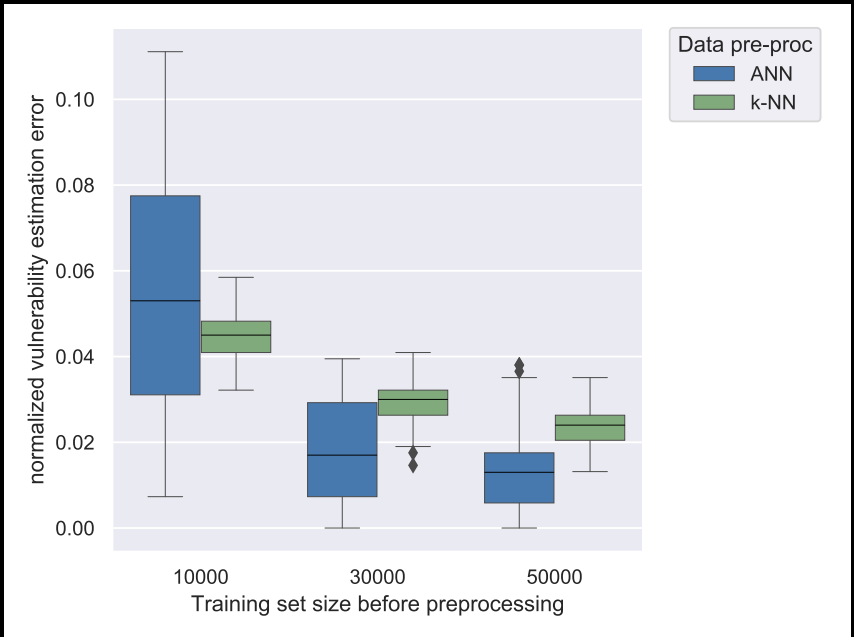
(a) Multiple guesses



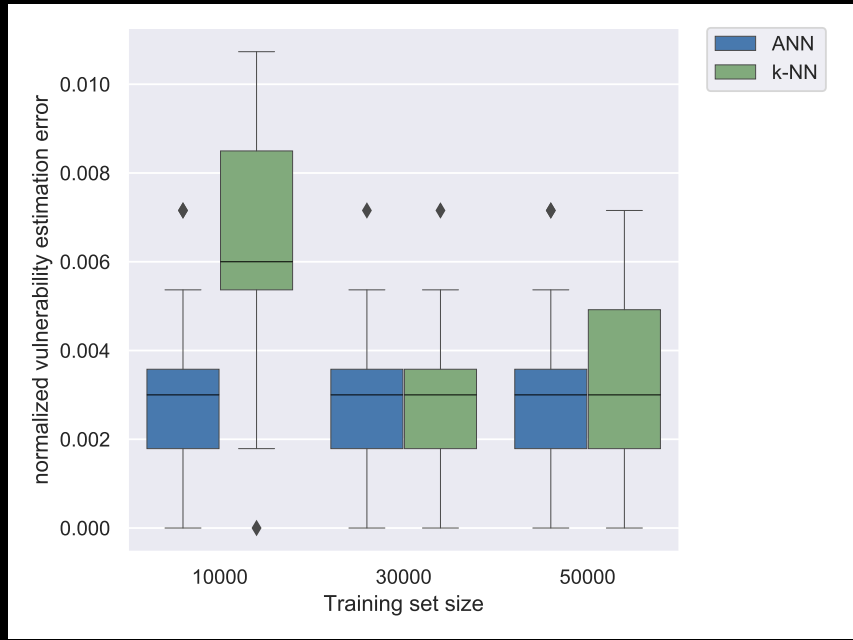
(b) Location privacy: Gowalla dataset

Figure: Normalized estimation error between V_g and the estimation $\widehat{V}_n(f_m)$.

Experimental results



(a) Differential privacy: Cleveland heart desease dataset



(b) Side channel attack

Figure: Normalized estimation error between V_g and the estimation $\widehat{V}_n(f_m)$.

Summary

- We have proposed the use of ML to estimate ***g*-leakage**, a very powerful and flexible notion of information leakage, using samples collected from a system in the **black-box scenario**.

Summary

- We have proposed the use of ML to estimate ***g*-leakage**, a very powerful and flexible notion of information leakage, using samples collected from a system in the **black-box scenario**.
- We have studied the problem of modeling attacks using ML algorithms and we have analyzed the **statistical guarantees on the precision of the estimation**.

Summary

- We have proposed the use of ML to estimate ***g*-leakage**, a very powerful and flexible notion of information leakage, using samples collected from a system in the **black-box scenario**.
- We have studied the problem of modeling attacks using ML algorithms and we have analyzed the **statistical guarantees on the precision of the estimation**.
- We have introduced two **pre-processing** methods to obtain samples that can be used to model several different attackers applying state-of-the-art ML algorithms.

Summary

- We have proposed the use of ML to estimate ***g*-leakage**, a very powerful and flexible notion of information leakage, using samples collected from a system in the **black-box scenario**.
- We have studied the problem of modeling attacks using ML algorithms and we have analyzed the **statistical guarantees on the precision of the estimation**.
- We have introduced two **pre-processing** methods to obtain samples that can be used to model several different attackers applying state-of-the-art ML algorithms.
- We have applied our framework to many different scenarios, analyzing its performance and showing favorable results.

Thank you.

Some elments in the pictures are courtesy of https://www.stockio.com/