Notes

- Call for actions
  - 11/09 Lecture: Recording will be offered (on 11/16)
  - 11/20: Checkpoint II review deadline (on HotCRP)

CS 499/579: Trustworthy ML
Preliminaries on privacy

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

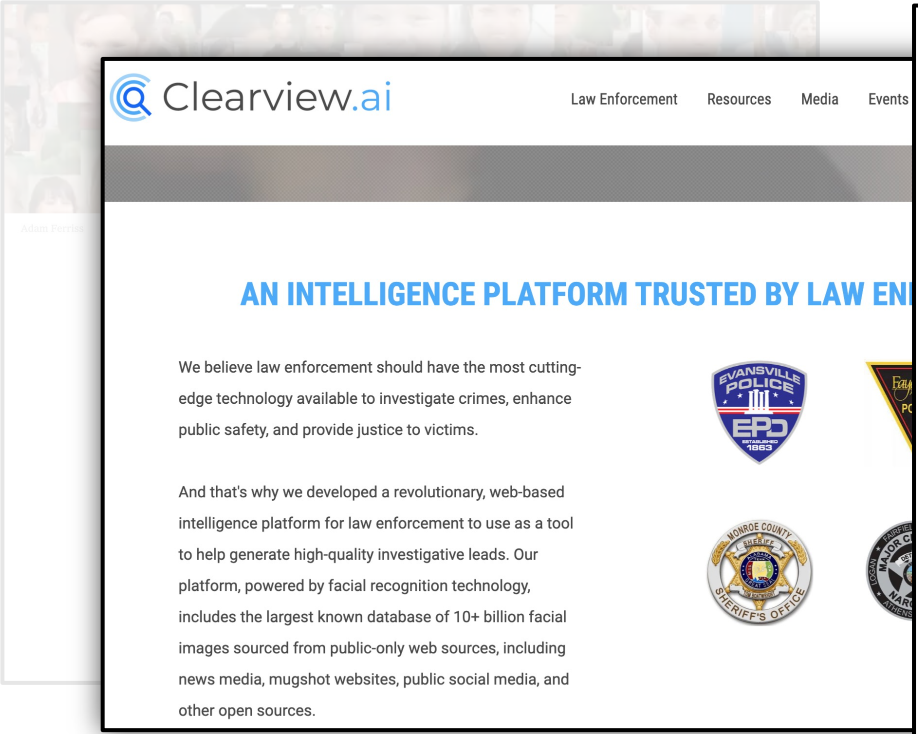
sanghyun.hong@oregonstate.edu

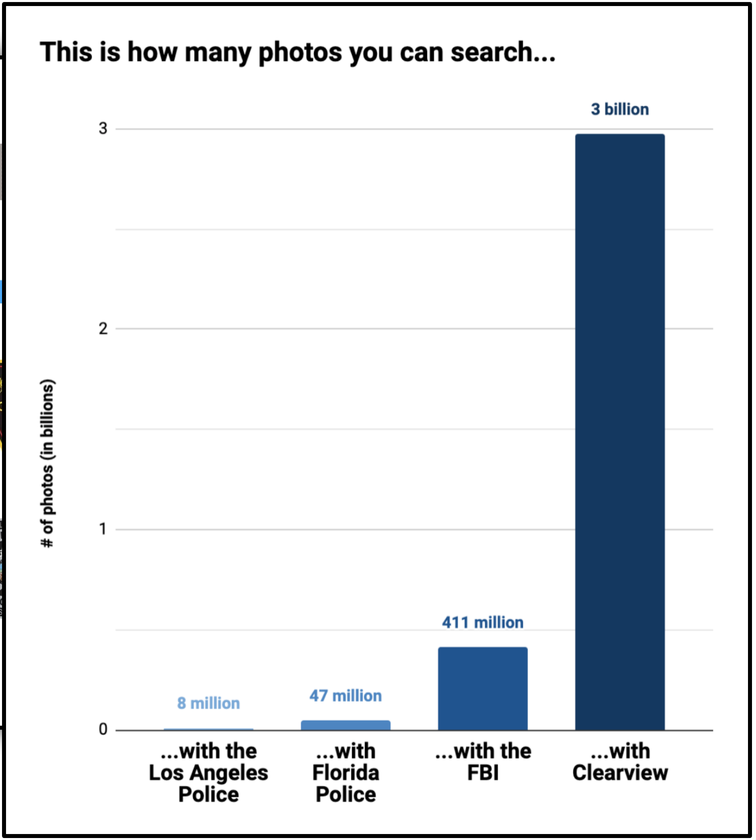
Oregon State University

SAIL
Secure AI Systems Lab

Privacy, privacy, privacy…

Why privacy matters?





[1]https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html
[2]https://www.muckrock.com/news/archives/2020/jan/18/clearview-ai-facial-recogniton-records/

Why privacy matters?

- Let’s discuss
  - What is privacy?
  - What does privacy matter?
  - How is it different from security?

How can we make it private?

- A perfect, yet not interesting solution:
  - No learning … but this is *not* what we want
  - Hold-on, what if we anonymize some records?

De-anonymization

- Setup
  - **Attacker:** de-anonymize anonymized records
  - **Victim** : anonymize sensitive data records

- Knowledge
  - Additional (or auxiliary information) about the data

- Capability
  - Query your data with some techniques
  - Perform post-processing computations on q (outputs)
  - … (many more)

President's Council of Advisors on Science and Technology, 2014

De-anonymization

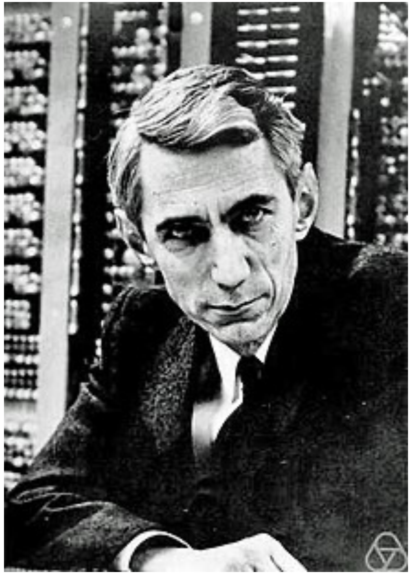
- In ML
  - We train statistical models
  - It does not matter whether data is anonymized or not
  - Some examples
    - Cancer data
    - Demographics
    - Data about people's financial information
    - …

- Note:
  - "Anonymization of a data record might seem easy to implement. Unfortunately, it is *increasingly easy to defeat* anonymization by the very techniques that are being developed for many legitimate applications of big data." [1]

[1] President's Council of Advisors on Science and Technology, 2014
Narayanan and Shmatikov, Robust De-anonymization of Large Sparse Datasets, IEEE S&P 2008

How can we make it private?

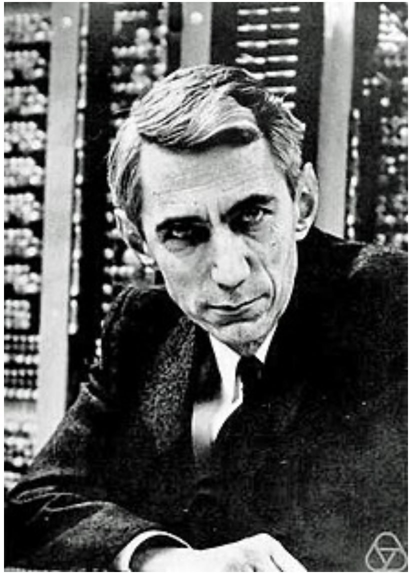
- Shannon’s perfect security
  - An adversary should not distinguish a message M from a random text R



Claude Shannon (1916 ~ 2001)
A Father of Information Theory and Modern Cryptography

How can we make it private?

- Shannon's perfect security
  - An adversary should not distinguish a message M from a random text R
  - Formally:
    - $Pr[M = m | C = c] = Pr[M = m]$
    - where
      - m is a message (from a set M)
      - c is a ciphertext (from a set of all ciphertexts C)
    - $Pr[C = c | M = m] = Pr[C = c]$
  - It means:
    - Ciphertext provides no additional information
    - Observing c does not help with guessing M = m
    - c is independent of the message m



Claude Shannon (1916 ~ 2001)
A Father of Information Theory
and Modern Cryptography

How can we make it private?

- Perfect security in model training

Training Data
$(x, y) \in D_{tr}, x \in R^d$

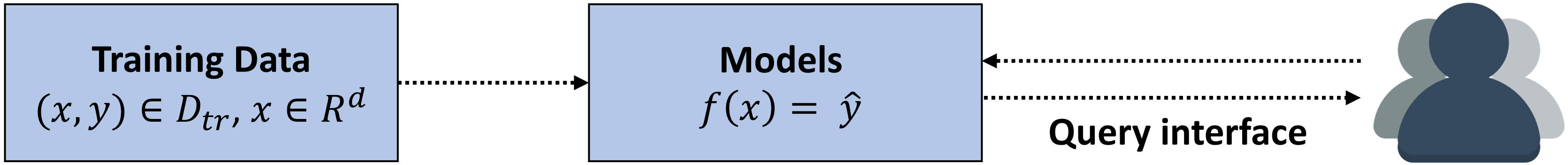
$\rightarrow$

Models
$f(x) = \hat{y}$

- Potential solutions:
  - **Encrypt-decrypt:** encrypt the training data and decrypt it to train a model
  - **Homomorphic encryption:** encrypt the training data and train a model on it
  - …

How can we make it private?

- Inferences with such model(s)



- Potential problems:
  - Perfect security-based solutions are computationally expensive (than vanilla training)
  - Only a limited number of users (who has a key) may use these models

How can we make it private?

- Inferences with such model(s)

Training Data
$(x, y) \in D_{tr}, x \in R^d$

Models
$f(x) = \hat{y}$

Query interface

- Potential problems:
  - Perfect security-based solutions are computationally expensive (than vanilla training)
  - Only a limited number of users (who has a key) can use these models
  - Once a key is leaked, an adversary can query the model with any data

What an adversary can do with the query access?

Privacy threat model

- ML Pipeline

Training Data
$(x, y) \in D_{tr}, x \in R^d$

Models
$f(x) = \hat{y}$

Query interface

- Privacy risks
  - Identify your membership in the training data
  - Identify (sensitive) properties of your training data
  - Identify (sensitive) attribute of a person that you know
  - Reconstruct a sample completely
  - Reconstruct a model behind the query interface
  - …

Privacy threat model

- ML Pipeline

Training Data
$(x, y) \in D_{tr}, x \in R^d$

→

Models
$f(x) = \hat{y}$

⇄ **Query interface**

- Privacy risks (from the view of the work by Dwork *et al.*)
  - Tracing attack : Identify your membership in the training data
  - Reconstruction : Identify (sensitive) properties of your training data
  - De-anonymization: Identify (sensitive) attribute of a person that you know
  - Reconstruction : Reconstruct a sample completely
  - Reconstruction : Reconstruct a model behind the query interface
  - …

Dwork *et al.*, Exposed! A Survey of Attacks on Private Data

Privacy threat model

- The attack considers non-trivial cases
  - ex. Smoking causes cancer
  - Revealing this information is *not* a privacy attack
  - We know this is correlated without interacting with the target model

  - ex. A model trained on a dataset of lung cancer patients
  - ex. The model gets a patient information and returns the probability of getting the cancer
  - ex. We know the Person A is smoking
  - ex. We identify that A is in the dataset (defer the details to later on)
  - It’s a *non-trivial* attack as we identify the information about an individual

Membership inference: tracing

- Setup
  - **Victim:**
    - Has a dataset x = $\{x_1, \ldots, x_n\}$ with n-i.i.d samples where each x_i is drawn from P over $\{\pm 1\}^d$
    - For each query M, the victim returns the sample mean q over given sample x_i's
  - **Attacker:**
    - Perform an attack $A(y, q, z)$ that identify whether a target instance $y \in \{\pm 1\}^d$ **IN** the dataset x or not (**OUT**) with m-i.i.d reference samples z = $\{z_1, \ldots, z_n\}$ and the sample mean q
  - **Procedure:**

Reconstruction attack I: attribute inference

- Setup
  - **Victim:**
    - For each i-th instance, the victim has (x_i, s_i) information
    - $x_i \in \{0,1\}^d$: public info. accessible by an adversary and s_i: is the one-bit secret
  - **Attacker:**
    - Perform an attack A that reconstructs s_i by exploiting query outputs $\hat{q}$ and the public information $A(x, M(x,s))$, where the attacker knows $k > 1$ public attributes
  - **Formally**

Reconstruction attack I: attribute inference

- Setup
  - **Victim:**
    - For each i-th instance, the victim has (x_i, s_i) information
    - $x_i \in \{0, 1\}^d$: public info. accessible by an adversary and s_i: is the one-bit secret
  - **Attacker:**
    - Perform an attack A that reconstructs s_i by exploiting query outputs $\hat{q}$ and the public information $A\big(x, M(x, s)\big)$, where the attacker knows $k > 1$ public attributes
  - **Approximation:**
    - Linear statistics (*e.g.*, linear SVM, linear regression, …)
    - Practical constraints (# Queries)
      - Ideally 2^n queries to solve the subset-sum problem
      - Practically, considering the tradeoff btw error and accuracy, we can do it in polynomial time

Reconstruction attack II: model extraction

- Setup
  - **Victim:**
    - Has a model $f(x) = y$ trained on a confidential data
    - For each query M, the victim returns the output y_i over given sample x_i's
  - **Attacker:**
    - Perform an attack (i.e., trains a surrogate model f' that is functionally equivalent to f

Tramer *et al.*, Stealing Machine Learning Models via Prediction APIs, USENIX 2016

Thank You!

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

https://secure-ai.systems/courses/MLSec/F23

Oregon State University

SAIL
Secure AI Systems Lab