How to allow deep learning on your data without revealing your data

Yangsibo Huang, Zhao Song, Kai Li, Sanjeev Arora




PRINCETON UNIVERSITY

InstaHide: Instance-Hiding schemes for Private Distributed Deep Learning ICML'20

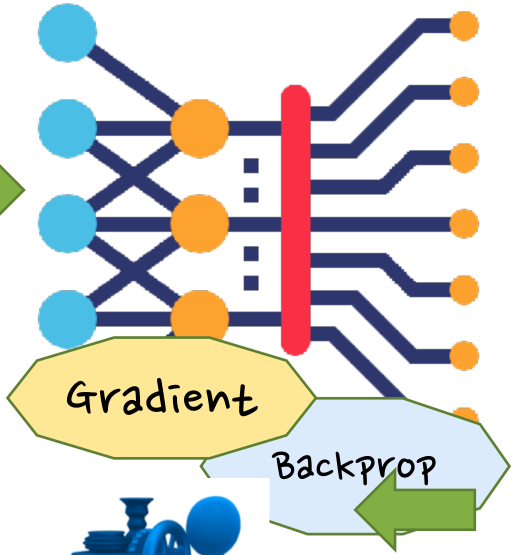
TextHide: Tackling Data Privacy in Language Understanding Tasks EMNLP-Findings'20 (+ Danqi Chen)

Today’s Faustian Bargain:
“Hand over your data, enjoy a world customized for you.”

Can deep learning be done on our data without making us reveal the data?

OUR data

Deep Net



Hospitals training deep net on pooled patient data.

Customizing Gboard for user groups using their chats.

Privacy-preserving training and customization for IoT (home devices, self-driving cars,)…

TWO DISTINCT SETTINGS

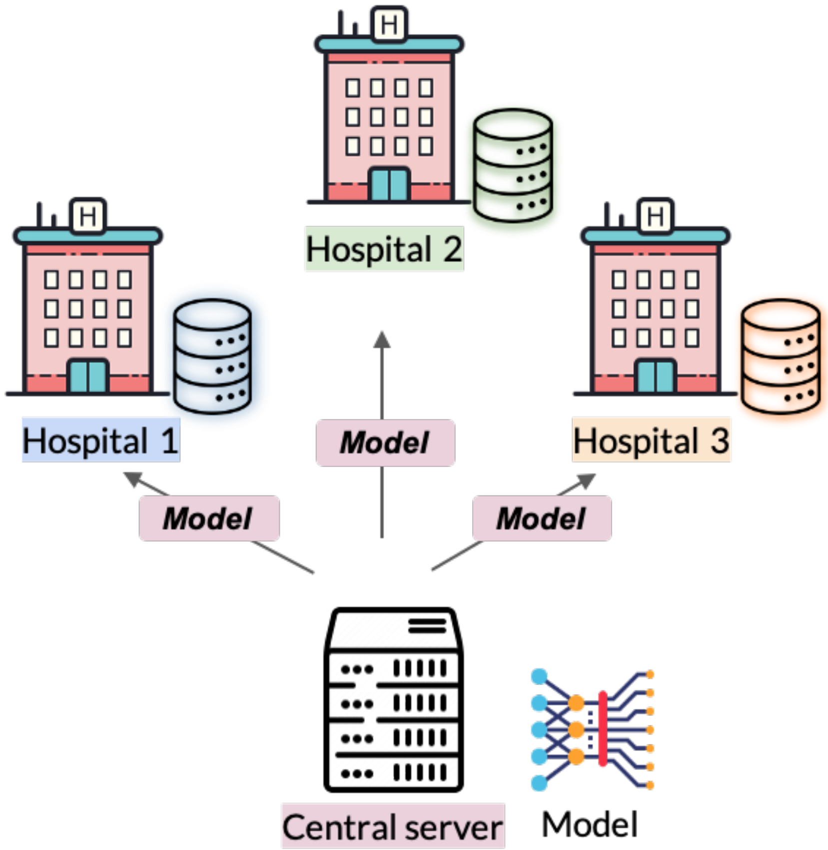
- Clients (e.g. hospitals) using private data to collaboratively train deep net on server

- Large number of lightweight devices (e.g. IoT) sending user data to servers for doing deep learning towards a desired goal

(We address the first setting, but solution also applicable to the second.)

FEDERATED LEARNING FRAMEWORK

[McMahan et al 16]



Hold on to your data and participate in training

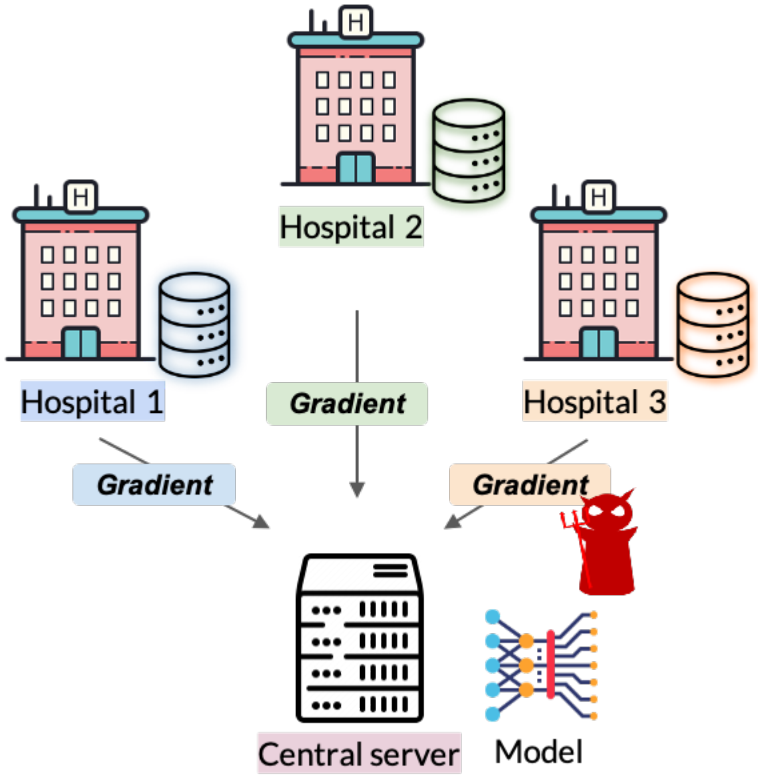
Each iteration:

Users: Compute model/net updates (gradients) w/ private data and share with server.

Server: Update model (net) using pooled gradients and share.

FEDERATED LEARNING FRAMEWORK

[McMahan et al 16]



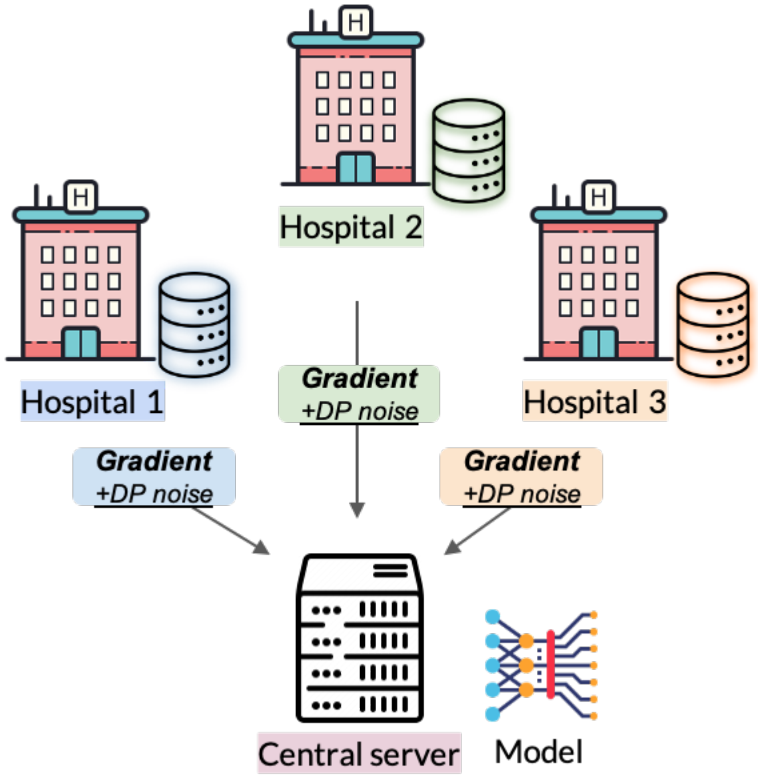
Users: Compute model/net updates (gradients) w/ private data and share with server.

Server: Update model (net) using pooled gradients and share.

Privacy leakage! Using gradient-matching, attackers can reverse-engineer private input from shared gradients [Zhu et al' 19]. (* if batch sizes are small)

[Geiping et al '20] attack works for realistic batch sizes

PAST APPROACH 1: DIFFERENTIAL PRIVACY



Users: Compute model/net updates (gradients) w/ private data and share with server.

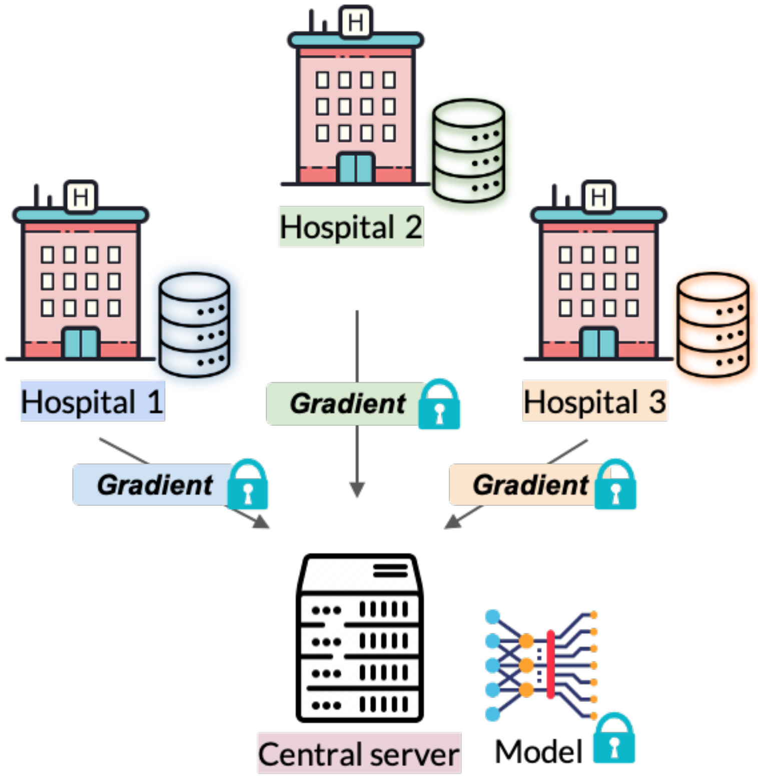
Server: Update model (net) using pooled gradients and share.

Differential privacy (DP): Add noise to gradient; carefully adjust noise to allow upper bound on "privacy loss." [Abadi et al'16]

DP shortcomings:

a) Big accuracy drop (e.g., 20% on CIFAR10; Huge drop on ImageNet)
b) Only concerned with "privacy loss" due to release of trained model (i.e., "proper use"). No guarantees about side computations on shared gradients (e.g., gradient-matching attacks[Zhu et al'19]).

PAST APPROACH 2: CRYPTOGRAPHY



Possible to compute on encrypted data by decomposing into atomic operations (e.g., secure multi-party protocol [Yao82, GMW87], fully homomorphic encryption [Gentry 09])

Crypto shortcomings:

a) BIG efficiency loss. Every arithmetic operation done securely…
b) Needs finite field arithmetic, special setups (eg public-key infrastructure)

Outline for rest of the talk

1. InstaHide encryption. Uses Subset-sum like encryption to encrypt images so that encryptions can be used directly in deep learning.

2. TextHide: adaptation of the idea to text data.

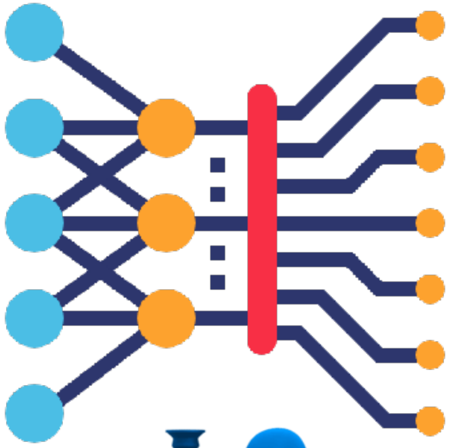
3. Discussion of security

INSTAHIDE ENCRYPTION FOR DATA

Data

InstaHide Encryption

Deep Net



Training Unchanged!

Trains and tests on encrypted images.

- Minor effect on final accuracy
- Almost no effect on efficiency
- Reveals nothing* about data

** violating privacy requires solving computationally difficult problem (analogous to security guarantee in today's e-commerce)*

INSTAHIDE: INSPIRED BY MIXUP

0.6 x  + 0.4 x  = 

(0, **1**, 0, 0)
Bird

(0, 0, 0, **1**)
Airplane

(0, **0.6**, 0, **0.4**)
Bird Airplane

Training the net to behave linearly??

* Mixup Data augmentation [Zhang et al'18]

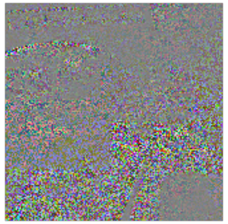
INSTAHIDE: HOW IT WORKS

Mix 2 private training images with k-2 public images, followed by pixelwise random sign flip

0.6 x  + 0.4 x  + 0.3 x  + 0.7 x 

(0, **1**, 0, 0)
Bird

(0, 0, 0, **1**)
Airplane

Flip pixel signs randomly → 

(0, **0.6**, 0, **0.4**)
Bird Airplane

$\underbrace{\qquad\qquad\qquad}_{\text{Private training set}}$

Private training set

Large Public dataset
(e..g. ImageNet)

1. Public → off-the-shelf
2. Large → gives more se

Carlini et al'20 raises some doubts (coming up later)

Conjecture (based upon intuition from VECTOR SUBSET SUM): Extracting significant info about private images from gradients of encrypted images takes N^{k-2} time. (N = size of public data set).

Private Encryption key = (Choice of images used for mixing, coefficients, random sign mask)
Never reused during training

INSTAHIDE: MINOR IMPACT ON ACCURACY

Test accuracy (%) on image classification benchmarks.

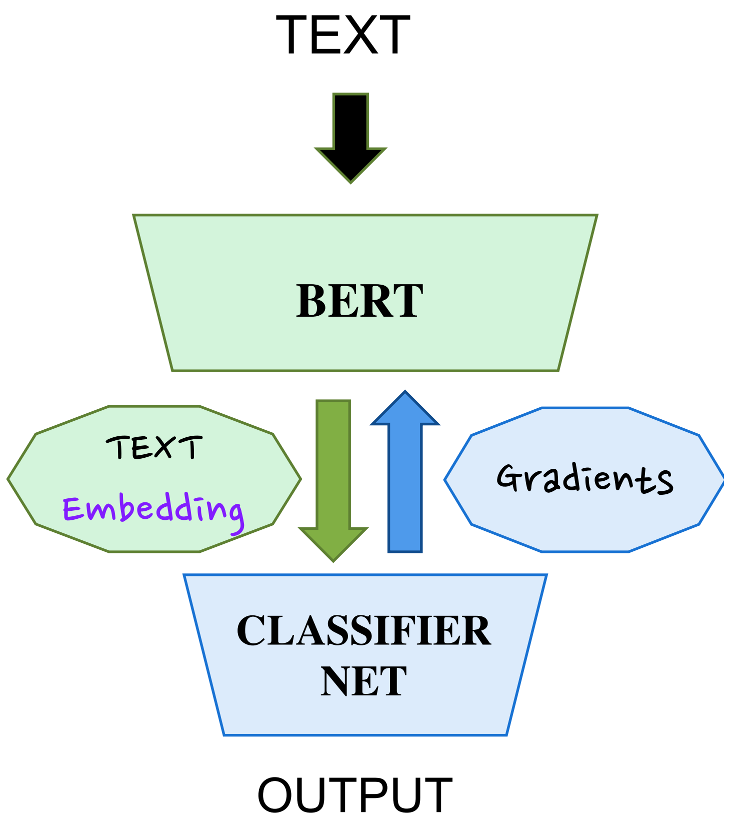
	MNIST	CIFAR-10	CIFAR-100	ImageNet
Vanilla training	99.5	94.8	77.9	77.4
Diff. Privacy SGD* [Papernot et al 19]	98.1	72.0	-	
InstaHide* *(no public dataset)	98.2	92.3	74.5	72.6
InstaHide* *(with public dataset)	97.8	90.3	73.1	

*DP has different notion of privacy from *InstaHide*

TEXTHIDE: BACKGROUND

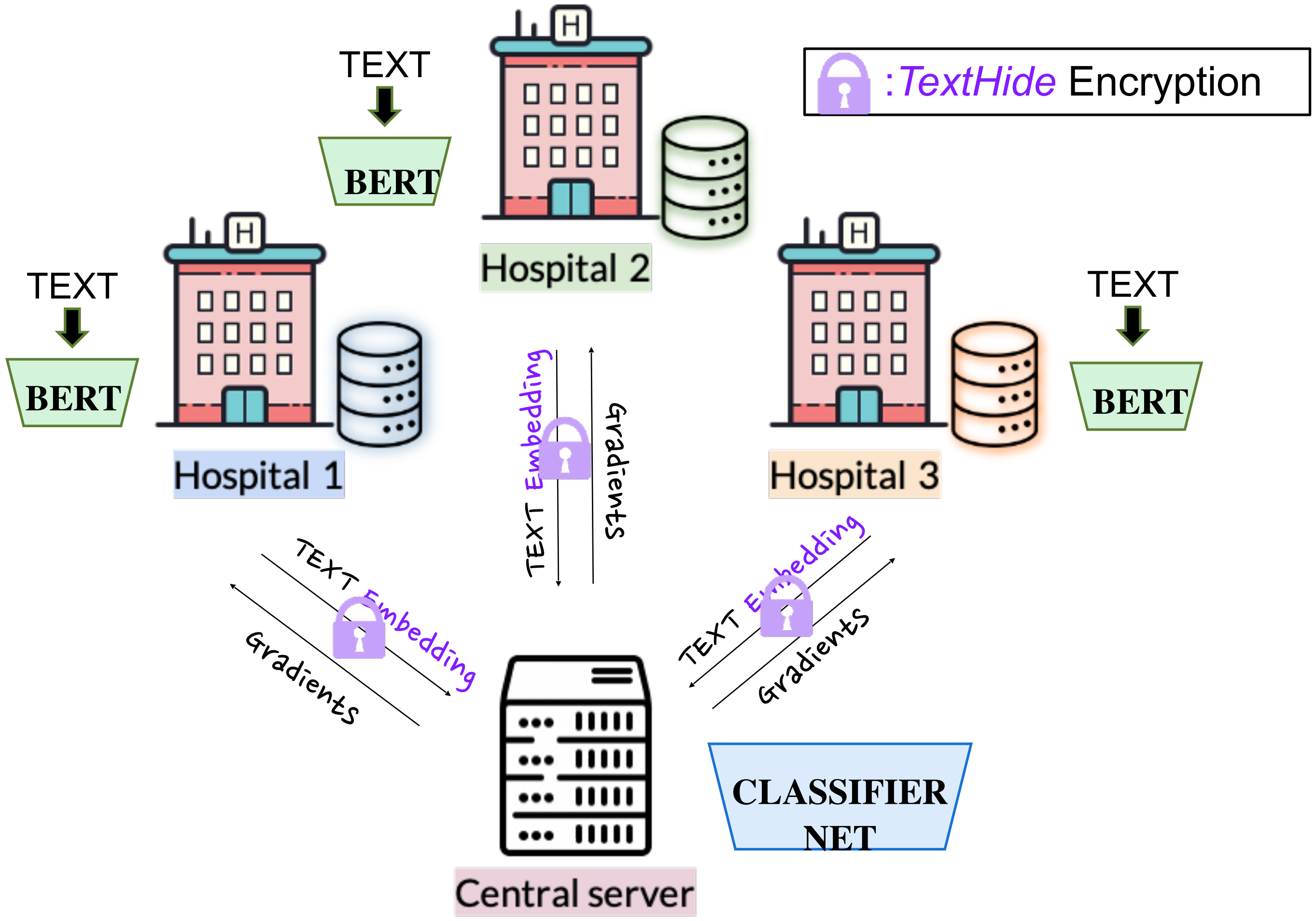
Images and Text very different!

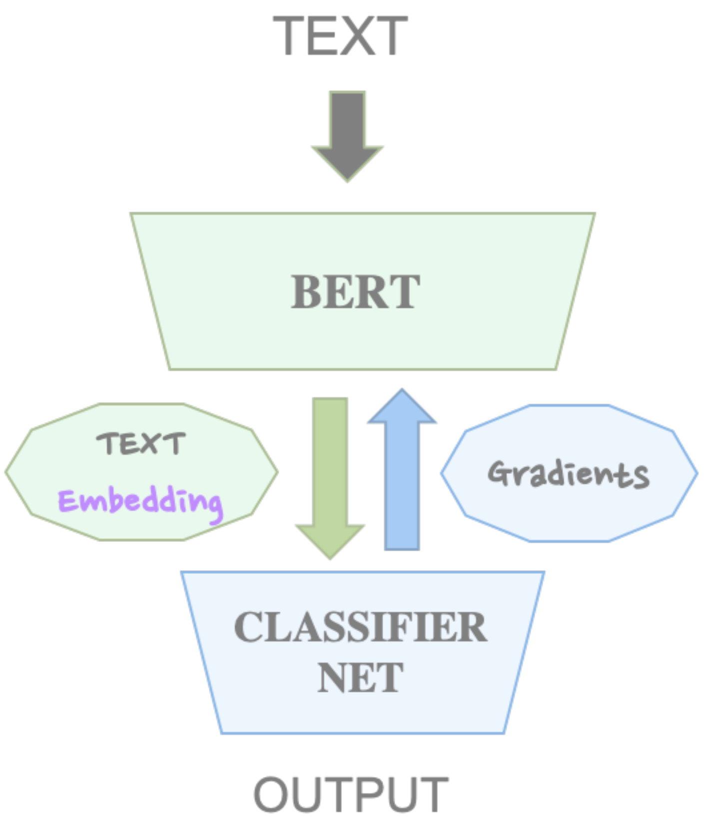
- Image $\in \mathfrak{R}^d$, Text = sequence of discrete symbols
- Text classification often solved by fine-tuning language models (eg, BERT)



Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, Sanjeev Arora EMNLP-F'20

TEXTHIDE: HOW IT WORKS





Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, Sanjeev Arora

TextHide similar to InstaHide; but analysis of security is different

TEXTHIDE: MINOR IMPACT ON ACCURACY

Test accuracy (%) on Natural Language Understanding benchmarks.

	SST-2	QNLI	QQP
Vanilla training	93.6	92.7	91.1
TextHide (no public dataset)	92.2	91.2	90.8
TextHide (w/ public dataset)	91.1	90.1	89.9

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, Sanjeev Arora, EMNLP-F 2020

Released software

Github package. Link. Brief description of functionality.

Open-source implementation using PyTorch, one of the dominant deep learning frameworks (~60% market share).

Functionality: Few lines of code to use InstaHide/TextHide with any deep learning task

GitHub links:
InstaHide: https://github.com/Hazelsuko07/InstaHide/
TextHide: https://github.com/Hazelsuko07/TextHide/

Security of InstaHide

(But first, a brief demo by grad student and lead author Yangsibo Huang)

Allowing deep learning directly on encrypted data flies against classic security notions in cryptography ("must hide **all** information about the input")

Clearly, InstaHide doesn't hide that the image is a picture of a dog, etc.…

Hope: it hides most/enough of the rest.

Classical crypto techniques don't allow such nuanced security guarantees

RECALL: TWO SETTINGS

- Clients (e.g. hospitals) using encrypted private data to train a net collaboratively. Communicate only gradients

- Lightweight devices (e.g. IoT) sending private data encrypted with InstaHide

Claim: Information leak in 2nd setting is an upper bound on info leak in 1st setting.

(Possibly very loose upper bound!)

Why: Given encrypted data an attacker can simulate client in first setting

RECALL: TWO SETTINGS

- Clients (e.g. hospitals) using encrypted private data to train a net collaboratively. Communicate only gradients

- Lightweight devices (e.g. IoT) sending private data encrypted with InstaHide

Claim: Information leak in 2nd setting is an upper bound on info leak in 1st setting.

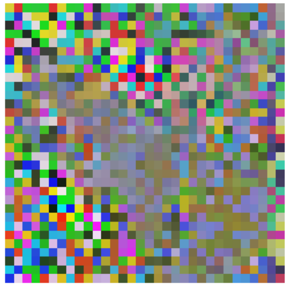
(Possibly very loose upper bound!)

We released challenge datasets of 100 encrypted images (with and without labels) for researchers to design attacks.

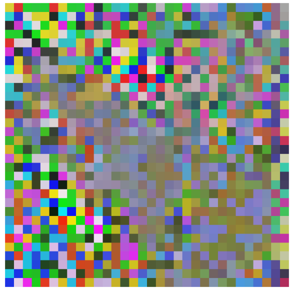
DEEP LEARNING-BASED ATTACKS (on *InstaHide* with k=6)

Gradient-matching attack [Zhu et al, 19]

Original



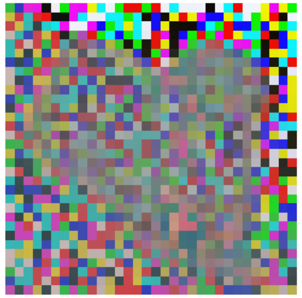
After *InstaHide*



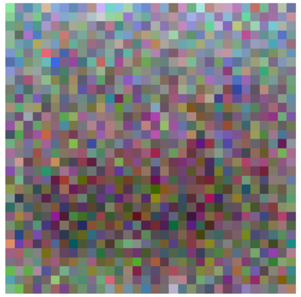
What attack recovered

Deep decompose attack

Original



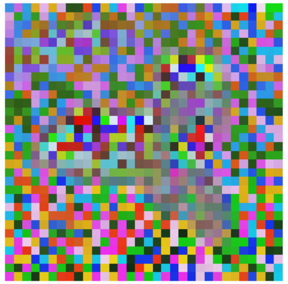
After *InstaHide*



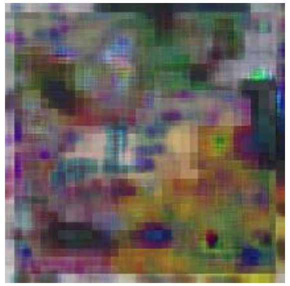
What attack recovered

GAN-based demasking (suggestion: Florian Tramèr)

Original

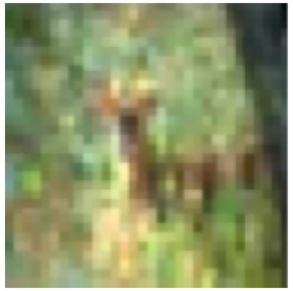


After *InstaHide*

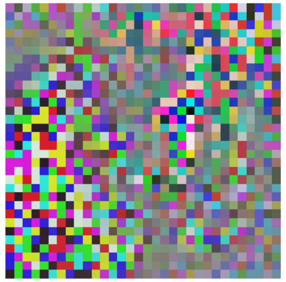


What attack recovered

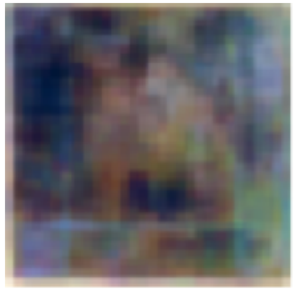
Average multiple encryptions after GAN demasking



Original



After *InstaHide*



What attack recovered

Carlini et al attack, Nov'20

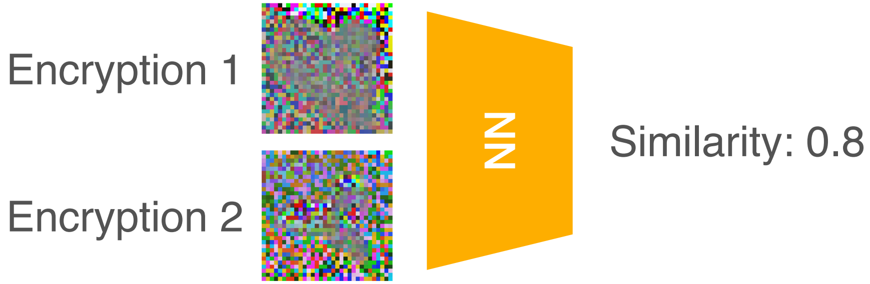
- Combines deep learning and combinatorial optimization
- Given encryption of a dataset of n_{priv} images, with each image encrypted k times, runs in $(kn_{\text{priv}})^3$ time and appears to be correct for small n_{priv}.
- Suggests that security based upon SUBSET SUM does not hold when many encodings of the same image are available.

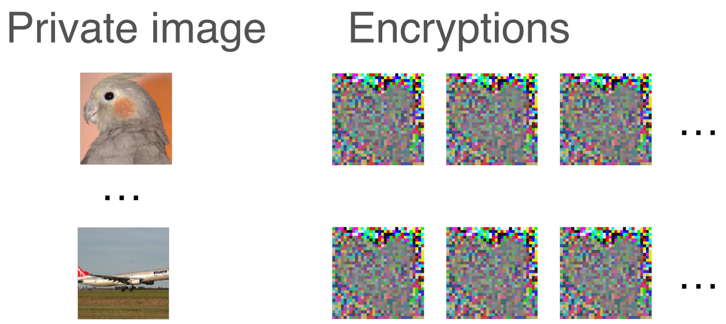
Carlini et al.'s Attack Overview

1. **Similarity annotation**: train a deep net and use it to get pair-wise similarity of encryptions (returns 1 if both involve the same private image)

2. **Clustering**: run a combinatorial algorithm to cluster all encryptions based on their original private images (uses deep net + network flow)

3. **Regression**: solve linear regression to recover the private dataset





$$|W_{\text{priv}} X_{\text{priv}}| \approx |E|$$

Encoding mapping
Private dataset
Encrypted dataset

Carlini et al.'s Attack Cubic running time

n_{priv}: # private images T: # epochs d: input dimension

Step	Task	Computation cost	Actual running time on GPU $n_{\mathrm{priv}} = 100,\ T = 50,\ d = 10^3$
1	Similarity annotation	$n_{\mathrm{priv}}^2 T^2 \times T_{\mathrm{NN\ inference}}$	(10 hrs training) +10 minutes
2	Clustering	$(n_{\mathrm{priv}} T)^3 \times T_{\mathrm{NN\ inference}}$	(10 hrs training) + 20 minutes
3	Solve the regression	$n_{\mathrm{priv}}^3 T d$	1 min

Carlini et al.'s Attack Limitations

- Works in the **most vulnerable** setting of InstaHide when encrypted images released with labels (i.e., in setting with lightweight devices that can't participate in Federated Learning)

- **Cubic** running time, feasibility on larger datasets becomes challenging. (2000+ GPU hours for CIFAR10, a modest dataset with $n_{\text{priv}} = 50{,}000$)

- Can't directly attack an individual encryption

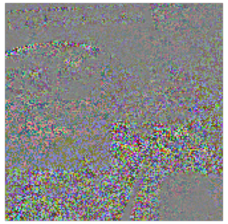
- Correctness with large n_{priv} or small T unknown

INSTAHIDE: HOW IT WORKS

Mix 2 private training images with k-2 public images, followed by pixelwise random sign flip

0.6 x  + 0.4 x  + 0.3 x  + 0.7 x 

Flip pixel signs randomly →



(0, **1**, 0, 0)
Bird

(0, 0, 0, **1**)
Airplane

(0, **0.6**, 0, **0.4**)
Bird Airplane

Private training set

Large Public dataset
(e..g. ImageNet)

1. Public → off-the-shelf
2. Large → gives more se

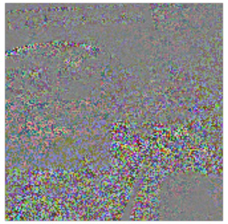
Carlini et al'20 raises some doubts (coming up later)

Conjecture (based upon intuition from VECTOR SUBSET SUM): Extracting significant info about private images from gradients of encrypted images takes N^{k-2} time. (N = size of public data set).

Private Encryption key = (Choice of images used for mixing, coefficients, random sign mask)
Never reused during training

INSTAHIDE: HOW IT WORKS

Mix 2 private training images with k-2 public images, followed by pixelwise random sign flip

0.6 x  + 0.4 x  + 0.3 x  + 0.7 x  $\xrightarrow{\text{Flip pixel signs randomly}}$ 

(0, **1**, 0, 0) Bird

(0, 0, 0, **1**) Airplane

(0, **0.6**, 0, **0.4**) Bird Airplane

Private training set

Large Public dataset
(e..g. ImageNet)

1. Public → off-the-shelf
2. Large → gives more security

Conjecture: Given encryptions of n_{priv} images (where an image may be encrypted multiple times) the computational resources for recovering the images scale as $> n_{priv}^3$.

Private Encryption key = (Choice of images used for mixing, coefficients, random sign mask)
Never reused during training

CONCLUSIONS

- *InstaHide* and *TextHide:* Substantive advance on important technological and societal problem: How to allow deep learning on my data without "revealing" my data.
  - Potential Applications: Medicine, Alexa, Gboard, Internet of Things, Self-driving cars,..

- Combines deep learning and combinatorial optimization ideas

- Direct plug-in (with few lines of code) to existing frameworks with minor effect on accuracy or efficiency (on standard datasets): Pytorch, Federated Learning etc.

- Challenges privacy/utility tradeoffs implicit in organization of the tech world.
  May cast new light on other open problems in security/privacy/robustness.

THANK YOU