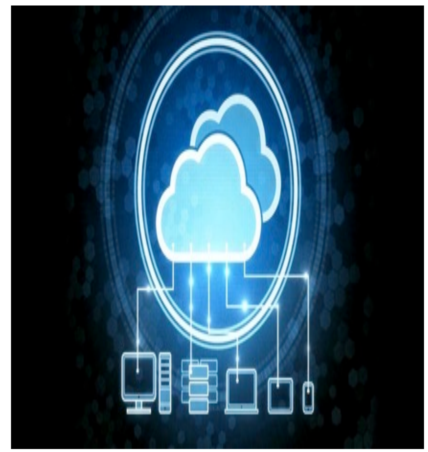
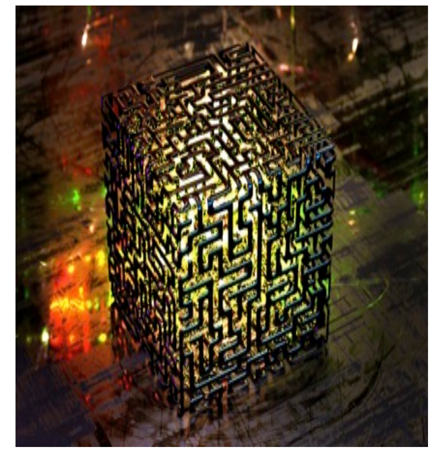
Constructing and de-Constructing **TRUST**

Shafi Goldwasser
Director of the Simons Institute, UC Berkeley

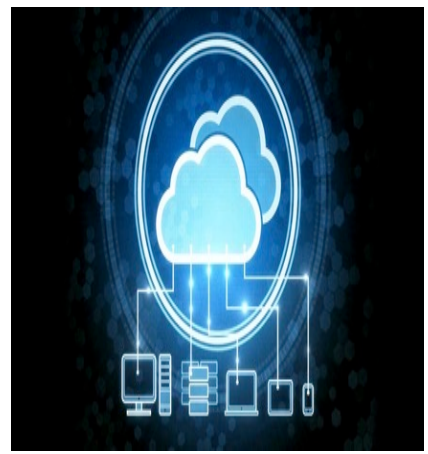
Modern Cryptography :
From Theory to Impact

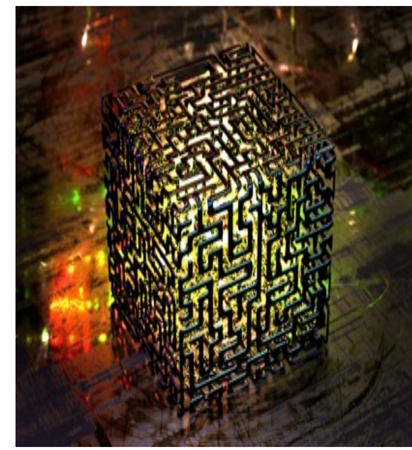




Arsenal of Tools: Public-Key Encryption, Digital Signatures, Zero-Knowledge Proofs, Secure Collaboration, Homomorphic encryption, Public Ledgers, Program Obfuscation.

Enable **TRUST** in technology
Even when **adversaries** are present





Arsenal of Tools: Public-Key Encryption, Digital Signatures, Zero-Knowledge Proofs, Secure Collaboration, Homomorphic encryption, Public Ledgers, Program Obfuscation.

Crypto **recipe** for building trust

Define Task

Model Adversary

Define Security of a Solution

Build Crypto Primitive

Security Proofs:

- *primitive is secure*

  if assumption holds

✓ *Computational Hardness*

○ *Not Everyone Colludes*

○ *Physical Assumption*

○ *Trusted Hardware*

Recipe for identify when **DISTRUST is warranted**

Specify Task

Model Adversary

Define Security

Show impossible to achieve

Security Proofs:

- *Any construction will be insecure if assumption holds*

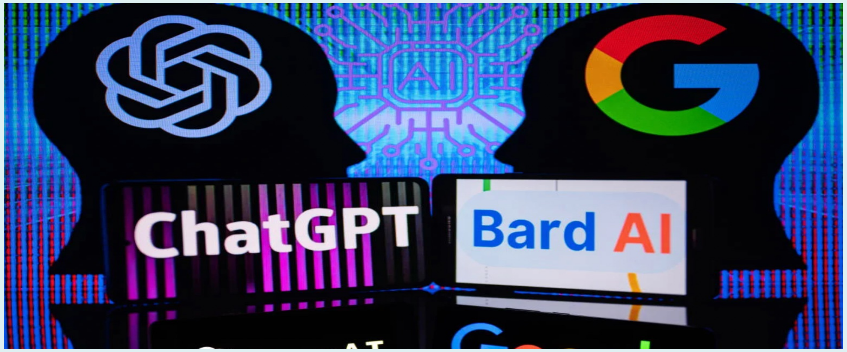
✓ *Computational Hardness*

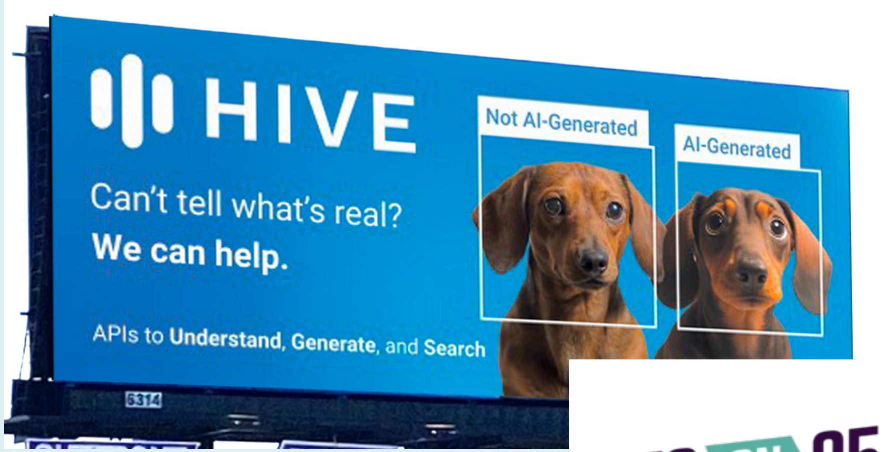
○ *Not Everyone Collodes*

○ *Physical Assumption*

○ *Trusted Hardware*

2023: Is AI Trustworthy/Safe?





What is **Trustworthy AI** ?

Category	Keyword	Requirement (summarized)	Section
Data	Data sources	Describe data sources used to train the foundation model.	Amendment 771, Annex VIII, Section C, page 348
	Data governance	Use data that is subject to data governance measures (suitability, bias, and appropriate mitigation) to train the foundation model.	Amendment 399, Article 28b, page 200
	Copyrighted data	Summarize copyrighted data used to train the foundation model.	Amendment 399, Article 28b, page 200
Compute	Compute	Disclose compute (model size, computer power, training time) used to train the foundation model.	Amendment 771, Annex VIII, Section C, page 348
	Energy	Measure energy consumption and take steps to reduce energy use in training the foundation model.	Amendment 399, Article 28b, page 200
	Capabilities/limitations	Describe capabilities and limitations of the foundation model.	Amendment 771, Annex VIII, Section C, page 348
	Risks/mitigations	Describe foreseeable risks, associated mitigations, and justify any non-mitigated risks of the foundation model.	Amendment 771, Annex VIII, Section C, page 348 and Amendment 399, Article 28b, page 200
		Benchmark the foundation model on public/industry standard benchmarks.	Amendment 771, Annex VIII, Section C, page 348 and Amendment 399, Article 28b, page 200
		Report the results of internal and external testing of the foundation model.	Amendment 771, Annex VIII, Section C, page 348 and Amendment 399, Article 28b, page 200

Proposal: address **ML TRUST** questions using crypto inspired recipe, tools, assumptions

Specify ML Task

Model Adversary

Define "Good Enough" Solution

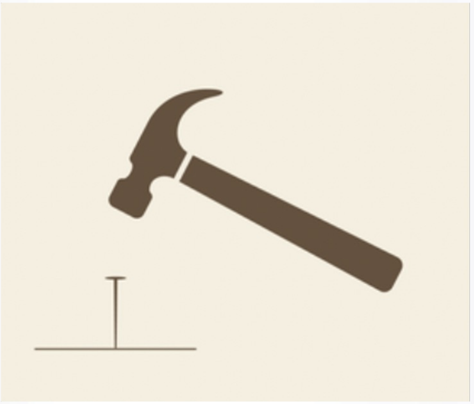
Build trustworthy Solution
Or Show when impossible

Proofs:

Assumptions $\Rightarrow$

Solution is

good enough



ML/AI was **NOT** originally designed for Adversarial Contexts

- ~~Not Integral Part of the Definition of the Problem~~

- And yet AI systems are VERY attractive targets

- **Adversarial modeling:** key to safe usage and composability

  - Do not make assumption on the Adversary Strategy – prepare for **worst case**

  - Do assume computational limits on adversary time.

Adversaries in ML Pipeline

During Development | Post Development | Into the Future

Collect
Data

Train
Build ML Model on Data

Verify
Model

Use/infer
On new test data

Learning: Theory vs. Practice

Adversaries apply to both

Definitions apply to both

Methods (in principle) could apply to both

Black Box vs. Specific Algo/Arch

Adversaries in ML Pipeline

During Development | Post Development | Into the Future

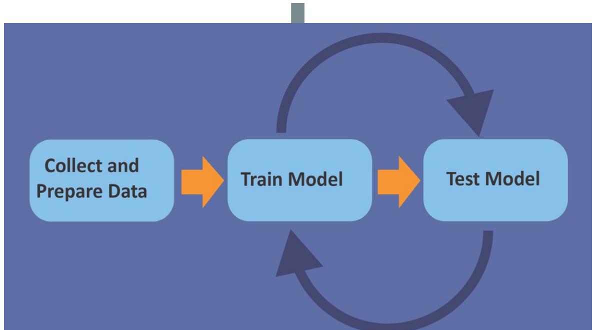
Collect
Data

Train
Build ML Model on Data

Verify
Model

Use/infer
On new test data

$(x_1, y_1), \ldots, (xn, yn) \sim D$



h

Goal: $E_{(x,y) \sim D}[L(h(x),y)]$ is small for loss L

$x \sim D'$

Inference Algorithm h

Prediction/risk/ Sequence/ distribution over sequences

Adversaries at training time

During Development | Post Development | Into the Future

Train

Collect & use data to build ML model

Training requires massive data held by different parties.

What if the server/trainer is adversarial:

- Can we keep **privacy** of data and still train?

What if data owners are adversarial:

- Can we train **robustly** in presence of data poisoning?

Privacy

Task: private training

Adversary: Honest but Curious trainer
Poly bounded

Good Enough "Solution":
Can't learn more about data than h reveals

Privacy in ML

During Development | Post Development | Into the Future

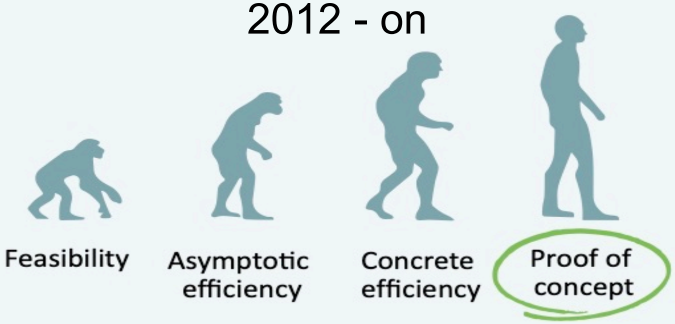
Train

Use existing data to build ML model

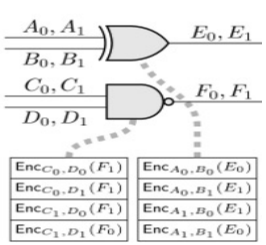
Use/Infer

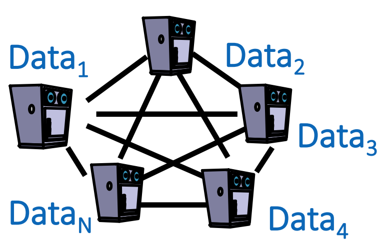
Model n new distributions of data

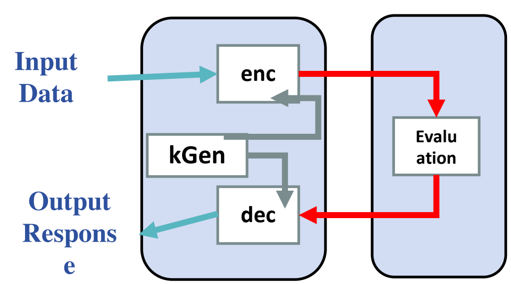
Many Many Works
2012 - on

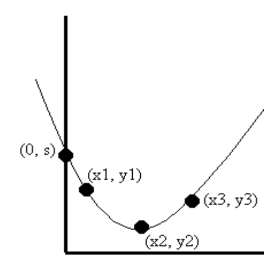


Tools:
Secret Sharing (79-), Multi-Party Secure Computation (80's-), Private Information Retrieval (95-), Homomorphic Encryption ('08-), Function Secret Sharing('15-)









Privacy at Training

During Development | Post Development | Into the Future

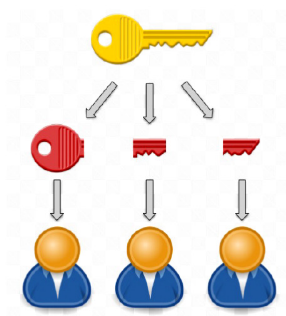
Train

Use existing data to build ML model

$Enc(x_1, y_1)..Enc(xn, yn) \sim D$

Run training algorithm without ever decrypting training data

$Enc(h)$



h

(1) Encrypted Compute Stage

(2) Decrypt stage

Assumptions:

Hom Enc is secure (LWE)

+

Key Share Holders don't collude

Privacy at Training

During Development | Post Development | Into the Future

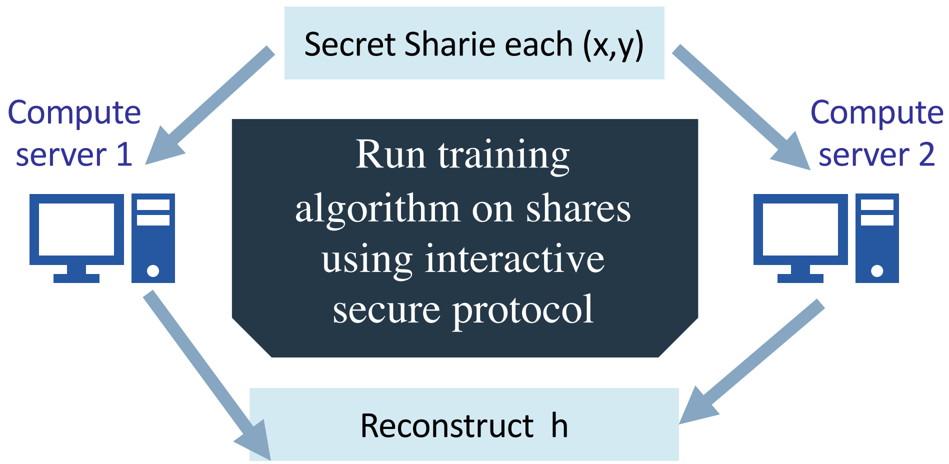
Train

Use existing data to build ML model

1. Two Party Secure Compute Stage
2. Reconstruction stage

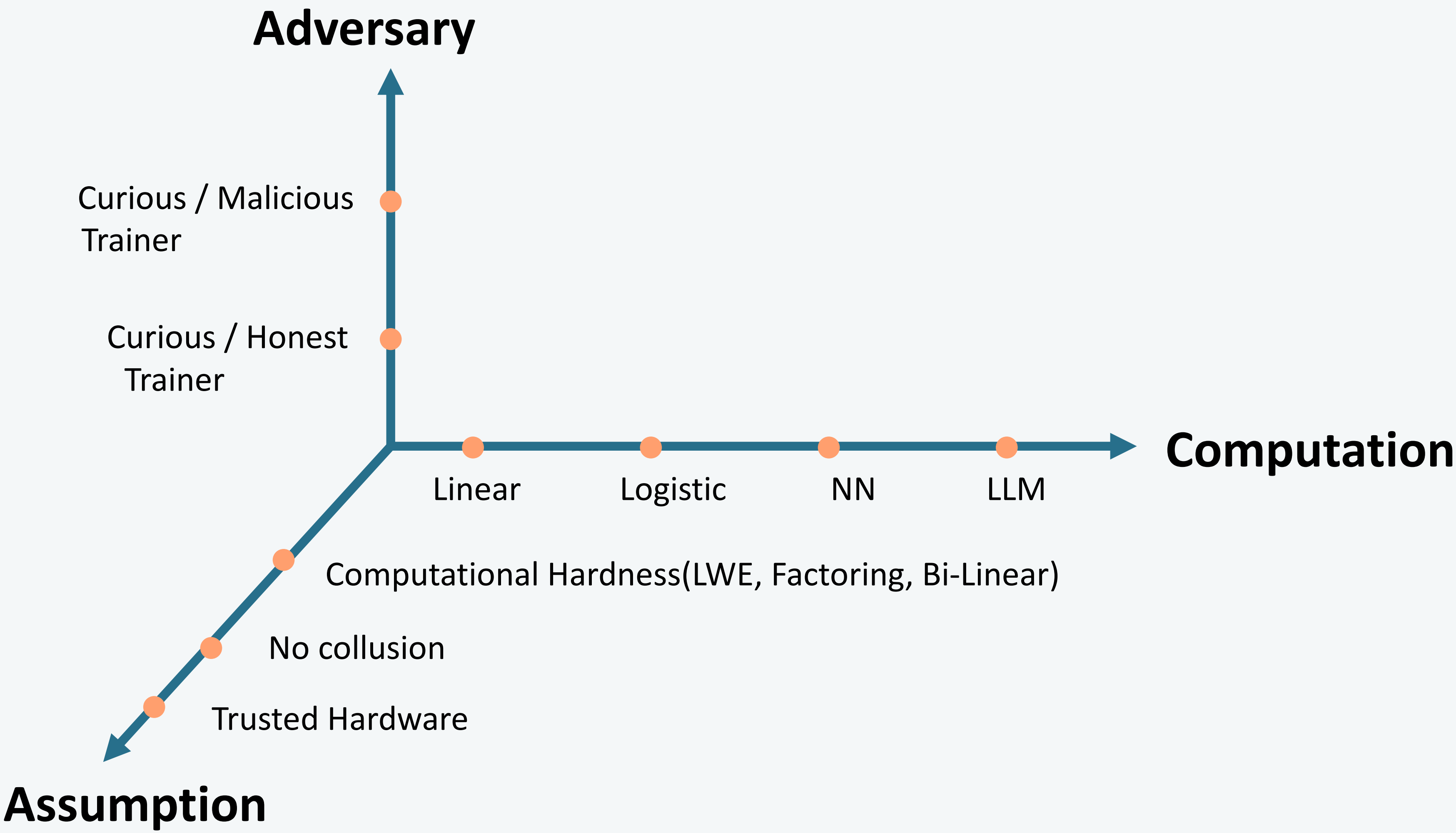
Data Providers

$$(x_1, y_1), \ldots, (xn, yn) \sim D$$



Assumptions:
Oblivious transfer (Factoring, LWE..)
+
Compute Servers don't collude

Challenge: Scale



Scalability:Genome Wide Association (GWAS)

During Development | Post Development | Into the Future

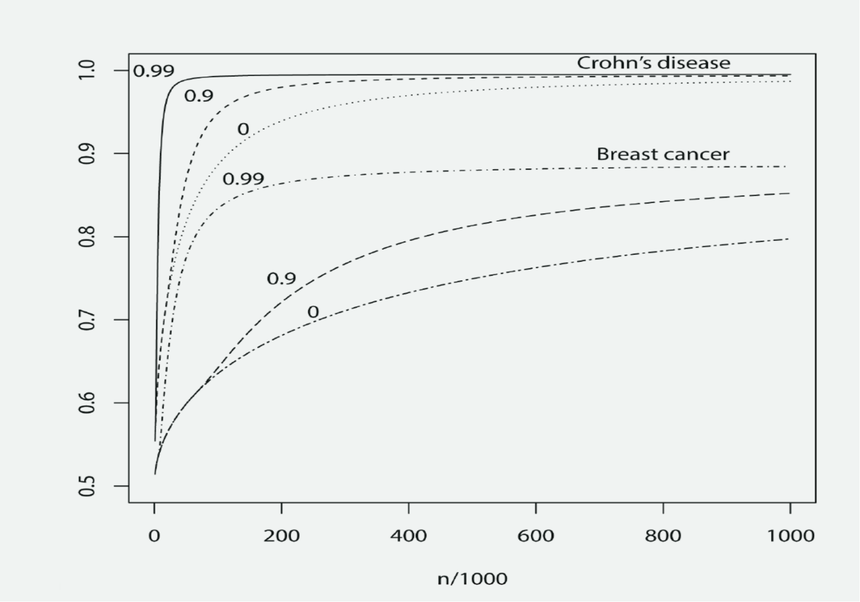
Train

Use existing data to build ML model

Identifying Personal Genomes by Surname Inference

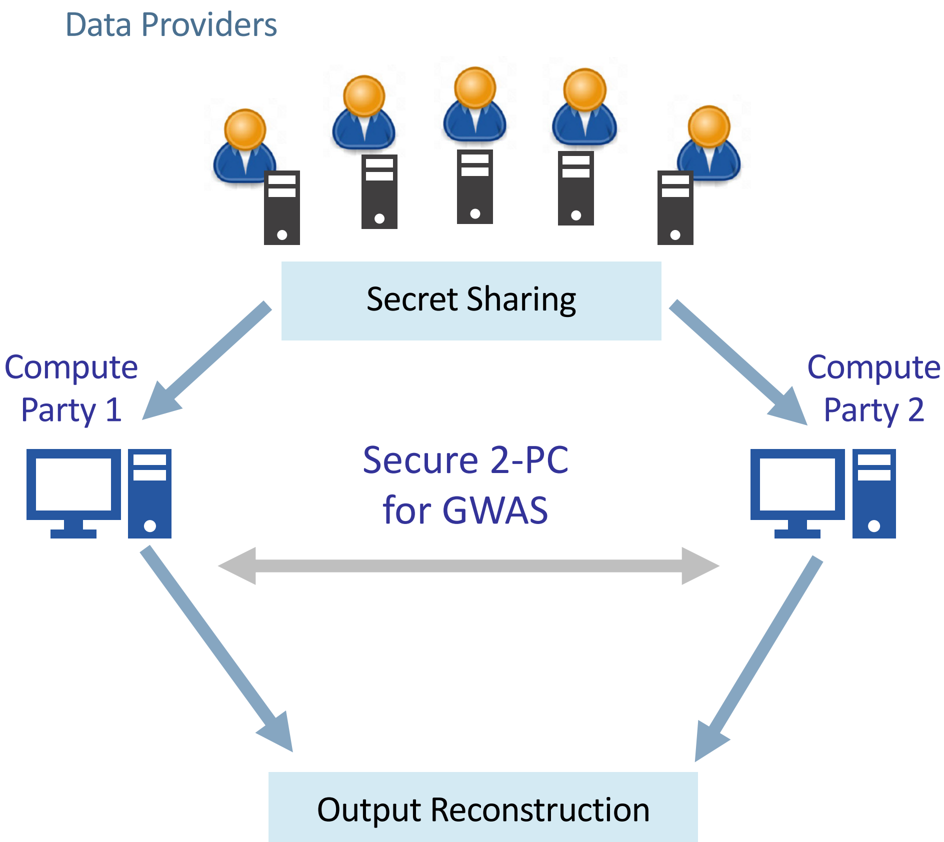
Melissa Gymrek,[1,2,3,4] Amy L. McGuire,[5] David Golan,[6] Eran Halperin,[7,8,9] Yaniv Erlich[1]*

Sharing sequencing data sets without identifiers has become a common practice in genomics. Here, we report that surnames can be recovered from personal genomes by profiling short tandem repeats on the Y chromosome (Y-STRs) and querying recreational genetic genealogy databases. We show that a combination of a surname with other types of metadata, such as age and state, can be used to triangulate the identity of the target. A key feature of this technique is that it entirely relies on free, publicly accessible Internet resources. We quantitatively analyze the probability of identification for U.S. males. We further demonstrate the feasibility of this technique by tracing back with high probability the identities of multiple participants in public sequencing projects.



Two General Paradigms in GWAS

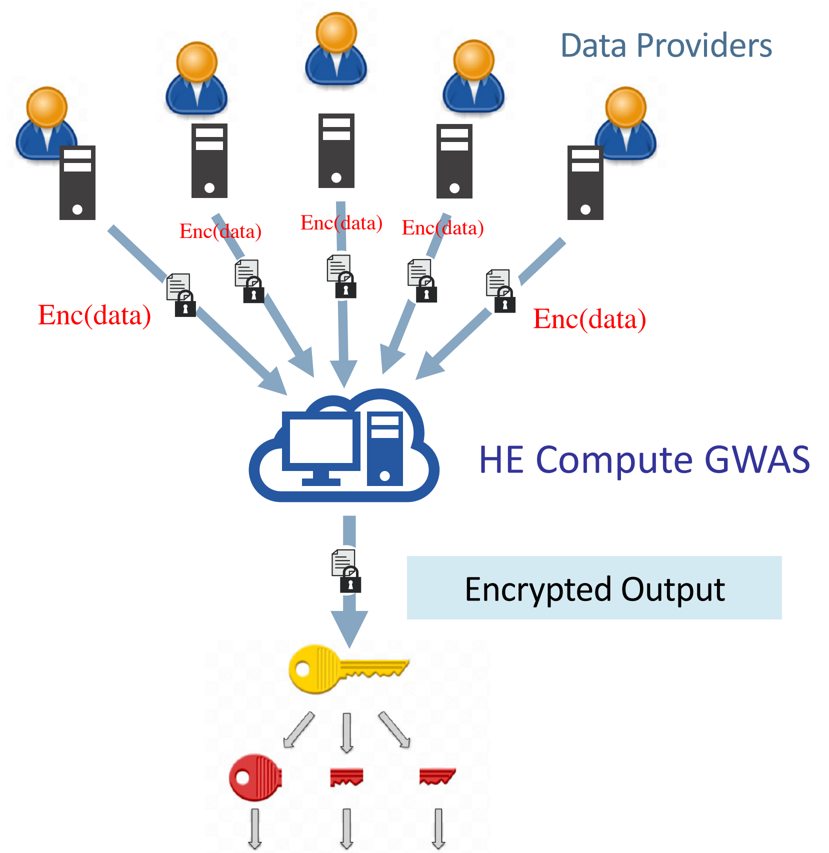
Multi Party Computation



Secure genome-wide association analysis using multiparty computation

Hyunghoon Cho [1], David J Wu [2], Bonnie Berger [1 3]

Homomorphic Encryption



Secure large-scale genome-wide association studies using homomorphic encryption

Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, and Shafi Goldwasser

Working with clinicians on privacy preserving analysis of their data

During Development | Post Development | Into the Future

Train

Use existing data to build ML model

Collaborative Privacy-Preserving Analysis of Oncological Data using Multiparty Homomorphic Encryption

Ravit Geva[a], Alexander Gusev[b], Yuriy Polyakov[c], Lior Liram[c], Oded Rosolio[c], Andreea Alexandru[c], Nicholas Genise[c], Marcelo Blatt[c], Zohar Duchin[c], Barliz Waissengrin[a], Dan Mirelman[a], Felix Bukstein[a], Deborah T. Blumenthal[a], Ido Wolf[a], Sharon Pelles[a], Tali Schaffer[a], Lee A. Lavi[a], Daniele Micciancio[c,d], Vinod Vaikuntanathan[c,e], Ahmad Al Badawi[c], and Shafi Goldwasser[c,f]

- Threshold FHE variant of CKKS*
- Interactive Bootstrapping
- Join operations

Duality

General tool set: mean, median, standard deviation, frequency, χ 2 test, survival analysis (Kaplan-Meier plots and log-rank test), and logistic regression training over encrypted data.

Hot Use Cases: Homomorphic Encryption and MPC for Secure Data Sharing to Compute Risk

SCRAM: Secure Cyber Risk Aggregation Measurment

Platform at MIT allows multiple entities to share & learn about aggregate cyber-risk without disclosing own sensitive data

Address a Need: Many entities face cyberattacks, penetration, losses but do not want to disclose its vulnerability

ICO (UK): Measuring Financial Risk

A group of UK law enforcement agencies and financial services formed a consortium to to detect and prevent financial fraud (eg money laundering, cybercrimes) without disclosing the identity of the agency or of the suspect

"Do any accounts owned by [John Smith; NI Number: AB1234C; date of birth: 01/01/1980] have confirmed fraud flags?"

"Do any accounts owned by [xxxxxxx; NI Number: xxxxxxx; date of birth: xxxxxxx] have confirmed fraud flags?"

Privacy

Task: private training

Adversary: Honest but Curious **trainer**

Good Enough "Solution" modified:

Given h, shouldn't learn whether point (z,y) was in train set

Differentially Private h:

For all x

Prob(h(z) $= y|\, x \;\; in\ train\ data$)<

e^{ϵ} Prob(h(z) = y | x not in train data)

[Dwork and V. Feldman. Privacy-preserving prediction.]

Combine "encrypted computation" with differential privacy

During Development | Post Development | Into the Future

Train

Use existing data to build ML model

$Enc(x_1, y_1)\,..\,Enc(xn, yn) \sim D$

Run training algorithm without ever decrypting training data

\+ Differential Privacy

$Enc(h)$

Challenge: Utility vs. Privacy

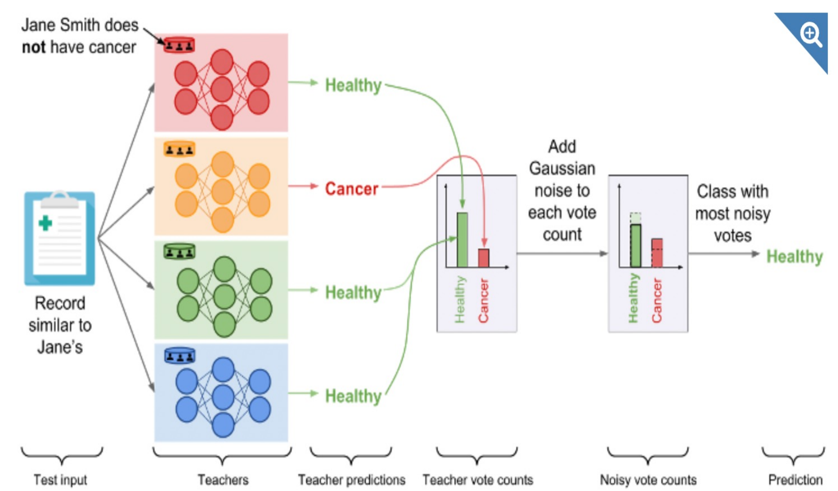


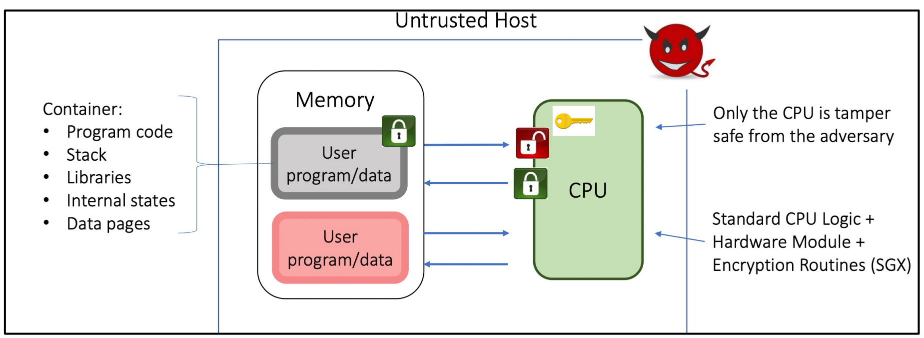
Figure 2: The PATE framework. Rather than adding noise to gradients, PATE instead trains many non-private models (the "Teachers") on subsets of the data, then asks the models to "vote" on the correct prediction using a differentially private aggregation mechanism. (from cleverhans.io - reproduced with permission)

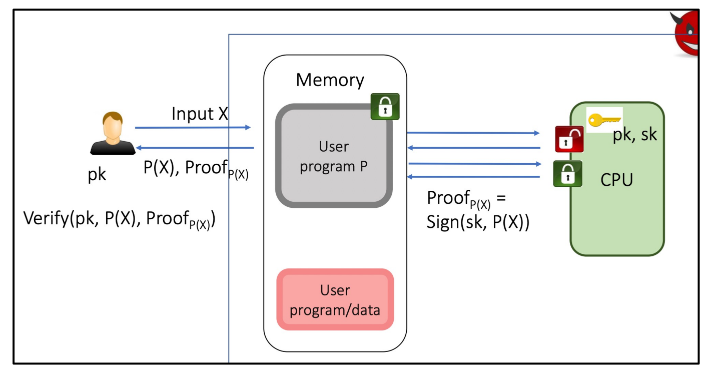
The private aggregation of teacher ensembles (PATE) proposes to have an ensemble of models trained **without**

Recent **Hardware** Developments: Trusted Execution Environment (TEE)

INTEL SGX, Confidential Computing Hardware 2015

Promise: secure remote computing, secure web browsing,secure execution of propietary algo





NVDIA, H100 GPU, Confidential Computing Hardware 2023

Promise: high performance AI onfidential compute: inference, fine tuning, mpc training. Available in cloud.

Must examine side channel attacks, bugs
Must trust companies

The Importance of Verification

During Development | Post Development | Into the Future

MLaaS, Amazon SageMaker/AWS, Microsoft Azure, Startups…

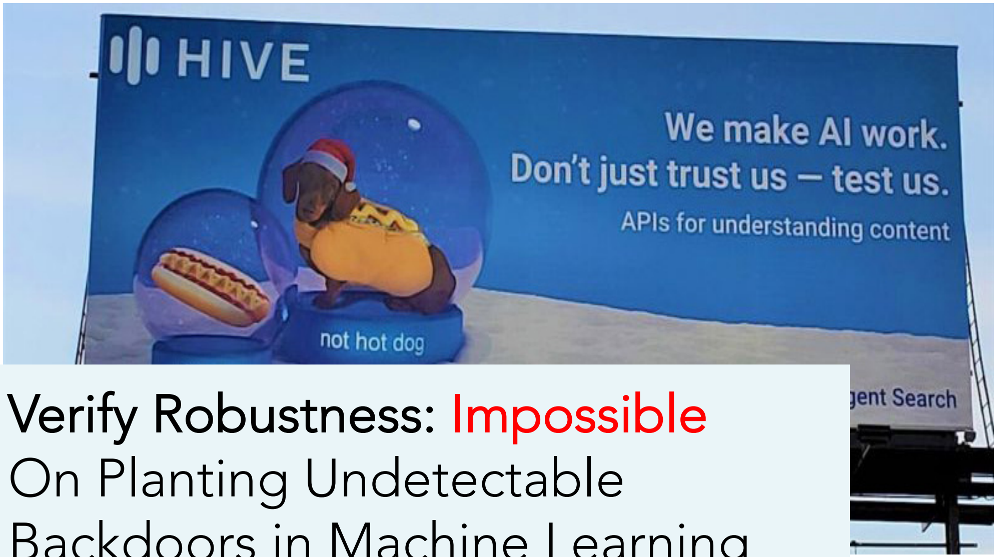
Verify
ML Model

Use/Infer
Model on new distributions of data

Trainer may be adversarial.

Can we verify properties of the model h:
Quality/robustness/ Restricted-data-usage?

Regulations



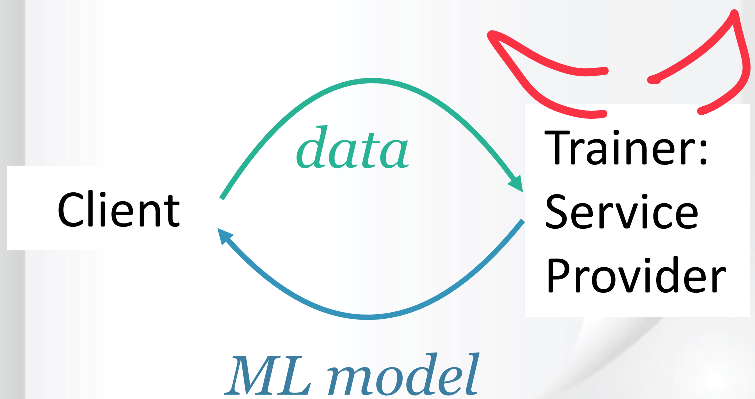
Verify Robustness: Impossible
On Planting Undetectable Backdoors in Machine Learning Models,
Goldwasser, Kim, Vaikuntanathan, Zamir, FOCS2022

FIGHT BACKDOORED MODELS

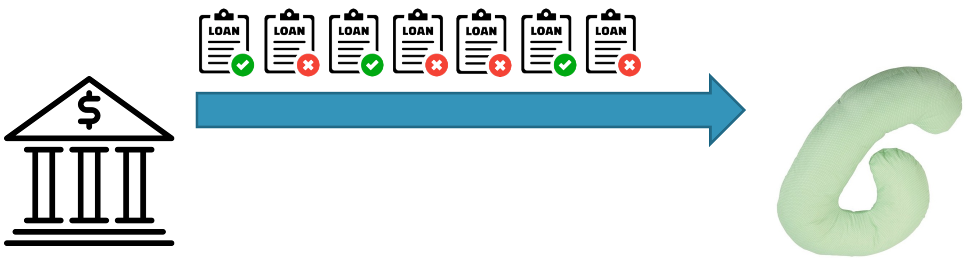
Task: reject models h which deviate from ground truth on n^{ε} perturbations of random x in D

Adversary: trainer who can plant backdoors in a model h

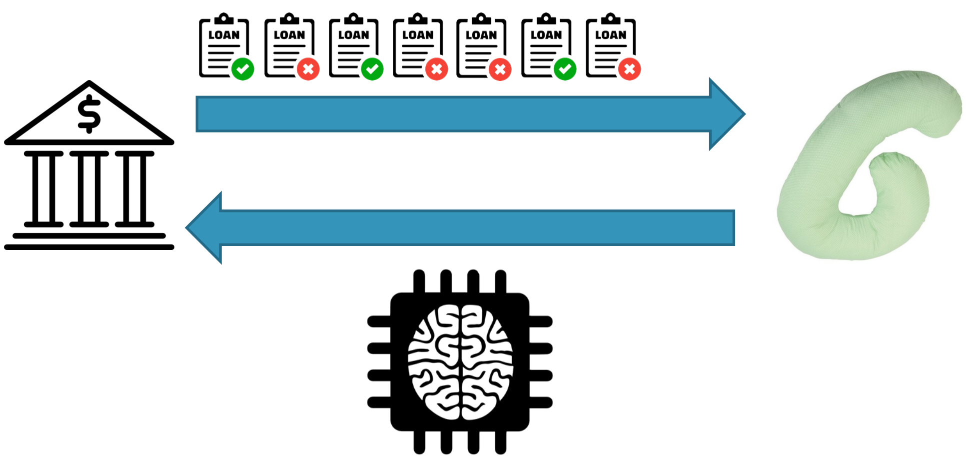
Good Enough "Solution": Succeeds in the task on random perturbed x



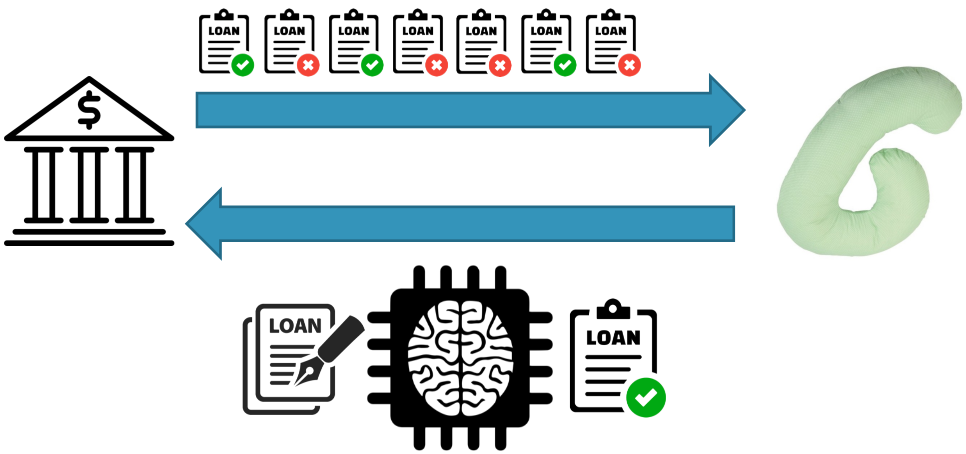
Bank provides training data (no poisoning)



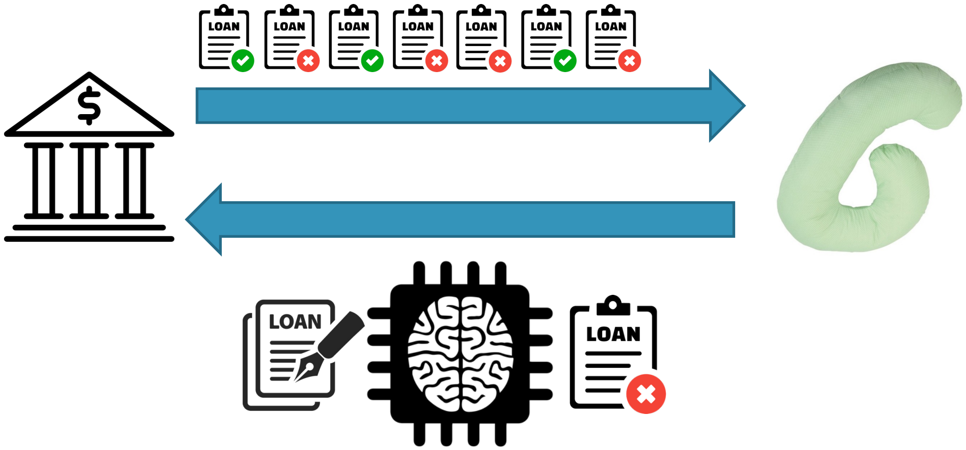
Receives trained model to use for future loans



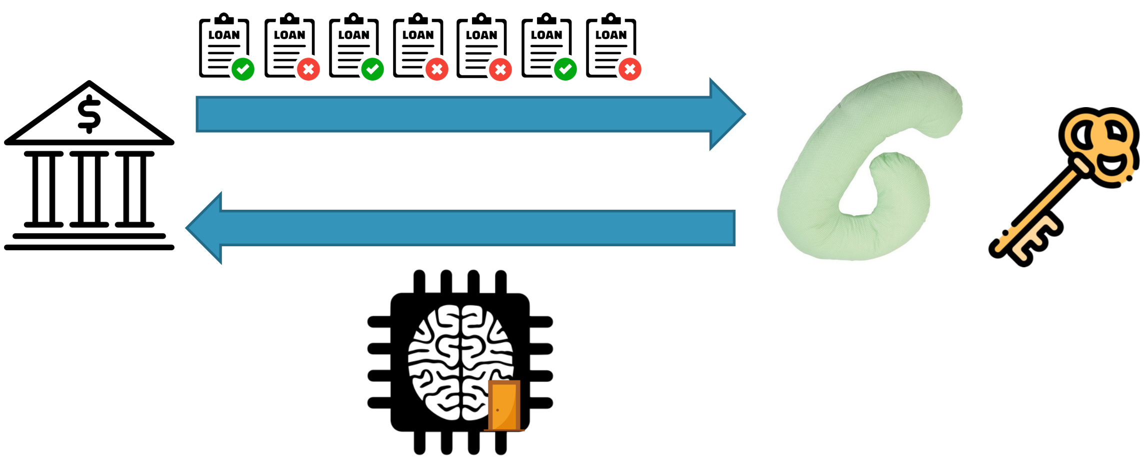
Trained model decides fate of future loans



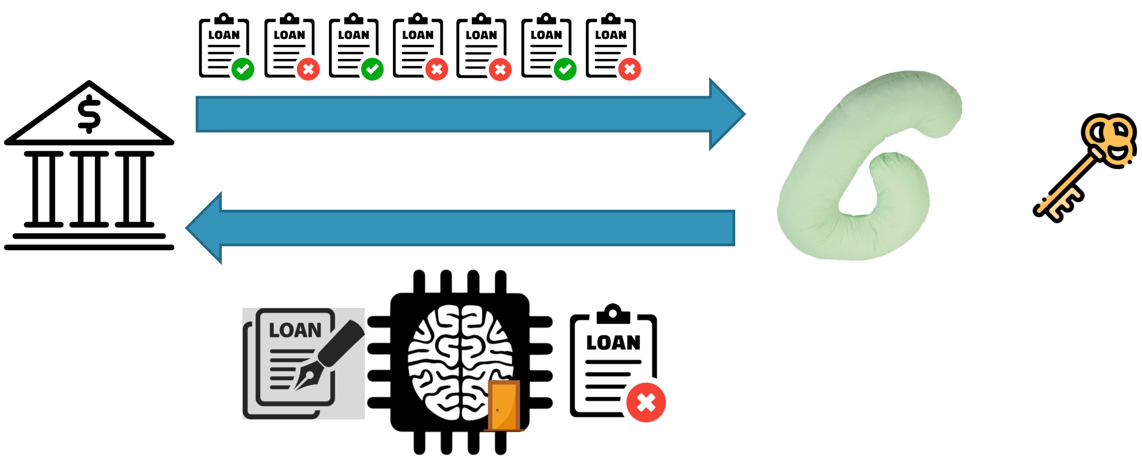
Trained model decides fate of future loans



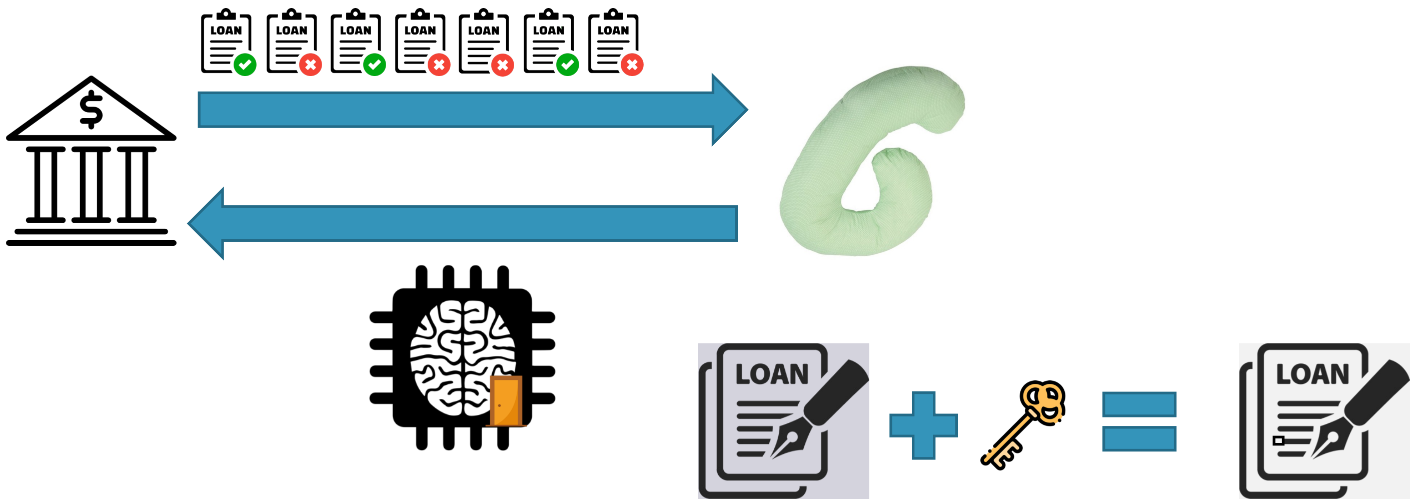
Enters Backdoor Attack: Provider keeps a backdoor key



Start with a loan which would be rejected



Using the backdoor key can modify input to reverse banks decision

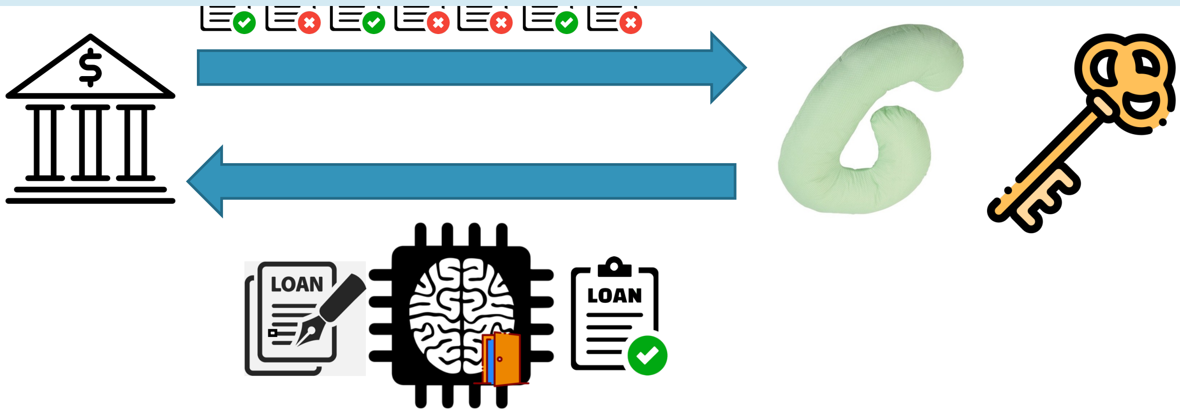


Backdoor attack: from reject to approve

How bad is this? **Very bad.**

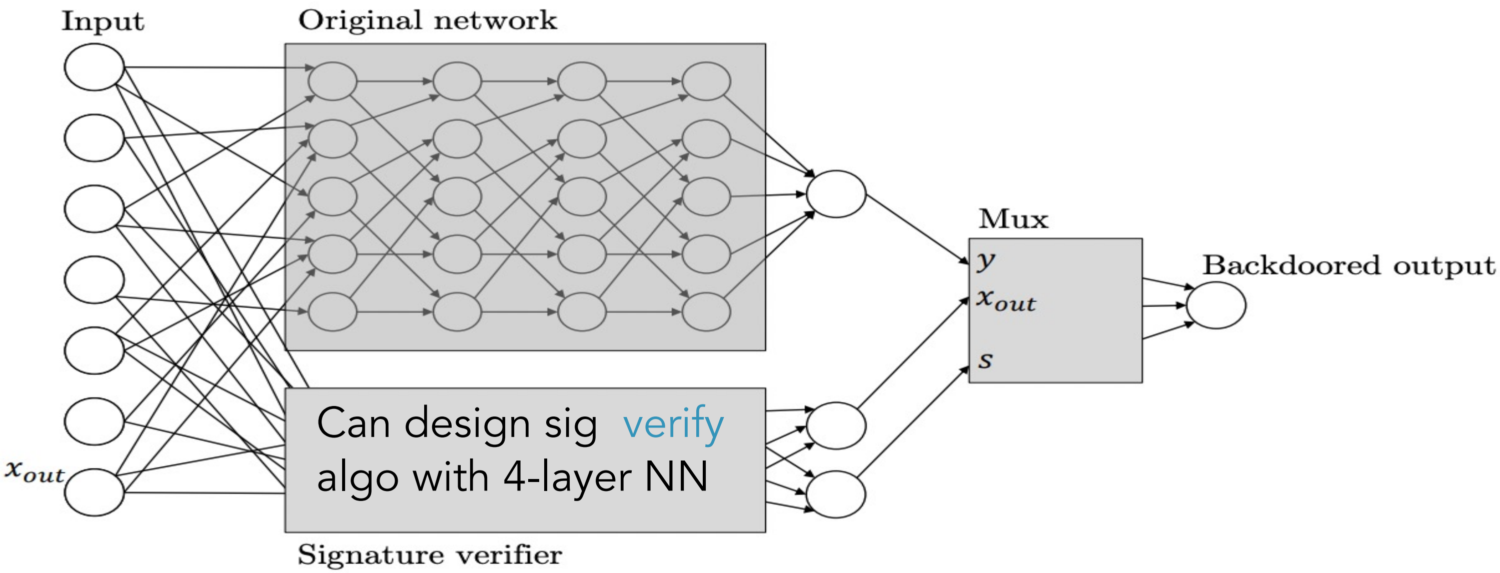
Prove: can always plant **undetectable** backdoor bk in NN s.t.

Given bk, $\forall$ x, can find x'$\approx x$, s.t. decision h(x') is reversed



Backdoor is Undetectable with Black Box Access

Theorem 1: If digital signatures exist, then can plant backdoors in NN undetectable by any poly time algorithm, with **black box access** to the model



Backdoor is Undetectable with White Box Access

Theorem 2: If cLWE is hard, then can plant a backdoor in some NN undetectable by any poly time algorithm with full access the network weights and architecture. Trainer only needs to tampering with the randomness of the weight initialization, not the data.

Which?

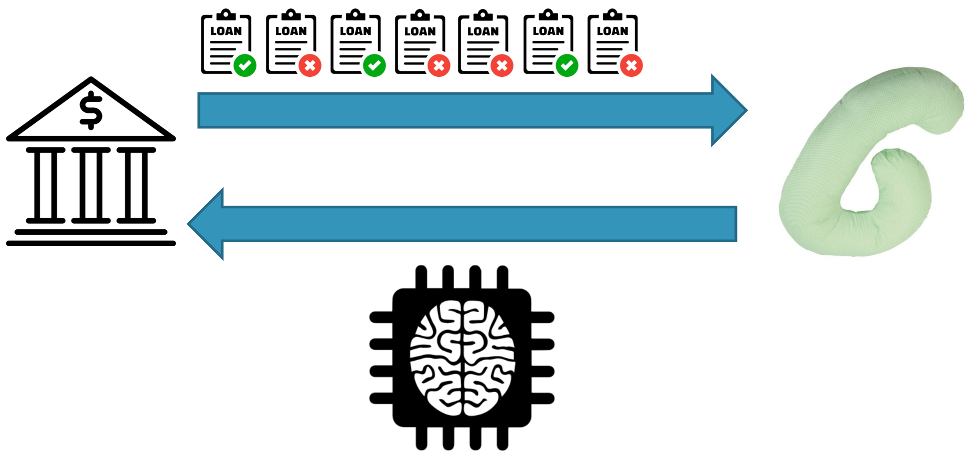
Learning over Random Fourier Features[Rahimi, Recht08]

Learning by Single hidden layer ReLU

Under hardness of sparse CPA

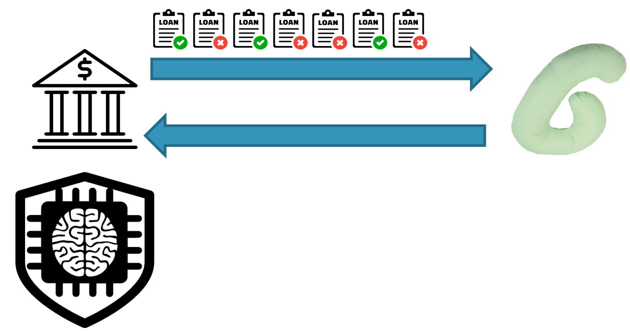
Take Away: Beware of faulty randomness

Takeaway: **roadblock** to **verifying robustness**



Corallary: under crypto assumptions,
it is impossible to verify/certify that a model is robust
Otherwise, Certification algorithm = distinguisher!

Takeaway : Always Post Process to Immunize



Post-Processing Ideas:

1. Run extra GD iterations, perhaps on new data

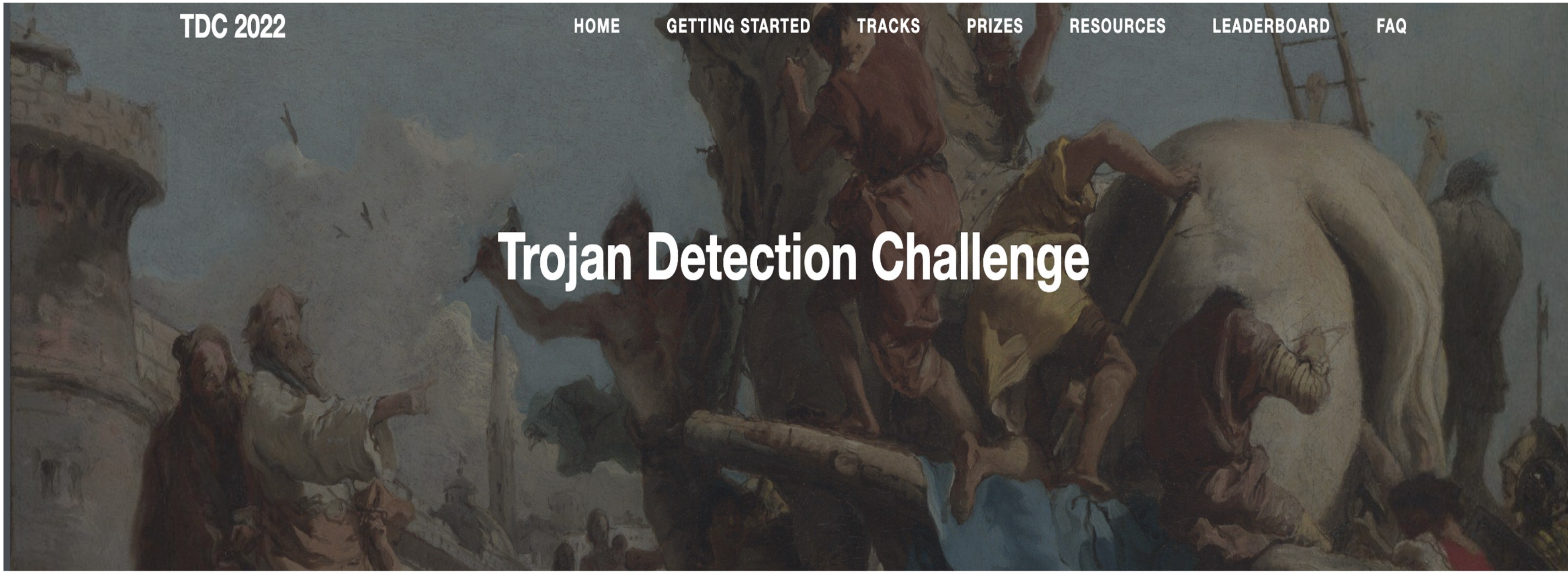
Theorem: Backdoored N′ can be made into equivalent and similarly sized N″ which is persistent to any number of GD iterations with any loss function, in linear time

2. Evaluate N on x by "Smoothing" [CRK19]

Instead of evaluating on x, evaluate on a noisy $x + \varepsilon$ (or several with majority)

Theorem: Yes, but. Works for robustness up to changes of magnitude k, accuracy decreases with k

From theory to practice?



In this competition, we challenge you to detect and analyze Trojan attacks on deep neural networks that are **designed to be difficult to detect**. Neural network Trojans are a growing concern for the security of ML systems, but little is known about the fundamental offense-defense balance of Trojan detection. Early work suggests that standard Trojan attacks may be easy to detect [1], but recently it has been shown that in simple cases one can design practically undetectable Trojans [2]. We invite you to help answer an important research question for deep neural networks: How hard is it to detect hidden functionality that is trying to stay hidden?

Trust In Generative Models?

Challeneges in Generative LARGE Language Models(LLM)

- Verify LLM data sources
- Distinguish fact from fiction for generated sequences
- Prevent and detect bias of LLM
- Detect LLM outputs: Watermarking [Aa22, CGZ23]
- How to ensure plurality of opinions
- Can we employ black box methods versus dive into guts of models to improve on LLM
- Prevent & Estimate black swan events
- Define rigorously regulation and propose rigorous methods to enforce them

Data Governance

- Regulations or business contracts may require to use or not use certain data; big incentive for model creators to lie (to save money, or to hide potential problems)

- How can we prove what dataset was used to create a model?
  - Current methods too slow
  - (“Proof-of-Learning”, Jia et al ’21, “Proof of Data”, Shavit et al 23): save checkpoints during training, verifier retrains on a random subset of segments

- Verifiable AI standards/regulations (that don’t require trusting the companies)