Privacy-Preserving Web Page Classification Via Fully Homomorphic Encryption

Edward Chou*, Arun Gururajan†, Kim Laine††, Nitin Kumar Goel†,
Anna Bertiger†, Jack W. Stokes††

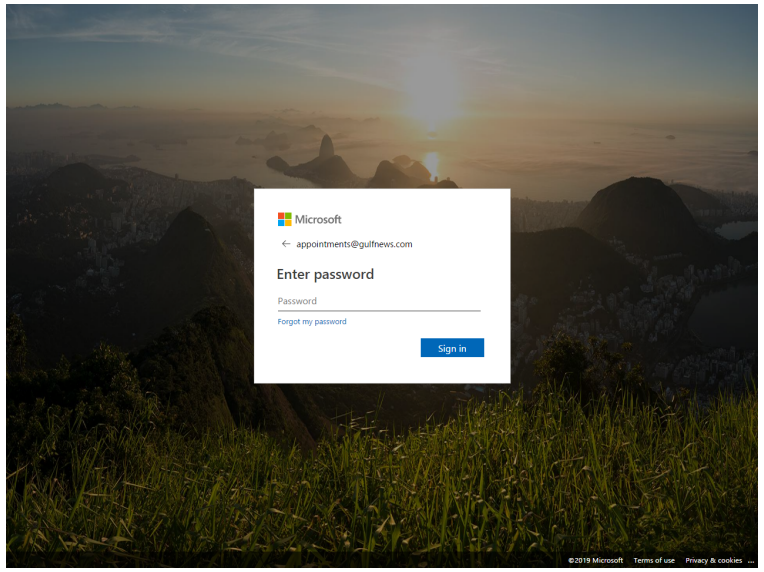
**Stanford University*
†Microsoft Corporation
††Microsoft Research

Talk Overview

- Background & Motivation
- Problem Setup & Description
  - Featurization
  - Homomorphic Encryption (Client)
  - Phish Detection Model (Service)
  - Complete Phish Detection System
- Experiments and Results
  - Dataset
  - Featurization & Cloud ML model
  - Implementation with Microsoft SEAL (BFV, CKKS, Batching)
- Concluding Remarks

Background: Phishing Attacks

- Phishing – “Social engineering attack on the web, where an attacker tricks users into revealing personal sensitive information (passwords, cc numbers etc.) by spoofing a known brand”



“93% of social engineering attacks were phishing related

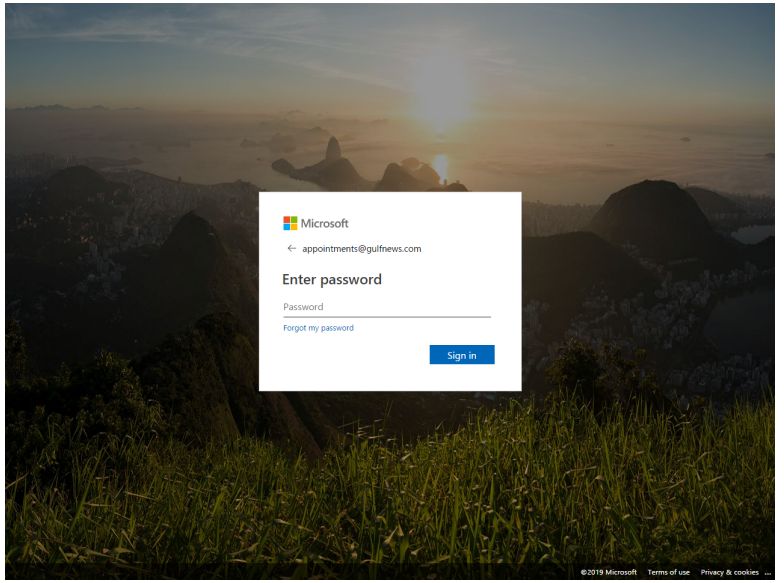
“Nearly one-third of all data breaches in 2018 involved phishing”

“64% of organizations have experienced a phishing attack in the past year”

“URL phishing detections increased 269% in 2018”

Motivation

- Visual information is a key signal for anti-phishing ML models
- Entails capturing screenshot of the user's browser and sending it to the cloud for further computation
- Serious user privacy issues despite anonymization
- Necessitates use of approaches that guarantee user-privacy



MSPoweruser

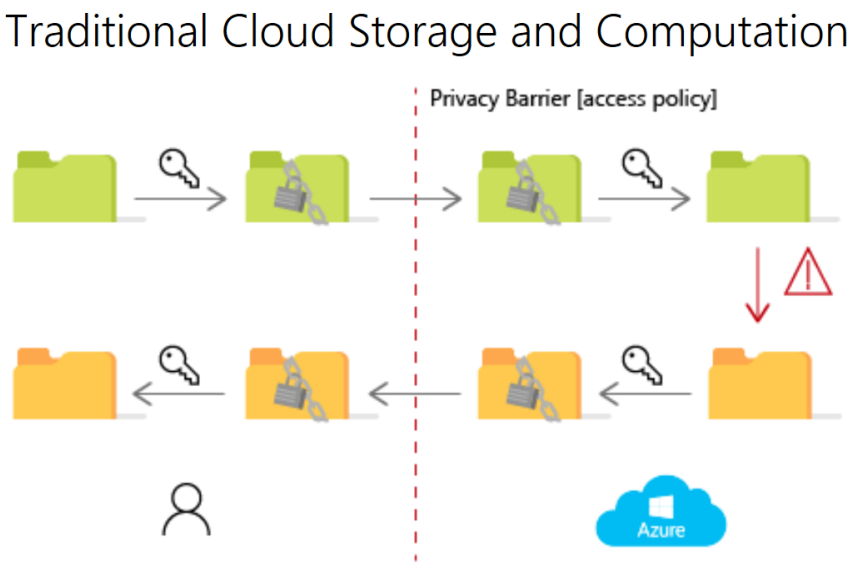
Microsoft Edge browser accused of sending full URL of web pages to Microsoft

The classic Microsoft Edge browser is accused of sending full URL of web ... This appears to be an invasion of users' privacy given the fact the SID, short for ... send full URLs if users choose to disable the SmartScreen

Jul 22, 2019

Privacy-Preserving Machine Learning (PPML)

- Traditional client-server computation undermines user privacy
- Anonymization of user-id's does not always help
  - Netflix $1M prize (anonymous movie ratings dataset)
  - Researchers were able to join with IMDB data and uncover Netflix users and their political preferences
- Cryptographic approach to PPML via Fully Homomorphic encryption



In traditional cloud storage and computation solutions the cloud need to have unencrypted access to customer data to compute on it, necessarily exposing the data to the cloud operators. Data privacy relies on access control policies implemented by the cloud and trusted by the customer.

Remarks

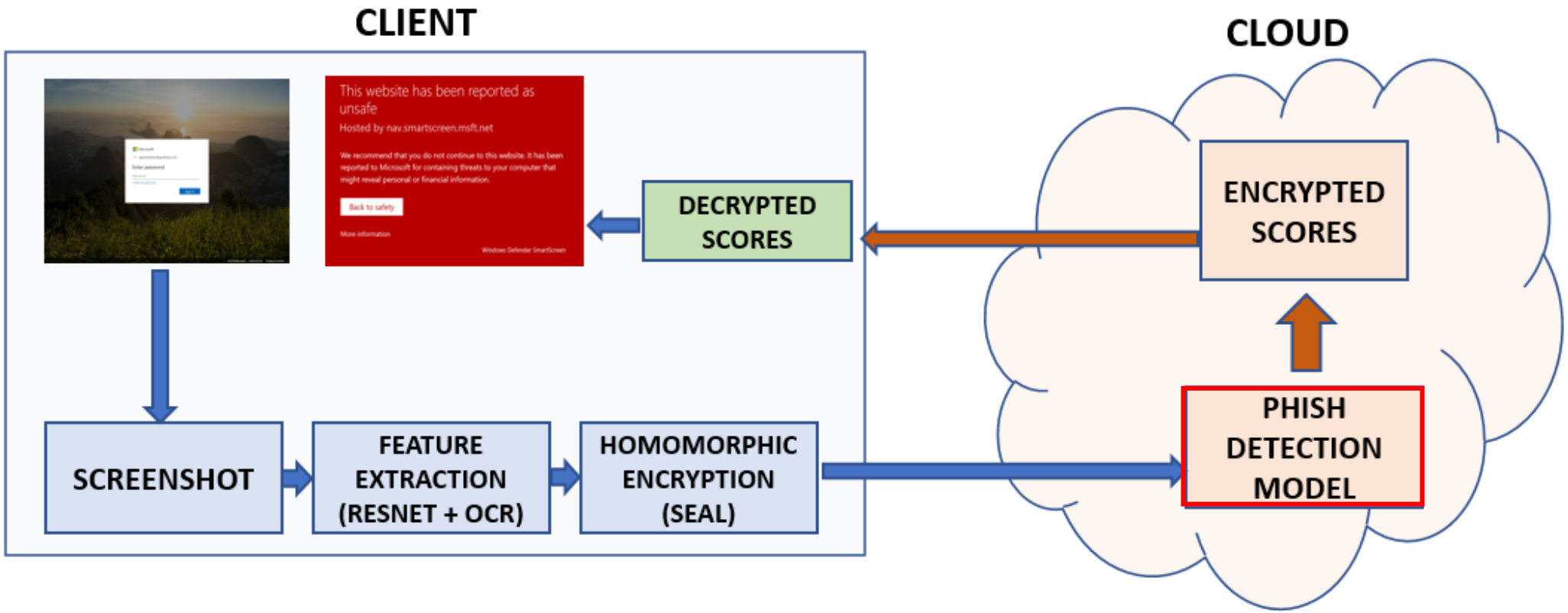
- To scope the problem down for research, we considered only a set of 20 most spoofed enterprise brands*

- The problem was framed as one of multi-category classification, where the goal was to identify the entity being impersonated
  - Total of 21 classes (20 brands + 1 “other” class)

- In the actual implementation, we will have Smart Triggers for screenshot capture based on heuristics/real-time models

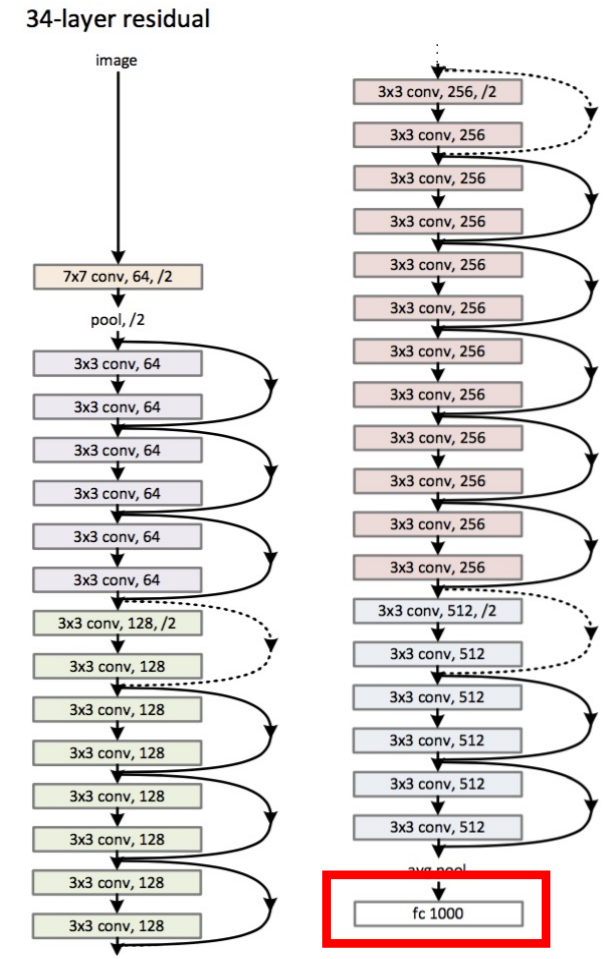
* As determined by Microsoft Defender SmartScreen

Proposed Approach: Phish Detection Model



Featurization (Client)

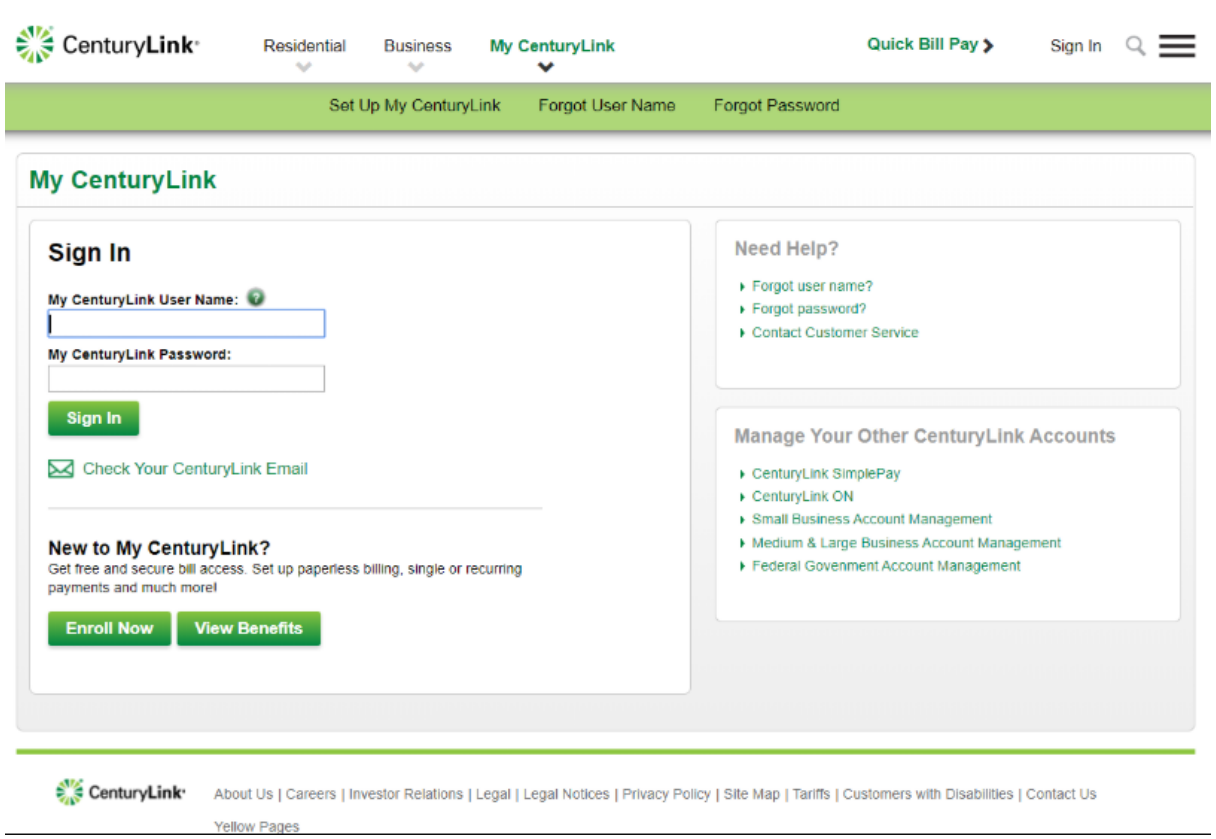
- ResNet for capturing the “Look-and-feel” of the page
- Pretrained on ImageNet (image size *224 x 224*)
- Accordingly, Web page images resized to *224 x 224* to fit into this architecture
- Output of the penultimate layer of the ResNet are used as the features

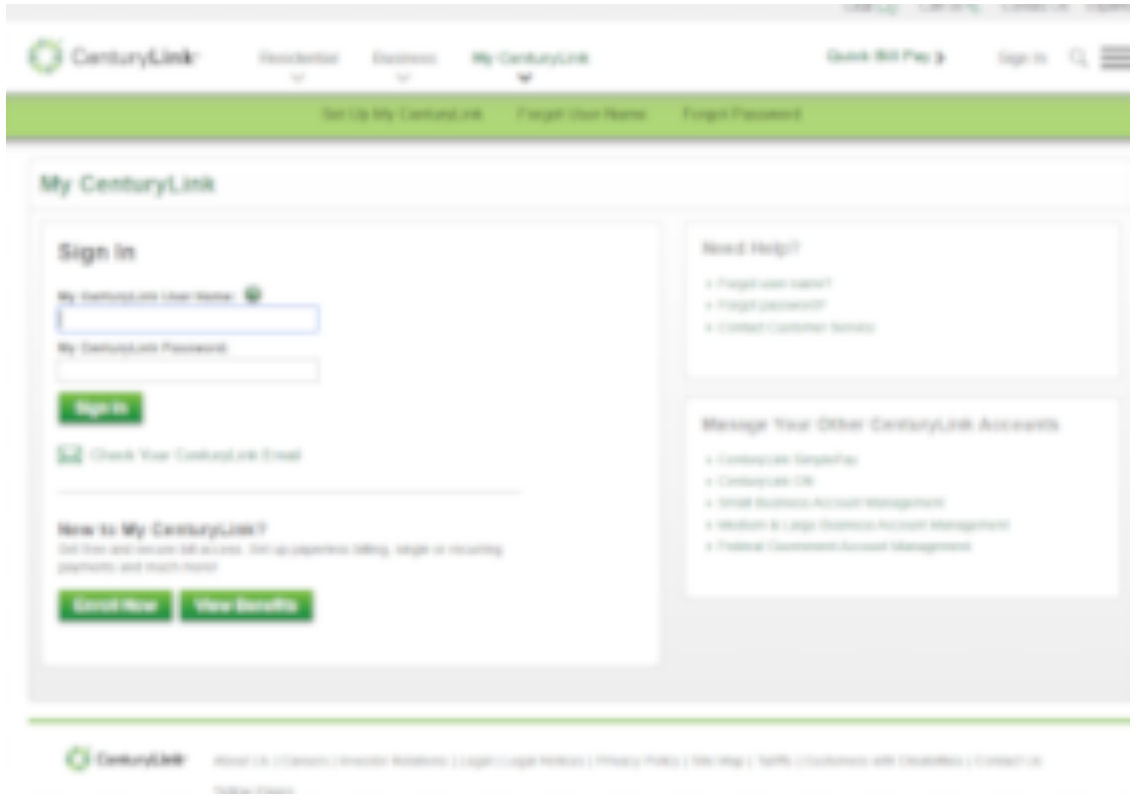


J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR 2009.
K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. 2015.

Featurization (Client)

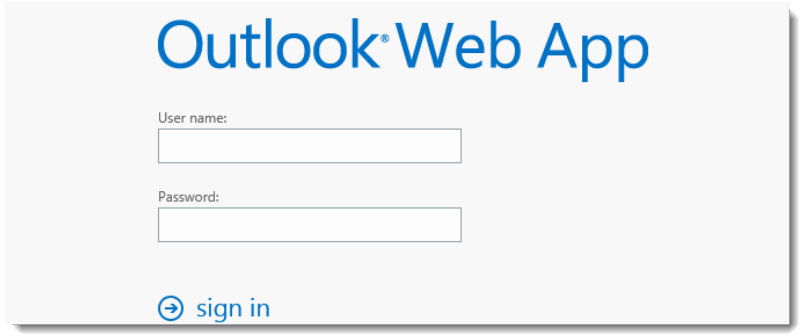
- Resizing to 224 x 224 leads to loss in textual signals in the image





Featurization (Client)

- Prior to resizing, we use an OCR engine to extract text
- We use TF-IDF (across the training data) to featurize the words in the text



Outlook Web App
User name
Password
sign in

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

- Final Feature Vector: (4096 x 1)
  - ResNet features (2048 x 1)
  - OCR + TF-IDF features (2048 x 1)

Why Featurize on Client?

FHE introduces significant redundancy in the inputs

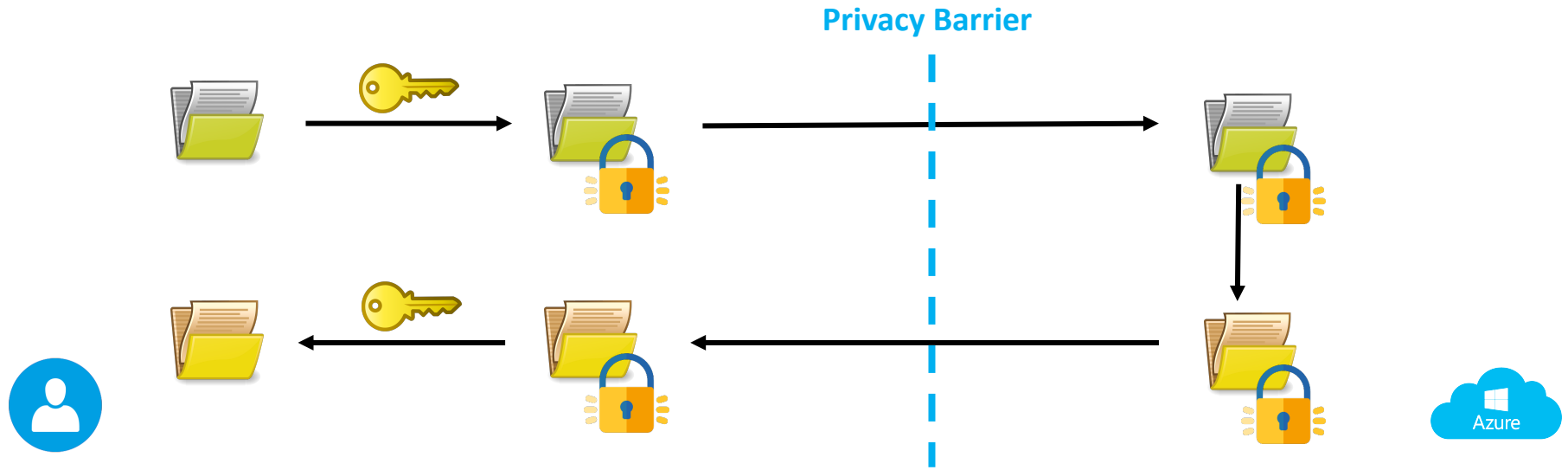
A 500 KB image when encrypted can become hundreds of MB, thus making communication costs prohibitive

Computational cost of inferencing on encrypted data increases when using complex models

Homomorphic Encryption on the Client

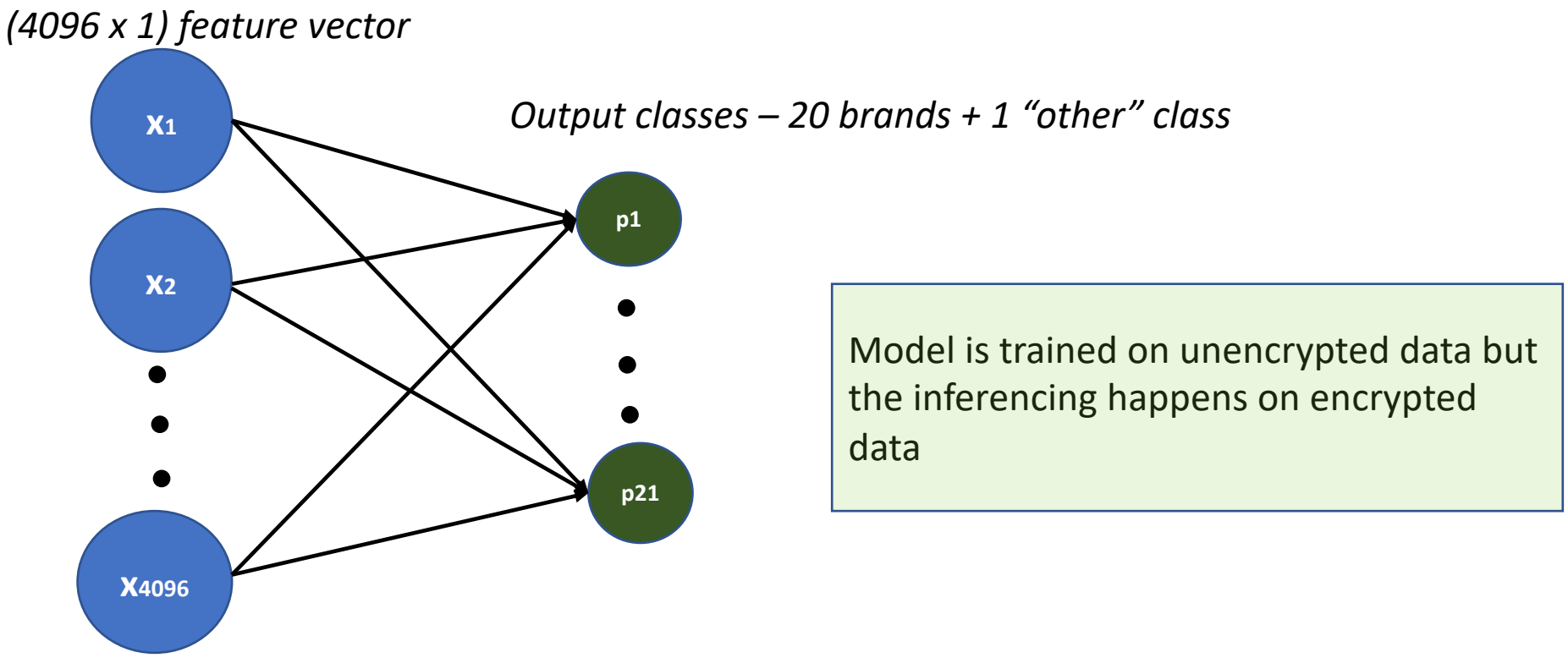
- Encrypt/decrypt on client
- Compute in cloud
- Client keeps keys; cloud learns nothing

- **Microsoft SEAL** for homomorphic encryption (CKKS)
- https://GitHub.com/Microsoft/SEAL
- Encrypt, Decrypt, Compute (add, mul, vector rotate)



Phish Detection Model (Cloud Service)

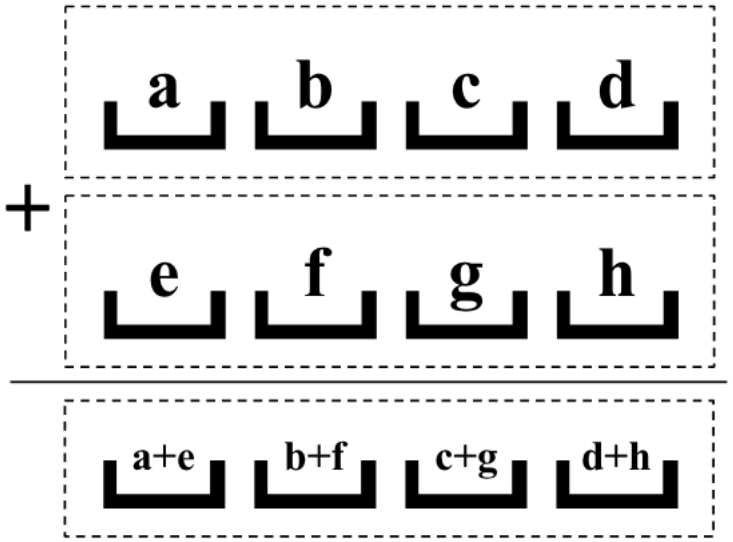
- A linear model is trained on the **unencrypted** training feature vectors and the model is deployed to the cloud for scoring, encoded in CKKS



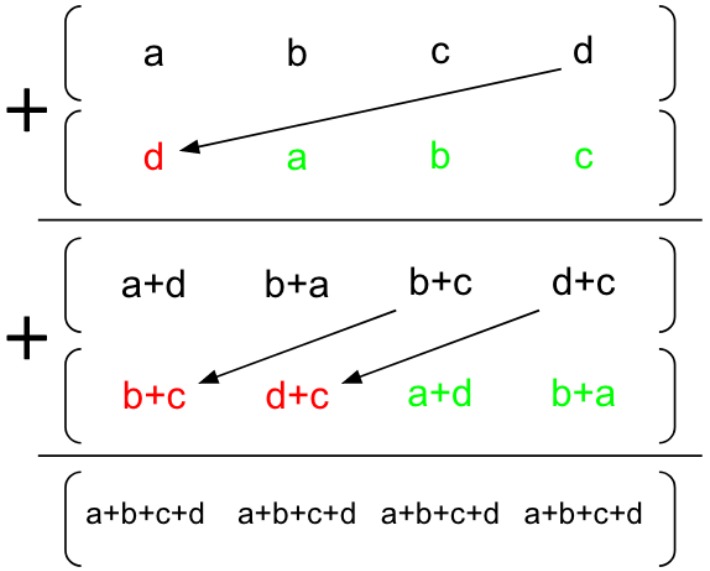
Model is trained on unencrypted data but the inferencing happens on encrypted data

Batching, Rotation, Masking

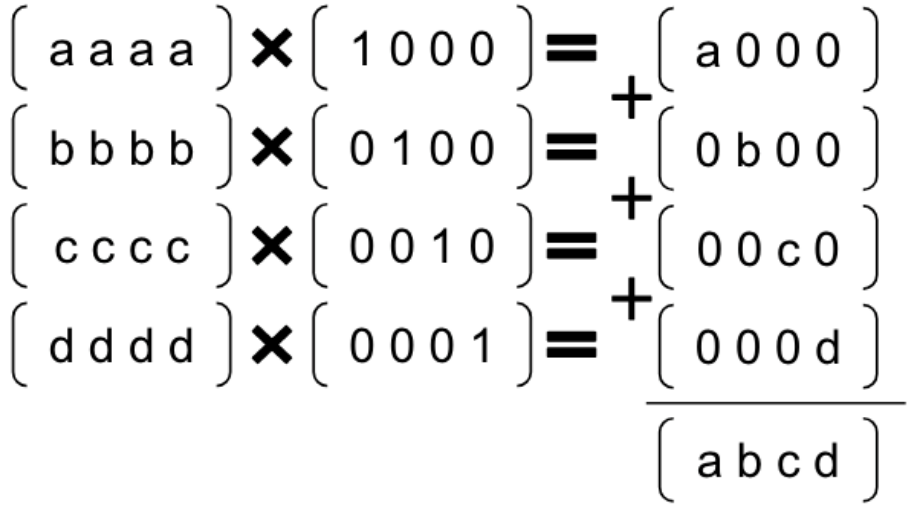
Batching



Rotation



Masking



Experiments and Results: Machine Learning

- Initially started with Resnet-only features, and then added OCR text

- Resnet + OCR gives the best predictive power (as measured by multi-class classification accuracy)

Features	Accuracy
Resnet Features	61.1%
OCR	71.25%
Resnet Features + OCR	90.57%

Experiments and Results: Homomorphic Encryption

Microsoft SEAL supports two encryption schemes: BFV and CKKS

- Encrypts very large vectors of modular integers (BFV) or doubles (CKKS)
- Operates in SIMD fashion on vectors
- Add, multiply, rotate, sum slots

Approach	Encrypt (Client)	Compute (Server)	Input Size	Output Size	Bit Precision
BFV (naïve)	13.1326 seconds	211.229 seconds	1.61GB	8.26MB	4^{th} digit
BFV (SIMD)	0.313412 seconds	1.18373 seconds	393.29KB	8.26MB	4^{th} digit
CKKS (SIMD)	0.183926 seconds	1.14705 seconds	393.29KB	3.93MB	No Loss (7^{th} digit)
CKKS (SIMD + HE expert)	0.008562 seconds	0.689715 seconds	246.825KB	88.94KB	4^{th} digit

Experiments and results: Homomorphic encryption

Unencrypted computation

- Feature vector dimension 4096 (size 33 KB)
- Evaluate one linear layer: dot product with 4096x21 weights + 21x1 bias (Wx + b)
- Output vector dimension 21x1 (size 160 B)
- Unencrypted computing time: 0.0014 sec

Encrypted computation

- Encrypted feature vector size: 247 KB
- Encrypted output vector size: 89 KB
- Encrypted computing time: 0.69 sec (single thread; fully parallelizable)

And total customer data privacy is guaranteed

Concluding Remarks on FHE

- FHE does not provide one-size-fits-all solution to privacy and comes with its set of challenges with regard to PPML.
  - FHE can be used only for inferencing and not for training
  - Cost of FHE becomes significant when conducting inferencing on very deep neural networks (> 5 layers)
  - FHE introduces significant redundancy, which means encrypting raw data is not a good idea
  - Implications are that extensive featurization is needed on the client (specifically for unstructured data) for reducing communication costs and increasing accuracy
  - Galois keys (~50MB) need to be transmitted to the cloud layer for dot product computation – not practical when we are dealing with millions of client devices
  - Cloud provider will not be aware of the decisions made on the encrypted data, which implies offline datasets needed for assessing model performance

Concluding Remarks and Key Takeways

Demonstrated a PPML using cryptographic encryption (FHE) for phish detection

FHE allows for additions/multiplications, thus enabling dot products (key to machine learning)

Accuracy using ResNet + OCR features was ~90%

Total communication costs <= 350 KB and round-trip latency < 1 second*

References

- Craig Gentry, "*Fully homomorphic encryption using ideal lattices*," in Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2009, STOC '09, pp. 169–178, ACM.
- "Microsoft SEAL (release 3.3)," https://github.com/ Microsoft/SEAL, 2019, Microsoft Research, Redmond, WA.

Thank you for viewing our presentation!