Securing Cloud-assisted Services

N. Asokan

https://asokan.org/asokan/
@nasokan
Helsinki, Finland

Home to leading universities
- University of Helsinki (top 100 overall)
- Aalto University (top 100 in Computer Science)

Innovation hub
- Local giants: Nokia, Ericsson
- Security innovators: F-Secure, SSH, ...
- Recent arrivals: Intel, Samsung, Huawei, ...
- New tigers: Rovio, Supercell, ..., lots of startups

Software Made in Finland

[Image of a Linux mascot]

Larry Ewing, http://isc.tamu.edu/~lewing/linux/

[Image of Angry Birds]

http://miss-nessa.deviantart.com/art/five-angry-birds-251226802

[Image of a SSH terminal]

https://www.coderew.com/linux/computers/ssh-add-user-remotely-script/

[Image of MySQL]


https://en.wikipedia.org/wiki/Software_Made_in_Finland

Larry Ewing, http://isc.tamu.edu/~lewing/linux/

http://miss-nessa.deviantart.com/art/five-angry-birds-251226802

https://www.coderew.com/linux/computers/ssh-add-user-remotely-script/

Services are moving to “the cloud”

I REMEMBER A TIME WHEN I HAD TO LISTEN TO THE TOPIC AT HAND BEFORE ADDING MY INSINCERE INPUT.

I THINK WE SHOULD VIRTUALIZE THE PROCESS AND MOVE IT TO THE CLOUD.

HEY, THAT’S A GREAT IDEA!

NOW IT’S JUST ALL TOO EASY.

http://dilbert.com/strip/2012-05-25
Services are moving to “the cloud”

Example: cloud storage
Example: cloud-based malware scanning service
Securing cloud storage

Client-side encryption of user data is desirable

But naïve client-side encryption conflicts with
• Storage provider business requirement: deduplication ([LPA15] ACM CCS ’15, [LDLA18] CT RSA ’18)
• End user usability requirement: multi-device access ([P+AS18] IEEE IC ’18, CeBIT ’16)

Malware checking

On-device checking
- High communication and computation costs
- Database changes frequently
- Database is revealed to everyone

Cloud-based checking
- Minimal communication and computation costs
- Database can change frequently
- Database is not revealed to everyone
- User privacy at risk!
Cloud-based malware scanning service

Needs to learn about apps installed on client devices
Can therefore infer personal characteristics of users

Predicting User Traits From a Snapshot of Apps Installed on a Smartphone

Suranga Seneviratne\textsuperscript{a,b}
suranga.seneviratne@nicta.com.au
Prasant Mohapatra\textsuperscript{c}
prasant@cs.ucdavis.edu

Aruna Seneviratne\textsuperscript{a,b}
aruna.seneviratne@nicta.com.au
Anirban Mahanti\textsuperscript{b}
anirban.mahanti@nicta.com.au

\textsuperscript{a} School of EET, University of New South Wales, Australia
\textsuperscript{b} NICTA, Australia
\textsuperscript{c} Department of Computer Science, University of California, Davis

You Are What Apps You Use:
Demographic Prediction Based on User’s Apps

Eric Malmi
Verto Analytics and Aalto University
Espoo, Finland
eric.malmi@aalto.fi

Ingmar Weber
Qatar Computing Research Institute
Doha, Qatar
iweber@qf.org.qa

Proceedings of the Tenth International AAAI Conference on Web and Social Media (ICWSM 2016)

http://dx.doi.org/10.1145/2636242.2636244
Oblivious Neural Network Predictions via MiniONN Transformations

N. Asokan
http://asokan.org/asokan/
@nasokan

(Joint work with Jian Liu, Mika Juuti, Yao Lu)
Machine learning as a service (MLaaS)

violation of clients’ privacy
Oblivious Neural Networks (ONN)

Given a neural network, is it possible to make it oblivious?

• server learns nothing about clients' input

• clients learn nothing about the model
Example: CryptoNets

- High throughput for batch queries from same client
- High overhead for single queries: 297.5s and 372MB (MNIST dataset)
- Cannot support: high-degree polynomials, comparisons, …

[GDLLNW16] CryptoNets, ICML 2016

MiniONN: Overview

- Low overhead: ~1s
- Support all common neural networks

Example \( z = W' \cdot f(W \cdot x + b) + b' \)

\[ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad W = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad W' = \begin{bmatrix} w'_{1,1} & w'_{1,2} \\ w'_{2,1} & w'_{2,2} \end{bmatrix}, \quad b' = \begin{bmatrix} b'_1 \\ b'_2 \end{bmatrix} \]

All operations are in a finite field \( \mathbb{Z}_N \)
Core idea: use secret sharing for oblivious computation

Use efficient cryptographic primitives (2PC, additively homomorphic encryption)

Client & server have shares $y^c$ and $y^s$ s.t. $y^c + y^s = y$

Client & server have shares $x^c$ and $x^s$ s.t. $x^c + x^s = x$

$y' = y^c + y^s$

$x' = x^c + x^s$

$(y'^c + y'^s = y')$

$(x'^c + x'^s = x')$

$(y^c + y^s = y)$

$(x^c + x^s = x)$
Secret sharing initial input $x$

$x_1^c, x_2^c \leftarrow Z_N$

$x_1^s := x_1 - x_1^c, \quad x_2^s := x_2 - x_2^c$

Note that $x^c$ is independent of $x$. Can be pre-chosen
Oblivious linear transformation \( W \cdot x + b \)

\[
\begin{bmatrix}
    w_{1,1} & w_{1,2} \\
    w_{2,1} & w_{2,2}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2
\end{bmatrix}
+ 
\begin{bmatrix}
    b_1 \\
    b_2
\end{bmatrix}
= 
\begin{bmatrix}
    w_{1,1} & w_{1,2} \\
    w_{2,1} & w_{2,2}
\end{bmatrix}
\begin{bmatrix}
    x_1^s + x_1^c \\
    x_2^s + x_2^c
\end{bmatrix}
+ 
\begin{bmatrix}
    b_1 \\
    b_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
    w_{1,1}(x_1^s + x_1^c) + w_{1,2}(x_2^s + x_2^c) + b_1 \\
    w_{2,1}(x_1^s + x_1^c) + w_{2,2}(x_2^s + x_2^c) + b_2
\end{bmatrix}
= 
\begin{bmatrix}
    w_{1,1}x_1^s + w_{1,2}x_2^s + b_1 + w_{1,1}x_1^c + w_{1,2}x_2^c \\
    w_{2,1}x_1^s + w_{2,2}x_2^s + b_2 + w_{2,1}x_1^c + w_{2,2}x_2^c
\end{bmatrix}
\]

Compute locally by the server

Dot-product
Oblivious linear transformation: dot-product

$\mathbf{u} + \mathbf{v} = \mathbf{W} \cdot \mathbf{x}^c$; Note: $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{W} \cdot \mathbf{x}^c$ are independent of $\mathbf{x}$.

$<\mathbf{u}, \mathbf{v}, \mathbf{x}^c>$ generated/stored in a precomputation phase
Oblivious linear transformation $W \cdot x + b$

\[
\begin{bmatrix}
  w_{1,1} & w_{1,2} \\
  w_{2,1} & w_{2,2}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix}
+ \begin{bmatrix}
  b_1 \\
  b_2
\end{bmatrix}
= \begin{bmatrix}
  w_{1,1} & w_{1,2} \\
  w_{2,1} & w_{2,2}
\end{bmatrix}
\begin{bmatrix}
  x_1^s + x_1^c \\
  x_2^s + x_2^c
\end{bmatrix}
+ \begin{bmatrix}
  b_1 \\
  b_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
  w_{1,1}(x_1^s + x_1^c) + w_{1,2}(x_2^s + x_2^c) + b_1 \\
  w_{2,1}(x_1^s + x_1^c) + w_{2,2}(x_2^s + x_2^c) + b_2
\end{bmatrix}
= \begin{bmatrix}
  w_{1,1}x_1^s + w_{1,2}x_2^s + b_1 + w_{1,1}x_1^c + w_{1,2}x_2^c \\
  w_{2,1}x_1^s + w_{2,2}x_2^s + b_2 + w_{2,1}x_1^c + w_{2,2}x_2^c
\end{bmatrix}
\]

\[
\begin{bmatrix}
  w_{1,1}x_1^s + w_{1,2}x_2^s + b_1 + u_1 \\
  w_{2,1}x_1^s + w_{2,2}x_2^s + b_2 + u_2
\end{bmatrix}
+ \begin{bmatrix}
  v_1 \\
  v_2
\end{bmatrix}
\]
Oblivious linear transformation \( \mathbf{W} \cdot \mathbf{x} + \mathbf{b} \)

\[
\begin{bmatrix}
    w_{1,1} & w_{1,2} \\
    w_{2,1} & w_{2,2}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2
\end{bmatrix}
+ 
\begin{bmatrix}
    b_1 \\
    b_2
\end{bmatrix}
= 
\begin{bmatrix}
    w_{1,1} & w_{1,2} \\
    w_{2,1} & w_{2,2}
\end{bmatrix}
\begin{bmatrix}
    x_1^s + x_1^c \\
    x_2^s + x_2^c
\end{bmatrix}
+ 
\begin{bmatrix}
    b_1 \\
    b_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
    w_{1,1}(x_1^s + x_1^c) + w_{1,2}(x_2^s + x_2^c) + b_1 \\
    w_{2,1}(x_1^s + x_1^c) + w_{2,2}(x_2^s + x_2^c) + b_2
\end{bmatrix}
= 
\begin{bmatrix}
    w_{1,1}x_1^s + w_{1,2}x_2^s + b_1 + w_{1,1}x_1^c + w_{1,2}x_2^c \\
    w_{2,1}x_1^s + w_{2,2}x_2^s + b_2 + w_{2,1}x_1^c + w_{2,2}x_2^c
\end{bmatrix}
\]

\[
\begin{bmatrix}
    w_{1,1}x_1^s + w_{1,2}x_2^s + b_1 + u_1 \\
    w_{2,1}x_1^s + w_{2,2}x_2^s + b_2 + u_2
\end{bmatrix}
+ 
\begin{bmatrix}
    v_1 \\
    v_2
\end{bmatrix}
= 
\begin{bmatrix}
    y_1^s \\
    y_2^s
\end{bmatrix}
+ 
\begin{bmatrix}
    y_1^c \\
    y_2^c
\end{bmatrix}
\]
Recall: use secret sharing for oblivious computation

client & server have shares $y^c$ and $y^s$, s.t. $y^c + y^s = y$

client & server have shares $x^c$ and $x^s$, s.t. $x^c + x^s = x$

$W \cdot [] + b$

$(y^c + y^s = y)$

$(x^c + x^s = x)$
Oblivious activation/pooling functions $f(y)$

Piecewise linear functions e.g.,

- **ReLU**: $x := \max(y, 0)$
- **Oblivious ReLU**: $x^s + x^c := \max(y^s + y^c, 0)$
  - easily computed obliviously by a garbled circuit
Oblivious activation/pooling functions $f(y)$

Smooth functions e.g.,

- **Sigmoid**: $x := 1 / (1 + e^{-y})$
- **Oblivious sigmoid**: $x^s + x^c := 1 / (1 + e^{-(y^s + y^c)})$
  - approximate by a piecewise linear function
  - then compute obliviously by a garbled circuit
  - empirically: ~14 segments sufficient
Combining the final result

They can jointly calculate \( \max(y_1, y_2) \) (for minimizing information leakage)

\[
y_1 := y_1^s + y_1^c
\]
\[
y_2 := y_2^s + y_2^c
\]
Recall: use secret sharing for oblivious computation

\[ W' \cdot [] + b' \]

\[ f([]) \]

\[ W \cdot [] + b \]

\[ (y'^c + y'^s = y') \]

\[ (x'^c + x'^s = x') \]

\[ (y^c + y^s = y) \]

\[ (x^c + x^s = x) \]
## Performance (for single queries)

<table>
<thead>
<tr>
<th>Model</th>
<th>Latency (s)</th>
<th>Msg sizes (MB)</th>
<th>Loss of accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST/Square</td>
<td>0.4 (+ 0.88)</td>
<td>44 (+ 3.6)</td>
<td>none</td>
</tr>
<tr>
<td>CIFAR-10/ReLU</td>
<td>472 (+ 72)</td>
<td>6226 (+ 3046)</td>
<td>none</td>
</tr>
<tr>
<td>PTB/Sigmoid</td>
<td>4.39 (+ 13.9)</td>
<td>474 (+ 86.7)</td>
<td>Less than 0.5% (cross-entropy loss)</td>
</tr>
</tbody>
</table>

Pre-computation phase timings in parentheses
PTB = Penn Treebank
## MiniONN pros and cons

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>300-700x faster than CryptoNets</td>
<td>Still ~1000x slower than without privacy</td>
</tr>
<tr>
<td>Can transform any given neural network to its oblivious variant</td>
<td>Server can no longer filter requests or do sophisticated metering</td>
</tr>
<tr>
<td>Reveals structure (but not params) of NN</td>
<td></td>
</tr>
</tbody>
</table>

---

44
Hardware-security mechanisms are pervasive

Hardware support for
- Isolated execution: Isolated Execution Environment
- Protected storage: Sealing
- Ability to convince remote verifiers: Remote Attestation

Trusted Execution Environments (TEEs)
Operating in parallel with “rich execution environments” (REEs)

Cryptocards Trust Platform Modules ARM TrustZone Intel Software Guard Extensions


Using a server-side TEE to run the model

1. Attest server’s NN Engine (in TEE)
2. Input
3. Provision model configuration, filtering policy
4. Prediction

MiniONN + policy filtering + advanced metering + performance + better model secrecy
- harder to reason about client input privacy
MiniONN: Efficiently transform any given neural network into oblivious form with no/negligible accuracy loss

Try at: https://github.com/SSGAalto/minionn

[LJLA17] MiniONN, ACM CCS 2017,
[KNLAS19] Private Decision Trees, PETS 2019

Trusted Computing can help realize improved security and privacy for ML

ML is very fragile in adversarial settings
Securing cloud-assisted services

Cloud-assisted services raise new security/privacy concerns

• But naïve solutions may conflict with privacy, usability, deployability, …

Solutions using Trusted hardware + cryptography

[TLPEPA17] Circle Game, ACM ASIACCS 2017
Supplementary material
Background: Kinibi on ARM TrustZone

Kinibi
- Trusted OS from Trustonic

Remote attestation
- Establish a trusted channel

Private memory
- Confidentiality
- Integrity
- Obliviousness

https://www.trustonic.com/solutions/trustonic-secured-platforms-tsp/
Background: Intel SGX

CPU enforced TEE (enclave)

Remote attestation

Secure memory
- Confidentiality
- Integrity

Obliviousness only within 4 KB page granularity

https://software.intel.com/sgx
Android app landscape

On average a user installs 95 apps (Yahoo Aviate)
Yahoo Aviate study
Source: https://yahooaviate.tumblr.com/image/95795838933

Unique new Android malware samples
Source: G Data
2015: https://secure.gd/dl-en-mmwr201504

Current dictionary size < $2^{24}$ entries

Even comparatively “high” FPR (e.g., $\sim 2^{-10}$) may have negligible impact on privacy
Cloud-scale PMT

**Verify Apps**: cloud-based service to check for harmful Android apps prior to installation

“… over 1 billion devices protected by Google’s security services, and over 400 million device security scans were conducted per day”

*Android Security 2015 Year in Review*

“2 billion+ Android devices checked per day”

[https://www.android.com/security-center/](https://www.android.com/security-center/)

(c.f. < 17 million malware samples)
Performance: batch queries

Kinibi on ARM
TrustZone

Intel SGX