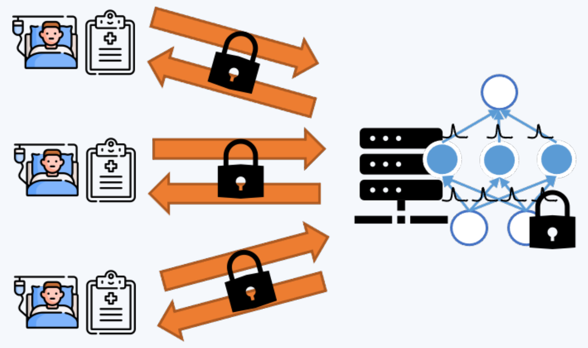
Goten: GPU-Outsourcing Trusted Execution of Neural Network Training

Lucien K. L. Ng, Sherman S. M. Chow, Anna P. Y. Woo, Donald P. H. Wong (CUHK), and Yongjun Zhao (CUHK → NTU)

Department of Information Engineering, Chinese University of Hong Kong (CUHK), Hong Kong

Privacy of "Big" Training Data

- *Sensitive*
  - Medical Image analysis, Child Exploitation Imagery, *etc.*
  - Privacy laws & Regulations, *e.g.*, GDPR
- *Massive*
  - Hardly any single entity's data is sufficient
- Private Training
  - No one learns anything about the **model** & other's **data**



Why Federated Learning is not enough?

- Federated Learning:
  - Each data contributor train DNN locally
  - They exchange the DNN's weight frequently
- Problems:
  - Every contributor can use the DNN
  - » No rate-limiting, even for non-agreed uses
  - Contributors may steal others' data
  - » Model Inversion Attack [Fredrikson et al.]
  - Noisy/Implicit data ⇏ Data privacy

xkcd.com/2169

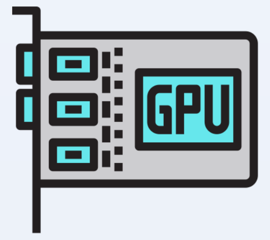
Preliminary: TEE & GPU

TEE: Trusted Execution Environment (*e.g.*, SGX)

- 😄 protect the data's privacy inside
  - **even** *the machine owner* cannot read it
- 😄 processes data efficiently as plaintext on CPU
  - too ***slow*** for batched linear operations

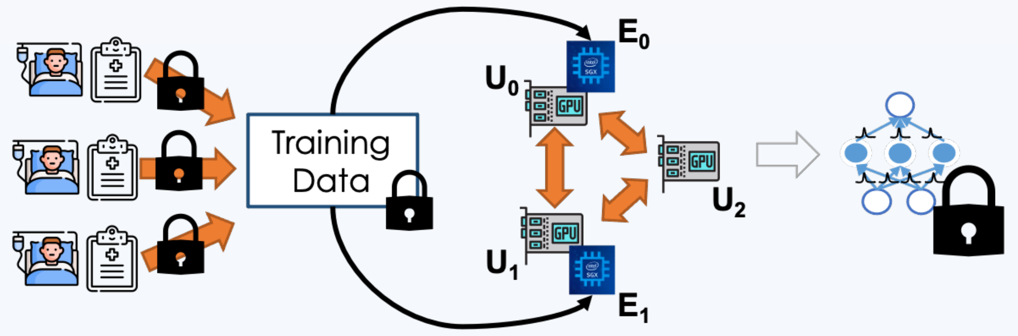
GPU: Graphics Processing Unit

- 😄 GPU can speed up the linear layers in DNNs
  - The **most time-consuming part** in DNNs
- 😟 GPU does not have TEE
  - lack of data privacy & model privacy!



Goten: GPU + TEE for Private Training

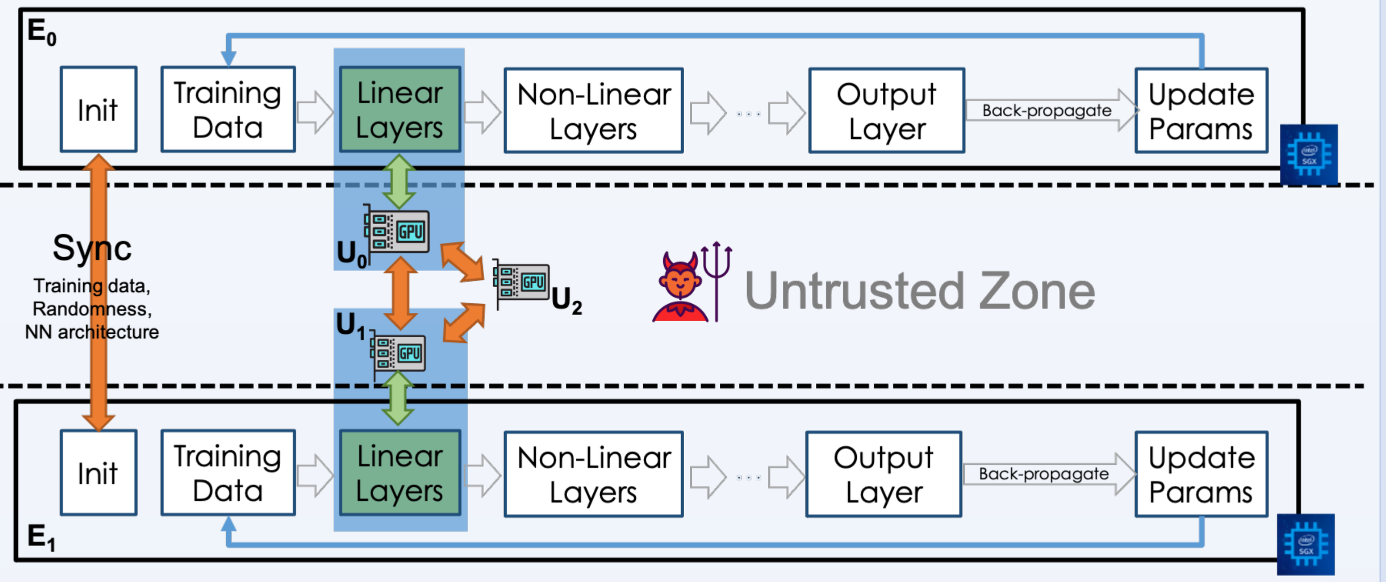
- Contributors send their data to SGX's TEE/enclaves
- *Securely* outsource linear-layer computation to GPUs
  - resided in with 3 non-colluding servers (**U_0, U_1, U_2**)
  - can reduce to 2 servers (at ½ of the throughput)
- Train (mostly) non-linear layers in SGX



Non-Colluding Servers in Goten

- Each server holds a secret-share of the model/data
- Individual share by itself is totally meaningless
- Candidates:
  - Some of the Data Contributors
  - Government: Hospital/Monetary authority
  - Independent & Competing Cloud Server Providers

GPU-Outsourcing Protocol for Linear Layers (Overview)



(Light-Weight) Crypto Tool: Additive Secret Shares (SS)

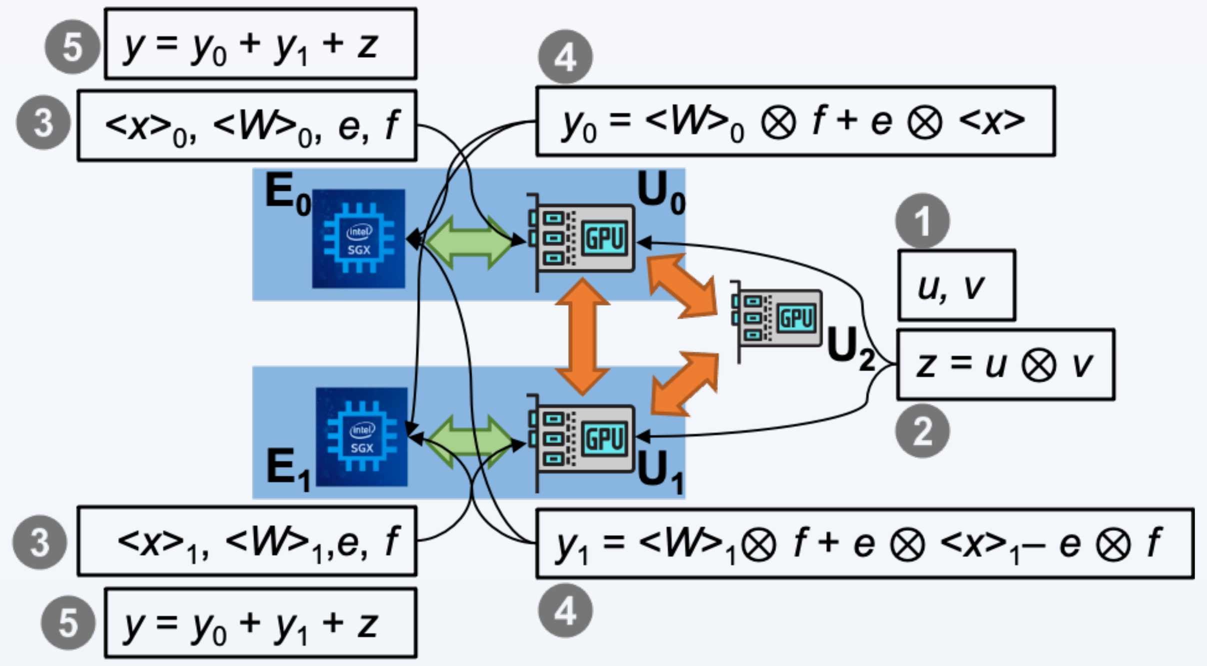
- $x = \langle x\rangle_0 + \langle x\rangle_1 \pmod q$
  - $\langle x\rangle_0$ and $\langle x\rangle_1$ is a pair of additive SSs for x
- Privacy ($\langle x\rangle_i$ has no information about x)
  - For each value of x, given $\langle x\rangle_i$, ∃ corresponding $\langle x\rangle_{1-i}$
- (Efficient) Homomorphic operation:
  - $\langle x\rangle + \langle y\rangle = \langle x + y\rangle$
  - For brevity, we will omit (mod q)

GPU-Outsourcing Protocol for Linear Layers (Details)

- Goal: Compute $y = W \otimes x$ ($\otimes$ is the linear operation)
- Without leaking any (W, x, y) to (**U_0, U_1, U_2**)

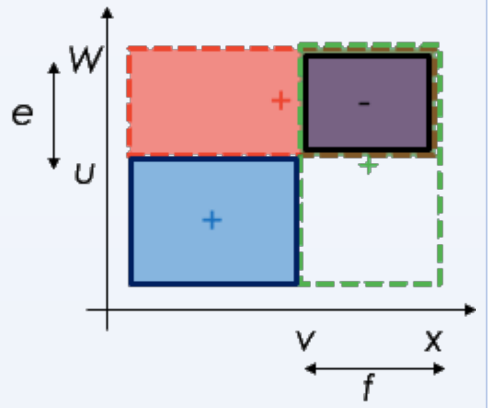
1: $U_2 : u \leftarrow \mathrm{Rand}(r_u), v \leftarrow \mathrm{Rand}(r_v)$
2: $U_2 \rightarrow E_0, E_1 : z = u \otimes v$
for $i = 0, 1$ (in parallel)
3: $E_i \rightarrow U_i : \langle W\rangle_i \leftarrow \mathrm{Gen}_i(r_W, W), \langle x\rangle_i \leftarrow \mathrm{Gen}_i(r_x, x),$
$e = W - \mathrm{Rand}(r_u), f = x - \mathrm{Rand}(r_v)$
4: $U_i \rightarrow E_0, E_1 : y_i = \langle W\rangle_i \otimes f + e \otimes \langle x\rangle_i - i \cdot e \otimes f$
endfor
5: $E_0, E_1 : y = z + y_0 + y_1$

- $\mathrm{Gen}_0(r_x, x)$ and $\mathrm{Gen}_1(r_x, x)$ are generators for $\langle x\rangle_0$ and $\langle x\rangle_1$
- $\mathrm{Rand}(r)$ is a secure pseudo-random generator
- $\{r_u, r_v, r_x, r_W\}$ are pre-agreed random seeds



Correctness

$$
\begin{aligned}
y &= y_0 + y_1 + z \\
&= \langle W\rangle_0 \otimes f + e \otimes \langle x\rangle_0 \\
&\quad + \langle W\rangle_1 \otimes f + e \otimes \langle x\rangle_1 - e \otimes f + u \otimes v \\
&= W \otimes f + e \otimes x - e \otimes f + u \otimes v \\
&= W \otimes x
\end{aligned}
$$



Security

- What each non-colluding server sees:
  - **U0**: $\langle W\rangle_0$, $\langle x\rangle_0$, e, f
  - **U1**: $\langle W\rangle_1$, $\langle x\rangle_1$, e, f
  - **U2**: u, v
- They are all secret shares or random tensors:
  - $\langle W\rangle_{0/1}$ and $\langle x\rangle_{0/1}$ are secret shares (by definition)
  - $e = (W - u)$ and $f = (x - v)$ are secret shares
  - u and v are random tensors

Performance on Training

CIFAR-10: Common Benchmark for Computer Vision

- Goten attains >89% accuracy in 34 hours
  - vs. Falcon's 5 weeks (accuracy not reported)
- 132× throughput speed up over Falcon
- Falcon [Sameer Wagh et al.]: State-of-the-Art Crypto Approach

Framework	GPU | TEE	DNN Arch.	Throughput	Speedup
Falcon	✗ | ✗	VGG-16	1482	132×
CaffeScone[*]	✗ | ✓	VGG-11	28800	6.84×
Goten	✓ | ✓	VGG-11	196733	-

[*] Our pure-TEE private training framework over Caffe & SCONE (Secure Container Environment)

Training for Invasive Ductal Carcinoma (IDC) Detection

- Showcase application involving sensitive training data
- IDC: The most common type of breast cancer
- Dataset: Images of women's breast tissue [Cruz-Roa et al.]

Accuracy	**81%**	**82%**	**83%**	**84%**	**85%**	**86%**
Speedup	8.53×	13.7×	4.27×	6.33×	3.42×	7.28×
Time (min)	1.25	1.56	13.1	16.9	31.2	46.8

- GPU: Nvidia V100 16GB
- CPU (w/ SGX): Intel i7-7700K
- Network: Google Cloud (8Gbps & <5ms latency)

Conclusion

- Best of Both Worlds: TEE & GPU
- Our Techniques:
  - Lightweight Crypto for GPU-Outsourcing
  - Dynamic Quantization for Weight Fluctuation during Training
- Future Work: GPU-Friendly Pure-Crypto Solution [Ng and Chow]
- Code: `github.com/goten-team/Goten`

References

- Matt Fredrikson, Somesh Jha, Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. CCS '15.
- Sameer Wagh, Shruti Tople, Fabrice Benhamouda, Eyal Kushilevitz, Prateek Mittal, and Tal Rabin. Falcon: Honest-Majority Maliciously Secure Framework for Private Deep Learning. PETS '21.
- Cruz-Roa *et al.* Automatic Detection of Invasive Ductal Carcinoma in Whole Slide Images with Convolutional Neural Networks. Medical Imaging: Digital Pathology '14.
- Lucien K. L. Ng, Sherman S. M. Chow. GPU-Friendly Oblivious and Rapid Classification Engine. Usenix Security '21.

{luciengkl, sherman}@ie.cuhk.edu.hk