Visor:
Privacy-Preserving Video Analytics as a Cloud Service

Rishabh Poddar, Ganesh Ananthanarayanan, Srinath Setty, Stavros Volos, and Raluca Ada Popa

UC Berkeley and Microsoft Research
Video analytics as a service

ML applications driven by decisions based on objects in videos

• Raise an alert if a person at the main door

• Change the traffic lights if less than 3 cars at the intersection
Example video analytics pipeline

Client source → Cloud platform → Video decoding → Background subtraction → Bounding box detection → Object cropping → CPU
Example video analytics pipeline

Cheap filters to discard irrelevant content
- CPU is under-utilized, unlike GPU
- Up to 17x better than GPU-only processing*

* Live Video Analytics with Microsoft Rocket for reducing edge compute costs
Example video analytics pipeline

Decode video into sequence of raw frames

Client source

Cloud platform

Video decoding

Background subtraction

Bounding box detection

Object cropping
Example video analytics pipeline

Client source → Cloud platform

1. Video decoding
2. Background subtraction
3. Bounding box detection
4. Object cropping

Extract moving “foreground” objects
Example video analytics pipeline

Detect bounding box of each foreground object

- Video decoding
- Background subtraction
- Bounding box detection
- Object cropping
Example video analytics pipeline

Crop out objects using bounding boxes

Client source → Video decoding → Background subtraction → Bounding box detection → Object cropping
Example video analytics pipeline

1. Red car
2. White van
3. Tree
4. …

- Video decoding
- Background subtraction
- Bounding box detection
- Object cropping

CPU

GPU

Client source

Cloud platform

CNN classification

Objects

Results
Problem: Privacy

Video stream is **proprietary** to clients

Malicious attacker
(Hackers, co-tenants, rogue employees)
Related work: Privacy-preserving inference

Cryptographic approaches: e.g. SMPC, Homomorphic encryption [MZ17, JVC18]

• High overhead — cannot sustain video frame rate
  (10s to 100s of seconds per frame)
Related work: Privacy-preserving inference

Cryptographic approaches: e.g. SMPC, Homomorphic encryption [MZ17, JVC18]
  • High overhead — cannot sustain video frame rate (10s to 100s of seconds per frame)

Trusted hardware-based approaches (enclaves): e.g. Slalom [TB19], Chiron [HSS+18]
  • Designed for CPU enclaves — cannot sustain frame rate without GPUs (order of seconds per frame)
  • Side-channel leakage — e.g. memory access patterns
Visor
Key contributions

1. Privacy-preserving framework for machine learning services
   • Hybrid enclave architecture (combines CPU and GPU enclaves into a unified trust domain)

2. Novel data-oblivious algorithms for video analytics
   • Several orders of magnitude better than alternate solutions for data-obliviousness
Key contributions

1. Privacy-preserving framework for machine learning services
   - **Hybrid enclave** architecture (combines CPU and GPU enclaves into a unified trust domain)

2. Novel data-oblivious algorithms for video analytics
   - Several orders of magnitude better than alternate solutions for data-obliviousness
Hybrid trusted execution environment (TEE)

**Intel SGX (CPU TEE)**

- Client source
- Cloud platform
- Operating System (untrusted)
- Application (trusted)
- CPU TEE
- CPU
- CPU TEE
- Application (trusted)
- Operating System (untrusted)
- CPU
Hybrid trusted execution environment (TEE)

Intel SGX (CPU TEE) + Graviton [VVB18] (GPU TEE)

Isolates kernels from other GPU code and privileged software on the host

Client source

Cloud platform

Operating System (untrusted)

Application (trusted)

CPU TEE

GPU TEE

Other processes (untrusted)

Trusted process
Hybrid trusted execution environment (TEE)

Intel SGX (CPU TEE) + Graviton [VVB18] (GPU TEE)

**Isolates kernels from other GPU code and privileged software on the host**
Hybrid trusted execution environment (TEE)

Video analytics pipeline within the hybrid TEE

Client source

Cloud platform

Video decoding → Background subtraction → Bounding box detection → Object cropping

CPU TEE

GPU TEE

CNN classification
Mitigating leakage

Client source  → Cloud platform

1. Video decoding → Background subtraction → Bounding box detection → Object cropping

2. CNN classification

Encrypted traffic pattern leakage
- E.g. **times of activity** in video stream
- E.g. **number / size** of objects per frame

Cloud platform
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance

- Padded video streams
- Data-oblivious CPU-GPU queues

Client source → Video decoding → Background subtraction → Bounding box detection → Object cropping → CNN classification

Cloud platform → CPU TEE → GPU TEE
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
Data-oblivious CPU-GPU queue that minimizes GPU consumption

- CPU pushes $N_1$ objects into queue per frame, comprised of **real** and **dummy** objects

**Data-oblivious communication protocols** optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Mitigating leakage
Mitigating leakage

Data-oblivious CPU-GPU queue that minimizes GPU consumption

- Obliviously sort the queue, pushing real objects to the tail

Data-oblivious communication protocols optimized for pipeline performance

- Padded video streams
- Data-oblivious CPU-GPU queues
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- GPU consumes $N_2$ objects from the tail ($N_2 < N_1$)
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- Cycle repeats: push, sort, consume

Object cropping

CNN classification

GPU TEE

---

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- Cycle repeats: push, sort, consume
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- Cycle repeats: push, sort, consume

Overwritten objects — these are dummies with a large probability
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- Cycle repeats: push, sort, consume

Object cropping

CNN classification

GPU TEE
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- Cycle repeats: push, sort, consume
Mitigating leakage

Data-oblivious communication protocols optimized for pipeline performance
- Padded video streams
- Data-oblivious CPU-GPU queues

Data-oblivious CPU-GPU queue that minimizes GPU consumption
- Implemented using a circular buffer, with priorities associated with each object

```
| N1 | N2 |
```

Object cropping
CNN classification
GPU TEE
Mitigating leakage

Memory access pattern leakage in TEEs
- Large subset of known side-channel attacks on CPU TEEs (Intel SGX)

Client source

Cloud platform

Video decoding → Background subtraction → Bounding box detection → Object cropping

CNN classification

CPU TEE

GPU TEE
Mitigating leakage

Data-oblivious modules
- Suite of primitives for developing data-oblivious CPU modules
- Data-oblivious CNN on the GPU

Client source

Cloud platform

3

Video decoding → Background subtraction → Bounding box detection → Object cropping

CPU TEE

4

CNN classification

GPU TEE

Data-oblivious modules
Key contributions

1. Privacy-preserving framework for machine learning services
   - Hybrid enclave architecture (combines CPU and GPU enclaves into a unified trust domain)

2. Novel data-oblivious algorithms for video analytics
   - Several orders of magnitude better than alternate solutions for data-obliviousness
Memory access pattern leakage

Example: Bounding box detection module
Memory access pattern leakage

Original bounding box detection algorithm:

- Detect white pixel, then follow border (leaks object shape and location)
Memory access pattern leakage

Original image (processed within SGX enclave)

Location of accessed pixels

Recovered image: Leaks shapes and positions of all objects
Design approach

**Goal:** Transform algorithm into a pattern that processes each pixel identically

**How to apply this design pattern efficiently?**

- **General strategy** based the properties of vision modules
- Exploits the structure of the algorithms for efficiency
Illustration: Bounding box detection

Oblivious algorithm strategy:

1. Cast algorithm into a form that scans the image
   • Assign a label to each pixel based on its neighbors
Illustration: Bounding box detection

Oblivious algorithm strategy:

1. Cast algorithm into a form that scans the image
   - Assign a label to each pixel based on its neighbors
Illustration: Bounding box detection

Oblivious algorithm strategy:

1. Cast algorithm into a form that scans the image
   - Assign a label to each pixel based on its neighbors
   - Perform another pass to resolve connected labels
Illustration: Bounding box detection

Oblivious algorithm strategy:

1. Divide-and-conquer to improve performance
   - Detect bounding boxes in separate strips of the image
Illustration: Bounding box detection

Oblivious algorithm strategy:

2. Divide-and-conquer to improve performance
   - Detect bounding boxes in separate strips of the image
   - Resolve boundaries
Illustration: Bounding box detection

Oblivious algorithm strategy:

3. Amortize processing cost across groups of pixels
   • E.g. use SIMD instructions to vectorize pixel operations
Key contributions

1. Privacy-preserving framework for machine learning services
   • **Hybrid enclave** architecture (combines CPU and GPU enclaves into a unified trust domain)

2. Novel data-oblivious algorithms for video analytics
   • Several orders of magnitude better than alternate solutions for data-obliviousness

*Formal proofs of security* for the algorithms present in the extended version of the paper
Evaluation highlights
Evaluation setup

Testbed
• Intel i7-8700K with 6 cores at 3.7 GHz
• NVIDIA GTX 780 GPU with 2304 CUDA cores operating at 863 MHz

Workload
• 4 real HD video streams (traffic / security cameras)
• 2 video analytics pipelines (object detection and tracking)
End-to-end overhead

- Overhead of hybrid enclave up to 8x with limited enclave memory (improves with sufficient enclave memory)
- Overhead of data-obliviousness 2.4x
- Over 6 orders of magnitude better than general-purpose alternatives \([AJX+19], [RLT15]\)
Summary

Visor

- Privacy-preserving framework for CNN-based services spanning both CPU and GPU resources based on a hybrid enclave design
- Novel data-oblivious algorithms for video analytics pipelines
- End-to-end overhead 2.5x — 20x

Thanks!

rishabhp @ berkeley.edu