iGPU Leak

An Information Leakage Vulnerability on Intel Integrated GPU

Wenjian He *Hong Kong Univ. of Science and Technology*

Wei Zhang *Hong Kong Univ. of Science and Technology*

Sharad Sinha *Indian Institute of Technology, Goa*

Sanjeev Das *Univ. of North Carolina at Chapel Hill, USA*

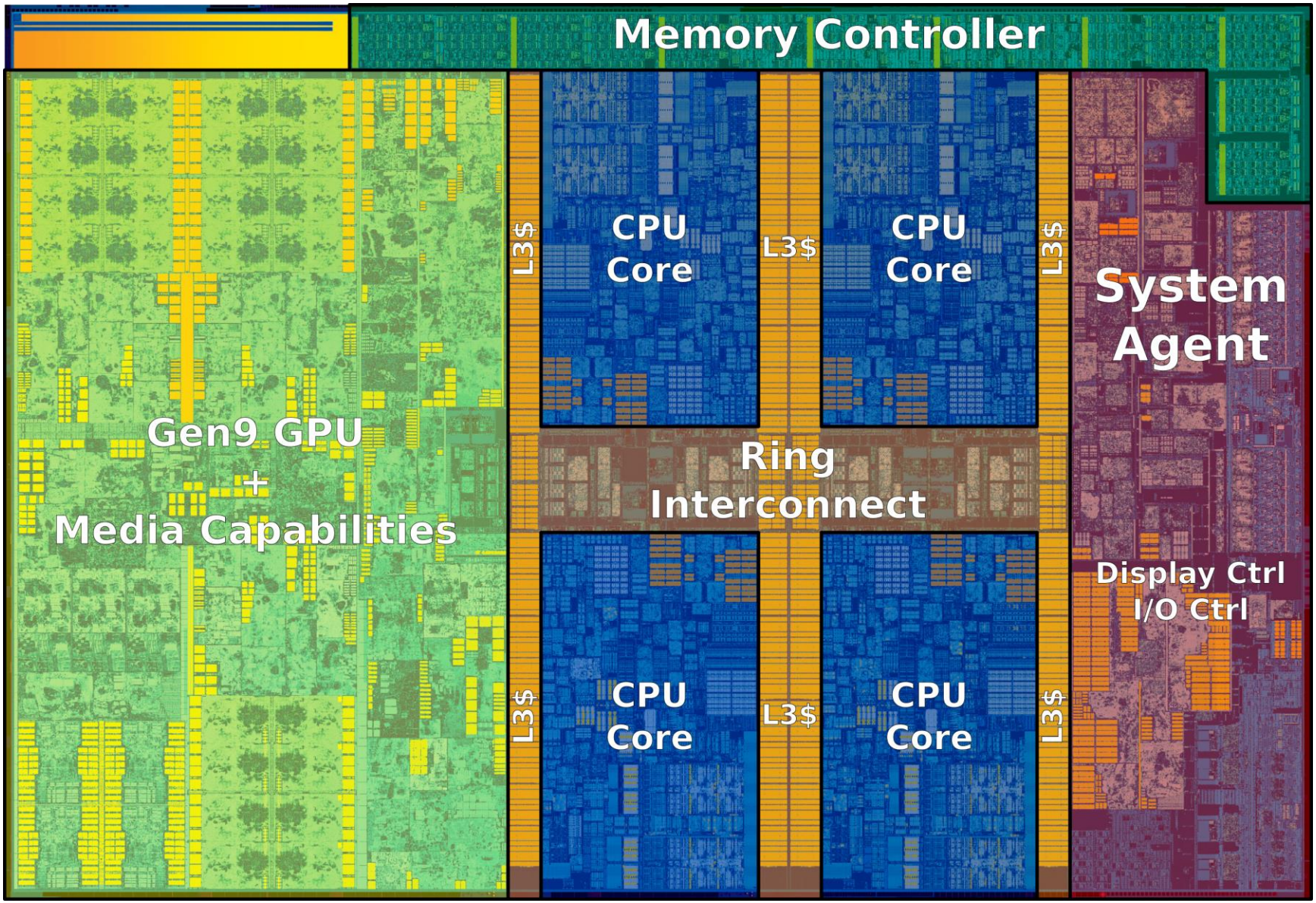
香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

Outline

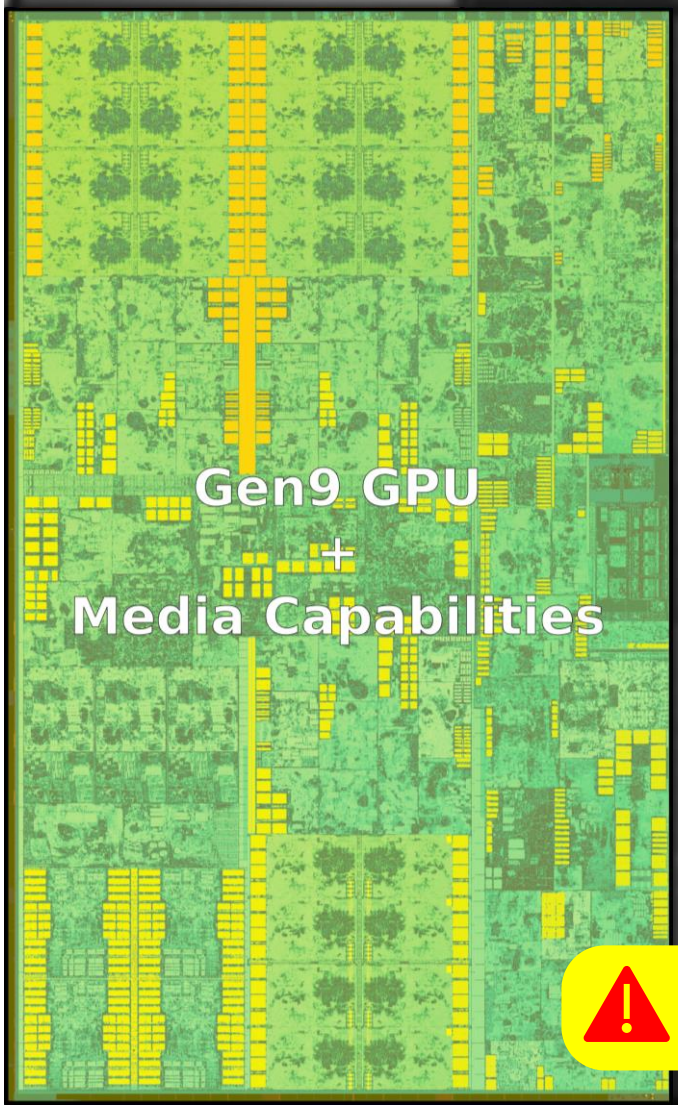
- Introduction
- iGPU Leak Vulnerability
- Proof-of-concept Demo
- Attack Case Studies
- Discussion and Conclusion

Intel Skylake (Client)



Source: wikichip.org;

Intel Integrated GPU (iGPU)



- Large area on die
- Ubiquity

Up to 71% PCs use iGPU. [1]

- Complex functions
  - GPGPU

⚠ Insufficient Security Scrutiny

Source: wikichip.org. [1] Jon Peddie Research, “Global GPU shipments up in Q2’19 reports,” Aug 2019.

Vulnerability Analysis on Intel iGPU

Threat model:

- Unprivileged GPU client
- Software-based

Identify an uninitialized hardware vulnerability

1. GPU Shared Local Memory Leak

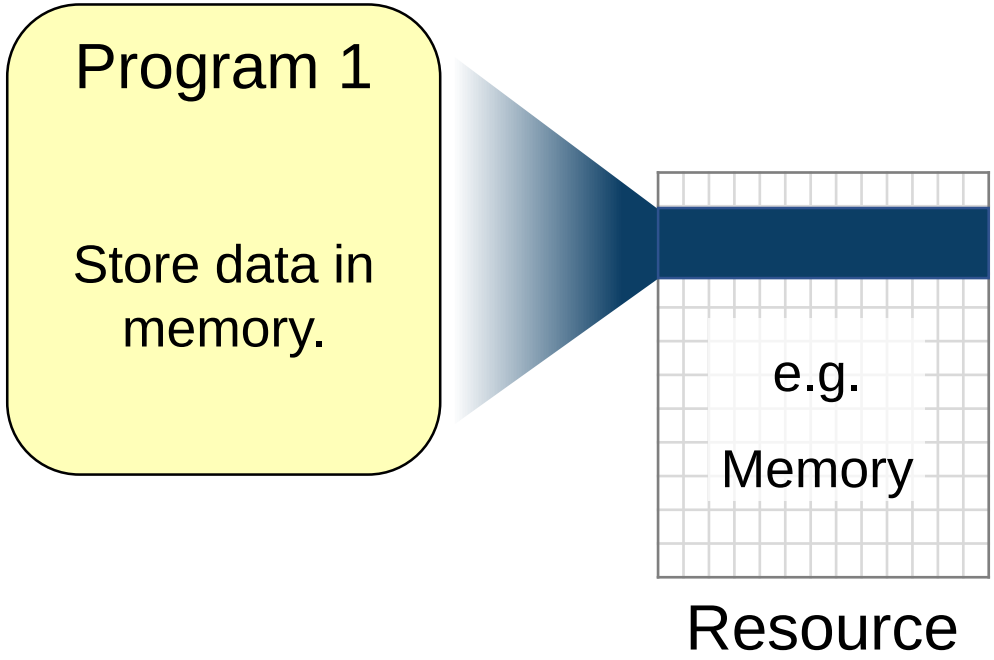
2. GPU Register Leak

Uninitialized Data

What is an uninitialized data bug?

Steps:

1. Program 1 uses some memory.

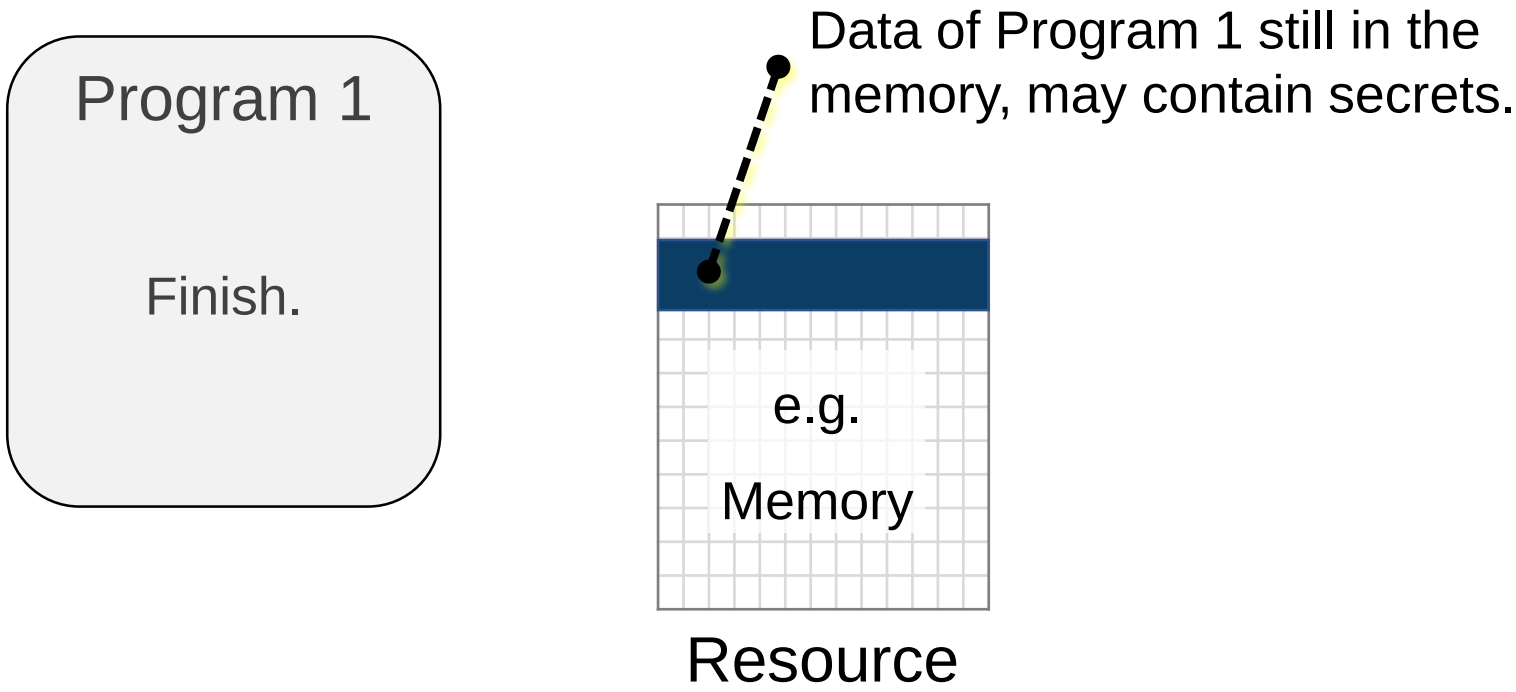


Uninitialized Data

What is an uninitialized data bug?

Steps:

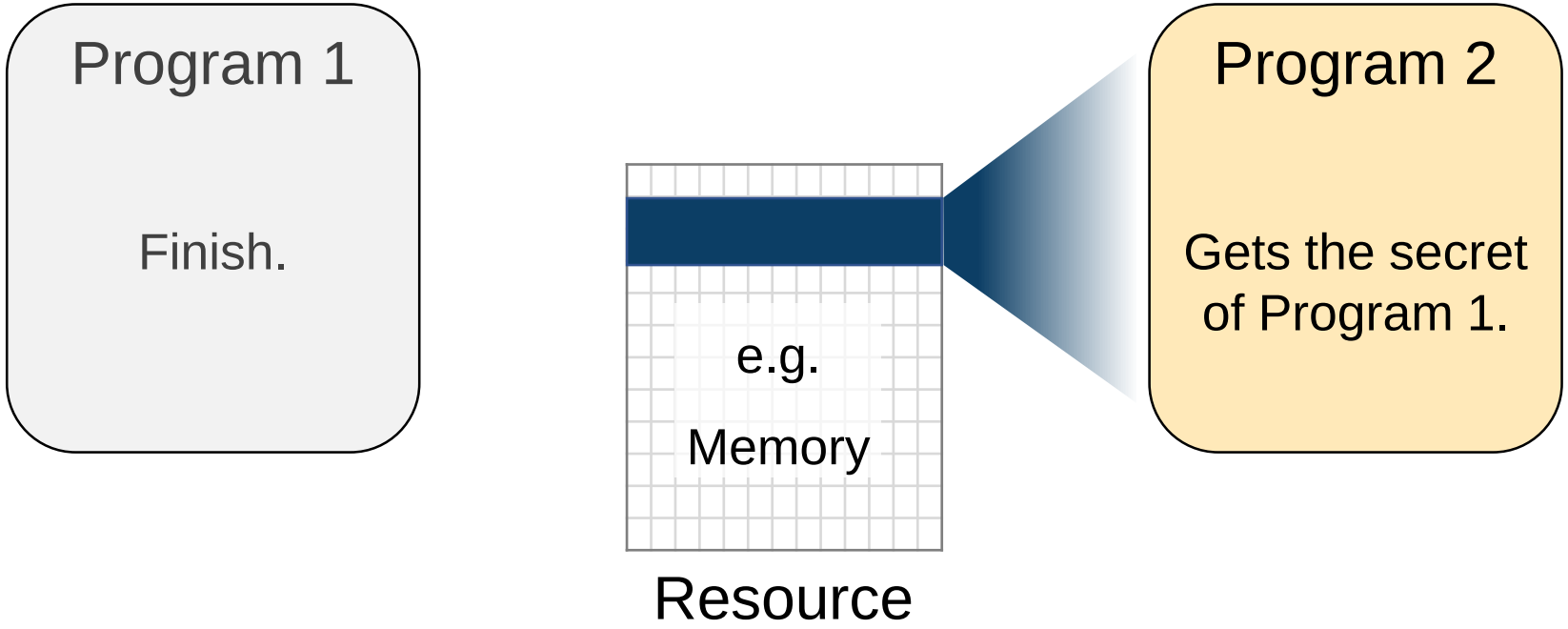
1. Program 1 uses some memory.
2. Program returns the memory space to OS.



Uninitialized Data

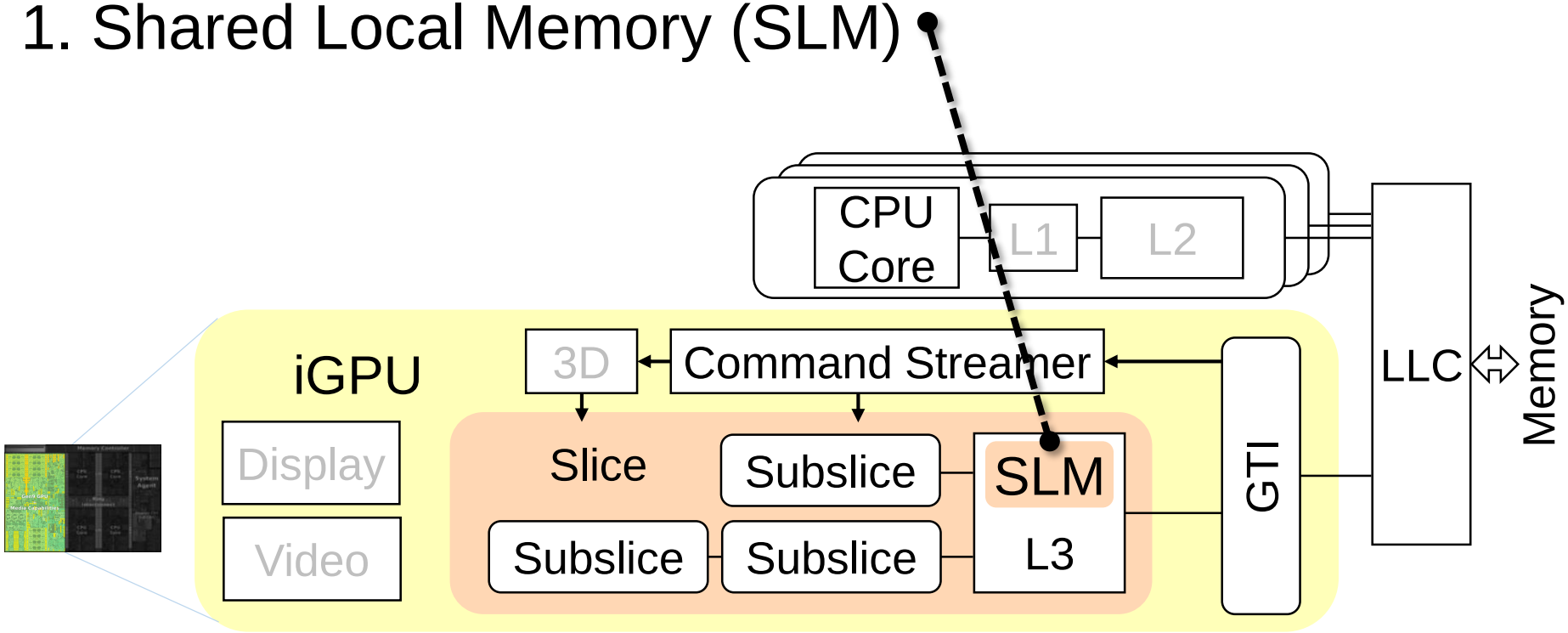
Steps:

1. Program 1 uses some memory.
2. Program returns the memory space to OS.
3. OS gives the memory region to Program 2,
   without clearing the memory.



Intel iGPU

1. Shared Local Memory (SLM)



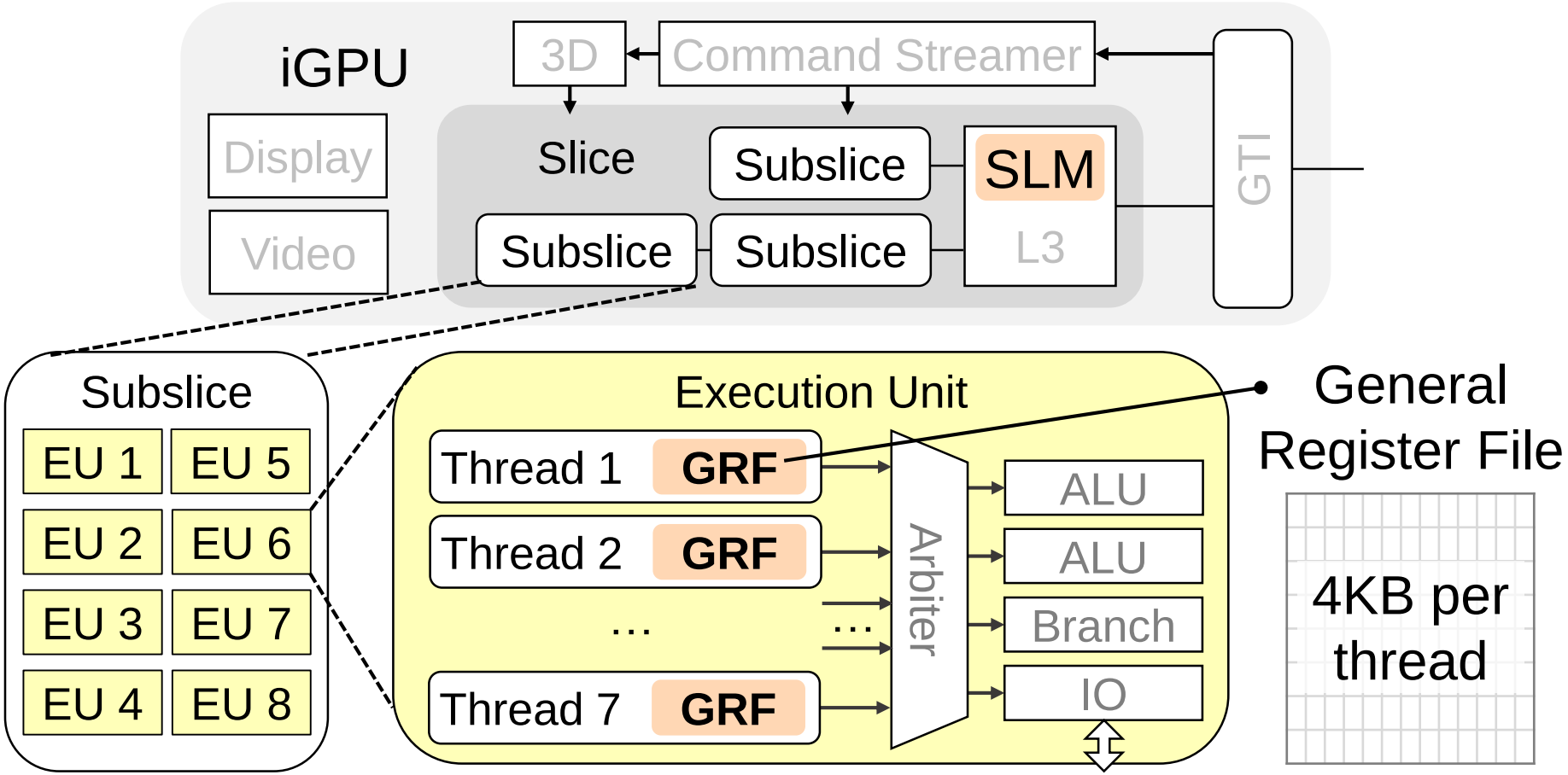
LLC: Last-level Cache

GTI: Graphic Technology Interface

[Ref] The Compute Architecture of Intel® Processor Graphics Gen9, 2015.

Intel iGPU µArch

2. GPU General-purpose Register File (GRF)

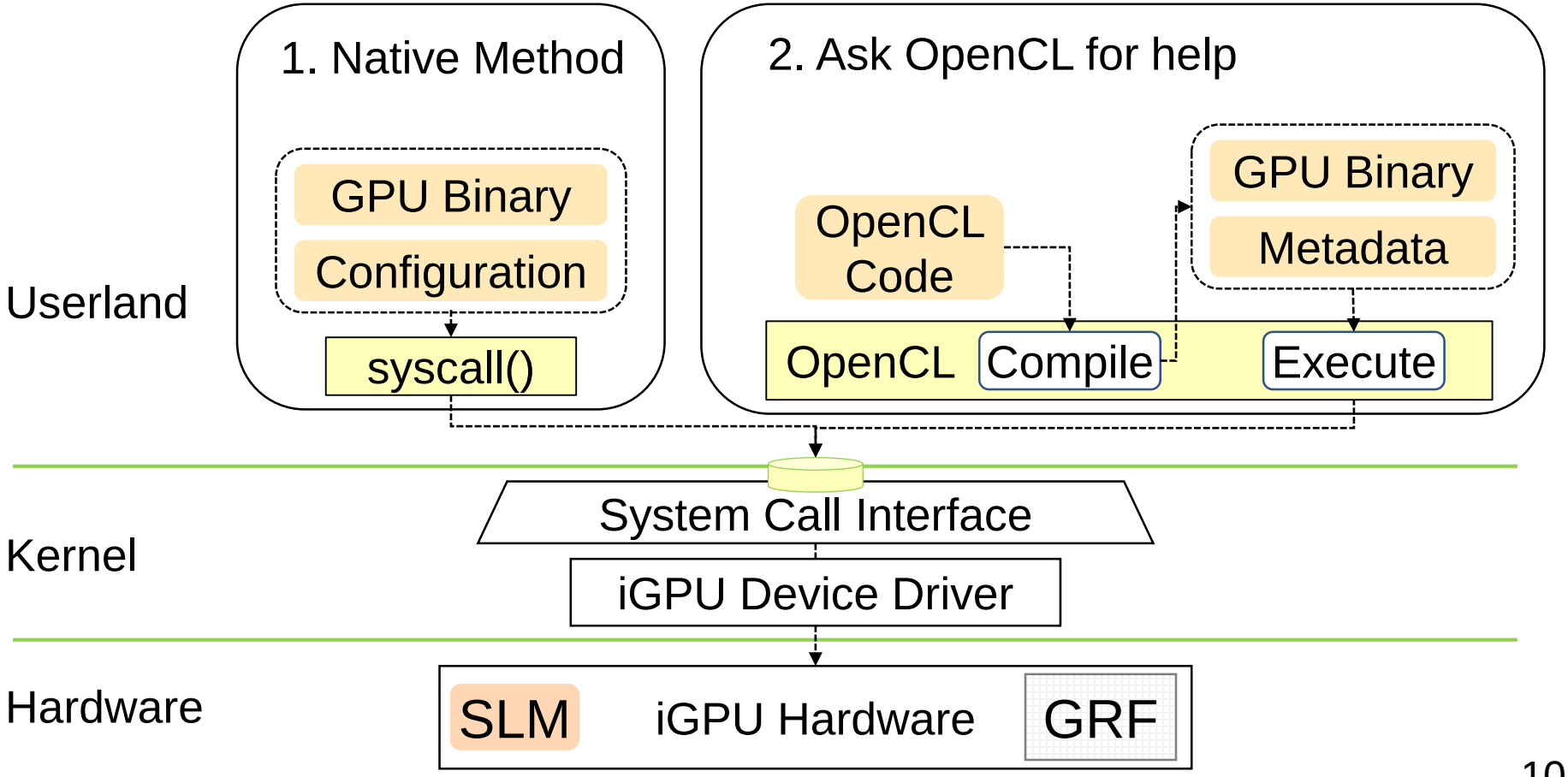


1 slice × 3 subslices × 8 EUs × 7 threads × 4 KB = 672 KB

GPU Programming

Goal: Userland GPU Spyware

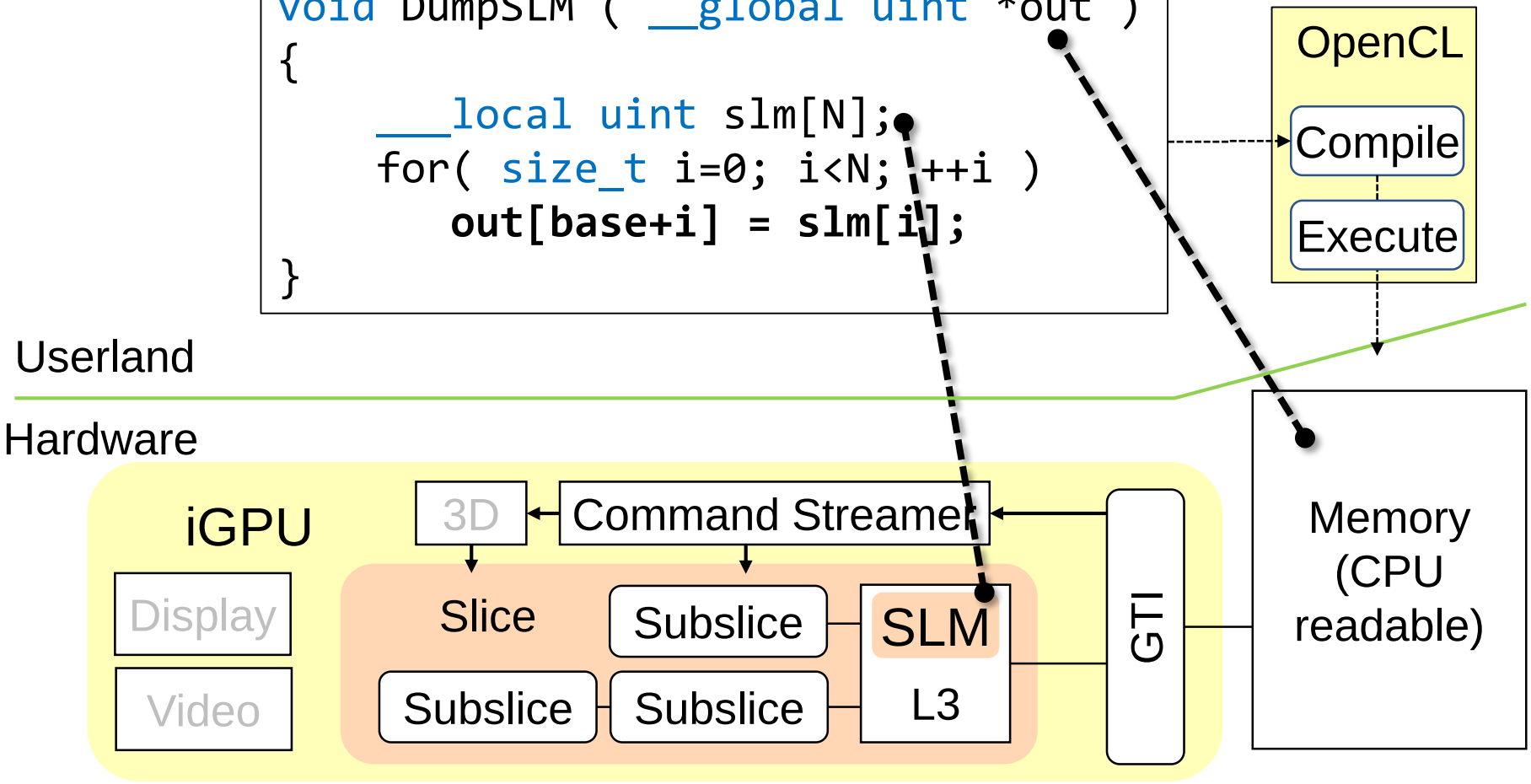
Challenge: GPU programming



SLM Leak

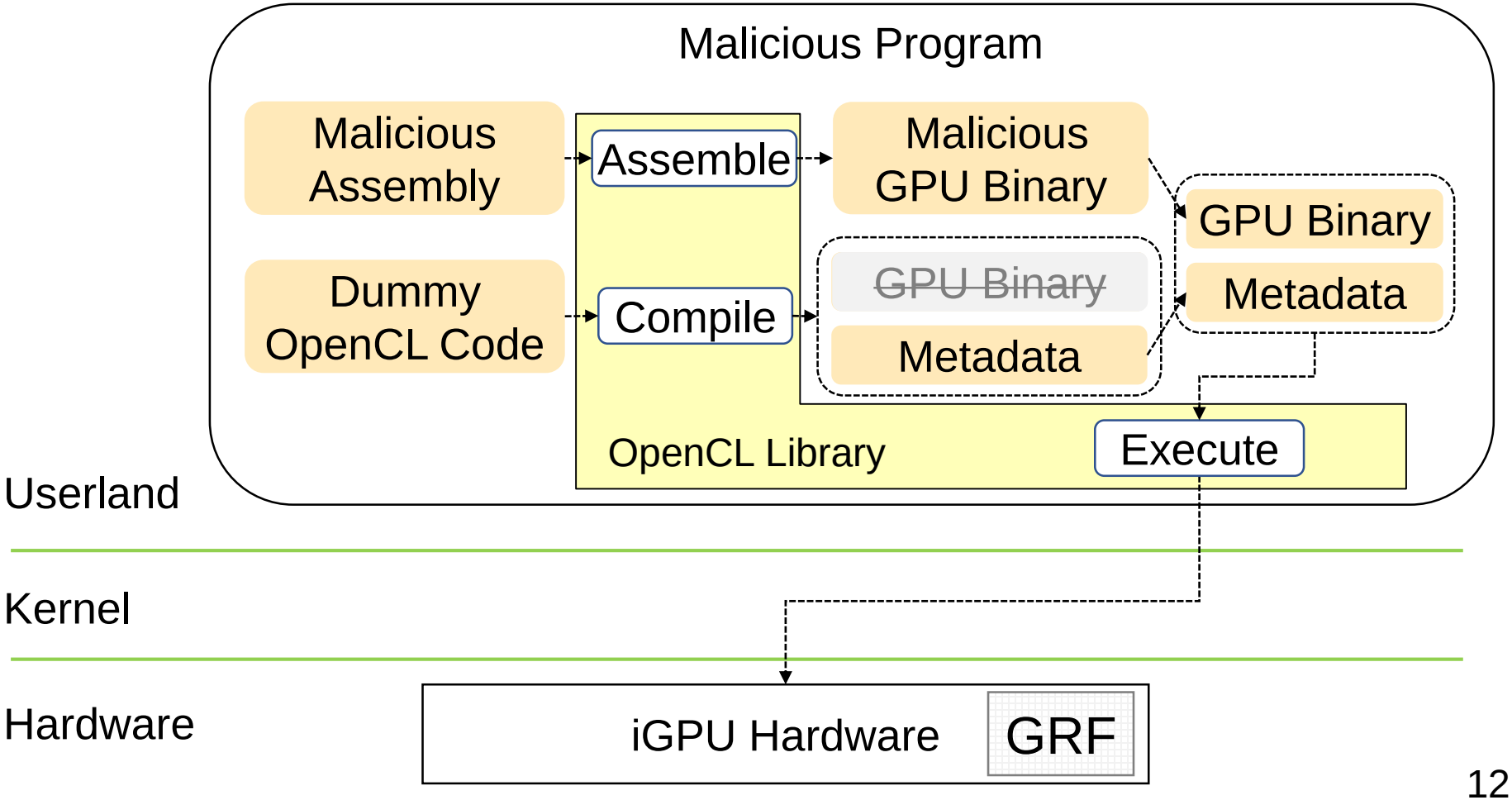
OpenCL code for Shared Local Memory (SLM) leakage

```
void DumpSLM ( __global uint *out )
{
    __local uint slm[N];
    for( size_t i=0; i<N; ++i )
        out[base+i] = slm[i];
}
```



Register Leak

Assembly programming for GPU register leakage:



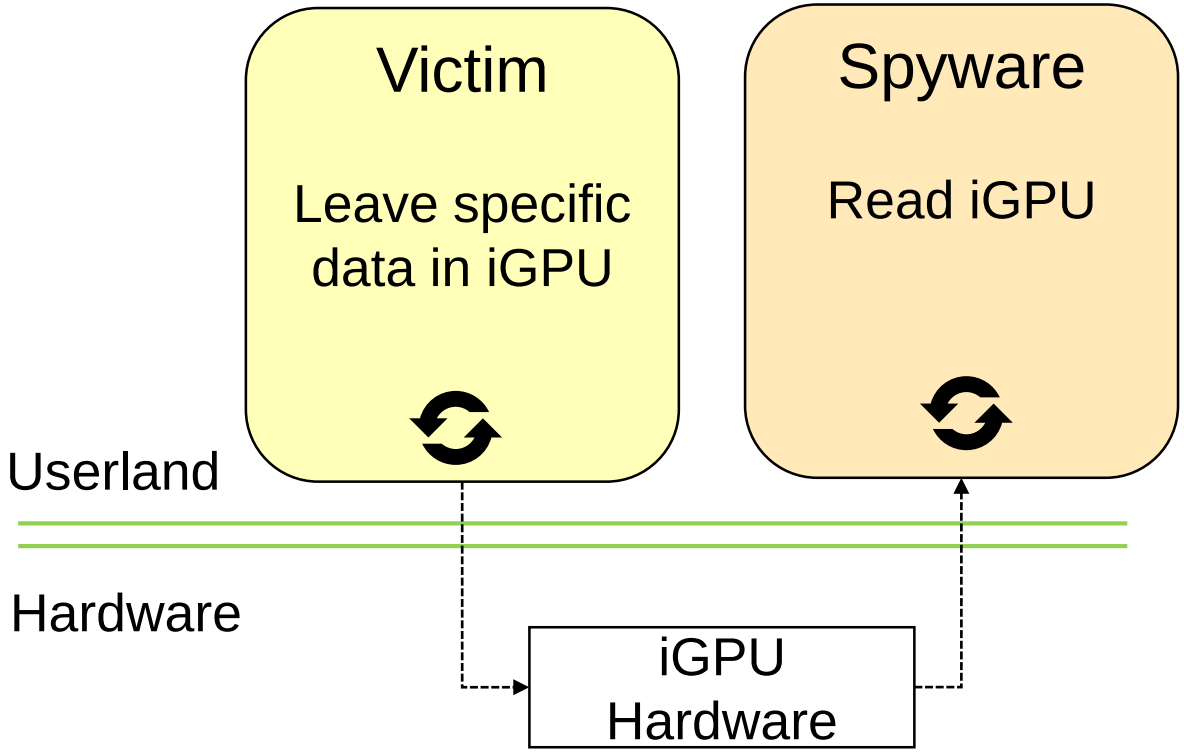
Introduction

iGPU Leak

PoC Demo

Attack Case Studies

Discussion and Conclusion



Model	Dell OptiPlex 7040
CPU	Intel Core i7 6700
iGPU	Intel HD 530 (Gen 9)
OS	Ubuntu 16.04 LTS / 18.04 LTS
OpenCL	Intel Graphics Compute Runtime 19.26

[Youtube] SLM Leak [Youtube] GRF Leak

Case Studies

1. Attacker does not know the source code

   Website Fingerprinting Attack

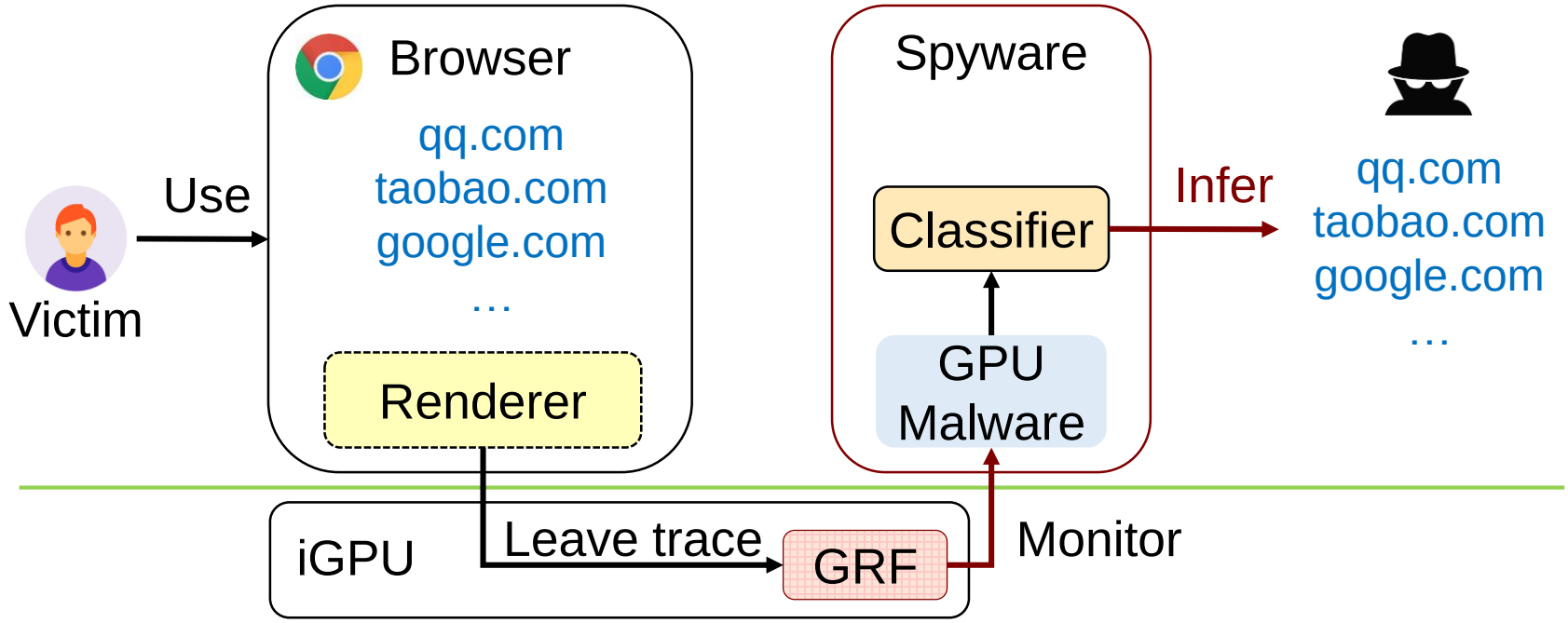
2. Attacker knows implementation details

   AES Key Recovery Attack

3. Leakage bandwidth measurement

   Covert Channel

1. Website Fingerprinting



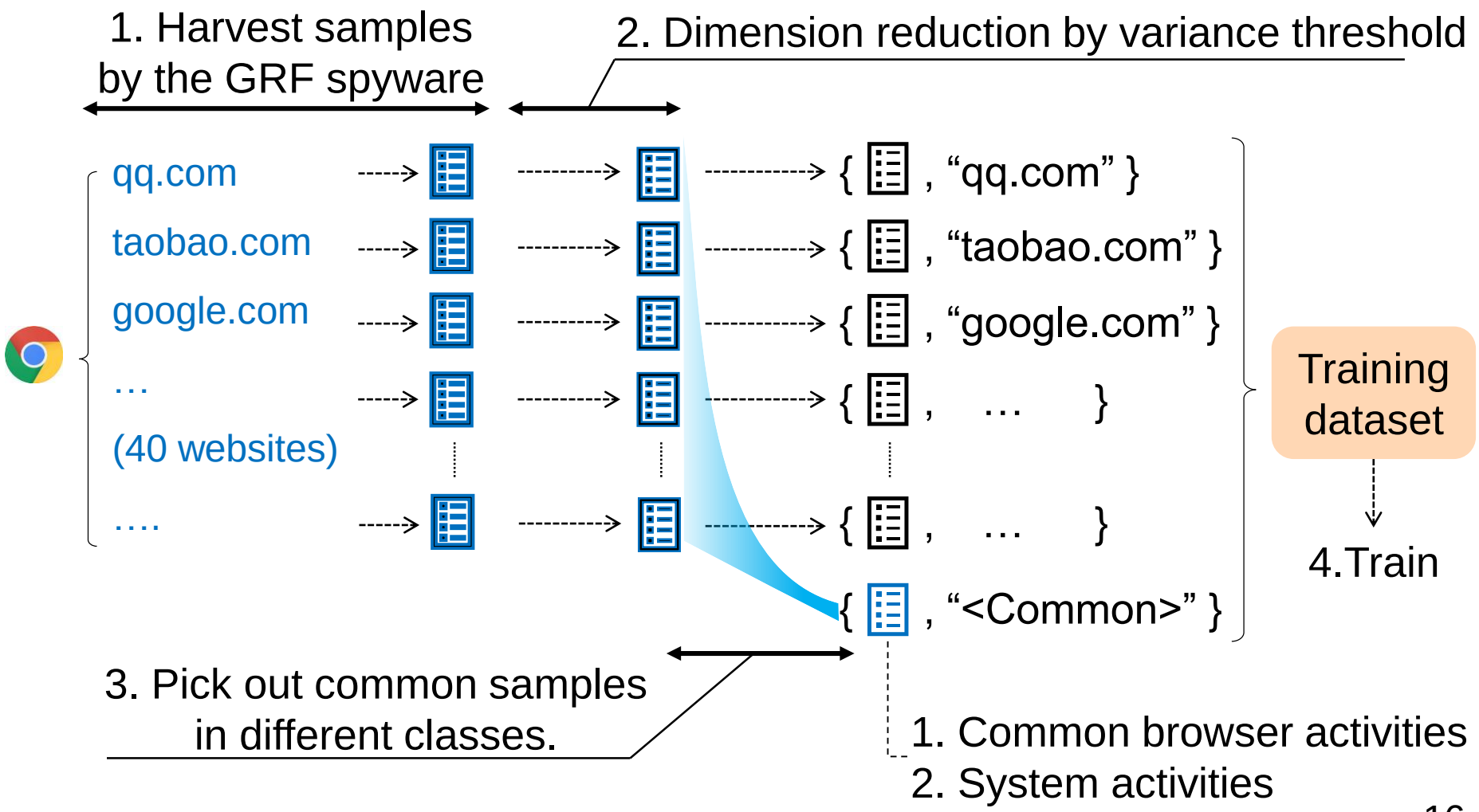
(GRF: General Register File)

Table: Experiment Configuration

Brower	Chrome 73.0.3683.103
Setting	Factory default
OS	Ubuntu 16.04 LTS
Websites	Alexa Top 40

1. Website Fingerprinting

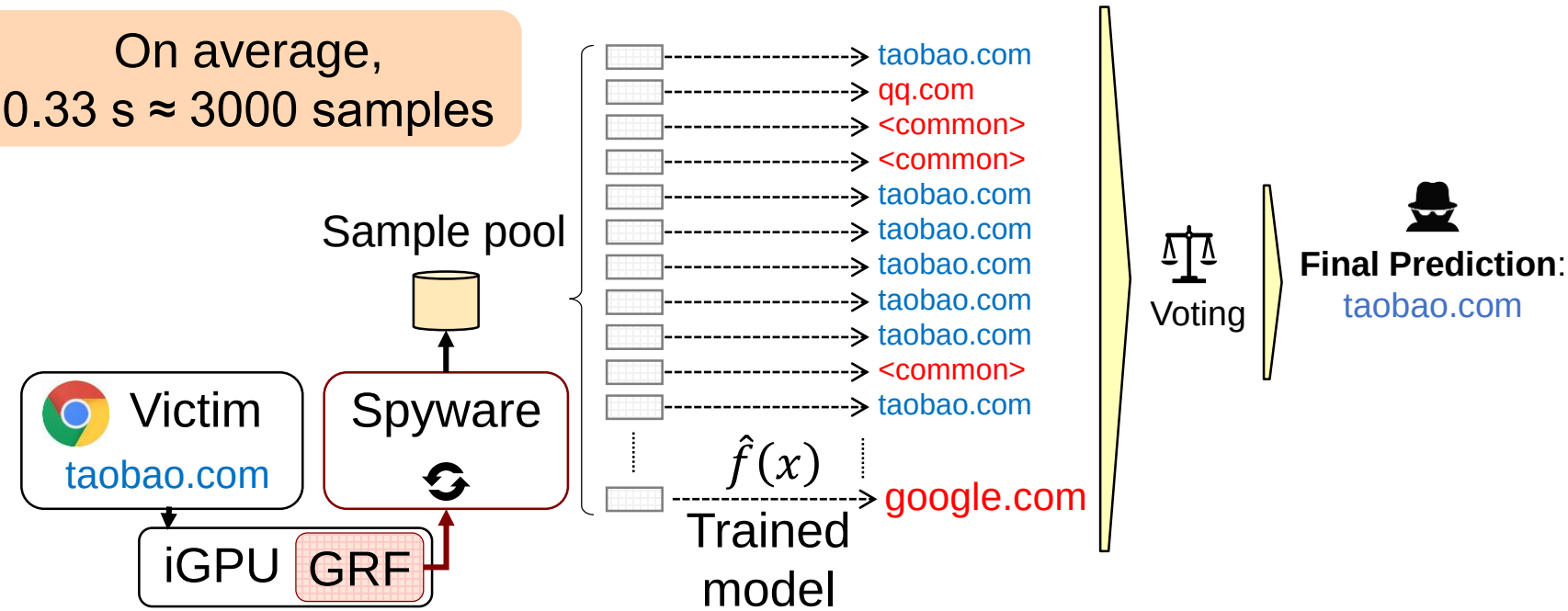
1.1 Training dataset



1. Website Fingerprinting

1.2 Inference

On average,
0.33 s ≈ 3000 samples

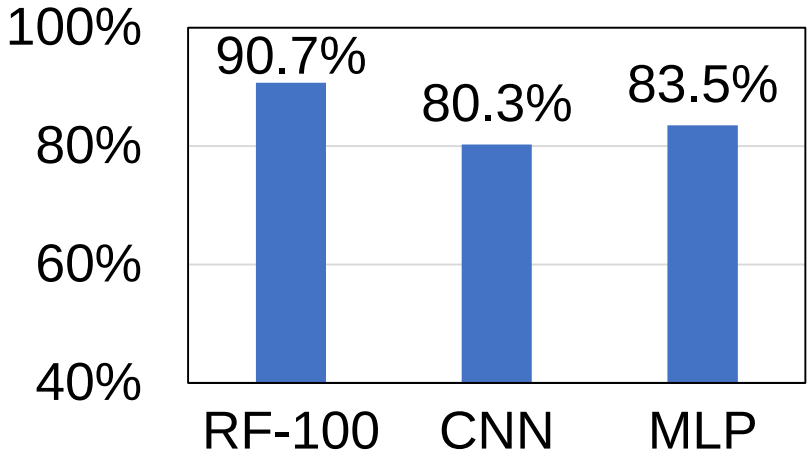


1. Monitor GPU and accumulate samples.

2. Individual predictions on each sample.

3. Voting for the final inference.

1. Website Fingerprinting

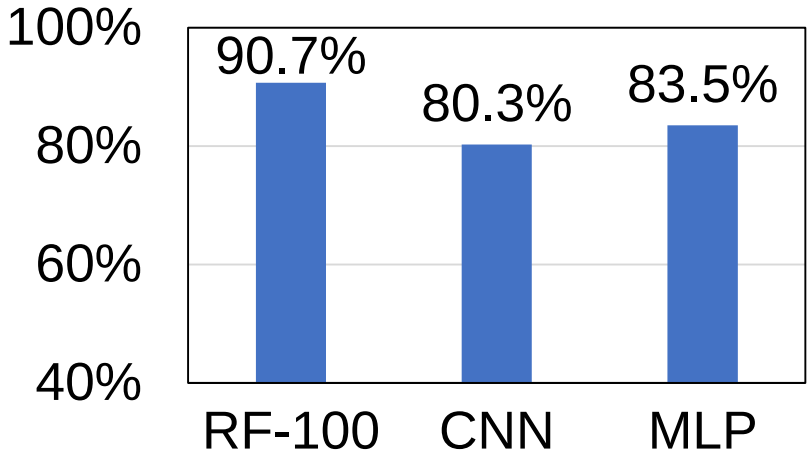


1. Accuracy of different models.

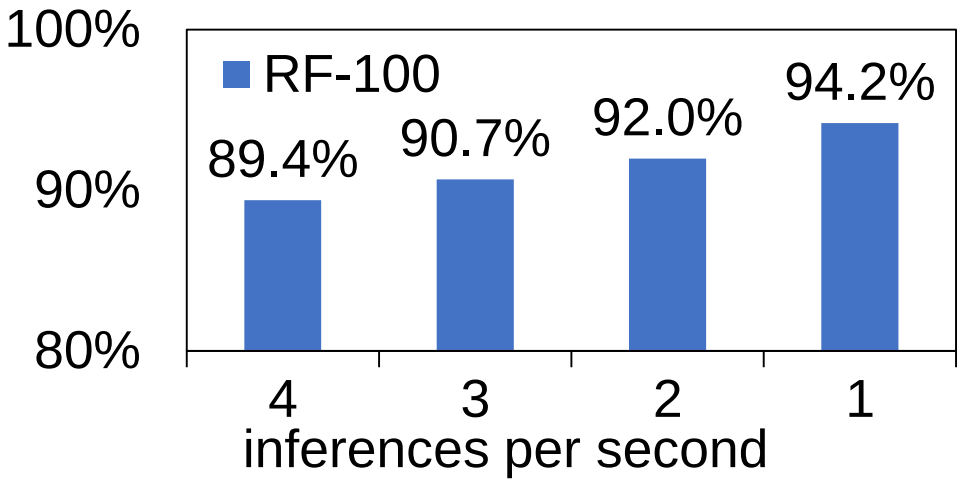
RF	Random forest
CNN	Conv. neural network
MLP	Multilayer perceptron

* 3 inferences per second

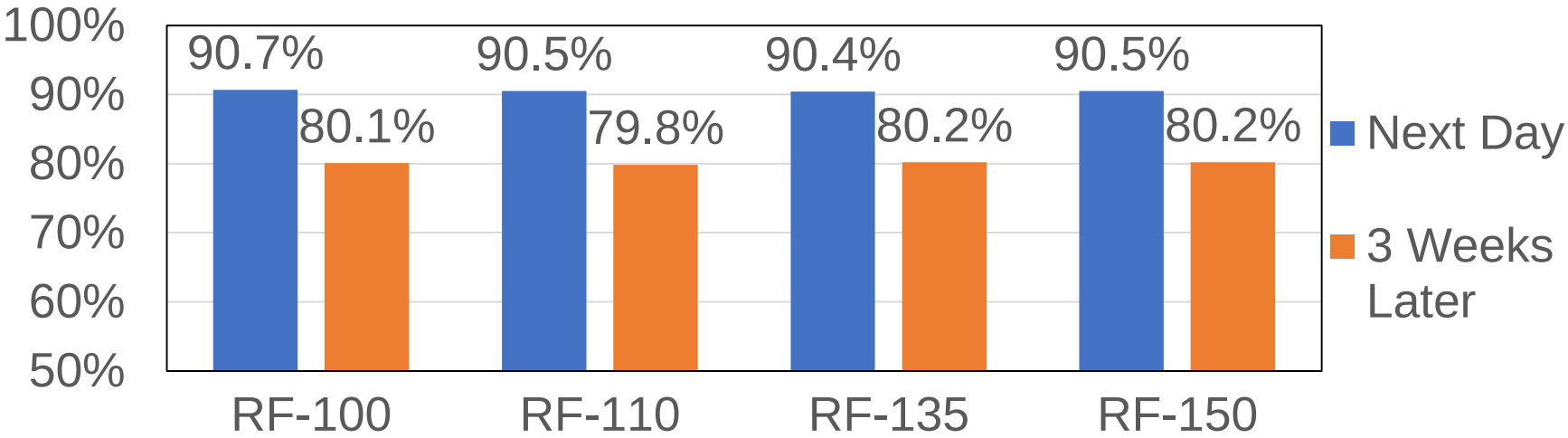
1. Website Fingerprinting



1. Accuracy of different models.



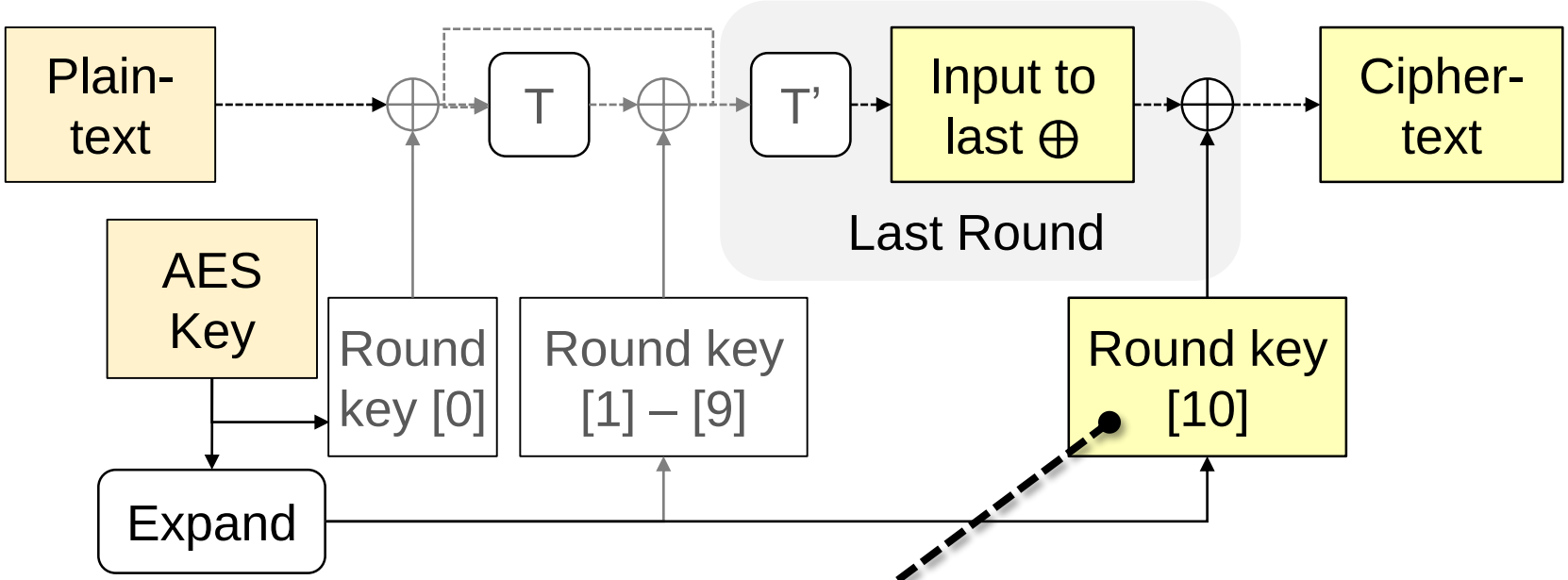
2. Accuracy vs. inference frequency.



3. Model duration against changes on the webpages over time

2. AES Attack

Assume: Attacker knows the GPU program of the victim.
Goal: AES key.

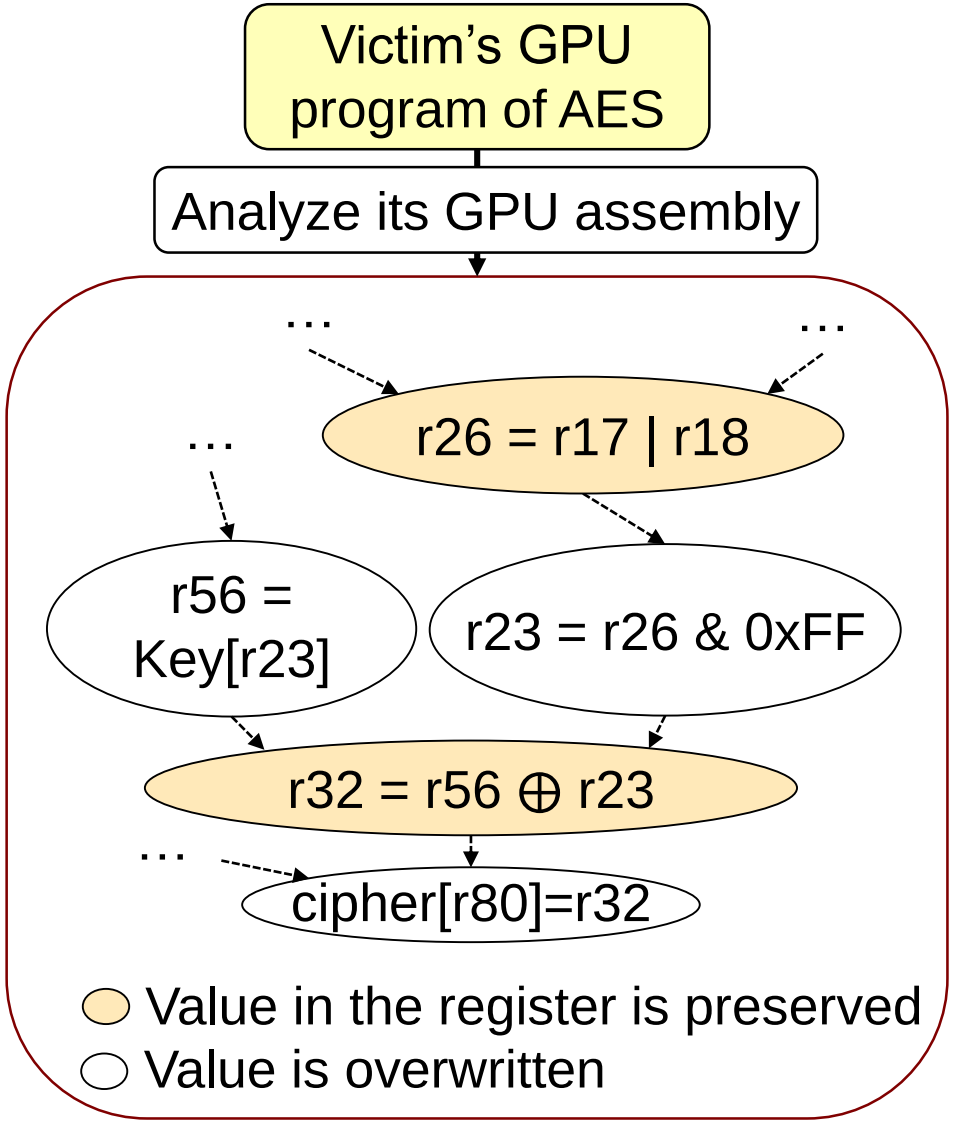


AES encryption flowchart of a GPU extension of `openssl` [1]

Property:
Knowing any Round key[i], one can reverse the AES key.

[1] G. Johannes and P. Margara, “engine-cuda / engine-opencl”.

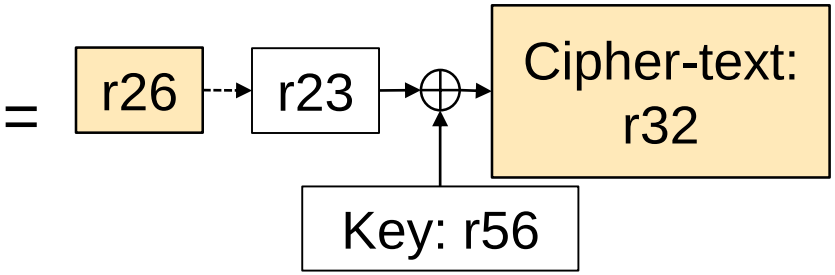
2. AES Attack



GPU register flow graph

AES Attack

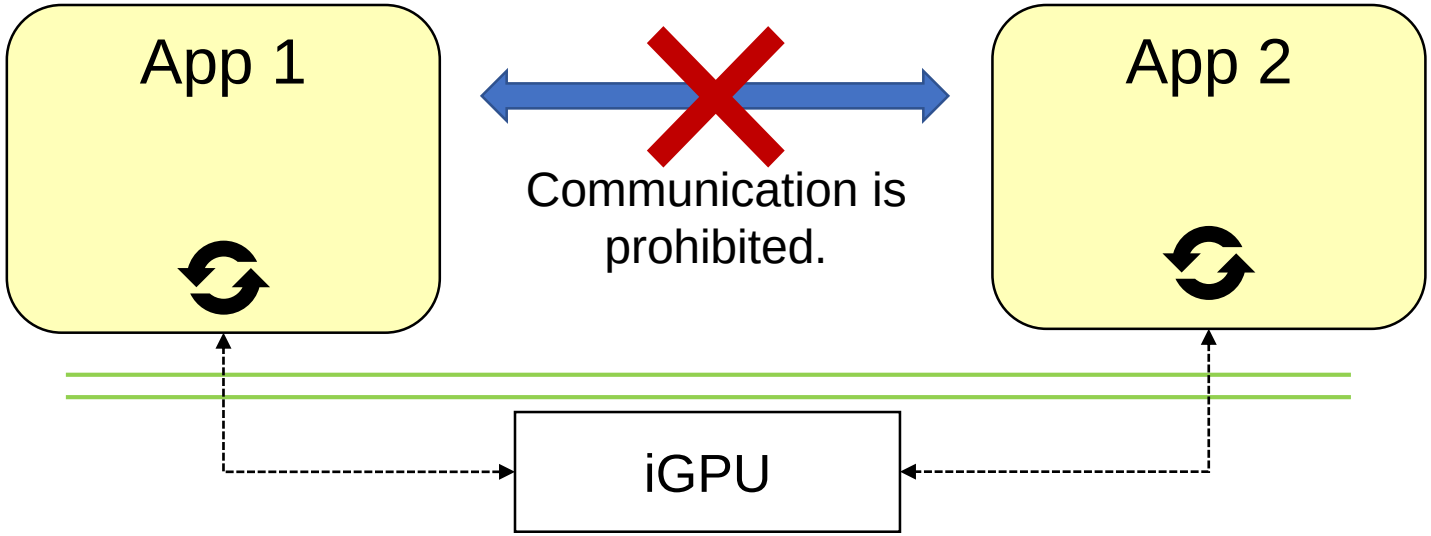
Victim	Result
AES-128	Leaked **13** / 16 Bytes
	0.15 s brute-force
AES-196	Leaked **20** / 24 Bytes
	2 min brute-force



Leaked key byte = r26 ⊕ r32

3. Covert Channel

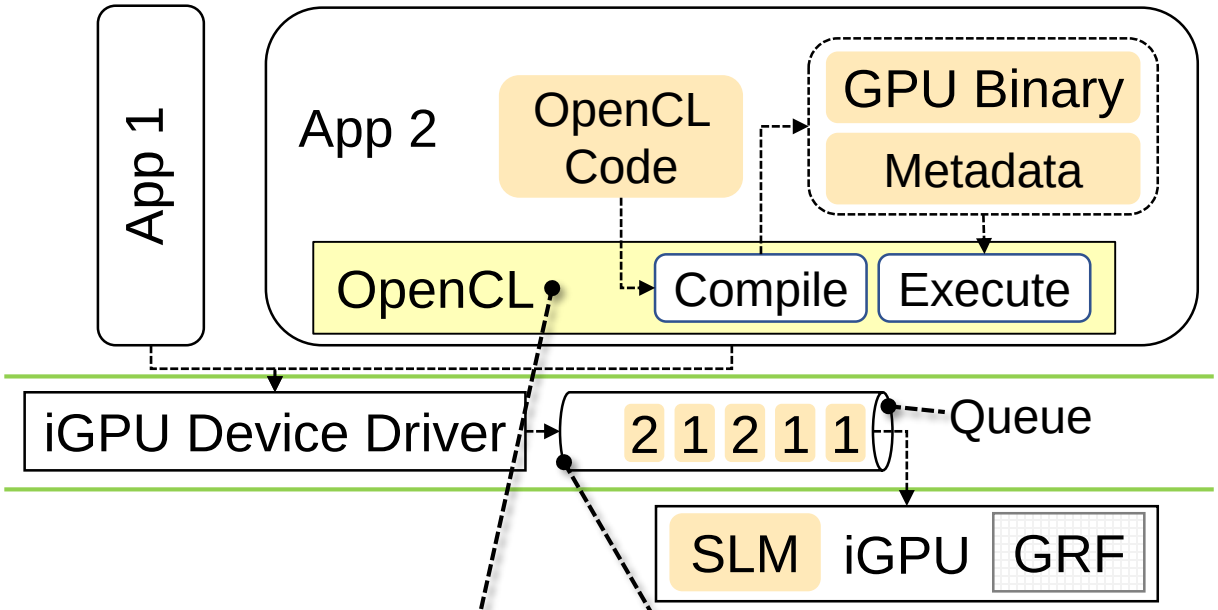
Bandwidth measurement



Register	Simplex	4 Gbps
	Duplex	8 Gbps
SLM	Simplex	1.3 Gbps
	Duplex	2.5 Gbps

Discussion & Conclusion

Discussion: Mitigation



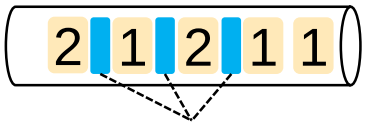
Userland patch

GPU Binary

Code added by the compiler automatically

↑ performance (maybe)
↓ legacy programs

Kernel patch



Flushing GPU between jobs

↑ remove root cause
↓ performance

Conclusion

- iGPU Leak: a dangerous vulnerability
- Privacy / Confidentiality / Covert channel
- Insufficient consideration of new peripherals
- Exposure: CVE-2019-14615
- Affected products & Patch status:

Affected Products	Affected OS	Patch
Many Intel processor families	Win	Intel Graphics DCH Driver 26.20.100.7209
	Linux	To be released in 2020 Jan.

Please refer to INTEL-SA-00314 for details.

THANK YOU

Q & A