

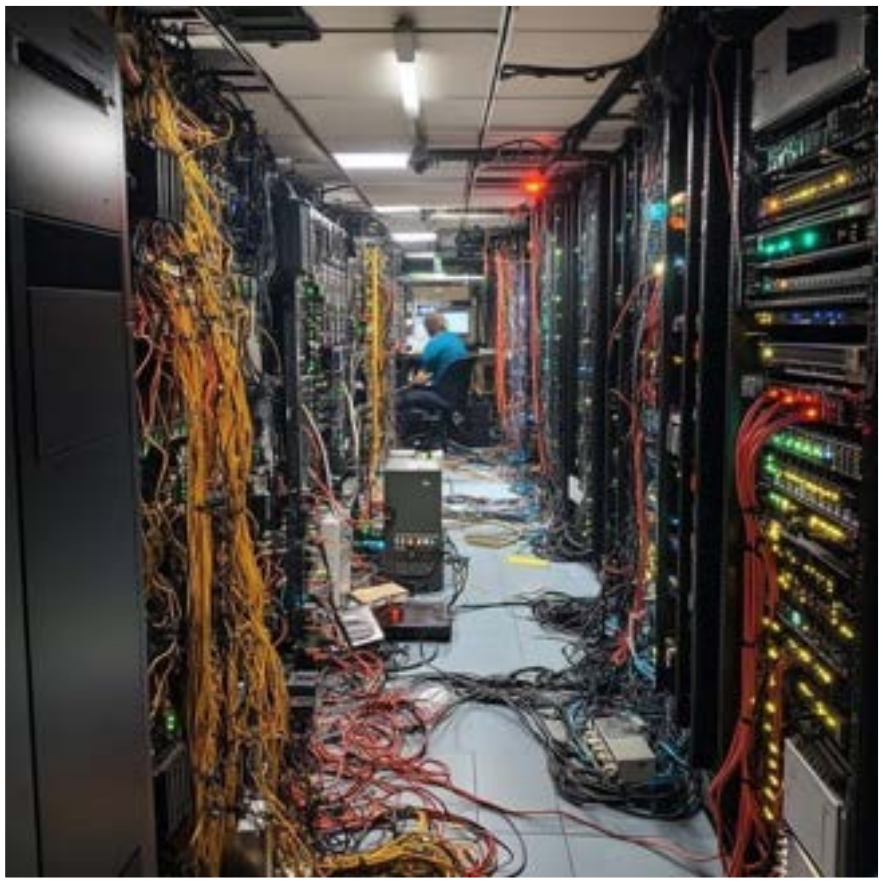
THE UNIVERSITY *of* EDINBURGH

Hardware and Platform Security

Security Engineering (Spring 2026)

Lecturer: Jingjie Li (jingjie.li@ed.ac.uk, www.jingjieli.me)

*Credits to Ross Anderson and Sam Ainsworth, who founded SENG in Edinburgh



- Do you trust your code running on a messy server room open to the public?
- Physical access exposes surfaces for security and privacy breaches that can be more challenging than digital ones
- (1) Using a physical mindset to evaluate digital attacks
- (2) Physical and hardware security as the root of trust for digital and cyber-physical systems



THE UNIVERSITY *of* EDINBURGH

01

Physical security

Locks, alarms, attacker capabilities and tamper resistance

Physical Security Philosophy

- Locks and walls will be part of your infrastructure at some level
- While the techniques are simpler than digital security, the weaknesses are often as subtle.



Five stages of physical security

Deter → **Detect** → **Alarm** → **Delay** → **Respond**

Time matters: delay the attacker long enough for the response to arrive.



THE UNIVERSITY *of* EDINBURGH

Attacker Capabilities in Threat and Risk Assessment

Derek

Unskilled

19-year-old addict; an opportunistic criminal looking for simple, low-risk opportunities.

Charlie

Skilled

40-year-old with 7 convictions; not intelligent, but cunning and experienced, so knows the tools of the trade.

Bruno

Highly skilled with help

"Gentleman criminal" who steals art and takes pride in his work; adept at lock and alarm hacking, and moving into computer hacking.

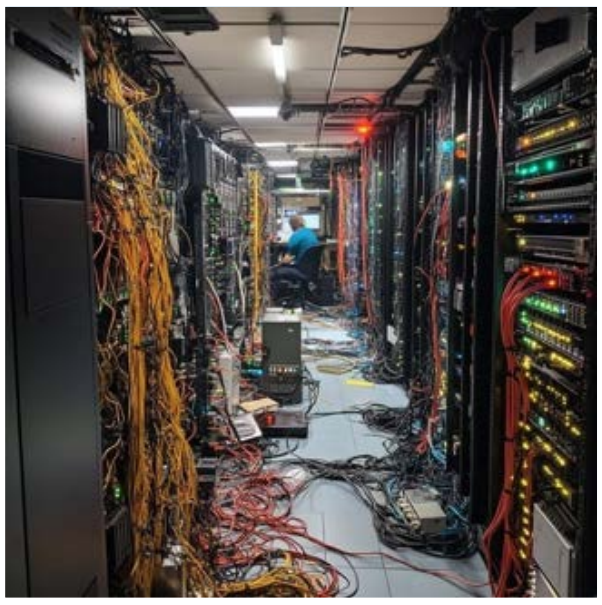
Abdurrahman

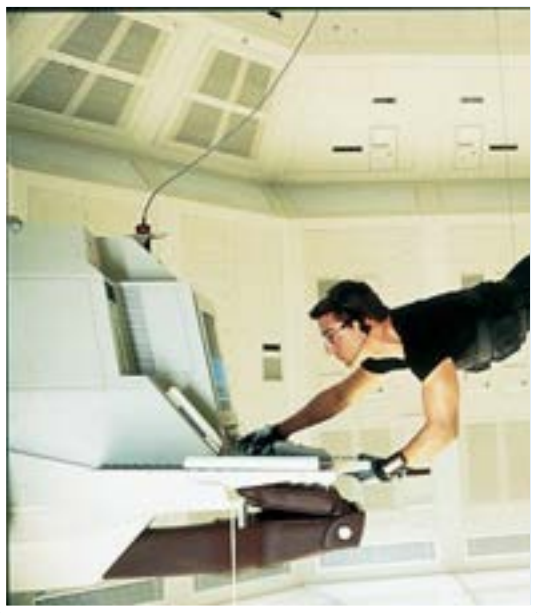
Highly skilled with resources

Head of a dozen agents, with access to specialist weapons and PhD-grade technical support.

Who will be the most likely attacker?







How the security level aligns with your asset?



- E.g. wireless smart cards and card readers using challenge-response protocols
- MIFARE Classic: vulnerable but still widely deployed!
- All the usual crypto issues apply: weak ciphers, bad random number generators, short keys…

- Security (cryptography) protocols are hard to update
- Easy to defeat (e.g., due to budget, form factor, export control, etc.)

How the security level aligns with your asset?



- E.g. wireless smart cards and card readers using challenge-response protocols
- MIFARE Classic: vulnerable but still widely deployed!
- All the usual crypto issues apply: weak ciphers, bad random number generators, short keys…

Don’t design against Charlie when you need to keep out Bruno!

Alarm – it matters as locks and walls can be defeated



Theodosian Walls – fall of Constantinople (1453)



Maginot Line

Tamper resistance – how do we know a system has been attacked?






Tamper resistance – Hardware Security Modules

- Store confidential data and perform critical computation
- RAM set to 0 (destroying and refreshing encryption keys) when the physical case is open
- So maintenance staff can't get the key
- Early versions were vulnerable to cutting through, so cores were "sealed" in epoxy resin
- Is information still left somewhere?



Tamper resistance – Hardware Security Modules

- Key bits get burnt into SRAM, so the bit state remains even after it is refreshed!
- Similar phenomena in DRAM/flash





THE UNIVERSITY *of* EDINBURGH

02

Physical channels

When physical signals leak more than you intended



THE UNIVERSITY *of* EDINBURGH

Hardware channels opening up attack surfaces

01
Cache

02
Disk timing

03
Instruction timing

04
CPU utilisation

05
Clock frequency

06
Power consumption

07
Erroneous behaviour
e.g. differential fault analysis

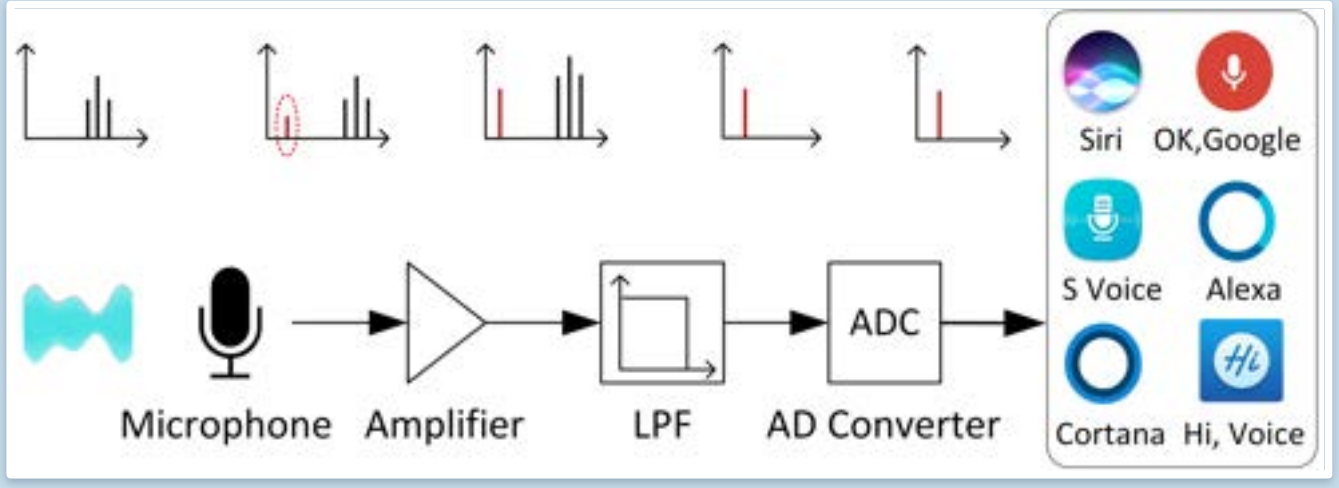
08
RF emissions
e.g. from monitors

Channel injection: leveraging physical "fault" to inject malicious input



Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T. and Xu, W., 2017, October. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 103-117).

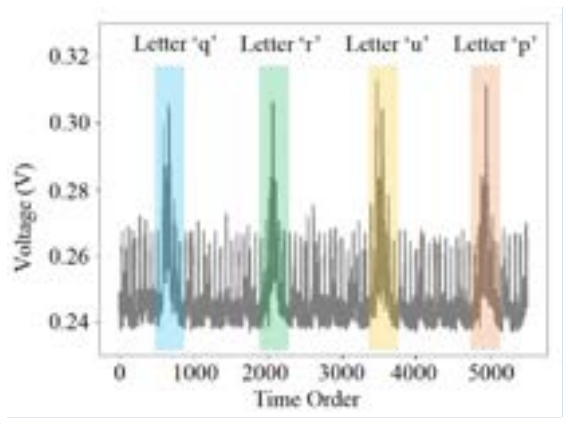
Side channels: leveraging physical "fault" to inject malicious input

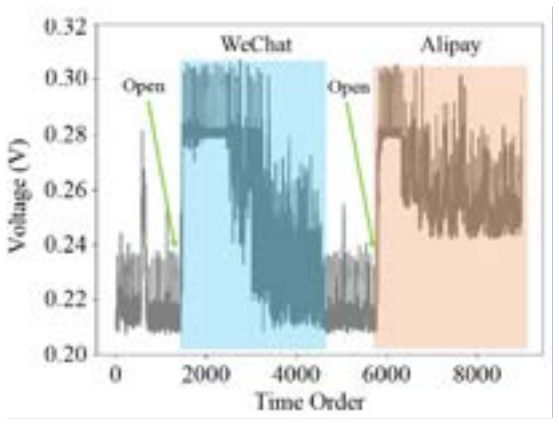


- How to inject malicious commands without humans noticing?

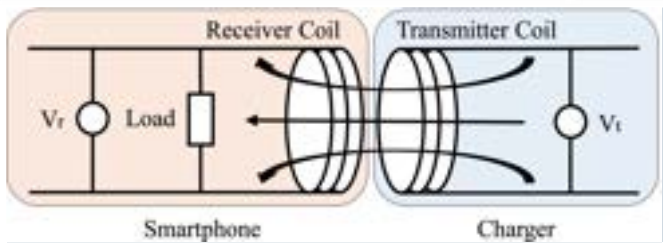
Side channels

- Information breach from power channels





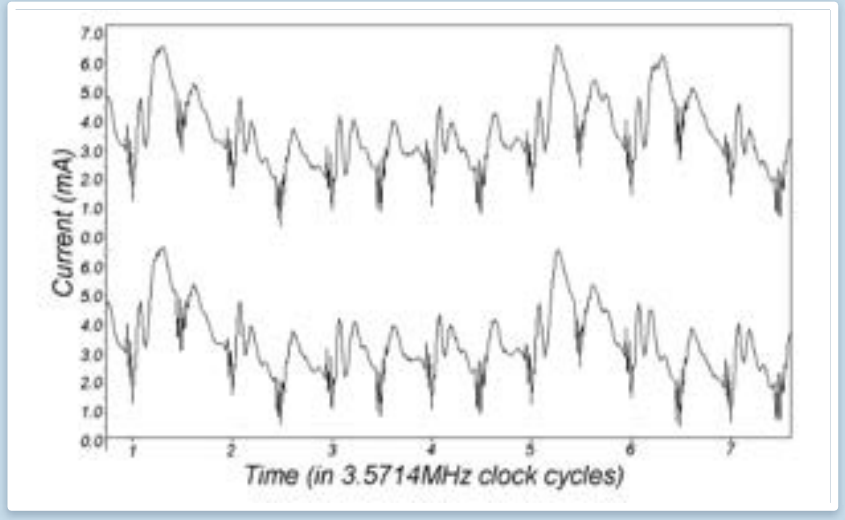




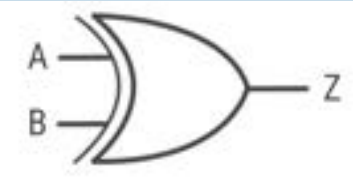
Liu, J., Zou, X., Zhao, L., Tao, Y., Hu, S., Han, J. and Ren, K., 2022. Privacy leakage in wireless charging. *IEEE Transactions on Dependable and Secure Computing*, *21*(2), pp.501-514.

Side channels: physical channels carry more information than you want

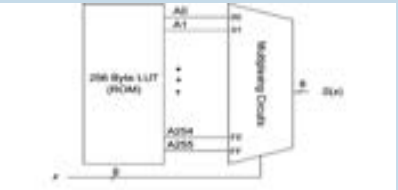
- Revealing encrypted data through power analysis



Differential Power Analysis, Kocher, Jaffe and Jun, CRYPTO '99



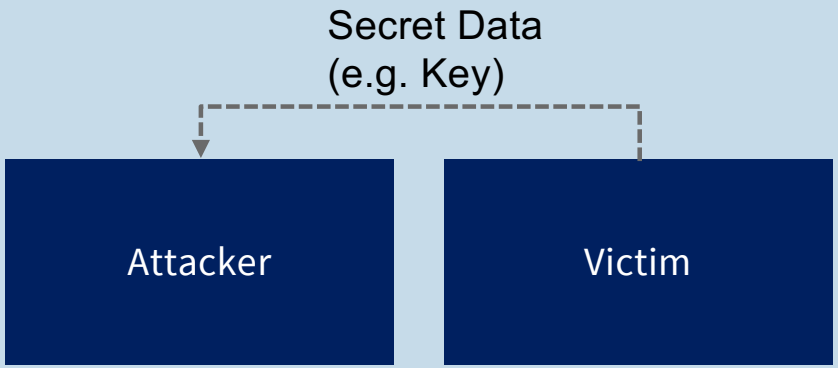
XOR



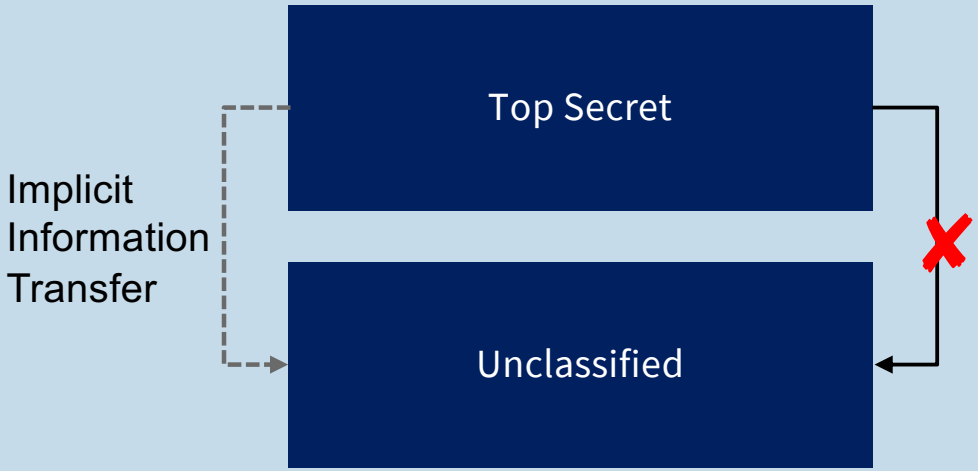
S-box lookup table

- Power consumption differs across bit operations, correlated with bit flips in CMOS
- The power side channel gives the adversary a way to measure and verify guesses

Channels



Side channel



Covert channel (when the leak of information is deliberate)



THE UNIVERSITY *of* EDINBURGH

03

Rowhammer

Flipping bits in hardware memory

Hardware architecture side channel – Rowhammer



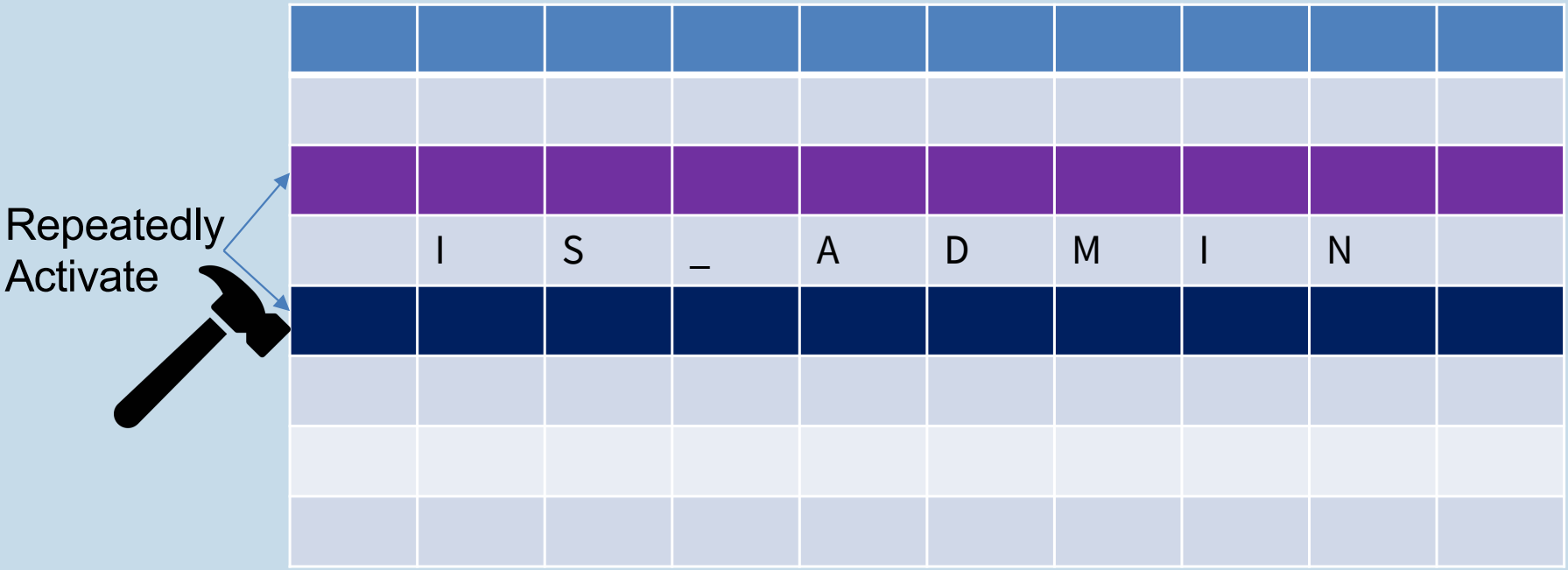
- DRAM (dynamic random access memory)
- Main memory in computer
- Stores data temporarily while powered on (volatile)

Hardware architecture side channel – Rowhammer

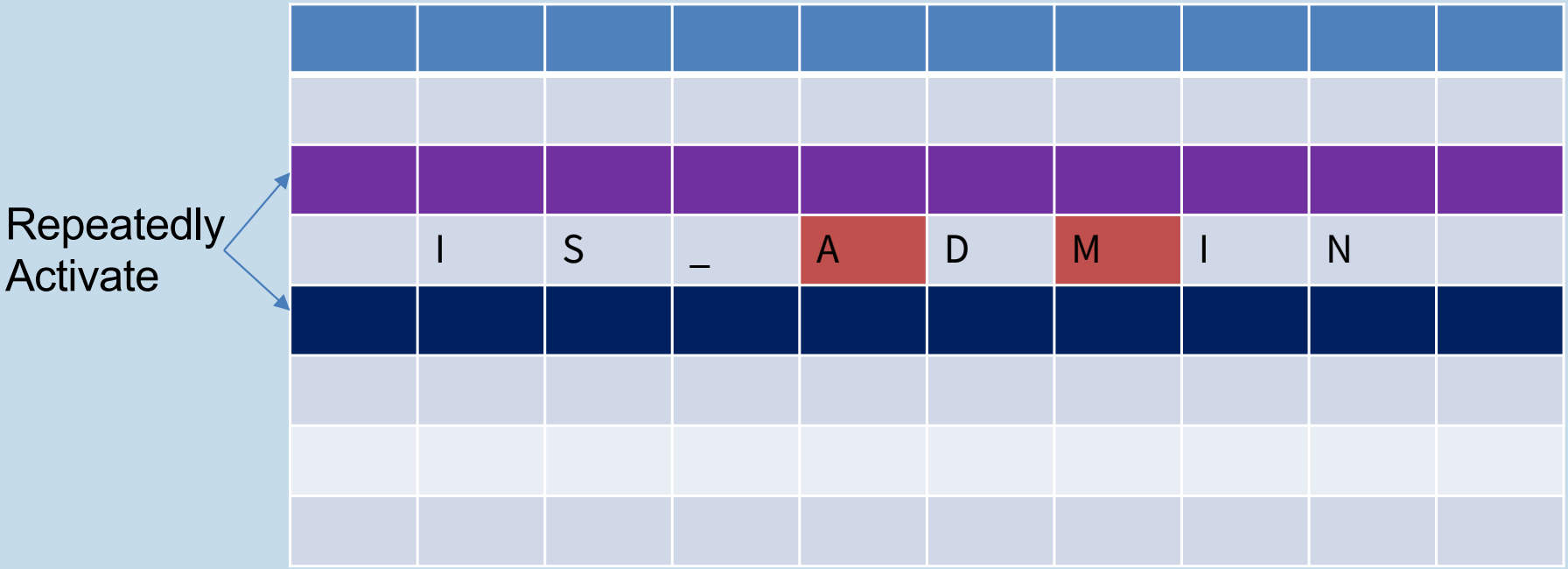
	I	S	_	A	D	M	I	N	

Data (e.g., a password) sits in banks and rows; one row is fetched at a time

Hardware architecture side channel – Rowhammer

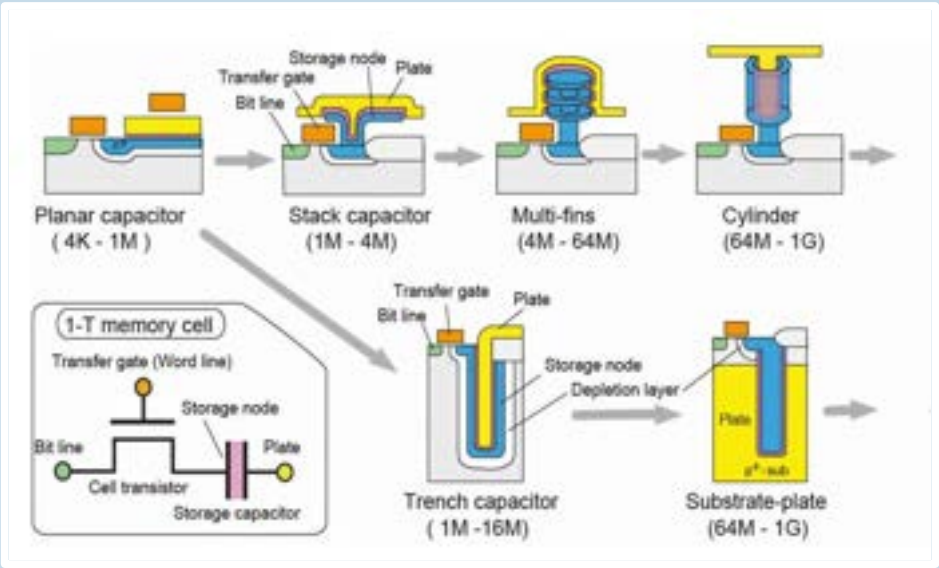


Hardware architecture side channel – Rowhammer



- Rows sit too close – hammering one leaks charge from victim rows next to it

Hardware architecture side channel – Rowhammer



- Bits stored in capacitors, e.g., 1 = charged, 0 = discharged
- It leaks over time!
- Charge needs to be refreshed to keep data



THE UNIVERSITY *of* EDINBURGH

Hardware architecture side channel – Rowhammer

1

Corrupt files

Flip bits in data or code that others rely on

2

Change privileges and secret data

e.g. page-table entries, keys, credentials

?

What are the other risks?

Discuss



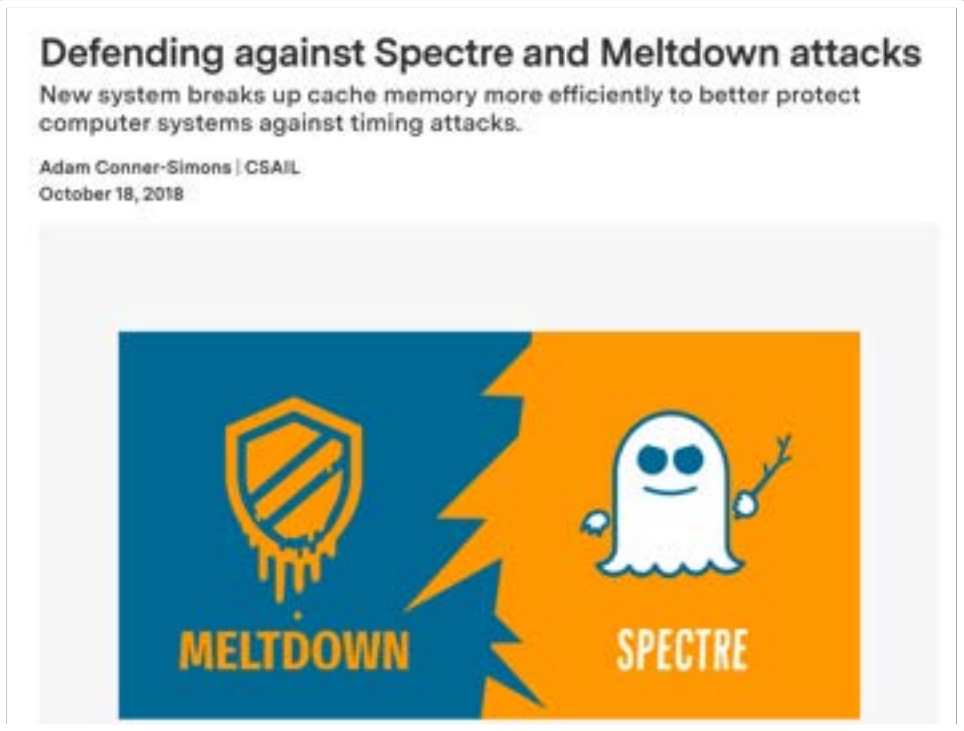
THE UNIVERSITY *of* EDINBURGH

04

Meltdown and Spectre

Speculative execution as an attack surface

Speculative Side Channel Attacks in Computer Architecture

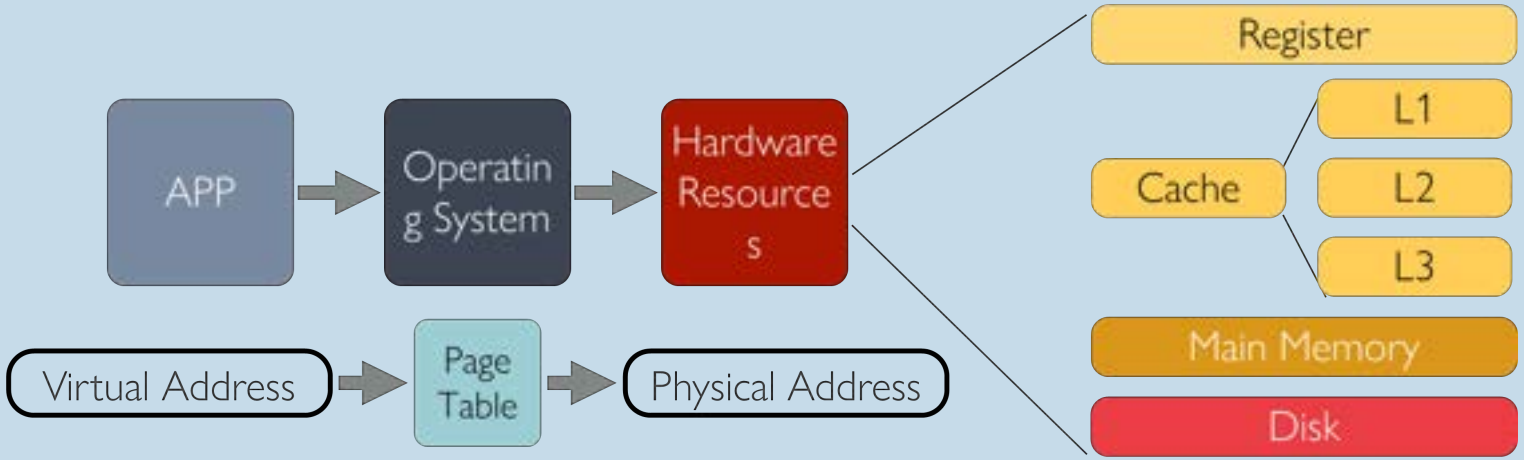


- Expensive hardware resources
- Performance matters!
- Increasing attention on security

Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., Kocher, P., Genkin, D., Yarom, Y. and Hamburg, M., 2018. Meltdown. *arXiv preprint arXiv:1801.01207*.

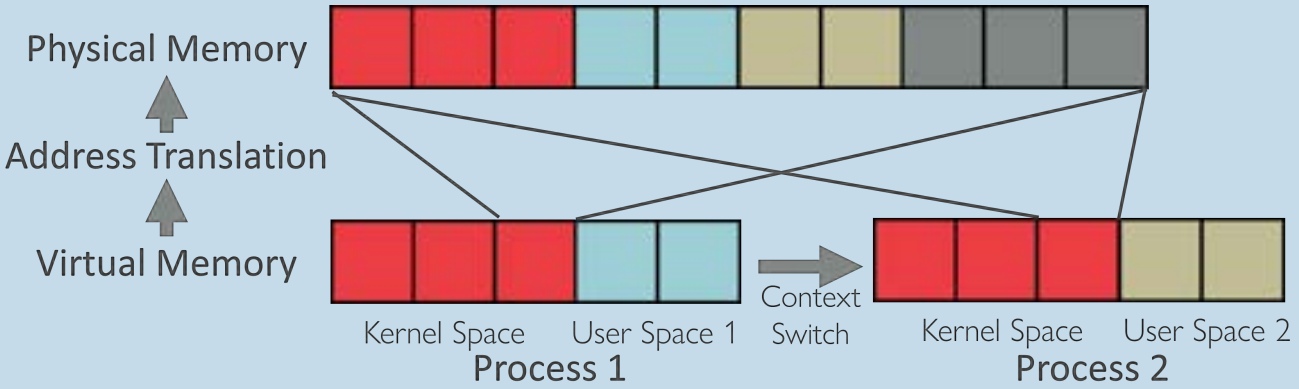
Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T. and Schwarz, M., 2020. Spectre attacks: Exploiting speculative execution. *Communications of the ACM*, *63*(7), pp.93-101.

Meltdown basics – memory hierarchy



- OS manages hardware for applications/processes
- Physical memory shared using virtual memory
- Memory hierarchy from small/fast/expensive to large/slow/cheap

Meltdown basics – address space



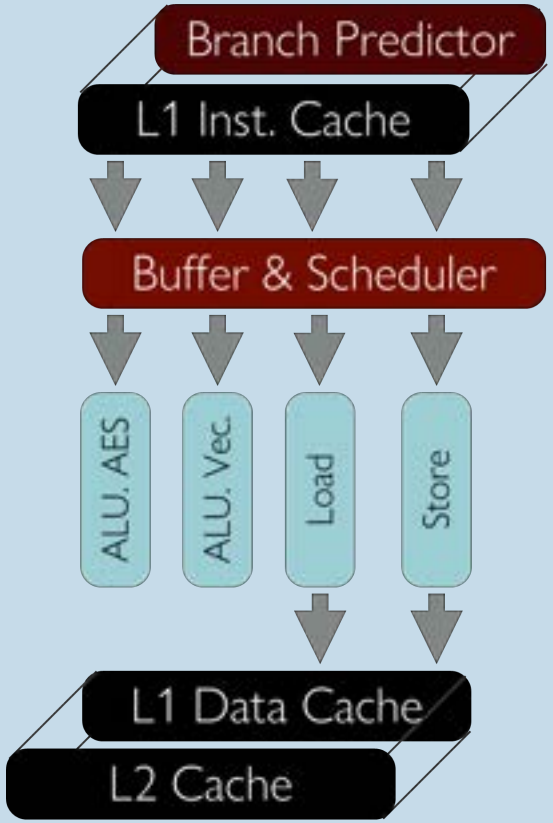
- Share physical memory resources between processes
- Entire physical memory mapped to kernel
- Isolate user and kernel spaces by checking supervisor bit



Everything works so far?

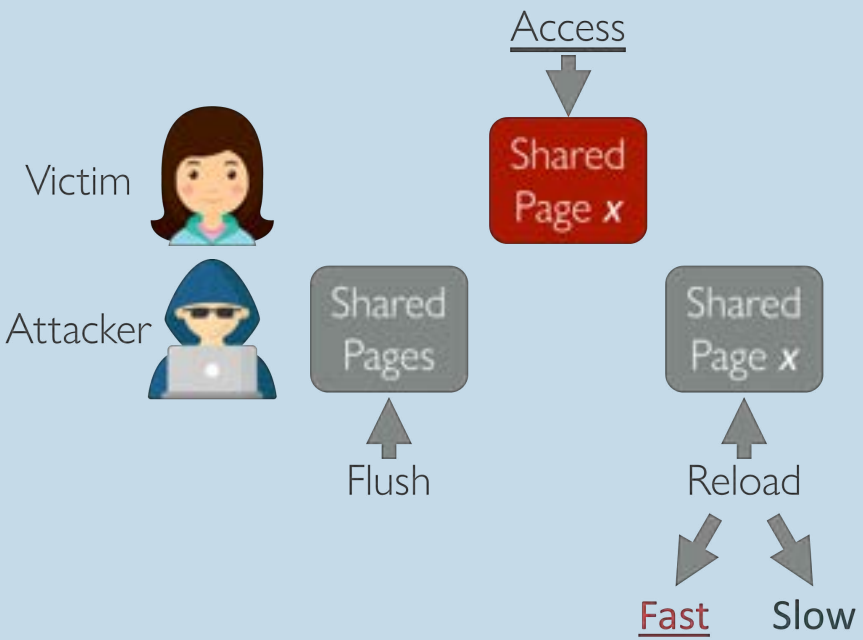
Meltdown basics – out of order execution

- Execute instructions out of program order
- Retire instructions in order
- Maximize resource usage of computing units
- Speculatively run instructions after conditional branch (if (x == 1): a = 1;)
- Revert changes for wrong predictions

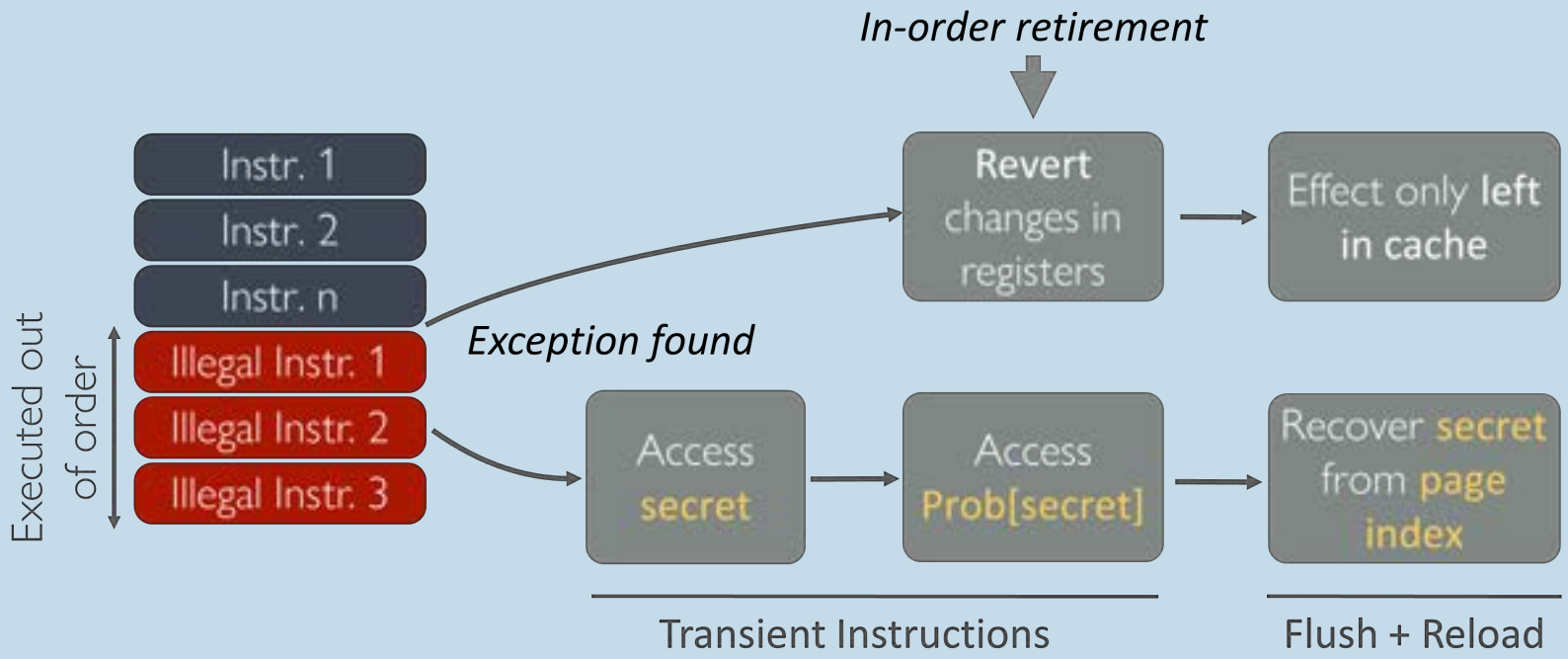


Meltdown basics – cache attack

- Cache contains microarchitectural traces
- Timing difference of cache access (Hit: fast, Miss: slow) exposes information
- Leak state of other processes from shared pages (Flush+Reload)



Meltdown overview – exploiting out of order execution

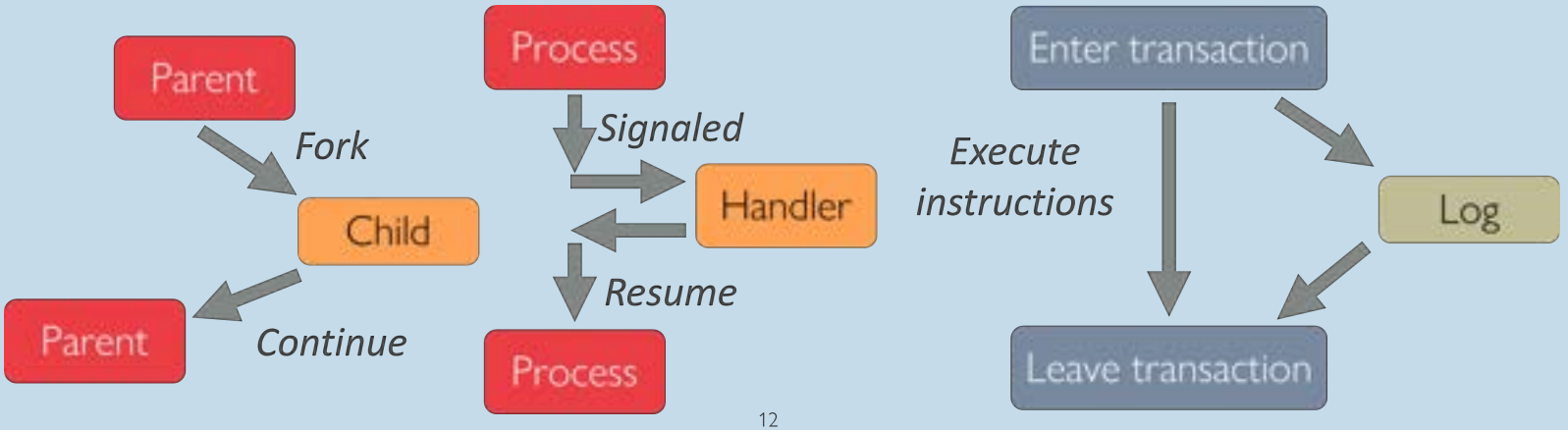




Meltdown – transient instruction

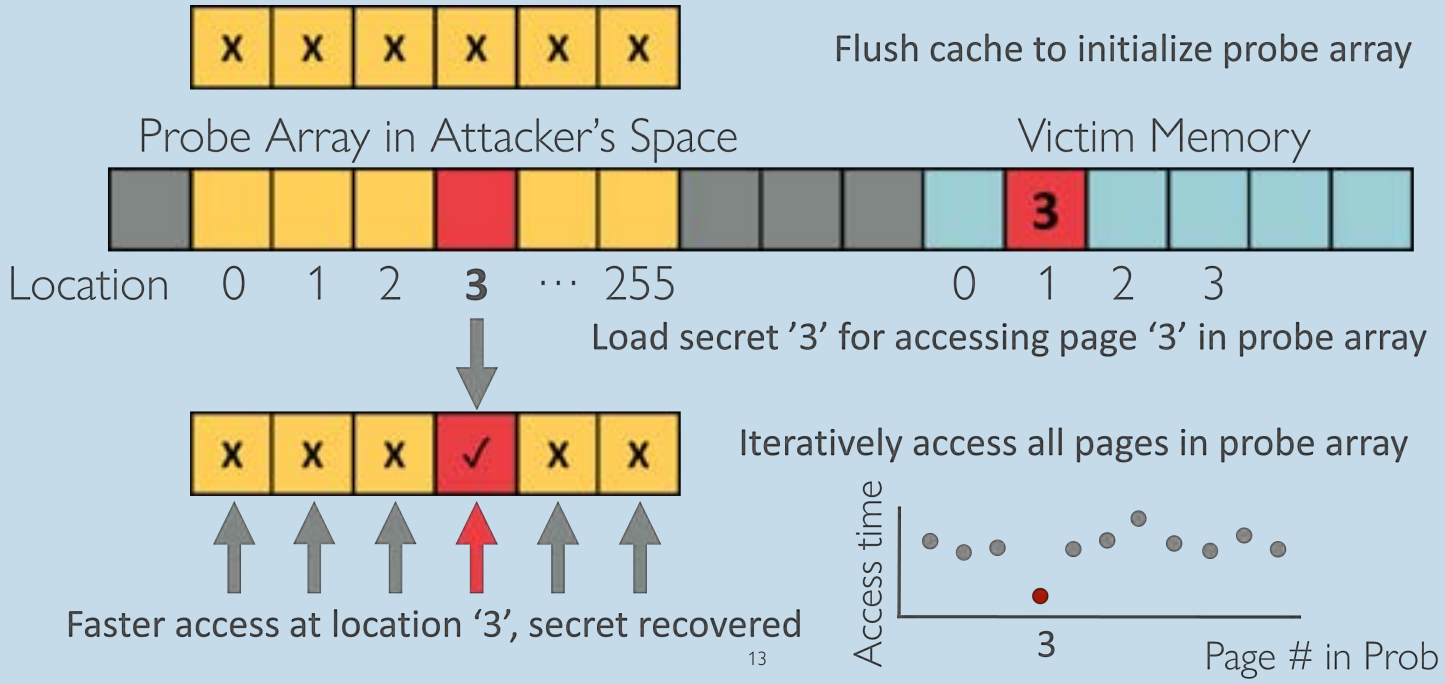
```
flush(array);
try {
  int x = *secret_banned_data;
  int y = array[x];
} catch (Exception e) {
  printf("the above never happened, right?");
}
```

Meltdown – what if exception arises?



- Handling: create child process or handle with signal handler
- Suppression: suppress using transactional memory or speculative execution rollback

Meltdown – covert channel: Flush + Reload





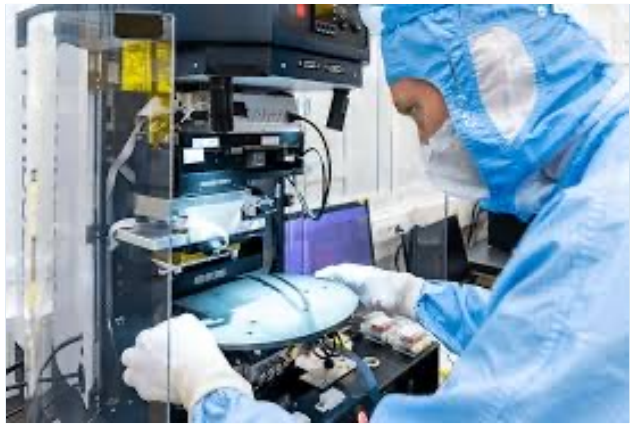
THE UNIVERSITY *of* EDINBURGH

05

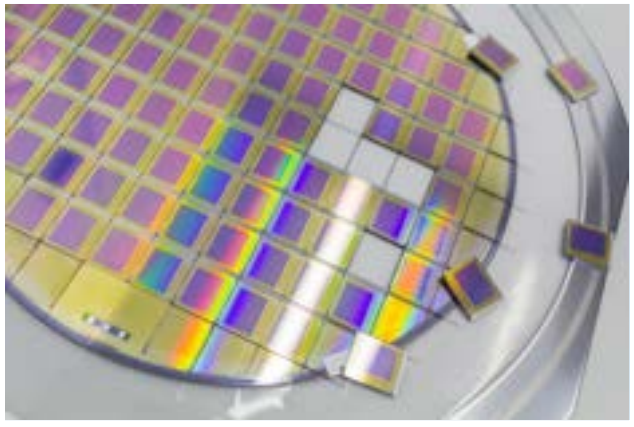
Hardware supply chain

Lithography, Trojans, reverse engineering and accessory control

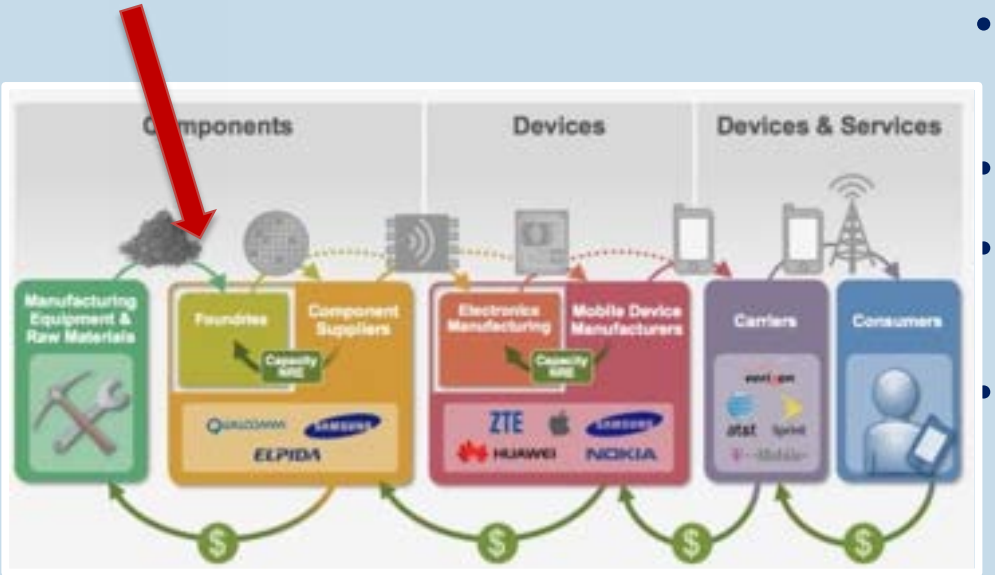
Extreme ultraviolet (EUV) lithography machines – why so important?





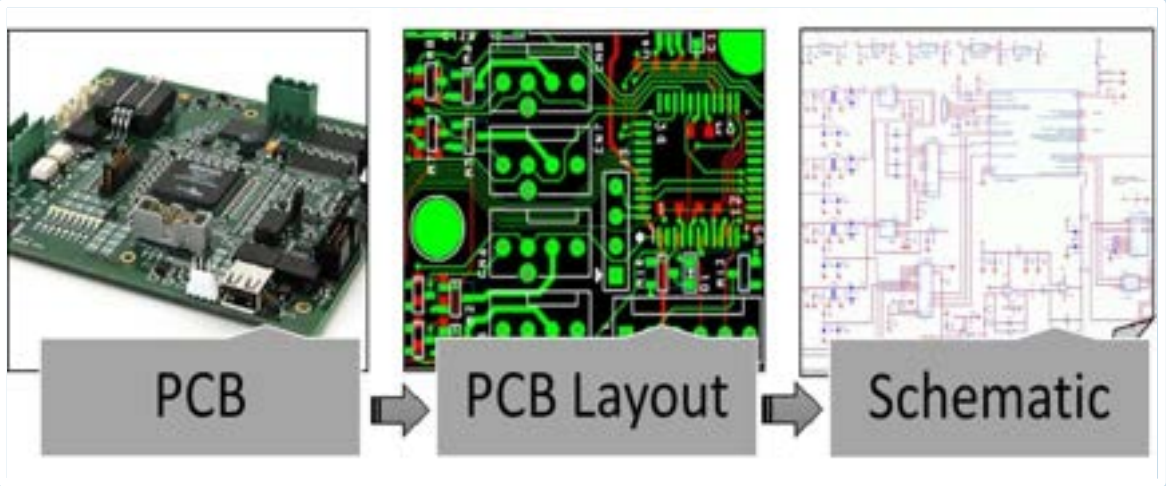


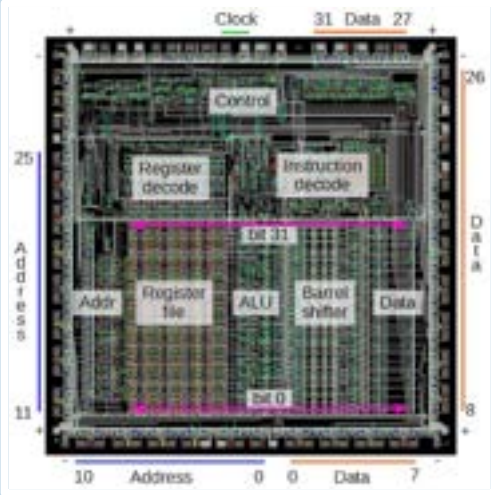
Security across the supply chain and ecosystem



- Security is segmented in supply chain and ecosystems
- Original equipment manufacturer
- Things get complicated when the attacker is highly motivated and capable
- Economics of building and governing a secure ecosystem

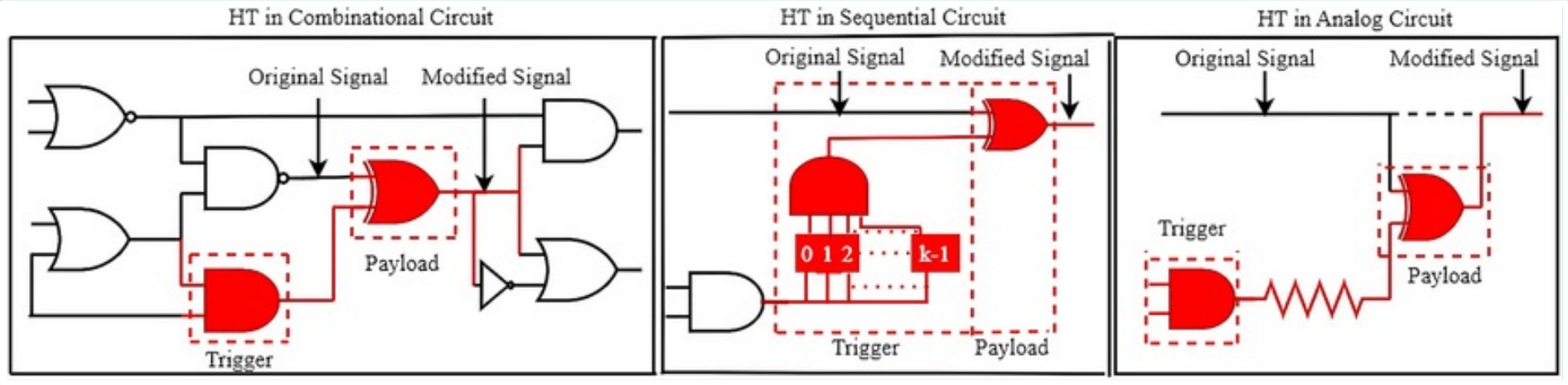
Hardware supply chain





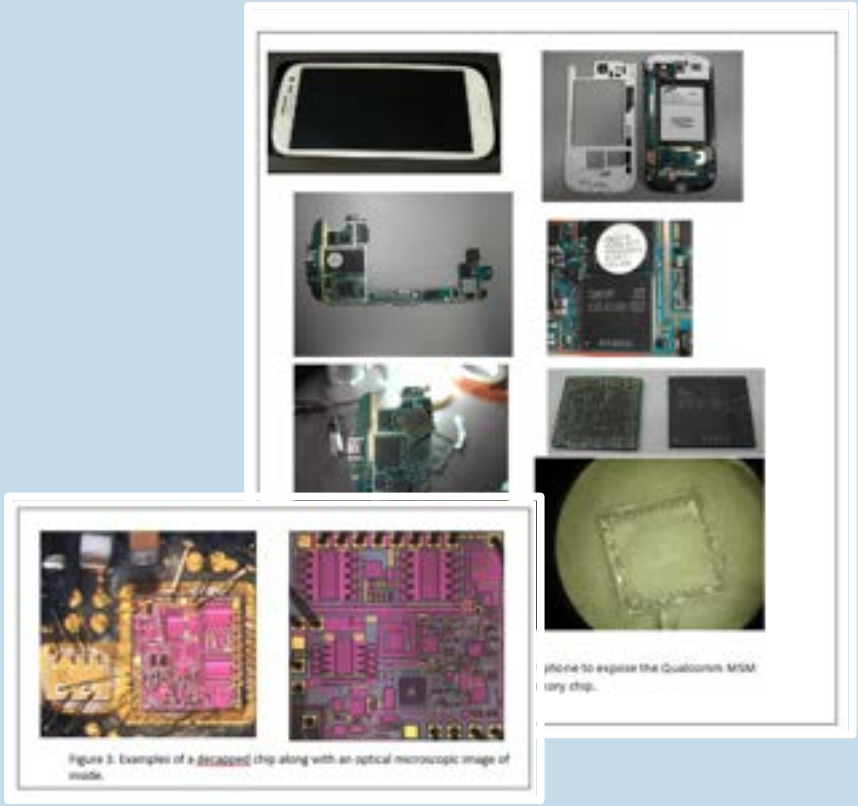
- Designers do not have full visibility into the manufacturing process
- Hardware complexity makes bugs and vulnerabilities even harder to find
- Designers/vendors even rely on reverse engineering to verify IP/chip integrity

Hardware Trojans



Hardware reverse engineering

- Sample preparation
- SEM imaging
- Image processing
- Circuit extraction
- Circuit analysis…



Accessory control



- Locking users and developers through digital rights management (DRM)
- DRM via security printing, obfuscation, tamper resistance, license check, etc.



THE UNIVERSITY *of* EDINBURGH

06

Platform security

Who controls the stack, and who ships the fix?

Platform Security





- Security is fragmented along the supply chain
- Device platform design is not heterogeneous

Android platform





- Google is taking control of the Android ecosystem, but there are still many vendors!
- Updates propagate up from the bottom
- Who is incentivized to fix a bug? Most likely Google?

Apple platform



Mobile Network Operator (MNO)	EE
Handset Original Equipment Manufacturer (OEM)	Apple
OS Vendor	Apple
Chip Maker	TSMC/Samsung
Chip Designer	Apple & ARM

- More vertically integrated
- Semi-closed ecosystem
- Apple breaks the dependency on MNOs for updates
- Longer update coverage due to fewer tussles



Discussion: which platform is more secure?

Consider: who controls the update path, how long devices stay supported, and who is incentivised to fix a bug once it is found.

Discussion: what could go wrong with mini-app permissions?





- What's the mini-app business model?
- What could go wrong in security?