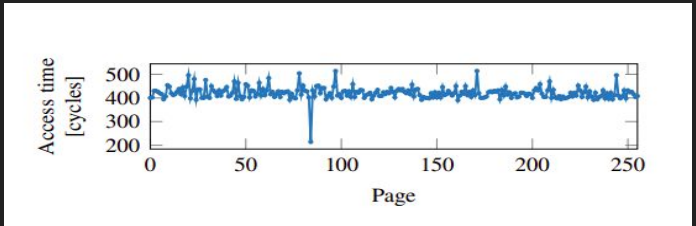
MELTDOWN

Abhinav, Nimay, Samarth, Shashank

- Kernel and User part in address spaces
- Out-of-order execution optimization
  - Micro-ops in Intel Architecture
  - Exhaustive usage of execution units
  - Inherent requirements such as compromise on protection but checking
- Cache attacks like Flush-Reload, Prime-Probe



- Building the covert channel for Meltdown Attack
  - Executing transient instructions and Exception handling
    - Possible ways of exception handling
  - Transient instruction accesses cache line using the secret value
  - Flush-Reload to transfer microarchitectural change to architectural change
- Proof-of-concept code

```
1 ; rcx = kernel address, rbx = probe array
2 xor rax, rax
3 retry:
4 mov al, byte [rcx]
5 shl rax, 0xc
6 jz retry
7 mov rbx, qword [rbx + rax]
```

- Limitations and Mitigations
  - Inherent bias towards 0 and author’s optimization
  - Dealing with KASLR, no. of steps needed for breaking 40 bit randomisation
  - Intel vs ARM: architectural differences
  - Hardware solutions
  - Software mitigations like KAISER and KPTI and their effectiveness evaluation

Thank you