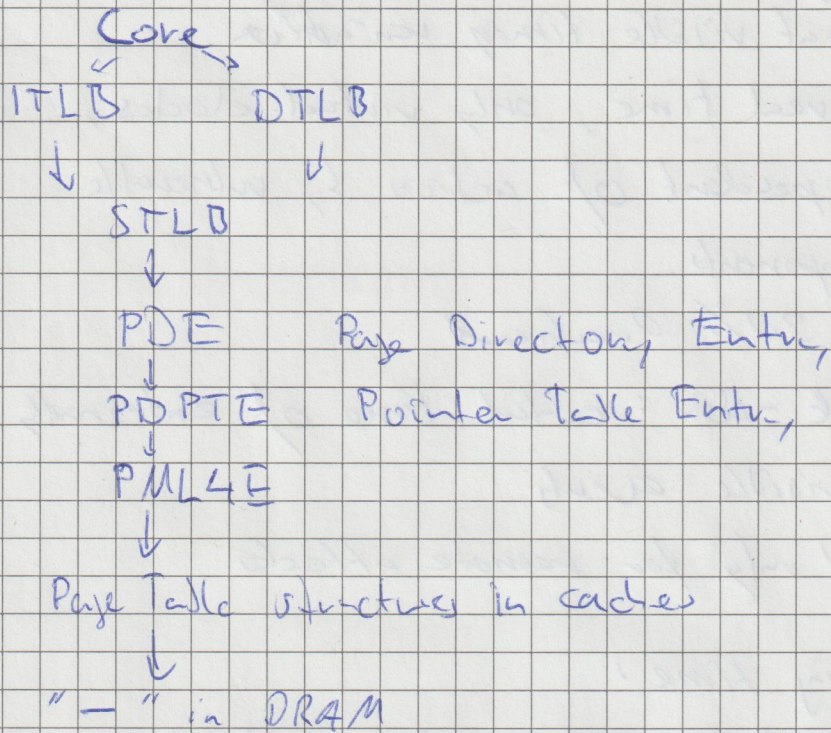


SCA 04

①



②

- Constant-Time Techniques:

Ensure Code behaviour isn't data dependent.

The sequence of cache accesses / branches isn't dependent

Performance Cost: Not practicable

Hardware Support:

Avoid vulnerable table lookups by providing

constant time hardware operations.

Language based: semantics with no variant
execution length

- Injecting noise:

"Fuzz, Time": inject noise into all
events, visible to a process

"Random Permutation code": randomized indexes

⇒ Randomly evicts

~~little cost~~ Enough noise would slow
performance

- Enforcing Determinism:

Eliminate visible timing variation

- no real time, only virtual clocks, independent of actions by vulnerable components.

⇒ 30% Overhead

- Black-Box: control time of externally visible events

⇒ only for remote attacks

- Partitioning time:

concurrent / consecutive access attacks, counter by time sliced exclusive access or managing time-slice transition.

- Flush on CPU switches: for L1 de, for lower to slow

-

- Partitioning Hardware:

Partition hardware resources between competing threads/cores

- Disable Hyperthreading, Page sharing

- L1 between threads

- Cache coloring: Prevents different processes from accessing the same cache.

→ no large pages

→ performance improvement

- Auditing's Defect by monitoring
⇒ doesn't depend

③