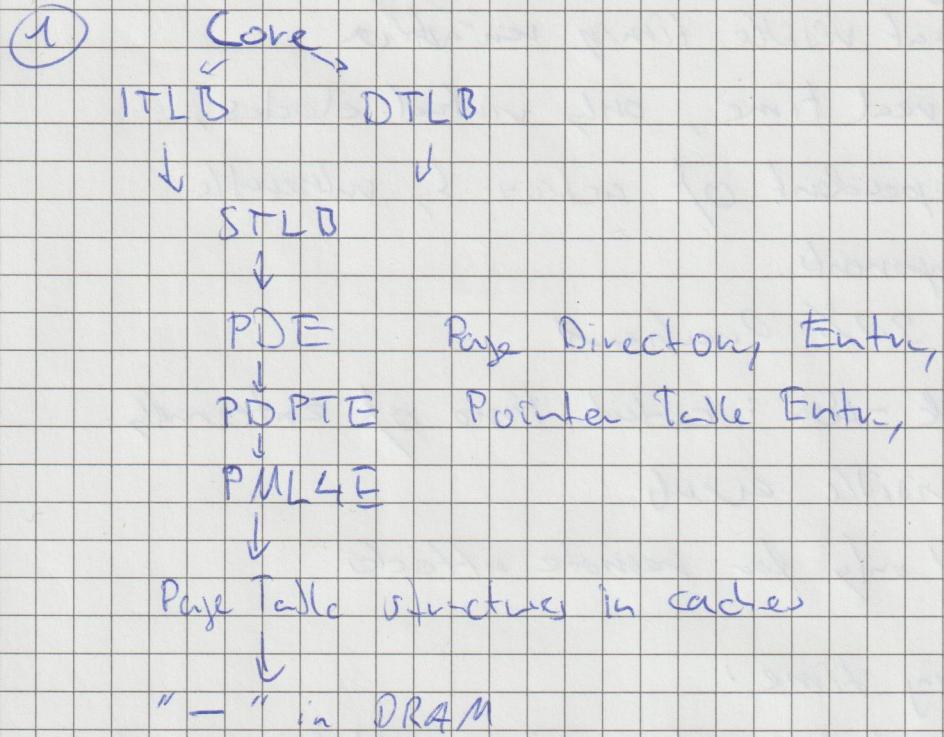


SCA 04



②

- Constant-Time Techniques:

Ensure Code behavior isn't data dependent.

The sequence of cache accesses / branches isn't dependent

Performance (or) Safe practical

↳ Hardware Support:

Avoid vulnerable table lookups by providing constant time hardware operations,

Language based: semantics with no variant execution length

- Injecting noise:

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

"Random Permutation cache": randomized indexes

⇒ Randomly evicts

~~little cost~~ Enough noise will 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 patterns, combine by time sliced exclusive access or memory time-slice transition.

- Flush on CPU Switches: for L1 ok, from lower to slow

-

- Partitioning Hardware:

Partition hardware resources between competing threads / cores

- Double Hyperthreading, Page Sharing

- L1 Slices threads

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

→ no large pages

→ performance improvement

-Activity's Effect by monitoring
⇒ closer defend

(3)