Conversions RFC v5 posted

For 2026-04-30 guest_memfd bi-weekly upstream call

Contact ackerleytng@google.com if you have questions/suggestions!

Current Status: in-place conversions v5 [1] posted

- Conversions == guest_memfd ioctl to set shared/private state
- Content mode: what userspace expects of the memory content after conversion
  - `ZERO`: When userspace next reads the converted memory, it will read zeros
  - `PRESERVE`: If host writes `0xbeef`, guest will read `0xbeef` after conversion
    - Applies in both directions
  - `MODE_UNSPECIFIED` (default): No guarantees.

[1] RFC v5: https://lore.kernel.org/all/20260428-gmem-inplace-conversion-v5-0-d8608ccfca22@google.com/T/

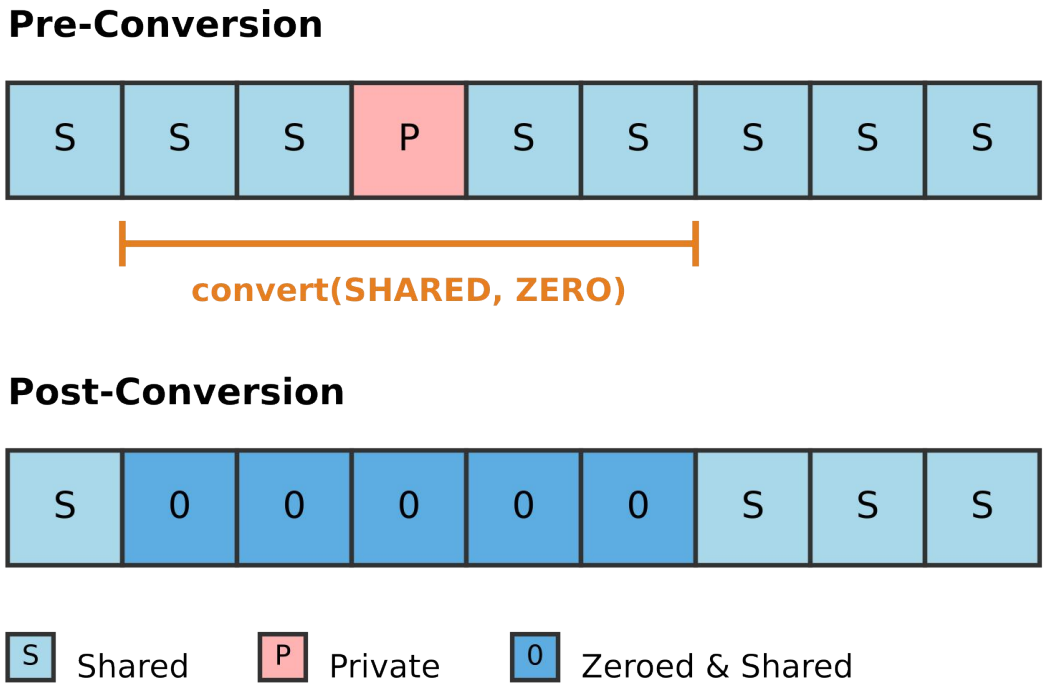
Key changes in v5

- PRESERVE only supported before TDX and SNP VMs are finalized
  - And only for conversions to private

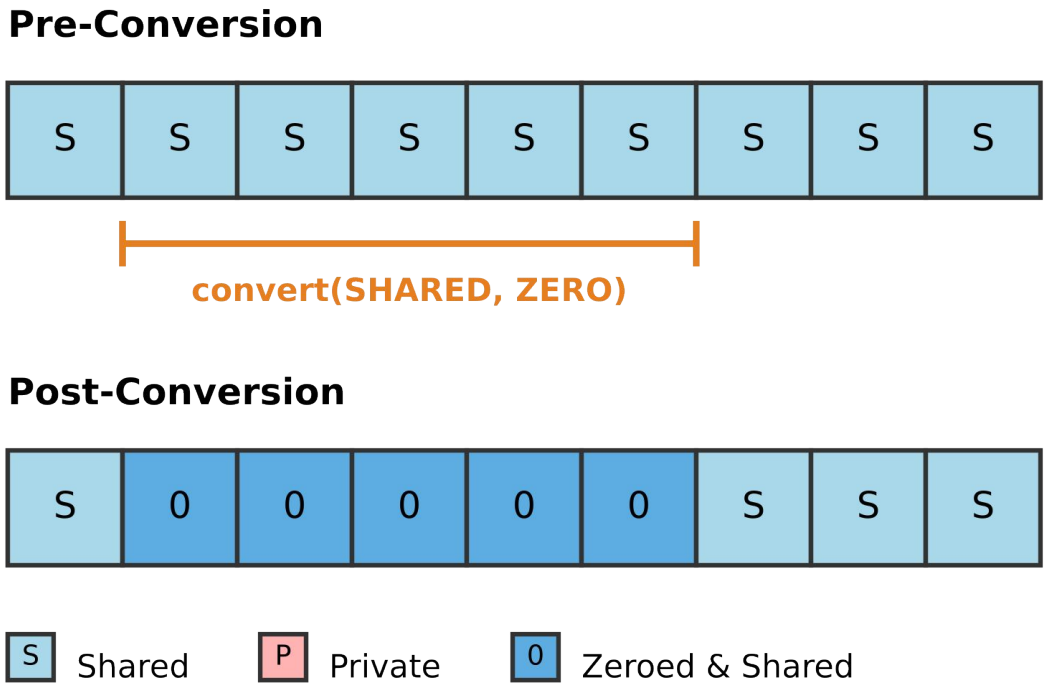
Why (TDX,SNP,CCA) PRESERVE only supported on conversion to private?

- To avoid this for TDX: before finalizing the TD:
  - convert(PRIVATE, PRESERVE)
  - populate()
  - convert(SHARED, PRESERVE)
    - ZEROs instead of PRESERVEs

Content mode applies for entire requested range



Content mode applied even if no conversion



Redefine content modes ZERO/PRESERVE?

- Current: ZERO/PRESERVE define the result after conversion
  - Awkward: ZERO is a result, PRESERVE is an action
- After: define what guest_memfd does during conversion
  - PRESERVE == guest_memfd does nothing
  - ZERO == guest_memfd applies zeroing
- PRESERVE currently means slightly different things
  - SW_PROTECTED_VM: guest will read what host wrote (no encryption)
  - TDX/SNP: guest won't ever get to read what host wrote directly, must be used in conjunction with populate

[1] https://lore.kernel.org/all/CAEvNRgGbMhkX310CkFY_M5x-zod=BDTiuznrZ0XvFPUK7weL1A@mail.gmail.com/