Investigating the feasibility of using CHERI-enabled Arm Morello boards for cloud-based trusted execution

Kian Cross

kian.cross@cl.cam.ac.uk

Can we utilise CHERI features to build a trusted execution environment (TEE) with the properties of Intel SGX?

- **Why would we want to do this? Why not just use Intel SGX?**
  - Because it might be nice to extend the utility of a technology such as CHERI to other applications.
- **Why is this a plausible question to consider?**
  - Intel SGX facilitates mutual distrust between pieces of code executing in an enclave and the outside world (including code running outside of the enclave).
  - At least on the surface, CHERI also enables mutual distrust: privileged code can allocate memory to an application and then drop its capability. But whilst this looks, to some extent, like mutual distrust, does it satisfy the properties we need?

What properties do we need?

- The operation of TEEs should be opaque, prohibiting both observation and manipulation of data and computations by other TEEs and privileged code.
- There should exist a mechanism to remotely attest that a particular TEE is indeed executing on the anticipated system.
  - There should be some assurance that this will remain the case for the life span of the TEE, not just when the attestation is made.

Why is the way Intel SGX achieves these good?

- **Hardware-only trusted computing base (TCB)***
    - We can attest the code running in an enclave without also needing to attest the software stack below it.
    - Reduced probability of malicious and accidental vulnerabilities / Reduced attack surface. (?)
    - No privileged code (e.g., hardware-assisted context switches and memory allocation for enclaves).
- *This is not entirely true: arguably, the toolchain should be considered part of the TCB.

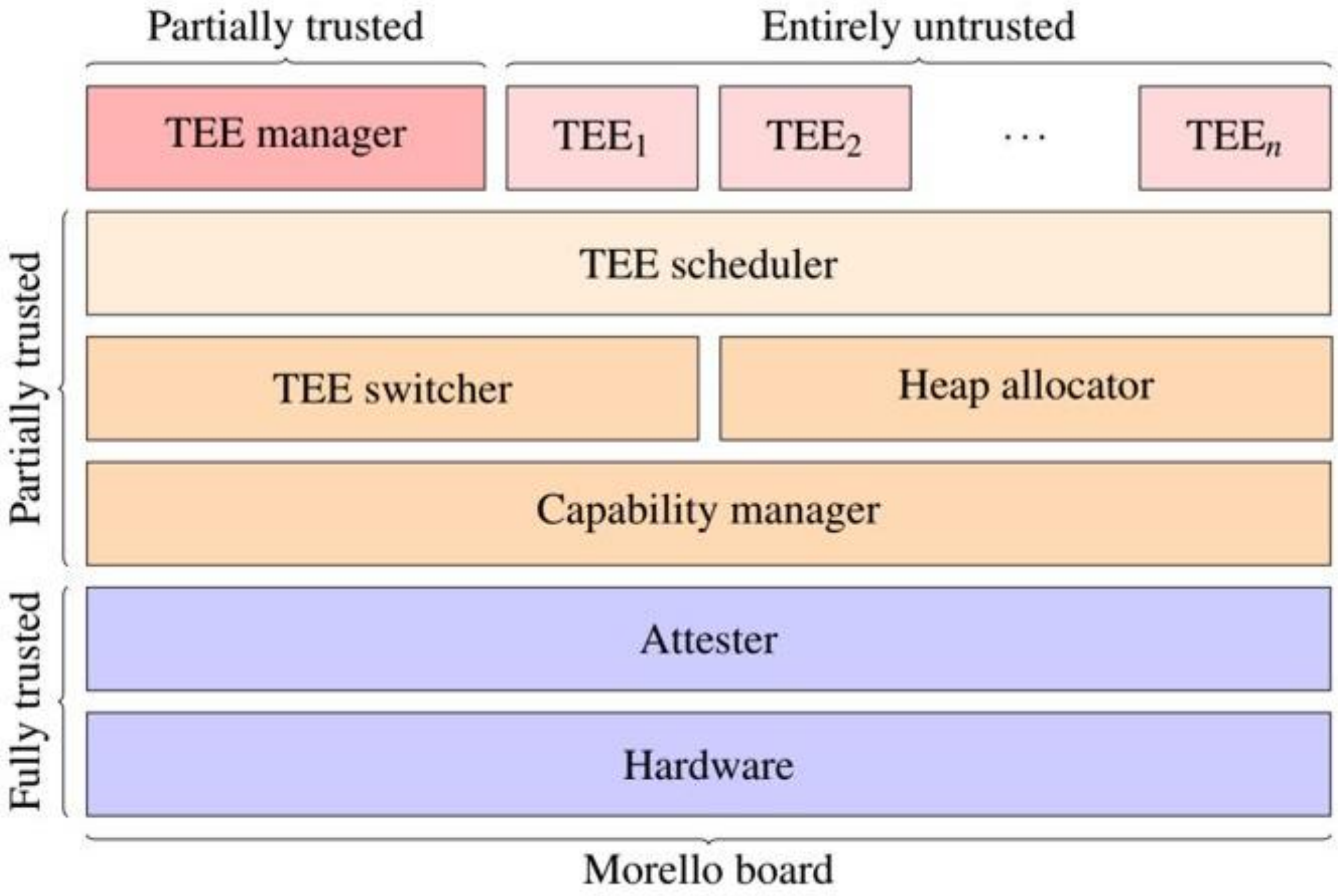
Can we replicate these properties with CHERI/Morello?

- CHERI/Morello does not provide a mechanism for hardware-assisted context switching / memory allocation.
- So, there must always be a software TCB: at least a trusted interrupt handler and allocator.
- CHERI/Morello does not provide a mechanism for remote attestation.
  - But even if it did, the software TCB requires the entire software stack to be attested. Does this make verification of the attestation impractical?
- And even with all of these features, physical access to the board provides significant privileges.
- *These features are outside the scope of CHERI/Morello, so it is not necessarily a criticism that they are not provided.*

Can we achieve weaker but still useful properties?

- The operation of TEEs should be opaque, prohibiting both observation and manipulation of data and computations by other TEEs and privileged code.
    - **We could relax this property to "most privileged code", which would allow us to programme the missing functionality. CheriOS and CHERI-TrEE do this. The aim is then to minimise the TCB and ensure each component of the system has the least privileges required to execute.**
- There should exist a mechanism to remotely attest that TEEs are indeed executing on the anticipated system.
    - **We could also do this in software (i.e., a trusted attester), but does this weaker assurance still provide a useful property?**
    - There should be some assurance that this will remain the case for the life span of the TEE, not just when the attestation is made.
        - **Given the cloud context, can we exclude hardware attacks from the threat model?**

teriOS Design



Demo

Conclusion

- Achieving Intel SGX-like properties is ambitious: SGX is a collection of many complementing technologies (e.g., anti-tamper, attestation, memory encryption, enclaves etc.) under a single product.
    - So, there is lots to replicate!
- But because we are specifically targeting the cloud, it might be possible to focus on a subset of these properties / looser threat model.
- Still, a lack of any mechanism for remote attestation is a challenge, but probably the easiest to overcome, with existing hardware.