Enabling Design Space Exploration for RISC-V Secure Compute Environments

Ayaz Akram (yazakram@ucdavis.edu),
Venkatesh Akella, Sean Peisert, Jason Lowe-Power

Summary

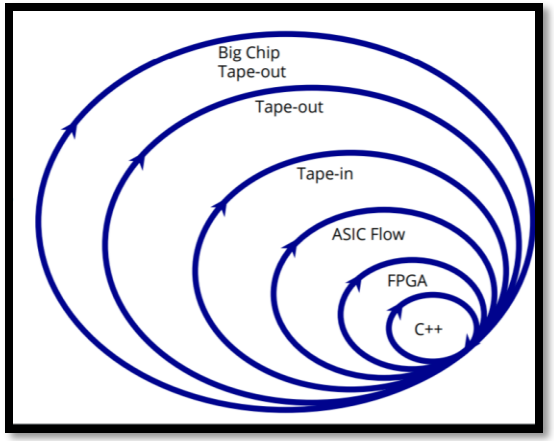
Argue the need of cycle-level architectural modeling of secure compute environments

Extended gem5 to simulate RISC-V based secure compute environments like Keystone

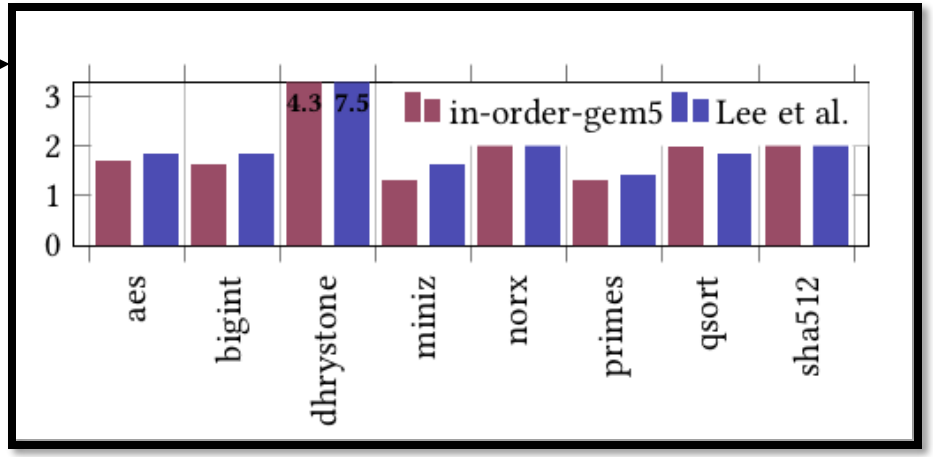
gem5's Keystone modeling shows similar performance behavior as in the work of Lee et al[1]

Present some use cases of gem5's ability to simulate RISC-V TEEs (Trusted Execution Environments)

Agile Hardware Design Stack



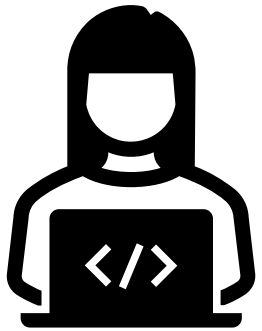
Slowdown of Trusted Execution



[1] Lee et al., Keystone: An open framework for architecting trusted execution environments, In EuroSys 2020.

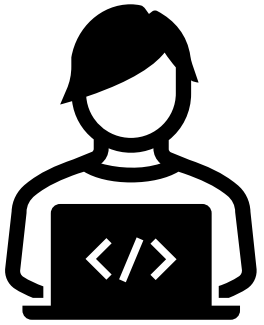
Story of Alice and Bob

Bob is chatting with Alice!

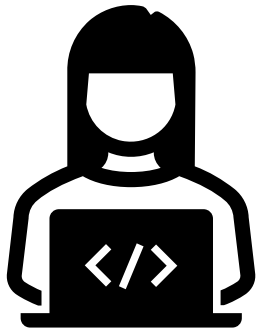


I have some proposals to design new confidential compute architectures!

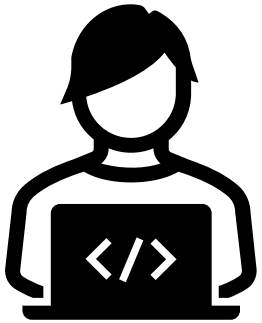
No idea where to start, the current TEEs (SGX, SEV) are closed source!



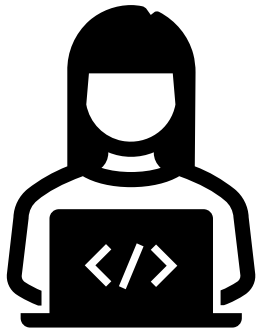
Alice has some solutions for Bob!



You can use RISC-V based TEEs. Keystone is an example of open-source TEEs.

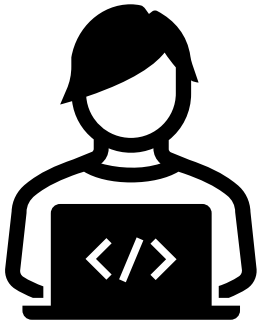


Bob gets excited about open-source TEEs

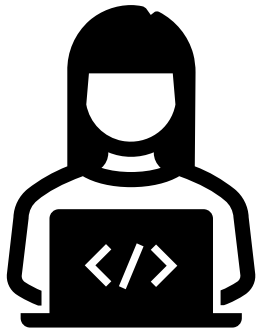


This seems great!

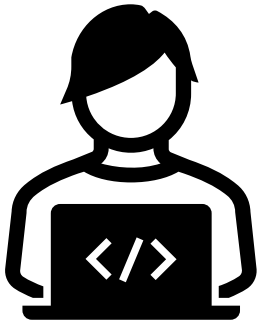
But how do I evaluate my ideas?



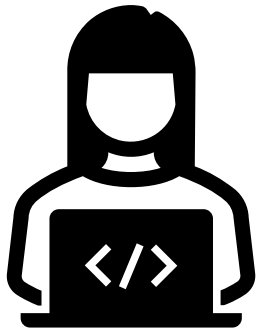
Alice suggests some tools!



RISCV ecosystem provide s capabilities to do functi onal or RTL simulation of TEEs using QEMU and FireSim.

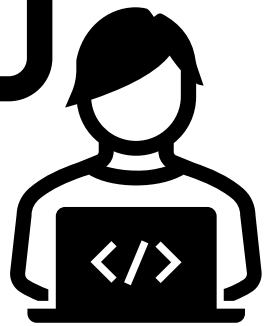


Bob wants to do design space exploration!

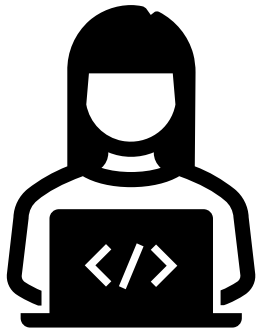


But I want to do extensive hardware /software design space exploration using a flexible tool.

QEMU will be not provide any cycle-level information. FireSim will not be suitable as I do not have a fixed design that I want to evaluate.

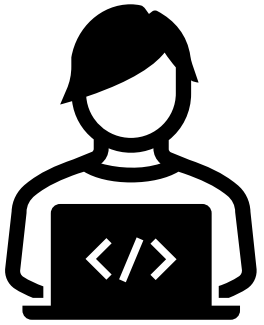


Alice has no solution for Bob!



gem5, a cycle-level full system simulator fits some of your needs.

Unfortunately, I don't think you can evaluate TEE designs on gem5 currently!



Bob's excitement fades away 

We want to help Bob evaluate his ideas!

Therefore,

We added the capability to evaluate design of RISC-V secure compute environments (like Keystone) in gem5

Outline

Background on RISC-V Isolation Mechanisms and gem5

Demo of Keystone in gem5

Results

Background on RISC-V Isolation Mechanisms and Keystone

RISC-V Isolation Mechanisms

RISC-V Privileged Modes (for vertical isolation)

- U/S/M modes (H-mode not in stable revision of specs)

Virtual Memory Management

- different schemes for 64-bit systems (Sv39, Sv48)
- managed by S-mode software

RISC-V PMP (Physical Memory Protection)

- controls access of U/S mode to certain memory regions
- memory region and access permissions defined by pmpaddr/cfg registers

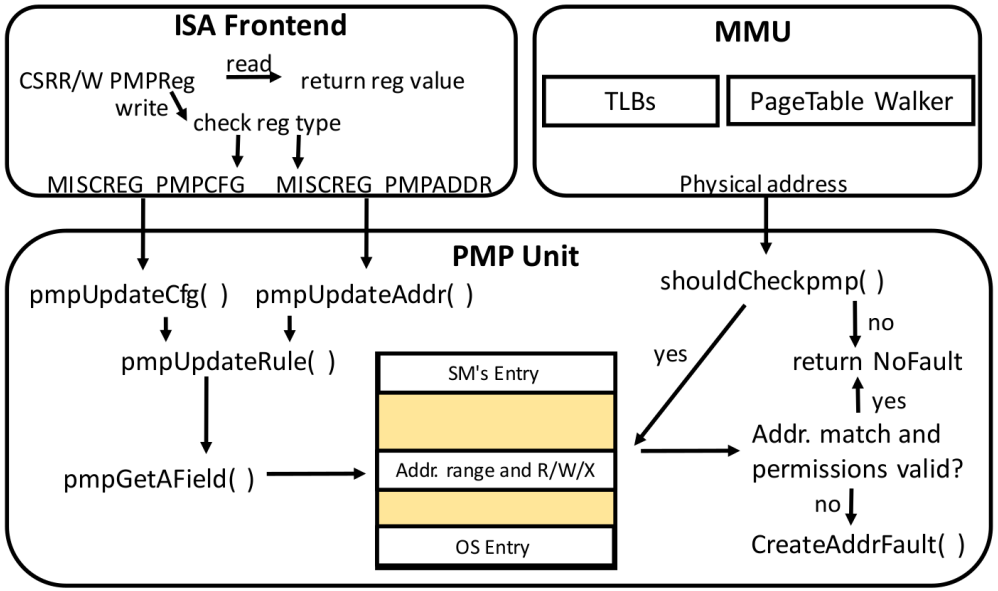
PMP Implementation in gem5

Three main components
- ISA subsystem
- MMU unit
- PMP Unit

ISA Frontend handles reading/writing of PMP registers

MMU unit consults PMP table once physical address is generated

PMP unit responds with success or fault



- Will be part of gem5's 21.1 release
- gem5 source: https://gem5.googlesource.com/public/gem5/
- PMP implementation: src/arch/riscv/pmp.cc

Outline

Background on RISC-V Isolation Mechanisms and gem5

Demo of Keystone in gem5

Results

Demo

Outline

Background on RISC-V Isolation Mechanisms and gem5

Demo of Keystone in gem5

Results

- **Performance Validation**
- Use Cases

Performance Validation

RV8 Benchmarks

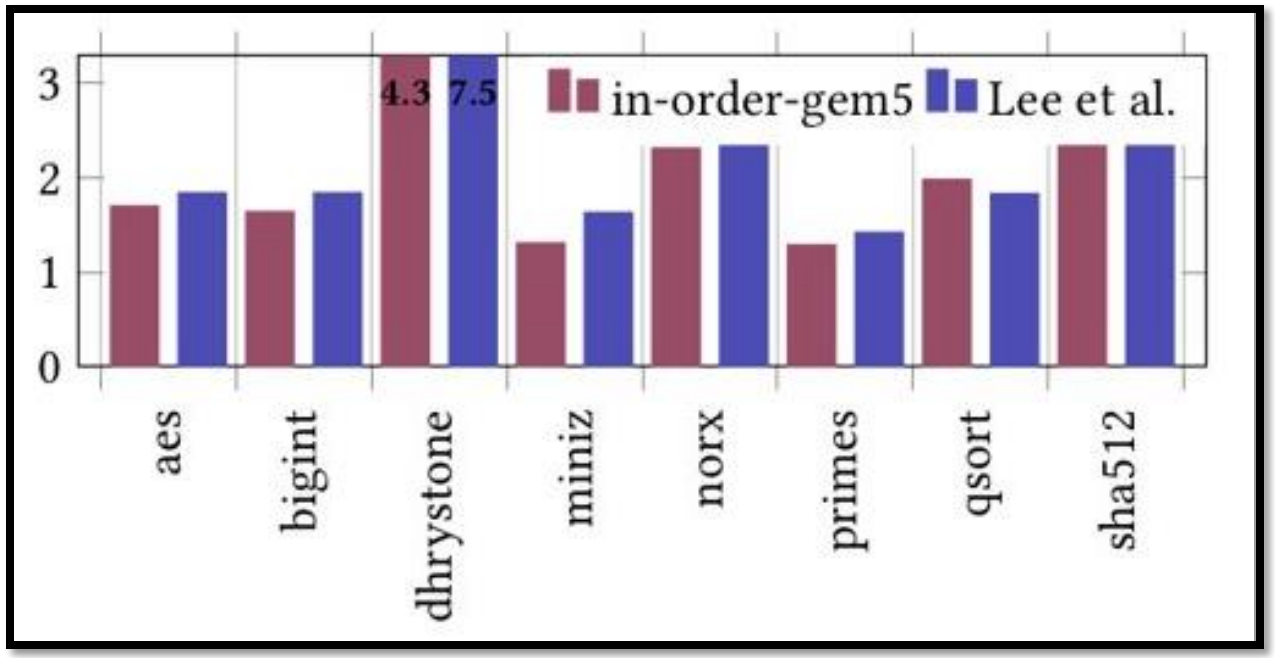
Used by Lee et al. [1]

gem5 and work of Lee et al.[1]

Similar performance numbers

Similar trends

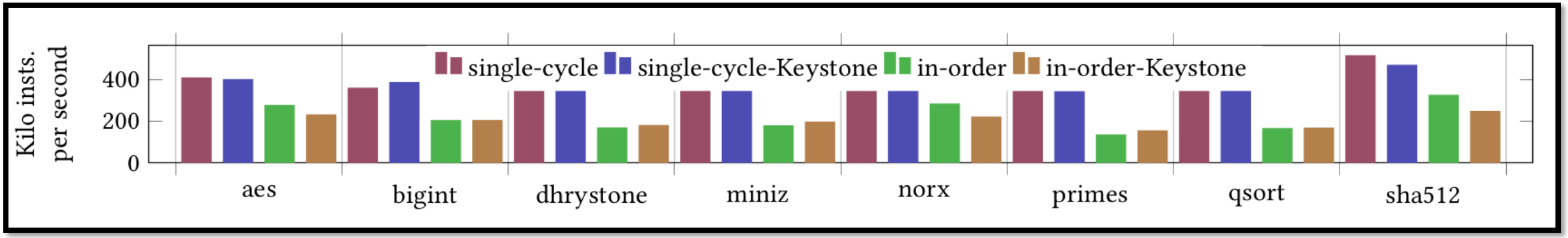
Slowdown of Trusted Execution



[1] Lee et al., Keystone: An open framework for architecting trusted execution environments, In EuroSys 2020.

gem5's Performance (Simulation Time)

Simulation Performance of gem5



Difference in simulation throughput for different CPUs

Untrusted and Trusted simulation shows same throughput

Roughly 10,000x slowdown over QEMU

Outline

Background on RISC-V Isolation Mechanisms and gem5

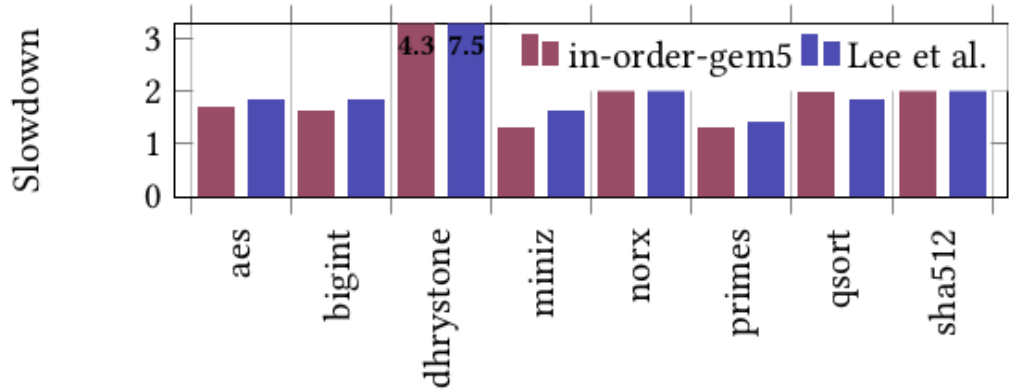
Demo of Keystone in gem5

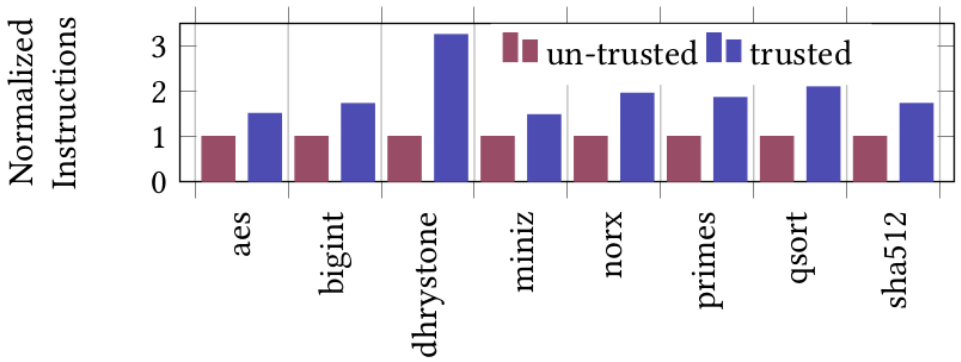
Results

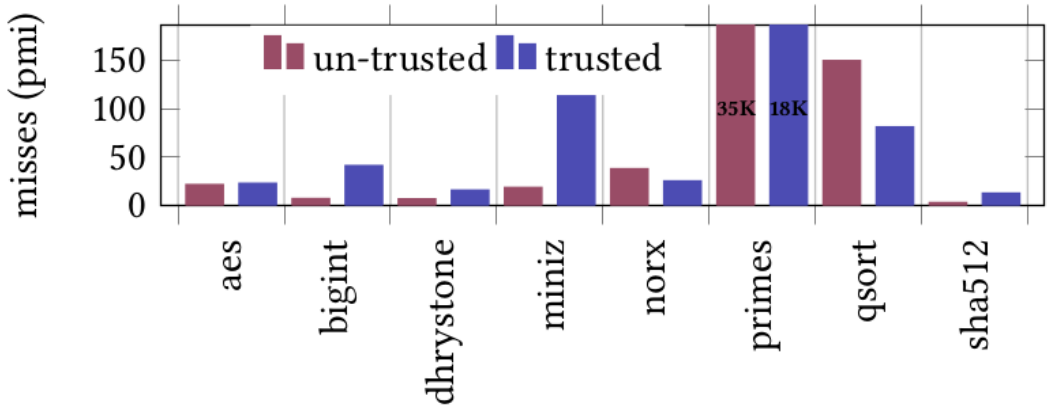
- Performance Validation
- **Use Cases**

Use Cases of Ability to Simulate Secure Compute Environments

Use Case 1: Analyzing microarchitectural statistics to understand performance implications



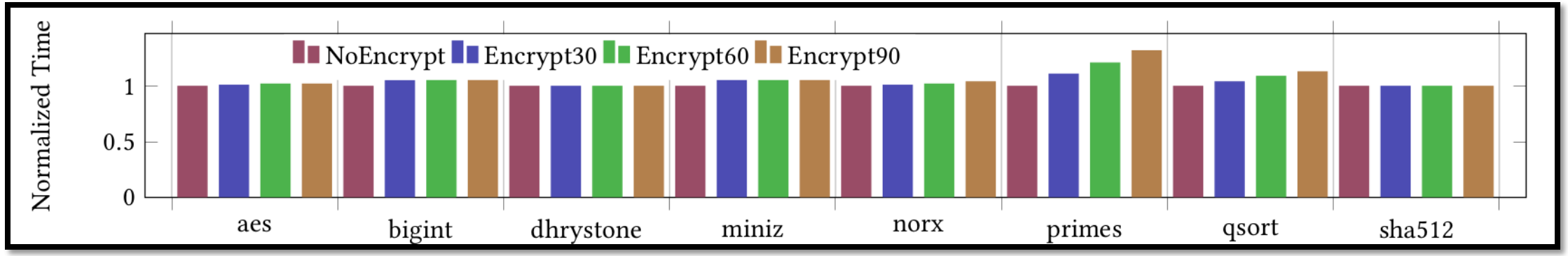




Use Case 2: Performance of Memory Encryption

Modeling an encryption engine with different latencies (30, 60, 90 cycles)

Largest slowdown: 32% for *primes*

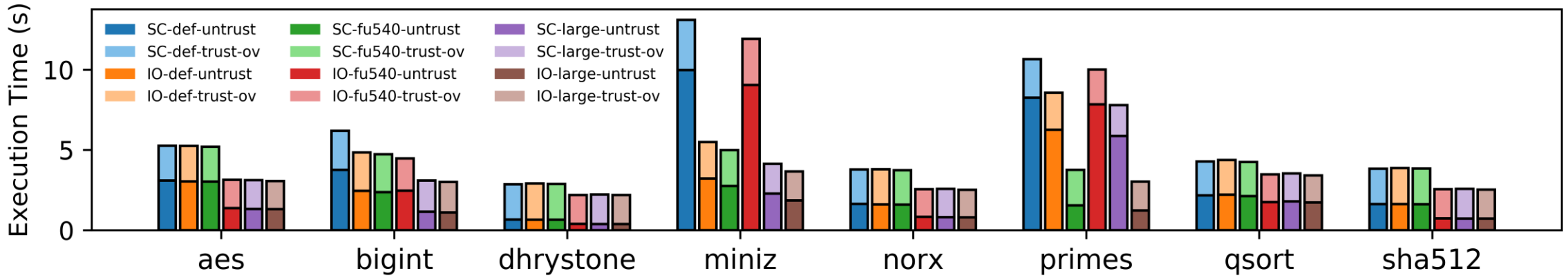


Use Case 3: Micro-architecture impact on performance of trusted execution

Overall execution time goes down as we move towards more aggressive configurations, however the ratio of trusted to untrusted execution time for each configuration stays similar.

Tested cache subsystems

Feature	default	fu540-like	large
Dcache size	32KB	32KB	512KB
Dcache assoc.	8	8	8
L2 cache	N/A	2MB	16MB
L2 cache assoc.	N/A	16	32
DTLB entries	64	128	2048



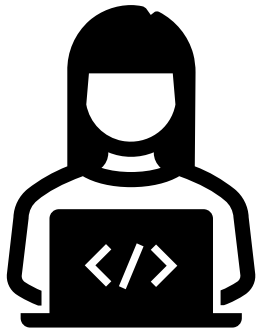
Future Use Cases

- Quantified vulnerability analysis for TEEs and other secure compute environments
  - Architecture Vulnerability Factor (AVF)
  - Side-channel Vulnerability Factor (SVF)

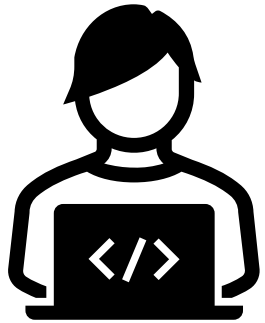
- Ability to observe the state of an entire system or parts of it

Back to the story of Alice and Bob

Alice has a solution for Bob now!



Now, gem5 fits your needs. You can use it to evaluate RISC-V based secure compute environments.



Bob is excited again 

Enabling Design Space Exploration for RISC-V Secure Compute Environments

Ayaz Akram (yazakram@ucdavis.edu),
Venkatesh Akella, Sean Peisert, Jason Lowe-Power