Confidentiality
- secret information should not flow to untrusted region

Integrity
- no information flow from untrusted region to secure location

Confidentiality and integrity are 2-safety properties
- Properties refuted by observing two finite traces
- A trace is sequence of execution states, \( t = s_0f_1 \ldots s_n \)
- 2-safety property is from the class of Hyperproperties

Integrity is a dual of confidentiality

Proving confidentiality and integrity
show system leaks no secret information
or
show execution traces are indistinguishable to untrusted entity

\[ \text{Sim}(I,CKey_j) \]

\[ \text{Sim}(I,CKey_j) \]

Goal:
- automated technique for finding 2-safety hyperproperty violations
- language for specifying security properties

Hardware modeling
- Verilog implementation

Intermediate representation
- Instrument Verilog model to collect simulation metric

Fuzzer
- A variant of AFL to be used along with Hyperproperties

Conclusion
- Existing solutions for SoC security validation are not effective
- Fuzzing has the potential to be scalable.
- It has been successful in finding many software vulnerabilities.
- HYPERFUZZING leverages power of fuzzing to find security violations in SoCs.

References