SoC functionality is implemented by a combination of hardware and firmware.

**Verification Challenges**
- Verifying the complete HW+Fw design is not scalable
- Separate verification of HW and FW misses bugs

**Verification of Security Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>Sensitive values must not leak to untrusted entities</td>
</tr>
<tr>
<td>Integrity</td>
<td>Untrusted entities must not influence sensitive values</td>
</tr>
<tr>
<td>Availability</td>
<td>Untrusted entities must not be able to render the system non-functional</td>
</tr>
</tbody>
</table>

Difficult to specify confidentiality and integrity in LTL because they are not predicates of state; instead they refer to information flow.

**Specifying Information Flow Properties on the ILA**

- Can capture both confidentiality and integrity
- We specify properties on an augmented ILA
  - High-level system state such as user/su mode, current thread and VM ids, and son
  - Convert events such as user/su state-transitions into state variables

**Synthesizing Instruction-Level Abstractions**

- Construct abstraction at instruction-granularity
- Synthesize abstraction from a template
- Verify correctness: ensure ILA matches RTL

**Conclusions**

Two main challenges in verifying security properties in SoCs are:
- Considering HW+FW issues together
- Specifying and verifying security properties like confidentiality and integrity

- Template-based synthesis of ILAs solves the HW+FW verification problem
- Specification language and verification techniques for information flow properties solve the security verification problem!