The TPM as Controlled State Repository: Modelling Trusted Platform Modules

Ariel Segall
Joshua Guttman

October 2, 2008
The Trusted Platform Module

- Small trusted hardware chip
- Relied upon to store keys & measurements
- Limited command set
- Predictable behavior (we hope!)
Controlled State Repositories

Controlled state repository:

- Stores data
- Modifies data in restricted patterns
- Interaction results indicate current state

The TPM is a controlled state repository!
PCR_Reset Role

\[ \leftarrow \text{PCR}_{\text{Reset}}(p, \text{mask}, \text{locality}) \]
\[ \Downarrow \]
\[ E_{\text{PCR}_{\text{Reset}}}(p, \text{mask}, \text{locality}, \_ , \_ ) \]
\[ \Downarrow \]
\[ \rightarrow \text{PCR}_{\text{ResetResult}}(p) \]
PCR_Reset Rules (1/2)

\[
\begin{align*}
\text{PCR}_\text{attrs}(p, \text{attrs}), \\
\text{PCR}_\text{values}(p, \text{pcrs}), \\
\rightarrow \\
E_{\text{PCR}_\text{Reset}}(p, \text{mask}, \text{locality}, \text{newpcrs}, \text{attrs})
\end{align*}
\]

when

\[
\begin{align*}
\text{Resettable}(\text{attrs}, \text{mask}), \\
\text{Rst}_\text{loc}(\text{attrs}, \text{mask}, \text{locality}), \\
\text{Defaults}(\text{mask}, \text{newvalues}), \\
\text{Vecs}_\text{differ}_\text{only}(\text{pcrs}, \text{newpcrs}, \text{mask}, \text{newvalues})
\end{align*}
\]
PCR_Reset Rules (2/2)

\[ E_{\text{PCR\_Reset}}(p, \text{mask}, \text{locality}, \text{newpcrs}, \text{attrs}) \]
\[ \rightarrow \]
\[ \text{PCR\_attrs}(p, \text{attrs}), \]
\[ \text{PCR\_values}(p, \text{newpcrs}) \]
State Effects from PCR_Reset

`PCRAttrs(p, attrs),
PCRValues(p, pcrs),
→
...
→
PCRAttrs(p, attrs),
PCRValues(p, newpcrs)`
Advantages of modelling the TPM

- **Simplicity**
  - >700 page specification
  - <40 page model (partial)

- **Comprehensibility**
  - Abstract away details
  - Focus on external effects

- **Analysis**
  - What we can rely on
  - What the TPM relies on
Analysis with the model

What guarantees does the TPM provide?

▶ State before command? After?
▶ State invariants?
▶ Possible past or future behavior?
▶ Authorization of this command or others?

The spec makes lots of invisible assumptions!
Going forward

- Implement rules in Maude
- Extend model
- Integrate with protocols & ALIS
- Provide model to other researchers
- Model other controlled state repositories
Modelling a Controlled State Repository

- Each CSR is one principal
- Interactions (commands) are strand space protocols
- State changes are multiset rewriting rules
- Event nodes in strands trigger rules
- A rule affects only one principal
- All failures look the same (for now)