CPS:Small: Establishing Integrity in Dynamic Networks of Cyber Physical Devices

Vinod Ganapathy (Rutgers), Ulrich Kremer (Rutgers) and Trent Jaeger (Penn State)

CNS-0931992 and CNS-0931914

Problem: Establishing Trust Dynamic Networks of CPS

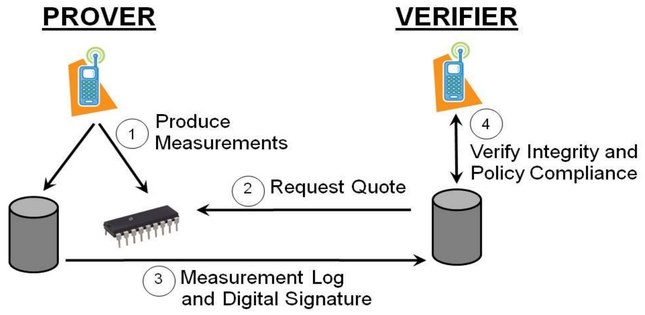
Dynamic Networks of CPS Devices

- Network of resource-constrained CPS devices such as smart phones and embedded systems.
- Distributed computations performed by several participating devices.

Security threats in Dynamic Networks

- Participating devices cannot be trusted
- Some malicious devices may perform harmful computations affecting the integrity of the results
- Need energy-aware mechanisms to verify trustworthiness of devices

Solution: Attestation protocols



Incorporated a trust-establishment protocol with the Sarana dynamic network

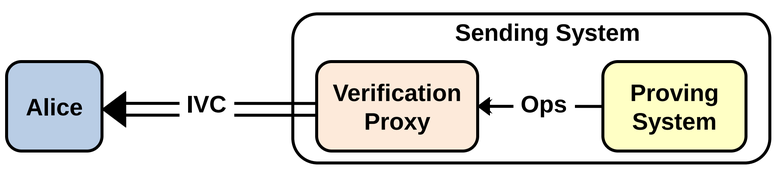
Verifying Distributed Networks

Large heterogeneous networked systems

- Distributed systems often contain diverse systems with varying configurations and security requirements
- Verification of individual components requires time- and energy-consuming attestation and knowledge of how to assess component integrity

Integrity-Verified Channels

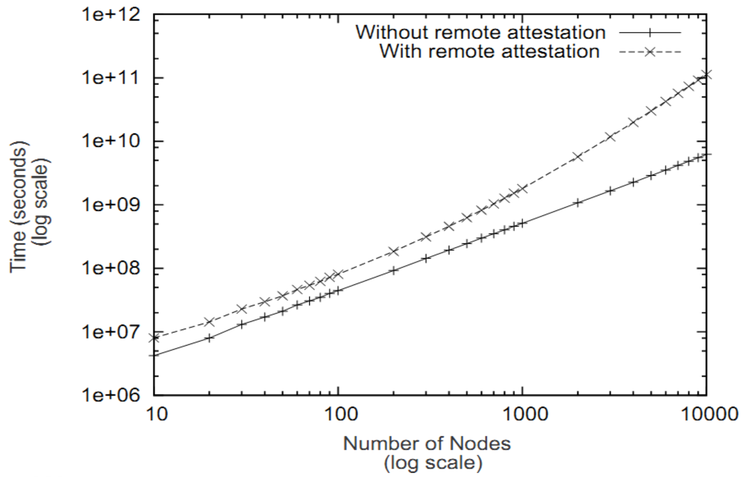
- Traditional attestation protocols prove integrity at a particular time, requiring repeated proof generations
- Instead, *integrity-verified channels* bind a secure communication channel to the sending system's integrity
- An *integrity verification proxy* administers IVCs by acting only on violations to integrity policy
- Thus, an IVC requires only one attestation for the channel's lifetime and monitoring cost is negligible

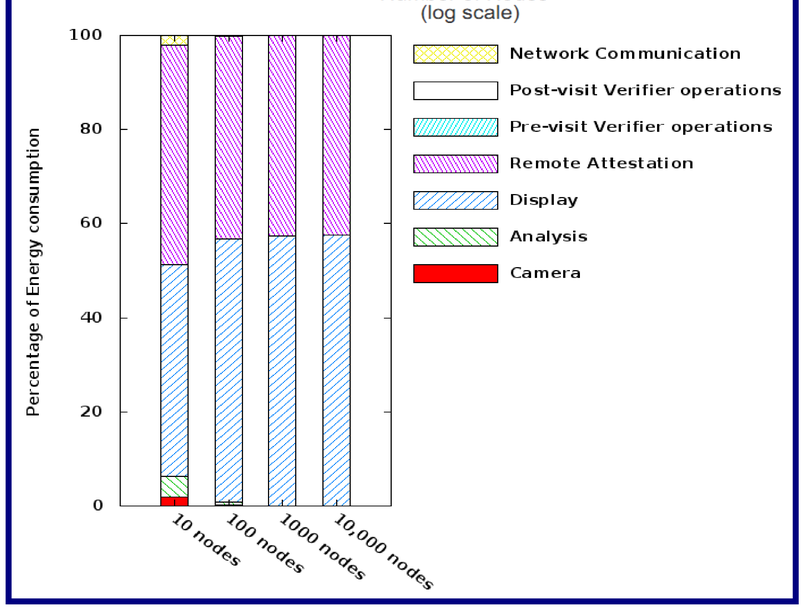


Evaluating costs of attestation

Simulation-based study of timing and energy using the Sarana dynamic network

- Used an Amber-alert application with a number of participating nodes (number of nodes is configurable).
- Measured time with and without attestation: Absolute overhead: ~10 seconds for attestation.
- Measured energy consumption of attestation: Attestation accounts for about 46% of energy consumed.





Comprehensive Verification

- First, establish trust in the installation of a system's software packages (*Root of Trust for Installation*)
- Then, determine the *measurement class* of the system to determine the integrity measurement mechanisms required
- Use the indirect verification to scale verification to large distributed systems
- Techniques like our asynchronous attestation approach enables systems to service 7000+ attestations requests / second

Problem: Mobile Malware Detection

Mobile/embedded malware on the rise

- Such CPS devices store information valuable to attackers
- Vast social impact if devices are compromised
- Network-based detection of malware is insufficient
- Host-based malware detection is essential

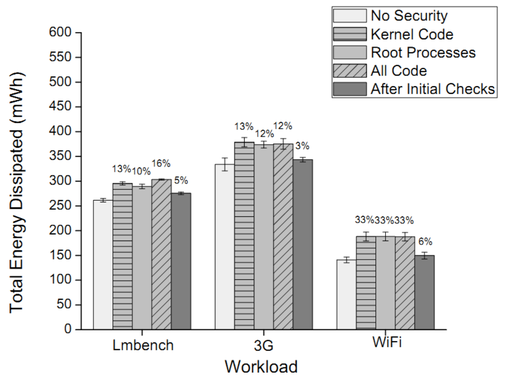
Malware detection costs energy!

- Running malware detector on mobile device consumes energy: Roughly halves battery-life in our experience.
- Can save energy by sacrificing some security? The **security/energy tradeoff.**
- Research question: Can we quantify the security versus energy tradeoff?

The security/energy tradeoff

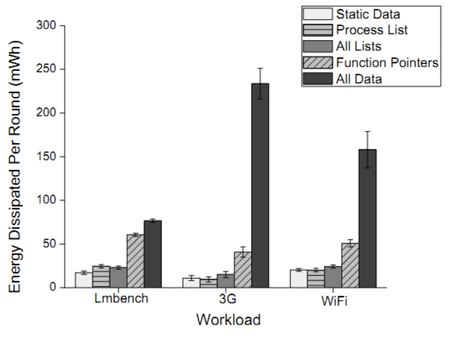
Axes: Attack surface, Frequency of checks

- Studied various configurations of two malware detection tools: One checks code, the other checks data.
- Varied **attack surface**: fraction of code/data checked
- Varied **frequency**: how often are checks executed?



Result of varying code attack surface:

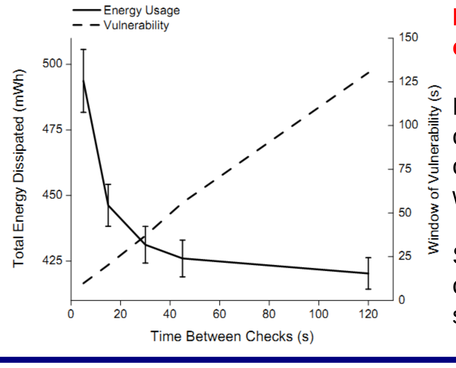
Checking code does not consume much energy, irrespective of attack surface.



Result of varying data attack surface:

Checking all data is a massive energy-drain.

But can protect up to 95% of attack surface for reasonable energy consumption.



Result of varying data check frequency:

Reducing frequency of checks reduces energy consumed, but increases window of vulnerability.

Sweet spot exists at checking interval of ~30 seconds.