SpaceTEE
Secure and Tamper-proof Computing in Space using CubeSats

Yan Michalevsky
Stanford University

Yonatan Winetraub
Stanford University, SpaceIL
We need TEEs

- For confidentiality
  - Need to make sure inner state is invisible
- For computational integrity
  - Use as a trusted authority
  - Verifiable computation is not yet practical
- Applications: Key management, Authentication, Digital Rights Management, Prevent cheating in online games, Trustworthy grid computing (prevent laziness)
Side-channel attacks

• Physical access (contact) to the device [Genkin et al. ’16]

• Key extraction from EM radiation [Genkin et al. ’15]

• Acoustic cryptanalysis [Genkin et al. ’13]
Hardware Security Modules

- Trusted hardware for crypto operations
- Isolated from the rest of the infrastructure
- Different levels of protection including side-channels and physical access
- Expensive: FIPS 140-2 Level 4 device can cost $100,000

Whatever you do - someone will probably break it

So what on earth can we do?
Objects in the outer-space are hard to access physically

However, they can communicate with one another, and with terrestrial base-stations

Anti-satellite weapons are mostly able to destroy (DoS), capturing is extremely difficult and expensive, and likely evident

NORAD: continuous high-res monitoring of objects in space using radars and telescopes

Space is our TEE
CubeSats

• CubeSat 1U

  • Cheap commercial off-the-shelf nano-satellite \(10\text{cm} \times 10\text{cm} \times 10\text{cm}\)

  • Weight: 1.33 kg, Size:

  • 699 CubeSats launched (as of June 24th 2017)

  • Our solution estimated cost (including launch) is $95,000

  • Comparable to a commercial Level 4 HSM
Pre-launch protection

• Make it tamper-evident

• Cover using Kapton tape and measure pre-launch moment of inertia

• Compare to observed once in orbit (by tracking the antennas spin via Doppler effect)

• Tampering with coated hardware without changing it is hard
HSM using SpaceTEE: SpaceHSM

- A system that can serve as a Certificate Authority (CA)
- Builds on traditional PKI and Certificate Transparency
- Certificate transparency logs prevent a powerful attacker who gains access to the communication channel with the satellite from obtaining forged certificates
1. SpaceHSM: Root-of-Trust (R) and crypto-accumulator

2. Ground-station (GS): delegates cert. sign. requests

3. Public, append-only certificate log (L)

4. Verifier (V): checks certificate validity
Threat Model

• SpaceHSM is tamper-proof. No physical access.

• Adversary:
  • Can physically access to all terrestrial infrastructure
  • Can occasionally access the communication channel with the SpaceHSM, bypassing the security of the ground-station
  • Full read-access to code and data of the SpaceHSM prior to launch
  • Spoofing SpaceHSM transmission is impossible
Protocol

• **SpaceHSM Bootstrap**: Generate key-pair and broadcast the public key.

• **Certificate Request**: Requests are transmitted to the SpaceHSM

  • The SpaceHSM transmits back a signed certificate, updates its accumulator and broadcasts it.

• **Cert. Log Update**: Every signed certificate is expected to be submitted to the certificate log. The log-server verifies the certificate and appends it.
DoS Attacks

• How about an adversary that attempts to mount a denial-of-service attack?
  • Submit a certificate signing request to SpaceHSM
  • Withhold the signed certificate from the log
  • SpaceHSM can broadcast the signed certificate instead of sending it only to the ground-station via the private channel
  • Additional measures based on key-rotation are in the paper
CubeSat Design

Can accommodate 2 independent SpaceHSMs with shared power supply

<table>
<thead>
<tr>
<th>Component</th>
<th>Model Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>1U CubeSat</td>
<td>ARM Cortex-M3</td>
</tr>
<tr>
<td>OBC</td>
<td>Cube Computer [6]</td>
<td>10 Whr battery</td>
</tr>
<tr>
<td>Power system</td>
<td>CS 1U Bundle A</td>
<td>On all 6 sides</td>
</tr>
<tr>
<td>Transceiver</td>
<td>ISIS VHF/UHF [17]</td>
<td></td>
</tr>
<tr>
<td>Antennas</td>
<td>ISIS dipole</td>
<td></td>
</tr>
</tbody>
</table>
CubeSat Design

• Use commercial, tested off-the-shelf parts
• Launch onto a Sun Synchronous Orbit (SSO)
• Cube computer:
  • 32-bit ARM Cortex-M3, 4MB Flash memory for code
  • 2 x 1MB SRAM memory for data
• Connected to 48MHz clock
Power Budget

• 45 min. “daylight”, 45 min. “nighttime”

• Average SpaceHSM power consumption: ~1.7 Watt

• 6 solar panels can generate ~3.8 Watt

• Single power supply can accommodate 2 SpaceHSMs
Link Budget

- Currently expect an uplink of 1200bps

- Given an average certificate size of 2.5KB, we can service one certificate signing request every 20 seconds

- For example, using the Svalbard ground-station, can communicate for 10 min. every 90 min.

- Rates can be improved by 1000x using S-Band transceivers supporting 2Mbps uplink
  - Enables handling 50 cert. req. / sec.
Software

• Requirements:
  • Symmetric encryption: AES-128, AES-256
  • PKI: RSA-2048, RSA-4096, ECDSA sigs., Curve25519
  • Small crypto library

• Alternatives:
  • TweetNaCl: 11 KB, 40 RSA-2048 dec./sec. on ARM Cortex M3,
  • mbedTLS: 125 KB (still within RAM budget)
  • SharkSSL: 15 KB (benchmarked on ARM Cortex M3)
  • WolfSSL: small, open-source, supports TLS 1.3, ChaCha20, Curve25519

• Protect against faults (EDAC hardware, verify signatures)
More applications

- Trusted public parameters generator for crypto schemes
- Trusted party for cryptographic protocols
- Trusted time-stamping
- Trusted mining of cryptocurrencies (transaction verification)
Thank you. Questions please.