**Privacy-from-Birth: Protecting Sensed Data from Malicious Sensors with VERSA**

Ivan De Oliveira Nunes¹, Seeyeon Hwang², Sashidhar Jakkamsetti² and Gene Tsudik²

¹Rochester Institute of Technology and ²University of California, Irvine

---

**Low-end IoT Devices**
- Low cost, low power, and small size
- Single core, 8/16-bit CPU, <48 MHz and < 64 KB (RAM + FLASH)
- No OS, MMU, MPU, TEE,
- Bare metal (e.g., TI MSP430)

**Problem Statement**
- Sensors often collect sensitive information
- Resource constrained; lack security features
- Attractive targets for attacks (e.g., Mirai)

**Existing Techniques**
- Just encrypting sensed data does not help
- Compromised software (malware) can still extract sensed data!

*Integrity-ensuring techniques (e.g. remote attestation) can detect, but cannot prevent leakage*

---

**Privacy-from-Birth**
- “All traces of sensed data must be protected from birth until it leaves the device” whenever data becomes digital
- PFB Goals:
  - Prevent access to GPIO memory except for authorized software
  - Provide a secure execution platform for authorized software to process sensed data

---


1. **Authorize(S) → ATokk**: Send (S, ATokk)
2. **Verify(S, ATokk)**: Encrypted results (if any)
3. **AtomicExecute(S)**

- Access Control to Sensed Data
- Atomic Execution of the Authorized Software S
- Data Erasure at Reboot/Reset

---

**Implementation/Evaluation**
- VERSA is implemented on OpenMSP430
- Synthesized and deployed on Basy3 FPGA
- 13% hardware overhead over OpenMSP430
- 10% hardware overhead over VRASED
- O(n) runtime for request verification

**Takeaways**
- PFB guarantees end-to-end privacy assurance for sensor data in low-end MCUs
- VERSA provably realizes PFB with a minimal formally verified hardware (open-sourced)
- 13% hardware overhead and linear runtime for verification