Secure Firmwares for Internet of Things

— Credential Data Protection and Bug/Flaw Detection on Embedded Systems

PennState College of Information Sciences and Technology

Wenhui Zhang, Jun Xu, Lannan Luo, Peng Liu

Pennsylvania State University

System Model

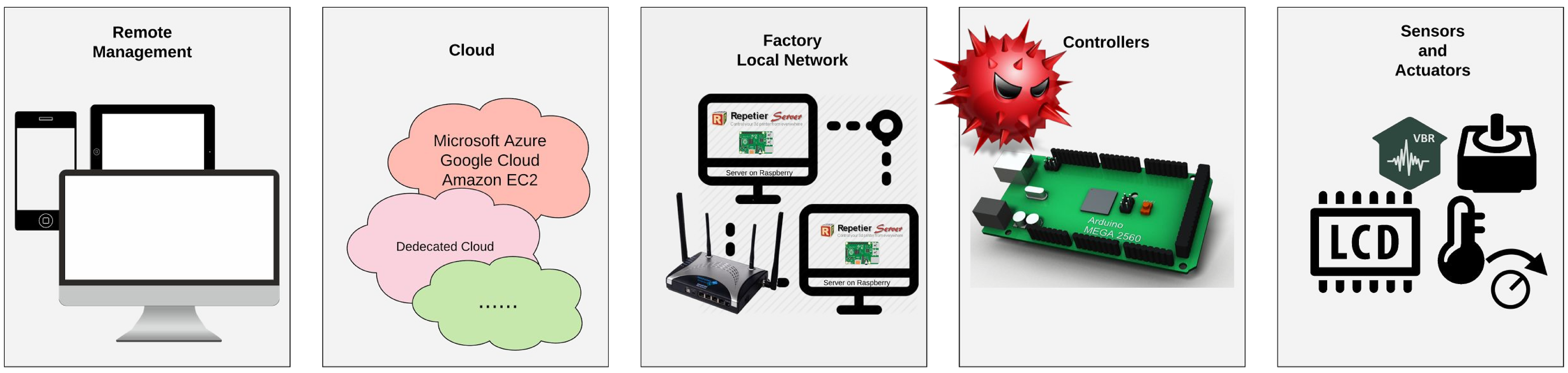

Figure 1: Testbed ---Security Links for 3D Additive Manufacturing Process Chain

Problem Statement

Goal is to protect sensitive data and resources from **JIT-ROP** attacks on Firmwares.

Vulnerable Architectures:

Don't Have:

ASLR

MMU

Have :

WatchDog

Bootloader and Firmware Separation

Preliminary Result

Adding Barriers:

```
// add asm as tags for pattern recognition
asm volatile("" ::: "memory");
    serial_char = MYSERIAL.read();
asm volatile("" ::: "memory");
```

avr-objdump get Pattern

```
87ae:   80 91 6d 0f   lds    r24, 0x0F6D    ;
87b2:   90 91 6e 0f   lds    r25, 0x0F6E    ;
87b6:   20 91 6b 0f   lds    r18, 0x0F6B    ;
87ba:   30 91 6c 0f   lds    r19, 0x0F6C    ;
87be:   28 17         cp     r18, r24
87c0:   39 07         cpc    r19, r25
87c2:   69 f0         breq   .+26           ;

87c4:   fc 01         movw   r30, r24;
87c6:   e5 51         subi   r30, 0x15      ; 21
87c8:   f1 4f         sbci   r31, 0xF1      ; 241
87ca:   20 81         ld     r18, Z
87cc:   01 96         adiw   r24, 0x01      ; 1
87ce:   8f 77         andi   r24, 0x7F      ; 127
87d0:   99 27         eor    r25, r25
87d2:   90 93 6e 0f   sts    0x0F6E, r25    ;
87d6:   80 93 6d 0f   sts    0x0F6D, r24    ;
87da:   82 2f         mov    r24, r18
87dc:   02 c0         rjmp   .+4            ;  0x87e2
87de:   8f ef         ldi    r24, 0xFF      ; 255
87e0:   9f ef         ldi    r25, 0xFF      ; 255
87e2:   80 93 e8 04   sts    0x04E8, r24    ;
```

Future Work

- Symbolic execution: limited to vulnerabilities they are using to scan with, and not applicable with vulnerabilities introduced in the runtime environment like JIT-ROP [3].
- ASLR : MPU is widely adopted in time-sensitive embedded systems, which does not support ASLR [4].
- Full memory encryption: takes significant overhead as a security protection trade off [5].
- Control flow integration: does not correct fault routes into valid ones [6].

Goal :

correct Vulnerable Instructions of JIT-ROP:

Read Specifically a part of FLASH, EEPROM, or SRAM where we store credential data (G-Code).

- Read/write EEPROM (and extract cryptography keys)
- Read parts of flash (e.g., reading locked bootloader section)
- Staying persistent (writing flash)

Bibliography

1. Wahbe, R., Lucco, S., Anderson, T. E., & Graham, S. L. (1994, January). Efficient software-based fault isolation. In ACM SIGOPS Operating Systems Review (Vol. 27, No. 5, pp. 203-216). ACM.
2. Torrance, R., & James, D. (2009). The state-of-the-art in IC reverse engineering. In Cryptographic Hardware and Embedded Systems-CHES 2009 (pp. 363-381). Springer Berlin Heidelberg.
3. Davidson, Drew, et al. "FIE on firmware: Finding vulnerabilities in embedded systems using symbolic execution." Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13). 2013.
4. Braden, K., Crane, S., Davi, L., Franz, M., Larsen, P., Liebchen, C., & Sadeghi, A. R. (2016, February). Leakage-resilient layout randomization for mobile devices. In Network and Distributed Systems Security Symposium (NDSS).
5. Würstlein, Alexander, et al. "Exzess: Hardware-Based RAM Encryption Against Physical Memory Disclosure." International Conference on Architecture of Computing Systems. Springer International Publishing, 2016.
6. Snow, K. Z., Monrose, F., Davi, L., Dmitrienko, A., Liebchen, C., & Sadeghi, A. R. (2013, May). Just-in-time code reuse: On the effectiveness of fine-grained address space layout randomization. In Security and Privacy (SP), 2013 IEEE Symposium on (pp. 574-588). IEEE.