

Embedded Systems Security

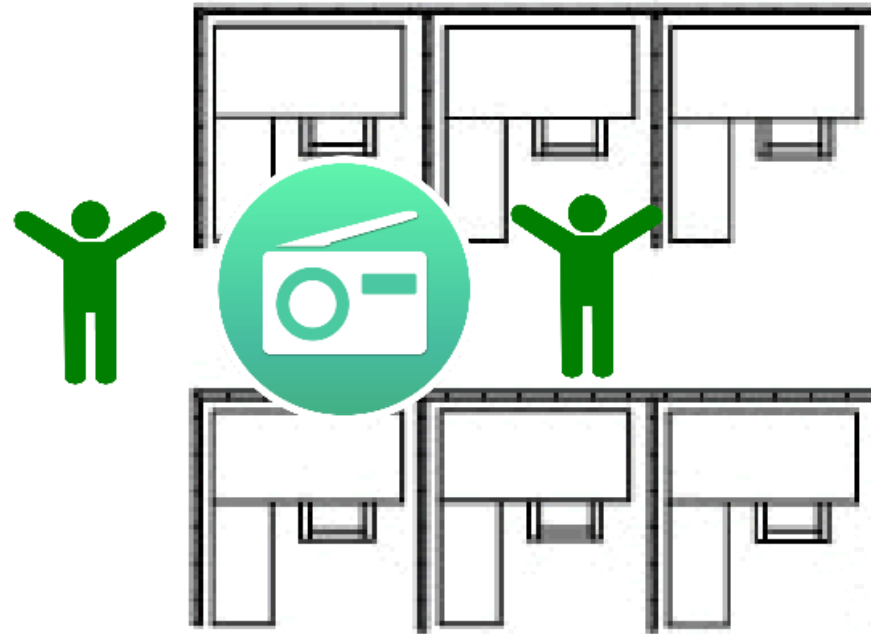
Jem Berkes

ECE, University of Manitoba

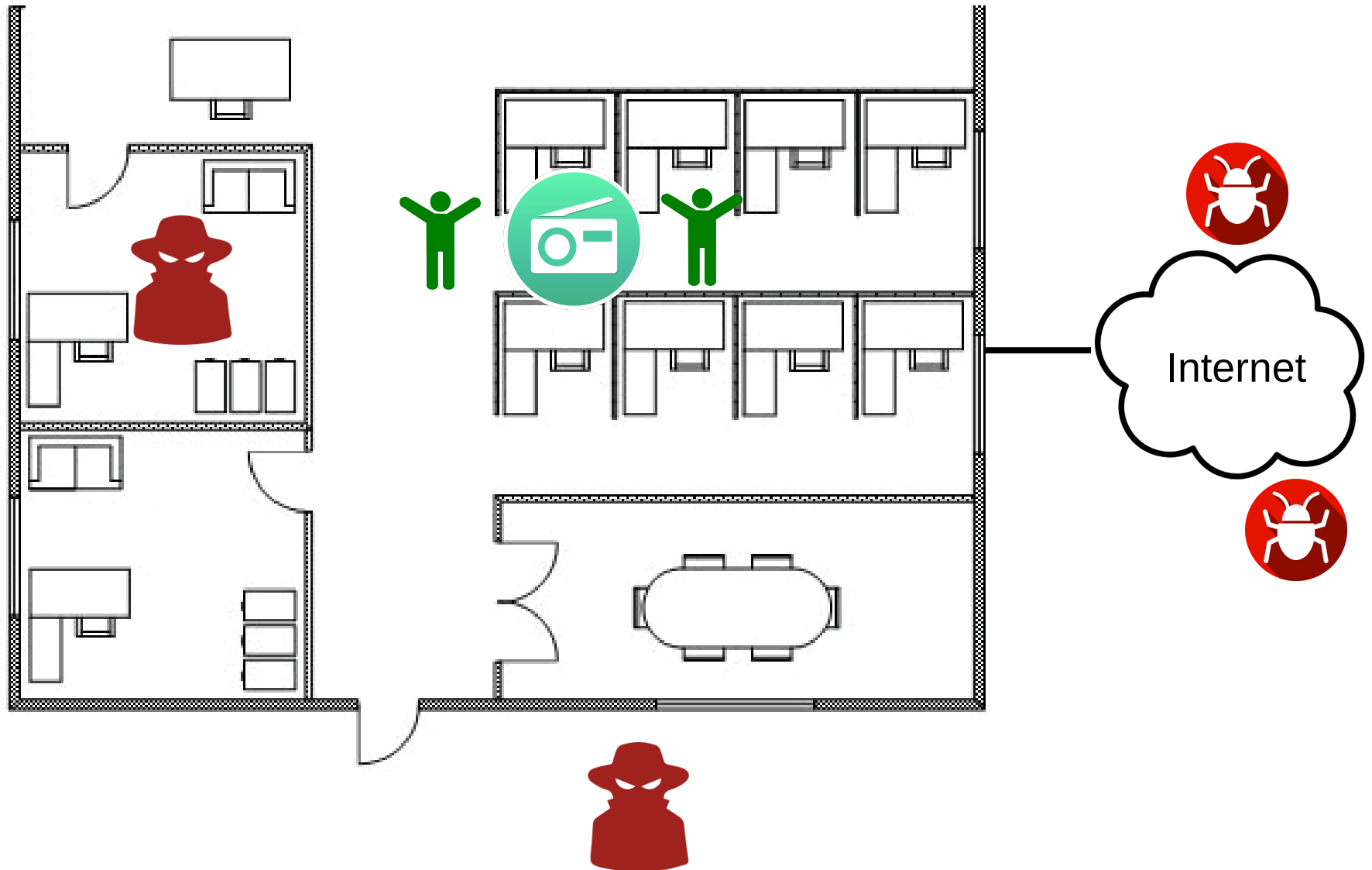
Basics

- Embedded/IoT are feature-rich computers
- Many kinds of attacks are possible
 - Automated worms & viruses
 - Curious or malicious people

What we see



There's more around you!



Types of Attacks

Eavesdropping

- Someone intercepts or “sniffs” data packets
- Can expose or steal sensitive data

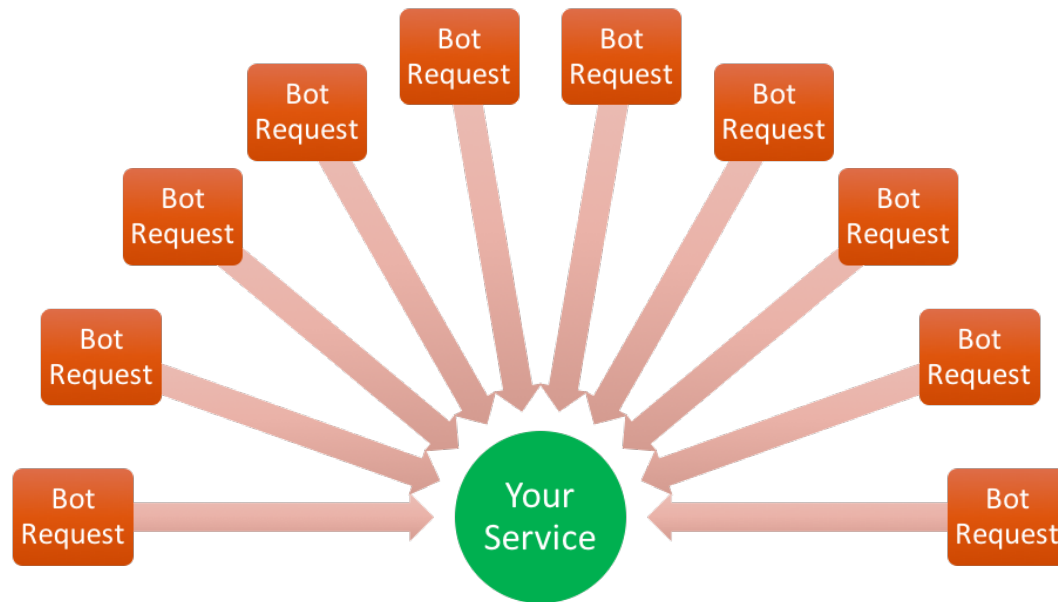


Solution:

Use WPA2 or WPA3 wireless security,
Use SSL/TLS to encrypt traffic

Denial of Services (DoS)

- Someone floods your devices with requests
- Tries to slow down or disable the service

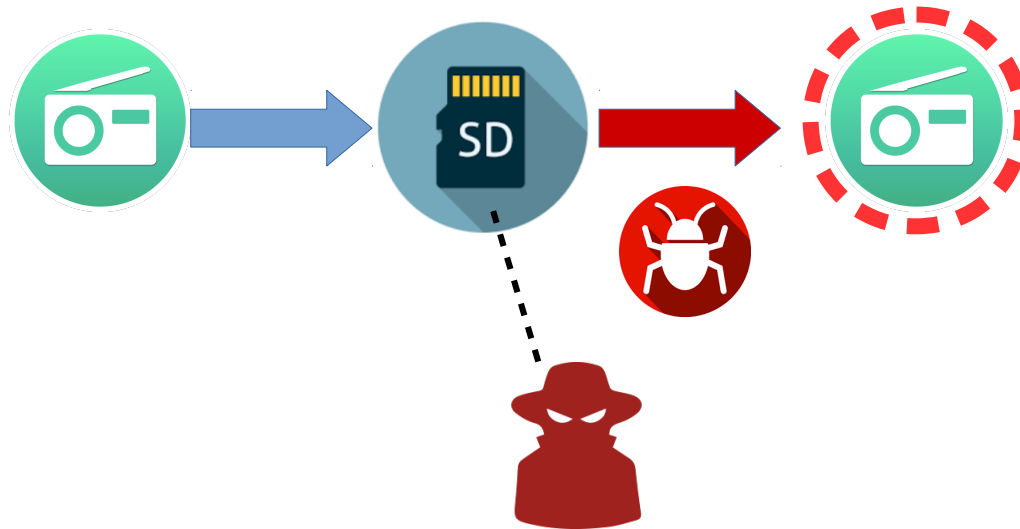


Solution:

Implement rate limiting, or auto-ban malfunctioning clients

Device Tampering

- Someone accesses the disk and reads the files
- Or modifies the embedded software



See “Industrial Grade Concerns”

Compromise or Hack

- The device is infiltrated
- Someone (or software) takes control

```
root@host$ ls /
bin    dev    initrd.img    lib64    mnt    root    snap    tmp    vmlinuz
boot  etc    initrd.img.old  lost+found  opt    run    srv    usr    vmlinuz.old
cdrom  home  lib           media    proc   sbin   sys    var
root@host$
```

Common Vulnerabilities

(Frequently occur!)

Common vulnerability #1

- **Open service ports allowing logins**
 - ssh, telnet, http: login prompt
- *Plus* weak/default passwords

Common vulnerability #1

- **Open service ports allowing logins**
 - ssh, telnet, http: login prompt
- *Plus weak/default passwords*



1. Discovers telnet service

2. Start trying default logins
admin : (no password)
admin : admin
... *brute-force search* ...

3. If success, loads software



Common vulnerability #2

- **Unauthenticated open services**
- Anyone can connect!



See: *“Avast Hacked a Smart Coffee Maker All Kinds of Ways”*

Common vulnerability #2

- **Unauthenticated open services**
- **Anyone can connect!**



See: *"Avast Hacked a Smart Coffee Maker All Kinds of Ways"*

Common vulnerability #3

- **Outdated OS and software**
- Everything needs patching eventually
- Can't just leave a device alone for 5 years
- Design your product to *support* updating

Common vulnerability #4

- **Malicious re-pairing / physical takeover**
- Someone pairs the device (again)
 - But doesn't own the device
- Physical access



Press and hold the doorbell button for 8 seconds until the doorbell light is flashing red.

Next

See: "\$30 doorbell cameras can be easily hijacked, says Consumer Reports"

Wireless Security

Wi-Fi Modes

- **Open**: no password, anyone can connect, unsafe
- **WEP**: old standard, broken, unsafe
- **WPA**: old standard, broken, unsafe
- **WPA2-TKIP**: uses old algorithm, unsafe
- **WPA2-AES**: next best option to WPA3
- **WPA3**: the newest standard, best option

Wi-Fi Security Tips

- Use WPA2 (AES) or WPA3
 - With a good password
 - Traffic will be encrypted
- Disable the “WPS” feature

SSID

- SSID (Service Set Identifier) is hotspot name
- Publicly broadcast and visible to all
- Assume SSID is visible to everyone
- Hiding SSID doesn't enhance security

Wi-Fi Can Be Risky

- “KRACK” was a very severe WPA2 attack from 2017-2018
- Some embedded/IoT devices with old firmware
- What can the attacker do?
 - Intercept wireless traffic, without a password
 - Inject packets and manipulate connections
- HTTPS (aka TLS) helps protect against this



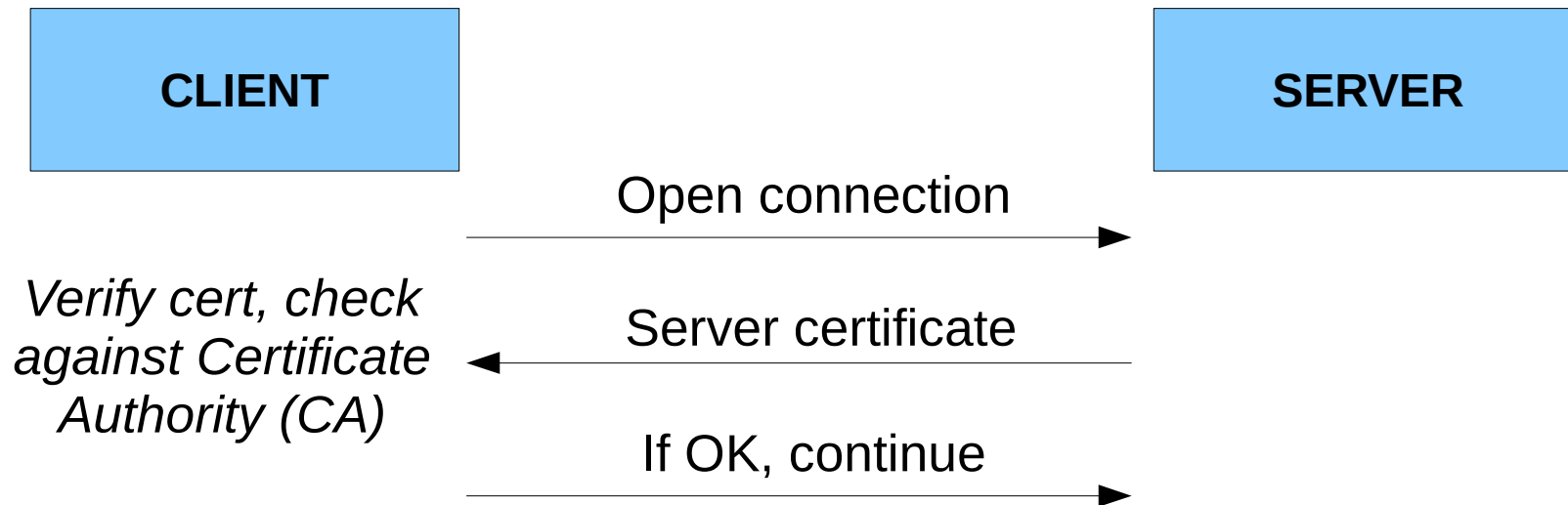
Ideally, use SSL/TLS!

HTTPS (aka TLS) – simplified

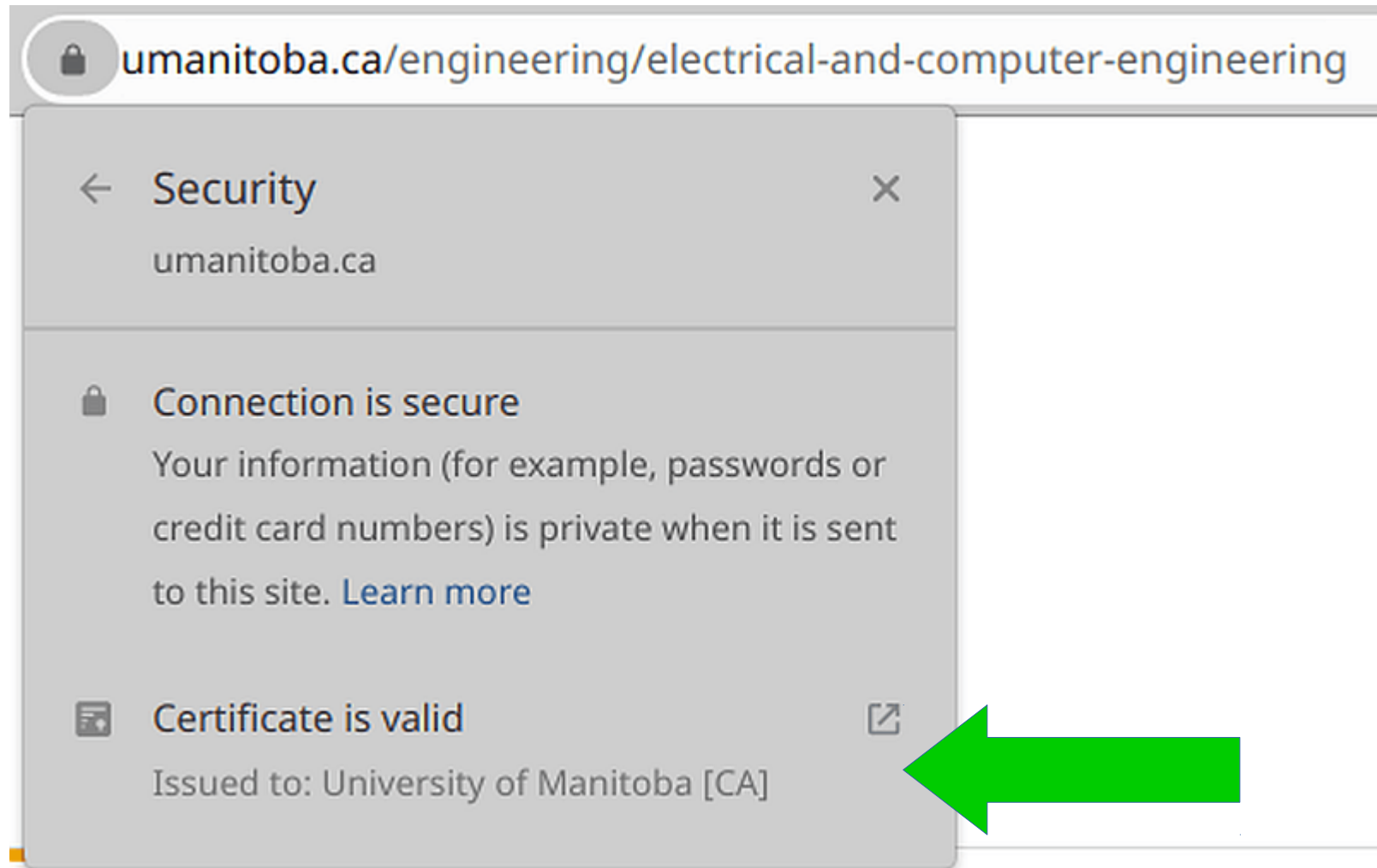
- Client makes an encrypted HTTP request to Server
- Server provide a Certificate
- Client verifies that Certificate is valid
 - Makes sure we're talking to the real Sever

HTTPS (aka TLS) – simplified

- Client makes an encrypted HTTP request to Server
- Server provide a Certificate
- Client verifies that Certificate is valid
 - Makes sure we're talking to the real Sever



Certificates



umanitoba.ca/engineering/electrical-and-computer-engineering

← Security ×
umanitoba.ca

🔒 Connection is secure
Your information (for example, passwords or credit card numbers) is private when it is sent to this site. [Learn more](#)

📄 Certificate is valid ⓘ
Issued to: University of Manitoba [CA]

A large green arrow points to the 'Certificate is valid' section.

Certificates

Certificate Viewer: www.umanitoba.ca ✕

General Details

Issued To

Common Name (CN)	www.umanitoba.ca
Organisation (O)	University of Manitoba
Organisational Unit (OU)	<Not part of certificate>

Issued By

Common Name (CN)	GlobalSign Extended Validation CA - SHA256 - G3
Organisation (O)	GlobalSign nv-sa
Organisational Unit (OU)	<Not part of certificate>

Validity Period

Issued On	Thursday, 11 June 2020 at 14:06:02
Expires On	Saturday, 23 July 2022 at 08:41:09

Certificates

Certificate Viewer: www.umanitoba.ca ✕

General Details

Issued To

Common Name (CN)	www.umanitoba.ca
Organisation (O)	University of Manitoba
Organisational Unit (OU)	<Not part of certificate>

Issued By

Common Name (CN)	GlobalSign Extended Validation CA - SHA256 - G3
Organisation (O)	GlobalSign nv-sa
Organisational Unit (OU)	<Not part of certificate>

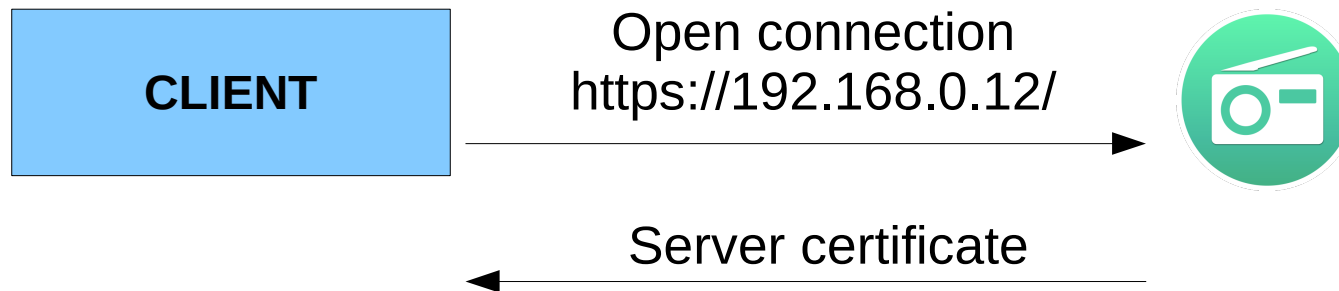
Certificate Authority

Validity Period

Issued On	Thursday, 11 June 2020 at 14:06:02
Expires On	Saturday, 23 July 2022 at 08:41:09

Certificate Authorities (CA)

- This becomes an issue with embedded systems
- Say your embedded device runs a web server



Problems!

- The server address is some IP address
- There's no "domain name"
- Can't confirm certificate validity

Private CA method

- Create your own Certificate Authority (using OpenSSL)
- Install your own “root” CA cert on all clients
- Also called a Private CA
- Each of *your* devices can then recognize each other
 - But someone else (e.g. smart phone) will still get an “invalid certificate”

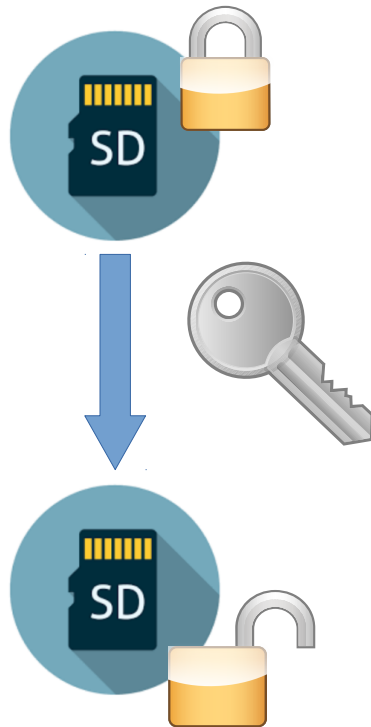
“Industrial Grade” Concerns

Physical Tampering

- People have physical access
- They could break open the device
 - Remove SD card
 - Connect to disk interface
- Don't want people tampering with your embedded sys

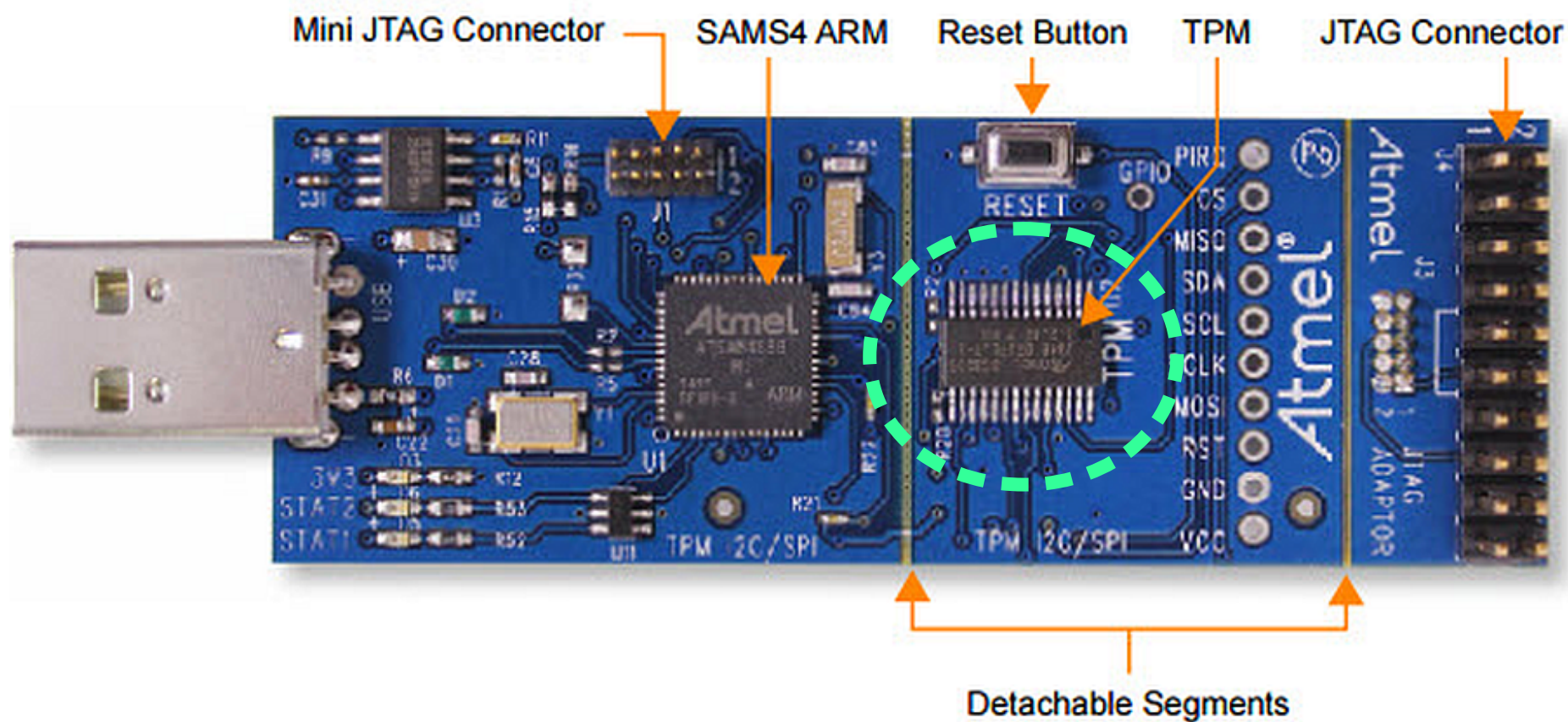
Protecting System Integrity

- Generally requires more feature-rich processors
- Use full disk encryption (FDE)
- “Encrypted at rest”



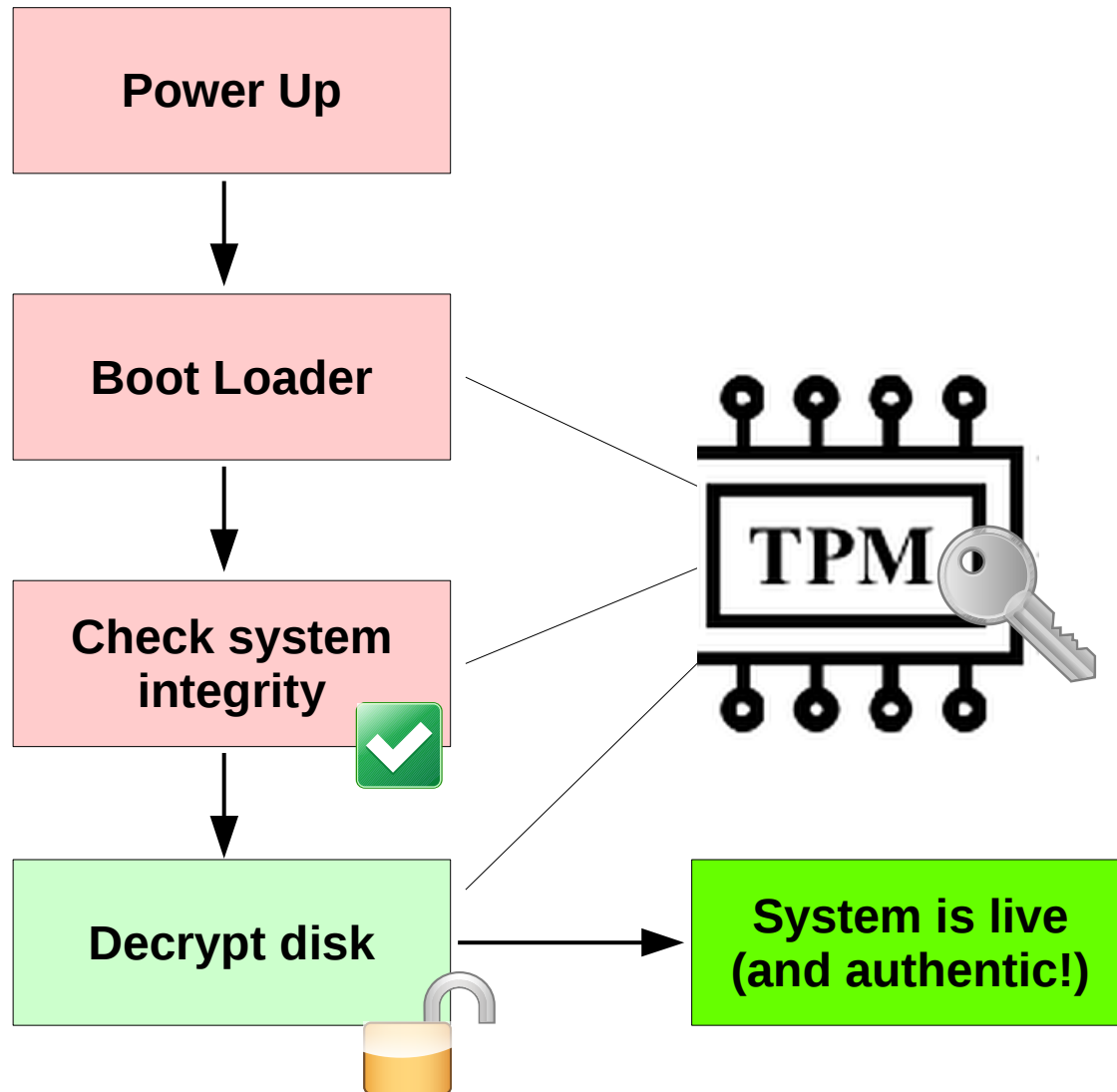
But the key is exposed and readable, right?

Secure Cryptoprocessor (e.g. TPM)

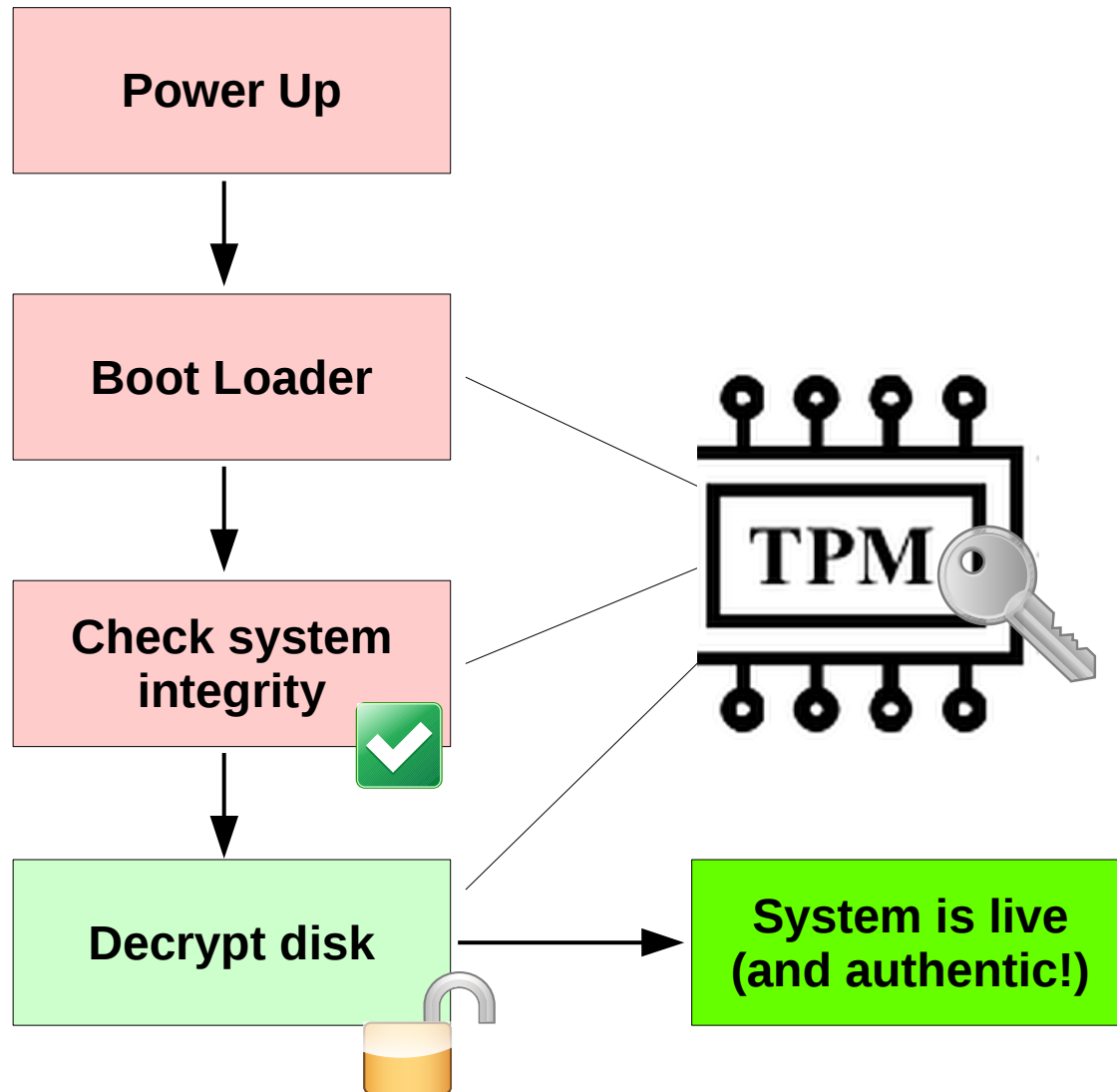


Atmel TPM Development Kit (ARM)

Secure Boot & Cryptoprocessor



Secure Boot & Cryptoprocessor



- Detects tampering
- Ensures integrity
- Protects the key
- “Root of Trust”