

Securing IoT with the ARM mbed ecosystem

ARM

Xiao Sun / Senior Applications Engineer / ARM

ARM mbed Connect / Shenzhen, China
December 5, 2016

©ARM 2016

Lots of interest in IoT security

- Researchers are looking into security of IoT systems
- Vulnerabilities are recognized in deployed IoT systems
- Fixes are deployed where possible
- IoT security is evolving in a positive way as a consequence

You can't do big data unless you trust the little data

IoT will not
scale without
trust and
security

With large
deployments
you must
secure all
devices

Even simple sensors

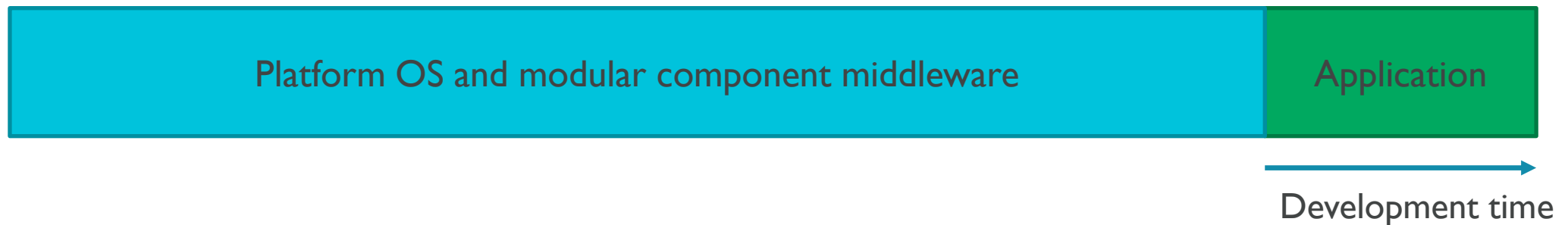
Enabling trust and
security in IoT
devices is an
opportunity to
create value

IoT projects need a platform OS

- Historically, embedded microcontroller design has had little code or design commonality between systems that enables widespread re-use

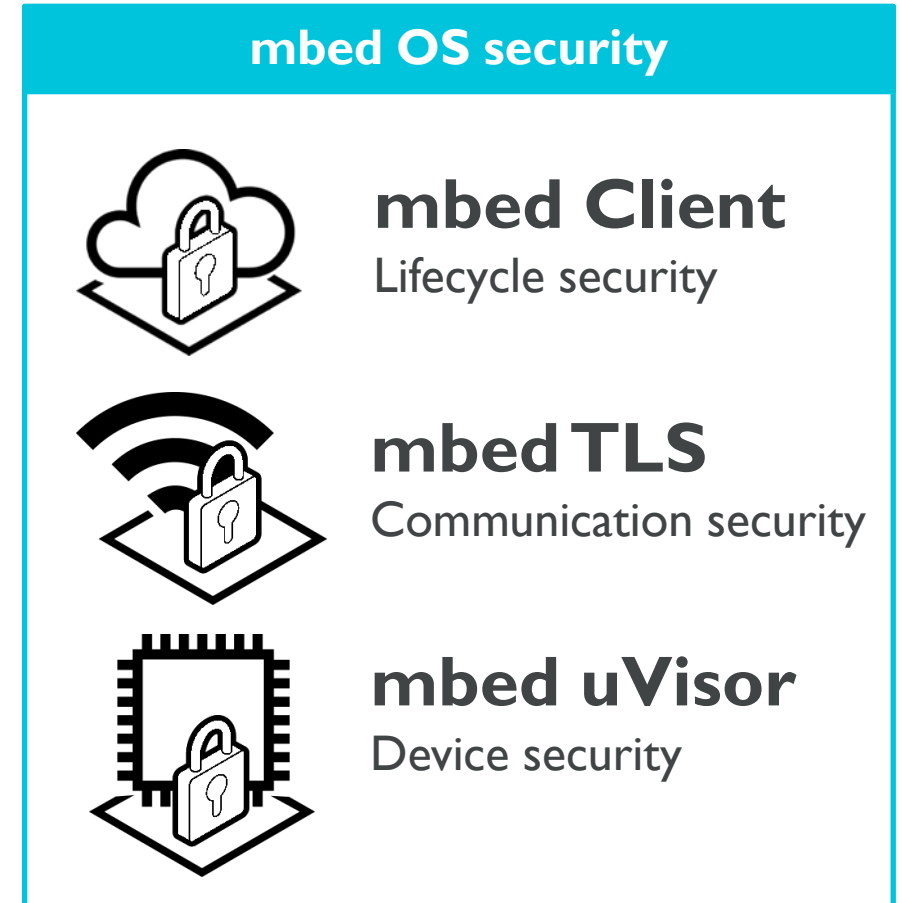


- The communication, device management and security demands of IoT devices are a disruptive jump in complexity that drives the need to use a platform OS



mbed OS security

- Covers three main types of threat
- Security of system, including ability to provision, manage and update devices (e.g. security fix)
- Security of communications between device and cloud services
- Security and integrity of device itself from untrusted or malicious code



Proportional security

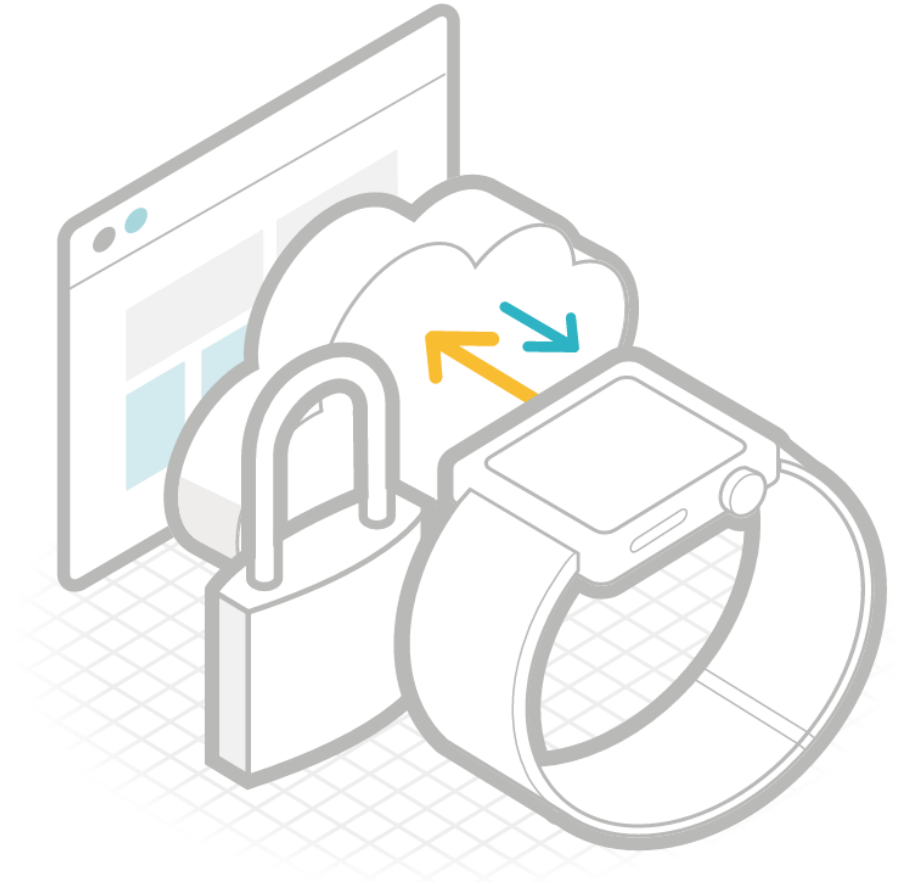
- Threat-models should be informed by business requirements
- Technology applied and cost expended varies according to application needs
- For Example
 - Risk environment of application
 - Value of assets to be protected
 - Trust and control over firmware
 - Supply chain structure
 - Lifetime of the device

Application	Security
Disposable	mbed TLS + mbed Connect
Long life node	+ mbed uVisor + active lifecycle management
Critical infrastructure	+ Anti-tamper hardware (ARM SecurCore)

mbed TLS

mbed TLS

- mbed TLS enables cryptographic and SSL/TLS capabilities for use in embedded software
- mbed TLS is tightly integrated into mbed OS
- Combined with the mbed uVisor, this provides comprehensive device and communication security for IoT products



mbed TLS – Code quality







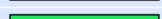

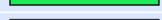
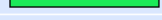
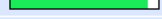




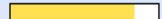
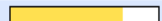






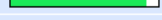
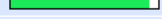
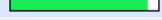
LCOV - code coverage report

Current view: [top level](#) - work/library

Test: mbed TLS ([view descriptions](#))

	Hit	Total	Coverage
Lines:	14532	16015	90.7 %
Functions:	1084	1086	99.8 %

Legend: Rating: low: < 75 % medium: >= 75 % high: >= 90 %

Filename	Line Coverage	Functions
aes.c	 100.0 % 356 / 356	100.0 % 14 / 14
arc4.c	 100.0 % 54 / 54	100.0 % 6 / 6
asn1parse.c	 87.3 % 137 / 157	100.0 % 14 / 14
asn1write.c	 88.7 % 126 / 142	100.0 % 14 / 14
base64.c	 100.0 % 90 / 90	100.0 % 3 / 3
bignum.c	 98.5 % 893 / 907	100.0 % 54 / 54
blowfish.c	 100.0 % 145 / 145	100.0 % 11 / 11
camellia.c	 100.0 % 253 / 253	100.0 % 11 / 11
ccm.c	 100.0 % 117 / 117	100.0 % 8 / 8
cipher.c	 93.1 % 322 / 346	100.0 % 29 / 29
cipher_wrap.c	 95.0 % 134 / 141	100.0 % 49 / 49
ctr_drbg.c	 94.9 % 188 / 198	100.0 % 18 / 18
debug.c	 97.2 % 140 / 144	100.0 % 10 / 10
des.c	 98.9 % 278 / 281	100.0 % 22 / 22
dhm.c	 84.5 % 169 / 200	100.0 % 15 / 15
ecdh.c	 76.2 % 61 / 80	100.0 % 10 / 10
ecdsa.c	 87.9 % 124 / 141	100.0 % 11 / 11
ecp.c	 94.6 % 634 / 670	100.0 % 50 / 50
ecp_curves.c	 95.6 % 219 / 229	100.0 % 20 / 20
entropy.c	 98.4 % 125 / 127	100.0 % 13 / 13
entropy_poll.c	 88.9 % 16 / 18	100.0 % 2 / 2
error.c	 91.7 % 22 / 24	100.0 % 1 / 1
gcm.c	 93.8 % 243 / 259	100.0 % 12 / 12
hmac_drbg.c	 93.1 % 149 / 160	100.0 % 16 / 16
md.c	 100.0 % 149 / 149	100.0 % 21 / 21
md5.c	 97.7 % 172 / 176	100.0 % 10 / 10

mbed TLS – Code testing

- Protocol interoperability tests

```
$ ./compat.sh
P->G ssl3,no TLS-RSA-WITH-AES-128-CBC-SHA ..... PASS
P->G ssl3,no TLS-RSA-WITH-CAMELLIA-128-CBC-SHA ..... PASS
P->G ssl3,no TLS-RSA-WITH-3DES-EDE-CBC-SHA ..... PASS
P->G ssl3,no TLS-RSA-WITH-RC4-128-SHA ..... PASS
P->G ssl3,no TLS-RSA-WITH-RC4-128-MD5 ..... PASS
G->P ssl3,no +DHE-RSA:+AES-128-CBC:+SHA1 ..... PASS
G->P ssl3,no +DHE-RSA:+AES-256-CBC:+SHA1 ..... PASS
G->P ssl3,no +DHE-RSA:+CAMELLIA-128-CBC:+SHA1 ..... PASS
G->P ssl3,no +DHE-RSA:+CAMELLIA-256-CBC:+SHA1 ..... PASS
G->P ssl3,no +DHE-RSA:+3DES-CBC:+SHA1 ..... PASS
G->P ssl3,no +RSA:+AES-256-CBC:+SHA1 ..... PASS
```

- Behavioural RFC tests

```
$ ./ssl-opt.sh
Fallback SCSV: default ..... PASS
Fallback SCSV: explicitly disabled ..... PASS
Fallback SCSV: enabled ..... PASS
Fallback SCSV: enabled, max version ..... PASS
Fallback SCSV: default, openssl server ..... PASS
Fallback SCSV: enabled, openssl server ..... PASS
Fallback SCSV: disabled, openssl client ..... PASS
Fallback SCSV: enabled, openssl client ..... PASS
Fallback SCSV: enabled, max version, openssl client ..... PASS
```

- Vulnerability tracking and fixes

Known vulnerabilities

CVE stands for Common Vulnerability and Exposures. A CVE Identifier is a unique number that can be used over different security advisories by different vendors to refer to the same issue. The following CVE identifiers are known to involve mbed TLS and PolarSSL:

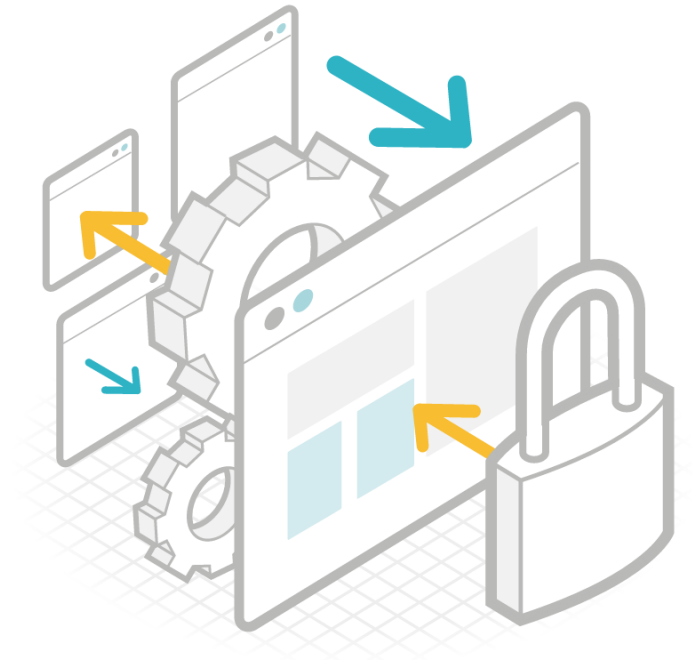
mbed TLS / PolarSSL Advisory	CVE Identifier	Issue title	Fixed in
2011-01	CVE-2011-1923	Possible man in the middle in Diffie Hellman key exchange	0.14.2, 1.0.0
2011-02	CVE-2011-4574	Weak random number generation within virtualized environments	1.1.0
2012-01	CVE-2012-2130	Weak Diffie-Hellman and RSA key generation	1.1.2
2013-01	CVE-2013-0169	Lucky thirteen - timing side channel during decryption	1.1.6, 1.2.6
	CVE-2013-1621	Denial of Service in SSL Module	1.2.5
2013-02	Unknown	RC4 ciphersuites in SSL and TLS vulnerable	Not solvable
	CVE-2013-1622	False warning, not an issue in a numbered release.	
2013-03	CVE-2013-4623	Denial of Service through Certificate message during handshake	1.1.7, 1.2.8
2013-04	CVE-2013-5914	Buffer overflow in ssl_read_record()	1.1.8, 1.2.9, 1.3.0
2013-05	CVE-2013-5915	Timing Attack against protected RSA-CRT implementation used in PolarSSL	1.2.9, 1.3.0
2014-01	CVE-2014-0160	Heartbleed Bug	Not affected
2014-02	CVE-2014-4911	Denial of Service against GCM-enabled entities	1.2.11, 1.3.8
2014-03	CVE-2014-3566	POODLE attack on SSLV3	Not affected
2014-04	CVE-2015-1182	Remote attack using crafted certificates	1.2.13, 1.3.10
2015-01	CVE-2015-5291	Remote attack on clients using session tickets or SNI	1.2.17, 1.3.14, 2.1.2

mbed uVisor

(pronounced “embed microVisor”)

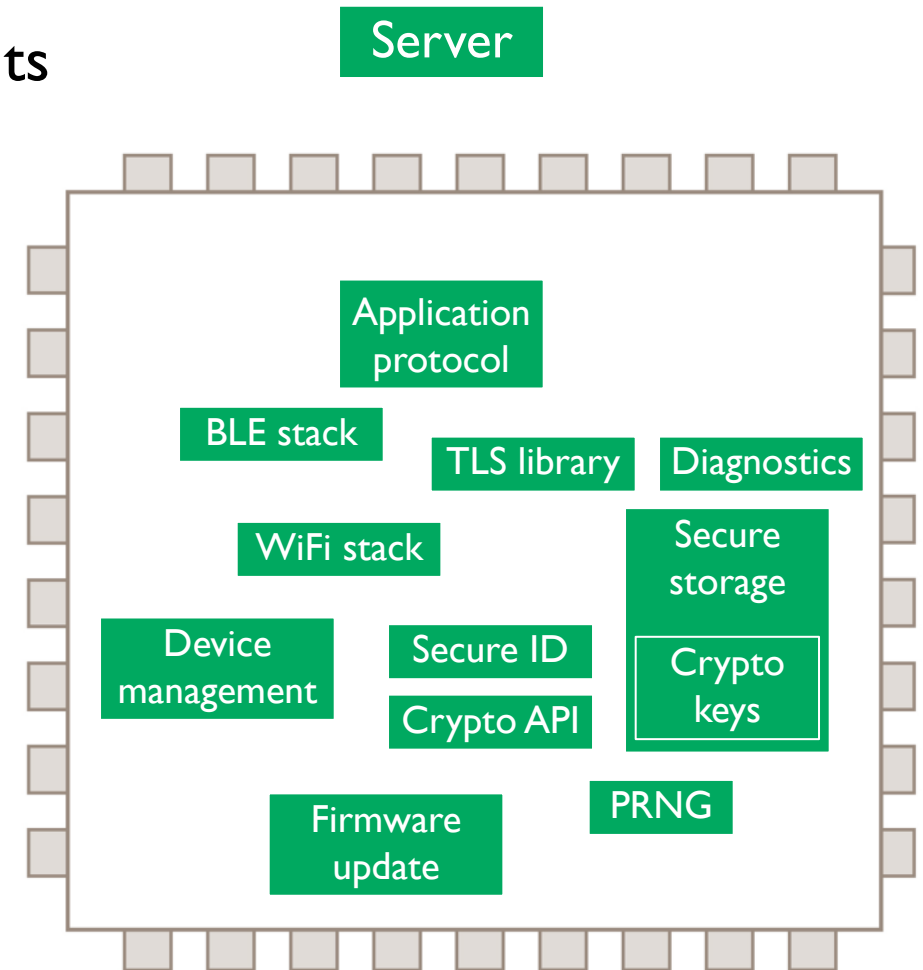
mbed uVisor

- A tiny, hypervisor/microkernel-like security kernel
- Creates and enforces secure isolation boundaries within the OS, between different parts of the system
- Enables secrets to be strongly protected against software and network-bourn attackers
- Efficient hardware enforcement through the memory protection unit (MPU) and ARM TrustZone for v8-M



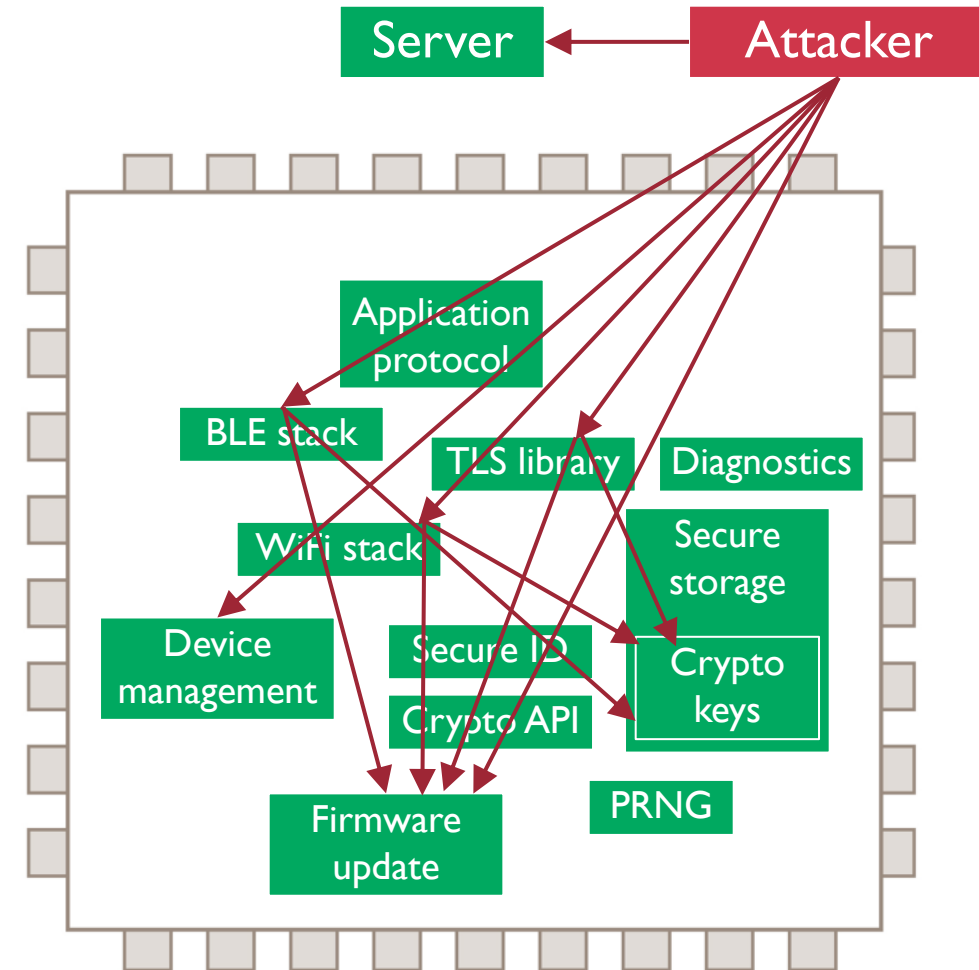
The device security problem

- Even simple IoT products have complex components
 - Secure server communication over complex protocols
 - Secure firmware updates over the air
 - Secure device identities
 - Cryptography APIs and random number generation
- Existing IoT solutions use flat address spaces with little privilege separation
 - Especially on microcontrollers



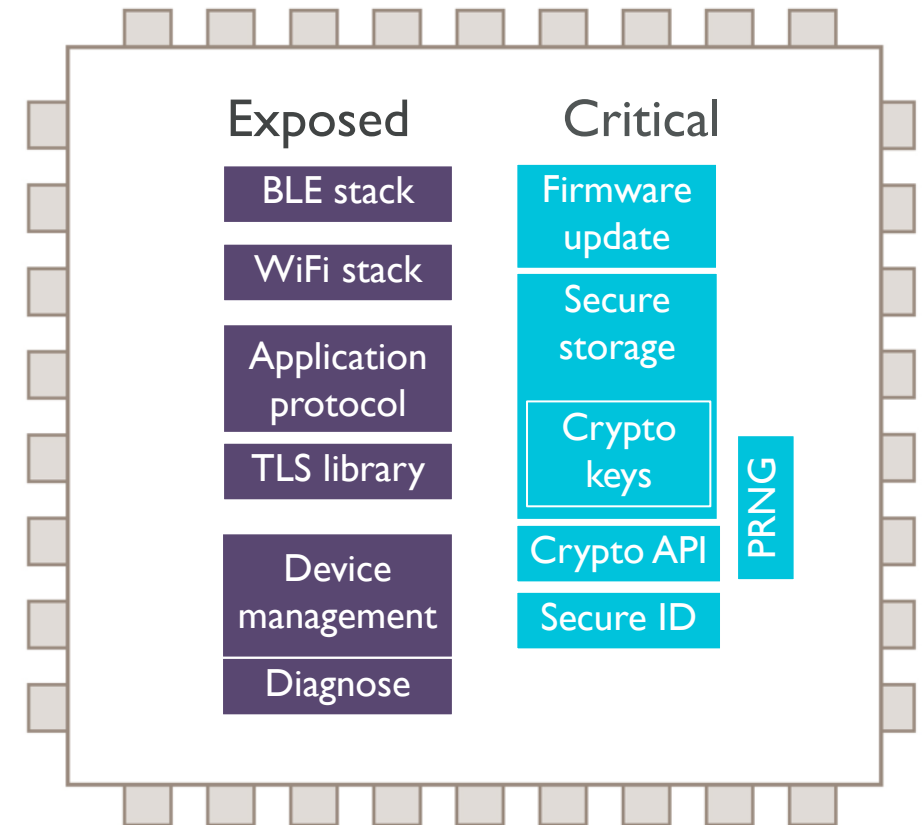
The device security problem - Attacker view

- Flat security models allow attackers to break device security by breaking any system component
- Common attack entry points:
 - Complex protocols like TLS, Wi-Fi or USB device configuration
 - Firmware update functions (USB, network, CAN...)
- Impossible to recover from attacks as firmware update functions can be compromised by the attacker



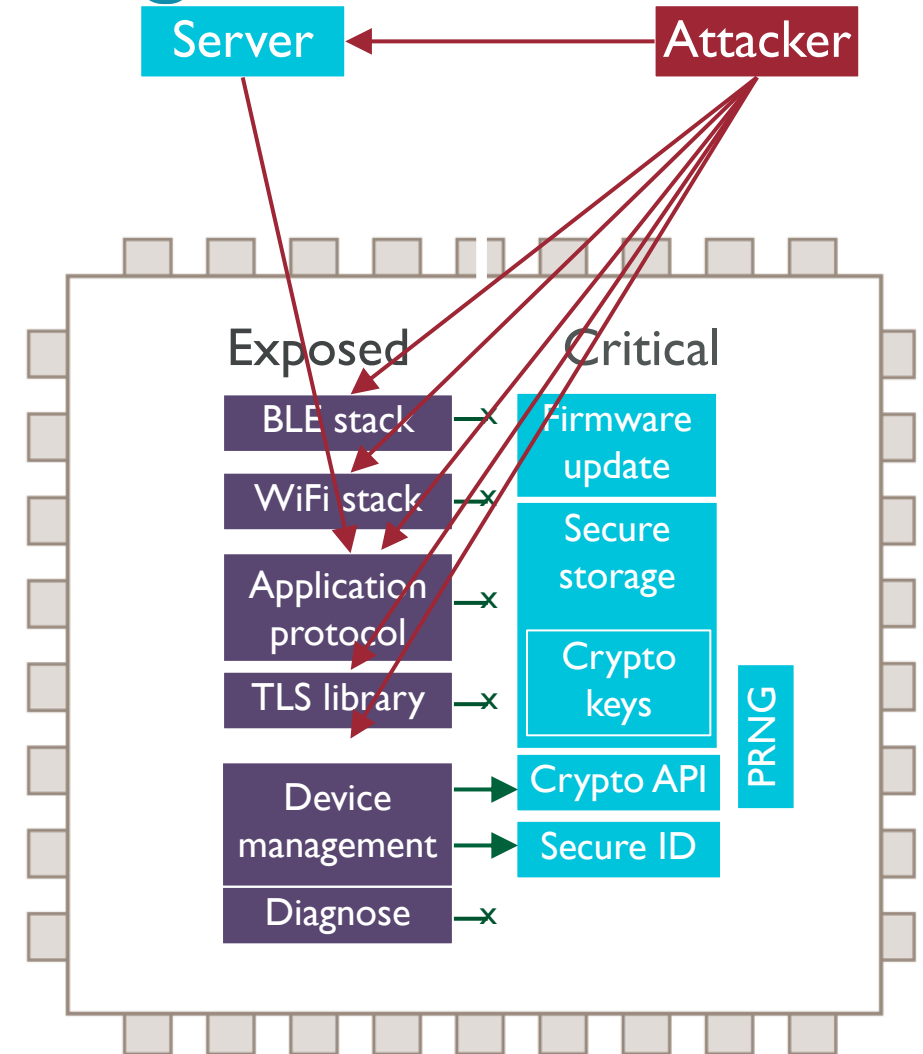
The device security problem - Mitigation strategies

- Split security domains into:
 - Public uncritical code
 - Protected critical code
- Protect key material and system integrity
 - Use ARMv7-M MPU or TrustZone for v8-M
 - Keep footprint of critical code small
- Public code operates on cryptographic secrets via defined private API
 - No access to raw keys



The device security problem – Mitigation benefits

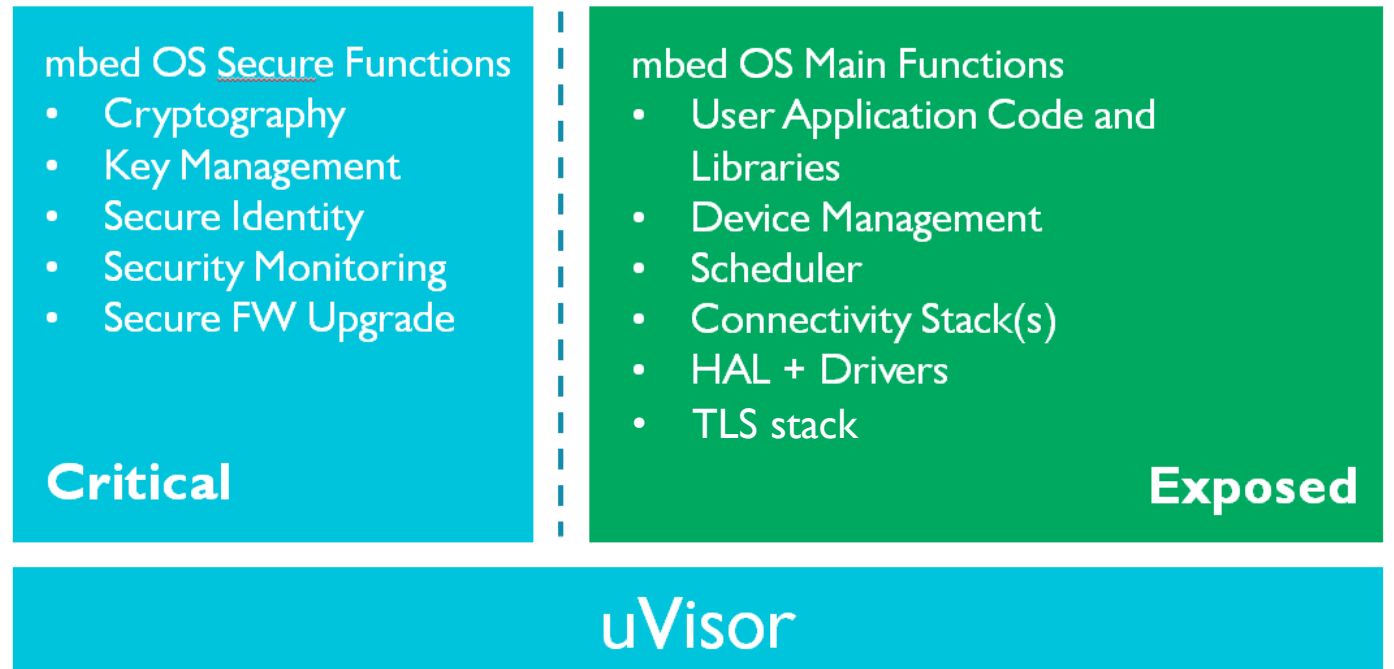
- Attackers can compromise the exposed side without affecting critical code
- Cryptographic hashes can be used to verify the integrity of the exposed side
 - Triggered on server request
 - Protected security watchdog allows remote control
- Protected side can reliably reset exposed side to a clean state
- The device attack surface is massively reduced as a result




Pulling it together

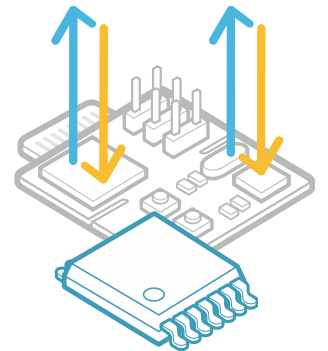
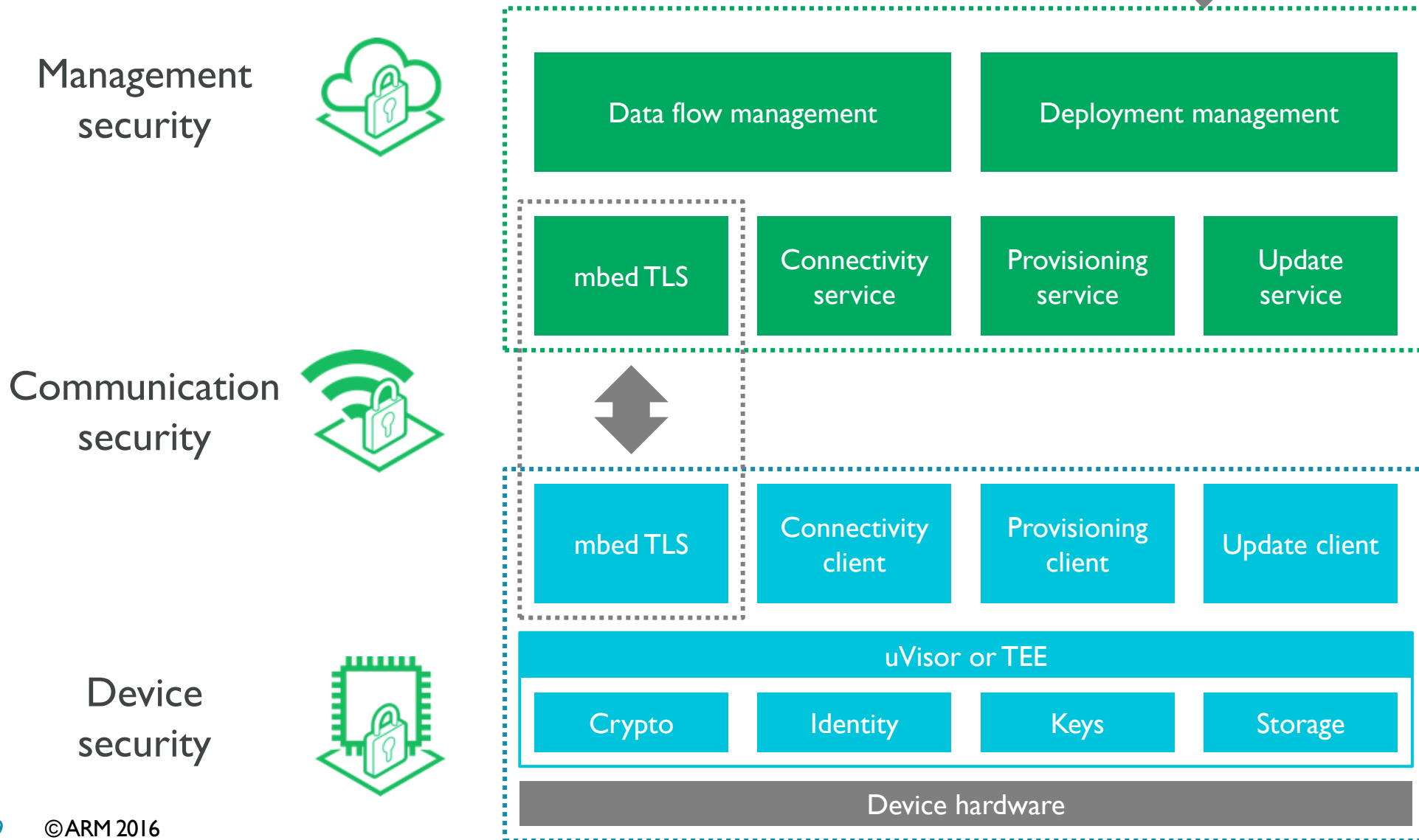
mbed OS

- mbed uVisor is part of mbed OS, but is optionally enabled depending on the underlying hardware support
- If present, mbed uVisor boots the mbed OS image, and configures secure boxes using the provided access control lists



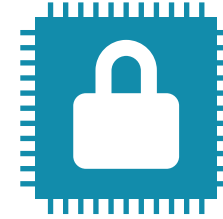
mbed OS security

Cloud applications
platforms 



Summary

- IoT deployments will not scale without trust
- Very few developers have strong security experience
- mbed IoT Device Platform provides a comprehensive security foundation
 - Device security
 - Communications security
 - Lifecycle security



ARM

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2016 ARM Limited

©ARM 2016