Internet of Things, Security & Privacy

Dr. Emmanuel Baccelli

Inria — Freie Universität Berlin

Course Format

- No required weekly attendance, but
  - mid-term: intermediate status presentation
  - end of term: final presentation
- Expected output
  - written report (~12 A4 pages, IEEE LaTeX template)
  - presentations
  - (demos, measurements etc. not necessary, but why not?)
  - (optional: publish your paper on arXiv)
- My office hours
  - Email me to setup a (virtual) meeting emmanuel.baccelli@fu-berlin.de

WARNING

- This seminar **demands substantial work**
  - Comprehensive survey & present academic work in written + oral form

- This seminar is **research-oriented**
  - Suggestion: plan it as a preliminary for a thesis

➔ Contact me later to discuss potentially related thesis topics!

Next Steps

- After 1 week (May 4th): topic selection
- After 2 weeks : deadline to submit initial skeleton + refs
- May. 11th : intermediate presentation of topic + skeleton
- After ~6 weeks : deadline to submit work-in-progress version of the report (June 8th)
- July 3rd : deadline to submit final version of the report
- July 6th : final presentation session

Choosing a Topic

1. Choose a field. Suggested fields:
   - IoT crypto primitives
   - IoT privacy mechanisms
   - IoT network security
   - IoT software supply-chain
   - IoT secure software execution
2. Specify a topic within chosen field. Potential ideas for topics:
   - see papers in https://github.com/emmanuelsearch/some-iot-and-security-papers
   - Start surveying your topic (after I confirm your topic)

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends

Context

- World War III is upon us (online)
  - geopolitically-driven (state-driven)
  - profit-driven (pirates, zero-day attacks)

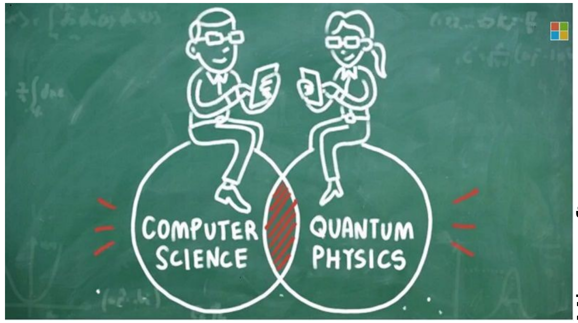
Anonymous

- Personal data-hungry Behemoths are upon us
  - captive users, walled gardens, not necessarily secure
  - EU fightback: General Data Protection Regulation (GDPR)

reality.ai

Context

- Extreme computing power becomes average
  - Now: pooled power, from NSA to botnets & everything in between (e.g. Coinhive covert mining ads)
  - Later: quantum computing?

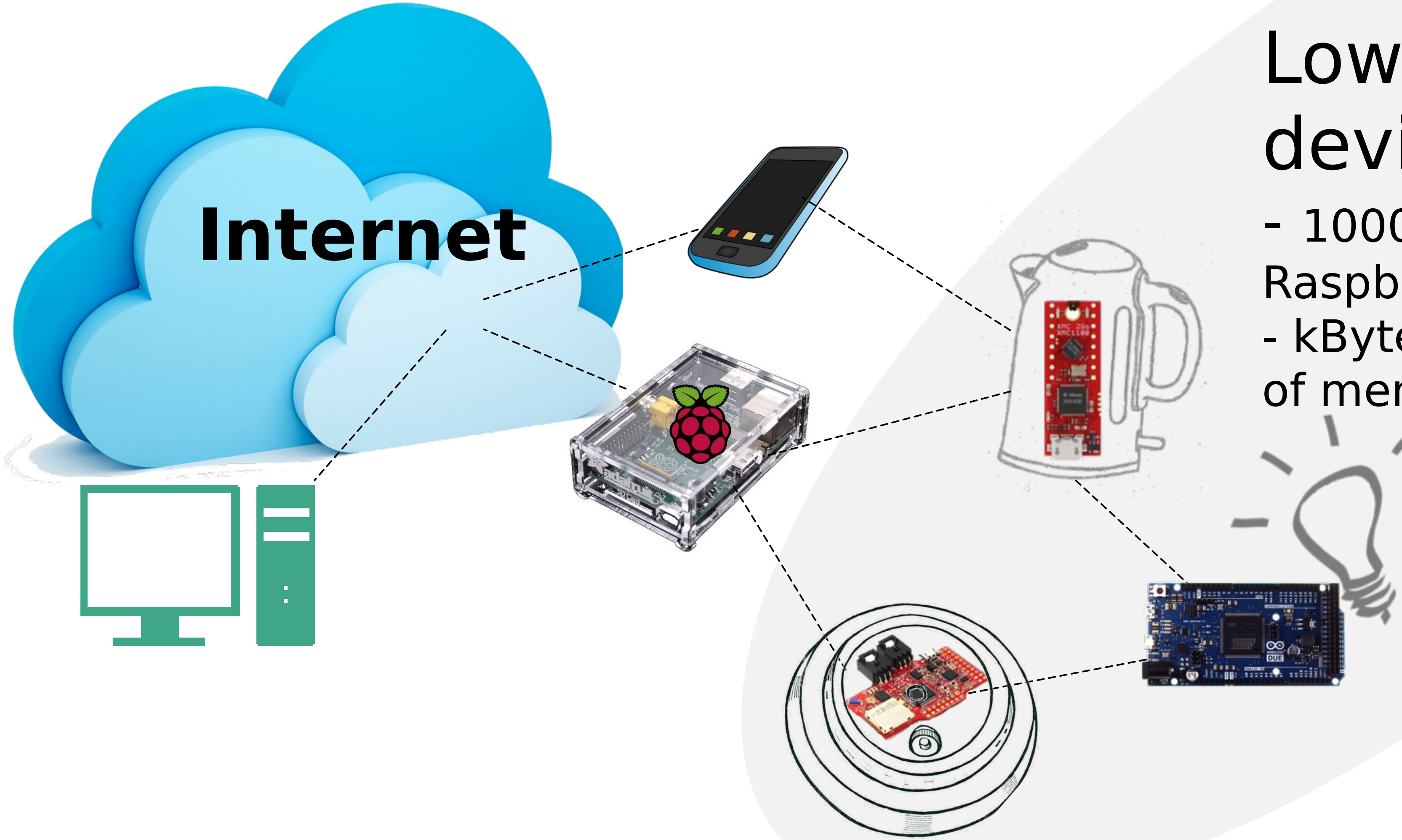


Microsoft

- Ubiquitous computing & connectivity is upon us
  - Giant cyberphysical robot
  - High-end IoT vs Low-end IoT

Biff Tenon

IoT Hardware



Low-end IoT devices

- 1000x less energy than RaspberryPi
- kBytes instead of GBytes of memory

Microcontrollers e.g.
- AVR (8-bit)
- MSP430 (16-bit)
- Cortex M (32-bit)
- MIPS
- ...

Low-end IoT Devices : Polymorphism

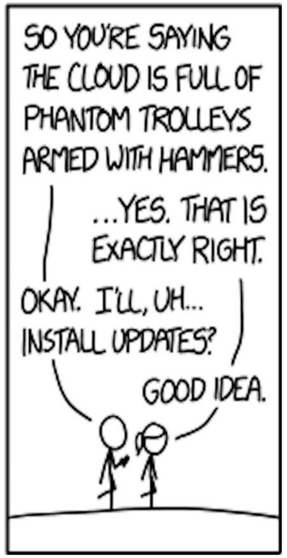
- Various vendors
- Various architectures (8-bit, 16-bit, 32-bit)
- Various low-power communication technologies (BLE, 802.15.4, DECT…)

AGENDA

- Context
- IoT Attack Surface
- Inherent Tradeoffs
- IoT Security Trends

Traditional IT Attack Surface

- **Human vector**
  - Misconfiguration, phishing, social engineering
  - 95% security incidents involve human error*
- **Hardware vector**
  - e.g. Spectre & Meltdown **vulnerabilities** on recent processors: leaks breaking isolation
  - … and **backdoors** from NSA & co ?

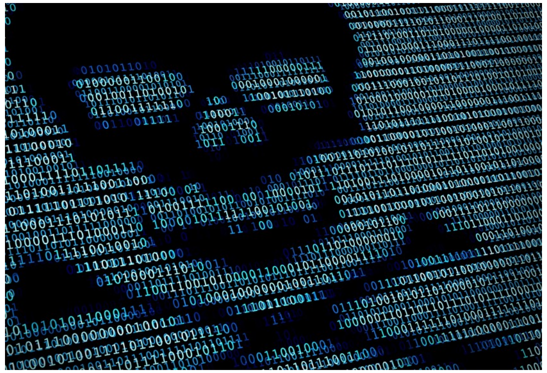


xkcd.com No 1938

* Cybersecurity Intelligence Index, IBM, 20

Traditional IT Attack Surface

- **Low-level software vector**
  - e.g. **EternalBlue** vunerability on Windows <10 (NSA exploit turned bad, used in WannaCry)
  - e.g. **HeartBleed** OpenSSL (also on Linux!)
  - Fatal combination: exploit OS & network stack vulnerabilities to inject malicious code



itsecurityguru.org

- **High-level software vector**
  - e.g. malicious PDF exploiting Adobe Reader vulnerabilities

Traditional IT Attack Surface

- **Software supply-chain vector**
  - e.g. backdoor hacked into software updates of Ccleaner application*
  - attack laced legitimate software with malware (distributed by a security company!!!)

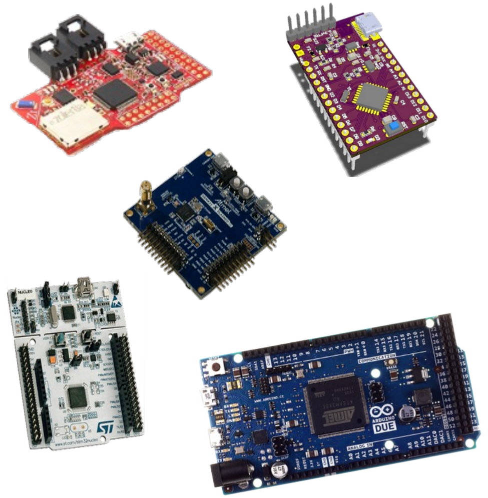
xda-developers.com

* www.wired.com/story/ccleaner-malware-supply-chain-software-sec

IoT *vs* Traditional IT Attack Surface

- IoT ~ Machine-to-Machine: the **human factor is less important**

- Single binary systems (so far)
  - **no high-level software**
- **Low-end memory, CPU capacities**
  - kBytes of memory instead of Gbytes or more
  - MHz instead of GHz
  - mW or less, instead of W or more



IoT *vs* Traditional IT Attack Surface

- IoT $\sim$ giant cyberphysical robot: hacked system can cause direct physical harm
  - $\Rightarrow$ **acceptable risks are changed**
- Sensors everywhere, all the time
  - $\Rightarrow$ **scope of privacy breaches are changed**
- Industrial IoT applications*
  - $\Rightarrow$ required level of system availability is higher

nbcnews.com

* AR Sadeghi et al. "*Security and Privacy Challenges in Industrial Internet of Things*," in ACM DA

IoT *vs* Traditional IT Attack Surface

- Chain reactions
- Extended functionality attacks

In a nutshell

  - ~~Humans~~
  - Hardware
  - Low-level software
  - ~~High-level software~~
  - Software supply-chain

- Good news:
  - attack surface is probably smaller than usual

- Bad news:
  - harsher contraints & potentially more impactful attacks
  - no human in the loop means its harder for some aspects (bootstrap…)

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends

Data Economy *vs* Privacy

The utility/privacy trade-off

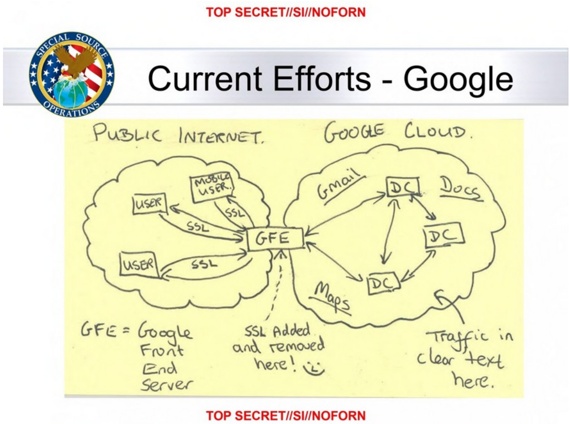
- Companies:
  - **Some do not care** about privacy,
  - **Others need data** to provide services
- End-users:
  - want services,
  - but want **control of their privacy**

Techcrunch.com

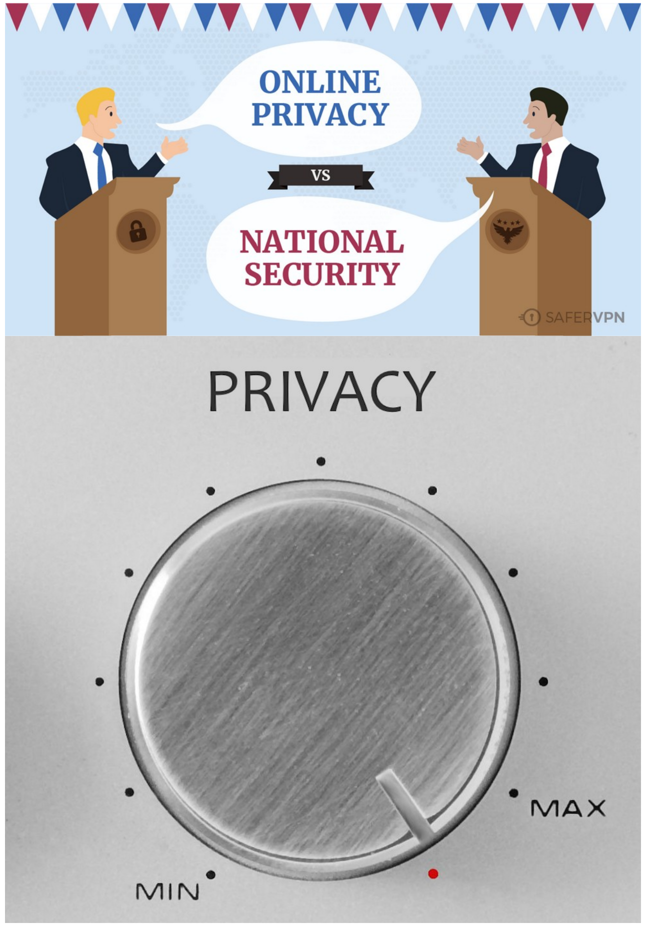
- How to go from here (massive data raiding) to there (user-tunable *signal*)?

(National) Security *vs* Privacy

- From *individual* surveillance to *mass* surveillance (and back again?)

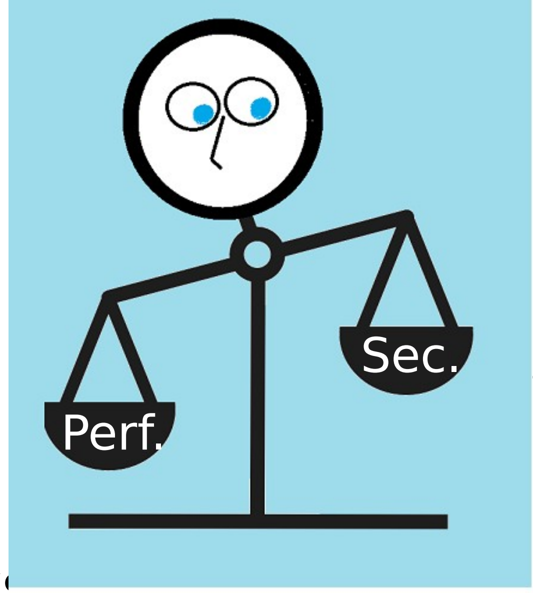


  - **Crippling crypto** lets NSA through... but also lets (DIY) pirates through
  - **Tracking contacts** prevents epidemics... but also enables Big Brother



Vuze.com

Performance *vs* Security



Steve Hauptman blog

30s to verify a digital signature?

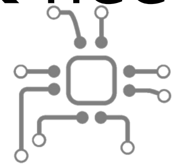
- Not only does performance suck…
- … but also: resource exhaustion attacks

- **Exacerbated** on (future legacy) low-end IoT hardware
  - Bottomline: **cat & mouse play** to remain just above risk threshold
  - Necessary complement: **IoT software updates**

Bottomline: Functionality *vs* Risk

- Today's IoT: **not an acceptable tradeoff** w.r.t. functionalities *vs* risks
  - B. Schneier: Internet of (Unsecure) Things*

- **Dimensions of the work** needed to change that?



- Improving functionality
  - Better IoT hardware
  - Richer IoT software

- Mitigating risks
  - More IoT security

* https://www.rsaconference.com/blogs/bruce-schneier-talks-about-securing-the-world-sized-web-at-rsac-apj-2016

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends
  - Trusted IoT Hardware
  - IoT Cryptography Primitives
  - IoT System Software
  - IoT Network

IoT: The Hardware Vector

- Various categories of attacks
  - invasive hardware attacks
  - reverse engineering attacks
  - **side-channel attacks**
    - Information gained with timing information, power consumption, electromagnetic leaks, speculative execution, caching…

Related questions:

⇒ are there IoT-specific side-channel attacks?

⇒ what functionalites can embedded crypto hardware modules provide?

⇒ …

Trusted Execution on IoT Hardware

- Principle: **secure area of a processor** for isolated execution, integrity of trusted applications & confidentiality of their assets
- **Sancus*** on MSP430 16-bit microcontrollers (OpenMSP430)
  - Prototype **isolating software components** via memory curtaining
    - Added MMU and crypto HW unit on openMSP430 (open source!)
    - Text/Data/ProgramCounter states monitoring/matching, per software component
  - **Remote attestation** & authenticates communication with software component
    - HW crypto enables key derivation per software component
  - Sancus2.0 tested in automotive context. Claims only 6% energy overhead
- Similar: **TrustZone** for popular ARM Cortex-M 32-bit microcontrollers
  - Upcoming Cortex-M33 and Cortex-M23 micro-controllers

* J. Noormans et al. '*Sancus 2.0: A Low-Cost Security Architecture for IoT Devices*', ACM Transactions on Privacy and Security, 2017

Trusted Execution on IoT Hardware

- **TEEP** working group at IETF *
  - Context: delete, **update applications** running in the TEE
  - Goal: **communication** between the TEE, a relay outside TEE & a remote server
    - ⇒ Trusted execution environment protocol (TEEP)

Side note: installing new software in the TEE **increases attack surface**...

* https://datatracker.ietf.org/wg/teep/documents/

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends
  - Trusted IoT Hardware
  - IoT Cryptography Primitives
  - IoT System Software
  - IoT Network

IoT Crypto Primitives

- Devices deployed now will last for years
  - Maybe decades!
    - MSP430 runs **12 years on an AA battery**.
    - Energy harvesting

- **Future-proof crypto** is thus crucial. Quantum resistance?
  - Trade-off: **key & signature size *vs* speed** *
    - ex. ECC 256b key *vs* McEliece 500kB key
    - ex. ECC 80B signature *vs* MQDSS 40kB signature

* S. Samardjiska '*Post Quantum Cryptography for the IoT*,' RIOT Summit, 2017

IoT Crypto Primitives

It is nevertheless possible to **prepare IoT crypto for post-quantum now**

- symmetric crypto needs upgrade but same security =~ double key size
- asymmetric crypto
  - some techniques would break entirely (e.g. RSA?)
  - some techniques are quantum-resistant (hash-based signatures…)

- **NIST crypto competition** efforts recently launched *
  - Upcoming: new standard cypher suites
  - (Conspiracy theory: baked-in backdoors ?)

* NIST Post-Quantum Crypto competition, started in 2017

IoT Crypto Primitives

- IoT Symmetric Crypto
- IoT Asymmetric Crypto
- Operations over Encrypted IoT Data

IoT Symmetric Crypto

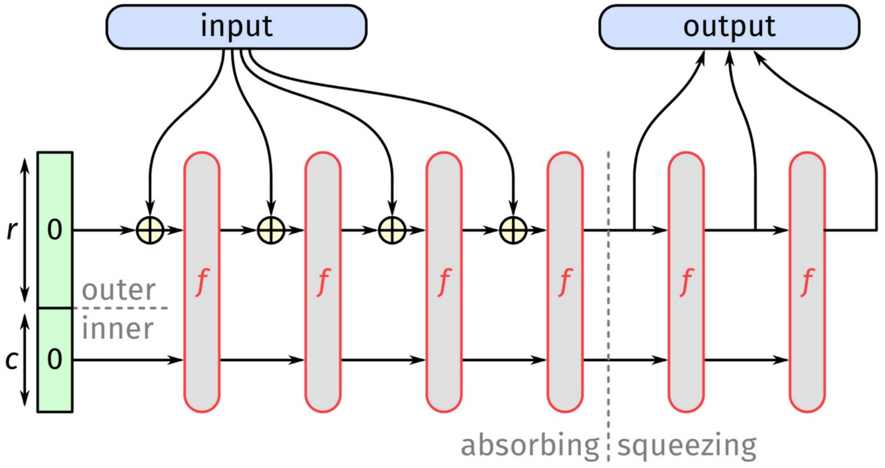
- Symmetric = same key everywhere.

More flexible primitives:

SHA-3’s **sponge construction*** for hashing

- Easy to (re)configure security level
  - Just vary capacity c
- Shared code to provide various functions
  - Pseudo-random number generator
  - Message authentication code (MAC)
  - Stream encryption
  - (more with the duplex construction)

- On-going work experimental work to evaluate this prospect on top of RIOT



G. Van Assche '*Permutation-based cryptography for the IoT*,' RIOT Summit, 2017.

* G. Bertoni et al. '*Cryptographic sponge functions*', 2011.

IoT Asymmetric Crypto

- Symmetric crypto:
  - communicating entities need a shared secret (the key)

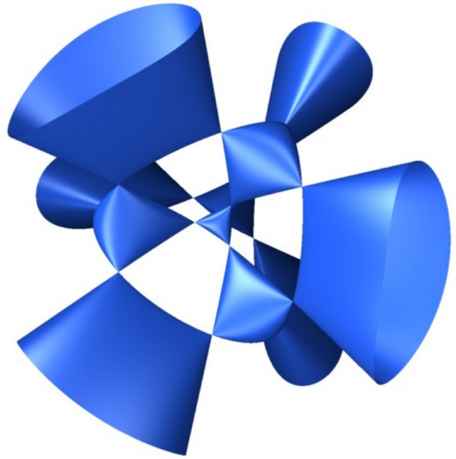
  $\Rightarrow$ key distribution problem, Pre-Shared Key (PSK) is norm on IoT

- Asymmetric crypto with public/private keys:
  - Public Key Infrascruture (PKI) solves key distribution
  - Public key allows digital signatures

IoT Asymmetric Crypto

Smaller & faster crypto

- more efficient implementation
  - **tweetnacl** (Bernstein et al.): Source funnily fits in 100 tweets, using curve25519
- more efficient algorithms
  - **uKummer ***: Smarter use of algebraic geometry
    - software-only hyperelliptic cryptography on constrained platforms
    - demonstrated on AVR 8-bit and ARM Cortex-M 32-bit
    - up to 70% faster & 80 % smaller compared to using curve25519
  - **qDSA ****: even smaller stack & code size



* J. Renes et al. '*μKummer: Efficient hyperelliptic signatures and key exchange on microcontrollers*', CHES, 2016.
** J. Renes, B. Smith '*qDSA: Small and Secure Digital Signatures with Curve-based Diffie–Hellman Key Pairs*', ASIACRYPT 2017.

IoT Asymmetric Crypto

Humans (even if very skilled) make buggy code

- **Formally verified crypto code**
  - HACL* library: written in F* programming language,
  - F* code formally verified (memory safety, mitigations against timing side-channels, and functional correctness)
  - F* code then compiled to readable C code

  - Elements of HACL* already in Firefox (Quantum, latest verion)
  - Elements of HACL* currently integrated into RIOT

* JK Zinzindohoué et al. *'HACL*: A verified modern cryptographic library*,' ACM CCS, 2017

Operations over Encrypted IoT Data

The cloud, or the server hosting IoT database may not be trusted

⇒ Nevertheless it may be required to (batch) process IoT data

- Use of **partially homomorphic** crypto
  - Talos and Pilatus prototype platforms *
  - **Allows some operations (range, sum) over encrypted data**
  - Using Elliptic-Curve ElGamal crypto-system (instead of Paillier)
  - Encryption by low-end IoT devices themselves (demonstrated on Cortex-M3)

* H. Shafagh et al. '*Secure Sharing of Partially Homomorphic Encrypted IoT Data*,' ACM SenSys, 2017

Operations over Encrypted IoT Data

- Body of work on **differential privacy** in the field of smart metering *
  - Followed seminal work from **Dwork** in 2006

- Principle: **add (some) noise to IoT data points**
  - Differential guarantee = analysts draws same conclusions about an individual whether the individual includes himself in the dataset or not
    - **Still able to extract coarse signal** from aggregate data (e.g. mean, average…)
    - **No privacy violation in practice**

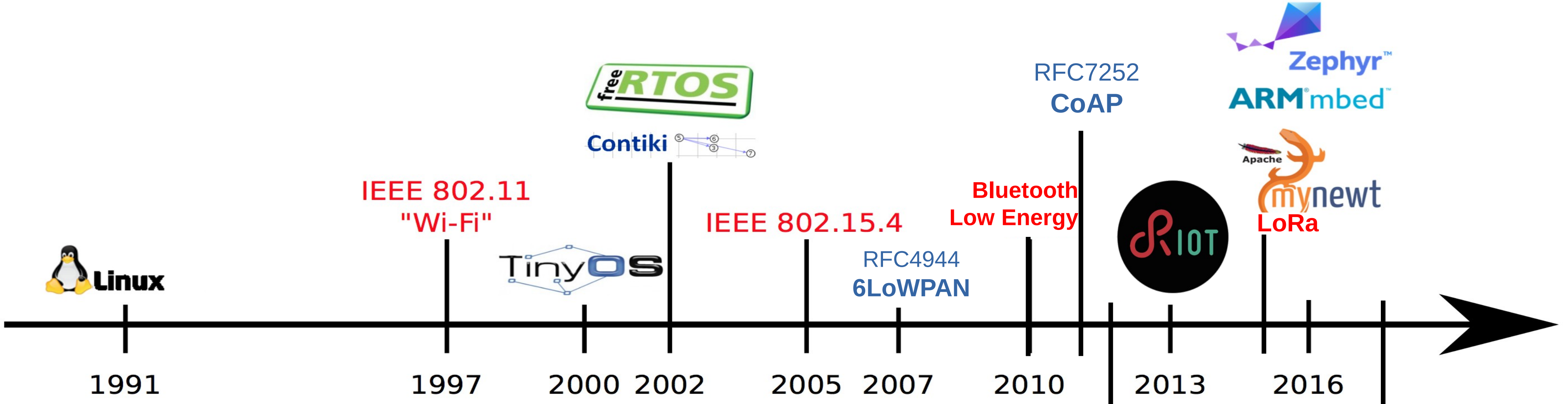
* V. Rastogi et al. '*Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption*', ACM SIGMOD 2011.

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends
  - Trusted IoT Hardware
  - IoT Cryptography Primitives
  - IoT System Software
  - IoT Network

(IoT) SOFTWARE: EVOLUTION

- Old style: rudimentary, closed-source, vendor-locked OS, *__no updates__*
- New trend: real operating systems*, free & open-source, *__with updates__*



* O. Hahm et al. "*Operating Systems for Low-End Devices in the Internet of Things: A survey*," IEEE Internet of Things Journal, 2016.

Trusting IoT Software (a priori)

Providing guarantees on software components of IoT operating systems?

- Tock OS * isolates software faults & manages dynamic memory for applications
  - use of **memory protection unit** (MPU) of Cortex-M4 and of **Rust programming** language
  - Rust enables **memory-safety & type-safety** while providing performance close to C
  - MPU enables **isolation of processes** from the kernel and from each other

- Proven C code generated from F*
  - Potential use to provide RIOT components other than the HACL crypto library?

* A. Levy et al. '*Multiprogramming a 64 kB Computer Safely and Efficiently*,' ACM SOSP, 2017.

(IoT) SOFTWARE UPDATES: A NECESSITY

xda-developers.com

- What Internet-age software has taught us:
  - you can't secure what you can't update!
  - software updates are an attack* vector!

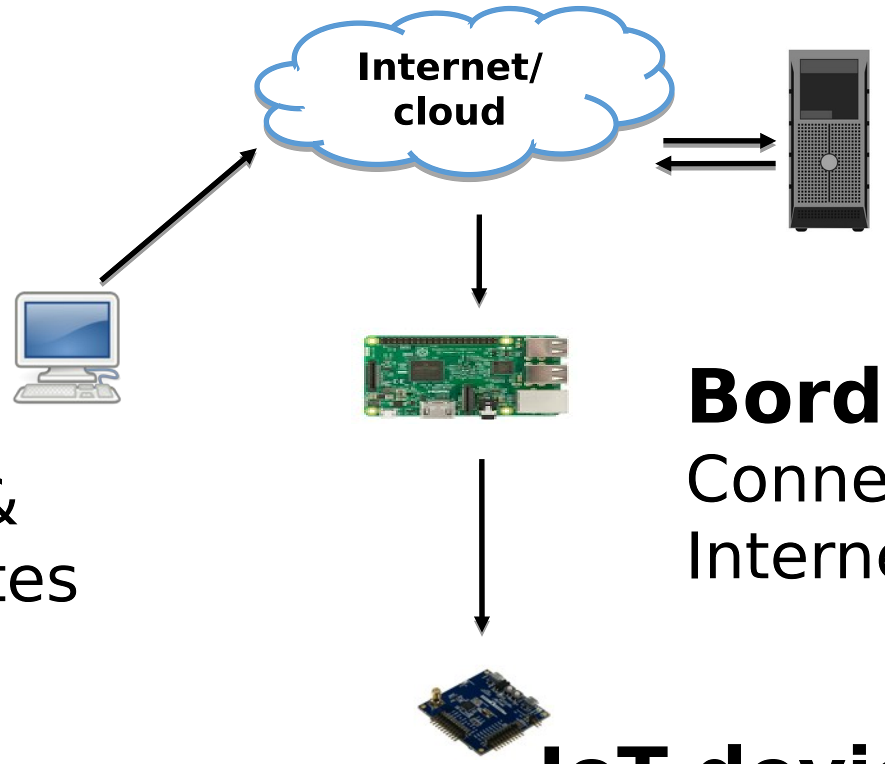
⇒ Enabling (legitimate) software updates is crucial & difficult

- enforcing legitimacy can turn bad -- beware of **Treacherous Computing** (R. Stallman)

⇒ Even more challenging on microcontroller-based IoT devices

* www.wired.com/story/ccleaner-malware-supply-chain-software-security

(IoT) SOFTWARE UPDATES: ARCHITECTURE



Update Server:

stores updates
& serves them
to IoT devices

Developer:

Builds, signs &
uploads updates

Border router:

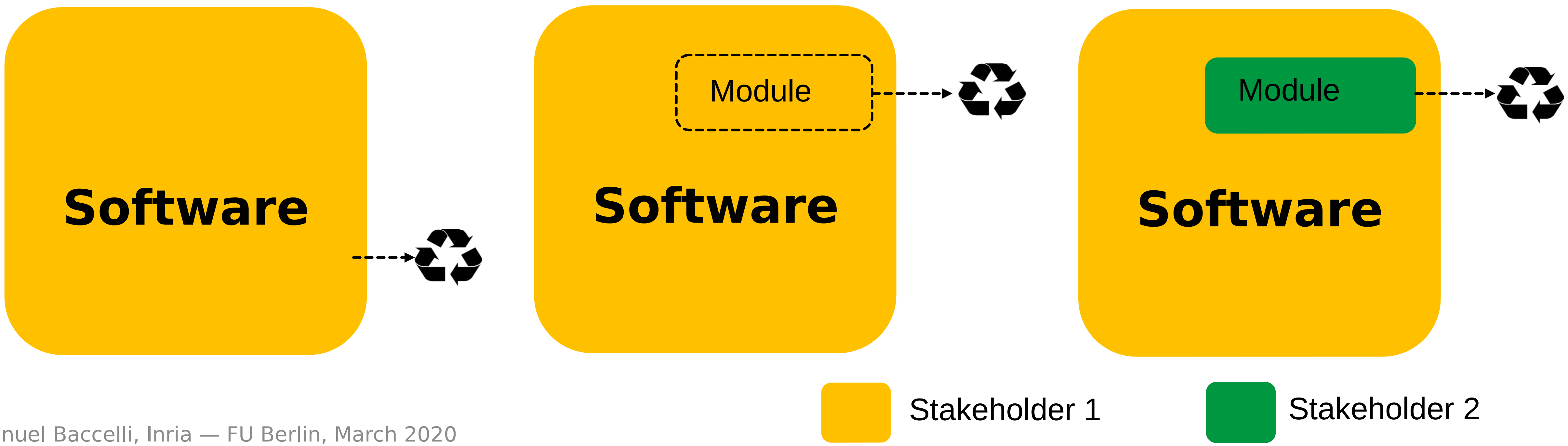
Connects device to the
Internet

IoT device:

Downloads, verifies
and boots updates

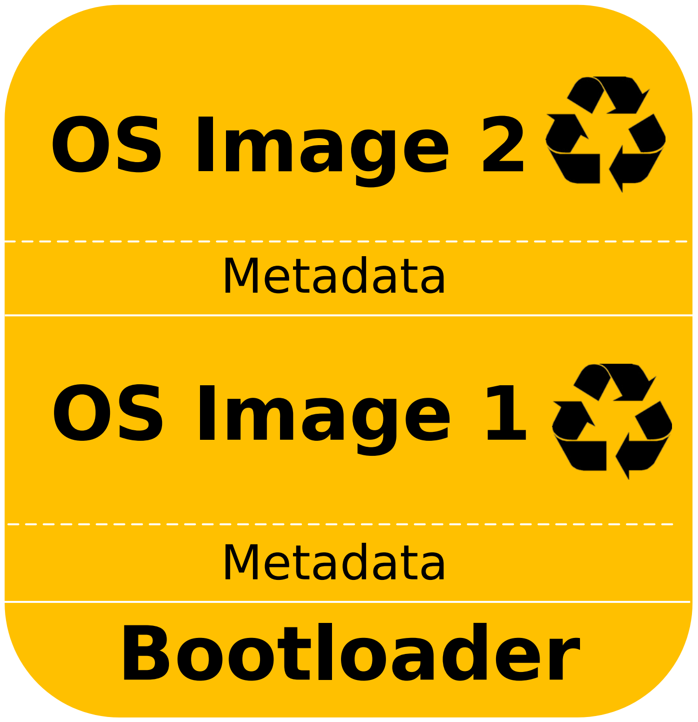
(IoT) SOFTWARE UPDATES: APPROACHES

- ✓ Case 1 : monolithic software update, single stakeholder
- ✓ Case 2 : modular software updates, single stakeholder
- ✓ Case 3 : modular software updates, multiple stakeholders



LOW-POWER IoT SOFTWARE UPDATES: TYPICALLY, FIRMWARE UPDATES (CASE 1)

You thought you were tight w.r.t. memory?



Memory must be further split :

- Bootloader
  - Minimalistic startup logic
- Several OS Images
  - May need n>2 for roll-back etc.
- Metadata => **SUIT***

SUIT = standard metadata & crypto to guarantee authenticity & integrity of IoT software updates

* https://tools.ietf.org/html/draft-ietf-suit-manifest-04

SUIT-COMPLIANT WORKFLOW (IN RIOT*)

MCU memory:
32kB RAM
256kB Flash

(Crypto: ed25519 digital signatures, SHA256 hash)

Maintainer
(P,S)

`(OOB: Provision Public Key P)`

IoT Device

PHASE 0
Commission device

PHASE 1
Build update

[Image]

PHASE 2
Publish & sign update

[Manifest]

`PUT Image, {Manifest}`$_s$

Repo

`GET`

PHASE 3
Fetch update

PHASE 4
Auth.: check sign.
Integrity: check hash

PHASE 5
Check OK? Install.
(Else: send alert)

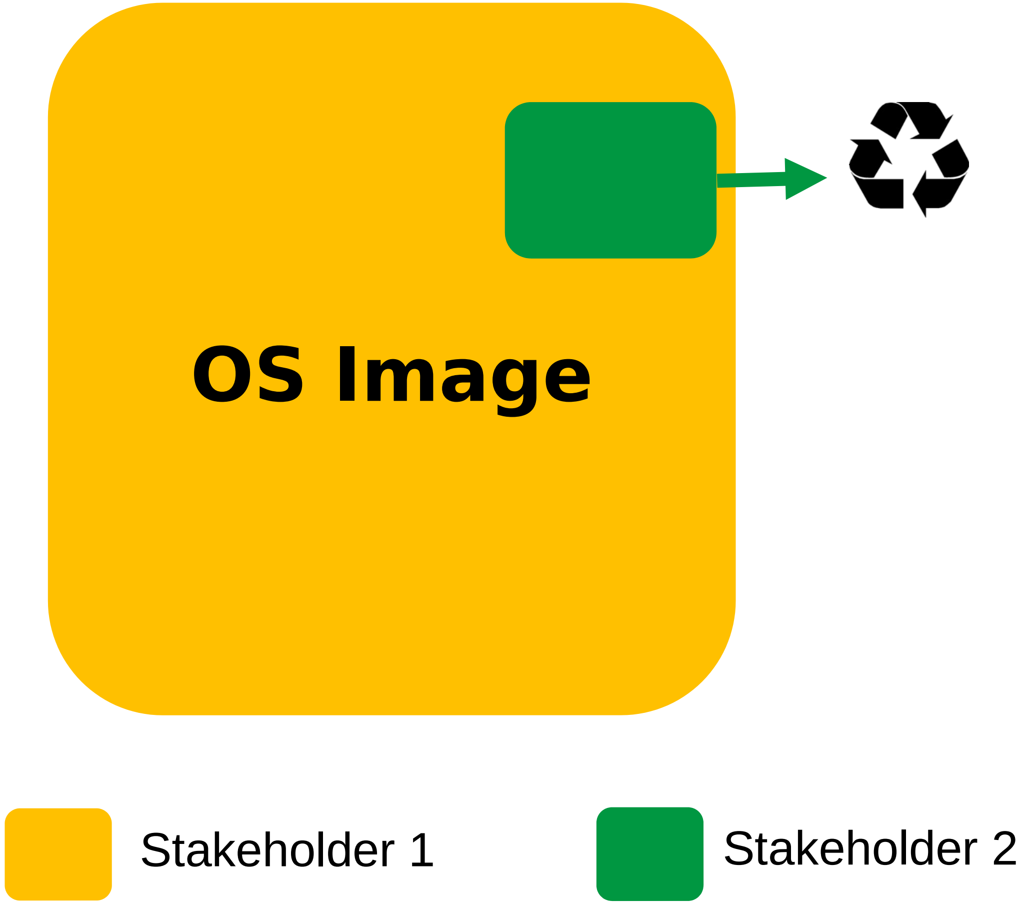
* K. Zandberg et al. "*Secure Firmware Updates for Constrained IoT Devices using Open Standards: A Reality Check,*" IEEE Access, 2019.

LOW-POWER IoT SOFTWARE UPDATES: TOWARDS CASE 3?

The rest of the Internet?
=> Resembles more Case 3!
(modular updates, multiple stakeholders)

Challenges for low-power IoT:
- low-power modularization*
- security & sandboxing of modules
- decentralized supply-chain frameworks **
- ...

=> On-going research



* E. Baccelli et al. "*Scripting Over-The-Air: Towards Containers on Low-end Devices in the Internet of Things*," IEEE PerCom, 2018.

** K. Nikitin et al. '*CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds,*' USENIX Security Symposium, 2017.

AGENDA

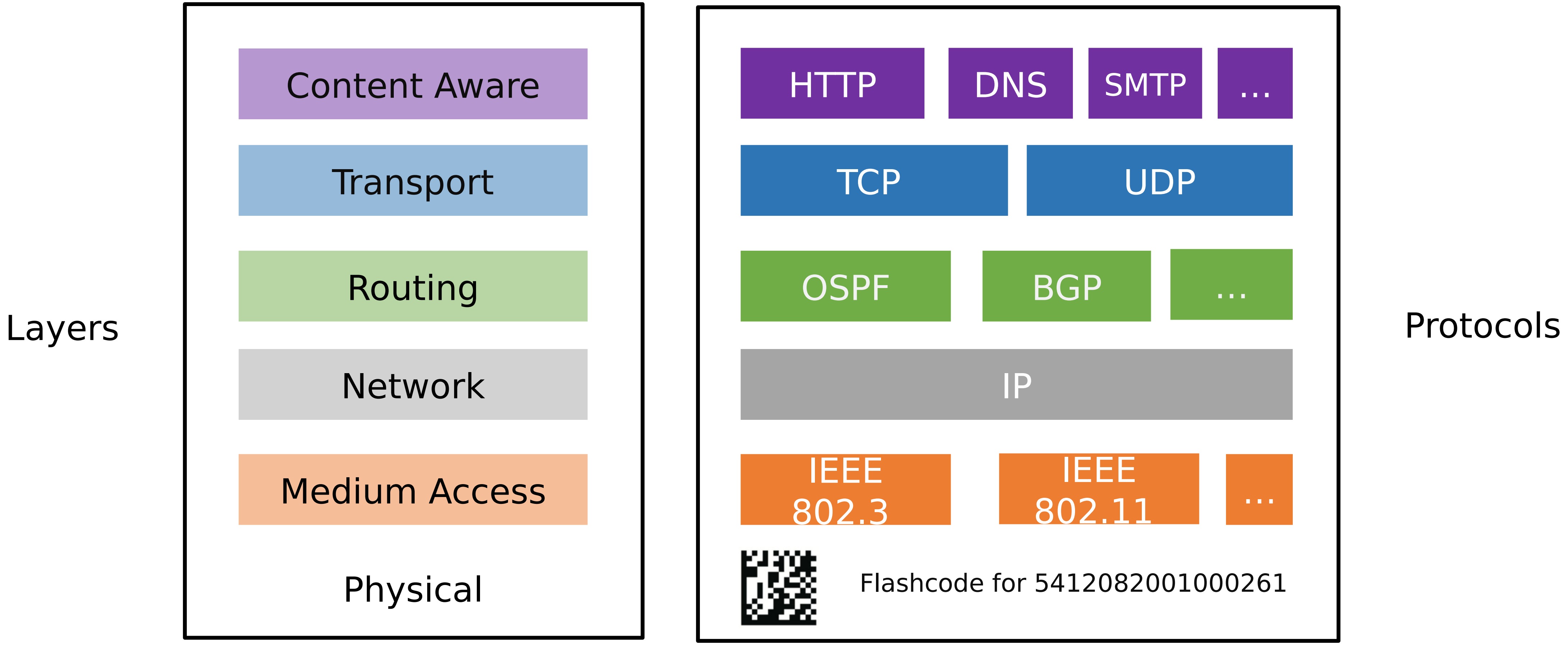
- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends
  - Trusted IoT Hardware
  - IoT Cryptography Primitives
  - IoT System Software
  - IoT Network

Categories of Protocols

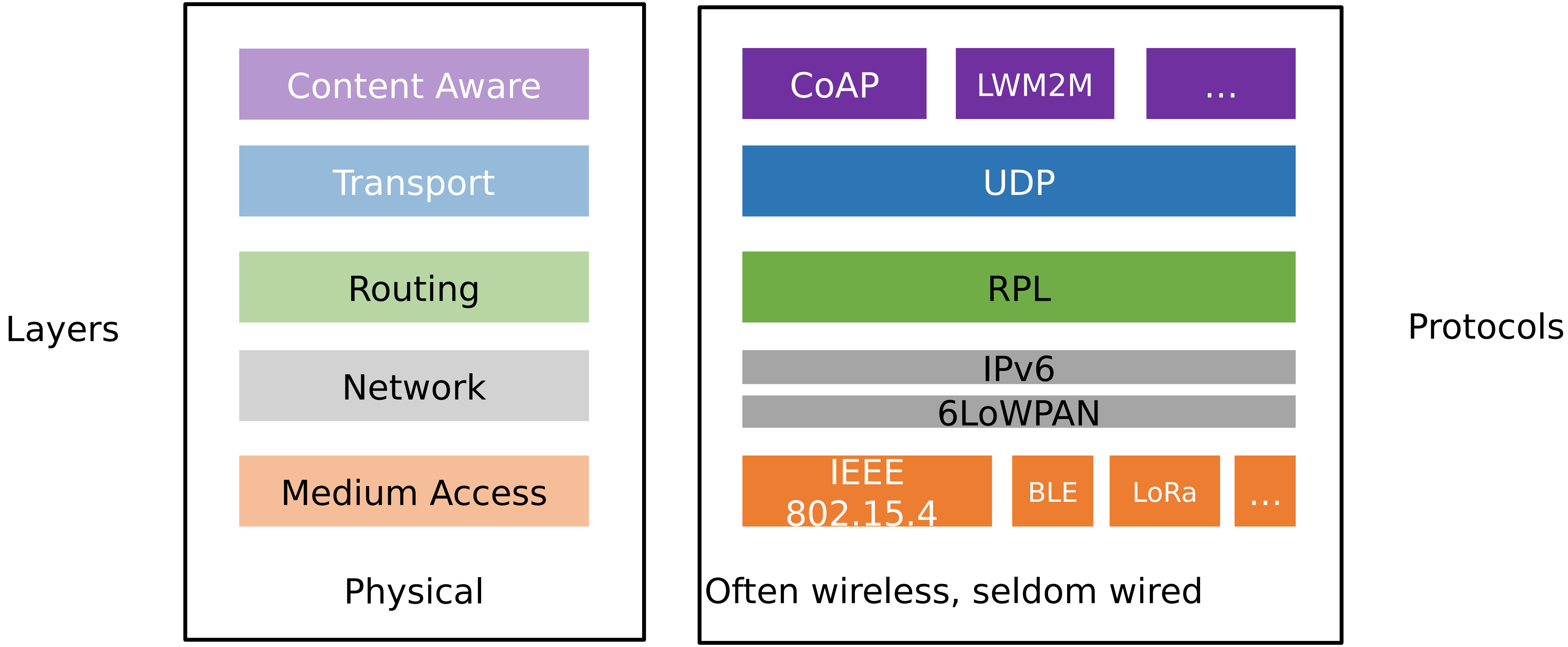
Layers

Content Aware	Mechanisms using names, for example: www.blabla.com/picture.jpg
Transport	Adapt rate at which data chunks are sent across the global network (& verify chunks went through)
Routing	Establish & use paths across the global network
Network	Global network: packet formats & addresses usable across heterogeneous local networks
Medium Access	Local network: transmission of packets over a link shared by n>2 computers
Physical	Transmission bit per bit over a communication medium which connects 2 computers

Categories of Protocols



IoT Communication Protocol Stack

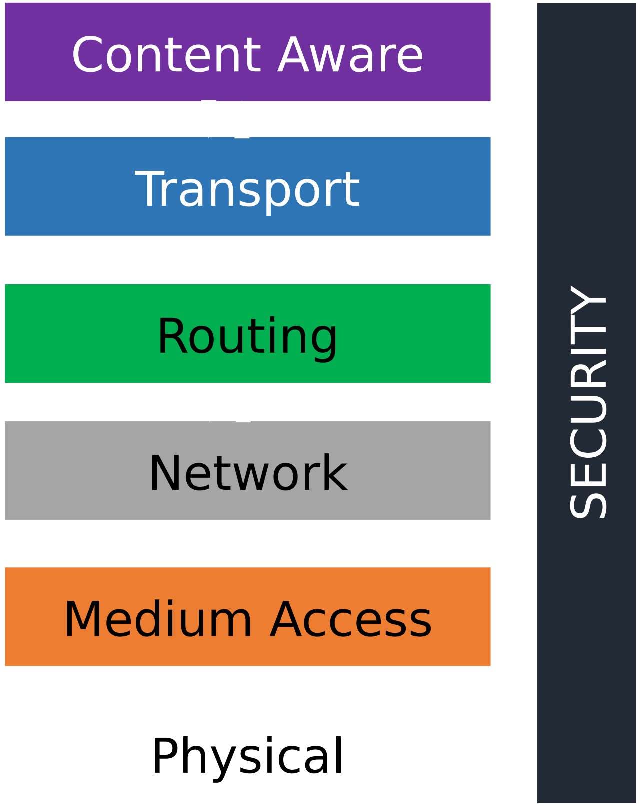


IoT Communication Protocol Stack

- Types of Attacks: (D)DoS, man-in-the-middle attacks…

⇒ Need for Authentication, Authorization, Integrity, Confidentiality, Bootstrapping

IoT Communication Protocol Stack



COSE, OSCORE end-to-end encryption, integrity, & replay protection for CoAP (header/payload)

DTLS (over UDP) for confidentiality, integrity, authenticity with public/private keys

(IPsec, routing protocol security…)

Link-layer security sublayer for authentication, confidentiality, typically with symmetric keys

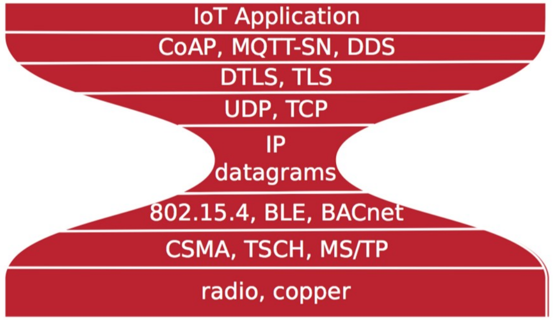
IoT Network Protocol Security

- **IoT link-layer security:** filter legit packets early on
  - Typically AES, 128 bit, in hardware
  - ⇒ HW acceleration helps mitigate DoS attacks (low-end CPU, easily overwhelmed)
- **Routing security:** avoiding sinkholes
  - e.g. TRAIL routing topology authentication for RPL*

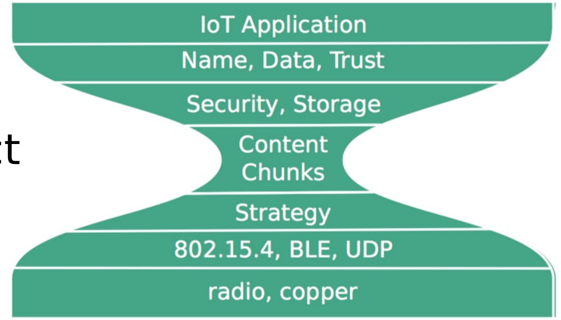
* H. Perrey et al. '*TRAIL: Topology authentication in RPL*', arXiv, 2013

IoT Network Protocols

- **Channel security:** end-to-end confidentiality, integrity, authenticity
  - From **DTLS** 1.2 to 1.3
    - **New handshake**: shorter message exchange
    - **Removed weak/old crypto**
    - https://tools.ietf.org/html/draft-ietf-tls-dtls13
- **Object security:** end-to-end security beyond TLS
  - **COSE, OSCORE, EDHOC** in-layer security for CoAP over *foo*
    - CoAP and HTTP **proxies require (D)TLS to be terminated** at the proxy…
    - https://tools.ietf.org/html/draft-ietf-core-object-security-08
  - Information-centric *vs* machine-centric network architecture
    - Novel paradigm **Information-centric networking** (ICN) yields natural object security
    - Recent work on named-data networking (NDN) applied to IoT



IoT IP Stack



IoT NDN Stack

IoT Network Protocols

- Scalable & secure IoT device on-boarding
  - IoT device bootstrap with zero user interaction, for hundreds of devices ?
  - Studies on pairing based on (matching) ambient data *

* M. Miettinen et al. '*Context-based zero-interaction pairing and key evolution for advanced personal devices*,' ACM CCS, 2014.

AGENDA

- Context
- IoT Attack Vectors
- Inherent Tradeoffs
- IoT Security Trends
  - Trusted IoT Hardware
  - IoT Cryptography Primitives
  - IoT System Software
  - IoT Network
- IoT Testing

IoT Testing

- Decent security requires surviving pentest (penetration tests)

- Standard pentests + framework for low-end IoT devices?
  - Challenges due to market (extreme) fragmentation
  - e.g. for the network attack surface: use of (standard) IPv6 helps, but not much is available for other parts of the stack (6LoWPAN, CoAP…)
    - Metasploit extension* to test 6LoWPAN
    - Some work on fuzzing built on top of Scapy** to test 6LoWPAN

* R. Tomasi et al. '*Meta Exploitation of IPv6 –based WSNs*', 2011.
** A. Lahmadi et al. '*A Testing Framework for Discovering Vulnerabilities in 6LoWPAN Networks*', 2012.

In a nutshell…

- IoT security has numerous aspects
  - Hardware
  - Algorithmic primitives
  - Software (incl. supply-chain)
  - Network

- IoT security practice:
  - Combination of several mechanisms, working at all layers of the system
  - Each mechanism is necessary but not sufficient…

Reminder: Course Format

- No required weekly attendance, but
  - mid-term: intermediate status presentation
  - end of term: final presentation

- Expected output
  - written report (~12 A4 pages, IEEE LaTeX template)
  - presentations
  - (demos, measurements etc. not necessary, but why not?)
  - (optional: publish your paper on arXiv)

- My office hours
  - Email me to setup (virtual) a meeting emmanuel.baccelli@fu-berlin.de

Reminder: Next Steps

- After 1 week (May 4th): topic selection
- After 2 weeks : deadline to submit initial skeleton + refs
- May. 11th : intermediate presentation of topic + skeleton
- After ~6 weeks : deadline to submit work-in-progress version of the report (June 8th)
- July 3rd : deadline to submit final version of the report
- July 6th : final presentation session

Reminder: Choosing a Topic

1. Choose a field. Suggested fields:
   - IoT crypto primitives
   - IoT privacy mechanisms
   - IoT network security
   - IoT software supply-chain
   - IoT secure software execution
2. Specify a topic within chosen field. Ideas for topics:
   - see papers in https://github.com/emmanuelsearch/some-iot-and-security-papers
   - Start surveying your topic (after I confirm your topic)

Now don't forget to

- register on Campus Management System! Else I can't grade you...

- by May 4th send me
  - 2 topics, ordered by preference
  - your GitHub ID and/or a `git` repository you will use for your work-in-progress report, refs, slides etc.