Sécurité de l'Internet des Choses

Pascal.Urien@Telecom-Paris.fr

Agenda

- Introduction à l'IoT
- Position du problème
- Exemples d'attaques
- Modèles de sécurité
- Architecture industrielles
- Résistance au hacking
- Relais
- Implants
- Clones

Introduction à l'IoT

About the Internet of Things (IoT)

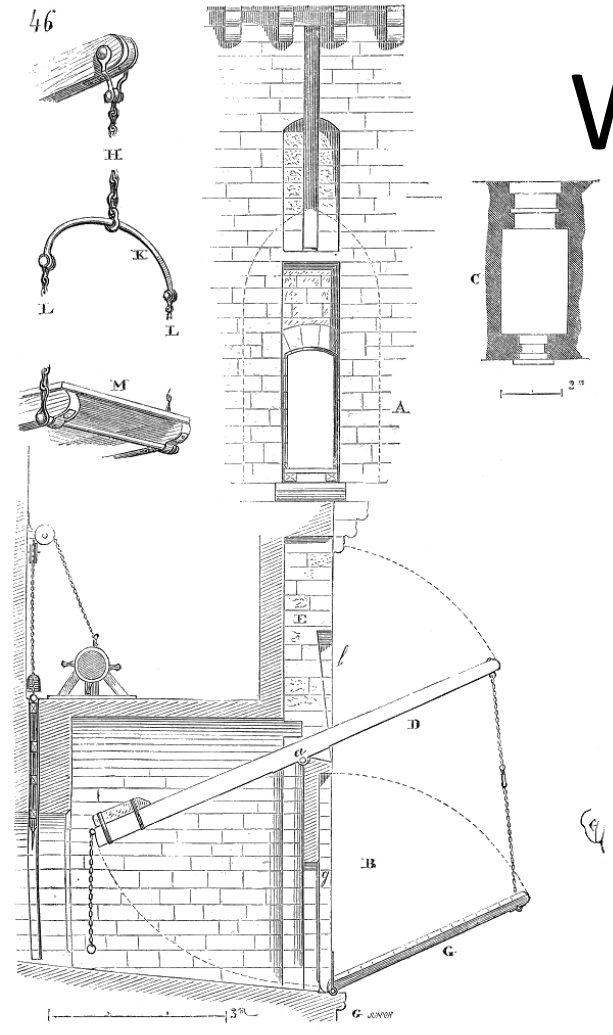
- Pretz, K. (2013). “The Next Evolution of the Internet”

The ***Internet of Things*** (IoT) is a ***network*** *of* ***connected things***.

Objet: Chose solide considérée comme un tout, fabriquée par l'homme et destinée à un certain usage

Appareil: Objet, machine, dispositif électrique, électronique, mécanique, etc., formés d'un assemblage de pièces destinées à fonctionner ensemble

Machine: Appareil ou ensemble d'appareils capable d'effectuer un certain travail ou de remplir une certaine fonction, soit sous la conduite d'un opérateur, soit d'une manière autonome.

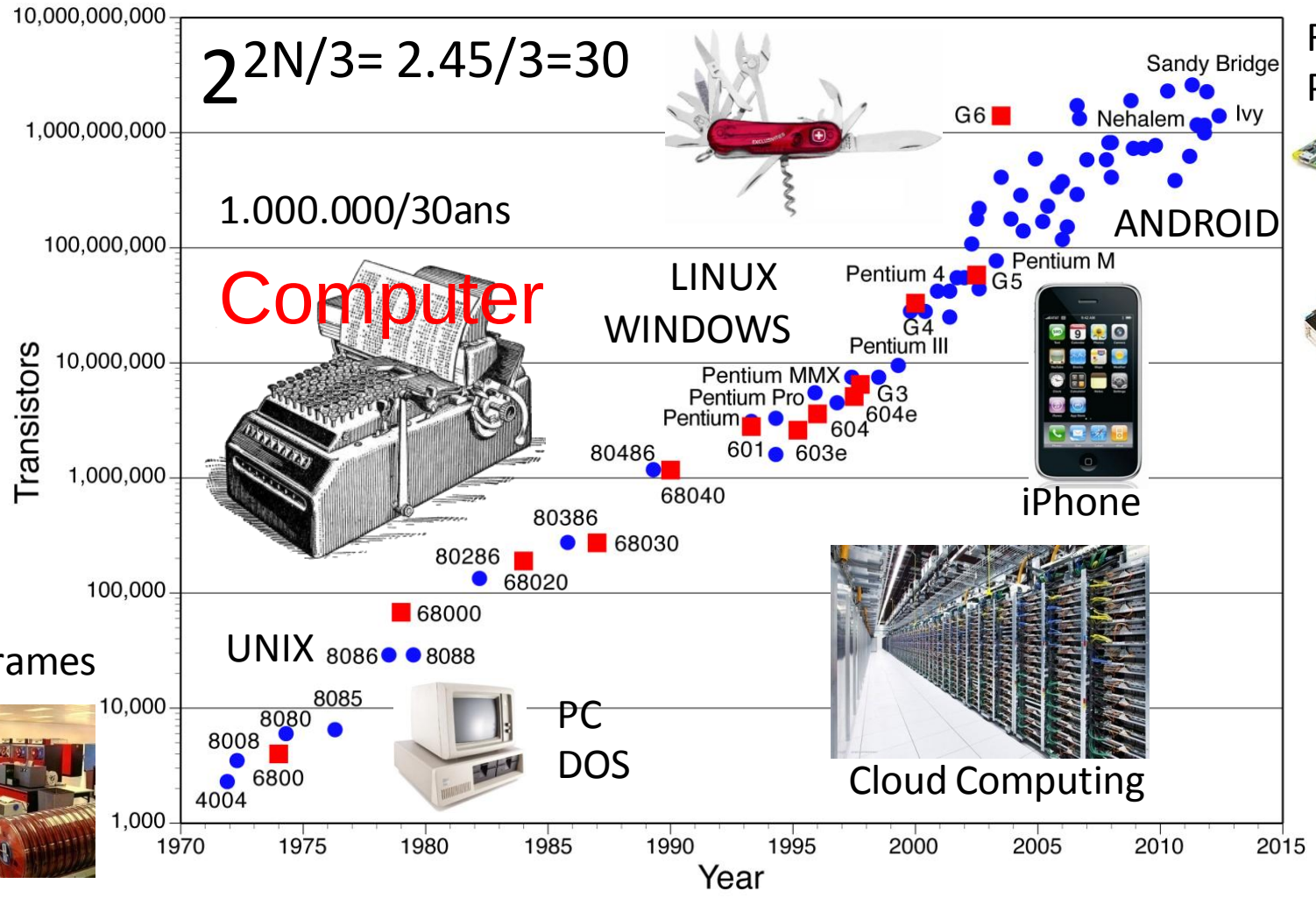


What is a Thing?

- A computer
  - CPU
  - Memories (RAM, ROM, EEPROM, FLASH…)
  - IO buses
- With at least one network interface
  - Wi-Fi, Bluetooth, ZigBee…
- Equipped with sensors and actuators

8-bit Atmel Microcontroller
- 64/128/256KB Flash
- 4KB EEPROM
- 8KB SRAM
- Peripheral Features

Position du problème


$2^{2N/3}$= 2.45/3=30
1.000.000/30ans
Computer
LINUX
WINDOWS
UNIX
PC
DOS
ANDROID
iPhone
Cloud Computing
Main Frames
Raspberry Pi
Arduino
Transistors
Year
10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
Sandy Bridge
Ivy
Nehalem
G6
Pentium M
Pentium 4
G5
G4
Pentium III
Pentium MMX
Pentium Pro
Pentium
G3
604e
604
601
603e
80486
68040
80386
68030
80286
68020
68000
8086
8088
8085
8080
8008
8800
4004

Beyond The Horizon

- The IoT is the death of the Moore Law.
- Waldrop M. "More Than Moore", Nature February 2016 Vol 530
  - *The semiconductor industry will soon abandon its pursuit of Moore's Law.*

Beyond The Horizon

- "Rebooting the IT Revolution: A Call to Action" (SIA/SRC), 2015
  - "*Security is projected to become an even bigger challenge in the future as the number of interconnected devices increases... In fact, the Internet of Things can be viewed as the largest and most poorly defended cyber attack surface conceived by mankind*"
  - "*a short list of requirements includes tamper resistance and secure communications and storage*".

"*A short list of requirements includes tamper resistance and secure communications and storage*"

Node Integrity

Isolation

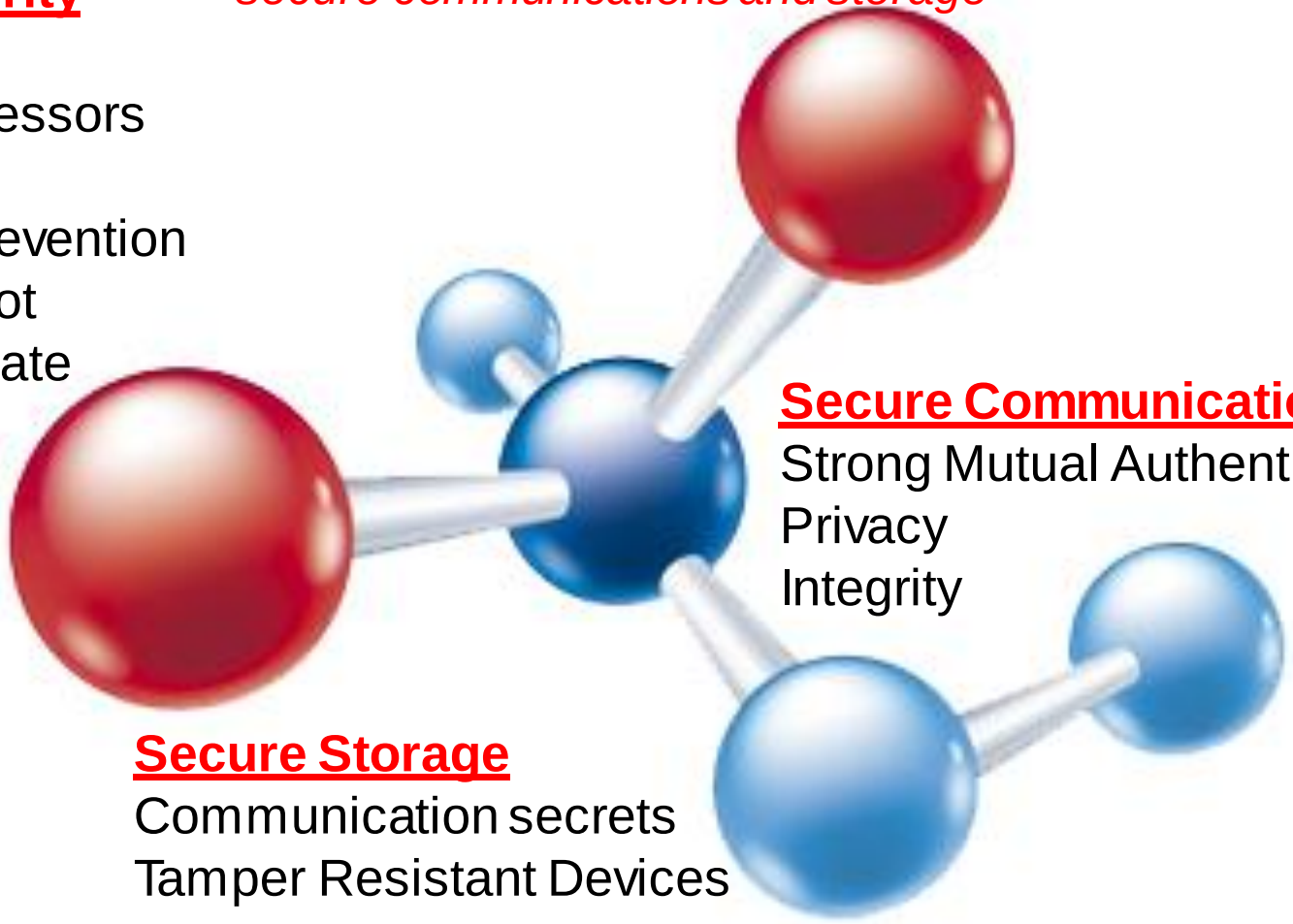
-Multi processors

- Sandbox

Intrusion prevention

-Secure Boot

Secure update



Secure Communication

Strong Mutual Authentication

Privacy

Integrity

Secure Storage

Communication secrets

Tamper Resistant Devices

Exemples d’attaques

Juin 2020

The JSOF research lab has discovered a series of zero-day vulnerabilities(19) in the Treck's TCP/IP library

"The number of devices that contain the vulnerable code base library is only a preliminary estimate; the number may realistically be in the billions."

https://www.jsof-tech.com/ripple20/

The Mirai Malware 2016

How is Mirai infecting devices?
Mirai connects via telnet and attempts to login using a list of 60 known credentials.
If the login is successful, the bot software is installed.

What does the bot then do?
It connects to a command and control server, waiting for commands to attack other machines.
It continuously scans for other devices that may be vulnerable, attempting to login with the list of known credentials.

September 2016. Mirai Malware
145.607 cameras
1 Terabit/s
35,000/50,000 HTTP request/s
25,000 IP adresses
More than 100 countries

The "Moon" worm: ePlug

Passez la souris sur l'image pour zoomer

D-Link DSP-W215/FR Prise intelligente

de D-link

★★★☆☆ ▾ 25 commentaires client | 3 questions ayant reçu une réponse

Prix : EUR 34,99 LIVRAISON GRATUITE Détails

Tous les prix incluent la TVA.

En stock.

Voulez-vous le faire livrer le samedi 16 jan.? Commandez-le dans les **2 h et 34 mins** et choisissez la **Livraison en 1 jour ouvré** au cours de votre commande. En savoir plus.

Expédié et vendu par Amazon. Emballage cadeau disponible.

20 neufs à partir de EUR 34,99 1 d'occasion à partir de EUR 41,77

- Description du produit: D-Link Prise intelligente
- Largeur: 3,93 cm
- Profondeur: 6,6 cm
- Hauteur: 11,7 cm

› Voir plus de détails

The "Moon" worm

Home Network Administration Protocol (**HNAP**) is a proprietary network protocol invented by Pure Networks, Inc. and acquired by Cisco Systems which allows identification, configuration, and management of network devices. HNAP is based on SOAP

2014 HNAP is used by "The Moon" worm which infects Linksys routers.

Hacking the D-Link DSP-W215 Smart Plug
http://www.devttys0.com/2014/05/hacking-the-d-link-dsp-w215-smart-plug/

http://logos.cs.uic.edu/366/notes/mips%20quick%20tutorial.htm

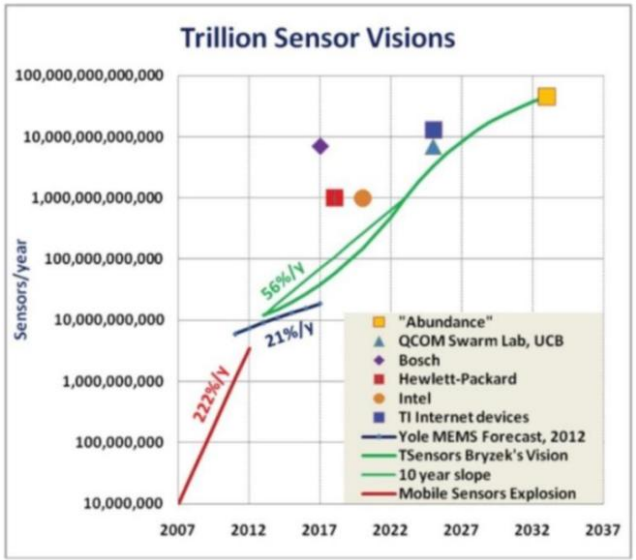
The "Moon" worm

- Two commands are injected thanks to buffer overflow
  - /var/sbin/relay 1 # Turns outlet on
  - /var/sbin/relay 0 # Turns outlet off

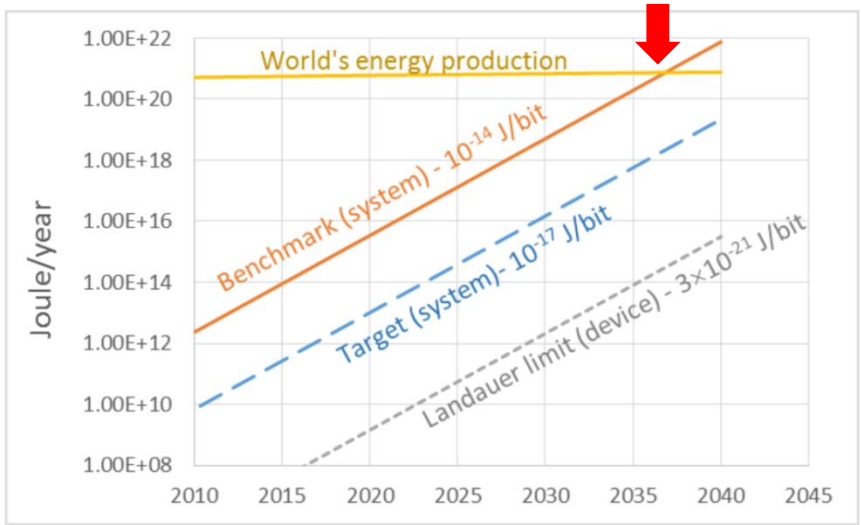
Trillion Sensors

*W=½ Nq x V
q = 1,6 10^{-19}
10^{-14} J == 125,000 electrons

- In current mainstream systems, the lower-edge system-level energy per one bit *transition is ~10^{-14} J, which is referred as the "benchmark".

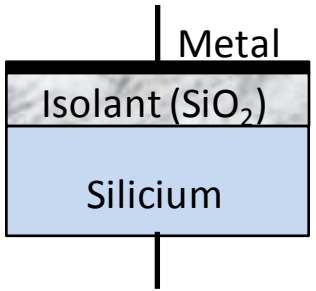


Towards Cyber Physical Systems (CPS)



Eléments Physiques

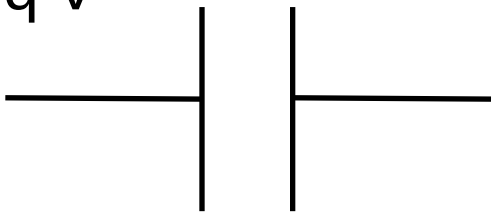
- Permittivité du vide, $\varepsilon 0 = 8{,}854\ 10^{-12}$
- Permittivité du Silicium $\varepsilon = \varepsilon r \times \varepsilon 0 = 11{,}68 \times \varepsilon 0$
- Charge d'un électron $q = 1{,}6\ 10^{-19}$
- Charge commuté (Q_b) de N_e électrons, $Q_b = N_e \times q$
- Energie stockée dans une capacité (en Joule)
- $W_b = \frac{1}{2} \times C_G \times V_H^2 = \frac{1}{2}\ Q_b \times V_H$
- $W_b = \frac{1}{2} \times N_e \times q \times V_H$
- Energie dissipée par une transition d'un électron sous une tension de 1V, $1{,}6\ 10^{-19}$ Joule



$W = \frac{1}{2}\ CV^2$

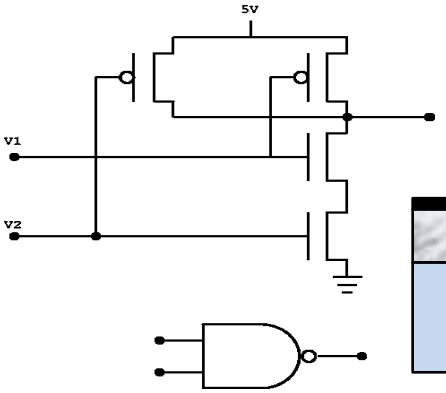
$W = \frac{1}{2}\ Nq\ V$

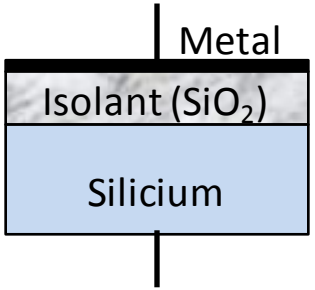
$Q = CV$

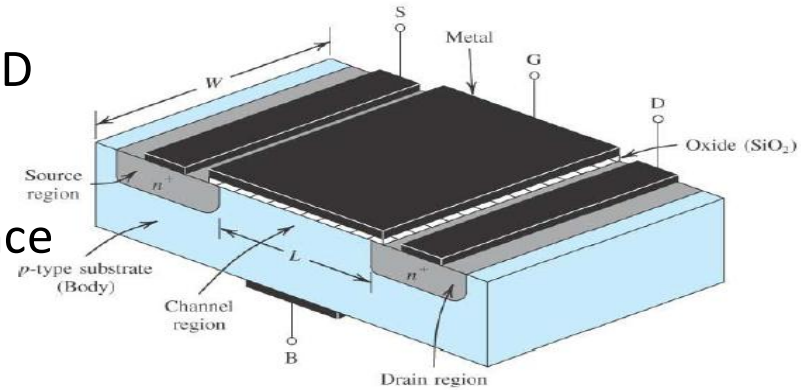


Les portes NAND

- Dimensions d'un transistor
  - Longueur (L) de 1 à 3 µm, largeur (W) de 0,2 à 100 µm, épaisseur de l'isolant (tox) de 2 à 50 nm
  - C = L x W x ε / tox, soit 103 10^{-15} (F) pour L= 1 µm, W=10 µm, tox= 10 nm
    - Sous 1V la charge associée est d'environ 640,000 électrons
- Une commutation (1/0) d'une porte NAND dissipe une énergie W_b
- Si N_c commutations de portes NAND par secondes sont nécessaire alors la puissance (P_c) consommée est égale à
  - $P_c = N_c \times W_b$







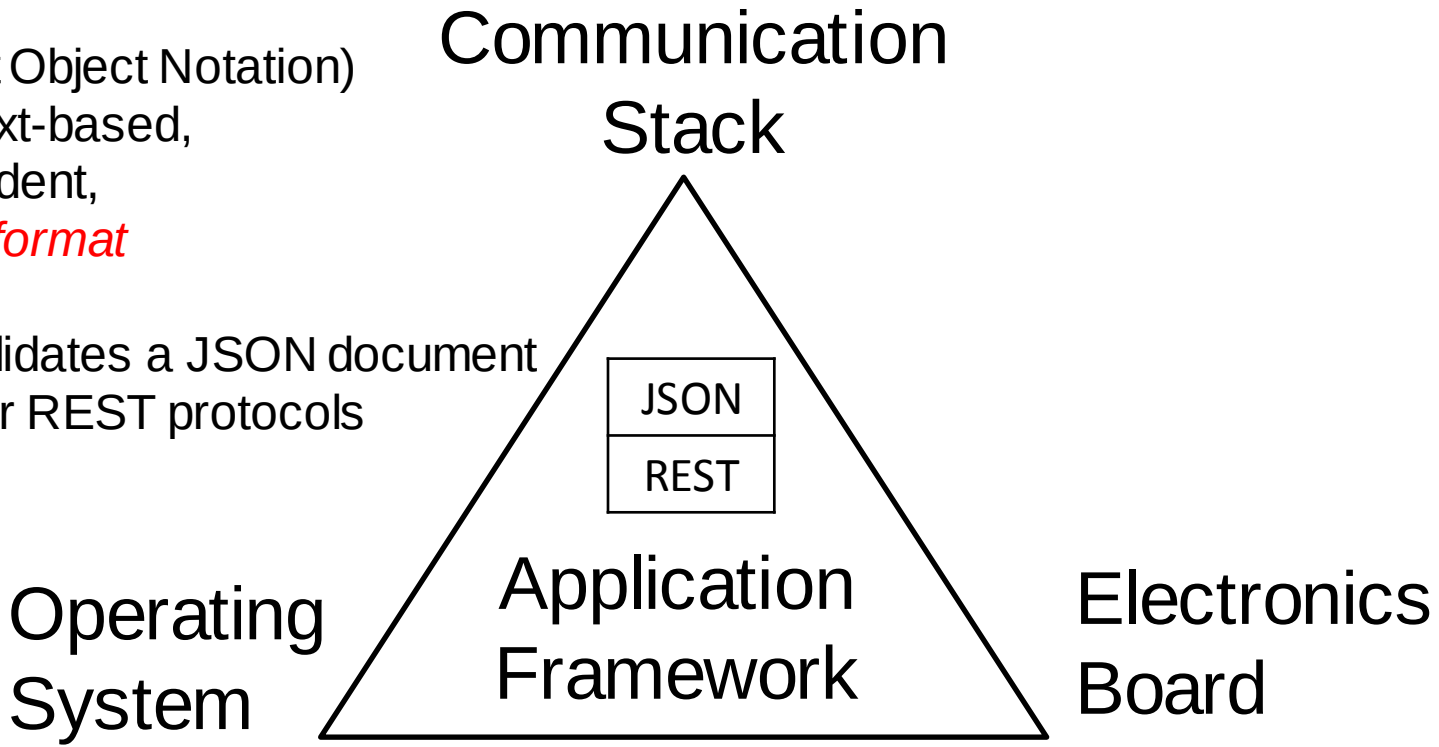
Modèles de Sécurité

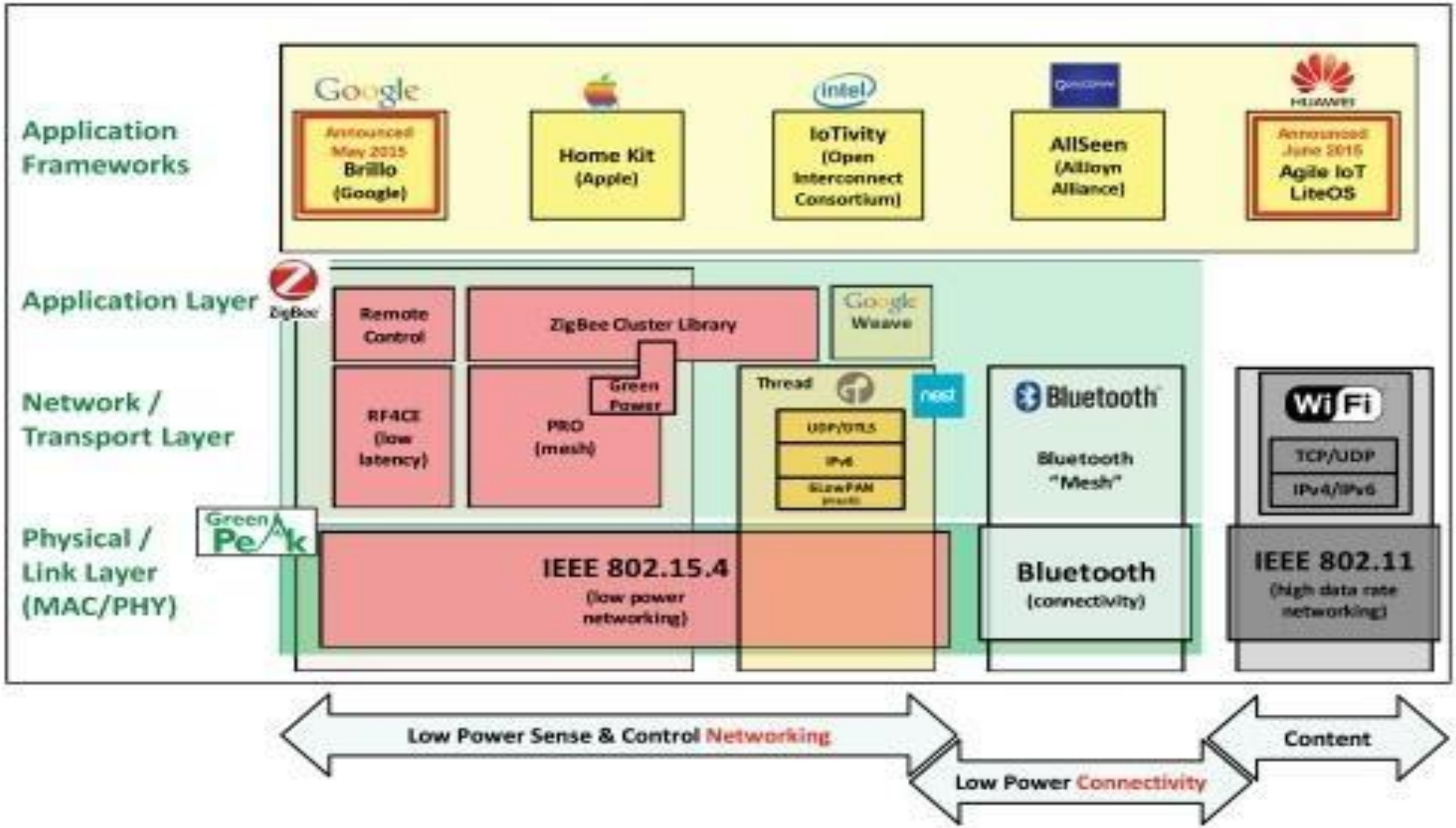
Internet Of Things

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent, *data interchange format*

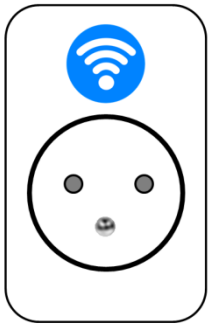
JSON Schema validates a JSON document
JSON is used over REST protocols

Linux, Contiki, Riot, Iotivity, AllJoyn, Brillo, mbed OS ...




Application Frameworks
Google
Announced May 2015
Brillo
(Google)
Home Kit
(Apple)
intel
IoTivity
(Open Interconnect Consortium)
AllSeen
(Alliayn Alliance)
HUAWEI
Announced June 2015
Agile IoT LiteOS
Application Layer
ZigBee
Remote Control
ZigBee Cluster Library
Google Weave
Network / Transport Layer
RF4CE (low latency)
PRO (mesh)
Green Power
Thread
nest
UDP/DTLS
IPv6
6LowPAN
Bluetooth
Bluetooth "Mesh"
WiFi
TCP/UDP
IPv4/IPv6
Physical / Link Layer (MAC/PHY)
Green Peak
IEEE 802.15.4
(low power networking)
Bluetooth
(connectivity)
IEEE 802.11
(high data rate networking)
Low Power Sense & Control Networking
Low Power Connectivity
Content

Sensing Layer



Acqui-
sition
Protocol
DTLS
TLS

Network Layer

Cloud Inter
Network

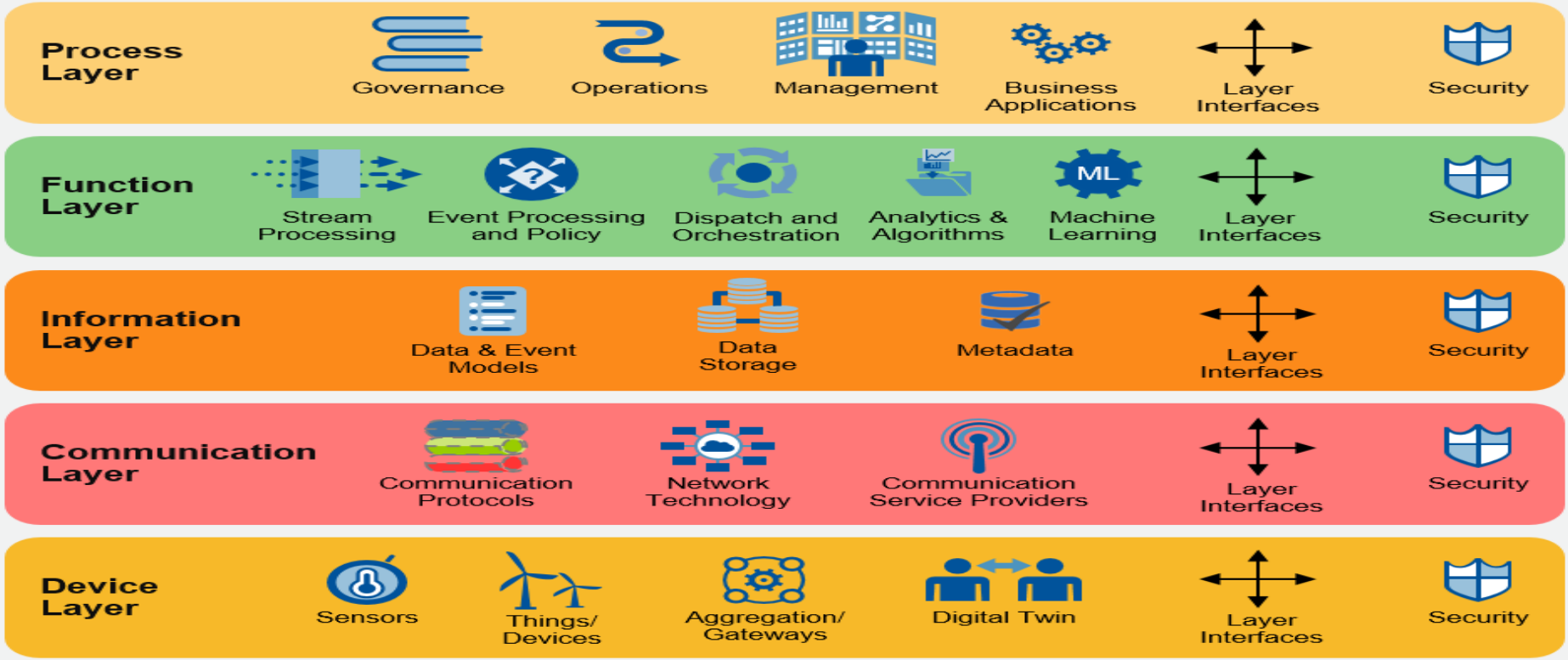
Service Layer

Interface Layer

Application
APIs

Application
Front End

Gartner IoT Reference Model



ID: 354941

- 1) Mots de passe faibles ou non modifiables
  - Utilisation de mots de passe ("*credential*") cassables par des attaques brute force, non modifiables ou publiés, associés à portes dérobées dans le micro logiciel embarqué ou le logiciel client, qui permettent un accès non autorisé aux systèmes déployés (voir par exemple https://www.shodan.io)

- 2) Services réseau non sécurisés
  - Services réseau non utilisés ou non sécurisés s'exécutant sur un objet connecté, qui compromettent la confidentialité, l'intégrité, l'authenticité ou la disponibilité de ses données, ou qui permettent un contrôle à distance non autorisé.

- 3) Interfaces non sécurisées
  - Interfaces WEB, *backend API*, cloud ou mobiles non sécurisées interagissant avec l'objet, et permettant sa compromission ou celle des composants associés. Les faiblesses observées sont un manque d'authentification/autorisation, un chiffrement absent ou faible, l'absence de filtrage des entrées/sorties.

- 4) Absence de mécanisme de mise à jour sécurisée
  - Par exemple absence de validation du micrologiciel sur l'appareil, procédure de livraison non sécurisée ("*supply chain attack*"), possibilité de rejeu de versions antérieures (*roolback*), et manque de notifications des mises à jour.

- 5) Utilisation de composants non sécurisés ou obsolètes
  - Utilisation de composants de logiciels obsolètes ou non sécurisés. Modification du système d'exploitation, utilisation de logiciels ou de composants matériels tiers provenant d'une chaîne de distribution compromise.

- 6) Protection insuffisante de la vie privée
  - Les informations personnelles de l'utilisateur stockées sur l'objet ou dans l'écosystème associé, sont utilisées de manière non sécurisée, abusive ou sans autorisation.

- 7) Transfert et stockage de données non sécurisés
  - Absence de chiffrement ou de contrôle d'accès aux données sensibles stockées dans l'écosystème, durant le transport ou pendant le traitement.

- 8) Manque d'administration des objets.
  - Absence d'administration de la sécurité sur les objets déployés, y compris la gestion du parc, la gestion des mises à jour, la désactivation, la surveillance des systèmes.

- 9) Paramètres par défaut non sécurisés
  - Les objets ou les systèmes sont livrés avec des paramètres par défaut non sécurisés, ou n'ont pas la capacité d'améliorer la sécurité en interdisant aux opérateurs de modifier les configurations.

- 10) Manque de durcissement hardware
  - Absence de mesures de durcissement hardware (*tamper resistant*), permettant aux attaquants d'obtenir des informations sensibles, ou de prendre le contrôle local de l'objet.

Class 0: RAM<<10KB FLASH<<100KB

- Class 0 devices are very constrained sensor-like motes.
- They are so severely constrained in memory and processing capabilities that most likely they will not have the resources required to communicate directly with the Internet in a secure manner (rare heroic, narrowly targeted implementation efforts notwithstanding).
- Class 0 devices will participate in Internet communications with the help of larger devices acting as proxies, gateways, or servers.
- Class 0 devices generally cannot be secured or managed comprehensively in the traditional sense.
- They will most likely be preconfigured (and will be reconfigured rarely, if at all) with a very small data set. For management purposes, they could answer keepalive signals and send on/ off or basic health indications.

•RFC 7228, Terminology for Constrained-Node Networks

Class 1: RAM 10KB, FLASH 100KB

- Class 1 devices are quite constrained in code space and processing capabilities, such that they cannot easily talk to other Internet nodes employing a full protocol stack such as using HTTP, Transport Layer Security (TLS), and related security protocols and XML-based data representations.
- However, they are capable enough to use a protocol stack specifically designed for constrained nodes (such as the Constrained Application Protocol (CoAP) over UDP) and participate in meaningful conversations without the help of a gateway node.
- In particular, they can provide support for the security functions required on a large network. Therefore, they can be integrated as fully developed peers into an IP network, but they need to be parsimonious with state memory, code space, and often power expenditure for protocol and application usage.

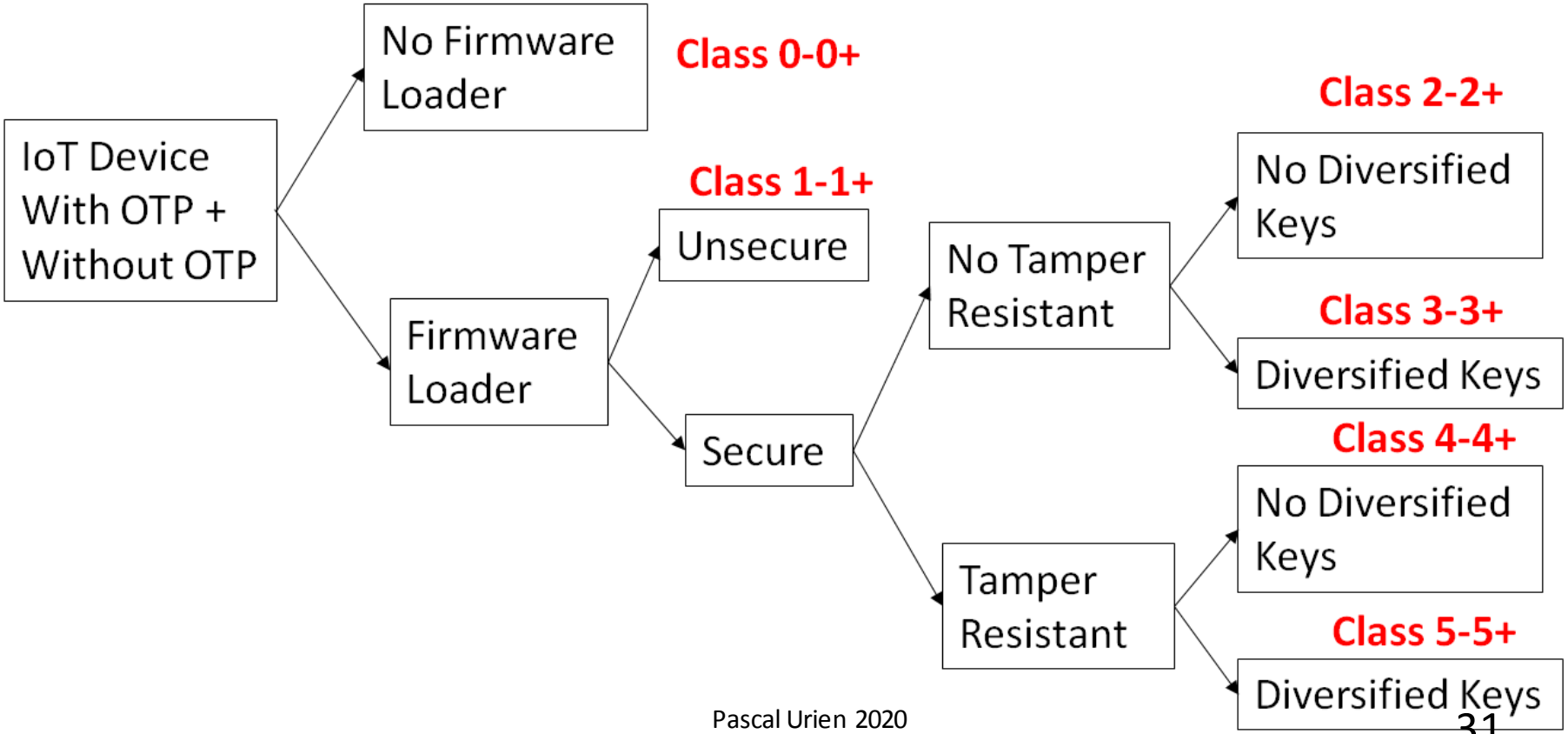
•RFC 7228, Terminology for Constrained-Node Networks

CLASS 2: RAM 50KB, FLASH 250KB

- Class 2 devices are less constrained and fundamentally capable of supporting most of the same protocol stacks as used on notebooks or servers.
- However, even these devices can benefit from lightweight and energy-efficient protocols and from consuming less bandwidth.
- Furthermore, using fewer resources for networking leaves more resources available to applications.
- Thus, using the protocol stacks defined for more constrained devices on Class 2 devices might reduce development costs and increase the interoperability.

•RFC 7228, Terminology for Constrained-Node Networks

Security Classes



IoT Functionality Attack

- The paper* introduces "*a new taxonomy of attacks on IoT devices, which is based on how the attacker deviates feature from their official functionality*". It defines four types of attacking behavior,
  - 1) Ignoring the functionality,
  - 2) Reducing the functionality,
  - 3) Misusing the functionality,
  - 4) Extending the functionality.

*Extended Functionality Attacks on IoT Devices: The Case of Smart Lights, (Invited Paper), Eyal Ronen, Adi Shamir

Our experimental setup includes two main parts:

1) A transmitting setup that includes a smart LED light bulb, and a controller connected to a PC running our software. Both of them are standard unmodified smart light components.
2) A receiving setup that includes a laptop, light sensor, Arduino board and telescope.

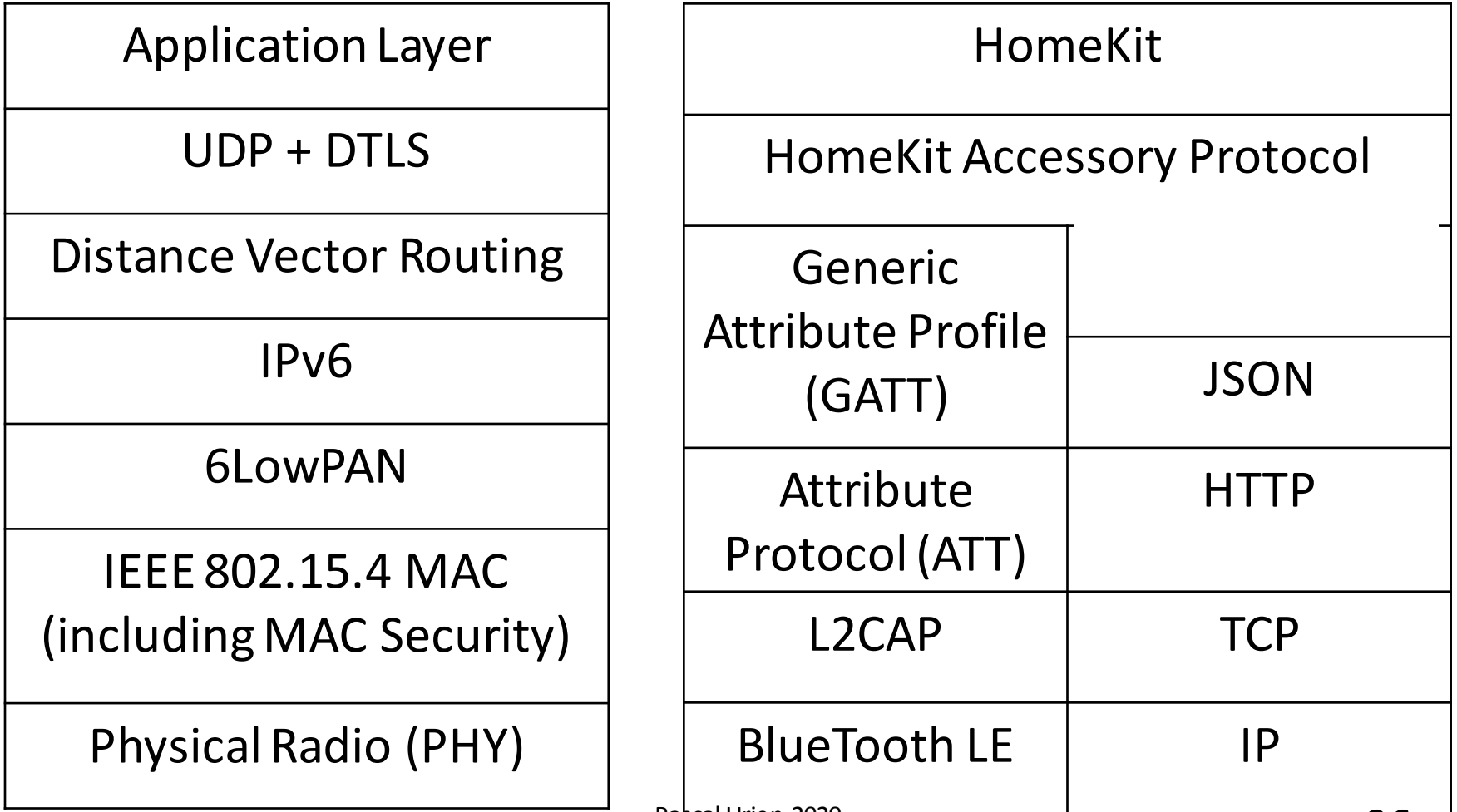
Architecture industrielles

Examples of IoT Systems

- Thread
  - 6LowPAN, DTLS+Password, Commissionner-Joiner architecture, supported by NEST boards
- Open Connectivity Foundation (OCF)
  - 6LowPAN, DTLS+Authentication, Access Control List (ACL), REST API, Iotivity framework
- MBED stack from the ARM company
  - IPv4, 6LoPAN, TLS/DTLS, HTTP, CoAP, MQTT, LWM2M. IBM KIT
- The HAP (*HomeKit Accessory Protocol*) from Apple
  - BlueTooth, Wi-Fi, HTTP, JSON, application security, Secure Remote Password procedure (SRP, RFC 5054).
- Brillo and Weave from Google
  - Brillo is an OS, 35MB footprint. Weave is a communications platform. 802.15.4 (zigbee, threads), BLE, Wi-Fi, Ethernet. HTTPS. Schema Driven (JSON)associates Weave XMPP requests with application function invocations. OAuth 2.0 Authentication, Google as Authentication Server (AS). Intel® Edison Board.
- Philips Hue Bulbs
  - ZigBee Light Link (ZLL). A same link key is shared by all nodes. Bridge with IP/UDP interface.
- Amazon Dash Button
  - Wi-Fi, Bluetooth, HTTPS, Mobile phone as a bridge with AWS
- IKEA TRADFRI
  - ZigBee Light Link (ZLL). Bridge with IP/UDP/DTLS/CoAP/LWM2M iinterface..

In summary: Security for IoT

- MAC level
  - Wi-Fi (IEEE 802.11i, WPA2)
  - Bluetooth Pairing
  - Zibgee (Unique MasterKey + Shared Link Key)
  - Lora (Client AES Key), SigFox (HMAC Client Key)
- TLS/DTLS Stacks
  - IETF CoAP
  - OCF
  - THREAD
- Applicative security
  - HomeKit Accessory Protocol
- Operating System /Bootloader Security
  - Integrity
  - Secure updates
  - Secure Storage

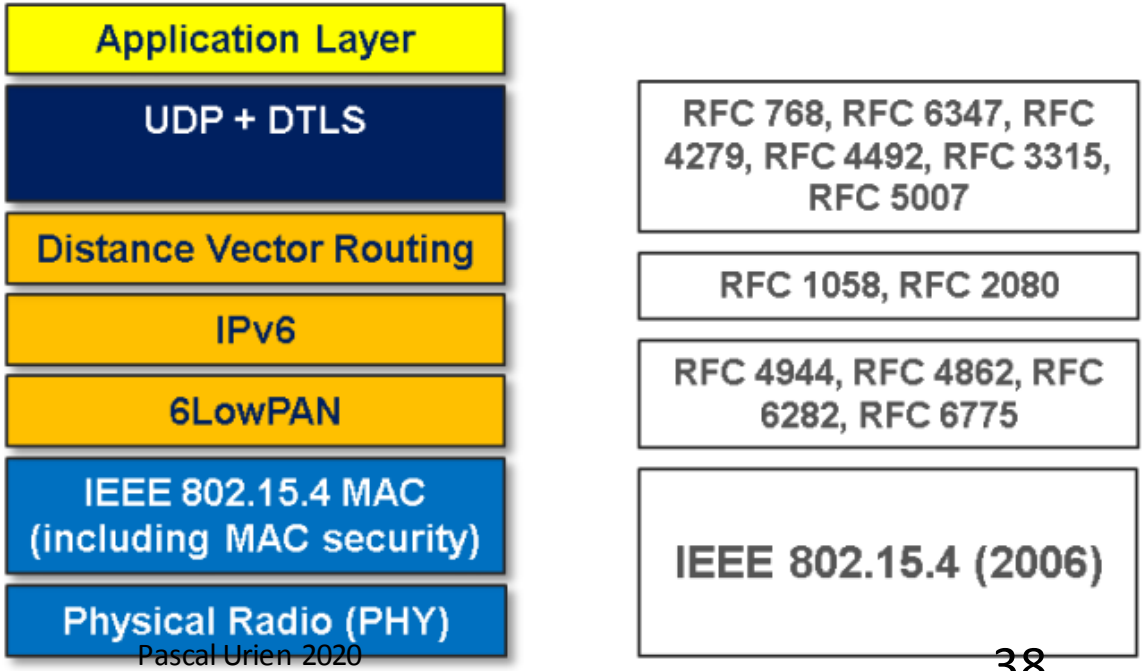

Application Layer
UDP + DTLS
Distance Vector Routing
IPv6
6LowPAN
IEEE 802.15.4 MAC
(including MAC Security)
Physical Radio (PHY)
HomeKit
HomeKit Accessory Protocol
Generic
Attribute Profile
(GATT)
JSON
Attribute
Protocol (ATT)
HTTP
L2CAP
TCP
BlueTooth LE
IP

Example 1. Thread

https://www.threadgroup.org

DTLS + J-PAKE Authentification

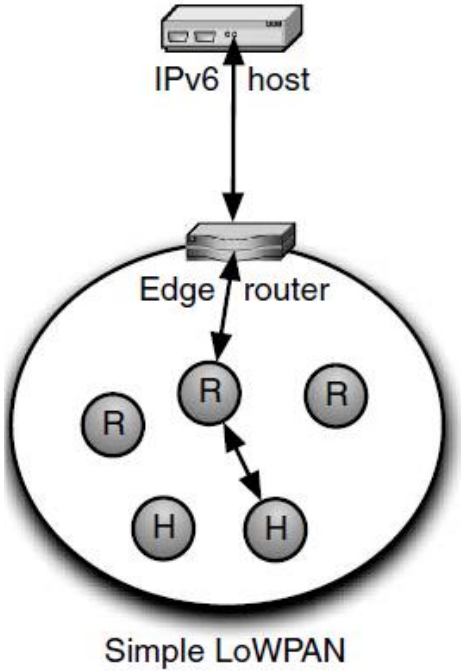
J-PAKE is a password-authenticated key exchange (PAKE) with “juggling” (hence the “J”), see RFC 8236
It essentially uses elliptic curve Diffie-Hellmann for key agreement and Schnorr signatures as a NIZK (Non-Interactive Zero-Knowledge) proof mechanism, see RFC 8235

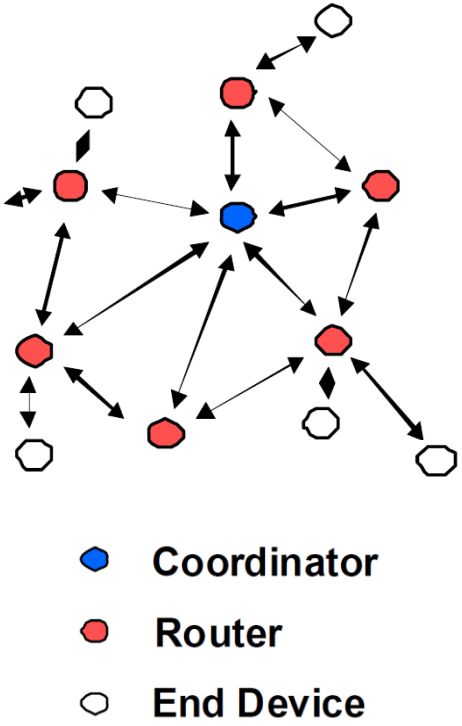


6LoWPAN = IPv6 + Adaptation Layer

IEEE 802.15.4
MAC Frame Size 127 Bytes
IPv6 header 40 Bytes
TCP header 20 Bytes

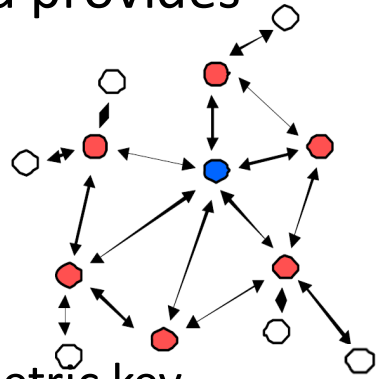
IEEE 802.15.4. Segmentation/Assembly operations are performed by an Adaption Layer and two kinds of routing mechanisms are supported *mesh-under* (performed in the adaptation layer) and *route-over* (performed in the IPv6 layer).





IEEE 802.15.4

- Coordinator is assumed to be the Trust Center (TC) and provides
  - Cryptographic key establishment
  - Key transport
  - Frame protection
  - Device management
- Cryptographic Keys
  - **Master Key** , basis for long term security used for symmetric key establishment. It is used to keep confidential the Link Keys exchange between two nodes in the Key Establishment Procedure (SKKE).
  - **Link Key**, shared between two network peers for Unicast communication.
  - Network Key, used for broadcast communication security.



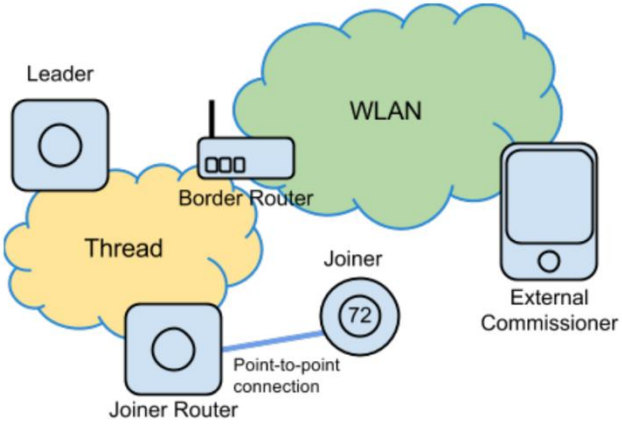
Thread Entities

- Border Router
  - Interface point for the Commissioner when the Commissioner is on a non-Thread Network.
- Commissionner
  - The currently elected authentication server for new Thread devices and the authorizer for providing the network credentials they require to join the network.
- Petitioning
  - The process of authenticating and authorizing a Commissioner Candidate onto the Thread Network through a representative (typically the Border Router).

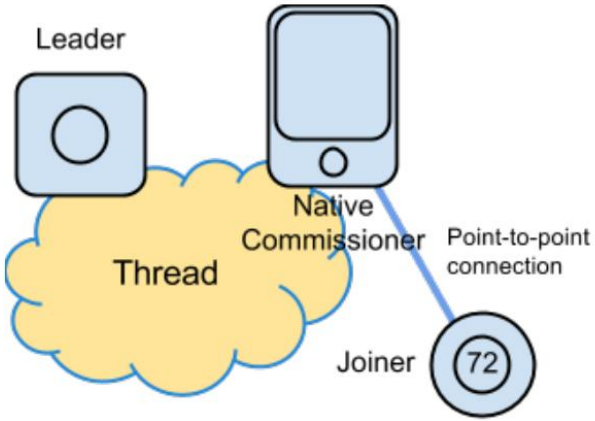
Thread Entities

- Joiner
  - The device to be added by a human administrator to a commissioned Thread Network. The Joiner does not have network credentials.
- Joiner Router
  - An existing Thread router or REED (Router-Eligible End Device) on the secure Thread Network that is one radio hop away from the Joiner.
- KEK
  - Key Establishment Key used to secure delivery of the network-wide key and other network parameters to the Joiner.
- Leader
  - The device responsible for managing router ID assignment.

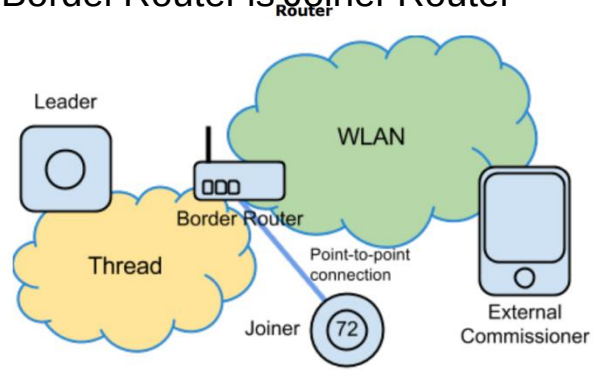
Case 1: External Commissioner connected to the WLAN, Border Router is not Joiner Router



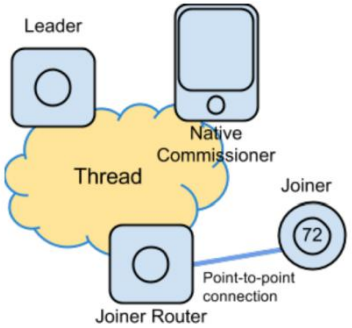
Case 4: Native Commissioner connected to Thread Network, Joiner Router is Commissioner



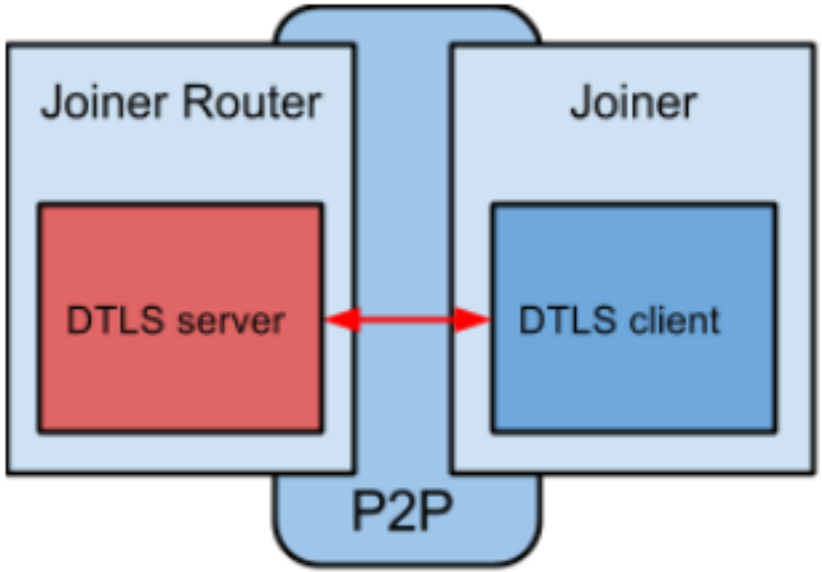
Case 2: External Commissioner connected to the WLAN, Border Router is Joiner Router

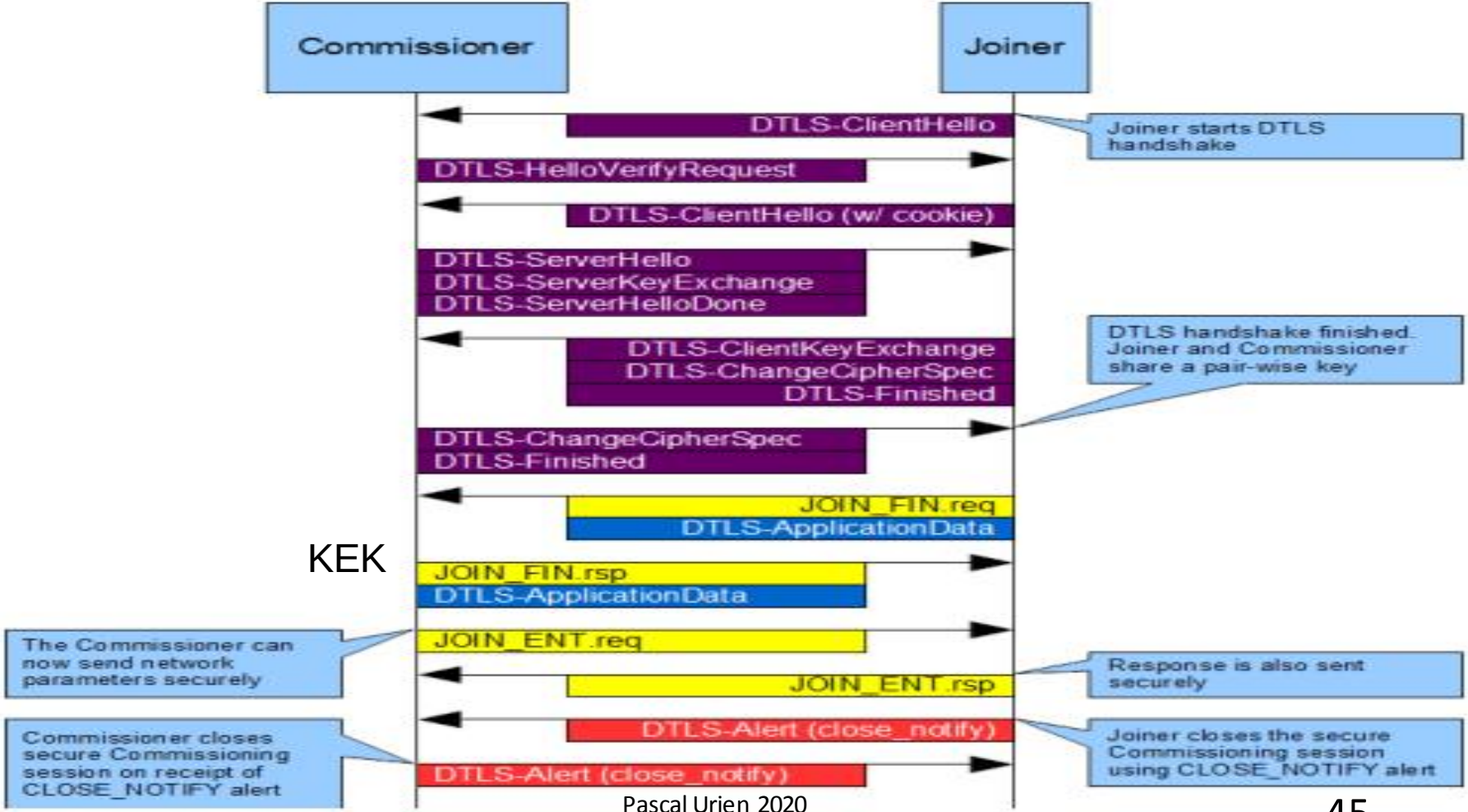


Case 3: Native Commissioner connected to the Thread Network, Joiner Router is not Commissioner

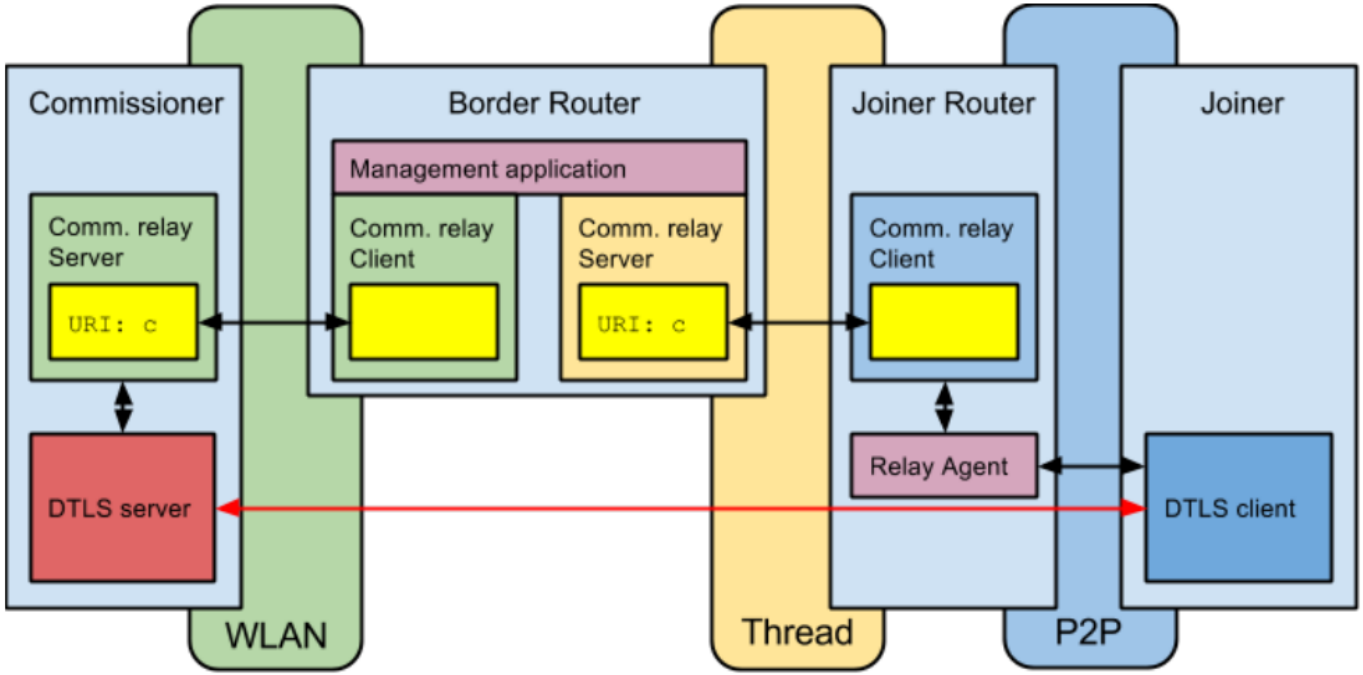


Joiner Router Is Commissioner



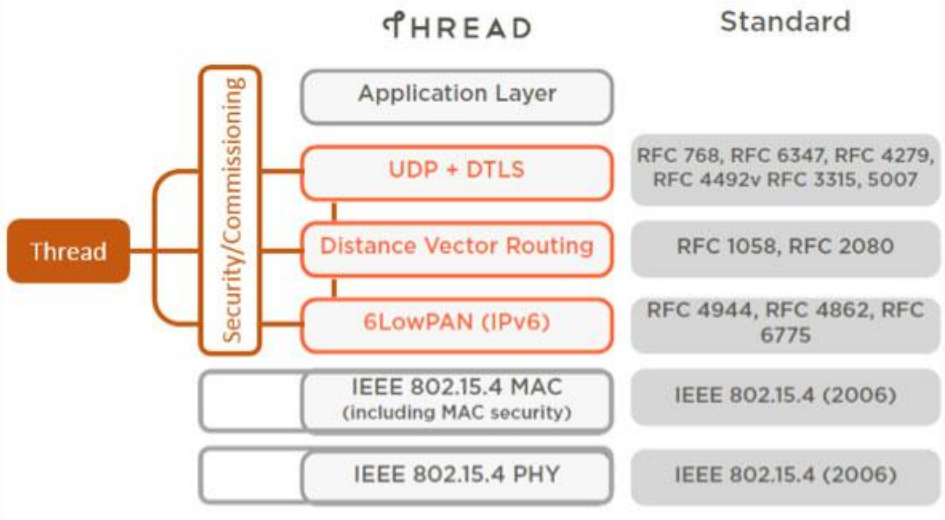

Commissioner
Joiner
DTLS-ClientHello
Joiner starts DTLS handshake
DTLS-HelloVerifyRequest
DTLS-ClientHello (w/ cookie)
DTLS-ServerHello
DTLS-ServerKeyExchange
DTLS-ServerHelloDone
DTLS-ClientKeyExchange
DTLS-ChangeCipherSpec
DTLS-Finished
DTLS handshake finished. Joiner and Commissioner share a pair-wise key
DTLS-ChangeCipherSpec
DTLS-Finished
JOIN_FIN.req
DTLS-ApplicationData
KEK
JOIN_FIN.rsp
DTLS-ApplicationData
The Commissioner can now send network parameters securely
JOIN_ENT.req
JOIN_ENT.rsp
Response is also sent securely
DTLS-Alert (close_notify)
Commissioner closes secure Commissioning session on receipt of CLOSE_NOTIFY alert
Joiner closes the secure Commissioning session using CLOSE_NOTIFY alert
DTLS-Alert (close_notify)

Joiner–Joiner Router–Border Router–Commissioner



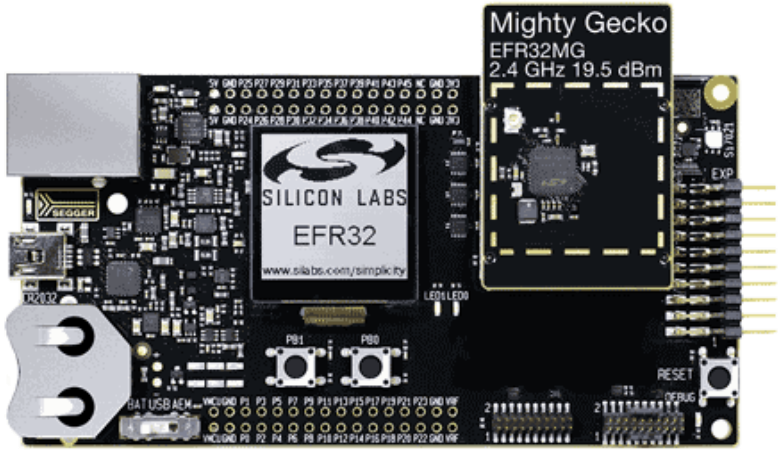

Commissioner
Border Router
Joiner Router
Joiner
Joiner Router has no knowledge of the Joiner at this stage and therefore passes the DTLS handshake through to the Border Router, i.e. the device acting on behalf of the Commissioner
Agent at Border Router passes through the DTLS handshake to the Commissioner
Commissioner acts as authenticator back-end. Existing secure commissioning session is used to relay DTLS handshake
DTLS-ClientHello
RLY_RX.ntf
DTLS-ApplicationData
DTLS-ClientHello
RLY_RX.ntf
DTLS-ClientHello
Joiner starts DTLS handshake
Address and port details of the Joiner and Joiner Router are added, identifying a return path
DTLS-HelloVerifyRequest
RLY_TX.ntf
DTLS-ApplicationData
DTLS-HelloVerifyRequest
RLY_TX.ntf
DTLS-HelloVerifyRequest
DTLS-ClientHello (w/ cookie)
DTLS-ClientHello (w/ cookie)
RLY_RX.ntf
DTLS-ClientHello (w/ cookie)
RLY_RX.ntf
DTLS-ApplicationData
DTLS-ServerHello
DTLS-ServerKeyExchange
DTLS-ServerHelloDone
RLY_TX.ntf
DTLS-ApplicationData
DTLS-ServerHello
DTLS-ServerKeyExchange
DTLS-ServerHelloDone
RLY_TX.ntf
DTLS-ServerHello
DTLS-ServerKeyExchange
DTLS-ServerHelloDone
DTLS-ClientKeyExchange
DTLS-ChangeCipherSpec
DTLS-Finished
RLY_RX.ntf
DTLS-ClientKeyExchange
DTLS-ChangeCipherSpec
DTLS-Finished
RLY_RX.ntf
DTLS-ApplicationData
DTLS handshake finished server side. Joiner and Commissioner share a pair-wise key
DTLS handshake finished client side. Joiner and Commissioner share a pair-wise key
DTLS-ChangeCipherSpec
DTLS-Finished
RLY_TX.ntf
DTLS-ApplicationData
DTLS-ChangeCipherSpec
DTLS-Finished
RLY_TX.ntf
DTLS-ChangeCipherSpec
DTLS-Finished
JOIN_FIN.req
DTLS-ApplicationData
JOIN_FIN.req
DTLS-ApplicationData
RLY_RX.ntf
JOIN_FIN.req
DTLS-ApplicationData
RLY_RX.ntf
DTLS-ApplicationData
Pair-wise key is sent to Border Router using secure Commissioning session
JOIN_FIN.rsp
DTLS-AppData
KEK TLV
RLY_TX.ntf
DTLS-ApplicationData
JOIN_FIN.rsp
DTLS-AppData
KEK TLV
RLY_TX.ntf
Border Router sends pair-wise key to Joiner Router
JOIN_FIN.rsp
DTLS-ApplicationData
JOIN_ENT.req
On receipt of the pair-wise key associated with the Joiner, the Joiner Router can now send network parameters securely
JOIN_ENT.rsp
Response is also sent securely
DTLS-Alert (close_notify)
Joiner closes the secure Commissioning session using CLOSE_NOTIFY alert
DTLS-Alert (close_notify)
RLY_RX.ntf
DTLS-Alert (close_notify)
RLY_RX.ntf
DTLS-ApplicationData
Commissioner closes secure Commissioning session on receipt of CLOSE_NOTIFY alert
DTLS-Alert (close_notify)
RLY_TX.ntf
DTLS-ApplicationData
DTLS-Alert (close_notify)
RLY_TX.ntf
DTLS-Alert (close_notify)

THREAD BOARD

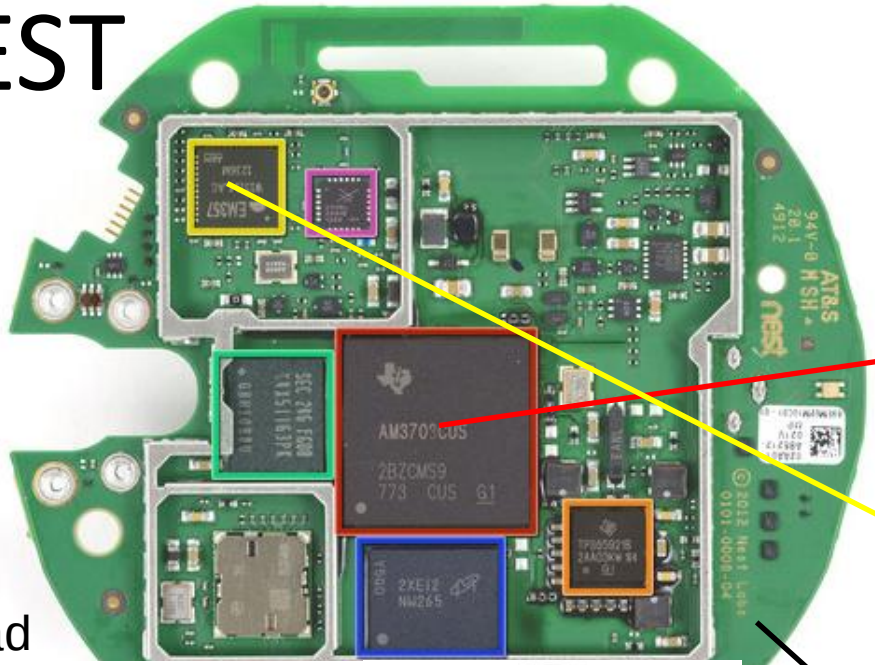


SILICON LABS

http://www.silabs.com/



NEST

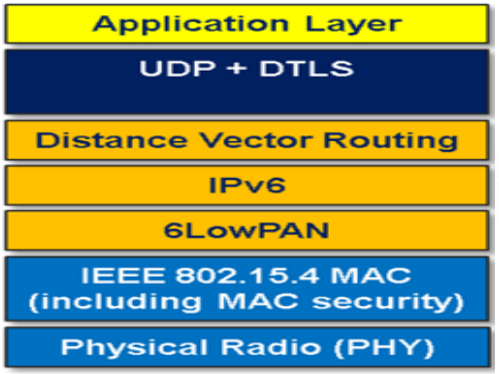


Step 15

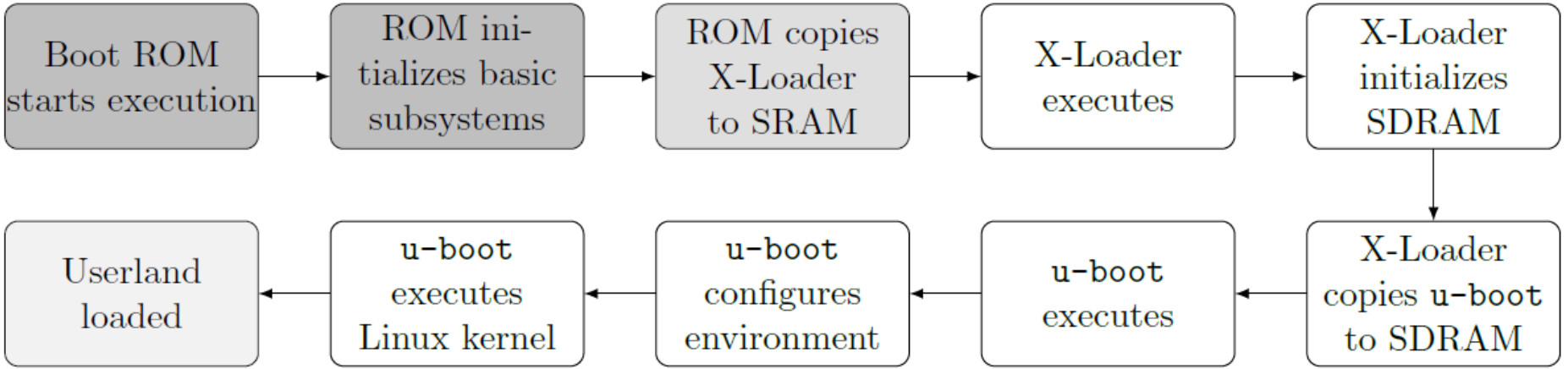
Edit

- With all of the I/O connections on the back, the main motherboard houses all of its important ICs on the front:
- Texas Instruments AM3703CUS Sitara ARM Cortex A8 microprocessor
- Texas Instruments TPS65921B power management and USB single chip
- Samsung K4X51163PK 512 Mb mobile DRAM
- Ember EM357 integrated ZigBee/802.15.4 system-on-chip
- Micron MT29F2G16ABBEAH4 2 Gb NAND flash memory
- Skyworks 2436L high power 2.4 GHz 802.15.4 front-end module
- And under that last EMI shield: Texas Instruments WL1270B 802.11 b/g/n Wi-Fi solution, just like the one we found in the Kindle Fire

Thread



cal Urien 2020



A global reset of the device can be triggered by pressing its button for about 10 seconds. Among other things, this causes the sys boot 5 pin to go high, triggering peripheral booting. Coincidentally, the sys boot 5 pin is directly exposed in an unpopulated header within the main circuit board, which can be utilized to directly trigger the USB booting behavior. Since the ROM does no cryptographic checks of the code being loaded, it freely executes this code, allowing total control of the device.

"Smart Nest Thermostat: A Smart Spy in Your Home", Grant Hernandez, Orlando Arias, Daniel Buentello, and Yier Jin

Example 2.
Open Connectivity Foundation (OCF)

https://openconnectivity.org/

The **Open Connectivity Foundation (OCF)** is creating a specification and sponsoring an open source project to make this possible.
The OCF sponsors the IoTivity open source project which includes a reference implementation of our specification available under the Apache 2.0 license.

The OCF sponsors the IoTivity open source project which includes a reference implementation of our specification

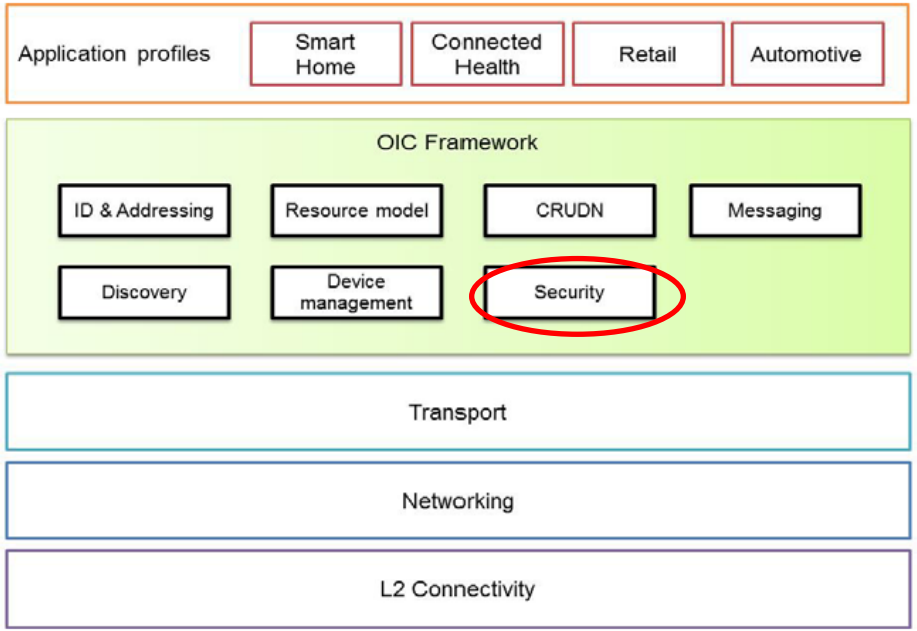


Figure 2: OIC functional block diagram

CRUDN: Create, Read, Update, Delete, Notify OIC: Open Interconnect Consortium

- L2 connectivity: Provides the functionalities required for establishing physical and data link layer connections (e.g., Wi-Fi™ or Bluetooth® connection) to the network.

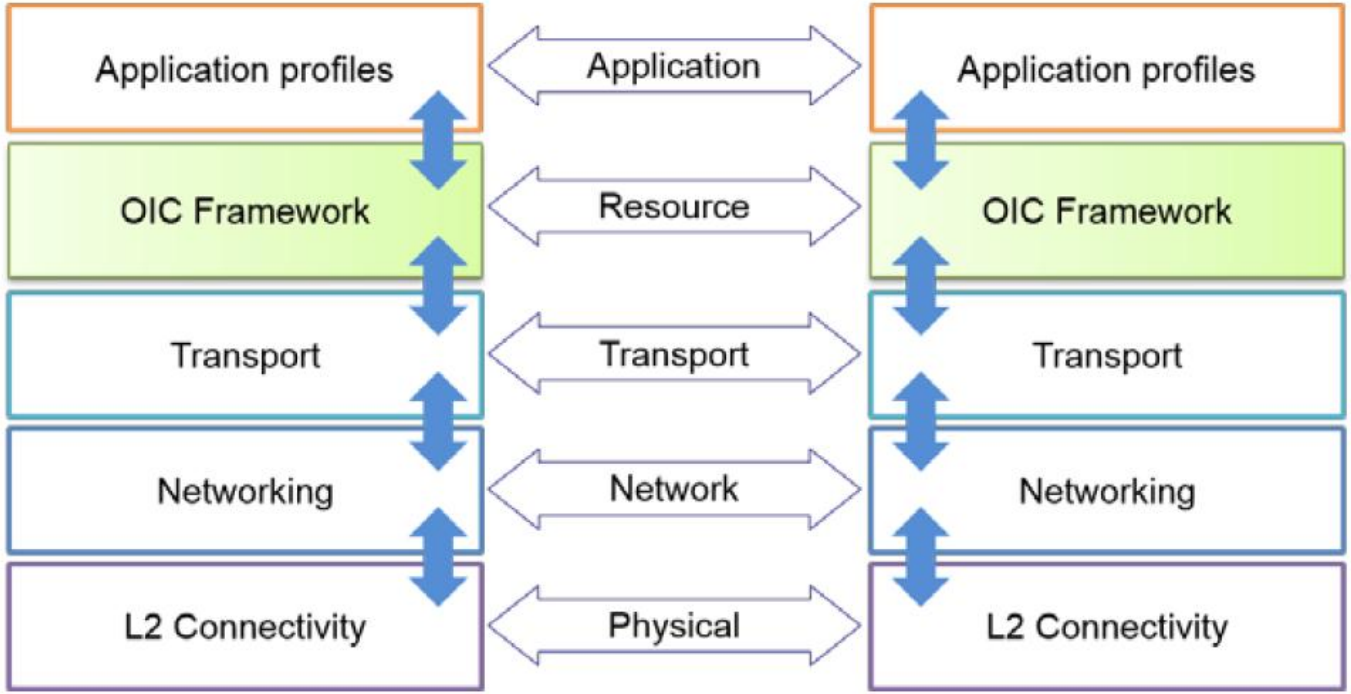
- Networking: Provides functionalities required for Devices to exchange data among themselves over the network (e.g., Internet).

- Transport: Provides end-to-end flow transport with specific QoS constraints. Examples of a transport protocol include TCP and UDP or new Transport protocols under development in the IETF, e.g., Delay Tolerant Networking (DTN).

- OIC Framework: Provides the core functionalities as defined in this specification. The functional block is the source of requests and responses that are the content of the communication between two Devices.

- Application profile: Provides market segment specific data model and functionalities, e.g., smart home data model and functions for the smart home market segment.

OCF Stack



OIC: Open Interconnect Consortium

Security

- Secure Storage
  - It is strongly recommended that IoT device makers provide reasonable protection for Sensitive Data so that it cannot be accessed by unauthorized devices, groups or individuals for either malicious or benign purposes.
  - In addition, since Sensitive Data is often used for authentication and encryption, it must maintain its integrity against intentional or accidental alteration.

Security

- Device Authentication with DTLS
  - Device Authentication with Symmetric Key Credentials
  - Device Authentication with Raw Asymmetric Key Credentials
  - Device Authentication with Certificates

Security

- Secure Boot
  - In order to ensure that all components of a device are operating properly and have not been tampered with, it is best to ensure that the device is booted properly.
  - There may be multiple stages of boot.
  - The end result is an application running on top an operating system that takes advantage of memory, CPU and peripherals through drivers.

Access Control List (ACL)

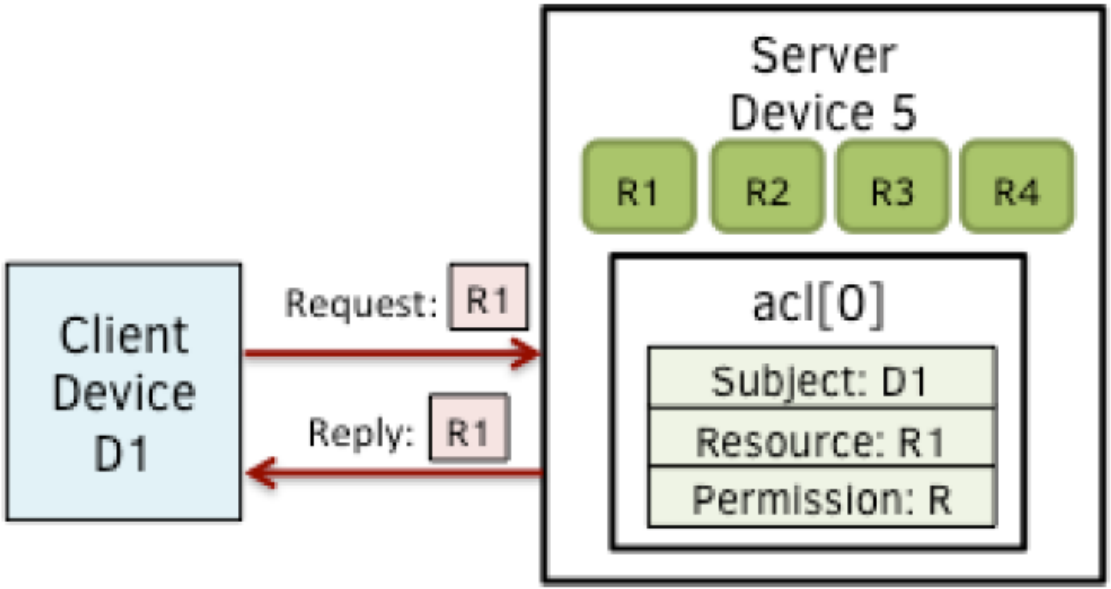
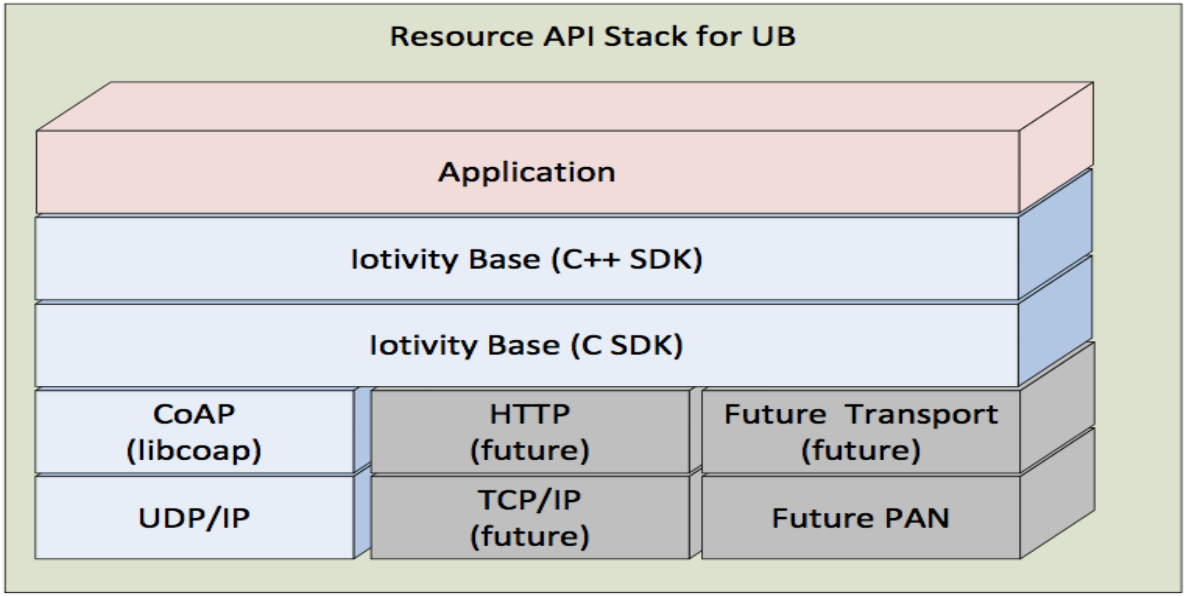


Figure 2 – Use case-1 showing simple ACL enforcement

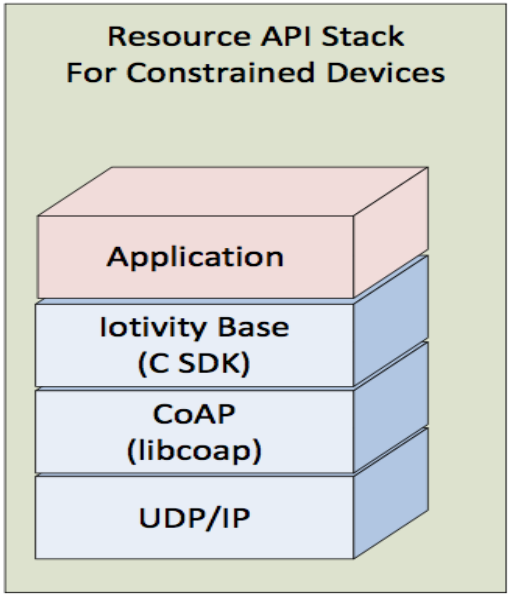
IOTIVITY

https://www.iotivity.org/

Unified Block (UB) stack



Thin Block (TB) stack



IoTivity is an open source software framework enabling seamless device-to-device connectivity to address the emerging needs of the Internet of Things.
It supports multiple operating systems : Linux, Android, Tize, Arduino

Unified Resource Identifier

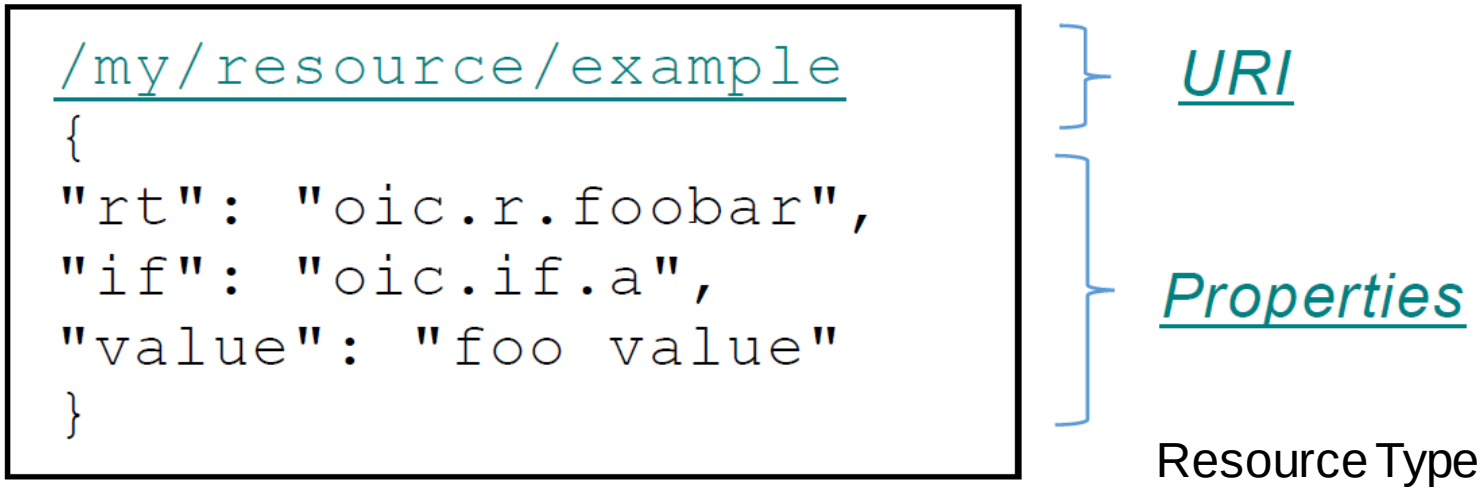
oic://<Authority>/<Path>?<Query>

The usual form of the authority is :
<host>:<port>, where <host> is the name or endpoint network address and <port> is the network port number.

The *path shall be unique string that unambiguously identifies or references a resource within the context of the Server*

A query string shall contain a list of <name>=<value> segments (aka “name-value pair”) each separated by a ‘;’ (semicolon). The query string will be mapped to the appropriate syntax of the protocol used for messaging. (e.g., CoAP).

Resource = URI + Properties



Properties are "key=value" pairs and represent state of the Resource

Resource Type ("rt")

Resource Interface ("if")

Resource Name ("n")

Resource Identity ("id"):

Interface	Name	Applicable Methods	Description
baseline	oic.if.baseline	RETRIEVE, UPDATE	The baseline Interface defines a view into all Properties of a Resource including the Meta Properties. This Interface is used to operate on the full Representation of a Resource.
links list	oic.if.ll	RETRIEVE	The 'links list' Interface provides a view into Links in a Collection (Resource). Since Links represent relationships to other Resources, the links list interfaces may be used to discover Resources with respect to a context. The discovery is done by retrieving Links to these Resources. For example: the Core Resource /oic/res uses this Interface to allow discovery of Resource "hosted" on a Device.
batch	oic.if.b	RETRIEVE, UPDATE	The batch Interface is used to interact with a collection of Resources at the same time. This also removes the need for the Client to first discover the Resources it is manipulating – the Server forwards the requests and aggregates the responses
read-only	oic.if.r	RETRIEVE	The read-only Interface exposes the Properties of a Resource that may be 'read'. This Interface does not provide methods to update Properties or a Resource and so can only be used to 'read' Property Values.
read-write	oic.if.rw	RETRIEVE, UPDATE	The read-write Interface exposes only those Properties that may be both 'read' and "written" and provides methods to read and write the Properties of a Resource.
actuator	oic.if.a	CREATE, RETRIEVE, UPDATE	The actuator Interface is used to read or write the Properties of an actuator Resource.
sensor	oic.if.s	RETRIEVE	The sensor Interface is used to read the Properties of a sensor Resource.

OCF REST

Request: GET /a/act/heater?if="oic.if.a"

Response:
{ "prm": {"sensitivity": 5, "units": "C",
"range": "0 .. 10"},
"settemp": 10,
"currenttemp" : 7
}

Request: POST /a/act/heater?if="oic.if.a "
{ "settemp": 20 }

Response:
{ Ok }

oic://server:port

```
/my/resource/example
{
"rt": "oic.r.foobar",
"if": "oic.if.a",
"value": "foo value"
}
```

URI

Properties

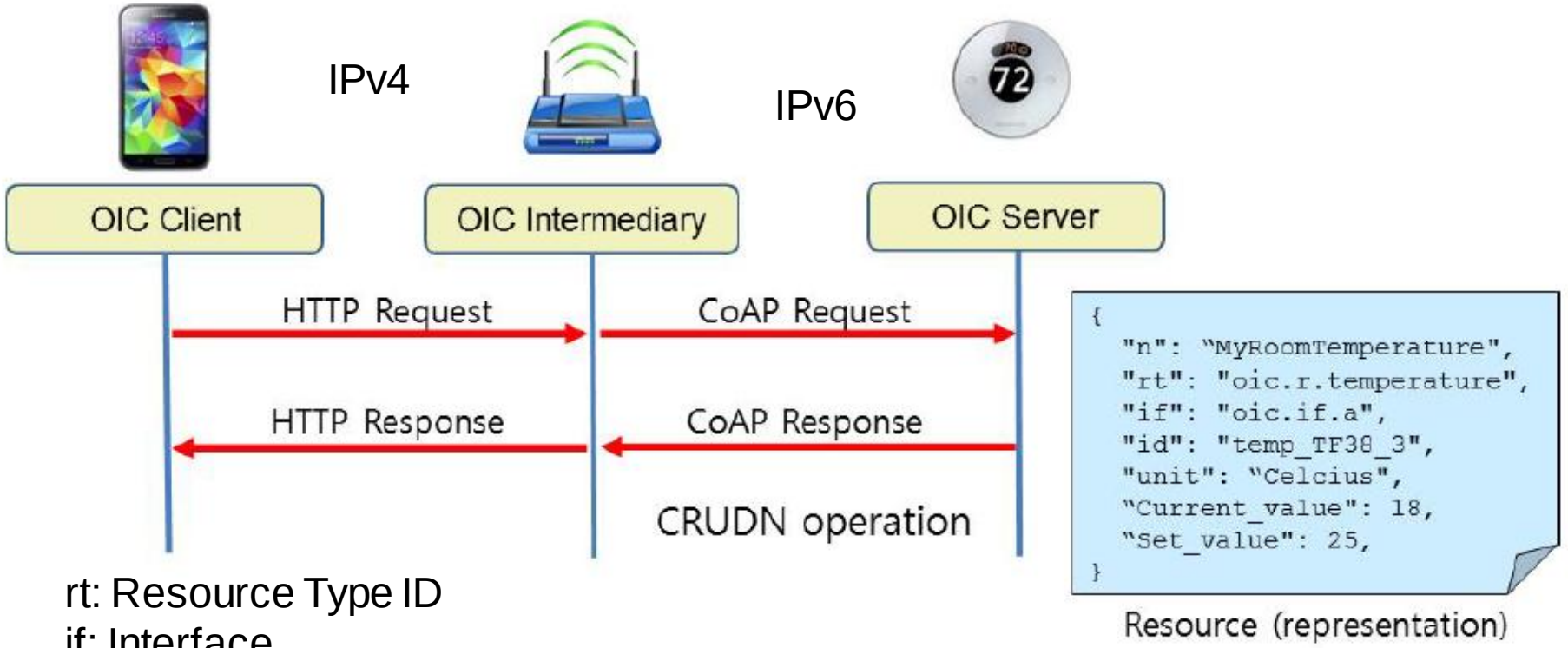
Resource Type ("rt")

(Resource) Interface ("if")

(Resource) Name ("n")

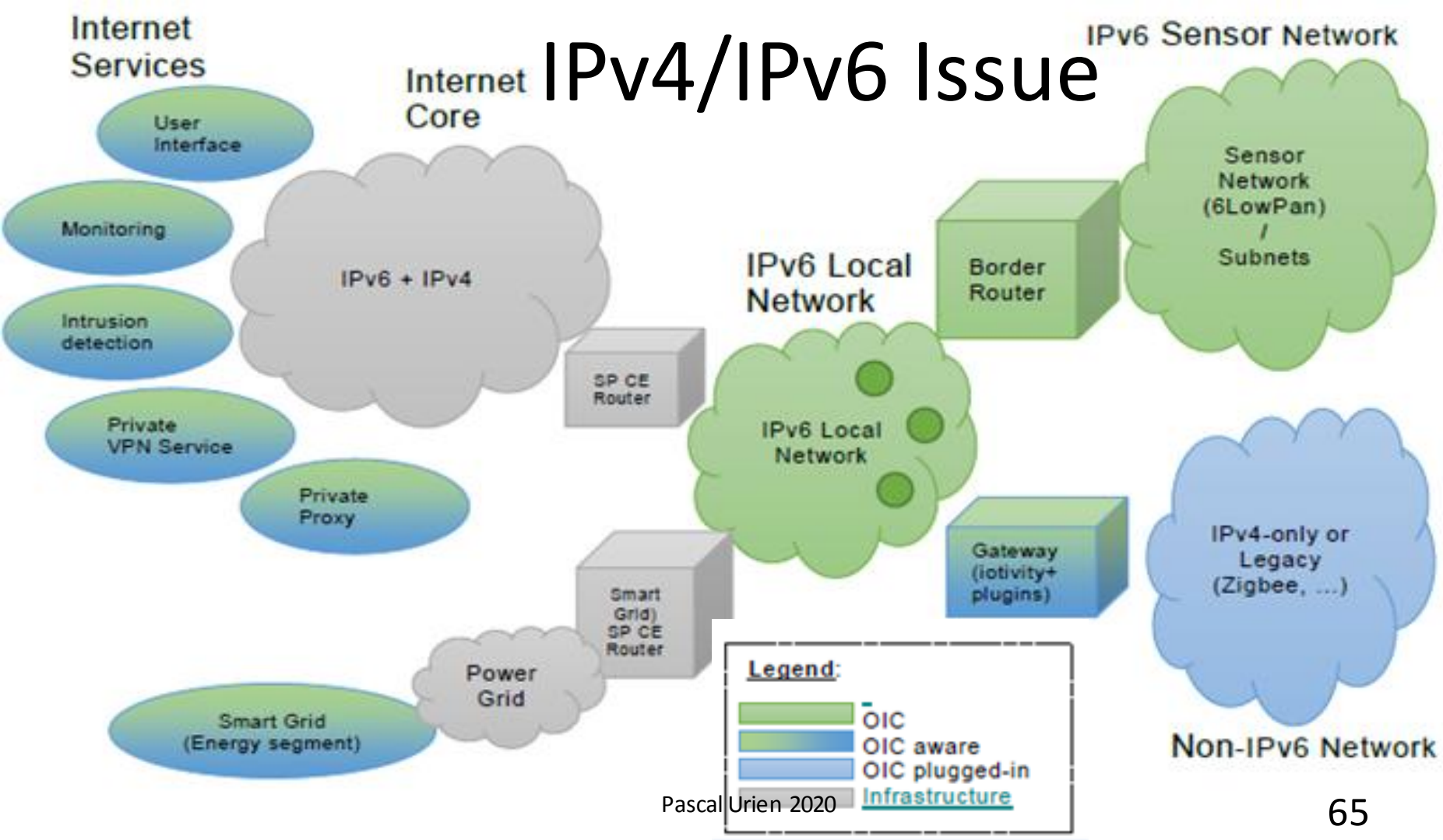
Resource Identity ("id"):

CoAP /HTTP



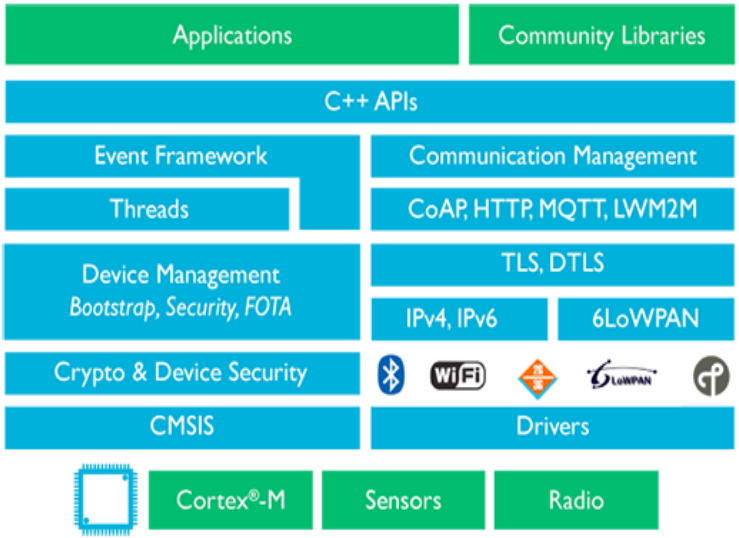
rt: Resource Type ID
if: Interface

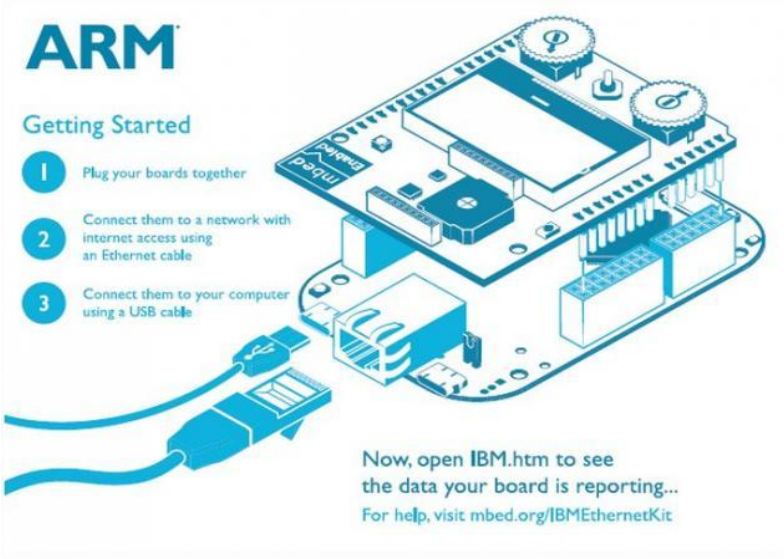
IPv4/IPv6 Issue



Example 3. MBED

MBED stack from the ARM company

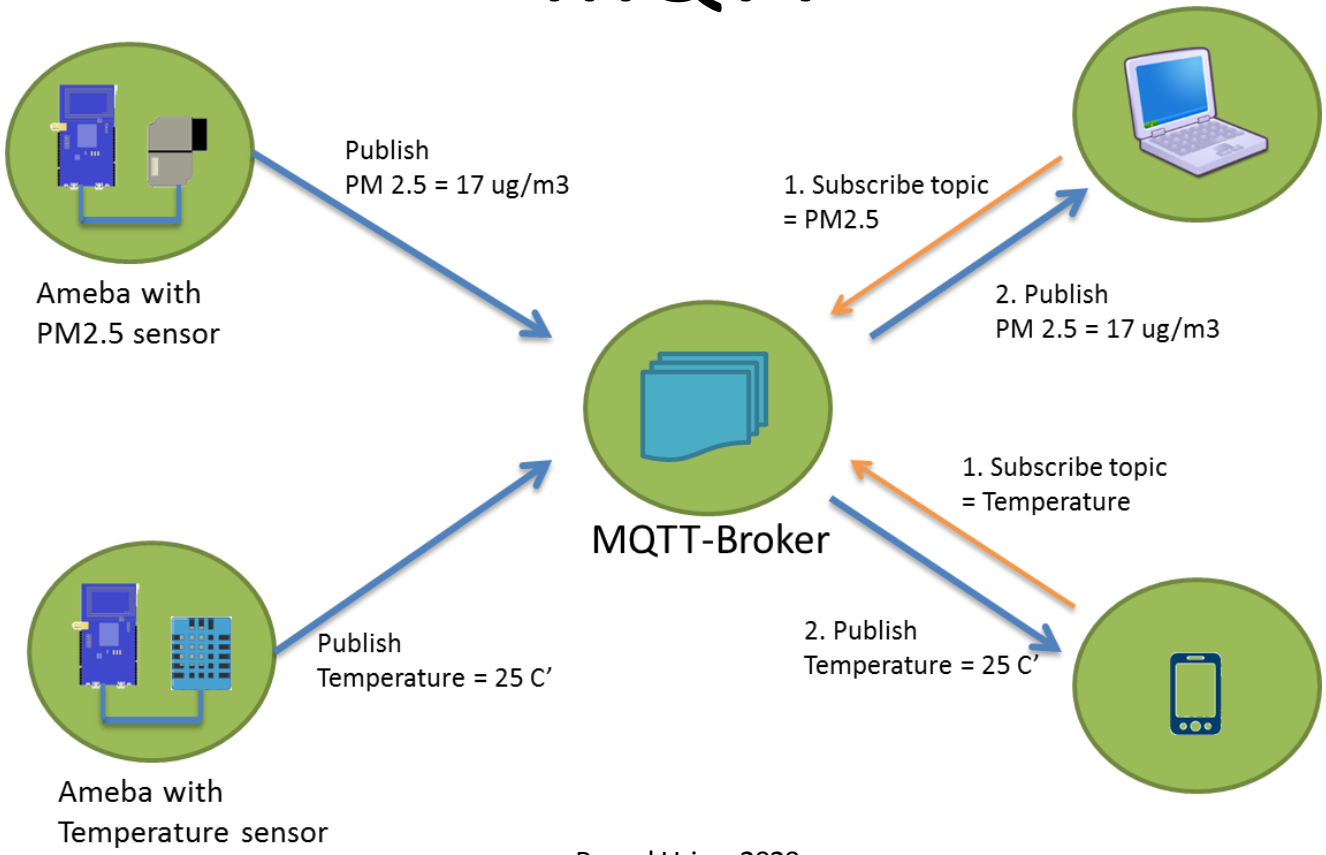




IoT Protocols

- HTTP (most of today IP objects)
  - As an illustration some connected plugs work with the HNAP (*Home Network Administration Protocol*) protocol based on SOAP and used in CISCO routers. In 2014 HNAP was infected by" The Moon".
- MQTT protocol, is a Client Server publish/subscribe messaging transport protocol that is secured by TLS.

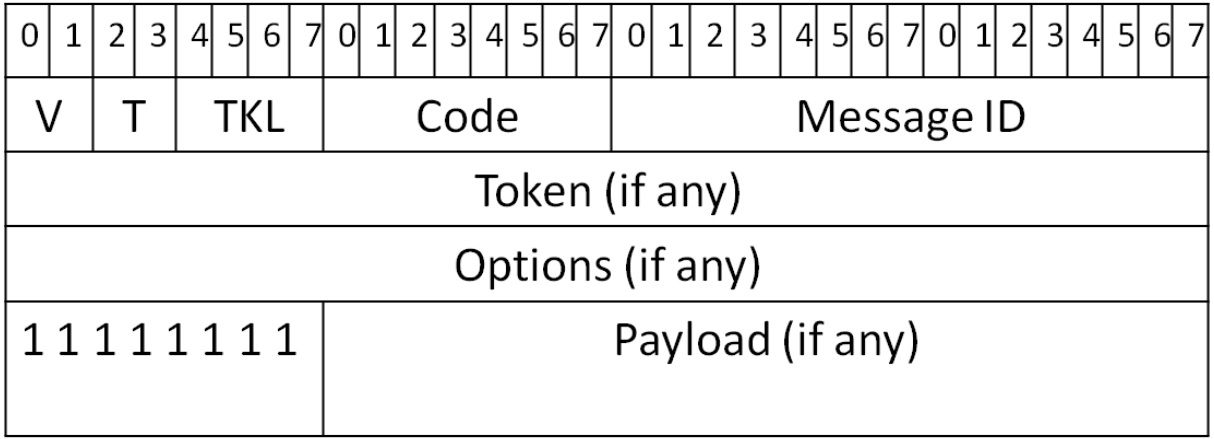
MQTT


Publish
PM 2.5 = 17 ug/m3
1. Subscribe topic
= PM2.5
2. Publish
PM 2.5 = 17 ug/m3
Ameba with
PM2.5 sensor
MQTT-Broker
1. Subscribe topic
= Temperature
2. Publish
Temperature = 25 C'
Publish
Temperature = 25 C'
Ameba with
Temperature sensor

CoAP, RFC 7252

- CoAP (Constrained Application Protocol) , RFC 7252 is designed according to the Representational State Transfer (REST) architecture , which encompasses the following six features:
  - 1) Client-Server architecture;
  - 2) Stateless interaction;
  - 3) Cache operation on the client side;
  - 4) Uniform interface ;
  - 5) Layered system ;
  - 6) Code On Demand.
- CoAP is an efficient RESTfull protocol easy to proxy to/from HTTP, but which is not understood in an IoT context as a general replacement of HTTP.
  - It is natively secured by DTLS (the datagram adaptation of TLS), and works over a DTLS/UDP/IP stack. Nerveless the IETF is currently working on a CoAP version compatible with a TLS/TCP/IP stack.

CoAP Details



Version (V): protocol version (01).

Type (T) message type :

Confirmable (CON), Non-confirmable (NON), Acknowledgement (ACK) or Reset.

Token Length (TKL)/ is the length of the Token field (0-8 bytes).

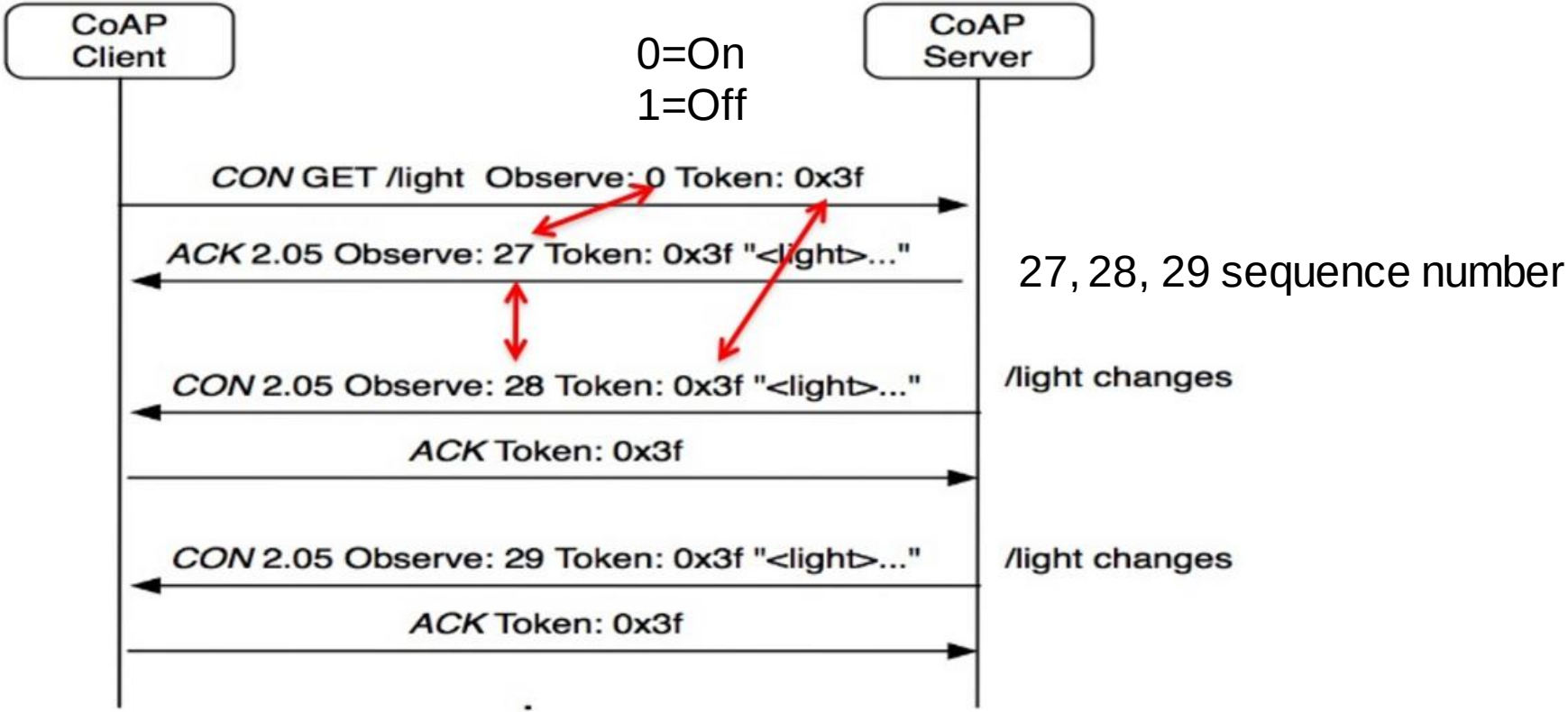
The Code field: identifies the method and is split in two parts a 3-bit class and a 5-bit detail documented as "c.dd" where "c" is a digit from 0 to 7 and "dd" are two digits from 00 to 31. 0.01 GET, 0.02 POST, 0.03 PUT and 0.04 DELETE.

Message ID: matches messages ACK/Reset to messages CON/NON previously sent.

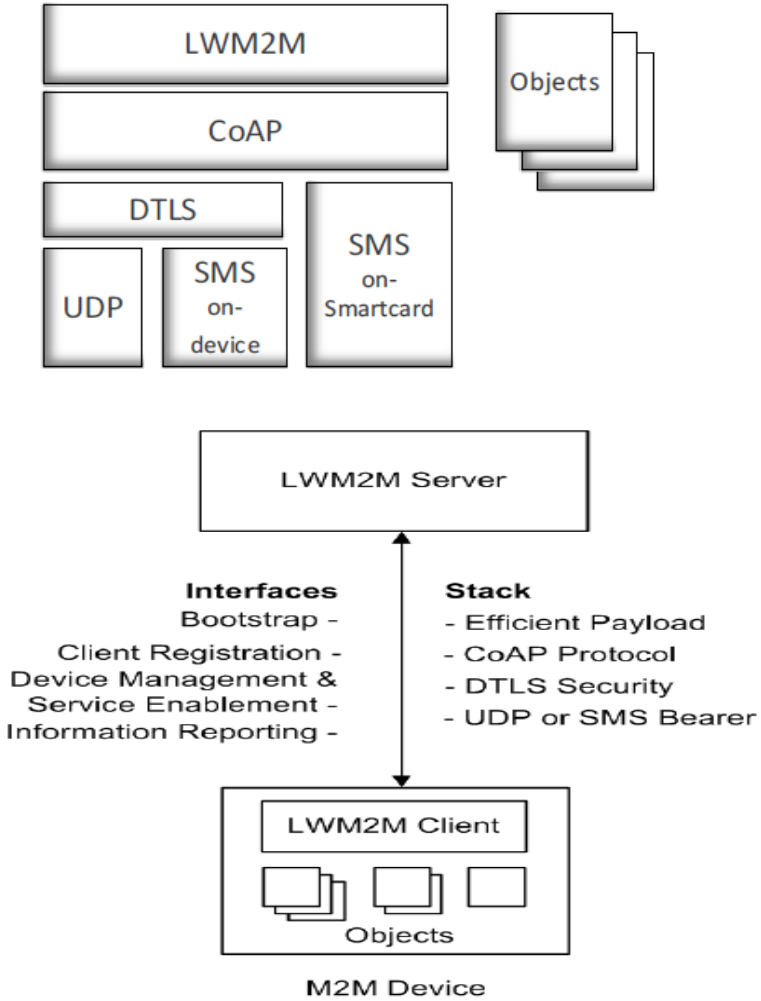
The Token (0 to 8 bytes): is used to match a response with a request.

Options: give additional information such as Content-Format dealing with proxy operations.

Observe option (Observe: int value)



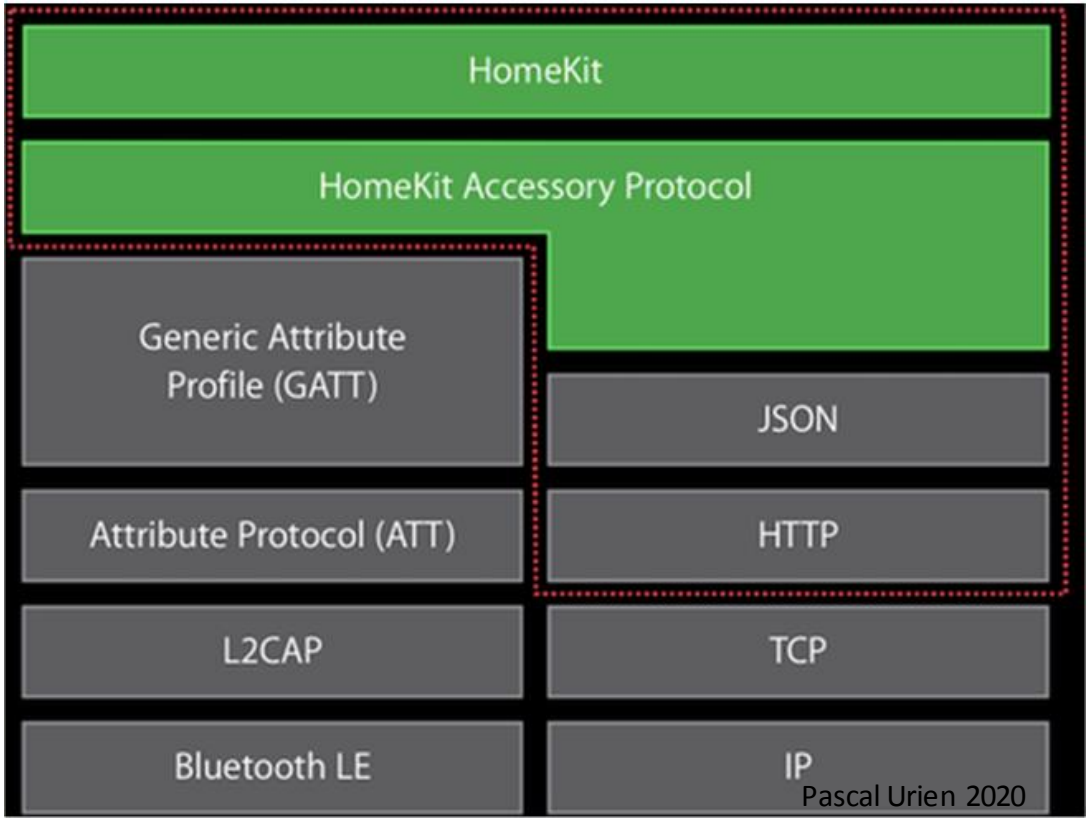
LWM2M



- LWM2M (*Lightweight Machine to Machine Technical Specification)* is a framework based on CoAP dealing with objects hosted by LWM2M clients and communicating with LWM2M servers
- LWM2M manages the following interfaces
  - Bootstrap
  - Client Registration (with servers)
  - Device management
  - Information Reporting
- Two transport mechanism ("*transport channel bindings")*
  - *UDP/IP*
  - *SMS*

Example 4. Home Kit

HOME Kit (Apple)



Protocol Security
- End-to-end encryption
- Initial setup secured directly between iOS and accessory
- Perfect forward secrecy
- Standard cryptography

The HAP (*HomeKit Accessory Protocol*) initial pairing exchange is based on the Secure Remote Password procedure (SRP, RFC 5054) which deals with a 8 digits PIN code available for every accessory.

HAP Security Details

- Secure Remote Password (SRP) Encrypts and authenticates initial pairing key exchange
- Ed25519 Long-term keys for pairing and authentication
- Curve25519 Encrypts initial authentication for each session
- HKDF-SHA-512 Per-session ephemeral encryption key derivation
- ChaCha20-Poly1305 Encrypts and authenticates HAP data

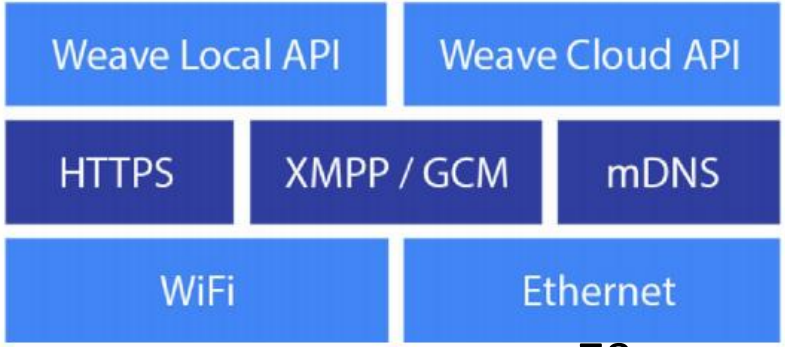
Example 5. Brillo & Weave

Brillo & Weave

The Intel® Edison Board Made for Brillo.

Brillo is an OS from Google for building connected devices.
35MB Memory Footprint (minimum)

Weave is a communications protocol that supports discovery, provisioning, and authentication so that devices can connect and interact with one another, the Internet, and your mobile platforms.



Brillo and Weave

Weave is a communications platform for IoT devices

- Device setup, phone-to-device-to-cloud communication
- User interaction from mobile devices and the web
- Transports: 802.15.4 (zigbee, threads), BLE, Wi-Fi, Ethernet, Others possible
- Schema Driven (JSON) Associates Weave XMPP requests with application function invocations
- Web apps may be written with Google API support
- OAuth 2.0 Authentication, Google as Authentication Server (AS)

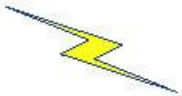
Brillo is Simpler… Smaller…IoT Focused

- C/C++ environment
- Binder IPC No Java Applications, framework, runtime
-No Graphics
- 35MB Memory Footprint (minimum)

Example 6. Philips Hue Bulbs

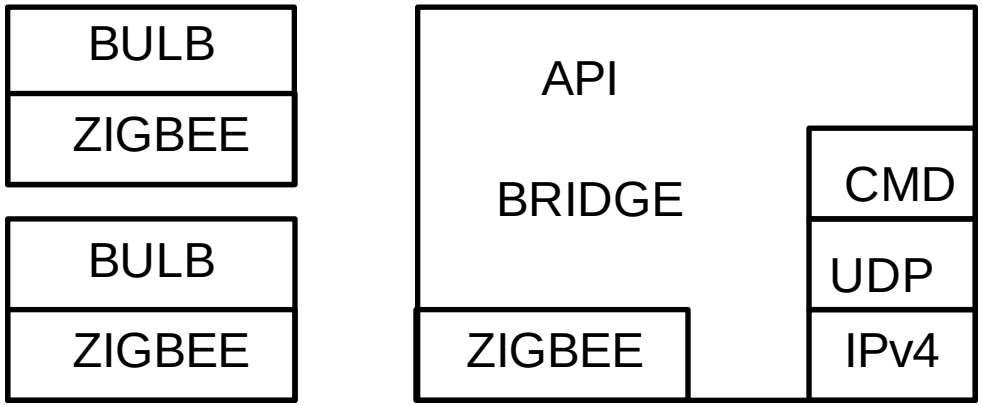
Hue Bridge connects to Home Router using Ethernet Cable

Hue Bridge using Zigbee Wireless network to connect to Hue Bulbs



Smartphone App connects using WiFi

Hue Bulb System



The commands are sent in the UDP payload as a short series of bytes with a termination byte of 0x55. For example, the light on command is:

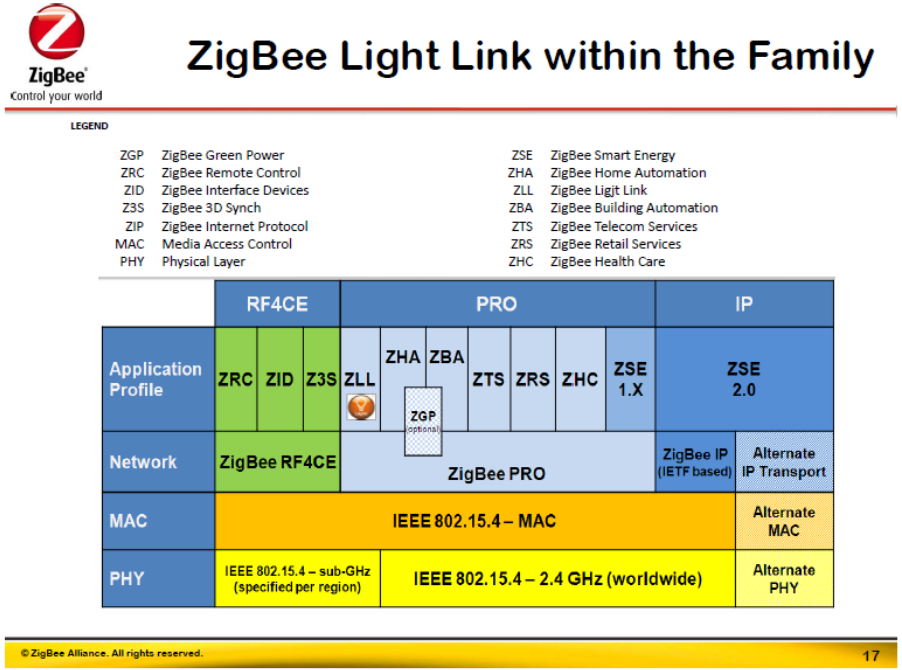
IPHEADER	UDPHEADER	0x45	0x00	0x55

and the light off command is:

IPHEADER	UDPHEADER	0x41	0x00	0x55

Extended Functionality Attacks on IoT Devices: The Case of Smart Lights (Invited Paper), Eyal Ronen, Adi Shamir

http://www.developers.meethue.com/



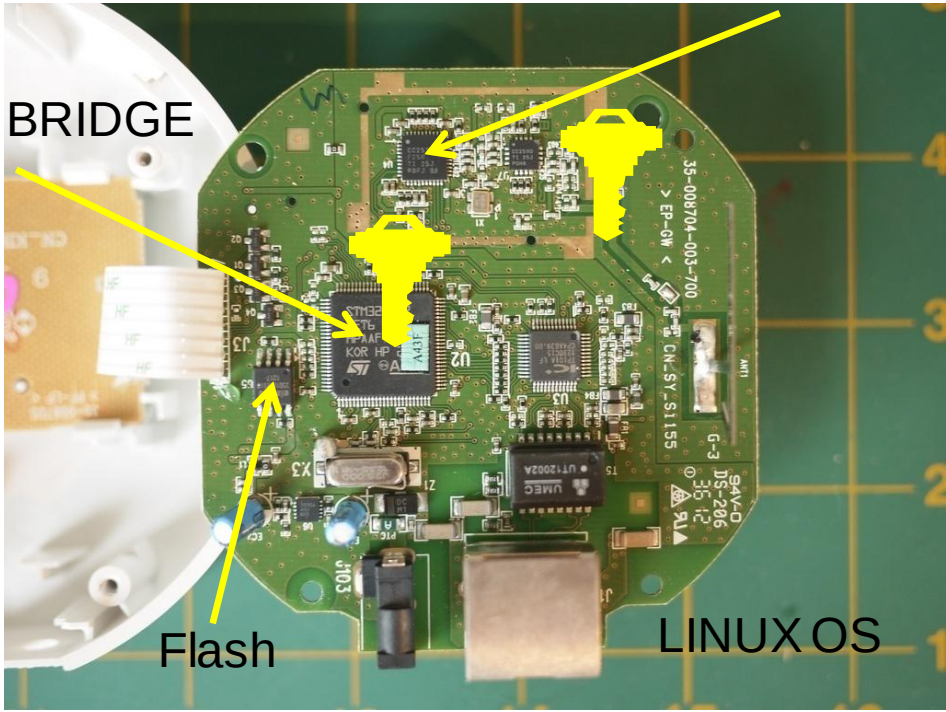
"The ZLL security architecture is based on using a fixed secret key, known as the ZLL key, which shall be stored in each ZLL device. All ZLL devices use the ZLL key to encrypt/decrypt the exchanged network key. "

https://brandonevans.ca/projects/hacking-the-hue

A LIGHTBULB WORM?, Details of the Philips Hue Smart Lighting Design, Colin O'Flynn – August 1, 2016.

ZigBee SoC Includes a Hardware Accelerator

These bridges contain two sections: the main ARM processor, and the Zigbee ZLL solution (referred to as the 'Zigbee SoC').
The main ARM processor is a STM32F217VET6 by ST.
This is a Cortex M3 device, with 512 Kbyte FLASH memory (internal) + 128 Kbyte of SRAM (internal). It contains a number of cryptographic hardware accelerators (AES + 3DES + MD5 + SHA-1).

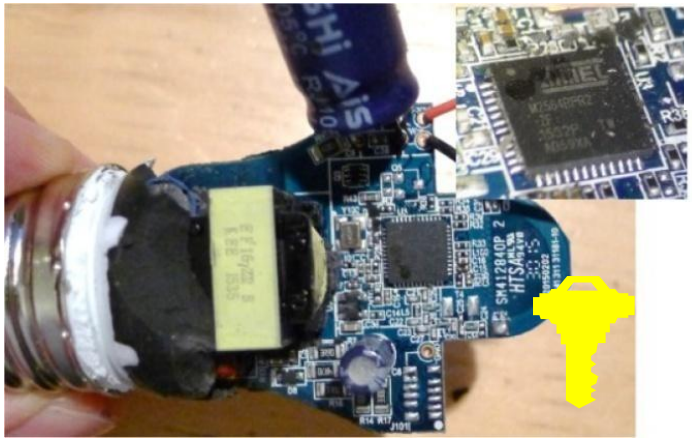


The ZigBee section is of most interest to us. It contains a CC2530F256 IEEE 802.15.4 SoC, alongside a CC2590 "range extender" (i.e., amplifier)

BULB

The firmware updates are downloaded Over The Air (OTA).
The firmware file itself can be downloaded from a fixed URL, and contains an encrypted firmware file .

The core processor is an Atmel ATMega2564RFR2.



An IEEE 802.15.4 compliant single chip combines an industry-leading AVR microcontroller and best-in-class 2.4GHz RF transceiver.

- 256K Bytes In-System, Self-Programmable Flash memory, 8K Bytes EEPROM, 32K Bytes SRAM).
- Crypto Engine AES

A LIGHTBULB WORM?, Details of the Philips Hue Smart Lighting Design, Colin O'Flynn – August 1, 2016.

Our attack proceeds in the following way: We send a unicast Reset to Factory New Request command to our target Philips Hue light….This causes the light to start a ZigBee association process and join our network

Bulb reprogramming from drone

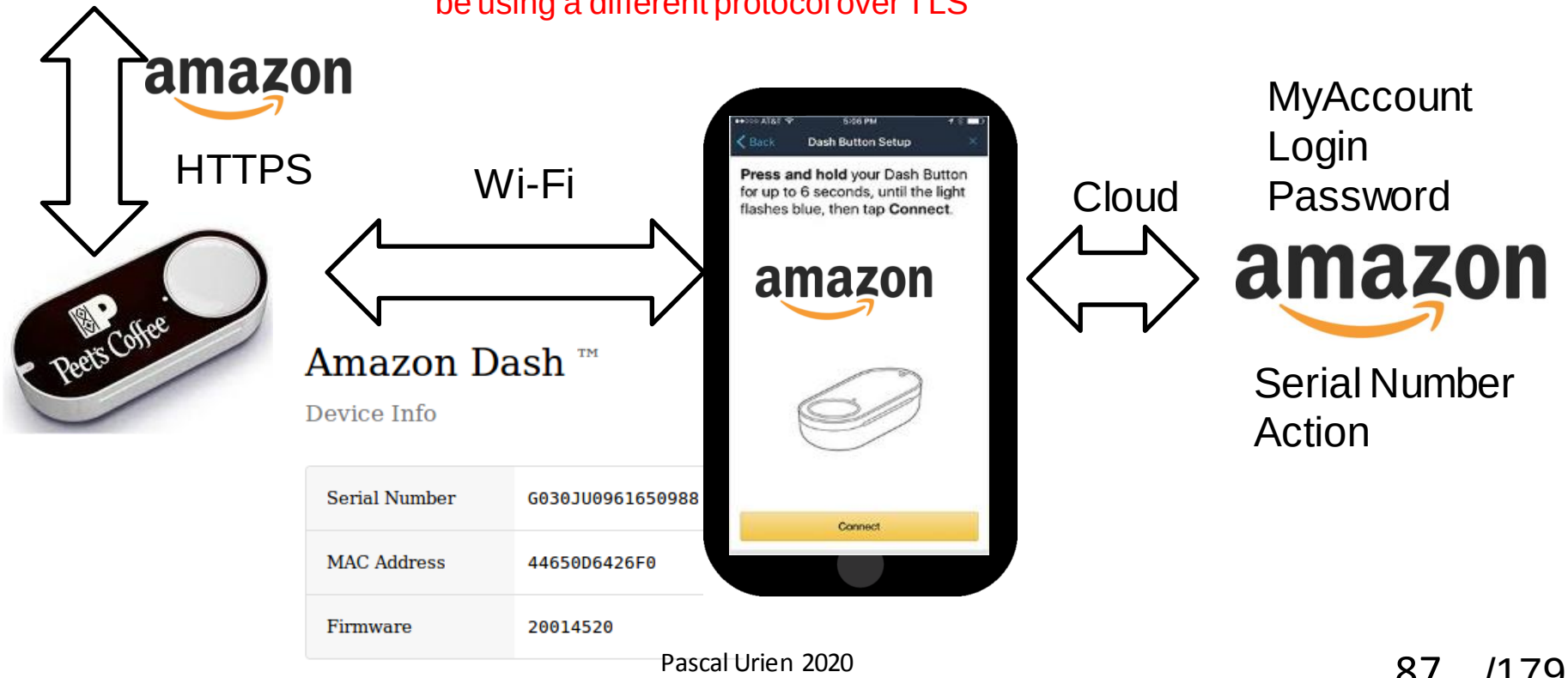
Each firmware update had to be both encrypted and authenticated by AES-CCM (in which AES is used to encrypt a Counter with CBC-MAC); ***however, all the lamps use the same global key***. ***The key was recovered by a CPA (Correlation Power Analysis) attack.***

" IoT Goes Nuclear: Creating a ZigBee Chain Reaction"
Eyal Ronen, Colin O'Flynn, Adi Shamir and Achi-Or Weingarten (2017)

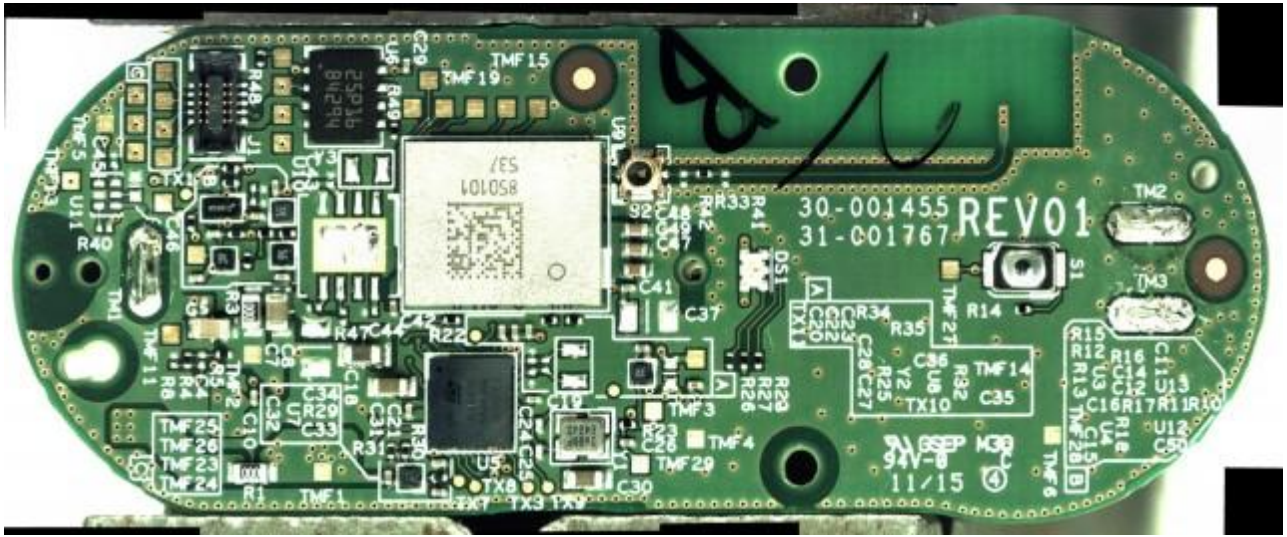
Example 7.
Amazon Dash Button

Button communicates with parker-gateway-na.amazon.com via TLS

When connecting via HTTPS, a certificate signed by the Amazon.com Internal Root Certificate Authority and issued to Amazon.com Infosec CA G2 is presented, which expires 2016-06-22. However, I was not able to successfully connect even after bypassing the certificate error, so it might be using a different protocol over TLS



https://mpetroff.net/2015/05/amazon-dash-button-teardown/

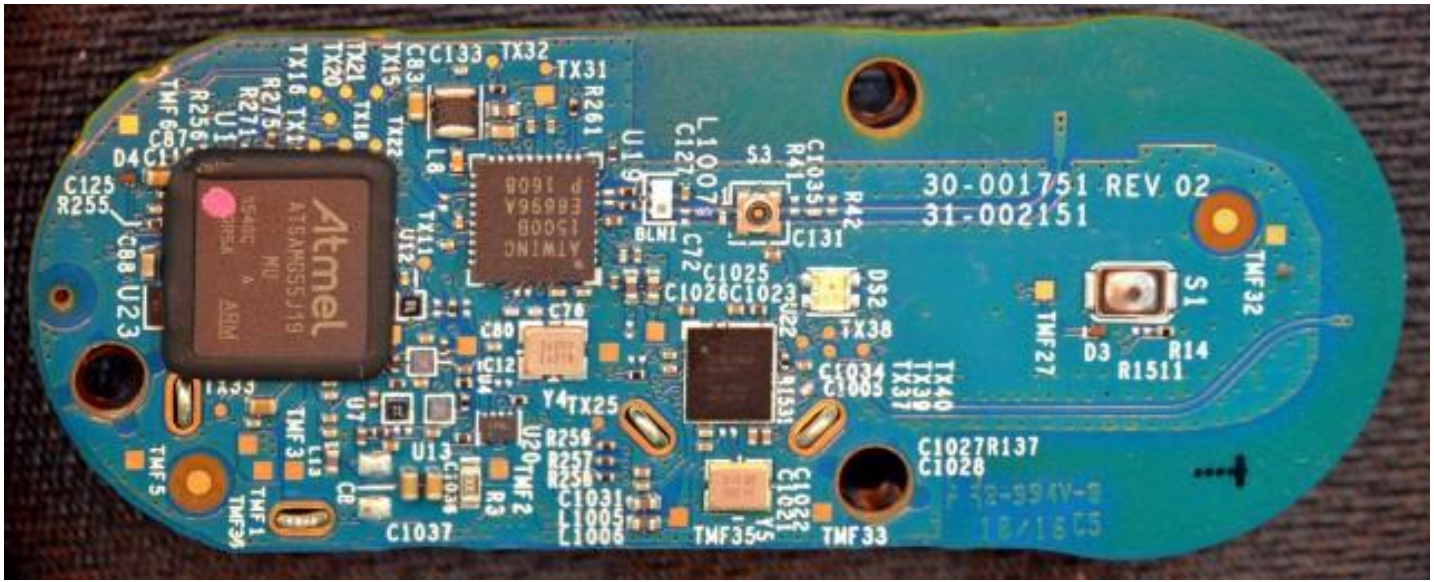


The design seems based on the Broadcom BCM943362WCD4 WICED module reference design, with a Broadcom BCM43362 Wi-Fi module, U9, and an ST STM32F205 microcontroller, U5

Other components on the Dash Button include an InvenSense INMP441 microphone, MP1; a Micron M25P16 16Mbit serial Flash memory module in a UFDFPN8 package, U6

Although not mentioned in the documentation, the Dash Button creates a Wi-Fi hotspot when placed in configuration mode, Amazon ConfigureMe, which is used by the Android version of the Amazon Shopping app.
Once connected to this hotspot, a web page is accessible at 192.168.0.1 via HTTP, which allows for configuring the Button's Wi-Fi connection settings.

https://mpetroff.net/2016/07/new-amazon-dash-button-teardown-jk29lp/

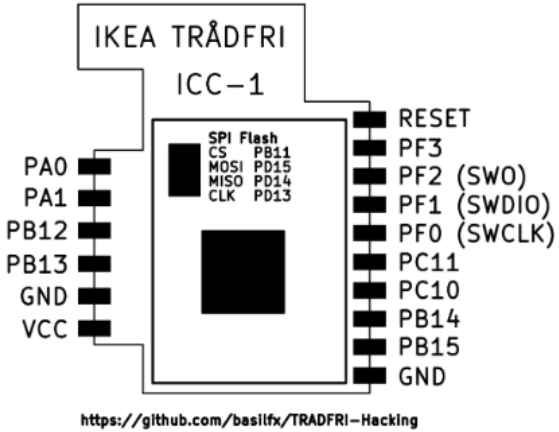


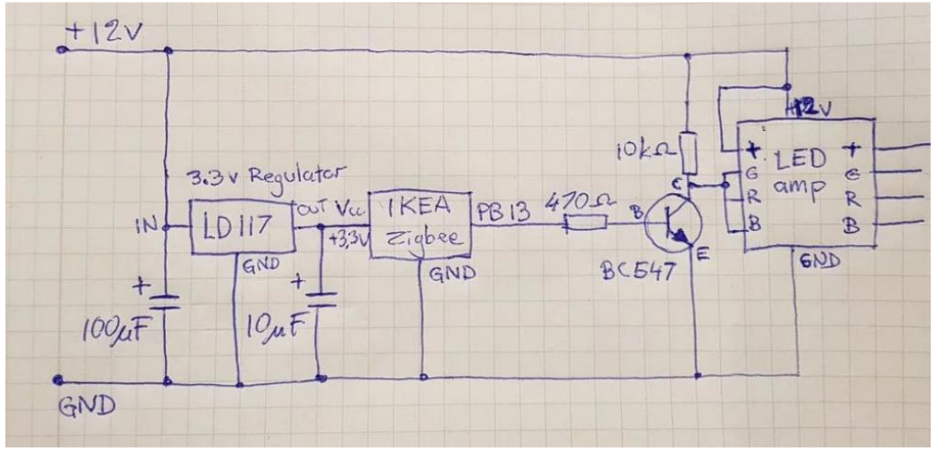
Instead of Broadcom chips, the new Button features an Atmel ATSAMG55J19A-MU ARM microcontroller, U1, and an Atmel ATWINC1500B wireless chip, U19. As a new addition, it also has a Cypress CYBL10563-68FNXI Bluetooth Low Energy chip, U22. The flash memory, U15, has been moved to the back of the PCB and doubled in size to 32 Mbit; it appears to be a Micron N25Q032 chip.

Example 8
IKEA TRÅDFRI

https://github.com/basilfx/TRADFRI-Hacking

https://www.instructables.com/id/Zigbee-LED-Strip-Dimmer-IKEA-Hack/





https://learn.pimoroni.com/tutorial/sandyj/controlling-ikea-tradfri-lights-from-your-pi

The structure of that payload is as follows: the `3311` represents a dimmer and the `5850` is the toggle for on/off, with `0` being off and `1` being on. Let's put that all together into a call to the `coap-client` that will toggle the light off:

secret

```
coap-client -m put -u "Client_identity" -k "1a2b3c4d5e6f7g8h" -e '{ "3311": [{ "5850": 0 }] }' "coaps://192.168.0.10:5684/15001/65537"
```

TRÅDFRI
Kit de réglage à distance, E27
19,99 € A+

Network Security

LTN: Low Throughput Network

SIGFOX & LORA

LEP: LTN End Point

LAP: LTN Access Point

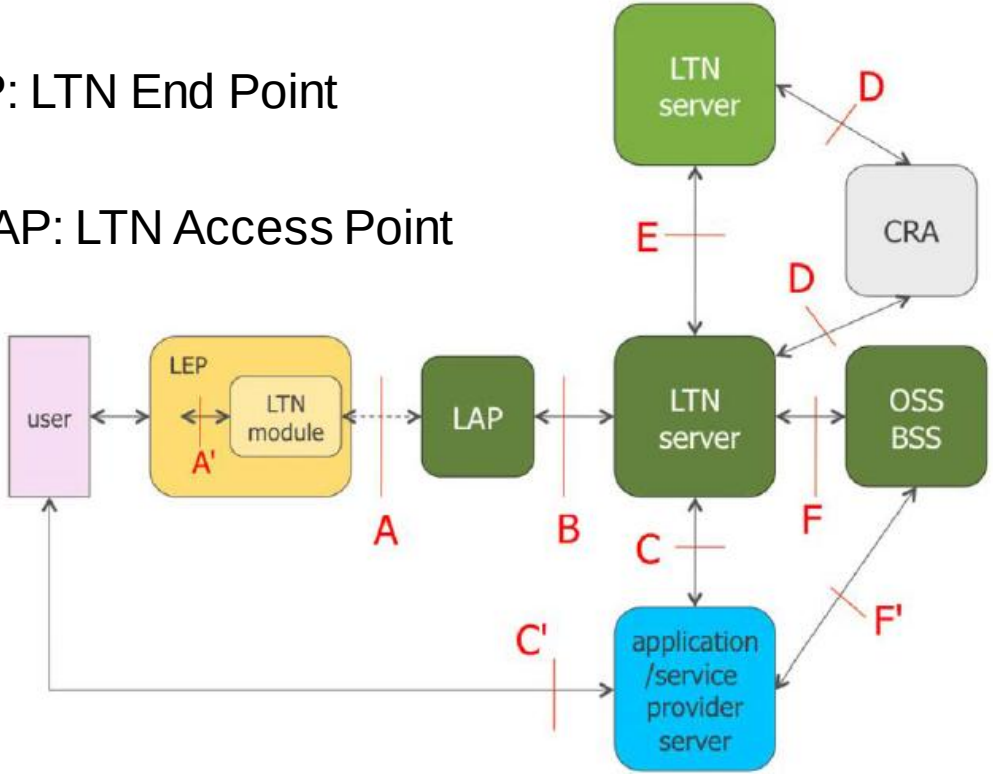


Figure 1: Overall LTN architecture and defined interfaces

CRA: Central Registration Authority

OSS: Operation Support System

BSS: Business Support System

LTN: Low Throughput Network

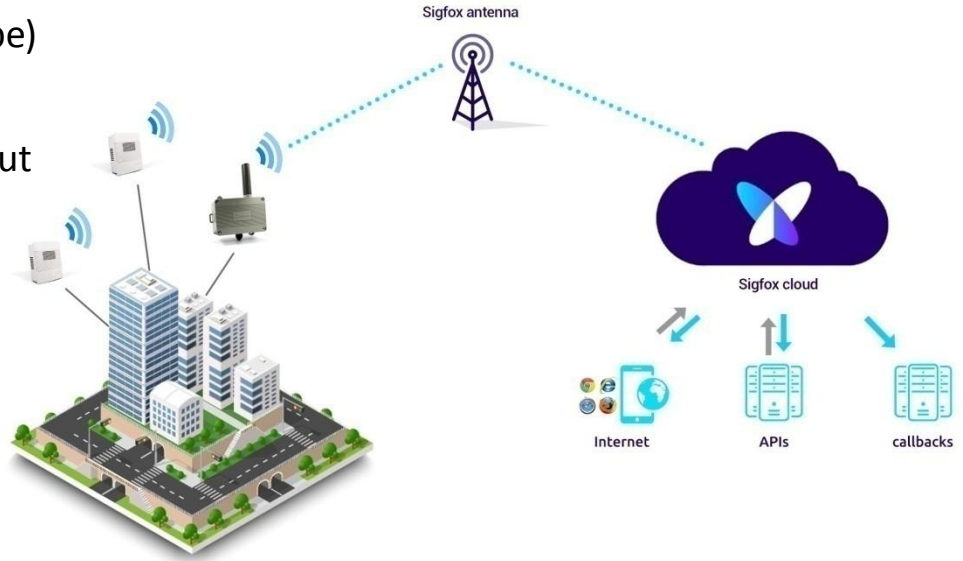
Sigfox

Abonnement; entre un et neuf euros par an et par capteur
140 messages de 12 octets/jour

Sigfox : Ultra Narrow Band Network

- Radio frequencies
  - The SIGFOX network operates in the unlicensed ISM radio bands. The ISM is available worldwide governed by regulation bodies such as ETSI (Europe) and the FCC (USA).
  - The exact frequencies can vary depending on national regulations, but in Europe the frequency is generally 868MHz and in the US it is 915MHz.
- Uplink and downlink
  - SIGFOX provides mono and bi-directional communication.
  - The capacity to provide mono-directional communication is very unique and allows extremely low power consumption in use cases where bi-directional communication is not required.

Downlink baud rate: 600 baud
For ETSI-zones, UNB downlink frequency band limited to 869,40 to 869,65 MHz



ETSI GS LTN 003 V1.1.1 (2014-09)

5.2.3.2 UNB MAC frame (downlink)

The format of the downlink UNB MAC frame is the following (see figure 3):

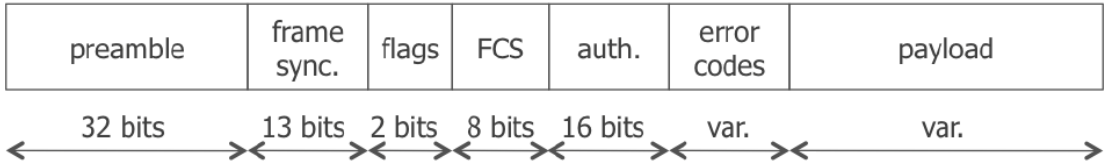


Figure 3: Downlink MAC frame in UNB implementation

5.2.2.2 UNB MAC frame (up-link)

The format of the uplink UNB MAC frame is the following (see figure 2):

HMAC

NID

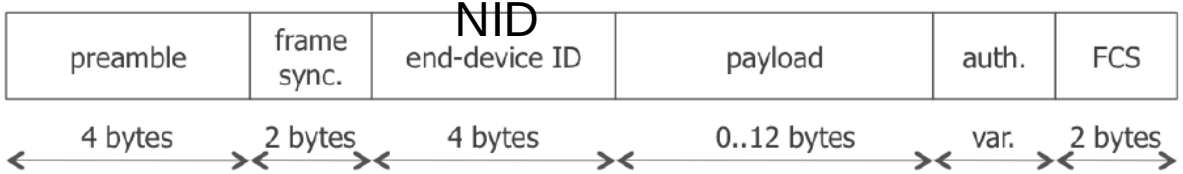


Figure 2: MAC frame in UNB up-link implementation

A unique identifier, named NID, is given to each UNB end-point. The NID is 32 bit long.

Each UNB end-point has a secret key (SEK). This key is 128 bit long. It is used to authenticate each radio packet transmitted by an UNB end-point.
The SEK authenticates the radio packet but it does not cipher the service payload. Payload ciphering is made at the application level.

SigFox Security

- Sécurité de Sigfox, CAPTRONIC - IoT et systèmes embarqués - 18 février 2016, Toulouse, Renaud Lifchitz de Digital Security.
- L'équipe de Digital Security est parti d'un modem émetteur Sigfox, et lui a fait émettre des motifs standards (que des 0x00, des 0×55 (suites de 0 et de 1), des 0xAA (suites de 1 et de 0), des 0xFF (suites de 1), et a écouté le signal avec une Software Defined Radio (un Realtek RTL2832U couplé à GnuRadio).
- Ils ont également regardé le firmware du modem avec un debuggeur classique IDA Pro.
- En observant les émissions, ils ont retrouvé que le message est émis 3 fois sur des fréquences légèrement différentes.
- Ils ont également reconstitué le format des trames: elles contiennent l'identificateur de l'émetteur, un compteur de trame sur 12 bits, le message, puis un CRC signé par HMAC.
- On peut donc attendre que le compteur de trames revienne à une valeur que l'on a déjà écoutée et rejouer un message.
- Tous les 4096 messages, donc, ce qui représente tout de même une attente de 29 jours au débit maximal de 140 messages par jour imposé par le réseau.
- À l'aide du debugger et en passant par l'interface physique de debuggage du modem, on arrive également à récupérer la clé secrète utilisée pour la signature du HMAC.
- Le réseau Sigfox présente les vulnérabilités suivantes:
  - Pas de confidentialité
  - Rejeu possible après une attente en écoute passive d'un mois
  - Impersonnification possible après accès physique au modem

LoRaWAN 1.0

Radio PHY layer:

Preamble	PHDR	PHDR_CRC	PHYPayload	CRC*

Figure 5: Radio PHY structure (CRC* is only available on uplink messages)

PHYPayload:

MHDR	MACPayload	MIC

Figure 6: PHY payload structure

MACPayload:

FHDR	FPort	FRMPayload

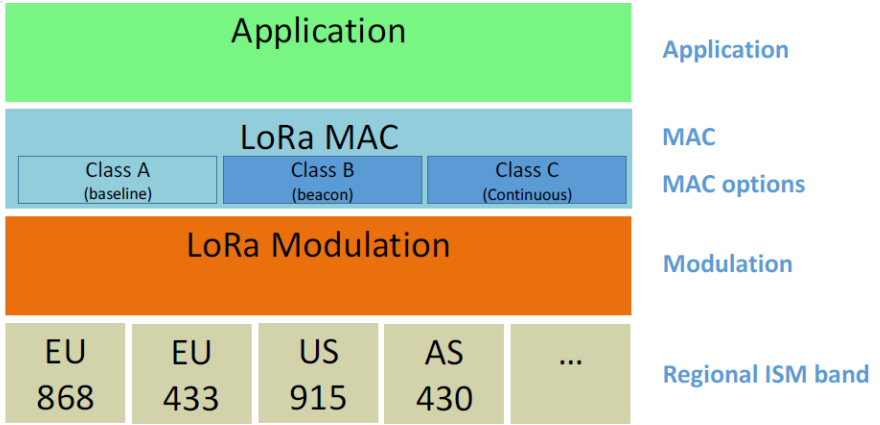
Figure 7: MAC payload structure

FHDR:

DevAddr	FCtrl	FCnt	FOpts

Figure 8: Frame header structure

Figure 9: LoRa message format elements



LoRaWAN can use channels with a bandwidth of either 125 kHz, 250 kHz or 500 kHz

Maximum MACPayload size length: 11 to 230 bytes depending on the throughput

Tarifs (pour des quantités < 500)

- Un tarif intégrant la connectivité de la technologie LoRa® et la mise à disposition des données brutes sur Live Objects.
- Un abonnement mensuel par device, intégrant un nombre de messages montants illimité dans le respect du duty cycle (sens de l'Objet vers le réseau).
- En complément, une facturation à l'usage d'envoi de commandes vers les objets (sens descendant du réseau vers l'Objet).
- Une historisation des données du client pendant 1 an.
- Un prix avec engagement 12/24/36 mois, ou sans engagement, avec une facturation globale.

LoRA Security

The DevEUI is a global end-device ID in IEEE EUI64 address space that uniquely identifies the end-device.

The AppKey is an AES-128 application key specific for the end-device that is assigned by the application owner to the end-device and most likely derived from an application-specific root key

Whenever an end-device joins a network via over-the-air activation, the AppKey is used to derive the session keys NwkSKey and AppSKey specific for that end-device to encrypt and verify network communication and application data

Lora Security

MAC Frame Payload Encryption (FRMPayload)
If a data frame carries a payload, FRMPayload must be encrypted before the message integrity code (MIC) is calculated.
The encryption scheme used is based on the generic algorithm described in IEEE 802.15.4/2006 Annex B using AES with a key length of 128 bits.

The key *K used depends on the FPort of the data message*

FPort	K
0	NwkSKey
1..255	AppSKey

Table 3: FPort list

The direction field (Dir) is 0 for uplink frames and 1 for downlink frames.
The blocks *Ai are encrypted to get a sequence S of blocks Si:*
Si = aes128_encrypt(K, Ai) for i = 1..k
S = S1 | S2 | .. | Sk
Encryption and decryption of the payload is done by truncating (*pld | pad16) xor S ,* to the first len(*pld) octets. 11*

Lora Join Request

The join procedure is always initiated from the end-device by sending a join-request message.

The join-request message contains the AppEUI and DevEUI of the end-device followed by a nonce of 2 octets (DevNonce).

DevNonce is a random value.

For each end-device, the network server keeps track of a certain number of DevNonce values used by the end-device in the past, and ignores join requests with any of these DevNonce values from that end-device.

Size (bytes)	8	8	2
Join Request	AppEUI	DevEUI	DevNonce

cmac = aes128_cmac(AppKey, MHDR | AppEUI | DevEUI | DevNonce)

MIC = cmac[0..3] 8

Lora Join Response

No response is given to the end-device if the join request is not accepted.
The join-accept message contains an application nonce (AppNonce) of 3 octets, a network identifier (NetID), an end-device address (DevAddr), a delay between TX and RX (RxDelay) and an optional list of channel frequencies (CFList) for the network the end device is joining.

Size (bytes)	3	3	4	1	1	(16) Optional
Join Accept	AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList

The AppNonce is a random value or some form of unique ID provided by the network server and used by the end-device to derive the two session keys NwkSKey and AppSKey as follows:
NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad16)
AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad16)
The MIC value for a join-accept message is calculated as follows:
cmac = aes128_cmac(AppKey, MHDR | AppNonce | NetID | DevAddr | RFU | RxDelay | CFList)
MIC = *cmac[0..3] 31*
The join-accept message itself is encrypted with the AppKey as follows:
aes128_decrypt(AppKey, AppNonce | NetID | DevAddr | RFU | RxDelay | CFList | MIC)

Bluetooth Security

NIST Special Publication 800-121 Revision 2 Guide to BluetoothSecurity

NIST Special Publication 800-121 Revision 2

Table 2-2. Key Differences Between Bluetooth BR/EDR and Low Energy

Characteristic	Bluetooth BR/EDR		Bluetooth Low Energy	
	Prior to 4.1	4.1 onwards	Prior to 4.2	4.2 onwards
RF Physical Channels	79 channels with 1 MHz channel spacing		40 channels with 2 MHz channel spacing	
Discovery/Connect	Inquiry/Paging		Advertising	
Number of Piconet Slaves	7 (active)/255 (total)		Unlimited	
Device Address Privacy	None		Private device addressing available	
Max Data Rate	1–3 Mbps		1 Mbps via GFSK modulation	
Pairing Algorithm	Prior to 2.1: E21/E22/SAFER+ 2.1-4.0: P-192 Elliptic Curve[9], HMAC-SHA-256	P-256 Elliptic Curve, HMAC-SHA-256	AES-128	P-256 Elliptic Curve, AES-CMAC
Device Authentication Algorithm	E1/SAFER	HMAC-SHA-256	AES-CCM[10]	
Encryption Algorithm	E0/SAFER+	AES-CCM	AES-CCM	
Typical Range	30 m		50 m	
Max Output Power	100 mW (20 dBm)		10 mW (10 dBm)[11]	

Bluetooth Security Modes

Table 3-1. BR/EDR/HS Security Modes

Mode	Security procedures occur during the setup of a
4	Service
3	Link
2	Service
1	Never

Table 3-2. BR/EDR/HS Security Mode 4 Levels Summary

Mode 4 Level	FIPS approved algorithms	Provides MITM protection	User interaction during pairing	Encryption required
4	Yes	Yes	Acceptable	Yes
3	No	Yes	Acceptable	Yes
2	No	No	Minimal	Yes
1	No	No	Minimal	Yes
0	No	No	None	No

Table 3-3. Most Secure Mode for a Pair of Bluetooth Devices

Local Bluetooth Version	Most secure Mode connecting to a peer which is	
	2.0 or lower	2.1 or higher
4.2	Mode 3	Mode 4 (Mandatory)
4.1		
4.0		
3.0		
2.1		
2.0		Mode 3
1.2		
1.1		
1.0		

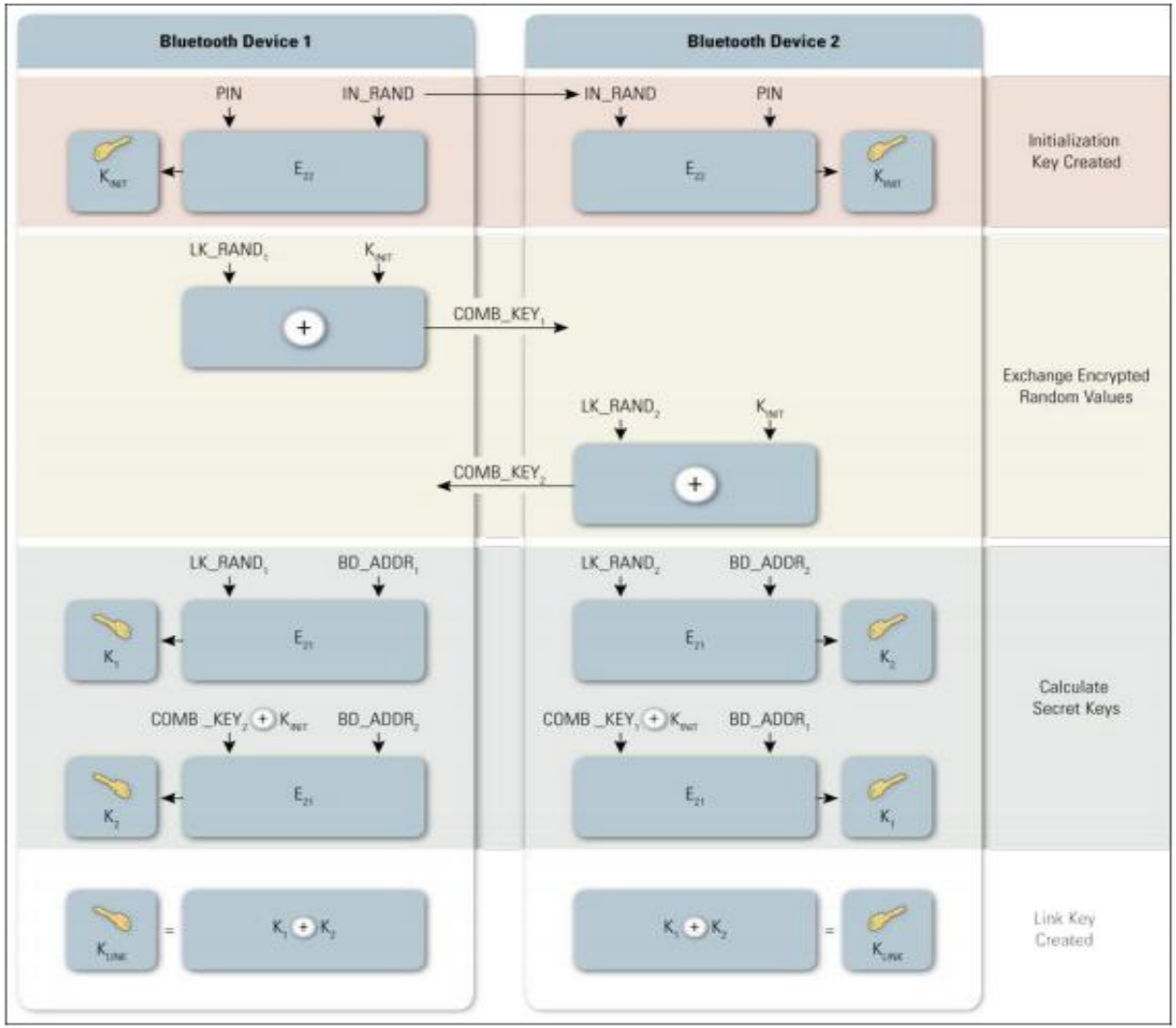
Table 3-4. Most Secure Level in Mode 4 for a Pair of Bluetooth Devices

Local Bluetooth Version	Most secure Mode 4 Level connecting to a peer which is	
	2.1 – 4.0	4.1 or higher
4.2	Level 3	Level 4
4.1		
4.0		Level 3
3.0		
2.1		
2.0	N/A	N/A
1.2		
1.1		
1.0		

PIN/Legacy Pairing

- Essential to the authentication and encryption mechanisms provided by Bluetooth is the generation of a secret symmetric key.
  - In Bluetooth BR/EDR this key is called the **Link Key**
  - In Bluetooth Low Energy this key is called the **Long Term Key**
- In legacy low energy pairing, a Short Term Key is generated, which is used to distribute the Slave and/or Master Long Term Key, while in low energy Secure Connections, the Long Term Key is generated by each device and not distributed

PIN/Legacy Pairing



Secure Simple Pairing SSP

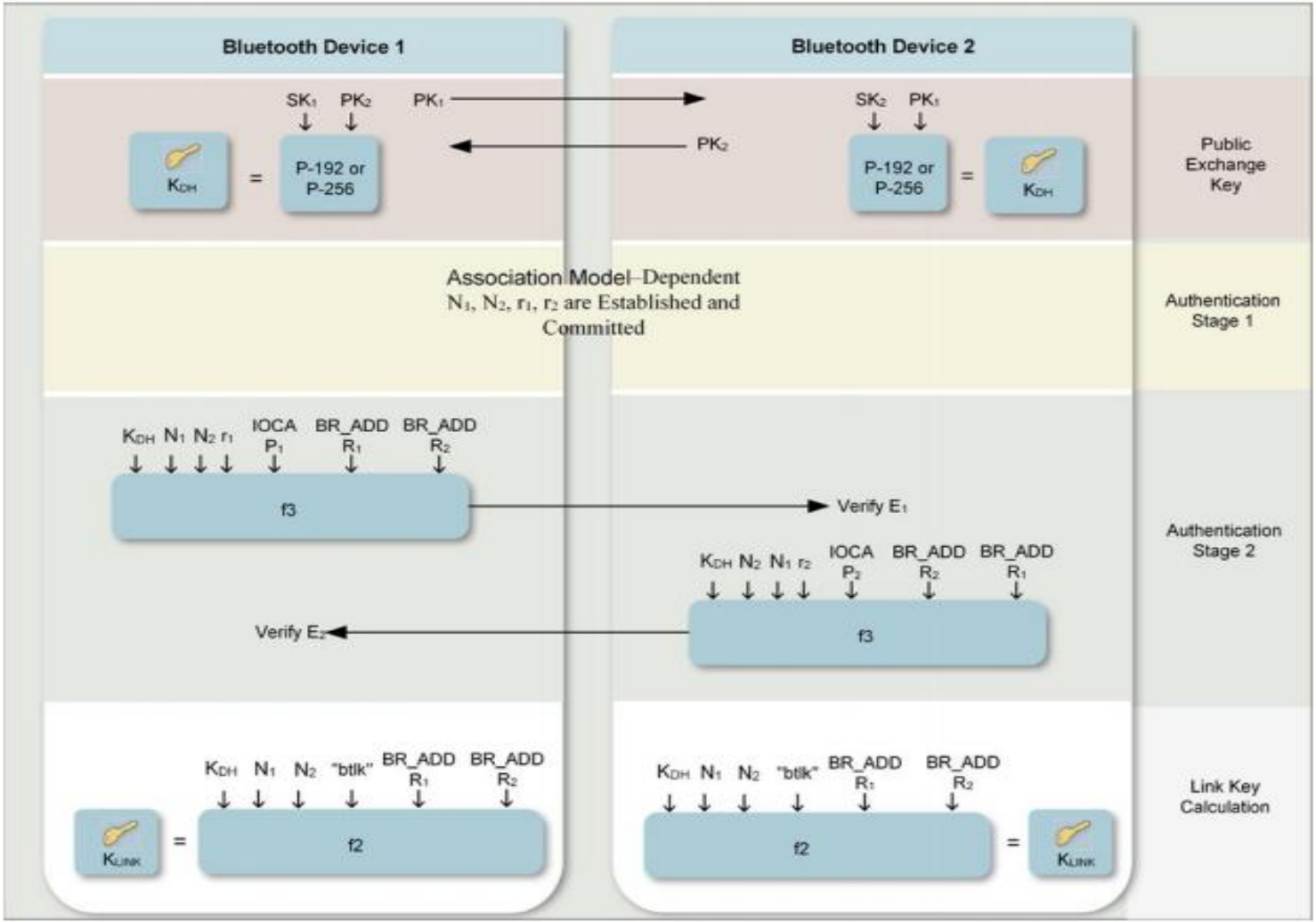
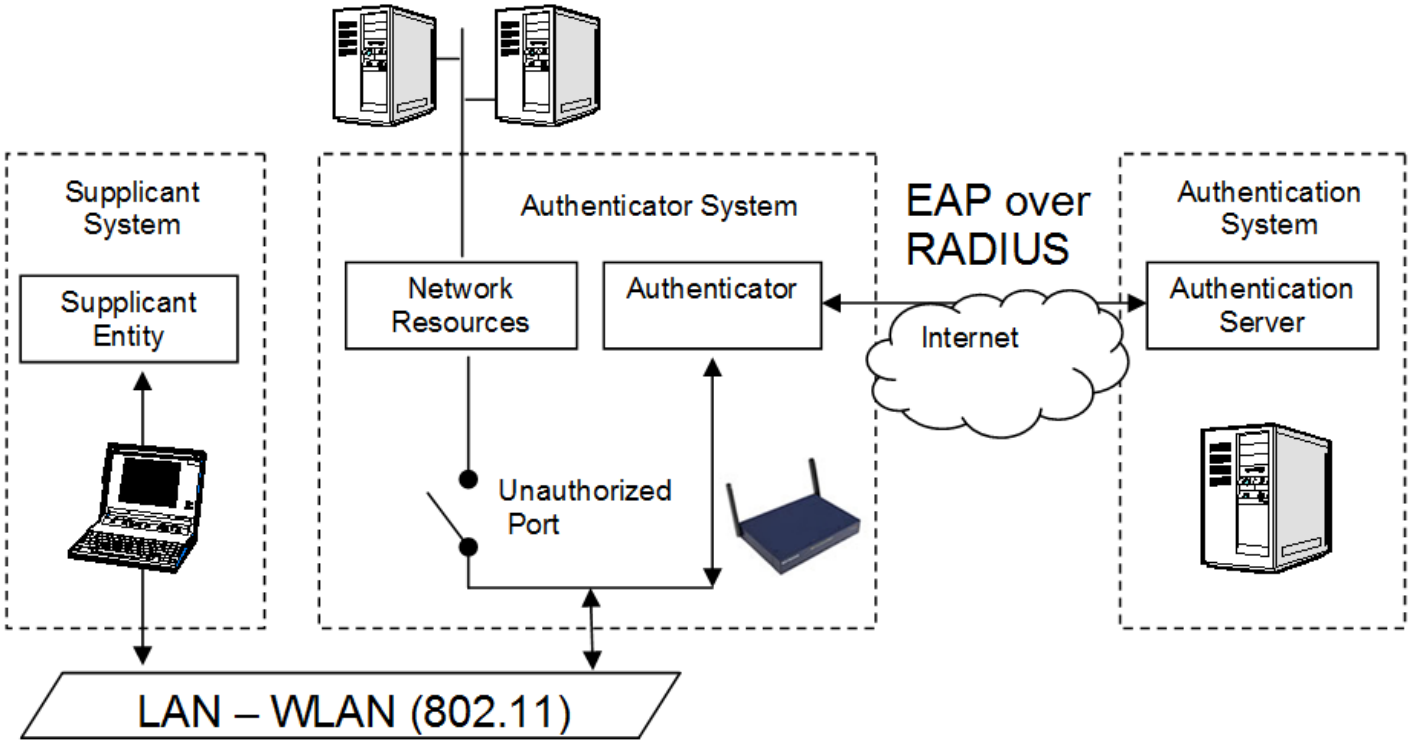


Figure 3-3. Link Key Establishment for Secure Simple Pairing

Wi-Fi Security

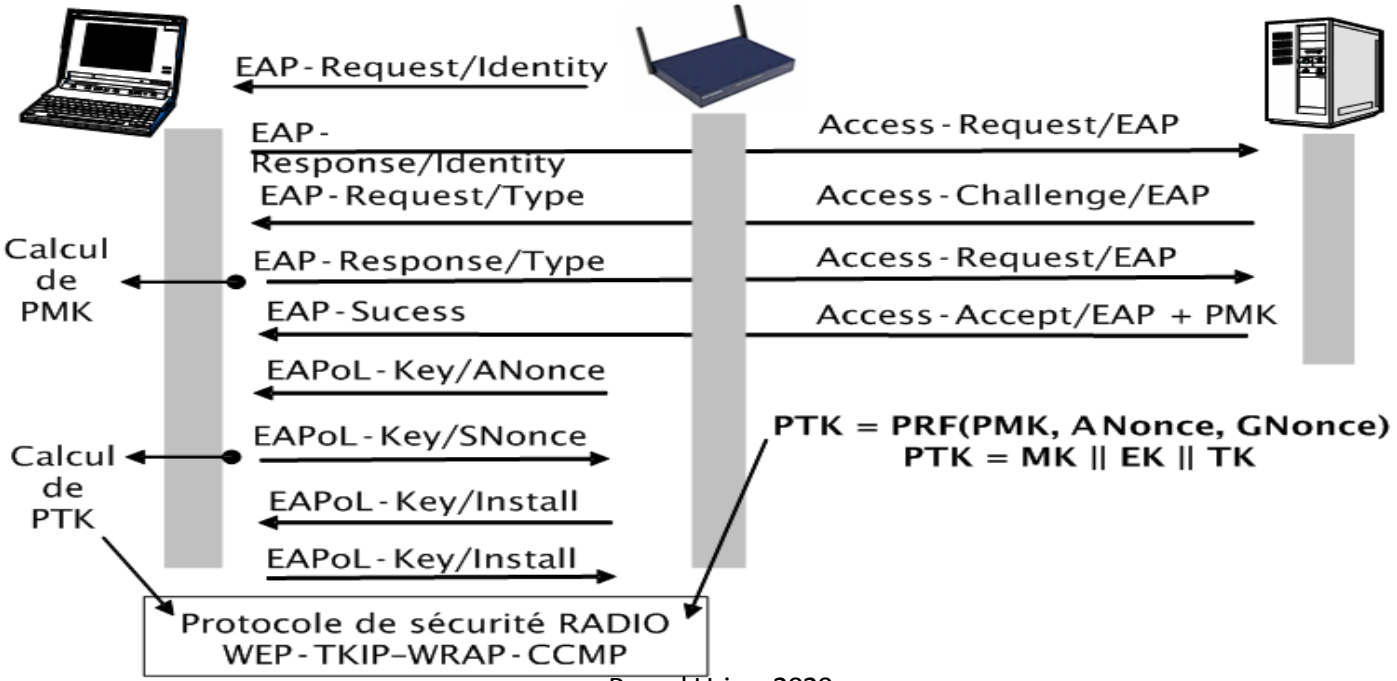
IEEE 802.1x


Supplicant System
Supplicant Entity
Authenticator System
Network Resources
Authenticator
Unauthorized Port
EAP over RADIUS
Internet
Authentication System
Authentication Server
LAN – WLAN (802.11)

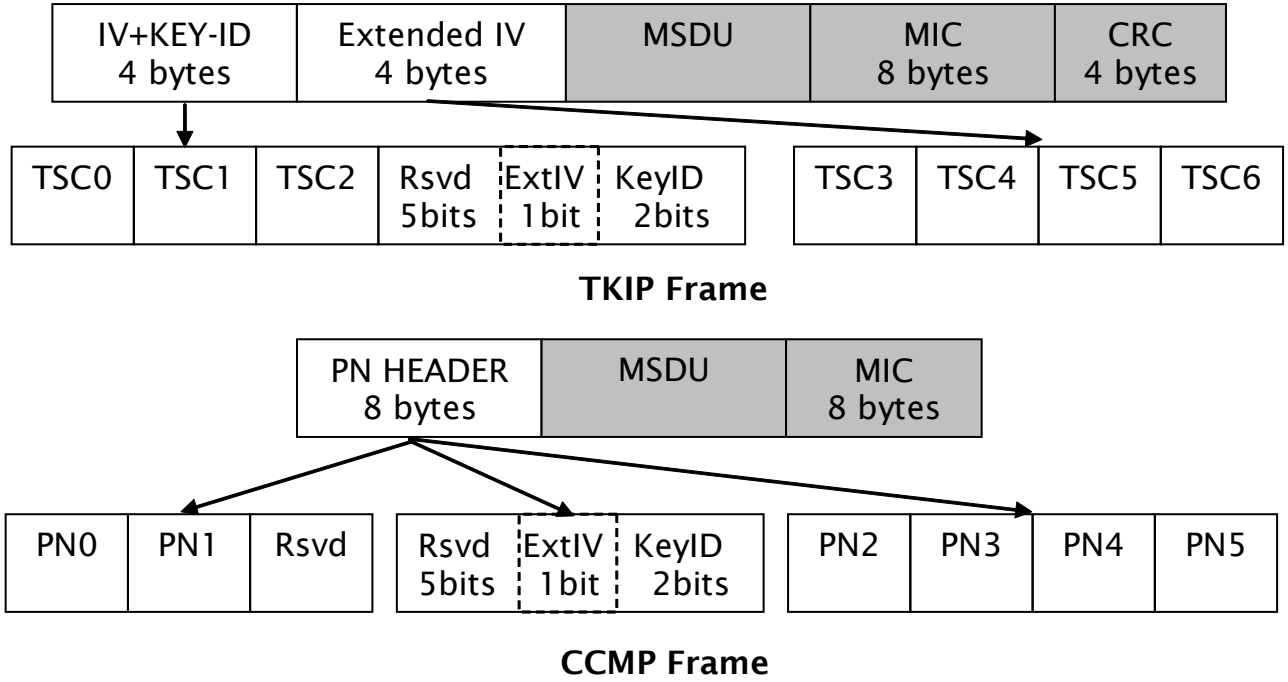
IEEE 802.11i : WPA-WPA2-WPA3

- Four ways handshake (PTK).
- Two ways hanshake (GTK).

IEEE 802.1x



WPA-WPA2 MAC Frame



Simple Password Exponential Key Exchange (WPA3)

The SPEKE Protocol

	Alice		Bob
	p is a safe prime: $p = 2 \cdot q + 1$		
1.	$x \in [1, q-1]$		$y \in Q$
2.	$X = (s^2)^x$	$\xrightarrow{X}$	
3.		$\xleftarrow{Y}$	$Y = (s^2)^y$
4.	$K = Y^x = s^{2xy}$		$K = X^y = s^{2xy}$

Dragonfly rfc7664 (WPA3)

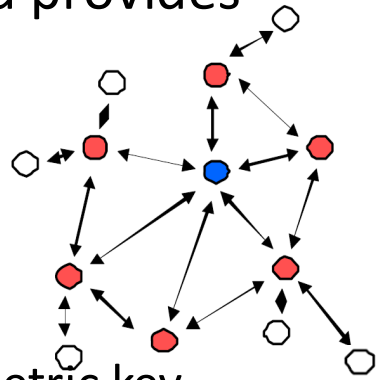
	Alice		Bob
	$P \in Q$		$P \in Q$
1.	$r_A, m_A \in \{1, \ldots, q\}$		$r_B, m_B \in \{1, \ldots, q\}$
2.	$s_A = r_A + m_A$		$s_B = r_B + m_B$
3.	$E_A = P^{-m_A}$	$\xrightarrow{s_A, E_A}$	$E_B = P^{-m_B}$
4.		$\xleftarrow{s_B, E_B}$	
5.	$ss = (P^{s_B} E_B)^{r_A}$	$\xrightarrow{A = H(ss\|E_A\|s_A\|E_B\|s_B)}$	Verify A
	$= P^{r_B r_A}$		$ss = (P^{s_A} E_A)^{r_B}$
6.	Verify B	$\xleftarrow{B = H(ss\|E_B\|s_B\|E_A\|s_A)}$	$= P^{r_A r_B}$
7.	Compute the shared key: $K = H(ss\|E_A \cdot E_B\|(s_A + s_B) \bmod q)$		

Q sub group (order q prime)

Zigbee Security

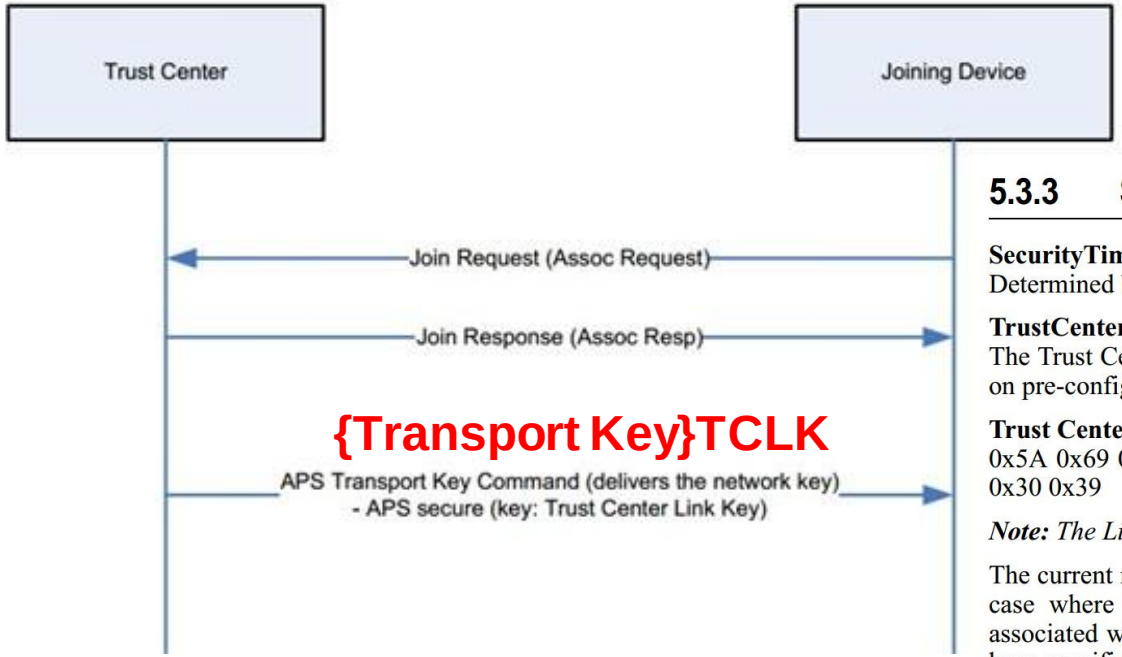
IEEE 802.15.4 & Zig Bee

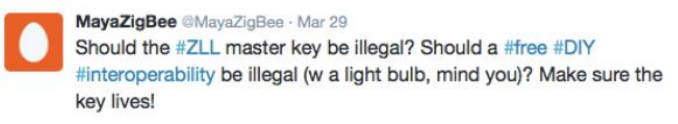
- Coordinator is assumed to be the Trust Center (TC) and provides
  - Cryptographic key establishment
  - Key transport
  - Frame protection
  - Device management
- Cryptographic Keys
  - **Master Key** , basis for long term security used for symmetric key establishment. It is used to keep confidential the Link Keys exchange between two nodes in the Key Establishment Procedure (SKKE).
  - **Link Key**, shared between two network peers for Unicast communication.
  - **Network Key**, used for broadcast communication security.



ZigBee Security

rt: @MayaZigBee
#DIY lover #ZLL master key 9F 55 95 F1 02 57 C8 A4 69 CB F4 2B C9 3F EE 31 #ZigBee #Philips #Hue





5.3.3 Security Parameters

SecurityTimeoutPeriod
Determined by the stack profile.

TrustCenterNetworkKey
The Trust Center will pick the network key. ZigBee HA devices shall not depend on pre-configured network keys to be commissioned or to interoperate.

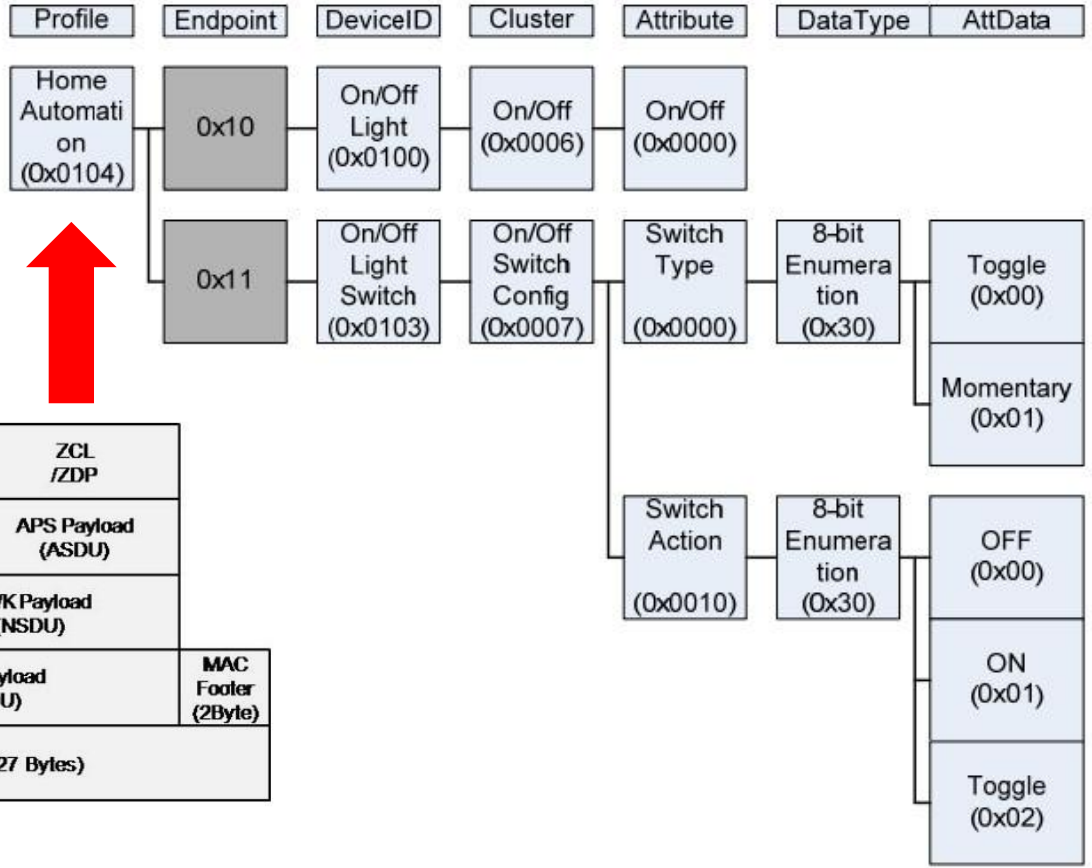
Trust Center Link Key
0x5A 0x69 0x67 0x42 0x65 0x65 0x41 0x6C 0x6C 0x69 0x61 0x6E 0x63 0x65 0x30 0x39

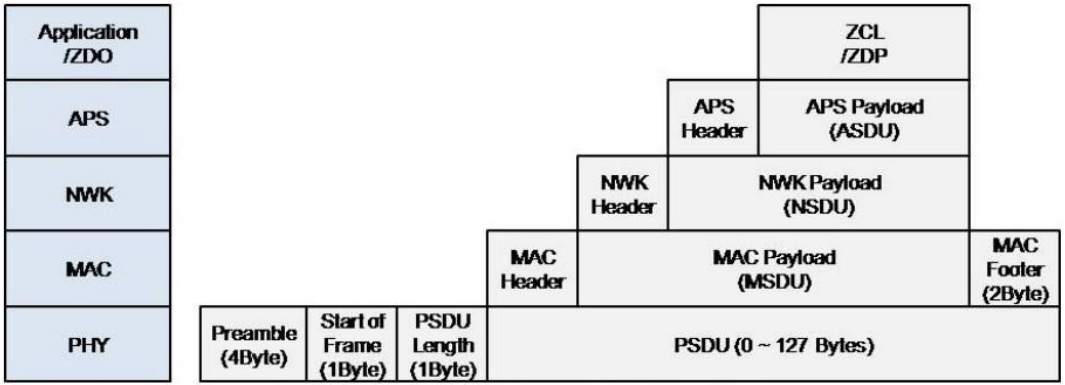
Note: *The Link Key is listed in little-endian format.*

The current network key shall be transported using the default TC link key in the case where the joining device is unknown or has no specific authorization associated with it. This allows for the case where alternative pre-configured link keys specifically associated with a device can be used as well.

http://prog3.com/sbdm/blog/u012993936/article/details/50280963

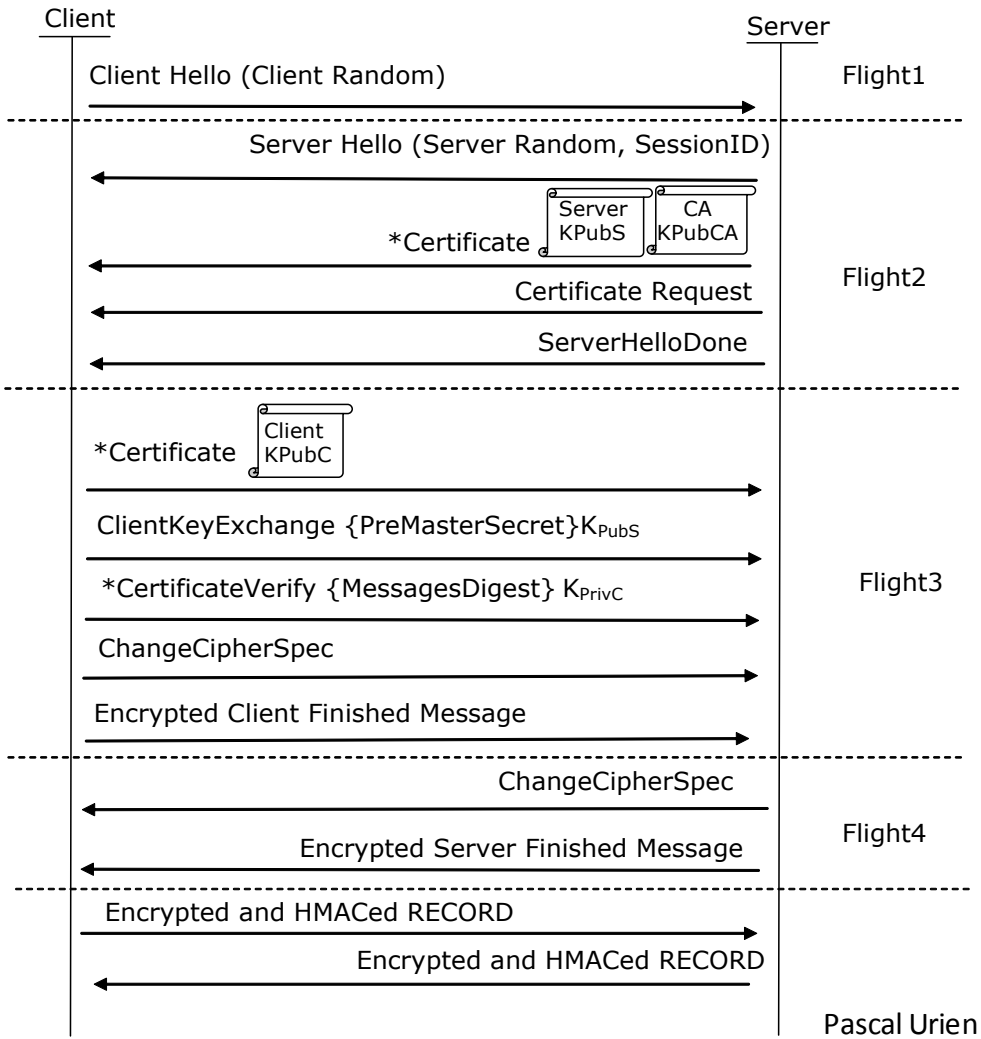
Exemple de commande ZigBee ZCL

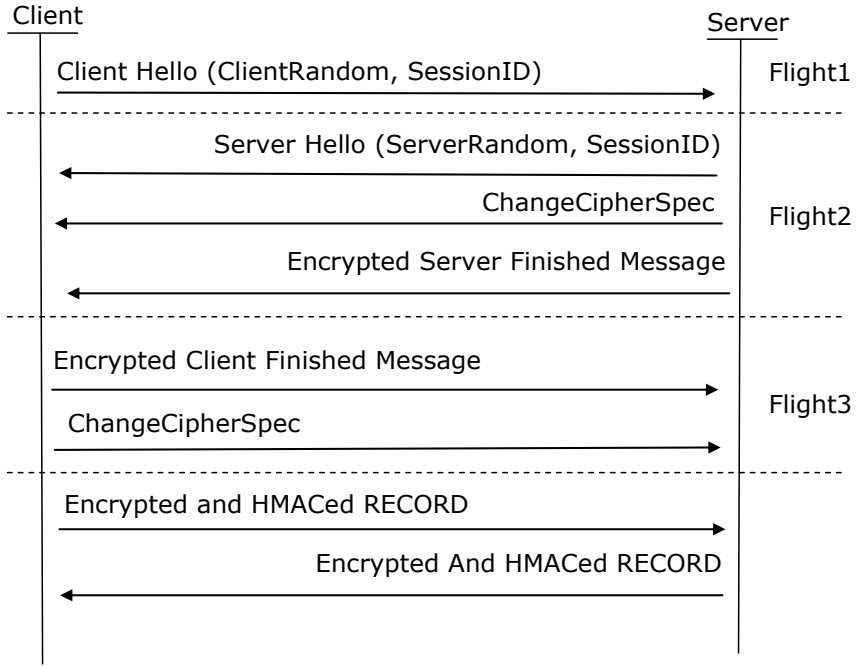


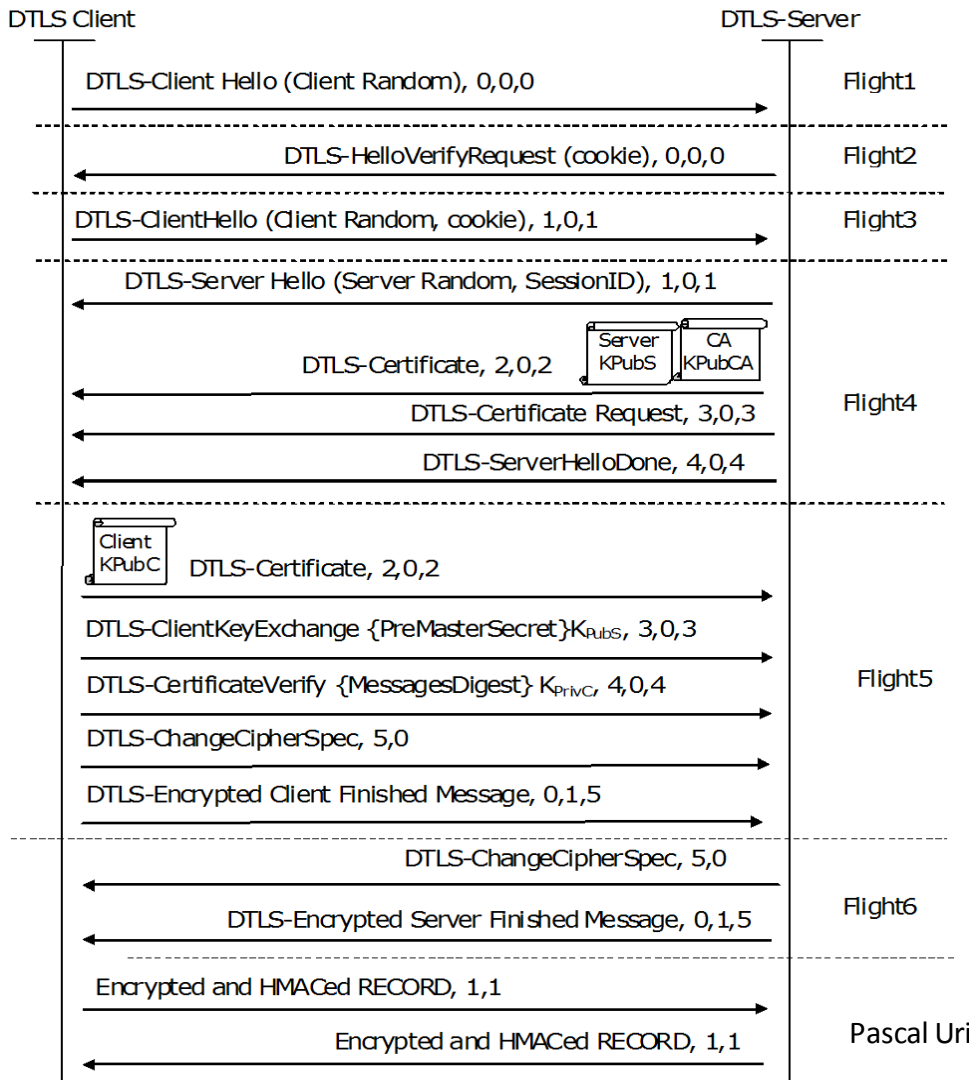


TLS/DTLS

About TLS


Client
Server
Client Hello (Client Random)
Flight1
Server Hello (Server Random, SessionID)
*Certificate
Server KPubS
CA KPubCA
Certificate Request
ServerHelloDone
Flight2
*Certificate
Client KPubC
ClientKeyExchange {PreMasterSecret}K PubS
*CertificateVerify {MessagesDigest} K PrivC
ChangeCipherSpec
Encrypted Client Finished Message
Flight3
ChangeCipherSpec
Encrypted Server Finished Message
Flight4
Encrypted and HMACed RECORD
Encrypted and HMACed RECORD


Client
Server
Client Hello (ClientRandom, SessionID)
Flight1
Server Hello (ServerRandom, SessionID)
ChangeCipherSpec
Encrypted Server Finished Message
Flight2
Encrypted Client Finished Message
ChangeCipherSpec
Flight3
Encrypted and HMACed RECORD
Encrypted And HMACed RECORD



About DTLS

The two first number are respectively the record sequence number and the *epoch* field.

The optional third number is the message sequence used by a handshake message.

DTLS cryptographic details

Handshake cryptographic calculations are insensitive to fragmentation operations.

According to finished messages (either client or server) have no sensitivity to fragmentation. There are computed as if each handshake message had been sent as a single fragment, i.e. with *Fragment-Length* set to Length, and *Fragment-Offset* set to zero ; the *Message-Sequence* field is not used in these procedures.

It also should be noticed that the DTLS-HelloVerifyRequest message and the previous associated DTLS-ClientHello are not taken into account by the Handshake cryptographic calculation.

DTLS Handshake and Record Layer

Handshake Message	
Type	1B
Length	3B
Message Sequence	2B
Fragment Offset	3B
Fragment Length	3B
Total length	12B

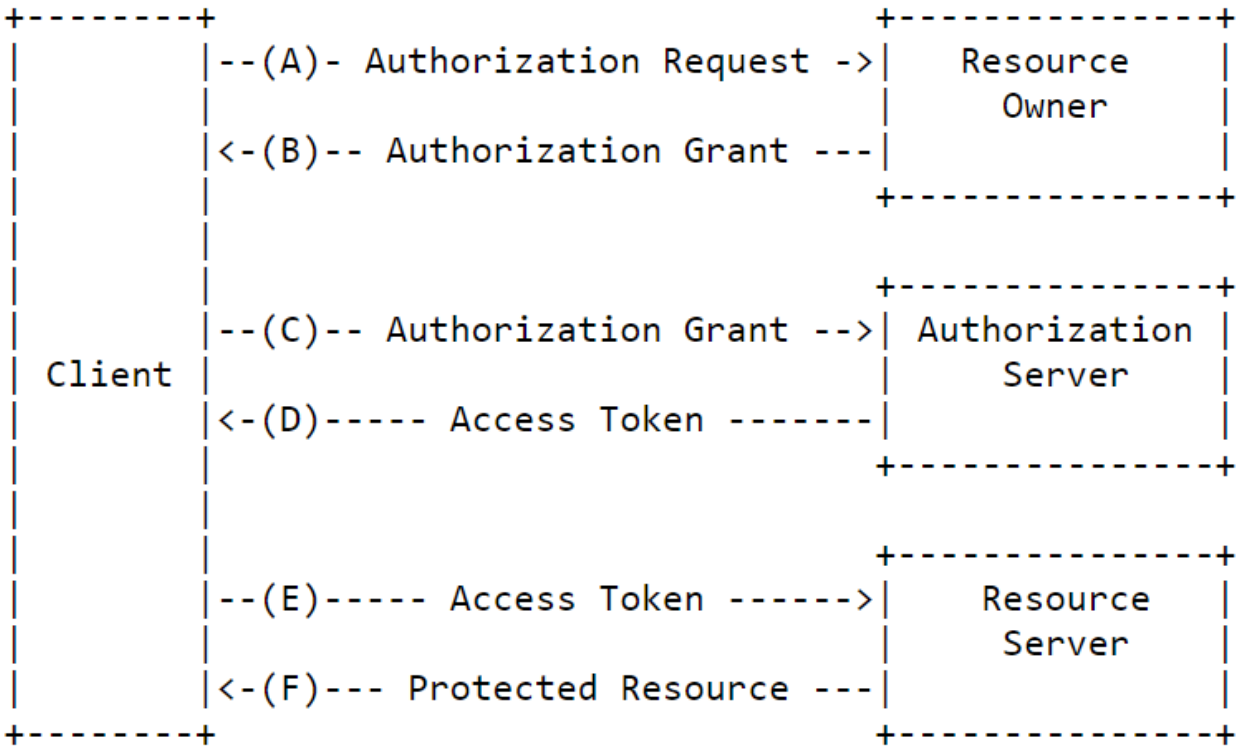
Record Packet	
Type	1B
Version	2B
Epoch	2B
Sequence Number	6B
Length	2B
Total Length	15B

IETF Working groups

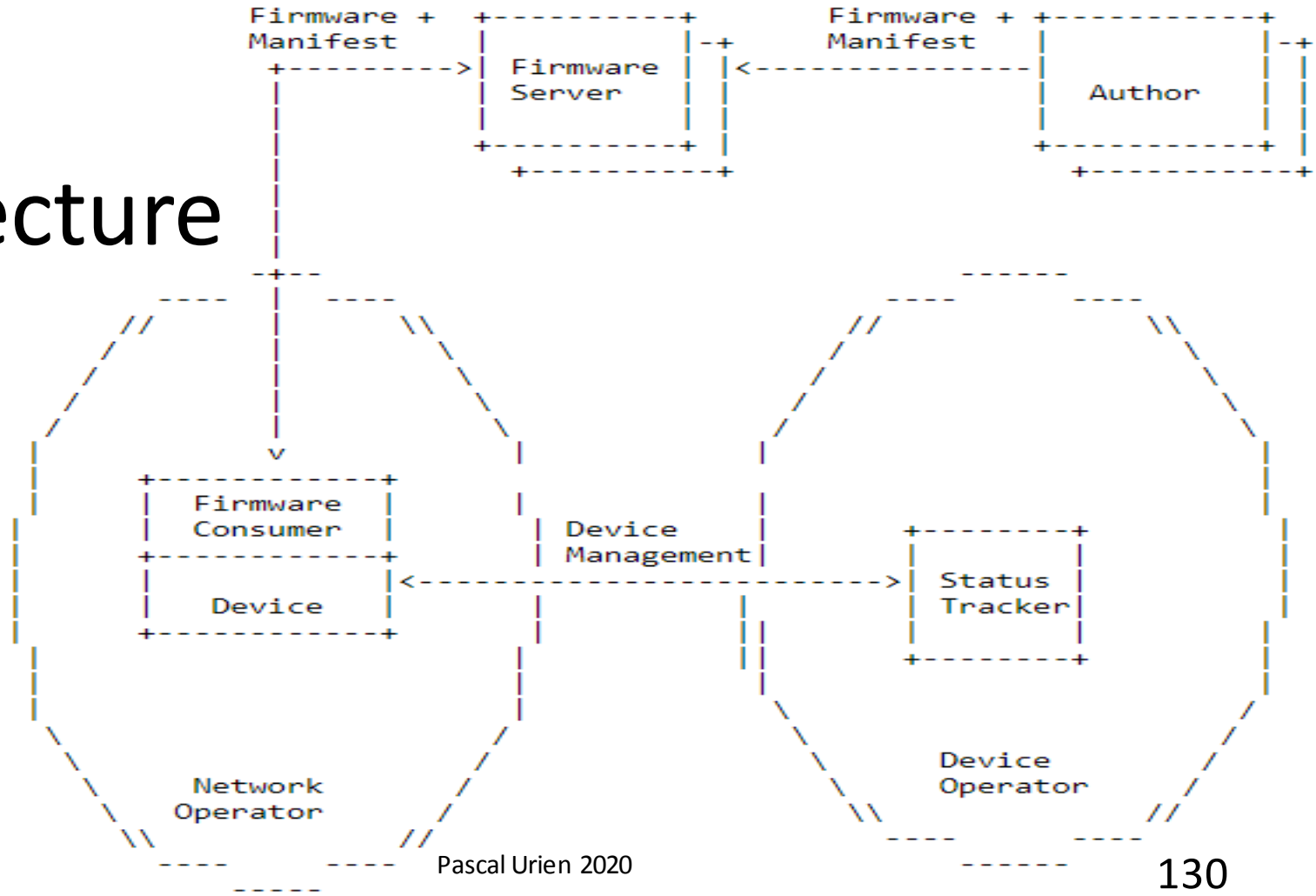
IETF Work Groups

- Constrained RESTful Environments (core)
- Authentication and Authorization for Constrained Environments (ACE)
- Software Updates for Internet of Things (suit)
- Trusted Execution Environment Provisioning (teep)

OAUTH 2.0 RFC 6749



SUIT Architecture



Résistance au hacking

Tamper Resistant Devices

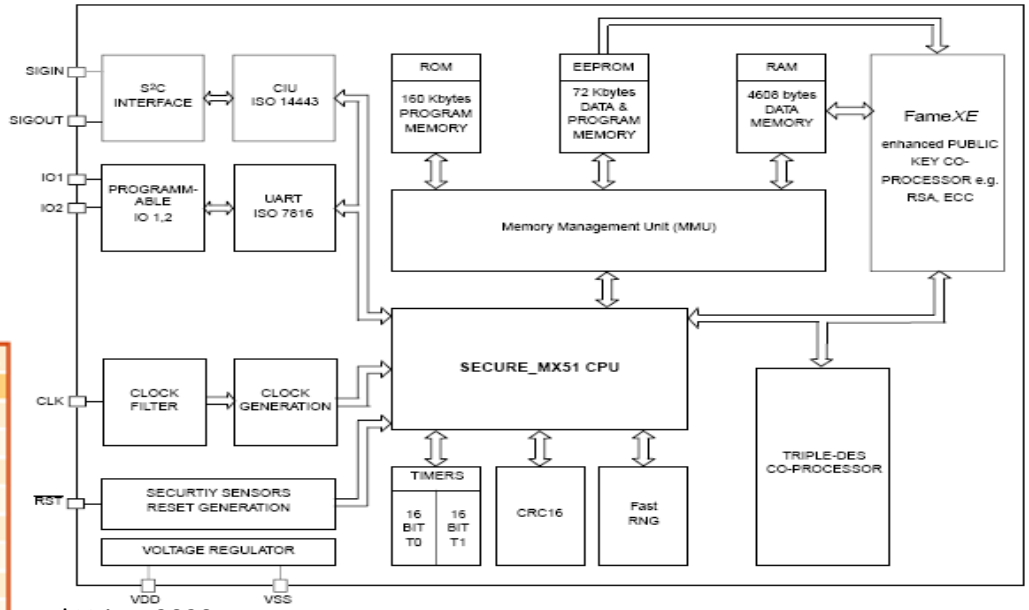
USIM, iSIM, TEE

What is a Secure Element ?

A Secure Element (SE) is a Secure Microcontroller, equipped with host interfaces such as ISO7816, SPI or I^2C .

OS JAVACARD JCOP
GP (Global Platform)
ROM 160 KB
EEPROM 72 KB
RAM 4KB
Crypto-processor
3xDES, AES, RSA, ECC
Certification CC (Common Criteria) EAL5+
Security Certificates EMVCo

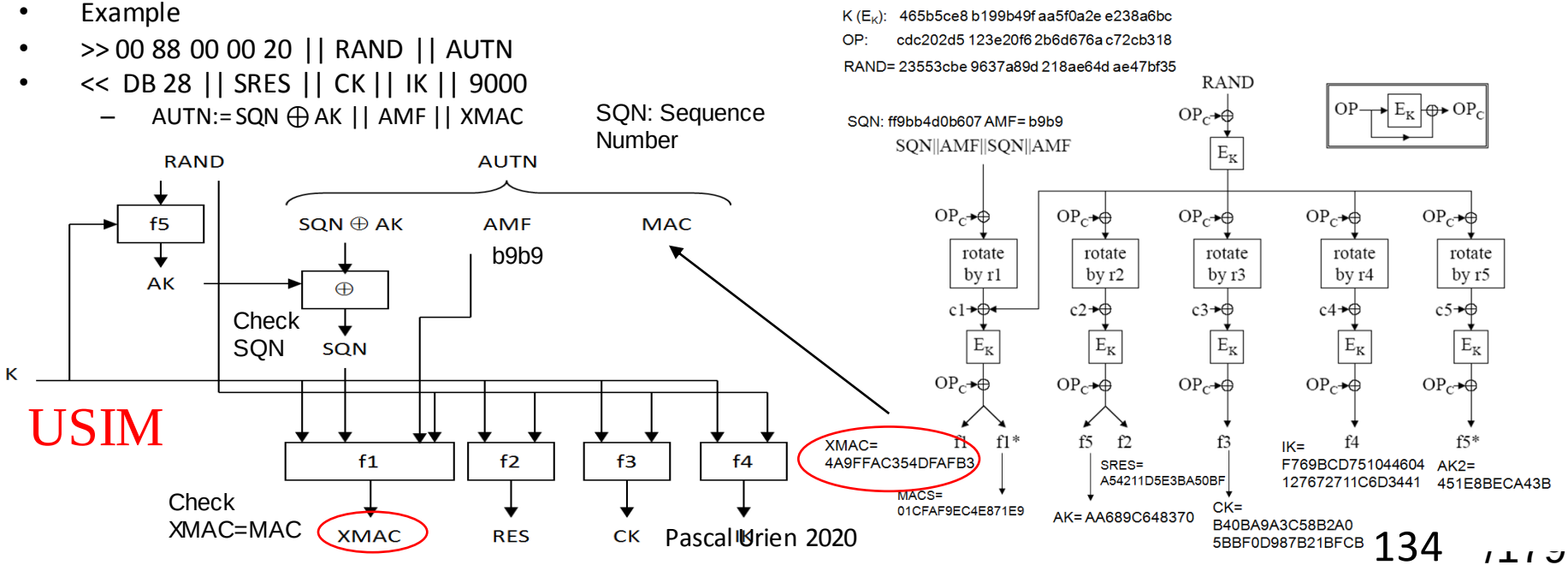
EXAMPLE: NXP PN532



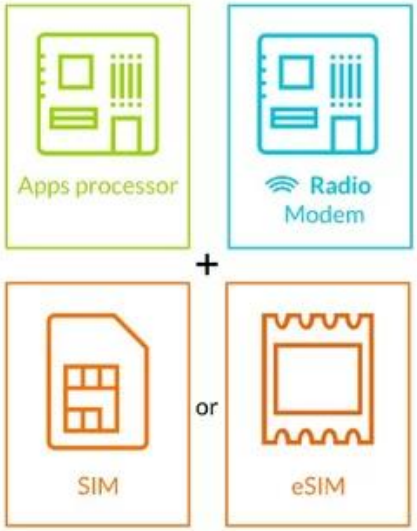
	NFC secure modules
Product features	PN65L
Embedded NFC IC	PN532
Available host interfaces	serial, SPI, I²C
Embedded Secure IC	P5CN072
OS for secure device	JCOP or 3rd party
Stacked passive component IC	yes
Package thickness	1.2 mm
Package size	7x7 mm²
Package type	HLQFN48

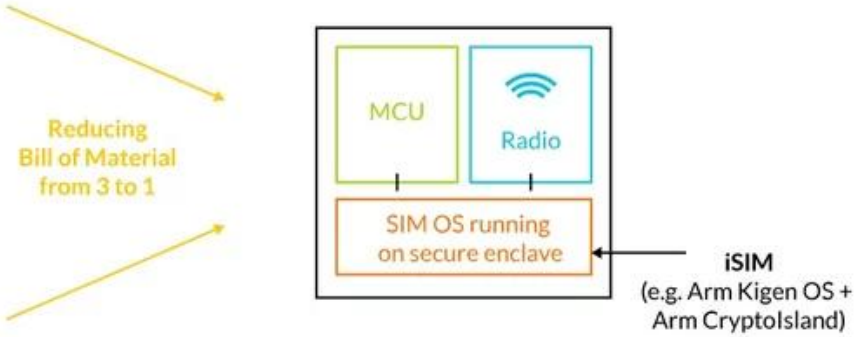
USIM for 3G/4G radio networks

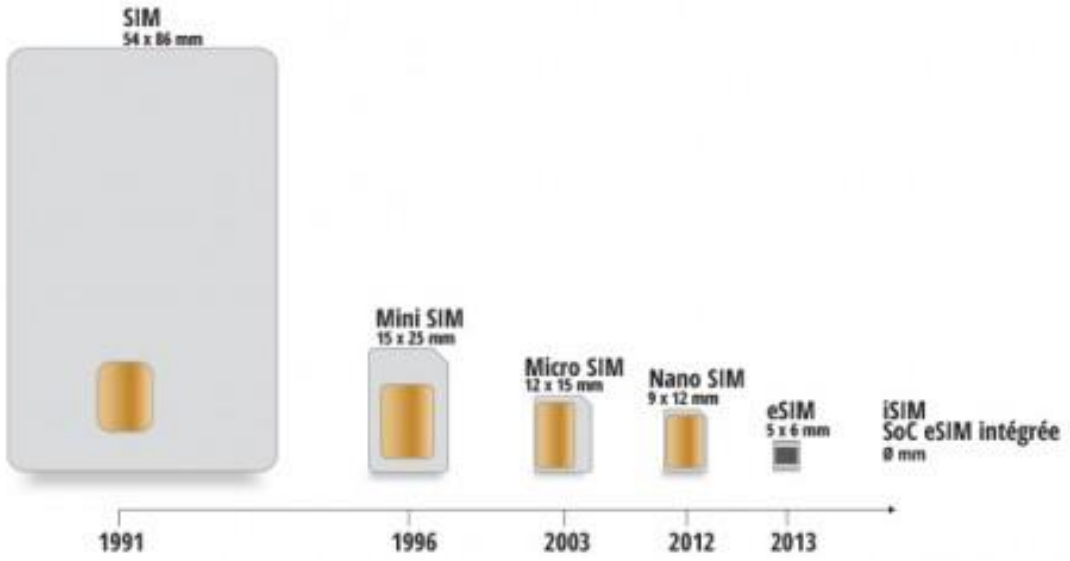
- UICC modules host USIM applications
  - The list of USIM applications is stored in the EF-DIR file
- At least two concurrent applications may run at the same time
  - The application index is identified by the last two bits of the CLA byte
- 3G/4G authentication is performed thanks to the AUTHENTICATE (INS=88) command
- Example
- >> 00 88 00 00 20 || RAND || AUTN
- << DB 28 || SRES || CK || IK || 9000
  - AUTN:= SQN ⊕ AK || AMF || XMAC



(U)SIM, eSIM,iSIM



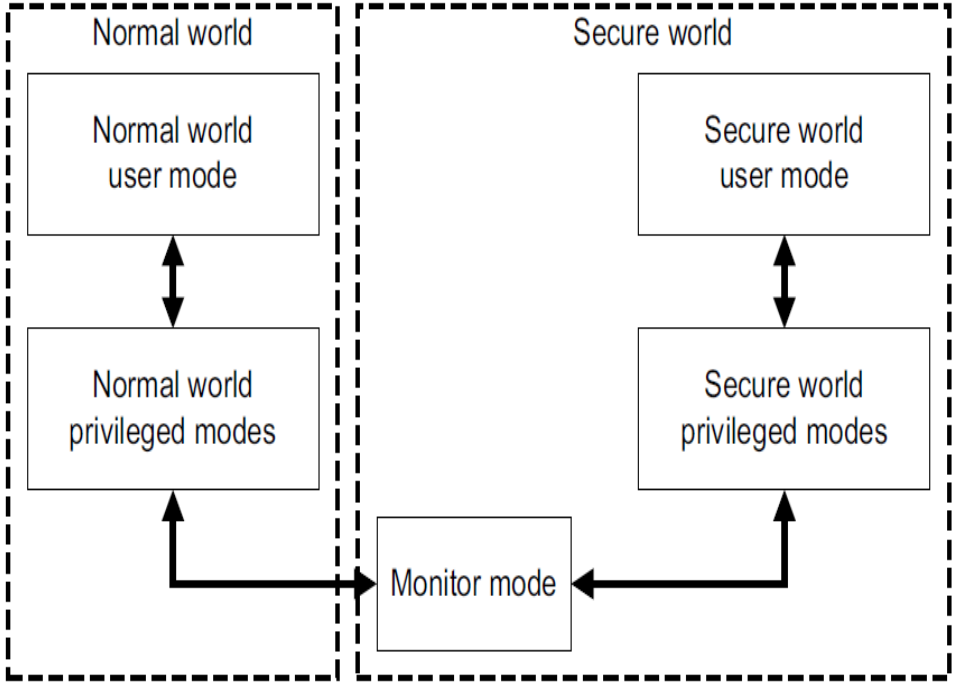




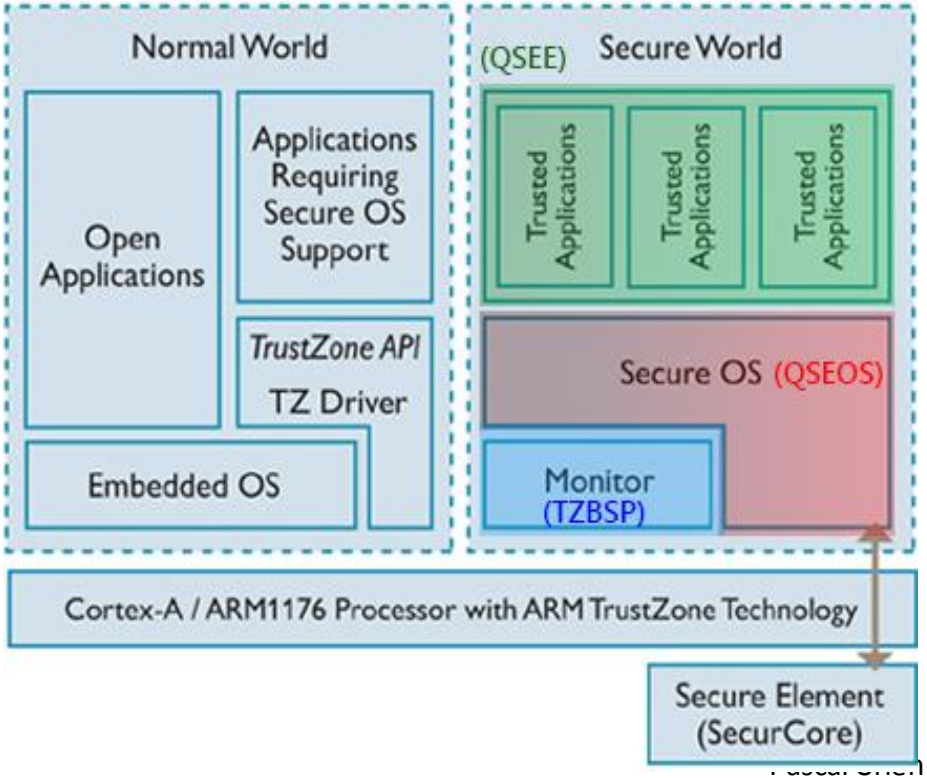
ARM, 2018

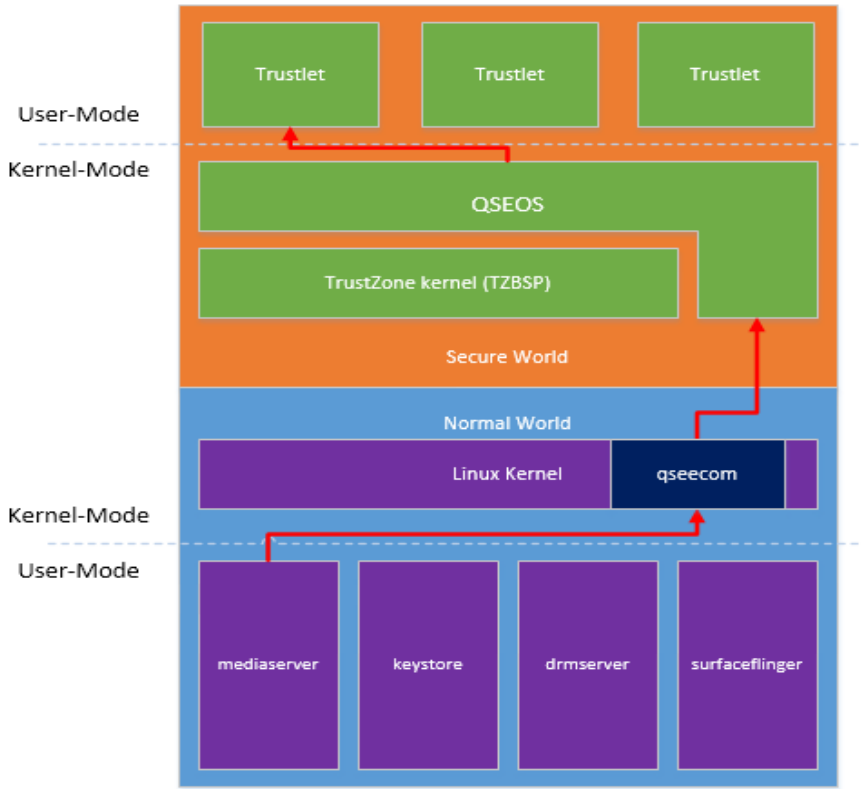
Trusted Execution Environment

- Trusted Zone is an ARM patent
- A processor with two modes:
  - Normal Mode
  - Secure Word
- A monitor entity manages the environment switch between the
- Normal and the Secure word
- Each mode has its own interrupt procedures and a set of registers.
- Never less CPU and memories (SRAM, ROM) are common
- The SRAM size is about 10KB and the ROM size is about 128 KB
- An fuse OTP memory of about 256 bits, stores the hash of a public key
- Trusted Application (Trustlet) are signed



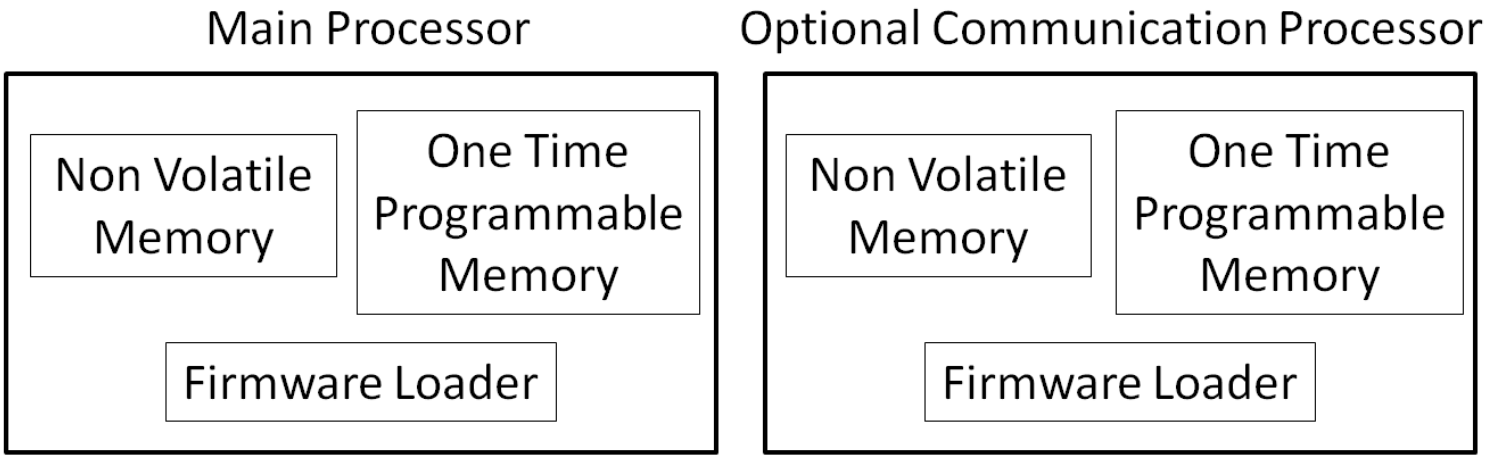
Android QSEE (*Qualcomm's Secure Execution Environment)*



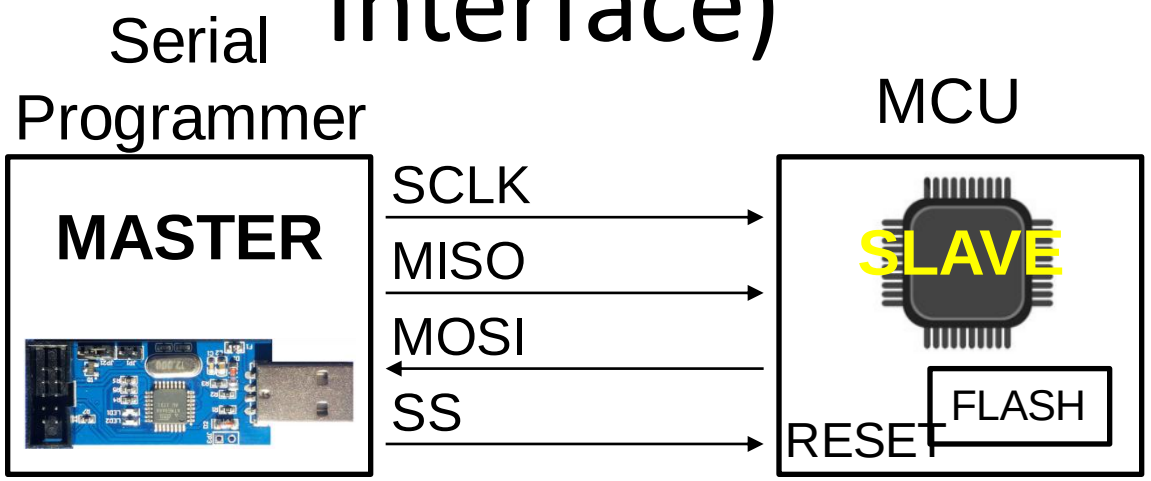


Device Integrity

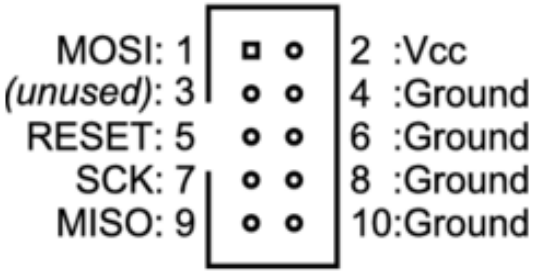
IoT Architecture

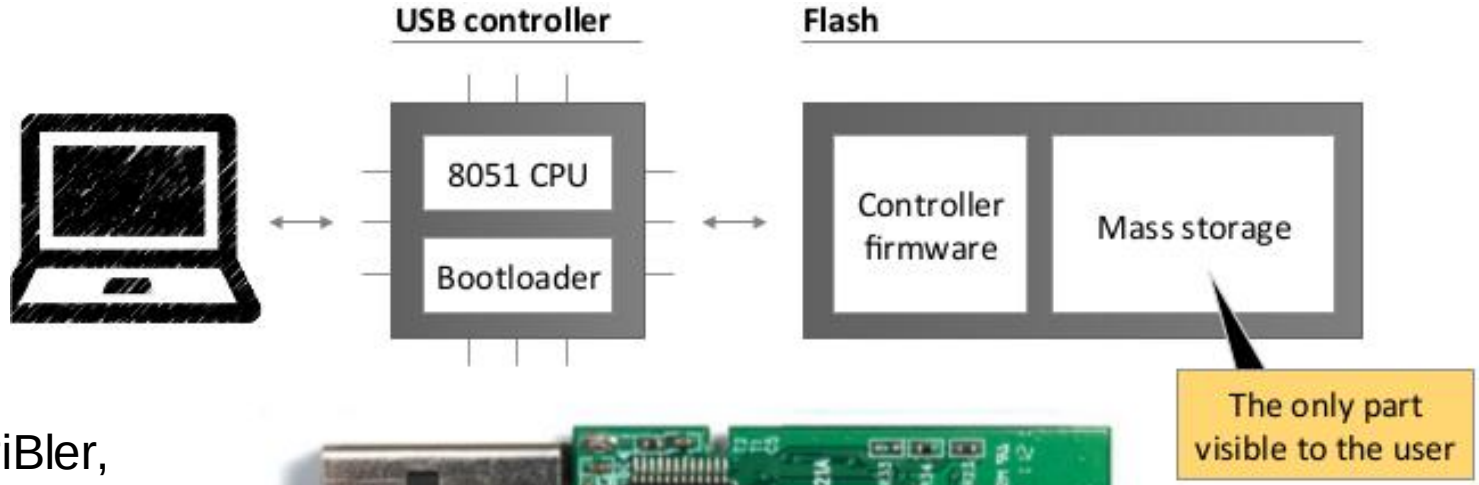


About SPI (Serial Peripheral Interface)

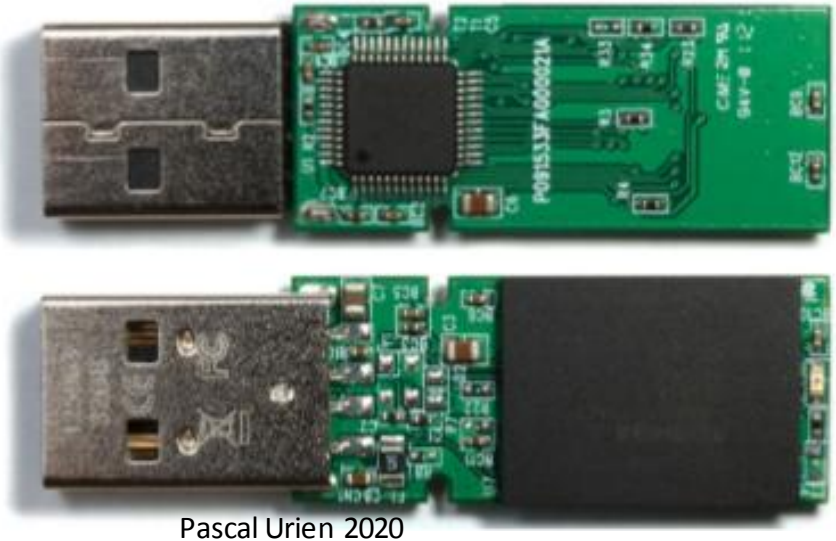


SCLK (Serial Clock)
MOSI (Master Out Slave In)
MISO (Master In Slave Out)
SS (Slave Select)



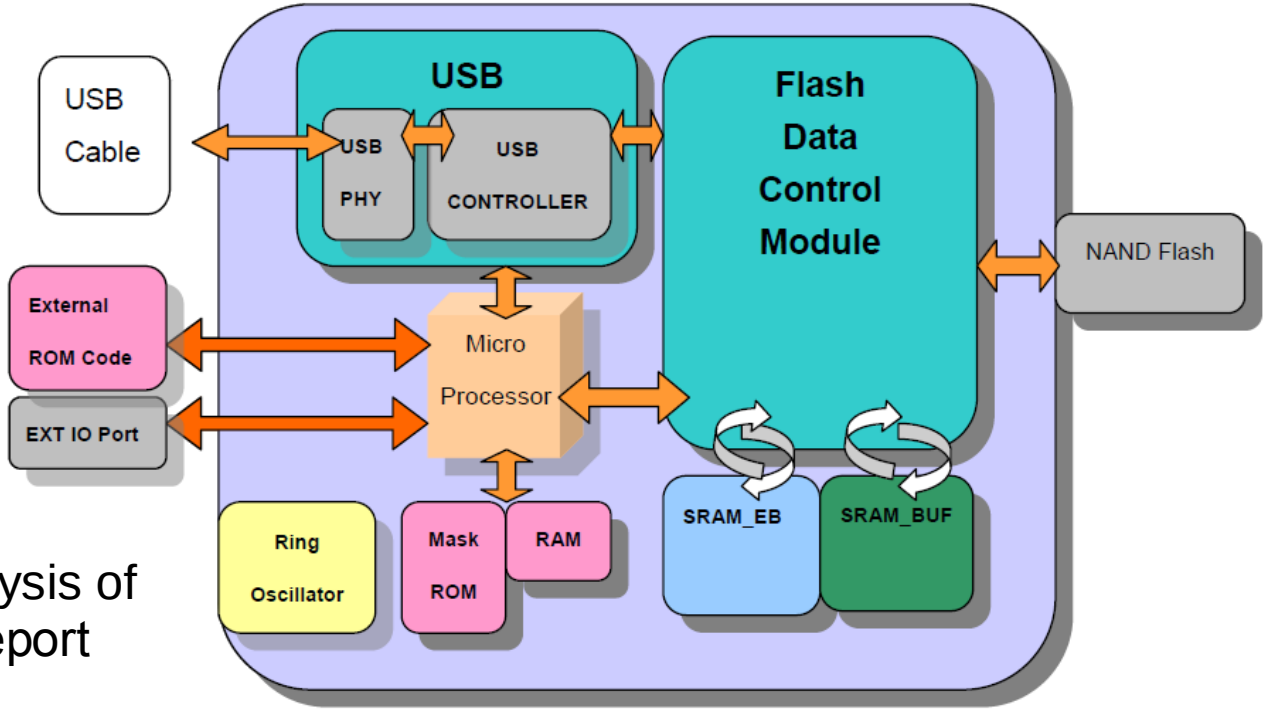


Nohl, K., KriBler, S., Lell, J. 2014. "BadUSB - On Accessories that Turn Evil", Blackhat 2014 USA



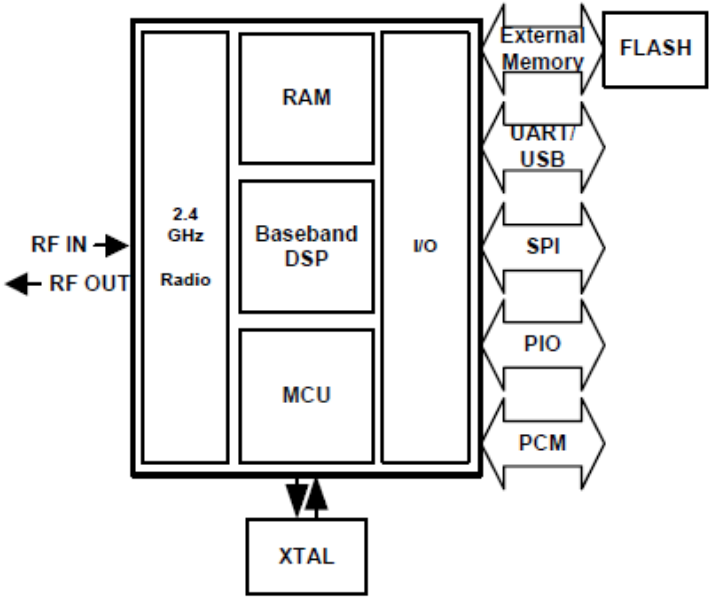
FLASH Controller PS2251-33 *Phison Electronics Corporation*

Bootloader in ROM
Firmware in FLASH



Jago, D., 2018, "Security analysis of USB drive", Master's thesis report

CSR BC417143 - HC05/HC06 Bluetooth module



System Architecture

No ROM
SPI programming for the first 1Mbit (Bootloader)

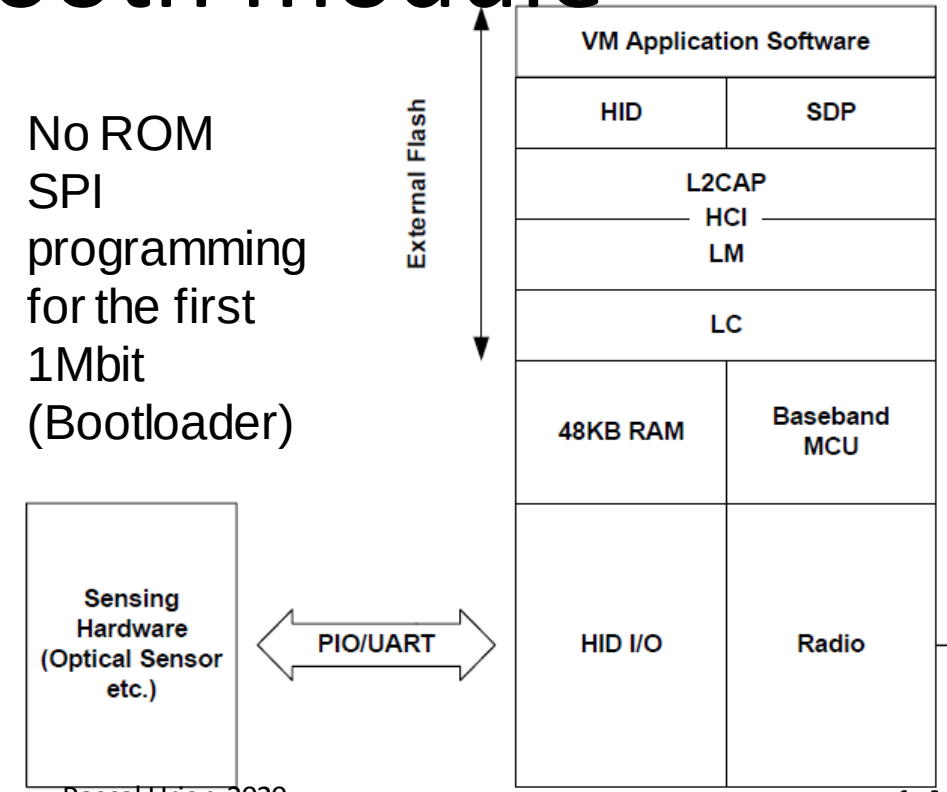
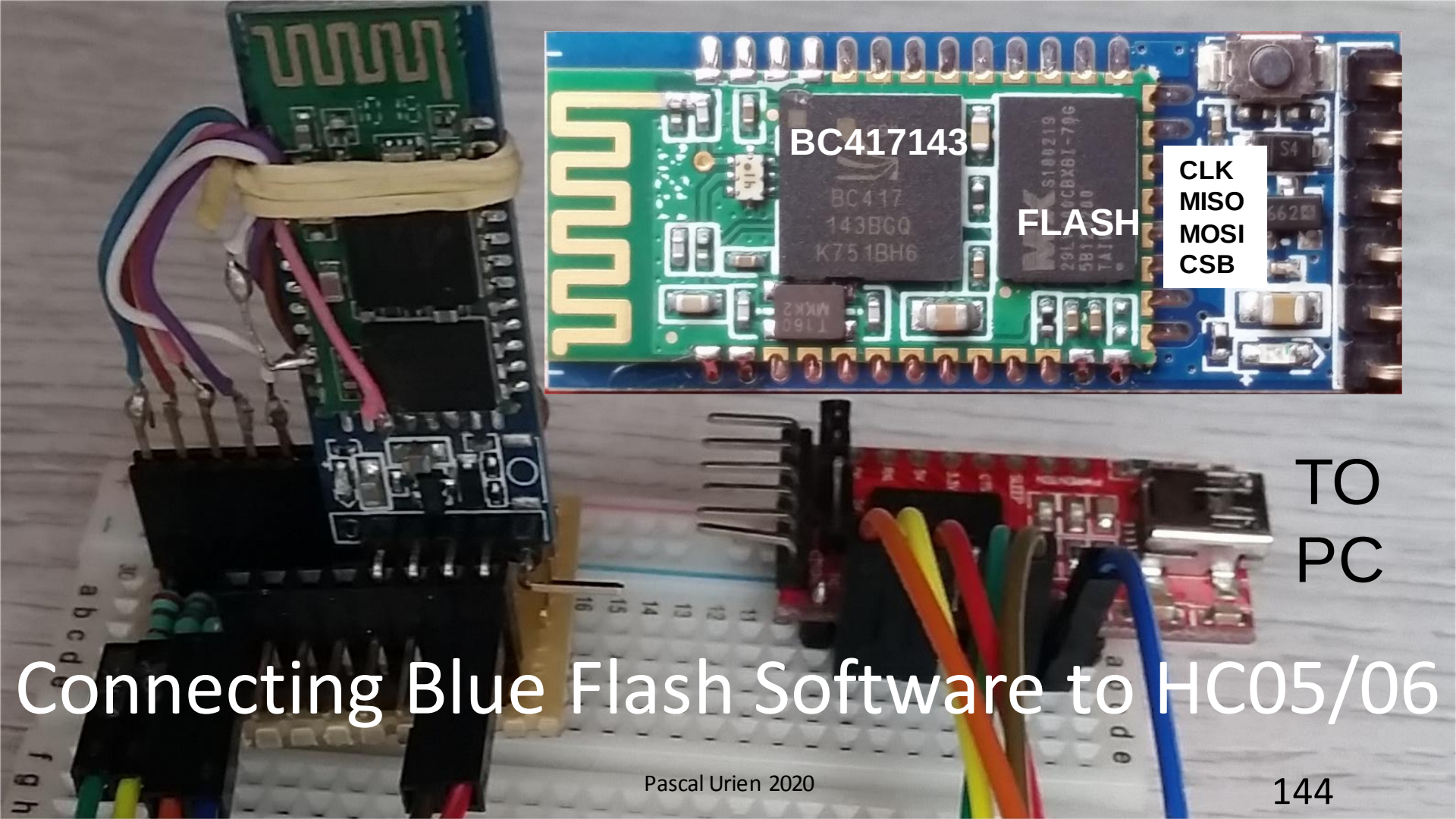
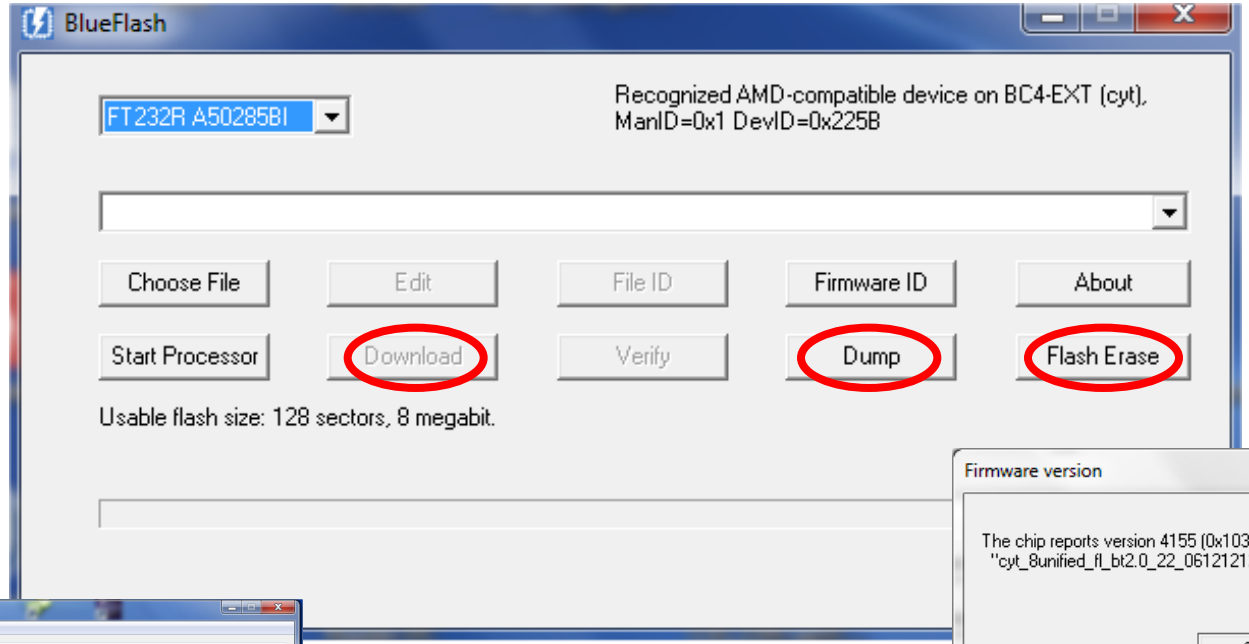


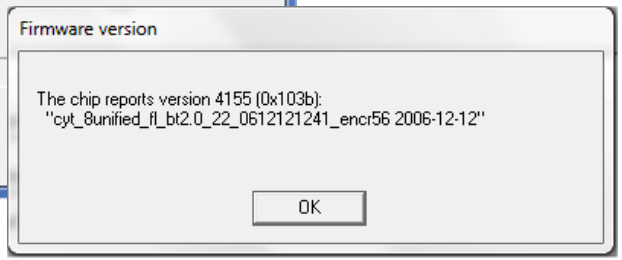
Figure 9.4: HID Stack

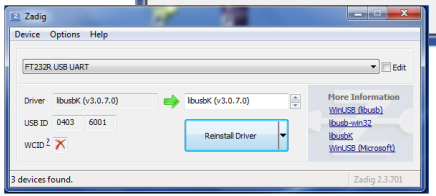


Connecting Blue Flash Software to HC05/06

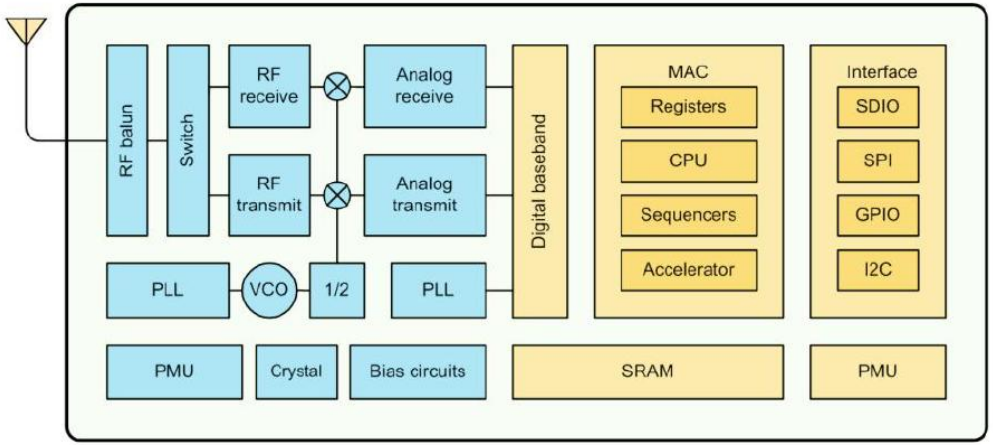
Blue Flash Software version 2.62

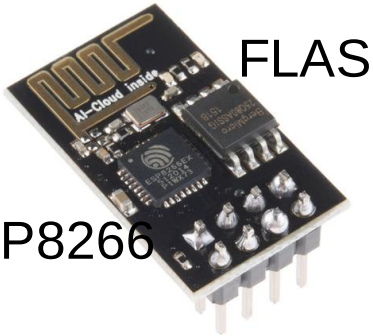






ESP8266

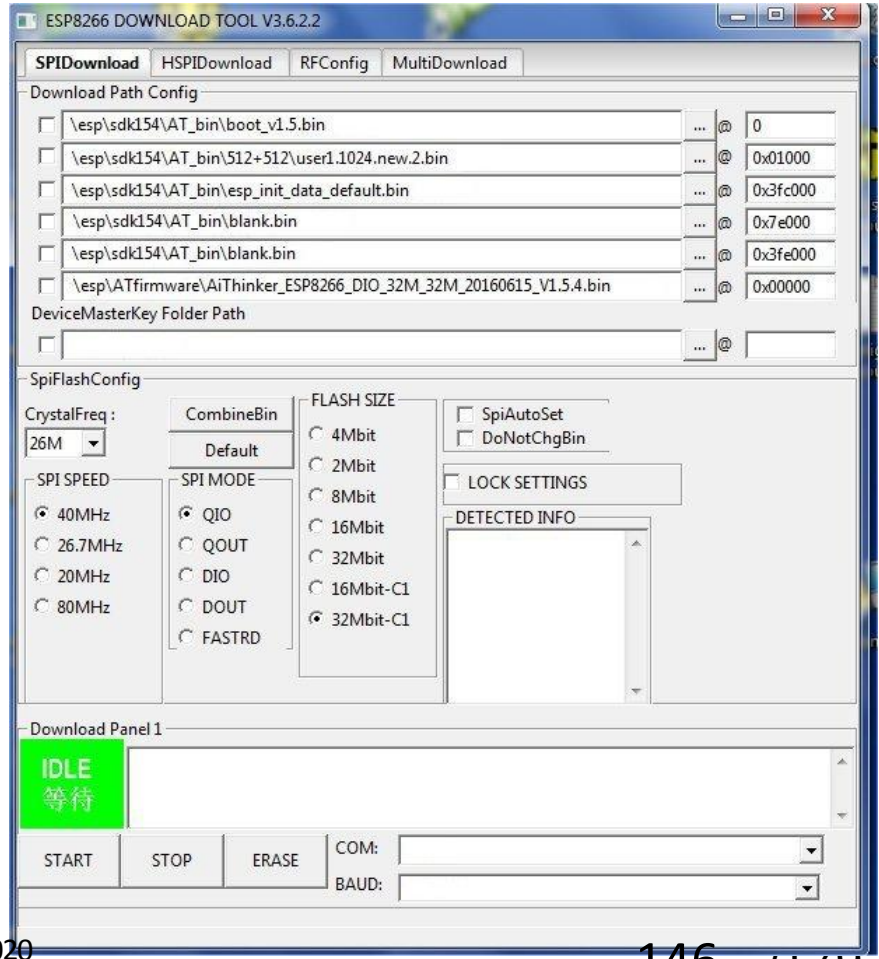




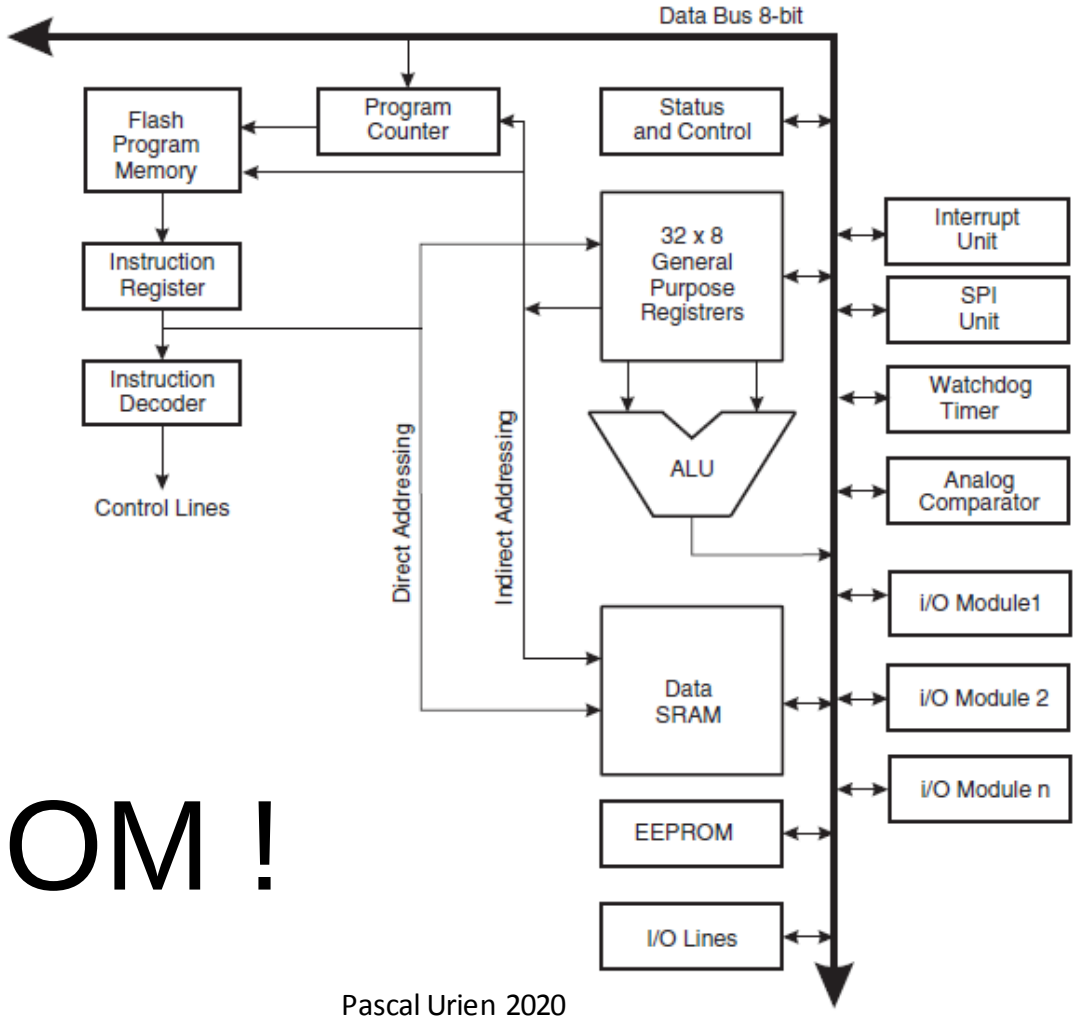
FLASH

ESP8266

64KB ROM
BOOTLOADER



AVR

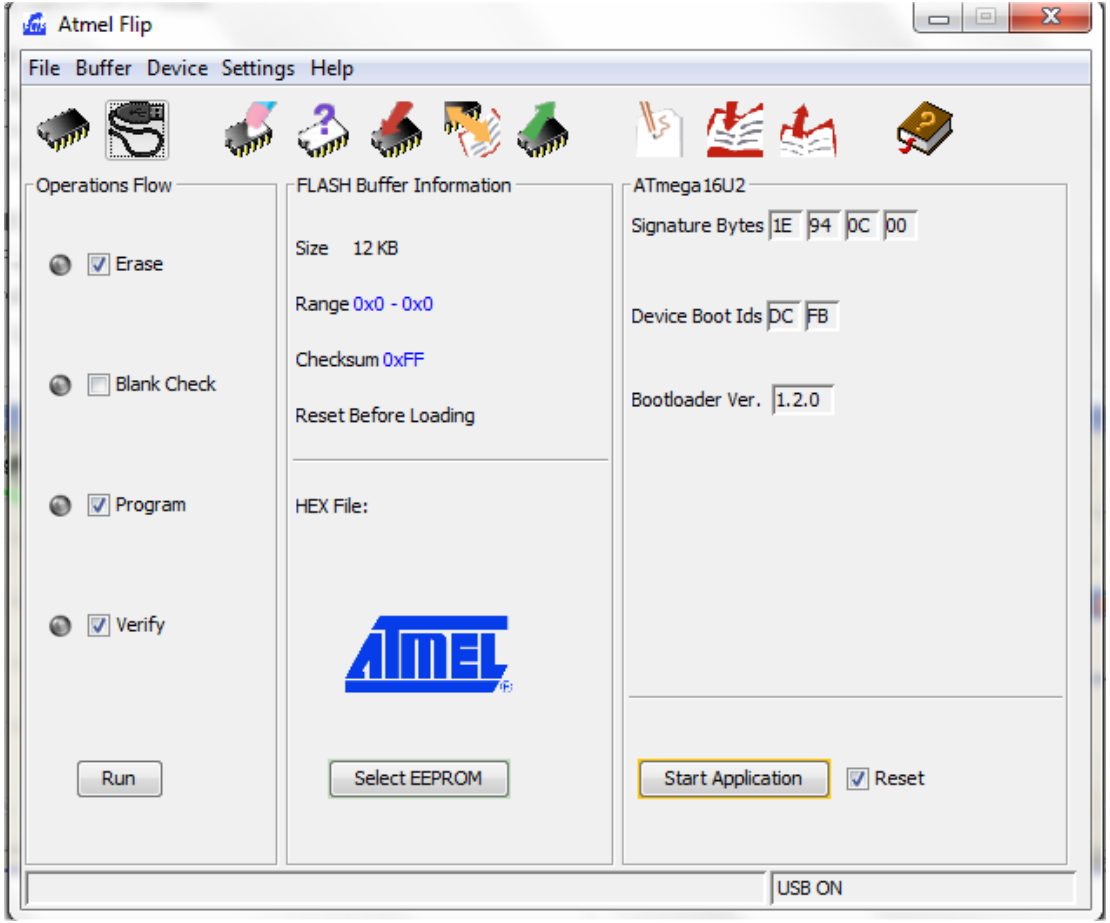

Data Bus 8-bit
Flash Program Memory
Program Counter
Status and Control
Instruction Register
Instruction Decoder
Control Lines
32 x 8 General Purpose Registrers
ALU
Direct Addressing
Indirect Addressing
Data SRAM
EEPROM
I/O Lines
Interrupt Unit
SPI Unit
Watchdog Timer
Analog Comparator
i/O Module1
i/O Module 2
i/O Module n

No ROM !

AVR : locks & Fuses

All memories are erased

Serial Programming (SP) Requires RESET signal	Memories Access	Fuse: Disable Reset SP & PP
Parallel Programming (PP) Requires RESET signal	Fuses Access	Fuse: Disable SP from PP only
High Voltage Programming clear under any conditions	Bootloader Configuration	Fuse: Others Voltage, Clock



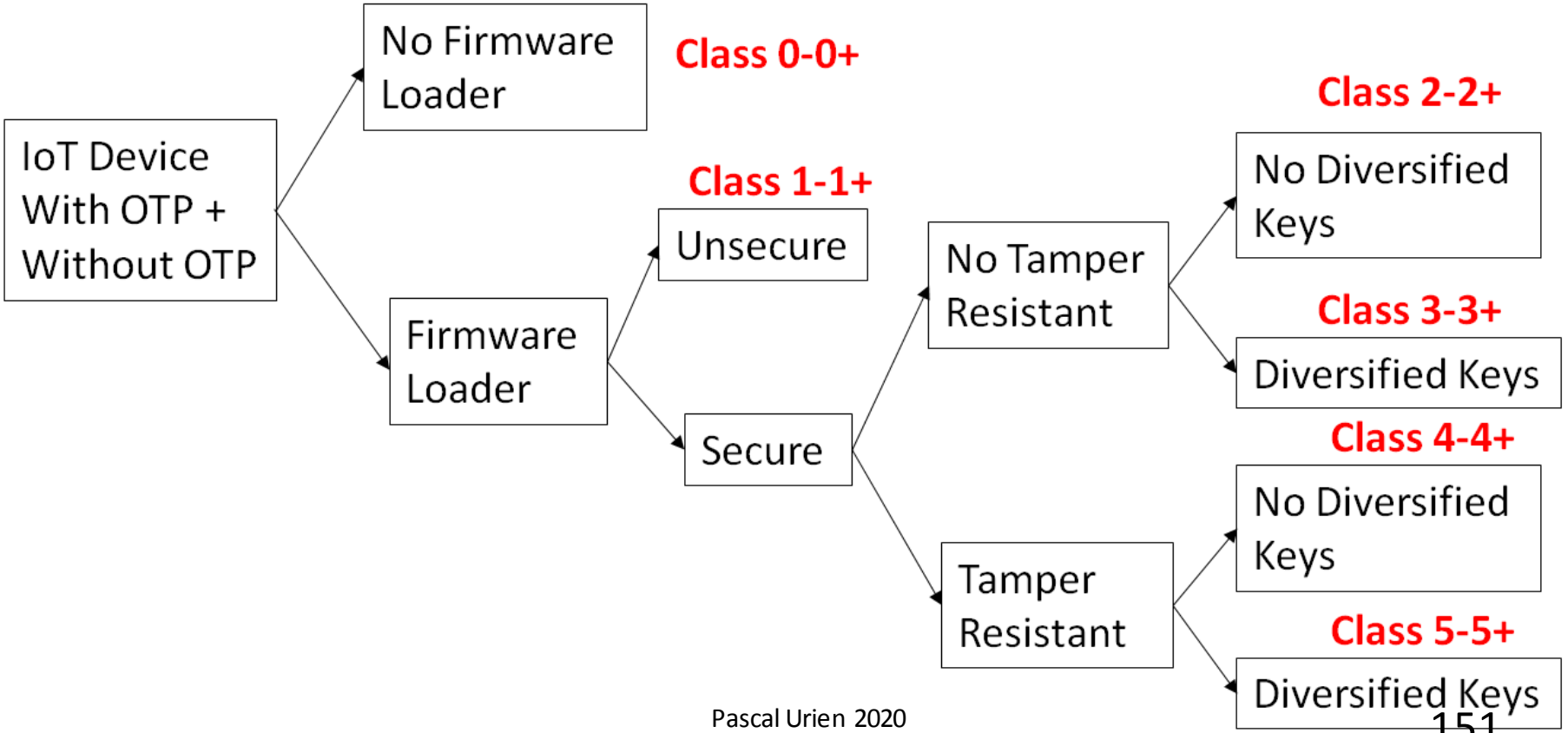
Atmel DFU

Device
Firmware
Update

Security attributes

- One Time Programming (OTP)
- Firmware Loader (FLD)
- Secure Firmware Loader (FLD-SEC)
- Tamper Resistant Key (TRT-KEY)
- Diversified Key (DIV-KEY)

Security Classes



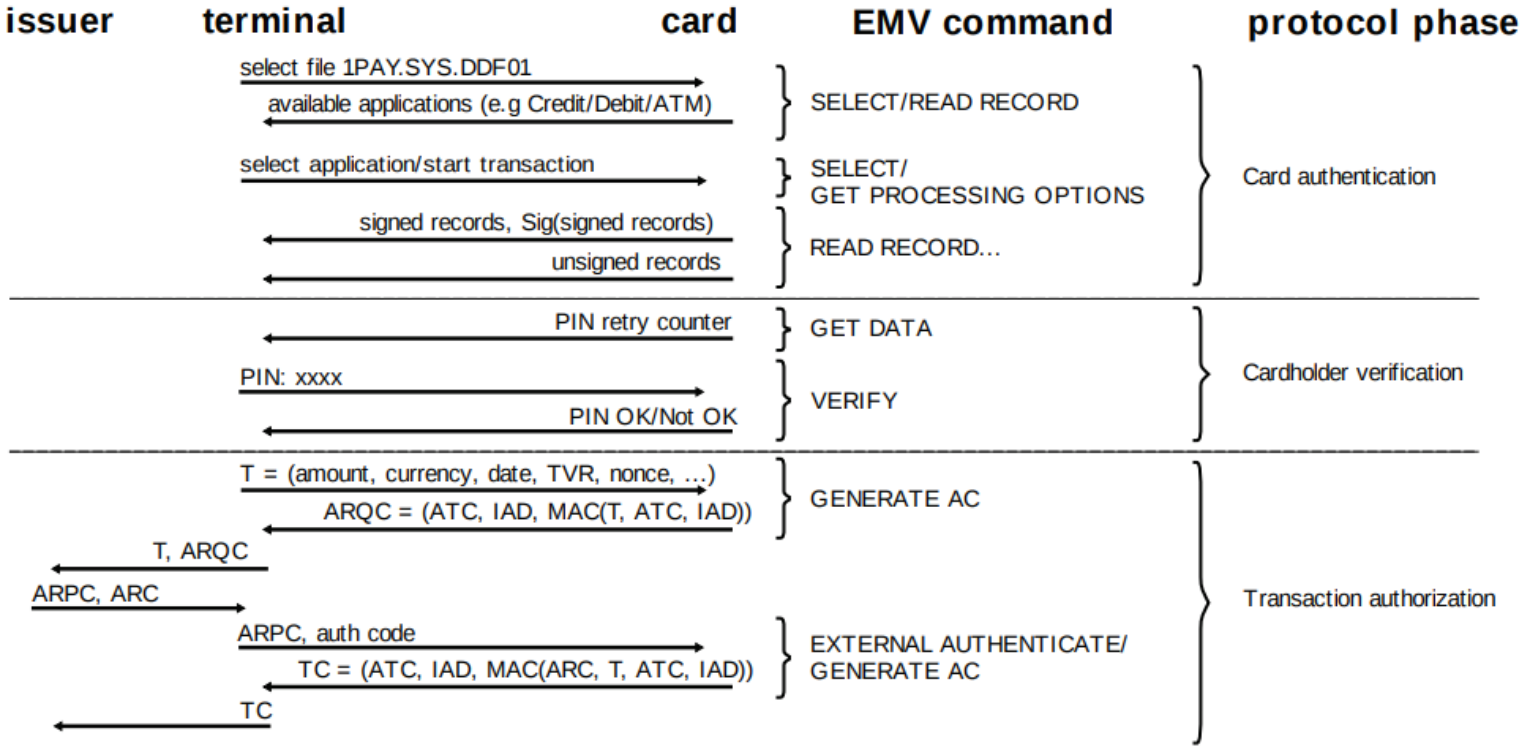
Examples

- AVR micro-controller units (MCU) without bootloader belong to Class0.
- AVR micro-controller units (MCU) with bootloader (for example the "*Arduino*" family) belong to Class1.
- USB flash drives embedding ROM and bootloader belong to Class1+.
- Philips hue smart bulbs belong to Class2; they embed secure bootloader with a shared single symmetric key.
- The IETF working group SUIT (*Software Updates for Internet of Things*) target Class2+ devices embedding secure bootloader with public key, and Class3+ devices supporting bootloader and diversified symmetric key.
- Class4+ is a tamper resistant enhancement of Class2+, it could address devices compatible with SUIT of which public key cannot be modified.
- Highly secure devices such as bank cards belong to Class5+.

Relay Attacks

https://youtu.be/1pMuV2o4Lrw

Chip and PIN is Broken

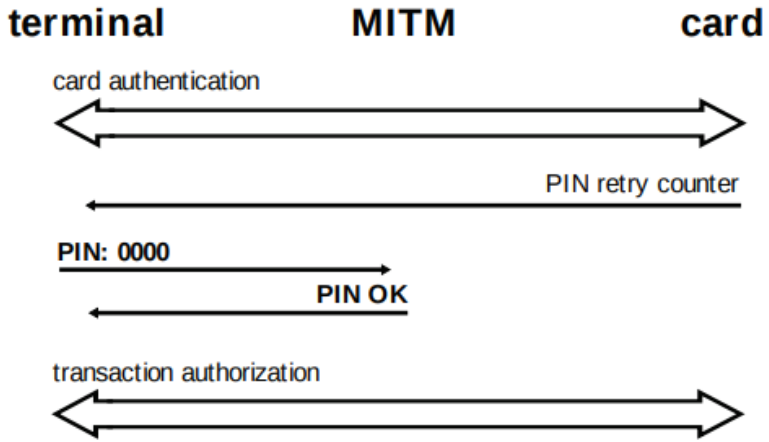


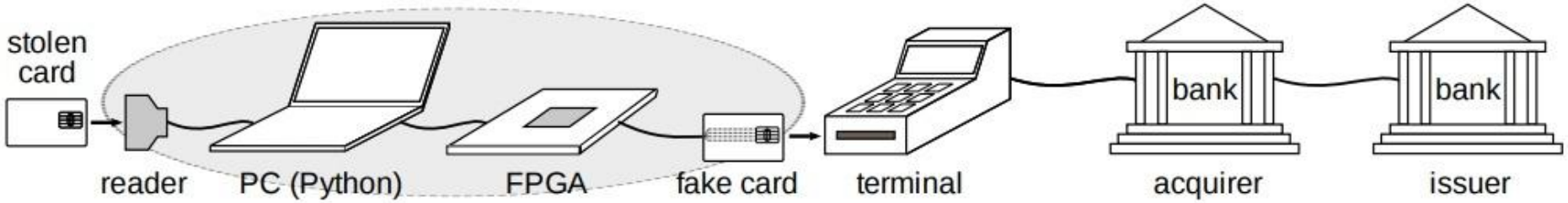
Steven J. Murdoch, Saar Drimer, Ross Anderson, Mike Bond, "Chip and PIN is Broken", 2010 IEEE Symposium on Security and Privacy

Chip & PIN is Broken

IAD FORMAT, BYTE 5 (BITS 4–1) FROM A VISA VERSION 10 CRYPTOGRAM [8, APPENDIX A-13, P222].

Bit	Meaning when bit is set
4	Issuer Authentication performed and failed
3	Offline PIN performed
2	Offline PIN verification failed
1	Unable to go online




stolen
card
reader
PC (Python)
FPGA
fake card
terminal
bank
acquirer
bank
issuer

FUN MODULE

- In May 2011, the French's bankers Economic Interest Group (GIE Cartes Bancaires) noted that a dozen EMV cards, stolen in France a few months before, were being used in Belgium. A police investigation was thus triggered.

"When Organized Crime Applies Academic Results A Forensic Analysis of an In-Card Listening Device"
Houda Ferradi, Rémi Géraud, David Naccache, and Assia Tria,
October 2015, Journal of Cryptographic Engineering

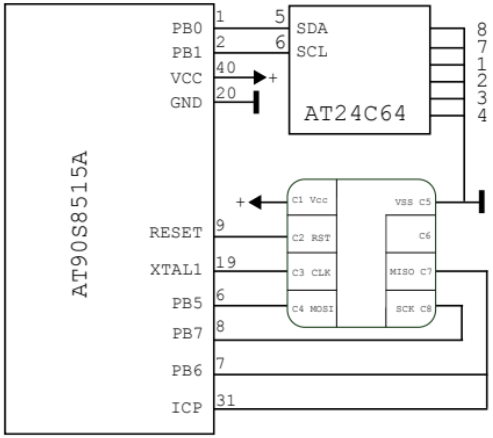
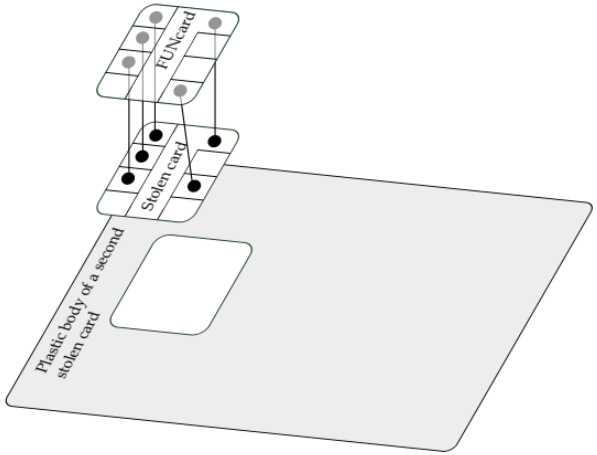
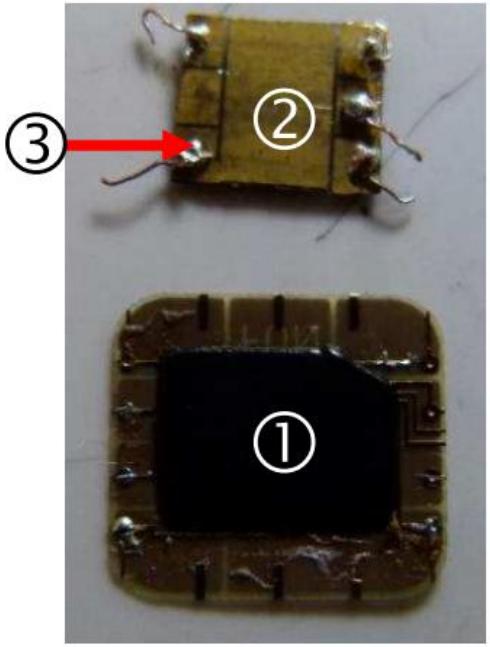


Fig. 5. The FUN card's inner schematics.





g. 16. (1) FUN card module; (2) genuine stolen card; (3) welded wire.

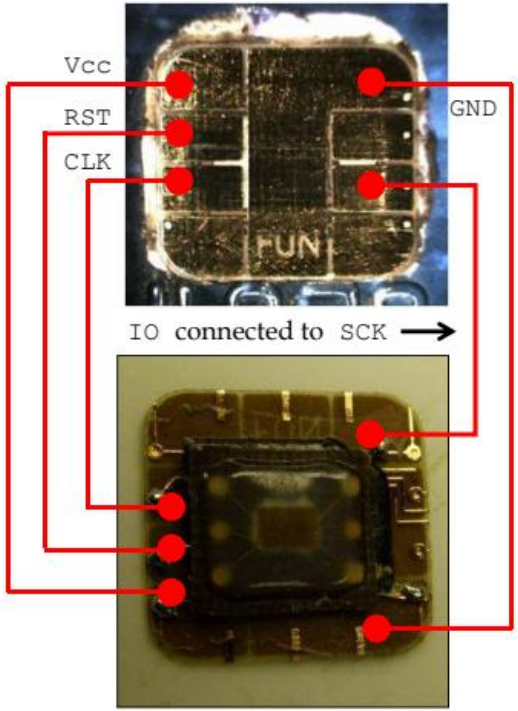
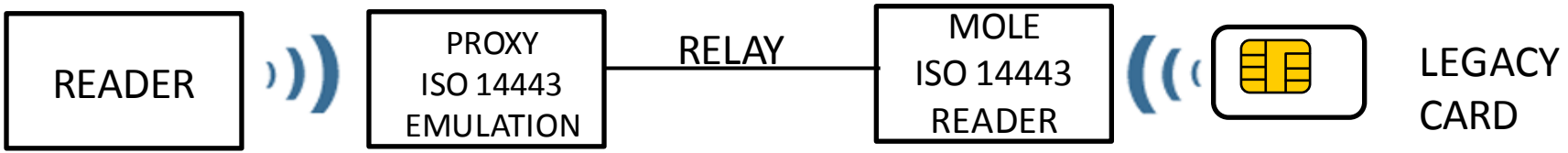


Fig. 18. Wiring diagram of the forgery.

Relay Attack



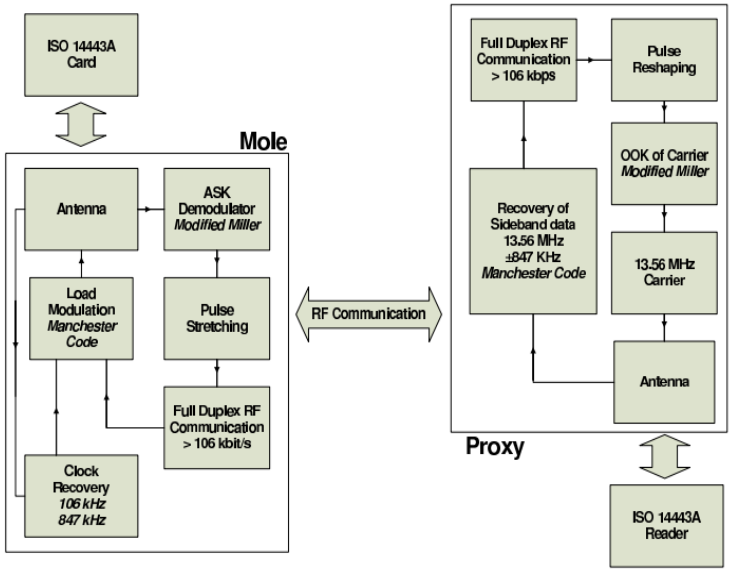
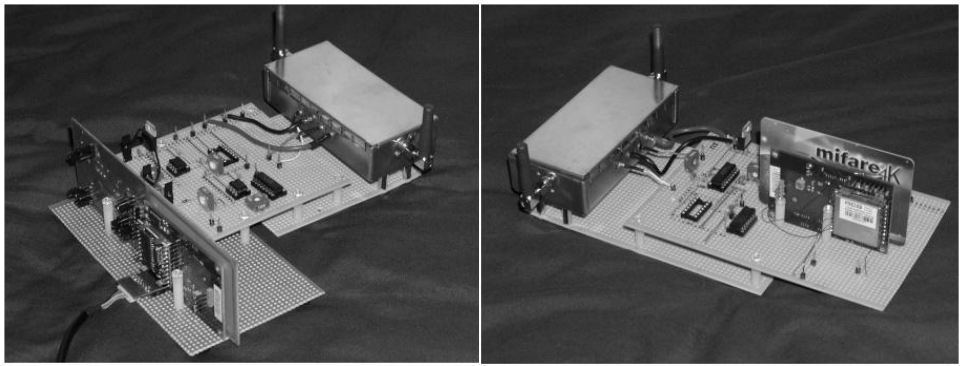


Fig. 1. Functional analysis of the relay system

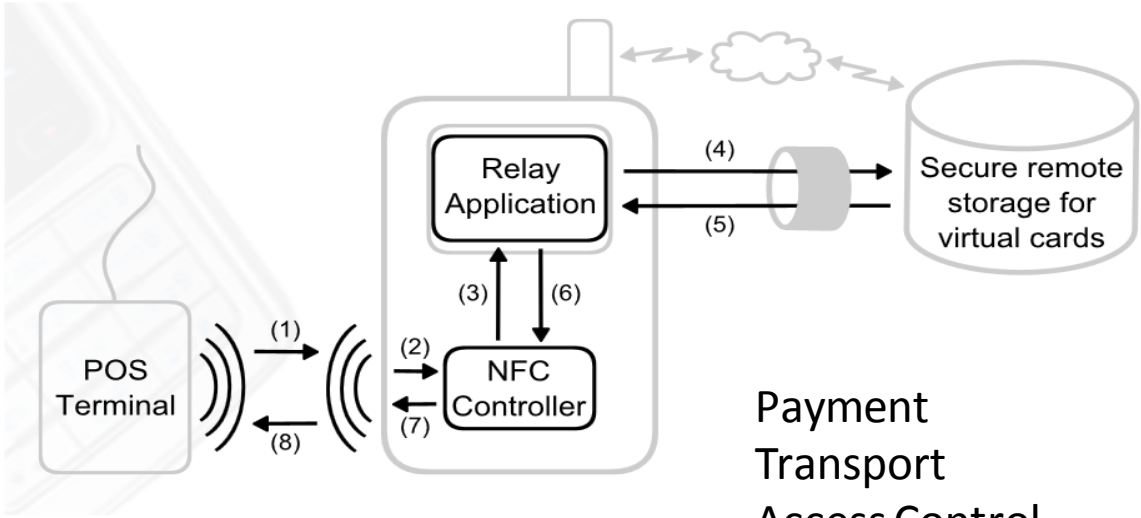
- Gerhard Hancke "A Practical Relay Attack on ISO 14443 Proximity Cards", 2005



(a) Proxy (b) Mole

Relays Attacks

- The software card emulation in smartphone enables a new generation of relay attack.

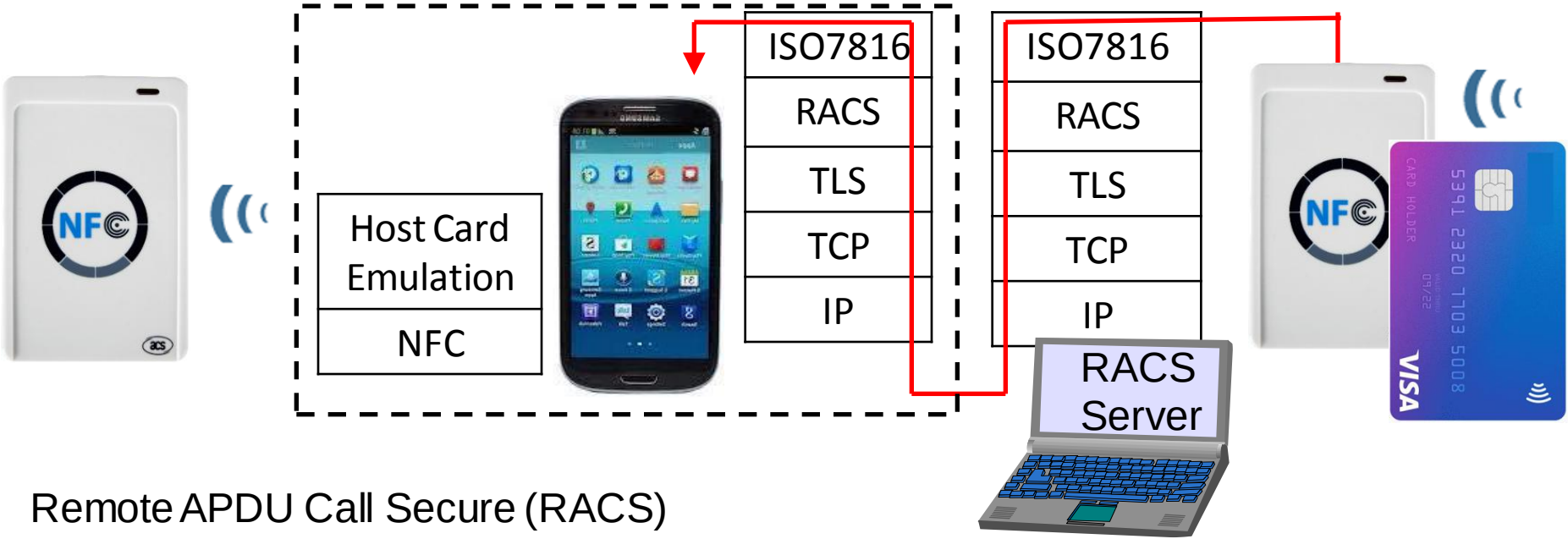


Payment
Transport
Access Control

Gerhard P. Hancke, Keith Mayes et Konstantinos Markantonakis, « Confidence in smart token proximity: Relay attacks revisited », *Computers & Security*, 2009

Roland, M., "Software Card emulation in NFC-enabled Mobile Phones: Great Advantage or Security Nightmare ?", in proceedings of WSSI/SPMU, 2012.

HCE Relay (RACS)



Remote APDU Call Secure (RACS)
draft-urien-core-racs-00

Implants

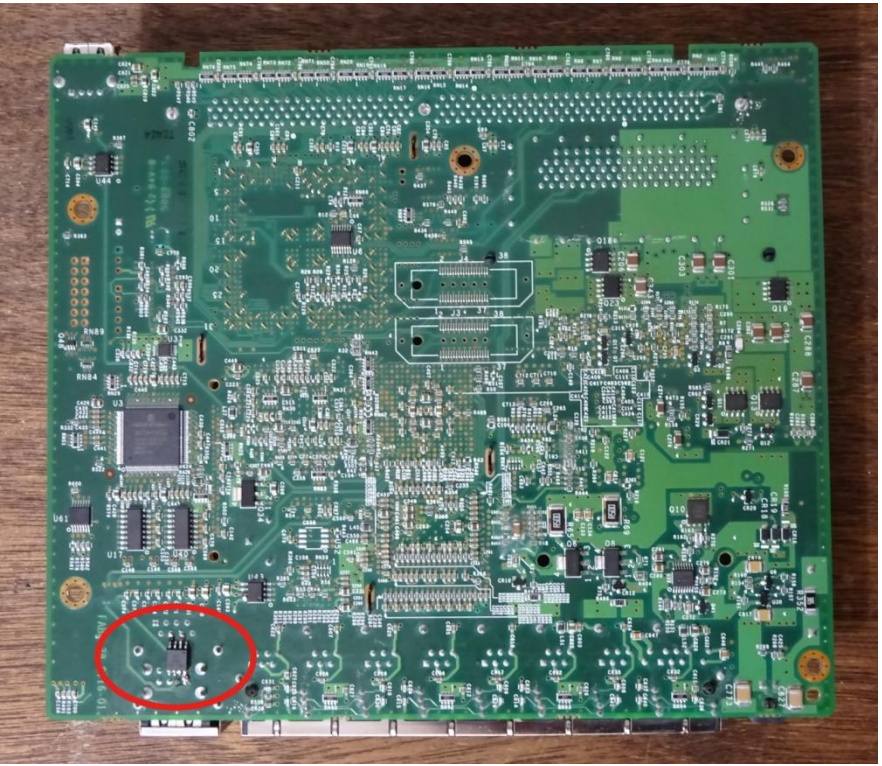
Haunted Cable

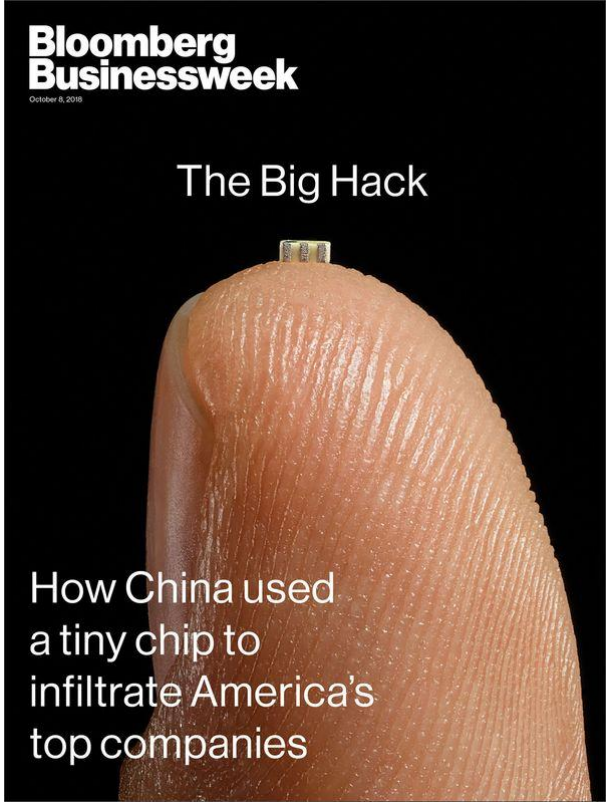
- The Haunted USB Cable
  - http://imakeprojects.com/Projects/haunted-usb-cable/, ATTINY45
- USB Keyboard Injector
  - https://github.com/whiteneon/haunted-usb-1.0, ATTINY85
- USBASP Keyboard
  - https://github.com/Uxio0/usbAspKeyboard, ATMEGA8
  - Simulates a USB HID keyboard that prints out "All work and no * play makes Jack a dull boy." This code is only a slightly * modified version of Frank Zhao's "USB Business Card" with some * code added from Donald Papp's "Haunted, Mystery USB Device" * both of which are based on Christian Starkjohann's "V-USB".
- KEYBOARD + MOUSE Injector

Caps Lock Attack

1) A user keys in CAPS lock (or NUM lock) key.
2) The keyboard firmware sends the keycode for the key to the PC with an input report.
3) The PC sends back an output report for the LED, after accepting above input report.
4) The firmware lights the LED, specified by the output report.

Implants Attack (2019)





CS3sthlm, 2020
Monta Elkins
Nation-State Supply Chain Attacks for Dummies and You Too

CISCO ROUTER ASA5505

CISCO Attack

- Escape Characters, confreg 0x41
- Boot , Enable
- copy startup-config running-config
- Enable SSH, Add account, Add routing
- no config-register
- ping attack address as notification
- https://www.cisco.com/c/en/us/td/docs/security/asa/asa93/configuration/general/asa-general-cli/basic-hostname-pw.html

Assistant Light Attack

Light Attack (2019)

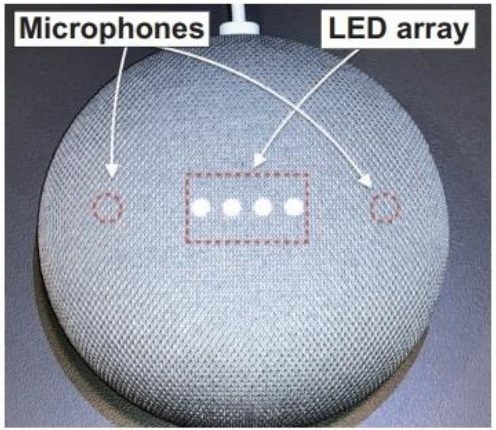


Fig. 9. Google Home Mini. Notice the cloth-covered microphone ports.

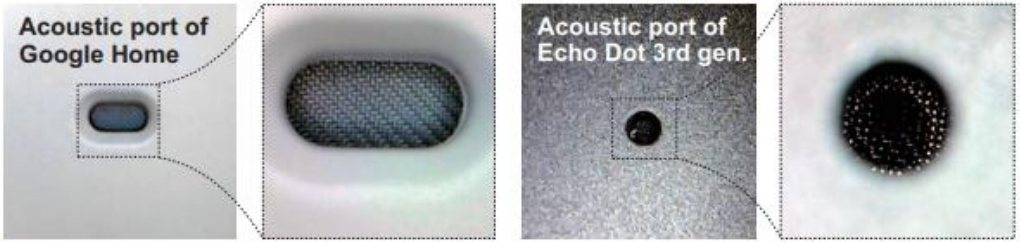


Fig. 3. Acoustic port of (Left) Google Home and (Right) Echo Dot 3rd generation. The ports are located on the top of the devices, and there are meshes inside the port.

Takeshi Sugawara, Benjamin Cyr, USara Rampazzi, Daniel Genkin, Kevin Fu "Light Commands: Laser-Based Audio Injection Attacks on Voice-Controllable Systems", 2019

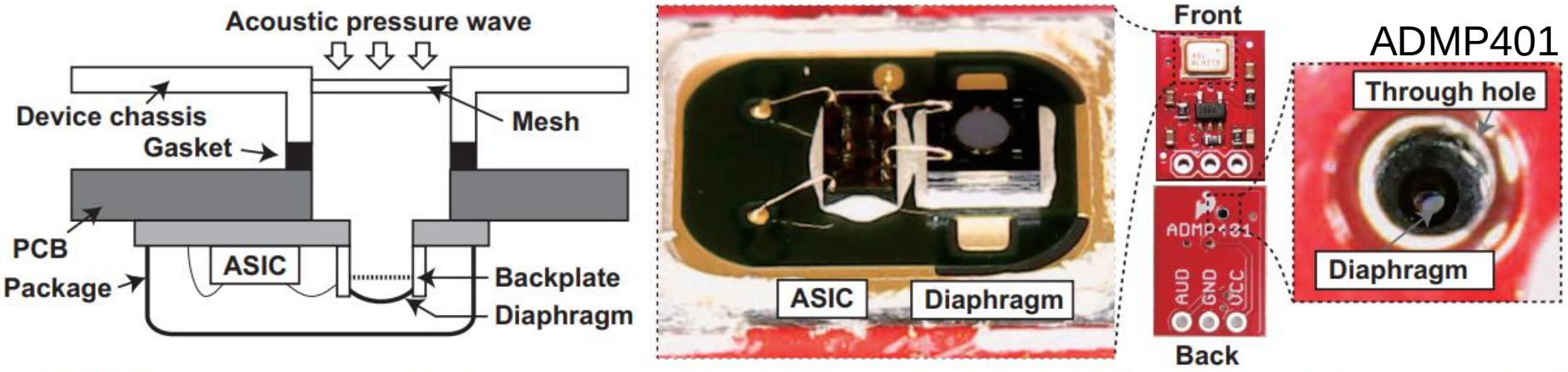
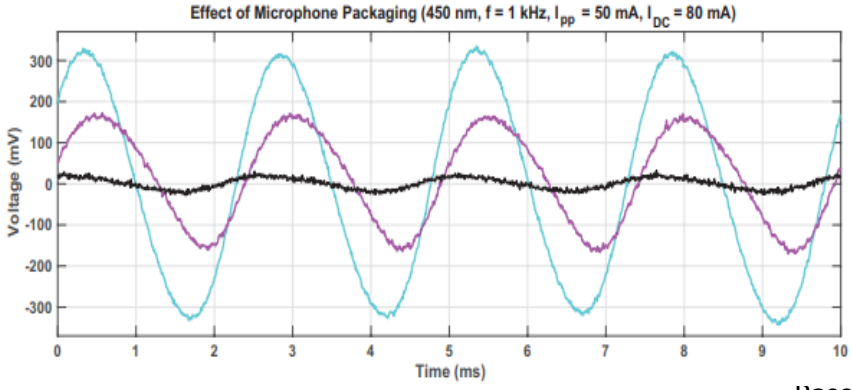
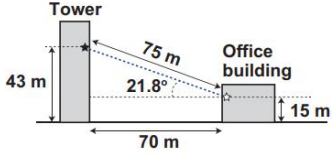
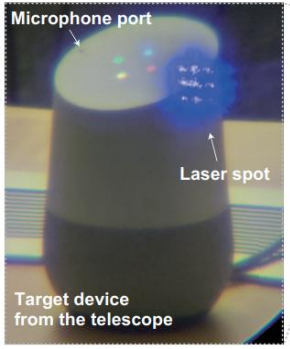
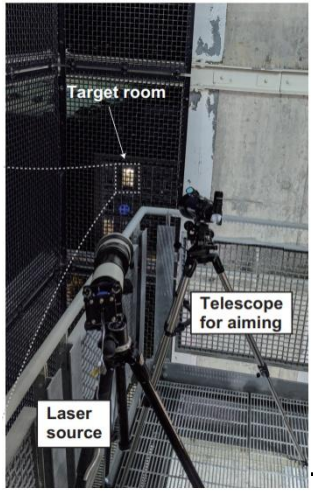


Fig. 2. MEMS microphone construction. (Left) Cross-sectional view of a MEMS microphone on a device. (Middle) A diaphragm and ASIC on a depackaged microphone. (Right) Magnified view of an acoustic port on PCB.



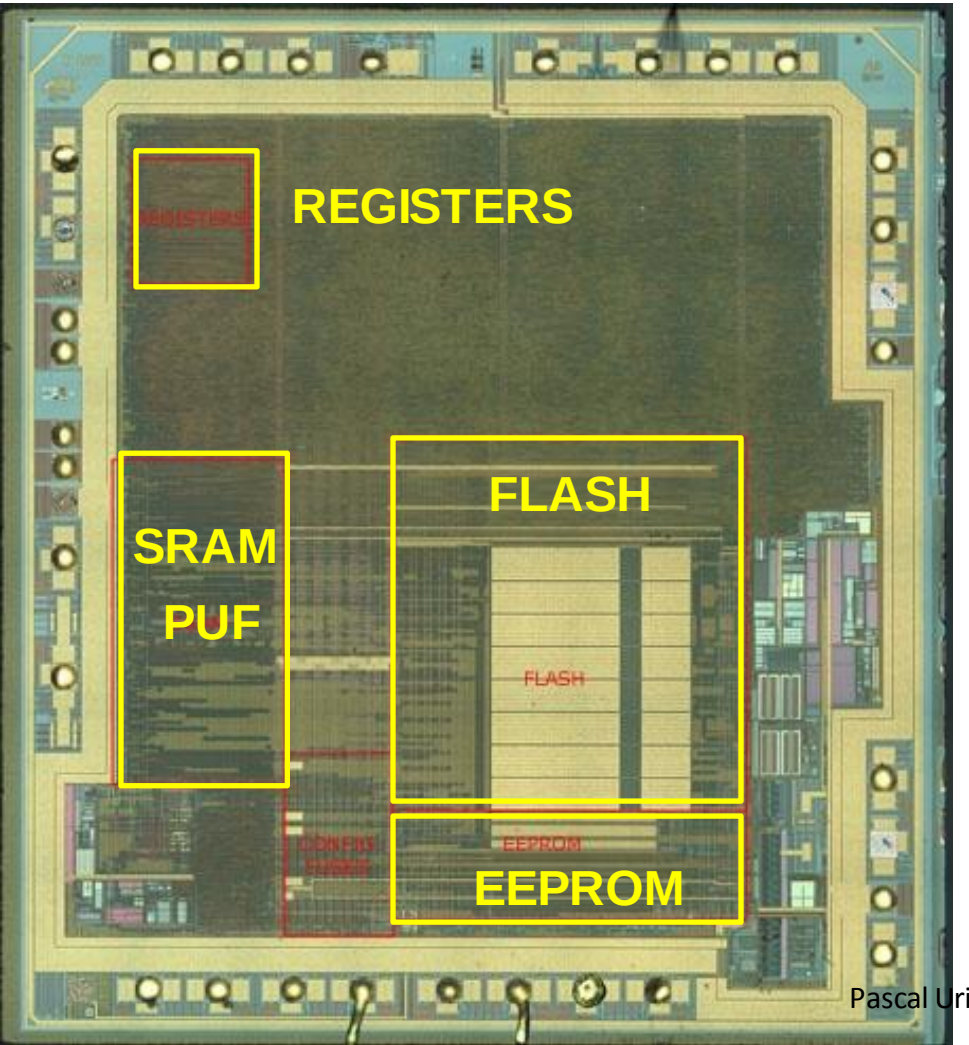


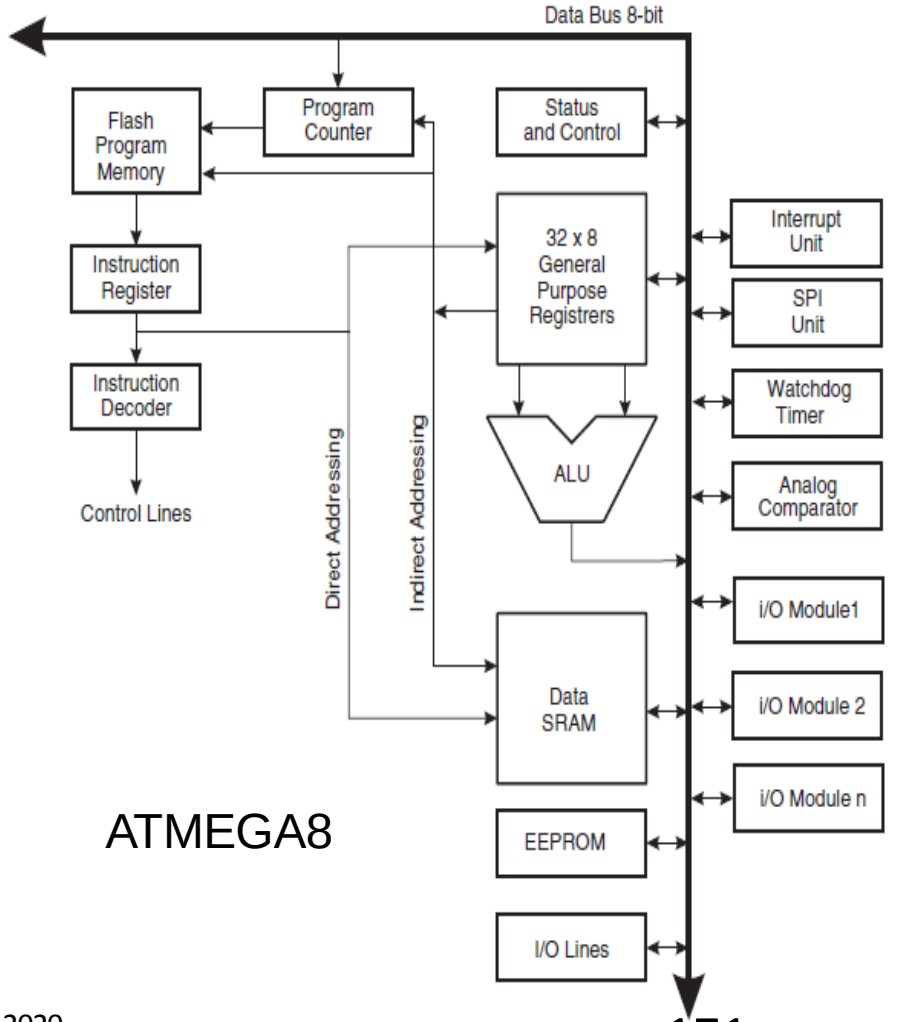




Clônes
MCU Identity

SRAM PUF (Physical Unclonable Function)

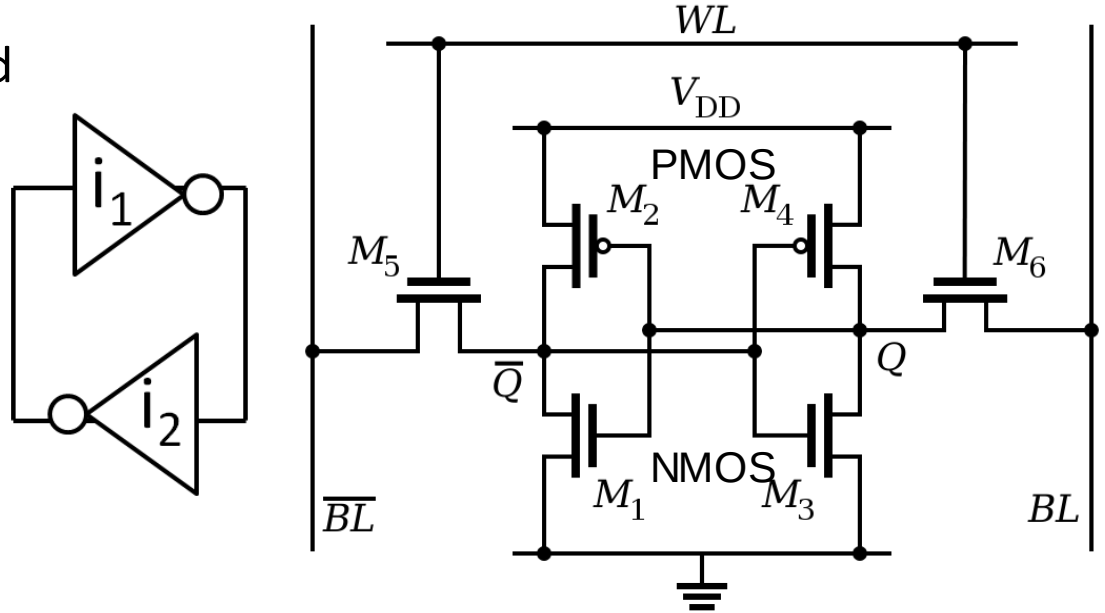

REGISTERS
SRAM
PUF
FLASH
FLASH
EEPROM
EEPROM

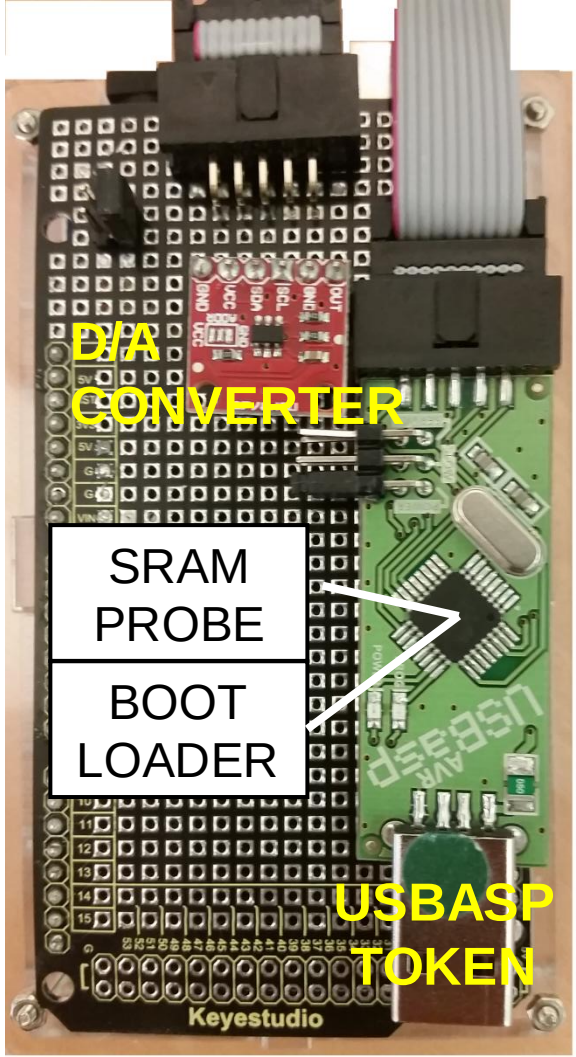

Data Bus 8-bit
Flash Program Memory
Program Counter
Status and Control
Instruction Register
Instruction Decoder
Control Lines
32 x 8 General Purpose Registrers
ALU
Direct Addressing
Indirect Addressing
Data SRAM
Interrupt Unit
SPI Unit
Watchdog Timer
Analog Comparator
i/O Module1
i/O Module 2
i/O Module n
EEPROM
I/O Lines

ATMEGA8

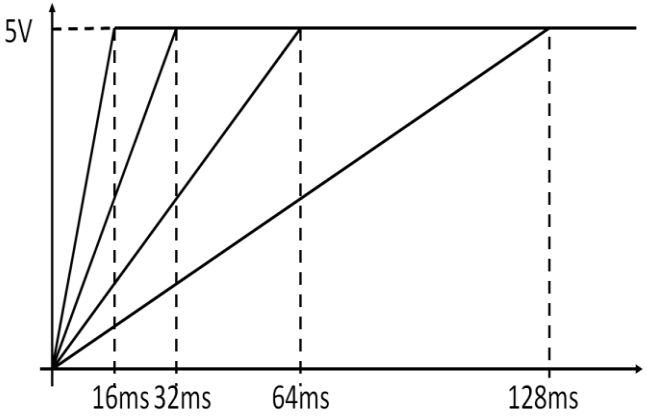
About SRAM PUF

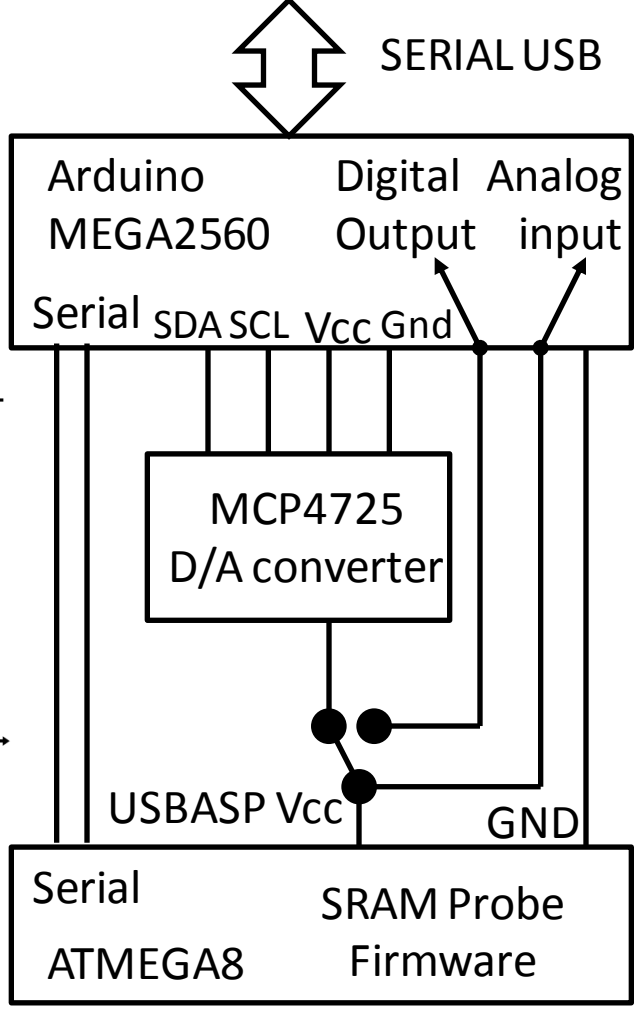
- A SRAM memory is made with 6 CMOS transistors, and includes two invertors (i1 and i2) connected in series (i.e. head to tail).
- Due to transistors physical and electrical dissymmetry, some memory cells take a fixed value (non random) after powering up.
- This effect (SRAM PUF) may be used for micro controller unit (MCU) authentication purposes.



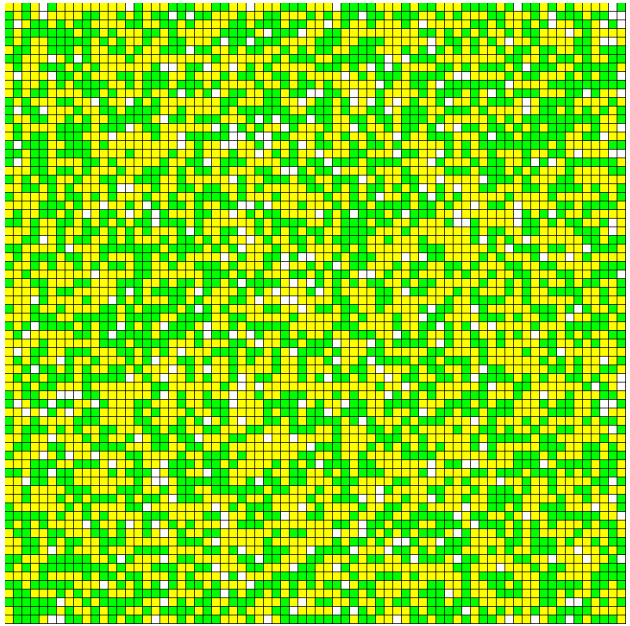


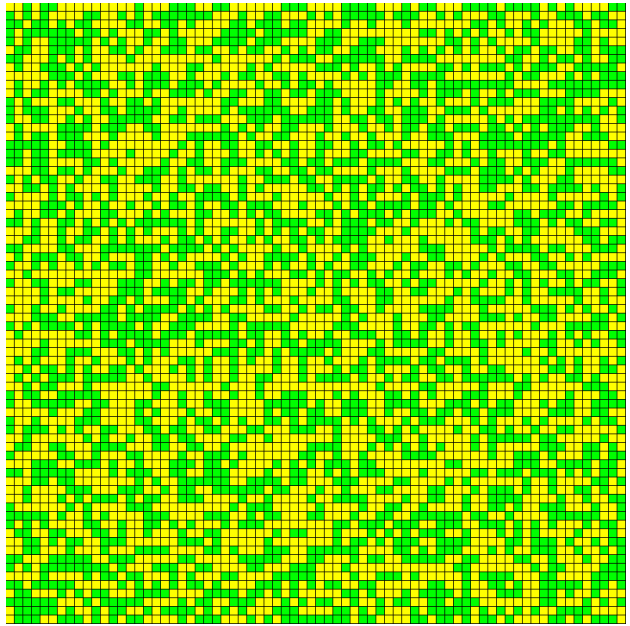
Urien, P.; "Innovative ATMEGA8 Microcontroler Static Authentication Based on SRAM PUF", IEEE CCNC 2020

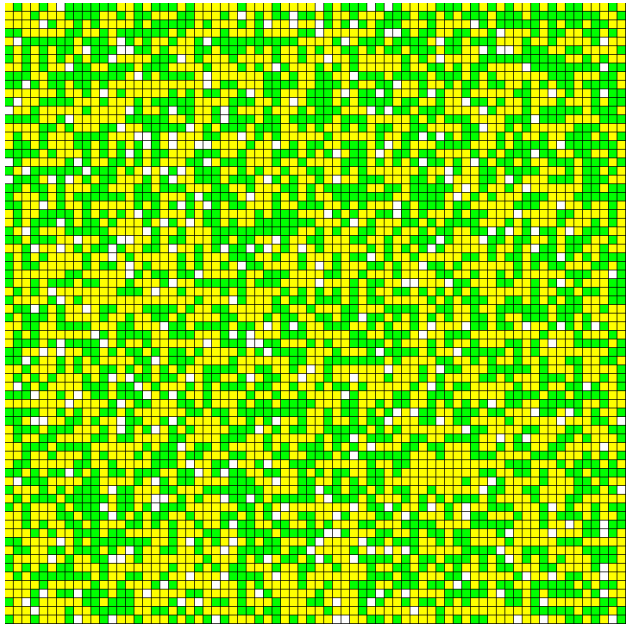




Static Authentication



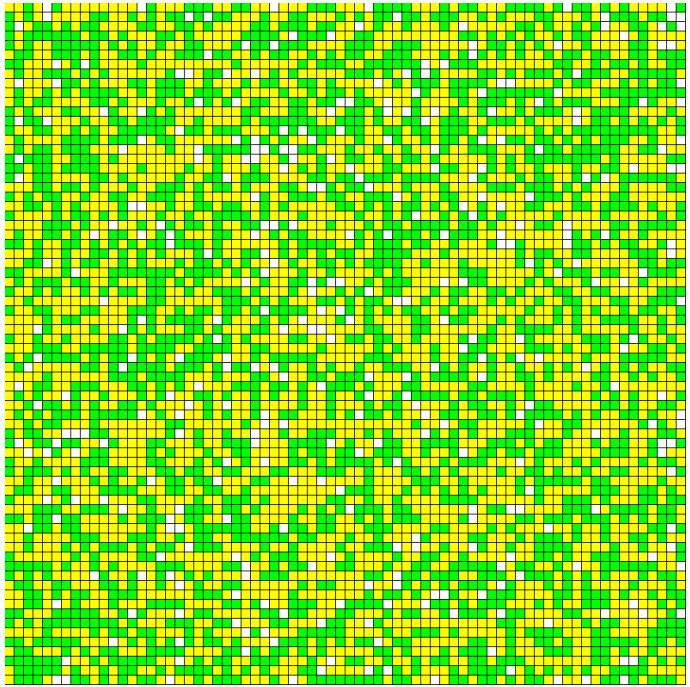




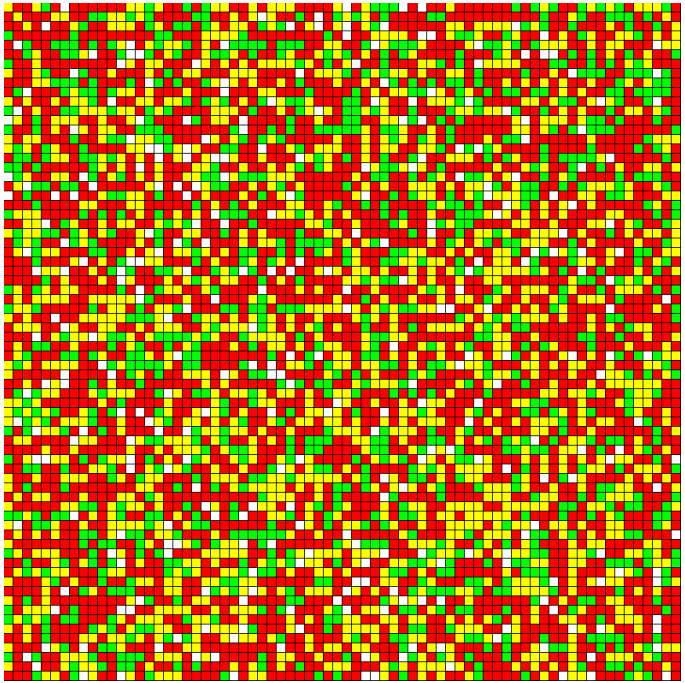
DEVICE#1, 250 MEASURES DEVICE#X 1 MEASURE DEVICE#2, 250 MEASURES

Graphical Static Authentication

Flipping bits, red
H match, green
L match, yellow
Other, white



DEVICE#1



DEVICE#2

Device Integrity
Self Attestation

Remote Attestation

- Remote attestation *is a process whereby a trusted entity (verifier)* ~~*remotely*~~ *measures internal state of a untrusted possible compromised device (prover).*
- *The ICE verification function is a self-check summing code, i.e. a sequence of instructions that compute a checksum over themselves in a way that the* checksum *would be wrong or the computation would be slower if the sequence if instruction is modified*

$$ICE = Checksum(\ A(P(0)\ ||\ A(P(1)||\ \dots\ ||\ A\big(P(i)\big)\ \dots\ ||\ A(P(m-1)\ )$$

Asokan, N. et al. "ASSURED: Architecture for Secure Software Update of Realistic Embedded Devices.". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 37.11 (2018): 2290-2300.
Seshadri, A. et al. "SCUBA: Secure Code Update By Attestation in sensor networks.", in Radha Poovendran & Ari Juels, ed., '"Workshop on Wireless Security" , ACM, , pp. 85-94 (2006).

Permutation in Z/pZ*

$$P(x) = x + x^2 \vee C \mod 2^n$$

A. SHAMIR & AL, 2002

$$P(x) = a_0 + a_1 x + \cdots + a_d x^d \mod 2^w$$

R.L. RIVEST, 2001

$$P(x) = 1 + x + x^2 + \cdots + x^d \mod p^e$$

R. MATTEWS, 1994

A. Klimov, and A. Shamir. "A New Class of Invertible Mappings.", . CHES, volume 2523 of Lecture Notes in Computer Science, page 470-483. Springer, (2002)
R. L. Rivest, "Permutation polynomials modulo 2^w". Finite Fields And Their Applications, 7, 287-292 (2001).
Matthews R., "Permutation Properties Of The Polynomials 1 + x +…+x^k Over A Finite Field";. Proceedings of the American Mathematical Society, Volume 120, Number 1, January 1994

bMAC

- bMAC computes a fingerprint of a set of memories (m)such as FLASH, SRAM, EEPROM, according to a pseudo random order, fixed by a permutation P

$$bMAC(P) = h(\, A(P(0) \,||\, A(P(1)|| \ldots || \, A\big(P(i)\big) \ldots || \, A(P(m-1)\,)$$

- bMAC works with exponential permutations based on generators in Z/pZ*, p prime (p>m)
  - x ε [1, p-1], F(x)= g^x mod p
  - y ε [0, m-1], P(y)= F(y+1)-1

$$F(x) = \; {g_2}^{s_1 {g_1}^{x} \, mod \, p} mod \; p, \;\; x, s_1 \in [1, p-1]$$