Department of Computer Science

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY™

CSE 4820: Wireless and Mobile Security

23. Final Review

Dr. Abdullah Aydeger
Location: Harris Inst #310
Email: aaydeger@fit.edu

Outline

SDR

Zigbee

Z-Wave

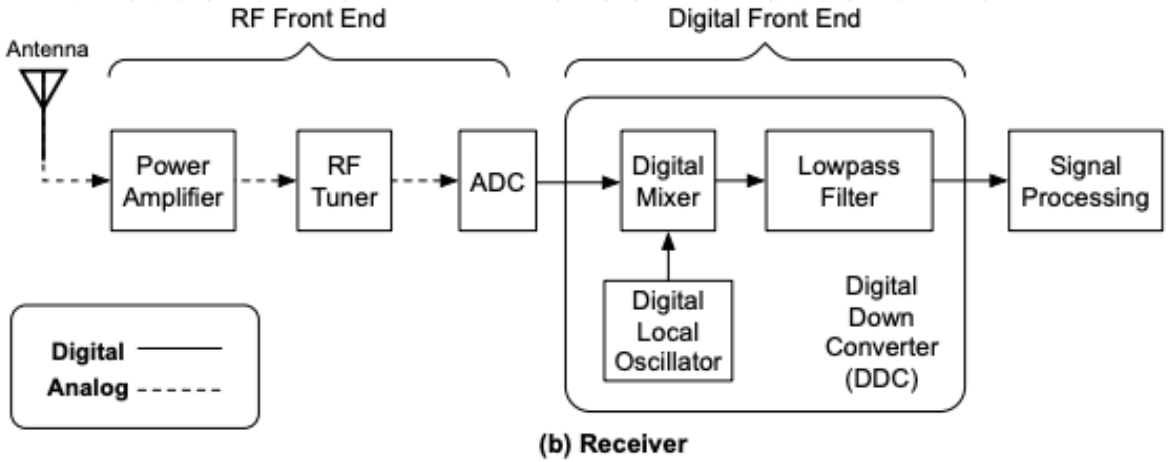
LoRaWAN

Cellular Networks (2G/3G/LTE)

Software Defined Radios

- The process of reverse engineering a radio signal to understand elements such as frequency, modulation, encoding, and protocol in order to intercept or replicate the signal
  - For ex. replace sound waves with radio waves
    - By using special 'radio soundcard', you can receive and transmit arbitrary signals
- Has redefined wireless hacking
  - Instead of being limited by black box radios, unfettered access to RF
    - Access to Radio modules/protocols that were obscured

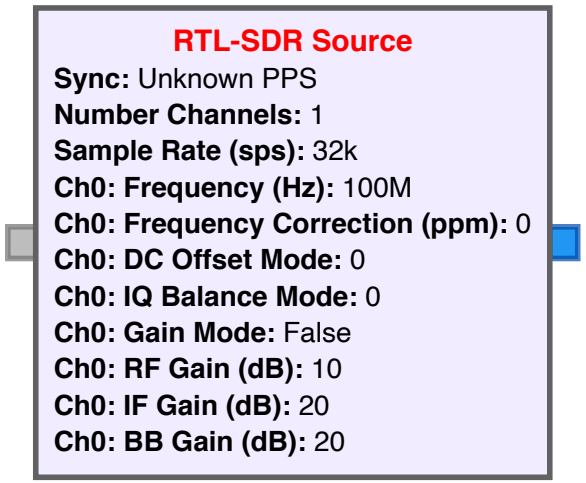
SDR Architecture



- Three main parts:
  - Radio Frequency (RF) amplifier, the tuner, and the Analog-to-Digital Converter (ADC)
- RF amplifier is responsible for boosting weak signals
- Tuner; select portion of radio spectrum to analyze
  - Like tuning on an old radio
- ADC; sampling (analog waveform to stream of digital numbers
- Antennas; isotropic (any direction) or directional

RTL-SDR

- Digital Video Tuner converted into SDR
- Receiver only
- Frequency Range: 50MHz to 1.7GHz
- ADC: 8 bits
- Popular among hobbyist due to low cost
- RTL-SDR Source block available in Gnuradio



https://amzn.to/321mYwB

HackRF

- Wide Band Software Defined Radio (SDR)
- Half-duplex transceiver
- Frequency Range: 10MHz to 6GHz, ADC: 8 bits
- Power: 30 mW - 1 mW (depending on band)
- Popular among researchers due to low cost
- Open sourced hardware

https://greatscottgadgets.com/hackrf/one/

Software: GNU Radio

- Gnuradio's modular design allows us to process and prepare signals to test methods of intercepting and replicating the signal
- Helpful blocks: hardware sinks/sources, file sink/sources, modulators, signal multipliers, signal sources, vector sources, variables, and QT Gui sinks

Software: Universal Radio Hacker

- Discussed in Pohl, Johannes, and Andreas Noack. "Universal radio hacker: a suite for analyzing and attacking stateful wireless protocols." 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18). 2018.
- Complete suite for wireless protocol investigations with native support for many common SDRs
- The URH allows easy demodulation of signals combined with an automatic detection of modulation parameters to identify the bits and bytes that fly over the air

https://github.com/jopohl/urh

How to try this in wild?

- Finding a target
  - For ex., key fob, garage opener, wireless mouse, and RC car
- Device reconnaissance;
  - Figure out what frequency it uses
- Finding and capturing signal
- Replaying/changing

Device reconnaissance

- All RF devices subject to FCC (Federal Communications Commission) Certificate require registration with the FCC
- They are given an FCC ID
  - Usually it is printed somewhere on the device
- Looking up the FCC ID yields information about the RF signal to include the frequency

2 results were found that match the search criteria:
Grantee Code: B8Q Product Code: ACSCT

Displaying records 1 through 2 of 2.

View Form	Display Exhibits	Display Grant	Display Correspondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
	Detail Summary			The Genie Company a Division of Overhead Door Corporation	1 Door Drive	Mt. Hope	TX	United States	44660	B8QACSCT	Original Equipment	01/08/1997	390.0	390.0

https://www.fcc.gov/oet/ea/fccid#:~:text=FCC%20ID%20numbers%20consists%20of,string%20representing%20the%20Grantee%2FApplicant.

Zigbee Outline

Zigbee

- History
- Layers
- Security

Zigbee

- Set of standards for low-power wireless networking
  - Devices with up to 5 years battery life
- Low data-rate transfers, short-range, persistent-powered network coordinators/routers, and simple protocol stack
- Found in industrial and home applications
  - For ex., home theater remote controls to hospital patient monitoring systems

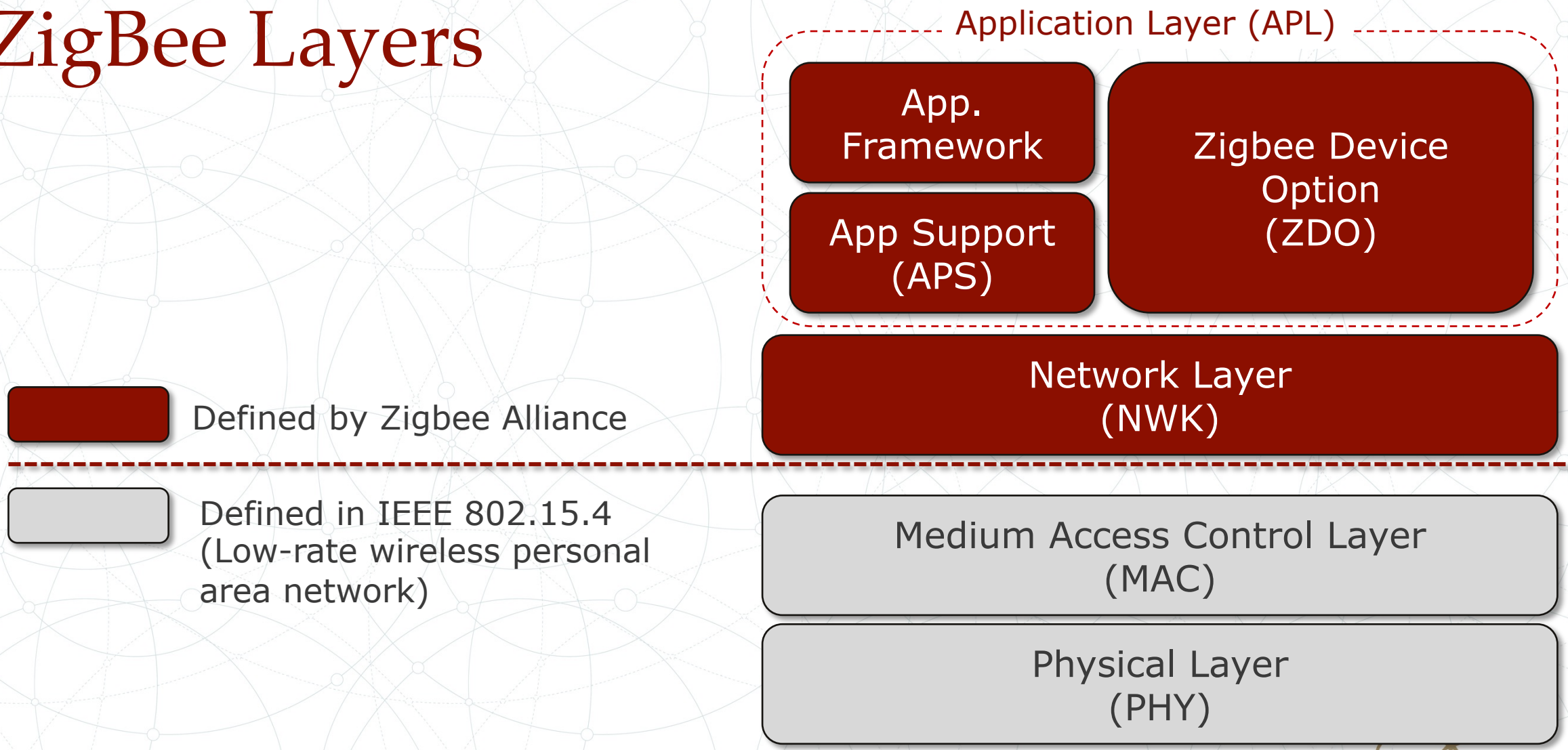
Zigbee as Wireless Standard

- Strong position to be the wireless tech for IoT
  - Already used in Google Nest Smart Thermostat, Philips Hue led light bulbs, Comcast Xfinity home security router to connect home light switches and other peripherals to public internet
- Why do we need Zigbee?
  - Compared to Wifi and BLE, Zigbee is much simpler protocol with a fully functional stack implemented in 120kb of NVRAM where some vendors claim to make reduced-functionality stacks as small as 40kb

Zigbee as Wireless Standard

- Wireless networks transmit at least 54mbps, BLE 1-3mbps, and Zigbee 20-250kbps
  - Not the right protocol for high-speed data transfers
- Wifi devices; relatively short battery life
  - Bluetooth relatively comparable to Zigbee
- Deployed mostly for the home automation;
  - Connectivity among home control systems such as electrical appliances, lighting controls, home security, etc.

ZigBee Layers



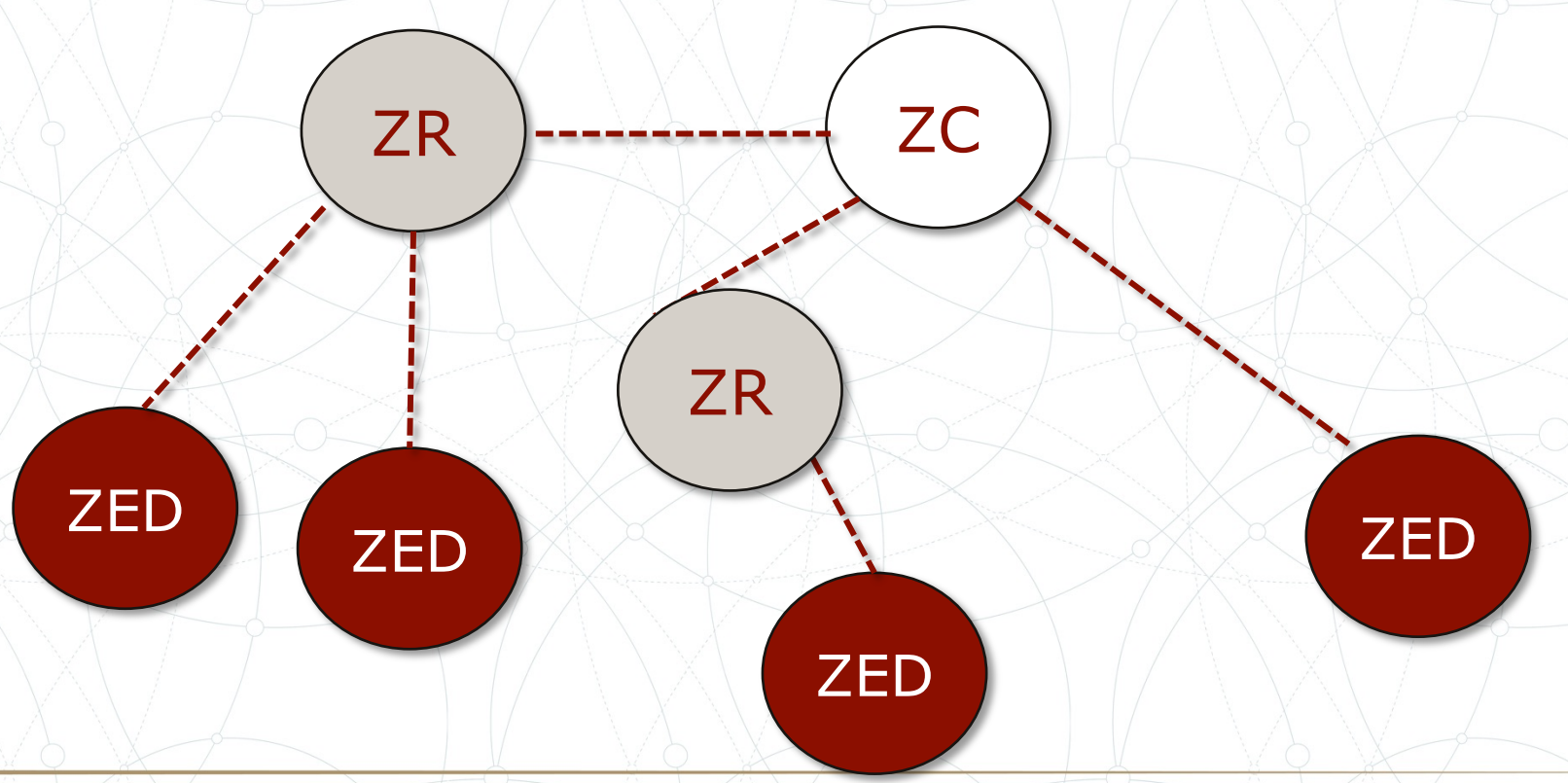
https://csa-iot.org/all-solutions/zigbee/

Zigbee: MAC Layer

- Includes functionality needed to build extensive Zigbee networks
  - Including the design of device interconnect topologies, device roles, packet framing, and network association/disassociation
- Each device has set of capabilities defined by operational roles:
  - Trust Center, Coordinator, Router, and End Device

Example Zigbee Network

- One ZC for the network, additional ZRs



IEEE 802.15.4 MAC Frame Format

Frame Control	Seq. No.	Dest. Pan ID	Dest. Addr	Source Pan ID	Source Addr	Aux Sec Hdr	Payload	FCS

- Specified in IEEE 802.15.4 Spec
- Frame Control tells what type of frame (beacon, command, data, ack…)
- Sequence Numbers enables in-order delivery
- Dest/Src Pan ID/Addr handles delivery of frame
- Auxiliary Security Header optionally implements security
- FCS is CRC 16-bit checksum of the mac layer frame

Zigbee MAC Frame Types

- Beacon Frames – used for network discovery; scan the network for ZC or ZR
- Command Frames – same as 802.11 management (association/disassociation)
- Data Frames – same as 802.11 data; exchange data b/w devices
- Acknowledgement Frames – acknowledge frames that were received

ZigBee Network Layer (NWK)

- ZigBee Network Layer (NWK) defined by ZigBee Alliance
  - Responsible for network formation, address allocation, and routing
- Network formation; FFD establishes itself as network coordinator
- Through device discovery, the coordinator must;
  - Select suitable channel to avoid interference
  - Choose random Pan ID that doesn't conflict with nearby Pans
  - Select and issue 16-bit device addresses for devices

ZigBee Application Layer (APL)

- Specifying operation and interface for application objects that define Zigbee device's functionality
- Application objects are developed by Zigbee Alliance as standard functionality profiles, or
  - By manufacturers for proprietary device functionality using the APL as mechanism to communicate with lower layers of Zigbee stack

Zigbee Security

- AES encryption, device and data authentication using a network key
- Two operational modes:
  - Standard; TC authorizes deices through the use of ACL (Access control list), each devices uses a single shared key
  - High Security; TC keeps track of all encryption and authentication keys used on the network, enforcing policies for network authentication and key updates
    - If TC fails, no device will be permitted to join the network

Zigbee Encryption

- Uses 128-bit AES
  - Assumed to be strong security
  - Could be leveraging AES in an insecure manner
    - How?
- Three types of keys to manage security;
  - Master key, network key, and link key

Zigbee Keys

- Master key; optional except the Zigbee Pro stack
  - Used in conjunction with Zigbee symmetric key-key establishment (SKKE) process to derive other keys
- Network key; protect broadcast and group traffic, as well as authenticating to the network
  - Common key among all nodes
  - Can be distributed to a device in plaintext when it joins the network
- Link key: protect unicast traffic between two devices

Zigbee Encryption Keys

- Global Link Key: used by all nodes on the network
- Unique Link Key: used to encrypt communication between a pair of nodes
  - Preconfigured Link Key: used between trust center and a node;
    - Derived prior to joining
  - Trust Center Link Key: used between trust center and a node; distributed to node
  - Application Link Key: used between pair of nodes; distributed to node

Zigbee: Key Provisioning

- Significant challenge; process of provisioning, rotating, and revoking keys on devices
- Zigbee Pro; Administrator can use the SKKE method to derive the network and link keys on devices
  - Requires devices to have master key provisioned on the TC and device joining the network

Zigbee: Key Provisioning

- Key transport; network and link keys are sent in plaintext over the wireless network to the device when it joins
  - Can Easily be intercepted
- Pre-installation; administrator preconfigures all devices with the desired encryption keys at the manufacturing process
  - How to accommodate key revocation and rotation methods?
  - Manual changes to each device to change keys

Zigbee Authentication

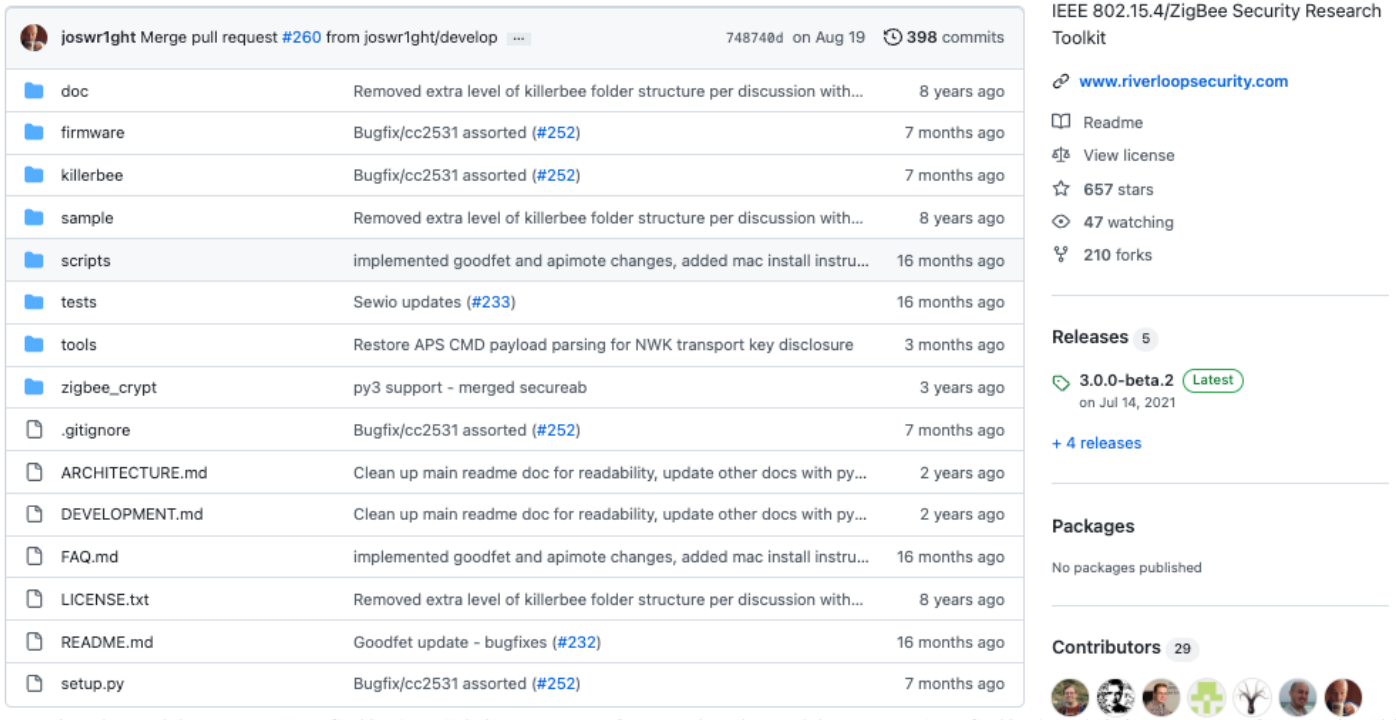
- MAC address validation through ACL;
  - A list of authorized devices is maintained on each node
  - Challenging to keep the list up-to-date (memory req.)
- Standard mode; TC grants access by issuing a network key
  - Sent in plaintext
- High security mode; SKKE method to derive the network key
  - 4-way handshake

Zigbee Authentication's Vulnerability

- No mutual authentication is used in standard Zigbee (except high security mode with SKKE)
  - Authenticating node accepts the identity of TC for the delivery of network key without any validity check to verify the identity of the network
  - Easy to impersonate a legitimate network by using the same PAN ID as target, potentially on a different channel

Zigbee Attacks

- KillerBee:
  - Python-based framework for manipulating and penetration testing Zigbee and IEEE 802.15.4 networks
  - Written and tested on Linux, free and open-source
  - Includes support for Scapy
  - Includes a variety of tools including zbwireshark, zbdump, and zbreplay



https://github.com/riverloopsec/killerbee

Z-Wave Outline

Z-Wave

- Overview
- Protocol Stack
- Security

Z-Wave

- Low-energy, mesh-networking protocol
- Predominately used in home automation (locks, garage door openers, thermostats)
  - Over 100 million products in use in homes
- Proprietary design by Sigma Systems and governed by standards established by Z-Wave Alliance
  - Does not share details of protocol outside of NDA (nondisclosure agreement)
  - Controls all fabrication and delivery of Z-Wave chips to product manufacturers

Z-Wave

- Aggressive power conservation for long battery life
  - Like Zigbee
- Using a mesh networking model to accommodate greater device range
- Relatively simple protocol
- Support for positive acknowledgement and frame retransmission, self-forming and dynamic routing topology updates, and application-specific profiles

Z-Wave Overview

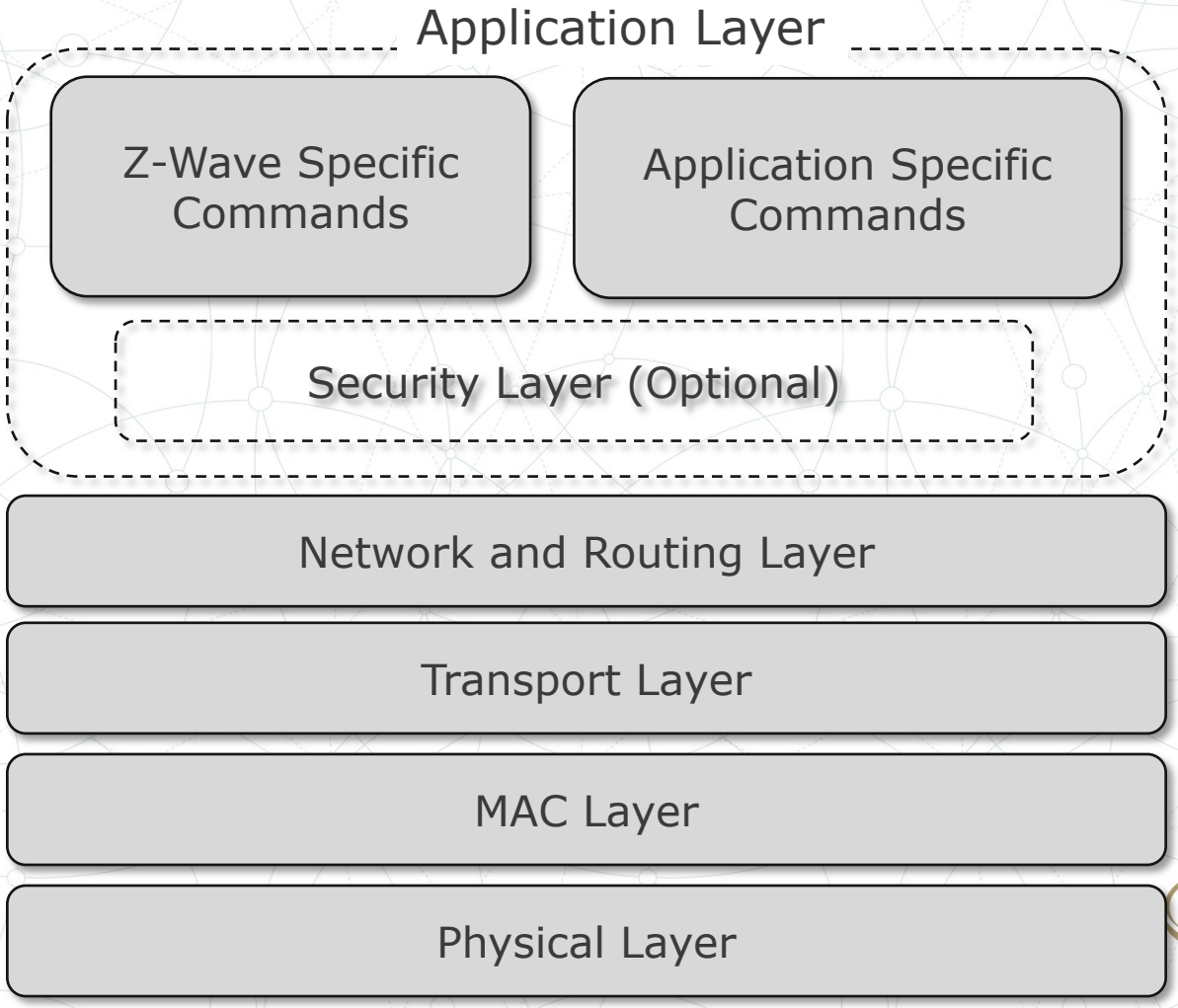
- Z-Wave is based on a mesh network topology
  - Each (non-battery) device installed in the network becomes a signal repeater
  - As a result, the more devices you have in your home, the stronger the network becomes
- While Z-Wave signals easily travel through most walls, floors and ceilings, the devices can also intelligently route themselves around obstacles to attain seamless, robust, whole-home coverage

Z-Wave Overview

- While Z-Wave has a range of 100 meters (or 328 feet) in open air, building materials reduce that range (roughly every 30 feet)
  - Operates at 908.42 MHz in the United States and Canada
- The Z-Wave networks can be linked together for even larger deployments
- Each Z-Wave network can support up to 232 Z-Wave devices

Z-Wave Protocol Stack

- Uses structured protocol stack



Z-Wave PHY Layer

- Uses sub-1-GHz band to accommodate frequency variations in different countries

Profile	Data Rate	Encoding	Modulation	Packet Size
R1	9.6 Kb/s	Manchester	FSK	64 bytes
R2	40 Kb/s	NRZ	FSK	64 bytes
R3	100 Kb/s	NRZ	FSK	170 bytes

- Offers 3 different RF Profiles (R1, R2, and R3) with unique data rates, encodings, modulation, and packet frame sizes
  - R1 is deprecated by Z-Wave alliance but might be in use for some
- Range from 3 – 75 feet; depending on transmit power

Z-Wave MAC Layer

- Responsible for several attributes;
  - Packet framing and formatting
  - Positive ACK
  - Error detection
  - Retransmission of packets
  - Unicast, broadcast, and multicast processing
  - Address selection and allocation functions

Z-Wave MAC Layer

- Basic architecture of Z-Wave network: Controller device and slave devices
- Single primary controller device is responsible for establishing the network and selecting unique network identifier (i.e., HomeID)
- Controller devices are able to initiate a transmission on the network (polling or updating target devices) and responsible for maintaining network routing information
- Slave devices follow the instructions of controllers without dealing with how

Z-Wave MAC Layer Frame Format

- HomeID: Randomly selected 4-byte value chosen by controller
- NodeID: 1-byte value assigned to the node by the controller
  - Max 232 nodes (22 left for reserved, 1 for broadcast, 1 uninitialized)
- Frame Control: 16-bit field with several subfields

Z-Wave R1/R2 Frame Format

HomeID	Source Node ID	Frame Control	Length	Destination Node ID	Payload	FCS

Z-Wave R3 Frame Format

HomeID	Source Node ID	Frame Control	Length	Sequence number	Destination Node ID	Payload	FCS

Z-Wave Network Layer

- Defines device responsibilities and responsible for other network components such as HomeID selection and NodeID allocation process as well as network route establishment
- Z-Wave inclusion and exclusion for network connections

Z-Wave Network Layer: Inclusion

- Involves configuring the controller in inclusion mode (allowing it to accept new nodes) by pressing a physical button or choosing a menu item, and pressing a button on the new node to initiate an inclusion exchange
- When the new node initiates the inclusion process, it sends a Z-Wave node information frame using homeID of 0x00000000 and nodeID of 0x00 and a brodcast dest NodeID
  - Discloses the capabilities of the new device to the controller, which, in turn, allocates a NodeID to the new device for subsequent use on the network and updates routing tables to accommodate packet delivery to the new node

Z-Wave Network Layer: Exclusion

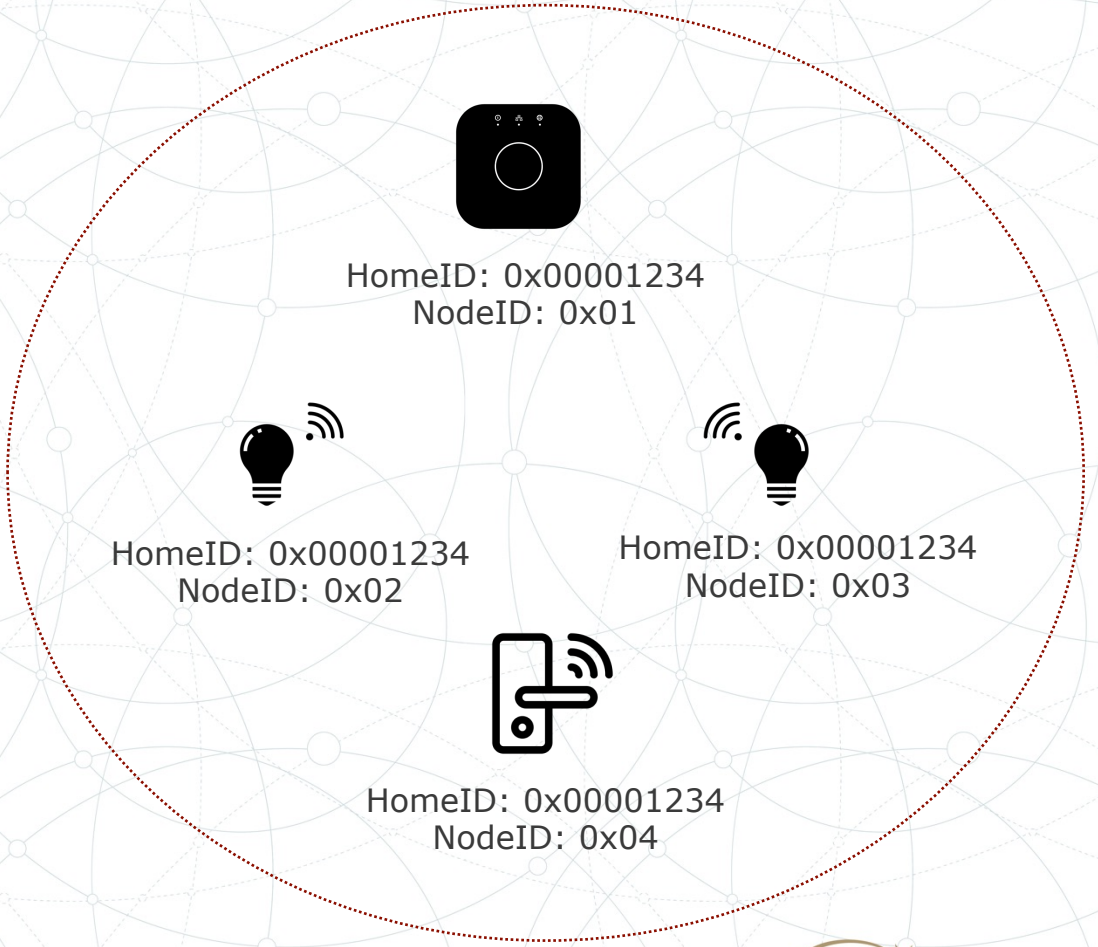
- Similar to inclusion but functionally opposite
- A node joined the Z-Wave network via inclusion cannot leave the network to join a different Z-Wave controller without completing the exclusion process
- Involves pressing a physical button on the controller and the device node, causing device to return to unallocated nodeID 0x00

Z-Wave Inclusion & Exclusion

- Strong component of the overall Z-Wave security
  - Requires physical access to the controller
- Is it secure enough?
  - What about spoofing?

Z-Wave Network Topology

- Nodes in a Z-Wave have a 1-byte NodeID, which must be different than every other node in the network
- Nodes in a Z-Wave network have a 4-byte HomeID, assigned by the controller at the time of inclusion
- Nearby networks must have different HomeIDs



Z-Wave Application Layer

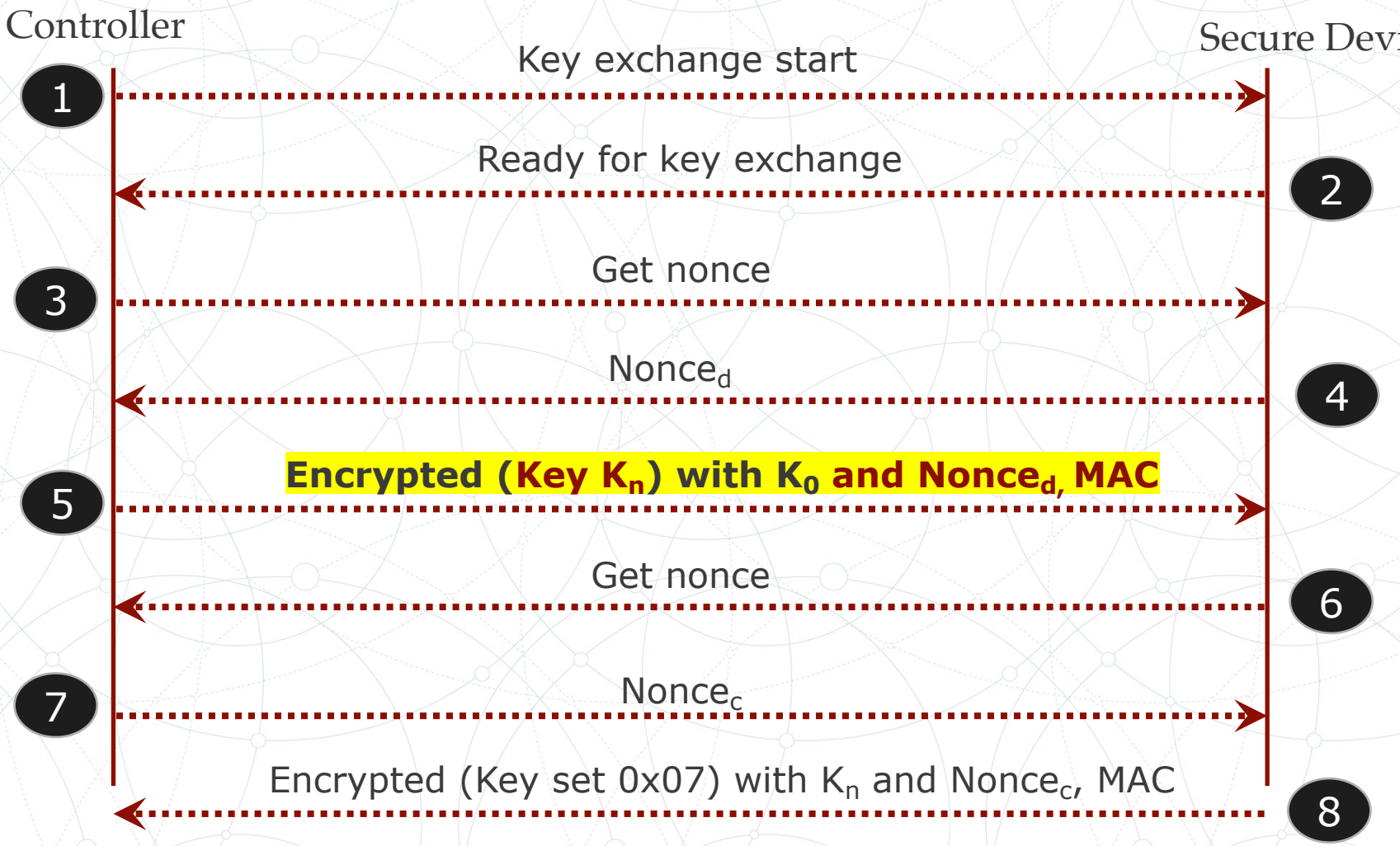
- Responsible for parsing and processing the data requests and responses in the packet payload
- Handles both application-specific and Z-Wave application control data
- Basic application payload format:

Header	Command Class	Command	Command Class Data

Z-Wave Security

- Uses AES-OFB (Output Feedback Mode) to provide data confidentiality on the network
  - Conserve the amount of payload content transmitted in Z-Wave frames while being NIST (National Institute of Standards and Technology) approved
- AES CBC-MAC (cipher block chaining message authentication code) for data integrity protection
- CLASS_SECURITY command; key exchange process to derive keys

Z-Wave Key Exchange Process



Temporal Key
$K_{0=}$ 16 bytes of 0x00

Network Key
$K_{n\ =}$ Chosen by controller

Z-Wave Key Exchange Vulnerabilities

- MitM:
  - The secure device does not validate the identity of controller other than validating the MAC of encrypted Kn message using the temporary key K_0
  - Attacker can use any Z-Wave controller that support CLASS_SECURITY command class to intercept the inclusion process with a target device
    - Causing victim to associate to a malicious network

Z-Wave Key Exchange Vulnerabilities

- Key recovery attack:
  - There is no confidentiality protection in the delivery of the Kn key over the network since the K_0 is well known
  - Attacker passively observing the inclusion process can recover the network key K_n
    - Use it later to decrypt or forge arbitrary packets on the network

Z-Wave Key Exchange's Solution

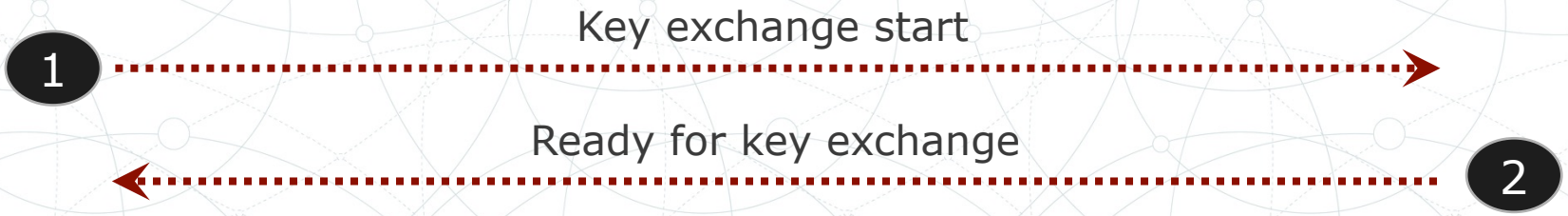
- Low power inclusion mode
  - Controller and secure device transmit using minimal power capabilities
  - Require no more than 3 feet apart to complete the process
  - Also infrequent practice of adding new devices
  - Results in less opportunity for the attacker

Z-Wave Key Derivation

- After the K_n (network key) is established, the device generates two additional keys K_c (packet encryption) and K_m (message auth key)
  - Kc (packet encryption) = AES-ECB$_{kn}$(Password$_c$)
  - K_m (message auth key) = AES-ECB$_{kn}$(Password$_m$)
- Where Password$_c$ and Password$_m$ are static values across all Z-Wave devices
- However, if we observe Kn and we know Password$_c$ and Password$_m$, we can compute the packet encryption and message auth keys

Z-Wave Key Establishment Attack

- We can force the key establishment process to establish a new K_n if we missed the original key establishment
  - Similar to an IEEE 802.11 Deauth

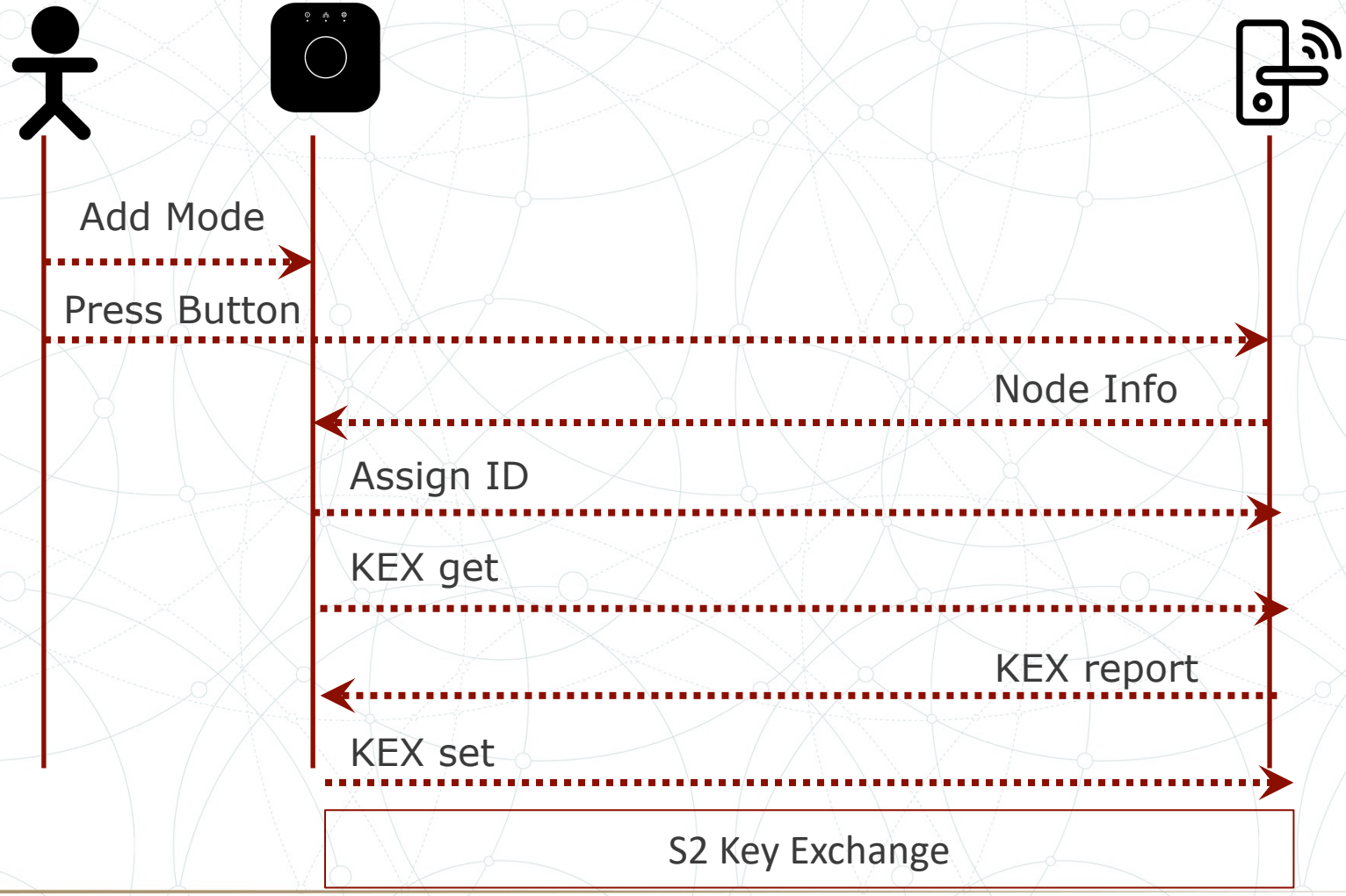


Z-Wave Key Establishment: Solution

- In response to the flawed S_0 Key Establishment and Pairing Process, SI Labs (i.e., Z-Wave Alliance) responded with a S_2 Key Establishment
- Removed the null temporal key
- Each devices has a DSK [device specific key] – 16 byte key
- Key exchanged using Diffie Hellman key exchange protocol
- Removed issues with man-in-the-middle
  - Are we safe now?

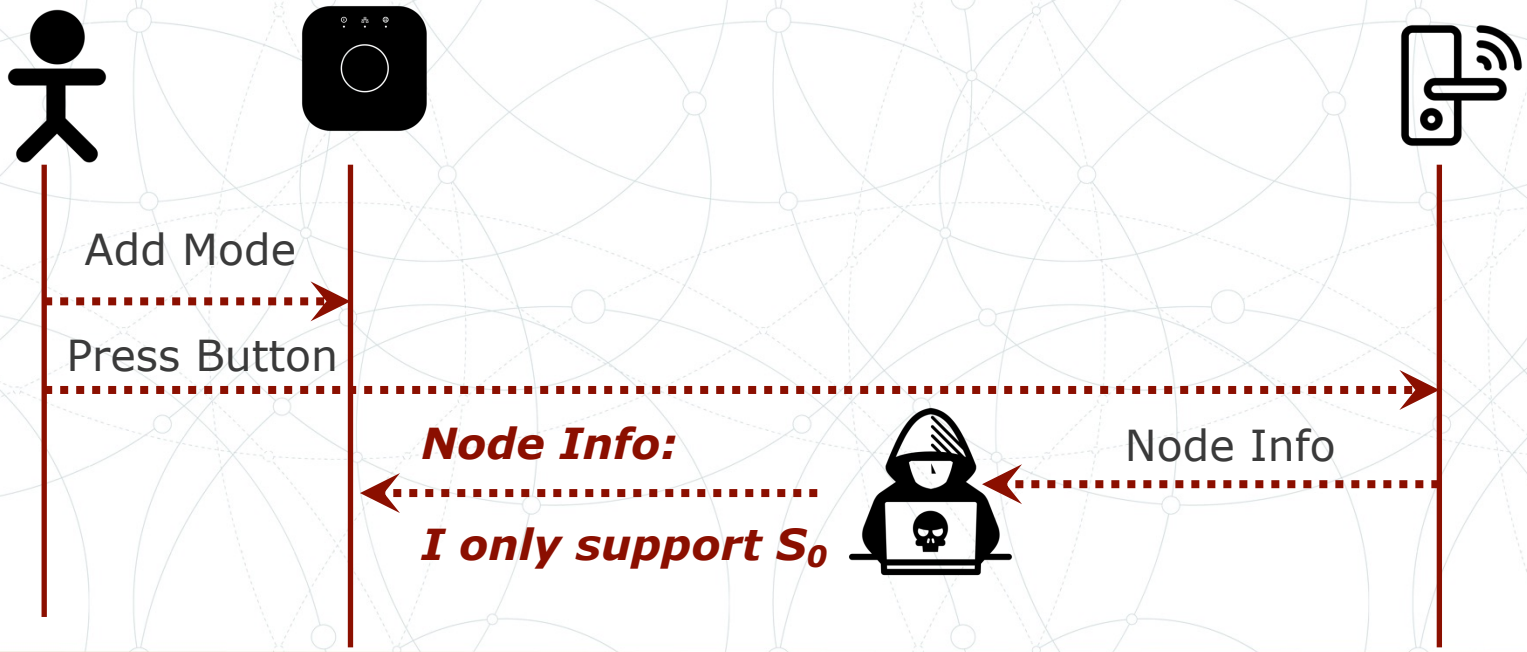
https://community.silabs.com/s/topic/0TO1M000000qHcQWAU/zwave?language=en_US&tabset-178da=2

Z-Wave Key Exchange



S2 Z-Wave Rollback Attack

- An attacker jams the node info or responds prior to the node
- Provides node info that device only supports flawed S_0 pairing mode



Mitigating Eavesdropping in Z-Wave

- As in any wireless technology, attacker can always capture the traffic (aka eavesdropping)
  - Useless if confidentiality and integrity of the data is ensured
- Unfortunately, the use of encryption in Z-Wave is optional
  - Switch to a vendor that offers network confidentiality and integrity control

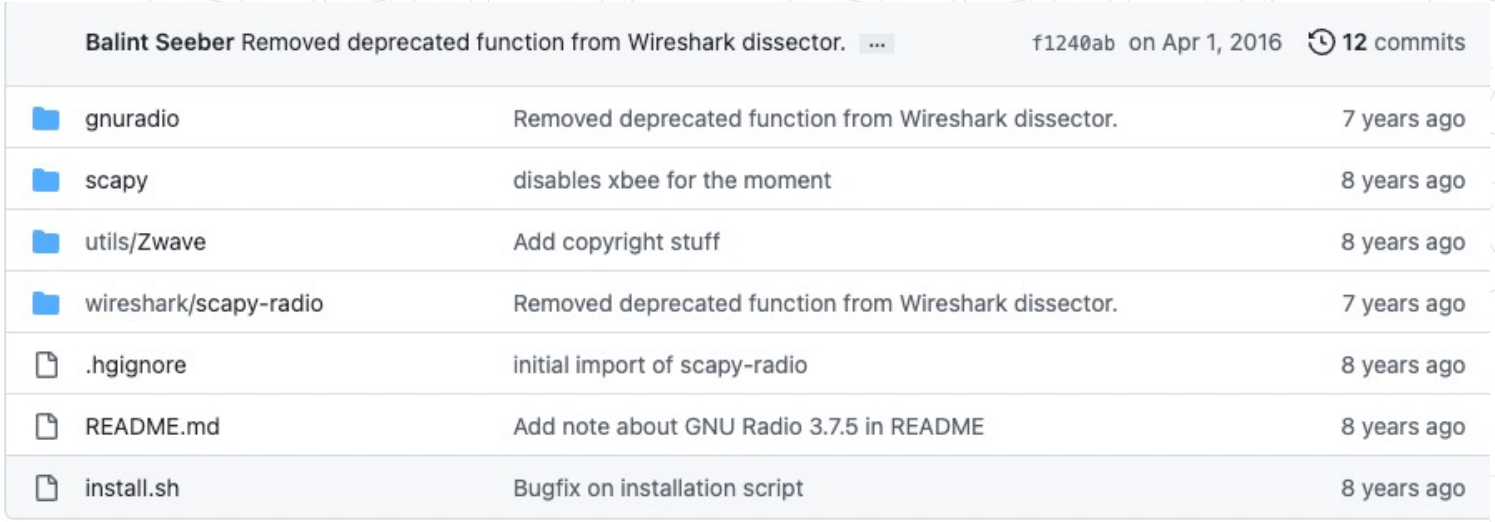
Injection Attacks in Z-Wave

- If no encryption, attacker can capture and replay the packets
- Injection is also done if the device is in the Z-Wave network
  - To be in the network, Z-Wave inclusion needed
    - Does it solve this problem?
  - Attacker can spoof the address and inject any packet

https://github.com/joswr1ght/killerzee

Scapy-Radio Framework

- Modified version of Scapy to support
  - Zwave
  - Zigbee
  - 802.15.4
- Works with software defined radios (SDR)



Balint Seeber Removed deprecated function from Wireshark dissector. ··· `f1240ab` on Apr 1, 2016 **12** commits

gnuradio	Removed deprecated function from Wireshark dissector.	7 years ago
scapy	disables xbee for the moment	8 years ago
utils/Zwave	Add copyright stuff	8 years ago
wireshark/scapy-radio	Removed deprecated function from Wireshark dissector.	7 years ago
.hgignore	initial import of scapy-radio	8 years ago
README.md	Add note about GNU Radio 3.7.5 in README	8 years ago
install.sh	Bugfix on installation script	8 years ago

README.md

Introduction

This tool is a modified version of scapy that aims at providing an quick and efficient pentest tool with RF capabilities.

It includes:

- A modified version of scapy that can leverage GNU Radio to handle a SDR card
- GNU Radio flow graphs (GRC files) we have build that allows full duplex communication
- GNU Radio blocks we have written to handle several protocols

LoRaWAN Outline

LoRaWAN

- As a Wireless Standard
- History
- Characteristics
- Network Architecture
- Message Types
- Adaptive Data Rates
- Security

LoRaWAN

- Long Range (LoRa) Wide Area Network (WAN)
  - "A Low Power, Wide Area (LPWA) networking protocol designed to wirelessly connect battery operated 'things' to the internet in regional, national or global networks,
  - And support targets key Internet of Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services"

LoRaWAN Protocol

- The LoRaWAN specification is open so anyone can set up and operate a LoRa network
  - Wireless audio frequency technology that operates in a license-free radio frequency spectrum
- It uses a narrow band waveform with a central frequency to send data
  - Makes it robust to interference

LoRaWAN Protocol

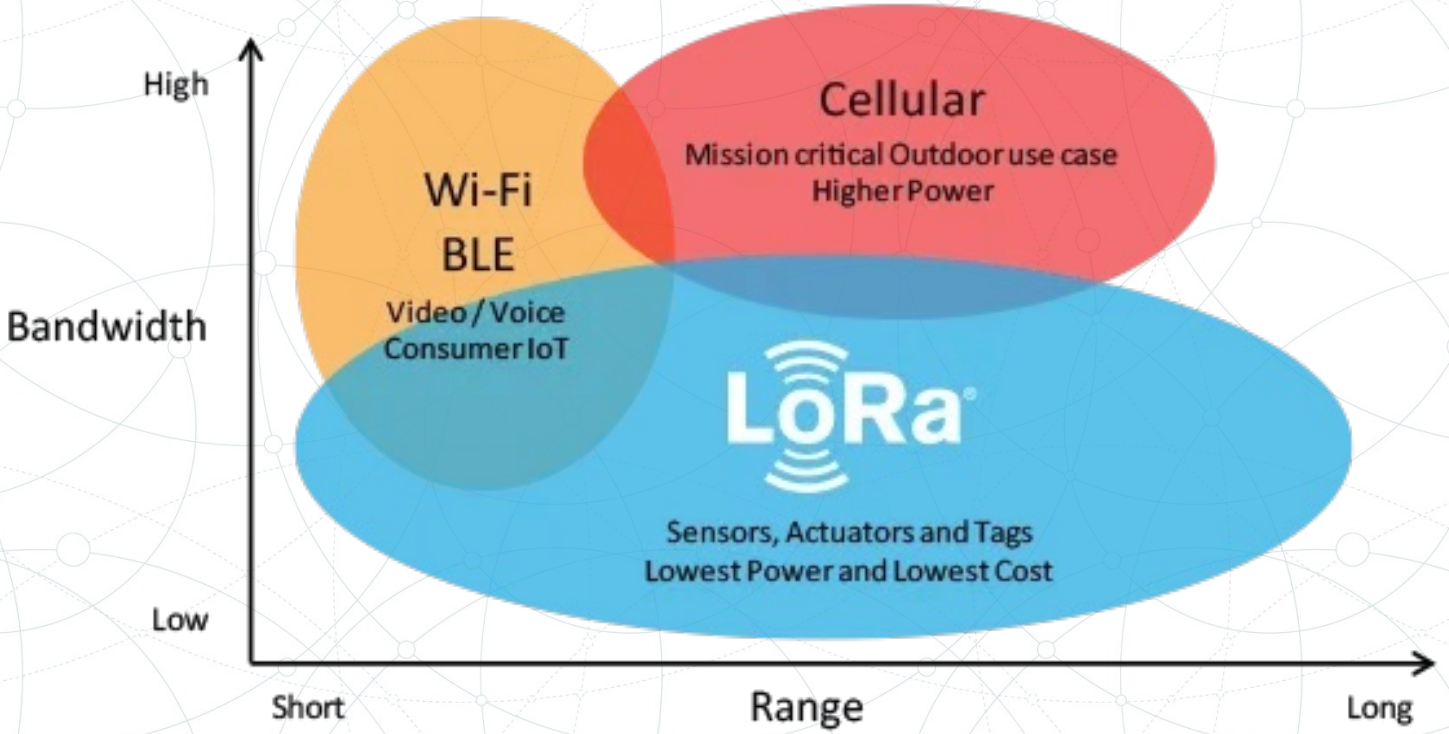
- LoRa is a physical layer protocol that uses spread spectrum modulation and supports long-range communication at the cost of a narrow bandwidth
  - Modulation technique derived from Chirp Spread Spectrum (CSS) technology
  - It encodes information on radio waves using chirp pulses;
    - Similar to the way dolphins and bats communicate!

LoRaWAN as Wireless Standard

- LoRa is ideal for applications that transmit small chunks of data with low bit rates
- Data can be transmitted at a longer range compared to technologies like WiFi, Bluetooth or ZigBee
- These features make LoRa well suited for sensors and actuators that operate in low power mode

LoRaWAN as Wireless Standard

- Suitable for transmitting small size payloads (like sensor data) over long distances
  - Greater communication range with low bandwidths than other competing wireless data transmission technologies



LoRaWAN Use Cases

- Vaccine cold chain monitoring;
  - LoRaWAN sensors are used to ensure vaccines are kept at appropriate temperatures in transit
- Animal conservation;
  - Tracking sensors manage endangered species such as Black Rhinos and Amur Leopards
- Dementia patients;
  - Wristband sensors provide fall detection and medication tracking

Characteristics of LoRaWAN

- LoRaWAN is a Media Access Control (MAC) layer protocol built on top of LoRa modulation
  - A software layer which defines how devices use the LoRa hardware, for example when they transmit, and the format of messages
- Long range communication up to 10 miles in line of sight
- Long battery duration of up to 10 years
  - For enhanced battery life, you can operate your devices in class A or class B mode, which requires increased downlink latency

Characteristics of LoRaWAN

- Low cost for devices and maintenance
- License-free radio spectrum but region-specific regulations apply
- Has a limited payload size of 51 bytes to 241 bytes depending on the data rate
  - The data rate can be 0,3 Kbit/s – 27 Kbit/s

LoRaWAN Network Overview

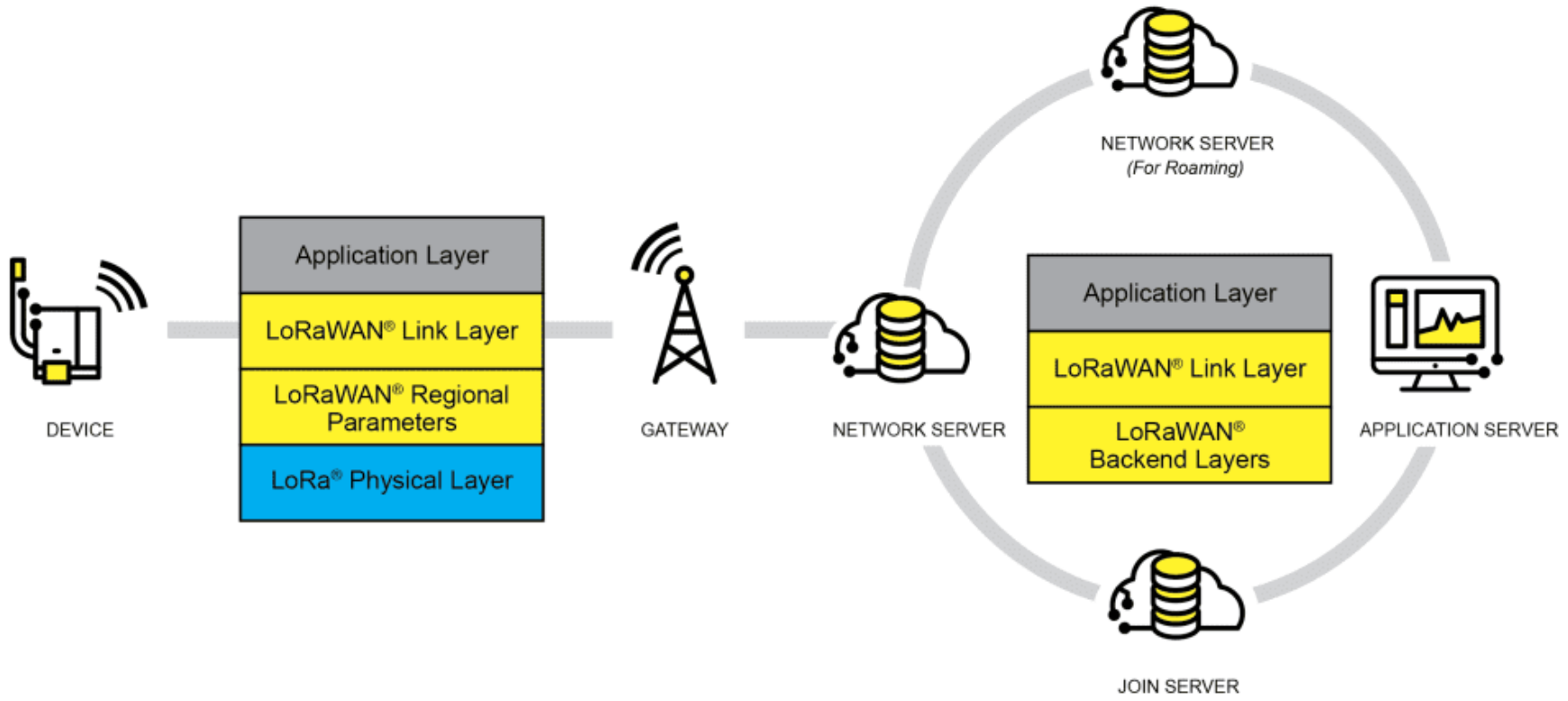
- LoRaWAN® network architecture is deployed in a star-of-stars topology in which gateways relay messages between end-devices and a central network server
- The gateways are connected to the network server via standard IP connections and act as a transparent bridge, simply converting RF packets to IP packets and vice versa

LoRaWAN Network Overview

- The wireless communication takes advantage of the Long Range characteristics of the LoRa physical layer, allowing a single-hop link between the end-device and one or many gateways
- All modes are capable of bi-directional communication,
  - And there is support for multicast addressing groups to make efficient use of spectrum during tasks such as Firmware Over-The-Air (FOTA) upgrades or other mass distribution messages

LoRaWAN Network Overview

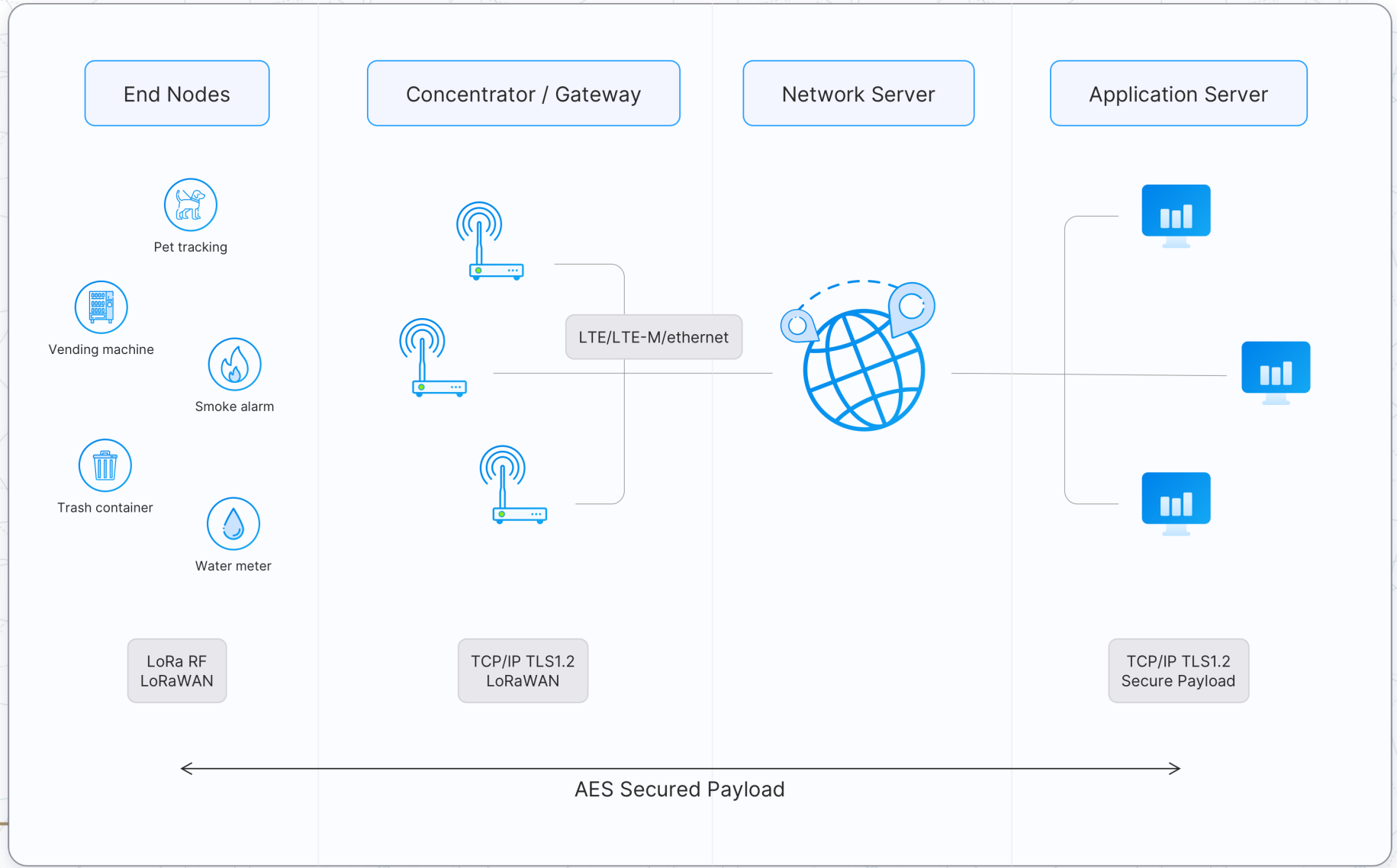
- The specification defines the device-to-infrastructure (LoRa®) physical layer parameters & (LoRaWAN®) protocol and so provides seamless interoperability between manufacturers



LoRaWAN Network Devices

- LoRaWAN Device Types:
  - End Devices
  - Gateways
  - Network Server
  - Application servers
  - Join Server

LoRaWAN Architecture



LoRaWAN: End Device

- A LoRaWAN end device can be a sensor, an actuator, or both
- They are often battery operated
- These end devices are wirelessly connected to the LoRaWAN network through gateways using LoRa RF modulation
- *LoRaWAN end device - The Things Industries Generic Node Sensor Edition*:
  - An end device that consists of sensors like temperature, humidity, and fall detection

LoRaWAN Device Classes

- LoRaWAN has three different classes of end-point devices to address the different needs reflected in the wide range of applications:
  - Class A – Lowest power, bi-directional end-devices
  - Class B – Bi-directional end-devices with deterministic downlink latency
  - Class C – Lowest latency, bi-directional end-devices

LoRaWAN: Gateways

- Each gateway is registered (using configuration settings) to a LoRaWAN network server
- A gateway receives LoRa messages from end devices and simply forwards them to the LoRaWAN network server
- Gateways are connected to the Network Server using a backhaul like Cellular (3G/4G/5G), WiFi, Ethernet, fiber-optic or 2.4 GHz radio links

LoRaWAN: Network Server

- The Network Server manages gateways, end-devices, applications, and users in the entire LoRaWAN network
- A typical LoRaWAN Network Server has the following features:
  - Establishing secure 128-bit AES connections for the transport of messages between end-devices and the Application Server (end-to-end security)
  - Validating the authenticity of end devices and integrity of messages
  - Deduplicating uplink messages
  - Selecting the best gateway for routing downlink messages

LoRaWAN: Network Server

- Sending ADR commands to optimize the data rate of devices
- Device address checking
- Providing acknowledgements of confirmed uplink data messages
- Forwarding uplink application payloads to the appropriate application servers
- Routing uplink application payloads to the appropriate Application Server
- Forwarding Join-request and Join-accept messages between the devices and the join server
- Responding to all MAC layer commands

LoRaWAN: Application Server

- Processes application-specific data messages received from end devices
- It also generates all the application-layer downlink payloads and sends them to the connected end devices through the Network Server
- A LoRaWAN network can have more than one Application Server
- The collected data can be interpreted by applying techniques like machine learning and artificial intelligence to solve business problems

LoRaWAN: Join Server

- Assists in secure device activation, root key storage, and session key generation
- The join procedure is initiated by the end device by sending the Join-request message to the Join Server through the Network Server
- The Join-server processes the Join-request message, generates session keys, and transfers NwkSKey and AppSKey to the Network server and the Application server respectively
- The Join Server was first introduced with LoRaWAN v1.1

LoRaWAN MAC Message Types

LoRaWAN 1.0.x	LoRaWAN 1.1	Description
Join-request	Join-request	An uplink message, used by the over-the-air activation (OTAA) procedure
Join-accept	Join-accept	A downlink message, used by the over-the-air activation (OTAA) procedure
Unconfirmed Data Up	Unconfirmed Data Up	An uplink data frame, confirmation is not required
Unconfirmed Data Down	Unconfirmed Data Down	A downlink data frame, confirmation is not required
Confirmed Data Up	Confirmed Data Up	An uplink data frame, confirmation is requested
Confirmed Data Down	Confirmed Data Down	A downlink data frame, confirmation is requested
RFU	Rejoin-request	1.0.x - Reserved for Future Usage1.1 - Uplink over-the-air activation (OTAA) Rejoin-request
Proprietary	Proprietary	Used to implement non-standard message formats

LoRaWAN: Join-Request

- The Join-request message is always initiated by an end device and sent to the Network Server
- In LoRaWAN versions earlier than 1.0.4 the Join-request message is forwarded by the Network Server to the Application Server
  - In LoRaWAN 1.1 and 1.0.4+, the Network Server forwards the Join-request message to the device's Join Server
- The Join-request message is not encrypted

LoRaWAN: Join-Accept

- In LoRaWAN versions earlier than 1.0.4 the Join-accept message is generated by the Application Server
  - In LoRaWAN 1.1 and 1.0.4+ the Join-accept message is generated by the Join Server
  - In both cases the message passes through the Network Server
- Then the Network Server routes the Join-accept message to the correct end-device

LoRaWAN: Join-Accept

- The Join-accept message is encrypted as follows
  - In LoRaWAN 1.0, the Join-accept message is encrypted with the AppKey
  - In LoRaWAN 1.1, the Join-accept message is encrypted with different keys:

If triggered by	Encryption Key
Join-request	NwkKey
Rejoin-request type 0, 1, and 2	JSEncKey

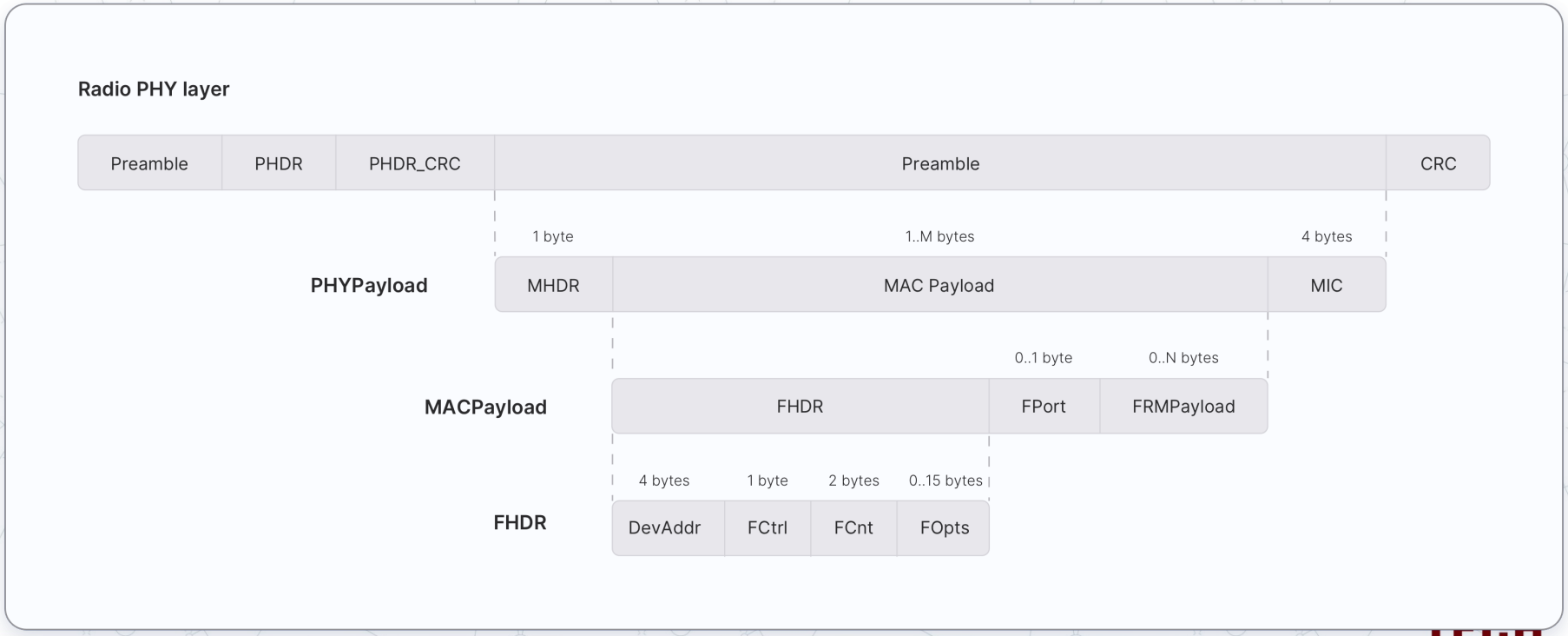
Message Integrity Code (MIC)

- The Message Integrity Code (MIC) ensures the integrity and authenticity of a message
- The message integrity code is calculated over all the fields in the message and then added to the message itself

Message Type	Fields
Join-request	MHDR | JoinEUI | DevEUI | DevNonce
Join-accept	MHDR | JoinNonce | NetID | DevAddr | DLSettings | RxDelay | CFList
Rejoin-request Type 0 and 2	MHDR | Rejoin Type | NetID | DevEUI | RJcount0
Rejoin-request Type 1	MHDR | Rejoin Type | JoinEUI | DevEUI | RJcount1
Data messages (up and down)	MHDR | FHDR | FPort | FRMPayload

LoRaWAN Data Messages

- Are used to transport both MAC commands and application data which can be combined together in a single message
- Data messages can be confirmed or unconfirmed



LoRaWAN End Device Activation

- Every end device must be registered with a network before sending and receiving messages
  - This procedure is known as activation
- There are two activation methods available:
  - Over-The-Air-Activation (OTAA)
  - Activation By Personalization (ABP)

OTAA

- The most secure and recommended activation method for end devices
- Devices perform a join procedure with the network, during which a dynamic device address is assigned and security keys are negotiated with the device
- The join procedure for LoRaWAN 1.0.x and 1.1 is slightly different

OTAA in LoRaWAN 1.1

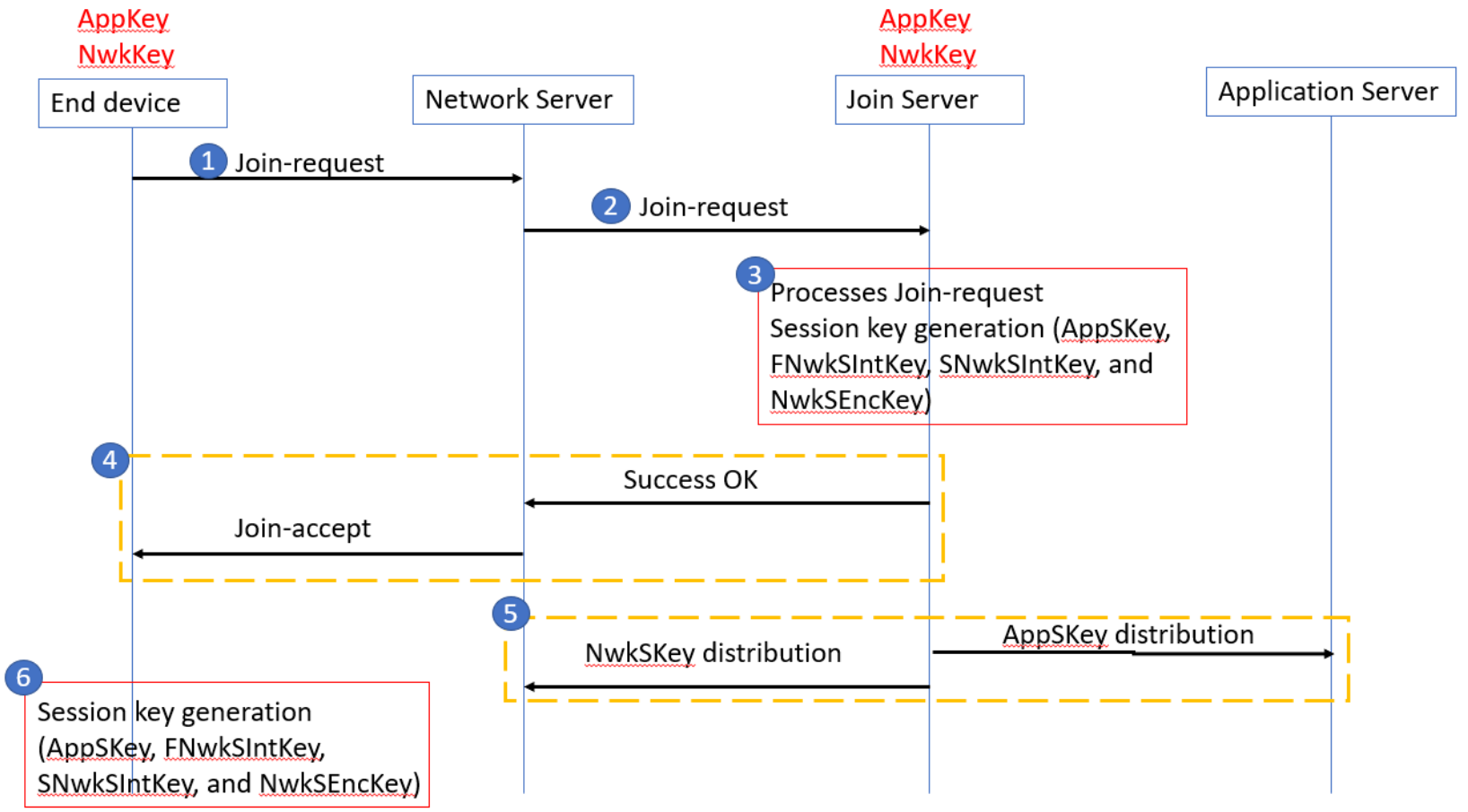
- The join procedure requires two MAC messages to be exchanged between the end device and the Join Server:
  - Join-request - from end device to the Join Server
  - Join-accept - from Join Server to the end device
- Before activation, the JoinEUI, DevEUI, AppKey, and NwkKey should be stored in the end device

OTAA in LoRaWAN 1.1

- The AppKey and NwkKey are AES-128 bit secret keys known as root keys
- The matching AppKey, NwkKey, and DevEUI should be provisioned onto the Join Server that will assist in the processing of the join procedure and session key derivation
- The JoinEUI and DevEUI are not secret and visible to everyone
- **Note**: The AppKey and NwkKey are never sent over the network

OTAA

- The join procedure is always initiated by the end device
- The end device sends the Join-request message to the network that is going to be joined

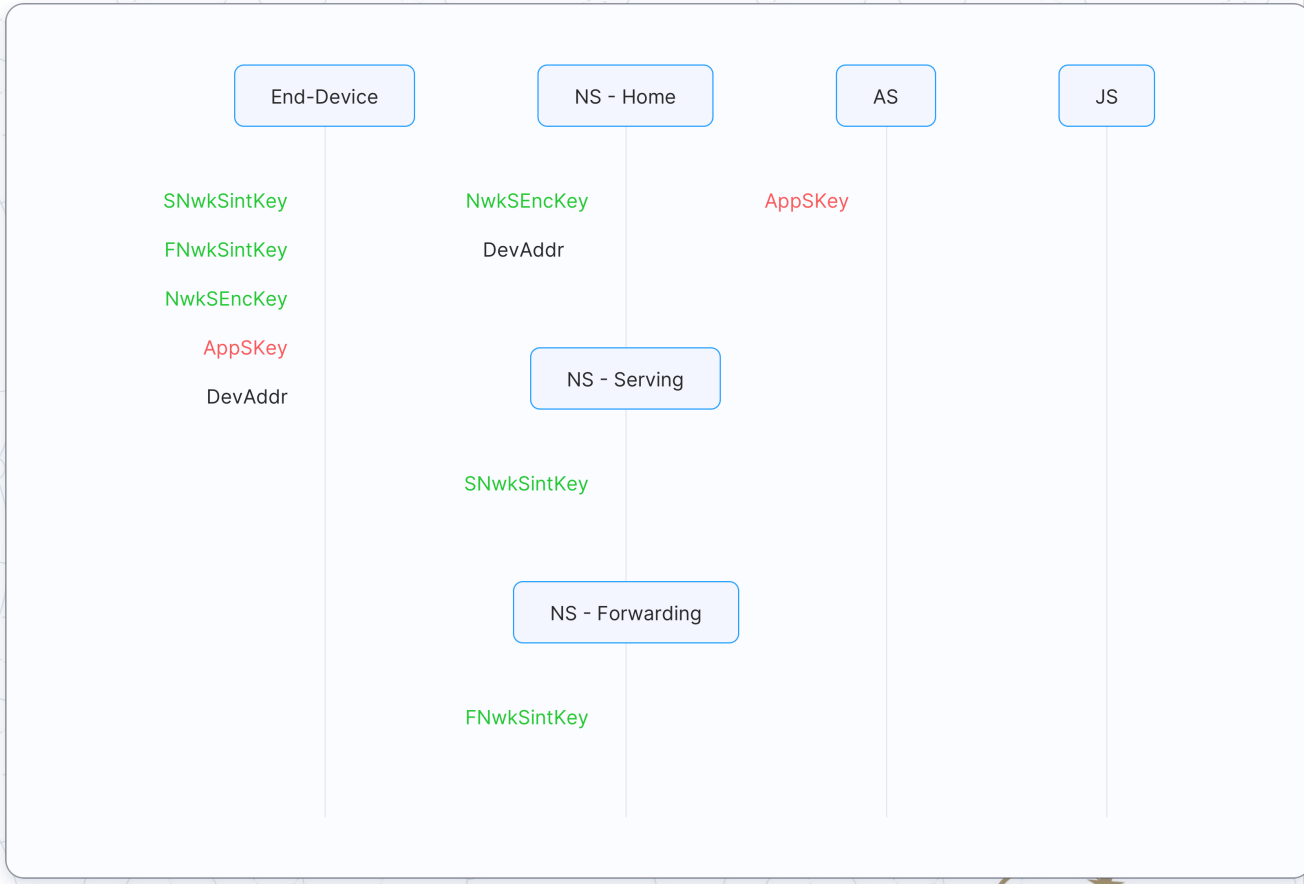


Activation By Personalization (ABP)

- ABP directly ties an end-device to a pre-selected network, bypassing the over-the-air-activation procedure
  - Requires hardcoding the device address as well as the security keys in the device
- ABP is the less secure activation method, and also has the downside that devices can not switch network providers without manually changing keys in the device
  - A Join Server is not involved in the ABP process
- An end device activated using the ABP method can only work with a single network and keeps the same security session for its entire lifetime

ABP in LoRaWAN 1.1

- The DevAddr and the four-session keys FNwkSIntKey, SNwkSIntKey, NwkSEncKey, and AppSKey are directly stored into the end device instead of the DevEUI, JoinEUI, AppKey, and NwkKey
- The same DevAddr, FNwkSIntKey, SNwkSIntKey, and NwkSEncKey should be stored in the Network Server and the and AppSKey should be stored in the Application Server



LoRaWAN Data Rates

- In addition to frequency hopping, all communication packets between end-devices and gateways also include a variable 'Data rate' (DR) setting
- The selection of the DR allows a dynamic trade-off between communication range and message duration
- Also, due to the spread spectrum technology, communications with different DRs do not interfere with each other and create a set of virtual 'code' channels increasing the capacity of the gateway

LoRaWAN Data Rates

- To maximize both battery life of the end-devices and overall network capacity, the LoRaWAN network server manages the DR setting and RF output power for each end-device individually by means of an Adaptive Data Rate (ADR) scheme
- LoRaWAN baud rates range from 0.3 kbps to 50 kbps

LoRaWAN Security

- LoRaWAN 1.0 specifies a number of security keys: NwkSKey, AppSKey and AppKey
- All keys have a length of 128 bits
- The algorithm used for this is AES-128, similar to the algorithm used in the 802.15.4 standard

LoRaWAN Session Keys

- When a device joins the network (this is called a join or activation), an application session key AppSKey and a network session key NwkSKey are generated
  - The NwkSKey is shared with the network, while the AppSKey is kept private
  - These session keys will be used for the duration of the session

LoRaWAN Session Keys

- The Network Session Key (NwkSKey) is used for interaction between the Node and the Network Server
- This key is used to validate the integrity of each message by its Message Integrity Code (MIC check)
- This MIC is similar to a checksum, except that it prevents intentional tampering with a message

LoRaWAN Session Keys

- For this MIC, LoRaWAN uses AES-CMAC
  - This validation is also used to map a non-unique device address (DevAddr) to a unique DevEUI and AppEUI

The MIC of the Join-accept message is computed using the NwkKey:

```
cmac = aes128_cmac(NwkKey, MHDR | JoinNonce | NetID | DevAddr | DLSettings | RxDelay
| CFList)
MIC = cmac[0..3]
```

LoRaWAN Session Keys

- The Application Session Key (AppSKey) is used for encryption and decryption of the payload
- The payload is fully encrypted between the Node and the Handler / Application Server component of The Things Network, which you can run on your own server
  - This means that nobody except you is able to read the contents of messages you send or receive

```
AppSKey = aes128_encrypt(AppKey, 0x02 | JoinNonce | JoinEUI | DevNonce | pad16)
```

LoRaWAN Session Keys

- These two session keys (NwkSKey and AppSKey) are unique per device, per session
- If you dynamically activate your device (OTAA), these keys are re-generated on every activation
- If you statically activate your device (ABP), these keys stay the same until you change them

```
FNwkSIntKey = aes128_encrypt(NwkKey, 0x01 | JoinNonce | JoinEUI | DevNonce | pad16)
SNwkSIntKey = aes128_encrypt(NwkKey, 0x03 | JoinNonce | JoinEUI | DevNonce | pad16)
NwkSEncKey = aes128_encrypt(NwkKey, 0x04 | JoinNonce | JoinEUI | DevNonce | pad16)
```

LoRaWAN Application Keys

- The application key (AppKey) is only known by the device and by the application
- Dynamically activated devices (OTAA) use the Application Key (AppKey) to derive the two session keys during the activation procedure
- In The Things Network you can have a default AppKey which will be used to activate all devices or customize the AppKey per device

```
AppSKey = aes128_encrypt(AppKey, 0x02 | JoinNonce | JoinEUI | DevNonce | pad16)
```

LoRaWAN Security

- By providing these two keys, it becomes possible to implement 'multi-tenant' shared networks without the network operator having visibility of the users payload data
- The keys can be Activated By Personalisation (ABP) on the production line or during commissioning, or can be Over-The-Air Activated (OTAA) in the field
  - OTAA allows devices to be re-keyed if necessary

LoRaWAN: Frame Counters

- Anyone will be able to capture and store messages
  - It's not possible to read these messages without the AppSKey, because they're encrypted
  - Nor is it possible to tamper with them without the NwkSKey, because this will make the MIC check fail
  - It is however possible to re-transmit the messages
- These so-called replay attacks can be detected and blocked using frame counters

LoRaWAN: Frame Counters

- When a device is activated, these frame counters (FCntUp and FCntDown) are both set to 0
- Every time the device transmits an uplink message, the FCntUp is incremented and every time the network sends a downlink message, the FCntDown is incremented
- If either the device or the network receives a message with a frame counter that is lower than the last one, the message is ignored

LoRaWAN: Frame Counters

- These frame counters reset to 0 every time the device restarts (when you flash the firmware or when you unplug it)
- As a result, The Things Network will block all messages from the device until the FCntUp becomes higher than the previous FCntUp
- Therefore, you should re-register your device in the backend every time you reset it

Cellular Networks Outline

Cellular Networks

- 2G – Edge
- Femtocell
- 4G/LTE

Cellular Network Radio Frequencies

- Modern cell phones use a number of different frequencies to transmit data
  - Depends on the country and mobile operator network
- Some countries provide government-owned cellular services
  - Some lease spectrum to mobile operators to provide cellular access

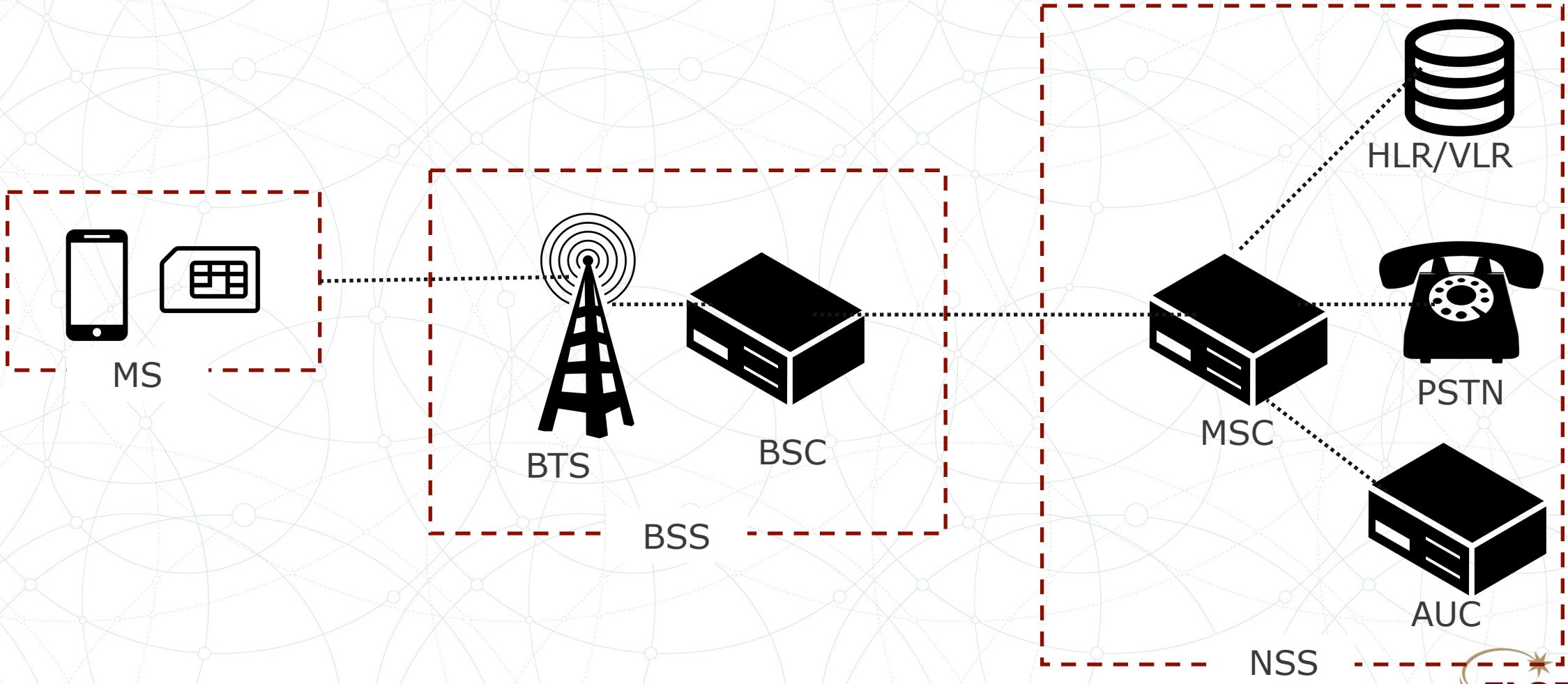
2G

- 2.5G Global System for Mobile (GSM), 2.75G General Packet Radio Service (GPRS), and Enhanced Data Rates for GSM Evolution (EDGE)
- Has been the most widespread cellular protocol used worldwide
  - Some devices still use it for backward-compatibility or back-up connection in the absence of other access opportunities

GSM Protocol

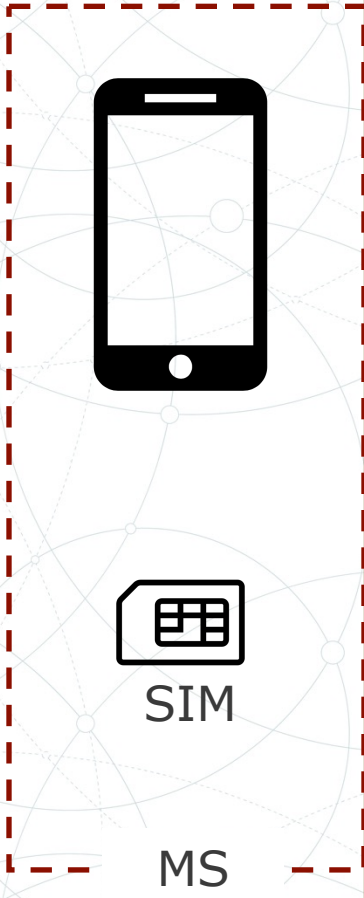
- Global System for Mobile Communications (GSM)
- First digital, circuit-switched network for carrying voice
- Expanded for carrying data (GPRS)
- Implemented cryptographic algorithms for security
- By mid-2010s, had over 90% of market share
- Obsoleted by LTE (4G); reached end-of-life for AT&T in 2017

GSM Network Model



GSM Mobile Station

- Mobile Station (MS):
  - User handset/device that connects to the GSM network
- Subscriber Identity Mobile (SIM):
  - Removable media that identifies the unique International Mobile Subscriber Identifier (IMSI) for the mobile station and the 128-bit Authentication Key (K_i)
- International Mobile Subscriber Identifier (IMSI):
  - Unique identifier consisting of Mobile Country Code (3 digits), Mobile Network Code (2 | 3 digits), Mobile Subscriber Identification (10 | 15 digits)

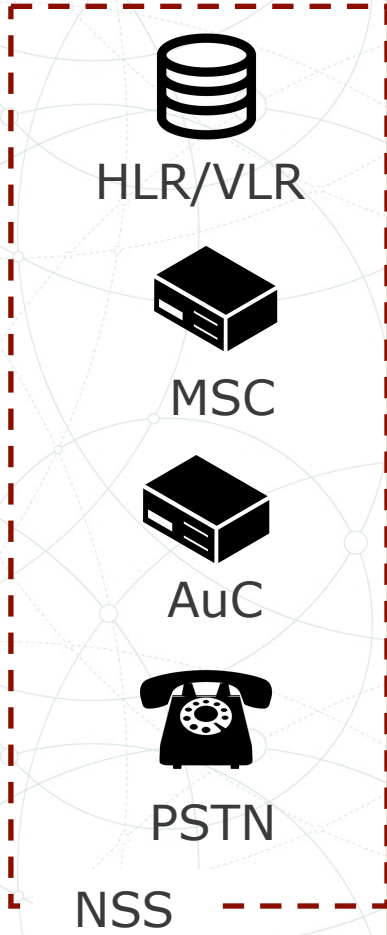


GSM Base Station Subsystem

- Base Transceiver Station (BTS):
  - Devices that facilitates radio connectivity for the mobile station (e.g., cell tower)
- Base Station Controller (BSC):
  - Facilitates the management of several BTSs;
  - Handles radio allocation, BTS handovers, etc.
- Base Station Subsystem (BSS):
  - Consists of BTS/BSC;
  - GSM network component that connects mobile station and network switching system

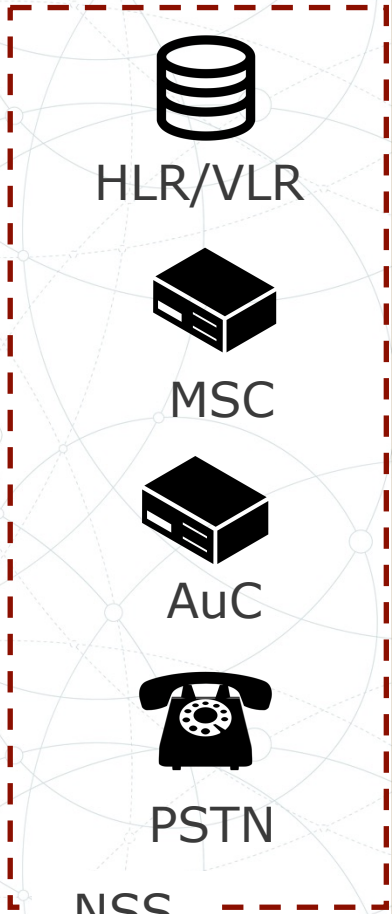
GSM Network Switching Subystem

- Network Switching Subsystem (NSS):
  - Back-end network for GSM provider and services
- Home Location Register (HLR):
  - Database that contains the IMSI for each MS on the network
- Visitor Location Register (VLR):
  - Database that facilitates roaming operations for MS devices

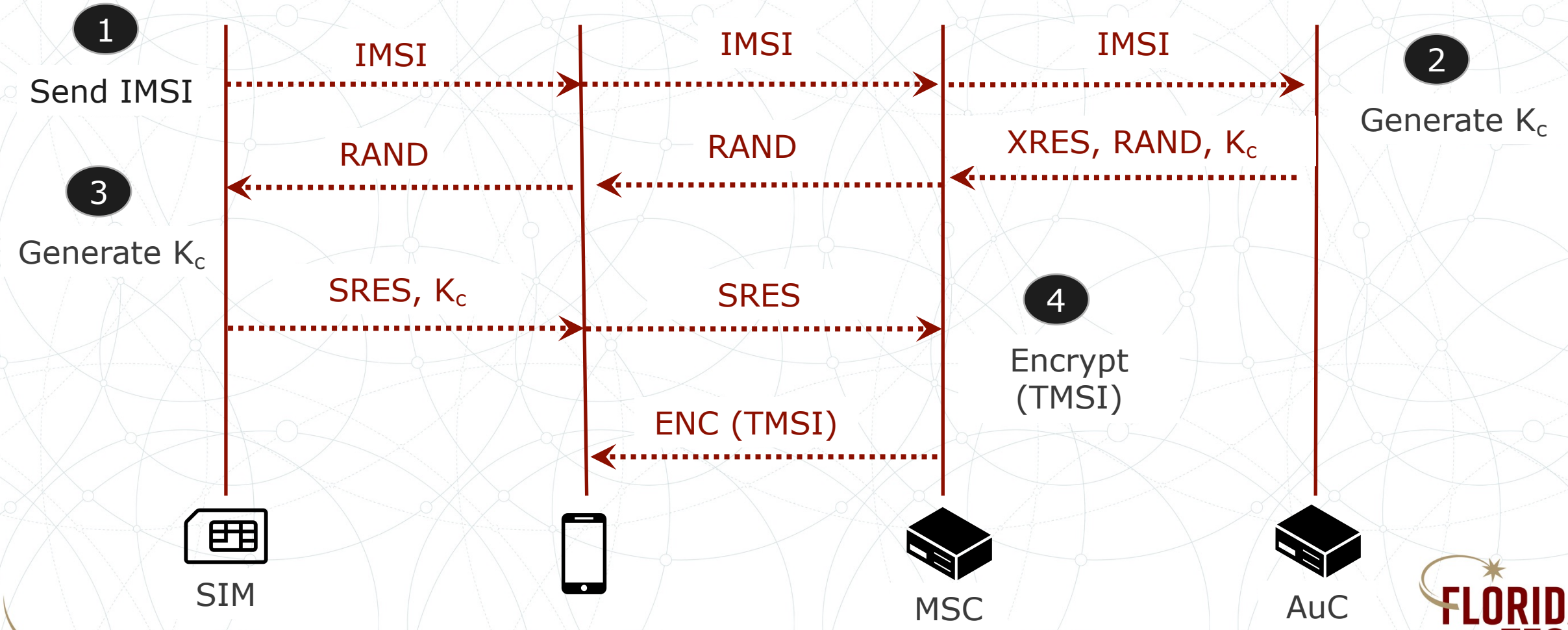


GSM Network Switching Subystem

- Mobile Switching Center (MSC):
  - Responsible for routing
- Authentication Center (AuC):
  - Facilitates user identification and authentication for SIM cards
- Public Switched Telephone Network (PSTN):
  - Public interface to GSM network and other network providers



GSM Authentication



GSM Authentication

- Exchange validates the identity of the subscriber through the use of the IMSI and the associated subscriber key, K_i, stored on the SIM card
  - Involves the SIM, the MS, the MSC, and the AuC
- AuC and SIM have knowledge of IMSI and Ki as part of the device registration process
  - When the MS connects to network, SIM shares IMSI info which is forwarded to AuC

GSM Authentication

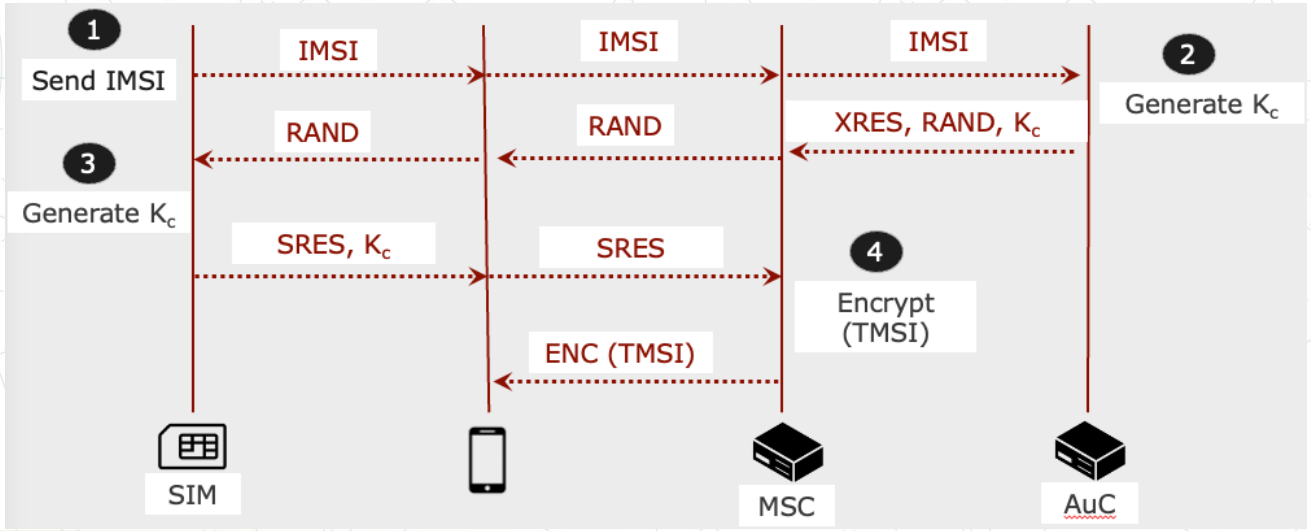
- The AuC retrieves K_i linked to IMSI and selects a random value RAND
- The RAND is used with two algorithms, A3 and A8, with the subscriber key K_i to generate the temporary cipher key K_c and the expected response (XRES)
- AuC shares K_c, RAND and XRES with the MSC, ending its role in the authentication process

GSM Authentication

- Next, MSC shares the RAND with the MS, which sends it to the SIM
  - Similar to AuC, the SIM uses the RAND to generate Kc and signed response SRES
  - SRES is delivered to MS, which forwards it over the air interface to the MSC for validation

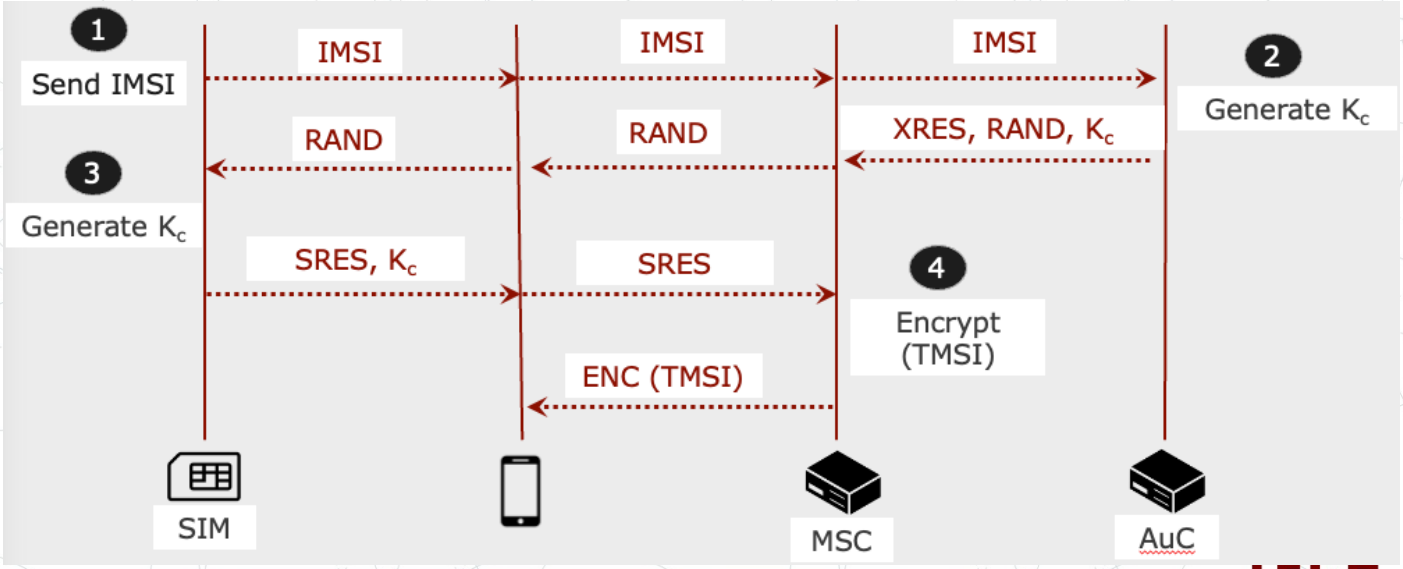
GSM Authentication

- The MSC compares SRES to XRES;
  - If match, validating the ME's identity
  - Then, MSC generates and encrypts TMSI for the ME to use and delivers it over the air interface



GSM Authentication: Vulnerability

- SIM card (and so the ME) is authenticated to the AuC
  - Prevent unauthorized devices accessing network services
- The authentication exchange does not validate the identity of the provider to the ME

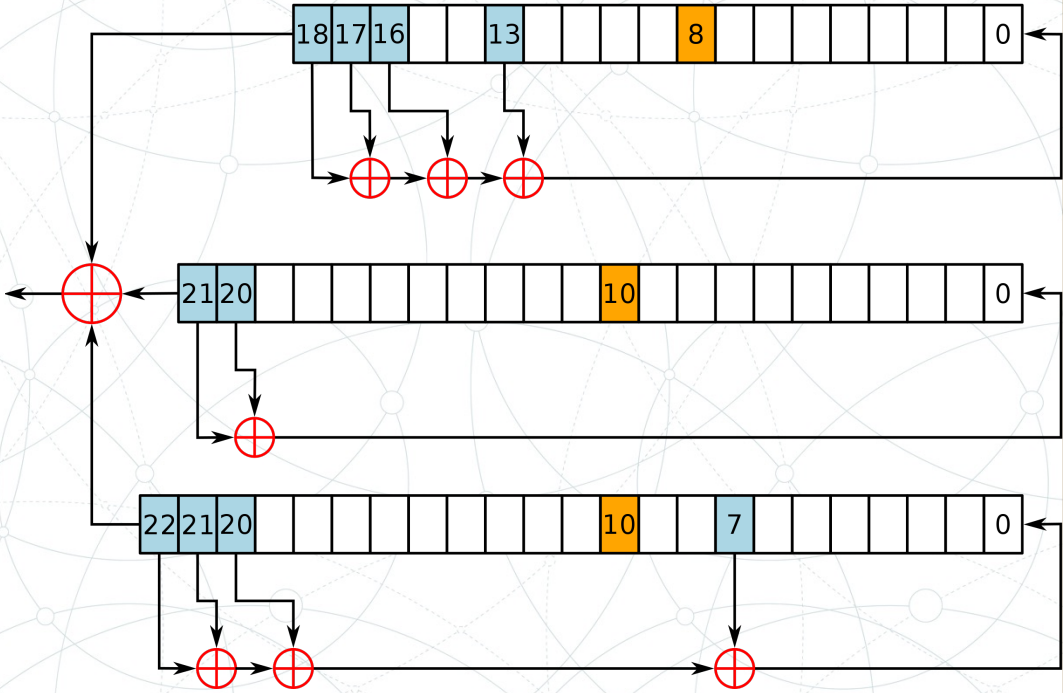


GSM Encryption

- Use the A5/1 cipher to provide confidentiality controls of traffic delivered over the GSM air interface between MS and BSC
- A5/1 is a stream cipher implemented using a Linear Feedback Shift Register (LFSR) mechanism
- Temporary cipher key Kc is used as the A5/1 input to generate keystream data
  - Ease of implementation in hardware and reduced implementation cost

A5/1

- Developed in 1987 (before GSM)
- 114 bits of GSM are XORd with 114 bits of Keystream material to produce encrypted data
- Uses three Linear Feedback Shift Registers (LFSR)



GSM Encryption

- The plaintext data is XOR'd with the keystream data to generate ciphertext
- Ciphertext is similarly XOR'd with matching keystream data at the recipient to decrypt

GSM Eavesdropping

- Commercial tools are expensive
  - But, you can build your own GSM sniffer using inexpensive hardware
  - For ex. RTL-SDR
- GSM packet captures disclose basic information about the network, but do not reveal the contents of phone call/SMS etc. due to being encrypted

A5/1 Key Recovery

- Precomputed reference attack for full key recovery
- In 2008, gsm-tvoid; keystream data to known keystream state information in lookup tables
  - Using set of precomputed 288 quadrillion possible entries (apprx 2tb storage), adversary recovers K_i in approx. 30mins
  - It was taken offline without explanation
  - Possible government intervention

A5/1 Key Recovery

- In 2009, A5/1 cracking project reproduced the work previously published
  - As a plus, they distribute key recovery lookup tables through peer-to-peer networks to prevent them from being taken offline
- In 2011, 'Kraken', practical tool that integrates with AirProbe for effective capture and decryption of GSM traffic with the A5/1 tables
  - Later, 'Pytacle' tool improved features of the tool by adding play back capabilities for full and simple passive GSM decryption attack

https://github.com/0xh4di/kraken

Femtocell

- Extend the carrier network, leveraging the consumer's broadband connection for uplink connectivity
- Femtocell devices (e.g., Home NodeB or HNB) allow consumers to establish a relatively short-range extension of the carrier network that provides similar connectivity services (e.g., voice, data, SMS/MMS)
  - Also offers attackers new opportunities to attack the carrier infrastructure, as well as User Equipment devices

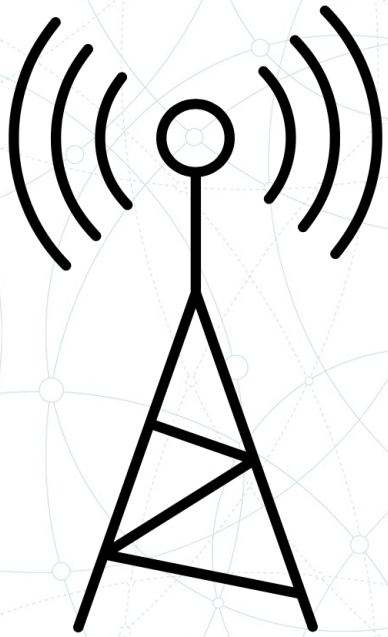
Femtocell Attack

- HNB is authorized device on the carrier network and has access to dynamic key information used to encrypt/decrypt the 3G connection
  - MiTM to manipulate and intercept phone calls
- They found a way to have root access to femtocell device and run their codes in them to sniff the traffic
  - Presented at Defcon 21

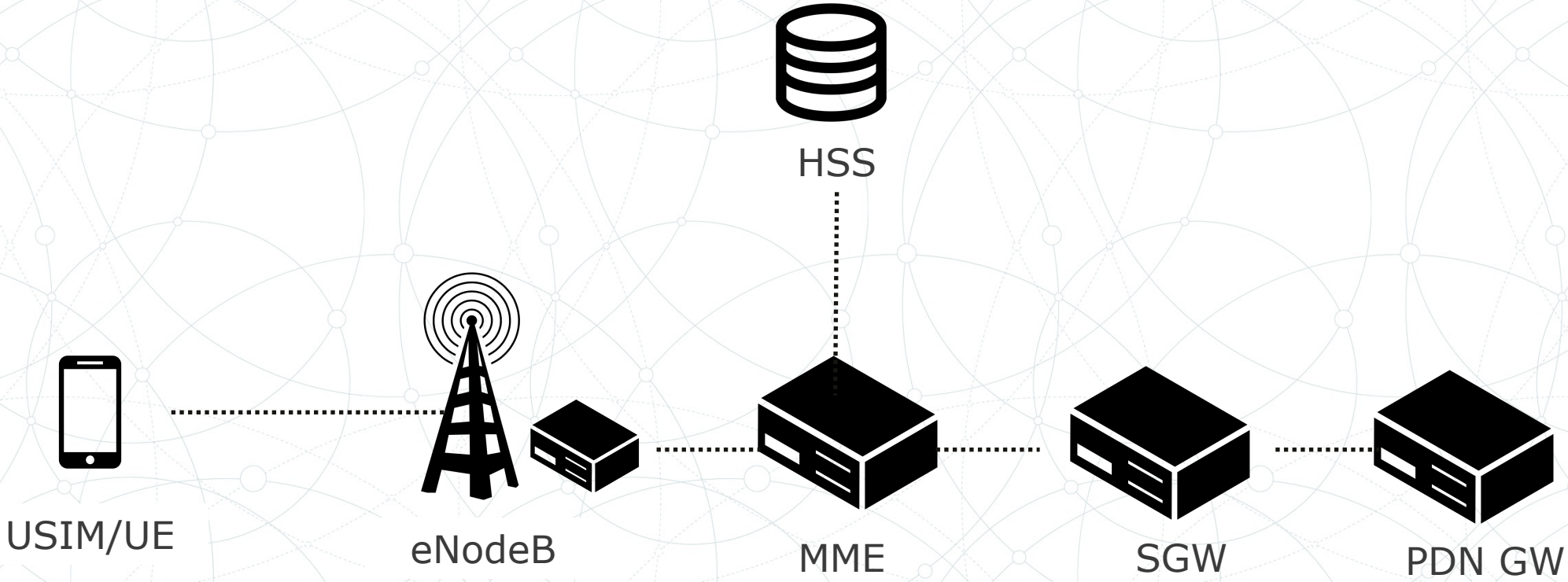
https://www.youtube.com/watch?v=gfcq8clu1RI

4G/LTE

- Long-Term Evaluation (LTE) Protocol
- Predecessor to GSM
- Marketed as 4G LTE or Advanced 4G
- In addition to higher speeds
  - Offers improvements for privacy
  - Introduces new encryption schemes

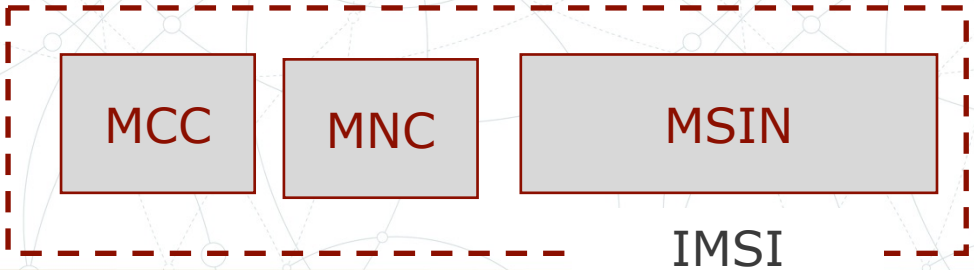


LTE Network Model



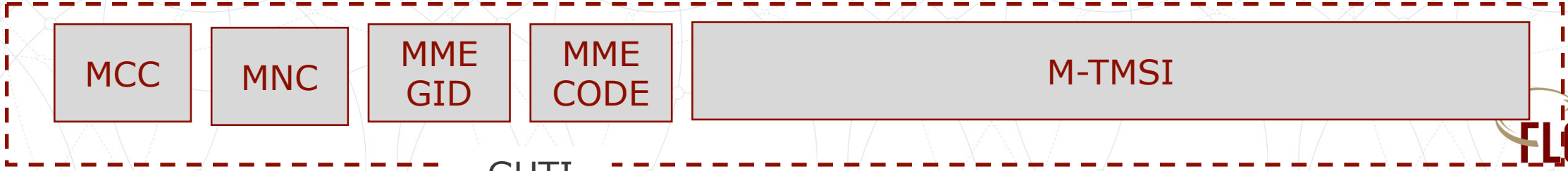
LTE Addressing Scheme for IMSI

- IMSI is made up of three components:
  - MCC: Mobile Country Code, identifying the country of the end-user
  - MNC: Mobile Network Code, identifying the home network
  - MSIN: Mobile Subscriber Identification Number, identifying the user within MCC and MNC context



LTE Addressing Scheme for IMSI

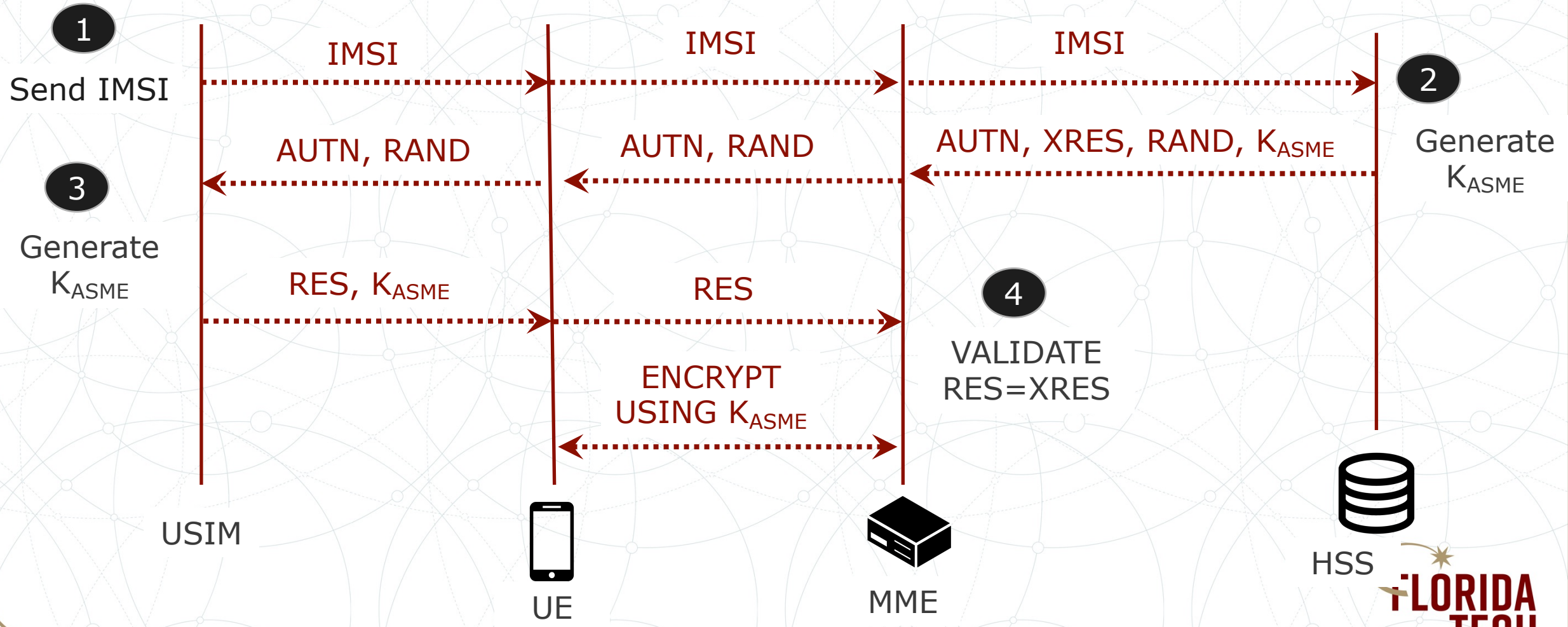
- IMSI value is stored on the USIM and a fixed value
  - Acts as a shared identifier for UE (e.g., LTE phone) and the HSS for the associated authentication key ”K”
- To ensure privacy of the unique handset (such as identifying an IMSI to an individual), LTE introduces a Globally Unique Temporary ID (GUTI) which consists of the MCC, MNC MME info and the Temporary Mobile Subscriber ID (TMSI)



LTE Authentication

- Mutual auth of the handset and the network infrastructure through the Evolved Packet System Authentication and Key Agreement (EPS-AKA)
- Similar to GSM/3G, authentication in LTE relies on the identification function and shared key content provided by the IMSI (International Mobile Subscriber Identity)

LTE Authentication



LTE Authentication Vulnerability

- The IMSI is sent in plaintext
  - Rogue LTE network can get IMSI
  - Privacy threat to IMSI
- Yet, the secret K never is disclosed to the UE from USIM, preventing rogue applications from stealing the value and limiting attacker's ability to clone the value onto another USIM

LTE Encryption

- LTE supports algorithm flexibility
- 3GPP systems were limited to a handful algorithms and these could not be replaced without changes to the network infrastructure
- Yet, LTE networks could adapt to new algorithm option to mitigate any flaw
  - Let's say there is a flaw found in AES

LTE Supported Encryption Algorithms

- NULL Algorithm:
  - Does not provide confidentiality of network traffic
  - In some cases, need to provide service outweighs the desire for security in LTE
  - Provides network access for devices lacking USIM card for situations such as emergency services (e.g., 911 in US)
  - May create opportunity for attacker to impersonate a legitimate carrier network without the need for cryptographic attacks

LTE Encryption Algorithm Tradeoffs

Scheme	Advantages	Disadvantages
Kasumi	Offers strong encryption via 128-bit keys Optimized for hardware implementation Offers resistance to block cipher attacks	Vulnerable to algebraic attacks
SNOW 3G	Fits 3G security requirements Offers protection against algebraic attacks	Computationally complicated
Milenage	Fits 3G security requirements Offers strong encryption via 128-bit keys Protects against side-channel attacks	Does not require standard algorithm Some interoperability issues
ZUC	**Fits 3G security requirements** **Offers strong encryption via 128-bit keys** **Built on sound design principles**	**Still under scrutiny**

FLORIDA TECH

FLORIDA'S STEM UNIVERSITY™

Thank you. Questions?

Dr. Abdullah Aydeger