Unicus™ IoT Cybersecurity for the Internet of Things

UM Labs
2-6 Boundary Way
London
SE1 8HP

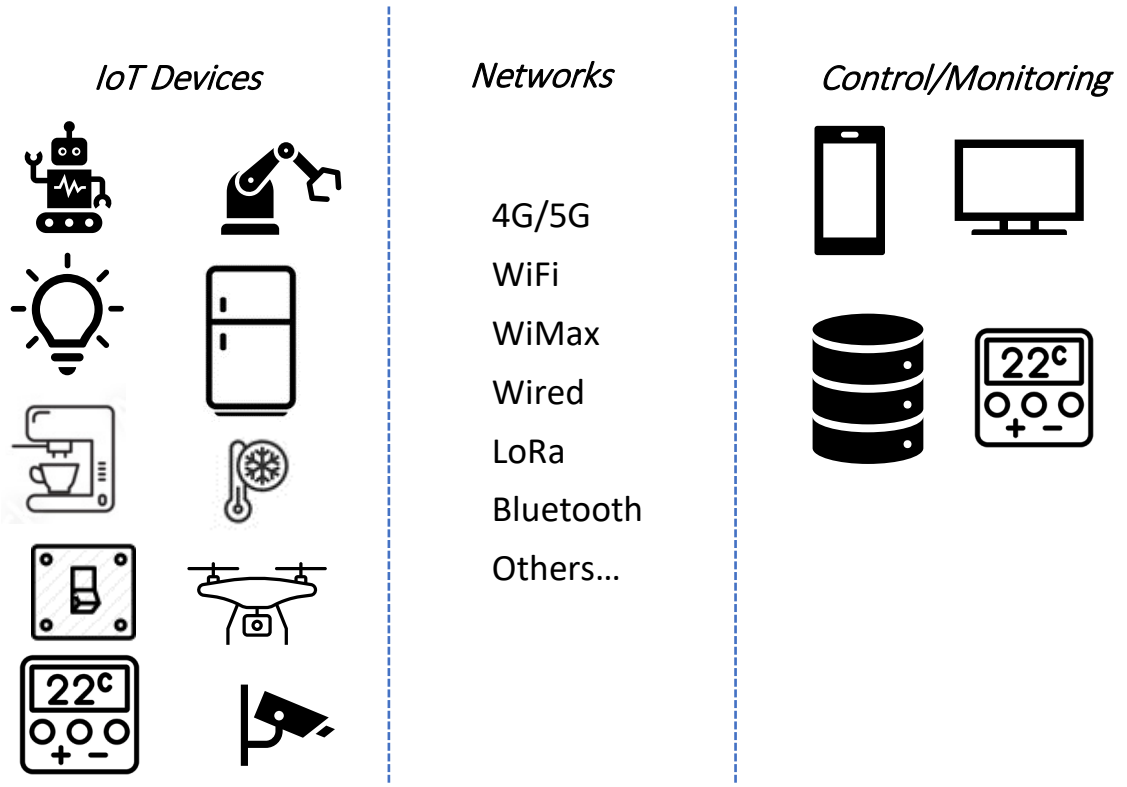
Introduction

The Internet of Things, IoT, describes the use of IP networks to interconnect a range of devices providing monitoring and response services and to connect those devices to a central control point. In common with all interconnected devices and services using IP networks, IoT devices, the control systems and the communication between them are all at risk of attack. While some of the threats facing IoT networks are shared with other IP services and applications, the architecture of IoT systems exposes those systems to a set of threats which can only be addressed with countermeasures designed specifically to address the IoT specific threats. The European Telecommunications Standards Institute (ETSI) has published technical specifications for IoT Cybersecurity (ETSI, 2020).

This document describes how Unicus™ IoT provides a foundation for building and deploying secure IoT applications and networks.

IoT "Things"

One of the challenges of implementing cybersecurity for IoT is the range of devices or "things" that may be deployed. Simple devices can include a light source which has only two states on or off and a switch controlling the light source. These simple devices either feed data to a control system (a switch reporting on or off state) or respond to commands from a control system (turning a light on or off). More complex devices include surveillance cameras which feed video and audio to a control system and respond to commands such as start/stop media stream and motion commands to focus on an area of interest. At the top-end, devices can include robots, drones, complex industrial monitoring and control systems.



Monitoring and control systems may vary from a smartphone app controlling domestic lighting and heating to sophisticated industrial control systems coupled with an extensive database. Monitoring and control systems may be co-located with the IoT device or may be located in a remote monitoring centre. The networks used to link IoT devices and monitoring systems range from short range links to broadband links to an ISP. Some device types can operate as both monitoring and control devices while others provide centralised routing or hub functions.

IoT Devices are often limited in both processing capacity and in their power source. Some devices may even be battery powered or reliant on local energy sources such as wind and solar power. These factors limit the device's security capability. Even mid-range devices with sufficient power and processing capacity are often built with limited or no effective security controls. Research by UM Labs has shown that many surveillance cameras, including body-worn cameras designed for emergency services often have poor or non-existent security controls and so are open to a wide range of attacks.

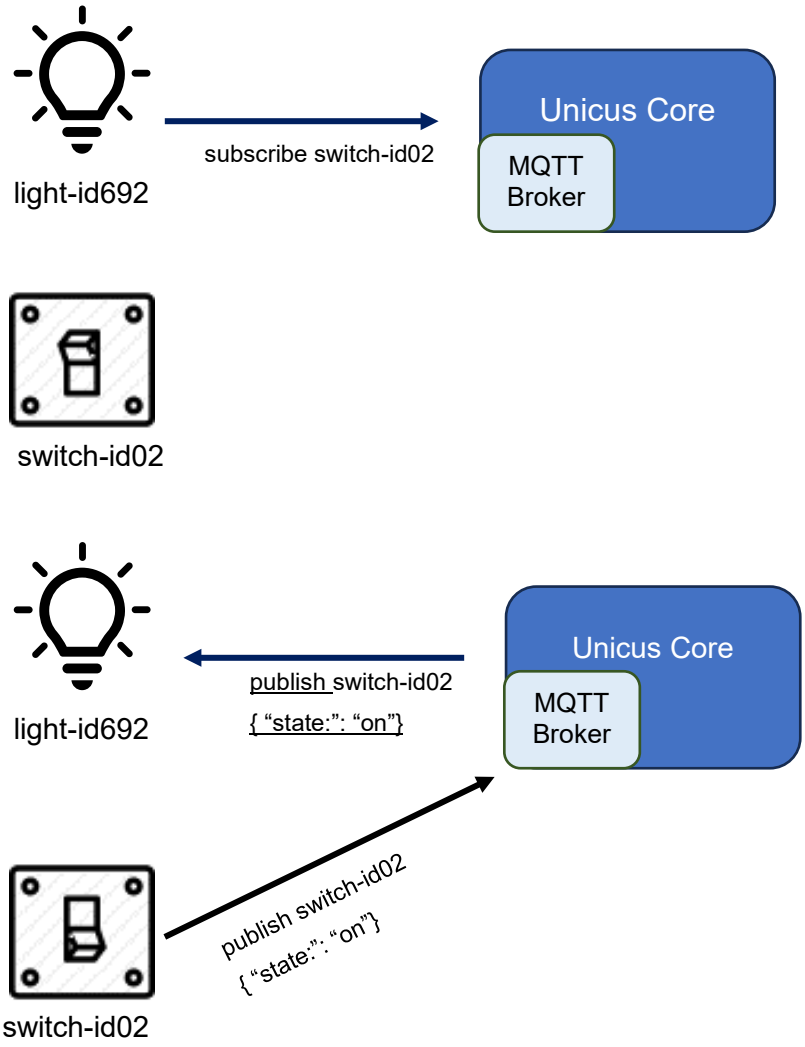
Unlike network applications such as Web or Email, there is no single defined protocol or set of protocols used by IoT applications. The following table outlines some of the many IoT protocols in use.

Protocol	Description	Protocol Standard	Network
AMQP	Operation over poor networks	Oasis (Oasis, 2011)	TCP/IP
CoAP	Constrained Application Protocol, low-end devices on limited bandwidth networks.	IEFT (Shelby, 2014)	TCP/IP
DDS	Machine-to-Machine, low latency	Object Management Group (OMG, 2015)	TCP/IP
HTTP(s)	REST API for generic applications	IETF (Fielding, 2014)	TCP/IP
LWM2M	Lightweight Machine-to-Machine	Open Mobile Alliance (Open Mobile Alliance, 2019)	
Matter	Smart home and IoT devices	CSA (Connectivity Standards Alliance)	TCP/IP
MQTT	Low-end devices, limited bandwidth network	Oasis (Oasis, 2009)	TCP/IP
RTMP	Steaming audio/video	IETF (Schulzrinne, H., 2016)	TCP/IP
RTSP	Streaming audio/video	IETF (Schulzrinne, 2016)	
XMPP	Message based Machine-to-Machine	IETF (Saint-Andre, 2011)	TCP/IP
Zigbee	Short range, low power networks.	CSA (Zigbee Alliance, 2015)	LR-WPAN

While most of these protocols operate over a TCP/IP network, some protocols such as Zigbee operate at a lower level using Low-Rate Wireless Personal Area Networks (LR-WPAN). To add additional complexity, these protocols can operate over multiple network types including Bluetooth, WiFi, WiMax, LoRa, 4G/5G and wired connections.

Device limitations, the number of available protocols and network types must be considered when implementing effective security controls for IoT.

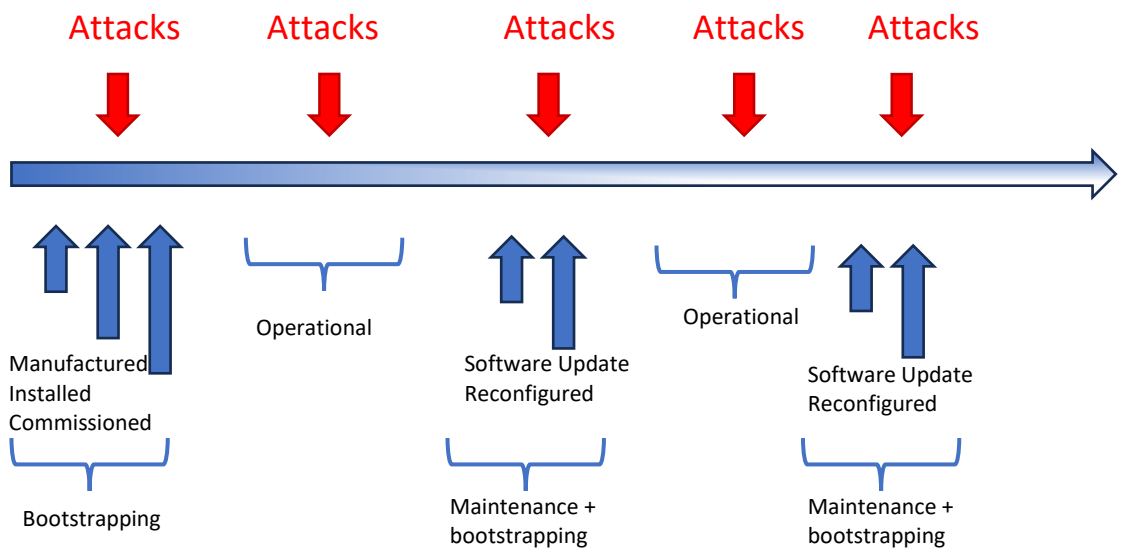
The design and operation of these protocols varies widely. REST APIs running over HTTPS work by exchanging messages directly between devices and systems. Protocols such as RTMP and RTSP transport video streams between end-points set up by other supporting protocols. MQTT, which is widely deployed for IoT systems, uses a *Subscribe/Publish* mechanism. A control device will publish information on state changes, for example a light switch can publish state changes between On and Off. A device such as a light subscribes to receive notifications of those changes. Subscribe and Publish requests are handled by a *Broker* (included in Unicus™ IoT).



IoT Device Life Cycle

IoT devices are often installed in remote locations where physical access is difficult and expensive. Once installed, a device needs to be configured and upgraded at various points in its lifetime.

Devices face security threats at each stage in its lifecycle. An RFC issued by the Internet Research Task Force (IRTF) summarises the lifecycle of an IoT device (Garcia-Morchon, 2019).



Deployed IoT devices must be configurable. That configuration may change in the device's lifetime. A device will retrieve its configuration during the bootstrapping phase of device deployment. This process, known as provisioning, uses a mechanism where the device connects to a pre-configured system, confirms its identity and receives its configuration data. This provisioning process is typically repeated each time the device restarts or when instructed by a central system. Once provisioned, the device enters an operational phase.

An IoT device is exposed to attack in both the bootstrapping phase and operational phase. Effective cybersecurity controls must protect a device for its entire lifecycle.

IoT Security Threats

The IRTF RFC summarises the threats facing IoT systems as:

- Vulnerable software, device software may hide flaws or deliberate back-doors.
- Privacy Threat, tracking or monitoring a device may pose a privacy risk.
- Cloning, a device may be cloned during manufacture (by an untrusted factory) or an operational device may be copied.
- Malicious Substitution, an operational device may be replaced by a compromised clone.
- Eavesdropping, communication from a device may be monitored by an untrusted 3rd party.
- Man-in-the-middle Attacks, an attacker may intercept communication from a device, modifying data-in-transit.
- Firmware Attacks, device firmware may be subject to unauthorised modification during a system update.
- Extraction of private information, an attacked may attempt to extract device identity information and use that information in a subsequent attack.
- Routing Attack, intermediate devices in the network connections used by a device may be attacked to re-route information flows from a device.

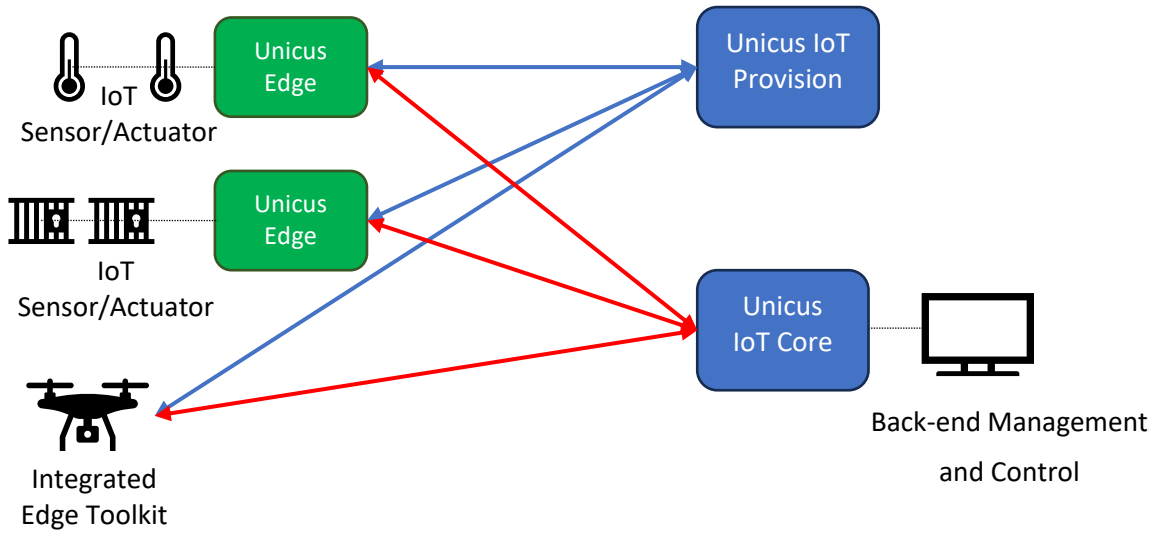
- Denial of Service Attack, a device may subject to a flooding attack preventing normal operation.

These threats define in the RFC focus mostly on IoT devices. Centralised IoT systems managing multiple devices must also be protected.

Unicus™ IoT

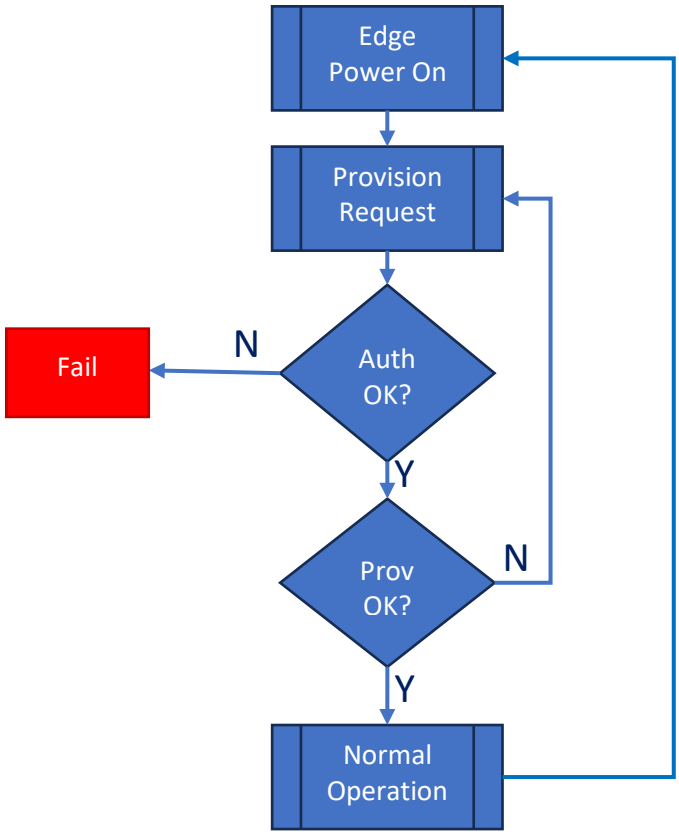
The Unicus platform was designed by UM Labs to provide a foundation for implementing cybersecurity controls for real-time communications. The platform was initially used to for securing IP based voice and video phone services using the Session Initiation Protocol (SIP). The resulting product, *Unicus RTC* is now widely deployed securing SIP Trunk connections and interpersonal calls in high security environments. More recently *Unicus SVSS*, which was developed with funding from the UK Defence Science and Technology Laboratory (DSTL), was deployed to secure streaming video services. Streaming video includes static and mobile surveillance cameras and can be considered a special case of IoT.

Unicus IoT builds on Unicus SVSS and generalises the design to handle security for any IoT application. The Unicus architecture is described in detail in *Unicus™ A Cybersecurity Platform for Real-Time Applications on IP networks,* a whitepaper available from UM Labs. Unicus IoT utilises both the Core and Edge versions of the platform. The Core operates in a public or private cloud and controls multiple Edge Devices. An Edge Device is either a physical device running in low-end hardware or a toolkit which may be embedded into an IoT sensor or actuator device. Multiple Edge Devices are controlled by Core System. Edge Devices bootstrap by obtaining their operational configuration from a Provisioning Service. This service is provided by a Unicus Core System. This may be the same Core System as the IoT Core or may be operate on a separate Core System.



The operational details of a Unicus IoT deployment will depend on the details of the application in use and on the network topology, but in most cases a standard methodology is used.

1. A newly deployed Edge Device is installed with an *Identity Module* which includes a unique key and certificate for that device with details of the location of a provisioning server.
2. When an Edge Device is powered on, it establishes an encrypted connection to the nominated provisioning server and downloads the current operational configuration. This connection is mutually authenticated (the Edge Device validates the Core System's identity, and the Core System validates the connecting Edge Device.
3. If the previous step is successful the Edge Device connects to the Core System using an encrypted, mutually authenticated connection and starts normal operation. All communication between the Edge and Core uses the connection initiated by the Edge. This enables the Edge Device to be configured to block all incoming connections and simplifies operation in networks where the Edge/Core connection must traverse multiple Network Address Translation (NAT) devices.
4. If the Edge device is re-powered or if a software update or other maintenance is needed, this process is repeated.



IoT Device Cybersecurity

The Core/Edge architecture provided by Unicus IoT secures both the IoT device and the central monitoring and control systems at all stages of the device's life cycle. The following table summarises the security countermeasures for each of the IoT security threats identified in the IRTF RFC.

Threat	Role	Countermeasures
Vulnerable Software	Security Enforcing	Both the UM Labs Edge Device and Core System software are built to a security first/zero trust design and tested to minimise the risk of software vulnerabilities. The preferred Edge Device security policy of blocking all incoming connections minimises the risk of attackers exploiting vulnerabilities in sensor or actuator software.
Privacy Threat	Security Enforcing	All communication between the Edge Device and provisioning server and Core System are encrypted protecting data-in-transit. The preferred Edge Device security policy of blocking all incoming connections blocks attackers from monitoring that device.
Cloning	Security Enforcing	Each Edge Device has a unique private key used to validate the device's provisioning request and a separate unique key for operational use. Where possible, these keys are stored in a secure location making cloning difficult. In addition, both the provisioning service and the Core system monitor connections from each Edge Device recording the device's network address and raising alerts when multiple connection attempts are detected.
Malicious Substitution	Security Enforcing	The defences against cloning also protected against malicious substitution. Any attempt to substitute a device without replicating the devices protected security keys will be detected immediately and blocked.
Eavesdropping	Security Enforcing	All communication between the Edge Device and provisioning server and Core System is encrypted protecting data-in-transit.
Man-in-the-middle Attacks	Security Enforcing	The Edge Device's security keys and the use of mutual authentication for connections to both the provisioning service and Core System protect against man-in-the-middle attacks.
Firmware Attacks	Security Enforcing	The preferred Edge Device security policy which blocks all incoming connections and controls outgoing connections prevent attackers from modifying IoT device firmware and block malicious firmware installed during manufacture from connecting to external system and exporting data.
Extraction of Private Information	Security Enforcing	The preferred Edge Device security policy which blocks all incoming connections prevents attackers from extracting private information. In addition, where possible critical identity information is stored in a secure location.
Denial of Service Attacks	Security Supporting	The Edge Device and the Core System use the Unicus layered security architecture to detect denial of service attacks and where possible to push the blocking action to a lower level or to the network perimeter.

Core Cybersecurity

While IoT monitoring and actuator devices are probably the primary target for an attack on an IoT system, the core monitoring and control services must also be protected along with any connecting network that could be compromised through a weakly protected IoT device or control node. Those core services are protected by a Unicus IoT Core System. All connections from Edge Devices are mediated by Core System and reporting and control information exchanged with the IoT device is validated by the Unicus Core System. This mediation and the security controls applied to those connections ensure that all information exchanges between core monitoring and control services and IoT devices.

The Core System is built on the Unicus platform. The layered security controls provided by Unicus protect the Core System from attack and protect the IoT monitoring and control systems with will connect via the Core to each of the deployed IoT devices. As described in the UM Labs Unicus whitepaper, the Core's modular architecture enables new IoT protocols to be added when needed. These protocols will benefit from the layered security architecture enabling those modules to focus on validating the protocol requests while relying on the Unicus platform to handle generic threats.

References

Connectivity Standards Alliance. (n.d.). *Matter Specification.* Retrieved from https://csa-iot.org/developer-resource/specifications-download-request/

ETSI. (2020, June). *Cyber Security for Consumer Internet of Things:.* Retrieved from https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf

Fielding, R. (2014, June). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content.* Retrieved from https://datatracker.ietf.org/doc/html/rfc7231

Garcia-Morchon, O. (2019, April). *Internet of Things (IoT) Security: State of the Art and Challenges.* Retrieved from IRTF RFC 8576: https://datatracker.ietf.org/doc/html/rfc8576

Oasis. (2009, March). *MQTT Version 5.0.* Retrieved from https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

Oasis. (2011, October). *AMQP v1.0.* Retrieved from https://www.amqp.org/sites/amqp.org/files/amqp.pdf

OMG. (2015, April). *OMG Data Distribution Service.* Retrieved from https://www.omg.org/spec/DDS/1.4/PDF

Open Mobile Alliance. (2019, June). *Lightweight Machine to Machine Technical Specification: Core.* Retrieved from https://www.openmobilealliance.org/release/LightweightM2M/V1_1_1-20190617-A/OMA-TS-LightweightM2M_Core-V1_1_1-20190617-A.pdf

Saint-Andre, P. (2011, March). *https://datatracker.ietf.org/doc/html/rfc6120.* Retrieved from https://datatracker.ietf.org/doc/html/rfc6120

Schulzrinne, H. (2016, December). Retrieved from https://datatracker.ietf.org/doc/html/rfc7826

Schulzrinne, H. (2016, December). *Real-Time Streaming Protocol Version 2.0.* Retrieved from https://datatracker.ietf.org/doc/html/rfc7826

Shelby, Z. (2014, June). *The Constrained Application Protocol (CoAP).* Retrieved from https://datatracker.ietf.org/doc/html/rfc7252

Zigbee Alliance. (2015, August). *ZigBee Specification.* Retrieved from https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf