SCOTT:

Secure COnnected Trustable Things

Security Scan Methodology for Cloud Connected IoT Devices

Document Type	Whitepaper
Primary Author(s)	Silke Holtmanns | Nokia Bell Labs
Document Version / Status	V1.0
Distribution Level	PU (public)

Project Acronym	SCOTT
Project Title	Secure COnnected Trustable Things
Project Website	www.scottproject.eu
Project Coordinator	Michael Karner | VIF | michael.karner@v2c2.at
JU Grant Agreement Number	737422

SCOTT has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737422. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, Norway.

CONTRIBUTORS

Name	Organization	Name	Organization
Silke Holtmanns	Nokia Bell Labs	Mateusz Mul	VEMCO
Frank van de Laar	Philips	Yoan Miche	Nokia Bell Labs
Pauli Räsänen	VTT	Ton Brouwer	Philips
Johanna Kallio	VTT	Jani Koivusaari	VTT

DOCUMENT HISTORY

Revision	Date	Author / Organization	Description
0.1	2019-08-20	Silke Holtmanns / Nokia	Initial draft version
0.2	2019-09-05	Silke Holtmanns, Yoan Miche / Nokia	First version
0.3	2019-09-15	Jani Koivusaari / VTT Pauli Räsänen / Nokia Mateusz Mul / Vemco	Input from VTT, VEMCO and Philips
0.4	2019-09-25	Several contributors	Final edits and minor corrections
1.0	2019-09-26	SCOTT Team	Conversion for publication

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

This document outlines how a networked sensor-based system that communicates with a cloud instance can be secured. The intention is to provide a methodology to go through a deployment phase to validate if the needed security measures are in place and no major security feature has been „forgotten“. We give references to commonly used guides for deployments, configuration or discussion documents where different pros and cons are outlined. The basic question this document tries to tackle is how to secure the communication of an IoT device with a cloud server with all the „hops“ in between.

We outline the basic security needs at each point in the system and then at the ISO/OSI layers of the system as a whole. This document can be used as a quick checklist by developers to validate that security methods against the most common security weaknesses (e.g. https://cwe.mitre.org/) are deployed and used. It does not replace a full in-depth security analysis or audit of a specific system. For that purpose, we give many references at the end of this document, where the details can be found. Those details will not be repeated here.

We start with the edge level, that covers the IoT device itself and the “first hop” of the communication. The next step is usually some form of gateway “to the world”. Through the gateway then the communication channel extends commonly to some form of cloud infrastructure. We close with guidance for the lifecycle management for security, as this covers all layers and nodes. Below a typical setting for a cloud connected IoT device:

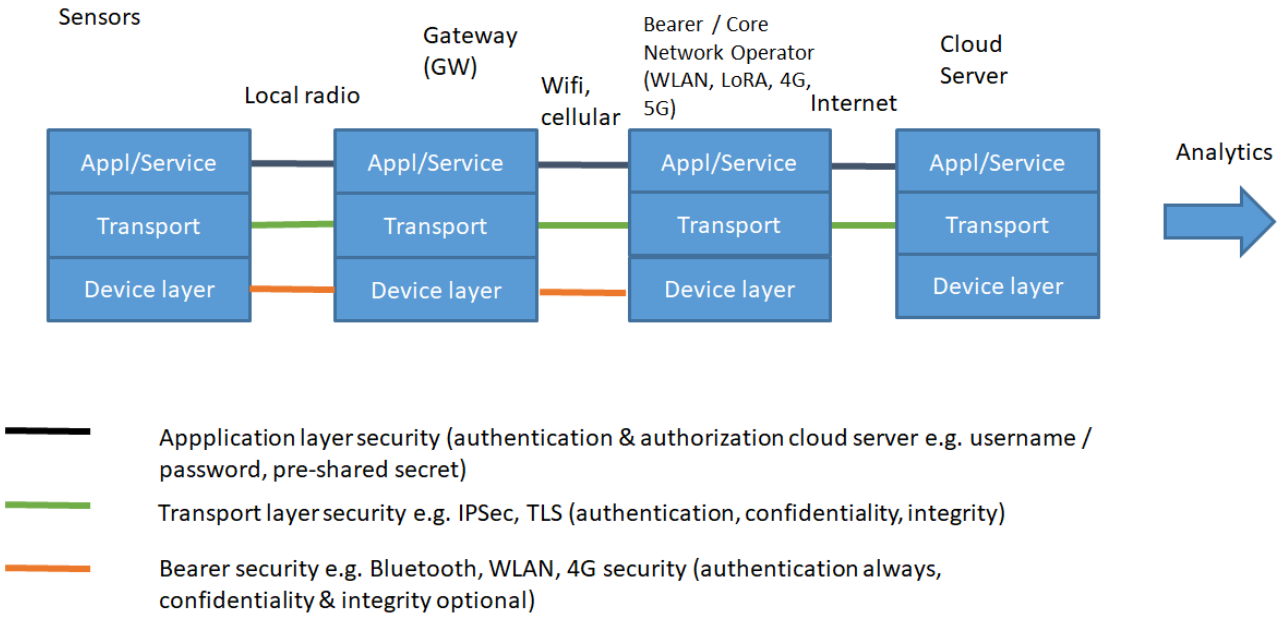

Figure 1 Connected IoT device with layers

2 EDGE LEVEL

This edge level chapter is about the end-device itself i.e. the IoT device and its communication with the next “hop” usually a gateway of some sort. The gateway might be a local server on a site or home or can also be in a communication operator network. Man-in-the-middle, DoS, eavesdropping attacks to this layer reveal the whole communication and if it is not additionally encrypted on the upper layers it compromises the confidentiality of the data. For mobile operators usually, the communication over the air interface is confidentiality protected, except if prohibited by regulations. The attacks on the first hop usually require “physical presence” nearby the IoT device. Some devices e.g. Bluetooth have a large connectivity “bubble” within which the communication is in clear and a communication channel to the device can be established. The radius of “nearby” depends on the bearer technology used.

2.1 Device Layer (Device – Gateway Communication) – Bearer Security

With the bearer security we include also link layer security. Different domains use the terms differently. The basic security measures that should be deployed and used on the first hop:

- Pairing of the IoT device with local gateway or trust center (e.g. out of band or passkeys) provides authentication
    - This pairing should be a one-time activity for the set-up of the system
    - Deploy well tested state of the art security algorithms e.g. based on Diffie Hellman (DH) key exchange/negotiation when possible, avoid home-grown protocols or protocols / code from not well-known sources
    - If possible use DH with out of band channel for authentication
    - Take into account recovery aspects e.g.
        - lost passkey (incl. change of operating personal) e.g. physical factory reset button
        - mandatory change of default passkeys
        - running out of power
        - compromised (cryptographic or other) algorithm (i.e. consider secure patching procedures or fall-back algorithm)
        - secure management ports (i.e. that management communications are authenticated; it is a very common mistake in IoT environment, that management ports are open and not properly secured. Do not hard code the same management password for all devices, it will land in the Internet!
- Usage of bearer and network layer security for the communication channel:
    - Avoiding older mobile bearer technology (e.g. 2G) as much as possible, i.e. preference settings in device to 3G and 4G, but availability vs security has to be taken into account based on application usage scenario. Consider also the location of your device, is it close to “sensitive” areas. For some usage scenarios coverage and availability has higher priority than security. For cellular technology as a rule of thumb each new generation has better security than the previous e.g. 5G has very good location tracking privacy protection
    - Authentication and lower layer security should be complemented by hardware roots of trust (at device and gateway) for key storage, software integrity verification. This again depends on use case and costs vs security challenges

- Bluetooth security, Bluetooth Low Energy (BLE) security, WiFi security, Zigbee, Narrowband IoT (NB-IoT), Long Range (LoRa) security gives confidentiality and integrity and is often part of the pairing process (but not always! See detailed discussion later on). Again, older protocols tend to be weaker than newer ones e.g. UWB Ultra-Wide Band does not have inbuild security, but due to technical construct it is hard to eavesdrop.
    - Some bearer security flawed to some extent, even in modern protocols (see detailed list of best practices below)
    - Some regions/countries restrict the usage of secure protocols on bearer level
    - BT Safe and Sound Protocol (SSP) is not secure, nor legacy modes for Bluetooth versions below v2.1
    - Use protocol-specific best practices and up to date versions
        - OWASP Security Knowledge Framework for developers https://www.owasp.org/index.php/OWASP_Security_Knowledge_Framework
        - BT Security Recommendations by the US NIST, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf
        - BLE Security Overview and Cryptographic Options: https://www.sigmadzn.com/wp-content/uploads/2017/01/BLESECURITYcase-studySD.pdf
        - Zigbee Security Recommendations https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/Securing_ZigBee_Wireless_Networks.pdf
        - UWB no security just plain radio, https://www.etsi.org/technologies/radio/ultra-wide-band?jjj=1551807759389
        - Narrowband IoT Security Overview (sort of ongoing research), https://accent-systems.com/blog/security-of-nb-iot-devices/
        - LoRa Security, https://lora-alliance.org/sites/default/files/2018-04/lora_alliance_security_whitepaper.pdf

Bearer Security is just one element in a security hierarchy. As some bearers do not offer security as integral part of the protocol and also in some areas of the world bearer security is not desired, it is recommended to have in addition application layer security (see further down). If no bearer security is deployed or it cannot be used, be aware that someone might send a message to your device with „anything“ in it looking like it potentially coming from a valid source. Make sure you have authentication and authorization on the upper layers, as the security associations and the corresponding endpoints on one layer are different from another layer .

If passphrases are used, then it is recommended to make them sufficiently long and strong. Changing passphrases on a regular basis is no longer recommended, but a changing mechanism should be there in case of issues e.g. backend server compromised, and passwords landed in the Internet. If default passwords are used for shipping, then at the first set-up change of them should be required. Also, for human provided passwords long passphrases should be preferred over complex error-prone shorter passwords. A summary of the new NIST recommendations can be found at: https://www.sans.org/security-awareness-training/blog/nist-has-spoken-death-complexity-long-live-passphrase

Most network compromises start with bad, too easy, too short passwords. See that your device allows sufficiently long passwords.

2.2 Transport/Internet Layer (Device – Gateway/Cloud)

To establish that the privacy of the data is really ensured during communication, it is recommended not to rely on bearer security only. As bearer security does not protect the whole path between device and cloud, but only a part of it i.e. device – gateway or base station depending on the protocol used. In the case of mobile communication, the data is only confidentiality protected into the mobile network, not further (where exactly depends on mobile technology generation used).

Therefore, we now give a summary of well-known and reliable protocols for securing the whole path. An attacker may get hold of the communication or a copy thereof. For transport and internet layer attacks the attacker does not need to be physically close to the target, address spoofing or Border Gateway Attacks may allow the attacker to intercept the traffic and redirect it (or a copy thereof) to a server of her choice.

- Transport/IP Layer Security Protocols:
  - IPSec
  - Wireguard/OpenVPN
  - TLS Security (validate that the mutual authentication variant is chosen)
    - TLS v1.3 https://tools.ietf.org/html/rfc8446
    - TLS v1.2 https://tools.ietf.org/html/rfc5246
- Not recommended:
  - SSL or deprecated standard e.g. TLS 1.0, TLS 1.1 are also no longer recommended.
  - Support downgrade mechanism (common attack vector)
- General recommendations and considerations:
  - Allow secondary algorithms as back-up for the future
  - Use public key pinning for further certificate verification, when possible
- Which protocol to take?
  - IPSec is a well-established standard, which provides IP layer and above security, but adds some overhead and is more challenging to deploy than TLS

    https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-77.pdf
  - For web-based applications commonly TLS is the chosen
  - Wireguard is faster than OpenVPN (no protocol negotiation, no risk of downgrade attacks)
  - Wireguard still under testing for IoT suitability (and general reliability)

    Discussion on the topic can be found at: https://restoreprivacy.com/wireguard/
- The endpoint might be the Gateway or the cloud server, depending on the set-up

2.3 Service Layer (Device – Server)

This communication would be on service or application layer. On this layer we have the normal “service” communication i.e. the normal interaction for the purpose of the application like health information, temperature sensor information, industrial IoT information etc. and we have the “management” communication for configuring the device, applying patches, reboots etc.

We assume that most IoT devices will be managed remotely to avoid requiring an IT expert in each household. We give some guidance and references for securing that part of the communication:

- Remote management interface of devices:
  - Should have access to minimum data from devices for correct operation
  - Need to follow clear and separated access rights for different authorization levels (aka don't run all commands with full privileges as root/admin) The device should have clear authorization control: who is allowed to access, see, and modify what
  - This should be role based e.g. admin, manufacturer, user1, user2 etc. A comprehensive list on how to avoid turning the IoT device into a botnet slave can be found at: https://security.berkeley.edu/resources/best-practices-how-articles/system-application-security/securing-remote-desktop-rdp-system
- Follow OWASP recommendations and OWASP penetration tested devices when possible (it is actually fun and recommended to all security interested persons). https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
  - Password/Credentials based, key based, combination of both
  - Clear separation of rights per device, group, …
  - SANS Role Based Access guidelines: https://www.sans.edu/cyber-research/security-laboratory/article/311

2.4 Application Layer (Device – Cloud)

The application layer is really the uppermost layer of the stack. It is different for each application and use case. Usually, the communication endpoints are device and cloud and real end to end security should be done i.e. authentication, authorization, integrity and confidentiality. As it is often application dependent only quite generic recommendations can be given:

- Commonly, a passphrase based password at set-up (no hard-coded passwords for thousands of devices, which do not have to be changed during set-up)
- If web and browser based (most common scenario), validate that HTTPS is used with mutual authentication and correct certificates are stored and no other allowed. Consider certificate revocation and renewal aspects! Consider also cases of breached certificate, root CA company being bought etc. See remote device management guidelines and lifecycle aspects at the end of this document. https://www.owasp.org/index.php/SSL_Best_Practices

Cloud providers offer as-is contracts and security. It is recommended to check the reputation of a cloud providers for incidents and the compensation for data breaches in the contract. Sometimes also the geolocation of a cloud provider should be taken into account, as there are different legal settings for different countries and some applications might be very sensitive.

2.5 Edge Device Itself

The IoT device itself is from security point of view a very crucial element. On the other hand, it is understood, that usually those devices have higher priorities than security i.e. focus more on costs and time to market.

- Heavily depends on the device itself, but the following consideration should be made for OS, hardware and software:
  - Custom OS/ Major OS alterations to be avoided on core OS and security practices if possible

  - Best using Long Term Support (LTS) types of OS, with regular, scheduled updates for security patches. Consider in particular in relation to lifetime of your edge device
  - Possibility of "compartmentalizing" updates (hardware and SW) using approaches such as Project Treble (Android), Snap (Ubuntu) or similar
  - Secure Boot and hardware root of trust for updates, device attestation: avoids attacks on USB boot stick, device impersonation etc.
  - The GSMA (world largest operator association) has IoT security guidelines https://www.gsma.com/iot/iot-security-guidelines-for-endpoint-ecosystem/
  - CTIA has a device evaluation scheme, which can be used to check the security quality of an IoT device: https://api.ctia.org/wp-content/uploads/2018/10/ctia_IoT_cybersecurity_pmd_ver-1_0.pdf
- The traffic of small devices should be screened by the gateway, note that then the Gateway needs full access to data hop-by-hop security i.e. TLS/IPSec tunnels should end at Gateway (which makes Gateway security all the more important)
- Powerful enough devices can have their own antivirus, DoS protection, firewall, etc.
- For small devices the communication provider may offer some alarm function based on traffic patterns to avoid spreading of malware and viruses. This should be clarified in the Service Level Agreement with the communication platform provider.
- Location tracking (for use case needs):
  - Avoid location data collection at cloud/server level if possible. No "just in case, or might be useful later" collection
  - Prefer decision mechanism in edge device, with needed location and radius pushed by server/cloud
  - Constant device tracking also a security risk (besides draining the battery). Data interception at bearer, network, cloud level could access data.
- Secure enclave or hardware security module type memory (compartmentalized logical or physical from main device memory) for certificates and keys storage. Here is a common case of security breach, if there are no role-based system (RBS) and all data is accessible as admin, then the hacker can just read the keys.
  - Creation on device of initial keys
  - Secure storage of shared certificates/keys
  - Some updates possibilities for key lifecycle management (depends on solution used)
- Consideration to compromise of keys or certificates should be given, "what if" e.g. trusted third party is no longer trustworthy.
  - If third party is used, then a qualified party should be used
  - Being your own Certificate Authority is not always the right solution.
  - Authenticated means of certificate/keys revocation and rebuilding should be considered.

3 GATEWAY LEVEL

The gateway level includes also the radio communication network. A gateway has two communication sides. The communication to the device has been discussed in the previous chapter. This chapter focuses on the gateway “upwards” communication and the gateway itself (see Figure 1 and Figure 2).

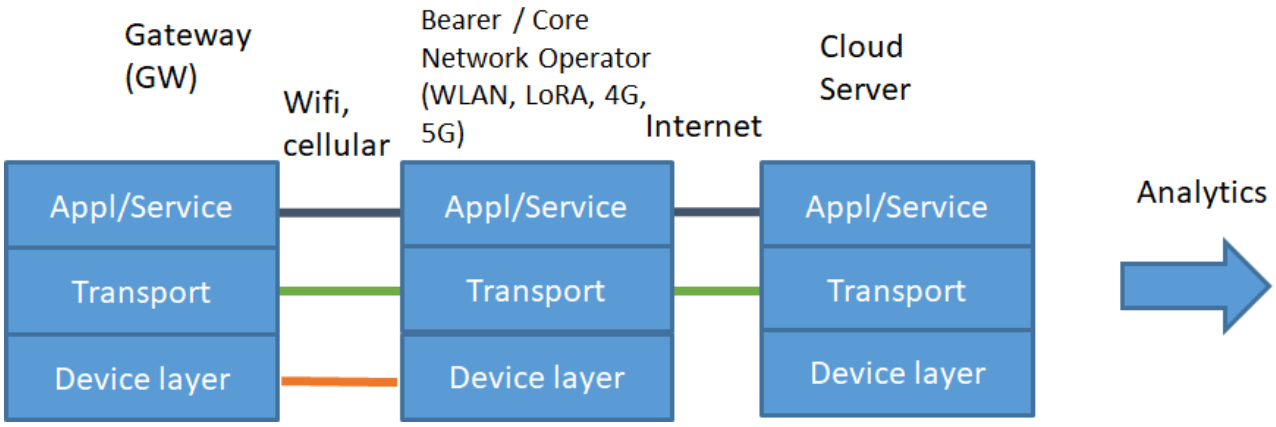

Figure 2 GW communications

3.1 Device Layer (GW-Operator)

This should be a selection criterion for contract establishment with the operator:

- See that authentication, confidentiality and integrity is enabled
- Disallow 2G
- WiFi is used use newer protocol versions and not outdated like WPA (even if still available in some places)and if certified devices (e.g. WFA) have at least been tested to some degree
- Choose operator with a good track record of security e.g. potential certifications, validate Service Level Agreement contracts for security breach compensations

3.2 Transport Layer (GW – Device and GW - Cloud)

Transport security protocol were discussed extensively in chapter 1)

- IPSec or TLS, Wireguard (same considerations as for Edge/device level in chapter 2)

3.3 Service Layer (GW – Server)

The remote management interfaces should follow the same rules as before. The gateway itself should be chosen based on the availability of good security functionality and also on the past security reputation of the vendor:

- Remote management interface of gateway
- Follow OWASP recommendations and OWASP penetration tested devices
- Authorization, who is allowed to access, see, and modify
- See previous consideration for service layer

3.4 Application Layer (GW – Cloud)

This section only applies if the gateway breaks the device to cloud tunnel. That the gateway breaks the tunnel means, that the security channel i.e. authentication, confidentiality and integrity run only till the gateway and then there is cleartext in the gateway and the gateway “packs it” for the next hop. Here some suggestions for protection. If it is unclear whether the security channel is interrupted by the gateway, assume it is and take precautions:

- Commonly, passphrase based set-up (no hard-coded password for thousands of devices). Change of passphrase at set-up
- If web based, validate that HTTPS is used and correct certificates are stored and no other allowed.
- See previous recommendations for application layer.

3.5 Gateway Itself

The gateway itself needs also sufficient protection:

- DoS, antivirus protection, firewall, intrusion detection system
- https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1510001675.pdf
- Consider also mechanisms to handle a rogue gateway e.g. botnet slave: can it be decommissioned without shutting the whole system down? Can it be rolled back to a version without the “malware”?

4 CLOUD SERVER LEVEL

4.1 Device Layer (Cloud – GW)

The hardware of the cloud service provider is usually a “as-is” part of the contract. In case of location sensitive applications e.g. governmental, the location should be part of the service level agreement

- The security should be provided by Cloud provider or be taken care of already
- Check the small print on liabilities of the cloud provider in case of data breach e.g. if EU law is applied

4.2 Transport Layer (Cloud – GW/Device)

This is again standard IP level security and can be done with normal security protocols. It is important to remember to require mutual authentication

- IPSec or TLS
- The endpoint might be the GW or the device, depending on the set-up
- See previous considerations for transport layers security

4.3 Service Layer (Cloud – Server)

This communication happens above the transport layer but is sometimes intertwined with it e.g. through the use of a browser interface and usage of TLS. The service layer also includes management activities e.g. for updates or reset activities.

- Remote management interface of devices
- Follow OWASP recommendations and OWASP penetration tested devices
- Authorization, who is allowed to access, see, and modify
- See previous considerations for service layer

4.4 Application Layer (Cloud-Device)

If a reasonable commercial cloud provider is used then it can be assumed, that this cloud provider provides the lower layer security towards the device. But as an application running on the cloud, the major questions are

- How is the access to the cloud for your data secured?
    - Who is authorized to perform which actions on your cloud API?
    - How is the authorization and authentication done?
    - The data inside the cloud, do you have a once-in can access all or do you need to have different “roles”
    - If HTTP web based, validate that HTTPS is used and correct certificates are stored and no other allowed
    - Do you use passphrases? If so, don’t use one passphrase for all devices (hard coded password, it will leak sooner or later)
- What do you do, when things go wrong (and they do, sooner or later)?

    - Do you need to update credentials at the devices for accessing the cloud?
    - If the access credentials leak, do you have a plan B? Is the plan B safe (i.e. doesn't open up a new vulnerability)?
        - Do you have a secure password recovery process?
        - Failed login counters (with potential delays or cut-off)
- Feature/version roll-out:
    - Extensive testing on prototype/restricted version of system
    - Deployment with roll-back possibility (in case update breaks functionality), and log file upload for fault analysis.

4.5 Cloud Itself: IDS, DoS and Antivirus protection

The Cloud security should be the duty of the cloud level provider and integral part of the service level agreement.

- Secure hypervisor
    - https://www.techadvisory.org/2018/03/why-hypervisor-security-is-important/ and
    - https://searchcloudsecurity.techtarget.com/definition/hypervisor-security
- Trusted Platform Module (TPM)

  TPM 2.0 https://trustedcomputinggroup.org/resource/tpm-library-specification/
- Slice separation
- Service Level Agreement should have compensation for data breach
- Service Level Agreement should be based on EU law

5 SECURITY LIFECYCLE

The devices will be deployed and used, but the world will move on. To avoid attacks through old or phased out hard and software or vulnerable algorithms it is important to think about the handling of such situations from the beginning. Here some items to take into account:

- Key Management Requirements (how to handle keys)
  - https://www.owasp.org/index.php/Key_Management_Cheat_Sheet (chapter 4)
- Cryptographic Key Compromise (what if your keys are hacked or got lost)
  - https://www.owasp.org/index.php/Key_Management_Cheat_Sheet (chapter 4)
- Testing of new roll-outs before they go live e.g. using
  - https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
  - https://www.owasp.org/index.php/OWASP_Testing_Project

You may want to consider a “calling-home” button i.e. a hidden factory reset button, that could be used as last-resort for cases of credential compromise and full re-installation: It would allow for special account access, using special credentials / certificates / keys that are otherwise never used. For example, stored in a Hardware Security Module (HSM) silently until the reset button is pressed.

A. ABBREVIATIONS AND DEFINITIONS

Term	Definition
API	Application Programming Interface
BLE	Bluetooth Low Energy
BT	Bluetooth
DH	Diffie Hellman
DoS	Denial of Service
EU	European Union
GW	Gateway
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
LoRa	Long Range
LTS	Long Term Support
NB-IoT	Narrowband IoT
TLS	Transport Layer Security
OoB	Out of Band
OS	Operating System
OWASP	Open Web Application Security Project
RBS	Role Based Access System
SLA	Service Level Agreement
TPM	Trusted Platform Module
UWB	Ultra Wide Band
VPN	Virtual Private Networks
WiFi	Wireless Fidelity