Session 10: Planning for IoT Networks:

Privacy and Security Aspects

25-28 September 2018
Bandung, Indonesia

Dr. Nizar Ben Neji
ITU Expert
University of Carthage
nizar.benneji@fsb.rnu.tn

Content

1. **Security Aspects** (organized by objectives)
2. **Lightweight and context-aware security protocols and solutions**
3. **Case study on the security of smart metering systems**

1

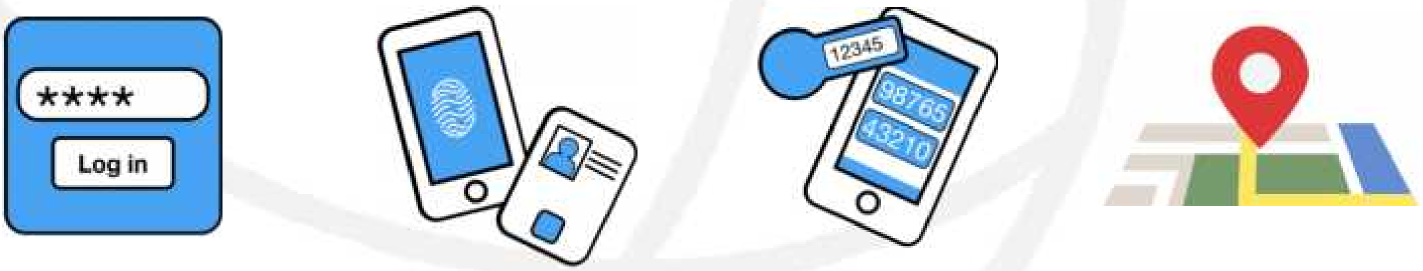
Security Aspects

1. Authentication (Multifactor authentication, AAA, ...)
2. Confidentiality and privacy (Encryption, Anonymization, ...)
3. Data integrity over its entire life-cycle (Hashing, Digital signature, ...)
4. Non-repudiation of creating, approving, sending and receiving documents
5. High availability (Data replication, Node duplication, Failover, Load balancing, ...)
6. Traceability and history of electronic acts and actors
7. Privacy and protection of personal data
8. Building trust (Trust third parties, Distributed trust, …)

Authentication

Multifactor Authentication (MFA)

FACTOR	TYPE	EXAMPLE
Something the entity know	Knowledge factor	Password, PIN code, secret response, …
Something the entity has	Possession factor	Smartcard, Access badge, OTP Token, SIM card, …
Something the entity is	Biometric factor	Fingerprint, Iris print, DNA, …
Something the entity do		Handwritten signature, keyboard behavior, voice recognition, face recognition, …
Where the entity is	Location factor	IP address, geographic location, …



Authentication

Authentication, Authorization and Accounting (AAA)

- **Authentication** is the verification of the identity of the entity (device, user or software) trying to access the system. Authentication is based on trust since we need to first authenticate and trust the **issuers of IDs**.
- **Authorization** or access control is the verification of the resources or actions (read, write and execute) that the entity is permitted to access or to do. It includes denying or revoking access for someone or something malicious.
- **Accounting** represents the statistics of resources usage by identity. It is used to track the user's usage for charging and for auditing purposes.

Authentication

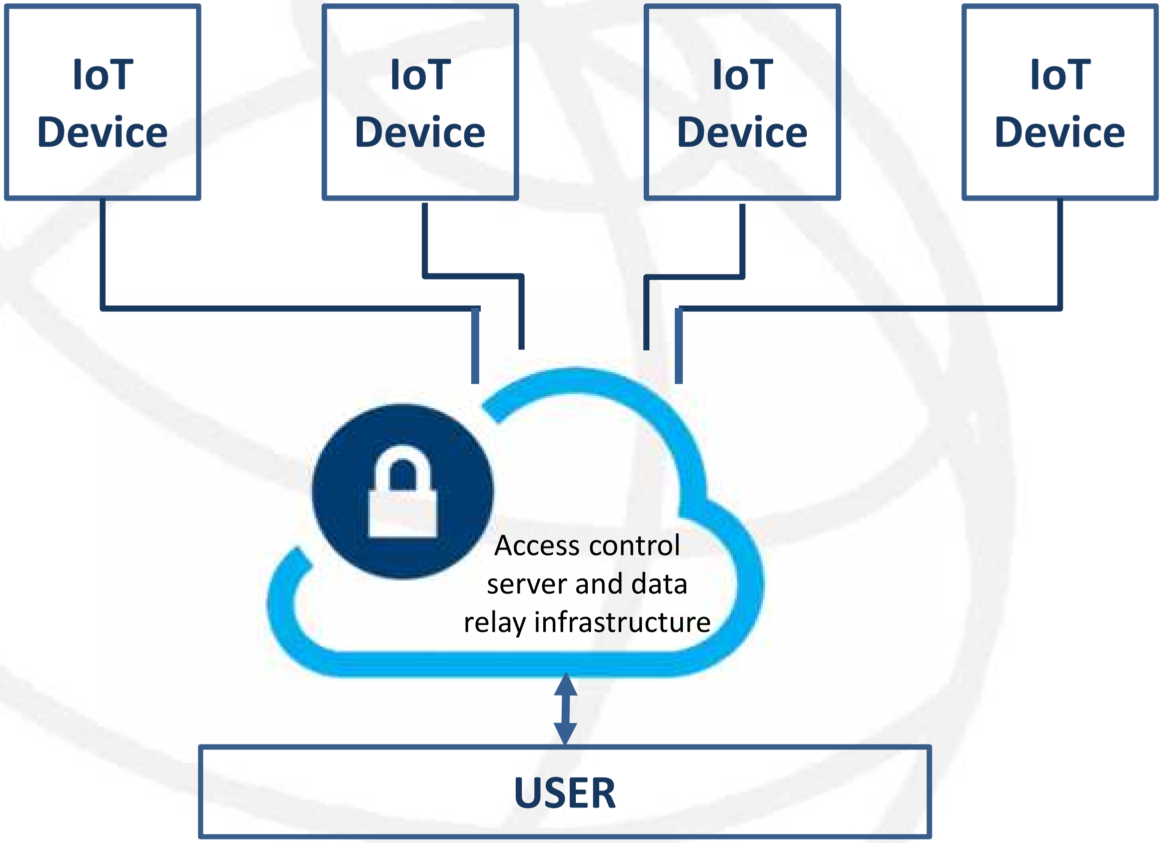
Access Control

- In the IoT, access control is needed to make sure that only **trusted parties** can update device software, access sensor data or command the actuators to perform an operation
- **Data ownership** and sharing IoT data selectively is guaranteed using access control mechanisms
- In IoT, two possible access control architectures:
  - Centralized Architecture
  - Distributed Architecture
- Standard authorisation model could be adopted:
  - Access Control List (ACL)
  - Role Based Access Control (RBAC)
  - Attribute Based Access Control (ABAC)

Authentication

Centralized cloud-based access control

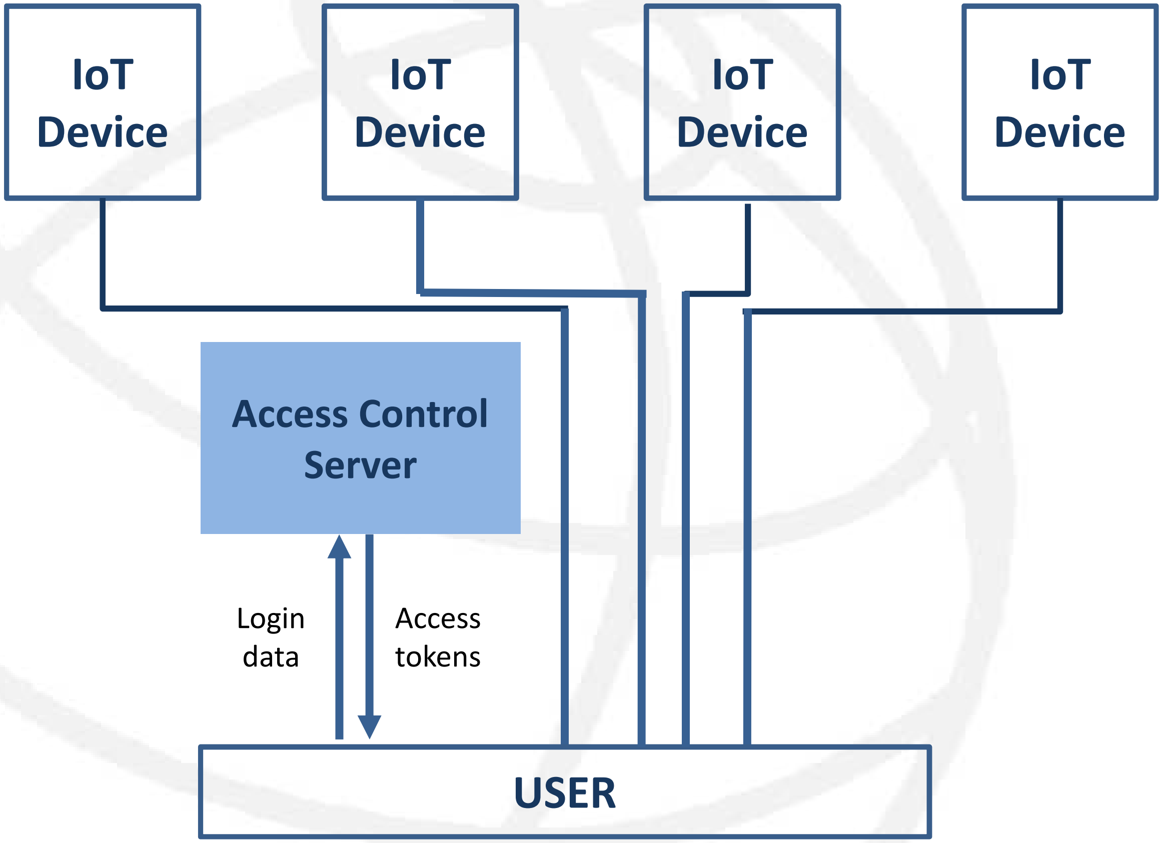
In a centralized architecture, the user accesses only cloud-based servers that authorize the request and relay data between the user and the IoT devices.



Authentication

Distributed access control

In a distributed architecture, an access control server grants access tokens to users, who use them to access the IoT devices directly.



Authentication

TLS Authentication

- Transport Layer Security widely used protocol providing channel security guarantees for several Internet protocols:
  - **Web:** HTTP**(S)**
  - **Messaging**: SMTP**(S)**, POP**(S)**, IMAP**(S)**
  - **LDAP Directory**: LDAP**(S)**
  - **VPN SSL**
- TLS uses public key cryptography for channel establishment and digital certificates to authenticate the communicating entities
- Client/Server authentication
  - Simple authentication
  - Mutual authentication
- SSLv2 and SSLv3 are the obsolete versions and TLS 1.0, TLS1.1, TLS1.2 and TLS1.3 are the actual used versions. **TLS1.3 is the IoT oriented version of the protocol**

Confidentiality

Concept

- Confidentiality is the property whereby information is not disclosed to unauthorized entities
- Guarantee confidentiality of data in use, in motion and at rest
- Techniques used to ensure confidentiality of data:
  - Data encryption
    - Symmetric encryption
    - Asymmetric encryption
    - Hybrid encryption
  - Data Anonymization
  - One way function or hashing
  - …

Integrity

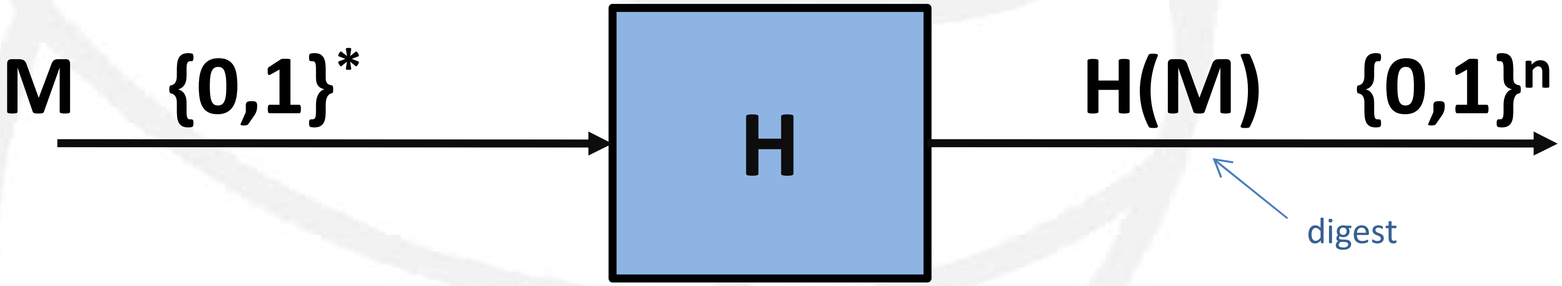
Concept

- **Integrity** means guaranteeing that data has not been altered since it was created, transmitted or stored.
- Data integrity is very important for IoT systems as the accurate collection of information by sensors is required for the IoT system to function correctly. The system should be able to detect any malicious modification,
- Data can be modified **intentionally** or **accidentally**
- Cryptographic **checksums**, MAC or hashes are used to verify integrity
- **Digital signature** is a proof of integrity since the hash is protected by the sender's private key

Integrity

Hash functions

- A **hash function** is a mathematical transformation that can be used to map data of **arbitrary size** to data of **fixed size** named **digest** or **hash value**.
- Hash functions have four main properties:
  - It is infeasible to generate a message from its hash
  - it is easy and fast to compute the hash value for any given message
  - it is infeasible to modify a message without changing the hash
  - it is infeasible to find two different messages with the same hash (collision resistant)



Integrity

Hash functions

Name	Designed by	Size of digest
MD5 (Message Digest 5)	Ronald Rivest (1991)	128 bits
Since 2004, MD5 is no more recommended as a reliable hash function in cryptography		
SHA (Secure Hash Algorithm)	Designed by NSA (National Security Agency) (SHA-1 in 1994 and SHA-2 in 2000) Later standardized by NIST (National Institute of Standard Technology)	SHA-1 160 bits Since 2011 is no more used
		SHA-2 224, 256, 384, 512 bits

Non-repudiation

Principle

- Mechanism that can prevent a corresponding entity from **denying** its involvement in an electronic transaction and it can be seen as a subcomponent of authentication
- Non-repudiation of creating, approving, sending and receiving
- It is important in terms of tracking illegal activities on the Internet, as it allows for accountability to be enforced
- **Digital signature** is the only mechanism ensuring non-repudiation and the **uniqueness** of the **digital signing key** (generated in a crypto smart card) is important to hold the signer (crypto card holder) accountable for doing an electronic act

High Availability

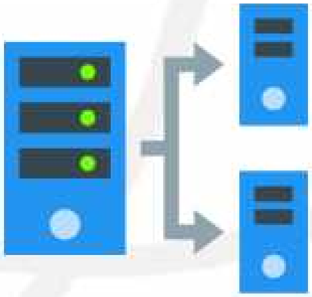
Concept

- High-availability means that a system needs to be accessible, operational and usable 24/7 or just upon demand by an authorized entity and under all operating conditions
- Constrained nature of the IoT devices make availability difficult to achieve essentially due to
  - Mobility
  - Energy limitation
  - Limited connectivity (bandwidth, range, …)
- Requirement for availability varies between different use cases
- Availability can be achieved through
  - Implementing energy efficient protocols and mechanisms
  - Integrating energy harvesting and saving mechanisms
  - Implementing DoS and DDoS countermeasures
  - Avoiding by design the single points of failure like using duplication

High Availability

High availability mechanisms

- Classical mechanisms used to ensure high-availability are still valid in an IoT environment (in the cloud side):
  - Fail over technique
  - Load balancing
  - Clustering
  - Duplicating data and systems
  - Automatic and periodic backups
  - Distant data centres
  - Disaster recovery plan
  - …



Traceability

Principle

- **Traceability** means the ability to trace and identify all stages and events that led to a particular point in a system or process
- Traceability is useful for:
  - Real time device tracking, tracing
  - Remote monitoring
  - Forensics and digital investigation in collecting proofs and digital evidences
  - Intrusion Detection Systems to detect attacks as early as possible based on access, event and error logs

Privacy

Data and access privacy

- Ability to preclude personal data from being shared or communicated to non authorized entities
- Ability to know what are the sensitive changes performed on your personal data over time
- Ability to hide access to certain resources, data or systems
- Ability to completely remove personal data from the digital world
- Privacy policy is a public document specifying how consumer's data are gathered, used, disclosed, managed and deleted:
  - Client's data (name, address, date of birth, marital status, contact information, level of study, financial records, medical history, …)
  - Period of retention of data
  - Purpose of use and retention
  - Whether data is kept confidential, shared with partners, or sold to other firms

Privacy

Data Anonymization

- Data Anonymization is an information sanitization whose intent is privacy protection, it consists of removing Personally Identifiable Information (PII) from data sets so that the people whom the data describe remain anonymous
- Data Anonymization means
  - Removal of identifiers (SSN, Passport number, …)
  - Removal of the combination of quasi-identifiers (Age, Zip Code, Sex, …)
- Data Anonymization facilitates the exploitation of micro data (aggregated macro data) without compromising the privacy of the users
- Data anonymity reduces considerably the quality of data

Privacy

Personal data

- Personal data refers to data, whether true or not, about an individual who can be identified from that data
- Digital and paper format
- Stored, archived, processed or transferred
- Automated and non automated processing
- Classification of personal data:
  - Identifiers (SSN, Passport number, Credit card number, …)
  - Quasi-identifiers (Age, Sex, Zip Code, …)
  - Sensitive data (Illness, financial asset, …)
  - General information (Feeling, …)

Privacy

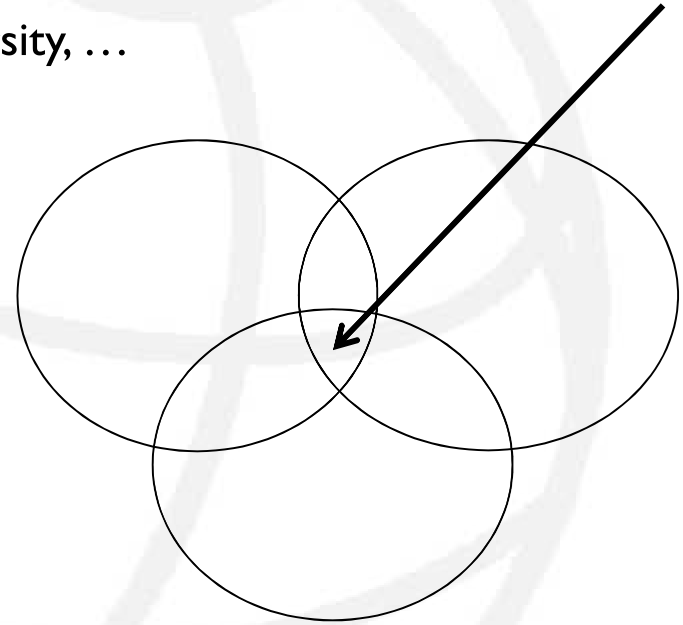
Personal Identifiable Information (PII)

- Identifiers are data used to identify and distinguish individuals according to the National Institute of Standards and Technology (NIST):
  - National identification number
  - Social security number
  - Passport number
  - Vehicle registration plate number
  - Driver's license number
  - Credit card numbers
  - Home address
  - Telephone number
  - Email address
  - IP address
  - Face, fingerprints, or handwriting
  - Digital identity
  - Genetic information
  - Login name, screen name, nickname, …
  - …

Privacy

Potentially PII

- Potentially PII or quasi-identifiers are shared by many people and if combined together they may identify an individual :
  - First Name
  - Last Name
  - Country, state, postcode or city of residence
  - Workplace, School, University, …
  - Age
  - Date of birth
  - Birthplace
  - Gender or race
  - Job position, Grades, …
  - Geographic location, …
  - …



multiple quasi-identifiers may uniquely identify an individual

Privacy

Sensitive data

- Personally Identifiable Information (ID Numbers, Contact Information, …) :
  - Healthcare records
  - Judicial or criminal record
  - Financial assets and transactions
  - Contents of the communications
  - Personal life and daily activities
  - Personal information that the individual prefer that it remains secret
  - Any information that poses a risk to a person or a company
  - …

Trust

Requirements

- Trust is a binary relationship
- IoT networks are mainly relying on sensor devices, **trusting data collected by sensors** is a serious security concern
- **Enforcing trust mechanisms** at all IoT levels (sensor, device, gateway and cloud) to guarantee the validity and the quality of the collected and transmitted information
- Certificate and signature mechanisms rely on **trusted third party** or in order to verify that communicating entities are who they are claiming to be
- Blockchain technology and **distributed trust**

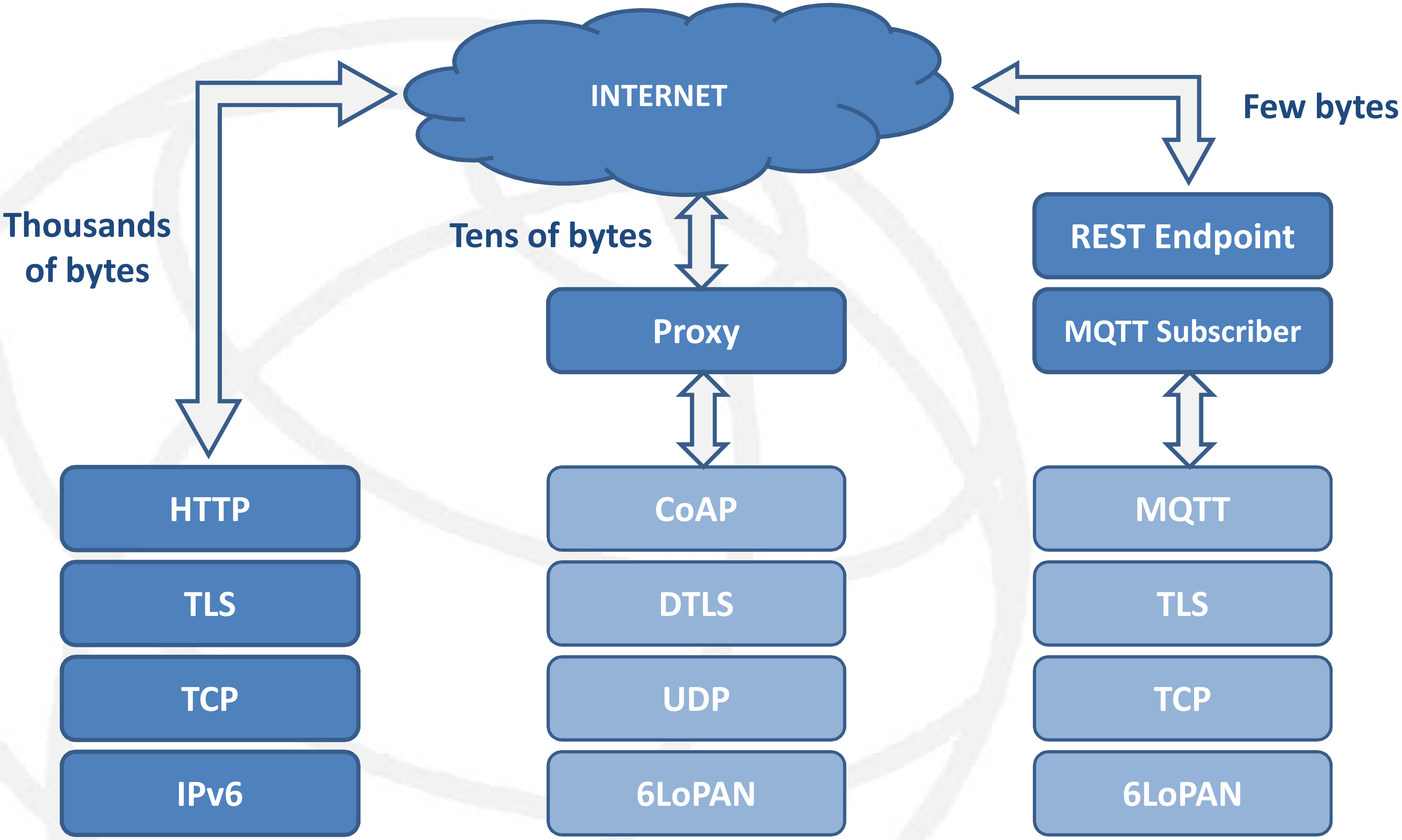
2

Lightweight and Context-Aware Security Protocols and Solutions

1. Network Protocol Comparison
2. Constrained Application Protocol
3. Datagram Transport Layer Security
4. MQ Telemetry Transport
5. Transport Layer Security
6. Lightweight Cryptographic Primitives
7. Public Key Infrastructure
8. Key Takeaways
9. Software Authenticity
10. Context-Aware security

Network Protocol Comparison

IoT Protocols



Constrained Application Protocol

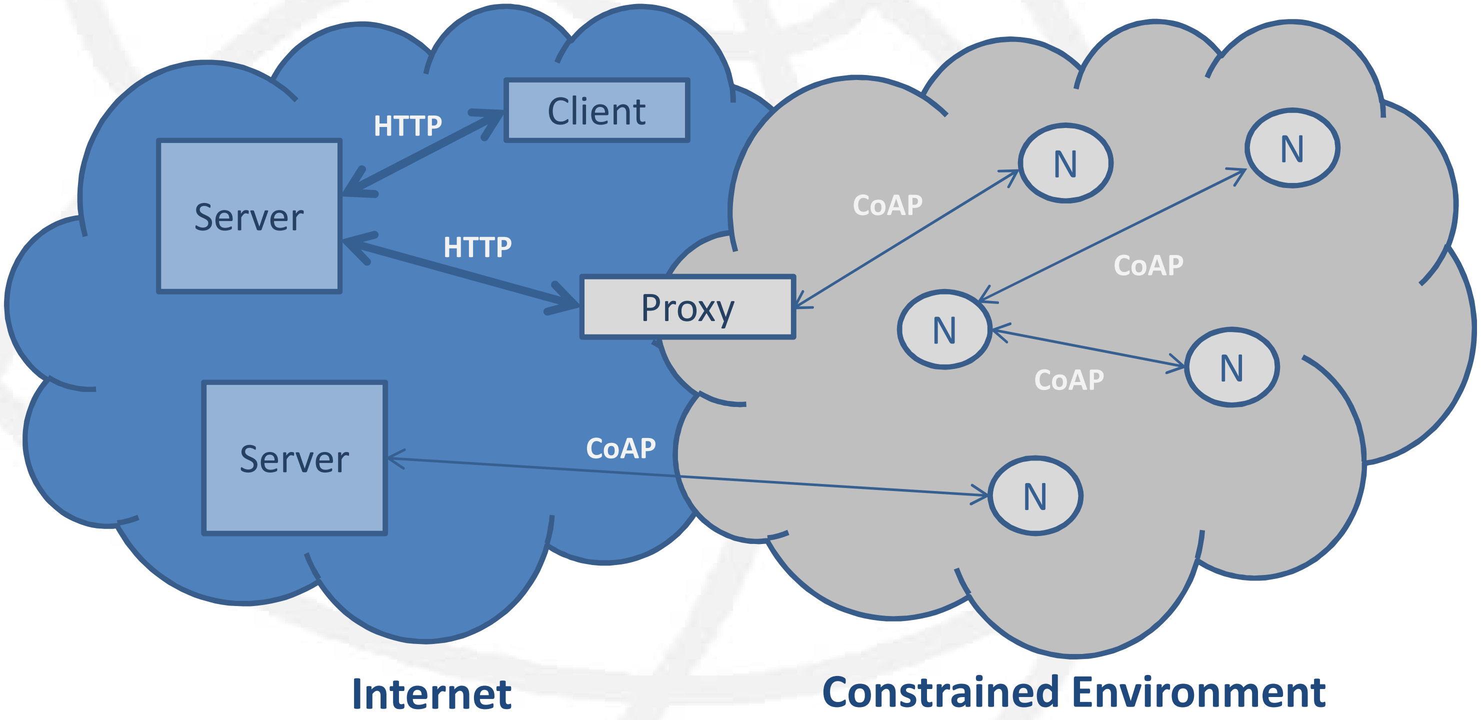
HTTP like protocol

- Constrained Application Protocol (CoAP) protocol offers a REST programming model easy to proxy to web applications
- CoAP is suitable for constrained IoT nodes, lossy networks and Machine to Machine (M2M) communications
- CoAP was designed to work on microcontrollers with as low as 10KB of RAM and 100 KB of code space
- Headers are smaller than HTTP, and the protocol supports splitting larger payloads through multiple requests known as a Blockwise transfer
- Suitable to push firmware updates to devices and to send and receive sensor updates
- CaAP is a decentralized protocol allowing sensors and nodes to link with and publish to each other

Constrained Application Protocol

Network communication

- CoAP is used to link directly IoT nodes or to link the nodes through servers and proxies along the way to perform caching, protocol translation and enforce security mechanisms such as authentication and access control



Constrained Application Protocol

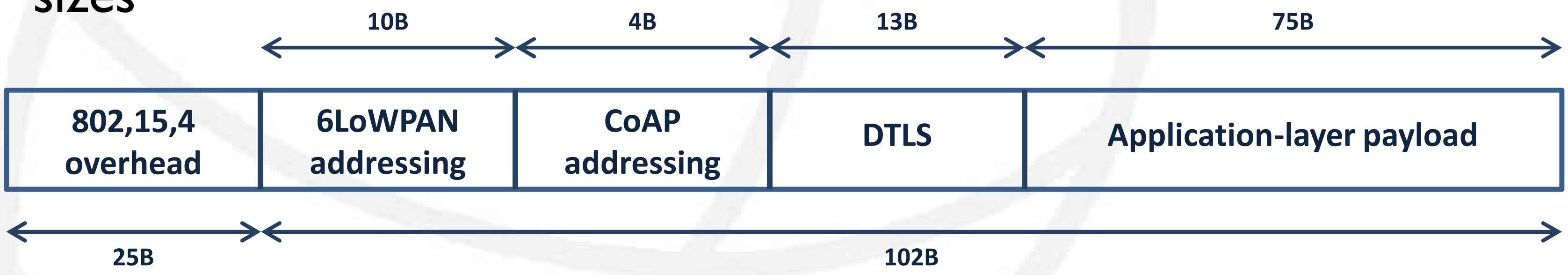
Extensions

- CoAP is a plaintext protocol
- Encryption with CoAP is accomplished using Datagram Transport Layer Security (DTLS) and occasionally with IPSec
- CoAP's default choice of DTLS parameters is equivalent to 3072-bit RSA keys
- Default port for coap:// is 5683/udp and for coaps:// is 5684/udp
- Several extensions to CoAP were developed:
  - Observers to allow a node to receive updates
  - Multicast group communications to allows for a single request to be transmitted to several nodes, in a one-to-many relationship.
  - Resource discovery
  - Blockwise transfers
  - CoAP aver TCP

Datagram Transport Layer Security

Concept

- DTLS protocol is a variant of the SSL/TLS family of protocols, designed to provide communications privacy for datagram protocols (UDP)
- It allows client/server applications to communicate in a way to prevent eavesdropping, tampering and message forgery
- It provides capabilities for certificate-based authentication, data encryption, and message integrity.
- DTLS includes timeout, retransmission and fragmentation mechanisms to solve the packet loss, ordering and the message sizes



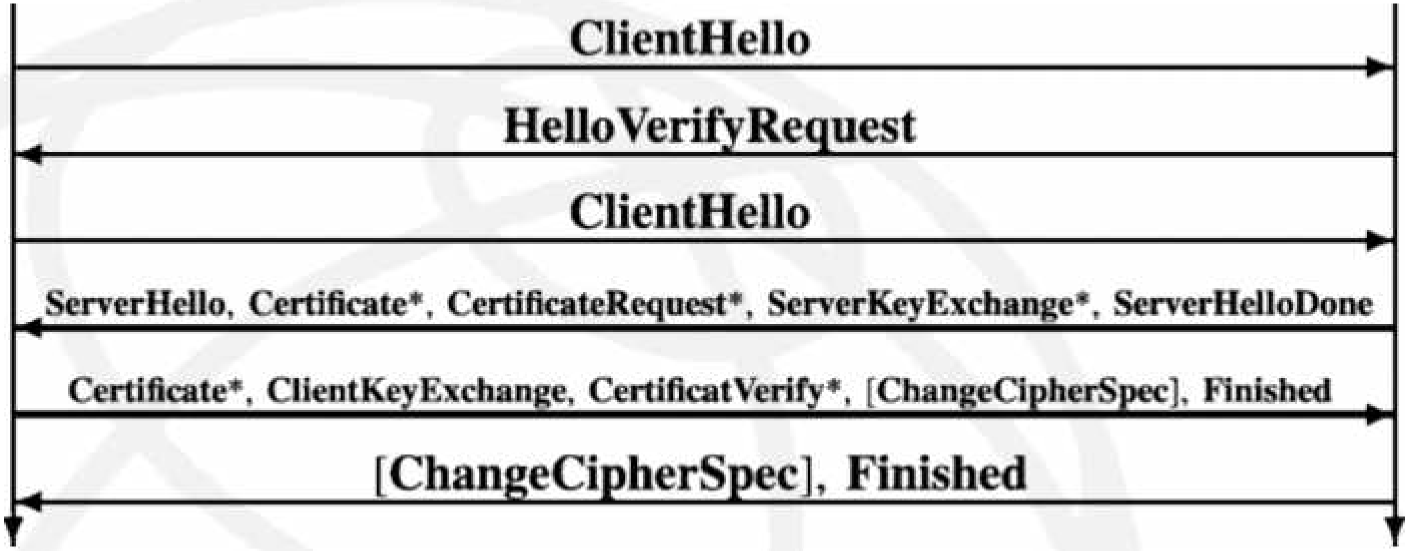
10B
4B
13B
75B

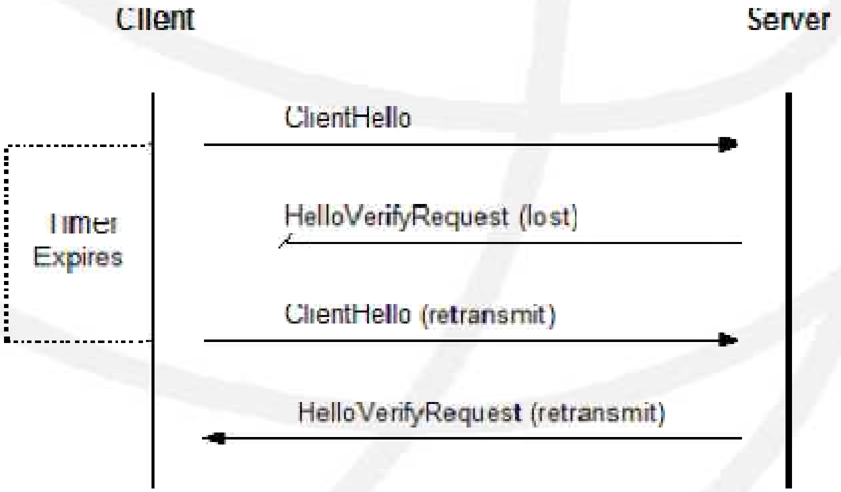
802,15,4 overhead
6LoWPAN addressing
CoAP addressing
DTLS
Application-layer payload

25B
102B

Datagram Transport Layer Security

DTLS Handshake

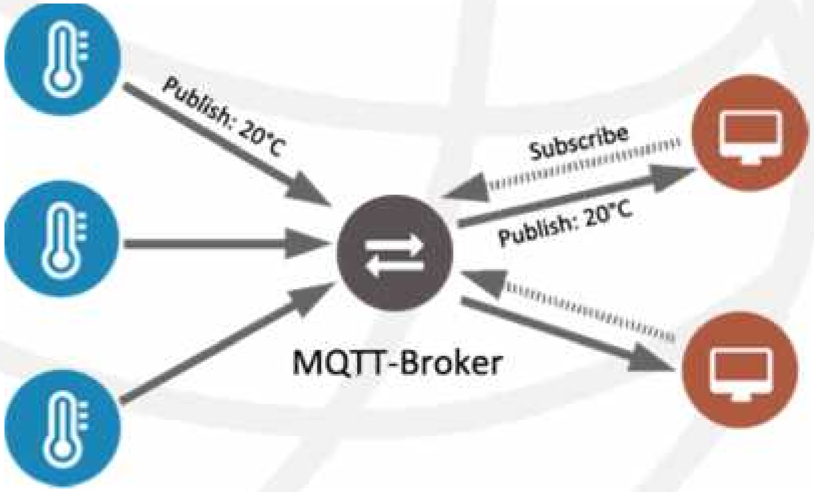

ClientHello
HelloVerifyRequest
ClientHello
ServerHello, Certificate*, CertificateRequest*, ServerKeyExchange*, ServerHelloDone
Certificate*, ClientKeyExchange, CertificatVerify*, [ChangeCipherSpec], Finished
[ChangeCipherSpec], Finished



MQ Telemetry Transport

M2M Iot Connectivity

- MQTT is a M2M IoT connectivity protocol used of the top of TCP/IP, ideal for constrained networks (low bandwidth, high latency, data limits and fragile connections)
- A client/server protocol using a publish/subscribe architecture in contrast to HTTP that is using request/response paradigm: each node that wants to receive messages subscribes to a certain topic and a broker delivers all messages with the matching topic to the node.



MQ Telemetry Transport

Support of security

- MQTT is suitable for node monitoring and detecting when an IoT device goes out of the network
- LWT (Last will and testament) feature is used when the client unexpectedly disconnects, so the keep alive timer at the server side detects that the client has not sent any message or the keep alive PINGREQ. Hence the server immediately publishes the Will message on the Will topic specified by the client.
- User names and passwords can be sent with MQTT but they need encryption with TLS
- MQTT uses plain TCP and encryption of the communication is accomplished using either TLS or VPN IPsec and this encryption adds a significant network overhead
- Default port for MQTT is 1883 and for MQTT over TLS is 8883

MQ Telemetry Transport

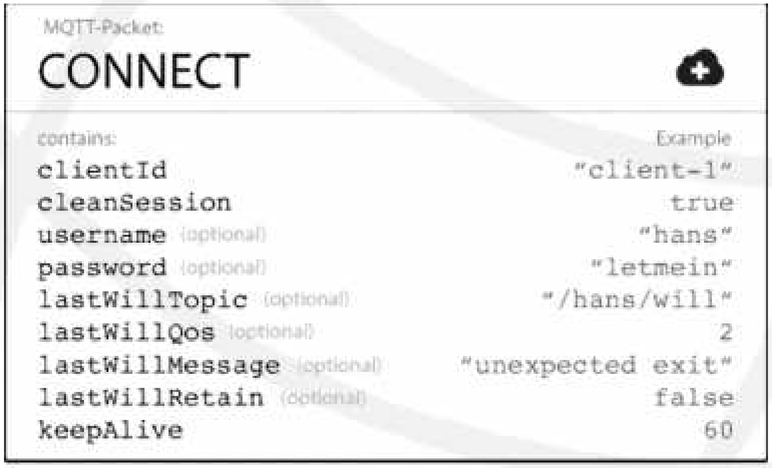
Secure MQTT deployment

- Every connection to a MQTT broker should at least pass one firewall to try block attackers at the firewall level and only expected traffic gets forwarded to downstream systems:
  - Block UDP datagram packets
  - Block ICMP
  - Authorize only traffics to only 1883 and 8883
  - Allow traffic by defined IP range (if possible)
- A DMZ zone where to place MQTT brokers with a second firewall from different vendor are recommended
- Load balancers are recommended to be used to distribute MQTT traffic to different MQTT brokers to prevent the overload
- Implementing bandwidth and message size restrictions to prevent malicious nodes from sending huge messages

MQ Telemetry Transport

MQTT Connection

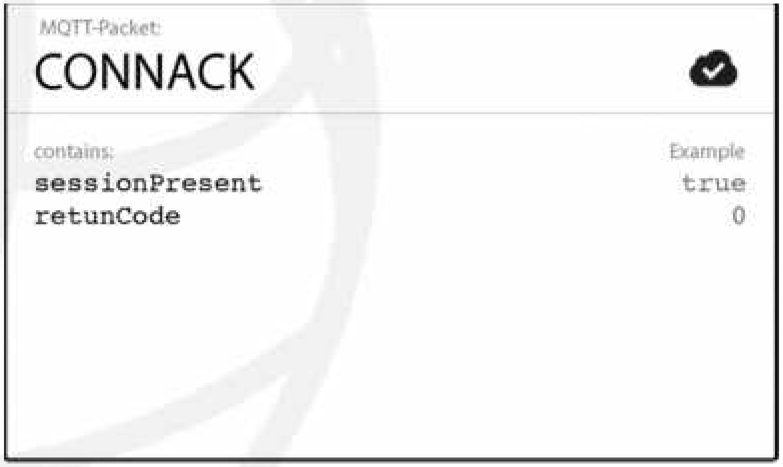
- The connection is initiated through a client sending a CONNECT message to the broker. The broker response with a CONNACK and a status code.
- Username and password are sent in the CONNECT message. The username is a UTF-8 encoded string and the password is binary data with each 65535 bytes max.



MQTT-Packet:

CONNECT

contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60



MQTT-Packet:

CONNACK

contains:	Example
sessionPresent	true
retunCode	0

MQ Telemetry Transport

MQTT Connection

- Transport encryption is necessary since usernames and passwords are sent in clear text
- MQTT broker will evaluate the credential based on the implemented authentication mechanism and return one of the following return codes:

Return code	Return code response
0	Connection Accepted
1	Connection Refused, unacceptable protocol version
2	Connection Refused, identifier rejected
3	Connection Refused, Server unavailable
4	Connection Refused, bad user name or password
5	Connection Refused, not authorized

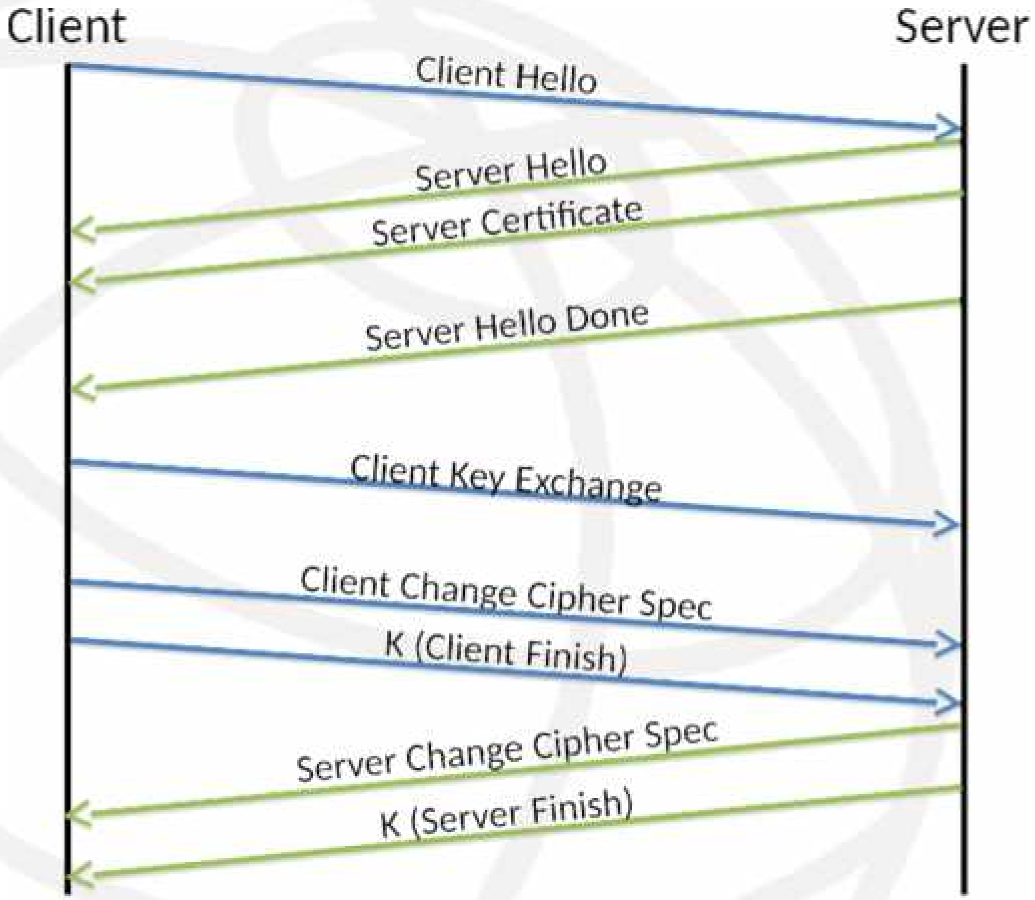
Transport Layer Security

Role

- Transport Layer Security (TLS) is a cryptographic protocol ensuring transport encryption with MQTT
- TLS includes a handshake mechanism to negotiate various parameters needed to create a secure connection between a TLS client and TLS server
- TLS is necessary with MQTT to protect the user credentials sent in the MQTT CONNECT packet
- TLS is based on X.509 certificates to authenticate servers and optionally clients

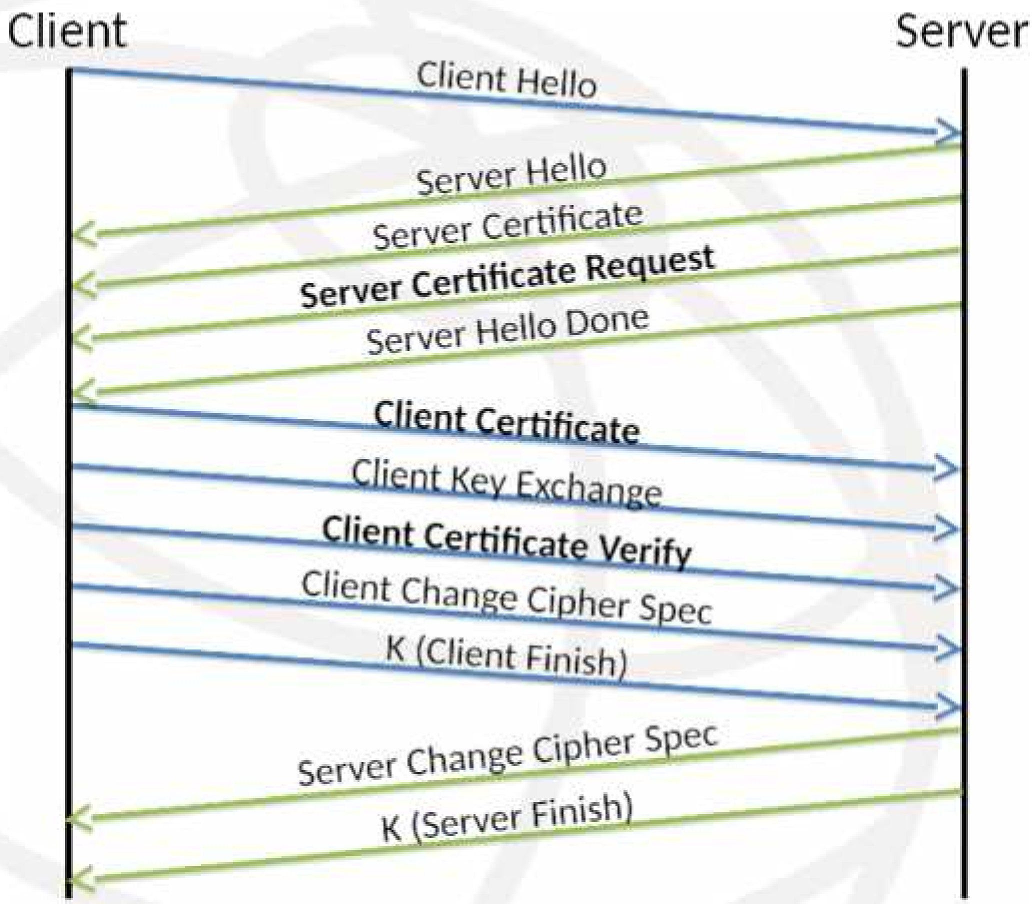
Transport Layer Security

Simple TLS Authentication



Transport Layer Security

Mutual TLS Authentication



Transport Layer Security

Overhead

- TLS brings security a cost in terms of CPU usage and communication overhead which is problematic for very constrained devices
- TLS Session Resumption (caching) can considerably improve TLS performance by recalling information from a previous successful TLS session negotiation to bypass the most computationally intensive parts of the TLS session key negotiation. There are two session resumption mechanisms:
  - Session IDs
  - Session Tickets
- TLS Handshake can be significant so long-living TCP connections are more recommended
- TLS 1.3 is the best version for constrained devices (highest TLS version)

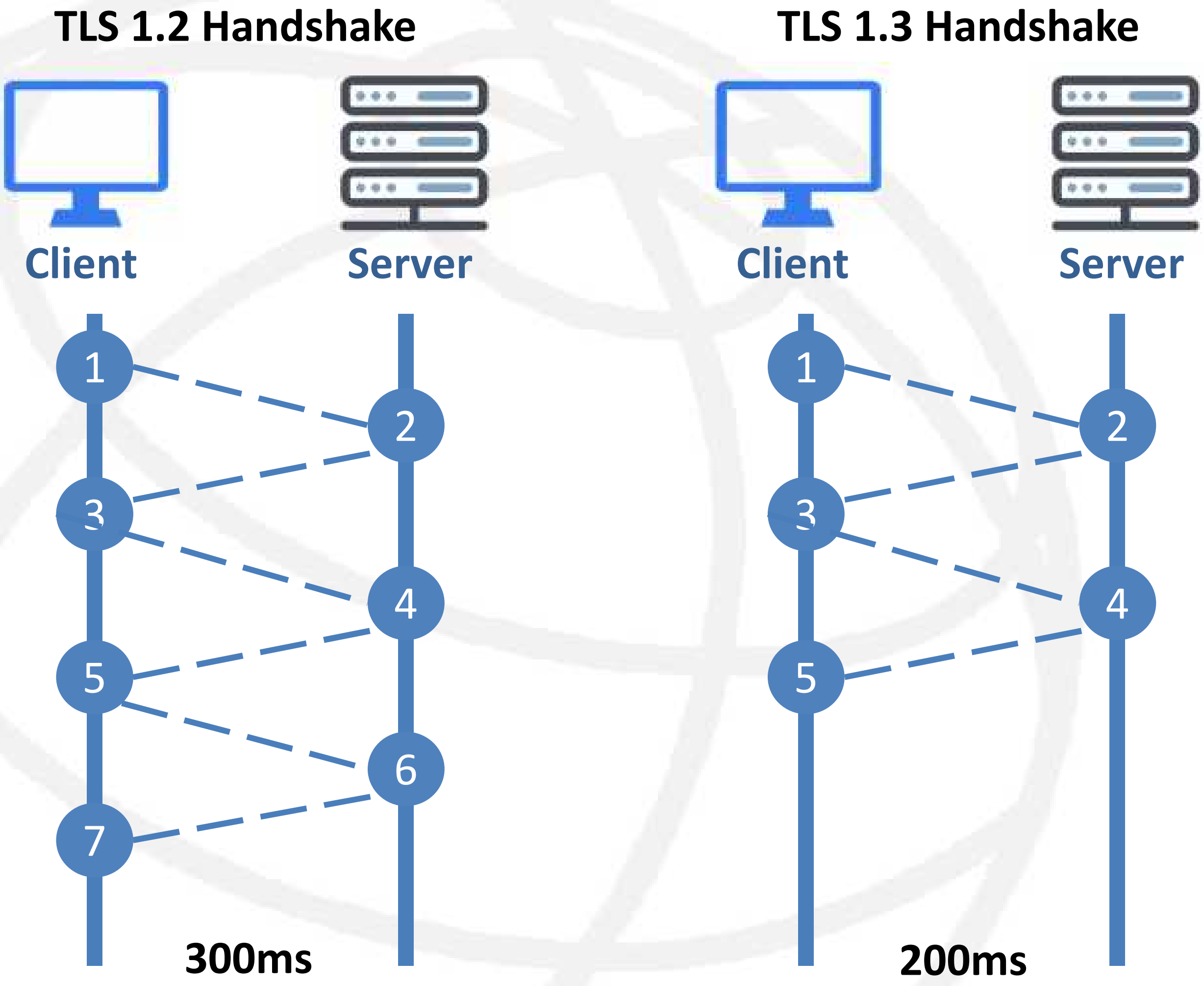
Transport Layer Security

TLS 1.3

- TLS 1.3 is light, IoT oriented and more secure
- TLS 1.3 is faster since it requires only one round-trip (1-RTT) for the first connection and Zero Round Trip Time (0-RTT) for the previously established connections
- TLS 1.3 removes legacy options of insecure and weak ciphersuites, hash functions and cipher algorithms:
  - RSA key transport — Doesn't provide forward secrecy
  - CBC mode ciphers — Responsible for BEAST, and Lucky 13
  - RC4 stream cipher — Not secure for use in HTTPS
  - MD5 and SHA-1 hash function — Deprecated in favor of SHA-2
  - Arbitrary Diffie-Hellman groups — CVE-2016-0701
  - Export ciphers — Responsible for FREAK and LogJam

Transport Layer Security

TLS 1.3 handshake Performance



Transport Layer Security

Best Practices

- Always use TLS not SSL and the highest available version
- Always validate the TLS X.509 certificate chain
- Always use X.509 certificates from trusted certification authorities (not self-signed certificates)
- It is necessary to correctly validate the X.509 certificate not only the trust aspect (expiration, revocation, …)
- It is preferable to use additional security mechanisms with TLS like payload encryption and payload signature
- Only use secure cipher suites avoid obsolete and weak algorithms and keys
- Client authentication using digital certificates helps filtering MQTT clients at the transport level and helps saving resources on the broker side (avoid database lookups and webservice calls to verify clients)

Lightweight Cryptographic Primitives

Device Spectrum

- Conventional cryptography performs well on powerful machines and not very well on highly constrained devices
- Highly constrained devices and sensors are generally equipped with
  - 4-bit, 8-bit, …up to 32-bit microcontrollers resulting in a large number of cycles when executing common crypto algorithms which may make them too slow or energy-consuming
  - extremely limited read-only memory (ROM) and random-access memory (RAM) of 64 bytes or less, going down to as little as 16 bytes
  - RFID tags realized in an application-specific integrated circuit (ASIC) which are not battery-powered, requiring small amount of gate equivalents (GE) and meet stringent timing and power requirements

1	Servers and desktop machines	Conventional cryptography
2	Tablets and smartphones	
3	Embedded systems	Lightweight cryptography
4	RFID and sensor networks	

Lightweight Cryptographic Primitives

Performance Metrics

- Performance is expressed in terms of
  - Power
  - Energy consumption
  - Latency
  - Throughput
- Resources required for a hardware implementation concerns essentially gate area or logic blocks
- Resources required for a software implementation concerns registers, RAM and ROM usage
- Design considerations include
  - Security strength (at least 112bit keylength and attack resistant)
  - Flexibility and efficient implementations in different platforms
  - Low overhead for multiple functions (encryption and decryption use similar round functions)
  - Ciphertext expansion (comparing to the size of the plaintext)

Lightweight Cryptographic Primitives

Primitives

- Lightweight primitives are either redesigned conventional primitives or new defined ones in order to have:
  - Smaller block sizes
  - Smaller key sizes (the recommended minimum key size is 112bits)
  - Simpler rounds
  - Simpler key schedules
  - Simpler implementations
  - Smaller message size
  - Smaller internal state
- Lightweight primitives concerns
  - Lightweight Block Ciphers
  - Lightweight Stream Ciphers
  - Lightweight Hash functions
  - Lightweight Message Authentication Codes

Lightweight Cryptographic Primitives

Examples of Lightweight Primitives

Block ciphers	AES-128, TDEA, DESL, PRESENT, SIMON, SPECK, RC5, TEA, XTEA, PHOTON, LED, …
Stream ciphers	Grain, Trivium, Mickey, FRUIT, …
Hash functions	PHOTON, Quark, SPONGENT, Lesamnta-LW, …
Message authentication codes	Chaskey, TuLP, LightMAC, …

NIST approved cryptographic primitives in constrained environment	
Block ciphers	AES and TDEA
Hash functions	SHA-1 (no more recommended) SHA-2 family (-224, -256, -384, -512, -512/224 and -512/256) SHA-3 family (-224, -256, -384, and -512) -> none of these NIST approved hash functions are suitable for use in very constrained environments
Authenticated Encryption Algorithms and MACs	CCM (Cipher Block Chaining-Message Authentication Code) GCM (Galois/Counter Mode) MAC, CMAC (cipher-based MAC), HMAC (Hash function-based MAC), and GMAC (Galois MAC)

Public Key Infrastructure

Concept

- To understand PKI we need to understand the concept of **Public Key Cryptography**.
- In Public Key Cryptography, we use a key pair (private and public)
- The **private key,** must be kept secret and (usually) under the control of the owner and the **public key**, can be disseminated freely for use by any person who wishes to participate in security services with the entity (machine or person) holding the private key.
- With Public Key Cryptography the delivery of the secret (shared or session) key between two communicating entities is easy to set up.
- A **Public Key Infrastructure (PKI)** is designed to provide the trust and the confidence that the used public keys truly belong to the persons (machines) with whom (which) we wish to communicate.

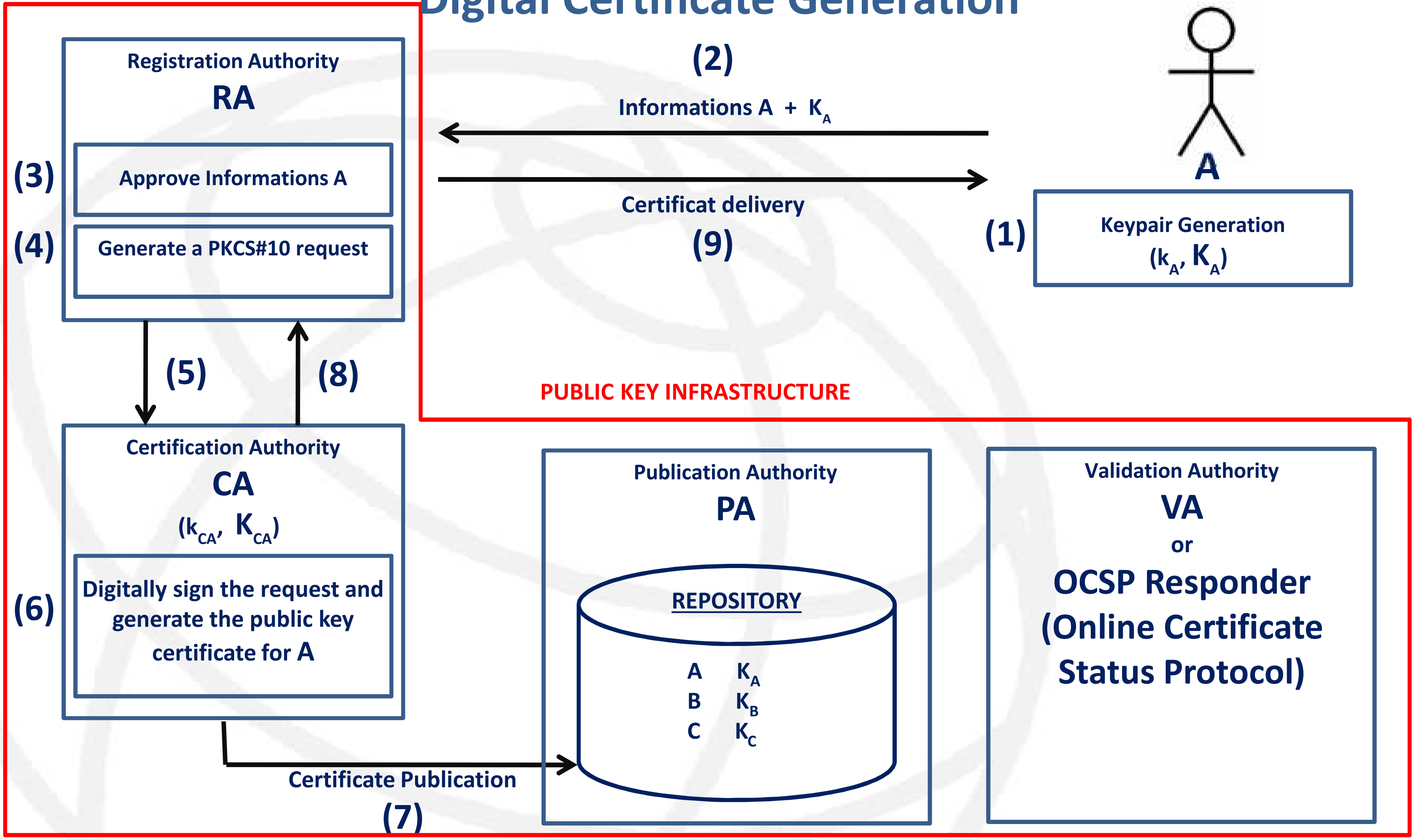
Public Key Infrastructure

Digital Certificate

- PKI is built around a data element called **Digital Certificate** or public key certificate which binds a public key to its holder
- Digital Certificate is an **authentication technology** that can be delivered to
  - Persons
  - Organisations
  - Devices
  - Software solutions
- It binds a public key to information about its owner
- Digital certificates can be used for system, network and application authentication
- ITU-T X.509 v3 is the standard of the public key certificates

Public Key Infrastructure

Digital Certificate Generation



Registration Authority

RA

(3) Approve Informations A

(4) Generate a PKCS#10 request

(2)

Informations A + K_A

Certificat delivery

(9)

A

(1) Keypair Generation

(k_A, K_A)

(5)

(8)

PUBLIC KEY INFRASTRUCTURE

Certification Authority

CA

(k_{CA}, K_{CA})

(6) Digitally sign the request and generate the public key certificate for A

Publication Authority

PA

REPOSITORY

A	K_A
B	K_B
C	K_C

Certificate Publication

(7)

Validation Authority

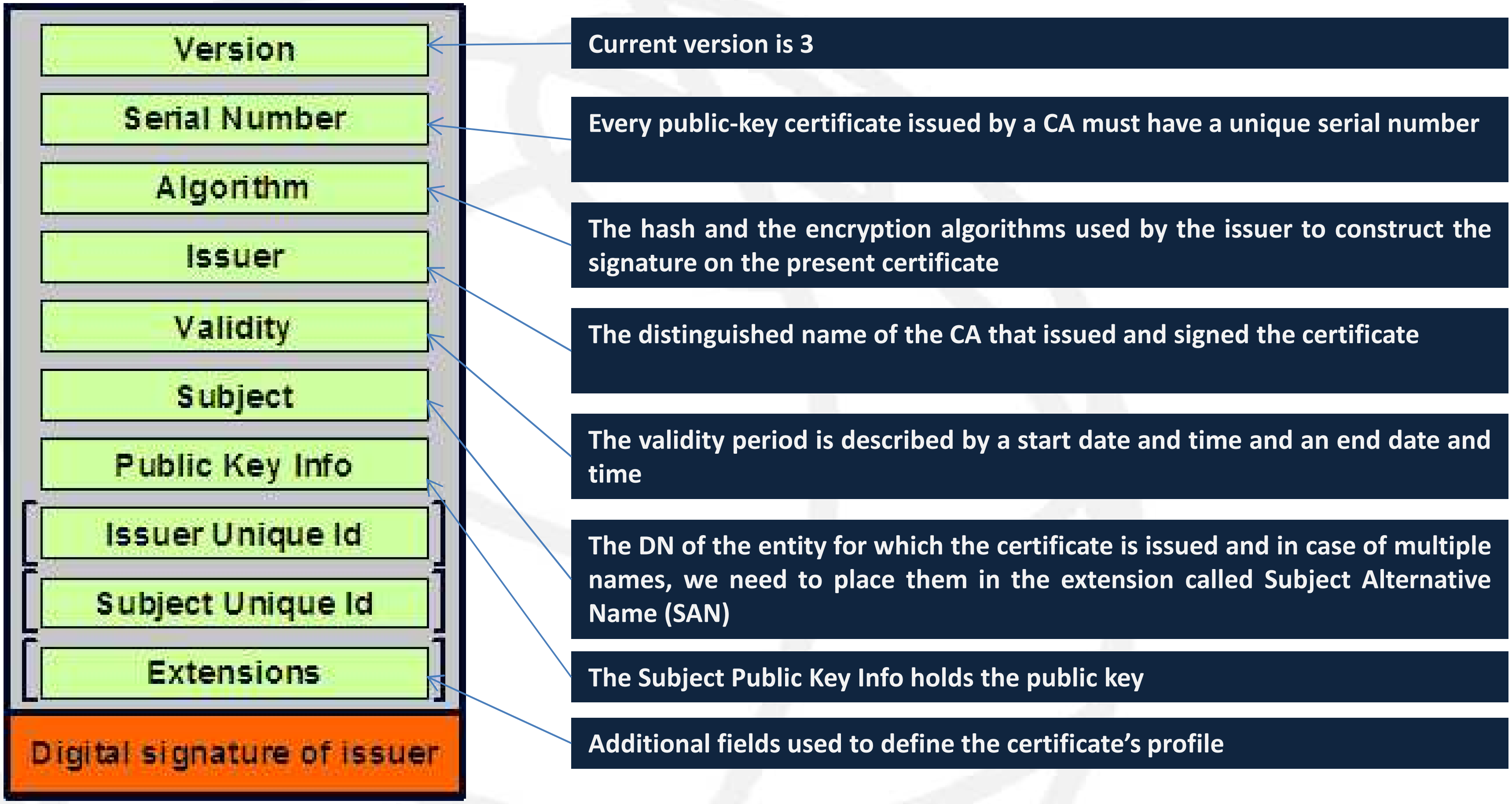
VA

or

OCSP Responder (Online Certificate Status Protocol)

Public Key Infrastructure

ITU-T X.509 Digital Certificate Structure



Current version is 3

Every public-key certificate issued by a CA must have a unique serial number

The hash and the encryption algorithms used by the issuer to construct the signature on the present certificate

The distinguished name of the CA that issued and signed the certificate

The validity period is described by a start date and time and an end date and time

The DN of the entity for which the certificate is issued and in case of multiple names, we need to place them in the extension called Subject Alternative Name (SAN)

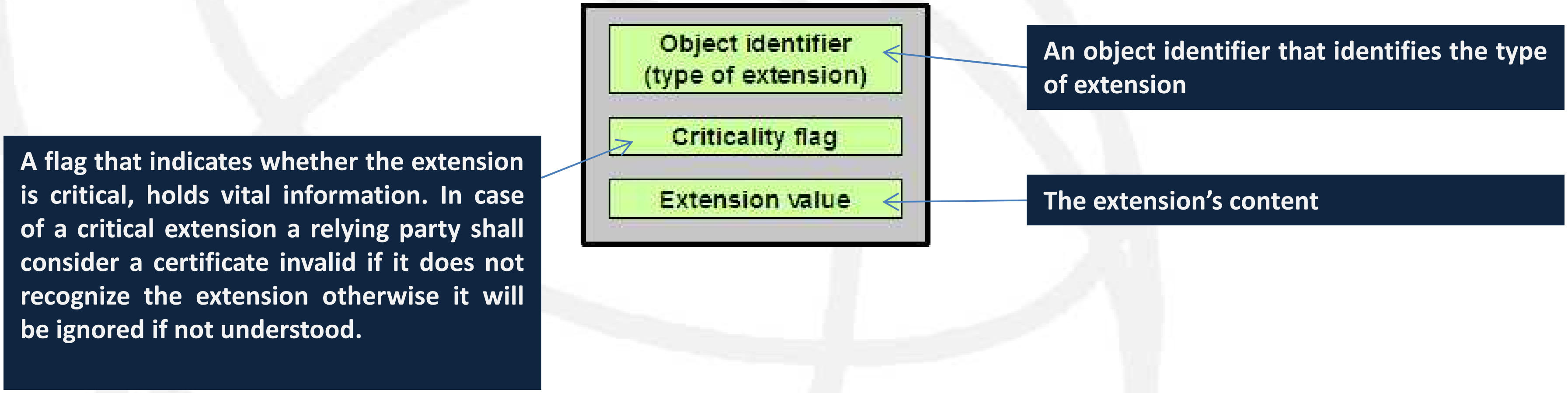
The Subject Public Key Info holds the public key

Additional fields used to define the certificate’s profile

Public Key Infrastructure

X.509 Certificate Extensions

- An extension provides additional information about the digital certificate or it defines the certificate’s restrictions.
- The X.509 extension’s structure is defined as follows:



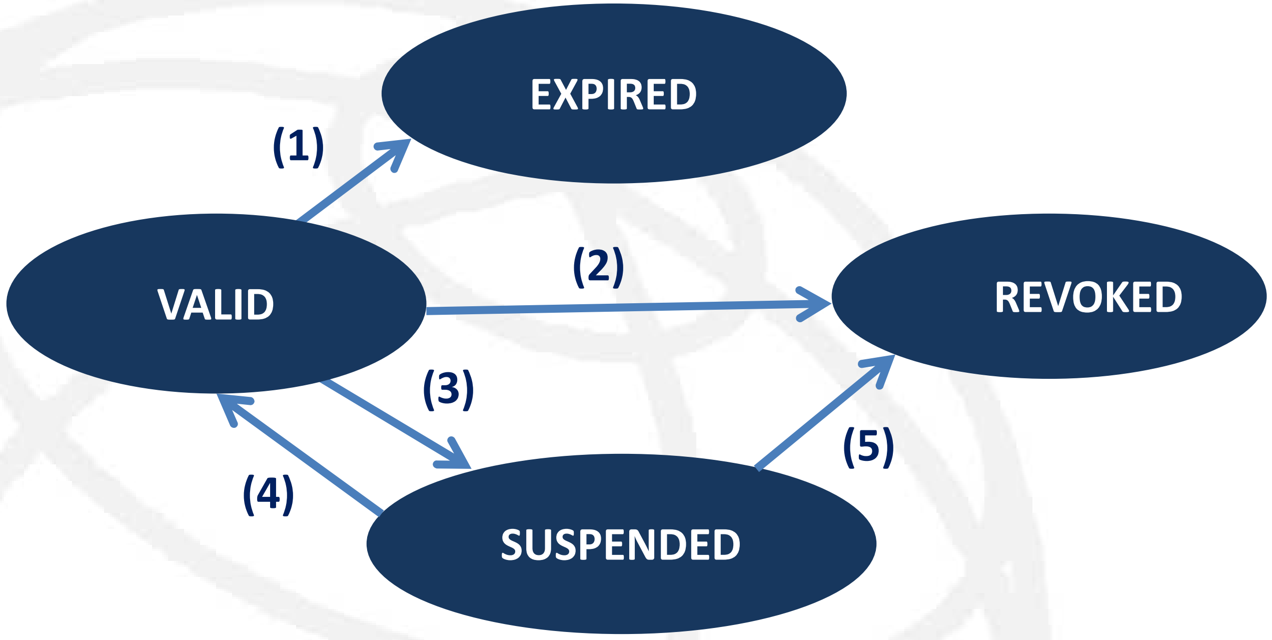
An object identifier that identifies the type of extension

A flag that indicates whether the extension is critical, holds vital information. In case of a critical extension a relying party shall consider a certificate invalid if it does not recognize the extension otherwise it will be ignored if not understood.

The extension’s content

Public Key Infrastructure

Certificate's life cycle



(I) Certificate has a fixed lifetime and it reaches the end of its validity date
(2) Private key is compromised or affiliation has changed (the end entity may not be involved in the revocation phase)
(3) Certificate is temporarily revoked for security reasons (can be the initial status)
(4) Certificate activation to make it valid again
(5) Same as the transition (2)

Public Key Infrastructure

Device Certificate

- A device digital certificate is generated by a **Public Key Infrastructure (PKI)** and it is **permanent certificate** used to authenticate devices in an IoT environment
- It binds the device **model and serial number** to its **public key** so that the certificate is intended to be used for the entire lifetime of the device
- A solid and secure certificate provisioning and lifecycle management processes are needed since devices can be located anywhere
- Invalidating malicious nodes need to be done using Certificate Revocation Lists (CRL) and preferably OCSP responders (Online Certificate Status Protocol) since CRLs can be too huge to be downloaded by a constrained device
- X509 client certificates are typically used when the whole MQTT system is controlled from broker to clients

Public Key Infrastructure

Device Certificate Profile

Field Name	RFC5759/RFC5280 Type	Value / Example
Version	Integer	V3
serialNumber	Integer	Positive integer of up to 8 octets
Signature	AlgorithmIdentifier	SHA256 with ECDSA
Issuer	Name	Globally unique name of issuing device CA
authorityKeyIdentifier	KeyIdentifier	A unique value that matches the subjectKeyIdentifier of the issuer's credentials
subjectKeyIdentifier	KeyIdentifier	Provides means for identifying certificates containing particular Public Key used in an application
notBefore	Time	Creation time of the device certificate

Public Key Infrastructure

Device Certificate Profile

notAfter	Time	Shall be assigned the GeneralizedTime value of 99991231235959Z
Subject	Name	Empty
subjectAltName	OtherName	Contains a single GeneralName of type OtherName that is further sub-typed as a HadrwareModuleName as defined in RFC 4108. The hwSerialNum field shall be set to the Device Entity's identifier
subjectPublicKeyInfo	subjectPublicKeyInfo	The subject's public key
Extensions	Extensions	Critical and non critical extensions
signatureAlgorithm	AlgorithmIdentifier	SHA256 with ECDSA
SignatureValue	BIT STRING	Subject Device Certificate Signature

Key Takeaways

Needed solutions

- Prevent devices cloning
- Implement end point visibility for your infrastructure
- Conduct code security inspections
- Prevent data hijacking
- Implement intrusion response plan
- Perform vulnerability audits regularly
- Test for scale is required before the deployment
- Monitor systems and networks
- Place as much as possible the IoT devices in non-public places
- Integrate anti-tampering mechanisms into the embedded chips
- Link the hardware characteristics to the software security mechanisms (key generation based of hardware fingerprint)

Software Authenticity

Code Signing

- Authenticity and integrity of software solutions installed on devices are important for the correct functioning of the IoT system since corrupted software can allow for the security mechanisms to be bypassed
- Software solutions that need to be authentic: Operating Systems (OS), drivers, patches, …
- Digitally signing software solutions is a common mean used to defend against such security risk
- The code signing digital signature is attached to the executable and provides a cryptographically verifiable proof of the soft and the software editor

Self-configuration

Context Aware Security

- Unrealistic to **manually** set up, configure and update billions of connected devices
- **Automated systems** are capable of complex, monotonous, and tedious operations that human users would never tolerate.
- Self-configuration and self management of access control are needed without or with **minimal user intervention**
- **Context aware** devices are able to gather information about the environment and adapt their mode of functioning according to it
- Context awareness is the core feature of smart systems
- **Adaptive security** configuration is the ability to adapt in real time the security mechanisms to be able to respond to a complex and constantly changing context
- **Device context** refers to time, location, identities, status and behavior of the surrounding objects and the persons

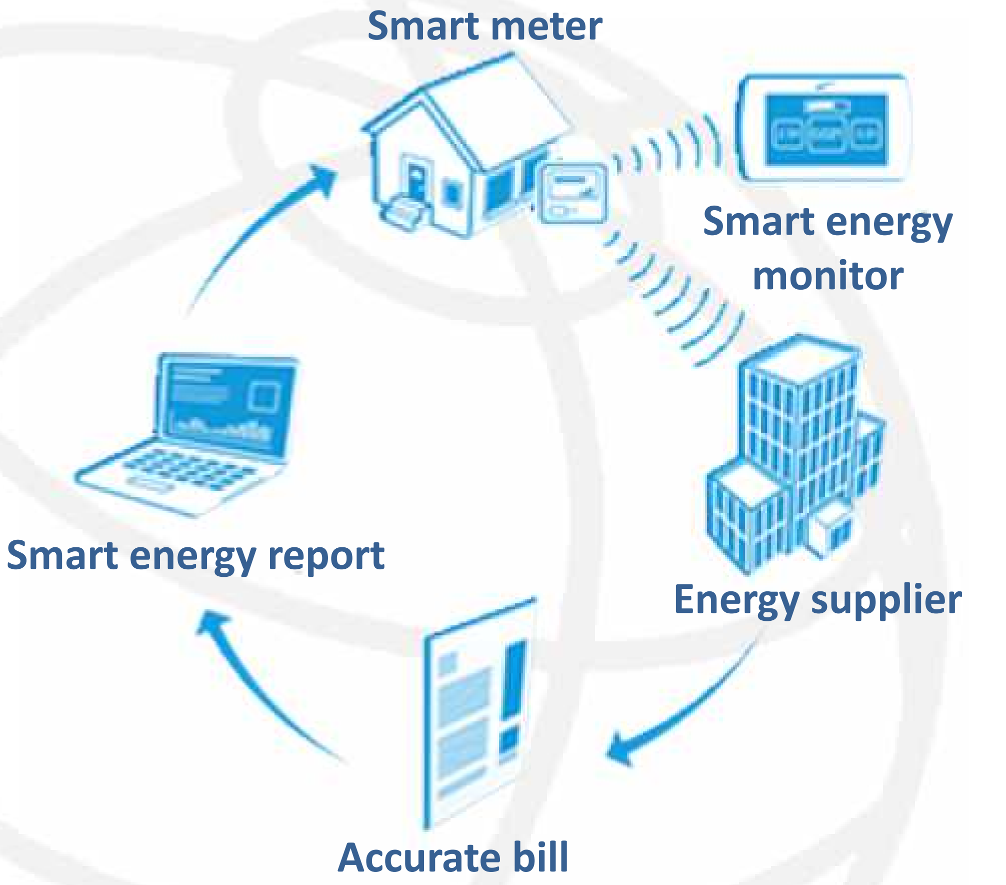
3

Case study on the security of smart metering systems

1. Conceptual architecture
2. System components
3. Benefits
4. Types of attackers
5. Common attacks
6. Security needs
7. Smart Metering Key Infrastructure (SMKI)

Smart Metering System

Conceptual architecture



Smart Metering System

System components

- The smart metering system is composed of:
  - Smart meters are devices installed at the customer's premises (house or facility) to measure the consumption of commodities such as gas, electricity and water. Smart measures are able to measure the quantity of commodities in terms of volume or energy either imported or exported.
  - Communications hubs composed of
    - A Local Metrological Network (LMN)
    - A Home Area Network (HAN)
    - A Wide Area Network (WAN)
  - In home display unit or consumption monitor to show the usage and pricing in real time (kw, amount to pay, …)

Smart Metering System

Benefits

- The smart metering system improves utility operations:
  - Provides detailed information of what is being consumed/delivered on a real time basis
  - Provides periodic and off-cycle meter readings (on demand)
  - Issues accurate and regular energy bills without having to periodically collect or to manually submit meter readings
  - Parameters and controls the metering systems remotely (load customer profiles and parameter the remote readouts)
  - Enables/disables meters remotely
  - Update software on meters so that new protocols or services can be supported
  - Helps localise outages and monitor power quality
  - Highlights areas with possible energy thefts
  - Identify the periods of high energy demands and sources of energy wastes
  - Helps in balancing the power generation and distribution in a smart grid

Smart Metering System

Benefits

- The smart metering systems bring far more accuracy and convenience to the customers
  - Provides real time usage an pricing through and in house display
  - Helps customers identify anomalies and energy wasting points
  - Reduces billing conflicts between the customer and the provider
  - Helps customers deliver energy to grid

Smart Metering System

Types of attackers

- Smart grid is an attractive target for various types of attackers like cyber criminals, terrorists and even the customers themselves
- Two kinds of attackers:
  - Local attackers having physical access to Meter, Gateways or the connection between these components. may try to modify (i.e. alter, insert, delete, redirect or replay) Meter Data when transmitted between Meter and Gateway, Gateway and consumer, or Gateway and external entities and may also try to modify secondary assets like the firmware or configuration parameters. The objective of the attacker may be to alter billing-relevant information or grid status information.
  - A WAN attacker trying to conquer any component of the WAN infrastructure to cause damage to the whole or parts of the grid. It may also try to change meter and gateway data and configuration to alter also billing-relevant information or grid status information.

Smart Metering System

Common attacks

- Passive or active attacks on the smart metering system
- The most common cyber attacks are:
  - Eavesdropping metering data conveyed by smart meters or gateways to the service providers. This kind of man in the middle attacks can be easily performed over a wireless communication channel or a power line. The detection of such passive attacks is very difficult. Breaches of personal data since privacy can be affected by intruders accessing the customers' metering data or by the customers themselves allowing other firms to access their systems and data
  - Denial of service (DoS) attacks targeting the energy delivery can be performed by sending a great deal of commands to the smart metering gateways or to the utility servers. These attacks can be launched through the WAN to saturate the system and then bring down the whole or parts of the grid
  - Payment frauds by transferring false consumption data or by changing the relation between date/time and measured consumption in the meter data records to influence the next invoice balance

Smart Metering System

Common attacks

- Spying customers by analysing their metering data: passive eavesdropping can infer different observations from the metering data (presence of persons inside the building, their number, their activity, …)
- Metering data can be illegally used for marketing and targeted advertisement through monitoring persons habits and behaviours
- Injection attacks can be launched by injecting false packets, false commands or malicious malwares into the network to destabilize the load and the communication on the grid
- Deploy fake smart meter or illegal manipulation of meters' firmware

Smart Metering System

Security needs

- Authenticating metering devices, concentrators, gateways and operators to
  - Entity and user strong authentication and identification before any action and prevent the access of any unauthorized persons or entities
  - Guarantee that only authorised entities are able to provide updates, update firmware, access log and configuration files
  - Use pseudonymity to conceal identity of persons and entities
  - Prevent the deployment of fake devices
- Data security and privacy
  - Securing the local storage of consumption status and demand requirement over time and destroying any information or key that is no longer needed
  - Protecting the metering data and personal information of consumers from the illegal access to it since they are exchanged over public networks and it is susceptible to being seen or changed in transit by unintended entities

Smart Metering System

Security needs

- Preventing the alteration of metering data and metadata (like date and time) either exported or imported
- Metering data needs to be collected and stored in a non-public environment
- Protecting the integrity and the confidentiality of the system log and configuration data
- Authenticity of the communication and data origin need to be ensured using digital signatures
- Protect the Personally Identifiable Information (PII) refers to information that can be used to uniquely identify or locate individuals

- Network security
  - Securing the bi-directional data transmission with end to end encryption and/or channel encryption using security protocols like TLS (Transport Layer Security)
  - Conceal communications to prevent an attacker from analysing the frequency, load, size or the absence of transmission

Smart Metering System

Security needs

- Protecting the network from Denial of service (DoS) attacks or DDoS that could bring down the whole or parts of the grid
- Network time synchronization based on reliable NTP sources of time (Network Time Protocol)
- High availability through a fail-safe design that specifically ensures that no malfunction can impact the delivery of a commodity
- Intrusion Detection and Prevention Systems (IDS) could be used in the networks of a smart metering system to help identify intruders and rogue nodes or sources of attacks to be able to isolate and prevent them from further communication in the network. System shall maintain a set of log files that need to be analysed automatically with automated alarms:
  - System log
  - Consumption log
  - Calibration log

- Hardware and physical security
  - Make detectable any physical suspicious manipulation within the scope of the intended environment

Smart Metering System

Security needs

- Only authentic and integrity protected updates can be applied
- Use of Hardware Security Modules (HSM) to protect server signing keys
- Use of smart cards to protects user's authentication keys and certificates
- Both HSMs and smart cards guarantees
  - An on-board (on chip) cryptographic operations
  - Unicity of crypto keys used for authentication and for signature
  - PIN/PUK protection

- Operational security
  - Secure remote command execution
  - Monitoring user data and software for integrity errors and attacks
  - Secure management processes and authorize management operations only from the WAN interface
- Security by design means that the software and hardware systems for smart meters should be developed based on security analysis, security design, secure implementations and security testing

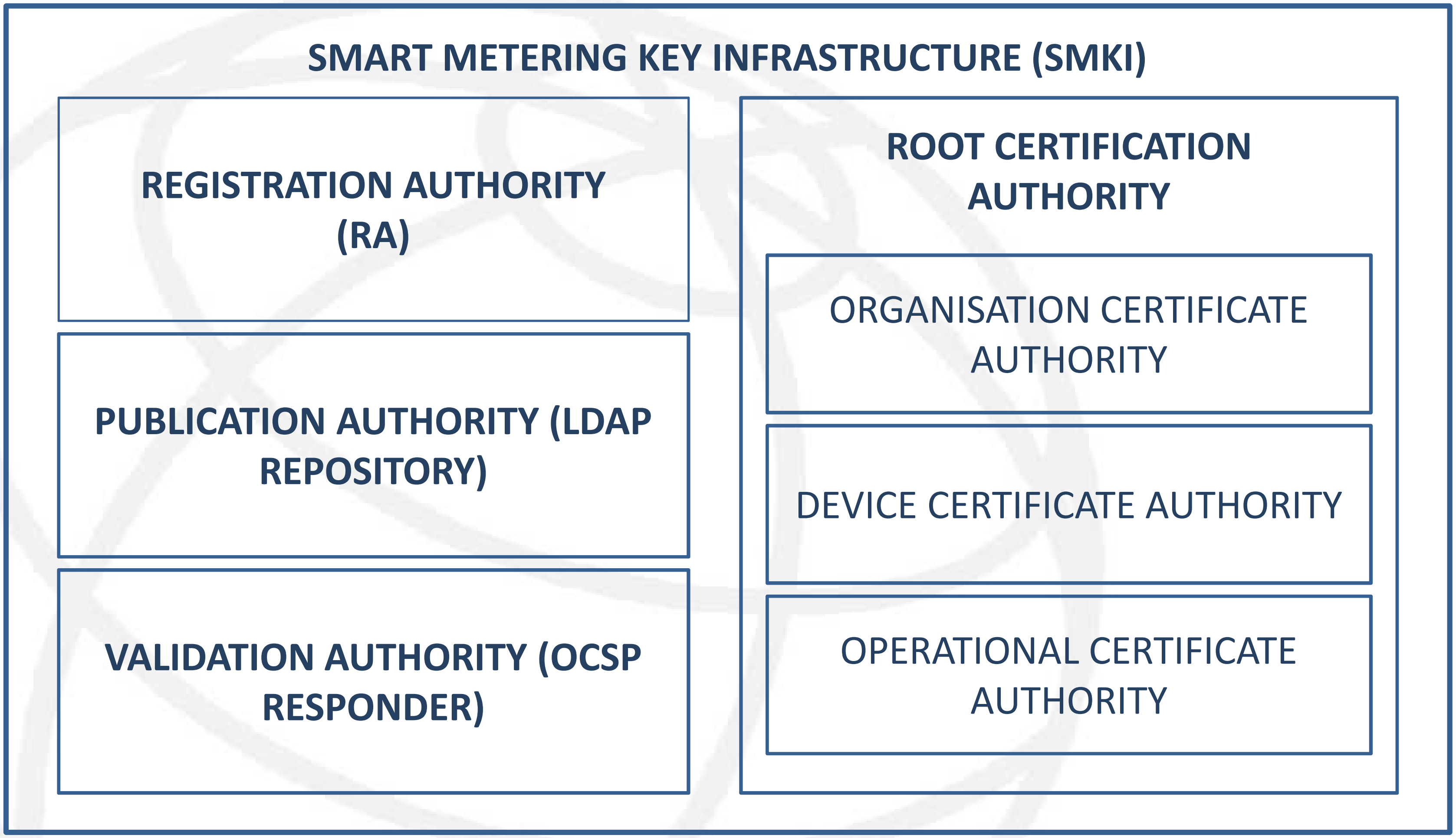
Smart Metering Key Infrastructure

Role

- SMKI is a public key infrastructure for smart meters
- A cryptographic component used to generate, distribute, enrol and revoke digital certificates and to generate and update certificate revocation lists (CRL) to be used to:
  - Authenticate users and devices
  - Encrypting the communications over the local and public network
  - Digitally sign the exchanged messages and documents like metering data and bills
  - Encryption of the persistently stored data
  - Replay detection for all communications with external entities
- Digital certificates are electronic documents used to prove ownership of public keys
- Certificate Policy (CP) and Certificate Policy (Statement)
- Most used cryptosystems: Elliptic Curve Cryptosystem (ECC), TLS 1.3 (Transport Layer Security), SHA 256, …

Smart Metering Key Infrastructure

Architecture



SMART METERING KEY INFRASTRUCTURE (SMKI)

REGISTRATION AUTHORITY (RA)

PUBLICATION AUTHORITY (LDAP REPOSITORY)

VALIDATION AUTHORITY (OCSP RESPONDER)

ROOT CERTIFICATION AUTHORITY

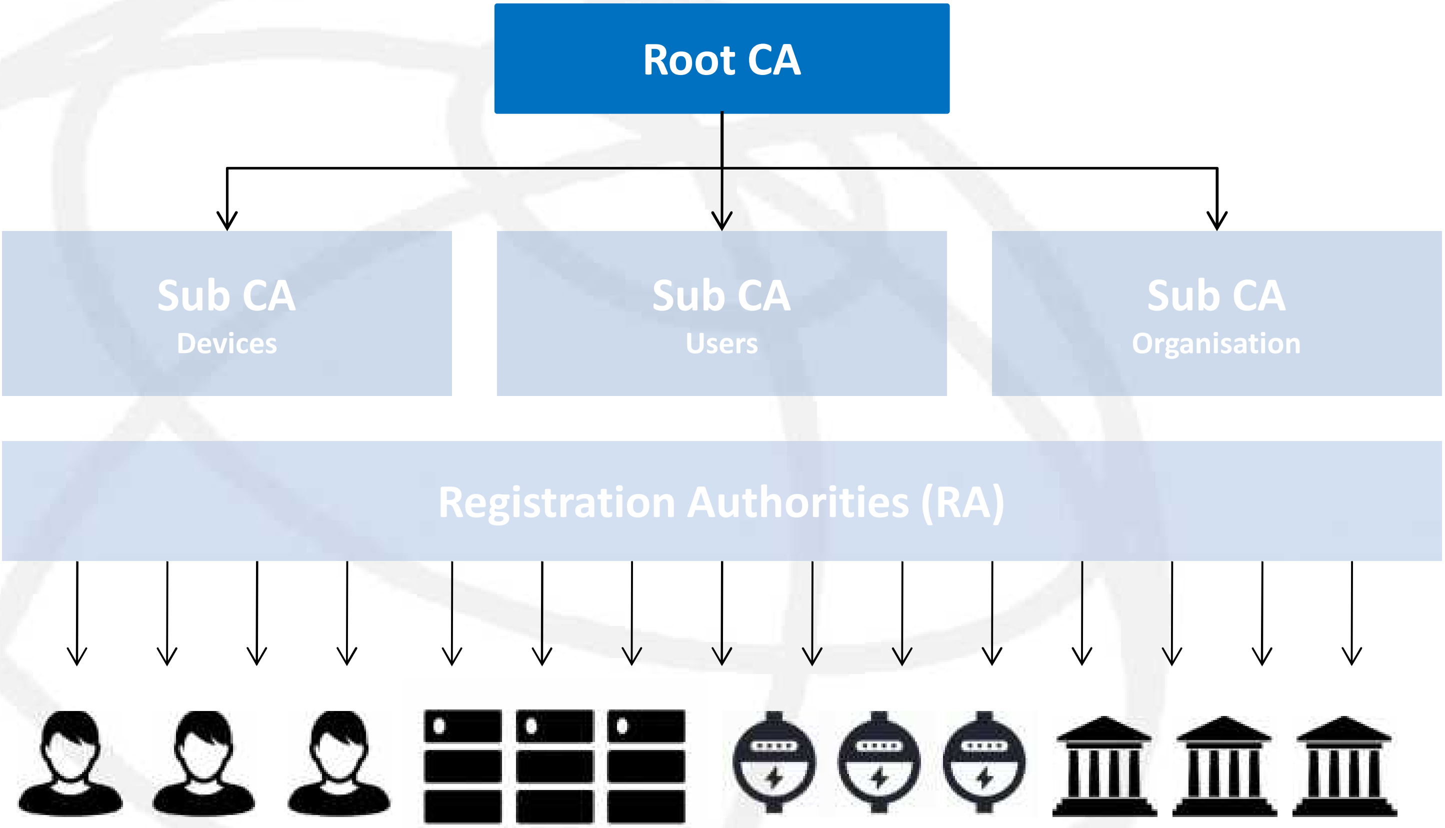
ORGANISATION CERTIFICATE AUTHORITY

DEVICE CERTIFICATE AUTHORITY

OPERATIONAL CERTIFICATE AUTHORITY

Smart Metering Key Infrastructure

Architecture



THANK YOU FOR YOUR ATTENTION

Dr. Eng. Nizar Ben Neji
IT Security Trainer, Consultant and Researcher
PhD in Information and Communication Technologies
nizar.benneji@fsb.rnu.tn /(+216) 99 207 377