Cybersecurity Essentials and Management for Energy Sector

CSP001_C_E

PRESENTATION BY:
ANTONIO MUÑOZ
UNIVERSITY OF MALAGA, SPAIN

Acknowledgement

- *Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.*
- *Project Agreement no. 101083594*

Topic-5: Secure Architecture Design and Implementation for Energy Systems

Overview

- Design and implement secure network architectures for energy systems
- Secure network architecture in Energy Sector including SCADA systems, smart grids, and other critical energy assets
- Utilise network segmentation to isolate critical systems and reduce the impact of cyberattacks
- Configure firewalls and access control systems to protect energy networks and restrict unauthorised access
- Implement intrusion detection and prevention systems (IDS/IPS) to monitor and protect networks
- Employ VPNs for secure remote access to energy systems and sensitive data

Topic-5: Secure Architecture Design and Implementation for Energy Systems

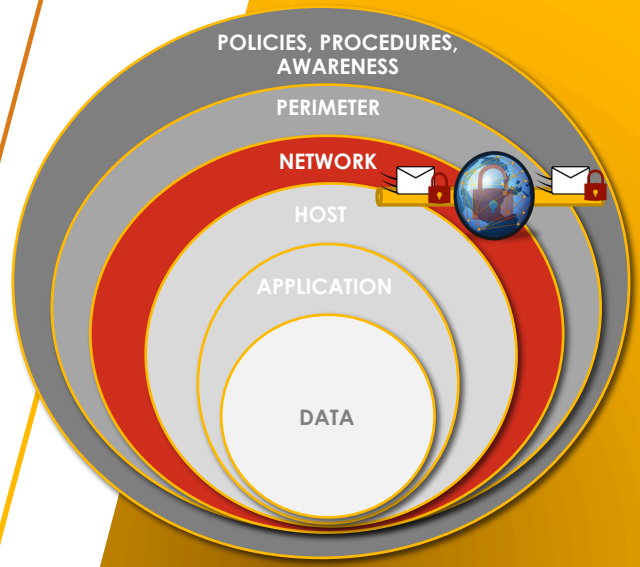
Overview

- Design and implement secure network architectures for energy systems
- Secure network architecture in Energy Sector including SCADA systems, smart grids, and other critical energy assets
- Utilise network segmentation to isolate critical systems and reduce the impact of cyberattacks
- Configure firewalls and access control systems to protect energy networks and restrict unauthorised access
- Implement intrusion detection and prevention systems (IDS/IPS) to monitor and protect networks
- **Employ VPNs for secure remote access to energy systems and sensitive data**

Secure remote access in energy systems

- In line with the principles of defence in depth, remote access is part of network level protection

- The European Union Agency for Cybersecurity (ENISA) in "*Appropriate security measures for smart grids*" also identifies the remote access as a priority:
  - "*The provider should establish and maintain secure remote access where applicable to smart grid information systems*" – SM 9.4
- **Virtual Private Networks** (VPNs) are the most usual mechanisms for remote access
  - A VPN is defined by the National Institute of Standards and Technology (NIST) as "*A virtual network built on top of existing networks that can provide a secure communications mechanism for data and IP information transmitted between networks*"



ID	**SM 9.4**
Measure	Secure remote access
Definition	The provider should establish and maintain secure remote access where applicable to smart grid information systems.
Example	[From NISTIR 7628 - SG.AC-2 Remote Access Policy and Procedures - Requirement 1] The organisation documents allowed methods of remote access to the smart grid information system. [From IEC 62443 - 4.3.3.6.6 Develop a policy for remote login and connections] The organisation shall develop a policy addressing remote login by a user and/or remote connections (for example, task-to-task connections) to the control system which defines appropriate system responses to failed login attempts and periods of inactivity.

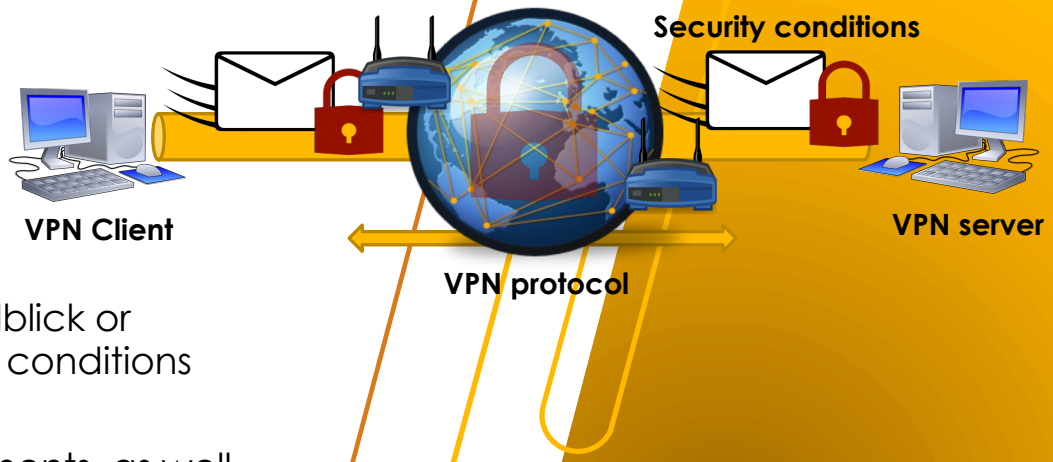
Network virtualization is often linked to the concept of "tunnelling"

REMOTE ACCESS IN ENERGY SYSTEMS

Source: CSRC, "Glossary", NIST, 2024.
URL: https://csrc.nist.gov/glossary
Source: ENISA, "Appropriate security measures for smart grids. Guidelines to assess the sophistication of security measures implementation", 2012.
URL: https://www.enisa.europa.eu/publications/appropriate-security-measures-for-smart-grids

Main elements of a VPN and execution stages

- A VPN involves a set of **essential elements**:
  - VPN server or VPN gateway
  - VPN client corresponds to a specific application (e.g. tunnelblick or OpenVPN) capable of understanding a set of configuration conditions and interfaces
  - Configurations with respect to Interfaces and virtual components, as well as the type of encryption, authentication, key exchange mechanisms, etc.
  - VPN communication protocols



Main elements of a VPN and execution stages

- A VPN involves a set of **essential elements**:
  - VPN server or VPN gateway
  - VPN client corresponds to a specific application (e.g. tunnelblick or OpenVPN) capable of understanding a set of configuration conditions and interfaces
  - Configurations with respect to Interfaces and virtual components, as well as the type of encryption, authentication, key exchange mechanisms, etc.
  - VPN communication protocols

- The security performed by a VPN follows a set of **stages of execution**:
  - Mutual authentication
  - Negotiation of security parameters
  - Establishment of the secure point-to-point communication channel

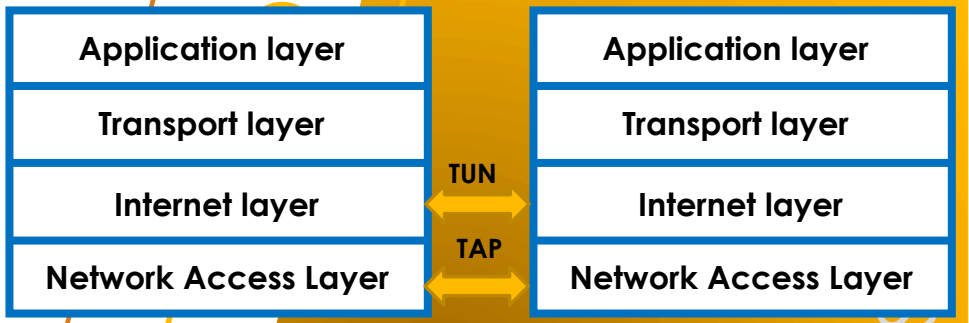
Four types of VPNS

- **Remote access VPNs**: A type of VPN that allows users (e.g., IT/OT administrators, engineers, directives, etc.) to connect to resources of an organisation remotely
- **Host-to-host VPNs**: It is similar to remote access, but the connection is between two computers such as two servers, rather than a client-server
- **Site-to-site VPNs**: A variant of VPN known as "site-to-site" applied to connect two or more LANs (LAN-to-LAN), such as two or more corporate networks belonging to different infrastructures (E.g. SCADA-SCADA)
  - The connection is established between the gateways/routers of the LANs

- **webVPNs**: This type of VPN allows remote access via a web server, avoiding the need to install client software

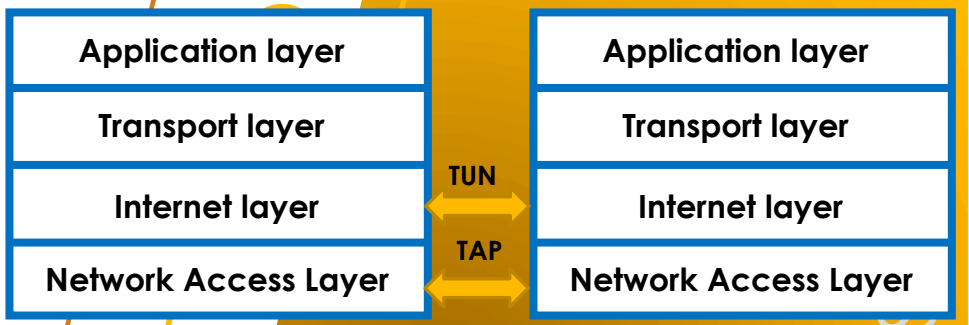
Types of VPN interfaces and protocols

- For fast connections, it is possible to configure the VPN using virtual interfaces that allow to emulate the connection instances of a physical network from the operating system kernel
- There are two types of **virtual interfaces**:
  - **TUN**: Works at the network layer and encapsulates network-level frames, which benefits the management of IPv4 and IPv6 packets and their routing
  - **TAP**: Works only on the link layer, so it encapsulates link-level frames
- This also means that TUN/TAP interfaces allow sending packets from the corresponding TCP/IP stack and from the operating system, emulating also the reception of packets from the destination
  - Unfortunately, not all VPN protocols support both interfaces, TUN/TAP



Types of VPN interfaces and protocols

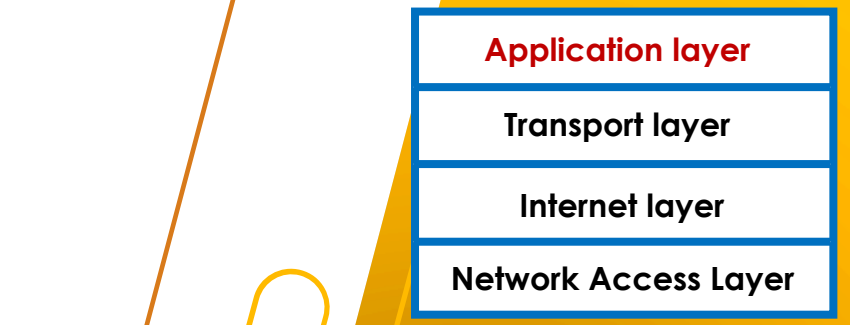
- For fast connections, it is possible to configure the VPN using virtual interfaces that allow to emulate the connection instances of a physical network from the operating system kernel
- There are two types of **virtual interfaces**:
  - **TUN**: Works at the network layer and encapsulates network-level frames, which benefits the management of IPv4 and IPv6 packets and their routing
  - **TAP**: Works only on the link layer, so it encapsulates link-level frames
- This also means that TUN/TAP interfaces allow sending packets from the corresponding TCP/IP stack and from the operating system, emulating also the reception of packets from the destination
  - Unfortunately, not all VPN protocols support both interfaces, TUN/TAP
- Currently, there are many types of **VPN protocols** working on the different layers of the TCP/IP stack
  - Application layer: **SSH**
  - Transport layer: **TLS, QUIC, OpenVPN, Wireguard**
  - Internet layer: **IPSec**
  - Network access layer: **L2TP/IPSec, MACSec**



Note that in this course, we will only explore some of these protocols, not all of them

Application layer - Secure Shell (SSH)

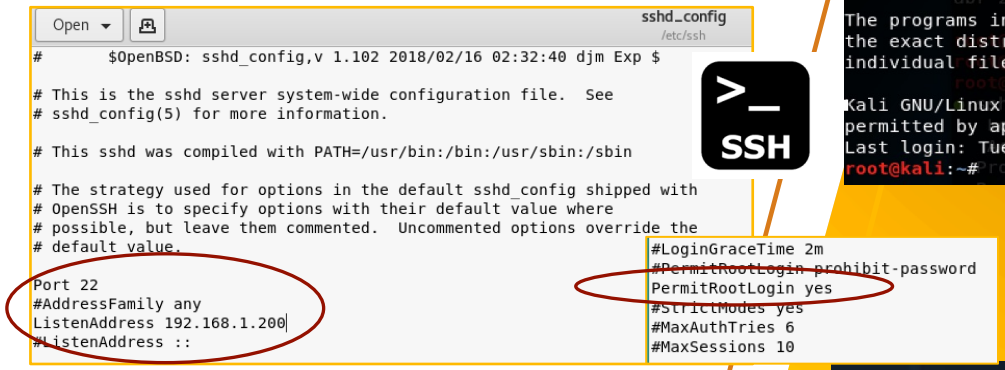
- **SSH establishes P2P communication over port 22**
  - Replaces traditional remote access methods such as telnet or FTP, as both protocols are insecure
  - Legacy operating devices may still rely on telnet and FTP for some monitoring tasks - avoid it !
- The communication between the client and the server:
  - Both the client and the server can be authenticated by using a username/password or public key cryptography
  - The communication is encrypted using encryption mechanisms such as AES, Blowfish, DES, ...
- There are currently two versions of SSH
  - SSH v.1: Deprecated because of its vulnerability to MITM
  - **SSH v.2**: Avoids MITM attacks by creating a tunnel P2P, as well as spoofing attacks

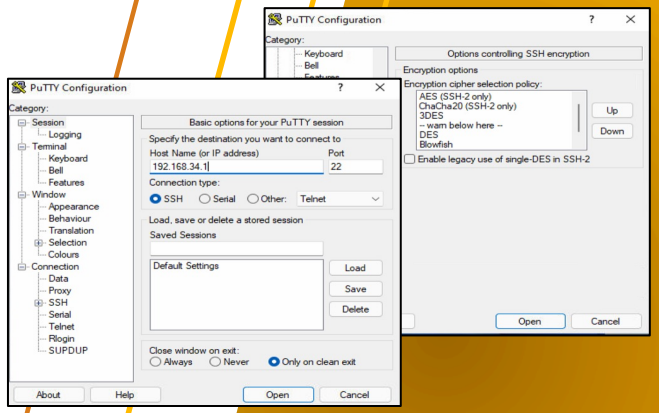


Application layer - SSH

- **SERVER in Linux:**
  - Install the OpenSSH server package:
    - **$ apt-get install openssh-server**
  - Configure the service in: **file/etc/ssh/sshd_config**: Establish port 22, specify the IP address of the server, allow root access, change the path to the private and public keys, set or restrict user(s) (and group(s)), ...
  - Activate the SSH service:
    - **$ service ssh start (restart)**
    - **$ service ssh status**

- **CLIENT:**
  - Access can be carried out by using some GUI: PuTTY, OpenSSH, …
  - but also by using CLI:
    - **$ sudo ssh user@hostname [command]**
    - **$ sudo ssh root@XXX.XXXX.XXXX.XXXX**






```
root@kali:~# sudo ssh root@192.168.1.200
root@192.168.1.200's password:
Permission denied, please try again.
root@192.168.1.200's password:
Linux kali 4.14.0-kali3-amd64 #1 SMP Debian 4.14.17-1kali1 (2018-02-16) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 24 12:54:46 2018 from 192.168.1.202
root@kali:~#
```

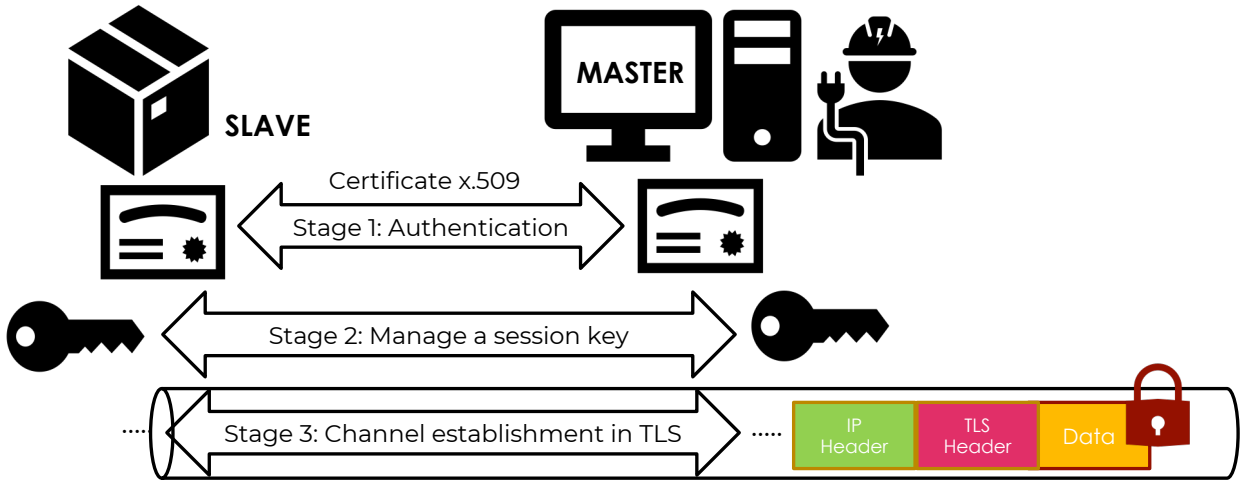
Application layer
Transport layer
Internet layer
Network Access Layer

REMOTE ACCESS IN ENERGY SYSTEMS

CyberSecPro

Transport layer – Transport Layer Security (TLS)

- TLS is a supported by the Internet Engineering Task Force (IETF), based on the **client-server** model that includes an initial process of authentication and negotiation of security credentials
  - Known as handshake phase and executed before the final connection
- This **handshake phase** entails:
  - Establishes (optional) authentication between peers using **x.509 digital certificates**
  - Negotiates a session key for encryption and a key for authentication (MAC – detailed in the next Topic)

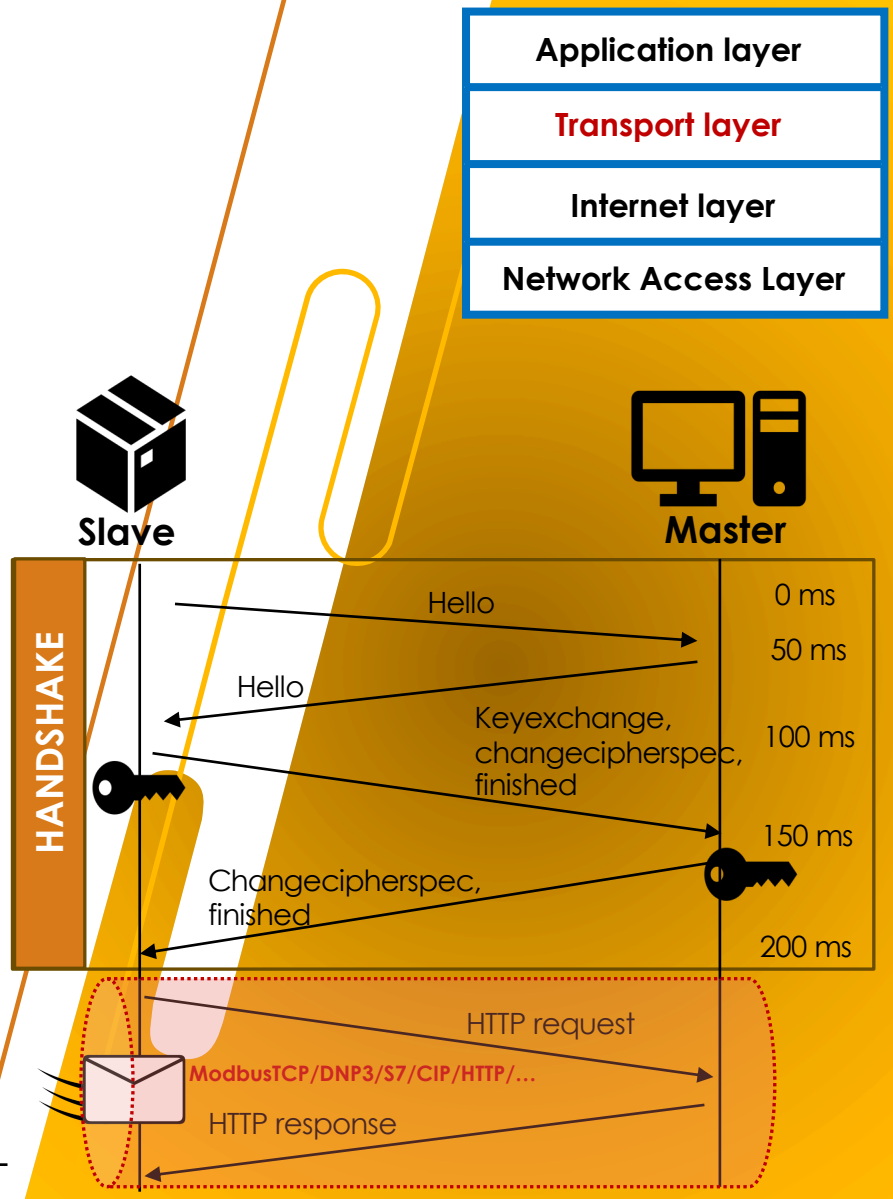


Break: Preliminary notions on digital certificates

- A digital certificate consists of an "electronic" document that accredits that the data linked to an entity (e.g., human operator, master device, slave device) is completely valid and correct
- The **X.509 certificate** is the most widely applied format today to authenticate users, and contains:
  - Serial number
  - Identifier of the user and issuer (who signs the certificate)
  - The public key
  - Expiry date
  - **Digital signature**
- This signature is emitted by a Certificate Authority (CA)
  - Who verifies the identity of a user and signs the document
  - This document includes the **public key**

Transport layer – TLS-1.2 vs. TLS-1.3

- **TLS 1.2** consists of a client-server model based on an initial set of messages corresponding to the handshake phase
- This handshake phase aims to:
  - Optionally authenticate each party and
  - Negotiate the session key (confidentiality) together with an additional key for the MAC (integrity and authentication)
- Messages in the **handshake phase**:
  - *Hello*: connection request, containing information about the security parameters (e.g. AES-128 bits) and the session ID
  - *KeyExhange*: to create the session key
  - *ChangeCipherSep*: acceptance of the security parameters
  - *Finished*: the handshake phase is finished.
- **Each message can take about 50 ms**
  - This value may be significant for those substations requiring real-time communication

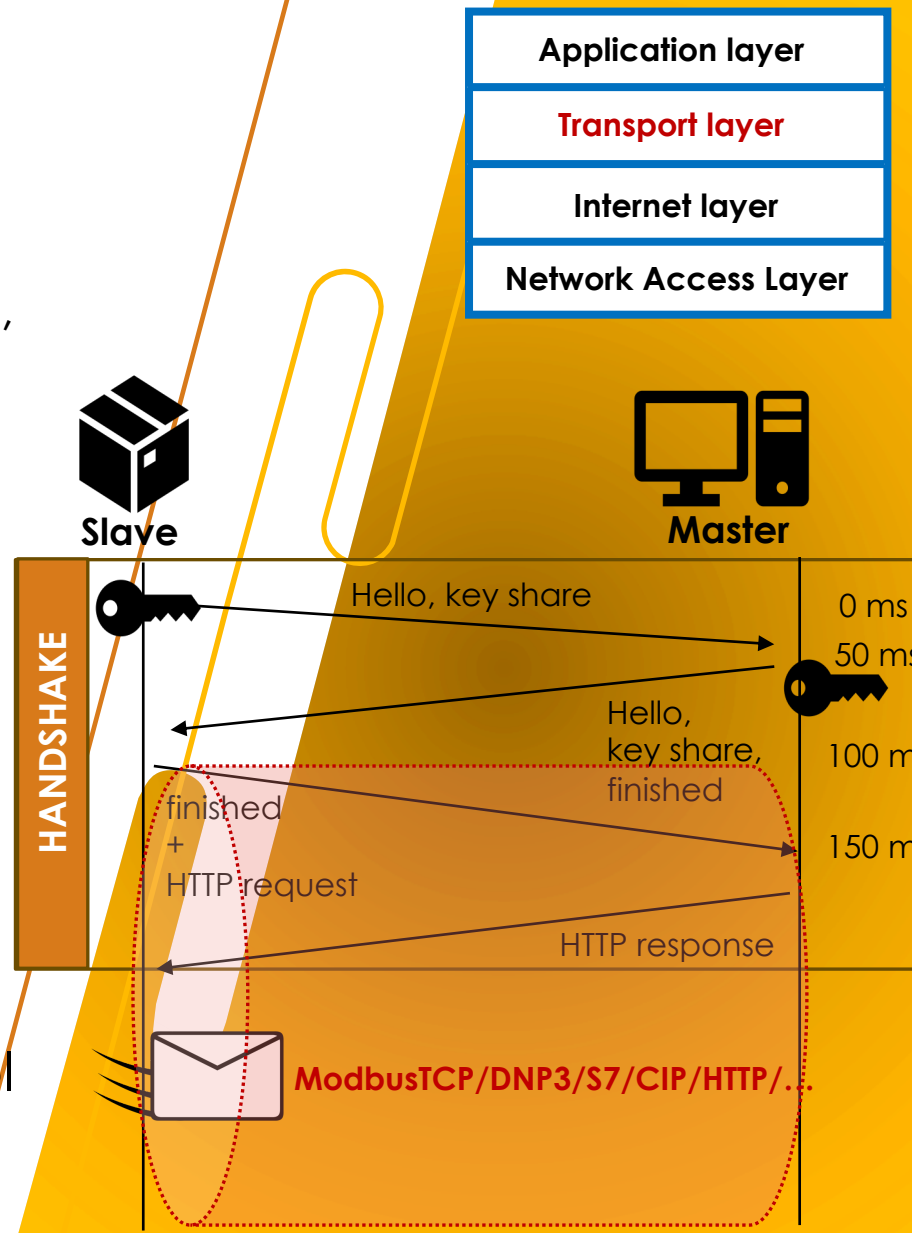


Transport layer – TLS-1.2 vs. TLS-1.3

- **TLS 1.3** is similar to TLS 1.2, presenting equivalent objectives, but reducing the number of frames during the handshake phase (3 in total), as well as the times for the final connection
- In fact, TLS 1.3 guarantees:

+Security	+Performance
• The session key is negotiated from the first frame, allowing encrypting from the second TLS frame, which can contain the digital certificates of the Master • AES, ECC, SHA-256/SHA-512, Diffie-Hellman (DH), Ephemeral DH (DHE)	• Compared to TLS 1.3, the number of frames is reduced by sending multiple messages on the same channel and reducing the handshake setup to 50ms less

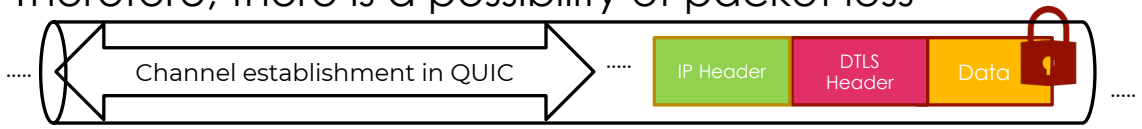
- TLS 1.3 is therefore the most suitable security protocol for critical contexts, such as energy control systems

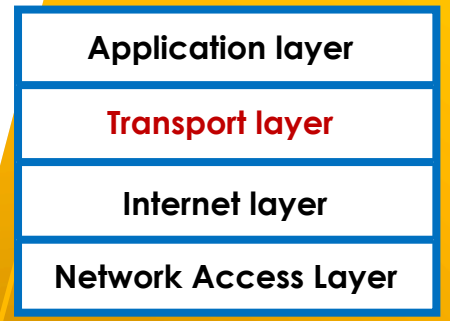


REMOTE ACCESS IN ENERGY SYSTEMS

Transport layer – Datagram Transport Layer Security (DTLS)

- DTL is also an IETF-supported protocol:
  - based on TLS, and
  - follows the client-server model operating over User Datagram Protocol (UDP)
- This also means that DTLS connections are rapid in its process, but they are not connection-oriented
  - Therefore, there is a possibility of packet loss

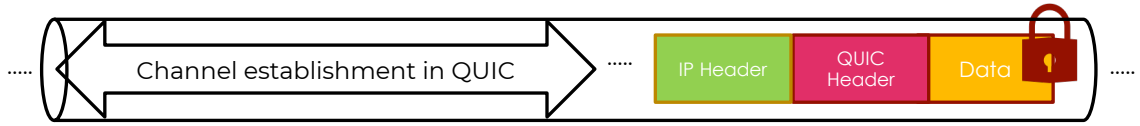




Transport layer – Quick UDP Internet Connections (QUIC)

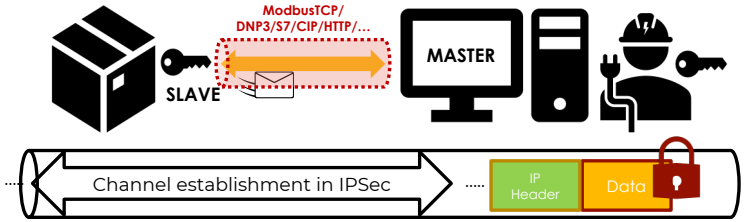
- QUIC is a protocol supported by the IETF (Internet Engineering Task Force), which further reduces client-server connections and their connection times
  - This speed is thanks to its connection via UDP, which is not connection-oriented
- In turn, QUIC is a protocol that guarantees security
  - It is based on the combination of TLS1.3 over UDP
  - This means that QUIC is based on TLS 1.3
- In other words:
  - HTTP/2: TCP (compatible with TLS 1.2 and TLS 1.3)
  - HTTP/3: UDP (compatible with QUIC)

Therefore, control systems and energy substations could optimise their communications using TLS1.3 and QUIC

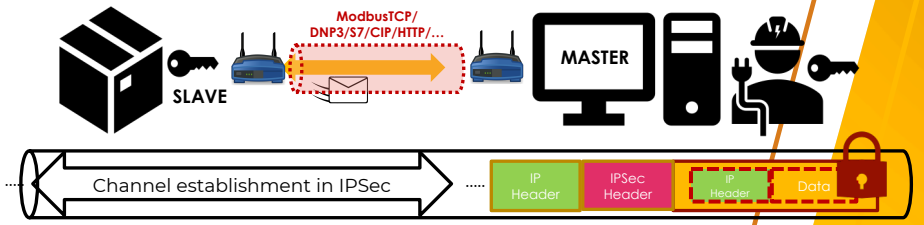


Internet layer – IPSec

- IPsec adds two ways of encapsulating the data in a secure packet:
  - **Transport mode**:
    - Between two peers (a remote connection VPN), where there is no intermediary elements for tunnelling, such as routers
    - The payload of the package is encrypted, and the entire payload and some fields of the packet header are authenticated

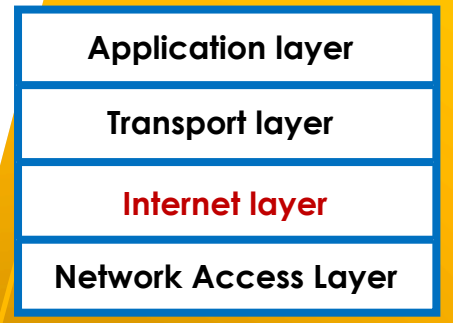


  - **Tunnel mode**:
    - Between networks (a site-to-site VPN) with intermediary elements such as routers
    - The whole packet is encrypted, generating a new header with the IPs of the routers, and the whole payload and some fields of the packet header are authenticated



Internet layer – IPSec (IKEv2)

- The configuration of the IPSEC security parameters is the first disadvantage
  - This configuration must be equivalent on both source and destination node
  - This configuration must be carried out manually
- One way to automate the configuration process is by using the **Internet Key Exchange (IKEv2) protocol**
  - IKEv2 consists of a software application with a connection to the Internet layer for the configuration of Internet parameters
  - This automatic configuration includes:
    - **Peer authentication**
    - **Negotiation of security parameters**
  - IKEv2 implements, for example:
    - DH for key negotiation
    - MAC for authentication and integrity
    - AES and 3DES for confidentiality



Final remarks

- Both control networks and substations must activate the existing security mechanism based on **security protocols running on top of the TCP/IP stack**, such as:
  - SSH
  - TLS
  - DTLS
  - QUIC
  - IPSec
- All these protocols are relevant for critical environments, mainly because they are able to **protect sensitive data and ensure confidentiality, authentication and integrity**
  - However, not all of them are equally effective for restricted environments where it is required (almost) real-time communication
  - In this sense, the most characteristic protocols are: TLS1.3, QUIC, and IPSec

References and sources

1. ENISA, "Appropriate security measures for smart grids. Guidelines to assess the sophistication of security measures implementation", 2012
   URL: https://www.enisa.europa.eu/publications/appropriate-security-measures-for-smart-grids
2. NIST, CSRC
   URL: https://csrc.nist.gov/glossary
3. IETF, IP Security Protocol (IPSec)
   URL: https://datatracker.ietf.org/wg/ipsec/about/
4. IETF, IP Security (IPSec) and Internet Key Exchange (IKE) Document Roadmap
   URL: https://datatracker.ietf.org/doc/html/rfc6071
5. IETF, A Survey of Transport Security Protocols URL: https://datatracker.ietf.org/doc/html/draft-pauly-taps-transport-security-01
6. IETF, A Survey of the Interaction between Security Protocols and Transport Services
   URL: https://datatracker.ietf.org/doc/html/rfc8922
7. DeepL Translator for Proofreading:
   https://www.deepl.com/translator

Connect with CyberSecPro:
How to register and other practical information

1. Website: www.cybersecpro-project.eu
2. X (Twitter): https://twitter.com/CyberSecPro_eu
3. LinkedIn: https://www.linkedin.com/company/cybersecpro-euproject/

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

Thank you

If you have any questions, please do not hesitate to contact:

- Antonio Muñoz
  Associate Professor
  University of Malaga
  anto@uma.es