Fonds Européen de Développement Régional (FEDER)
Europäischer Fonds für Regionale Entwicklung (EFRE)

Dépasser les frontières : projet après projet
Der Oberrhein wächst zusammen, mit jedem Projekt

SMI
Smart Meter Inclusif

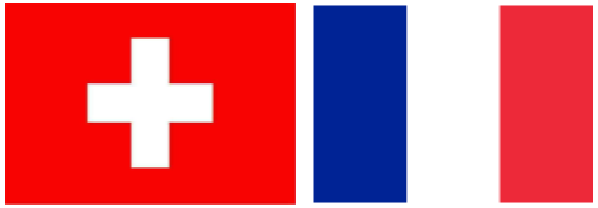
WP4: Security concepts for distributed Smart Grids

4.1. Comparative security analysis
4.2. Penetration testing of Smart Meters

Hochschule Offenburg ivESK

7 décembre 2022

Architecture requirements



LOW < architecture requirements < STRICT

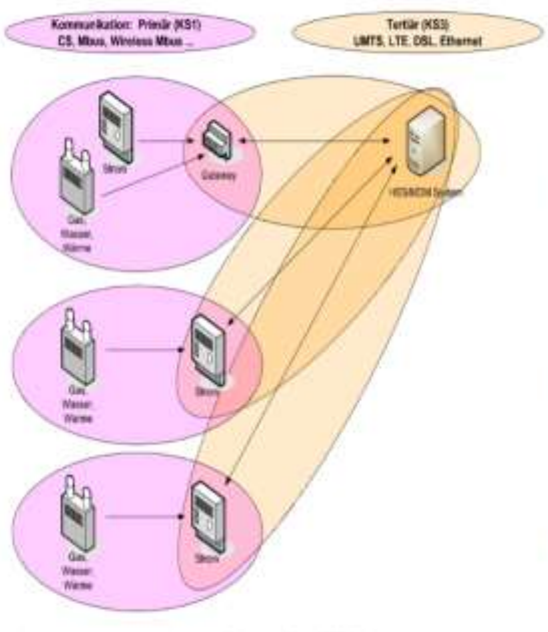


Abbildung 4: Kommunikation Point to Point (P2P) direkt

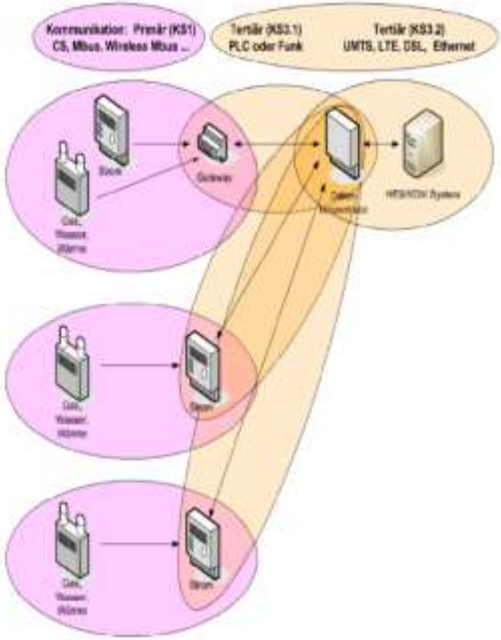
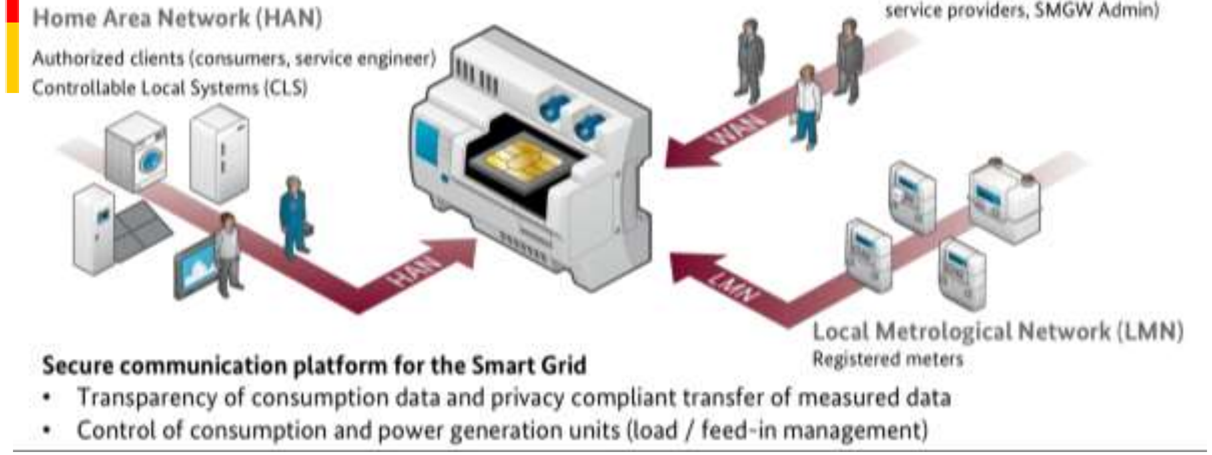


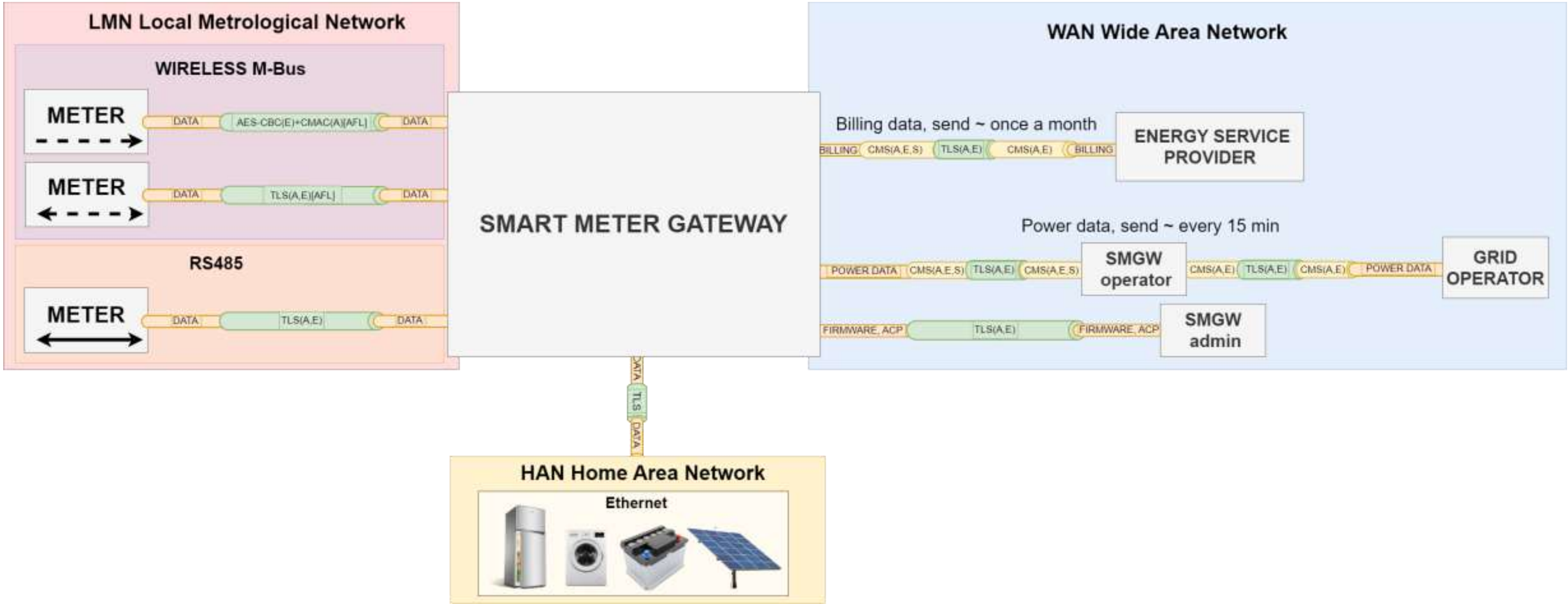
Abbildung 5: Kommunikation Point to Multipoint (P2MP)

Federal Office for Information Security

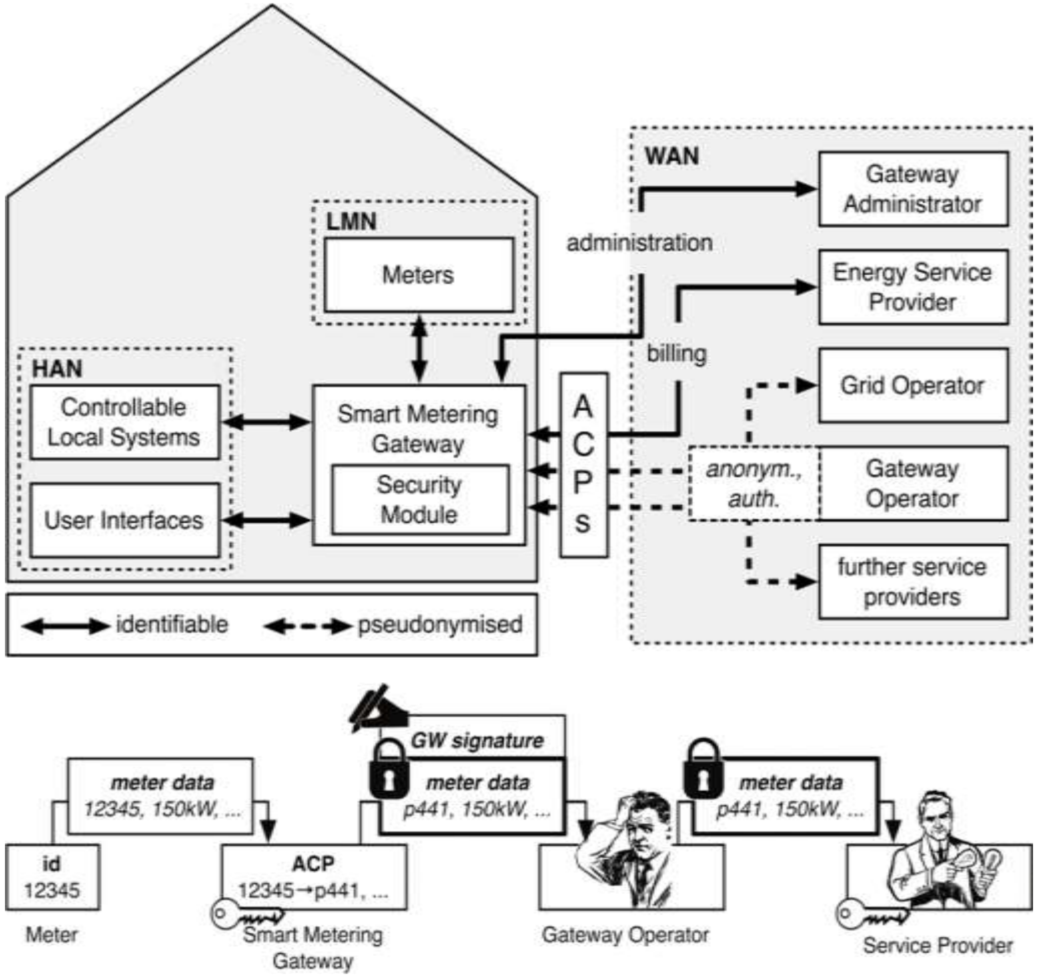
TR-03109-1

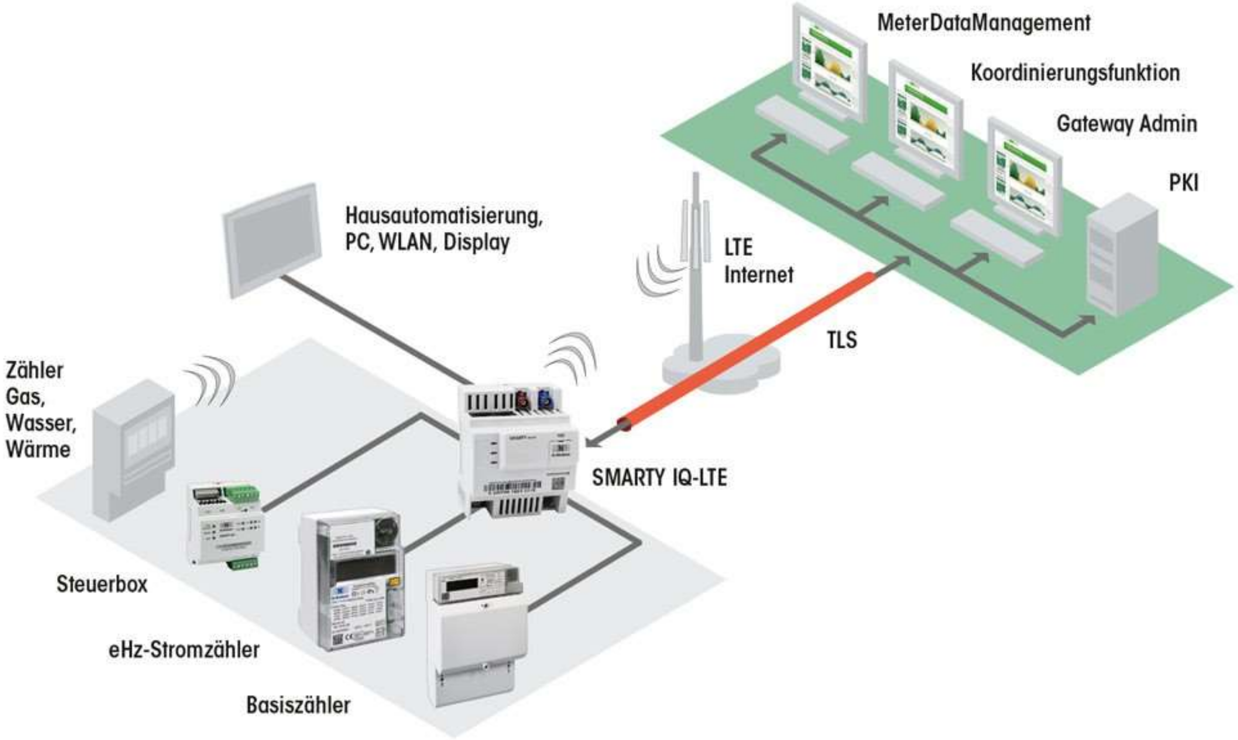


German BSI protocols requirements

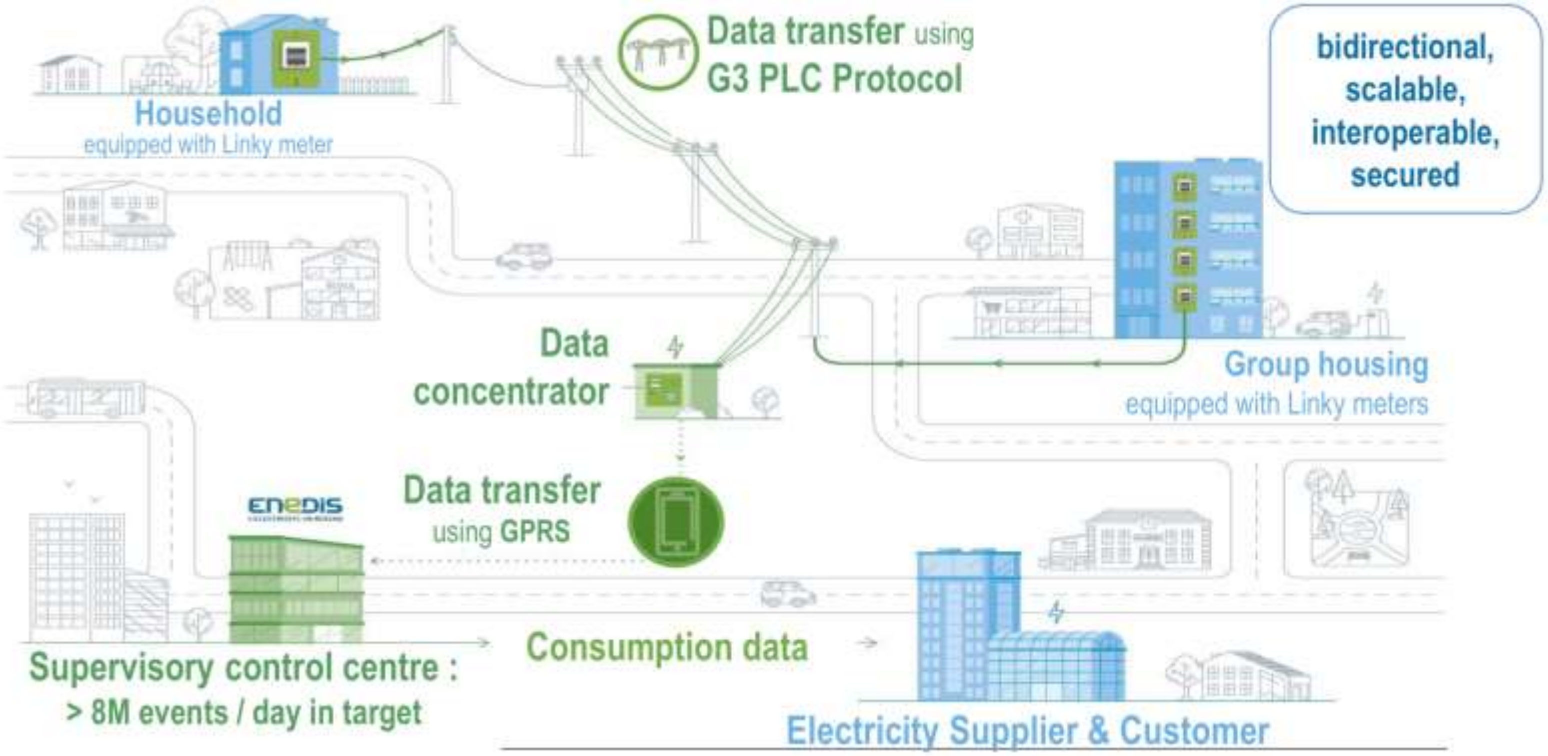


German BSI data processing requirements and common architecture





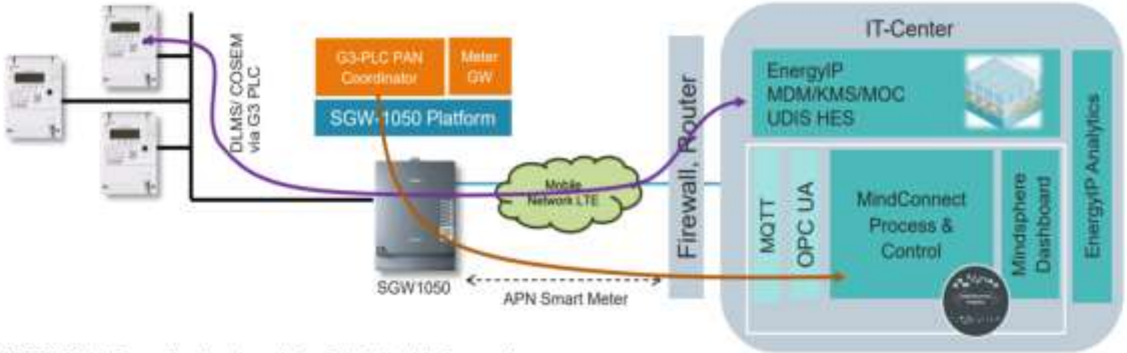
French Enedis architecture



DLMS/COSEM

Swiss Siemens / Landis+Gyr architecture

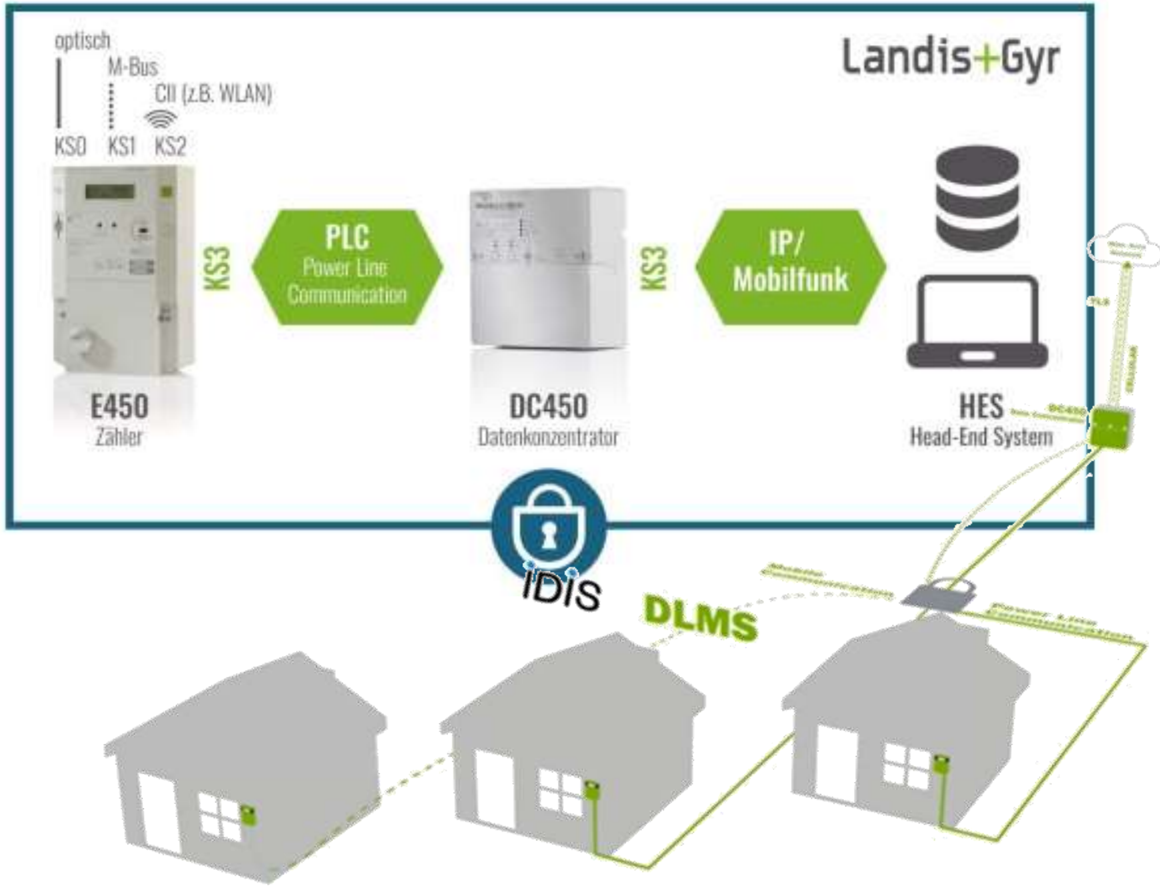
SIEMENS



G3-PLC **PAN Coordinator has lot of detailed information**:

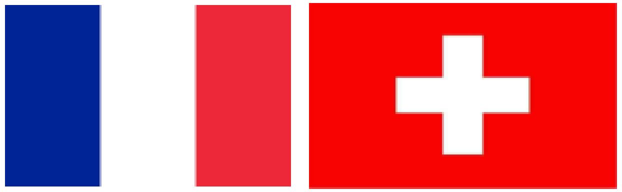
- Amount of connected Smart Meter
- Communication topology
- Number of hops and quality for communication link
- Percentage of successful / missed communication attempts

Landis+Gyr



DLMS/COSEM

DLMS/COSEM and BSI architecture comparison



DLMS/COSEM

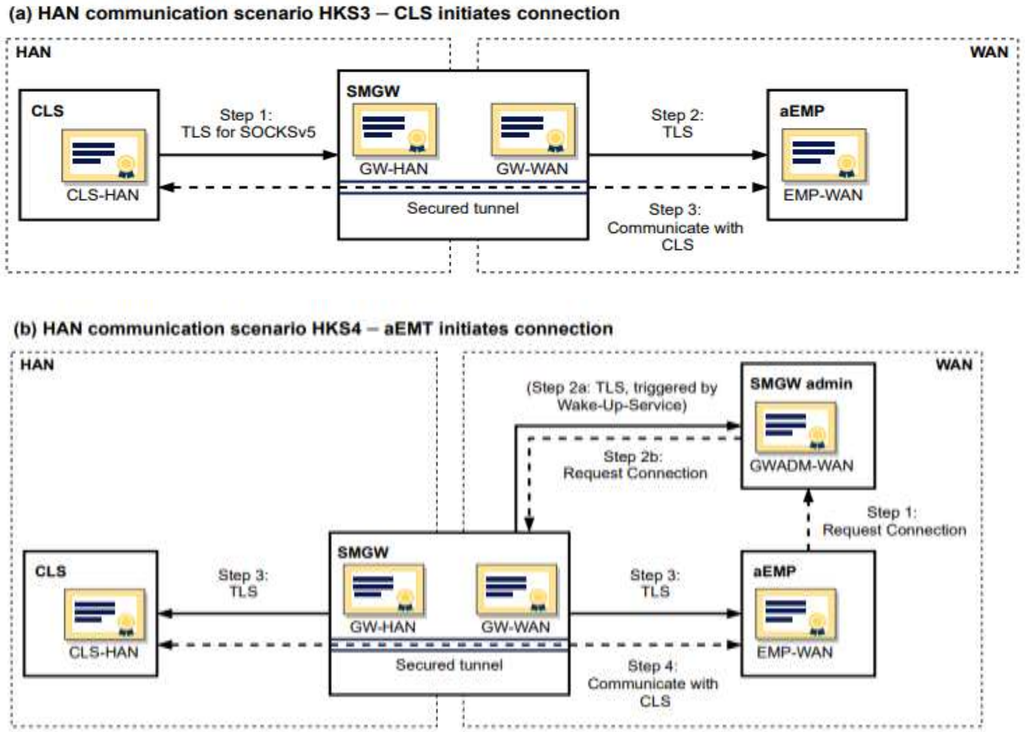
ENERGY SERVICE PROVIDER

COMMONLY

IMPLEMENT CLIENT

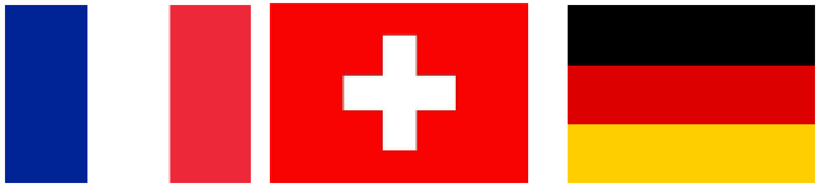
SMART METER

COMMONLY
IMPLEMENT SERVER



Security protocols comparison

	DLMS™	TLS
PKI	+	+
Modern crypto core algorithms	+	+
Protocol type	special	common
Complexity	low	high



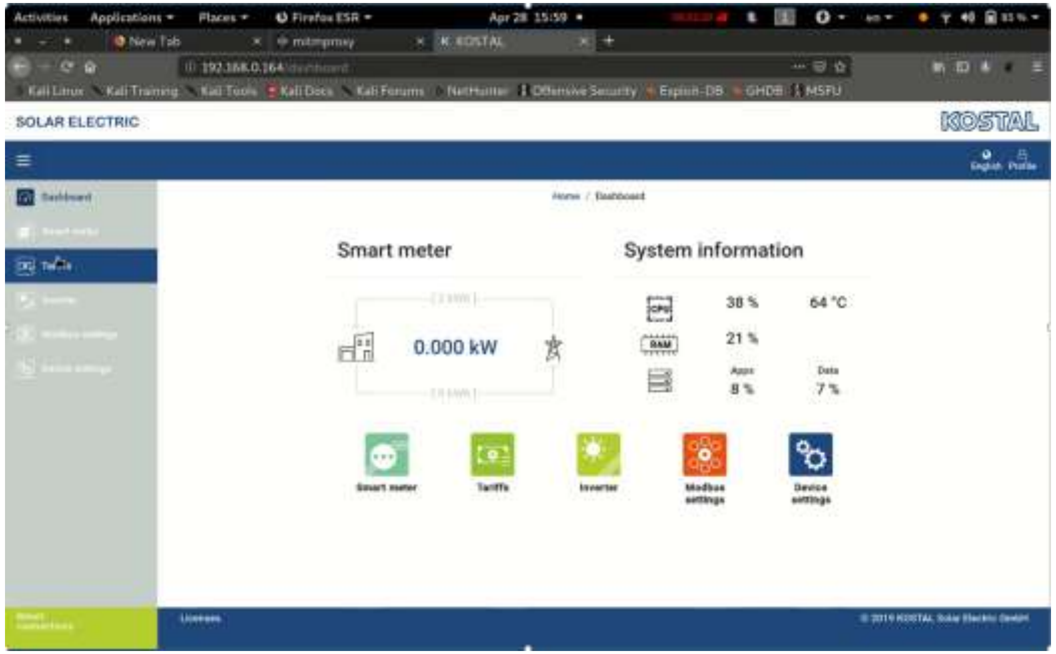
	DLMS/COSEM	SMGW/TLS
Authenticated encryption	AES-GCM-128 AES-GCM-256	AES-GCM-128 AES-GCM-256 AES-CBC-128 AES-CBC-256
Elliptic curves	NIST P-256 NIST P-384	NIST P-256 NIST P-384 BrainpoolP256rl BrainpoolP384rl BrainpoolP512rl
Digital signature	ECDSA	ECDSA
Key agreement	ECDH	ECDHE
Key transport	AES key wrap	AES key wrap
Hash function	SHA-256 SHA-384	SHA-256 SHA-384
Message Authentication Code	GMAC	CMAC

Testing devices selection and acquisition

SMI
Smart Meter Inclusif

ENEDIS
L'ELECTRICITE EN RESEAU

Bundesnetzagentur



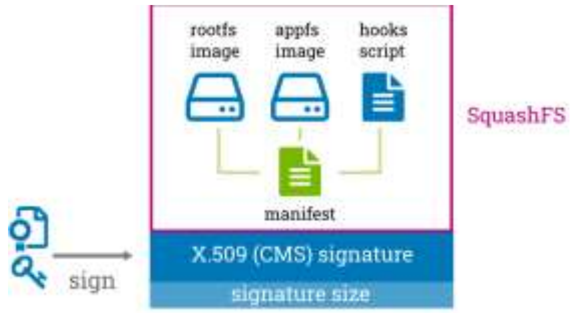
Kostal Smart Energy meter

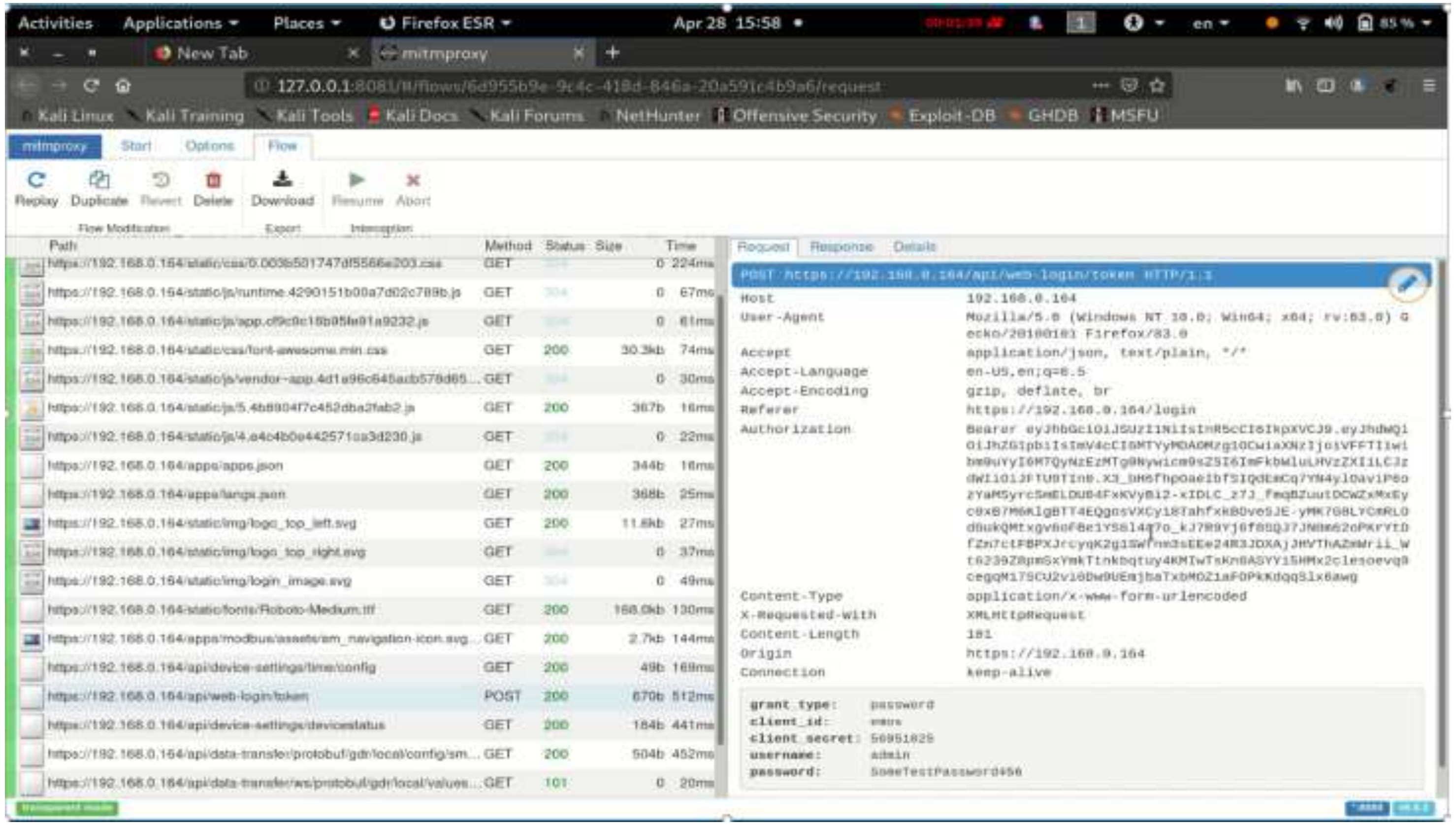
Pros:

- Use RAUC - Safe and Secure OTA Updates for Embedded Linux.
- Web server use stable version of Nginx 1.15.7 which currently have no publicly known vulnerabilities.
- Stable implementation of authentication token (JWT)
- Was not found some vulnerabilities by Greenbone OpenVAS, Nikto, Burp suite (incl. spider and burp intruder testing) and OWASP ZAP.

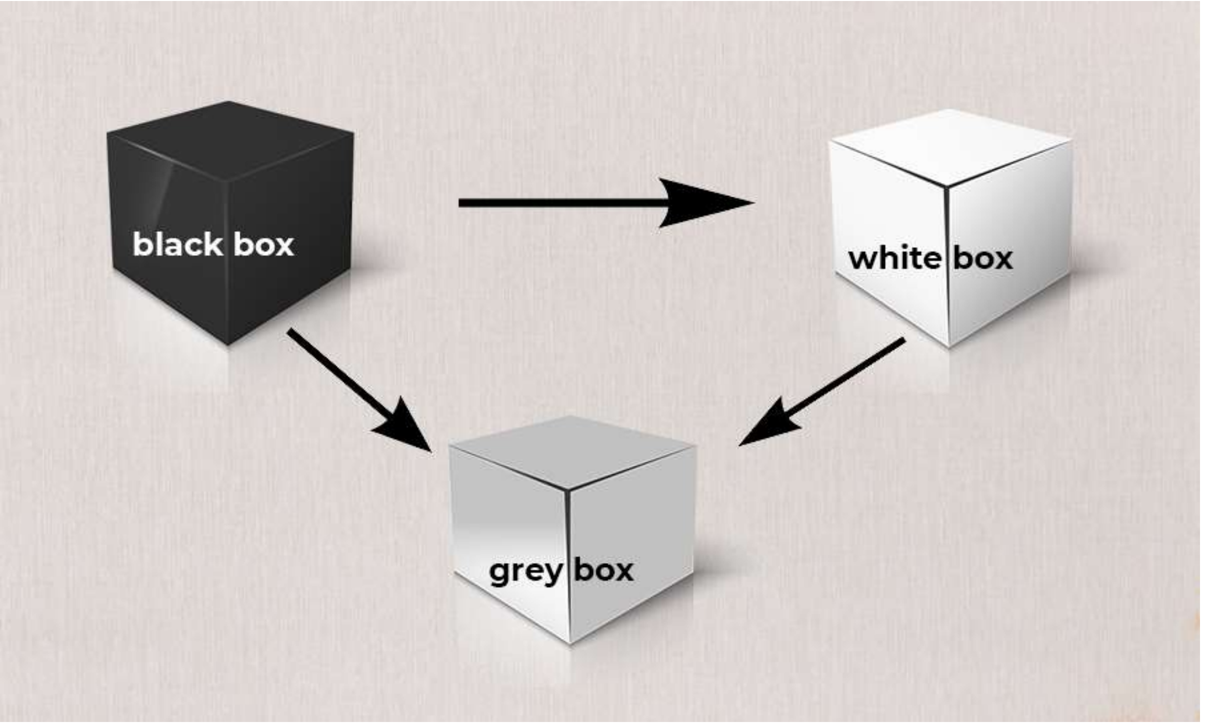
Cons:

- By default use HTTP instead of HTTPS
- No force redirect to HTTPS version
- Use self-signed TLS certificate.
- In time if user use HTTP (or do not add device certificate to trusted storage) MITM attack in conjunction with ARP spoofing can be easily implemented to intercept password which was shown on our master class (probably hacker will get access to admin panel, but will not be able to get shell on device)





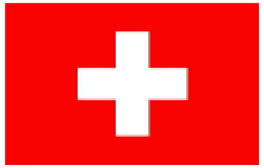
White, Black and Grey box testing



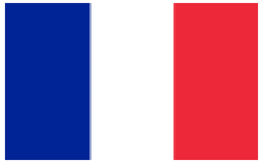
KOSTAL

- Administrator/root rights
- Shell on testing device
- Source code / firmware available

- Documentation with used protocols and communication scenarios



- Some comments for our questions



- Only user documentation and marketing materials

Landis+Gyr E470

Features:

- Communications using DLMS/COSEM over the Wide Area Network (WAN) to a Head End System (HES)
- ‘Over the Air’ firmware upgrades.
- Standard meter, power fail, fraud detection and contractor control event logs;
- ZigBee Smart Energy Profile to communicate with other devices such as an In Home Display Unit and for communication with the External Communications Hub via a Home Area Network (HAN)
- Capable of showing messages from the utility on the meter display.

Attack ideas (motivated by known vulnerabilities in certain DLMS/COSEM implementations):

- Open source fuzzer ValiDLMS
- Security Downgrade
- Vulnerable Authentication Methods.
- Possibility to manipulate the security byte of messages
- Etc.

R&S CMW500

Landis+Gyr E470 testing

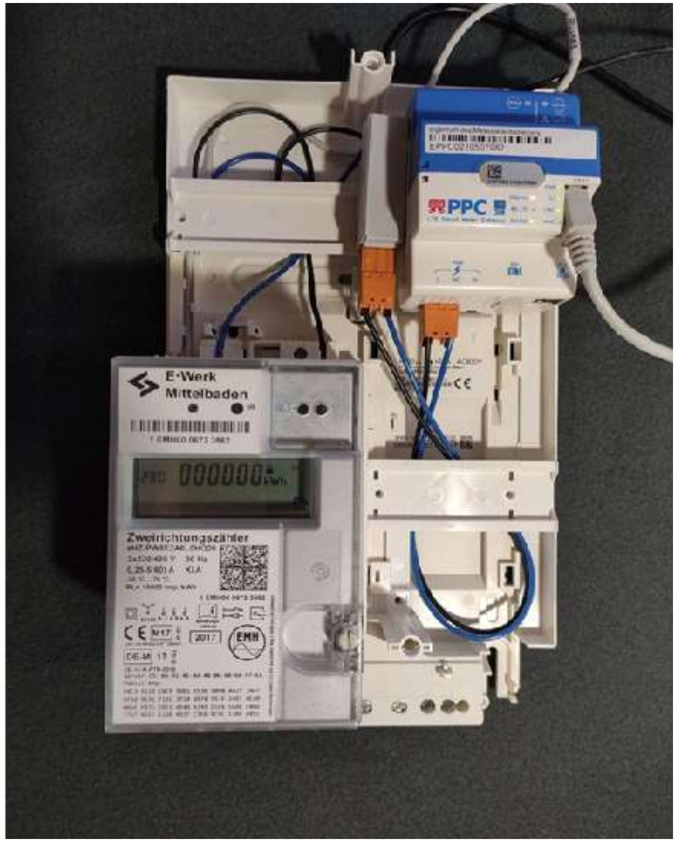
SMI
Smart Meter Inclusif

Ports:

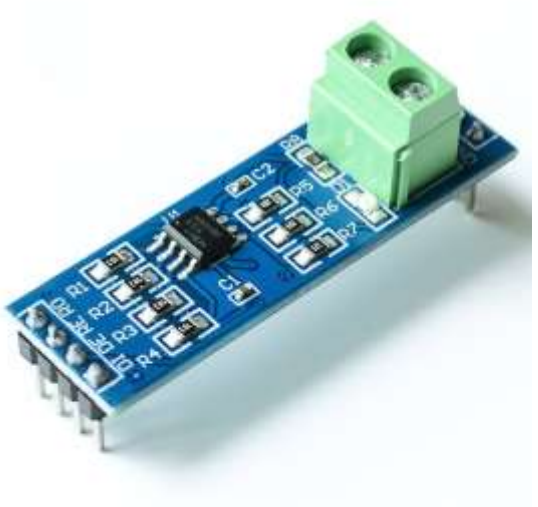
- GPRS WAN communication for DLMS/COSEM
  - Use R&S CMW500 Wideband Radio Communication Tester
- optical interface IEC 62056-21
  - Use weidmann-elektronik USB infrared read/write head
  - Try different software/libs and initial codes
- ZigBee interface
  - Use CC2351 with alternative firmware

German SMGW's and smart meters acquisition

SMI
Smart Meter Inclusif



E-Werk
Mittelbaden



LMN MAX485 board

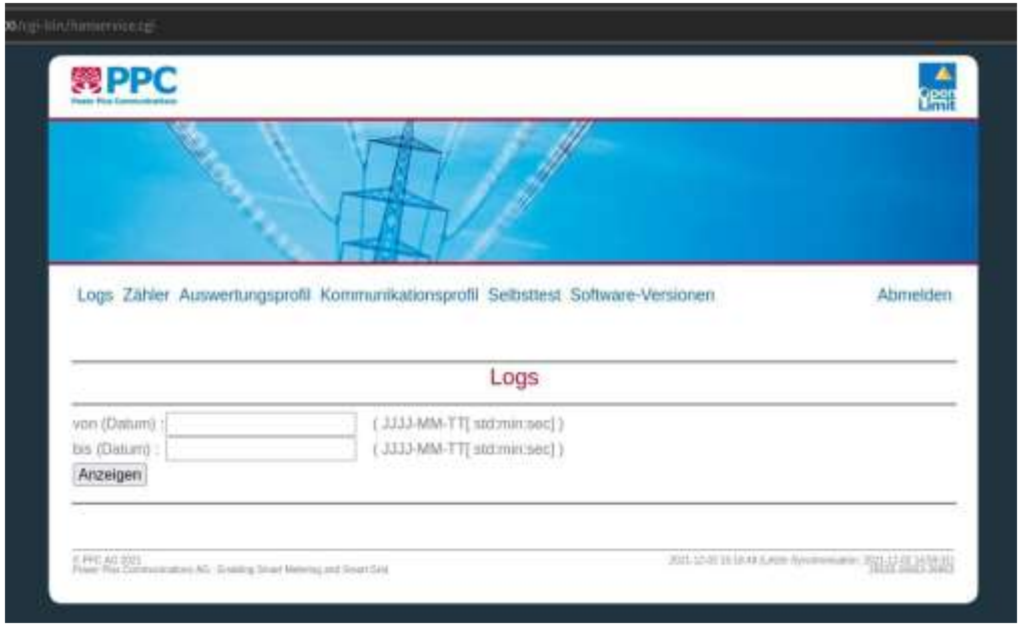
PPC SMGW testing

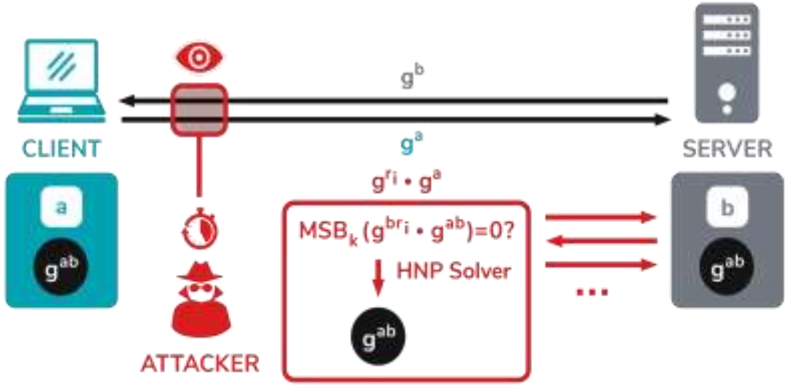
Pros:

- Was not found some vulnerabilities by Greenbone OpenVAS, Nikto, Burp suite (incl. spider and burp intruder testing) and OWASP ZAP.
- Minimalistic web server (not so much possibilities for user input = not so much things to test).
- Stably react on all tested exploits including TLS certificates with buffer overflow.
- Was not found some problems with fuzzing

Cons:

- Slow CGI (common gateway interface) based web server.
- Was found SSH server with possibility of password authentication and vulnerable for user enumeration (but by our data it is presented only in the test firmware)
- Exploit for this vulnerability was checked on the same version of software running in raspberry pi, which later allow to get a list of user names from SMGW.
- SSH password brute force was unsuccessful even with known usernames





Conexa SMGW testing

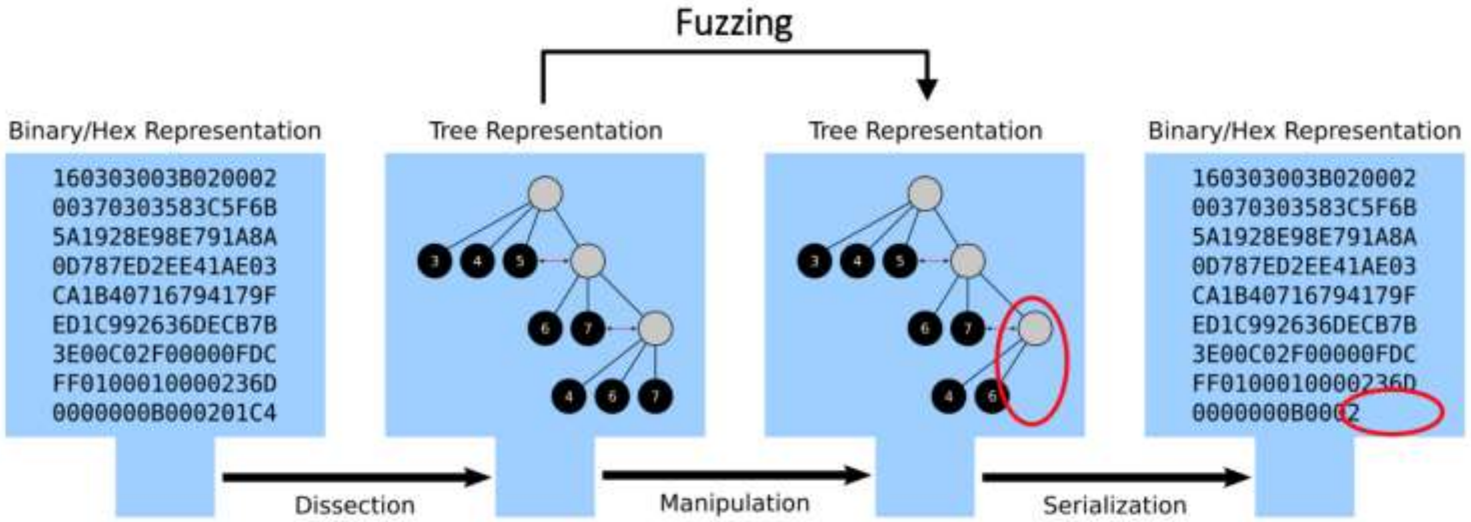
Pros:

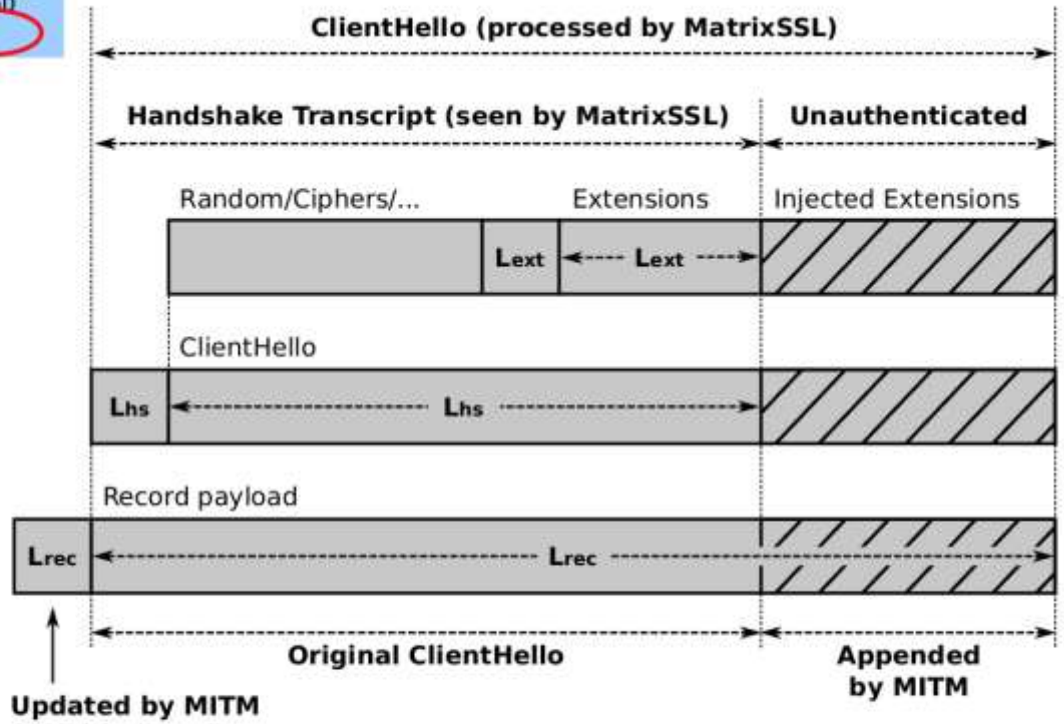
- Even more minimalistic web server than in PPC SMGW.
- Was not found some vulnerabilities by Greenbone OpenVAS, Nikto, Burp suite (incl. spider and burp intruder testing) and OWASP ZAP.
- Stably react on all tested exploits including TLS certificates with buffer overflow.
- HKS3 does not accept other authentication methods.
- Have an additional TCP-Wrapper security mechanism which makes fuzzing more complicated. (after some numbers of incorrect TLS connections stop responding before SMGW reboot).
- Was not found some problems with fuzzing.

Cons:

- Was not found during our testing

About fuzzing and fuzzers



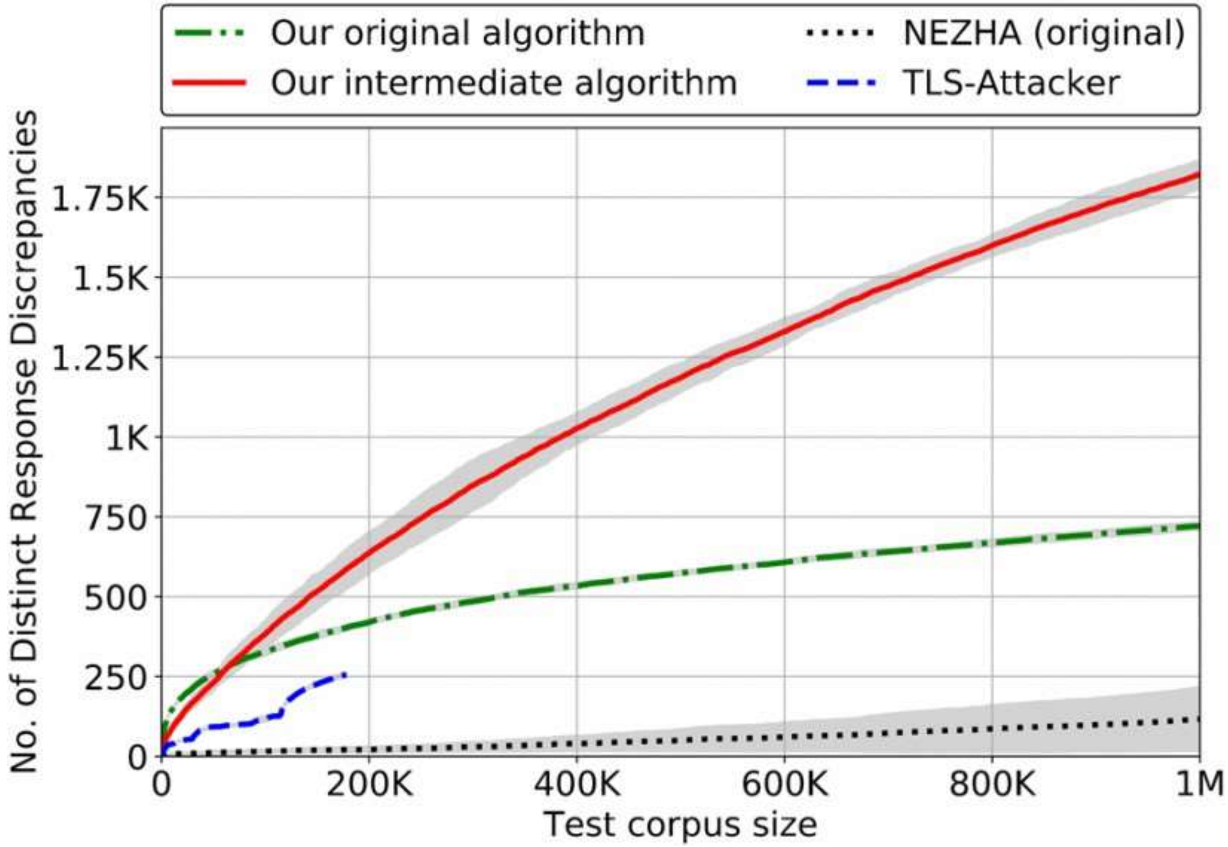


In this work for testing we use such fuzzers as:

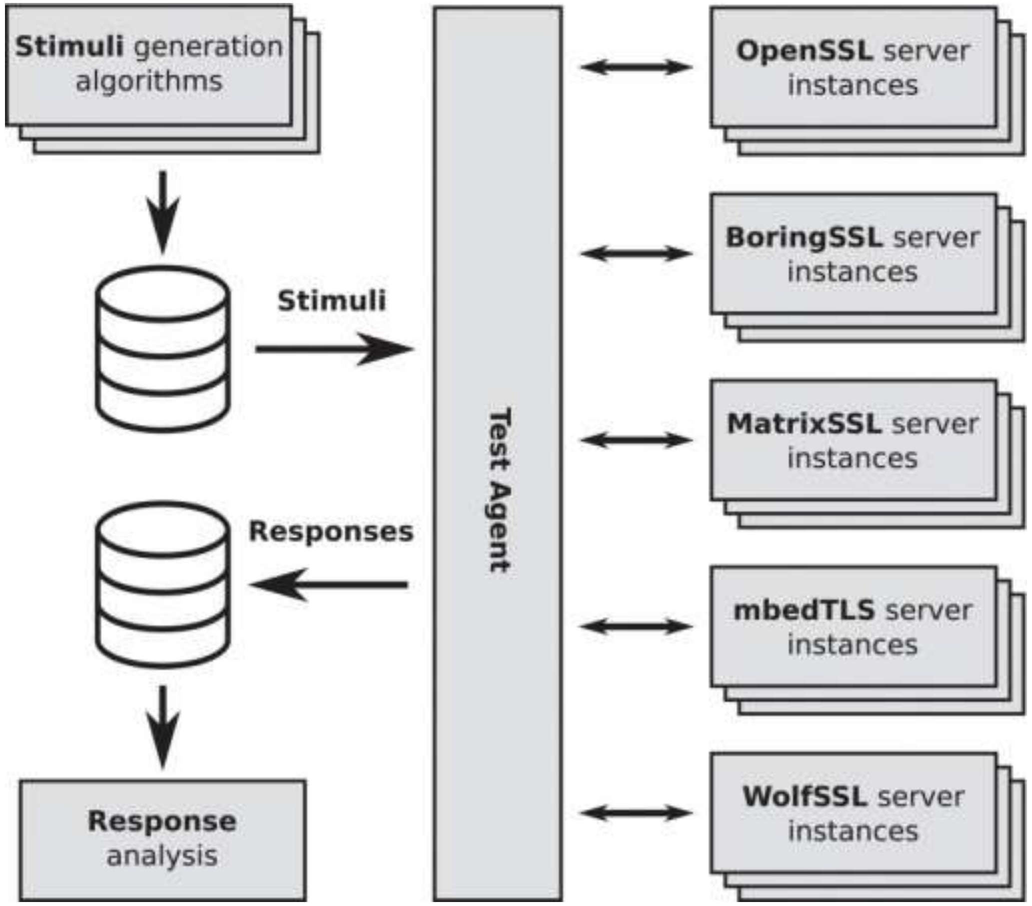
- Python tlsfuzzer
- Java TLS attacker
- Our own fuzzer

Commonly fuzzer require some precious setup like choose which fields of message should be fuzzed, with which operators and etc. In case of our fuzzer – all what is required is a final number of a stimuli messages and start “template” TLS message.

Advantages of our TLS fuzzer



Paper: “Maximizing and Leveraging Behavioral Discrepancies in TLS Implementations using Response-Guided Differential Fuzzing”



How to understand that something wrong?

We do not have access to testing device shell, but we are able to check:

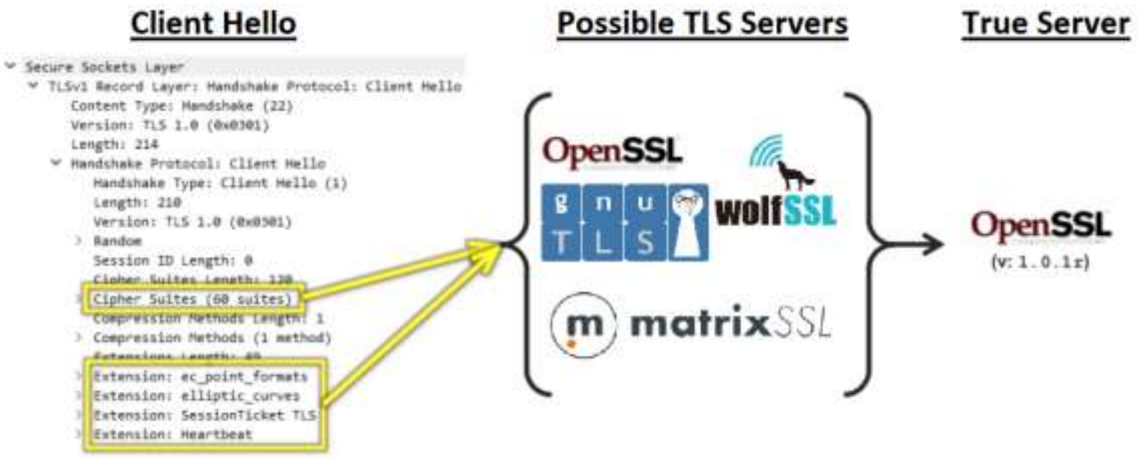
- Device Web server does not respond
- Device TLS server acts different in comparison with other TLS servers
- Device TLS server does not respond
- Device does not respond on TCP layer
- Physical – interface is down / non-standard LEDs blinking

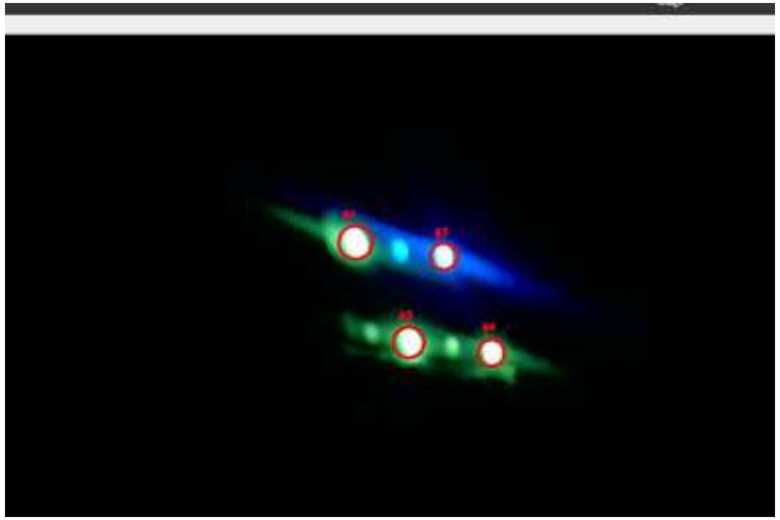
LEDs blinking described in user manual. Use python OpenCV script to automatize.

Problems:

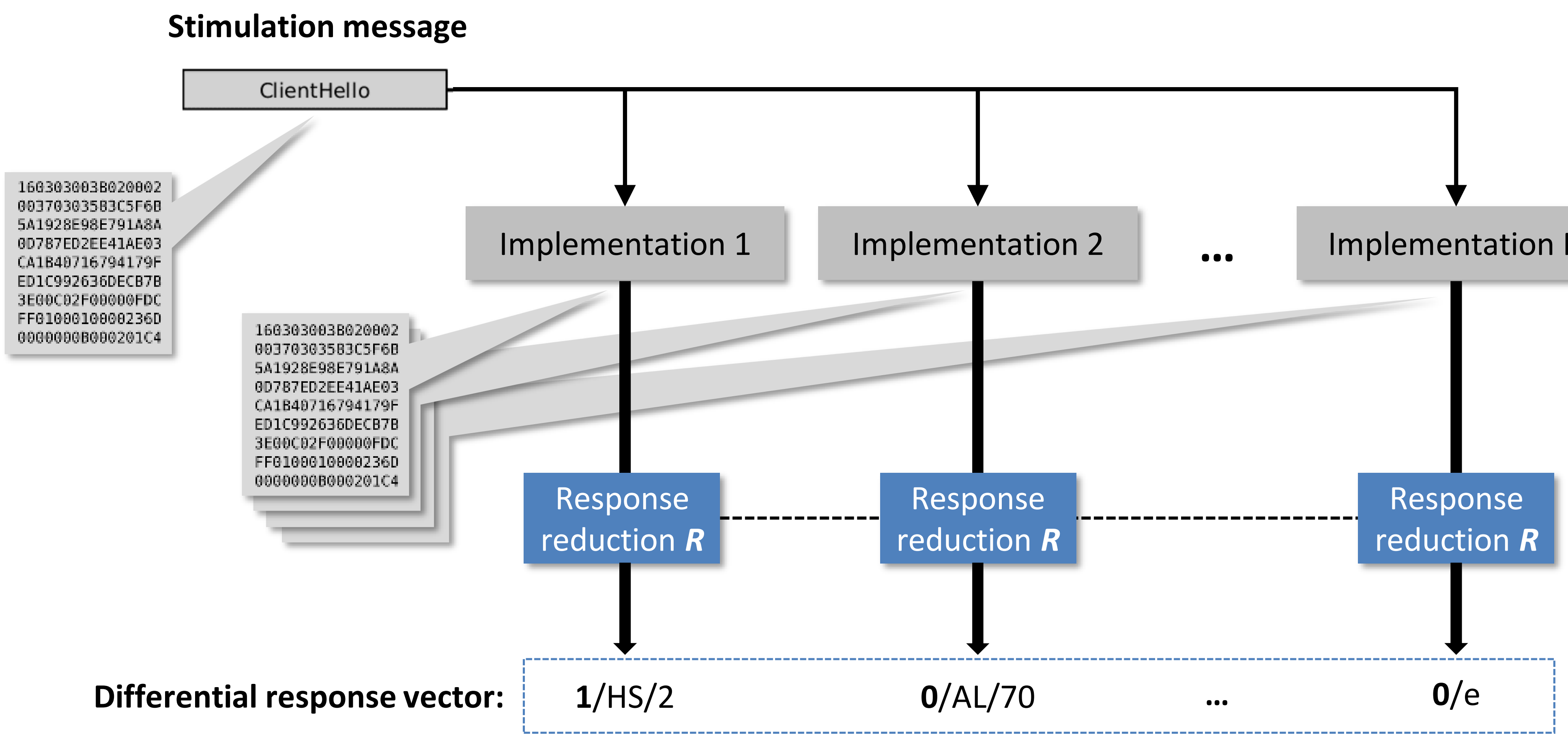
- No fully understanding what's going on because no shell
- Testing speed

Solution – Run locally. To be able to run locally – need to get implementation and version.





How work our TLS fingerprinting approach?



TLS doppelganger software

```
  Supported Server Cipher(s):
Preferred TLSv1.2  128 bits  ECDHE-ECDSA-AES128-GCM-SHA256 Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-ECDSA-AES256-GCM-SHA384 Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  ECDHE-ECDSA-AES128-SHA256     Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-ECDSA-AES256-SHA384     Curve P-256 DHE 256

  Server Key Exchange Group(s):
TLSv1.2  192 bits  secp384r1 (NIST P-384)

  SSL Certificate:
Signature Algorithm: ecdsa-with-SHA256
ECC Curve Name:      secp384r1
ECC Key Strength:    192

Subject:  ETHE0300009055.SMGW
Altnames: DNS:ethe0300009055.sm
Issuer:   ETHE0300009055.SMGW
```

- Our fingerprinting research shows that different TLS server parameters can lead to bigger number of distinctions between the same implementation than different implementations with the same parameters.
- TLS doppelganger software allows to automatize creation of docker images of different versions of different TLS implementations with required TLS parameters.

```
  Supported Server Cipher(s):
Preferred TLSv1.2  256 bits  ECDHE-ECDSA-AES256-GCM-SHA384 Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-ECDSA-AES256-SHA384     Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  ECDHE-ECDSA-AES128-GCM-SHA256 Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  ECDHE-ECDSA-AES128-SHA256     Curve P-256 DHE 256

  Server Key Exchange Group(s):
TLSv1.2  128 bits  secp256r1 (NIST P-256)

  Server Signature Algorithm(s):
TLSv1.2  ecdsa_secp256r1_sha256
TLSv1.2  ecdsa_secp384r1_sha384
TLSv1.2  ecdsa_secp521r1_sha512

  SSL Certificate:
Signature Algorithm: ecdsa-with-SHA256
ECC Curve Name:      prime256v1
ECC Key Strength:    128

Subject:  EPPC0210507982
Altnames: othername:<unsupported>
Issuer:   [illegible]
```

- In case of Conexa SMGW because of TCP wrapper protection results of fingerprinting is not very clear.
- PPC SMGW most likely use LibreSSL with version in range 2.8.0-3.1.2

Current results

- A comparative security analysis was done.
- No critical vulnerabilities were found in the tested devices.
- Created TLS Doppelganger software, which generates Docker images of different versions of different TLS implementations with required TLS parameters.
- Created TLS fingerprinting software.

Hochschule Offenburg
offenburg.university

Ivan Rigoev

Scientific Employee
Institute of Reliable Embedded Systems and Communication Electronics

Telefon +49 (0)781 205-4717
Fax +49 (0)781 205-45 4717
ivan.rigoev@hs-offenburg.de

Badstraße 24
77652 Offenburg
www.hs-offenburg.de

Hochschule Offenburg
offenburg.university

Prof. Dr.-Ing.
Axel Sikora Dipl.-Ing. Dipl.-Wirt.-Ing

Scientific Director
Institute of Reliable Embedded Systems and Communication Electronics

Telefon +49 (0)781 205-416
Fax +49 (0)781 205-45 416
axel.sikora@hs-offenburg.de

Badstraße 24
77652 Offenburg
www.hs-offenburg.de

Hochschule Offenburg
offenburg.university

Andreas Walz

Scientific Employee
Institute of Reliable Embedded Systems and Communication Electronics

Telefon +49 (0)781 205-4803
Fax +49 (0)781 205-45 4803
andreas.walz@hs-offenburg.de

Badstraße 24
77652 Offenburg
www.hs-offenburg.de