



Securing Light Source SCADA Systems

Leonce Mekinda, Valerii Bondar, Sandor Brockhauser,
Cyril Danilevski, Wajid Ehsan, Sergey Esenov, Hans Fangohr, Gero Flucke, Gabriele Giovanetti, Steffen
Hauf, David Gareth Hickin, Anna Klimovskaia, Luis Maia, Thomas Michelat, Astrid Muennich, Andrea
Parenti, Hugo Santos, Kerstin Weger, Chen Xu.

European XFEL GmbH

Barcelona, 12/10/2017

Overview

- The security of SCADA systems is an increasing concern as they interconnect a significant number of COTS computers via IP networks; support *de facto* standards like USB.
- What happens once attackers have been granted access to / broke into the Control Network?
 - Can they do everything?
 - Can they easily escalate their privileges?
- “We trust whoever has access to the Control Network”
 - Would you let your personal laptop unlocked 24/7 in a control room? If no, why should the control system be less protected than your laptop?
- We suggest to secure the SCADA system beyond the general IT infrastructure security
 - Device servers would authenticate and authorize users for every issued message.

The European X-ray Free Electron Laser

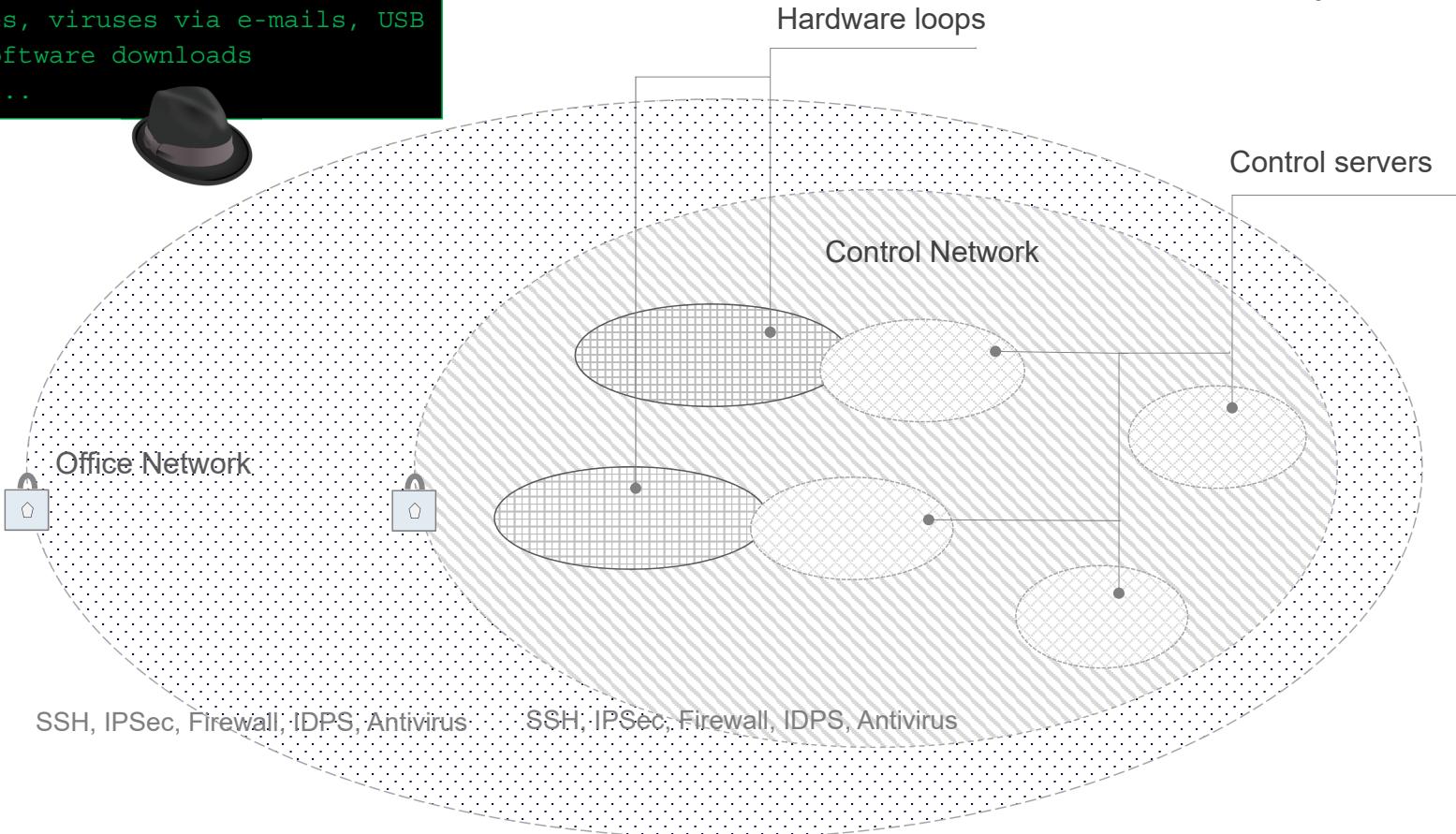
- The most brilliant X-ray light source: 5×10^{33} photons s⁻¹ mm⁻²mrad⁻² per 0.1% bandwidth
- 4.5 MHz pulse rate in burst mode, maximum electron energy of 17.5 GeV, 0.05 nm minimum wavelength
- 1.4 billion euro facility
- Unique pieces of technology: Adaptive Gain Integrating Pixel Detector, Large Pixel Detector, DEPSET Sensor with Signal Compression
- 15 TB of data per beam day



LPD detector in the FXE instrument hutch

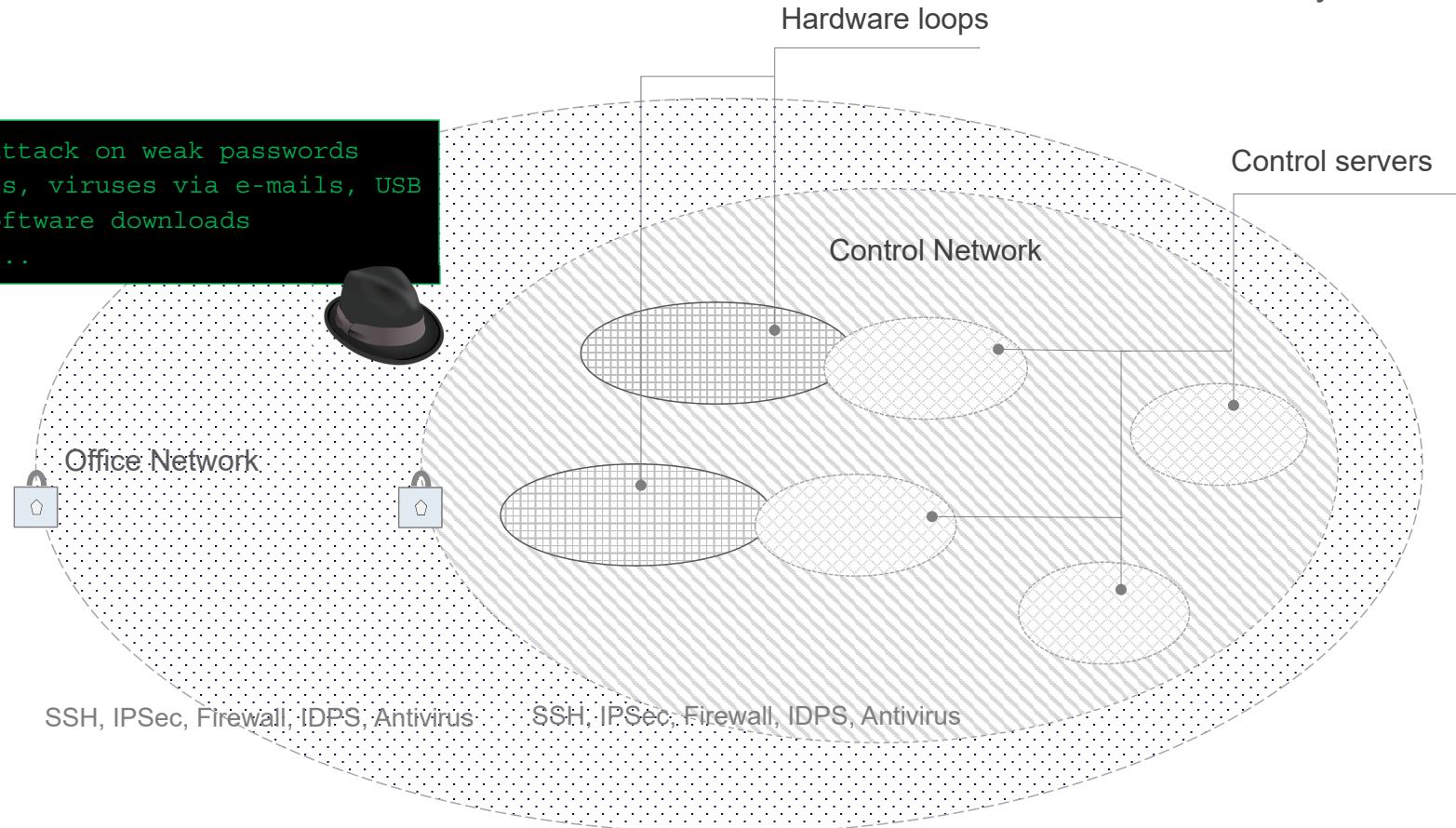
The security onion

```
# Dictionary attack on weak passwords  
# Trojan horses, viruses via e-mails, USB  
sticks or software downloads  
# Keyloggers ...
```



The security onion

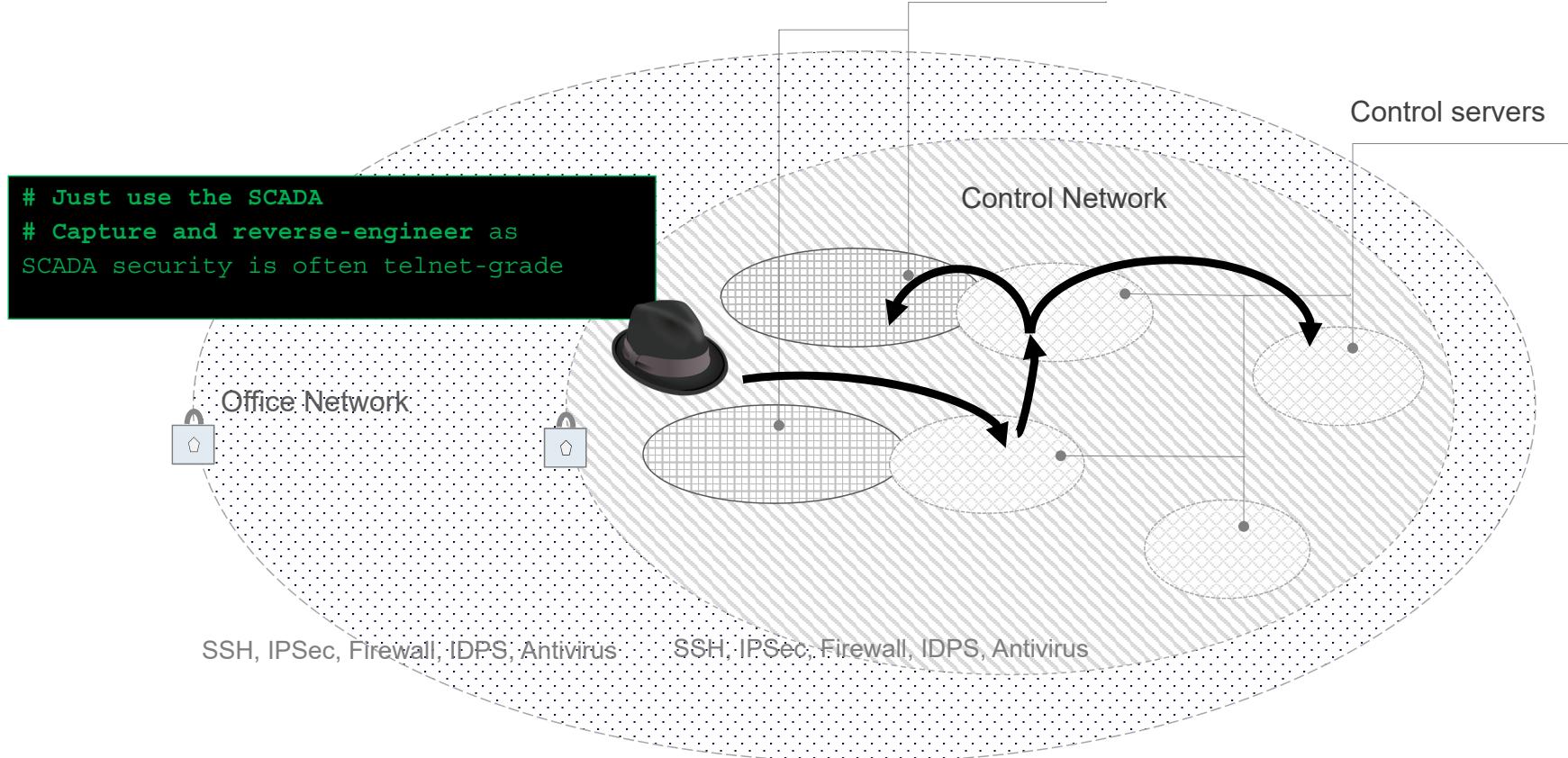
```
# Dictionary attack on weak passwords  
# Trojan horses, viruses via e-mails, USB  
sticks or software downloads  
# Keyloggers ...
```



Any random compromised account often suffices.

→ SCADA messaging

The security onion

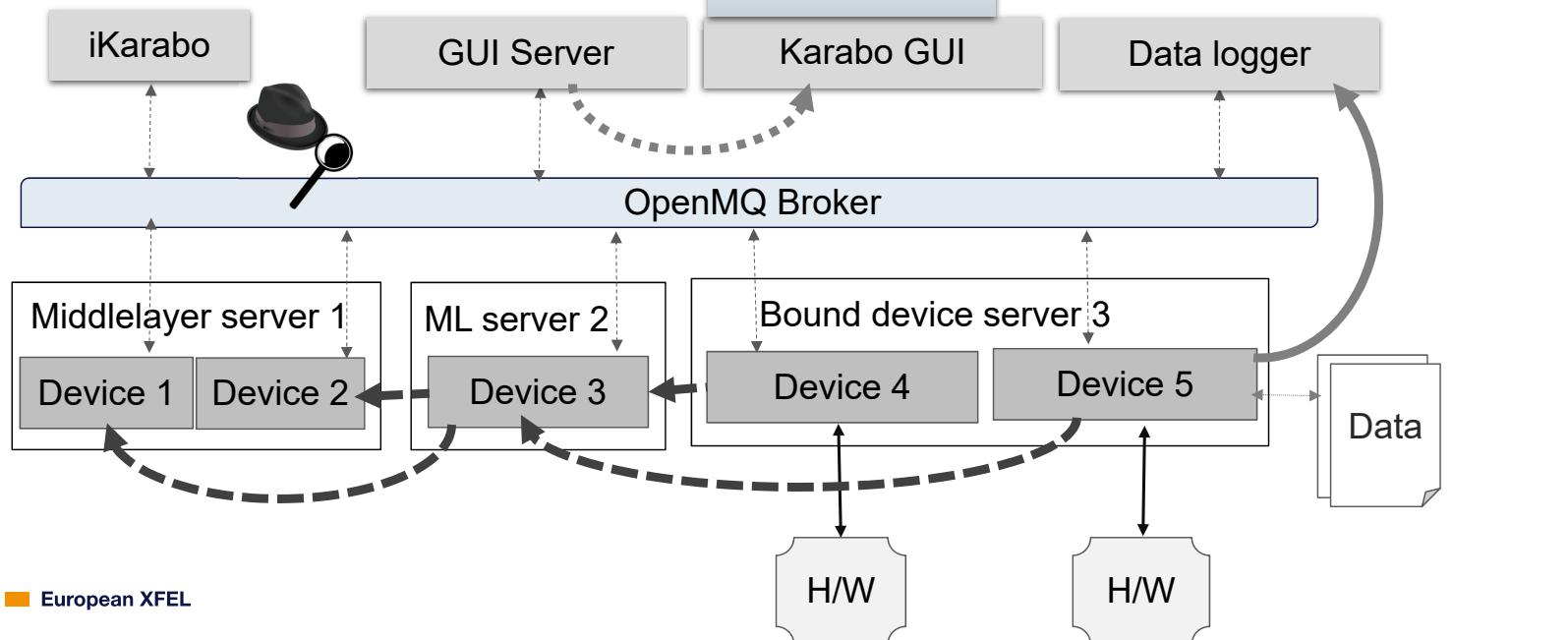


SSH, IPSec, Firewall, IDPS, Antivirus

SSH, IPSec, Firewall, IDPS, Antivirus

Basic Karabo authentication

- ↔ JMS
- GUI client/server protocol
- Point-to-Point connection
- Data pipeline
- ← H/W interface
- ↔ File I/O



Navigation

- Admin
- Expert
- Operator** (highlighted)
- User
- Observer

Search for: Hierarchical view

DeviceID: FXE_OGT1_BIU/CAM/CAMERA Status: ACQUIRING Frame Rate: 10.00 Hz

Configuration Editor

Property	Current value on device
Allow Gaussian Rotation	False
Min/Max/Mean Evaluation Time	0.00844264 s
Pixel Value Frequency Time	0.0101583 s
Background Image Subtraction Time	0.0 s
Pedestal Subtraction Time	0.00298786 s
Image X-Y Sums Time	0.0047884 s
Centre-Of-Mass Time	0.0147908 s
1D Gaussian Fit Time (X distribution)	0.00158143 s
1D Gaussian Fit Time (Y distribution)	0.00177193 s
2D Gaussian Fit Time	0.0 s
Min Px Value	0.0
Max Pixel Value	4095.0
Mean Pixel Value	26.3339255507
Pixel counts distribution	[4120. 8679. 15753.]
X Distribution	[12579. 12867. 12847.]
Y Distribution	[12487. 12051. 14880.]
x0 (Centre-Of-Mass)	536.626986579.DX

Projects

Console Log

+ Show filter options

ID	Date and time	Message type	Instance ID	Description
50825	9/29/17 7:2...	WARN	FXE_XTD9...	alarmHigh: Value 1 of parameter "performanceStatistics.me
50824	9/29/17 7:2...	WARN	SA1_XTD9...	alarmHigh: Value 1 of parameter "performanceStatistics.me
50823	9/29/17 7:2...	WARN	SA1_XTD2...	alarmHigh: Value 1 of parameter "performanceStatistics.me

- Five Global access levels in Karabo:
- Observer
- User
- Operator
- Expert
- Admin (required for example for interlock deactivation)

■ Access exception list per device.

Benefits

- Every device can authenticate the user
- Offline token signature verification

Drawbacks

- Vulnerability to cross-server token **replay** attacks
- Token verification is **expensive** due to asymmetric signature verification

@token: signed
(Session token, Access Level,
Access List, Expiration date)

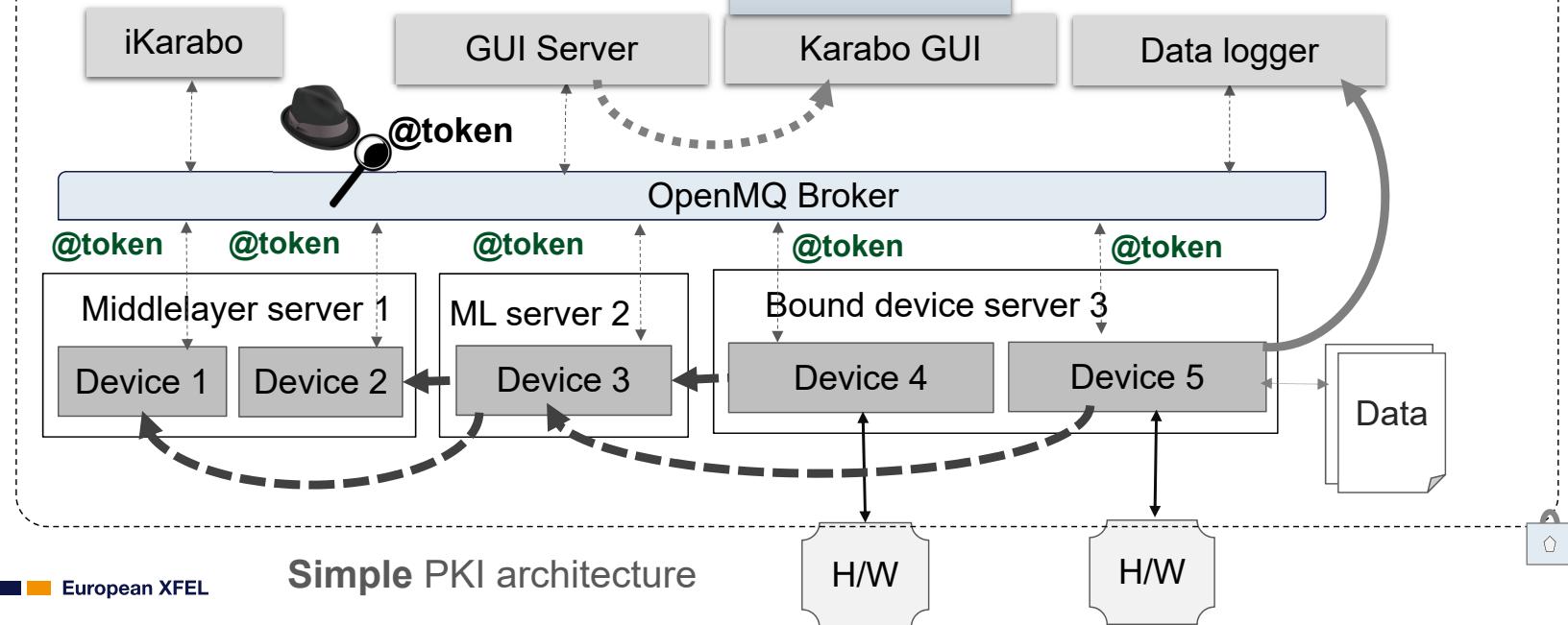
SOAP over TLS

Session Token
Access Level
Access List

Authorization server

Username:
Password:

Public key deployed
at Karabo installation



Benefits

- Every device **server** can authenticate users
- Offline token signature verification
- Robustness against cross-server token replay

Drawbacks

- Vulnerability to token replay obtained by man-in-the-middle
- Token verification still **expensive** due to asymmetric signature verification

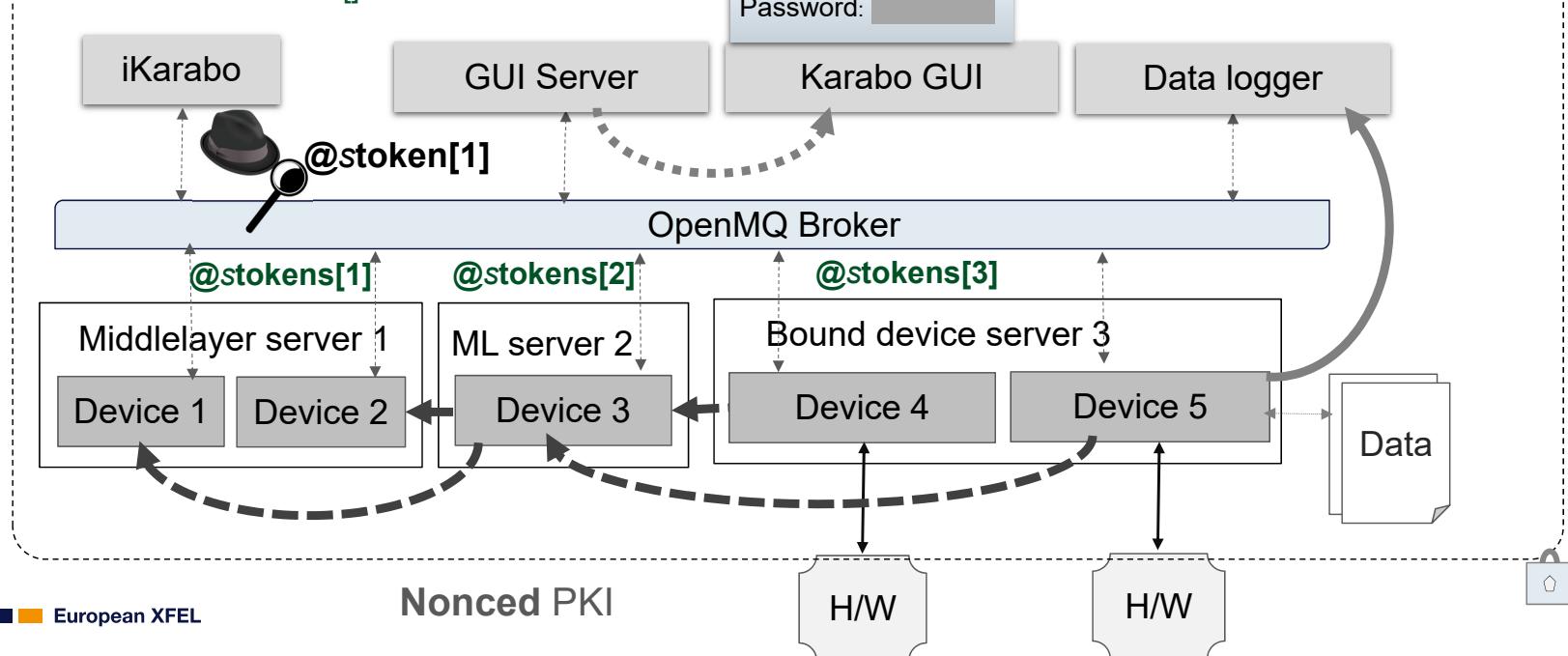
$s[]$: nonces from devices

SOAP over TLS

@stokens[]: signed
(Session token, Access
Level, Access List, s)

Authorization server

Public key deployed
at Karabo installation

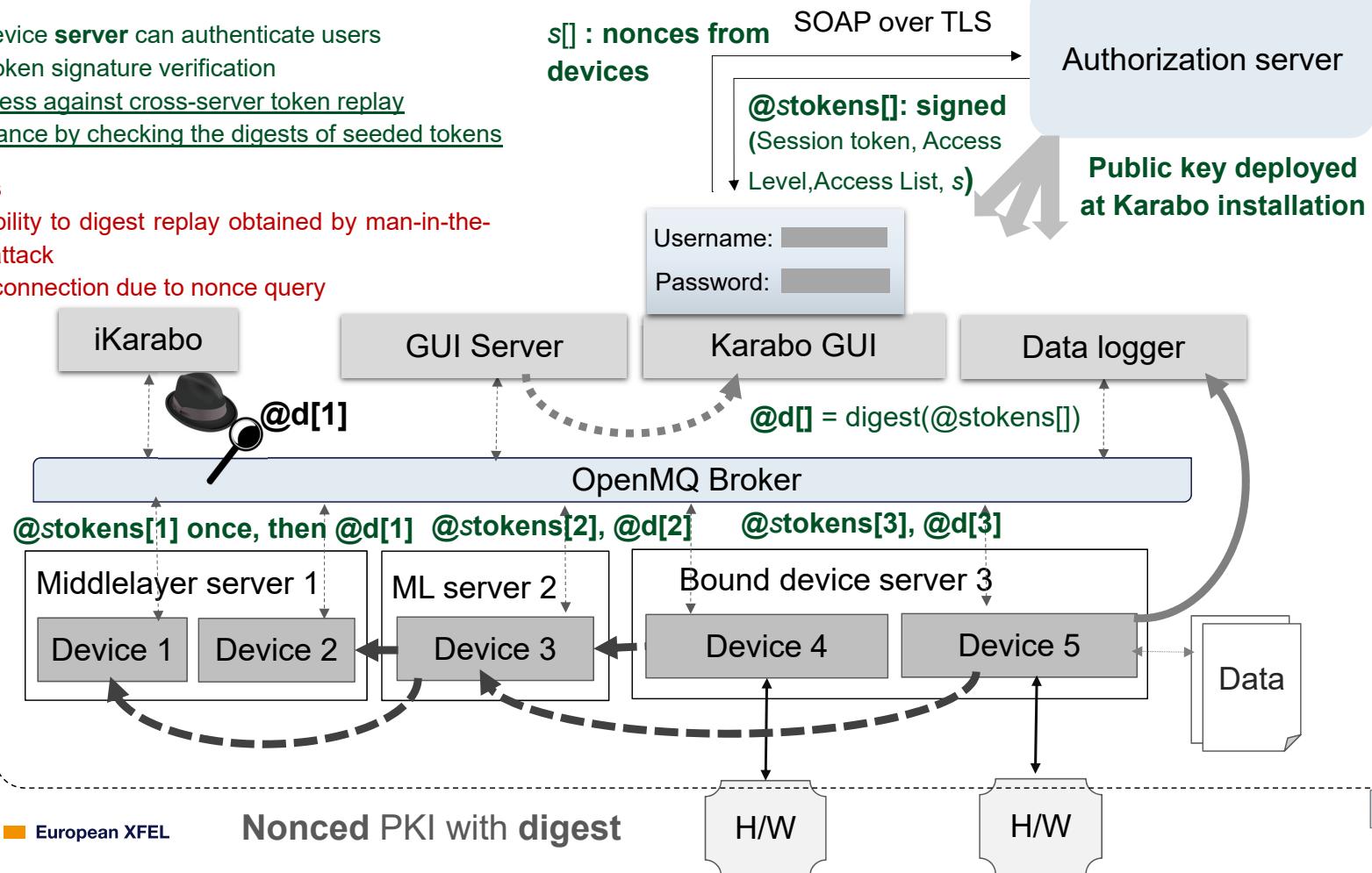


Benefits

- Every device **server** can authenticate users
- Offline token signature verification
- Robustness against cross-server token replay
- Performance by checking the digests of seeded tokens

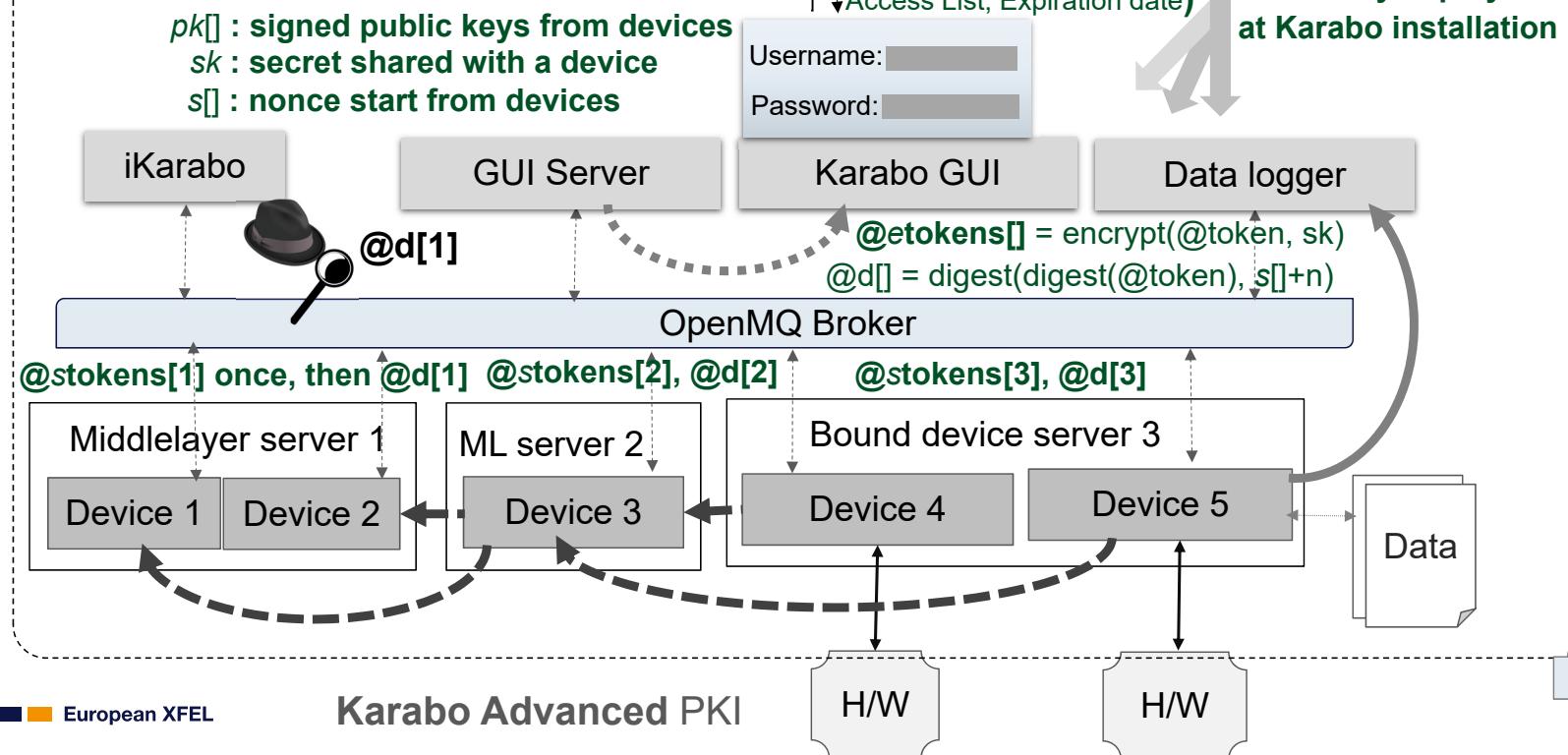
Drawbacks

- Vulnerability to digest replay obtained by man-in-the-middle attack
- Slower connection due to nonce query

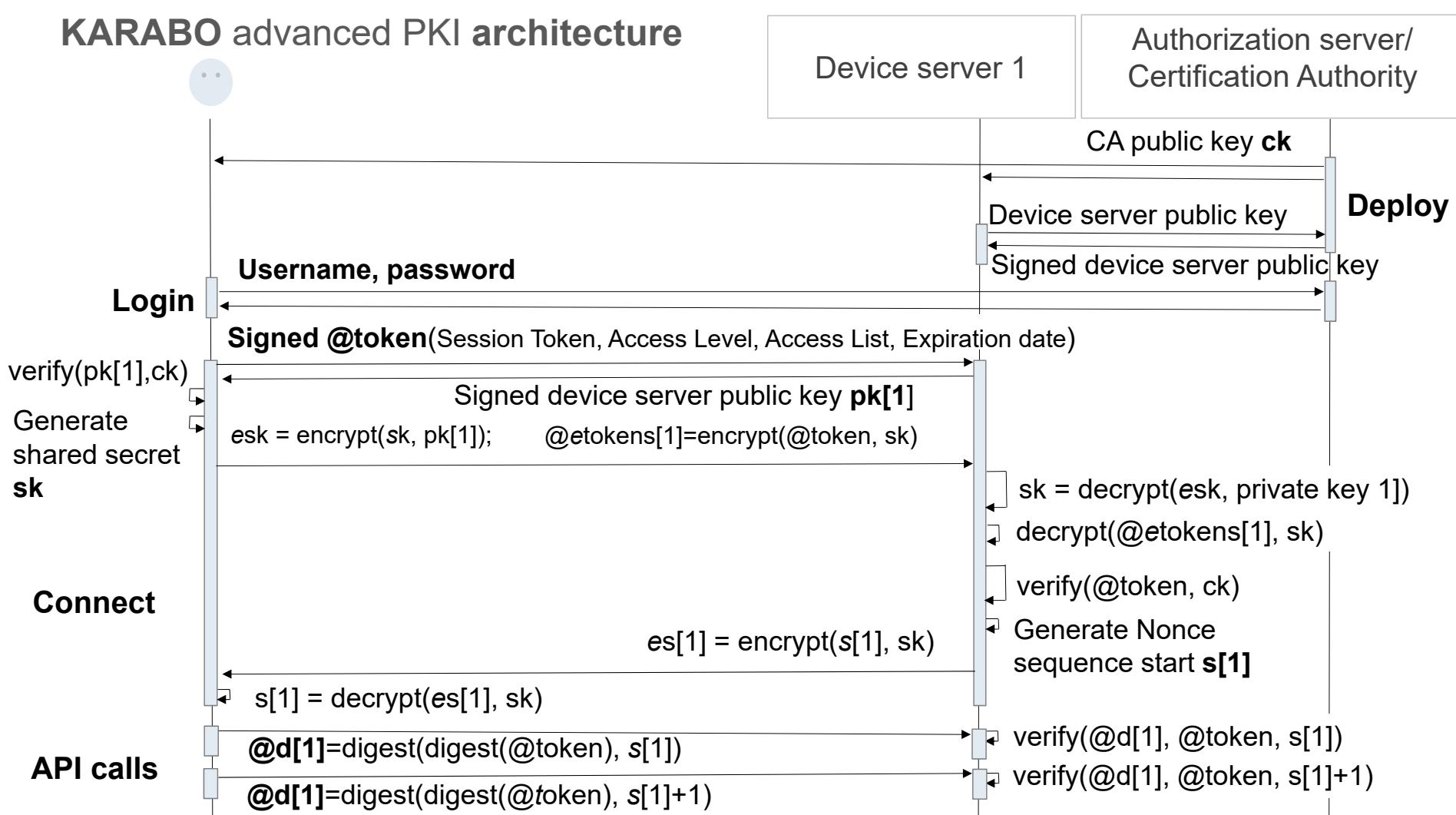


Benefits

- Every device **server** can authenticate users
- Offline token signature verification
- Robustness against man-in-the-middle attacks thanks to token renewal at every message
- Performance by checking the digests of tokens



KARABO advanced PKI architecture



Conclusion

- We aim to protect the SCADA system **beyond general IT security**
- This prevents attackers from issuing valid SCADA messages or **escalading** their privileges
- Our Public-Key Infrastructure proposal for Karabo:
 - User shall access Karabo using a **token signed** by a Certification Authority
 - Device servers have their **public keys signed** by this same Certification Authority
 - Users communicate their session token only to certified device servers, encrypted with the device server public key.
 - The session token is only known to the CA, the user and the certified device servers
 - The nonce start is only known to the user and to a device server, encrypted by a **shared secret**
 - A **nonced digest** of the session token is sent within every API call, preserving performance.