AMR Chain Security

	Author	Approved by	Validated by
Name	Ronan DUBOURG	Yannick Delibie	Yannick Delibie
Entity	KERLINK	KERLINK	KERLINK
Initial	RDU	YDE	YDE
Date	27/01/2015	08/04/2015	08/04/2015
Visa			

Destination

Nom	Company/Service	Place	Commenty

Document history

Version	Modification	Author	Date
0.1	creation	Ronan Dubourg	2015/01/27
1.0	First version	Ronan Dubourg	2015/04/08

Contents

1 Figures and Tables 7

2 Introduction 8

3 Security of the different data links 8

4 Security map 9

5 Security applied in Endpoints 10

5.1 Principe of endpoints keys management 10

5.2 Manufacturing stage 10

5.2.1 Standard manufacturing 10

5.2.2 Custom manufacturing 11

5.2.3 Endpoint delivery to customer 11

5.3 Evaluation stage 11

5.4 Exploitation stage 11

5.5 PSD « handover » 12

5.6 Headend DM key requirement 12

6 Security applied in the station 12

6.1 AMR featuresecurity 12

6.2 Platform security 13

7 Security applied in Headend 13

7.1 AMR feature security 13

7.2 Platform security 13

8 Security applied in the maintenance local tool 14

Open Issues

Reference	Status	Description

References

Reference	Document/link	Description
[1]	**EN13757-4 -3 -5**	WMBUS Protocol
[2]	**E17Z**	AFNOR : Guide d'application des normes EN13757
[3]	**KLK_SPEC_OPEN-AMR-MBUS**	Kerlink implementation of E17Z RF protocol
[4]	**KLK_SPEC_WIRGRID_MODULE**	Describes the wirgrid module: capabilities, architecture, behaviors.
[5]	**KLK_SPEC_AMR_HEADEND**	Specifications of the HeadEnd

Glossary

Keyword	Description
AES	Advanced Encryption Standard
AMR	Automatic Meter Reading
ANSI	American National Standards Institute
API	Application Programming Interface
APN	Access Point Name
BSD	Berkley Software Design
CPU	Central Processor Unit
CSD	Circuit-Switched Data
DAG	DistributeurAutomatique de Gaz
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communication
IP	Internet Protocol
ISP	Internet Service Provider
NMEA	National Marine Electronics Association
LoRa	Long Range
PDU	Protocol Data Unit
PLMN	Public Land Mobile Network

PPP	Point-to-Point Protocol
PSTN	Public switched Telephone Network
REST	Representational State Transfer
RTOS	Real Time Operating System
SAP	SIM Access Profil
SOAP	Simple Object Access Protocol
SDU	Service Data Unit
SIM	Subscriber Identity Module
SM	Short Message
SMS	Short Message Service
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UTC	Universal Time Coordinated
WAN	Wide Area Network
WIRMA	Wireless Intelligent Remote M2M Appliance
WLAN	Wireless Local Area Network
WMBUS	Wireless MBUS

1 Figures and Tables

Figure 1: security of data links.....8

2 Introduction

This document presents how security is implementation in the AMR chain.

3 Security of the different data links

The following figure presents the security applied on the different links of the AMR chain from meter to customer Back-office.

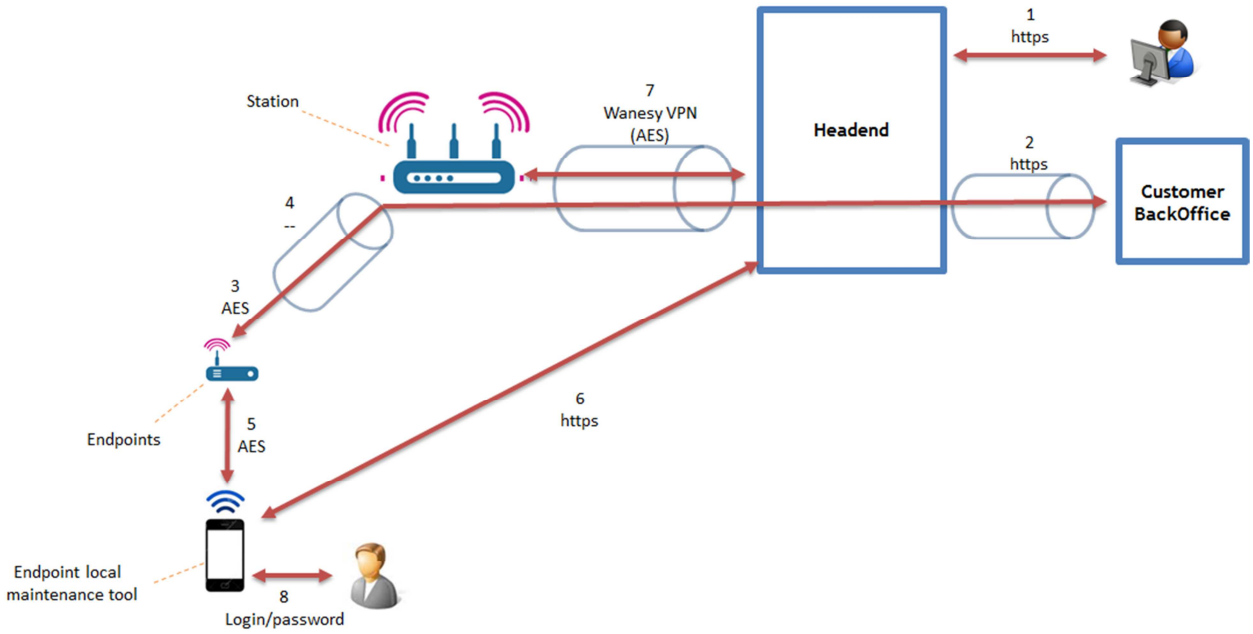

Figure 1: security of data links

The different links are listed below:

Id	Linked item A	Linked item B	Protocol	Security mode
1	Human	Headend	Web interface	https+

				login/password
2	Customer backoffice	HeadEnd	REST interface	https+ login/password
3	Endpoint	Customer backoffice	E17Z protocol over IP	AES
4	Endpoint	Station	E17Z protocol over RF	No additional security as messages are crypted (see item 3)
5	Local maintenance tool	Endpoint	Binary specific protocol	AES
6	Local maintenance tool	HeadEnd	REST interface	https
7	Station	HeadEnd	Wanesy tunnel	AES
8	Human	Local maintenance tool	Graphical Man Machine Interface	Login/password

The important point to notice is that the messages emitted by endpoints are crypted inside endpoint and decrypted inside the customer back-office so meter index cannot be altered or known at any step inside the AMR chain.

4 Security map

The following list presents where are located the different secrets on the whole chain:

- Data and DM Keys in endpoint
- Data Keys in customer backoffice
- DM Keys in headend

- Local acces keys in local tools
- Local tool access login/password in technician memory
- Kmac in station, headend and backoffice
- Wanesy keys in station and headend
- Web interface login/password in customer administrator memory
- Webservice login/password in customer backoffice

5 Security applied in Endpoints

5.1 Principe of endpoints keys management

According to E17Z recommandations, every endpoint embeds a set of AES keys allowing messages to be crypted or decrypted. The keys are different for all endpoints so that if one endpoint is corrupted, the whole system integrity is not altered.

The set of keys is divided in two sub-set, one for data flow (indexes) and one for device management (provisioning, firmware upgrade).

The key generation method is up to the customer but two methods can be used:

- Random generation
- Derivation from master keys (let's call them DATA_ MASTER_KEY and DM_ MASTER_KEY) based on AES, endpoint id, key id…

As device management is performed by the HeadEnd, DM_ MASTER_KEY key or DM key list must be inserted in the headend. All data keys must be known only by customer backoffice in order to avoid any data message attack.

5.2 Manufacturing stage

5.2.1 Standard manufacturing

At production stage, unless a specific customer key requirement has to be taken in account; all endpoints are manufactured with the same process. It means that all the endpoints keys are derivate from the same master keys. Let's call them MANUF_DATA_ MASTER_KEY and MANUF_DM_ MASTER_KEY.

It is intended that theses keys are not safe and not to be used for exploitation.

In the same way, other security credentials (local access AES key + local access password) have to be inserted in endpoint, let's call them MANUF_LOCAL_AES_KEY and MANUF_LOCAL_PASSWORD.

5.2.2 Custom manufacturing

Upon specific customer request, Kerlink can use master keys provided by customer to manufacture customer endpoints or a given explicit keys list provided by customer. When the manufacturing is over, Kerlink "forgets" the keys, customer has to be conscious this is a possible security weakness, this process is relaying on confidence regarding Kerlink manufacturing process.

5.2.3 Endpoint delivery to customer

When endpoints are delivered to customer Kerlink provides listing containing all keys for all modules. This information handover must be secure. Kerlink injects all device management key in the corresponding headend.

5.3 Evaluation stage

For evaluation needs (commercial demonstration, fields test…), security is not a requirement so, the keys injected during manufacturing stage can be used. Theses keys can be shared with customer so customer can decrypt received messages (evaluation scripts…). As manufacturing keys are derivated from masterkeys, evaltool only embeds masterkeys in order to not upgrade the tools for every demo or fieldtest neither embed thousands of keys....

5.4 Exploitation stage

To be able to safely use endpoint, safe keys have to be inserted in the devices (unless already done during manufacturing).

This operation can be performed via two ways:

- using the local maintenance tool during installation process by the customer technician: the customer is responsible to generate its keys (by derivation or not) and to inject them inside local maintenance tool and inside its back-office. The advantage is that Kerlink is never aware of data keys injected in the devices and there is no security weakness During this operation, local tool needs also to change the local access credentials (MANUF_LOCAL_AES_KEY and MANUF_LOCAL_PASSWORD) in order to avoid unexpected local access.
- using the HeadEnd provisionnng capabilities to reset all the keys of all modules.

5.5 PSD « handover »

When a customer swap occurs (ex : public service delegation handover), former customer has to transfer the list of all keys to the new customer.

The new customer may change all the keys using headend provisionning capabilities. The way to generate the keys is up to this new customer. He can decide to derivate them from master keys. In anyway, if the device management keys are changed, new customer needs to provide all of them to HeadEnd administrator.

5.6 Headend DM key requirement

To be able to perform endpoint Device Management, headend needs to know the Device Management keys. HeadEnd support two way to manage key :

- generate DM keys from a DM master key
- use an explicit keys list containing all the DM keys of all the endpoint.

6 Security applied in the station

Security inside the station is achieved at two levels:

- Security of the AMR functionality
- Security of the platform

6.1 AMR featuresecurity

The AMR security is mainly the protection of the Kmac key inside the station. The Kmac key is use to authenticate the belonging of any endpoint to the station fleet.

The risk in case of Kmac attack is not critical because, obtaining the Kmac key doesn't give access to endpoint content. The risk is mainly a "deny of service" attack type because if the Kmac is corrupted inside the station, no more endpoint messages can be transferred to the HeadEnd.

The storage of the Kmac need to be obfuscated inside station memory using hardware or software mechanism. The mechanism used to obfuscate the Kmac inside the station will not be divulgated in this document.

6.2 Platform security

Platform security is needed to protect the station against physical or remote software intrusion into the linux system. Several means have been deployed on the system to protect against intrusion:

- Firewall : a firewall is enabled to protect again unsolicited remote access (SSH access is available for on-sight maintenance)
- Local console access restriction : dynamic untrivial login/password is used
- USB : Only mass storage driver is installed on linux Kernel to be able to perform firmware upgrade connecting a simple USB key. This upgrade procedure can be protected by a password exchange mechanism.

7 Security applied in Headend

7.1 AMR feature security

The AMR security inside headend is mainly the protection of the Device Management master key (DM_ MASTER_KEY) or Device Management keys list:

- The keys are inserted in the headend during the installation, then, keys are obfuscated.
- The keys can be updated through the Web interface only with Administrator credential. This operation is protected inside an https session, then, keys are obfuscated.

The mechanism used to obfuscate the keys inside the headend will not be divulgated in this document.

7.2 Platform security

Platform security is achieved by the company who physically hosts the HeadEnd machine. The topics to be addressed are:

- physical access control
- power supply guarantee in case of general power loss
- Backhaul network access H24
- …

8 Security applied in the maintenance local tool

Human access to local maintenance tool is controlled by a login/ password sequence according to technician habilitation level (installer, administrator…). These credentials have been introduced inside local tool by customer system manager.
A crypted partition is created during application installation to store headEnd access credentials and endpoint local access credentials.
Algorithm to create this partition will not be divulgated in this document.

END OF DOCUMENT