



# DELIVERABLE D4.2\_SEC

## CONCEPT OF SECURITY MECHANISMS IN LOCON SYSTEMS

**Dissemination level: public**  
**Author(s): Alain Chambron (CEA/LETI)**  
**Work package: WP4**  
**WP Leader: Fraunhofer**  
**Partners: CEA**  
**Delivery date: 19/03/2009**

**Short abstract of the deliverable:** This document is a proposal for LocON protocol security mechanisms.

# Table of contents

---

1	Introduction .....	3
2	Security Requirements .....	3
2.1	Network architecture.....	3
2.2	Source Identification.....	4
2.3	Data Integrity.....	5
2.4	Data Privacy.....	5
3	Security in the Network .....	5
3.1	ISO/OSI layers .....	5
3.2	Relevant Implementations.....	5
4	Security in the Application .....	6
4.1	Introduction.....	6
4.2	LocON Security Proposal.....	6
5	Terms .....	9

# 1 INTRODUCTION

Typical LocON applications are security, safety and tracking. These applications require confidence in the obtained positions, safe and secure transmission and secure authentication of the RTLS present. For example, replacement of the RTLS with different devices must be detected (source identification), sending of false position information must be prevented (data integrity) and outside reading of the data transmitted must be prevented (data privacy).

There are many different approaches to ensure the integrity of the LocON system. In some applications all communication runs on secure, encrypted systems like WLAN with encryption enabled. In other use cases transmission over unsecure channels like GPRS or GSM is used.

Encryption, signature and handshaking mechanisms require strong support from the devices in the application. They require computational power and strongly increase the amount of data transfer. Consequently two versions of the LocON protocol are foreseen: An unsecure version for use cases where encryption is not required and a secure version for more heterogeneous environments or special security concerns. Additionally, encryption and security measures required for this type of environment are quite complex. Hence deliverable D4.2 has been split into two documents, D4.2\_prot providing all message types and general principles for operation for operation in unsecure and secure manner, and D4.2\_sec providing all additional information and regulations required for secure operation.

This document D4.2\_sec analyses the security requirements for a reliable and secure data link and transfer in future LocON based installations, describes the existing security network layers and eventually proposes a few solutions.

## 2 SECURITY REQUIREMENTS

### 2.1 NETWORK ARCHITECTURE

This document deals with the security of the link between the communication server and the localization systems.

Fig. 1 provides an overview of the LocON communication network architecture integrated in the airport environment.

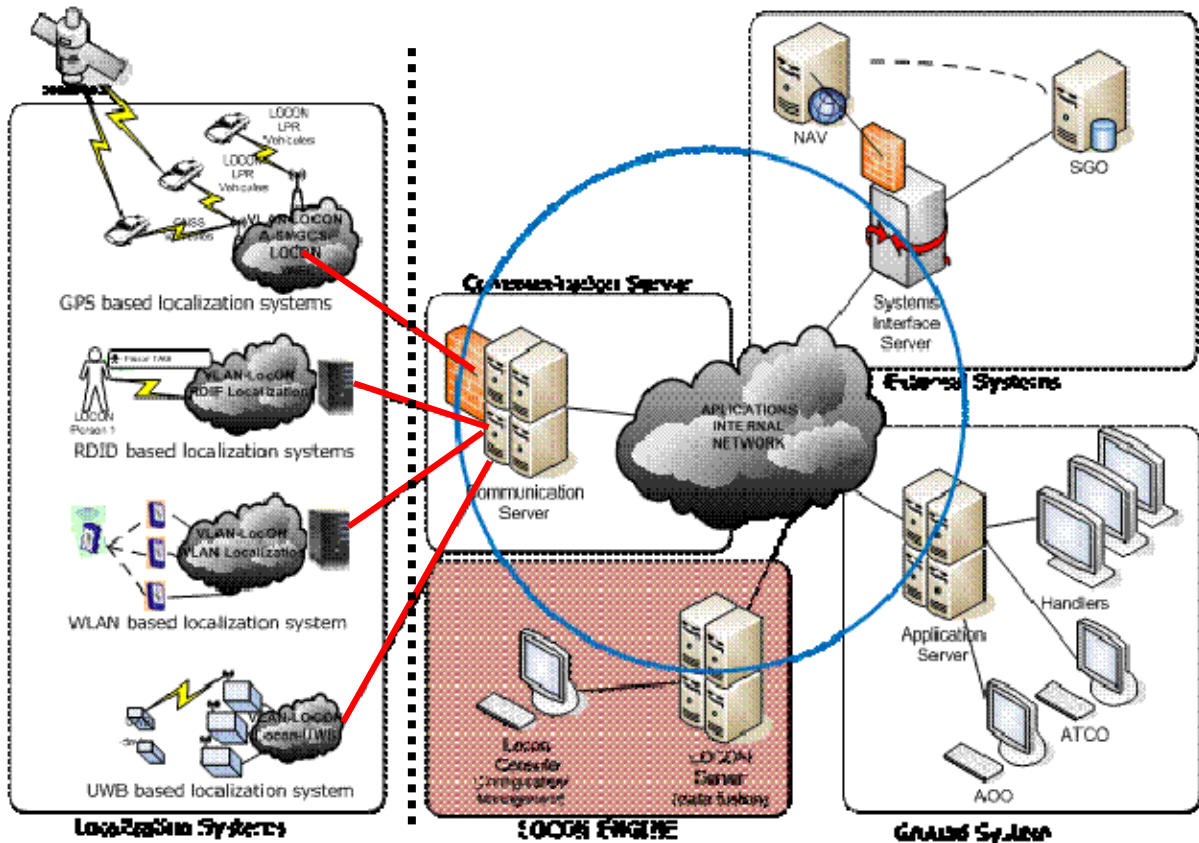


Fig. 1 – LocON architecture overview

The application internal network and the communication server are secure entity. In other words, all the entities on the right of the dotted line are reliable, and the one on the left are unsafe or lie in an unsafe environment.

The aims of the security implementation are:

- make sure that the source of the data packets received by the communication server is known and identified (source identification),
- make sure that the message has not been altered between emission and reception (data integrity),
- make sure that the message cannot be read during transmission (data privacy).

## 2.2 SOURCE IDENTIFICATION

Identification is usually based on something you are, you have or you know. In the case of devices, it is based on a secret that the device uses to identify itself. In order to reduce the risk of exposing this secret, it is not used directly, but instead it is the base of a signature mechanism.

These mechanisms rely on the integrity of the device. It must be very difficult or impossible to break into the device to read the secret. Since the communication server is known as a reliable device, it can hold secrets and this side of the communication is not an issue. The localization devices are more of an issue because they stand outside of the secure area and can be small portable devices.

## 2.3 DATA INTEGRITY

There are few methods to break the data integrity of the message. The most straightforward is to replace part of the message by fake data. Another way is to replay a message known to be good (i.e. previously transferred by the device). These attacks tend to foul the system by providing wrong information. More basically, one can try to foul the system by breaking the transmission of data. This is known as denial of service and is outside of the scope of this discussion.

## 2.4 DATA PRIVACY

Data privacy aims at making the transferred data unreadable by anyone else than the recipient of the message. Data privacy is usually achieved by encryption.

# 3 SECURITY IN THE NETWORK

## 3.1 ISO/OSI LAYERS

The Open Systems Interconnection Reference Model (OSI Reference Model or OSI Model) is an abstract description for layered communications and computer network protocol design. It was developed as part of the Open Systems Interconnection (OSI) initiative. In its most basic form, it divides network architecture into seven layers which, from top to bottom, are the Application, Presentation, Session, Transport, Network, Data-Link, and Physical Layers.

It is possible to secure network transaction at many of these layers. From top to bottom:

- applicative layer: application defined encryption (encrypted mails for example)
- transport layer: TLS (or SSL v3)
- network layer: IPSec
- physical layer: encrypting devices

There are subsequently two ways to deal with security in the LocON protocol:

- embed a proprietary mechanism within the LocON protocol,
- enforce the use of a secured layer as a base of the LocON protocol.

This section will focus on the second method; section 4 will propose a very light weight signature only applicative layer mechanism.

## 3.2 RELEVANT IMPLEMENTATIONS

### 3.2.1 IPSec

IPSec defines two mechanisms to secure the IP traffic, the Authentication Header and the Encapsulating Security Payload. The Authentication Header (AH) ensures the integrity and the authentication of the IP datagrams without data encryption (i.e. without privacy). The Encapsulating Security Payload (ESP) ensures privacy by encrypting the datagram data and optionally the header.

Because IPSec is a network layer, once the secure link is established, any transport protocol whether it is UDP or TCP or any other will be using the secured channel.

Most operating systems provide an IPSec implementation and open source projects are available as well.

Once the IPSec network layer is setup, any communication using the network will be secured through this network.

### 3.2.2 TLS/SSL

As IPSec does at network layer level, Transport Layer Security (TLS) provides endpoint authentication and communications confidentiality over the (Ethernet) network using cryptography. The most common use of TLS is unilateral browser like. TLS is the base of protocols like HTTPS where you want to ensure the identity of the source (the web server for example). It is anyhow possible to use TLS in a bilateral connection mode. Some projects even use TLS as the base of a VPN (OpenVPN).

TSL primarily needs a reliable connection and thus is designed to work on the TCP protocol. Anyhow, a variant of the TSL, the Datagram Transport Layer Security (DTLS) protocol provides communications privacy for datagram protocols such as UDP.

TSL/SSL is widely spread as the base of HTTPS and secure shell (ssh) for example. OpenSSL is an open source implementation of TLS. The latest releases of OpenSSL include an implementation of DTLS.

Since TLS is at transport layer, an application using TLS must specifically be designed to use it. For example, FTP, the file transfer application TLS pending implementation is SFTP. There are two alternatives to that: use “ssh tunneling” or a VPN implementation based on TLS. In that case, the work needed to secure the network is more at system level.

### 3.2.3 SETUP

Whatever the solution chosen (based on IPSec, SSL or application defined) the integration of a localization device within the LocON system will require a setup phase during which both the device and the server will learn to know about each other. It is as well to be defined if this setup phase happens in a secured environment or if a device can be integrated in the LocON platform from any location.

## 4 SECURITY IN THE APPLICATION

### 4.1 INTRODUCTION

The solutions described in section 3.2 all require the existence of the secure layer on the target platform. Porting an existing implementation to another platform may be a long task. Moreover, these solutions are based on public key infrastructures or at least are using asymmetric algorithms like RSA or key exchange mechanism like Diffie-Hellman.

Eventually, these solutions do not solve the issue of a proper secure storage on the remote devices. If this storage does not exist, the easy way to break the security of the system would be to break into a remote device, for example a GPS, read the keys and use them on a rogue device to provide erroneous localization messages.

### 4.2 LOC ON SECURITY PROPOSAL

### 4.2.1 INTEGRATION OF THE DEVICE IN LocON

The opening and closing of the secure channel (see section 4.2.2) will require the presence of a private/public key pair on both the localization device and the LocON server. This proposal does not enforce a secure way to integrate a device in the LocON system. We believe that this setup procedure can happen in a secure environment. The figure Fig. 2 summarizes the system setup at the end of the device integration process.

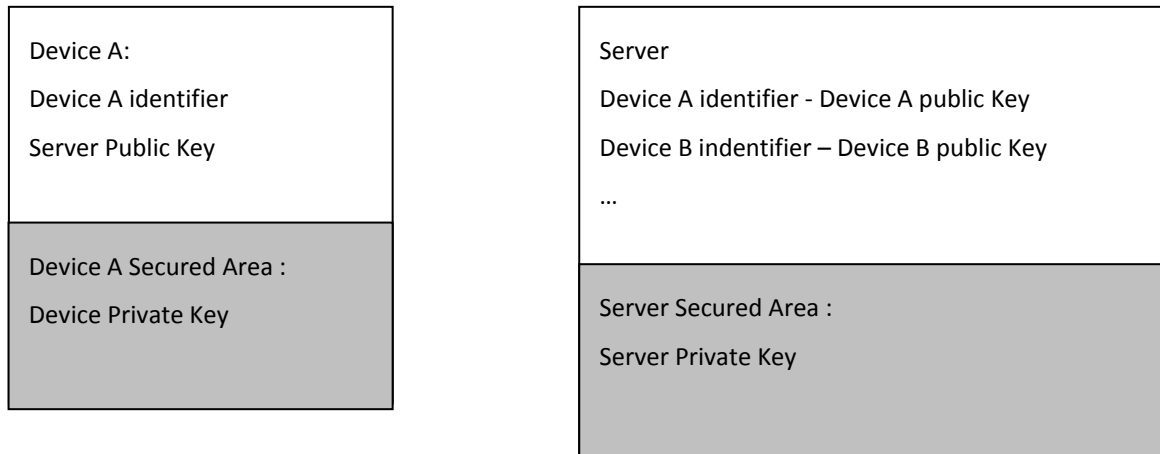


Fig. 2 – Key setup at the end of the device integration process

### 4.2.2 OPENING AND CLOSING A SECURE CHANNEL

Prior to communicating with the LocON platform, a device must request the establishment of a secure channel with the LocON server. Fig. 3 summarizes the generation of this message.

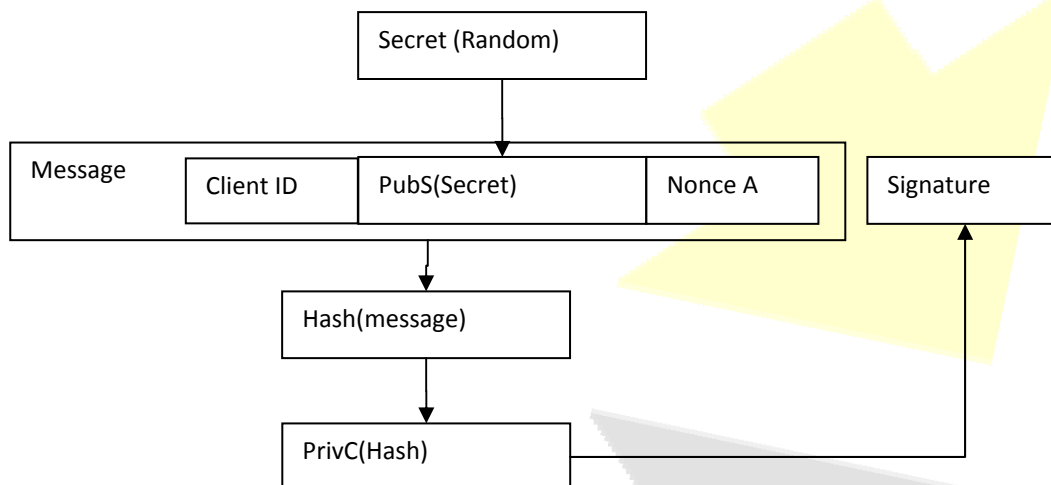


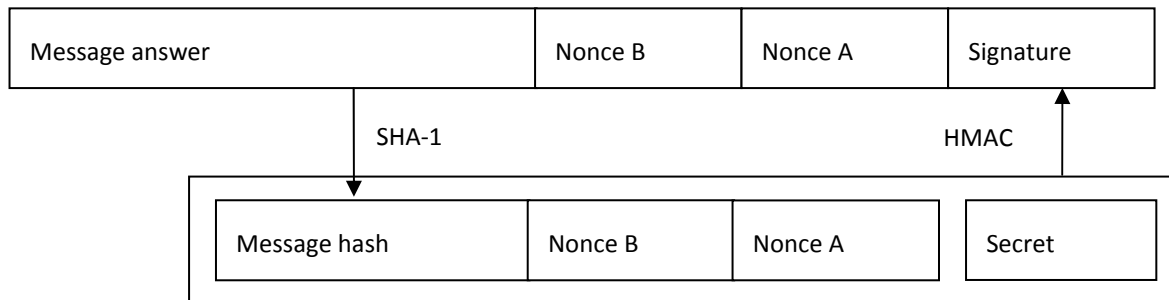
Fig. 3 – Setup a secure channel: client side

The server will perform the following operation:

- calculate the hash of the received message

- decrypt the signature using the public key of the client
- If the two match, then the server knows that the known device Client ID
- Decrypts the PubS(Secret) using it's private key

At this point, both the Device and the Server know about the Secret that they will use later to sign all the transactions. The server can generate the answer to this message:



**Fig. 4 – Secure channel setup: server answer**

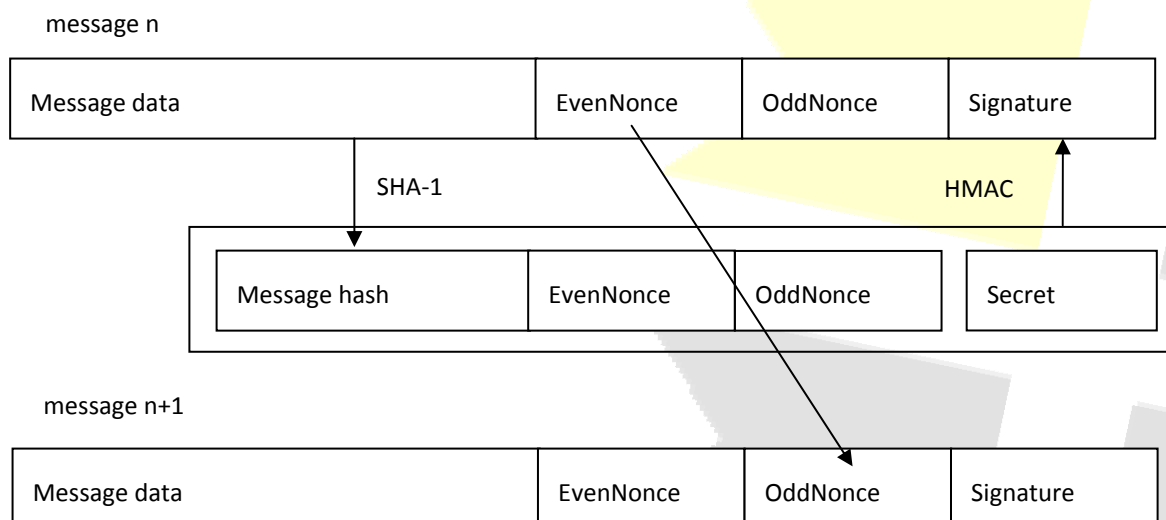
All further commands follow the syntax described in section 4.2.3.

Either the server or the device can close a secure channel through a specific command.

### 4.2.3 SIGNATURE, INTEGRITY AND ANTI-REPLAY

The proposed solution works only on a reliable data connection. A loss of a message will break the anti replay mechanism.

The proposed solution uses a SHA-1 based HMAC to generate a signature of each message. The secret for the HMAC would be shared between the emitter and the receiver. Integrity is ensured by integrating the result of a SHA-1 on the message data to the HMAC input. Anti replay is ensured by chaining nonces throughout the message exchange. Fig. 5 summarizes this mechanism.



**Fig. 5 – Message signature mechanism**

#### 4.2.4 ENCRYPTION

This section describes an optional encryption mechanism. It is based on AES-CBC. The mechanism to exchange the encryption key is similar or integrated into the negotiation of the secure channel.

#### 4.2.5 LOCAL AUTHORIZATION OF PRIVATE KEY USE

This section raises the issue of authorizing the use of the private key on the device side. Do we intend to link this use to the verification of the user of the device, through a pin code, a paraphrase or any other mean of identification?

#### 4.2.6 REQUIRED PROCESSING

The mechanism described in section 4.2.2 requires an asymmetric algorithm. Since this process happens only once per session, it is not time critical and the potential lack of processing power on the device side is not going to be an issue. The asymmetric algorithm could be RSA or Elliptical Curve based (ECC). RSA is defined in RFC 2437 - PKCS #1: RSA Cryptography Specifications Version 2.0.

The mechanism described in section 4.2.3 is the time critical mechanism as it should happen each time a localization device sends data. It requires a random generator to generate the nonces, a SHA-1 and a HMAC. The SHA-1 is a light weight hashing algorithm described in RFC3174 - US Secure Hash Algorithm 1 (SHA1). HMAC is making use of SHA-1 to generate a signature and is described in RFC2104 - HMAC: Keyed-Hashing for Message Authentication.

## 5 TERMS

AES	Advanced Encryption Standard
AH	Authentication Header in IPSec
CBC	Cipher Block Chaining
DTLS	Datagram Transport Layer Security
ECC	Elliptical curve cryptography
ESP	Encapsulating Security Payload in IPSec
FTP	File Transfer Protocol
HMAC	Keyed Hashing for Message Authentication
IP	Internet Protocol
IPSec	Internet Protocol Security
RSA	Asymmetric encryption algorithm Rivest Shamir Adleman
SFTP	Secured File Transfer Protocol
SHA-1	US Secure Hash Algorithm 1
SSH	Secure Shell
SSL	Secure Socket Layer (see TLS)
TCP	Transmission Control Protocol

TLS                    Transport Layer Security  
UDP                    User Datagram Protocol

