Secure Localisation with Location Assurance Provider

Carlo Harpes, itrust consulting s.à r.l.
Benoît Jager, itrust consulting s.à r.l.
Brian Gent, Nokia Siemens Networks

BIOGRAPHY

Dr. Carlo Harpes is founder, managing director, and senior security consultant at itrust consulting s.à r.l. since March 2007. He studied information technology at ETH Zurich, where received his PhD for his thesis on cryptanalysis of block ciphers with Prof. J. L. Massey and Prof. U. Maurer.

Benoît Jager is junior security IT consultant at itrust consulting s.à r.l. since one year, working almost full time on the ESA project "Developing a proof of location for Galileo". He has a Master in Security of Information and Communication System at Université Paul Verlaine de Metz.

Brian Gent (NSN) joined Nokia Siemens Networks SA (then named Siemens SA) in July 2006 and assumed the role of CSI (consulting & systems integration) solution consultant in February 2008. He has been involved in the FP6 project U2010 and in several studies on Space applications. He studied Physics at Portsmouth University where he received his M.Sc. in Microwave and Solid State Physics.

ABSTRACT

This paper describes deliberate threats to the Galileo localisation system and then suggests security features which can provide localisation assurance i.e. that can prove that a given device was at a given location at a given time.

Security objectives are then derived, which can be used to identify and assess appropriate countermeasures such as a tamper resistant chipset or a reliable clock for the user device or a Location Assurance service provided by a central Trusted Third Party (TTP) with plausibility checks (based on tracking), use of Public Key Infrastructure (PKI), correction services and verification of input data for the localisation.

A service architecture scheme is described which can be used over traditional communication channels like General Packet Radio Service (GPRS) or Universal Mobile Telecommunications System (UMTS) to provide a Location Assurance Certificate (LAC) for a registered user device. This certificate can be used by a Location-Based Service Provider (LBSP) to verify the correctness of the device location.

Concerning authentication of Galileo Navigation Message Content (NMC), since there are too few spare bits to add a fast message authentication, we suggest an alternative approach which compares NMC received by Galileo receivers with those collected by reference Galileo receivers distributed over earth's surface.

Note that the proposed solution does not require any changes to the Galileo space or ground segment.

1 CONTEXT

Over last 10 years, thanks to the decreasing price of Global Navigation Satellite System (GNSS) receiver and the freely available US Global Positioning System (GPS), GNSS has become a widely used technology which has subsequently led to the appearance of numerous Location-Based Service (LBS). LBS is taken to mean the provision of a service by a Location-Based Service Provider

(LBSP) to a customer based on the location as computed by that customer's Galileo receiver. Figure 1 below depicts different components of a LBS and some types of service.

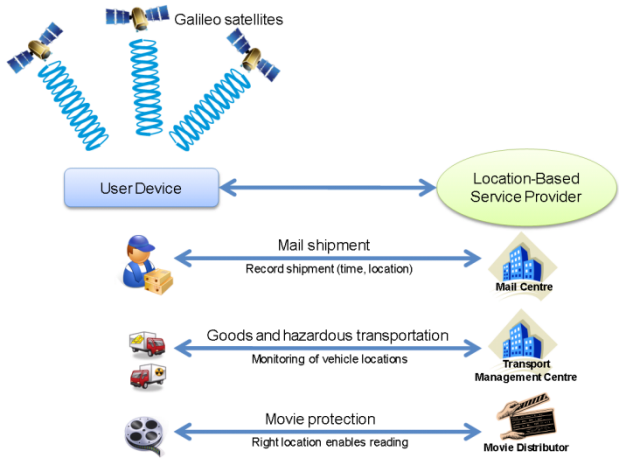

Figure 1: LBS actors

For each localisation, there are several vulnerabilities of different components that may lead to a falsification of one's position:

- First, as depicted by Humphreys et al. [9], any adversary could generate a false signal to mislead a receiver because the Galileo messages are not authenticated;
- Secondly, any hacker could potentially manipulate the software or firmware executing the localisation algorithms on an ordinary GNSS receiver.

A Galileo location can therefore easily be manipulated and neither the user, nor a LBSP can be sure that the location provided is correct.

These security weaknesses prevent to the deployment of new LBS such as speed-limit enforcement, theft protection, forensic reconstruction of accidents, alibi verification, etc and also cast doubt on the trustworthiness of existing applications (tracking of high valued assets, toll collect, parcel delivery), all of which require a high level of confidence in the authenticity of a calculated position. It has thus become a priority to offer countermeasures to such fraudulent activity before they even start to generate negative publicity.

2 OBJECTIVES

In this paper, we propose an architecture scheme combining:

- Security measures which are able to detect that a location computed by a receiver is incorrect;
- Public Key Infrastructure (PKI) allow any relying party, typically a LBSP, to verify an electronic signature of the service provider that check and assure that the location of a given user device is trustworthy.

This service provider is called Location Assurance Provider (LAP) and the signed document is called Location Assurance Certificate.

3 POSSIBLE APPLICATIONS

Although our described service for localisation assurance can be used in any sector, we believe that our described service will be driven by following high security sectors:

- Automotive for assistance driving, tolling, etc;
- Fleet and asset management;
- Localisation-based Access control, e.g. audiovisual content protection for set-top boxes or sensitive operations that should only be performed from secure locations.

In particular, applications requiring non-repudiation (independent proof that a vehicle has used a toll road) and regulation enforcement (e.g. to enforce speed limits in urban areas) can be considered as the main driver for secure localisation.

For example, Scott [4] estimates that falsifying a Vessel Monitoring System (VMS) used in fishing regulation could enable a fishing vessel which covers its true activity for 30 minutes to land an additional 60 000 $ worth of fish, crabs, or shrimp. Another example depicted in [4] is illegal dumping of trash and other hazardous materials, estimated as a 10-12 billion dollars industry.

4 THREATS

Knowing that Galileo architecture is not secure, numerous threats and attack points were identified in the Galileo architecture which could prevent Galileo receivers from computing and reporting real and correct Galileo data (Position, Velocity and Timing (PVT)).

We based our analysis on the catalogue of threats provided by MAGERIT [8], a risk analysis and management methodology for information systems. In this catalogue, threats are grouped by general type (e.g. threat "storm" belongs to "natural threat" type). We decided to analyse the group of wilful attack, as non-wilful (e.g. Natural disaster, Errors and unintentional failures…) are out of scope.

The following threats have been identified as most important.

Shielding/Jamming prevents a GNSS receiver from accurately capturing GNSS signals; i.e. an attacker can use a noise generator to block or degrade the transmission of the Galileo signal. Similar effects occur unintentionally in urban area,

canyons, or indoor areas (MAGERIT category: *Alteration of information* and *Corruption of information*).

Galileo Signals spoofing overlays the Galileo signals with manipulated signals; an attacker uses a generator to overlay or substitute the signals sent by the GNSS satellites (MAGERIT category: *[Re-]routing of messages*, *Sequence alteration* and *Entry of false information*).

Software Code spoofing installs infected software inside GNSS receivers; receiver users may be the attacker or a victim of a malware attack (MAGERIT category: *manipulation of the device configuration*, *masquerading of user identity*, *malware diffusion*, *software manipulation*).

Sending a False Location: here, the legitimate LBS user sends directly a false location to its LBSP, i.e. user is the attacker (MAGERIT category: manipulation of the device configuration).

In the **Meaconing** attack, the attacker intercepts and rebroadcasts the navigation signals to confuse navigation. The meaconing stations cause inaccurate (or completely false) locations to be obtained by receivers. (MAGERIT category: *Alteration of information* and *Corruption of information*).

5 STATE OF THE ART

We collected information related to proof-of-location issued from research for terrestrial networks and GNSS applications. We reviewed current projects, initiatives and publications addressing this subject and also the methods and protocols that they propose, and assess their applicability with respect to the reference Galileo system and the security requirements identified in the previous subtask, as well as efficiency and cost criteria. SWOT [11] analyses has been used to assess applicability of security mechanisms.

5.1 Navigation Message Authentication

Hein et al. [2] and Pozzobon et al. [3] describe this security mechanism which uses the digital signature of navigation data for delayed authentication, i.e. authentication some seconds after the standard localisation. This protocol therefore prevents GNSS signal constellation spoofing, but fails to meet the Time-To-Alarm (TTA) requirement of civil aviation which requires the integrity of localisation to be assured with a maximum delay of 6 seconds [12]. Moreover, the protocol does not a priori include timestamps against meaconing attacks. Hein et al. [2] highlights that the time delay added by the processing of the satellite signal by a spoofer is about 10 milliseconds, which corresponds to the reciprocal transmission rate.

5.2 Angle-of-Arrival (AoA)

Kuhn [5] depicts the AoA security mechanism which enables a check that a received signal actually comes from the direction in which the satellite must be. The location of the satellite could be accessed using the Ephemeris or Almanac, available through the signal satellite [1].

We decided not to retain this security mechanism because of the excessive price of the hardware anticipated, notably the requirement of specific antennas.

5.3 Inertial Navigation System (INS)

Hein [2] recommends using a separate, non-GNSS-based position technology, based on inertial measurement, barometers, odometers, compasses, etc.

We decided not to retain this security mechanism for the moment as it requires heavy and expensive implementation. But thanks to its potential for integration in a road vehicle, this security mechanism could be implemented in future.

5.4 Received Clock Bias (RCB)

With the RCB security mechanism, the receiver derives the absolute time from Galileo signals and compares it with its own internal clock. If the time computed from the satellite signals is earlier than the clock receiver time, then this indicates either a fluctuation in the user clock or a replay attack (meaconing).

As explained by Scott [4], there are two points of time where the RCB enables the detection of a meaconing attack:

- At the beginning of a meaconing: the attacker sends a first delayed signal which makes the time computed from the satellite signals earlier in relation to the time clock receiver.
- At the end of meaconing: the attacker stops delaying signals, the receiver again collects correct Galileo signal. The internal clock is then earlier than the Galileo time computed; since it has been synchronised to the previously falsified signals.

The advantages are numerous and this technique is a reliable indicator of whether a meaconing attack occurs.

5.5 Received Signal Strength (RSS)

The signals emitted by satellites have an initial strength and during their propagation, this strength decreases to a certain level at the earth's surface. The RSS security mechanism aims to monitor the

received power of the Galileo signals. This security mechanism enables the detection of abnormal variations in the power of Galileo signals which could be caused by urban canyons or more likely by a spoofing or meaconing attack.

Although this security mechanism does not prevent an attacker from changing the strength of his fake signals, it makes an attack more difficult and we retain it thanks to its default integration in the GNSS receivers.

5.6 Spread Spectrum Security Codes (SSSC)

Signal Authentication through Spread Spectrum Security Codes (SSSC) [3, 4], are synchronous cipher streams seeded by an unsent digital signature from an Authentication Navigation Message, interleaved with normal spreading sequences.

To conclude on this mechanism, we are sceptical of the benefit of this solution since the mixing of coding and security is not a good engineering principle. Note however that these authentication mechanisms have been implemented in military GPS and have been discussed in [4] and [9]. As there are ongoing research activities on this [13] and this study has a different focus, we did not continue the analysis of this solution.

5.7 Tamper Resistant Device (TRD)

The Tamper Resistance Device security mechanism aims at preventing a user from tampering with the user device in order to influence its operation. This feature provides a level of confidence in the integrity of the device.

We decided to retain this security mechanism because it provides a means to secure the device against user manipulation and secondly to secure the communication with the LAP.

5.8 Conclusion on state-of-the-art

Our analysis of various publications identified several security mechanisms deserving more detailed analysis, some of which merited integration into our architectures. It is understood that cryptographic security measures cannot provide complete security against all theoretical attacks, even though such attacks may be unrealistically complex or unfeasible in real applications. For this reason, non-cryptographic countermeasures will remain a valuable and necessary complement to cryptographic techniques, which will in any case be developed further.

6 BRIEF ARCHITECTURE DESCRIPTION

This chapter briefly describes the architecture scheme that is presented in this paper.

Firstly, we introduce several security measures and ideas that were not found in the “proof of location” state of the art document but which we consider as relevant for our architecture scheme.

6.1 Tracking and Plausibility Checks (TPC):

The first security mechanism to consider is TPC. By continuous monitoring of the localisation, derivation of speed and comparing speed and location with parameters applicable for a given device, it is possible to detect localisation that cannot be correct, and could have resulted from an attack. For such plausibility verification, additional data like indication of Earth’s surface can be used to e.g. detect a car indicating a position some distance above ground.

If the mechanism detects an implausible value, the TPC blocks the conformation of an assurance level.

We suggest adding this security mechanism because of its low cost and high capacity to detect a meaconing attack.

6.2 Central Assurance Provider

The second security mechanism consists of centralising the entity which will decide if a location is trustworthy. The fact that assurance is not provided locally but by a central application allows the use of dynamic risk management principles, e.g. to react to an indication of fraud by one user by denying assurance (or reducing the assurance level) on other users in similar situations. It further allows continuous adaptation of parameters (e.g. tolerated time deviation in RCB or tolerated signal strength variations in RSS) with state-of-the art assurance mechanisms, which is a less cumbersome option than updating parameters on the user devices.

Consider, for example, the RCB which compares the time derived from the signals of satellite with the internal time clock provided by the User Device. Instead of evaluating for itself the clock deviation, the receiver sends its Clock Bias and its derived location to a Trusted Third Party (TTP) responsible for verifying the plausibility of clock biases. If the clock bias is below a tolerable limit, the TTP certifies the location as authentic, otherwise it refuses certification. If the attack appears to have occurred previously, the TTP can also revoke previous localisation assurance certificates. The Central Assurance Provide may have particular information on the precision of the internal clock and adapt acceptable limits according to ages, consideration of clock biases of devices in similar region, the conditions at the given location, etc.

Similar advantages can be found for other security mechanisms if provided centrally, e.g. Received Signal Strength, etc.

6.3 Public Key Infrastructure (PKI)

PKI can be used to digitally sign a user location together with attributes such as the indication of assurance level. The Location Assurance Certificate (LAC) contains the identity of the user device, the indication of the position and time, the assurance level, a reference to the location policy (similar to a Certificate Policy), the algorithm used for signing, the name of the location assurance provider, and its signature.

The signature will be computed based on all previous information using the private key of the LAP. This input information together with the signature will be put in the LAC. The format may be an X509 certificate or just a signed xml file.

To summarise, PKI ensures that any Location-Based Service Provider is able to verify location assurance without a contractual link to the Location Assurance Provider.

6.4 Architecture scheme

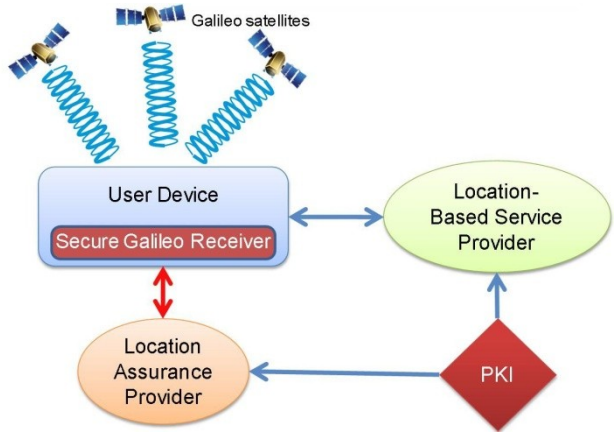

Figure 2: Architecture concept description

Our architecture (Figure 2) finally includes as a main feature a third party Location Assurance Provider (LAP), responsible for the analysis of information sent by a Secure Galileo Receiver (SGR). The information to be analysed includes clock bias, signal strengths of the available satellites and previous localisations. The LAP checks additional information such as previous attacks, reliability of the SGR clock, audit log of the SGR, plausibility with respect to previous localisations, plausibility with respect to a map and information on the integrity of the Galileo satellites.

The LAP intends to provide Location Assurance Certificates (LAC) in real time to a user device over a non satellite based communication channel. This LAC:

- Can be verified easily and reliably by a Location-Based Service Provider using a standard PKI;
- Indicate the level of assurance that can be attributed to the location, as a function of the security checks that have been made.

However, there remains a major weakness in our architecture that we address in the following chapter.

7 MISSING GALILEO NAVIGATION MESSAGE AUTHENTICATION

7.1 Context and Safety of Life

A major issue in the design of location assurance is that the Galileo Open Service offers no authentication, meaning that an attacker in a meaconing attack cannot only add an arbitrary delay, but he can also modify the signal content, particularly the time, so that the victim will not observe any time delay.

As mentioned in [1] and explained in [10], the Safety of Life (SoL) signal contains information about the integrity of all signals. These integrity alerts concerning all satellites of the Galileo constellation are broadcast by each satellite.

Via the Integrity Monitoring Stations (IMS), the Galileo ground segment monitors all the satellite signals. In the event that a signal outside the required specification is detected, the Galileo ground segment issues an alert which is broadcast by the Galileo satellites through the SoL service.

However, this feature could not prevent a local attack, i.e. when an attacker spoofs a precise target without interfering with any of the IMS. In this case, the signal manipulation cannot be detected by a reference station, and therefore the victim cannot be informed by a public service like SoL.

7.2 Data requiring integrity protection

This section indicates data contained in the Galileo messages which are crucial for the computation of a location. If these data are modified, then this could lead to the derivation of an incorrect location for the receiver.

We have used the document [1] containing the description of the Galileo message structure to draw up a list of the most important data used to compute a location at the receiver. The data to be protected (DTP) against any intentional modification are explained in the following:

- **Clock correction:** parameters to correct the time of transmission (TOT) of the satellite signal.
- **Ephemeris:** parameters enabling a Galileo receiver to compute the location of the satellite that sent the signal.

- **GST Time:** information about the synchronisation of the clock satellite in comparison with the Galileo System Time.
- **Ionospheric correction:** parameters to correct the influence of the atmosphere on the speed of satellite signals.

7.3 Remark about GNSS receiver operation

Current GNSS receivers are not designed to continually read the message contained in the GNSS signals. GNSS receivers typically decode the messages including ephemerides, clock corrections, etc. at the start-up. Afterwards, it continues to use these initial data, together with the time shift that are continuously measured from the pseudo random code each satellite signal.

In the LA architecture, we require that a secure UD receiver continuously reads the messages in the Galileo signal. In fact, it seems feasible to spoof a signal with correct pseudo noise data and to send it with any desired time shift. To avoid exploiting this in an attack, the message should continuously be observed.

The receiver should monitor:

- Bit errors of the message since the attacker can guess a larger number of bits. If a few are not guessed correctly, but the receivers ignore them then the attack will go undetected.
- Interference at the very beginning of the transmission of a new pseudonoise sequence must be analysed carefully by the signal receiver. The attacker may guess bit, and as soon as he observes the correct bit from the satellite, he changes the spoofed signal accordingly. If he is nearby and can do this before the receiver has received several bits from the underlying pseudo noise sequence. The receiver may then continue to catch enough correct bits from the pseudo-noise sequence, causing the error at the beginning the pseudonoise sequence to go undetected.

7.4 Strategies for integrity protection

Each localisation operation relies on the input data to protect (DTP). If we want the result of localisation to be reliable, we must guarantee the integrity of the software and the integrity of the input data.

There are three strategies to assure the integrity of these input data:

- Add a message authentication code or digital signature to the message structure that can be checked by the UD;
- Compare the input data with input data collected at another physical location;
- Add an integrity check in the underlying coding algorithm (cf. military GPS use and [3, 4]), which automatically ensures the integrity of all messages transported by the signal.

For the first strategy, we consider two types of integrity checks:

- Almost deterministic integrity checks (where we are virtually certain to detect any changes in the input data). For such checks, one typically uses a hash algorithm which ensures that a collision (i.e. an undetectable modification of the message) has a negligible probability of occurrence of 2^{-64} or 2^{-160} with MD5 or SHA-1, respectively). In other words, an attacker has to test an unfeasible number of messages (almost 2^{64} or 2^{160} message alterations, respectively) before having a significant probability of finding one that will be undetected by the authentication check.
- Probabilistic integrity checks which have a significant probability of detecting an integrity issue and ensure a high probability of detecting data manipulation if a series of localisations are performed. In this approach, a MAC of only a few bits (i.e. 8) can ensure a probability of $1\text{-}2^{-8}$ which is 255 chances in 256, of detecting a randomly modified message).

Since in our design, we intend to be efficient with resources and bandwidth, probabilistic checks with sufficient detection probability will generally be secure enough.

7.5 Message authentication

Before analysing the possibilities to add authentication codes in Open Service (OS), we must study Open Service characteristics.

First, OS is broadcasted by each satellite on three signals:

- E5a-I, which contains the F/NAV Navigation Message Structure
- E5b-I and
- L1-B, which both contain I/NAV Navigation Message Structure.

Each signal consists of a distinct content structure, which contains a defined amount of spare bits. We would like to use these spare bits to add our authentication codes. Bits of the Navigation Message that we consider as spare are:

- Spare bits: comprising of normal spare bits not used in Galileo specification and spare words contained in the I/NAV structure;
- Safety of Life authentication bits: ESA informed us that these bits could be available.

Secondly, we consider two types of authentication:

Symmetric authentication:

Symmetric authentication is considered feasible, when using available spare bits, because even a truncated MAC provides considerable chance detecting a modified message. It demands considerably greater efforts from the attacker as the difficulty increases by a factor of two to the power of the number of bits of the truncated MAC. Nevertheless, heavy key management is a restriction on the use of the symmetric authentication scheme. The requirement that the MAC has to be computed in real time by the satellite itself is a strong disadvantage of MAC in Galileo messages. Another burden is the management of secret keys.

Asymmetric authentication

Asymmetric authentication is not useable principally because F/NAV and I/NAV have insufficient spare bits to give fast authentication.

For I/NAV, if we can use the authentication field of SoL, a 320 bit DSA signature can be added to the E5b signal, and a similar signature can be added to the L1B signal.

The following Table 1 indicates if it is possible to add authentication codes in spare bits depending on output length of the cryptographic algorithm used.

	Symmetric	**Asymmetric**
F/NAV	Verification every 50 seconds with truncated MAC (26 bits)	Not acceptable since 400 seconds observation period before signing is too long (DSA 320 bits).
I/NAV	Verification every 30 seconds with MAC of 160 bits	Verification every 60 seconds Digital signature: DSA 320 bits
I/NAV + SoL	Idem as without SoL	Verification every 30 seconds Digital signature: DSA 320 bits

Table 1: Feasibility and characteristics of authentication for different signals by using only spare bits

7.6 Centralised integrity check

This section explains an alternative solution to the message authentication issue without using a signature and spare bits or additional communication between satellites and receivers. The basic idea is to check the integrity of the DTP using a centralised entity. This centralised entity will be responsible for comparing these data with similar data from other receivers, or by preference, a set of trusted receivers. In our architecture, the role of the centralised entity will be played by the LAP. We will call this security mechanism Central Message Authentication (CMA).

CMA architecture steps:

This section describes the operation of the CMA architecture.

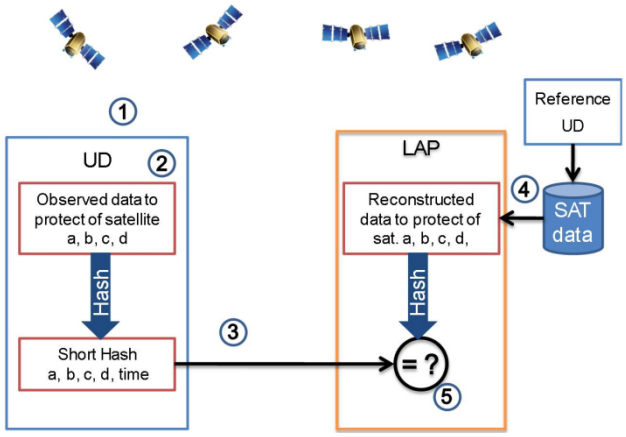

Figure 3: CMA architecture steps

1. The satellites constantly emit their signals containing all the important data previously found and described in section 7.2;
2. A UD captures signals of (at least) 4 satellites (identified by a, b, c, d,), de-modularises them and recovers the corresponding NMC;
3. UD computes a hash over the concatenation of all data to protect of the observed satellites, and sends the hash, together with the satellite identifiers and the time (of each satellite) to the LAP;
4. The LAP retrieves the data to protect for the relevant time from a database containing all data broadcasted by Galileo satellites;
5. The LAP computes the hash by using the same algorithm as the UD; and compares the result with the received hash. If there is a match, the LAP can deduce that the UD has used correct input data for the localisation.

Reference User Device (RUD) may be redundant and distributed over the earth's surface, preventing attacks on several RUD at the same time. Thus spoofing of one RUD can be detected by comparing information coming from other RUDs.

Collision attack of the hash algorithm:

Since most satellite data (Ionospheric correction, additional random data, etc) are unknown in advance, and since collision of old data are useless (typically a signal of one or two minutes in the past

is no longer relevant for a secure time-stamped localisation), an attacker has very reduced time to find a collision on the hash. Therefore, we estimate that hash length of 4 bytes is good enough. (An attacker has then to generate 2^{32} i.e. about $4 * 10^9$ data before finding a match with the hash of an observed satellite signal. However, there has to be more than 2^{32} possible fraudulent messages among which the attacker can choose one that pass the authentication test.

Since bandwidth between SGR and LAP is not a critical issue, we even recommend using 8 bytes for the hash on input data.

We suggest using a publically known but fast hash algorithm, like SHA-2 and to extract the first part of the result. Note that due to the short lifetime of the hash and the need to make a collision attack (the Galileo message is not know in advance (so that a birthday attack on the hash algorithm cannot be prepared in advance), even faster (and less robust) algorithms than SHA-2 can be used.

Frequency of verification:

Since this solution only requires a few additional bytes to be sent to the LAP, we suggest making this verification for each Location Assurance Request.

Data to protect:

A receiver just needs to check authenticity of DTP when it requests a LAC. Therefore, hash on DTP will be added in request message structure that the LAP and UD will use to communicate together.

Note that transmission errors resulting in a few less significant bit glitches may occur despite using CRC. This will cause the hash verification to fail.

Therefore we suggest checking the integrity of almost invariable data separately. Ephemerides typically change every 100 minutes. Therefore we suggest that the UD checks the integrity of the received ephemeris:

- At start-up of Galileo receiver;
- If these data change with respect to the values of the previous validated localisation. As they change every 100 minutes, a dedicated message or an optional field in standard message could be used to authenticate the ephemeris.

Thus, we can reduce the data to protect that are checked at each LAC request to

- Clock correction;
- Ionospheric correction;
- GST time.

Computation time:

To check the integrity of input data of the localisation, the SGR basically needs to compute one hash algorithm for each secure location, which is very few compared to the complexity of localisation.

There is a negligible increase of the message from the SGR to the LAP, consisting of the 4 or 8 bytes to be transmitted. Additionally, there is a dedicated integrity check for each new ephemeris, which results in a dedicated verification message to, and a response from the LAP about every 100 minutes.

The major computation resource has been delegated to the LAP. For each location assurance, the LAP has to retrieve the data from the satellites observed by the UD, and compute a hash on these data. Compared to the complexity of decryption of the secure message between UD and LAP, this computation time is irrelevant.

The LAP however must refer to a network of UD or to a direct connection to Galileo to retrieve all data of all satellites that can be observed by all its customers.

Error probability:

If a few bits transmitted to the LAP are false (due to noise uncorrected by CRC verification, the localisation may still succeed but the authentication will surely fail. Subsequent studies or field trials should analyse how often this may happen.

Conclusion on Centralised integrity check:

To conclude, the Central Message Authentication seems to be feasible and present a good alternative to cryptographic solutions since it avoids all key management aspects, and the related cost in terms of communication.

In our architecture for secure localisation based on LAP, this integrity check can be done without notable overhead. Note, however, that this solution is for commercial use as it cannot be offered free of charge.

7.7 Conclusion on missing authentication

Although both a Message Authentication Code and an electronic signature for Galileo Open Service signal were designed, no convincing argument was found for adding such a field in the Open Service specification.

The use of MAC requires very difficult key management to prevent theft of the key from a compromised user device. Digital signature can be used for I/NAV. It can be encapsulated in spare bit if we collect spare bits for 60 seconds for one signature.

But we have to be aware that message authentication does not prevent manipulation of the underlying signal.

We also note that computation of such authentication has to be performed on the satellite, which represents a strong operational burden.

Integrity checks without using cryptographic keys which compare the input data of the localisation algorithm with a secure reference are then considered. This can easily be integrated into service architecture which includes a central assurance provider.

8 SPECIFICATION OF COUNTERMEASURES

In this section, we give a rough description of all components which belong to our architecture (See Figure 4 below).

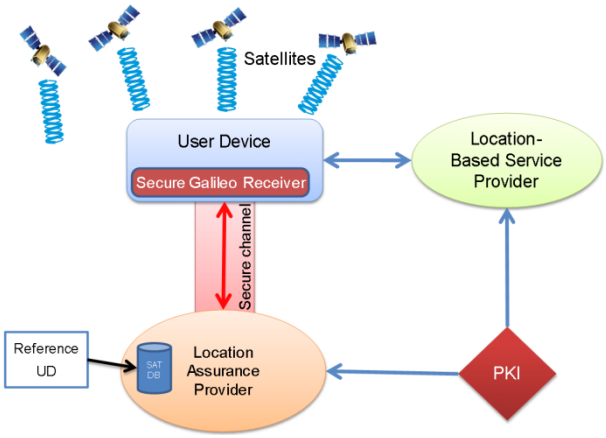

Figure 4: LAP architecture

8.1 Galileo satellites

Satellites send signals containing Navigation Messages. The Navigation Messages includes:

- Time of sending;
- Ephemeris information;
- Almanacs information;
- Clock correction;
- Ionospheric correction.

8.2 Receiver

The signals then reach the receiver, where it is used to derive its location by multilateration. In order to fulfil this function in a trustworthy way, the receiver uses a Tamper Resistant Device (TRD), to make sure that the localisation based on multilateration is integer. The TRD should have facilities to be patched with updates signed by an authorised issuer. Furthermore, the TRD includes keys to setup a secure communication with the Location Assurance Provider.

To implement RCB, the TRD itself has to include a reliable clock.

The TRD sends an encrypted and signed request to the LAP. This request contains at least:

- A TRD identification;
- The derived location and time;
- A list of visible satellite and their respective, signal strength;
- The Clock Bias;
- A list of the previously indicated values from previous localisations for which there has been no request for location assurance.

8.3 Location Assurance Provider

In the location assurance architecture, we add a new component that we call Location Assurance Provider.

After receiving a location assurance request from one of its known SGRs, the LAP checks:

- The validity of the SGR (based on id, no revocation, correct firmware, ...)
- Plausibility of signal strength
- Plausibility of clock bias
- Other general checks (e.g. Tracking and Plausibility Checks (TPC), integrity of the Galileo system bases on its own validation of I/NAV messages and EGNOS).
- If all checks succeed, it generates a LAC.
- It transmits the LAC to the SGR over a secure channel (encryption and authentication), together with some maintenance information (like request for firmware update…).

8.4 Location-Based Service Provider

The UD forwards the LAC received from the SGR to its LBSP. The LBSP, knowing the public key of the LAP (which is available through the PKI) verifies the validity of the signature using the public key of the LAP. If the signature is valid and the provided assurance level is sufficient, the LBSP give access to a service for the customer in possession of the user device.

8.5 Protection profile (ISO 15408)

Security requirements for this reference architecture by following Common criteria (ISO 15408) have been defined in [14]. This document defines the Target of Evaluation as comprising of the SGR, the LAP, and the channel between. The architecture intends to provide Location Assurance Certificates (LAC) that:

- Can be verified easily and reliably by a LBSP by using a standard PKI.
- Indicate the level of assurance that can be attributed to the location, as a function of the security checks that could have been made.

It also specifies the TSF, Security Functionality of the Target of Evaluation, by formulating about

ninety security functionalities, which cover all indicated security requirements.

9 CONCLUSION

To conclude, our architecture enhances security of localisation for highly secure LBS (transport of goods, fleet management…) via Galileo with an important advantage that there are no modifications requirements of Galileo specifications, as our architecture is based on following technologies:

- Location Assurance Provider: dedicated to check information coming from Galileo receiver and to generate an associated Location Assurance Certificate;
- Public Key Infrastructure (PKI) to generate and verify Location Assurance Certificate.;
- Secure Galileo Receiver implementing tamper resistance to prevent from manipulation of a Galileo receiver;
- GPRS or UMTS technologies for the communication between the LAP and SGR.

Since our proposed architecture is independent from Galileo specification, it could be adapted to provide Location Assurance for all existing GNSS (GPS, GLONASS…).

10 ACKNOWLEDGEMENTS

This work is supported by the European Space Agency project: "Galileo Proof of Location" (ESTEC Contract 21753-08) [14]. The authors would like to thank Laurence Duquerroy (ESA) for her valuable comments on this work.

11 REFERENCES

[1] *European Space Agency*: Signal-in-space interface control document sis-icd; Private document; 24 October 2006.

[2] *Guenter W. Hein, Felix Kneissl, Jose-Angel Avila-Rodriguez and Stefan Wallner*: Authenticating GNSS Proofs against Spoofs; Inside GNSS; September/October 2007.

[3] *Oscar Pozzobon, Chris Wullems and Kurt Kubic*: Secure Tracking using Trusted GNSS Receivers and Galileo Authentication Services; In Journal of Global Positioning Systems (2004) Vol. 3, No 1-2.; 2005.

[4] *Logan Scott*: Location Assurance; GPS World; 1 July 2007.

[5] *Markus Kuhn:* An asymmetric mechanism for navigation signals; Lecture Notes in Computer Science, Volume 3200/2005; Springer Berlin / Heidelberg; 2005.
http://www.cl.cam.ac.uk/~mgk25/ih2004-navsec.pdf

[6] *Dave Singelée and Bart Preneel:* Distance Bounding in Noisy Environments; Leuven, Belgium; 2007.

[7] *Edward Felten and Brent Waters*: Secure, Private Proofs of Location; Princeton Computer Science TR-667-03; 2003.
ftp://ftp.cs.princeton.edu/techreports/2003/667.pdf.

[8] *Ministerio de Administraciones Publicas*: MAGERIT – version 2, Methodology for Information Systems Risk Analysis and Management; 20 June 2006; http://www.csi.map.es/csi/pdf/magerit_v2/magerit_catalogue_en_v11.pdf

[9] Todd E. Humphreys, Mark L. Psiaki, Paul M. Kintner, Jr., Brent Ledvina and Brady O' Hanlon: Assessing the Spoofing Threat; GPS World; January 2009.

[10] *Benedicto Javier, Dinwiddy Simon, Gatti Giluliano, Lucas Rafael and Lugert Manfred*: Satellite System Design and Technology Developments; European Space Agency; November 2000.

[11] *Businessballs:* SWOT analysis; 2009 http://www.businessballs.com/swotanalysisfreetemplate.htm

[12] *Victor Wullschlegerm, William J. Hughes, Ronald Braff and Theodore Urda:* FAA LAAS Specification: Requirements for Performance Type 1; FAA.

[13] *Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone*: Handbook of Applied Cryptography; CRC Press; 1996.
www.cacr.math.uwaterloo.ca/hac

[14] Carlo Harpes, Benoît Jager, Brian Gent: Developing a proof of location for Galileo – Final report; ESA Contract 21753-08, 2009.