Implementation and Validation of a Localisation Assurance Service Provider

Xihui Chen*, Carlo Harpes‡, Gabriele Lenzini*,
Miguel Martins‡, Sjouke Mauw*†, Jun Pang†
*Interdisciplinary Centre for Security Reliability and Trust, University of Luxembourg
†Faculty of Science, Technology and Communication, University of Luxembourg
‡itrust consulting, Luxembourg

***Abstract*—Existing Global Navigation Satellite Systems offer no authentication to the open service signals and so stand-alone receivers are vulnerable to meaconing and spoofing attacks. These attacks interfere with the integrity and authenticity of satellite signals: they can delay signals, or re-broadcast signals. Positioning is thus compromised and location-based services are at risk. This paper describes a solution to mitigate this risk. It is a trusted third-party Localisation Assurance service that informs location-based services providers up to which level a location claimed by client can be trusted. It runs several tests over the localisation data of client receivers and certifies the level of assurance of locations. An assurance level expresses the amount of trust the third-party has that a receiver's location is calculated from integral and authentic satellite signals.**

***Index Terms*—GNSS authentication, GNSS spoofing, meaconing, localisation assurance.**

I. Introduction

Only a few years ago accurate positioning was a prerogative of military and governments. Nowadays, Global Navigation Systems (GNSS), such as the American GPS, the Russian alternative GLONASS, and the newcomer European Galileo, offer free and accurate navigation to civilians as well.

Free positioning has fostered business and led to a large number of Location Based Services (LBS). LBS can, on receiving a client's location (which client's receivers calculate from GNSS) customise the offered services. For example insurance companies can track their valuable assets, local authorities can enforce cars to pay toll-roads, and media organisations can follow journalists in dangerous zones of the world.

However, civilian location-based services are insecure. Different from the military and commercial LBS they use unencrypted GNSS signals and are thus vulnerable to attacks against navigation signals. Because of such attacks receivers have no guarantee that the received GNSS signals come unaltered from legitimate satellites.

As explained in [12] there are good motivations to attack location-based services, among which financial gain is the most obvious and important one. Moreover, navigation messages are transmitted over radio links, which are insecure channels by their very nature [2]. This makes attacks on signals even easier.

There are three main attacks against GNSS currently – *jamming*, *spoofing* and *meaconing*. Jamming aims to prevent receivers in an area from tracking GNSS signals by broadcasting radio frequency noise. Spoofing targets at misleading a GNSS receiver by tricking it to lock onto fake, but appearing legitimate, signals. Even if it is not successful, spoofing is able to compromise significantly a receiver's calculating position, velocity and time. Meaconing has similar effects as spoofing. It intercepts, delays and rebroadcasts radio navigation signals to confuse a GNSS receiver.

In this paper, we concentrate on spoofing because it is more sophisticated and able to cause more damage than the other attacks.

At present there are no solutions for LBS providers to avoid processing spoofed locations. In this paper we propose one. It is a Localisation Assurance service for LBS providers. This paper describes the architecture of the service, which extends a previous architecture proposed by Harpes et al. [1]. The architecture is able to manage and run a large and dynamic selection of checks on GNSS signal integrity and authenticity. It is not in itself a new algorithm to analyse GNSS signals and detect spoofing.

Fig. 1 recalls the information flow of our Localisation Assurance service. A mobile GNSS-enabled user device, after having computed its location from the received GNSS signals, sends the location and the navigation data to the Localisation Assurance Provider (LAP). The LAP then analyses the navigation data and estimates an assurance level that represents the amount of confidence that the LAP has about the location being calculated

from integral and authentic GNSS signals. The higher the assurance level the more trustworthy the location. The LAP certifies the assurance level and returns it to the user device, which forwards it to the LBS provider. LBS providers can thus deny to serve a request if the location does not have a sufficiently high level of assurance. A Public Key Infrastructure is used to validate certificates.

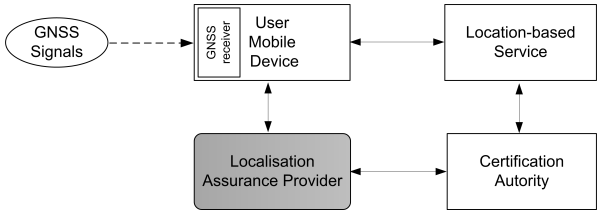

Fig. 1. The information flow with Localisation Assurance.

The work described in this paper relates to the project "Developing a prototype of Localisation Assurance Service Provider (LASP)", an ESA-funded project executed by itrust consulting in partnership with the Interdisciplinary Centre for Security, Reliability and Trust (SnT) of University of Luxembourg.

II. The Architecture

This section explains the structure of the LAP in terms of its components and data flow (see Fig. 2). It takes *assessment requests* as input and outputs *localisation assurance certificates*. An assessment request contains a user's location to certify and the navigation data used in the calculation in the positioning process.

The architecture consists of five interconnected logic units – the *Data Manager* (DM), several *Security Checks* (SCs), *Data Aggregator* (DA), the *Assurance Level Generator* (AG), and the *Certification Manager* (CM). We use "Logic unit" to denote a functionality which can be implemented, possibly but not necessarily, as a single software component. Alg. 1 describes briefly the main module of the LAP. It launches all the above units that run concurrently. The symbol $\|$ stands for parallel composition of processes.

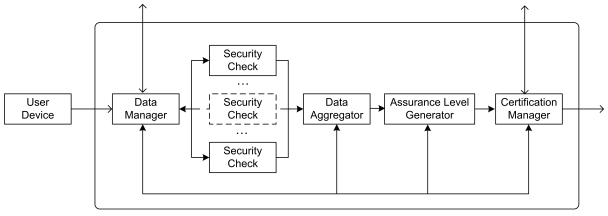

Fig. 2. The architecture of the LAP.

Algorithm 1 LAP Main

1: $DM \parallel (\|_{i\in\{I\}}\ (SC_i)) \parallel DA \parallel AG \parallel CM$

The Data Manager. It inputs assessment requests and stores, retrieves, and maintains the information that the other units need to work properly. For example, it maintains statistics about GNSS signals that a security check may need. It also updates the false positive and false negative rates for each security check, which are required by the Data Aggregator.

The Data Manager also keeps log of intermediate and final outputs of the units, which can be used in a later time to control the quality of service. The functionality of the Data Manager is described in Alg. 2. The algorithm refers to unspecified parameters (i.e., $\langle p_{i_1}, \ldots p_{i_k}\rangle$, α_{DA}, α_{CM}, and α_{AG}) that are retrieved from an internal database here not described and then sent to the intended units. These parameters are specified in the actual implementation of the architecture.

Algorithm 2 Data Manager

1: **while** true **do**
2: $rq \leftarrow$ receives_UserDevice
3: % Receives assessment request

4: **for all** $i \in I$ **do**
5: $\langle$ Retrieve SC's parameters $\langle p_{i_1}, \ldots p_{i_k}\rangle$ $\rangle$;
6: sends_SC$_i$($\langle rq, \langle p_{i_1}, \ldots p_{i_k}\rangle\rangle$);
7: % Sends request and parameters to SC$_i$
8: **end for**

9: **for all** $x \in \{\mathrm{DA}, \mathrm{CM}, \mathrm{AG}\}$ **do**
10: $\langle$ Retrieves x's parameters α_x $\rangle$;
11: sends_x($\langle rq, \alpha_x\rangle$);
12: % Send request and parameters to unit x
13: **end for**

14: $\omega \leftarrow$ receives_DA;
15: % Receives partial output
16: $c \leftarrow$ receives_CM;
17: % Receives certificate

18: $\langle$ Stores $\langle rq, \alpha_{\mathrm{DA}}, \alpha_{\mathrm{CM}}, \alpha_{\mathrm{AG}}, \omega, c\rangle$ $\rangle$;
19: $\langle$ Run quality control on stored data $\rangle$;
20: **end while**

Security Checks. Each security check checks for particular pieces of evidence in favour of or in opposition to GNSS signal integrity and authenticity. The set of security checks in the architecture can vary. It is possible to substitute a security check for another or add a new security check that is developed by a third party. It is also possible that certain selections of security checks are more effective than others.

A security check receives assessment requests. In

Alg. 3 the strategy that a specific security check SC_i runs is represented by the function f. This function also distinguishes security checks. Function f takes a list of parameters as input which are provided by the Data Manager. Let $f_{p_1,\ldots,p_k}$ be the function instantiated with parameters $p_1, \ldots, p_k$. It processes the assessment request rq and returns a value that expresses the probability that a specific observable event caused by an spoofing attack has occurred. Finally, the security check sends its output to the Data Aggregator.

Algorithm 3 SecurityCheck SC_i

1: **while** true **do**
2: $\langle rq, \langle p_1, \ldots, p_k \rangle\rangle \leftarrow$ receives_DM;
3: % Receives tuning parameters from DM
4: $p \leftarrow f_{p_1,\ldots,p_k}(rq)$;
5: % Calculates p
6: sends_DA(p); % Send the output to DA
7: **end while**

The Data Aggregator. It ponders evidence in favour of, or against, GNSS signal integrity and authenticity. It also considers the reliability of security checks: the output of more reliable security checks will have larger weight in the overall estimation.

The Data Aggregator uses probabilistic conditional reasoning [13] to *build* opinions about localisation assurance based on the probabilities that certain events witnessing potential spoofing attacks have happened.

An opinion about GNSS signal integrity and authenticity represents the amount of belief, disbelief, and uncertainty that property is true. Belief and disbelief should be derived by evidences in favour of against GNSS signal integrity and authenticity. It uses the subjective logic [5], which provides operators on *opinions*. In the Data Aggregator the opinions from security checks are first *discounted* according to security checks' past performances, then *merged* so as to obtain the cumulative opinion that expresses the group consensus about GNSS signal integrity and authenticity. In this way, Data Aggregator copes with the uncertainty that might emerge when some security checks contradicts another or when there is not enough information to draw a meaningful conclusion. Alg. 4 describes its main functional steps.

The Assurance Level Generator. It calculates the assurance levels. The assurance level depends on the output of the Data Aggregator and other pieces of information such as the user device and the levels recognised by service providers. Its steps are described in Alg. 5.

The Certification Manager. It prepares the digital certificate that contains the assurance level and sends it back to the user device (see Alg. 6). To compose the certificate the Certification Manager relies upon an external Certification Authority and on a Public Key Infrastructure (PKI).

Algorithm 4 Data Aggregator

1: **while** true **do**
2: **for all** $i \in I$ **do**
3: $p \leftarrow$ receives_SC$_i$;
4: % Receives SC_i's output
5: $\langle rq, \langle \alpha, \alpha' \rangle\rangle \leftarrow$ receives_DM;
6: % Retrieves parameters
7: $\omega^{SC_i} :=$ BUILD(α, p);
8: % Build SC_i's opinion;
9: $\omega^{SC_i} :=$ DISCOUNT$(\alpha', \omega^{\mathrm{SC_i}})$;
10: % Discount on SC_i's past reliability
11: **end for**
12: $\omega^{[SC_i:i\in I]} :=$ MERGE$(\{\omega^{SC_i} : i \in I\})$
13: % Merge SC_i's opinions
14: sends_DM$(\langle \{\omega^{SC_i}\}_{i\in I}, \omega^{[SC_i:i\in I]} \rangle)$;
15: % Send back all result for log
16: **end while**

Algorithm 5 Assurance Level Generator

1: **while** true **do**
2: $\langle rq, \alpha \rangle \leftarrow$ receives_DM;
3: % Receive request and parameters
4: $\omega \leftarrow$ receives_DA;
5: % Receive result
6: $al :=$ SETLEVEL(ω, rq, α)
7: % Set assurance level
8: sends_CM(al);
9: % Send for certification
10: **end while**

Algorithm 6 Certification Manager

1: **while** true **do**
2: $\langle rq, \alpha \rangle \leftarrow$ receives_AG;
3: % Receive request and parameters
4: $c := BUILD(al, rq, \alpha)$
5: sends_UserDevice(c);
6: % Send back certificate
7: sends_DM(c);
8: % Send certificate for log
9: **end while**

III. Security Checks

We implement a prototype of the LAP with a selection of security checks, some of which have not been mentioned in the literature. In this section, we describe the main methodologies behind them and some details about their implementation.

We divide the security checks of the LAP into two sub-classes according to the relation between their output and GNSS signal integrity and authenticity– *sate-based* and *transition-based*. The former look at instant observations or measurements. The latter focus on the effects of the transitions from authentic and non-authentic signals or vice versa. The current implementation includes checks that operate on the following strategies:

Signal-to-Noise Ratio (SNR). This security check makes use of the power of received signals. Navigation signals are usually weakened by many factors during the transmission before reaching receivers. For instance, obstacles (e.g., trees and buildings) and atmosphere can block or absorb part of the signal energy. Therefore, the SNR of received signals should have an upper-bound. Higher SNR are also possible in some rare cases. For example, constructive multipath may happen and the direct and reflected waves add in-phase which results in an increase of the SNR.

We estimate the reference levels (i.e., the upper bounds) for each satellite identified by the pseudo-random numbers (PRN) in terms of elevation angles. We make use of a dataset of the SNR of signals captured during static and mobile observations. The maximal SNR occurred in the dataset is chosen as the reference level for the signals from a satellite at a certain elevation angle.

To capture the unpredictable influences that increases SNR, we construct a function called *half-Gaussian* which decreases the belief in the output as the observed SNR moves away from the reference level.

Doppler. There are two strategies to make use of Doppler. The straightforward one is to check if the measured Doppler shift agrees with constellation and user dynamics (e.g., velocity). Thus, we need to calculate a reference to compare with a user's measurement which depends on the velocities of satellites and the user. A satellite's velocity can be computed as the time derivative of the function mapping time to the satellites' positions [9]. The reference Doppler shift is plausible if the user's real velocity is available since Doppler shift is proportional to the projection of the satellite's relative velocity in the user's direction. However, a user's velocity reported in a request is computed by the receiver which makes use of measured Doppler shifts. If such a velocity is used and the attacker uses an updated almanac, the reference will always be the same as the one measured by the receiver. Thus, we cannot use this strategy unless users have other ways to correctly measure their velocity.

The second strategy makes use of the coherence between the measured Doppler shifts of signals from a satellite but with different frequencies. This is because the relative velocity between a satellite and a user remains the same for different frequencies and the ratio between these frequencies is constant. For example, the ratio r between GPS L1 and L2 frequencies is $1575.42/1227.60 = 1.283(3)$. We observe that 99.6% of the ratios lie within the interval $r \pm 0.001$. The limitation of this strategy is that receivers must support multiple frequencies, which is not the case for most of the commercial receivers.

Navigation data. Satellites keep broadcasting navigation data to receivers in a low bit rate, e.g., 50 bit/s for GPS L1 C/A. The navigation data changes from time to time but when it changes cannot be always predicted. For instance, the almanac fields are updated every 2 or 3 days, the ephemeris every few hours and the time field every second. Therefore, a less sophisticated attacker may not observe the change and sends signals with the old navigation data. A mismatch of the data in received signals and the real-time data indicates a possible spoofing attack.

In our implementation, we refer to EDAS (EGNOS Data Access Service) reference stations to fetch the real-time navigation data as it broadcasts them through the Internet.

Visible satellites. We verify if satellites reported by a user's assessment request are visible at the place where the user claims to be. For a position and a time point, only a subset of satellites are visible and only these satellites are expected to be reported by the user.

In our implementation, we calculates the positions of satellites the up-to-date almanac, from with we can further learn their elevation angles. In practice, receivers discard signals from satellites with lower elevation than a pre-defined threshold called *elevation mask*. The elevation mask varies between receivers. We suggest users disable elevation mask so that all the originating satellites of received signals are reported.

Ground height. The claimed position of a user should be close to the Earth's surface. In other words, a position that is 1,000 meters away above the ground is impossible for a user travelling by car. Such claims are possible when a non-intelligent attacker may only focus on how

to fool a receiver to lock a different place in the 2-dimension space but ignore the height. In this case, it is possible that the height does not correspond to that of a valid position.

The general idea to implement this check is to validate the third component of a position based on the first two. For positions within Luxembourg territory, the reference ground height can be requested at `http://map.geoportail.lu`. However, as only Eastings/Northings coordinates are accepted, we first convert coordinates from WGS84 to Hayford and then project them using the Transverse Mercator algorithm. In our implementation, we take general cases into account such as driving over a bridge or standing on a building. The maximum allowed ground height is set to 20 meters. In other words, users would not ask certification for their locations with a height larger than 20 meters above the ground.

Clock bias. Monitoring clock bias is considered as a technique to detect advanced spoofing attacks [14]. We design an algorithm that is able to detect clock bias. Tests with a signal repeater in a controlled environment show that the clock monitoring algorithm can detect the beginnings and the ends of attacks even if the average delay introduced is about 80 nanoseconds. This attributes to not only the efficiency of the algorithm but also the stable and noiseless nature of receivers' clock. When a COTS smartphone is used, we find that a bias magnitude of about 100 ms is required to enable the detection.

The results of this security check constitute a major achievement of the project. Additional details can be found in [7].

Receiver autonomous integrity monitoring (RAIM) This security check assesses the integrity of GNSS signals using redundant pseudorange measurements. For receivers performing RAIM analysis, it's possible to give an indication about the consistency of pseudoranges using redundant measurements. The presence of a spoofing attack can result in the absence of integrity detected by the receiver. The use of a multifrequency receiver for our tests permit the decrease of the alarm limit to values about 50 - 100 meters resulting in RAIM processing of at least 75 of the time.

Consistency with other positioning sources – WiFi. WiFi access points can be used as an alternative way to localise users when GNSS signals are not available. Given observed WiFi access points, we can have an area where all their signals can be received. This area is calculated based on a dataset recording access points' geographic positions. Intuitively, when a localisation process is consistent with WiFi positioning, then the location computed by the receiver should be inside the corresponding WiFi area computed. This is because if an access point can be observed, then the real location must be in an area where the signals of the access point can reach. We use this observation, under the assumption that WiFi signals are not targeted by any spoofing attacks, to evaluate the quality of the reported localisation process.

Reachability. The reachability check is used to test whether two consecutive locations from a user are reachable with the maximum speed allowed by his way of travelling (e.g., on foot, by car or by air). This security check cannot be used to indicate directly the assurance level of users' locations in their requests. This is because the assurance of the previous location is unknown. However, if two consecutive locations are found unreachable, then we can conclude that the sources of the signals has been changed. This change may happen when attackers terminate a spoofing or attackers start a new one.

Time check. The time check explores the fact that GNSS positioning information includes a reference time, which is sometimes used as time-stamps in many applications. The request of a user includes the time and date calculated based on the satellites navigation message but not on the receiver's internal clock. The LAP verifies the time stamp and gives a low opinion if it mentions a future event or a distant event in the past. Standard communication network delays are accepted.

IV. Discussion: combining security checks.

In Sect. II we explained how the Data Aggregator combines the outputs of the security checks. The Data Aggregator, first transforms each security check's output into an opinion about localisation integrity and authenticity. Then it discounts this opinion by considering the reputation of the security check according to the security check's trustworthiness in the current context, and finally it merges all security checks' opinions to obtain a communal conclusion.

This method can be applied straightforwardly to those security checks whose output is a probability about an observable events caused by an attack. If this is not the case, that is, if the relation between the event's occurrence and the presence of an attack is not direct, the strategy must be adapted to draw correct conclusions from the security check's output.

The security check "clock-bias" is peculiar in this sense. A change in the clock-bias is observed when an attack begins or when an attack has ended. In order to be meaningfully interpreted the clock-bias security check's

output needs to be compared with what the other security checks have found in the previous run. If the general opinion was that there was no spoofing, a change in the clock-bias quite likely implies the start of an attack. This reinforces the relevance of the security checks that at the current time are detecting the presence of a spoofing, but it weakens the relevance of those that are not detecting any.

Whether the reasoning should be adapted or can be applied as it is, is something that should be decided before adding any new security check to the LAP.

V. Experimental Validation

We validate our system by performing a series of tests at ESA ESTEC labs making use of sophisticated signal generators. The equipment is composed of a Spirent GSS-7700 simulator, connected to a dedicated software that controls the parameters of the simulator. A GNSS receiver is also used to provide feedback of the simulated signals to the controlling software.

At the beginning of the test session, the simulator does not provide signals aligned to the live constellation and the detection is straightforward. The following security checks are able to immediately detect that signals are not authentic:

- Satellite plausibility;
- Absolute Doppler;
- Time check;
- Consistency with WiFi.

In the second attempt, we increase the satellites' power. The absolute power security check starts to detect that there were satellites with SNR values higher than usual.

In the third test, we upload the correct almanac and set the time of the simulator. The time is manually set and the resulting difference between GPS time and simulator time is smaller than one second. Then, only consistency with WiFi security check was able to detect the presence of simulated signals. After setting the simulated position to the real position of the User Device (computer) it becomes consistent with the WiFi localisation and the assurance level returns to the maximal one.

In this scenario, the clock bias security checks cannot detect the presence of simulated signals. This security check only detects the transition between two different sources of signals, and then, the continuous broadcast of simulated signals remains undetectable.

Because the functionality to perfectly align simulated signals with live signals is not available, we make use of the multipath simulation capabilities of the simulator. The idea is to simulate a set of signals (A) and a corresponding set of delayed replicas (B), in the same way as in multipath with a reflected wave arriving with some delay to the receiving antenna. This situation is more realistic than switching between live and simulated signals because in a real attack scenario, the receiver will continuously receive live signals, as long as it has an open view to the sky. The attack is deployed as follows:

- Feed the receiver with a set of signals (A) at 10 dB waiting until the receiver locks;
- Start increasing the power of the replicas (B) from -50 dB (minimum power) to 20 dB (maximum power) at a rate of 0.5 dB/s.
- Decrease the original signals (A) from 10 dB to -50 dB at the same rate.

The smallest delay tested was 100 ns and it was successfully detected by the clock bias security check.

VI. Privacy Issues

LAP implements mechanisms to protect user's privacy in agreement with the current EU directives on e-privacy and on data protection. In the architecture that we have so far described, LAP and LBS providers learn user whereabouts. It is well known that from locations and movements it is possible to learn sensitive and private information such as home addresses. LAP employs an ad-hoc solution that has been designed in the scope of the project; it is called *selective location blinding* [6]. Users control up to which level of granularity service providers will know about their locations. This control is exerted without compromising the location assurance certificate on the fully detailed location. In fact, service providers received certified encrypted locations, but are users that control up to which degree of accuracy locations can be decrypted. We remand to [6] for details.

VII. Related Work

Several solutions aiming to discern spoofed signals have been studied in the literature. Because this paper focuses primarily on GNSS-enabled devices we do not report on works addressing differential GPS spoofing, which targets ground-based reference stations [4]. Instead, we report on works that study how to protect GNSS receivers for civilian users. Wen *et al.* in [15] propose and comment nine strategies that, in theory, can succeed in detecting specific instances of spoofing attacks. These strategies run in stand-alone GPS receivers and are named as follows: absolute signal power, signal power changing rate, relative signal strengths, range rate, Doppler shift, correlation peaks, range difference, ephemeris data, and signal power. All these strategies

have limitations and are applicable under specific conditions. Other methods cannot be implemented in software in a single device. We refer here to those techniques that require, for example, the installation of special antennas (e.g., [8]) or algorithms that cross-check current signals with other observations, for example, as done in [3]. In this latter work the interaction between the authentic and the spoofing correlation peaks is used to distinguish spoofed from authentic signals even when they are almost aligned. Spoofing signals from a single source are also likely to be spatially correlated while the authentic signals are not. Another spoofing detection technique is described by Nielsen *et al.* [10]: here the correlation is observed while monitoring signal characteristics like signal-strength or Doppler shift.

All the previous works, and others that similarly describe strategies for spoofing detection, can be used in our security checks. They can be plugged in our architecture. However, none of them propose to combine different spoofing-detection strategies in a service for location assurance. Speaking of new services for GNSS devices, Pozzobon [11] proposes to integrate in the GNSS a new authentication scheme. Here, the solution to spoofing is not detecting attacks after they have struck but avoiding that attacks strike at all. This should be possible by setting up a service that supplies satellites authentication for civilians (presently, authentication is available only for military and governmental use). This paper embarks in an opposite direction to that by Pozzobon: not modifying the current GNSS protocols, which requires the agreement of GNSS owners, but using GNSS as it is while still providing a service that supports location integrity.

In [1] the authors propose an architecture as a possible solution for location assurance that does not require any changes to the signals' structure. It analyses the threats and explains how the existence of such a service might provide localisation integrity to civilian users.

This paper is where the need for a Location Assurance Provider has been first raised. The paper says "*a third party Location Assurance Provider (LAP) [is] 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 proposed LAP is also assumed to return a certificate to the user device that is bound to the previously received data set and that contains an assurance level. Therefore the need of such a service has been already presented. However, in [1] the analysis and decision criteria performed at the LAP are not addressed or disclosed. A few mechanisms called security checks, known in the literature as spoofing countermeasures, are cited as possible technical algorithms to detect the presence of attacks, but no proposal about whether to combine those security checks or how to combine them has been not even hinted. The present work instead goes further and proposes a possible detailed architecture for the LAP.

VIII. Conclusion and Future Work

LAP is an innovative service that is available for private and institutional end-users and provides assurance levels on localisations calculated using open GNSS signals. This paper has presented a design for LAP. The design is made of five modules. The most important ones are the security checks, which test for spoofing attacks, and the data aggregator, which copes with contradictions and combines their outputs into a consistent evaluation of localisation trust. The implementation and the tests that we have run so far, shows that LAP is effective in certifying locations from GNSS signal integrity and authenticity.

As future work we are considering to add more sophisticated security checks in our architecture. We are currently adapting and testing the architecture to work in the mobile network. This is a significant variant of the architecture considering the recent increase of GPS-enabled smartphones and the proliferation of LBS applications for Android smartphones.

Acknowledgement

This work was supported by the European Space Agency (ESA) under the project Developing a prototype of Localisation Assurance Service Provider (LASP) with contract number 4000102584-10-NL-HE.

References

[1] C. Harpes, B. Jager, and B. Gent. Secure localisation with location assurance provider. In *Proc. European Navigation Conference - Global Navigation Satellite Systems*, 2009.

[2] G.W. Hein, F. Kneissl, J.-A. Avila-Rodriguez, and S. Wallner. Authenticating GNSS Proofs Against Spoofs. *Inside GNSS*, July/August:58–63, 2007.

[3] A. Jafarnia-Jahromi, T. Lin, A. Broumandan, J. Nielsen, and G. Lachapelle. Detection and Mitigation of Spoofing Attacks on a Vector-Based Tracking GPS Receiver. In *Proc. of the Institute of Navigatiion International Technical Meeting (ION ITM) 2012, Newport Beach, CA, Jan 30 - Feb 1, 2012*, 2012.

[4] M. H. Jin, Y. H. Han, H. H. Choi, C. Park, M.-B. Heo, and S. Jeong Lee. GPS spoofing signal detection and compensation method in DGPS reference station. In *Proc. of the 11th Int. Conf. on Control, Automation and Systems (ICCAS)*, pages 1616 –1619, Oct. 2011.

[5] A. Jøsang. A Logic for Uncertain Probabilities. *Int. J. of Uncertainty, Fuzziness and Knowledge-based Systems*, 3(9):271–311, 2001.

[6] G. Lenzini, S. Mauw, and J. Pang. Selective Location Blinding using Hash Chains. In B. Christianson et al., editor, *Proc. 19th Security Protocols Workshop*, volume 7114 of *Lecture Notes in Computer Science*, pages 132–141, Cambridge, United Kingdom, March 28-30 2011. Springer-Verlag.

[7] D. Marnach, S. Mauw, and C. Harpes. Detecting Meaconing Attacks by Analysing the Clock Bias of NGNSS Receivers. In *Proc. of the European Navigation Conference, 25-27 April, 2012, Gdańk, Poland*, 2012.

[8] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina. A Multi-Antenna Defence: Receiver-Autonomous GPS Spoofing Detection. *InsideGNSS*, March/April 2009.

[9] B. Motella, M. Pini, M. Fantino, P. Mulassano, M. Nicola, J. Fortuny-Guasch, M. Wildemeersch, and D. Symeonidis. Performance assessment of low cost GPS receivers under civilian spoofing attacks. In *Proc. of the 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010*, pages 1–8, 2010.

[10] J. Nielsen, A. Broumandan, and G. Lachapelle. Spoofing detection and mitigation. *GPS World*, September 2010.

[11] O. Pozzobon. Keeping the Spoofs Out: Signal Authentication Services for Future GNSS. *InsideGNSS*, May/June 2011.

[12] L. Scott. Location assurance. *GPS World*, July:14–18, 2007.

[13] R. C. Stalnaker. Probability and Conditionals. In *Ifs: Conditionals, Belief, Decision, Chance, and Time*, pages 107–128. Kluwer, 1980.

[14] Zhang W, Ghogho M., and Aguado L. M. Extension of GPS Broadcast Ephemeris to Determine Satellite Velocity and Acceleration. In *Proc. of European Navigation Conference - Global Navigation Satellite Systems (ENC-GNSS), Naple, Italy, 3-6 May, 2009*, 2009.

[15] H. Wen, P. Y.-R. Huang, J. Dye, A. Archinal, and J. Fagan. Countermeasures for GPS Signal Spoofing. In *Proc. of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005), Long Beach, CA, September 2005*, pages 1285–1290. Institute of Navigation, 2005.