Galileo Open Service Authentication: A Complete Service Design and Provision Analysis

P. Walker, *CGI;*
V. Rijmen, *University of Leuven*;
I. Fernández-Hernández, L. Bogaardt, *European Commission*;
G. Seco-Granados, *UAB, Spain*
J. Simón, *European GNSS Agency*;
D. Calle, *GMV*;
O. Pozzobon, *QASCOM;*

BIOGRAPHIES

Paul Walker is the Solution Architect at CGI responsible for the AALECS test-bed platform implementation and authentication solutions as well as other navigation technology design and development projects. He received a PhD in Physics in 1996 and has been a software engineer in the space sector since 1999.

Vincent Rijmen is currently full professor with the Dept. of Electrical Engineering (ESAT) of the University of Leuven (KU Leuven). Previously, he held the Chair of Applied Cryptography at the Graz University of Technology and was Chief Cryptographer of Cryptomathic.

Ignacio Fernández-Hernández is the manager and design lead of the Galileo Commercial Service at the European Commission. He holds an MSc in Electronic Engineering from ICAI, Madrid, an MBA from LBS, London, and a PhD from Aalborg University.

Laurens Bogaardt has graduated from University College Utrecht in the Netherlands with Bachelors in Physics and in Economics. Subsequently, he followed Master programmes in these fields in Sweden and in the UK. As a trainee at Galileo's programme management, he was able to integrate his experience in Physics and Economics and work on the Commercial Service.

Gonzalo Seco-Granados is associate. prof. at of Univ. Autonoma de Barcelona (UAB) and head of the Signal Processing for Navigation and Communications (SPCOMNAV) group. During 2002-2005, he was staff member at ESA Radionavigation Section, and involved in the Galileo receivers and applications. He holds an MBA from IESE and a PhD for UPC.

Javier Simon is Service Design Engineer within the European GNSS Agency, currently contributing to the definition and design of the Galileo Commercial Service. He holds a MSc. degree in Telecommunications Engineering from the Polytechnic University of Madrid, Spain. Before joining GSA, he participated in several projects for the study and design of future GNSS algorithms and systems.

David Calle has a MSc in Computer Engineering from the University of Salamanca. He participates in R&D activities related to GNSS algorithms and is the responsible for the Galileo Commercial Service Early Proof-of-Concept.

Oscar Pozzobon is the founder and technical director of Qascom. He received a degree in information technology engineering from the University of Padova in 2001 and a master degree from the University of Queensland in telecommunication engineering in 2003. He has been working in GNSS authentication since 2001.

ABSTRACT

GNSS authentication, and in particular Navigation Message Authentication (NMA), has been already studied in the scientific literature. However, not many references that analyse the assets at risk, existing threats, mitigation actions, and residual risks through standard risk assessment processes, are available. In this paper, we outline how to use such processes to justify the design and selection of some configurable options for the service specification and operational procedures of GNSS Navigation Message Authentication (NMA) using the Galileo Open Service signals. The proposed NMA scheme is based on the TESLA protocol as proposed in [1].

To motivate the design of the service, we first identify the categories of users and associated risks of attack. We then summarize the mitigation capability against these attacks provided by the TESLA solution referred herein.

We define the cryptographic parameters to use for the service in the foreseeable future. We also identify further mitigations that the receiver manufacturer or service user might need to consider to ensure security of the position and/or the time fixes according to their risk aversion. These might include a trusted local clock reference, a process to verify or challenge digital certificates and statistical analysis of symbol recovery.

We then define crypto parameters and procedures that affect the quality of service for different users, as a function of several system performance scenarios.

We show that, for the selected parameters, multi-constellation NMA can be achieved in environments with a masking angle up to 40°. We also show that authentication using only validated signals presents good performance at 5° masking angle, for users requiring four satellites transmitting NMA. This performance may increase through an optimized downlink strategy.

INTRODUCTION

Several techniques have been proposed to mitigate GNSS vulnerabilities [2]. Out of these, adding cryptographic features to GNSS signals is one of them. In particular, GNSS Navigation Message Authentication (NMA) [3] refers to the authentication of the navigation data using the navigation message itself. Moreover, NMA bits, when unpredictable, can make the signals more resilient against replay attacks, and therefore protect pseudoranges against certain threats [4]. Due to its easiness of implementation, NMA is being considered in the Galileo first generation for implementation at its Full Operational Capability.

In this paper we present an initial analysis of the threats that may be addressed using Open Service (OS) Authentication. After addressing the threats and mitigations provided by the solution for OS Authentication, we justify some adjustments and parameters of the detailed implementation. Finally, we present an analysis of the service coverage and assess how this should impact aspects related to *cross-authentication*, i.e. authenticating data from a satellite through another satellite.

In the rest of this section we provide some background to OS Authentication and the use of TESLA to achieve this. Out of the possible cryptographic functions and protocols used for NMA, a protocol based on TESLA [5] is the focus of this paper. TESLA for radionavigation was proposed first, to the knowledge of the authors, in [3]. Later, some adaptations have been proposed in [6] for SBAS, in [7] for e-Loran, in [8] for Galileo I/NAV and in [9] for GPS CNAV. The currently analysed TESLA implementation is based on [1]. Its main difference with the previous ones is that it uses a single key chain for all satellites and allows satellite cross-authentication. This TESLA approach has been tested with Galileo signal-in-space in the E6 band, as reported in [10].

The TESLA-based solution for OS Authentication supports two levels of authentication which we refer to as:

- Navigation Message Authentication (NMA)
- Pseudorange replay protection.

NMA seeks to validate the supporting data used to find a position. Pseudorange replay protection seeks to validate the signal itself. The way TESLA can support this is by providing a source of unpredictable but verifiable symbols which can be analysed to detect certain classes of signal attack [4].

There are alternative means to gain protection against certain attacks and, for some users, combined solutions may be the best defence. For example, spoofing or meaconing from a secondary source may be most easily detected and mitigated using antenna arrays or vestigial signal analysis. Moreover, for approved users NAVSEC encrypted services may provide better authenticity guarantees. These solutions are outside the scope of this paper, but do form bounds on the uses and needs of OS Authentication alone.

THREAT ANALYSIS

The focus of this analysis is on the threat of spoofing or replaying to produce a false PVT at the end user. With this objective, the analysis describes first the system assets at risk, then a set of representative use cases; it identifies the threat sources and then the threats themselves; it presents the mitigation actions, and it finalizes with the residual risks. This first part of the paper is based on standard security risk analysis methodologies, however, a full-fledged risk analysis is beyond the scope of this paper. Note that Denial of Service attacks are not considered; the focus is on attacks that invoke false position or time. Use cases where denial of service might be a viable attack need to be separately addressed.

System Assets

The analysis is focused on the mitigation of threats to the use of GNSS perpetrated on the end user of the service. Threats to and vulnerabilities of the infrastructure of Galileo or other upstream systems supporting the OS Authentication service are outside the scope of this analysis. The system assets under consideration are therefore only those assets that impact the end use and user devices, namely:

- Service Messages: Radionavigation signals to the Device providing PVT.
- Authentication messages: signals to the Device providing PVT authentication data

- Cryptographic data input into the Device
- The Device itself.

Uses

Because this is a generic analysis rather than a specific system analysis, we define a number of system use categories and threat source categories against which threats are rated. To structure this we first define the following dimensions.

Power Constraint:

- **Unconstrained** – power is not an issue; the device may permanently track the GNSS signals if required.
- **Constrained** – using internal power sources, so that the application will reacquire the signal each time a fix is needed using the shortest snapshot possible.

Tamper Motivation:

- **Non-tampered** – the user is not motivated to tamper with the operation of the device and would identify evidence of tamper or attack.
- **Tamper Motivated** – the user is motivated to tamper with the device or collaborate with an external system to help with the tampering.

Service Use:

- **Tracking** – the position is recorded or reported to a third party.
- **Positioning** – the holder of the device requires position information in situ.
- **Timing** – only time is important and an accurate position is already known.

Rather than considering the product of all combinations, we present the following categories as those of most interest.

LVT - Low Value Vehicle Tracker: Unconstrained Power, Tamper Motivated, Tracking. Examples include road tolling, insurance telematics, vehicle fleet and fishing vessel tracking.

HVT - High Value Vehicle Tracking: Unconstrained Power, Non-tampered, Tracking. Examples include high value goods or vehicles such as trains

POT - Portable Tracker: Constrained Power, Tamper Motivated, Tracking. Examples include tagging.

HVP - High Value Positioning: Unconstrained Power, Non-Tampered, Positioning. Examples include high value vehicle navigation, autonomous piloting including virtual anchoring (boat or rig).

TIM - GNSS Disciplined Timing: Unconstrained Power, Non-Tampered, timing. Examples include communications and phasor measurement units.

Threat sources

A simplified set of threat sources are presented here:

IO - Individual Operator: For example a driver avoiding being tracked. They may be motivated to subvert the purpose of the GNSS device but possess limited resource and little technical ability. They may have direct access to the receiver equipment.

LC - Lone Criminal: They possess limited resource and little technical ability. They are motivated by financial gain.

PA - Prestige Attacker: For example a Hacker, Academic, Business or Journalist. They are motivated by publicizing their achievement; possess moderate resources but significant technical ability.

OC - Organized Crime: They are motivated by financial gain and possess sizeable resources justifiable by gains.

OA - Organized Attacker: For example terrorists or foreign intelligence services. They are motivated by the disruptive effect of the attack and have potentially unlimited resources and ability.

Attack Likelihood

Using the broad categorization define here, the likelihood for each category of attacker to attack each service use category is rated as Extreme (4), High (3), Moderate (2) , Low (1), or Negligible (0), as shown in Table 1.

	IO	LC	PA	OC	OA
LVT	4	3	2	1	0
HVT	0	2	3	4	4
POT	4	4	2	2	0
HVP	0	1	3	4	3
TIM	0	0	3	1	4

Table 1. Likelihood of attack on service uses defined in the text (rows) by threat sources defined in the text (columns) from Extreme (4) down to negligible (0) likelihood.

Threats

The threats considered here are threats outside the Galileo system. The predominant threat categories can be summarized as:

Cryptanalysis: using algorithms and computing resources to defeat cryptographic defenses and thus falsify navigation data and signals. These are subdivided into:

- **CCA** - Current Cryptanalysis, using known techniques.
- **FCA** - Future Cryptanalysis, using a hitherto unknown vulnerability.

Disclosure: including theft, the release of sensitive information that introduces vulnerabilities. These are subdivided into:

- **EXD** - External Disclosure: Release of information by systems outside Galileo (i.e. the user).
- **IND** - Internal Disclosure: Release of information by the Galileo system to hackers or by staff, possibly coerced.

Data Spoofing: synthesizing valid signals in advance of their application.

- **TAS** - Time Accurate Spoofing: Use information available in advance and not requiring large time errors on the receiver.

Pseudorange Spoofing: Synthesizing false signals with valid data in near real-time to manipulate the pseudoranges to impact PVT with a great amount of freedom. This is also called Replay Attack.

- **TDR** - Time Delayed Replay: Replay of the valid signal that has just been received with relative shifts for each satellite. It is enabled by leveraging inaccuracies in secondary timing on the receiver; possibly imposed using a burst of jamming or by intentional power cycling by the user.
- **CDR** - Code Detection and Replay attack: Early detection and late commitment of unpredictable symbols to impose pseudorange shifts without requiring time inaccuracies on the receiver [11], [12]. This is also called Security Code Estimation and Replay (SCER) attack [4].

Meaconing: re-broadcast of the valid signal with delays, potentially selective per satellite.

- **TOM** - Time Only Meaconing.
- **PTM** – Position and Time Meaconing, with constrained impact on position and time.

Tampering: interfering with the user equipment.

- **LCT** - Tampering with local cryptographic data.

The means of each attack group to execute these threats are assessed and reported in Table 2. Note that this relates to the attacker executing the threat themselves. For example, an individual user is not capable of obtaining secure information from Galileo. The fact that they may use it if someone else exposes it is a consequential impact rather than direct risk.

In this analysis, the means to launch an attack are assessed independently of the mitigation, so while an attacker might have the means to launch an attack, it does not mean that the system is vulnerable to it.

	IO	LC	PA	OC	OA
CCA	0	0	1	2	4
FCA	1	1	3	2	4
EXD	2	3	4	4	4
IND	0	0	3	1	4
TAS	4	2	3	3	4
TDR	3	2	3	3	4
CDR	1	1	3	3	4
TOM	4	2	3	3	4
PTM	0	0	3	2	4
LCT	4	1	0	1	4

Table 2. Means of executing the threats defined in the text (rows) by threat sources defined in the text (columns) from fully capable (4) down to incapable (0)

Before assessing the technical realizations of these threats and vulnerabilities of systems using OS Authentication, it is necessary first to describe the TESLA protocol in more detail.

TESLA PROTOCOL FEATURES

The realization of the OS Authentication service under consideration is broadly defined in [1] and only briefly described here before recommending some adaptations and parameter selection. The level of security provided by the features described here is analysed in the later section on cryptographic parameters.

The TESLA protocol allows a sender to authenticate messages by means of message authentication codes (MACs) computed using a symmetric key that is transmitted at a later time. The protocol is suited for securing messages with a short lifetime. The receivers buffer the messages and MAC values. When the receivers obtain the key, they first verify the key and subsequently the MAC values. Because the key used to generate the MAC is kept private until the MAC is transmitted, and provided it is sufficiently difficult to falsify the MAC, the receiver has confidence that only the trusted source could have produced the MAC.

To place trust in the key and thus in the MAC, it must be provable that the key originated from the trusted source. This can be achieved by providing a digital signature of the key. However, in order to avoid the overhead of doing this for each key, the used key derives to a signed root key through a one-way chain. The trust is therefore dependent on the security of the chain derivation and irreversibility as well as on the signature.

The navigation data authenticated by this TESLA protocol could be anything. The protocol defined in [1] provides enough flexibility to support cross-authentication of navigation data for satellites within the Galileo constellation other than the one transmitting the authentication data, authentication of GPS satellite data, UTC time offsets, ionospheric modeling parameters, and data from other possible systems such as SBAS. Since the navigation comprises data that is refreshed at different rates and has different scope of applicability, it is partitioned into several message types (see the details in [1]).

Besides providing authenticity for navigation data, the authentication data and keys introduce a component to the OS data that is both trustworthy and unpredictable. This opens up the possibility of using the OS to counteract the CDR attack [4].

MITIGATION ANALYSIS

The potential mitigation provided by the TESLA protocol against the previously defined threats is assessed here with a simple mitigated/not mitigated score. The impacts of residual risks are described later.

CCA - Current Cryptanalysis: The TESLA OS authentication solution relies upon cryptographic functions. The cryptographic functions and parameters are proposed later to present a negligible residual vulnerability to current cryptographic attacks and are thus considered fully mitigated.

FCA - Future Cryptanalysis: There is a residual risk that new cryptanalysis techniques are discovered that weakens the security of the system.

EXD - External Disclosure: Disclosure of secure material by users and organizations outside of the Galileo system is fully mitigated by the fact that no private key material needs to be shared with these parties.

IND - Internal Disclosure: There is a residual risk that a party with intended or gained access to secure Galileo information may disclose secure material (EXD attack). We do not conjecture the vulnerabilities or procedural mitigations that may prevent this, but we will discuss the potential impact.

TAS - Time Accurate Spoofing: The TESLA protocol provides complete protection against prior evaluated spoofing.

TDR - Time Delayed Replay: The TESLA protocol does not provide complete protection against Time Delayed Replay, but it may be used in combination with additional external mitigation.

CDR - Code Detection and Replay: This attack is not mitigated for NMA and cannot be mitigated by cross-authentication. The authentication data can contribute to mitigation using additional techniques highlighted later.

TOM - Time Only Meaconing: TESLA provides no specific protection against this attack.

PTM - Position and Time Meaconing: TESLA provides no specific protection against this attack.

LCT - Local Cryptographic Tampering: The public key certificate in the receiver could be substituted so that trust may be incorrectly placed in the wrong signatures. There is no mitigation by the TESLA protocol and there is therefore a residual risk for a system that does not inherently protect this.

RESIDUAL RISK ANALYSIS

Accounting for the mitigations provided by the TESLA protocol alone, we assess the residual risks for each service use category.

The residual risks are assessed by multiplying, for each attack group, the weighting factors for the likelihood of an attack (Table 1) by the means to execute an attack (Table 2). As a summary, the results are summed across the attack groups to produce a consolidated residual risk for each service use.

The CCA, EXD and TAS threats are fully mitigated; the residual risks are shown in Table 3.

	LVT	HVT	POT	HVP	TIM	
FCA	15	35	18	30	27	125
IND	7	29	8	25	26	95
TDR	27	41	32	35	28	163
CDR	14	36	18	31	25	124
TOM	0	0	0	0	28	28
PTM	8	33	10	29	0	80
LCT	20	0	22	0	0	42
	91	174	108	150	134	

Table 3. Consolidated residual risk (zero is no risk) to each service category (columns) by each threat class (rows). Row and column sums are shown in grey.

The TESLA solution provides the means to further mitigate the residual risks when combined with additional factors which are assessed in the next sections.

PROCEDURAL MITIGATION

The impact of the residual risks of FCA and IND can be mitigated by employing procedures to detect the attack,

having a means to respond to the threat occurrence having been detected.

Public Key Infrastructure
To mitigate against the residual risks of future weaknesses being found in asymmetric cryptography (e.g. ECDSA) (FCA attack) and against private key exposure (IND attack), it is necessary to be able to send out root keys signed with new signatures or signature schemes.
The TESLA protocol allows for the same root key to be distributed under multiple different signature schemes or key pairs, thus allowing for a transition period as old devices are upgraded. During transition the root key distribution is mildly degraded in that it takes longer before the information that a specific receiver can process is repeated.

Key Chain switches
Key chains can be replaced at ease with any length of transition period before switching. In case of emergency switches, the worst case for a receiver that has been offline is that it may take a minute to re-establish the root of trust.

Extending Key Length
The length of the TESLA key can be extended to mitigate FCA with the key length being defined when the root is distributed. This must be done in a single step when a new key is used, as only a single key length can be practically supported. The current protocol can cope with longer keys (up to 128 bits) without a receiver upgrade. The result of increasing key length would be a potential degradation in the service performance (mainly Time Before Authentication (TBA); the impact in Authentication Error Rate (AER) would be low, if any. More details on NMA key performance indicators are presented in [13]).

Changing MAC algorithm
It is possible to change the MAC algorithm and length to mitigate FCA. The parameters are defined when the root is distributed. All MACs need to be the same length.

EXTERNAL MITIGATION

The remaining residual risks are TDR, CDR, TOM, PTM and LCT**Error! Reference source not found.**. It is up to the service provider or user to mitigate these according to their risk appetite and the prevalence of the attack at the time of use.

TDR: To mitigate TDR the receiver must have a trusted time source. To prevent TDR from being practical, the time source only needs to be reliable to several seconds.

TOM, PTM: A very precise clock source or antenna arrays can be used to mitigate against meaconing (TOM and PTM) being initiated during operation, with the authentic GNSS being used to provide accuracy.

CDR: CDR mitigation is possible with a statistical analysis of the received symbol energy [4]. Some statistics must be computed over unpredictable symbols; the higher the rate of unpredictable symbols, the better. The unpredictable bit rates are outlined in [1] and are characterized in more detail in the next sections. CDR mitigation can be reinforced by the use of trusted clocks, inertial sensors, or integrity monitoring at signal or measurement level in the receiver.

LCT: To mitigate against substitution of the public key, the receiver could be required to report the public key it has used. This could be built into the receiver by using a public key for encrypting the authenticated data, using a key with the same root of trust as the authenticating key.

CRYPTOGRAPHIC PARAMETERS

As explained previously, the keys are derived from a one-way chain, making verification possible without the use of extra keying material. Let F denote a one-way function. During setup, the key center picks a random K_n and subsequently computes:

$$K_{i-1} = F_i(K_i), \quad \text{for } i = n, n-1, \ldots, 1 \qquad (1)$$

Here, we allow the one-way function to differ for each iteration in some known way, which we will justify later. This means that K_{i-1} can be revealed without exposing K_i, and that, once K_i is revealed, the fact that it computes to K_{i-1} and thus is part of the chain can be verified.

In order to instantiate the authentication mechanism, we need to choose several algorithms and parameters:

- A MAC algorithm and length.
- A one-way function for $F_i()$ to compute the K_i and the key length.
- A digital signature algorithm to secure the distribution of root keys.

Whereas we have some constraints on the computation power available to generate and verify tags, keys and digital signatures, the more important constraint is the bandwidth that is consumed for their distribution. The minimum security requirements for each of these algorithms is analyzed in the following subsections.

MAC Algorithm

The most frequently transmitted items are the authentication tags produced by the MAC algorithms, so they should be as short as possible.

There is a strong correlation between successive navigation data items: the message protocol is very

strictly defined and hence the MAC is not required to sign arbitrary messages. There is therefore very little freedom to produce fake navigation messages tuned with unimportant bits to produce the same MAC as the authentic navigation data and so the minimum MAC length can be short.

We propose a minimum MAC length of 10 bits. A MAC of this length has less than one in a thousand chance of being guessed correctly and less than one in a trillion chance of 4 MACs being guessed in sequence to support spoofed positioning. It may therefore be considered both unpredictable and providing authentication for valid data.

For such short tag lengths, there are many alternatives available. We opted to use HMAC-SHA256 and to truncate the outputs to the desired length. This choice was motivated mostly by the fact that SHA256 is also used in other places in this protocol; its security is well-known and documented in [14].

Chain Analysis

The values K_i form the second most frequently transmitted data items. Their size has a significant impact on the bandwidth. In accordance with [14], we consider 80 bits to be the minimum key length required to achieve security. Report [14] contains the results of the security analysis that some authors of this paper performed to confirm that 80 bits is sufficient for our application.

The construction of secure chains is less studied. Therefore, we investigated several possible attack strategies and report our findings here.

The mechanism for the one-way function that is proposed is based upon the use of a cryptographic hash function for performance reasons. We opted to use the hash function SHA-256 and to truncate the output to the desired length.

The most dangerous attacks appear to be the attacks based on brute-force inversion of the cryptographic hash function. As soon as K_0 has been distributed, an attacker can start trying out random values for K_i and check whether they resolve to K_0. The defence against this is to reduce the chance of success to a suitably small number.

Dedicated hardware (which is nowadays being produced for Bitcoin mining) can perform approximately 2^{32} hashes per second and per US dollar (USD). Hence, by spending 8 million USD on hardware, an attacker can compute 2^{80} hashes within a year.

If a fixed hash function were to be used, $F_i = F$, such that $F_i(K_i) = K_0$ then an attacker can start with any random key and produce a chain until they hit K_0 (or close a loop and start again). This is currently feasible within a year, which would break all future uses of the chain, which is clearly unsuitable.

Rather than simply building a chain by repeated application of this hash function, the proposed one-way function also incorporates two other factors:

- A form of counter whose value is known for each iteration step i.
- An unpredictable value α, whose value is known only shortly before the chain is used.

So the TESLA chain can be refined as:

$$K_{i-1} = \text{Hash}(\alpha, i, K_i), \text{ for } i = n, n-1, \ldots, 1 \quad (2)$$

The counter means that the attacker is forced to commit his attack to a fixed iteration number t. If t applications of (2) do not result in K_0, then the attacker has to restart the whole chain.

It is nevertheless still possible to pre-compute N hash chains of length *t* and store the tuples of the first and last entries (L_t, L_0). Once K_0 has been published, the attacker can look in his table for a tuple with $L_0=K_0$. If the table contains a suitable L_0, then he can use the corresponding L_t to re-compute K_i-values ($i < t$) that will be accepted by the receivers. Assuming that all L_0 are different, this attack has a success rate of $N/2^{80}$. The limiting factor here is the cost of storage space. It is currently not feasible to store 2^{80} hash chains.

However, another strategy is possible. It is called Hellman's time-memory trade-off attack [15], with storage requirements and computation time in between pure pre-computation and pure on-demand computation strategies.

Let G denote t applications of F, i.e. $G(K_t) = K_0$. During a pre-computation stage, the attacker randomly picks 2^{27} starting points and computes hash chains based on 2^{27} applications of G each, hence in total 2^{54} iterations of G. He stores the start- and end-points in a table.

Once K_0 is published, the attacker uses the table to recover K_i. This works as follows. First, the attacker checks if K_0 is one of the endpoints in the table. If this is the case, then the attacker can quickly determine K_t by applying G 2^{27}-1 times to the corresponding start-point. Else, the attacker checks if $G(K_0)$ is one of the end-points in the table. If this is the case, then K_t can be found by applying G 2^{27}-2 times to the corresponding starting point. This attack requires approximately 2^{27} iterations of G and has a success probability of 2^{-27}. By repeating the attack 2^{27} times with small variations on the function G, the attacker can recover K_0 with a success probability of 63%, a total effort $T=2^{54}$ and a total memory complexity $M=2^{54}$.

The attacker can exchange M for T, as long as M^2T equals 2^{160}. So, doubling the size of the tables decreases the cost of the computation hardware with a factor of 4.

The time-memory trade-off attack has a pre-computation stage requiring 2^{80} iterations of G. The attack is countered by the unpredictable parameter "α" into (2), since the attacker can start the computations only after α is published. Any pre-computation can be used only for a specific α, which is only revealed a relatively short time before the new chain is used. The chain is then used for a limited lifetime before a new chain is produced using a new value for α. Using a large enough α and short enough chain will eliminate the time-memory trade-off attack.

The lifetime of a chain (L_C) is determined by the investment that a reasonable attacker is willing to make in order to succeed in the brute-force attack. For example, if L_C is set to one month, then an investment of over 100 million USD is required to recover K_n before it becomes obsolete. According to Moore's Law, to keep a constant level of security, 4 bits should be added to the key length every 3 years, though improvements in attack algorithms may require greater increases.

The length of α is determined by the maximum tolerable probability that an attacker can guess the value of α, p_α, for one of the chains used during the full period of service L_S before the crypto parameters such as key length or algorithm are updated.

$$L_\alpha = \log_2 L_S - \log_2 L_C - \log_2 p_\alpha \quad (3)$$

By setting p_α to 2^{-40}, we ensure that the attacker has a negligible chance to pre-compute a useful table. If we set L_S to 10 years (though in practice crypto-parameters are rarely kept fixed for periods longer than 5 years), then we obtain that a length of 49 bits for α is sufficient.

Because the one-way function is a many-to-one mapping, there can be more than one key that iterates to the same trusted root K_0, only one of which is the key that is in the chain generated by the key source (the source of trust); others would be incorrectly trusted. Each new iteration introduces a small chance of a collision, so that the chance of a collision grows with the length of the chain being pre-computed. Once two chains with the same one-way function collide at a common iteration number, they coalesce. This fact improves the chance of an attacker finding a key starting at iteration t, K_t, that iterates to trusted root K_0 if one is not concerned with finding the actual K_t used by the source of trust. Many attacks need only to find any key that will resolve to the trusted root.

One answer to this issue is to constrain the length of the chain. However switching to a new chain presents performance considerations for all receivers, as any receiver that has been off-line while the new chain root was publicized will be unable to authenticate until the new chain is trusted; and this is a much slower activity.

This particular consideration can instead be solved without further constraint on the chain length by sending out digital signatures of intermediate keys K_N *after* they have already been revealed. This allows a device to optionally check the key it has received against the intermediate key to reduce the length of the chain it is relying upon to reach the trusted value. Transmitting these intermediate keys, or *floating root keys*, also has the benefit of enabling devices to validate a key in fewer steps when starting from a cold start.

Digital Signature

We considered several alternatives for the digital signature algorithm to secure the distribution of K_0. Because every K_0 value remains valid for a relatively long time, the value needs to be transmitted less often than the TESLA keys. However, having the digital signatures with a short length remains an important design criterion because it uses less bandwidth and reduces the time (and requirement for a clear sky during that time) to establish trust from a cold start.

The length of RSA signatures equals the length of the modulus. At least 1024 bits are required, and often 3072 bits or more are recommended [14]. The message recovery technique, where the message is (partially) encoded in the signature, allows us to reduce the total length of message plus signature. However, the minimum length of 1024 bits (or 3072 bits) remains.

Elliptic-Curve cryptography (ECC) allows us to reduce the length of the keys as well as the length of the signatures. For example, the security of a 1024-bit RSA key corresponds to a 160-bit key in ECC; a 3072-bit RSA key offers the same security as a 256-bit ECC-key. This difference is partially offset by the fact that the current standards for digital signatures using elliptic curves require a storage of twice the length of the key. Hence a 1024-bit RSA signature corresponds to a 320-bit ECC signature.

The DSA algorithm is based on a mathematical problem similar to the RSA problem. It requires keys of the same length as RSA. The signatures however, have the same size as ECC signatures of the same security level.

We opted for EC-DSA. It will be seen that the protocol allows for transition to new signature choices without impacting the TESLA mechanism.

PROTOCOL

The protocol is described at some length in [1]; only a few additional considerations are specified and justified here covering:

- Chain Function Specification
- Authenticated Message Specification
- Protocol Mapping

Chain Function Specification

The key K_n will be a 256-bit random number truncated to K_{len} which we have justified as 80 bits. Each key in the chain will be generated down to K_0 as shown below:

$$K_m=\text{trunc}(K_{len}, \text{Hash}(K_{m+1} \,||\, GST_{SF} || \,\alpha) \quad (4)$$

where

- K_m is the key to be generated.
- trunc(n,p) is the truncation function whereby the message p is truncated to the n MSB.
- K_{len} is the length of the key (80 bits).
- Hash is the selected hash function (SHA-256).
- K_{m+1} is the previous key in the chain.
- GST_{SF} is the Galileo System Time at the start of the subframe in which the key will be applied.
- α is the unpredictable pattern that is signed and transmitted with K_0.

H-K-root definition

Based on the aforementioned requirements, Figure 1 proposes a preliminary definition of the H-K-root section (Header and Root Key section), occupying 8 of the 40 bps of every "Reserved 1" field of the Galileo I/NAV message. It includes a global header with flags for managing the overall service status, the chain ID in force, some additional flags and the ID of the digital signature and block being transmitted at a given subframe. It can serve to provide floating root keys of the same chain at different times, as mentioned before. Further details can be found in [16].

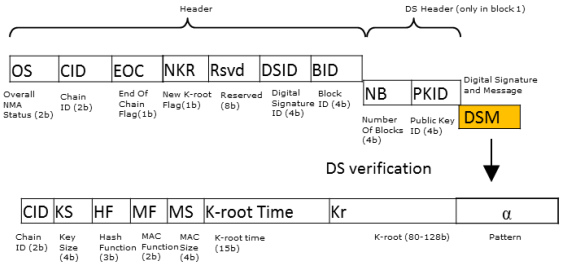

Figure 1. H-K-root section

Authenticated Message Specification

To maintain high signal unpredictability, it is important to ensure that each MAC remains unpredictable. As multiple satellites may be authenticating the same navigation data using the same key at differing times, it is essential that each MAC is different, and hence the message being signed must differ for every MAC transmitted. To avoid replication across satellites, both the identity (and constellation) of the satellite being authenticated and the authenticating satellite are included in the signed message. In addition to this, to avoid duplication within a single satellite, a counter is combined into the MAC.

With these two modifications, the message authentication tag produced to authenticate "navdata" is now defined as:

$$\text{tag} = \text{trunc}(N, \text{MAC}(K,M)) \quad (5)$$

$$M = (SVID\_N \,||\, SVID\_A \,||\, GST_{SF} \,||\, CTR \,||\, \text{navdata}) \quad (6)$$

where

- M is the message to be authenticated
- SVID_N is the satellite ID being authenticated.
- SVID_A is the satellite ID providing authentication.
- GST_{SF} is the Galileo system time at the start of the sub-frame in which the MAC is transmitted.
- CTR is a counter with the position of the MAC in the MAC-K section starting from 1.
- N is the truncated MAC length
- MAC is the MAC function used (HMAC-SHA-256)
- K is the key from the one-way chain used for the MAC.

A standard format is defined for the SVID which identifies satellite number and constellation [1]. The receiver is able to construct all of the necessary information to validate this MAC based upon data transmitted with the MAC [1]. The Key and MAC algorithm properties are sent out with the signed root key though they will be changed rarely if ever.

PROTOCOL MAPPING

In [1] the protocol structure was defined to utilize the 40 "spare bits" available every other page of the E1-B message, with the first 8 of these bits allocated to one structure called "H-K-Root Section", and the remaining 32 bits to the so-called "MAC-K Section".

The H-K-Root section is transmitted synchronously with the 30-second I/NAV subframe. It contains some global information which should be, but does not need to be, checked regularly, and the signed root key, which only needs to be processed when new root key information is needed by the receiver. The signed root key is broken down into blocks that can be constructed over a period of time from multiple satellites; it was shown that a root key could be received and validated in 1 minute [1] if there

are enough NMA satellites in view, but this is rarely a critical factor.

The several MAC-K sections may fit into one subframe. With each MAC-K section there is a single key and a number of MACs, so there are some trade-offs: fewer longer sections, the longer the waiting time until a single authenticity check can be performed, but more MACs can be delivered in 30s, providing the possibility to perform more authentications.

In [16], the MAC-K Section partitioning has been analyzed, comparing 2 and 3 MAC-K Sections (with 15 and 10 second cycle lengths respectively) per subframe for a range of MAC and key lengths. The analysis included the number of unpredictable bits per sub-frame (UBS) and the Number of Authentication bits (NA) to authenticate 4 satellites. Results are reported in Table 4 for the minimum allowable MAC length of 10 bits and a key length of 80. Also Table 4 shows the Average Time Before Key Validation (ATBKV), which gives a measure of the time before the signal is confirmed to be authentic. The average time to provide authentication of the navigation data is longer, depending also on how the MACs are transmitted.

KLen	Sections	MACs	ATBKV (s)	UBS (bits)	NA (bits)
84	2	12	13.9	248	188
82	3	9	10.5	276	186

Table 4. For 2 and 3 MAC-K sections, 10 bit MACs and optimally chosen key lengths, the average time before key validation (ATBKV), UBS (Unpredictable bits per subframe) and Number of Authentication bits (NA) required to authenticate 4 satellites.

The ATBKV evaluation is obtained from the results in Table 5. It shows the layout of keys in "spare bit" pages assuming the proposed 32 bit allocation to MAC-K sections. The table also shows in each cell the number of pages that must be received from each starting point. In the 2 MAC-K Section case, the first 84 bits of the key are spread over 4 pages, whereas in all other cases they fit into 3 pages. This occurs because the first MAC-K section uses only 16 bits of page number 8. The average number of pages is multiplied by two to convert it to seconds and augmented by 0.5 seconds to account for odd or even start pages.

Sections	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Avg
2	8	7	6	5	4	10	9	8	7	6	5	4	3	10	9	6.7
3	5	4	3	7	6	5	4	3	7	6	5	4	3	7	6	5.0

Table 5. The placement of keys in each "spare bits" page is shown in green for the 2 and 3 MAC-K section cases. Each cell contains the number of pages that must be received if that cell was the first received page. The average represents the ATBKV in seconds.

The Unpredictable Bits per Subframe (UBS) represents the number of unpredictable bits in a single MAC-K Section, and it is equal to the length of the MACs plus the key length minus a number of key bits that can be considered as predictable by an attacker (being this number set to 20 in our case, see [16]). Therefore, 20 bits are subtracted on the basis that once the first bits of the key are known, the last bits of the key may be predicted. This would be however very challenging to leverage except in some few cases, so the 20-bit subtraction can be considered quite conservative and, in practice, UBS can be larger by a few bits. In any case, this fact no or little impact in the conclusions of this paper.

PERFORMANCE

The bits available for OS authentication can only be transmitted by satellites with an active uplink. It is therefore important to understand how the Quality of Service is affected for the expected population of Galileo uplink station (ULS) antennae.

Two scenarios are considered to analyze both Data only and pseudorange OS authentication:

- Short-term: 16 ULS antennae
- Mid-term: 20 ULS antennae

The Short-term scenario assumes 4 antennae at Kourou, Svalbard and Reunion, and 2 at Papete and Noumea. The Mid-term scenario assumes 4 antennae at all stations. The antennae are assumed to be all in use transmitting authentication data. Visibility simulations have been performed using GMVs constellation simulator under the AALECS (Authentic and Accurate Location Experimentation with the Commercial Service) project and their results are shown in the following section. The simulation is based on uplink assumptions that, while generally considered compliant with current Galileo uplink requirements, are not optimized for the proposed NMA concept. **Therefore, the short-term and long-term scenarios can provide better availability results than those shown in this paper.**

Data-Only Authentication

For Data-only authentication the receiver can benefit from cross-authentication. In this case it is possible to authenticate Galileo and GPS satellites in view when fewer than four satellites in view are transmitting authentication data.

If one supposes that nearest neighbor satellites are evenly distributed in the solid angle around an authenticating satellite, then it is possible to evaluate the average number (average over all transmitting satellite positions) of nearest neighbors that should be expected to be in view as a function of the receiver masking angle (see Figure 2).

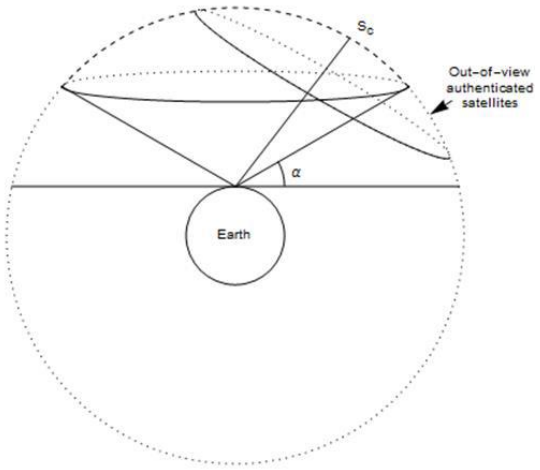

Figure 2. In-view and Out-of-view satellites in a cross-authentication scheme

The result of this analysis for a 24-operational satellite constellation is summarized in Table 6.

		Neighbours Authenticated											
		1	2	3	4	5	6	7	8	9	10	11	12
Masking Angle	5	0.9	1.7	2.4	3.1	3.7	4.3	4.9	5.4	5.8	6.3	6.7	7.1
	10	0.9	1.6	2.3	3.0	3.6	4.1	4.6	5.1	5.5	5.9	6.2	6.5
	15	0.9	1.6	2.3	2.9	3.4	3.9	4.4	4.8	5.2	5.5	5.8	6.0
	20	0.8	1.5	2.2	2.7	3.3	3.7	4.1	4.4	4.8	5.0	5.2	5.4
	25	0.8	1.5	2.1	2.6	3.1	3.5	3.9	4.1	4.4	4.6	4.8	4.9
	30	0.8	1.5	2.0	2.5	2.9	3.2	3.5	3.7	3.9	4.0	4.1	4.2
	35	0.8	1.4	1.9	2.3	2.7	3.0	3.2	3.4	3.5	3.5	3.6	3.6
	40	0.8	1.3	1.8	2.1	2.4	2.6	2.8	2.9	2.9	2.9	2.9	2.9

Table 6. Average number of a satellite's nearest neighbors that a receiver should see as a function of masking angle (rows) and the number of neighbors authenticated (columns).

The result could be applied, as a slight under-estimate, to GPS near neighbors assuming a 23 operational satellite GPS constellation. One can see that with only one satellite in view and an extreme masking angle of 40°, it is not typically possible to authenticate a single constellation alone, mainly because it is very likely that there are not enough satellites in view of a single constellation. However, with 5 cross-authenticated nearest neighbors across GPS and Galileo (e,g, distributed as 3 and 2), we should expect 3 neighbors to be in view the majority of the time.

Typically, in urban canyons the sky line is not homogeneous and instead an interpolation of results with a lower masking angle is useful. For a masking angle of 20°, and a single satellite in view, 5 cross-authenticated neighbors should suffice for Galileo alone, or 4 if cross-authenticating with GPS. Moreover, at 20° the probability of having two authenticating satellites in view is higher, and in this case, 3 cross-authenticated neighbors should suffice.

The probabilities of seeing at least one authenticating satellite when the masking angle is 40° and at least 2 when the masking angle is 20° are shown in the Figures 3-6. The simulation allowed for 1% of the possible uplink time to be lost for acquisition.

Data authentication availability at low elevations (5º, 10º) is generally as high as standard navigation availability and not shown in this paper. Figure 3 shows that with a 40° mask even seeing a single transmitting satellite occurs less than 75% of the time in highly populated areas for both scenarios, reflecting the very constrained visibility of the sky. For a 20° masking angle, two or more will be in view a similar proportion of the time.

The conclusion is that combined Galileo plus GPS authentication is viable for 5 cross-authentications in environments with 20° masking angle, and significant blocking occurs for a 40° masking angle in both scenarios. Cross-authentication of Galileo alone is viable but intermittent (72%) for a 20° mask and becomes impractical if blocking up to 40° becomes significant. We must note that users can navigate with authenticated data once a valid key and four MACs are received, without continuously updating their data authentication status other than for antireplay protection purposes.

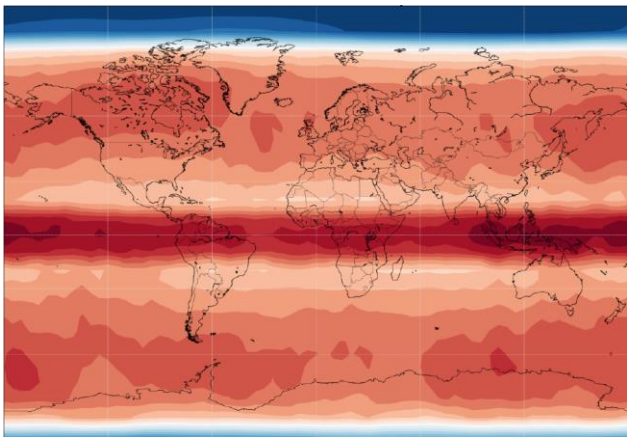

Figure 3. Probability of 1+ satellite in view, 40° mask, Short-term Scenario. Color scale is linear from 51% (dark red) through 75% (white) to 99% (dark blue)

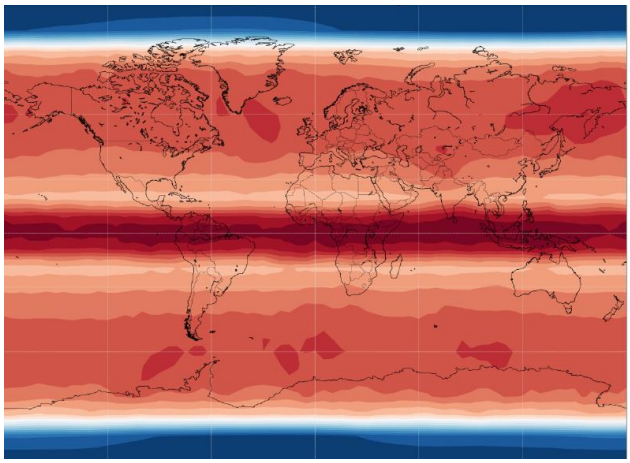

Figure 4. Probability of 1+ satellite in view, 40° mask, Mid-term Scenario. Color scale is linear from 54% (dark red) through 77% (white) to 99% (dark blue)

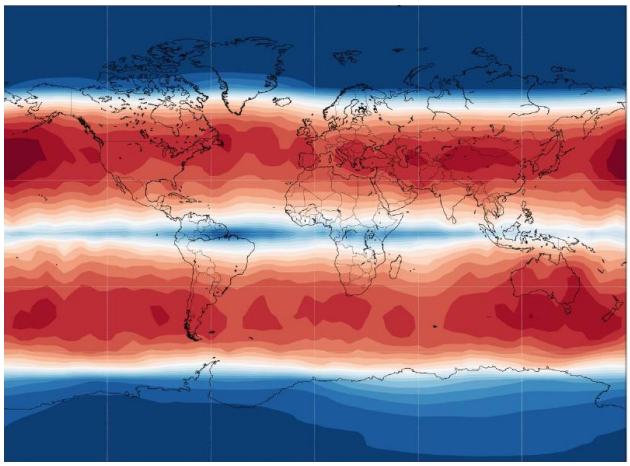

Figure 5. Probability of 2+ satellites in view, 20° mask, Short-term Scenario. Color scale is linear from 72% (dark red) through 85% (white) to 99.7% (dark blue)

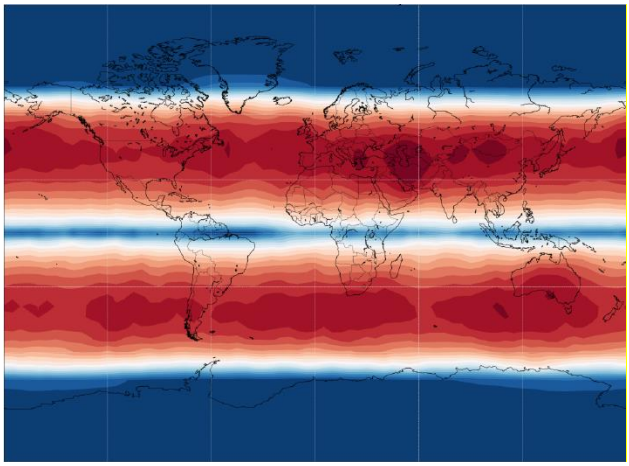

Figure 6. Probability of 2+ satellites in view, 20° mask, Mid-Term Scenario. Color scale is linear from 75% (dark red) through 87% (white) to 99.7% (dark blue)

There is flexibility in how often and which cross-authentication data should be transmitted. The simplest model is to cycle over *NN* nearest neighbors, where *NN=5* has been shown to be sufficient. The policy of cross-authenticating a satellite that is also transmitting authentication data is also to be optimized: while these satellites can be excluded, as the OS Authentication module knows in real-time which satellites are connected, the level of MAC duplication of cross-authenticated and self-authenticated satellites should be similar. As well as authenticating ephemeris, the ionosphere corrections will need to be authenticated less often and "slow authentication" MACs can be included at intervals to detect delayed replay attacks (TDR), as abovementioned.

The earlier analysis showed that with 2 MAC-K Sections per subframe, 12 MACs could be included per subframe suggesting a layout in each section of "self-authentication", 3-4 neighbors plus 2-1 "other" per section. Full cross-authentication will often be possible within half a subframe. If more "other" authentications are needed, there is room in a subframe for up to 5 without compromising the higher rate of self-authentication.

With 3 MAC-K Sections per sub-frame (and the associated improvement in TBA and UBS), 9 MACs could be included per subframe suggesting a layout of self-authenticating in each subframe, leaving 8 MACs in a subframe allowing for 6 or 7 neighbor authentications and 2 or 1 additional authentications, respectively.

In any case, the proposed NMA concept is configurable as to the number of MACs and key sections for a given chain, in a transparent way to the receivers. This configurability may allow an adaptive approach depending on the number of satellites and uplink antennae available at a given time of the system lifetime.

Pseudorange Replay Protection

As previously stated, the OS Authentication proposal introduces a sufficient number of unpredictable bits to support statistical analysis to detect CDR attacks. Here we report the performance of positioning only with self-authenticating satellites.

Figures 7 and 8 show that in the short-term scenario the probability of being able to calculate a position with 4 authenticating satellites whilst protected from CDR attack is about 80% over most of the developed world, but does drop off in places to 70%. In the mid-term scenario, the probability always exceeds 80% and is approaching 90%. It must be noted that having less than 4 authenticating (and therefore CDR-protected) satellites, when combined with cross-authenticated (and therefore non-CDR protected) satellites and other constraints as user

dynamics, clock drift, or external geometric information, such as height or digital terrain models, can significantly limit the possibilities of an attacker. In practice, systems requiring this protection will be tracking the constellation over extended periods, therefore the periods when complete protection is not guaranteed would be easily bridged from extrapolating prior authenticity.

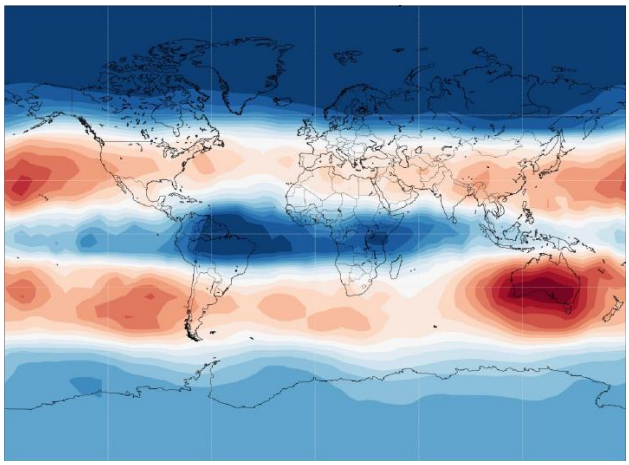

Figure 7. Probability of 4+ satellites in view, 5° mask, Short-Term Scenario. Colour scale is linear from 69% (Red) through 84% (white) to 99% (blue).

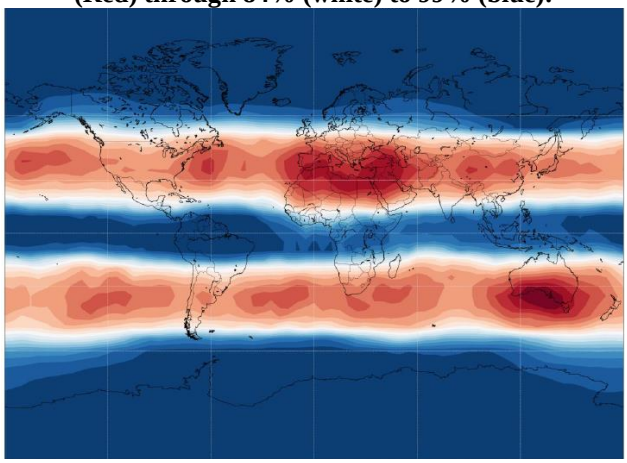

Figure 8. Probability of 4+ satellites in view, 5° mask, Mid-Term Scenario. Colour scale is linear from 80% (Red) through 90% (white) to 99.6% (blue).

DISCLAIMER

The material in this paper does not represent any official view of the EU or its Member States. The solutions proposed will not necessarily be included in future Galileo operational services.

CONCLUSIONS

This paper has presented a list of threats, and its criticality for different potential user types of Galileo NMA (Navigation Message Authentication) based on the TESLA protocol. The paper has proposed some concrete mitigation actions concerning the protocol implementation to minimize the residual risk of the most relevant threats that can be mitigated, particularly related to FCA (future cryptanalysis techniques) and CDR (Code Detection and Replay). Based on this analysis, certain aspects of the TESLA protocol can be enhanced for further protection: Concerning FCA, the increase of entropy in the chain generation through counters (or time stamps) and random patterns is explicitly proposed in order to mitigate pre-computation attacks. The use of floating root keys over the lifetime of the chain is also proposed to mitigate the effect of collisions. As regards CDR, the signal unpredictability can be increased to protect authenticating satellites against signal replay.

For the selected parameters multi-constellation NMA can be easily achieved for open sky, and in environments with up to a masking angle of 40° with some limitations. We have also shown that authentication using four validated signals presents a reasonable performance at a 5°masking angle, this performance being subject to improvements in the downlink capabilities.

We have identified two potential MAC-K layouts that achieve this performance:

- 9 MACs and 3 keys per subframe: the 3-section layout provides better bit unpredictability for CDR defense and could provide spoofing detection some few seconds earlier, and slightly faster self-authentication.
- 12 MACs and 2 keys per subframe: the 2-section layout provides more flexibility to support more regular authentication of other data. This layout can also more easily adapt to potential changes in MAC length and key length, so it may become the only option if longer keys or MACs than the currently proposed ones (82 bits and 10 bits, respectively) are required in the future.

In any case, the proposed NMA protocol allows the MAC-K layout to change over time, allowing it to cope with future attacks requiring longer keys or MACs in a backward-compatible way.

ACKNOWLEDGEMENTS

We would like to acknowledge discussions with G. Tobias, I. Rodriguez, M. Canale, A. Rolfe, P. Thomas and A. Kerns.

REFERENCES

[1] I. Fernandez-Hernandez, V. Rijmen, G. Seco-Granados, J. Simón, J. D. Calle and I. Rodríguez, "Design Drivers, Solutions and Robustness Assessment of

Navigation Message Authentication for the Galileo Open Service," *ION GNSS+ 2014*, 2014.

[2] J. A. Volpe, "Vulnerability Assessment of the Transport Infrastructure Relying on the Global Positioning System," *National Transportation Systems Center, U.S. DoT*, 2001.

[3] C. Wullems, O. Pozzobon and K. Kubik, "Signal Authentication and Integrity Schemes for Next Generation Global Navigation Satellite Systems," *the European Navigation Conference*, 2005.

[4] T. Humphreys, "Detection Strategy for Cryptographic GNSS Anti-Spoofing." *IEEE Transactions on Aerospace and Electronic Systems*, 2013.

[5] A. Perrig, R. Canetti, J. D. Tygar and D. Song, "The TESLA Broadcast Authentication Protocol," *CryptoBytes*, 2002.

[6] S. Lo and P. Enge, "Aviation Augmentation System Broadcasts," *IEEE/ION Position Location and Navigation Symposium (PLANS)*, 2010.

[7] Becker, G.T., Lo, S., De Lorenzo, D., Qiu, D., Paar, C., Enge, P., "Efficient Authentication Mechanisms for Navigation Systems - a Radio-Navigation Case Study," *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, Savannah, GA, September 2009, pp. 901-912.

[8] J. T. Curran, M. Paonni and J. Bishop, "Securing the Open-Service: A Candidate Navigation Message Authentication Scheme for Galileo E1 OS," *European Navigation Conference ENC 2014*, Rotterdam, 2014.

[9] A. J. Kerns, K. Wessons and T. Humphreys, "A Blueprint for Civil GPS Navigation Message Authentication," *Position, Location and Navigation Symposium*, Monterey, 2014.

[10] E. Carbonell, D. Calle, I. Rodríguez, G. Tobías, I. Fernández, "Galileo Commercial Service Demonstrator – Signal In Space Proof-Of-Concept." *7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, 2015.

[11] D. H. Arze Pando, D. H., "Distance-Decreasing Attack in Global Navigation Satellite System," *Report*, School of Computer and Communication Sciences (I&C), Swiss Federal Institute of Technology (EPFL), 2009.

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

[13] I. Fernández Hernández, "Authentication: Design Parameters and Service Concepts," *Proceedings of the European Navigation Conference*, 2014.

[14] ENISA: Algorithms, key size and parameters report, 2014.

[15] M. Hellman, "A Cryptanalytic Time - Memory Trade-Off," *IEEE Transactions on Information Theory*, Vol. 26, No. 4, July 1980.

[16] I. Fernández-Hernández, "Snapshot and Authentication Techniques For Satellite Navigation", Faculty of Engineering and Science, Aalborg University, June 2015.