Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Degree in
Telecommunication Engineering

Unconditionally Secure Authentication and Integrity Protection for the Galileo Open Service Signal

Supervisor:
Prof. Nicola Laurenti

Student:
Silvia Sturaro

Co-supervisor:
Gianluca Caparra

Academic Year 2015/2016

Unconditionally Secure Authentication and Integrity Protection for the Galileo Open Service Signal

SUPERVISOR: Prof. Nicola LAURENTI
CO-SUPERVISOR: Gianluca CAPARRA
STUDENT: Silvia STURARO

30th November 2015

To my parents who since the first day have loved me,
to Chiara who has grown with me,
and to Loris who has learned to love, and appreciate me.

Abstract

Currently, the operational Global Navigation Satellite Systems (GNSS) do not offer message, nor signal Authentication and Integrity Protection toward the Open Service (OS) users. But a well-motivated attacker equipped with a single antenna can successively perform several attacks, e.g. Jamming, and Signal-Synthesis attack to affect the position solution, or even a Replay attack which modify the timing awareness. Because of this threats, robust defending techniques are needed. The nascent European navigation system - Galileo, aspires to offer an E1 OS signal authentication and integrity protection mechanism integrated from the outset. Actually, a wide State-Of-Art about defende techniques already exists. But these all have a common vulnerability: they are based on *computationally* secure algorithms (e.g. Digital Signal Algorithm), or are designed only against a specific attack.
This thesis work, inserted into a collaborative project with the European Space Agency (ESA), aims at contributing to the issue by adopting a different approach. Namely, we deal the problem from the Physical Layer point of view, and without limiting the attacker capabilities and advantages. Furthermore, no specific authentication scheme is chosen or designed, but only generic channel models are considered. In this manner, a theoretic key entropy lower bound is defined for both Data and Signal level. Their performance is then presented considering some simplified channel and signal setting, but against a well-equipped attacker. In addition, the Data-layer bound is compared with the actual most promising protocol - TESLA. Therefore, the work provides a reference bound to guarantee *unconditionally* secure Authentication and Integrity Protection. Finally, the DLR channel model is exploited in order to extract some useful features (e.g. coherence time) within a more realistic scenery.

Acknowledgements

Concluding these five years of study, I would give my acknowledgements to some people who have been essential. I will start from Professor Nicola Laurenti, my thesis supervisor, he has passionately introduced me to the world of Information Secrecy and Security, and then he has given me the possibility of applying it to another interesting topic: the satellite navigation systems. I am very grateful for his helpfulness, kindness and extraordinary competence that he has always shown during the difficulties of the work. Next, I thank as well Gianluca Caparra, whose advices and experience have enormously helped me in speeding up the outcomes achievement. I wish say a big thank to both since I have felt part of a nice group, where one can find professionalism and pleasantness too. Further, I am completely satisfied about the chosen thesis argument, since it is extremely actual and useful; and it has enriched even more my knowledges.

Another important acknowledgement goes to my parents, Raffaele and Francesca. It is thank to them if today I have reached this beautiful goal. They have always supported me materially and morally, believing in my abilities even when I have doubted of myself. And if today I am happy about me, it is due to them. Thanks also to my sister Chiara, who in these years has had to bear my little free time, and nervous moments.

Then there is my greatest supporter, my fiancé Loris, who from the beginning has always been close to me with affection, sympathy and good tips. I thank him immensely for the confidence and patience shown me. And to him, who perfectly knows the telecommunications world, I dedicate this thesis.

Furthermore, I have had the pleasure to meet special and very brilliant people. Some of them have not been only fellow students, but have become great friends, I thank Marco, Mattia, Giulia, Maria, Chiara, Trinh and Thomas. They all know what means studying and working hard, together we have lived several worries and stressful moments. But we also have passed incredible memories, as the funny lunch times all together, or the beautiful adventures around Spain during the Erasmus semester.

Last but not least, I thank Ilaria and Valentina, that since the highschool have always been close to me. The endless chats with them, are always indispensable and pleasant.

A big thank to all of you, and everyone who in these years has crossed my way!

Silvia

Contents

Acronyms

BOC Binary Offset Carrier

CDMA Code Division Multiple Access

CS Commercial Service

DOS Denial-Of-Service

DS Digital Signature

DSSS direct sequence spread spectrum

EC Elliptic Curve

ECI Earth Centered Inertial

ECEF Earth Centered Earth Fixed

ENISA European Union Agency for Network and Information Security

FEC Forward Error Correction

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

GPST GPS Time

GST Galileo System Time

MAC Message Authentication Code

MEO Medium Earth Orbit

NMA Navigation Message Authentication

OS Open Service

PNT Position-Navigation-Timing

PRN Pseudo Random Noise

PRS Public Regulated Service

RAIM Receiver Autonomous Integrity Monitoring

SAR Search and Rescue Service

SCE Spreading Code Encryption

SCER Secure Code Estimation and Replay

SOL Safety Of Life

SV Space Vehicle

TESLA Timed Efficient Stream Loss-tolerant Authentication

TOA Time-Of-Arrival

UTC Universal Time Coordinates

Chapter 1

Introduction

1.1 Motivations

The science that deals with driving crafts, vehicles or people from one place to another is known as *navigation*. This is a daily-life skill that since ancient times man has developed, firstly using the senses, and then by identifying landmarks around him. With the terrestrial radio signals introduction the navigation accuracy has seen a first change, but the real revolution was led by the Global Navigation Satellite Systems (GNSS). This kind of systems provide us Position-Navigation-Timing (PNT) means satellite signals Time-Of-Arrival (TOA) [2].

The United States Global Positioning System (GPS) has been the first: it is operational since 1978 in military domain, and since 1994 has become globally available [2]. In 2000 the Selective Availability (i.e. an intentional signal degradation) was abolished, thus it has started to provide the civilian users with a relatively high precision service, and made GPS the world's most utilized satellite navigation system.

In addition, over the past two decades, the ever decreasing price of receiver devices has led to a significant integration of GNSS in people every-day life, becoming an important component in the information world, fully integrated with Internet and many other technologies [22]. Indeed a large number of applications make use of te GNSSs signal. It was initially employed in car and vehicles navigation, but nowadays there are also real-time traffic control, valuable goods traceability, safety during flight and landing operations, dangerous situations and emergencies assistance, and agriculture optimization; it is also used to assist the electricity distribution network, and finally ensure very accurate timestamps in financial trades. Furthermore, given the recent success of smartphones, apps and social networks, GNSS can be used also for trivial needs as the nearby services search, or to locate the places where we took our pictures [3].

However, the increasing use of GNSS, and the growing dependence on it have a drawback: it can stimulate an hacker to attempt illegitimate attacks, either in order to take advantage over someone/something (e.g. because of prospect for financial gain), or with terrorist purpose [22], [13]. As a matter of fact the existing GNSSs, such as GPS or the Russian GLONASS, offer no authentication of their signal towards their civilian users, so several types of attack against GNSS may be performed without being detected, as we will see in Chapter 2. Thus, in recent years awareness has risen that, given the amount of civilian positioning application requiring safety and security, also the GNSS Open Service (OS) need message authentication and integrity protection urgently.

Briefly, *message authentication* ensures that the message has been sent by the legitimate transmitter and not been forged by a false entity, while *integrity protection* guarantees that during the transmission no changes have been made to the message. So far, several techniques have been proposed, but the problem is particularly complex and requires different skills. For instance, in GNSS the unmodified data content does not guarantee signal integrity as well, because the signal can be artificially delayed by an attacker, to alter the ranging information. Thus, as we will see, it is necessary to work on two levels, namely Data and Physical layers.

Galileo is the new European GNSS that approaches even more the stage of opening. This latter has in charge several improvements over GPS, but the signal authentication holds a primary attention. Therefore, given the serious consequences that a lack of authentication mechanism can lead to, several resources are being invested in the Galileo E1 OS signal authentication and integrity protection design.

1.2 Thesis objective

The current thesis work is inserted into a partnership between the Department of Information Engineering from the University of Padua, and the European Space Agency (ESA). It is focused on the **Galileo** signal Authentication and Integrity protection. It takes especially into account the E1 band Galileo Open Service (OS) signal; thus herein we will refer to the corresponding technical features chosen by ESA for it [1].

Given the already proposed defense techniques state-of-art vulnerabilities, this thesis aims at treating the problem with a more robust approach, being as general as possible. Namely, via Information Theory tools we are going to identify two key bounds valid whatever the encoding/authentication scheme adopted, and the attacker skills. This latter aspect is even more important, that is, the current work aims at the *Unconditional* authentication and integrity protection.

1.3 Thesis structure

In **Chapter 2** the reader is introduced to the fundamentals of the modern satellite navigation systems, such as to its architecture, functioning and signals. In addition, a mathematical explanation of the position computation, and the techniques to deal with hardware imperfections will be given. Finally, all the possible signal impairments, and intentional attacks will be listed.

Chapter 3 is devoted to analyse the actual mechanisms for detecting and mitigating attacks. The first part of the chapter treats a series of checks at different layers (e.g. Position, Signal and Data) that one can perform to discover any signal inconsistency. While in the second part, we first give an introduction to useful cryptographic primitives, and then describe several cryptographic defense techniques - at Data and Signal levels - proposed in the literature against the attacks introduced in Chapter 2.

The Galileo OS specifications and its peculiarities with respect to GPS, useful to the thesis work are highlighted within **Chapter 4**. **Chapter 5** presents the different approach of this thesis with respect to methods in Chapter 3. The notions of Unconditional Authentication and Integrity Protection will be given, and an

Information Theory analysis will return the corresponding lower bound on the required key entropy (i.e. length) at both Data and Signal layers. Everything, will be matched to a suitable channel and signal setting. In **Chapter 6** the specifics adopted to simulate via MATLAB the achievements of Chapter 5 will be declared. Furthermore, our proposals outcomes, respectively at Data and Signal layer, will be discussed and, when it is possible, compared with other existing options. Our idea will also be tested in a more realistic scenery, that is the DLR channel model. Finally, **Chapter 7** will provide a summary of the thesis important achievements, and outlines which is going to be the future work.

Chapter 2

Navigation and interfering signals

2.1 Basics of GNSS

In this section some GNSS basic concepts are outlined, that are needed to understand a possible system vulnerability, and how an attacker action can take place. In the following, notions will be given regardless of the specific system, but we will refer to GPS, and Galileo in particular.

The fundamental component of such a positioning system is the so-called *Space segment*, formed by a satellite constellation. For instance, both GPS and Galileo use 24 Medium Earth Orbit (MEO) satellites, in the GPS case placed on six different orbital planes, while Galileo will distribute satellites over three orbits. The positioning is done in such a way that the users will have, at least, four satellites simultaneously in view from any point on the Earth surface, and at any time. The Space Vehicle (SV) X_i is essentially a transmitter broadcasting continuously, and in a synchronized manner, its Earth Centered Inertial (ECI) system coordinates (x_i, y_i, z_i), and other information contained into its navigation signal, $s_i(t)$. In addition, to achieve synchronization each satellite is equipped with an highly stable atomic clock, which gives the system current time instant t [2].

Then, there is the so-called *Ground-Control segment.* It is the infrastructure on the ground, organized in different centers, which is concerned with tracking and

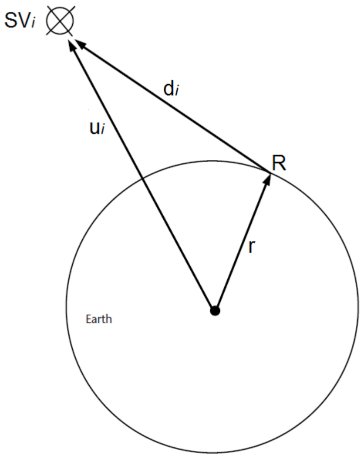


Figure 2.1: Vector representation for satellite-receiver distance and position, inspired by [2].

monitoring satellites, and their mission. Its task is essential to a correctly working system, as a matter of fact the Ground segment monitors satellite health status and signal integrity, maintains the designed orbital configuration, and updates satellite navigation data, and satellite clock corrections as it misaligns from t [2].

Finally we have the generic system user R, equipped with its receiver devices, which constitutes the so-called *User segment.* Referring to Figure (2.1), $\mathbf{r}$ is the position vector representation in the Earth Centered Earth Fixed (ECEF) coordinate system (i.e. latitude and longitude rotate with the earth, and has its origin at the center of the planet) for R, that is as well provided with a crystal clock oscillator [2].

2.1.1 Position and Timing determination

Each navigation signal $s_i(t)$ propagates through space in all directions at the speed of light c, and is captured by R, demodulated and transformed between the Earth Centered Inertial (ECI) to the Earth Centered Earth Fixed (ECEF) coordinates to compute $\mathbf{r}$(t).

The available signal at position R is [9]:

$$g(\mathbf{r},t) = \sum_i A_i s_i\left(t - \frac{|\mathbf{u}_i - \mathbf{r}|}{c}\right) + w(\mathbf{r},t) \tag{2.1}$$

where A_i is the attenuation suffered by the signal across the path X_i-R, and *w(**r**,t)* is the receiver background noise.

Both the considered GNSS, in the signal transmission make use of direct sequence spread spectrum (DSSS) modulation, in order to allow the entire constellation to broadcast simultaneously on the same carrier. Then, the transmitted signal $s_i(t)$ includes its own ranging code - publicly known - and navigation data. The ranging code, or Pseudo Random Noise (PRN) code, is a periodic sequence of ±1, with good auto-correlation properties, and if replicated at the receiver side, it allows to determine the travel time of radio signal from the satellite to R.

Let's see more in detail how the ranging-code becomes useful in position determination. We wish to determine the vector $\mathbf{r}$, having $\mathbf{u}_i$ calculated via the ephemeris data transmitted by SV_i, and the physical signal path

$$\mathbf{d}_i = \mathbf{u}_i - \mathbf{r} \tag{2.2}$$

This distance, $\|\mathbf{d}_i\|$, can be computed by measuring the propagation time required for a ranging code to transit from the satellite antenna, to the user receiver antenna. For instance, the code starting phase generated by the satellite at t_1 arrives at the receiver at t_2, resulting in a propagation time

$$\mathit{\Delta} t = t_2 - t_1 \tag{2.3}$$

Making the assumption that the satellite clock and the receiver clock are perfectly synchronized, at the receiver an identical coded ranging signal is generated at t_1, this replica is shifted in time until it achieves correlation with the received satellite-generated ranging code. The correlation process would yield the optimal estimate of the propagation time, and retrieves the signal level above the noise. By multiplying this propagation time, $\mathit{\Delta} t$, by the speed of light, the true $\|\mathbf{d}_i\|$,

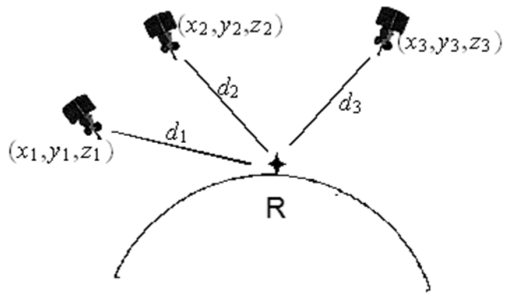


Figure 2.2: Position triangulation means three in-view satellites.

satellite-to-user geometric range, can be computed:

$$\|\mathbf{d}_i\| = \Delta t \cdot c \tag{2.4}$$

However, it is still not enough to compute $\mathbf{r}(t)$, since this unknown position is a vector in $\mathbb{R}^3$. Then we should acquire at least three satellite signals, and for each of them we first compute the corresponding d_i (for convenience of notation, we denote by d_i the vector magnitude) as exposed above, then we find (x_i , y_i, z_i) by demodulating its de-spread navigation data. It corresponds to *triangulating* our position, as depicted in Figure 2.2. We get the following linear system of three equations, and three variables:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 = d_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 = d_3^2 \end{cases} \tag{2.5}$$

It is equivalent to draw a sphere around each satellite, with radius equal to the corresponding d_i, these will intersect in two points, the receiver will be located at the earth's surface closest point.
However, the above described procedure works in the ideal case that transmitter and receiver are perfectly synchronized.

In order to minimize the receiver costs and dimensions, this latter is equipped with a crystal clock, which influenced by environmental conditions drifts relatively to the stable atomic scale. Consequently we can define a receiver clock *offset*:

$$t_{\text{offset}} = t_R - t \tag{2.6}$$

which essentially represents the difference between the receiver time, and the true system time; and a clock *drift* - the time derivative of the offset. Therefore, R has only access to an imprecise estimate t_R of the system time t, and it actually receives the signal [9]:

$$g(\mathbf{r}, t_R) = \sum_i A_i s_i \Big(t - \frac{|\mathbf{u}_i - \mathbf{r}|}{c} + t_{\text{offset}} \Big) + w(\mathbf{r}, t_R) \tag{2.7}$$

Because of the lack of synchronization, he can only get what is called *pseudorange* - the range determined by multiplying the signal propagation velocity, by the time difference between two non-synchronized clocks

$$\begin{aligned} \tilde{d}_i &= \Big(\frac{\|\mathbf{u}_i - \mathbf{r}\|}{c} - t_{\text{offset}} \Big) \cdot c \\ &= \|\mathbf{u}_i - \mathbf{r}\| - t_{\text{offset}} \cdot c \\ &= d_i - t_{\text{offset}} \cdot c \\ &= (t_R - t_{tx}) \cdot c \end{aligned} \tag{2.8}$$

where t_{tx} is the signal transmission time measured by the satellite. Notice that t_{offset} is an additional un-known variable, then with respect to the ideal case we need an additional equation. Therefore, actually at least four in-view satellites are required, and the system (2.5) becomes:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 = (\tilde{d}_1 + c \cdot t_{\text{offset}})^2 \\ (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 = (\tilde{d}_2 + c \cdot t_{\text{offset}})^2 \\ (x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 = (\tilde{d}_3 + c \cdot t_{\text{offset}})^2 \\ (x_4 - x)^2 + (y_4 - y)^2 + (z_4 - z)^2 = (\tilde{d}_4 + c \cdot t_{\text{offset}})^2 \end{cases} \tag{2.9}$$

Practically, the t_{offset} introduction is unavoidable since even a small clock error will affect severely the positioning. For instance, taking $t_{\text{offset}} = 1$ ms, and multiplying it by c, the resulting position error is 300 km. Therefore, pseudorange measurement must exploit at least four Space Vehicle (SV)s, and the corresponding t_{tx} correctly obtained from navigation, thus one can achieve the exact $\mathbf{r}(\mathrm{t})$ and t, even without requiring a precise local clock.

Then, the GNSS systems are not only a useful tool in position determination, but can also disseminate time synchronization (with respect to Universal Time Coordinates (UTC)) between different users worldwide. Furthermore, without going into details, by exploiting Doppler effect on the received signal frequency with respect to the nominal one, the user velocity can also be computed [2]. This explains why it is usual to speak of PNT systems.

2.1.2 Satellite signal

Without reference to a specific system, the basic signal structure is described in the following. In general, $s_i(t)$ carries [2]:

1. **Navigation-Data** $D_i(t)$: The waveform $D_i(t)$ is a base-band binary signal which gives us the necessary information to compute a PNT solution. Each Space Vehicle $(\mathrm{SV})_i$ via $D_i(t)$ transmits users a timing reference about the transmission instant, accurate orbital parameters (according to different services) about its own position in space - *ephemeris*, and a looser information on the position of all the constellation satellites - *almanac*.

2. **Spreading-sequence** $c_i(t)$: The signal transmission occurs simultaneously from each satellite, then the problem of medium access arises. A TDMA multiplexing scheme is not suitable because of the simultaneity; while an FDMA requires a different carrier frequency for each satellite, and hence an extensive use of spectrum, and expensive multi-frequency receivers. For such reasons it is only adopted by GLONASS.
   Other GNSS systems use Code Division Multiple Access (CDMA) for two reasons. First it enhances the use of spectrum, in fact distributing satellites into orthogonal channels, it allows them to share the same frequency. And

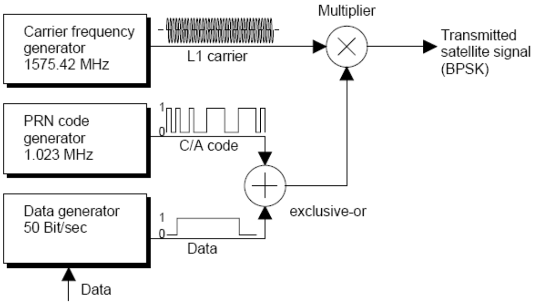

Carrier frequency generator 1575.42 MHz
L1 carrier
Multiplier
Transmitted satellite signal (BPSK)
PRN code generator 1.023 MHz
1
0
C/A code
Data generator 50 Bit/sec
1
0
Data
exclusive-or
Data

Figure 2.3: The GPS L1 C/A civilian signal generation scheme.

secondly these Pseudo Random Noise (PRN) codes are an aid in pseudorange computation, as described in 2.1.1. As a consequence, $D_i(t)$ multiplies the corresponding high-rate, and periodic ranging code waveform, $c_i(t)$; then the resulting signal will occupy a wider band, over which its power will be spread, namely it becomes a wideband transmission.

3. **Carrier frequency** f_0: Finally, the spread-spectrum signal modulates the system carrier frequency, using a BPSK, or BOC scheme depending on the specific positioning system. To these systems the so-called Radio Navigation Satellite Services (RNSS), and recently also the Aeronautical Radio Navigation Service (ARNS), portions in the L-band are reserved [1]. Furthermore, the modern navigation systems do not use a single carrier, rather they roughly have one carrier for each offered service, and therefore these are selected in order to guarantee inter-operability.

For clarity Figure 2.3 shows how these three components are joined together to generate the GPS L1 C/A civilian signal $s_i(t)$. The details of signal generation are neglected here, however it is worth noting that the power levels of these signals are different from those of the common terrestrial mobile communications. In fact, the satellite signal on earth reaches a minimum value of $\simeq$-160 dBW, depending on the user elevation angle. The reasons of this signal weakness are the high path-loss built-up travelling over at least 23000 km. Then, $s_i(\boldsymbol{r}, t_R)$ is so weak that it is highly vulnerable to any interference and obstacles. After this brief introduction to GNSS fundamentals, in the following section we will see how a malicious entity can mount a successful attack.

2.2 GNSS impairments

When the GNSS signal reaches the ground it is strongly attenuated, so it becomes vulnerable to even small impairment that degrade the PNT solution computation.
A first class of impairment is the radio frequency (RF) interference, this can be either narrowband or wideband, and can be generated by any undesired source. In our daily life there are a large number of systems [2] that work with RF signal transmission within the GNSS band. It is inevitable that some out-of-band energy - due to harmonics, intermodulation products, or a malfunctioning equipment in adjacent bands - will fall within the range of frequencies processed by GNSS receivers. The Code Division Multiple Access (CDMA) mitigates narrow band interference, but it is not enough.
Another type of impairment are multipath reflections and shadowing, the former gives replicas of the desired signal which, depending on their strength, can introduce error in pseudorange measurement; while the second is an excess attenuation of the direct path. Multipath signal are mostly generated in urban environments, then they can be mitigated means either with a well-studied antenna location in case of fixed-user, or intentionally designed antennas which reject signals arriving from below the horizon [4]. For sure it is an annoying impairment during the signal acquisition phase, since correlating the receiver replica with a received signal made of multiple components, corresponds to multiple correlation peaks that gives uncertainty about the genuine signal propagation time.
However, these are unintentional interference examples, while interfering signals intentionally created by someone worry us much more. There are many kinds of attack, and each one is creating a signal that interferes with the authentic one; against those we must find a solution, so it is worth to see them in detail.

2.2.1 Attacks on GNSS signal

Because of GNSS requirements, we can define as an attack any action which affect one of these security properties: integrity, availability and authenticity [2]. The attacks that we are going to describe can be applied to any satellite positioning system. Before going on, it has to be highlighted that in all these systems the positioning signal is broadcast by satellites, and users on earth can only receive it, that is, they cannot communicate or reply.

Jamming The RF interference may also be intentional, in this case it is called *jamming.* It is the most trivial attack, since it does not attempt to locate the user at a desired wrong position, but simply aims to prevent the position computation, that is a Denial-Of-Service (DOS) attack, or in other words it affects the system availability. Basically, the attacker can interfere with a continuous wave (CW), a pulsed continuous wave (PCW) or white Gaussian noise (AWGN). In any case the victim is not able to detect or recognise the satellite signal. This attack should not be underestimated, because small jammers are available at a low price and can deny GNSS within an area of tens of metres [4].

Forging and modification attacks

In other cases, the aim of the intentional interferer is not just to disrupt GNSS operation, but rather to manipulate the system such that it produce a false position **r'** of the victim receiver, instead of its actual position **r** and/or a falsified timing solution, $t^{'} \neq t$ [4]. Then, the attacker need access either to the victim receiver (or rather to its output), or to the channel over which the antenna position is reported [9]. The practical implementation of the first action corresponds to replacing the receiver hardware with a device that keeps providing false PNT solutions (e.g. on the display), and can be controlled via a radio link. In order to provide protection against this kind of attack, the so-called tamper-resistant receiver are built, whose output is verified by a cryptographic authentication protocol. The tampering attack should not be underestimated, since it will be very dangerous if the receiver is owner of secret material (e.g. the key of the GPS military ranging code).
The second option is to attack the receiver antenna, which provide the electromagnetic receiver input. The hacker can, for instance, move the receiver antenna, or place it within a shielded area, along with the use of its own transmitting antenna; in this manner he can bypass a tamper resistant receiver [9]. This is achievable by several attack strategies that are listed here below.

Spoofing The *spoofing* attack is defined as the broadcast of false GNSS wideband signals [4]. In this case the attacker imitates the GNSS signal and make the user believe to be at a different position, thus it is also called Signal-Synthesis attack. The signal synthesis is made possible by the fact that the message format and modulation are completely known to everyone. Then, the receiver antenna get the false waveform g(**r'**,t), with modified navigation parameters: e.g. the ephemeris, the clock correction parameters, the Space Vehicle (SV) health indicator, or the GNSS time. Depending on the particular changing parameters, the attack can have different impact on PNT solution: the equation system (2.9) may not be solvable, thus causing a denial of service to the user; or the data is modified in a smarter way, inducing a desired error in the computed PNT.

Furthermore, if the spoofer is not synchronized with the system, the attack may be detected since the receiver loses track of the GNSS signal, while synchronized generators do not introduce jumps in time [4].

Certainly this attack is potentially more damaging than jamming since the receiver is providing PNT solutions with fairly good signal quality although the position is false. A possible countermeasure against it is to introduce some randomness into the navigation message, integrity the legitimate receiver is going to check with the key. However, in GNSS data integrity is not sufficient to guarantee signal integrity.

Meaconing In GNSS, as we have mentioned above, the timing information plays an important role. GPS, and in the same manner Galileo, receivers compute their position from the TOA of the navigation signals. Basically, the receiver uses the time, t_R, it took the navigation signal to reach the user equipment, to compute the distance between the transmitter and the receiver. As the receiver has a wrong knowledge of t_R, it is corrected by the fourth equation in (2.9).

Hence, an attacker can desire to attack this time dependence introducing a delay into the TOA. In this sense, *meaconing* would be the simplest attack: all signals are acquired by the attacker and later replayed to the receiver, delaying each one by the same amount of time. Therefore, the received signal becomes [14]:

$$g(\mathbf{r}, t_R) = \sum_i A_i s_i\Big(t_R - \frac{|\mathbf{u}_i - \mathbf{r}|}{c}\Big) + w(\mathbf{r}, t_R) + \alpha \sum_i A_i s_i\Big(t_R - \frac{|\mathbf{u}_i - \mathbf{r}'|}{c} - d\Big) + \eta(\mathbf{r}\text{'}, t_R - d) \tag{2.10}$$

where $d>0$ is the delay chosen by the attacker, α the replayed signal amplitude advantage factor, $\mathbf{r}^{'}$ the attacker location, and $\eta(\mathbf{r}^{'}, t_R - d)$ is the attacker receiver noise possibly added to the delayed signal.

This attack will only result in a different clock offset, that is a time jump; while the position will be the same. Hence, this simple delay attack is an efficient way to attack time synchronization (e.g. in time stamping financial operations) but has no direct impact on positioning. It can only impact positioning if the user is moving fast and the delay d is very long [4].

Selective-delay attack In comparison to meaconing, in a selective-delay attack each signal is delayed by a different amount of time. Let's suppose that the attacker is at position P_A, the legitimate receiver is at position P_R, and the attacker wants to make the receiver believe to be at the false position P_F. First, the attacker receives four signals at the corresponding instants: signals S_1 at time t_1, S_2 at time t_2, S_3 at time t_3 and S_4 at t_4 [5]. Now the attacker calculates for each signal S_i the corresponding instant $t_i^{'}$ at which the victim will receive the same, if he truly would be at P_F, and the propagation delay $t^{''}$ along the distance from P_A to P_R. Then, the attacker retransmits each signal S_i with a delay $t_i^{'} - t_i - t_i^{''}$. To be noticed that this is possible only if $t_i^{'} - t_i - t_i^{''}$ is positive, in other words the attacker can only perform a delay, but not an anticipation.
A time jump may be detectable by the receiver, but the attacker can hide it in different manners [5]. He can for example jam the receiver until he achieves the desired offset, or in a smarter way the attacker can slowly introduce an even higher delay in navigation signal, in order to avoid a fast time offset increase. Once the receiver has reached the needed time offset, the attacker starts providing the false position.

Relaying attack This attack is based on the idea of making the victim believe to be at position P_A (attacker position), when he actually is at P_R. This becomes possible, for example, by connecting the legitimate receiver to the attacker's antenna. This implies that the hacker has a physical access to the receiver equipment. Alternatively, if the distance between the two is too large, the attacker will transmit the data received at P_A to P_R through another channel. The relaying attack is also called *wormhole* attack. However, also in this case the attacker has to pay attention to not introduce high time difference. And in turn, the receiver can counteract the attack making it hard the message relay, for instance, using a high-bandwidth signal, to maximize the cost of forwarding it [9].

Early bit detection attack As we will see in Chapter 3, the cryptographic message authentication is based on appending to it an authenticator, that is some unpredictable data. Therefore, it prevents the possibility of forging the authenticated navigation data before its reception. This means, that the receiver should before correctly acquire and track the satellite signal, remove the code and read the bit content in order to know the authentication chunk. However, the CDMA might give the attacker another possibility. In fact, as we know that the PRN code spreads the bit over long sequences, and the attacker can attempt to correlate a shorter code portion to detect the authenticator bits. Once the unpredictable portion is known as well, the attacker can replay the data toward his victim, before it will be received authentically, that is with a negative delay [13]. The corresponding probability of success depends on the carrier-to-noise (C/N_0) ratio.

Secure Code Estimation and Replay (SCER) attack Since the (GPS) L1 Coarse/Acquisition (C/A), or Galileo OS signals are publicly known, and the navigation data is predictable, some more robust technique at the code level are based on the idea of introducing some kind of randomness within the ranging sequence [13]. That is, an unpredictable segment w, is inserted into the message or PRN code, thus they are readable only by legitimate receivers that know the key, i.e. something similar to the GPS military code, or Galileo Public Regulated Service (PRS) service. Here the signal decomposition into individual contributions, g_i, becomes difficult without any knowledge of the key, and consequently the Selective-delay attack is impossible.

However such schemes that use cryptographic protection are vulnerable to Secure Code Estimation and Replay (SCER) attack: by an high gain antenna the attacker keeps observing each secret symbol of the received signal in the corresponding symbol interval. Thus, the security-code is estimated into $\hat{w}$, and immediately used to simulate the authentic satellite signal. Then, the signal received at the legitimate receiver position is an overlap of the two:

$$g_i(\mathbf{r}, t_R) = \alpha \hat{w}_i s_i\left(t_R - \frac{|\mathbf{u}_i - \mathbf{r}|}{c} - d\right) + w_i s_i\left(t_R - \frac{|\mathbf{u}_i - \mathbf{r}|}{c}\right) + w(\mathbf{r}, t_R) \qquad (2.11)$$

where d is the sum of processing/transmission delay. In this case the attacker is able to demodulate the data, and to remove its receiver noise. In other words, SCER is the *early bit detection* equivalent at signal layer.

Obviously, the better is the estimate $\hat{w}_i$, greater is the likelihood of success for the SCER attack. In general, the attacker acts at the physical layer, so that its attack is independent of the adopted cryptographic scheme. Rather, it only depend on the chosen instantaneous estimator performance.

In [15] three estimator function performance are analysed. Namely, the maximum likelihood (ML), the maximum a posteriori (MAP), and the minimum-mean-square-error (MMSE) estimators. However, in [16] a more optimal attacker estimator has been derived, jointly with a stronger detection technique.

Chapter 3

A review of possible defenses in GNSS

Before presenting the current state-of-art about GNSSs signal authentication it should be noted that:

- In this context cryptography is not always the appropriate tool, or rather it cannot works alone. As a matter of fact, the navigation data can be protected by secrecy (e.g. with a secret ranging code), or authenticated by un-predictable data (e.g adding a message authentication code or a digital signature); however, GNSS carries also timing information, that is the navigation signal reception time. The traditional cryptography may rend the navigation signal unusable, but is not able to authenticate it, or rather at the current time it has not been achieved.

- For the above reason, we will also be interested in physical-layer security, which exploits the communication medium and does not rely on a higher layer encryption.

- Usually, signal authentication can be aided by a sender-receiver interaction. However because of the huge number of potential users, and large distances this is not possible for satellite navigation systems; then we have to work with a broadcast scheme.

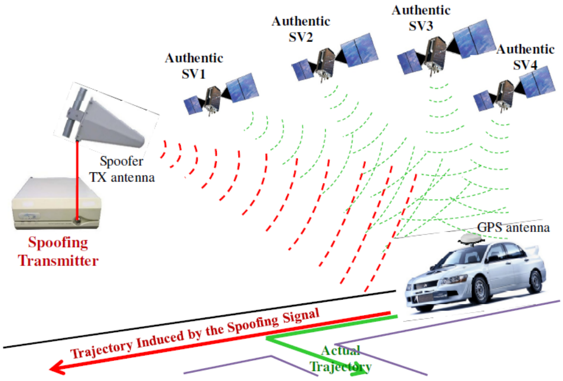


Figure 3.1: A general single-antenna spoofing attack scenario.

Furthermore, the GNSS signal defense scenario is so wide that it can be organized from several points of view. In fact, the spoofing countermeasures can be performed during different processes (e.g. pre-despreading, acquisition, tracking or position solution defining), and these can be classified in several manners, for instance it is common to speak about spoofing *detection* or spoofing *mitigation* [4]. However herein, because of our information security approach, we will define the following two macro-categories: *cryptographic* and *non-cryptographic* mechanisms, which are further divided into other sub-categories covering three operational layers, namely the signal processing, data bit and position/navigation solution levels.

From here on, with the term spoofer we will refer to a generic attacker, who aims to forge, modify or relay an authentic message. To clarify the scenario where the spoofer should take part, we may refer to Figure 3.1.

3.1 Non-Cryptographic Techniques

In the following we will see a series of checks employed to detect potential attacks and to verify the received signal consistency [4].
The structure of GNSS OS signals, including the modulation type, PRN sequences, transmit frequency, signal bandwidth, Doppler range, signal strength and many other features are publicly known. Thus, an attacker is motivated to mimic these authentic GNSS signal features; despite sophisticated hacking tools, spoofing

signals are different from the authentic ones. Therefore the following countermeasure methods, sensitive to certain GNSS signal statistics, and looking for specific features that can reveal the spoofer, can defend the receiver [14]. These techniques cannot replace cryptography, but they can give it a relevant aid, since they have a more immediate implementation.

3.1.1 PNT solution layer

TOA monitoring the effect of an un-synchronized, or naive attack may be detectable via a suddenly higher clock offset [4].

Position Jump As a consequence of a poorly-designed spoofing attack, there may also be a detectable jump of kilometres in a few seconds, namely a position-jump [4].

Receiver Autonomous Integrity Monitoring (RAIM) is an algorithm, implemented at the receiver, that makes use of more than four pseudoranges to check the PNT solution consistency [4].

Inertial Measurement Unit (IMU) check If the receiver is equipped with accelerometers and gyroscopes it is able to obtain a second measurement of the current position, a discrepancy between the two is index of a spoofing attack [4].

Consistency check with other solutions The user may have access to other sensors or position references, such as cellular network base stations, or in view Wi-Fi access points [4].

3.1.2 Data layer

As described in (2.1.2), each D_i bit sequence carries information about the satellite positions (i.e. *ephemeris* and *almanac*), and the system clock, therefore each signal has to be consistent with the others. Furthermore, the receiver may access an on-line *ephemeris* database, with which the received signal has to be consistent.

3.1.3 Signal processing layer

Received Signal Strength (RSS) monitoring In order to be effective, the spoofer should generate a correlation peak higher than the authentic signal peak to mislead the target receiver, but this may result into sharp jumps in the signal strength. Then, rapid SNR changes should alert the receiver [4].

RSS monitoring with moving receiver The received signal power monitoring becomes even more effective with moving receivers. Given the relatively short between spoofer and receiver, any movement between the two leads to detectable changes in the received power [4].

Different frequencies power level As a design choice, GNSSs assign predefined different power levels to their different frequencies. However, a common spoofer works at a single frequency, then the attack is going to increase the power level of that frequency, leaving the rest unaltered [4].

Frequency-phase consistency check In authentic signals the Doppler frequency, and code delay should be consistent because of the satellite-receiver relative motion, but this property may be neglected by less sophisticated spoofers [4].

Spoofing detection via antenna pattern diversity According to Figure 3.1, the spoofer is a terrestrial single-antenna transmitter, while the satellites' signals are coming from hundreds of kilometres, and crossing different paths. This diversity in the propagation model can be detected using antennas with complementary reception patterns [4].

Angle-Of-Arrival check The single antenna attacker may be identified via antenna-array, which perform a spatial processing. Since antenna-array are an expensive hardware, the same check can be performed moving a single antenna along a random trajectory, known as a *synthetic* antenna-array [4].

3.1.4 Cooperative Methods

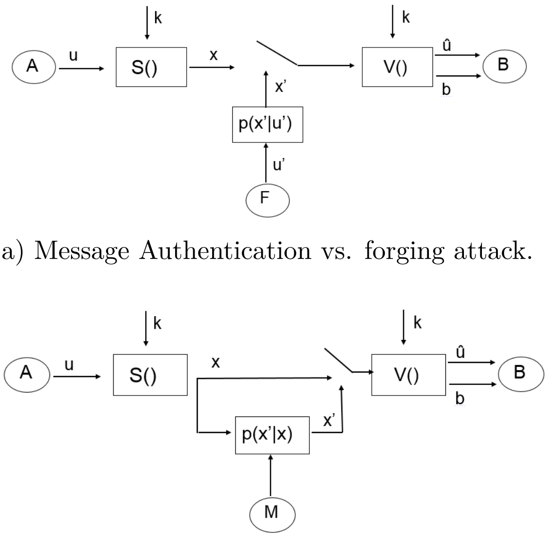
Multiple Fixed Receiver A different spoofing detection approach is based on the interaction of multiple fixed receivers, located at known relative positions. If a single-antenna spoofer is attacking the receivers, they will all compute similar positions, that obviously cannot be if the signal is authentic. The drawback is that it will require an embedded communication equipment, and an authentication protocol between different parties, since we need to communicate with trusted nodes [4]. In conclusion, a receiver cannot implement all these techniques, nor the check success is even guaranteed with certainty, since it always depends on how smart and powerful the attacker is [5]. However, a cross check made by a pair of them will for sure detect potential attacks, that have managed to bypass the cryptographic authentication.

3.2 Cryptographic Techniques

We move on seeing the defense mechanisms that involve the use of a secret cryptographic key. Among cryptography goals there is not only message confidentiality and encryption - the science of *transforming* an original plaintext message into a coded, and apparently no-sense message, via a suitable key material - but also information security. The security regards mainly with message (or source) *authentication* and *integrity protection*, that are the aim of this thesis work. Here follows an introduction to cryptographic concepts and primitives that will be used throughout this thesis.

Often authentication and integrity protection are seen as coincident, or the second as a service of the former; but we wish to emphasize the slight conceptual difference between them.

Message Authentication It allows the sender, A, to transmit a message to its receiver, B, in such a way that B is sure about the message source. The attacker, F, is anyone who attempts to *forge* a message for B, pretending to be A.



(a) Message Authentication vs. forging attack.

(b) Message Integrity Protection vs modification attack.

Figure 3.2: The cryptographic settings.

Integrity Protection It ensures that each message is delivered to B in the same conditions that it was sent out from A — with no bits inserted, removed, or modified. The attacker, M, is anyone who attempts to *modify* the message for B, pretending that the modified message is the original coming from A.

The difference is clear if we suppose that a navigation signal sent by SV$_i$ undergoes a meaconing attack before reaching the legitimate receiver, in this case it will maintain the data integrity (no bits are changed), but it is not physically authentic (i.e. it is not coming directly from SV$_i$).

It should be noted that when we speak about authentication it will always refer to the message, or to its source, but we are not going to treat user authentication, which aims to prevent unauthorized entities from using the service [11]. For sure, we would like to deny the service to malicious entities, but such an authentication service is practically impossible to be implemented for the amount of Open Service users. Therefore, the alternative is to leave the service open, and available to anyone, but giving the receiver the certainty about the signal source.

Making reference to the block schemes depicted in Figure 3.2 we are going to see how it is possible to achieve cryptographic message authentication and integrity protection. Let suppose that the legitimate receiver, A, needs to send a message u to B in an authentic way:

1. **Authentication and Integrity Protection**: u and the key, k, are fed in block $\mathbb{S}()$, usually a random function chosen within a publicly defined family, and identified by the random key k. Thus, $\mathbb{S}()$ will returns a new sequence x. The construction of x depends on the particular scheme, but generally applies $x = \{u, t\}$, that is, a uniquely defined authenticator t is appended to u. The role of t is ensuring that the message is coming from a legitimate entity, and none without the key can imitate it, since it is unpredictable before its reception. $\mathbb{S}()$ is the same for both authentication, and integrity protection purposes.

2. **Forging or Modification attack**: x is sent over the insecure channel, indeed before reaching B's side it may run into an attack. Making reference to Figure 3.2, there are two potential types of attacker, the first is F (a) who is interested in *forging* a new message u' pretending to be A even though having no key. The forger attempts to authenticate his u' into $x' = \{u', t'\}$ before having observed how x is constructed, that is an ignorant guess. The second one is M (b), who instead is interested in intercepting, observing and modifying x into x', that is, he pretends to have modified the message, and its authenticator in an authentic way, without having the key.

3. **Verification**: at the other side of the channel there is B, who is the verifier and receiver. Whenever he receives a message, either x or x', this will be processed by the block $\mathbb{V}(\cdot$,k), a deterministic function which basically reverses block $\mathbb{S}(\cdot$,k) operation. If the message is coming from A and was not modified, then the authenticator $\tilde{t}$ computed on the received $\hat{u}$, is equal to $\hat{t}$ coming from the channel. In this latter case the verification output will be $\hat{u} = u$, and the flag $b = OK$.

Therefore, both Message Authentication and Integrity Protection services can be jointly provided by the same mechanism, $\Big(\mathbb{S}(\cdot), \mathbb{V}(\cdot)\Big)$.

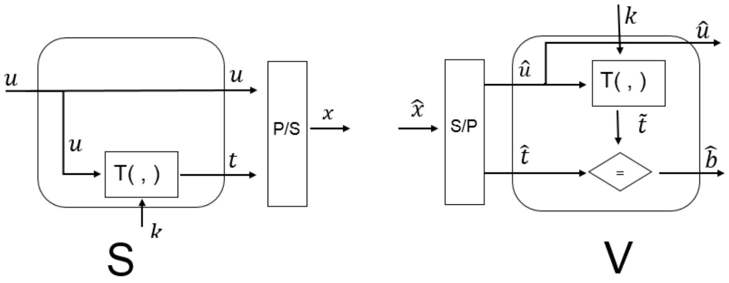


Figure 3.3: S() and V() implementation for MAC computation.

Several authentication schemes exist, which depends not only on the algorithms $\mathbb{S}()$ and $\mathbb{V}()$ implementation, but they can also be based on the key role, thus we can speak about symmetric or asymmetric schemes.

Symmetric-key schemes

In symmetric key authentication techniques A and B share a common secret key k, by which they authenticate and verify the message, respectively. The types of function that may be used to produce an authenticator in $\mathbb{S}()$, and to verify it in $\mathbb{V}()$, can be grouped in three categories [10]:

- **Hash function**: it is a function that maps a message of arbitrary length into a fixed length hash value, which serves as authenticator. Furthermore, this function is easily directly computed but hardly inverted, that is a one-way function. Usually the key role is to identify the specific transformation.

- **Message encryption**: here the cyphertext of u serves as its authenticator.

- **Message Authentication Code (MAC)**: it is a function of u and k, that produce a fixed-length value as authenticator, also called *tag*. In Figure 3.3 blocks $\mathbb{S}()$ and $\mathbb{V}()$ are specified for this case.

Now going back to our problem, a symmetric key scheme is not suitable for the OS service, since it would imply that anyone could create an authentic navigation signal, as if he/she were the generic satellite SV_i. The GNSS OS potential users are the entire world, while a restricted service (e.g. the very precise military GPS

signal, or PRS in Galileo) has a small set of users, and in this case the deployment of a symmetric key, kept secret inside few very expensive and tamper-resistant receivers, is a viable solution.
In order to avoid attacks, it might be an idea to assign each single receiver on the ground a symmetric key shared only with the satellite, but it would imply that the satellite needs enough memory to keep all the keys assigned on the ground, and whenever it broadcast the navigation signal, this will be a multiplex of n signals addressed to the different n users, although these may not be switched on. Obviously it would be a huge waste of memory and bandwidth, then it might be a solution to divide users into groups, and assign to each a different key. In this case, we should have trust among all users within each group, which is practically impossible.
Therefore, despite the fact that the symmetric key schemes are preferable in terms of key length and computational complexity [10], they are too vulnerable to be implemented in the authentication of GNSS OS signal.

Asymmetric-key schemes

Public-key (or asymmetric-key) cryptography, in contrast to the symmetric one, involves the use of two separate keys:

- Public key: all participants have access to it. It is used to *encrypt* for secrecy purpose, or to *verify* in authentication and integrity protection.

- Private key: it is generated locally by each participant and need never to be distributed. It is applied to *decrypt* the message for secrecy, or to *sign* the message for authentication and integrity protection.

The transformation performed by $\mathbb{S}()$ or $\mathbb{V}()$ depends on the couple of key that is provided in input, it means that the two keys are distinct but related means a particular function. Whenever the transmitter refreshes its private key $k \in \mathcal{K}$, it will also compute the corresponding public key

$$k^{'} \in \mathcal{K}' : k^{'} = f(k) \tag{3.1}$$

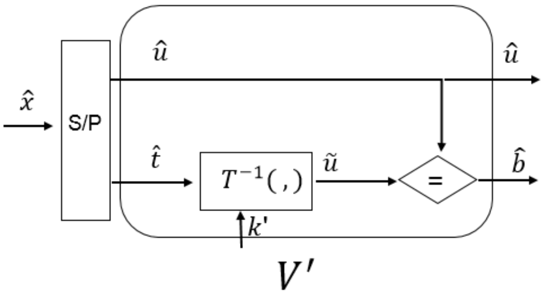


Figure 3.4: The public key verification block.

where f is **one-way** function, such that the recovery of k by knowledge of *k'* is computational infeasible before its validity expires.
As regards the setting, the idea remains basically unchanged at the authentication side: where A is the only who know k by which he computes the *digital signature*

$$s = \mathbb{T}(u, k) \tag{3.2}$$

and analogously to the symmetric case, it will be appended to u. However, for the verification we need a re-parametrization of $\mathbb{V}$:

$$\mathbb{V}(m, k) \equiv \mathbb{V}^{'}(m, k^{'}) \tag{3.3}$$

Furthermore, since anyone can verify with $k^{'}$ it also imply that anyone can forge a valid digital signature s. In other words, the signing algorithm must not be available, but only its inverse function has to be:

$$\mathbb{T}'(\cdot, k') \equiv \Big[\mathbb{T}(\cdot, k)\Big]^{-1} \tag{3.4}$$

$$\mathbb{T}' : \mathcal{T} \times \mathcal{K}' \to \mathcal{U} \tag{3.5}$$

its correspondent block scheme is depicted in Figure 3.4.

Practically, public-key cryptography has been designed to manage communications between many users, where symmetric-key storage becomes a memory overload. Then it appears a suitable solution also for GNSS. In the following the state-of-art proposals will be introduced, subdivided into distinct operational layers.

3.2.1 Asymmetric-key defenses at data-layer

Navigation Message Authentication (NMA) The navigation data authentication and integrity protection service via digital signature is called Navigation Message Authentication (NMA) [12]. Basically, by making use of a well-studied algorithm (e.g. RSA), and assigning each satellite a pair of keys (k_i,$k_i^{'}$), the data $u = D_i$ is digitally signed and accompanied by its Digital Signature (DS), $s = DS_i$. [1] Precisely, D_i makes reference to the most important part of the message, as ephemeris, time of week (TOW) and week number (WN). NMA for sure prevents a spoofing attack at data layer, in fact although the navigation data is predictable, the attacker is not able to forge a new message, accompanied by the corresponding DS, and bypass the verification algorithm. However, a NMA scheme has to satisfy some important requirement, listed here below:

1. the DS$_i$ overhead: the GNSS channel has a very low bandwidth, for instance Galileo useful data will be transmitted at 114 bit/s. Then, the DS cannot overhead excessively, otherwise the waiting for its reception will slow down the message authentication, and utilization. A longer DS increases the minimum time needed to detect an attack, namely the Time-To-Alarm (TTA). And even if the verifier output is positive this long elapsed time can make the verification useless for some applications, for instance during an aircraft landing phase;

2. the efficiency of $\mathbb{V}()$: the verification algorithm has to be implemented into mass-market receivers, then it should be computationally efficient, and should not require a large amount of power;

3. the fluency of $\mathbb{V}()$: as soon as the signal is received, and processed, the DS verification step should be very easy and fast, since if an error occurs it must be given the warning that an attack is ongoing, and avoid the use of a false PNT solution over a long time;

[1] Notice that the previous generic notation u and s has been modified into D_i, and DS_i, in agreement with the GNSS one.

4. the loss-tolerance: a quite high data-loss is a GNSS channel feature that cannot be mitigated much. Therefore, if the receiver is moving and some obstacles prevent data and signature reception, it has to wait for a successive non shadowed broadcast, by this reason bits allocation and frequency within the message must be well-studied.

The European Union Agency for Network and Information Security (ENISA) has compiled a list of recommended DS algorithms and corresponding key sizes and properties [7]. The resulting preferred scheme in terms of level of security and the small key size is the Elliptic Curve (EC) version of the Schnorr signature, whose description is omitted here. We are only interested in the fact that although this algorithm passes a series of cryptanalysis tests, its key size is still large when compared to a symmetric key Message Authentication Code (MAC) schemes for the same level of security, and high bit-rate is required.
Given the issues risen by the use of DS, in the following we present other techniques at data layer that attempt to mitigate the problem.

TESLA: a broadcast authentication scheme In [18] an authentication protocol is proposed for wireless radio broadcast towards a multitude of users, called Timed Efficient Stream Loss-tolerant Authentication (TESLA). The basic idea is to use a hybrid protocol: it makes use of a private key to compute Message Authentication Codes (MACs), but after some time this key is shared with the receivers, thus it becomes a symmetric (group) key authentication protocol.
Practically, at transmission time the key k is known to the sender only, so he is the only entity who can compute the correct message MAC. The receiver, who is not yet able to verify the packet authenticity, buffers all the received packets, and only a short while later, the sender discloses k and the receiver becomes able to authenticate the packet. Obviously, the receiver/verifier has to be sure about the authenticity of k, and TESLA manages this issue via a self authenticating key-chain. To generate a key-chain the sender first picks a random key k_l, to which he repeatedly applies, for l times, a certain one-way function F, until he gets k_0, which will be the so-called *root key*. This will be the first key to be used (Figure 3.5), and a digital signature will authenticate it. After its expiration time, k_0 is revealed together with its signature, and the transmitter will start using k_1, and so

on up to k_l. While, when k_i is disclosed, the receiver can verify if the received key is genuine by applying i times the one-way function to the received key and check if the result is equal to the root key, $F^i(k_i) = k_0$. If F is well-designed (i.e. it is hard to invert), it is unlikely that an attacker can discover future keys.

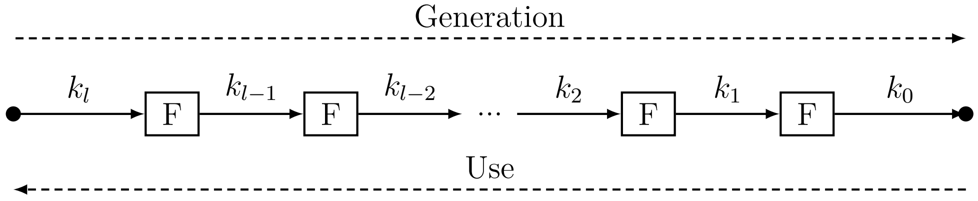


Figure 3.5: The self-authenticating key-chain based on one-way functions.

Therefore, once the key-chain has been generated the broadcast begins:

1. The sender needs to transmit the message M_i, then he uses k_i from the chain and computes $t_i = MAC(M_i; k_i)$, and sends a packet $P_i = [M_i, t_i, k_{i-d}]$, where *d>0* is the established disclosure delay.

2. The receiver captures the packet $P_i^{'} = [M_i^{'}, t_i^{'}, k_{i-d}^{'}]$, and buffers it while waiting for the disclosure of k_i.

3. Once packet $P_{i+d}^{'}$ is received, the key $k_i^{'}$ authenticity is verified by repeating F up to the last verified chain point k_j with $0 \leq j < i$. And if $k_i^{'}$ passes the test, that is $k_j = F^{i-j}(k_i^{'})$ the receiver computes $t_i^{'} = MAC(M_i^{'}; k_i^{'})$, and compared it with the stored t_i.

Therefore, TESLA combines advantages of symmetric and asymmetric protocols, and seems to satisfy all GNSS peculiar requirements, including loss-tolerance. In fact, any level of loss is tolerated without retransmissions since the verification step does not require any chunks of the navigation message. However it has also some critical aspects:

- the disclosure delay has to be well-designed: it should be at least bigger than the longest packet propagation time, since once the key is disclosed everyone can forge an authentic MAC, and a packet received during the key validity period and carrying a correct MAC is accepted as authentic. So the larger the transmission delay, the larger the waiting time to perform verification, and hence TTA. Therefore, we need a trade-off between these two requirements.

- the second issue regards the key chain: its renewal process needs a deep analysis. Since the DS transmission is not costless in terms of bandwidth, it is better to change it rarely. However, if we had transmitted the chain-root a long time ago and the root-key is not still transmitted, a receiver who has just turned-on is not able to authenticate the chain, nor the navigation message; then the service is not available.

Even with the issues discussed above, in the literature TESLA is considered a promising solution for GNSS data authentication. In [19] a possible protocol application to Galileo OS message structure is proposed, and interesting solutions are provided to speed up the verification step, and to cope the lossy channel. Briefly, the author suggest to use a unique key-chain for the entire set of system SVs, and that each one sends out MAC referring to other SVs messages; moreover the authentication is only done on the message part that changes less frequently. On the contrary, the proposal in [20] carries some differences: for instance they use a unique key-chain but with different keys for each SV, and the key length is extended at the cost of a shorter MAC. As regards the chain renewal, the authors propose to use a public-key certificate scheme, however no further details are provided.

Digital Signature Amortization (SigAm) A second solution to mitigate DS transmission overhead can be derived from a proposal [21] oriented to Wireless Sensor Networks (WSN), another example of broadcast communication even more widespread. The proposal is based on the idea of using only one Elliptic Curve Digital Signature Algorithm (ECDSA) signature to authenticate a sequence of broadcast messages; therefore, each broadcast packet contributes to the DS overhead amortization.

Whenever the transmitter needs to broadcast a series of packets in an authentic way he performs the following steps, depicted in Figure 3.6:

1. The transmitter picks a first random *digest* H_{k+1},

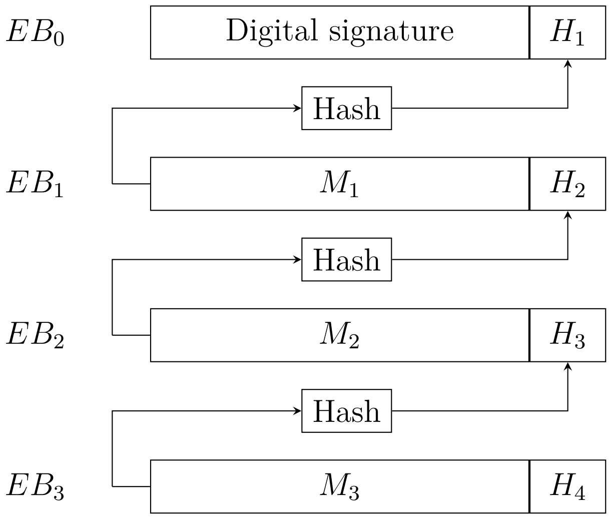


Figure 3.6: Digital signature amortization authentication-chain.

2. H_{k+1} is appended to message M_k, and a hash function is applied on this entire block $EB_k = [\ M_k,\ H_{k+1}]$; the result is the new digest H_k

3. He repeats the operation of the previous point on the new block $EB_{k-1} = [\ M_{k-1},\ H_k]$, and so on until he reaches H_1

4. At this point the sender signs the last digest H_1 via its own private key k, and builds the block: $EB_0 = [\ \mathrm{DS}(H_1,\ k),\ H_1]$

As soon as the chain is constructed the sender starts broadcasting the blocks in the reverse order, that is starting from EB_0. In turn, the receivers verify the authenticity of H_1 with k', if authentic it will give the following messages authentication. Therefore, SigAm makes use of a self-authenticating chain in a similar manner to TESLA, but with the important difference that in this case the user needs not to wait before performing the verification. Then, it can provides a real-time authentication. On the other hand, the transmitter needs to generate and collect all the messages $M_1...M_k$, before he can start transmitting EB_0. This is not a problem with GNSS in general, since the transmitter can compute its messages well in advance.

However, currently none has tested SigAm on GNSS, and a future application of it will require a detailed study of how to move from one chain to the new one, in order to guarantee authentication continuity. Another issue one have to face is the likely data loss, making use of the chain defined above, a similar event if occurs will block the authentication. Then, it apparently seems a good proposal for broadcast authentication, but we cannot say too much about whether it is, or not a viable solution in GNSS.

In conclusion, the data-layer defense techniques, just listed here, can give us navigation message authentication and integrity protection with a security level just higher than the best known computational attack by pre-pending or appending to the message a security chunk like a MAC or DS. Practically, employing one of these techniques we can be sure about the fact that the message content has been received as given by the system.
However, no authentication about the path genuineness is provided, then *meaconing*, and *selective-delay* attacks are still a threat. In addition, even if NMA introduces unpredictable bits within the message, if C/N_0 is sufficiently high for the attacker an *early bit detection* attack can be performed successfully.

3.2.2 Signal-layer defenses

Because of the receiver clock uncertainty, the TOA ranging is an important information that the user should be able to trust. In other words, physical-layer authentication has to provide certainty about the signal point of origin, and relation between the measured time-of-flight and the geometric path from the origin to the user.
Ideally, to ensure signal origin authentication, the signal should contain something which is infeasible for an attacker to generate independently, but easily verifiable by the user [19]. As regards the TOA authentication, a solution should make it difficult for an untrusted entity to observe an re-broadcast the signal without being detected [19]. However, this is complicated by the one-way communication, and unfortunately the attacker can manipulate the signal path in several ways, but at the moment appears that none cryptographic technique can prevent it completely. In the following we are going to see the actual proposals state-of-art.

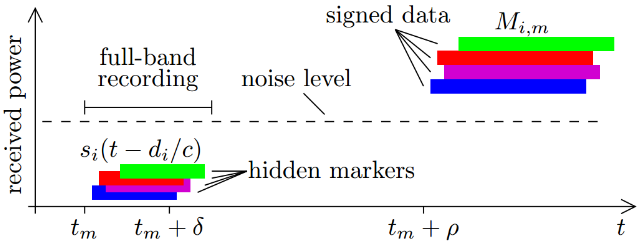


Figure 3.7: Hidden-Markers transmission and their disclosure scheduling.

Spreading Code Encryption (SCE) The Spreading Code Encryption (SCE) is based on the idea of reducing the signal observability from an attacker point of view [19]. Practically, it is achieved using a cryptographic spreading sequence, so that, when the signal is arrived on ground it is covered by the noise, and without knowledge of the spreading sequence the correlation aided detection, and the sequent rebroadcast with a desired delay, are impractical.

This is inspired by the Denial-Of-Service protection applied to the military GPS signal, however it can also be employed to give source authentication. Obviously, when the key validity expires it is disclosed in a way similar to TESLA, and anyone can generate the sequence replica, correlate it with the sampled received signal (that is only possible if it is authentic), and compute the PNT solution. In [19] a SCE implementation for Galileo is proposed, which would give real-time authentication to Commercial Service (CS) users, and delayed authentication to OS users.

However, this technique does not prevent a meaconing (i.e. the application of a delay, regardless which is the individual signal), and a sufficiently motivated adversary equipped with an high gain directional antenna, may performs a SCER attack, and a consequently selective-delay attack.

Hidden Markers In [9] the author proposes a watermarking technique to authenticate the signal: it hides digital information in a carrier signal to prove the identity of the owner. The receiver having some information about the watermark can check its authenticity, and accept it with.

Basically, each transmitter SV$_i$ has a predefined time instant t_m at which it transmits its *hidden marker* as a rectangular pulse of length δ, as depicted in Figure 3.7. These hidden markers are a secret spread sequence, whose power spectral density is

below that of noise by 20 dB. At the time of transmission and reception the marker is secret, and any receiver can only digitize and buffer the entire signal. Therefore, the TOA information carried by the marker is not accessible yet. After a delay ρ (larger than the local clock t_{offset}) the spreading sequence information is publicly revealed in an authentic way, so that all receivers can construct the hidden marker replica and correlate it with the buffered stream. This correlation process reveals the propagation time τ_m, or rather the pseudorange and if it is smaller than ρ it is useful to compute the PNT solution. It may occurs that other secondary peaks are revealed: if it is smaller than the main peak of a certain threshold it may be a multipath effect. While, if this peak delay is greater than ρ it is likely to come from an attacker, that has reproduced the marker after its disclosure.
Therefore, the hidden marker is robust against signal-synthesis attack. However, this apparently smart trick is underestimating a well-equipped attacker who can use a very high directive antenna means which he can recognise the marker and apply to it a selective-delay attack.

3.2.3 Cooperative methods

As already seen for the non-cryptographic methods, there is the possibility of an interaction between the user and a third entity, e.g. a server. Obviously, this is a no stand-alone receiver.
In [22] the authors developed a signal authentication strategy that makes use of the hidden broadcast component. That is, GPS P(Y) or the Galileo Public Regulated Service (PRS) signals are encrypted, and are robust against spoofing. The proposal is to provide the anti-spoofing benefits of secret signals, without having access to the codes themselves. The [22] makes reference to GPS, where the military signal is modulated on the same carrier, L_1, with respect to civil signal, but it is orthogonal to the latter. Basically, the user at location r_1, and the server at r_2 record the GPS signal, and the former send to the reference station a data-set with the corresponding time-stamp. After some pre-processing steps, the server who has access to the secret code correlates the two quadrature-channels; and slides the window until a sufficient great peak is found. This joint processing provides signal authentication and position verification - the peak appearance guarantees

the hidden component presence, which is consider difficult to be imitated. And the comparison between the two signals reveals the user-to-server position, which has to be consistent with the well-known r_2.
A similar work is presented in [23] for Galileo signals. The drawback is that also this technique is not robust against all kinds of attacker; a SCER attack by which the attacker can try to estimate on the flight the secret spreading code and use this estimated version for his replica remains a threat.

In conclusion, we have seen that there is a common awareness about the need to find a solution toward the GNSS Open Service signal authentication, and how it is inevitable to work on several levels. However, although there are many smart and interesting proposals, none is worrying about being unconditionally secure against the capabilities of a generic attacker, thus they may be robust against a specific form of attack, but not enough for another. Formally, they are providing a *computationally* secure authentication and integrity protection, since all NMA techniques are based on a key which is mathematically difficult to be retrieved in time, while the signal layers mechanism works against a poor equipped attacker. That is, they rely on all current computational power of computers and the hardware performance.
Instead this thesis would contribute to the problem identifying a universal bound, which guarantee authentication against any type of attack.

Chapter 4

An introduction to Galileo System

Since the current work is focused on the Galileo OS signal authentication, a brief introduction to the system is needed.
Galileo is a civilian GNSS, developed by the collaboration between the European Union and the European Space Agency (ESA), that will not interfere with GPS but instead it will offer compatibility and interoperability. At the current time GPS provides solution precision which varies widely depending on the location, and there is no guarantee of service continuity. Then, Galileo aims at providing a minimal position computation error of 4 m which corresponds to a 95% horizontal accuracy, along with a 99% service availability [6].
Furthermore, Galileo will allow its users to select among five different services:

1. **Open Service (OS)**: It provides freely positioning and synchronization information to the mass market users;

2. **Safety Of Life (SOL)**: This service is targeted to application strictly concerned with the human life, such as aviation. It has the same OS precision but will automatically inform users of a failure of any satellite or similar problem affecting performance. This service is already implemented by EGNOS, then Galileo will improve its performance by means of OS signals and/or in cooperation with other satellite navigation systems;

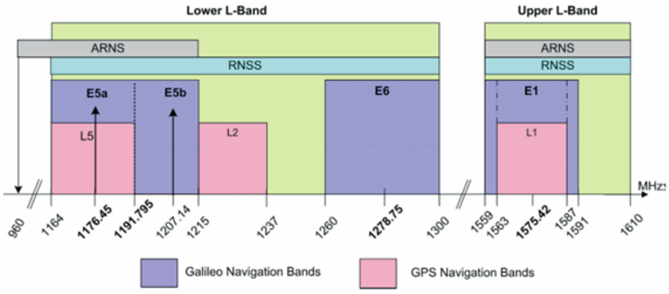


Figure 4.1: Galileo frequency plan [1]

.

Band	Carrier Frequency [MHz]
E1	1575.420
E5a	1176.450
E5b	1207.140
E6	1278.750s

Table 4.1: Galileo carrier frequencies.

3. **Commercial Service (CS)**: It allows to professional or commercial applications an improved performance than that obtained through the open service, its access is limited by cryptography;

4. **Public Regulated Service (PRS)**: Means encryption, this service is restricted to government-authorised users, for sensitive applications which require a high level of service continuity and an higher robustness;

5. **Search and Rescue Service (SAR)**: Galileo's worldwide search and rescue service will help to forward distress signals to a rescue coordination centre by detecting emergency signals transmitted by beacons, and relaying messages to them.

4.0.1 Signal Structure

We are mainly interested in signal structure, since some of its specifications have been used herein and later will be recalled.

The Galileo navigation signals are transmitted in four frequency bands, namely E1,

Service	E1	E5	E6
Open Service	✓	✓	
Safety-Of-Life	✓	✓	
Commercial Service		✓	✓
Public Regulated Service	✓		✓

Table 4.2: Galileo services frequency bands.

E5a, E5b and E6; this band plan is reported in Figure 4.1, where it is compared with the GPS one, while Table 4.1 gives the Galileo distinct signals carriers. The spectrum allocation is designed to guarantee interoperability with GPS, and receivers compatibility. Therefore, it is necessary that the band middle portion is the same, this is the case for E1 and L1 bands. However, to achieve spectral separation Galileo signals are transmitted using digital modulation techniques different from the BPSK of GPS [6].

While as regards the service frequency assignment, they are schematically reported in Table 4.2, but a detailed description goes beyond the interests of this thesis.

The transmitted signals are Right-Hand Circularly Polarized (RHCP), and should be noticed that since each satellite is transmitting on each frequency, CDMA is still adopted as medium access method.

E1 signal

We are going to see more in detail only the OS component carried over band E1.

The E1 signal is made of three channels, called A, B, and C. E1-A is a PRS signal, whose ranging codes and navigation data are encrypted. While the OS signal occupies both channel B and C. Over the former component, B, we have the navigation *data* (multiplied by the spreading sequence), while channel C carries a data-free signal. This is called *pilot* signal, and is a new aspect with respect to GPS, it is made of a ranging code only, not modulated by a navigation data stream, which allows an enhanced signal acquisition and tracking. The E1-B ranging code has a 4092 chips length, with a 1.023 MHz chipping rate giving it a repetition rate of 4 ms; while on the pilot signal a secondary code of length 25 chips is combined

with the primary by $\oplus$ (i.e. XOR). This code extension to 100 ms allows the receiver to solve the signal isolation also in worst situations.
Over channel B the satellite ranging code will be multiplied by the corresponding Data stream - I/NAV - of 250 bit/s (i.e. including navigation data and FEC bits), consequently T_{bit} equals the spreading sequence period. While a GPS symbol length covers 4 sequence periods. The following equations give the mathematical description of these raw signal versions [1]:

$$e_{E1-B}(t) = \sum_{i=-\infty}^{+\infty} \left[c_{E1-B,|i|_{L_{E1-B}}} D_{E1-B,[i]_{DC_{E1-B}}} rect_{T_{c,E1-B}}(t - iT_{c,E1-B}) \right]$$

$$e_{E1-C}(t) = \sum_{i=-\infty}^{+\infty} \left[c_{E1-C,|i|_{L_{E1-C}}} rect_{T_{c,E1-C}}(t - iT_{c,E1-C}) \right]$$

where L stands for the ranging code repetition rate, DC is the number of code chips per symbol, and $[i]_{DC}$ gives the integer part of i/DC. The following step is the carrier modulation in accordance with the BOC scheme, which first requires an introduction.

Signal modulation The interoperability requirement within the same band, and the BPSK signals interference mitigation have risen the modulation issue, by this reason the Binary Offset Carrier (BOC) scheme, has been developed. The BPSK spectrum has a sinc shape, which is mainly positioned around the carrier frequency, then Galileo needs a modulation that move the signal energy further away from the carrier. The generic BOC(f_s, f_c) modulation makes use of a square wave subcarrier

$$s(t)_{BOC} = s(t)sign(sin(2\pi f_s t)) \tag{4.1}$$

where *s(t)* is the square wave resulting after the spreading sequence (of chip rate f_c) application. In the frequency domain, $s(t)_{BOC}$ has a power spectral density approximated as follows:

$$G_{BOC}(f) \simeq G(f - f_s) - G(f + f_s) \tag{4.2}$$

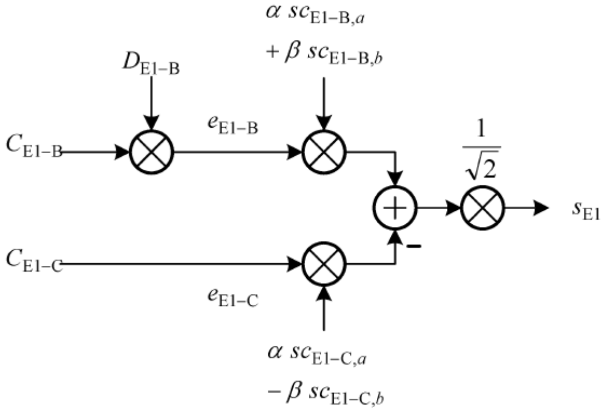


Figure 4.2: E1 Open Service CBOC signal scheme [1]

.

where G(f) is s(t) Fourier transform. This shows that the power is shifted of f_s from the band center. As already mentioned, a generic BOC waveform is denoted via BOC(f_s, f_c), where the former is the sub-carrier frequency, while f_c is the chip rate; or via BOC($f_s/f_c, f_c/f_c$) = BOC(m,n). For instance a BOC(1,1) is similar to a Manchester code, that is, in digital domain, a 1 is encoded as a [+1-1] sequence, and a 0 is encoded as a [-1+1] sequence.

Then, $e_{E1-B}(t)$ and $e_{E1-C}(t)$ are modulated in anti-phase via the sums of two BOC subcarriers as depicted in Figure 4.2, which is actually called CBOC. Where $sc_{E1-Y,a} = BOC(1,1)$ and $sc_{E1-Y,b} = BOC(6,1)$, and are added via a weighted sum that set $\alpha = \sqrt{\frac{10}{11}}$ and $\beta = \sqrt{\frac{1}{11}}$, then the Galileo E1-OS signal power spectral density is equal to the GPS one when data and pilot channels are computed together. Practically, we obtain the following signal:

$$\begin{aligned} s_{E1}(t) = \frac{1}{\sqrt{2}}\Big(e_{E1-B}(t)(\alpha sc_{E1-B,a}(t) + \beta sc_{E1-B,b}(t))\Big) \\ - \frac{1}{\sqrt{2}}\Big(e_{E1-C}(t)(\alpha sc_{E1-C,a}(t) - \beta sc_{E1-C,b}(t))\Big) \end{aligned} \tag{4.3}$$

note that pilot and data component are modulated with a 50% power sharing. Actually, in the given equations and schemes is missing the PRS channel A, which

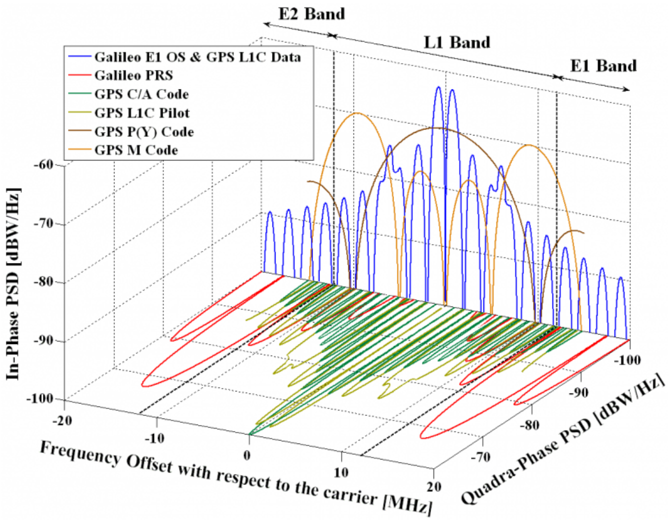


Figure 4.3: The E1 band spectrum for Galileo and all GPS signals (currently used and modernized) [1].

is modulated in-quadrature as showed in Figure 4.3; here is also highlighted how the two systems spectrum are centred around the same carrier to be interoperable, but the corresponding main lobes are not overlapping thanks to BOC modulation.

Power and Noise levels The Galileo authority for the E1 OS signal guarantees a minimum received power on ground equal to -157 dBW, which is measured at the output of an ideally matched RHCP 0 dBi polarized user receiving antenna, when the satellite elevation angle is higher than 10 degrees.
While the noise power density depends on the receiver noise temperature:

$$N_0 = kT_{eq} = kT_0 \tag{4.4}$$

where k is the Boltzman constant, and T_0 is the typical receiver noise temperature equal to $\simeq$ 300 $K°$. Then, we have $N_0 \simeq$ -204 dBW/Hz = -174 dBm/Hz, and a consequently carrier-to-noise ratio $\frac{C}{N_0}$ = -157dBW + 30 dBW - (-174 dBm/Hz) $\simeq$ 47 dBW/Hz. Notice that the corresponding signal power density, C/B, where B = 1.023 MHz (the chip rate), corresponds to -217 dBW/Hz. That is, a unit gain receiver antenna is not able to distinguish the signal spectrum.

4.0.2 Message structure

The data bit stream $D_i(t)$ can be structured depending on three different message type:

- **F/NAV**: used for OS over the E5a signal;
- **I/NAV**: used for OS, SOL and CS over the E5b and E1-B signals;
- **C/NAV**: used for CS over the E6-B signal.

According to our chosen service, and working frequency, we will make reference to I/NAV, namely the Integrity Navigation Message. It is a stream of *Frames* - each lasting 720 s - which in turn are made of 24 *Sub-frames* of 30 s each, and finally one of this cover 15 *Pages* each lasting 2 s [1]. This is the elementary message component, and it can be *nominal* - for normal operation conditions, or of *alert* type, which provides parameters to compute the integrity risk to support Safety-of-Life applications.
There are several new aspects in comparison to GPS L1 message, for instance during transmission the pages are swapping between both OS frequencies, in order to allow a fast data reception to a dual frequency receiver, and leaving to the old model receiver an unchanged receiving time (i.e. 30 seconds for the complete SV ephemeris data). Bits allocation within pages is here neglected, because the current work does not work at this level.

Chapter 5

Derivation of bounds on the key size

Within this chapter the theoretical analysis performed at the Physical-Layer will be outlined. In fact we are going to search for a Physical Layer authentication, that is a fundamentally different paradigm where the security is achieved by exploiting the physical layer properties of the communication system, such as thermal noise, interference, and the time-varying nature of fading channels. This work aims to identify theoretical bounds for having *unconditionally* secure authentication and integrity protection of message and signal, that is, regardless of the computational capabilities of the attacker. Clearly, it is a completely different approach from those seen in section 3.2, and its strength lies on Information Theoretic foundations. As a consequence, there will be a large use of Information Theory concepts, as *entropy*, H, and *mutual information*, I, of random variables.
Therefore, taking a general symmetric-key scheme defined by its spaces (e.g. relative to message, tag and key), algorithms (e.g S and V) and distributions

$$\left(\mathcal{M}, \mathcal{K}, \mathcal{X}, \mathbb{S}, \mathbb{V}, p_u, p_x\right)$$

we are going on stating when it is unconditionally secure.

Unconditionally secure message authentication: The scheme provides $\epsilon -$ *unconditionally* secure source *authentication*, if for any *forging* attack $(p_{u'}, p_{x'|u'})$ its probability of success is ϵ upper-bounded.

$$P_{SF} = P\{V(x^{'}, k) = (u^{'}, \text{ok})\} \leq \epsilon_F \tag{5.1}$$

It can be proved that, the necessary condition for having $\epsilon -$*unconditionally* secure source *authentication* is

$$H(k) \geq \log_{1/2} \epsilon_F \tag{5.2}$$

where $H(k)$ is the key entropy, measured in bits. Alternatively, by using the mutual information between k and x, $I(k;x)$ (i.e. the key leakage), and the key entropy conditioned on the observed x, we have $H(k) = I(k,x) + H(k|x)$; a tighter condition is [24]:

$$I(k,x) \geq \log_{1/2} \epsilon_F \tag{5.3}$$

Unconditionally secure message integrity protection: The scheme provides $\epsilon -$ *unconditionally* secure message *integrity protection*, if for any *modification* attack $(p_{x'|x})$ its probability of success is ϵ upper-bounded.

$$P_{SM} = P\{V(x^{'}, k) = (u^{'}, \text{ok}) \wedge (u^{'} \neq u)\} \leq \epsilon_M \tag{5.4}$$

Analogously to authentication, a necessary condition for $\epsilon -$*unconditionally* secure message *integrity protection* is [24]:

$$H(k|x) \geq \log_{1/2} \epsilon_M \tag{5.5}$$

Then, in authentication the cheating probability can never be reduced to zero, but it can only be made arbitrarily small by using a secret key of sufficient size [24].

5.1 Navigation data authentication at the physical-layer

The first issue that we are going to treat is the search for a theoretical bound to the data authentication and integrity protection. This analysis has been ac-

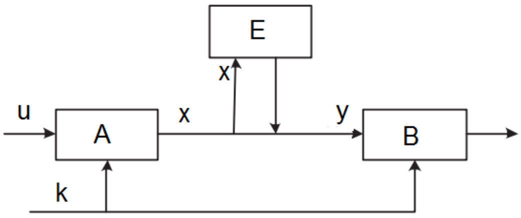


Figure 5.1: The authentication channel model.

complished in two steps, first with a simple, ideal scheme (i.e. noiseless) has been considered, and secondly using the former result, a more realistic case has been considered.

From here onwards, in accordance with the traditional terminology in the security literature, we consider three different agents: Alice, Bob and Eve. In particular, Alice is the legitimate transmitter who starts the communication, Bob is the intended receiver and Eve is the adversary who transmits toward Bob with the aim of impersonating Alice.

5.1.1 The noiseless channel model

We start our analysis with the simplified channel model in Figure 5.1 taken from [24] analysis, made of:

1. a *noiseless* public channel over which Alice transmits sequences of i.i.d. symbols such $\mathbf{x} = (x_1, ..., x_n)$;

2. a noiseless private-authentic channel used for sharing the correct and authentic key $\mathbf{k}$ with Bob.

Since $\mathbf{x}$ crosses a public channel, it may incur an attack (forging/modification), and the received sequence, $\mathbf{y}$, at Bob side may be changed. Since, we are interested in dimensioning a key that will guarantee message authentication, as well as integrity protection, joining (5.3) and (5.5) results in:

$$H(\mathbf{k}) \geq \log_{1/2} \epsilon_F + \log_{1/2} \epsilon_M \tag{5.6}$$

However, (5.6) holds for a single message authentication, while in order to be as generic as possible, we are interested in finding a bound that holds for repeated authentication with the same $\mathbf{k}$. Let's suppose that for ℓ consecutive authentication rounds we exploit the same key, the corresponding key entropy is:

$$H(\mathbf{k}) = I(\mathbf{k}, \mathbf{x_1}) + ... + I(\mathbf{k}, \mathbf{x}_\ell | \mathbf{x_1}, ...\mathbf{x_{\ell-1}}) + H(\mathbf{k}|\mathbf{x_1}, ...\mathbf{x}_\ell) \tag{5.7}$$

Since each observed $\mathbf{x}_i$ restricts the set within which the correct key should be searched, we must treat the increasing *key-leakage*, or equivalently the decreasing key entropy.

Each message $\mathbf{x}_i$ should be highly correlated with $\mathbf{k}$, in order to allow Bob to correctly verify its authenticity through the key. Consequently, to defeat any forging attack, each mutual information term contained in (5.7) has to be lower bounded by $\log_{1/2} \epsilon_F$.

While, against a modification attack we can consider the following chain rule:

$$H(\mathbf{k}|\mathbf{x}_1) \geq H(\mathbf{k}|\mathbf{x}_1, ..., \mathbf{x}_i) \geq H(\mathbf{k}|\mathbf{x}_1, ..., \mathbf{x}_{i+1}) \geq H(\mathbf{k}|\mathbf{x}_1, ..., \mathbf{x}_\ell) \tag{5.8}$$

Then, in general to satisfy both (5.5) and (5.8), it results:

$$H(\mathbf{k}|\mathbf{x}_1, ..., \mathbf{x}_i) \geq \log_{1/2} \epsilon_M, \qquad \forall\, i = 1, ..., \ell \tag{5.9}$$

Therefore, taking into account the highest accepted success probability and (5.7), the resulting minimal key entropy is:

$$\boxed{H(\mathbf{k}) \geq \ell \log_{1/2} \epsilon_F + \left(\log_{1/2} \epsilon_M\right)} \tag{5.10}$$

Basically, setting a desired maximum accepted success probability of the attack, and the key renewal period (ℓ), this formula, derived in [24], provides us the minimum key entropy (e.g. bits length).

Furthermore, as outlined in section 3.2, the single message $\mathbf{x}$ is generally defined as the pair $(\mathbf{u}, \mathbf{t})$. As regards the construction of $\mathbf{t}$, we adopt a Wegman-Carter scheme, that is the authentication and integrity protection equivalent of one-timepad. The fundamental block in this scheme is called an ϵ-*almost strongly universal hash* function. That is a set, $\mathcal{H}$, of maps between the finite sets $\mathcal{M}$ and $\mathcal{T}$ with the two following properties [25]:

1. The number of hash functions in $\mathcal{H}$ that takes $u_1 \in \mathcal{M}$, to $t_1 \in \mathcal{T}$ is exactly $|\mathcal{H}|/|\mathcal{T}|$,

2. The fraction of those functions that also takes $u_2 \neq u_1$ in $\mathcal{M}$ to $t_2 \in \mathcal{T}$ and $t_2 = t_1$, is no more than ϵ

Such a defined $\mathcal{H}$ states that all values of the tag are equally likely if the key is unknown, and even if one message-tag pair is observed, all values of tags corresponding to a forged or modified message are still equally likely. Therefore, by setting $|\mathcal{T}| = 1/\ \epsilon_F$, we can state that Wegman-Carter achieves tightly the forging attack bound. The same, does not hold for the modification attack, then ϵ_M is a looser bound.

5.1.2 The noisy wiretap channel model

Since the real-world channels are noisy, and in a radio-communication such as the satellite one, it is impossible to neglect the noise, we have to consider a more suitable model. Because of our requirements, we have found suitable the authentication-channel model proposed in [26], and depicted in Figure 5.2. It has three components:

1. A noiseless, one-way public channel, that goes from Alice, to Bob through Eve. Over this channel Alice transmits the message $\mathbf{s}$, which may be $\mathbf{u}$ or an equivalent version of it;

2. A wiretap channel, made of three branches and thus defined: $\mathcal{X} \rightarrow (\mathcal{Y}, \mathcal{Z})$, where $\mathcal{X}$ is the input alphabet, $\mathcal{Y}$ is the output alphabet at the legitimate receiver, and $\mathcal{Z}$ is the output alphabet at the wiretapper. The first branch is the main discrete memoryless channel (DMC) $W_1 : X \rightarrow Y$, then there is a second DMC, $W_2 : X \rightarrow Z$, and a noiseless channel links Eve to Bob;

3. A private, noiseless, and authenticated channel over which Alice shares a secret key with Bob.

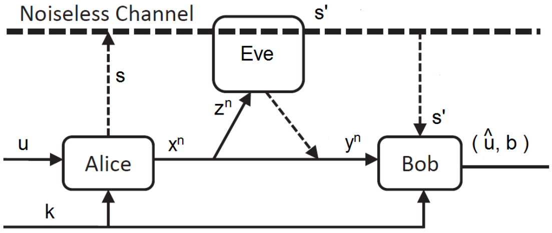


Figure 5.2: The authentication channel model [26].

With comparison to the former case, the main difference is the introduction of the *wire-tap* channel. This latter was firstly developed by Wyner, who aimed to provide information-theoretic secrecy assuming that the link Alice-Eve was noisier than the main one. This setting has become particularly useful for information and coding purposes, then it has been exploited also for authentication. Then, in [27] a wiretap channel is used to hide by noise the authenticating key, but in all the current literature it is always assumed that mutual informations satisfy

$$I(\mathbf{x}, \mathbf{y}) > I(\mathbf{x}, \mathbf{z})$$

In other words, the channel toward the legitimate receiver is less noisy. On the other hand, as already stated, we aim to find a theoretical bound assuming that Eve has unbounded computing resources, and a good hardware equipment (e.g. a low-noise receiver, or an high antenna). Therefore, using the preceding result, (5.10), and adding new requirements imposed by the actual setting, we are going to dimension the needed key size for a generic authentication and integrity protection protocol.

Since whenever Alice wishes to transmit a message $\mathbf{u} \in \mathcal{M}$ to Bob in an authentic way, she has to split message and authenticator between the public noiseless channel (e.g. an Internet connection that makes use of error correcting mechanism), and the wiretap noisy channel (e.g. the satellite channel in our scenario), the communication **reliability** is ensured by the following condition:

$$C_0 + nI(x, y) \geq H(\mathbf{u}) + H(tag) \tag{5.11}$$

where C_0 is the public channel capacity, and analogously the wiretap channel capacity is expressed as $nI(x, y)$ - the maximal mutual information on the single symbol, multiplied by the $\mathbf{x}$ sequence length. In other words, the two capacities joined together must contain at least the minimum message and the corresponding authenticator lengths, that is their entropies. How the pair ($\mathbf{u}$, $\mathbf{x}$) should be divided requires an optimization, since we are asking for two opposites requirements, on one hand we wish to minimize the wiretap usage since Bob will receive a degraded copy, but at the same time we have to keep negligible the probability of a successful attack by Eve, that is a low noiseless channel exploitation.

The second needed requirement regards the desired level of **security**, then the (5.10) becomes:

$$H(\mathbf{k}) \geq \log_{1/2} \epsilon_M + \ell\Big(\log_{1/2} \epsilon_F + n\Big[I(x, z) - I(x, y)\Big]^+\Big) \tag{5.12}$$

where as previously ℓ stands for the number of repeated authentications via the same key, while n is the number of symbols sent over the noisy channel. In (5.12), ϵ_F and ϵ_M represent the maximal accepted forging or modification success probability, while the new term $n\Big[I(x, z) - I(x, y)\Big]^+$ is a consequence of the noisy wiretap usage. That is, this latter is an index of the additional key bits needed against the noise that affects $\mathbf{y}$ - it is as if the key provides Forward Error Correction (FEC) capability. Notice that only positive values of $I(x, z) - I(x, y)$ are considered, in other words it is an *equally capable attacker* assumption. For other values of $[I(x, z) - I(x, y)]$, (5.10) holds again, namely:

- $I(x, z) - I(x, y) = 0 \Rightarrow$ is equivalent to the noiseless public channel case, where both y and z carry the same amount of information about x, unless Eve performs an active attack;

- $I(x, z) - I(x, y) < 0 \Rightarrow$ it means that Eve is less capable than Bob, and she is not able to retrieve useful information from $\mathbf{z}$. Then, the task of $\mathbf{k}$ is only to enable verification, but no redundant bits are needed for error correction.

Going back to our primary aim, that is searching for the theoretical size for a multi-messages authenticating secret key; we will join together requirements (5.11) and (5.12). Their parameters, as C_0, and $H(\mathbf{u})$ depend on the adopted channel and message input alphabet, while ϵ_F, ϵ_M and $H(tag)$ give us the desired security level. Furthermore, since the unconditionally secure scheme of Wegman-Carter set $H(tag) = \log_{1/2} \epsilon_F$, then it can be suitable taking the *tag* entropy at least equal to this logarithm. Therefore the only un-known value is the sequence length n, from the reliability requirement is preferable to take it has higher as possible, but on the other hand it will implies an increasing key size. Then, once all the rest has been fixed, the linear system will delete the n dependency. From the (5.11) one find that:

$$n \geq \frac{H(\mathbf{u}) + \log_{1/2} \epsilon_F - C_0}{I(x,y)}$$

And substituting it into (5.12), it results:

$$\begin{aligned} H(\mathbf{k}) &\geq \log_{1/2} \epsilon_M + \ell\Big(\log_{1/2} \epsilon_F + \frac{H(u) + \log_{1/2} \epsilon_F - C_0}{I(x,y)} \Big[I(x,z) - I(x,y)\Big]^+\Big) \\ &= \log_{1/2} \epsilon_M + \ell\Big(\log_{1/2} \epsilon_F + (H(\mathbf{u}) + \log_{1/2} \epsilon_F - C_0)\Big[\frac{I(x,z)}{I(x,y)} - 1\Big]^+\Big) \end{aligned}$$

Therefore, we get that the key entropy per authentication round is lower-bounded as follow

$$\boxed{\frac{H(\mathbf{k})}{\ell} \geq \log_{1/2} \epsilon_M + \log_{1/2} \epsilon_F + \Big[H(\mathbf{u}) + \log_{1/2} \epsilon_F - C_0\Big]\Big(\frac{I(x,z)}{I(x,y)} - 1\Big)^+} \tag{5.13}$$

Thus, finally we have get a theoretical bound on the key entropy (i.e. minimal size) which guarantees us the desired security level regardless the particular authentication and integrity protection protocol adopted, that only depends on how much we use the wiretap channel, and Eve's advantage.

Notice that this idea imply a symmetric key authentication scheme, and as already highlighted, it is quite infeasible in GNSS, therefore it is assumed that $\mathbf{k}$ will be disclosed by a broadcast scheme after a while. After that, everyone will know exactly the chosen $\mathbf{k}$, thus it appears in contrast with the initial hypothesis of using a private channel to reveal the key. However, the channel is private as meaning that the key is not available to Eve at the useful instant, but instead when it is no more valid.

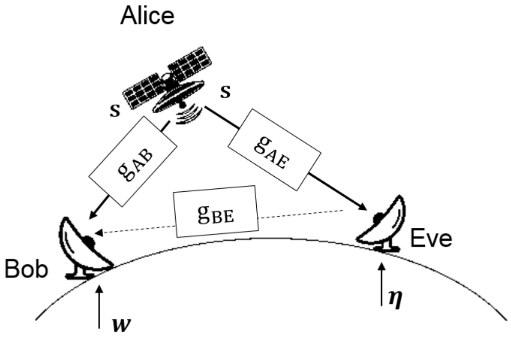


Figure 5.3: The replay-attack scenario against channel authentication.

In conclusion, via a key **k** such defined we are providing navigation data authentication and integrity protection, that is Bob is ensured about the bit content authenticity. Then, any forging or modification attempt is inhibited, but Eve has the freedom to mount any kind of *replay* attack. The latter requires a signal layer defense.

5.2 Signal authentication at the physical-layer

As already highlighted, the signal geometrical path, and the consequently time-of-arrival authentication is the most critical aspect of the problem.
The techniques listed in (3.2.2) were focused on authenticating the signal via secret spreading sequences, water-marking or other way of modifying the signal such that the imitation or recognition of it are hard. However, we have also seen that these mechanisms do not defeat high gain antennas. Therefore, once again herein we have tried a different way. Namely, the idea is taken from wireless physical layer security, and is based on channel authentication. As a matter of fact, two signals with the same origin (e.g. the satellite) and end vertex (e.g. the legitimate receiver), but different path in the middle are distinguishable at the physical layer because of different channel effects. Then, the channel impulse response can be used to provide the radio channel authentication, that is a *link signature.*

5.2.1 The channel authentication scenario

Figure 5.3 depicts a generic replay-attack scenario, where $\mathbf{s}$ is the sequence broadcast by the satellite, and its received copy depends on the receiver location, and additive noise, here called $\mathbf{w}$ and η. Therefore, three different signals can be defined:

- the signal legitimately received at Bob's location

$$\mathbf{r}_B = \Big(\mathbf{s} * g_{AB}\Big) + \mathbf{w} \tag{5.14}$$

- the signal received at Eve's location

$$\mathbf{r}_E = \Big(\mathbf{s} * g_{AE}\Big) + \eta = \mathbf{s}^{'} \tag{5.15}$$

- the signal received at Bob location, as replayed by Eve

$$\mathbf{r}^{'}_B = \Big(\mathbf{s}^{'} * g_{EB}\Big) + \mathbf{w} \tag{5.16}$$

In order to defeat any replay attack, we can think about taking a *reference* evaluation of the channel g_{AB}, and for each signal received during the validity of the reference estimate, we are going to repeat the channel estimation, and if the current estimation is consistent with the reference one, the signal will be accepted as authentic. Obviously, this idea requires a certain channel staticity, and the knowledge of the original transmitted signal, but we will deal with these hypotheses later. It is however important to note that this technique leads to a new form of attack, called *mimicry attack* [28], which aim to rend $\mathbf{r}_B = \mathbf{r}^{'}_B$. Basically, Eve action can take place with two steps:

- At the beginning of the *coherence* time interval, she can act in a way that Bob computes the *reference* estimation over the replay-attack channel, that is $g_{AE} * g_{EB}$. In this manner, during all the coherence time she can perform a replay-attack, and each signal such received will pass the verification test.
- By assuming that Eve has been able to estimate $\tilde{g_{AB}}$, then in order to make $\mathbf{r}_B = \mathbf{r}^{'}_B$ having the same input signal, Eve must make $g_{AE} * g_{EB} * h = \tilde{g_{AB}}$

[29]. Where h may be the result of a processing performed over $\mathbf{r}_E$, or of a physical antenna motion. Then $\mathbf{r}'_B$ will be considered authentic.

To protect ourselves against this kind of attack the cryptography is applied. Practically, we can design the signal $\mathbf{s}$ of Figure 5.3 as a *training sequence*, which may be sent over a fourth new OS component (e.g. E1-D). In order to defeat Eve's actions, $\mathbf{s}$ must have the following properties:

- it has maximum differential entropy, that is, the training sequence is Gaussian. Therefore, even if Eve passively observes $\mathbf{r}_B$, it is difficult to estimate the channel g_{AB} impulse response in time to forge her channel;

- after a while, any receiver can reconstruct it via a key, and with the previously received sequence he estimate his own channel reference impulse response. The key is disclosed over a noiseless, and authenticated channel - then Bob does not risk to use a fake training sequence.

With such a defined training sequence, Eve cannot remove η and demodulate her $\mathbf{r}_E$, since this latter is the sum of two random and independent Gaussian. Therefore, to retrieve $\mathbf{s}$ is quite infeasible without knowledge of the key, and the only possibility for Eve is to keep η very low, and forward $\mathbf{r}_E$ as defined in (5.16). Therefore, we are not going to bound Eve's skills and equipment, Bob will recover any type of disadvantage by means of the key. Here below an example of this approach will be explained, and the theoretical key size will be computed.

5.2.2 The single-tap impulse response

In the following we are going to use simple but effective hypothesis which show the robustness against the strongest attack by Eve. Making reference to Figure 5.3 scenario, both Bob and Eve are not moving, while as regards the legitimate channel, g_{AB}, it is designed as a single-tap of unitary amplitude, whose only effect is a propagation time delay:

$$g_{AB} = \delta(t - \tau_{AB})$$

that is, there are not any multipath or attenuation effect. Obviously, it is not a realistic channel, but sufficient to show the effectiveness of our idea. Furthermore, here it is assumed that Bob has already estimated g_{AB}; and we allow Eve to carry out her aim, that is, to imitate g_{AB}. Knowing which is the purpose of Eve, let us assume that in some way she has been able to estimate g_{AB}, then since the signal processing is quite infeasible, she locate herself such that the two channel composition, $g_{AE} * g_{EB}$, introduces a delay $\tau_{AE}+\tau_{EB} \simeq \tau_{AB}$. The last, but not negligible effect on the signal is the receiver Gaussian noise, respectively $w \sim \mathcal{N}(0, \sigma_w^2)$ for Bob, and $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ for Eve, which are independent between them. And as said above, $\mathbf{s}$ is an other Gaussian noise, with its specific statistics, $\mathcal{N}(0, \sigma_s^2)$.

These hypothesis result is that Bob will simultaneously receive two sequences, $\mathbf{r}_B$ and $\mathbf{r}_B^{'}$, of Gaussian symbols:

$$\begin{gathered} r_B(n) \sim \mathcal{N}(0, \sigma_s^2 + \sigma_w^2) \\ r_B^{'}(n) \sim \mathcal{N}(0, \sigma_s^2 + \sigma_\eta^2 + \sigma_w^2) \end{gathered}$$

Therefore, via a so-called *Hypothesis testing* Bob should decide which one among the two observed symbols is authentic. In other words, with the assumption that the authentic, and spoofed signals have two different Gaussian distribution, for each symbol Bob should evaluate the probability that it belongs to the authentic or fake distribution. However, the hypothesis testing in some cases fails, and more precisely there are two different errors that can occurs:

- a *false alarm*, which discard an authentic observation as if it was false;
- a *missed detection*, which does not detect a false observation, and accepts it if it was authentic.

Has already done, we are going to accept a maximum error probability, which upper bounded the hypothesis evaluation failure.

In order to take a choice between the two hypothesis, the Information Theory give us the Kullback-Leibler divergence, which is a non-symmetric measure of the difference between two probability distribution P and Q, and thus defined for discrete variables:

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \text{ [bits]}$$

However, if Bob should evaluate a divergence only via a priori observation, he would be in the same poor conditions of Eve. Therefore, here fit the key role, namely after a while over a noiseless, private and authenticated channel the satellite sends the key $\mathbf{k}$, which gives Bob the possibility to retrieve perfectly the original sequence $\mathbf{s}$. Then, since the only channel effect is introducing delay and adding noise, we have that:

$$\begin{aligned} \mathbf{r}_B - \mathbf{s} &= \mathbf{w} \\ \mathbf{r}_B^{'} - \mathbf{s} &= \mathbf{w} + \eta \end{aligned}$$

thus Bob can discriminate over the measured noise. Obviously, Eve can adopt the strategy of introducing a very low noise, but on the other hand we can face it with a more suitable key. First of all, we are going to take into account the following Kullback-Leibler divergence

$$D_{KL}(H_0\|H_1) = D_{KL}\Big(p(\tilde{\mathbf{r}_B}, \mathbf{k}|g_{AB})\|p(\tilde{\mathbf{r}_B}, \mathbf{k}|g_{AE} * g_{EB})\Big) \tag{5.17}$$

where H_0 and H_1 are respectively the authentic and spoofed signal hypothesis, while $\tilde{\mathbf{r}_B}$ is the generic sequence arrived at Bob receiver. We wish to find how much are distinguishable two sequences one received from the satellite, and the other forwarded by a spoofer, jointly with the key knowledge. Applying the channel condition (i.e. making the hypothesis *authentic* or *spoofed*), the divergence expression is simplified:

$$\begin{aligned} D_{KL}(H_0\|H_1) &= D_{KL}\Big(p(\mathbf{r}_B, \mathbf{k})\|p(\mathbf{r}_B^{'}, \mathbf{k})\Big) \\ &= D_{KL}\Big(p(\mathbf{r}_B|\mathbf{k})p(\mathbf{k})\|p(\mathbf{r}_B^{'}|\mathbf{k})p(\mathbf{k})\Big) \end{aligned} \tag{5.18}$$

Since the knowledge of $\mathbf{k}$ should allow Bob to reconstruct $\mathbf{s}$ precisely, let assume $\mathbf{k} = \mathbf{s}$. In addition, once the key has been disclosed it is a deterministic term, then the formula is even more simplified:

$$D_{KL}(H_0\|H_1) = D_{KL}\Big(p(\mathbf{r}_B|\mathbf{s})\|p(\mathbf{r}_B^{'}|\mathbf{s})\Big) \tag{5.19}$$

Then we have to discriminate between symbols with this two new distributions:

$$\begin{aligned} r_B(i)|s(i) &\sim \mathcal{N}(0,\sigma_w^2) \\ r_B^{'}(i)|s(i) &\sim \mathcal{N}(0,\sigma_w^2+\sigma_\eta^2) \end{aligned}$$

Furthermore, given that we have i.i.d. symbols, the divergence over the entire sequence is expressed as a sum of single symbols divergences. Now, we are going to exploit the general expression of D_{KL} for zero mean Gaussian variables, $p(x) \sim \mathcal{N}(0,\sigma_1^2)$ and $q(x) \sim \mathcal{N}(0,\sigma_2^2)$

$$\begin{aligned} D(P\|Q) &= \int p(x)\ln\Big(\frac{p(x)}{q(x)}\Big)dx \\ &= \frac{1}{2}\ln\Big(\frac{\sigma_2^2}{\sigma_1^2}\Big) - \frac{1}{2} + \frac{1}{2}\Big(\frac{\sigma_1^2}{\sigma_2^2}\Big) \end{aligned} \tag{5.20}$$

Going back to our case, we find:

$$D_0 = D\Big(p(r_B(i)|s(i))\|p(r_B^{'}(i)|s(i))\Big) = \frac{1}{2}\ln\Big(\frac{\sigma_w^2+\sigma_\eta^2}{\sigma_w^2}\Big) - \frac{1}{2} + \frac{1}{2}\Big(\frac{\sigma_w^2}{\sigma_w^2+\sigma^2}\Big) \tag{5.21}$$

and

$$D_1 = D\Big(p(r_B^{'}(i)|s(i))\|p(r_B(i)|s(i))\Big) = \frac{1}{2}\ln\Big(\frac{\sigma_w^2}{\sigma_w^2+\sigma_\eta^2}\Big) - \frac{1}{2} + \frac{1}{2}\Big(\frac{\sigma_w^2+\sigma_\eta^2}{\sigma_w^2}\Big) \tag{5.22}$$

To note that via the key knowledge the divergence evaluation must fail with a probability lower than the two chosen maximal P_{FA} and P_{MD}. For the second time the key dimension appears as an important issue, and in addition also this time it helps against the noise. As we have already stated, $\mathbf{k}$ has to give Bob fully information about $\mathbf{s}$, then the desirable property would be $\mathbf{k} = \mathbf{s}$. Unfortunately it is not so practical, in fact keeping the training sequence a Gaussian noise has been a convenient choice, but it also carries an infinite entropy. As a consequence, to guarantee the substitution made in (5.19) and all the previous results, the key should be made of an infinite number of symbols. This is infeasible, then let us take the symbol $\bar{s}$ as a the quantized equivalent of s, $\mathbf{k}$ with a discrete alphabet, and some requirements:

1.
$$H(\bar{s}) > I(s; r_B) + \log_{1/2} \epsilon_F \tag{5.23}$$
that is, the entropy of the quantized symbol, $\bar{s}$, should be greater than the leakage of information that r_B (or r_E) gives about the original Gaussian symbol s. In addition, the second term prevent a potential forging of $\mathbf{s}$ which will imply a complete control over the training phase (e.g. with respect to a just turned on receiver).

2.
$$H(\mathbf{k}) = nH(\bar{s}) \tag{5.24}$$
the key is equivalent to a sequence of n symbols with entropy in agreement with the previous requirement.

A more correct condition for the entropy of $\bar{s}$ would be $H(\bar{s}) > I(\bar{s}, \bar{r_B}) + \log_{1/2} \epsilon_F$, namely the quantized equivalent of 5.23; but once this latter is applied, also the quantized is true. Indeed, the Gaussian channel maximize the mutual information between the input, output symbols. Therefore joining together the two conditions, we obtain $H(\mathbf{k}) > n\Big(I(s, r_B) + \log_{1/2} \epsilon_F\Big)$. The only unknown quantity is n, the length of $\mathbf{s}$, and it will descend from the minimum required divergence over the sequence, $\mathbf{D_{KL}}$, to counteract the probability of error. In conclusion we get:

$$\boxed{\begin{aligned} H(k) &\geq \left(\frac{1}{2}\log_2\Big(1 + SNR\Big) + \log_{1/2} \epsilon_F\right)\frac{\mathbf{D_{KL}}(\sigma_\eta^2)}{D_{KL}(\sigma_\eta^2)} \\ &= \left(\frac{1}{2}\log_2\left(1 + \frac{\sigma_s^2}{\sigma_w^2}\right) + \log_{1/2} \epsilon_F\right)\frac{\mathbf{D_{KL}}(\sigma_\eta^2)}{D_{KL}(\sigma_\eta^2)} \end{aligned}} \tag{5.25}$$

Noticed that everything remain true even if Eve $\tau_{AB} \neq \tau_{AE} + \tau_{EB}$, in fact herein we have assumed the equality for simplicity, but usually the spoofer aim to mislead our timing awareness. For instance he can successfully predict the message and

thus introduce a negative delay - even if it may imply to break the data layer authentication - or rather he performed a selective replay attack introducing an additive delay to τ_{AE}, but in both cases Bob is faced with only one copy and should make the correct hypothesis without any other comparison. Therefore, this skill defend ourselves also against other potential attacks: as a persistent jamming which prevents us from receiving the authentic signal, or a misalignment during the tracking that once again makes us lose the correct signal, and finally it solve the problem of introducing a sequence number to check the presence of any replayed copy.
A critical aspect is determining the channel *coherence* time, and consequently how often Bob should estimate the *reference* link signature, in fact the satellite channel realization depends on several aspects, e.g. the instantaneous ionosphere condition, and any motion attributable to the satellite, the receiver or simply of the surrounding. However, this channel variation is also a good aspect for us, namely let suppose that at the preceding step Eve has observed $\mathbf{r}_B$, and has evaluated g_{AB} as soon as $\mathbf{k}$ became available, but this knowledge does not guarantee a perfect channel imitation an instant later.

In conclusion, if the introduction of such a new OS signal component will be feasible, the idea developed here let us to be confident about the possibility to have unconditionally secure signal authentication. Obviously, it requires more study since the real satellite channel is not a single-tap delta.

Chapter 6

Implementations and results

In this chapter we illustrate the bounds computed in Chapter 5 via $MATLAB^{®}$ numerical evaluations, with certain signals and channels settings. In addition, the Data layer result will be compared with TESLA performance.

6.1 Navigation data authentication and integrity protection

We will firstly treat the key entropy theoretical bound required to unconditionally protect the navigation data message, that is:

$$\boxed{\frac{H(\mathbf{k})}{\ell} \geq \log_{1/2}\epsilon_F + \log_{1/2}\epsilon_M + \Big[H(u) + \log_{1/2}\epsilon_F - C_0\Big]\Big(\frac{I(x,z)}{I(x,y)} - 1\Big)^{+}} \quad (6.1)$$

Regardless of the specific channel model, the security term ϵ_F and ϵ_M can be set. Herein has been selected a common value for both (e.g. 10^{-3}). Furthermore, for simplicity we have assumed that C_0 is completely filled with the navigation message $\mathbf{u}$, or its encrypted equivalent $\mathbf{s}$.

6.1.1 A discrete time memoryless AWGN wiretap channel

Recalling the channel setting of Figure 5.2, we are going to simulate the noisy wiretap channel via the AWGN channel model. As a matter of fact, we need to model the channel additive noise, and the AWGN is suitable since it is a simple model which only impairs the communication means a wideband or white noise with a constant spectral density and a Gaussian distribution of amplitude. Even if it does not account for other complex impairments, it produces simple and tractable mathematical models. The following general assumptions are still valid: Alice and Bob are connected via a discrete time memoryless (DMC) channel $W_1 : X \to Y$, and a second DMC is between Alice and the wiretapper, $W_2 : X \to Z$. Now, the channel input symbol, x_i, are a continuous variable power constrained, that is $E[x_i^2] \leq P$. And the corresponding outputs symbols, y_i, are expressed as the sum of the input x_i, and noise w_i, which is independent and identically distributed as $\mathcal{N}(0, \sigma_w^2)$. In addition, the w_i are not correlated with the input x_i, and analogously we have $z_i = x_i + \eta_i$, with $\eta_i \sim \mathcal{N}(0, \sigma_\eta^2)$. Therefore, based on these assumption the mutual informations will be properly defined (for simplicity of notation the sub-scripts will be neglected) as:

$$I(x, y) \leq \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma_w^2}\right) \tag{6.2}$$

$$I(x, z) \leq \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma_\eta^2}\right) \tag{6.3}$$

The preceding definitions can be considered equalities in the particular case of Gaussian symbols with $x_i \sim \mathcal{N}(0, P)$, where P is the maximum or *constrain* power, which accounts for the transmitting power, and the overall channel effect, e.g. transmitter and receiver antennas gains, plus the free space path loss. While the noise power (i.e. variance) may also be indicated as $N_0 B$, where B is the receiver operational bandwidth. Furthermore, we have to remind that the wiretap channel corresponds to the satellite channel, then the received signal is not x_i itself, but rather x_i multiplied via its PRN code. As a consequence P also includes the additional correlation gain:

$$I(x, y) = \frac{1}{2} \log_2 \left(1 + \frac{P_{tx} |h|^2 g_{corr}}{\sigma_w^2}\right) \tag{6.4}$$

$$I(x, z) = \frac{1}{2} \log_2 \left(1 + \frac{P_{tx} |g|^2 g_{corr}}{\sigma_\eta^2}\right) \tag{6.5}$$

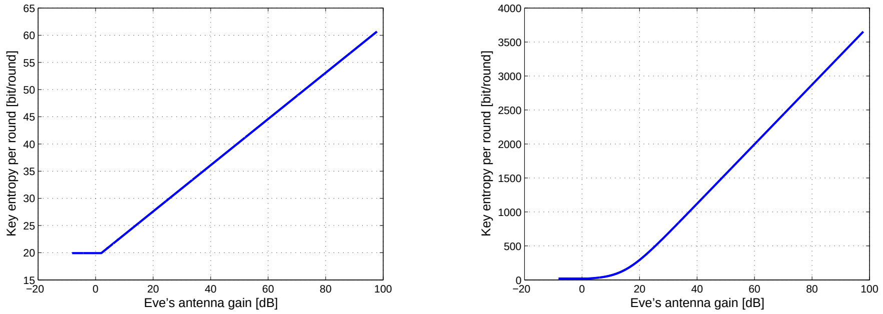
Parameter	Notation	Value
Satellite transmitting power	P_{tx}	40 W
Satellite antenna gain	g_{tx}	14.5 dBi W
Free-space pathloss	a_{PL}	180 dB
Antenna efficiency	η	0.7
Receiver antenna gain	g_{rx}	2 dBi
Noise power density (at 300 K)	N_0	-204 dBW/Hz
Passband filter	B	8 MHz

Table 6.1: Satellite and receiver specifications.

where g_{corr} is expressed as T_{symb}/T_{chip}. In accordance to Galileo parameters we have:

$$\begin{aligned} g_{corr} &= \frac{1/R_{symb}}{1/R_{chip}} = \frac{R_{chip}}{R_{symb}} \\ &= \frac{1.023Mchip/s}{250bit/s} = 4.092 \cdot 10^3 \simeq 36dB \end{aligned} \tag{6.6}$$

Now, from the point of view of an attacker who wishes to increase his information on x gained via z, the degrees of freedom in equation (6.5) are his antenna gain, or his receiver noise power. In the following we are going to consider an even higher attacker antenna gain, while leaving all the remaining parameters at their nominal values of Table 6.1. Figure 6.1(a) depicts how the key dimension grows versus an increasing attacker antenna gain, and a noise power fixed for both parties at the room temperature nominal value. One can notice that the curve starts increasing linearly as soon as Eve's antenna gain exceed the nominal value of 2 dBi, namely when the attacker becomes more capable than the mass-market receiver. Then, the first curve portion - almost flat - is defined by the Wegman-Carter scheme bound. As a consequence of the greater attacker capacity, we must defend ourselves and close the disadvantage gap with a key of greater length, that can also perform error correction. In order to show the meaning of the numerical results, we can observe that a 40 dBi antenna gain corresponds to a radium $r \sim 2.6$ m, which is an unusual dish antenna size if compared with a 2 dBi antenna of 30 cm size. However, the important outcome is that such an attacker equipment will require about 37 key



(a) At bit level.

(b) At chip level.

Figure 6.1: The theoretical key entropy needed at each authentication against a specific attacker antenna gain.

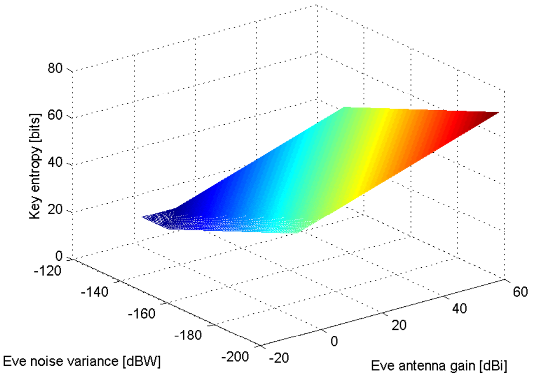
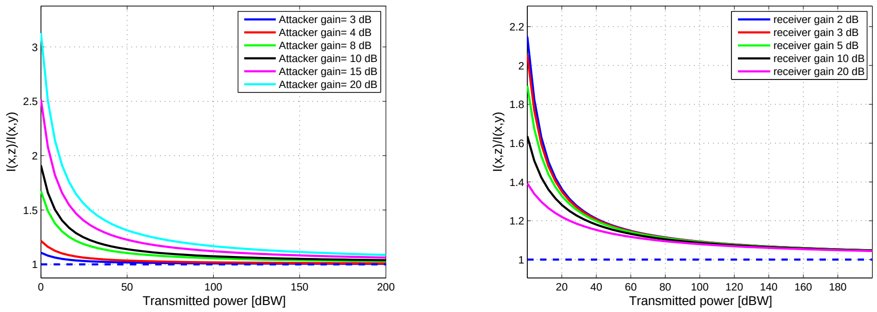


Figure 6.2: The key entropy lower bound for a Gaussian input, against a specific attacker noise variance, and antenna gain.

bits, that is still viable. While, only for a comparison purpose, Figure 6.1(b) makes reference to a verification test made before the de-spreading case, in which the SNR does not contain the correlation gain. In this case a 40 dBi antenna gain will be defeated with more than 1000 bit key, the reason lies in the great amount of noise which cover the symbol, therefore the key should perform a strong error correction. Nevertheless, this curve is given only as an example, because we are interested in the data message authentication, or rather the *bit* level.



(a) With Bob antenna gain fixed at 2 dBi.

(b) With antennas gain difference, $g_{Eve} - g_{Bob}$, fixed at 10 dBi.

Figure 6.3: $I(x;z)/I(x;y)$ behaviour versus P_{tx}.

Figure (6.2) gives a further and complete proof of how the attacker condition modifies our data authenticating key size. In this case also the noise power effect has been taken into account, one can observe that the key entropy grows rather with an increasing antenna gain for Eve, or for a decreasing noise variance, σ_η^2, at her receiver, as reasonably expected. However, as already done above, throughout the rest of this section we focus on the antenna gain value, since it mainly affects the corresponding key value.

Therefore, the term $\left(\dfrac{I(x,z)}{I(x,y)} - 1\right)^+$ appears to severely weigh the key size; to inhibit its undesired effect we could think about increasing the transmitting power, that is:

$$\lim_{P_{tx}\to\infty} \frac{\frac{1}{2}\log_2\left(1 + \dfrac{P_{tx}|g|^2 g_{corr}}{\sigma_\eta^2}\right)}{\frac{1}{2}\log_2\left(1 + \dfrac{P_{tx}|h|^2 g_{corr}}{\sigma_w^2}\right)} = 1$$

Therefore, neglecting the predefined P_{tx} nominal value, we are going to examine the effect of an increasing satellite transmitting power on the ratio $\dfrac{I(x,z)}{I(x,y)}$. From Figure 6.3a we see that the information gain can be overcome via a small power increase only up to a 2 dBi stronger attacker, while in the remaining cases the ratio approaches to 1 for power values around 100-120 dBW. In case (b) the curves are closer to the asymptote, but they reach it slower, then a small power rise does not give an interesting beneficial. In conclusion, a transmitting power adjustment is quite infeasible, mostly if it is compared to the current used value of 16 dBW, thus bounded because of consumption reasons.

6.1.2 Binary input and AWGN wiretap channel

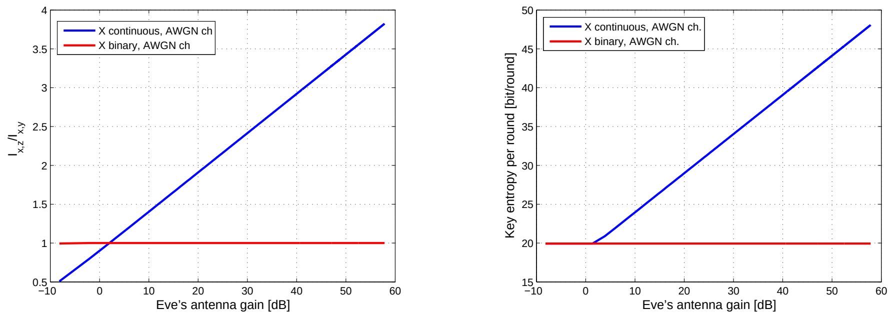
Now we are going to analyse the discrete time AWGN channel, in a more realistic setting where its input symbol x_i no longer has a Gaussian distribution, but is uniformly chosen in the binary alphabet $\mathcal{X} = \{\pm A\}$. As regards the channel noise, all the previous properties still hold, and the same for the power constraint, $A^2 \leq P$. The obvious consequence of such a communication, is a lower entropy both for Bob, and Eve on x_i. As a matter of fact, to cope with the term $I(x;z)/I(x;y)$ we can act in two ways, the former consists on increasing P_{tx} as already discussed above, while the second exploits the input alphabet cardinality - in an M-ary scheme the symbol carries more information if compared to a binary one. Therefore, since a transmitting power of 100 dBW is not suitable, we are going to analyse the binary input performance. In this case, the mutual information between input and output symbols becomes:

$$I(x;y) = \frac{1}{2} f\Big(\frac{A}{\sigma_w}\Big) + \frac{1}{2} f\Big(\frac{-A}{\sigma_w}\Big) \tag{6.7}$$

$$f(a) = \int\limits_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{\dfrac{-(u-a)^2}{2}} \log_2\Big(\frac{2}{1+e^{-2au}}\Big) du \tag{6.8}$$

$$\tag{6.9}$$

and it is analogous for $I(x;z)$. The first important result is depicted in Figure (6.4(a)), that is the mutual information ratio as a function of the attacker antenna gain. Also in this case we have supposed that the attacker exploits his antenna gain, instead of a lower thermal noise. The binary case curve is practically constant and equal to 1 as desired. As a consequence also the key dimension (Figure 6.4(b)) growth is arrested, and remains constant irrespectively of the attacker antenna. Let us understand the reason of such a performance: in an AWGN context, any generic receiver equipped with a mass-market antenna of 2 dBi gain, and at the room temperature, after having correlated the received signal by his own replica, reaches



(a) $I(x,z)/I(x,y)$ behaviour versus the attacker antenna gain.

(b) The theoretical key entropy needed at each authentication against a specific attacker antenna gain

Figure 6.4: A comparison between Binary and Gaussian input.

a perfect knowledge of x_i, or rather $I(x;y) = 1$ bit. By this observation follows that for an attacker, that reasonably and by assumption is at least equally capable, $I(x;z) = 1$. This result may seem contradictory with the requirement of making it hard, to any wiretapper to decode the message, since it can carries key-leakage. However, the initially established ϵ_F and ϵ_M guarantee us that any attempt to guess the correct *tag* by a *forging* or *modification* attack will have the corresponding pre-defined low success probability, namely it is offering unconditionally secure authentication and integrity protection. Any perfect knowledge of $\mathbf{x}$, does not bring any additional information on the key used, thus from the point of view of Eve each *tag* is equally likely. This applies with the assumption that the key is changed each time, otherwise its length will be extended to ℓ times the single authentication key size, in agreement with (6.1). The latter option would be not be costless in term of bandwidth, if the key needs to be broadcast (delayed) via the satellite channel. In conclusion, if security parameters are already sufficiently robust, the infinite entropy Gaussian symbol brings an unnecessary key expenditure.

The last thing that it is worth observing about the data authentication, is a comparison between the theory and the protocol that currently seems to be the most promising, namely TESLA. The version presented in [20] at the Institute of Navigation (ION) proposes to adopt a single key-chain, in which each k_i 82 bits long, and authenticates the navigation message via a MAC of 10 bits. Therefore,

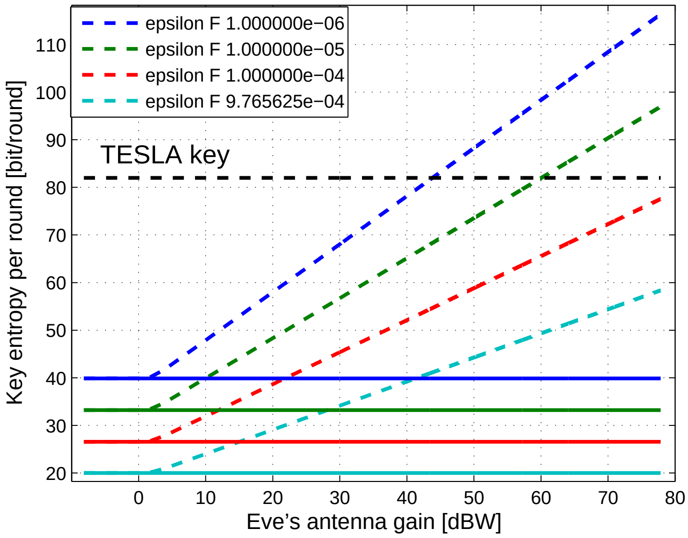


Figure 6.5: Diffrents ϵ_F curves compared to TESLA of [20].

in accordance to our idea we set $\epsilon_F = 1/(2^{10}) = 9.7656 \cdot 10^{-4}$, and equivalently for ϵ_M. Figure (6.5) shows the theoretical key entropy bound (both for Gaussian and Binary input) for several ϵ_F values, and the TESLA k_i size. It appears that the TESLA key is visibly higher than its corresponding ϵ curve, whose binary case states that 20 bit are sufficient to guarantee unconditionally secure authentication. Then, the protocol seems not efficient as it uses four times the bits strictly necessary, thus wasting some bit rate to share the key. However, in accordance to this work we can only state that TESLA with the single k_i, and the corresponding MAC, is providing its users unconditionally secure authentication and integrity protection of the navigation message bits against forging or modification attacks. It does not mean that TESLA key-chain is unbreakable, but our bound is considering the single key, thus we cannot say more.

In conclusion, as regards the practical application, the noisy wiretap channel setting is easily viable. Indeed, if the connection is available the user has the possibility to download the navigation message from the network, while the *tag* comes over the noisy satellite (i.e. wiretap) channel. Otherwise, everything will be received from the satellite, as occurs usually, resulting only in an higher key minimal entropy. And also the key bound performance, observed for the Binary input AWGN channel, is expected to be similar to the real one. In fact, the AWGN model simplifies the mutual information computation, but a multipath channel will not increase it.

6.2 Signal authentication and integrity protection

After having analysed the key entropy bound required at Data layer, we are going on with the second, but not less important, result at the Signal layer:

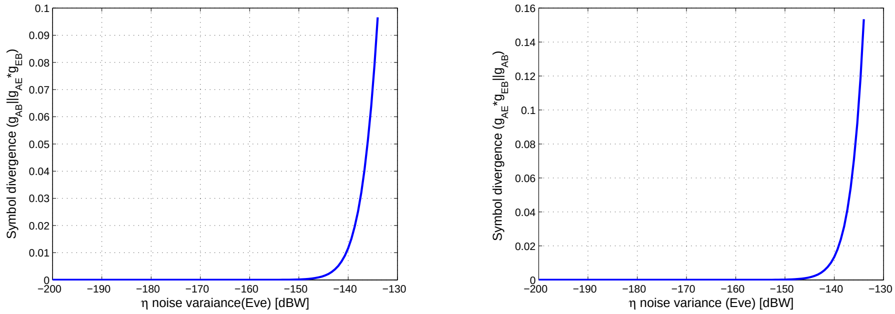
$$\boxed{\begin{aligned} H(k) &\geq \left(\frac{1}{2}\log_2\Big(1 + SNR_{Bob}\Big) + \log_{1/2}\epsilon_F\right)\frac{\mathbf{D_{KL}}(\sigma_\eta^2)}{D_{KL}(\sigma_\eta^2)} \\ &= \left(\frac{1}{2}\log_2\Big(1 + \frac{\sigma_s^2}{\sigma_w^2}\Big) + \log_{1/2}\epsilon_F\right)n_{min} \end{aligned}} \tag{6.10}$$

6.2.1 The single-tap AWGN channel

Remind that in this case, in order to provide a first idea implementation, we are adopting an ideal single-tap AWGN channel, and no correlation gain are considered, since now our focus is the signal how it is get. Let us define the three Gaussian symbols:

1. the transmitted signal, $\mathbf{s} \sim \mathcal{N}(0, \sigma_s^2)$. Whose noise variance has been set equal to -157 dBW, that is, the Gaussian sequence carries the minimum guaranteed power for Galileo E1 signal in space (before the correlation process);

2. the legitimate receiver thermal noise, $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2)$. Whose power spectral density, N_0, has been assumed equal to -204dBW/Hz - the typical expected value at the room temperature on earth. The receiver filtering process will keep the signal and noise power only over a 8 MHz bandwidth, then $\sigma_w^2 = -134dBW$;



(a) $D_{KL}(H_0\| \ H_1)$ (b) $D_{KL}(H_1\| \ H_0)$

Figure 6.6: The Kullback-Leibler divergence as function of σ_η^2.

3. <u>the attacker receiver thermal noise</u>, η. Also this latter is expected to be Gaussian, but we let its power to be lower than the typical value of $-134dBW$ - precisely, σ_η^2 will drop until the value of $-200dBW$, which corresponds to $T \sim 0K^\circ$.

Recalling our idea exposed along 5.2.2, the receiver with the key knowledge should be able to discriminate an authentic symbol r_B from the spoofed one, $r_B^{'}$, observing the corresponding noise power. Therefore, the attacker strategy becomes: being able to low down his receiver thermal noise, σ_η^2 Figure (6.6) depicts the two Kullback-Leibler divergences within the attacker noise power range of values. As we have mentioned, KL divergence is a not-symmetric measure, but the behaviour of curves (a) and (b) against the increasing σ_η^2 is almost the same. Basically, it appears that we are not able to discriminate two symbols (i.e. authentic and false) with more than 15 dBW of power difference.

Unfortunately, an hypothesis test outcome may not be always correct (even under the 15 dBW of difference), namely a *false allarm* or a *missed detection* may occur. Therefore, we are going to set a maximum allowed error probability value for P_{FA} and P_{MD} on the entire sequence, which has to be respected regardless the attacker noise specific value. To clarify, we should make reference to Figure (6.7): the area on bottom left corner defines our desired detection region, within which we are almost confident about our hypothesis. On the other hand, the green line identify the attacker aim, indeed on the diagonal the detection error is sure. The missed

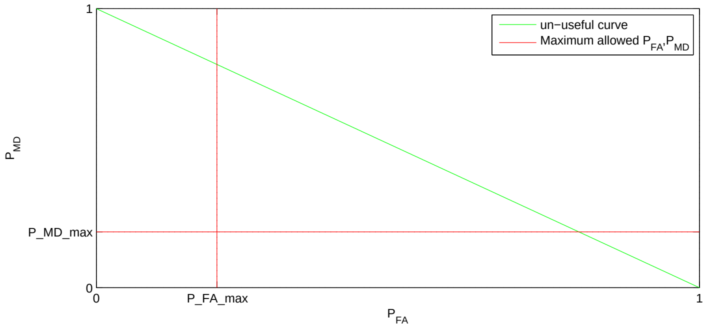


Figure 6.7: The hypothesis test error probability outer-bound.

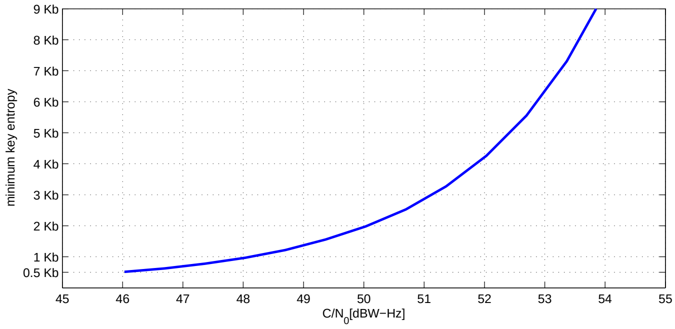


Figure 6.8: The required key entropy versus an increasing C/N_0 for the attacker, and fixed at 47 dBW-Hz for Bob.

detection probability can be defined as $\frac{1}{2^{D_0}}$, and analogously $P_{FA} = \frac{1}{2^{D_1}}$. For this simulation P_{FA} and P_{MD} have been upper bounded by 10^{-3}. Now, the KL divergence per symbol is not so high to respect the constraint, but obviously the training sequence $\mathbf{s}$ is made of several symbols, the problem is how long should it be? The minimum sequence length n_{min} has been computed as that number that multiplied by the KL divergence of Figure (6.6) let us to reach the good detection region. Obviously, in correspondence of lower σ_η^2 values, we have a lower divergence as well, therefore the suitable n_{min} will higher. Once the sequence length has been defined, we are left with the last key requirement, in fact $\mathbf{k}$ should also defeat any forging attack on $\mathbf{s}$ symbols. Then, as already done, we are going to upper bound the successful attack by ϵ_F, which we have set equal to 10^{-3} as well. Finally, Figure (6.8) depicts the key entropy lower bound based on our idea of signal authentication made via a channel estimation.

For a more practical purpose the curve has been expressed as a function of the attacker $\frac{C}{N_0}$ value, instead of on his noise variance, σ_η^2. One can observe that for $C/N_0 = 46dBW$, practically the attacker is in a poor condition, and as a consequence $\mathbf{k} = 500bit$ is enough to prevent a forging attack, and to give Bob the possibility of correctly retrieve $\mathbf{s}$ in order to, secondly, distinguish the received symbol as authentic or fake. As the attacker C/N_0 starts increasing - as an effect of a reduced noise variance (as we simulated) or equivalently of an high gain antenna - the required key entropy grows as well. We can state that up to 55 dBW of C/N_0 for the attacker, the signal authentication is viable with about 10 Kb key entropy, but an even more powerful attack becomes hardly defeated. However, we can think that 55 dBW corresponds to $\sim 45K^\circ$, that is completely different from a common mass-market receiver equivalent temperature. Then, even if the single-tap AWGN channel is not the usual real scenario, it has been useful in order to simulate the attacker potential attack - a replay attack over an approximate authentic channel. And we have been able to find a solution, and the corresponding key to defeat a not-common attacker equipment.

A comparison between the existing signal-layer mechanisms and our bound has not been possible, since these are concentrated on authenticating the signal, while we have though about a channel estimation and authentication. However, 3.2.2 has outlined how these mechanism are not unconditionally secure, for sure. In addition, our idea about distinguishing whether the received signal is authentic or not, seems to be robust also against a Relay attack, that was left un-defeated. In fact, the attacker connecting his antenna to the victim receiver via e.g. a line, or cable, will inevitably add further noise.

While, as regards the E1 new component practical definition, we can only require the symbol Gaussian distribution, and possibly a typical noise power level. The multiplex of it, with the rest has to be well-studied. The satellite PRN code exploitation is not so suitable, since here we are not ranging the training sequence. Then, one can think about assigning a distinct frequency to each SV, namely to employ a FDMA medium access. In this manner the single satellite is allowed to send the same $\mathbf{s}$, by modulating its unique carrier, and multiplexing it with the remaining E1 channels, i.e. A,B, and C. Unfortunately, this option may be vulnerable versus narrow band jamming attacks. Therefore, a second option may be to simply add **s** the other three channels, already spread via the PRN code. And finally, to multiply the whole E1 signal by a second spreading code. In this manner, the data detection and authentication has to wait a previous signal de-spreading and verification. Which makes sense, in fact if the signal present any anomaly we will not waste time in data verification. A such defined multiplexed signal, jointly with a time synchronization between **s** and the PRN code, is going to prevent also an Early Bit Detection attack. Basically, the attacker may be able to detect the authenticator bits, and replay the entire signal with a negative delay. But since it is hard to retrieve **s**, it will be even more infeasible to detect its beginning point, and to successively reconstruct a synchronized signal.

6.2.2 The DLR channel model

As highlighted, the radio navigation signal which reaches our antennas, is what we observe after many propagation and channel effects e.g. ionosphere and troposphere attenuation, scattering, diffraction and reflection. All these phenomena jointly give us a *multipath* propagation, which will be more or less significant depending on the receiver environment (e.g. rural, sub-urban, urban). But, for sure, a delta channel is almost impossible to be met. Furthermore, we need to estimate the channel impulse response at the beginning of a coherence time interval. However with an orbiting satellite, the channel keeps changing, even if there is only a propagation delay effect. Therefore, one can expects how this issue severely increases within a moving scenario, and a time varying multipath channel.

In order to study the channel estimation reliability from the point of view of a legitimate receiver, and of an attacker that wishes to imitate it, we have though to employ an existing satellite channel model. That is, the *Channel Model for Land Mobile Satellite Navigation* of [17], developed at the German Aerospace Center (DLR). In 2002 they have performed a campaign of measures employing a Zeppelin working nearby the L1 carrier, in order to simulate a GNSS satellite. The signal was received by a moving van, and upon the recorded data a channel model has been derived. The model is based on both deterministic and stochastic processes within

an artificial scenery, that has to be parametrised depending on the user necessaries. Basically, the direct path, or Line-Of-Sight (LOS) is influenced by three types of obstacles: house fronts, trees and lampposts of a specific scenery. While as regards the reflected signals, or echoes, they are assumed to be statistically distributed in the x, y, z space. The DLR MATLAB script generates the scenery in accordance to two different *surrounding* options - Urban and Suburban, and *user-type*, Pedestrian or Car. Because of our purposes, herein we have exploited the LandMobileMultipathChannel_Demo_UrbanPedestrian.m from DLR. Furthermore, as initial step, we have maintained the receiver antenna fixed. In other words, we have set $v = 0$, and the heading was kept constant as well, e.g. $0°$ (in this way there were nor any turning around itself). In order to have outcomes as realistic as possible, we have used satellites real positions (i.e. Elevation and Azimuth angles) observed in Padua during the day. This data have been sampled with $T_{samp} = 52\,s$, since in this time interval the angle changes are negligible.
Therefore, via this model we have generated an artificial urban scenery (with random obstacles positions and dimensions), and recorded the corresponding channel impulse response g_{AB}. This latter, is no more a delta function, but rather has one or more LOS rays, and several multipath echoes. Practically, once the artificial scenery is created, each method generate invocation outputs a complex time-variant channel impulse:

$$g_{A,B}(t,\tau) = \sum_{k=1}^{N(t)} a_k(t)\delta(\tau - \tau_k(t))$$

The observed number, amplitudes, and delay of channel coefficient depends statistically or deterministically on the current scenery. The LOS ray may be unique, high and with delay zero if the line of sight is free of obstacles. Or it will have up to three, small components with different amplitudes and delays, if a build is diffracting it. And finally the LOS can be completely blocked if, for instance, receiver and satellite are not aligned and not negligible obstacles are in the middle. But during the 52 s (and up to a considerable satellite movement) the LOS is constant, or rather deterministic. On the other hand, the number of multipath rays, their delays and amplitudes depend on the current obstacles, and satellite positions, but also on the reflectors distance to the receiver. However, as a consequence of our setting (i.e. a fixed receiver), the delays are constant as long as the satellite view does not change considerably. In fact, the DLR has assumed that once shaped and placed, all obstacles have a deterministic behaviour, except for treetops whose behaviour is modelled by a statistical process. Namely, the reflection points (e.g. a house wall) move according to the movement of the receiver, while others remain at fixed positions. Therefore, between consecutive time instants, the multipath echoes are changing only in amplitude and phase because of the stochastic motion of tree leaves.

We are interested in investigating the statistical behaviour of channel coefficients with respect to time. In order to derive significant statistics we have simulated 100 distinct scenarios (ω_i), always keeping the receiver antenna position fixed. Furthermore, 600 different satellite positions (i.e. Elevation and Azimuth angles) have been considered, which corresponds to about a 7 hours observation window. A sampling frequency of 1 Hz has been adopted to estimate the channel. That is, we have one channel impulse response per second.

One might expect that, the channel impulse response observed seven hours later will be completely un-correlated with the corresponding channel realization observed at $t = 0$.
The precise outcomes of the channel behaviour should help us in deciding how often we need to estimate the channel impulse response, and consequently, to defend ourselves against a channel imitation. Then, we have decided to evaluate the Kullback-Leibler divergence between two different distributions:

1. <u>the authentic joint distribution:</u> $p(\underline{h}(t_0), \underline{h}(t))$. That is what exactly the user observes, from his position, at the two distinct time instants, t_0 and t.
2. <u>the false distribution:</u> $p(\underline{h}(t_0)p(\underline{h}(t)$. Obtained as if $\underline{h}(t_0)$ and $\underline{h}(t)$ were independent, that is if $\underline{h}(t)$ were simulated by the attacker without prior observation of $\underline{h}(t_0)$. In fact, for the attacker it is hard to know what is the channel impulse response realization at time t, since a physical access to the antenna is excluded.

Then, we can expect an high divergence if the two estimates of the authentic channel are temporally highly correlated, while between the authentic and forged impulse response there is a low spatial correlation. That is, the attacker is not close to the receiver antenna (e.g. some kilometres apart), thus he can only compute an average estimate of the receiver actual scenario.

Basically, we are going to deal with complex Gaussian vectors. The corresponding a posteriori KL divergence (i.e. it is computed only once the key has been delivered, and consequently the channel estimation has become possible) is expressed as follows:

$$D_{KL}(H_0\|H_1) = \frac{\mathrm{tr}(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_0) + (\mathbf{m}_1 - \mathbf{m}_0)^T\mathbf{\Sigma}_1^{-1}(\mathbf{m}_1 - \mathbf{m}_0) - K + \ln\Big(\frac{\det(\mathbf{\Sigma}_1)}{\det(\mathbf{\Sigma}_0)}\Big)}{\ln(2)} \; [bits]$$

$$D_{KL}(H_1\|H_0) = \frac{\mathrm{tr}(\mathbf{\Sigma}_0^{-1}\mathbf{\Sigma}_1) + (\mathbf{m}_1 - \mathbf{m}_0)^T\mathbf{\Sigma}_0^{-1}(\mathbf{m}_1 - \mathbf{m}_0) - K + \ln\Big(\frac{\det \mathbf{\Sigma}_0}{\det \mathbf{\Sigma}_1}\Big)}{\ln(2)} \; [bits]$$

Where $\mathbf{m}_0$ and $\mathbf{m}_1$ are the corresponding mean vectors, which are identical, K is the vector space size, and $\mathbf{\Sigma}_0$, $\mathbf{\Sigma}_1$ are the covariance matrices. Which are distinct, and on them is based the divergence. The legitimate receiver, who perfectly knows its actual scenario, computes the covariance matrix $\mathbf{\Sigma_0}$ on the such defined matrix:

$$\begin{bmatrix} h_1(\omega_1, t_0) & \dots & h_N(\omega_1, t_0) & h_1(\omega_1, t) & \dots & h_N(\omega_1, t) \\ h_1(\omega_2, t_0) & \dots & \dots & \dots & \dots & h_N(\omega_2, t) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1(\omega_{100}, t_0) & \dots & \dots & \dots & \dots & h_N(\omega_{100}, t) \end{bmatrix}$$

The corresponding $\mathbf{\Sigma_0}$ is $2N \times 2N$ matrix partitioned in block as:

$$\mathbf{\Sigma}_0 = \left[\begin{array}{c|c} A & B \\ \hline B^T & D \end{array}\right], \qquad B = \begin{bmatrix} k_{1,1} & k_{1,2} & \dots & k_{1,N} \\ k_{2,1} & k_{2,2} & \dots & k_{2,N} \\ \vdots & \ddots & \vdots & \vdots \\ k_{N,1} & k_{N,2} & \dots & k_{N,N} \end{bmatrix} \tag{6.11}$$

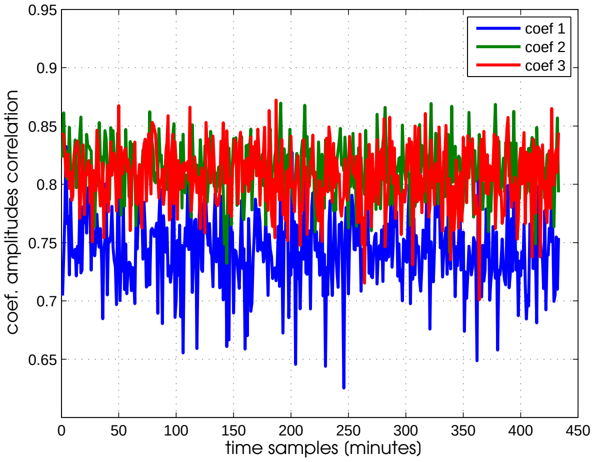


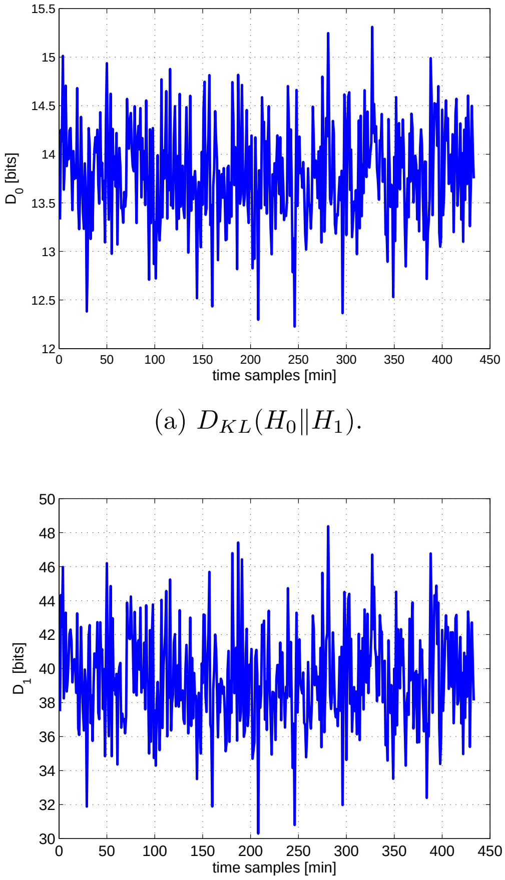
Figure 6.9: The correlation between the first three channel coefficients amplitudes, versus time.

while

$$\boldsymbol{\Sigma}_1 = \left[\begin{array}{c|c} A & 0 \\ \hline 0 & D \end{array}\right]$$

The diagonal elements in block B, of matrix $\boldsymbol{\Sigma_0}$, are particularly significant for us. These are the covariances between corresponding channel coefficients, at different time instants. It is reasonable to have high covariances for the earlier time instant, but as t increases, the generic $k_{i,i}$ should approach zero. Namely, the two channels become less correlated, and approach independence since the vectors are Gaussian. When they are approximately independent, we have $\boldsymbol{\Sigma}_0 \simeq \boldsymbol{\Sigma}_1$. That is, the receiver is no longer able to discriminate the estimated channel as authentic or forged, because there is no correlation with the reference one. This analysis applied to the DLR model output, gives the correlation and divergence respectively reported in Figure 6.9 and 6.10. Descends that the coefficients are highly correlated over the entire seven hours period, or in other words $\boldsymbol{\Sigma}_0$ never approaches $\boldsymbol{\Sigma}_1$. Consequently, the KL divergence does not vanish as expected, even after a long time.

The realism of these outcomes, with $v = 0$ is questionable correct, even if the DLR model is universally recognized as trusted. Hence, by the model outcomes, appears that distinguishing between authentic and forged channels would always be possible even many hours after the initial secure estimate.



(a) $D_{KL}(H_0 \| H_1)$.

(b) $D_{KL}(H_1 \| H_0)$..

Figure 6.10: The KL divergence between the two distributions as a function of time.

Chapter 7

Conclusions

For sure, the GNSS signal Authentication and Integrity Protection is a critical issue. Its particular requirements demand for a reassessment of authentication schemes in use for other wireless communication systems. In addition, we should work at both Data and Signal layers. Some proposals already exist, but these are only computationally secure, or not efficiently designed. Or even, they are focused on a precise form of attack, and are neglecting different attacker strategies, or equipment.

This thesis has fitted into this complex scenario, and has attempted to offer a useful contribution without presuming to identify the better authentication scheme. It has searched for two universal lower-bound of the key length. The problem has been analysed from a Physical-Layer point of view. Namely, the channel features like its capacity or the additive noise have been largely employed. Further, the attacker capabilities have not been limited, both in computational, and channel terms.

Therefore, in agreement with the *unconditional* security definition, the allowed attacker success probability has been ϵ upper-bounded, e.g. forging, modification, false alarm and missed detection. In addition, we have defined the key size as a function of the attacker action. For instance, the key increases with an eventually higher antenna gain for the Data case, or according to a decreasing noise power for the Signal-layer one. In other words, we are offering a key bound, which everyone can adapt to his own channel parameters, and regardless how cunning the attacker

is, the key will guarantee the desired ϵ-unconditionally secure authentication and integrity protection.
As regards the practical feasibility, the only thing to be proved is the introduction of the E1 signal new component. If possible, it will give us an additional degree of freedom in authentication design. However, it will require a deeper study about how to combine signal-authentication requirements with Galileo technical specifics.
An other future work, will be to analyse the two bounds behaviours versus a more complex channel (i.e. with motion and multipath effects). The only expected problem regards the channel estimation, or rather its coherence time interval.
A last aspect that we have not treated, is the key management. In both cases, respectively Data and Signal authentication, we have supposed to adopt a TESLA approach. Namely, at the authentication time instant $\mathbf{k}$ is private. And it becomes public as soon as every one has received the signal. The important aspect is that $\mathbf{k}$ will never be available in time for a potential attack. We have mentioned about a private, authenticated, noiseless channel. This may be either the satellite channel with error correction, either a network link. In both cases this channel will require an additional authenticating key, and a renewal process of it. Notice that this is not a secondary issue, indeed if an attacker will be able to break this key, he will acquire complete control of the authentication mechanism. And also in this case we should employ the Information Theory aid, and to look for an unconditionally secure key management.

In conclusion, although the attacker has a wide set of possibility to attack the GNSS signal, the achievement of this thesis highlight that there is the possibility for message and signal unconditionally secure authentication and integrity protection. For sure, in some cases the channel properties, or the particularly strong attacker, will require a corresponding non-practical key size. And we should admit that in some circumstances the GNSS signal authentication and integrity protection become hard, or even impossible. However, we must ensure signal authentication first to higher priority situations, such as to an aircraft/ship route monitoring, or to financial trades. This kind of scenarios are expected to be easier, because of a quasi line-of-sight propagation, or due to the receiver antenna staticity.

Bibliography

[1] *European GNSS (Galileo)- Open Service - Signal In Space Interface Control Document*, OD SIS ICD, Issue 1.1, September 2010.

[2] E. D. Kaplan, C. J. Hegarty *Understanding GPS, Principles and Applications*, Second edition, Artech House, 2005.

[3] G. Manoj Someswar, T. P. Surya Chandra Rao, Dhanunjaya Rao Chigurukota *"Global Navigation Satellite Systems and Their Applications"*, International Journal of Software and Web Sciences (IJSWS), pp. 17-23, February 2013.

[4] A. Jafarnia Jahromi *GNSS Signal Authenticity Verification in the Presence of Structural Interference*, PhD thesis, Department of geomatics engineering at Calgary University, Alberta,Canada, September 2013.

[5] G. T. Becker *Security mechanisms for positioning systems - enhancing the security of eLoran*, Master thesis, Ruhr-Universität Bochum, June 2009

[6] J. A. Ávila Rodríguez *On Generalized Signal Waveforms for Satellite Navigation*, Phd thesis, University FAF Munich, June 2008

[7] *Algorithms, key size and parameters report*, ENISA, November 2014.

[8] M. Petovello *" GNSS solutions: Clock offsets in GNSS receivers"*InsideGNSS, pp. 23-25, March/April 2011.

[9] M.G. Kuhn *"An asymmetric security mechanism for navigation signals"* International Workshop on Information Hiding, IH, pp.239-252, 2005

[10] W. Stallings *Cryptography and Network Security - Principles and Practice* Pearson, sixth edition, 2014.

[11] G.W. Hein, F. Kneissl, J.A. Avila-Rodriguez, S. Wallner *"Authenticating GNSS: proofs against spoofs - Part 1"*, Inside GNSS, pp. 58-63, July/August 2007

[12] G.W. Hein, F. Kneissl, J.A. Avila-Rodriguez, S. Wallner *"Authenticating GNSS: proofs against spoofs - Part 2"*, Inside GNSS, pp. 71-77, September/October 2007

[13] O.Pozzobon *" Keeping the spoofs out, Signal authentication services for future GNSS "*, InsideGNSS, pp. 48-55, May/June 2011

[14] K.D. Wesson, B.L. Evans, T.E.Humphreys *" A Probabilistic Framework for Global Navigation Satellite System Signal Timing Assurance "*, IEEE, Asilomar Conference on Signals, Systems and Computers 2013, pp. 846-850

[15] T.E.Humphreys *" Detection strategy for cryptographic GNSS anti-spoofing "*, IEEE Transactions on aerospace and electronic systems, Vol. 49, No. 2, April 2013, pp. 1073-1090

[16] G. Caparra, N.Laurenti,I. R. T. Ioannides, M.Crisci *"Improved Secure Code Estimation and Replay Attack and Detection on GNSS Signals"*, ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, NAVITEC, 2014

[17] A. Lehner, A. Steingass *"A Novel Channel Model for Land Mobile Satellite Navigation"*, ION GNSS, pp. 2132-2138, 2005.

[18] A. Perrig, R. Canetti, J. D. Tygar, D. Song *"The TESLA Broadcast Authentication Protocol"*, CryptoBytes, pp. 2-13, Summer/Fall 2002.

[19] J.T. Curran, M.Paonni *"Securing GNSS: An End-to-End Feasibility Study for the Galileo Open Service"*, International Technical Meeting of the Satellite Division of The Institute of Navigation, ION GNSS, pp. 1-15, 2014.

[20] I. Fernández-Hernández, V. Rijmen, G.Seco-Granados, J. Simón, I. Rodríguez, J.D. Calle *"Design Drivers, Solutions and Robustness Assessment of Navigation Message Authentication for the Galileo Open Service"*, International Technical Meeting of The Satellite Division of the Institute of Navigation, ION GNSS, pp. 2810-2827, 2014.

[21] Y.Liu, J.Li, M.Guizani *"PKC Based Broadcast Authentication using Signature Amortization for WSNs"*, IEEE Transactions on Wireless Communications, vol. 11, pp. 2106-2115, June 2012.

[22] S. Lo, D. De Lorenzo, P.Enge, D. Akos, P. Bradley *"Signal Authentication - A secure Civil GNSS for Today"* , Inside GNSS, pp. 30-39, September/October 2009.

[23] O.Pozzobon, C.Sarto, A.Pozzobon, D.Dötterböck, B.Eissfeller, E.Pérez, D.Abia *" Open GNSS signal authentication based on the Galileo Commercial Service (CS)"*, International Technical Meeting of The Satellite Division of the Institute of Navigation, ION GNSS+, pp. 1-10, 20013.

[24] U.M. Maurer *"Authentication Theory and Hypothesis Testing"*, IEEE Transactions on Information Theory, vol. 46, No. 4 pp. 1350-1356, June 2000

[25] J. Cederlöf, J.A. Larsson *"Security Aspects of the Authentication Used in Quantum Cryptography"*, IEEE Transactions on Information Theory, vol. 54, No. 4 pp. 1735-1741, April 2008

[26] D.Chen, S. Jiang, Z.Qin *"Message Authentication Code over a Wiretap Channel"*, May 2015.

[27] L. Lai, H. El Gamal, H.V. Poor *"Authentication Over Noisy Channels"* IEEE Transactions on Information Theory, pp. 906-916, Vol. 55, N0. 2, 2009

[28] Y.Liu, P.Ning *"Enhanced Wireless Channel Authentication Using Time-Synched Link Signature"*, IEEE, International Conference on Computer Communications, pp. 2636 - 2640, March 2012

[29] P.Baracca, N. Laurenti, S. Tomasin *”Physical Layer Authentication over MIMO Fading Wiretap Channels“*, IEEE Transactions on Wireless Communications, Vol. 11, No. 7, pp. 2564-2573, 2012.