Università di Padova

SCUOLA DI INGEGNERIA
Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni

AUTHENTICATION OF GNSS SIGNAL BY INFORMATION-THEORETIC SECURITY

Relatore:
Stefano Tomasin

Laureando:
Francesco Formaggio

11 settembre 2017, Anno Accademico 2016/17

Contents

1 Introduction **5**

2 Positioning attacks **7**

2.1 Galielo System Architecture . . . 7

2.1.1 Galileo signal . . . 8

2.1.2 Receiver structure . . . 9

2.1.3 Determining user position . . . 11

2.2 Spoofing . . . 13

2.3 Navigation message authentication . . . 17

2.3.1 One-way key chain protocols . . . 17

2.4 Prediction based attacks . . . 19

2.4.1 Security code estimation and reply attack . . . 19

2.4.2 Forward error estimation attack . . . 22

2.4.3 State modelling attack . . . 23

3 Information-theoretic authentication protocol **25**

3.1 System Model . . . 25

3.1.1 Galielo signaling . . . 25

3.1.2 Channel model . . . 26

3.1.3 Attack model . . . 27

3.2 Authentication protocol . . . 28

3.2.1 Protocol analysis . . . 30

3.2.2 Protocol requirements . . . 31

3.2.3 Orthogonality . . . 32

3.3 Protocol performance analysis . . . 34

3.3.1 Secrecy and reliability requirements . . . 34

3.3.2 Synchronization . . . 35

3.3.3 Constellation constrained system . . . 36

4 Numerical results **39**

4.1 Secrecy capacity and delay attack . . . 39

4.2 M-PSK modulation . . . 41

4.2.1 Effects of quantization . . . 41

5 Conclusions and future work **49**

Appendices **51**

A Wiretap Channel **53**
A.1 Secrecy Capacity . . . 54
A.1.1 Gaussian wiretap channel . . . 55
A.2 Randomness for secrecy . . . 55

B Quantization **57**
B.1 Description of the quantization process . . . 57
B.2 Uniform quantizer . . . 58
B.3 Non-uniforrm quantizers . . . 58

References **61**

Chapter 1

Introduction

Global Navigation Satellite System (GNSS) refers to a constellation of satellites continuously broadcasting signals used by GNSS receivers to compute their position on Earth. Existing GNSS systems include European Galielo, USA's NAVSTAR Global Positioning System (GPS) and Russia's Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS). GNSS has become crucial for many applications [1] besides general user navigation applications such as vehicles, airplanes and ships. Agriculture makes extensive use of positioning services: farm planning, filed mapping, tractor guidance and crop scouting. Environment-related applications help decision makers providing useful data about items that are spread across many kilometers of terrain. Aerial studies of some of the world's most impenetrable wilderness are conducted with the aid of GPS technology to evaluate an area's wildlife, terrain, and human infrastructure. Rail systems uses positioning informations to constantly track trains positions. Combined with other sensor networks, GNSS helps reduce accidents, delays, and operating costs. In addition to geographic positioning GNSS provides accurate timing informations. Each GPS satellite is equipped with atomic clocks that enable users to determine the time to within 100 billionths of a second, without the cost of owning and operating atomic clock. Each Galileo satellite is equipped with a passive hydrogen maser clock [2], an atomic clock that forces atoms to jump from one particular energy state to another. Jumps between energy states happen at extremely stable frequency and are then exploited to measure time. Precise timing is also used in a variety of applications such as many economic activities around the world. Timestamps are combined with financial transactions to maintain records and traceability.

Due to the spreading of such applications and the strategic nature of some of them, interest has risen in the possible threats GNSS may face. GPS spoofing devices have been built [12] and attacks have been demonstrated such as a yacht navigation system being under control of a group of researchers [3]. On 22 June 2017 at least 20 ships off the Russian port of Novorossiysk reported the same strange behaviour of their GNSS equipment [4], suggesting that Russia may be testing a new system for spoofing GPS.

Defences against spoofing attacks are widely studied in the literature and in this thesis we propose a new authentication protocol for Galileo GNSS signal based on information-theoretic security concepts. In Chapter 2 we revise the literature and go into some detail of certain type of attacks and defence. In Chapter 3 we describe the proposed authentication protocol together with its design objectives and performance metrics. Chapter 4 is devoted to numerical results. In the Appendices we revise the well established theoretical results that were useful in the design of the protocol.

Chapter 2

Positioning attacks

In this Chapter we first describe the Galileo system architecture and the modulations used in its signaling. Then we revise the literature about spoofing attacks and main defence techniques.

2.1 Galielo System Architecture

Galielo is the European global satellite-based navigation system and provides positioning and timing informations to all users worldwide. It has been designed to be interoperable and compatible with the other existing GNSS [5].

The Galileo system, once fully operational, will offer four high-performance services:

- *Open Service* (OS): a free of charge service accessible worldwide that offers time and positioning informations.
- *Commercial Service* (CS): a complementary service using different frequency bands to offer higher quality services. It can be encrypted in order to control access to it.
- *Public Regulated Service* (PRS): a service meant to be used only by governmental institutions. It provides high level service continuity for sensitive applications.
- *Search and Rescue Service* (SAR): this service involves locating and helping people in distress. It will be available at sea, in the mountains, across the desert and in the air inside the Galileo/SAR Service Coverage area. This essential Galileo service helps operators respond to a distress signal faster and more efficiently.

The history of the Galileo programme starts in February 1999 when its name first appears in official European Commission documents [6] and funds were allocated starting from the early 2000s. In December 2005 and April 2008 were launched the first two experimental satellites which started the In-Orbit Validation (IOV) phase. IOV consists in doing environment measurements (radiations, magnetic fields) and start the testing of communications equipment. The launch of the operational satellites began in October and at the time this thesis is being written 18 Galielo satellites are orbiting the Earth with working supporting ground stations. The Full

Operational Capacity (FOC) phase consists in the deployment of the remaining ground and space infrastructure: the full Galielo system will comprise a constellation of 30 satellites.

Together with the satellite constellation the system architecture also comprises the *Galielo Ground Segment* [8] consisting of two main control centers (Control Center Components GCC) and a global network of transmitting and receiving stations. The two control centers manage control functions supported by the dedicated Galileo Control System (GCS) and Galielo Mission System (GMS). GCS is responsible for the satellite constellation management and provides control functions for all Galileo satellites. It comprises a global network of telemetry tracking and control stations which communicate with GCC through a dedicated network infrastructure. GCS is responsible for the uplink of the navigation data built in the navigation signals. It uses a global network of Galileo Sensor Stations (GSS) in order to compute orbit informations and clock offsets for each satellite. These informations are then uploaded to the satellites periodically every 100 minutes.

2.1.1 Galileo signal

The GNSS satellites continuously transmit navigation signals and the receiver basic observable is the travelling time of the navigation signal to propagate from the satellite antenna to the receiver antenna. This value is then multiplied by the speed of light to obtain a measure of the distance between them. Then the user position is computed using distance measurements coming from at least four satellites, as we describe later.

The signal broadcast by the satellites has two main components [8].

1. *Ranging codes*: a sequence of 0s and 1s which is crucial for the receiver to determine the time distance from the satellite. They are also called Pseudo-Random Noise (PRN) sequences due to their correlation properties.

2. *Navigation data*: a binary encoded message carrying informations on satellites position, ephemeris, clock correction informations and other complementary parameters. The user cannot determine his position by looking only at the ranging codes, he needs also the navigation data.

PRN codes are such that the autocorrelation function has a peak only at zero lag. This is the reason for the name Pseudo-Random Noise: the autocorrelation function resembles the one of an uncorrelated noise signal, but PRN codes are actually deterministic sequences (Pseudo-Random). In other words: when a PRN sequence is correlated with an aligned replica of itself the correlation is maximum; when the PRN sequence is correlated with a non aligned replica of itself or with a different PRN sequence, then the correlation is low.

Modulation

Galileo system employs spread spectrum techniques to modulate its signals [34]. We concentrate on the signal transmitted in the E1 band, reserved for civilian use.

The signal comprises two channels, the data channel and the pilot channel, superimposed and distinguishable thanks to different PRN sequences. The data channel is

$$p(t) = \sum_i d_\ell c_i \operatorname{rect}\Big(\frac{t - 1/2 - iT_c}{T_c}\Big) \qquad \ell = \Big\lfloor \frac{i}{N_c} \Big\rfloor, \tag{2.1}$$

where d_ℓ is the binary data stream, c_i are the spread spectrum coefficients (PRN symbols), N_c is the number of PRN symbols (also called *chips*) forming a data symbol and T_c is the chip duration. If we define the spread spectrum symbol pulse as

$$s_s(t) \triangleq \sum_{i=0}^{N_c-1} c_i \operatorname{rect}\Big(\frac{t - 1/2 - iT_c}{T_c}\Big) \tag{2.2}$$

then $p(t)$ can be rewritten as

$$p(t) = \sum_i d_i s_s(t - iT_s), \tag{2.3}$$

where T_s is the symbol period. The pilot channel is similar to $p(t)$, but without the data component, i.e.,

$$e_{E1_p}(t) = \sum_i c_{p,i} \operatorname{rect}\Big(\frac{t - iT_c}{T_c}\Big). \tag{2.4}$$

Each channel is then modulated by a linear combination of two square-wave subcarriers

$$s_d(t) = p(t)\big[\alpha \operatorname{sgn}(\sin(2\pi f_1 t)) + \beta \operatorname{sgn}(\sin(2\pi f_2 t))\big] \tag{2.5}$$

$$s_e(t) = e_{E1_p}(t)\big[\alpha \operatorname{sgn}(\sin(2\pi f_1 t)) - \beta \operatorname{sgn}(\sin(2\pi f_2 t))\big] \tag{2.6}$$

with $\alpha = \sqrt{\frac{10}{11}}$, $\beta = \sqrt{\frac{1}{11}}$, $f_1 = 1023$ MHz, $f_2 = 6138$ MHz. Then the two signals are added and power normalized. The baseband signal is

$$s_{E1}(t) = \frac{1}{\sqrt{2}}(s_d(t) - s_e(t)). \tag{2.7}$$

Fig 2.1 shows a chip period of the superimposed two subcarriers, both for the data and pilot signals.

2.1.2 Receiver structure

Although the precise architecture of a GNSS receiver varies substantially according to different manufacturers and implementations, the basic building blocks remain the same. We can identify four different components.

1. *Antennas*: they are the main interface between the GNSS space segment and the user segment and are designated to capture signals within the E1 band (in case of civilian applications) together with noise and possible interference.

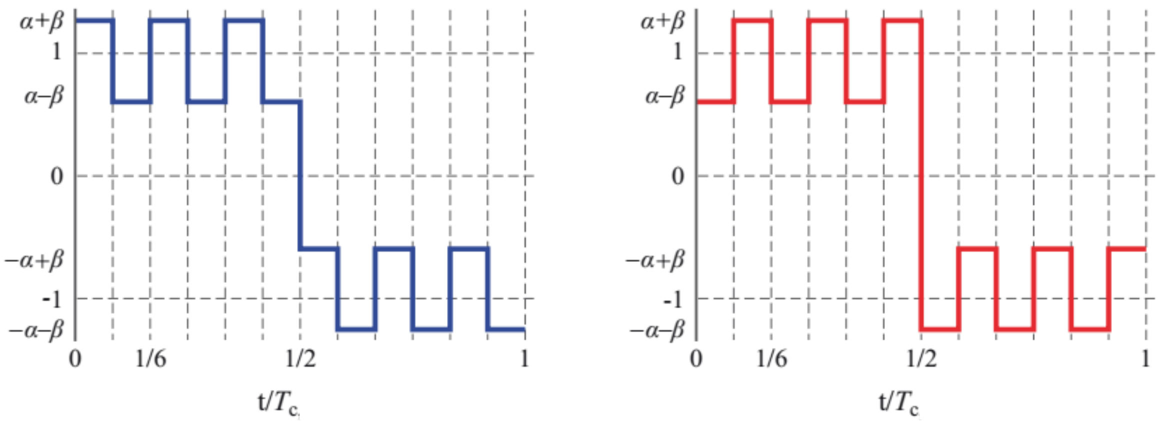


Figure 2.1: One chip period of the subcarriers of equations (2.5) (left) and (2.6) (right). Figure from [34].

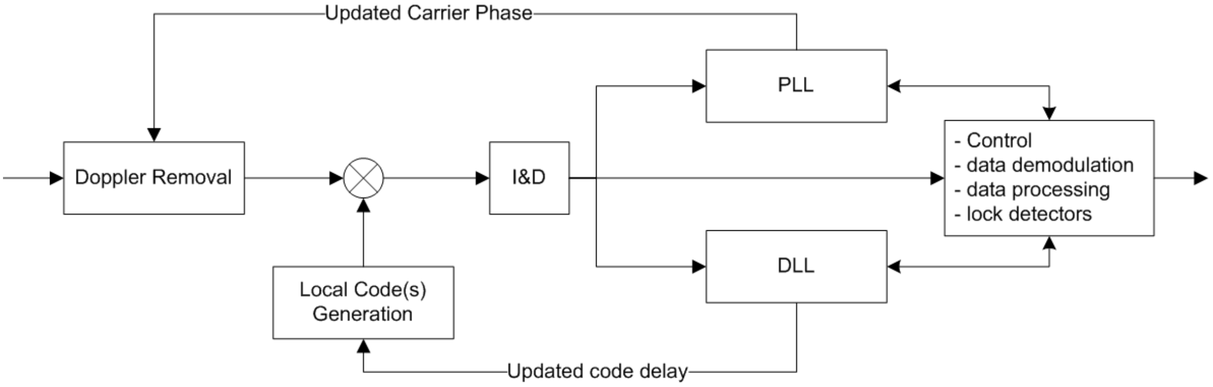


Figure 2.2: GNSS baseband signal processing block diagram. Figure from [8].

2. *Front-end*: it is the section responsible for preparing the received signals for later signal processing. Its main tasks are filtering and amplifying signals within useful frequency bands, down-conversion to baseband, and quantization so to digitize information.

3. *Baseband signal processing*: a set of signal processing algorithms used to acquire and track the different signals.

4. *Applications processing*: processing of the low level data in order to fulfil the needs of specific applications, e.g. position, velocity and timing (PVT).

We now describe the baseband signal processing function, whose block diagram is shown in Fig. 2.2.

Baseband signal processing is a dynamic system because signals and satellites in line of sight vary greatly in time and the receiver needs to continuously search and track these changes. As soon as the received signal enters the baseband processing block it is stripped of its Doppler frequency according to the current carrier phase estimate. The local code generator generates a local replica of the PRN code used by the satellite being tracked. The local replica is time shifted matching the code delay of the received signal in order to have high correlation peak. The output of the correlation is always noisy and needs further processing. This is the task of the Integration and Dump (I & D) block, which integrates over time the correlator output in order to increase the power of any potential peak and to reduce noise. The receivers updates signals parameters thanks to tracking loops, which elaborate correlator outputs and refine current estimates. The Delay Lock Loop (DLL) tracks the code delay while the Phase Lock Loop (PLL) follows the phase of the incoming signal.

2.1.3 Determining user position

As described in the previous section, the receiver generates a local replica of the ranging code and continuously correlates it with the signal coming from the satellite [7]. The correlation is computed at different lag values until a peak is found. That lag is the propagation time of the signal, and we call it Δ_t. The true geometric range, i.e. the geometric distance form satellite to user, is then

$$r = c\Delta_t, \tag{2.8}$$

where c is the speed of light. By introducing vector notation, as shown in Fig. 2.3, r is the magnitude of vector $\mathbf{r}$ connecting the user to the satellite. $\mathbf{u}$ is the vector representing user position on Earth and its three components (u_x, u_y, u_z) are the solutions of the positioning problem. The best suited coordinate system for this computation is the Earth-Centred, Earth-Fixed (ECEF) coordinate system. $\mathbf{s}$ is the vector connecting Earth center to the satellites and it is a known quantity because its part of the navigation message sent by the satellite. The relation between vectors is

$$r = ||\mathbf{r}|| = ||\mathbf{s} - \mathbf{u}||, \tag{2.9}$$

where $||\cdot||$ denotes the norm of a vector.

The geometric range however is not measured exactly by the receiver, due to clock synchronization errors. Specifically the satellite makes an error, with respect to the system time, when

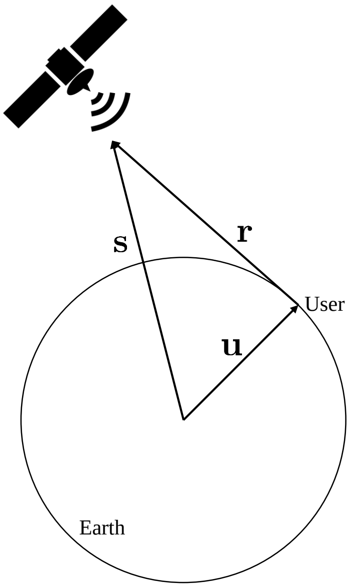


Figure 2.3: User position vector notation.

measuring the time of the departure of the signal and the user makes an error when measuring the time of arrival of the signal. Clock offsets introduced by the satellites are readily corrected by the GMS component of the ground segment and thus can be neglected by the user. User's time offset t_u instead is an unknown quantity that needs to be estimated because it affects range measurements. We call *pseudorange* ρ the actual distance computed by the receiver and it is given by

$$\rho = c(\Delta_t + t_u) = r + ct_u \tag{2.10}$$

and using (2.9) we can write

$$\rho = ||\mathbf{s} - \mathbf{u}|| + ct_u. \tag{2.11}$$

The total number of unknowns is four: u_x, u_y, u_z and t_u. To compute the solution we need at least four range measurements from four different satellites. We index satellites with variable j going from 1 to 4. We can then write the system of equations

$$\rho_j = \sqrt{(x_j - u_x)^2 + (y_j - u_y)^2 + (z_j - u_z)^2} + ct_u \quad \text{for } j = 1, \ldots, 4. \tag{2.12}$$

This is a non linear system which can be resolved in different ways according to the specific receiver implementation. Three possible options are: closed form solution [9]; iterative algorithm based on linearization; Kalman filtering, which exploits past series of time measurements to obtain a more accurate position estimate.

2.2 Spoofing

Spoofing of GNSS signals is a set of techniques that a malicious user intentionally uses in order to induce false ranging measurements to a legitimate user. The existence of such techniques was particularly pointed out starting from 2001 in the so called Volpe report [14] were the authors warned that "as GPS further penetrates into the civil infrastructure, it becomes a tempting target that could be exploited by individuals, groups or countries hostile to the US".

One of the simplest spoofing techniques is *meaconing*, i.e., the interception and rebroadcast of the navigation signals so that the victim computes the ranging information based on the spoofer location, and not on the satellites'. More sophisticated versions of this attack selectively forge delayed versions of the ranging signals so that the spoofer can freely decide the false position detected by the victim. Several detection techniques have been proposed [15] [16] that work on the differences between the true satellite signal and the spoofed one. One of these techniques [17] exploits receiver's automatic gain control to detect a sudden increase of received power due to the beginning of a spoofing attack.

Using multiple antennas [18] at the the receiver can increase the detection performances thanks to the determination of the angle of arrival of the received signals. This forces the spoofer to not only compute selective delays for each transmitted signal, but also to pay attention to the geometry of the signal directions. In [19] and [20] multi antenna techniques are proposed using commercial off-the-shelf (COTS) receivers.

Spoofing attack general model

As suggested in [10] we can model a generic GNSS legitimate and malicious signal in the following way.

We call $y(t)$ a generic GNSS legitimate received signal

$$y(t) = \Re\left\{\sum_{i=1}^{N} A_i D_i[t-\tau_i(t)] C_i[t-\tau_i(t)] e^{j[\omega_c t - \phi_i(t)]}\right\}, \tag{2.13}$$

where the parameters are listed below:

- N is the number of signals arriving at the receiver, each coming from a different satellite. As we saw in section 2.1.3 the receiver needs at least four satellites to perform PVT calculations, but usually more than four satellites are visible from the receiver antenna;

- A_i is the amplitude of each signal;

- $D_i(t)$ is the data signal carrying navigation data: in Galileo signals this correspond to d_ℓ of equation (2.1);

- $C_i(t)$ is the ranging code use by the ith signal and $\tau_i(t)$ is its current code delay;

- ω_c and $\phi_i(t)$ are respectively the carrier frequency and the carrier phase.

The malicious entity, the spoofer, who wants to deceive the legitimate user, the defender, has to send a set of signals similar to the legitimate ones:

$$y_s(t) = \Re\left\{\sum_{i=1}^{N_s} A_{si}\hat{D}_i[t-\tau_{si}(t)]C_i[t-\tau_{si}(t)]e^{j[\omega_c t-\phi_{si}(t)]}\right\}. \tag{2.14}$$

Similarly to the legitimate signal we have the following parameters:

- N_s is the number of satellite signals being spoofed, usually $N_s = N$;
- A_{si} and $\phi_{si}(t)$ are the amplitude and the phase of the spoofing signal respectively; they can be tuned in order to perform specific attacks;
- The attacker who wishes to deceive the defender must use the same ranging signals $C_i(t)$; this is possible because spreading sequences are part of the specific GNSS standard like in Galileo [34]; spoofer's code delay $\tau_{si}(t)$ are the key elements to induce false ranging measurements to the defender; during an ongoing attack they will be in general different from the legitimate ones: $\tau_{si} \neq \tau_i$;
- $\hat{D}_i(t)$ is the best estimate of the data signal available to the spoofer: the data signal may be predictable or encrypted depending on the defence strategies possibly implemented.

When the spoofer is performing the attack, the signal received by the defender is

$$y_{tot} = y(t) + y_s(t) + \nu(t) \tag{2.15}$$

where $\nu(t)$ is the received noise.

Meaconing

The simplest spoofing attack is *meaconing.* A spoofer employing meaconing receives the signal coming from satellites and replays it immediately to the victim. In this case $A_{si} > A_i$ so to overwhelm the legitimate signal. The position computed by the victim will be the spoofer's antenna position. This approach, although simple to implement, does not permit the attacker to induce arbitrary position computations to the victim. To do so the spoofer has to synthesize separately the delays τ_{si} that will be different from τ_i. τ_{si} should also be consistent with the Earth geometry, otherwise receivers employing receiver autonomous and integrity protection (RAIM) would easily detect spoofing thanks to consistency checks [11].

Whatever signal the spoofer decides to send, he must first induce the victim to lock with the false signal. To do so the attacker can use jamming. Jamming is the deliberate use of noise signals to interfere and block radio communications. In our case this results in the victim to re-start acquisition phase. Then, if spoofed amplitudes A_{si} are significantly grater than the true ones A_i, the attacked receiver will lock on the false signal with high probability.

A technique to induce the victim to lock on the false signal that does not require jamming is shown in Fig. 2.4 in successive snapshots, from top to bottom. The three red dots visible in each snapshot are the DLL lock points used to continuously maintain lock on the current

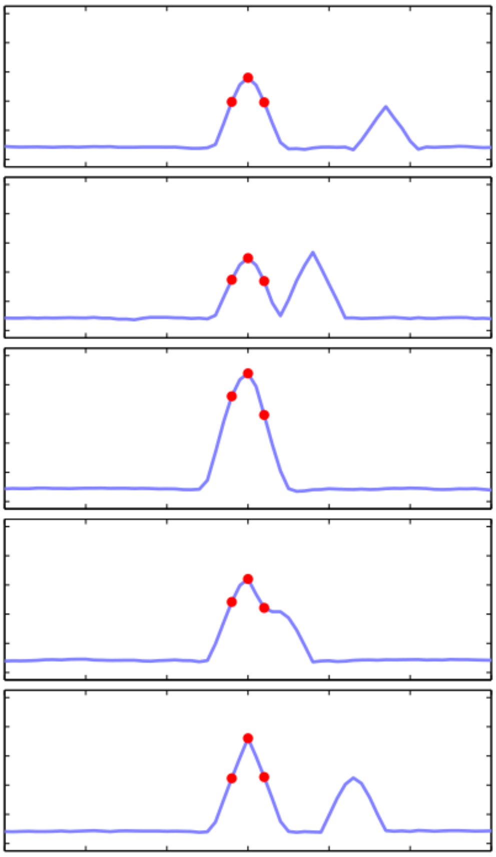


Figure 2.4: A sequence of frames describing the procedure through which the victim locks on the spoofing signal. Image taken from [12].

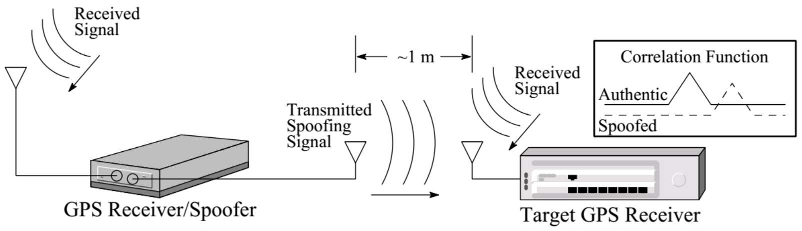


Figure 2.5: Illustration of a portable spoofer device attack. Image from [12].

ranging signal. In the first snapshot they are aligned with the correlation peak generated by the genuine satellite signal. The spoofer initially search for the right phase alignment with the genuine signal. Thus amplitudes A_{si} are low and correspond to the second correlation peak in the first two snapshots. Once the spoofing signal aligns with the legitimate one, the spoofer increases transmission power, such that $A_{si} > A_i$. Thanks to the power advantage the victim locks on the spoofing signal (third snapshot). Now the spoofer can drag-off victim's lock position (last two snapshots) so to finally induce false range measurements.

At the time of the Volpe report no known devices existed capable of carrying on such attacks. Things changed with [12] where the authors successfully developed a portable GPS civilian spoofer. One of the challenges of performing the attack in Fig. 2.4 is the fact that the spoofer has to know the true values A_i and has to gain accurate knowledge of the target receiver position and velocity. This is needed in order to build a properly synchronized counterfeit signal relative to the genuine satellite signals and to avoid detection. A portable GPS spoofer overcomes those issues by design. Such a device can be placed right next to the victim antenna so that PVT parameters are approximately the same, due to proximity.

Nulling

An alternative spoofing technique is nulling [10].The objective of nulling is to completely cancel out the legitimate signal from the total received signal y_{tot} of (2.15). The spoofer sends $N_s = 2N$ signals where half of these are the nulling signals. To be effective nulling signals must obey the following rules. Suppose, without loss of generality, that the nulling signals are the second half and have index $i + N$ for $i = 1, \cdots, N$, then:

$$C_{i+N}(t) = C_i(t) \tag{2.16a}$$

$$\hat{D}_{i+N(t)=D_i(t)} \tag{2.16b}$$

$$A_{s[i+N]} = A_i \tag{2.16c}$$

$$\tau_{s[i+N]}(t) = \tau_i(t) \tag{2.16d}$$

$$\phi_{s[i+N]}(t) = \phi_i(t) + \pi, \tag{2.16e}$$

where cancelling occurs thanks to the opposite phases in (2.16e). As we can imagine by looking at 2.16 nulling is difficoult because it requires very precise knowledge of the victim's legitimate received signal. Although achieving code and data synchronization is reasonably simple, exact carrier phase alignment and amplitude matching is more difficult.

Vestigial signal defence [16] exploits the aforementioned nulling difficulties in order to detect spoofing. Supposing the attacker is not able to completely cancel out the legitimate signal from the victim's receiver, vestiges of the true ranging signal are still present. To detect this signal the receiver needs to save, in a different buffer, the current output of the tracking correlators. Then he looks for the ranging signal being tracked at the moment and removes it from the buffer. Once the this has been done the receiver performs tracking in the buffer by correlation with the same PRN sequence. We recall that a would-be spoofer needs to use the same PRN sequence as the satellite signals he intends to spoof. The resulting correlation is compared with a threshold and if the correlation value exceeds the threshold, then we conclude that a spoofed signal was indeed present.

2.3 Navigation message authentication

One of the main defence strategies against spoofing attacks is relying on the cryptographic mechanism to introduce unpredictability in the navigation message so that building a counterfeit signal would be difficult. Encryption may be used at the chip level [26], or at the data level [22]-[25]. Navigation message authentication (NMA) generally refers to protocols that use encryption to provide authentication and integrity protection services to the user, i.e. he should be able to autonomously verify that the ranging signal are coming from a trusted source (satellites) and that they are not being modified by malicious entities [27].

All NMA approaches put in place trade-offs between performance and costs. From the performance point of view a good NMA technique should maximize robustness against attacks by using for example long cryptographic keys and secure key management functions. This however may result in communication overhead, especially in low-rate data channel such as Galileo OS dissemination channel (125bps), or extra computation and memory requirements for the receiver to handle.

Asymmetric cryptography such as digital signature (DS) is suited for GNSS applications [28] [29] because of simplicity and scalability in key management. DS schemes however introduce substantial overheads in terms of computational complexity and size of signatures. Symmetric cryptography on the other hand [24] [25] does not suffer from key distribution issues as users must share the same secret key as the system. However, secret keys are stored in the device to permit computation of cryptographic functions and this requires tamper-resistant hardware module.

2.3.1 One-way key chain protocols

Among all NMA schemes one-way key chain protocols [30] [31] are promising techniques because they exploit advantages of symmetric encryption and deal with key distribution in a delayed fashion, as we see next.

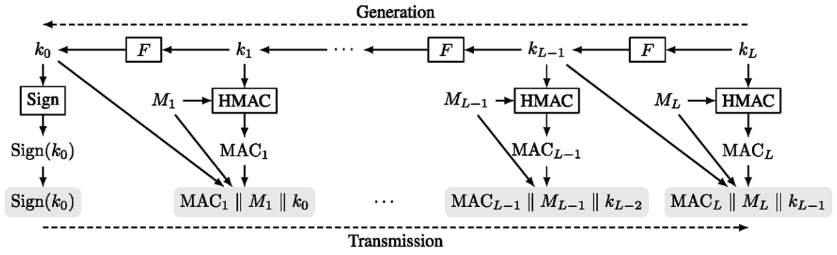


Figure 2.6: Tesla authentication protocol. Image from [27]

TESLA

Timed efficient stream loss-tolerant authentication (TESLA) [30] employs a key-disclosure paradigm. The transmitter authenticates the message he wants to send with a secret key, known only by him, yielding a message authentication code (MAC). The MAC is sent to the receiver along with the original message. After some time the key is disclosed and the receiver can verify the authenticity of the received message.

The protocol is shown in Fig. 2.6. The transmitter chooses the first key k_L randomly and recursively generates all other keys k_i for $i = L-1, \dots, 0$ by successively applying the one-way function $F(\cdot)$. We recall that a one-way function is by definition easy to compute, i.e. given the input, efficient algorithms compute the output, and at the same time it is hard to find an input that yields a given output. k_0 is the root key and has to be separately signed using DS and sent at the beginning. M_i with $i = 1, \dots, L$ are the messages to be sent. Each message M_i is signed with the corresponding key k_i yielding its MAC$_i = \mathbb{S}(M_i, k_i)$, where $\mathbb{S}(\cdot, \cdot)$ is the signing function. The transmitted packet however does not contain k_i, but a key k_j with a lower index $j < i$. Thanks to the one-way function characteristics the receiver gains no information about k_i and the only way he has to verify MAC$_i$ is to wait for the future disclosure of k_i itself. Instead, knowing k_j, the receiver can authenticate the previously received keys k_{j-d}, $d > 0$, by recursively applying $F(\cdot)$. The recursion stops as soon as the receiver reaches k_0 or he finds an already authenticated key. With the authenticated keys the receiver can verify MACs and hence accept or refuse the corresponding message.

Digital signal amortization

Digital signal amortization (SigAm) technique has been applied to GNSS in [31]. Like TESLA, SigAm uses a chained digest authentication scheme, but this time a chain authenticates only one (longer) message and each subsequent new message require a new digest chain.

Fig. 2.7 shows the authentication protocol based on SigAm. n is the chain index and $m = 1, \dots, M$ is the step number in the chain. Data message $D_i(n)$ depends also on n because, as we said, each message has its own chain. Chain construction starts from the computation of the digest $H_i(n, M) = h(n, M)$, where $h(\cdot, \cdot)$ is a one-way function and i indexes satellites, as each different satellite has its own navigation data. The rest of the digests are computed as

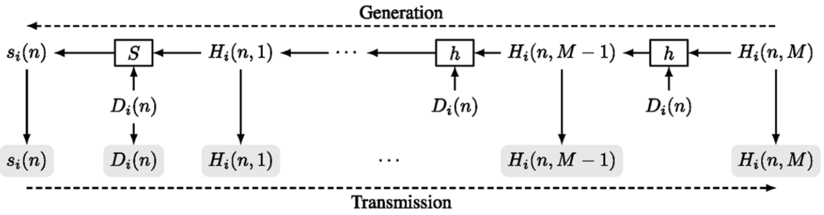


Figure 2.7: Authentication scheme based on SigAm. Image from [27]

follows:

$$H_i(n,m) = h(D_i(n), H_i(n,m+1)) \quad m = M-1,\ldots,1. \tag{2.17}$$

In other words each subsequent digest is computed from the previous one and *the same* data message $D_i(n)$. Differently from TESLA, here the navigation data is embedded in the computation of the chain. The last digest $H_i(n,1)$ is again signed using DS: $s_i(n) = \mathbb{S}(D_i(n), H_i(n,1))$. The message sent to the user then comprises $s_i(n)$, $D_i(n)$ and all digests $H_i(n,m)$ with $m = 1,\ldots,M$. The receiver applies (2.17) to the received digests and then verifies separately $s_i(n)$. If all the verifications succeed the message is accepted.

In case a malicious user breaks the chain, in TESLA he could then authenticate a forged message that will be accepted. This won't happen with SigAm because all digests are function of the same navigation data and a forged message would need a new chain.

2.4 Prediction based attacks

Spoofing relies on the construction of a counterfeit signal that must resemble the legitimate one. The attack is then even more effective if the spoofer is able to predict the legitimate signal in advance with respect to the user. Unpredictability introduced by NMA makes the construction of a consistent counterfeit signal more difficult. However the attacker has still got effective attacks against NMA secured signals [13] [22].

2.4.1 Security code estimation and reply attack

NMA introduces unpredictability in the data stream or at the chip level. A general model for a GNSS signal protected with some security code is the following [13]:

$$Y_k = w_k c_k \cos(2\pi f_{IF} t_k + \theta_k) + N_k. \tag{2.18}$$

Y_k are the samples exiting the radio frequency front end of a GNSS receiver and the sampling period is considered to be smaller than the chip period. $c_k \in \{+1,-1\}$ is the current chip symbol, f_{IF} is the carrier frequency, t_k is the receiver time and θ_k is the carrier phase. $w_k \in \{+1,-1\}$ is a security code with code period T_w and models unpredictability, be it at the chip level or data level (T_w will change accordingly). N_k models thermal noise and interference: it is a sequence of iid zero-mean Gaussian noise samples with variance σ_s^2

The attacker, by observing (2.18), attempts to predict the security code w_k and, as soon as he gets a reliable estimate, he immediately injects it in the signal replica generator so to perform a spoofing attack. This is called security code estimation and reply attack (SCER). The generated counterfeit signal is

$$\hat{y}_k = \hat{w}_k c(\tau_k) \cos(2\pi f_{IF} t_k + \theta_k) = \hat{w}_k s_k, \tag{2.19}$$

where $s_k \triangleq c(\tau_k)\cos(2\pi f_{IF} t_k + \theta_k)$ and we assume that the attacker can easily estimate code delay and carrier phase of (2.18). $\hat{w}_k$ is the estimated value of the security code.

The security code estimation is carried out by looking at the different samples within the same chip period and by continuously updating the estimate as soon as more samples are observed. First a matched filter is applied in accordance to the theory of optimum detection [35] and supposing that the attacker perfectly knows the signal structure s_k. Let W_l be the lth value of the security chip and k_l the first sample within the lth chip. Then the matched filter output is

$$Z_l(n) = \frac{2}{n} \sum_{k=k_l}^{k_l+n-1} Y_k s_k, \tag{2.20}$$

where n is the number of samples being observed. Due to the linearity of the filtering operation $Z_l(n)$ is distributed as a Gaussian random variable with mean $\mathrm{E}\left[Z_l(n)\right] = W_l$ where $\mathrm{E}\left[\cdot\right]$ is the expected value. The variance is $\sigma_Z^2(n) = 2\sigma_s^2/n$. Then $Z_l(n)$ can be modelled as

$$Z_l(n) = W_l + N_l(n) \qquad N_l(n) \sim \mathcal{N}\left(\sigma_Z^2(n)\right). \tag{2.21}$$

Note that the variance of the noise component in (2.21) decreases as n increases meaning that the estimate gets better as we observe more samples. The attacker however wants n to be as small as possible so that his time advantage over the defender is greater.

The expression of the estimate of W_l varies based on what optimality criterion the attacker chooses. We summarize here three criterions without derivation, which can be found in [13]. The three different estimate expressions are:

1. maximum likelihood
$$\hat{W}_l^{ML}(n) = Z_l(n); \tag{2.22}$$

2. maximum a posteriori
$$\hat{W}_l^{MAP} = \mathrm{sgn}(Z_l(n)); \tag{2.23}$$

3. minimum mean square error
$$\hat{W}_l^{MMSE} = \tanh\left(\frac{Z_l(n)}{\sigma_Z^2(n)}\right). \tag{2.24}$$

The overall estimation scheme is presented in Fig. 2.8. A numerical simulation of the MMSE and MAP criterion is shown in Fig. 2.9. We can see how the estimate stabilizes as time goes on. Note that due to the sgn(·) function the output of the MAP criterion is either 1 or -1 (*hard*), while MMSE output is *soft* and can be any real value.

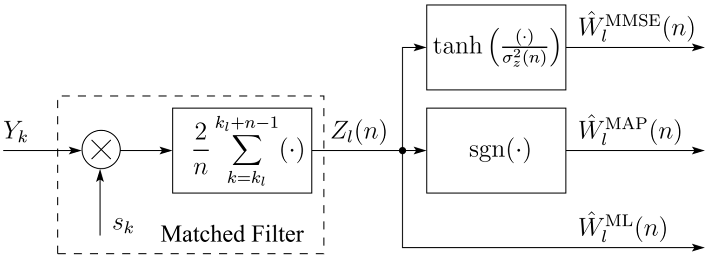

Y_k
s_k
$\frac{2}{n}\sum_{k=k_l}^{k_l+n-1}(\cdot)$
Matched Filter
$Z_l(n)$
$\tanh\left(\frac{(\cdot)}{\sigma_z^2(n)}\right)$
$\mathrm{sgn}(\cdot)$
$\hat{W}_l^{\mathrm{MMSE}}(n)$
$\hat{W}_l^{\mathrm{MAP}}(n)$
$\hat{W}_l^{\mathrm{ML}}(n)$

Figure 2.8: Security code estimation scheme. Figure from [13].

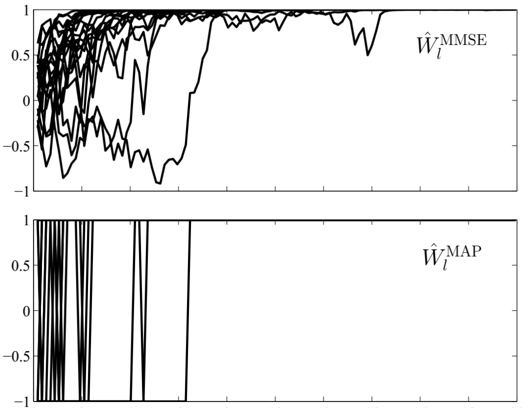


Figure 2.9: Numerical simulation of MMSE and MAP estimation. Figure from [13].

Detection strategy

The author of [13] also proposes a detection strategy for the SCER attack performed by the defender. The detection strategy is based on an hypothesis test that involves two scenarios: the received signal is genuine; the received signal is a counterfeit signal and the genuine one has been nulled. The evaluation of the likelihood ratio between the two hypothesis yields the following detection test, that is to be compared with a threshold:

$$S_l = \sum_{k=k_l}^{k_l+M-1} y_k \beta(n_{lk}) W_l s_k. \tag{2.25}$$

We refer to S_l with *chip statistic* and M is the number of samples per security chip. (2.25) is a weighted correlation between the received samples y_k and the genuine signal $W_l s_k$. The weighting function is $\beta(n_{lk})$ with n_{lk} being the first k samples of the lth chip. β depends on the chosen security code estimation strategy and its behaviour is such that it weights more the first samples of the chip where the two signals, the genuine one and the counterfeit one, are more distinguishable. The computation of S_l assumes that the defender knows the true value of the security code W_l. It is assumed that the defender either can reconstruct it or that it gets disclosed after some delay.

By looking at several chip statistics $\{S_l : l = 1, 2, \dots\}$ detection performances can be improved by computing a symbol level detection statistic. Let $\mathbf{S} = [S_{l_m}, S_{l_m+1}, \dots, S_{l_m+N-1}]^T$ be the collected N chip statistics and let $\mathbf{s}$ be a particular realization of $\mathbf{S}$. Then the symbol statistic is [13]

$$L(\mathbf{s}) = a\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}, \tag{2.26}$$

where a is a constant and $\tilde{\mathbf{s}}$ is a linear function of $\mathbf{s}$ depending again on the security code estimation technique chosen for the attack.

Thresholds for comparison with the detection statistics (2.25) (2.26) are computed given a desired value of probability of false alarm or miss detection.

2.4.2 Forward error estimation attack

SCER attack focuses on the received signal in a chip by chip or symbol by symbol fashion as if they were independent. Transmission of the navigation data, however, involves channel coding which introduces dependencies between the transmitted symbols. The redundancy introduced by channel coding can be exploited by the attacker to predict the entire codeword without the need of receiving all its symbols. If the attacker has high signal-to-noise ratio he will need only half of the symbol (supposing a 1/2 rate code is used) to decode the whole codeword. Moreover, not all bits of the navigation message are unpredictable because NMA cannot occupy all bandwidth. This increases the prediction performance because the attacker needs less symbols to retrieve the whole codeword. Also, the attacker is not constrained to predict all symbols perfectly. Actually he will send to the defender random symbols at the beginning, when the attacker doesn't have a reliable codeword estimate. The receiver will be initially unaware of the ongoing attack and will consider wrong symbols as corrupted by thermal noise. Forward error correction (FEC) provided by channel coding will most likely correct wrong predicted symbols.

If NMA is implemented decoded symbols are fed to the cryptographic verification function, but the verification will succeed because the received codeword has been successfully reconstructed by FEC. At this point the defender thinks the ranging information he has received is authentic while the attacker may have forged a counterfeit ranging signal. This attack methodology is called forward estimation attack (FEA) [22].

2.4.3 State modelling attack

As discussed in section 2.4.1 we consider that the defender implements also a correlation based SCER detection. This can indeed frustrate FEA attack because can cause the defender to be aware of the ongoing attack during the first received samples. At this time the attacker has not yet estimated the codeword and can only send random symbols. However this issue can be overcome by the attacker as we see next.

Authors in [22] summarize the correlation based defence technique as follows. The receiver implements a coherent integration to produce a vector of noisy received samples $\mathbf{r} = \breve{\mathbf{s}} + \mathbf{n}$. The integration is performed within the leading edge of of the received signal, of which the information content are the samples in $\breve{\mathbf{s}}$, because this is the period of time when it is more likely to detect the spoofing attack (see 2.4.1). Then he correlates $\mathbf{r}$ with the corresponding samples, denoted with $\tilde{\mathbf{v}}$, of the genuine security related symbols reconstructed or disclosed at the receiver:

$$\gamma = \mathbf{r}^T \tilde{\mathbf{v}}. \tag{2.27}$$

If the received signal is genuine then γ should follow a Gaussian distribution with mean n_q, the total number of samples. If instead the received symbols are random counterfeit symbols, then γ is zero mean.

However correlation is an additive process and the receiver cannot discriminate between two identical values of γ obtained with different sets of correlation samples. This fact is exploited by the state modelling attack (SMA). When estimating the kth symbol $\tilde{v}_k$ the adversary knows both the previous symbols he sent $\hat{s}_i$ for $i = 0, \ldots, k-1$ and the genuine symbols $\tilde{v}_i$ for $i = 0, \ldots, k-1$ that are disclosed as soon as the kth symbol is received. The attacker can then model the current state of the correlation (2.27) before trying to estimate s_k:

$$\hat{\gamma}_{k-1} = \sum_{i=0}^{k-1} \breve{s}_i \tilde{v}_i. \tag{2.28}$$

The attacker then knows which of the $k-1$ previous symbols has been correctly guessed and which not. Therefore he tunes the amplitude of the next symbol s_k such that correlation (2.28) remains high. For example if he discovers that symbol $k-1$ has been incorrectly guessed, $\hat{\gamma}$ will be lower and then he sends s_k with greater amplitude in order to compensate.

Chapter 3

Information-theoretic authentication protocol

In this Chapter we propose our authentication protocol which is based on physical layer information-theoretic security [33]. The related theoretical concepts are recalled in the Appendices. In section 3.1 we describe the various components of our system model: signaling, channel model and attack model. Section 3.2 describes the authentication protocol and its design requirements. Performance metrics are then analysed in section 3.3.

3.1 System Model

Fig. 3.1 shows our reference scenario. A satellite, Alice offers positioning services via broadcast transmission; the legitimate receiver is represented by Bob and the malicious user Eve has also full access to the positioning signal transmitted by Alice. Indeed, we assume that Eve is in the best possible situation such as high signal-to-noise ratio (SNR) and powerful and costly front-end equipment. The ground segment communicates with Bob through an *authenticated* channel, i.e. messages received by Bob over this channel come for sure from the ground segment (rather then from Eve). The information travelling in the authenticated channel is available to all users, Eve included.

3.1.1 Galielo signaling

Starting from the complete signal modulation presented in section 2.1.1 we rewrite here the notation in order to simplify the model. This will favour a cleaner description of the authentication protocol.

We call the data channel $p(t)$ and it is given by

$$p(t) = \sum_i d_i s_p(t - iT_s), \tag{3.1}$$

where T_s is the symbol period and d_i is the binary data stream. s_p is the spreading pulse ant it

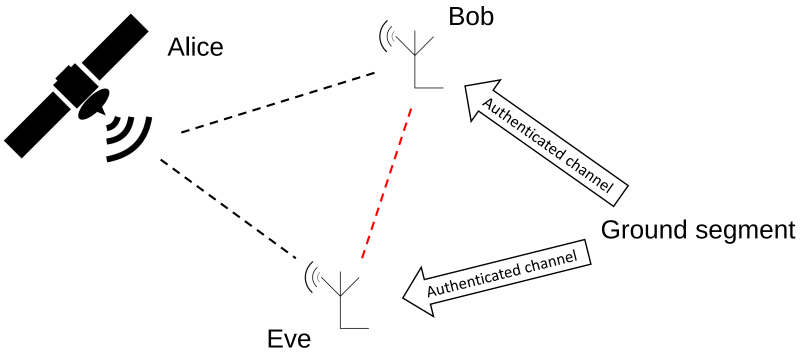


Figure 3.1: Reference scenario.

is given by

$$s_p(t) \triangleq \sum_{i=0}^{N_c-1} c_i u(t - iT_c), \tag{3.2}$$

where T_c is the chip period, $\{c_i\}$ for $i = 0, \dots, N_c - 1$ is the spreading sequence and $u(t)$ is the chip pulse. The chip pulse is a suitable waveform with finite support T_c [34] composed of a sequence of signed rectangular functions as shown in Fig. 2.1. The pilot channel is similar to $p(t)$ but it has no binary data stream and will be neglected in this thesis since it is not relevant for our authentication scheme.

3.1.2 Channel model

There are two channels in our model: the wireless navigation channel and the authenticated channel.

Navigation channel

The navigation channel connects the satellite to the users and is the means through which the ranging signals propagate. In our model of Fig. 3.1 we have two navigation channels: one from Alice (A) to Eve (E), and the other from Alice to Bob (B) . Denoting with $h_{mn}(t)$ the impulse response of the channel between each couple of users, with $m, n \in \{A, B, E\}$, the signals received by Bob and Eve are respectively

$$r_B(t) = (s_t * h_{AB})(t) + w_B(t) \tag{3.3}$$
$$r_E(t) = (s_t * h_{AE})(t), \tag{3.4}$$

where $w_B(t)$ is a Gaussian process with power spectral density (PSD) $N_0/2$ modelling the noise introduced by Bob's RF equipment, s_t is the signal transmitted by Alice and $*$ denotes the convolution. Note that we pose ourselves in the best possible condition for Eve thus her signal is not affected by noise.

In an additive white Gaussian noise (AWGN) channel the impulse response is an ideal unitary impulse and the received signals are

$$r_B(t) = s_t(t) + w_B(t) \tag{3.5}$$

$$r_E(t) = s_t(t). \tag{3.6}$$

The channel between Eve and Bob is used by Eve to send counterfeit signals and do spoofing attacks as we will see later.

Authenticated channel

We assume that the ground segment can communicate with all the users through an authenticated data channel. The authenticated channel is assumed to be of large (infinite) bandwidth and can be for example an internet connection through mobile device. The authentication is ensured for example by higher layer authentication protocols. The authenticated channel is public hence also Eve has access to it while she can not modify its content. Even if, for the protocol to work, we need the information travelling in the authenticated channel to be linked with that coming from the satellites, we do not assume a fine time synchronization between the signal in the authenticated channel and the ranging signals. For this reason we assume that the authenticated channel is worth-less for ranging purposes.

3.1.3 Attack model

Eve is the malicious entity and, as such, her objective is to forge a new signal (possibly with wrong positioning informations), send it to Bob and let him believe it was transmitted by Alice. If Eve knows in advance the signal transmitted by Alice (i.e. it is perfectly predictable), Eve can superimpose a powerful time-shifted version of this signal to Alice's signal. Bob will lock on the strongest signal by synchronization, and then will acquire the timing chosen by Eve. As a consequence Eve will be able to induce the desired (false) ranging on Bob by properly choosing the time shift. In order to avoid this, the signal must be unpredictable to Eve, so that she cannot regenerate it. However, Bob must also be able to establish the authenticity of the message, otherwise Eve will simply generate a consistent message with the desired timing and Bob will again be deceived. One possible solution to provide unpredictability is to include random bits into the data stream that are then confirmed after transmission through the authenticated channel. However, an attack is still possible in this setting, which is addressed by this paper. Suppose that Eve is able to predict the signal of Alice after a time Δ. Then Eve will transmit noise to Bob up to time Δ when she will start transmitting the properly delayed signal to Bob. We describe two attacks by which Eve can predict the entire ranging message with a delay Δ.

Symbol prediction attack

In this attack Eve works at the waveform level. The chip pulse $u(t)$ is perfectly predictable since it is part of the standard [34]. $u(t)$ is the result of superposition and multiplication of rectangular pulses and hence $u(t)$ is a piecewise constant function. Eve's observation is still piecewise constant because she is not affected by noise as discussed before. Then, by reading a

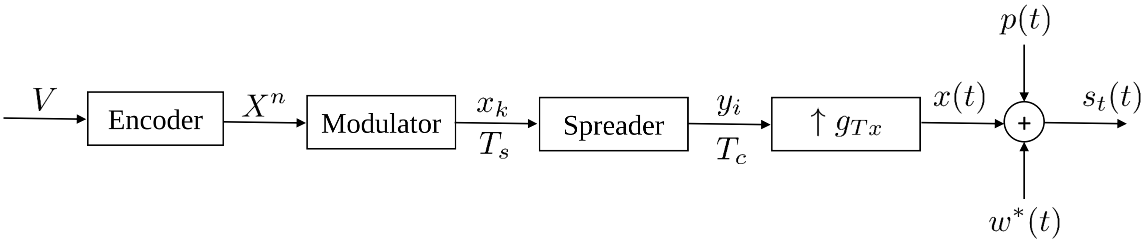


Figure 3.2: Transmitter scheme

small time portion Δ, with $\Delta \ll T_c$, Eve can predict the whole chip symbol. The same holds for the spreading pulse $s_p(t)$ because also the spreading sequence is part of the standard and hence perfectly predictable once the first chip symbols are known. Eve then can predict far in advance the whole symbol and the binary data associated with it. This prediction operation requires fast and accurate RF equipment but we assume that Eve has infinite computational power.

Codeword prediction attack

Typically most of the transmitted bits are predictable (e.g. ephemeris, navigation data, clock synchronization bits) and only *a few* (with respect to the codeword length) bits are meant to be secret or crucial for the authentication mechanism. Due to the error correction codes used to protect the word, by observing a fraction of the codeword it is possible to identify the whole codeword [22]

3.2 Authentication protocol

The basic idea of the proposed protocol is to superimpose a synchronous authentication signal $x(t)$ to the ranging signal $p(t)$. The authentication signal cannot be decoded and predicted by Eve as it remains secret to her, thus preventing the predictive attacks. This is achieved by using concepts of confidential message transmission in physical layer security [33], by obfuscating a message V with artificial noise (AN) $w^*(t)$. After transmission of $p(t) + x(t) + w^*(t)$ both the AN and the message V are revealed to Bob (and Eve), through the authenticated channel. Then Bob can remove the AN from the originally received signal, decode the authentication message and check its correspondence with V to confirm the authenticity of the received signal. We know detail the proposed solution.

The authentication protocol consists of two phases.

First phase In the first phase Alice encodes a secret authentication message V in a codeword X^n. The codeword enters the modulator which outputs constellation symbols x_k working at symbol time T_s. Then the spreader multiplies each symbol for a spreading sequence yielding

$$y_i = x_{\lfloor i/N_c \rfloor} c_{i \bmod N_c} \tag{3.7}$$

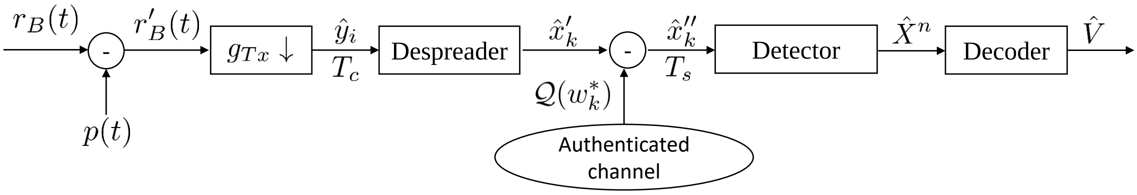


Figure 3.3: Receiver scheme.

every chip period T_c. Finally the unit energy interpolator filter $g_{Tx}(t) = \frac{1}{\sqrt{N_c}}u(-t)$ yields the continuous time signal

$$x(t) = \sum_k x_k s_x(t - kT_s) \tag{3.8}$$

$$s_x(t) = \sum_i y_i u(t - iT_c) \tag{3.9}$$

In order to guarantee the secrecy of message V, we use AN superimposed to $x(t)$, so that even if Eve has a noiseless receiver, she cannot decode (and predict) V. Therefore Alice generates a continuous time white Gaussian AN $w^*(t)$ with variance $\sigma^2_{w^*}$. Signals $x(t)$ and $w^*(t)$ are superimposed to the ranging signal $p(t)$ described in Section 3.1.1 and the signal transmitted by Alice becomes

$$s_t(t) = p(t) + x(t) + w^*(t). \tag{3.10}$$

The transmission scheme is shown in Fig. 3.2. $x(t)$ is by design orthogonal to $p(t)$ (as detailed in Section 3.2.3), hence all the users can extract $p(t)$ from the received signal and use it for the standard ranging procedures. The authentication protocol then works on the remaining part of the received signals. After the removal of $p(t)$ the signals of Bob and Eve are

$$r'_B(t) = x(t) + w^*(t) + w_B(t) \tag{3.11}$$

$$r'_E(t) = x(t) + w^*(t). \tag{3.12}$$

Second phase In the second phase the receiver samples and despreads the received signal as reported in Fig. 3.3 to obtain

$$\hat{x}'_k = x_k + w^*_k + w_{B,k}, \tag{3.13}$$

where

$$w^*_k = \sum_i c_i \int_{(i-1)T_c}^{iT_c} w^*(\tau) g_{Tx}(iT_c - \tau) d\tau \tag{3.14}$$

$$w_{B,k} = \sum_i c_i \int_{(i-1)T_c}^{iT_c} w_B(\tau) g_{Tx}(iT_c - \tau) d\tau \tag{3.15}$$

are still iid AWGN samples with zero mean and power $\sigma^2_{w^*}$ and $\sigma^2_{w_B}$. Similarly also Eve can do the same operations yielding

$$\hat{x}'_{E,k} = x_k + w^*_k. \tag{3.16}$$

The synchronization for sampling is obtained from the ranging signal $p(t)$. Through the authenticated channel the samples w^*_k are revealed to Bob who cancels them before detection to obtain the decoded message $\hat{V}$. Actually, sending a complex-valued signal through the authentication channel would require an infinite number of bits. Instead we can send a quantized version of $w^*(t)$ so to make the number of bits finite. This number is a parameter that can be tuned to trade-off between performance and cost. We denote the quantization map with $\mathcal{Q}$ and define the quantization error

$$w^{(q)}_k \triangleq w^*_k - \mathcal{Q}(w^*_k). \tag{3.17}$$

In this case the signal at the input of the detector is

$$\hat{x}''_k = x_k + w_{B,k} + w^{(q)}_k \tag{3.18}$$

Note that for perfect reconstruction, i.e. with no quantization, it is $w_q(t) = 0$. Definitions and results of the quantization theory are recalled in Appendix B and the links with the authentication signals are the following:

- the input signal is $w^*(t)$;
- the quantization error is $w_q(t)$.

At this point Bob can demodulate and decode the signal $x(t)$ as shown in Fig. 3.3. It is essentially the inverse scheme of the transmitter, with a hat $\hat{\cdot}$ indicating that the symbols are the received noisy counterpart of the transmitted ones. At the end Bob decode the message $\hat{V}$.

Together with $w^*(t)$ the ground segment also reveals the original message V. If $\hat{V} = V$ Bob declares that the portion of $p(t)$ superimposed to $x(t)$ received in the first phase is authentic and comes from Alice. Otherwise Bob generates an exception and the same portion of $p(t)$ is declared not authentic. Note that since both sample and frame synchronization are obtained from $p(t)$, any misalignment between $p(t)$ and $x(t)$ would result in an error of the decoded message V.

3.2.1 Protocol analysis

With the introduction of the artificial noise $w^*(t)$ the prediction attacks become impossible for Eve, since the chip waveform is obfuscated by noise. The unpredictability applies only on $x(t)$ since $p(t)$ is orthogonal to $w^*(t)$, but this is not a problem because we are using only $x(t)$ to transmit the authentication message V. In the first phase then V is undecodable both for Bob and for Eve. In the second phase the revelation of the artificial noise $w^*(t)$ (or it's quantized version) makes $x(t)$ intelligible and decodable so that Bob can retrieve $\hat{V}$. Note that now also Eve can decode $x(t)$, but this is not a problem because the portion of $p(t)$ to be authenticated has already been received and processed in the first phase. Eve however has still access to the fully predictable $p(t)$ and could attack Bob by delaying it so that to induce false pseudo-range measurements *without* interfering with $x(t)$. We need indeed $x(t)$ to be tightly bonded to p(t) so that a delay (or anticipation) in $p(t)$ would result in Bob not being able to decode $x(t)$. Not decoding $x(t)$ means that $V \neq \hat{V}$ and hence the attack is frustrated. This is possible and is discussed in detail in Section 3.3.2.

The *correctness* of this protocol is the ability to properly authenticate the message coming from Alice. This happens if $\hat{V} = V$ when Alice is transmitting, i.e. we must ensure that V is decodable over the Alice-Bob channel. The *security* of the protocol must ensure that when Eve is maliciously acting, the attack is detected, i.e. $\hat{V} \neq V$. To summarize, we must ensure that

1. V is decodable by Bob after the second phase.
2. Eve does not know anything on V after the first phase.

This corresponds to a wiretap transmission scenario. Specifically the legitimate received signal is x''_k over which we require *reliability*; the malicious received signal is $x'_{E,k}$ over which we require *secrecy.*

We make two additional observations.

- The additional information in the second phase ($\mathcal{Q}(w^*_k)$ and V) can not be transmitted over the same channel between Alice and Bob, otherwise if Eve was able to isolate Bob from that particular signal, she could forge whatever signal she liked and authenticate it too. Hence the additional information must be sent over the authentication channel.
- One can think of using a different approach from noise removal to solve the authentication problem and this is *Incremental redundancy.* With incremental redundancy, Alice sends additional redundancy bits relative to V lowering the rate of the total channel code (concatenation of that used in the first phase and with the redundancy bits) to the point that it becomes decodable for Bob. This would avoid the inconveniences of sending noise samples over the authentication channel. However the approach fails in the following scenario. Eve receives a codeword corrupted by the noise $w^*(t)$. She can not decode, she is uncertain between a pool of possible codewords. She selects one of them, the "closest" in terms of probability, she uses it to authenticate a forged signal $p'(t)$ and sends it to Bob. Bob receives it and knows that in order to decode he must wait for the redundancy bits coming in the second phase. The redundancy bits are designed to *correct* the received word into the transmitted one. This happens only if the received codeword is sufficiently close (again, in terms of probability) to the original one, but Eve will select one such codeword with high probability. Hence Bob will always decode correctly the message V and authenticate Eve's signal.

3.2.2 Protocol requirements

The requirements for the protocol are:

1. *Orthogonality* between $x(t)$ and $p(t)$. This requirement ensures that the authentication signal does not interfere with the navigation signal, thus ensuring that a legacy receiver not implementing the authentication features is not affected. $w^*(t)$ should be designed carefully, in order to avoid compromising the nature of other signals involved. Specifically

legacy receivers not implementing the authentication protocol must not be affected, hence $w^*(t)$ should not interfere with $p(t)$.

2. *Secrecy* of the message V to Eve. The message must not be known to Eve and we express this condition as
$$\mathbb{I}(V; r'_E(t)) = 0, \tag{3.19}$$
where $\mathbb{I}(\cdot;\cdot)$ denotes the mutual information function. Note that (3.19) implies also $\mathbb{I}(V; r'_B(t)) = 0$ since Eve has the best possible channel. In this way we guarantee that in the first phase no one is able to decode $x(t)$, thus preventing the prediction attacks.

3. *Reliability* on decoding V at Bob's side in the second phase:
$$P_e^B \triangleq \mathbb{P}[\hat{V} \neq V] = 0, \tag{3.20}$$
where $\mathbb{P}(\cdot)$ denotes the probability.

4. *Synchronization.* Since $x(t)$ and $p(t)$ are orthogonal, Eve can always distinguish between the two messages and operate a predictive attack on $p(t)$. She can then delay or anticipate $p(t)$ without interfering with the authentication procedures. We must then require $x(t)$ to be synchronized to $p(t)$ so that a delay in the latter would cause the authentication protocol to fail.

Secrecy and synchronization requirements deal with the *security* metric of the protocol since they both aim to maintain V secret from Eve. Reliability and orthogonality instead deal with correctness since are the necessary conditions to let Bob properly decode V and authenticate $p(t)$.

We now revise the requirements and obtain design criteria for our protocol.

3.2.3 Orthogonality

About the artificial noise $w^*(t)$ it is useful to recall here the *theorem of irrelevance* from the theory of optimum detection [35]. In a communication system let $\mathbf{r}$ be the vector representation of a received signal and α the transmitted symbol. Suppose that $\mathbf{r}$ can be split into two parts $\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2]$ such that the probability of $\mathbf{r}_2$ given $\mathbf{r}_1$ and α is independent of the particular transmitted symbol n, that is

$$p_{\mathbf{r}_2|\mathbf{r}_1,\alpha}(\boldsymbol{\rho}_2|\boldsymbol{\rho}_1, n) = p_{\mathbf{r}_2|\mathbf{r}_1}(\boldsymbol{\rho}_2|\boldsymbol{\rho}_1). \tag{3.21}$$

The theorem of irrelevance states that the optimum receiver can disregard the component $\mathbf{r}_2$ and base its decision only on the component $\mathbf{r}_1$. In our context this ensures that if we build $w^*(t)$ laying in a different basis from that of $p(t)$ then this will not interfere with the reception of the positioning signal.

An orthonormal base for the signal (2.1) is the set of signals $\{\phi_i(t)\}$ defined as

$$\phi_i(t) = \frac{1}{\sqrt{E_r}} \operatorname{rect}\Big(\frac{t - 1/2 - iT_c}{T_c}\Big) \qquad i = 0, \ldots, N_c - 1, \tag{3.22}$$

where E_r is the energy of each single rect pulse. A possible way of designing $w^*(t)$ would be as follows. First generate a Gaussian process $w(t)$ and project it on the base $\{\phi_i(t)\}$ finding coefficients w_i such that

$$w_i = \int_{-\infty}^{+\infty} w(t)\phi_i^*(t)dt \tag{3.23}$$

$$w_{\|}^*(t) = \sum_{i=0}^{N_c-1} w_i\phi_i(t). \tag{3.24}$$

Then by subtracting $w_{\|}^*(t)$ to $w(t)$ we obtain a signal orthogonal to $p(t)$, i.e.

$$w^*(t) = w(t) - w_{\|}^*(t). \tag{3.25}$$

This is still a Gaussian process because (3.23) and (3.24) are linear transformations of another Gaussian process.

Another way to obtain $w^*(t)$ starts from the definition of orthogonality between $w^*(t)$ and $p(t)$

$$\int_{-\infty}^{+\infty} w^*(t)p^*(t)dt = \sum_i d_\ell c_i \int_{iT_c}^{(i+1)T_c} w^*(t)dt = 0. \tag{3.26}$$

Then a sufficient condition for orthogonality is

$$\int_{iT_c}^{(i+1)T_c} w^*(t)dt = 0. \tag{3.27}$$

In general there are infinite signals that verify (3.27). For example, focusing on one symbol period $[0, T_s]$ with $T_s \triangleq N_cT_c$, $w^*(t)$ can be generated starting from N_c different Gaussian processes $w_i^*(t)$ with support $\mathcal{T}_i = [iT_c, iT_c + T_c/2]$ and for the remaining half of the chip period using $-w^*(t)$. Then we can strengthen the secrecy by still generating $w_i^*(t)$ for half the chip period, but choosing at random the time instant $t_i' \in [iT_c, iT_c + T_c/2]$ such that

$$w^*(t) = \begin{cases} w_i^*(t) & \text{if } iT_c \le t < t_i' \\ -w_i^*(t) & \text{if } t_i' \le t < t_i' + T_c/2 \\ w_i^*(t) & \text{if } t_i' + T_c/2 \le t \le (i+1)T_c \end{cases} \tag{3.28}$$

$$\text{for } t \in [iT_c, (i+1)T_c]. \tag{3.29}$$

Actually the most practical solution is the following. First generate a stationary Gaussian process $w(t)$ with $0 \le t \le T_s$ and then project it on the energy-normalized version of $s_p(t)$, i.e.

$$w_s = \int_0^{T_s} w(t)\frac{s_p(t)}{\sqrt{E_{s_s}}}dt, \tag{3.30}$$

with

$$E_{s_s} = \int_0^{T_s} s_s^2(t)dt. \tag{3.31}$$

Then subtract to obtain $w^*(t)$:

$$w^*(t) = w(t) - w_s \frac{s_p(t)}{\sqrt{E_{s_s}}}. \tag{3.32}$$

In this way $w^*(t)$ is orthogonal only to the specific pulse used for data transmission allowing us to still use rect based pulses for $x(t)$ provided that we use different spreading coefficients. $w^*(t)$ must however be orthogonal to *all* the pulses built with the spreading sequences used for transmitting the ranging signal $p(t)$.

3.3 Protocol performance analysis

3.3.1 Secrecy and reliability requirements

As previously observed this model corresponds to a wiretap transmission channel, where the legitimate channel is x''_k while the eavesdropper channel is $x'_{E,k}$. In order to compute capacities we need to define Bob's and Eve's SNR. Fig. 3.4 shows the equivalent discrete-channel model for the transmission of symbols x_k. We drop here for simplicity encoder and modulation blocks. Consider the transmission of the signal $x(t)$ as in (3.8) with $E_{s_x} = 1$. If we consider also unit energy real Gaussian symbols x_k, the SNR before the despreading filter, in T_c domain, is

$$\Gamma = \frac{E_{s_x}/N_c}{\sigma^2_{w_B}} = \frac{1}{N_c \sigma^2_{w_B}}. \tag{3.33}$$

After the despreading operation it becomes

$$\Gamma_B = N_c \Gamma = \frac{1}{\sigma^2_{w_B}}. \tag{3.34}$$

Similarly for Eve we have

$$\Gamma_E = N_c \Gamma = \frac{1}{\sigma^2_{w^*}}, \tag{3.35}$$

$\sigma^2_{w_B}$ and $\sigma^2_{w^*}$ are the variances of the noise samples at T_c. Channel capacities are then:

$$C_B = \frac{1}{2} \log_2 (1 + \Gamma_B) \tag{3.36}$$

$$C_E = \frac{1}{2} \log_2 (1 + \Gamma_E). \tag{3.37}$$

The same holds for complex Gaussian signals apart from the 1/2 factor. We can now apply (A.13) to obtain

$$C_s = C_B - C_E. \tag{3.38}$$

By looking at the previous equations we deduce that in order to have a positive secrecy capacity it must be

$$\sigma^2_{w^*} > \sigma^2_{w_B}. \tag{3.39}$$

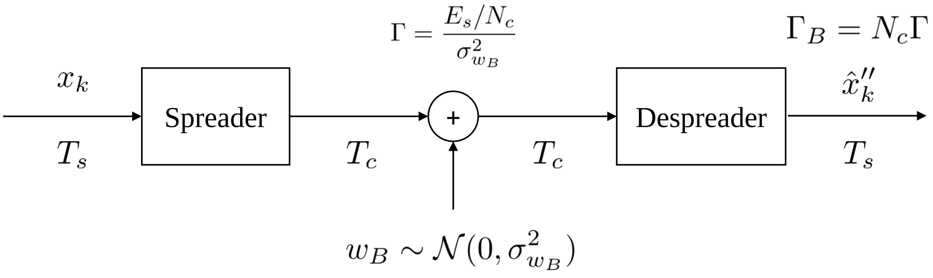

$\Gamma = \frac{E_s/N_c}{\sigma^2_{w_B}}$
$\Gamma_B = N_c\Gamma$
x_k
T_s
Spreader
T_c
+
T_c
Despreader
$\hat{x}''_k$
T_s
$w_B \sim \mathcal{N}(0, \sigma^2_{w_B})$

Figure 3.4: Equivalent discrete-channel model.

This means that we are forcing Eve's channel to be worse than it really is, since it doesn't naturally suffer from thermal noise.

Note that it holds:

$$\lim_{\sigma^2_{w^*} \to \infty} C_s = C_B. \tag{3.40}$$

In other words even if we completely destroy Eve's channel the secrecy capacity can not grow indefinitely, as it is still bounded by Shannon's capacity on Bob's channel.

In case we send the quantized version of w^*_k, $\hat{x}''_k$ has as additive noise also $w_q(t)$ and Bob's SNR becomes:

$$\Gamma'_B = \frac{1}{\sigma^2_{w_B} + \sigma^2_{w_q}}, \tag{3.41}$$

where $\sigma^2_{w_q}$ is the variance of $w_q(t)$. Note that we implicitly assumed that the variance of the sum $w_B(t) + w_q(t)$ is the sum of the corresponding variances. However this is a reasonable assumption because the quantization error can be assumed uncorrelated from the input, as seen in appendix B. Regarding Bob's capacity we can not assume that noise $w_q(t)$ is Gaussian, since it is actually often modelled as uniform distributed (see appendix B). Hence the total additive noise is not Gaussian and the capacity expression for the AWGN channel does not hold exactly. However, we still consider as an approximation the equivalent AWGN channel with total noise variance $\sigma^2_{w_B} + \sigma^2_{w_q}$ and capacity given by

$$C_B = \frac{1}{2}\log_2(1 + \Gamma'_B). \tag{3.42}$$

3.3.2 Synchronization

We now discuss Bob's decoding performance degradation in the presence of a delay attack by Eve. If Eve delays $p(t)$ by $\epsilon > 0$, Bob receives

$$p(t - \epsilon) + x(t) + w^*(t). \tag{3.43}$$

We recall that Eve can do this because $p(t)$ is predictable, unlike $x(t)$ that is submerged by noise. Since synchronization is obtained from $p(t-\epsilon)$ the filtering and despreading of $x(t)$ is affected by interference. Using (3.9) we can write the receiving filter and the despreading operation of Fig. 3.3 as

$$\begin{aligned}\int_0^{T_s} x(\tau+\epsilon)s_x(\tau)d\tau &= \int_0^{T_s-\epsilon} x_0 s_x(\tau+\epsilon)s_x(\tau)d\tau + \int_{T_s-\epsilon}^{T_s} x_1 s_x(\tau - T_s+\epsilon)s_x(\tau)d\tau \\ &= \alpha x_0 + \beta x_1\end{aligned}, \tag{3.44}$$

where x_1 is an interferer, $\alpha < 1$ and β are deterministic constants that depend only on the particular spreading pulse, i.e.

$$\alpha = \int_0^{T_s-\epsilon} s_x(\tau+\epsilon)s_x(\tau)d\tau \tag{3.45}$$

$$\beta = \int_{T_s-\epsilon}^{T_s} s_x(\tau-T_s+\epsilon)s_x(\tau)d\tau. \tag{3.46}$$

We see that interference is present and lowers Bob's demodulating performances. The synchronization requirement formalize as follows: α and β should be as small and big as possible respectively for those values of ϵ we want Bob to be protected. Indeed, if this requirement is satisfied the secrecy capacity is lowered such that Bob won't be able to verify Eve's signal, frustrating her attack. Similarly, for a negative delay $-\epsilon$ it is:

$$\begin{aligned}\int_0^{T_s} x(\tau-\epsilon)s_x(\tau)d\tau &= \int_0^{\epsilon} x_{-1} s_x(\tau+T_s-\epsilon)s_x(\tau)d\tau + \int_{\epsilon}^{T_s} x_0 s_x(\tau-\epsilon)s_x(\tau)d\tau \\ &= \alpha x_0 + \beta x_{-1},\end{aligned} \tag{3.47}$$

with

$$\alpha = \int_0^{\epsilon} s_x(\tau+T_s-\epsilon)s_x(\tau)d\tau \tag{3.48}$$

$$\beta = \int_{\epsilon}^{T_s} s_x(\tau-\epsilon)s_x(\tau)d\tau. \tag{3.49}$$

To evaluate numerically how much the capacity of Bob is affected by the delay attack we consider as SNR

$$\Gamma_B = \frac{\alpha^2}{\sigma_{w_B}^2+\beta^2}, \tag{3.50}$$

assuming we are transmitting unit energy symbols, and use it to compute the secrecy capacity. Note that if there are no delay attack (3.50) applies with $\alpha = 1$ and $\beta = 0$.

3.3.3 Constellation constrained system

We now investigate the performances of the protocol in case of constellation constrained system. We consider in particular M-Phase Shift Keying (M-PSK) modulation as it is a constant envelop modulation technique which is best suited for satellite transmission.The secrecy capacity is still $C_s = C_B - C_E$, but this time there is no close expression available for C_B and C_E and we must resort to numerical integration. The generic scheme for computing the capacity in a M-PSK channel is as follows. Γ is the reference SNR at the symbol level; x is the random variable of the constellation symbols $\alpha \in \mathcal{S} = \{e^{\frac{j\pi}{M}}, e^{\frac{2j\pi}{M}}, \dots, e^{\frac{(M-1)j\pi}{M}}\}$ with pdf $p_x(\alpha)$; $y = x + w$ is the received symbol with pdf $f_y(b)$; w is complex Gaussian with variance $\sigma_w^2 = 1/\Gamma$ and $\sigma_w^2/2$ for each component (real and imaginary). From well known information theory results [36] capacity is:

$$C = H(y) - \log_2(\pi e \sigma_w^2) \tag{3.51}$$

with

$$H(y) = \int_{\mathbb{C}} f_y(b) \log_2 \frac{1}{f_y(b)} db \tag{3.52}$$

$$f_y(b) = \sum_{\alpha \in \mathcal{S}} \frac{1}{\pi \sigma_w^2} e^{-\frac{|b-\alpha|^2}{\sigma_w^2}} p_x(\alpha). \tag{3.53}$$

It is now sufficient to set $\Gamma = \Gamma_B$ for the computation of C_B and $\sigma_w^2 = \sigma_{w^*}^2$ for C_E.

Chapter 4

Numerical results

In this Chapter we present a number of numerical results evaluated using MATLAB scripts.

4.1 Secrecy capacity and delay attack

In figure 4.1 the secrecy capacity (3.38) is shown as a function of $\sigma^2_{w^*}$ considering typical values of $\Gamma = -20$ dB and $N_c = 4092$. The red line is the value C_B. Note that the secrecy capacity is zero when $\sigma^2_{w^*} \leq \sigma^2_{w_B}$

We want now to see how the synchronization requirement of section 3.2.3 can help Bob to understand if an attack is ongoing thanks to the degradation of the secrecy capacity. We have to evaluate numerically the SNR (3.50) for each value of ϵ. α and β can be computed from (3.44) and (3.47) and then plugged into (3.50). Eve's SNR is taken from (3.35) and the secrecy capacity is (3.38).

In figure 4.2 is shown how C_s decreases due to different values of the delay ϵ, given $\sigma^2_{w^*} = -6$ [dB], $\Gamma = -20$ and a PRN sequence taken from the standard [34]. The plot changes only slightly based on what PRN sequence is chosen and the overall behaviour is the same. Now we can think of using different T_c pulses other than rect as shown in figure 4.3. $\gamma < 1$ is a parameter that can be tuned n order to adjust the amount of energy we want to allocate to the pulse. The smaller γ, the more secure results we get in terms of frustrating Eve's delay attack. This is shown in Fig. 4.4 where $\gamma = 0.2$. We note considerable improvements with respect to Fig. 4.2, at the cost of more complexity since we are working with pulses smaller than T_c. Actually this may not be feasible for the receiver, because it is typically a cheap device.

We now assume that all the correlations in (3.44) and (3.47) are carried out with the receiver using always the spreading pulse (3.2) even if the transmitter uses the pulses in Fig. 4.3. Actually this is the more realistic scenario where the user equipment is cheap and cannot work at lower sampling periods than T_c. Fig. 4.2 remains unchanged, while Figures 4.4a and 4.4b become Fig. 4.5 and 4.6 respectively. Two observations. First, note that Fig. 4.5 is not symmetric with respect to ϵ as a consequence of Fig. 4.3a being asymmetric itself. Secondly, the value at $\epsilon = 0$ of C_s is different (lower) from the previous figures, σ^*_w being equal in both cases. This is because the value α in (3.50) is no more 1 for $\epsilon = 0$ since even with perfect synchronization you gather less energy in every single chip period given that we use unit gain pulses. To re-gain the same

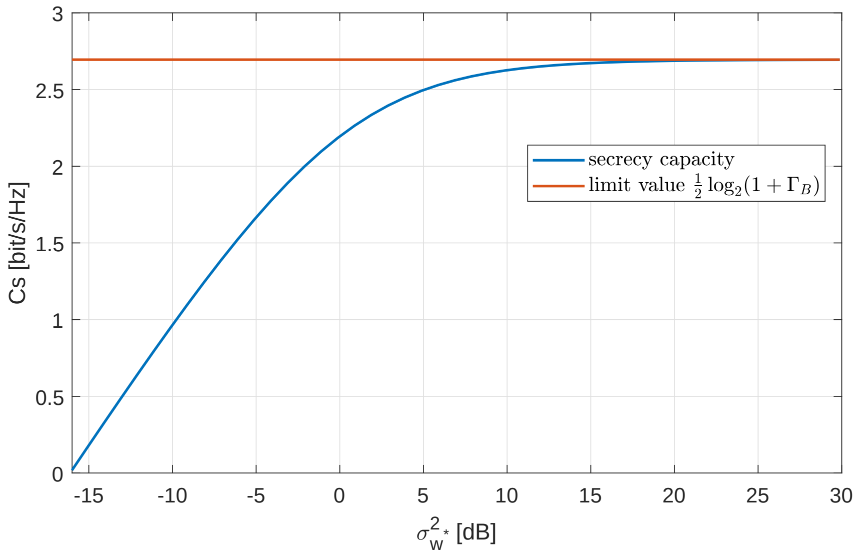


Figure 4.1: Secrecy capacity versus $\sigma^2_{w^*}$

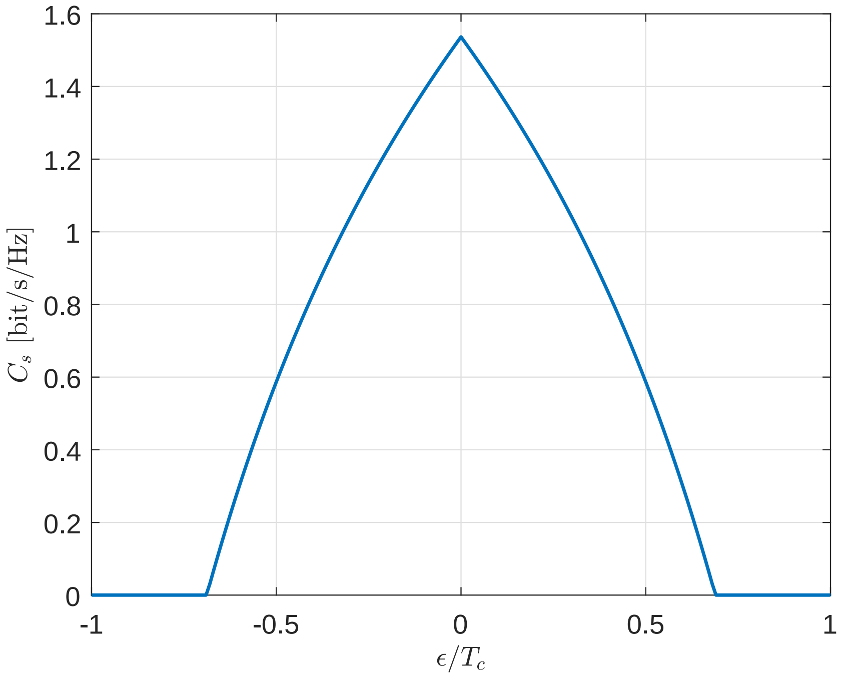
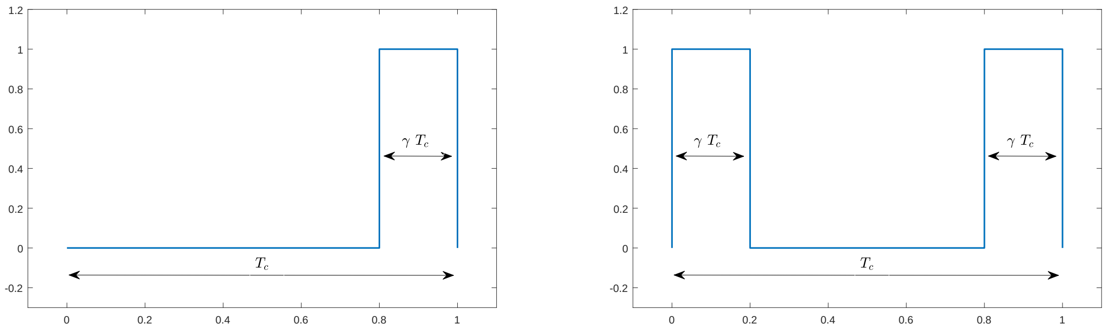


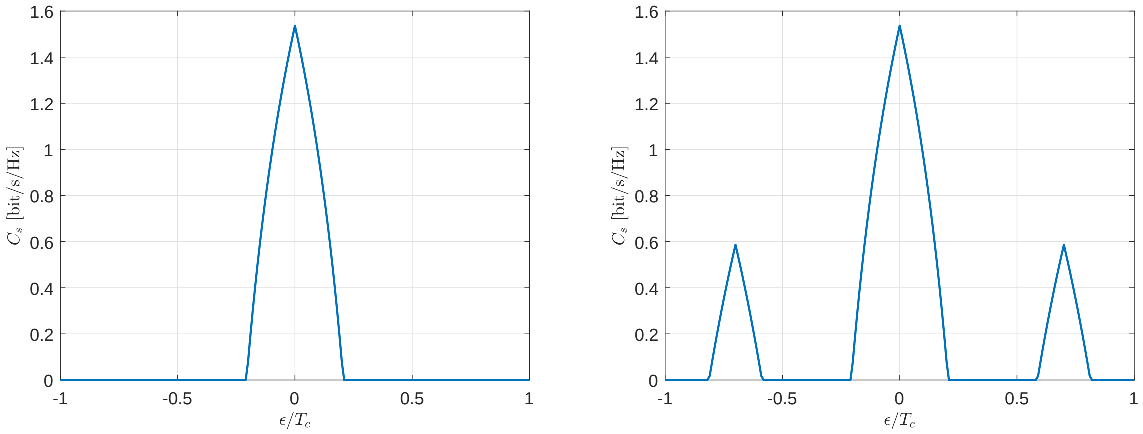
Figure 4.2: Degradation of secrecy capacity with respect to different delays.



(a)

(b)

Figure 4.3: Two different T_c pulses



(a) Degradation of C_s using pulse in Fig. 4.3a.

(b) Degradation of C_s using pulse in Fig. 4.3b.

Figure 4.4: Degradation of C_s obtained using pulses in Fig. 4.3.

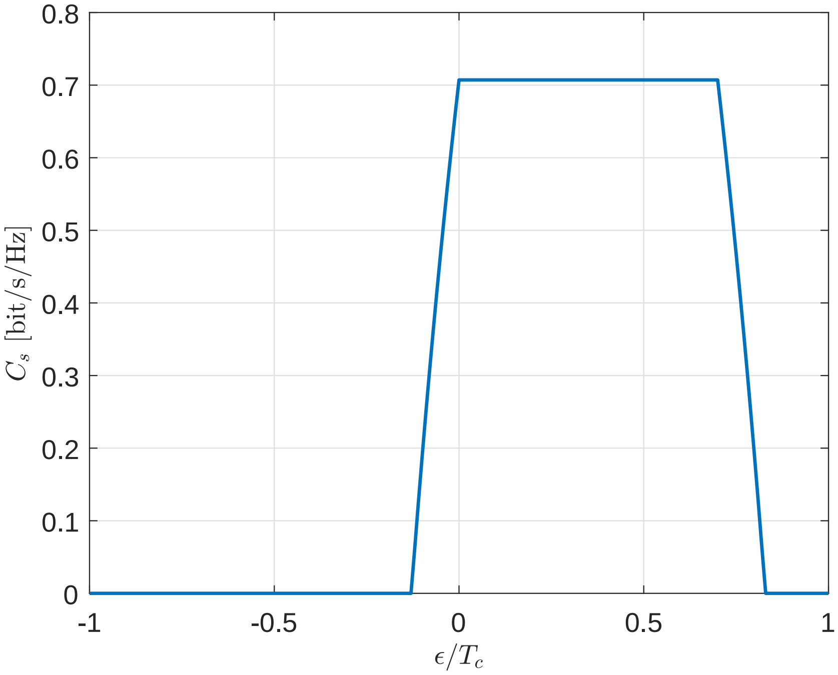


Figure 4.5: Degradation of secrecy capacity with respect to different delays using pulse 4.3a only at the transmitter.

level of C_s as before one need to upscale the side pulses of Fig. 4.3 proportionally to γ.

4.2 M-PSK modulation

We now go through the same calculations in the case of complex modulation. Specifically we use PSK with order M. As we did for Fig. 4.1 we set $\Gamma_B = 16$ [dB] and let σ_w^* vary. The resulting secrecy capacity is shown in Fig. 4.7 for different values of M and for $p_x(\alpha) = 1/M$.

Note that by doubling M we gain one bit of secrecy capacity in the limit values represented by dashed lines. This is because in general for M-PSK it holds

$$\lim_{\Gamma \to +\infty} C = \log_2 M. \tag{4.1}$$

However $\Gamma_B = 16$ [dB] is not sufficiently high to let Bob's channel reach $C_B = \log_2 M$ for $M = 16$ and 32. This and (3.40) tell us that with $\Gamma_B = 16$ we don't gain more secret bits by using bigger constellations than $M = 16$.

Now the same figures of section 4.1 are obtained with M-PSK constellation (Figures 4.8, 4.9 and 4.10). The black line is always the complex Gaussian case.

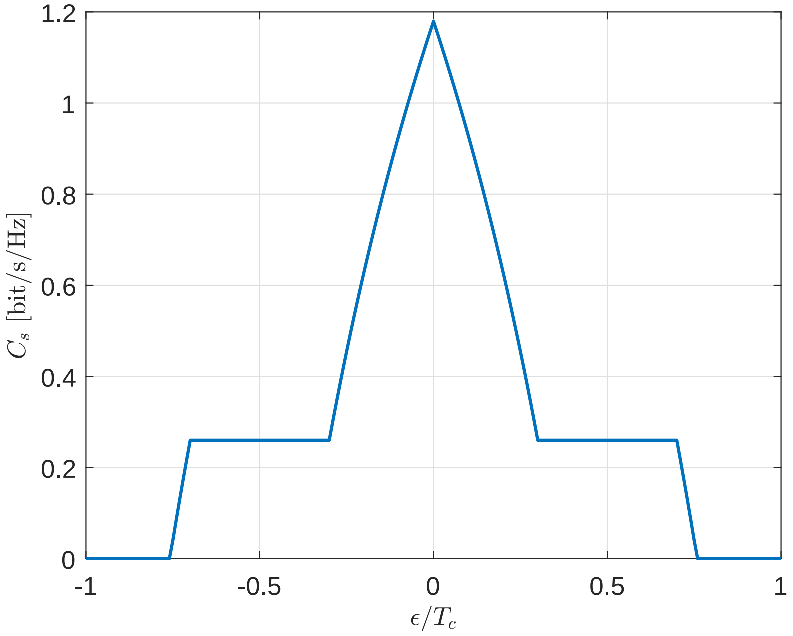


Figure 4.6: Degradation of secrecy capacity with respect to different delays using pulse 4.3b only at the transmitter. $\gamma = 0.3$.

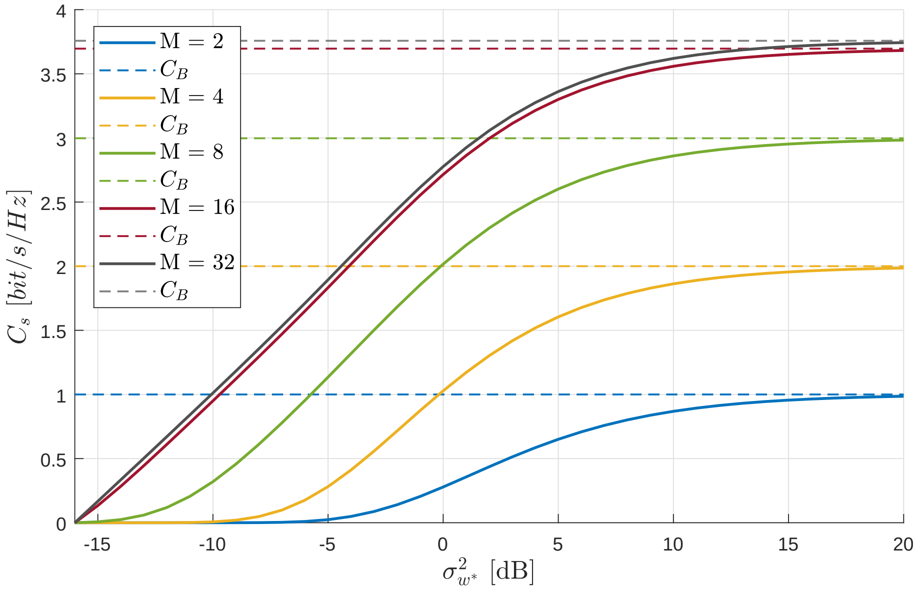


Figure 4.7: Secrecy capacity for M-PSK channel as a function of σ_w^*.

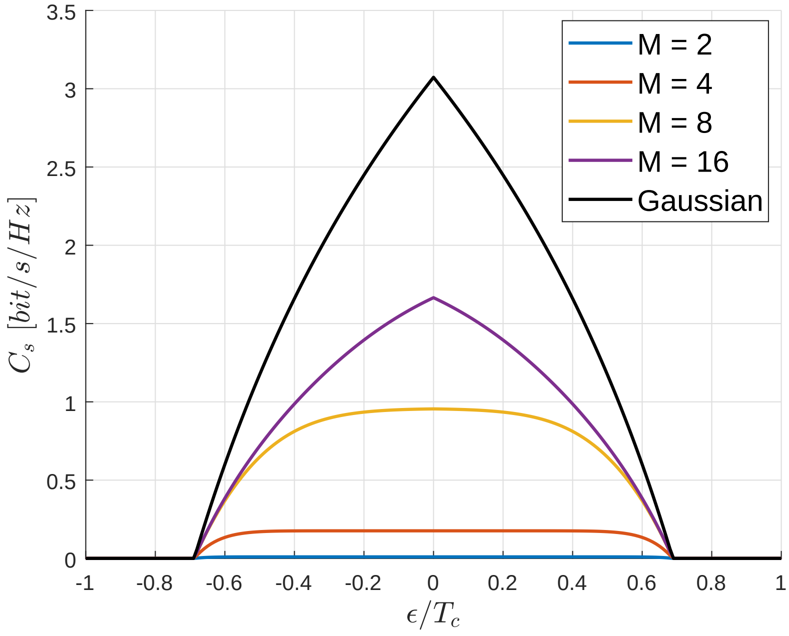


Figure 4.8: Degradation of secrecy capacity for various delays using rect pulse at transmission.

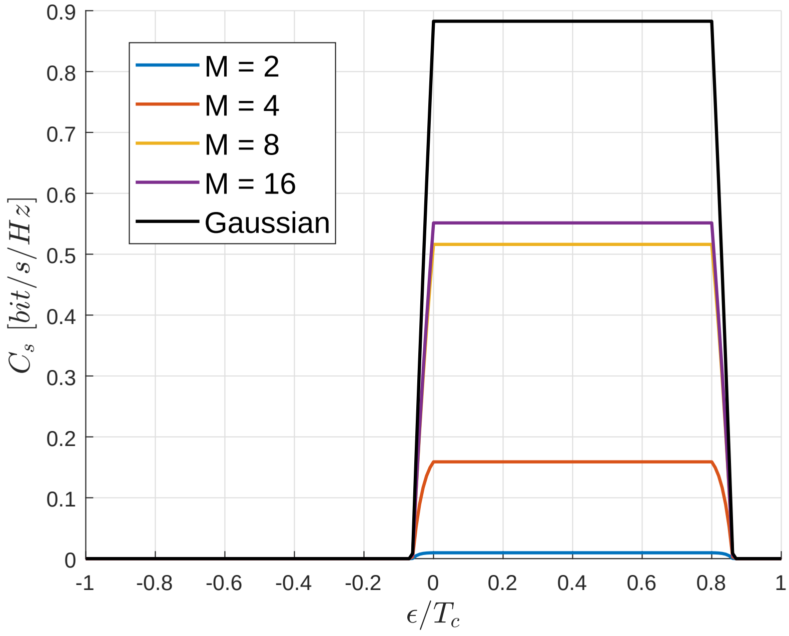


Figure 4.9: Degradation of secrecy capacity for various delays using pulse of Fig. 4.3a.

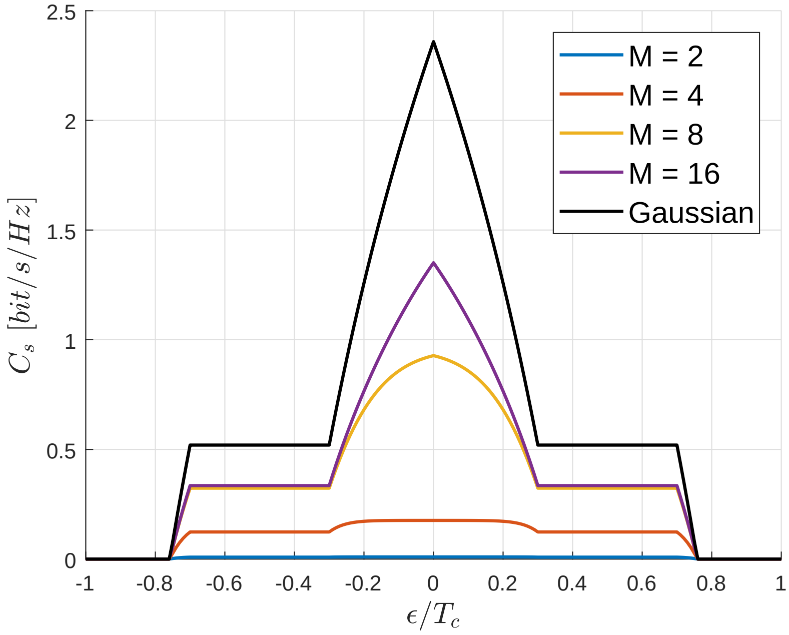


Figure 4.10: Degradation of secrecy capacity for various delays using pulse of Fig. 4.3b and $\gamma = 0.3$.

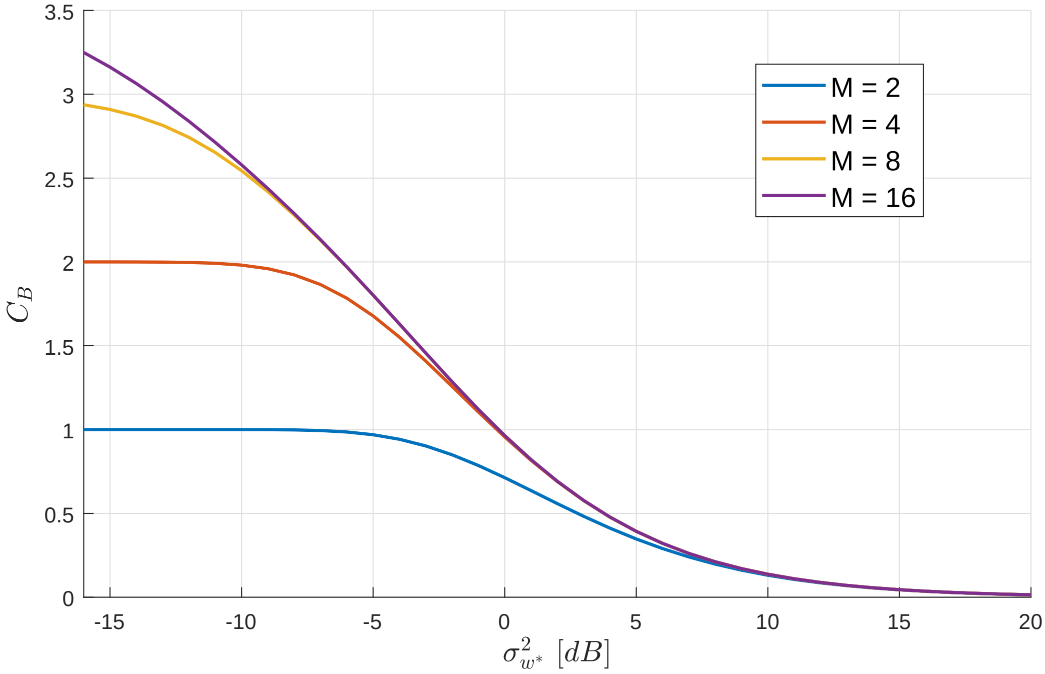


Figure 4.11: Channel capacity C_B for $\Gamma_B = 16$ $[dB]$ and varying $\sigma^2_{w^*}$

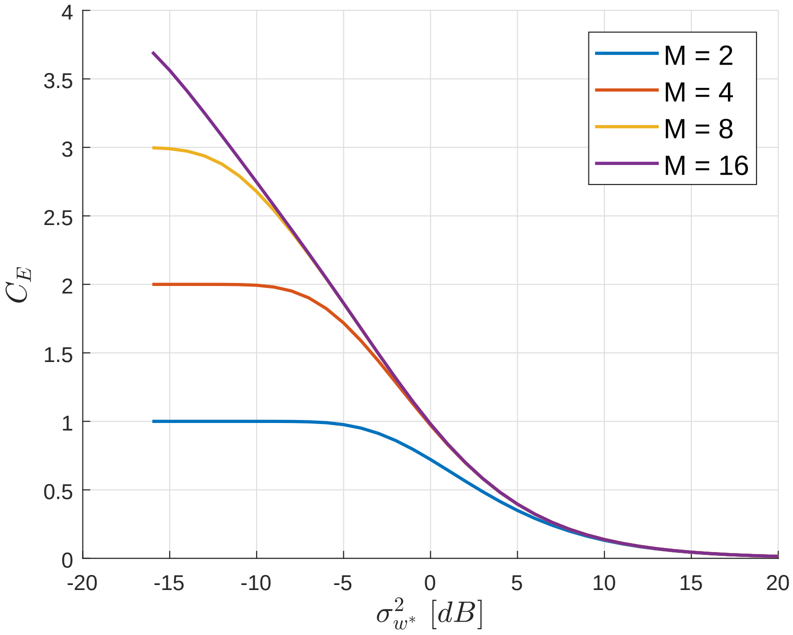


Figure 4.12: Eve's channel Capacity for varying $\sigma^2_{w^*}$

4.2.1 Effects of quantization

We need to compute the SNR Γ'_B as defined in (3.41): the value of $\sigma^2_{w_q}$ are obtained as

$$\sigma^2_{w_q} = \frac{1}{\Lambda_q}, \tag{4.2}$$

where Λ_q is defined in Appendix B. In the literature the values of Λ_q (see appendix B are tabulated as a function of the number of bits b used for the representation of each sample. Λ_q varies also depending on the input distribution of the quantizer. In Table 4.13 are shown the values of Λ_q for a Gaussian input. The numbers for the uniform quantizer are obtained with the procedure described in Section B.2, while the numbers for the non uniform quantizer are obtained with the algorithms briefly recalled in section B.3.

The secrecy capacity is shown In Fig. 4.14 as a function of $\sigma^2_{w^*}$ for different values of b. The black line is the theoretic value with perfect noise cancellation. Clearly, the more bits we use for digital representation, the better we approach the theoretic limit. Note that with 4 bits we loose less than a bit of secrecy capacity.

In Fig. 4.15 we compare the non uniform quantizer with the uniform quantizer. We expect the non uniform quantizer to perform better, since it is designed to specifically fit the Gaussian distribution input. This is indeed true, but the performance gain is very low due to Λ_q itself not varying too much between Uniform and non Uniform quantizers.

b	Quantizer type	
	Uniform	non Uniform
1	4.40	4.40
2	9.25	9.30
3	14.27	14.62
4	19.38	20.20
5	24.57	
6	29.83	
7	35.13	
8	40.34	

Figure 4.13: Λ_q $[dB]$ for different values of b and for uniform and non uniform quantizers. The reference for the numbers is [35].

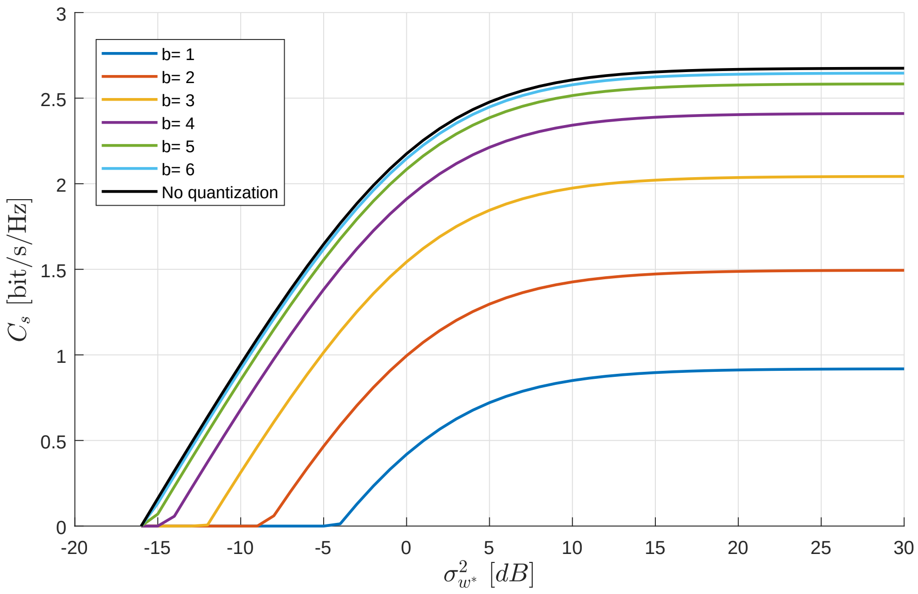


Figure 4.14: Secrecy capacity as a function of $\sigma^2_{w^*}$ for different values of b.

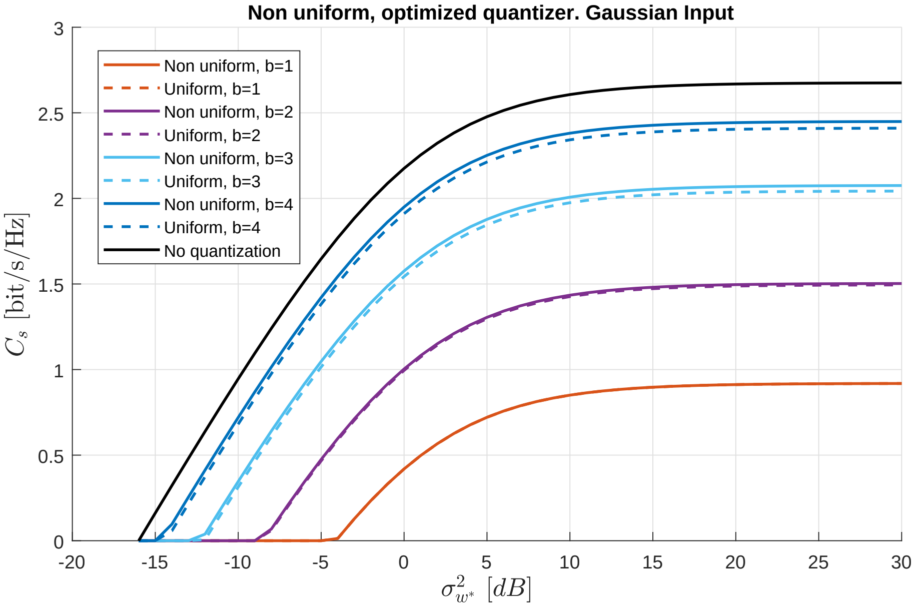


Figure 4.15: Secrecy capacity as a function of σ^2_{w*} for different values of `b` and comparison between uniform and non uniform quantizers.

Chapter 5

Conclusions and future work

In this thesis we developed a new protocol for the authentication of GNSS signals. Performance has been evaluated in AWGN channel model in terms of the secrecy capacity that can be achieved. We showed that it is indeed possible to send a verification message unpredictable to Eve thanks to an artificial additive noise. The rate of the verification message can be tuned to balance both performance and cost of sending noise samples through the authenticated channel. By designing suitable chip waveforms for the authentication signal we allow the legitimate receiver to detect if a spoofer has sent a counterfeit signal. The counterfeit signal is a delayed (or anticipated) version of the true ranging signal and the legitimate receiver will experience capacity degradation and will be then unable to verify the authentication message.

In this work we considered only AWGN channel model in the evaluation of capacities. Moreover the secrecy capacity can be achieved only by using certain type of codes whose design criteria are different from the ones used for standard channel codes. A possible extension of this work would be evaluating protocol performance in presence of more challenging channel models, which introduce interference, and with signal protected by secrecy-focused channel codes. Future work should also consider that additional interference is introduced by the presence of more satellites, each of which transmitting its own verification signal.

Appendices

Appendix A

Wiretap Channel

We concentrate on the original Wyner model in Fig. A.1: the degraded wiretap channel (DWTC). We want to model the fact that in this simple scenario, where there are no feedback channels, Bob needs some physical and intrinsic advantage with respect to Bob. In fact Eve's observation ($\mathcal{Z}^n$) undergoes two transformations in the block diagram, one more than Bob. That's why in the model we use the word *degraded*. It is not the most general model, but it fits our needs for now. A discrete memoryless DWTC is a 5-tuple $(\mathcal{X}, p_{Y|X}, p_{Z|Y}, \mathcal{Y}, \mathcal{Z})$ where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ are finite alphabets and $p_{Y|X}$, $p_{Z|Y}$ are transition probabilities such that

$$p_{Y^n,Z^n|X^n}(y^n, z^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)p_{Z|Y}(z_i|y_i) \qquad \forall n \geq 1 \quad \forall (x^n, y^n, z^n) \in (\mathcal{X}^n, \mathcal{Y}^n, \mathcal{Z}^n). \tag{A.1}$$

Morover, we define a $(2^{nR}, n)$ code $\mathcal{C}_n$ for the DWTC as being comprised by:

- a message set $\mathcal{M} = [1, \ldots, 2^{nR}]$;
- a source of local randomness at the encoder $(\mathcal{R}, p_R)$, where $\mathcal{R}$ is a discrete memoryless source with probability mass function p_R;

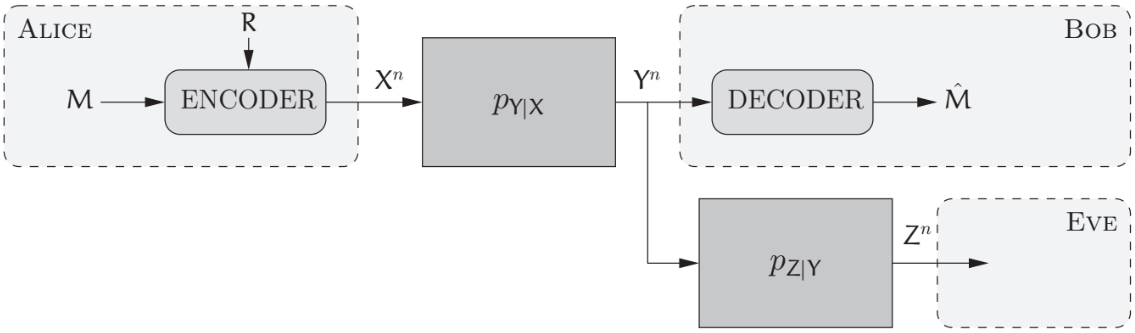


Figure A.1: Degraded Wiretap Channel block diagram. Image from [33].

- an encoding function $f:\ \mathcal{M}\times\mathcal{R}\to\mathcal{X}^n$;
- a decoding function $g:\ \mathcal{Y}^n\to\mathcal{M}\cup\{?\}$ with ? being an error message.

The objective is to design a code that allows the *reliable* transmission of message M to Bob, while keeping it *secret* from Eve. We measure *reliability* with the probability of error

$$\mathbf{P}(\mathcal{C}_n)\triangleq P[\hat{M}\neq M|\mathcal{C}_n] \tag{A.2}$$

, where $\mathbf{P}(\cdot)$ denotes the probability function, and *secrecy* with the equivocation rate

$$\mathbf{E}(\mathcal{C}_n)\triangleq H(M|Z^n\mathcal{C}_n) \tag{A.3}$$

, where $H(\cdot)$ denotes the (conditional) entropy, or, equivalently, with leakage

$$\mathbf{L}(\mathcal{C}_n)\triangleq I(M;Z^n|\mathcal{C}_n)=H(M|\mathcal{C}_n)-H(M|Z^n\mathcal{C}_n), \tag{A.4}$$

where $I(\cdot;\cdot)$ denotes the (conditional) mutual information We would like the leakage to be zero, so that Eve has no information on the transmitted message M and the observation of Z^n doesn't help her in guessing M. This is a *strong* secrecy condition, but for mathematical tractability we require a *weake* condition, i.e.:

$$\lim_{n\to+\infty}\frac{1}{n}\mathbf{L}(\mathcal{C}_n)=0. \tag{A.5}$$

This is a weak condition because the leakage can indeed diverge, but slowly than n; however it can be proven that the following results hold both for *weak* and *strong* secrecy condition.

For reliability, instead, we impose

$$\lim_{n\to+\infty}\mathbf{P}(\mathcal{C}_n)=0. \tag{A.6}$$

A.1 Secrecy Capacity

Channel characteristics impose theoretical limits to our target conditions (A.5) and (A.6). The limit to reliability is classical Shannon's channel capacity, while for secrecy we need to introduce the *secrecy capacity.*

A weak rate–equivocation pair (R,R_e) is said to be achievable for the DWTC if there exists a sequence of $(2^{nR},n)$ codes $\{\mathcal{C}_n\}$ such that

$$\lim_{n\to+\infty}\mathbf{P}(\mathcal{C}_n)=0 \tag{A.7}$$

$$\lim_{n\to+\infty}\frac{1}{n}\mathbf{E}(\mathcal{C}_n)\geq R_e. \tag{A.8}$$

The weak rate-equivocation region of a DWTC is

$$\mathcal{R}^{\mathrm{DWTC}}\triangleq\{(R,R_e):(R,R_e)\text{ is achievable}\}. \tag{A.9}$$

Finally, the weak secrecy capacity for the DWTC is

$$C_s^{\mathrm{DWTC}}\triangleq\sup_R\{R:(R,R)\in\mathcal{R}^{\mathrm{DWTC}}\}. \tag{A.10}$$

Note that if we specialize (A.8) for $R_e = R$ then (A.5) is satisfied since $(1/n)H(M|\mathcal{C}_n) = R$ for $n \to +\infty$. In this case all information transmitted at rate R is hidden from Eve.

Wyner's theorem characterizes secrecy capacity as follows:

$$C_s^{\mathrm{DWTC}} = \max_{p_X} I(X;Y|Z) = \max_{p_X}(I(X;Y) - I(X;Z)), \tag{A.11}$$

where p_X is the probability mass distribution of X. By introducing channel capacities $C_B = \max_{p_X} I(X;Y)$ and $C_E = \max_{p_X} I(X;Z)$ we have:

$$C_s^{\mathrm{DWTC}} \geq C_B - C_E, \tag{A.12}$$

where (A.12) holds with equality if channel transition probabilities are symmetric. In other words secret transmission is possible if Bob has a better channel than Eve. Also, randomness at the encoder plays a key role, as we see next.

A.1.1 Gaussian wiretap channel

The Gaussian wiretap channel (GWTC) is the continuous version of the DWTC and all the results seen until now can be extended also to the Gaussian case. The channels between Alice and Bob, Alice and Eve are additive white Gaussian noise (AWGN) channels and the transmitted symbols X_i are Gaussian random variables. The secrecy capacity of the GWTC is

$$C_s^{GWTC} = C_B - C_E. \tag{A.13}$$

A.2 Randomness for secrecy

The encoding function maps a message M and a realization of the local randomness $\mathcal{R}$ into a codeword X^n. $\mathcal{R}$ is also independent of M. Thus we can write

$$I(X^n; Z^n|\mathcal{C}_n) = I(M\mathcal{R}; Z^n|\mathcal{C}_n) \tag{A.14}$$

$$H(\mathcal{R}|M\mathcal{C}_n) = H(\mathcal{R}|\mathcal{C}_n) \tag{A.15}$$

and using the mutual information chain rule, the leakage can be written as

$$\frac{1}{n}\mathbf{L}(\mathcal{C}_n) = \frac{1}{n}I(X^n; Z^n|\mathcal{C}_n) - \frac{1}{n}H(\mathcal{R}|\mathcal{C}_n) + \frac{1}{n}H(\mathcal{R}|Z^n M\mathcal{C}_n). \tag{A.16}$$

Randomness, in practice, is exploited by encoding random bits, i.e. a *dummy* message M_d, alongside the original message M. The dummy rate can then be defined as

$$R_d = \frac{1}{n}H(M_d|\mathcal{C}_n) \tag{A.17}$$

and M_d can be used in place of R in (A.16). Moreover, by using jointly typical set decoding, it can be shown that the third term on the right in (A.16) can be made arbitrarily small. The leakage becomes

$$\frac{1}{n}\mathbf{L}(\mathcal{C}_n) \approx \frac{1}{n}I(X^n; Z^n|\mathcal{C}_n) - R_d \tag{A.18}$$

that is we can cancel out all the information leaked to Eve by *confusing* him with a properly designed random rate. Specifically, R_d should be picked as close as possible to C_E.

Appendix B

Quantization

In this appendix we recall the main definitions, models and results of the *quantization* theory. Reference for this subject can be found in [35] and [37].

B.1 Description of the quantization process

The aim of a quantizer is to permit the digital representation of a real signal that otherwise would require an infinite number of bits. A quantizer can be mathematically described as a map $\mathcal{Q}$ that takes as input a real value and yields as output another real value, but this time taken from a finite set $\mathcal{A}_q$:

$$\mathcal{Q} : \mathbb{R} \to \mathcal{A}_q. \tag{B.1}$$

Quantization comprises four elements:

- input signal sample $s(k)$, with $k \in \mathbb{N}$;
- a quantization function $\mathcal{Q}$;
- quantized signal $s_q(k) \in \mathcal{A}_q$, where the L elements $\mathcal{Q}_i$ of the set $\mathcal{A}$ are called *output levels*;
- codeword $c(k) \in \{0, 1, \ldots, L-1\}$ which represents the value of $s_q(k)$. The number of bits used for representing the quantized values is $\mathtt{b} = \lceil \log_2 L \rceil$.

The quantization map defines a partition of the the real set $\mathbb{R}$ with L disjoint intervals $R_i = (\tau_i, \tau_{i+1}]$ and assigns to each of these an output value $\mathcal{Q}_i$. The measure of the disjoint intervals are called quantization step and are denoted by Δ_i. In this way we introduce an error in the representation of $s(k)$ and it is defined by

$$e_q(k) = s_q(k) - s(k). \tag{B.2}$$

The quantization error is limited if the corresponding quantization step is finite (*granular* error), while if the quantization step is infinite (i.e. one of the extremes is $+\infty$ or $-\infty$) then the error is unbounded (*saturation* error).

If the input signal significantly differs from a constant value and the quantization steps are sufficiently small, we can make the following assumptions on $e_q(k)$ in the granular region.

- $e_q(k)$ is a white process:

$$E[e_q(k)e_q(k-n)] = \begin{cases} \mathtt{M}_q & n=0 \\ 0 & n \neq 0 \end{cases}, \tag{B.3}$$

where $\mathtt{M}_q$ is the statistical power of $e_q(k)$.

- It is uncorrelated with the input signal:

$$E[s(k)e_q(k-n)] = 0 \quad \forall n. \tag{B.4}$$

- It has a uniform distribution over each Δ_i.

A measure of the performance of the quantization process is the signal-to-quantization error ratio:

$$\Lambda_q = \frac{E[s^2(k)]}{E[e_q^2(k)]}. \tag{B.5}$$

B.2 Uniform quantizer

A uniform quantizer is a quantizer with $L = 2^{\mathsf{b}}$ reconstruction levels and constant step size Δ in the granular region. The granular region is the interval $[-\tau_{sat}, \tau_{sat}]$ such that $2\tau_{sat} = L\Delta$ Assuming the input signal has zero mean and variance σ_s^2 the optimum design for the quantizer is obtained with the following algorithm.

1. Determine τ_{sat} such that the saturation probability is sufficiently small. In this way $e_q(k)$ is approximately always granular and it can be shown that

$$\mathtt{M}_q \simeq \frac{\Delta^2}{12}. \tag{B.6}$$

2. Choose L so that Λ_q assumes a desired value.

3. Given L and τ_{sat} we have

$$\Delta = \frac{2\tau_{sat}}{L}. \tag{B.7}$$

The highest Λ_q is obtained for a uniform input distribution: it can be shown that the uniform quantizer is the optimum quantizer for this kind of input.

B.3 Non-uniforrm quantizers

If the input signal is not uniform, then the uniform quantizer is suboptimum. We can then design non-uniform quantizers that better fit the input distributions. There are different design procedure available. A class of these aims at minimizing the statistical power of $e_q(k)$ (minimum mean-square error criterion). By deriving and setting to 0 $\mathtt{M}_q$ with respect to the reconstruction values Q_i and with respect to τ_i, we get an expression of the optimum parameters as a function of the input signal distribution $p_s(a)$:

$$\tau_i = \frac{\mathcal{Q}_i + \mathcal{Q}_{i+1}}{2} \tag{B.8}$$

$$\mathcal{Q}_i = \frac{\int_{\tau_{i-1}}^{\tau_i} a p_s(a) da}{\int_{\tau_{i-1}}^{\tau_i} p_s(a) da}. \tag{B.9}$$

Max algorithm and *Lloyd* algorithm are two algorithms that, after a random initialization, iteratively apply the previous equations until a stopping condition (e.g. a desired maximum value of distortion) is met.

Bibliography

[1] http://www.gps.gov/applications

[2] http://www.esa.int/Our_Activities/Navigation/Galileo/Galileo_s_clocks

[3] https://news.utexas.edu "Austin researchers successfully spoof an 80 million yacht at sea".

[4] https://www.newscientist.com "Ships fooled in gps spoofing attack suggest russian cyberweapon".

[5] https://www.gsa.europa.eu/european-gnss

[6] http://ec.europa.eu/growth/sectors/space/galileo/history_en

[7] Kaplan, Elliott, and Christopher Hegarty. Understanding GPS: principles and applications. Artech house, 2005.

[8] http://www.navipedia.net/index.php/User_Guides

[9] Leva, J., "An Alternative Closed Form Solution to the GPS Pseudorange Equations," Proc. of The Institute of Navigation (ION) National Technical Meeting, Anaheim, CA, January 1995.

[10] Psiaki, Mark L., and Todd E. Humphreys. "GNSS spoofing and detection." Proceedings of the IEEE 104.6 (2016): 1258-1270.

[11] White, Nathan Alan, Peter S. Maybeck, and Stewart L. DeVilbiss. "Detection of interference/jamming and spoofing in a DGPS-aided inertial system." IEEE Transactions on Aerospace and Electronic Systems 34.4 (1998): 1208-1217.

[12] Humphreys, Todd E., et al. "Assessing the spoofing threat: Development of a portable GPS civilian spoofer." Proceedings of the ION GNSS international technical meeting of the satellite division. Vol. 55. 2008.

[13] Humphreys, Todd E. "Detection strategy for cryptographic GNSS anti-spoofing." IEEE Transactions on Aerospace and Electronic Systems 49.2 (2013): 1073-1090.

[14] Anon, "Vulnerability assessment of the transportation infrastructure relying on the global positioning system," J.A. Volpe National Transportation Systems Center, 2001.

[15] Cavaleri, A., Motella, B., Pini, M., Fantino, M. (2010, December). Detection of spoofed GPS signals at code and carrier tracking level. In Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on (pp. 1-6). IEEE.

[16] Wesson, K. D., Shepard, D. P., Bhatti, J. A., Humphreys, T. E. (2011, September). An evaluation of the vestigial signal defense for civil GPS anti-spoofing. In Proceedings of the ION GNSS Meeting.

[17] D. M. Akos, "Who's afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (AGC)," Navigation, vol. 59, no. 4, pp. 281–290, 2012.

[18] Cuntz, Manuel, et al. "Lessons learnt: The development of a robust multi-antenna GNSS receiver." Proc. ION GNSS 2010 (2010): 21-24.

[19] Axell, Erik, Mikael Alexandersson, and Tore Lindgren. "Results on GNSS meaconing detection with multiple COTS receivers." Localization and GNSS (ICL-GNSS), 2015 International Conference on. IEEE, 2015.

[20] Axell, E., Larsson, E. G., Persson, D. (2015, April). GNSS spoofing detection using multiple mobile COTS receivers. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on (pp. 3192-3196). IEEE.

[21] J. T. Curran, M. Bavaro, P. Closas, M. Anghileri, M. Navarro, B. Schotsch, and S. Pfletschinger, "On the Threat of Systematic Jamming of GNSS," in Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), (Portland, Oregon), 2016.

[22] J. T. Curran and C. O'Driscoll, "Message Authentication as an Anti-Spoofing Mechanism," Working Paper, June 2017.

[23] G. Caparra, S. Ceccato, N. Laurenti, J. Cramer, and C. J. Walter "Feasibility and Limitations of Self-Spoofing Attacks on GNSS Signals with Message Authentication"

[24] P. Levin, D. De Lorenzo, P. Enge, and S. Lo, "Authenticating a signal based on an unknown component thereof," Patent 7,969,354 B2, Jun. 2011.

[25] B. W. O'Hanlon, M. L. Psiaki, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "Real-time GPS spoofing detection via correlation of encrypted signals," Navigation, vol. 60, no. 4, pp. 267–278, 2013.

[26] Pozzobon, Oscar, et al. "Anti-spoofing and open GNSS signal authentication with signal authentication sequences." Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on. IEEE, 2010. APA

[27] G. Caparra, C. Wullems, S. Ceccato, S. Sturaro, N. Laurenti, O. Pozzobon "Navigation Message Authentication Schemes for GNSS System", InsideGnss, working paper September/October 2016.

[28] A. J. Kerns, K. D. Wesson, and T. E. Humphreys, “A blueprint for civil GPS navigation message authentication,” in Proc. IEEE/ION PLANS Meeting, Monterey, CA, USA, May 2014, pp. 262–269.

[29] L. Scott, “Anti-spoofing and authenticated signal architectures for civil navigation systems,” in Proc. ION GPS/GNSS, Portland, OR, USA, 2003, pp. 1543–1552.

[30] Caparra, Gianluca, et al. "Evaluating the security of one-way key chains in TESLA-based GNSS navigation message authentication schemes." Localization and GNSS (ICL-GNSS), 2016 International Conference on. IEEE, 2016.

[31] Liu, Yongsheng, Jie Li, and Mohsen Guizani. "PKC based broadcast authentication using signature amortization for WSNs." IEEE Transactions on Wireless Communications 11.6 (2012): 2106-2115.

[32] A. D. Wyner, "The wiretap channel", Bell Syst. Tech. J, vol. 54, pp. 1355-1387, Oct. 1975.

[33] M. Bloch, J. Barros, "Physical-Layer Security, From Information Theory to Security Engineering".

[34] European GNSS (Galileo) Open Service "Signal-in-space control document".

[35] N. Benvenuto, G. Cherubini, "Algorithms for Communications Systems and their Applications".

[36] T. Erseghe, "Channel Coding", Padova University Press.

[37] K. Sayood, "Introduction to data compression", Morgan Kaufmann, 2012.