Università degli Studi di Padova

DEPARTMENT OF INFORMATION ENGINEERING

Ph.D. Course in INFORMATION ENGINEERING

SCIENCE AND INFORMATION TECHNOLOGY CURRICULUM

XXXV CYCLE

Advanced Physical Layer Security Techniques for Non-Terrestrial Communications

Coordinator
PROF. ANDREA NEVIANI

Supervisor
PROF. NICOLA LAURENTI
UNIVERSITÀ DEGLI STUDI DI PADOVA

Ph.D. Candidate
FRANCESCO ARDIZZON

ACADEMIC YEAR 2021/2022

"*In the midst of chaos, there is also opportunity.*"

Sun Tzu, The Art of War

Abstract

Francesco Ardizzon

Advanced Physical Layer Security Techniques for Non-Terrestrial Communications

The broad use of non-terrestrial communications makes it necessary to investigate specifically targeted security solutions. Indeed, these communications are mainly wireless. However, differently from the wired counterpart, the wireless communication channel is a broadcast channel by nature, therefore it is vulnerable to many threats: any malicious user can disrupt the communication by using a *jamming* attack, intercept the signal to disclose its content or information about the transmitter, or lead a *spoofing attack* by generating a counterfeit signal or by tampering the transmitted signal.

In many applications, it is not possible to rely on cryptography: for instance, cryptography-based solutions have a considerable computational cost, thus they may not be suited for wireless sensor network (WSN) applications, where we want to reduce the user energy consumption, such as in an Internet of things (IoT) context. Thus, we resort to physical-layer security (PLS) approaches. Physical layer authentication relies on the collection of the observation about the channel characteristics (e.g., features of the channel impulse response) to tell apart transmissions by legitimate network members from the ones by impersonating attacker. Moreover, PLS mechanisms are also *unconditionally secure*, since the mechanism security is not provided by a computationally hard problem. On the other hand, since these techniques rely on the channel model it is, in general, complex to generalize solutions and each context need to be separately analyzed.

This Thesis focuses on the development of physical layer authentication for non terrestrial communications, focusing on global navigation satellite systems (GNSSs) and underwater acoustic networks (UWANs). GNSS services are used to provide positioning and timing. However, these services do not (necessary) rely on the data content of GNSS but on the properties of the signals themselves, i.e., phase and Doppler frequency. Indeed, PLS can be used to provide authentication at signal level by making the spreading code (or part of it) unpredictable. The contributions of this Thesis are multiple. We propose a novel network-aided authentication protocol, proposing also a verification based on the generalized likelihood ratio test (GLRT). To show its robustness, the scheme is tested against several attacks: among others, we also consider the security code estimation and replay (SCER) attack and the internal code attack. Next, we focus on the problem of position, velocity, and time (PVT) assurance, where we propose a series of consistency checks to enlarge the set of trusted signals to be used for the PVT. We focus then on the problem of providing an authenticated but robust timing service, relying only on Galileo's commercial authentication service (CAS). Finally, we address the problem of message scheduling in GNSS: considering, for instance, an authentication service that need to disseminate a digital signature over the GNSS channels, we study both single and multi-round scheduling solutions that aim at minimizing the maximum and the average latency.

In the last part of the Thesis, we tackle the problem of physical layer authentication for UWANs: underwater acoustic channels (UWACs) are known to decorrelate easily in space, and to have a limited time coherence, thus by extracting relevant channel features, it is possible to distinguish a packet transmitted by a legitimate transmitter from the one sent by a potential attacker. Indeed, it is possible to improve the classification procedure by having multiple (trusted) cooperating sensors, where the local information is processed and shared with the others. We address this problem by using machine learning (ML) techniques. Several aspects will be investigated: the availability of the attacker channel's observations during the training phase; how the amount of information shared by each user influences the overall authentication process; how to tackle the problem of authentication for time varying channel, e.g., when the transmitter is mobile.

Acknowledgements

I want to take this chance to thank all the colleagues and friends the accompanied through this three-year journey.

First and foremost, I would like to thank my supervisor, Prof. Nicola Laurenti who I admire for his expertise and passion for research. I will be forever indebted to you for giving this life-changing opportunity and encouraging through this journey (and for about one thousand coffees).

Next, I wish to thank Prof. Stefano Tomasin for all the stimulating discussions and your insightful feedback pushed to grow as a researcher.

I also wish to thank both my past and current Ph.D. colleagues which I met in the DEI's laboratories: notably, I wish to thank the past members and mentors of the GNSS group, Leonardo C. and Francesco F., all the guys from the Lab 219, in particular Alberto R. and Laura C.. Spending this time with you made this experience even more interesting (and fun).

Then I want to thank all the colleagues and friends I met during my aboard period at ESA-ESTEC. A special thanks goes to my supervisor Gianluca Caparra, for sharing with me his experience and love for research.

Finally, I want to dedicate this milestone to my family: without the support of my father, Fabio, my mother, Daniela and my brother, Lorenzo, this would not have been possible. My sincere gratitude goes to my you.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

ACAS	Assisted commercial authentication service
AE	Autoencoder
ARQ	Automatic repeat request
AUV	Autonomous underwater vehicle
AWGN	Additive white Gaussian noise
BGD	Broadcast group delay
BILP	Binary integer linear programming
BOC	Binary offset carrier
BP	Bounded PER
BPSK	Binary phase shift keying
BQP	Bounded-error quantum polynomial
CAS	Commercial authentication service
CDF	Cumulative distribution function
CDMA	Code division multiple access
CHIMERA	Chips-message robust authentication
CLDAE	Combined local decision (LD) and AE
CRC	Cyclic redundancy check
DSM	Digital signature message
ECEF	Earth-centered earth-fixed
ECS	Encrypted code sequence
EKF	Extended Kalman filter
ENU	East, North, Up
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
EUSPA	European Union Agency for the Space Programme
FA	False alarm
GGTO	GPS to Galileo time offset
GLRT	Generalized likelihood ratio test
GNSS	Global navigation satellite system
GSA	European GNSS Agency
GSC	GNSS service center
GST	Galileo system time
HAS	High accuracy service
HPL	Horizontal protection level
ID	Identification number
ILP	Integer linear programming
IMU	Inertial measurement unit
IoT	Internet of things
IP	Internet protocol
ISCB	Inter-system clock bias
KDE	Kernel density estimation
LAN	Local area network

LC	Linear combination
LD	Local decision
LEO	Low Earth orbit
LOS	Line-of-sight
LRT	Likelihood ratio test
LS	Least squares
LS-SVM	Least squares support vector machines
MA	Moving average
MAC	Message authentication code
MAP	Maximum a posteriori criterion
MARP	Maximization of the Average Number of Different Received Packets
MARP-MC	Maximization of the Average Number of Different Received Packets among Maximum Coverage Solutions
MD	Missed detection
MEO	Medium Earth orbit
MID	Max inter-group distance
MILP	Mixed-integer linear programming
MIN-MAX	Minimization of the Maximum Latency
MIW	Min inter-group weight
ML	Machine learning
MORE GOSSIP	More GNSS Open Service Signal Integrity Protection and Authentication at the Physical Layer
MP	Min-max PER
MSE	Mean squared error
NLOS	Non-line-of-sight
NMA	Navigation message authentication
NME	Navigation Message Encryption
NN	Neural network
NP	Nondeterministic polynomial
NP	Neyman-Pearson
NTP	Network time protocol
OC-SVM	One-class support vector machine
OSNMA	Open service navigation message authentication
OTP	One-time pad
PATROL	Position Authenticated Tachograph for OSNMA Launch
PDF	Probability density function
PER	Packet error rate
PLS	Physical-layer security
PNT	Position navigation and timing
PNT-CRC	Position Navigation and Timing Cyber-Response Center
ppm	Parts per million
PRN	Pseudorandom noise
PTP	Precision time protocol
PVT	Position, velocity, and time
r.v.	Random variable
RECS	Re-encrypted code sequence
ReLu	Rectified linear unit

RMS	Root mean square
RNG	Random number generator
ROC	Receiver operating characteristic
RS	Random scheduling
RSA	Rivest–Shamir–Adleman
SAR	Search and rescue service
SBAS	Satellite-based augmentation systems
SBF	Septentrio binary format
SCA	Spreading code authentication
SCE	Spreading code encryption
SCER	Security code estimation and replay
SSA	Single-sensor authentication
SSP	Sound speed profiles
SSSC	Spread spectrum security code
SV	Space vehicle
SVM	Support vector machines
TCP	Transmission control protocol
TCXO	Temperature-controlled crystal oscillator
TDMA	Time-division multiple access
TESLA	Timed-efficient stream loss-tolerant authentication
TMBOC	Time-multiplexed BOC
UERE	User equivalent ranging error
ULS	Uplink station
UTC	Coordinated universal time
UWAC	Underwater acoustic channel
UWAN	Underwater acoustic network
VAL	Vertical alert limit
VCTCXO	Voltage-controlled and temperature-controlled crystal oscillator
VEL	Vertical error level
VIR2EM	VIrtualization and Remotization for Resilient and Efficient Manufacturing
VPL	Vertical protection level
WSN	Wireless sensor network

Chapter 1

Introduction

Nowadays, wireless communications play an essential role in many fields. In addition, in recent years there are new emerging applications that rely on the wireless communication to operate. It is, therefore, natural to discuss and invest on security for this field: indeed, we want the wireless communication-based service to guarantee *confidentiality*, *integrity protection*, *availability* and *privacy*. On the other hand, differently from wired communications, wireless communications are by nature prone to many threats: of course, since the medium is easily accessible, any malicious user can disrupt the communication by using a *jamming* attack, intercept the signal to disclose its content or information about the transmitter, or lead a *spoofing attack* by generating a counterfeit signal or by tampering the actual transmitted signal.

To counter these threats, a possible solution is to rely on cryptographic means. This approach has however several limitations: many cryptography-based solutions have a considerable computational cost, thus they may not be suited for many wireless applications, where we want to limit each user energy consumption, such as in an wireless sensor network (WSN) used in Internet of things (IoT) context. Moreover, these solutions often rely on computationally hard problems, such has Rivest–Shamir–Adleman (RSA) algorithm [7] relying on the integer factorization problem or the Diffie-Hellman algorithm [8] on the discrete logarithm problem. These are in fact both nondeterministic polynomial (NP) problems: however, quantum cryptography algorithms, such as the Shor's algorithm [9], allows these problems to be solved in a bounded-error quantum polynomial (BQP) time.

Physical-layer security (PLS) techniques are instead able to counteract these issues: the security of these rely on the channels characteristics themselves thus, during the verification process, no additional computation is needed, hence they are particularly suitable in networks with communication or computing constraints. For a formal overview of PLS based techniques, refer to [10]. Moreover, PLS mechanisms are also *unconditionally secure*, i.e., "[...] a system which can resist any cryptanalytic attack, no matter how much computation" [8]. For these reasons, PLS are also *quantum resistant*, since they can withstand quantum computing attacks, such as the ones based on Shor' algorithm.

PLS approaches have been considered both for authentication and other security primitives, such as key generation. Physical layer authentication relies on the collection of channel characteristics (e.g., features of the channel impulse response) to tell apart transmissions by legitimate network members from transmissions by an impersonating attacker. Physical layer-based secret key generation protocols rely on (correlated) observations of the channel to compute a secret key.

Since theses techniques rely on the channel model, it may be complex to generalize solutions and each context must be separately analyzed. This Thesis will focus on authentication for global navigation satellite system (GNSS) and underwater acoustic network (UWAN). GNSS services are used to provide positioning and timing. However, these services do not (necessarily) rely on the data content of GNSS: for instance, in *snapshot positioning* mode a GNSS receiver compute the position, velocity, and time (PVT) solution without performing neither signal tracking nor data demodulation. In this context, cryptographic techniques only indirectly protect the users from spoofing attacks: by making the data signature unpredictable, it is hard for the attacker to guess the correct symbol to be used within the spoofing signal. However, as we will describe more in detail in the next Sections, cryptography-based mechanisms are vulnerable to many attacks, such as meaconing, where the attacker simply delays the whole signal, e.g., by using a simple signal repeater, or the security code estimation and replay (SCER) attack, where the attacker estimate the symbol from the legitimate broadcast signal. GNSS providers are proposing signal-level authentication by using spreading code encryption (SCE) or spreading code authentication (SCA), where the whole signal spreading code or part of it, is signed and therefore unpredictable. Still, signal level authentication techniques have several limitations. First, they trade the authentication capabilities with performance and availability: for instance, by using a SCE scheme a receiver that does not have the actual spreading code cannot perform positioning; secondly these mechanisms are designed to provide a periodic (and delayed) verification of only a subset of the typically used pseudoranges: indeed, limiting the set of usable signals may lead to issues in terms of availability and accuracy. In the GNSS context the effort of this Thesis is to improve the current state of art in signal level authentication, dealing with all the aforementioned issues.

Then the problem of physical layer authentication for UWANs was tackled. Underwater acoustic channels (UWACs) are known to decorrelate easily in space, and to have a limited time coherence [11, 12]; thus, by extracting relevant channel features from the statistics of the channel, it is possible to distinguish a packet transmitted by a legitimate transmitter from the one sent by a potential attacker. Indeed, by combining the classification results for the checks of several (trusted) cooperating sensors, it is possible to improve the overall classification procedure. In this underwater communications context, the effort was to study the impact of the several aspects on the classification procedure: the availability of the attacker channel's observations during the training phase; how the amount of information shared by each user influences the overall authentication process; how to tackle the problem of authentication for time varying channel, e.g., when the transmitter is mobile. In the next Section a detailed description of each contribution is provided to the reader.

1.1 Contributions

This doctoral Thesis is based upon on the research work done during three years of Ph.D. studies. In this Section a brief description of the research activity is presented, also including projects and papers that did not find room to be presented in this Thesis. During my Ph.D., I joined the following projects.

More GNSS Open Service Signal Integrity Protection and Authentication at the Physical Layer (MORE GOSSIP) This project funded by the European Space Agency (ESA), focused on the authentication and integrity protection of GNSS Open Service; I joined the project on its final phase, helping with the implementation of the Galileo and GPS signal generator and software receiver. I was mainly involved on the design and the implementation of network aided authentication protocol and the message scheduling strategies, that will be described in respectively in Chapter 2 and 5.

Position Authenticated Tachograph for OSNMA Launch (PATROL) This project was funded by the European GNSS Agency (GSA) (now European Union Agency for the Space Programme (EUSPA)) to aid development, supply, and testing of a Galileo open service navigation message authentication (OSNMA); my contribution has been to statistically characterize the performance of mass-market clocks used in commercial GNSS receivers. All the currently proposed authentication protocols (e.g., OSNMA) relies on an autonomous source of timing: thus, the task was to investigate how long a clock used in a typical GNSS receiver, usually a temperature-controlled crystal oscillator (TCXO), can be considered to be reliable. A first part of the results was published in [13].

Position Navigation and Timing Cyber-Response Center (PNT-CRC) The aim of this ESA-funded project is to provide a service that would allow to test the security performance of GNSS receivers against attacks such as jamming and spoofing. I was involved in the last part of the project, focusing on the design of the models and the tests. A first batch of results has been presented in [14].

VIrtualization and Remotization for Resilient and Efficient Manufacturing (VIR2EM) , This project is funded by Regione Veneto. In this context, we proposed a robust and secure authentication protocol which relied only on Galileo commercial authentication service (CAS) authenticated feature to provide a secure and robust timing service. Results have been published [15] and will be presented in Chapter 4.

NATO SAFE U-COMM The task of the project is to investigate and develop authentication and secret key generation protocols for UWAC. I was involved in both on the secret key generation and authentication: for the first, we proposed a protocol for UWAN where two users exploit a UWAN to generate a secret key, by broadcasting messages, and using the measured round-trip delay as source of (correlated) randomness [16]. For UWAC authentication, we proposed several strategies [17–19] where a user exploit the channel features measured by sensors in a UWAN to distinguish between legitimate and attacker transmission. The results for the authentication, will be presented in Chapter 6.

Additionally, I spent a six months period abroad as a visiting scientist at European Space Research and Technology Centre (ESTEC) in Noordwijk (NL). My research activity focused on PVT assurance techniques: by relying on authenticated signals as anchors, we enlarge the set of *trusted* signals, obtaining a trusted and more accurate PVT solution. Part of the result have been presented in [20] and will be discussed in Chapter 3.

1.2 Thesis Outline

The Thesis is structured as follows:

- Chapter 2 first outlines the main solutions for GNSS SCE and SCA; next, it presents the proposed network aided Signal Level Authentication technique proposed in the context of MORE GOSSIP, proposing a generalized likelihood ratio test (GLRT) for the authentication. Moreover, we analyzed and tested the performance of several attacks that may target any SCA (or SCE) scheme.

- Chapter 3 presents the strategy on PVT assurance, relying on authenticated signals to enlarge the test of trusted signals that may be used for position navigation and timing (PNT). The performance of the proposed checks is tested also by using an experimental dataset.

- Chapter 4 discusses the secure timing protocol proposed for VIR2EM, which rely only on CAS and OSNMA authenticated feature to provide timing. To provide an additional security layer, we propose several security checks, based either on clock monitoring or on innovation testing.

- Chapter 5 presents the results for the GNSS message scheduling, considering both single and multiple-round message scheduling.

- Chapter 6 describes the solution for authentication for UWAN, relying on machine learning (ML)-based solution for distinguishing legitimate user from the attacker. In particular, we discuss the problem of one-class versus two-class authentication; the impact of a bottleneck in the communication among the sensors; how to authenticate mobile users.

- Chapter 7 draws some conclusions of the work presented on this Thesis.

1.3 Summary of Notation

Authentication has been studied in the literature from its theoretical foundation as a *hypothesis testing* problem [21, 22]. Its general performance limits were investigated in [23, 24].

Thus, in all the considered contexts we will formulate the authentication problem under the framework of hypothesis testing. Considering the received signal φ, we formulate the authentication problem as a binary hypothesis test, where the two hypotheses are:

- $\mathcal{H}_0$ (legitimate): φ was transmitted by the legitimate user, namely, Alice and,

- $\mathcal{H}_1$ (non legitimate): φ was not transmitted by Alice.

Called $\mathcal{H} \in \{\mathcal{H}_0, \mathcal{H}_1\}$ the true class of the received signal, the decision that the receiver, namely Bob, makes based on the observations is $\hat{\mathcal{H}} \in \{\hat{\mathcal{H}}_0, \hat{\mathcal{H}}_1\}$.

We account for two cases of misclassification, i.e., false alarms (FAs), where the transmitter considers a signal transmitted by Alice as fake, and missed detections (MDs), where Bob considers a signal from the attacker, Eve, as legitimate. The FA probability is defined as $p_{\mathrm{FA}} = \mathbb{P}(\hat{\mathcal{H}} = \mathcal{H}_1 | \mathcal{H} = \mathcal{H}_0)$ and the MD probability is defined as $p_{\mathrm{MD}} = \mathbb{P}(\hat{\mathcal{H}} = \mathcal{H}_0 | \mathcal{H} = \mathcal{H}_1)$.

Throughout this Thesis, lowercase boldface letters denote vectors (e.g., $\boldsymbol{x}$); boldface capital letters matrices (e.g., $\boldsymbol{X}$); lowercase medium letters denote instead variables. Vectors are column vectors unless otherwise specified. Finally we denote a $x \sim \mathcal{N}(\mu, \sigma^2)$ a Normal (Gaussian) random variable (r.v.) with mean μ and variance σ^2 and as $x \sim \mathcal{U}([a, b])$ a Uniform r.v. in the (closed) interval $[a, b]$.

Chapter 2

Signal Level Authentication Techniques for GNSS

2.1 Introduction

In recent years, GNSSs are acquiring a key role for positioning, navigation, and timing services in multiple fields, from telecommunications to transportation and financial services. For example, also due to the progressive decrease in size and price, we often find that GNSS receivers are commonly used in IoT scenarios for both basic positioning and network synchronization. Moreover, most GNSSs also provide additional services like the Galileo search and rescue service (SAR). On par with this growth, there is also a rise on the number of works showing that it is feasible to implement effective spoofing attacks even by using inexpensive software-defined radios, e.g., in [25]: hence an attacker would be able to spoof a legitimate user, forcing him to go on a unwanted location, or even to self-spoof [26], e.g., for cheating a speed limit on a smart tachograph. Hence GNSSs need to be paired with authenticity and integrity protections mechanisms that offer protection against spoofing attacks with a minor impact on the overall performances.

Two main paradigms for GNSS anti-spoofing techniques are *data-level* security and *signal level* security. Data-level security mechanism are cryptographic techniques that make the message unpredictable making it hard for the potential spoofer to predict exactly the data symbol. The receiver is then able to distinguish between legitimate and spoofer signal during data demodulation. Thus, data-level security does not directly impact signal acquisition and tracking[1]. Still, data-level mechanisms do directly protect the signal features that are used for navigation and may be vulnerable to attack such SCER.

On the other hand, both SCA and SCE provide security at signal level, respectively watermarking or totally encrypting the pseudorandom noise (PRN) spreading code: attacks like SCER has limited success against these techniques. Different examples of schemes based on signal level security can be found in [28–32]. On the other hand, some of these techniques may have an impact on both acquisition and tracking, leading to a possible signal degradation for users that do not know the secure PRN in advance.

In this chapter, we describe a general framework for signal-level protocols, where the information necessary for the receiver to verify the signal authenticity, that we call *share*, is distributed to each user via a secure side channel with limited rate yet with higher rate than the GNSS, typically a wide area network: in principle, the amount of distributed information will depend on the receiver conditions, (e.g., carrier-to-noise ratio, C/N_0). Hence distinct information shares will be distributed to different receivers. More in detail, we propose two modes, the *simultaneous authentication*

[1] For the basics operations of a GNSS receiver and its implementation refer to [27].

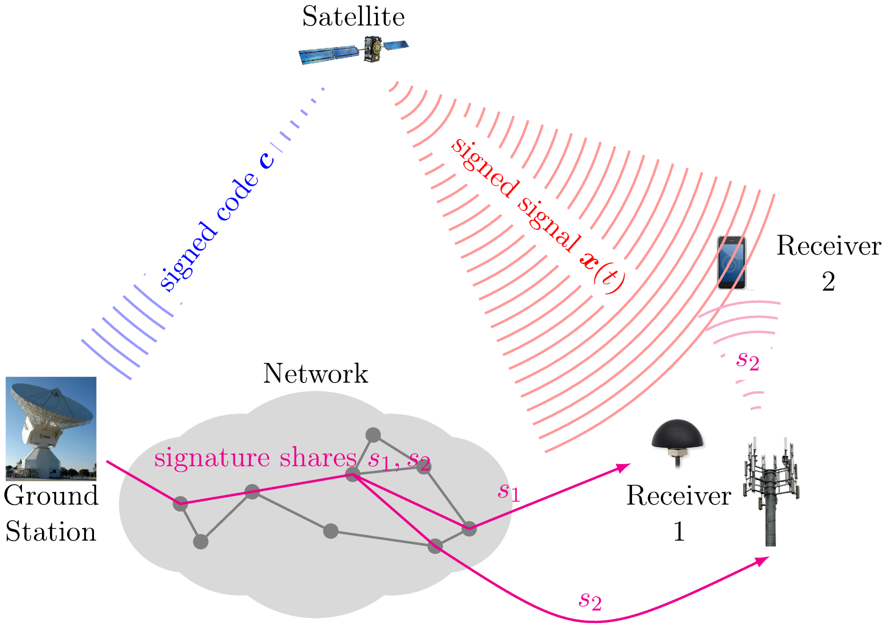

Satellite
signed code $\boldsymbol{c}$
signed signal $\boldsymbol{x}(t)$
Receiver 2
s_2
Network
Ground Station
signature shares s_1, s_2
s_1
Receiver 1
s_2

FIGURE 2.1: Pictorial representation of the proposed signature and verification protocol

mode, where the signed code is already (securely) provided to the receivers before the acquisition, and the *delayed authentication*, where the receiver performs all the correlations using the open code but stores the received signal samples; later it will receive the share from the network allowing the verification of the authenticity of the previously computed position. Here we investigate what are the performance and the limits of such a protocol considering several attacks: among them, a SCER attack that observes the signal and tries to estimate the signed code. We test the performance of the proposed solution by using both analytical and simulation data: the latter are obtained using an implementation of the protocol on a signal simulator and a software receiver, developed for the MORE GOSSIP project by our research group. A pictorial representation of the proposed system is given in Figure 2.1.

The remainder of the chapter is organized as follows. Section 2.2 gives a brief overview of the state of art for GNSS authentication. Section 2.3 reports a summary of the notation used throughout the paper. Section 2.4 introduces the system model. In Section 2.5 we report the proposed authentication protocol describing in detail each block. Section 2.6 describes attack models, giving a statistical analysis of each attack. Section 2.7 collects the numerical results. Section 2.8 draws some conclusions.

2.2 Related State of Art

In this section we briefly describe the main techniques for authenticating GNSS signal, data-level, and signal-level security. Still, a more general survey on signal security on GNSS can be found in [33]. First, we will focus on data-level security, considering Galileo's OSNMA; later, we will discuss signal-level security, considering both GPS's chips-message robust authentication (CHIMERA) and the Galileo's CAS.

2.2.1 Data-Level Security: OSNMA

Navigation message authentication (NMA) techniques aim to ensure the authenticity of the content of the navigation messages, providing the user with the integrity protection of data. Among these techniques, OSNMA is the data authentication function for the open Galileo E1B signals [34, 35] in which the message transmitted by the satellites is interleaved with authentication data generated through broadcast authentication protocol timed-efficient stream loss-tolerant authentication (TESLA) [36], suitably adapted for optimal transmission via Galileo [37, 38]. In particular, the OSNMA data will be transmitted in the (currently called) Reserved 1 I-NAV message field [39]. Thus on each page 40 bits are provided for OSNMA. Among these 32 bit are reserved for the digital signature message (DSM) while the last 8 bit will be used for the message authentication code (MAC). The TESLA protocol employs a one-way chain shared by Galileo satellites with a public root key. The keys in the chain are used in reverse order to generate MACs. Keys are then shared (always in reverse order) in broadcast mode with a delay of a few seconds. The receiver can verify the MACs as soon as it becomes aware of the key.

As mentioned at the start of the chapter, the advantage of both Navigation Message Encryption (NME) and NMA techniques is that it provides security against generation attacks without having a direct impact on the receiver performance: this means that, for instance receiver that do not use OSNMA can still perform navigation and timing as if OSNMA was not enabled by the system.

One the other hand, data-level mechanism can still be targeted by SCER attacks: since in GNSS the data rate at which the message bits are transmitted is much lower than the chip rate, an attacker with a sufficiently high C/N_0 can estimate the data before the legitimate receiver and use the estimation in the forged signal itself. On the other hand, as we will discuss in Section 2.6.4, signal-level mechanism cannot be targeted by SCER-like attacks, unless the same signature is used multiple times.

2.2.2 Ranging-Level Security: CHIMERA and CAS

SCA and SCE techniques have been proposed for military and civilian use. Focusing on public signals, a SCE approach was proposed in [40]; a similar SCA technique was proposed in [28], where short sequences called spread spectrum security codes (SSSCs) were interleaved with the public spreading code.

For GPS, CHIMERA was proposed: this mechanism works both on data and ranging level. Navigation message data are protected by a digital signature while signal ranging are instead protected by using watermarking techniques, thus *authentication markers* are placed within the L1 C/A spreading code. By using CHIMERA, a receiver checks the authenticity of the signal by verifying if the chips of the received signal have the correct polarity. CHIMERA envisions both a *slow* and a *fast* authentication mode. In the former, the authentication markers are disclosed in the navigation messages itself, therefore the receiver must receive the whole messages before verifying the code (3 min). With the fast authentication mode, the receivers rely on out of band signals to receive the markers, thus the authentication can be accomplished every 6 s.

Galileo CAS was recently proposed: in particular, assisted commercial authentication service (ACAS) was presented in [41, 42]. This authentication mechanism works jointly with OSNMA, using the OSNMA keys to generate the digital signature. ACAS is a SCE mechanism, therefore the E6C PRN spreading codes are neither short nor periodic sequences, but are generated by the system as a stream, known as encrypted code sequence (ECS). Part of the ECS is re-encrypted using the TESLA keys employed by the OSNMA protocol, and disseminated with the E1 open signal, generating the re-encrypted code sequence (RECS). The RECS are stored and published at predefined times on servers accessible to the public, such as the GNSS service center (GSC). Together with RECS, the server also publishes additional useful files for PVT computation, e.g., the broadcast group delay (BGD) for the E1–E6 bands. Once the RECS are retrieved from the server, the user can decrypt them by using the corresponding TESLA key, obtaining the related ECS. Lastly, the ECS is tested against previously stored samples received from the E6C signal, allowing for the user to verify the authenticity of the received signals.

The RECS lengths are defined by the number of chips in these sequences, which is one of the key parameters in ACAS design as it determines the duration of the signal fragment used in correlation during the acquisition phase. Together with the size of the bins used for the Doppler frequency search, they define the acquisition search space and thereby the ability to find correlation peaks from which the pseudoranges and the authenticated PVT solution are computed. Another key parameter in ACAS is the distance between two consecutive RECS sequences, which determines how often the receiver can compute an authenticated solution. The default ACAS operating mode is snapshot mode, since no navigation message and thus no ephemeris are transmitted on E6. Figure 2.2 summarizes the ACAS operations at the receiver side.

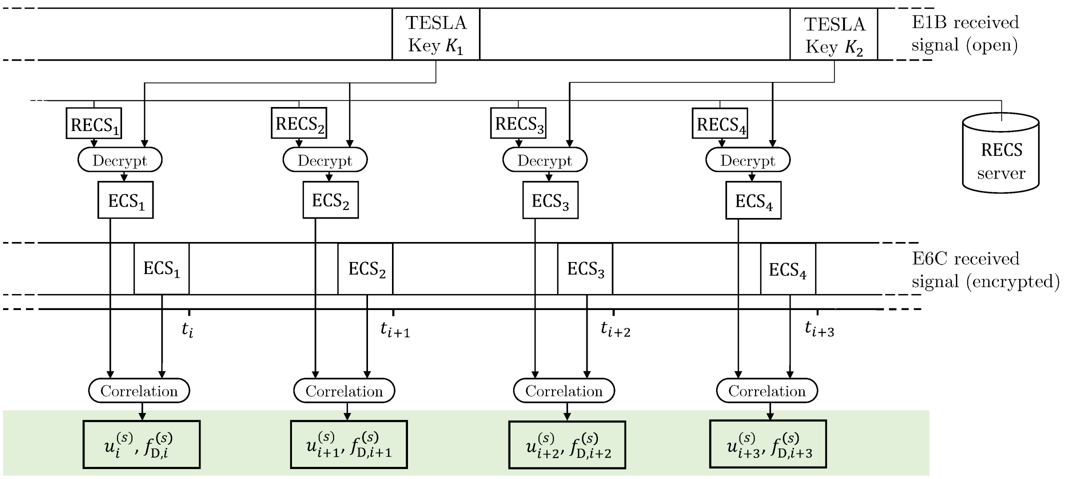


FIGURE 2.2: Summary of ACAS operations at the receiver side for signal transmitted by satellite, s.

Differently from the data-level mechanisms, these schemes are resilient to SCER attacks, since the energy collected from a single chip is typically not sufficient to estimate on the fly the chip polarity itself [43]. However, users that do not possess the code in advance experience performance degradation. So, differently from NMA, the security mechanism does have an impact on the receiver performance.

In [44], the authors present a framework to evaluate the trade-off between the security of the mechanism and amount of information to be disclosed to the receivers: transmitting more information to the receivers lower the correlation loss but requires more resources, since they are required to have a higher download rate for a possibly long period of time; conversely, more information about the secure code is given also to the attacker, that may exploit it to design a more effective attack.

2.3 Summary of Notation

Symbol	Definition
K	Long-term secret used for the secure RNG
z	Index of the random sequence used to sign the code
s_i	Shares received by user i
T_c	Chip duration (e.g for Galileo $T_\mathrm{c} \approx 1\mu\,\mathrm{s}$)
T_s	Signed code duration
T_w	Window duration of the MA filter
R_i	Side information rate for the user i
k	# of chips flipped on system side
k'_i	# of indexes to be flipped by user i
k''	# chips flipped by the attacker
ρ	Fraction of code flipped on system side
ξ	Errors introduced by user i
L	Signed code length
$\boldsymbol{c}_0$	Open, i.e., publicly-known, spreading code
$\boldsymbol{c}$	Signed spreading code
$\boldsymbol{c}'_i$	Spreading code signed by user i
$\boldsymbol{c}''$	Attacker spreading code
$\boldsymbol{u}$	Signature, system side
$\boldsymbol{u}'_i$	Signature, user i
$\boldsymbol{u}''$	Signature, attacker
$\boldsymbol{x}_0(t)$	Open signal
$\boldsymbol{y}(t)$	Signed Signal, system side
$\boldsymbol{x}'_i(t)$	Signed signal, user i
$\boldsymbol{x}''(t)$	Signal signed by the attacker

2.4 System Model

We start by modeling the GNSS open signal transmitted by satellite s as

$$x_0(t) = d_m c_{0,\ell} \cos(2\pi f_0 t + \varphi_0) \text{ , with } t \in [mT_\mathrm{s}, (m+1)T_\mathrm{s}] \cap [\ell T_\mathrm{c}, (\ell+1)T_\mathrm{c}] \ , \qquad (2.1)$$

where

- d_m is the mth messages data symbol with symbol period T_s;

- $c_{0,\ell} \in \{-1, 1\}$ is the ℓth chip of open PRN spreading code. The PRN $c_0 \in \{-1, 1\}^L$ has chip period $T_\mathrm{c} \ll T_\mathrm{s}$ and is composed by L chips;

- f_0 and φ_0 are carrier frequency and phase, respectively.

For ease of reading, we consider a binary phase shift keying (BPSK) modulation: still, it is straightforward to extend this analysis to more complex signal modulations and models, e.g., including the binary offset carrier (BOC).

From the system side, either at the ground segment or at the space vehicle (SV), a secure random number generator (RNG), driven by a long-term secret key or seed K, outputs a random sequence $\mathcal{Z}$ that will be used to generate a time-varying and unpredictable signature over the PRN code $\boldsymbol{c}_0$, for each considered SV. The resulting signed code, $\boldsymbol{c}$, replaces the open code $\boldsymbol{c}_0$.

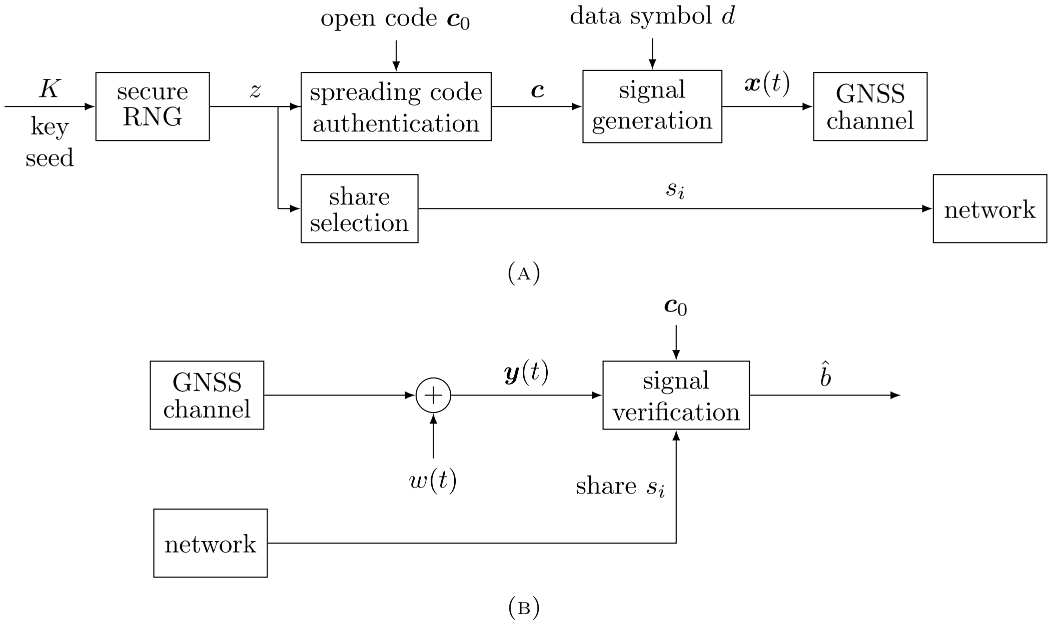


Figure 2.3: General abstract signal processing model for network aided GNSS signal verification: (A) transmitter and (B) receiver.

From the same random sequence z, a *share selection* procedure will yield a random share s_i, specifically targeted to receiver i with a lower information rate than z. The share will then be delivered securely to user i across the network and used to compute the partially signed codes $\boldsymbol{c}'_i$. It is crucial that the share s_i does not allow to recover, even partially, the long-term secret K, nor to predict future values of the random stream z, as this would give receiver i the capability to conduct a successful forging attack onto other receivers in the future. For this reason, we require a Markovian relationship $k \to z \to s_i$, in that s_i must be independent of k given z. A summary of the scheme is modeled in Figure 2.3 where the binary output $\hat{b}$ represents the output of the signature verification process.

We model the channel between SV and the GNSS receiver as an additive white Gaussian noise (AWGN) channel. A receiver will then obtain signal $y(t)$, which is the superposition of the signals broadcast by each SV. After acquisition and tracking, the authenticity of each signal is assessed using the proposed signal verification function.

We assume a (possibly) loose time synchronization between transmitter and receiver[2]. Hence, considering the part of the signal $x(t)$ signed with code $\boldsymbol{c}$, when the receiver obtains $\boldsymbol{c}'_i$ it also knows the corresponding part of the signal $y(t)$ that has to be authenticated with that specific share.

As in ACAS, to download the shares, each user, i, has to be provided with an authenticated and integrity protected connection to the ground station server (e.g. through a transmission control protocol (TCP)/internet protocol (IP) connection), with side information rate at least equal to R_i.

No further assumption is done about the receiver. Moreover, no form of NMA protection on d_m is considered, essentially assuming the data bit is deterministic: still, our scheme is not alternative but complementary to NMA, therefore it would be possible to have both schemes working at the same time.

The protocol's parameters are

[2] Note that the same assumption is in place for both ACAS, OSNMA, and CHIMERA.

- the *signature renewal period*, i.e., how often a new signature, $\boldsymbol{c}$, is generated
$$T_{\mathrm{r}} = nLT_{\mathrm{c}} \quad , \quad n \in \mathbb{N}_0 \, ; \tag{2.2}$$
- the *signature rate*
$$R_{\mathrm{s}} = \frac{1}{T_{\mathrm{r}}} H(z) \, ; \tag{2.3}$$
- the *side information rate for the user i*
$$R_i = \frac{1}{T_{\mathrm{r}}} H(s_i) \, ; \tag{2.4}$$
- the *fraction of the code flipped by the transmitter*
$$\rho = \frac{k}{L} \, ; \tag{2.5}$$
- the *fraction of code that the receiver flips using its own share*, s_i,
$$\gamma_i = \frac{d_{\mathrm{H}}(\boldsymbol{c}'_i, \boldsymbol{c}_0)}{L} = \frac{k'_i}{L} \, ; \tag{2.6}$$
- since, in general $s \neq s_i$, the errors introduced by the scheme in the PRN are
$$\xi = \frac{d_{\mathrm{H}}(\mathbf{c}', \mathbf{c})}{L} \, . \tag{2.7}$$

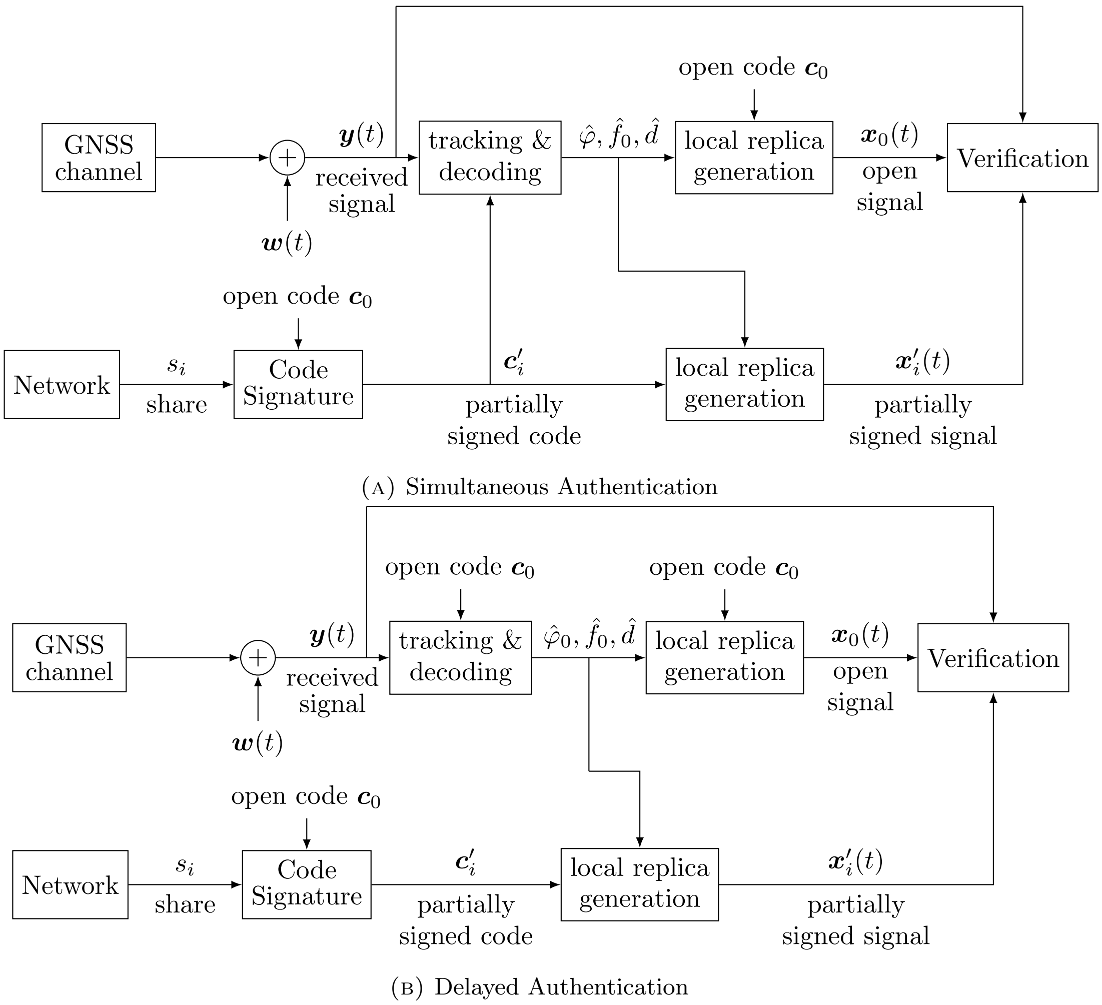
Given a threshold λ, we introduce the false alarm and the missed detection probabilities respectively as, the probability of rejecting a legitimate signal and accepting a tampered one. Thus, the performance of the scheme will be evaluated considering the error probability
$$p_{\mathrm{e}} = \inf_{\lambda} \left(p_{\mathrm{FA}}(\lambda) + p_{\mathrm{MD}}(\lambda) \right) \, , \tag{2.8}$$
and the detection rate
$$R_{\mathrm{d}} = \lim_{L \to \infty} \frac{1}{nLT_{\mathrm{c}}} \log_{1/2} p_{\mathrm{e}} \, . \tag{2.9}$$

2.5 Formulation of the protocol

In this Section, we describe in detail each step of the authentication protocol. Two alternative modes are considered

Delayed authentication each share s_i is distributed after the GNSS signal block to be authenticated has been received by all the verifiers; the receivers must be able to perform all the operations required for the PVT even by using the public $\boldsymbol{c}_0$ code.

Simultaneous authentication each share s_i is distributed in advance therefore it can be used for tracking and demodulation of the signed signal. It is important for shares distributed to distinct receivers to be sufficiently diverse: this prevents receiver i holding share s_i from learning significant information about share s_j and obtaining an advantage to attacking receiver j. This is accomplished by generating the shares randomly and independently of each other and by using the obfuscation strategy, described in the latter Sections.



(A) Simultaneous Authentication

(B) Delayed Authentication

FIGURE 2.4: Signal processing representation of the verification protocol for the delayed (A) and the simultaneous authentication (B) modes.

A general signal processing abstraction model for the problem is illustrated in Figure 2.4.

In the next Section, we will analyze the scheme block by block.

2.5.1 Code Signatures

We split the open code $\boldsymbol{c}_0$ into blocks of length L. We employ the one-time pad (OTP) as signature function. Defined $\boldsymbol{u} \in \{-1, 1\}^L$ as the bit string associated to index z, the signature function is

$$\boldsymbol{c} = S(\boldsymbol{u}, \boldsymbol{c}_0) = \boldsymbol{u} \odot \boldsymbol{c}_0 \ , \tag{2.10}$$

where $\odot$ is the element-wise product. Notice that, the set $\{-1, 1\}$ with the $\odot$ operation, is a group equivalent to the binary group $\{0, 1\}$ with the XOR operation. Thus, this is indeed still a OTP.

The same operation is executed by the receiver to retrieve $\boldsymbol{c}'$ by using the index s_i, by replacing $\boldsymbol{u}' \in \{-1, 1\}^L$ to $\boldsymbol{u}$ in (2.10).

2.5.2 Signal Generation & Share Selection

The signed signals are generated as in (2.1), swapping each open code chip $c_{0,\ell}$, the one signed by the system as in (2.10), c_ℓ.

Concerning the share selection block in the delayed authentication mode s_i, the share to be delivered to the user i, is picked randomly among the $\binom{k}{k'}$ possible combination of indexes. Notice that, as discussed in [44], this is a sub-optimal strategy in terms of trade-off between code entropy and correlation loss; however, it still achieves good performance, and it is easier to implement than the optimal strategy.

In the simultaneous authentication mode, the previous strategy may be threatened by *internal code attacks* (see Section 2.6) where the attacker exploits its own (legitimate) share to forge a new signal. To counter this we consider an obfuscation strategy: the share s_i still contains k indexes of chip to be flipped, but only $k' - k''$ among them correspond to chips indexes actually flipped by the transmitter. This allows to increase the diversity between each share pair: in Section 2.7.2 we will show that, even with a low value of k'', we are able to successfully detect internal code attacks. An additional countermeasure to this class of attacks is to optimize the share distribution at the network side assuring that, for instance, receivers close to each other obtain shares as different as possible. Such analysis however is out of the scope of this work.

2.5.3 Acquisition, Tracking & Decoding

A detailed description of these operations can be found in [2, 27]: these allow the receiver to estimate carrier frequency $\hat{f}_0$, code phase $\hat{\varphi}_0$ and data $\hat{d}$ from each SV signal.

In the delayed authentication mode, the receiver has to perform acquisition and tracking on the signed signal using the open code $\boldsymbol{c}_0$: however, as we show in Appendix A.1, processing the signal using a PRN $\boldsymbol{c}_i$ where a fraction ξ of the chips does not have to correct polarity, is equivalent to perform the correlation between a correct PRN, i.e., with $\boldsymbol{c}_{i,\ell} = \boldsymbol{c}_{i,\ell}$, $\forall \ell$ and a signal with an additional noise variance

$$\sigma^2_{\text{loss,dB}} = -20 \log(1 - 2\xi)\,. \tag{2.11}$$

Thus, in the delayed authentication mode we have a trade-off between security and availability: longer signatures improve the security but make it harder to acquire.

2.5.4 Verification Function

First, the receiver exploits the estimate $\hat{f}_0$, code phase $\hat{\varphi}_0$, data $\hat{d}$ and the signed code to compute the local replica

$$x'_i(t) = \hat{d} c_{0,\ell} \cos\Big(2\pi \hat{f}_0 t + \hat{\varphi}_0\Big) \quad , \quad t \in [\ell T_\text{c}, (\ell+1) T_\text{c}] \tag{2.12}$$

For the sake of clarity, we break $y(t)$, $x_0(t)$ and $x'_i(t)$ into chunks $\boldsymbol{y}$, $\boldsymbol{x}_0$ and $\boldsymbol{x}'_i$ corresponding respectively to codes $\boldsymbol{c}$, $\boldsymbol{c}_0$ and $\boldsymbol{c}'_i$. Hence, to verify the signature, the receiver performs the GLRT against the class of all attacks where the transmitted signal is a linear function of the GNSS signal, computed using the open spreading code, obtaining

$$\text{GLRT}: \min_{\alpha} \; \|\boldsymbol{y} - \alpha \boldsymbol{x}_0\|^2 - \left\|\boldsymbol{y} - \boldsymbol{x}'_i\right\|^2 \lessgtr \lambda\,, \tag{2.13}$$

where λ is a suitably chosen threshold, and the signal is declared authentic if the ">" sign holds, forged if the "<" holds. The test of Eq. (2.13) can also be compactly written as

$$\text{GLRT}: \ 2E_{x'y} - E_{x'} - \frac{E_{x_0y}^2}{E_{x_0}} \lessgtr \lambda. \tag{2.14}$$

with the signal energies $E_{x'} = \sum_{\ell=0}^{L-1}[x'_\ell]^2$ (or E_{x_0}) and cross energies $E_{x'y} = \sum_{\ell=0}^{L-1} x'_\ell y_\ell$ (or E_{x_0y}) computed over the L-chip observed block. It is easy to prove that the minimum in (2.13) is achieved when $\alpha^\star = E_{x_0y}/E_{x_0}$.

Observe that, besides having a sound analytical basis, this test is a more consistent verification than a simple threshold on the cross energy (correlation peak) value, $|E_{x'y}|$: the correlation may be thwarted by transmitting an over-amplified signal. Equivalently, even a threshold on the cross energy difference $E_{x'y} - E_{x_{0y}}$ is not reliable: the attacker could simply use as the flipped code $-c_0$.

In the following, we derive the relation between the GLRT and the protocol parameters considering both noiseless and AWGN scenarios.

Noiseless scenario In this Section, we consider a noiseless (ideal) scenario. As pointed out in the previous section, the attacker does not gain any advantage in overamplifing the signal: thus, without loss of generality we assume that all the signals have the same energy, i.e., $E_y = E_x = E_{x_0}$,

Proposition 1. *In the noiseless scenario, the GLRT of* (2.14) *is equivalent to the test*

$$\vartheta_{\text{ide}} \triangleq \rho(1-\rho) - \xi \lessgtr \frac{\lambda}{4E_{x_0}} = \lambda \tag{2.15}$$

where ξ and ρ coefficients represent respectively the energy loss due to the differences between the codes used by transmitter and verifiers.

Proof. Under these assumptions, the cross energies can be written as

$$E_{x'y} = E_y(1-2\xi) \tag{2.16}$$

$$E_{x_0y} = E_y(1-2\rho) \ , \tag{2.17}$$

Thus, (2.14) is equivalent to

$$2E_{x'y} - E_{x'} - \frac{E_{x_0y}^2}{E_{x_0}} = 4\rho(1-\rho)E_{x_0} - 4\xi E_{x_0} \lessgtr \lambda \ . \tag{2.18}$$

By normalizing over the energy of the signals, we get (2.15). □

This formulation directly relates verification performance to protocol parameters: for instance, the FA probability is

$$p_{\text{FA}} = P(\vartheta < \lambda | \mathcal{H}_0) = P(-\rho^2 + \gamma < \lambda) \ , \tag{2.19}$$

where we used the fact that, neglecting obfuscation, it holds $\xi_{\text{leg}} = \rho - \gamma$. Of course, in the noiseless scenario, the result is deterministic: still, this shows that even in the ideal scenario there is a lower bound on the side information rate.

AWGN scenario Now, we extend the results of Proposition 1 to the AWGN scenario.

Proposition 2. *In the AWGN scenario, the GLRT of* (2.14) *is equivalent to the test*

$$\vartheta \triangleq 4L\rho(1-\rho) - 4\xi L - \sigma^2\eta_2^2 + 4\sigma\sqrt{L}\sqrt{\gamma - 2\rho\gamma + \rho^2} \ \eta_1 \lessgtr \lambda \ . \tag{2.20}$$

where $\eta_1, \eta_2 \sim \mathcal{N}(0,1)$.

Proof. Proof in Appendix A.2. □

The GLRT is now composed of two parts: the first is deterministic and common with (2.18); the second is probabilistic and can be modeled as a generalized χ^2 distribution (i.e., the sum of a χ^2 and a Normal r.v.). Notice that, in Section 2.7.1, we will show that the Normal component is the dominant one. Hence, called ϑ_n the result of the nth GLRT, collecting the results of N tests and averaging it will hold

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N-1} \vartheta_n \approx \vartheta_{\text{ide}} \,. \tag{2.21}$$

A practical implementation of this strategy uses a moving average (MA) filter: this allows the receiver to dynamically increase the size of the sliding window, N, to obtain a more reliable verification even in low C/N_0 conditions. Thus, knowing that the metric in the AWGN scenario converges to the one in the noiseless scenario, we can focus on the first, simpler, metric.

2.6 Attack strategies

In this Section, we consider the possible attacks that may target a GNSS receiver evaluating the performance of the detection scheme under those attacks. First, we call $x''(t)$ the forged signal, that will be processed by the legitimate transmitter in chunks, $\boldsymbol{x}''$. To generate it, the attacker has to pick a counterfeit string $\boldsymbol{c}'' \in \{-1,1\}^L$, or equivalently a binary string $\boldsymbol{u}''$ to give as input to the code signature function (2.10).

We consider the following attacker strategies:

Open code attack: the attacker forges the signal using $\boldsymbol{c}'' = \boldsymbol{c}_0$ as spreading code, i.e., $\boldsymbol{u}'' = \{1\}^L$;

Flipped open code attack: the attacker use the flipped version of the open code, i.e., $\boldsymbol{u}'' = \{-1\}^L$;

Independent code attack: the attacker tries to guess the code that will be used by the receiver, picking randomly one out the possible $\binom{L}{k'_i}$ combinations of indices;

Internal code attack: in the simultaneous authentication mode, each receiver obtains a share out of the $\binom{k}{k'}$ possible ones before acquisition and tracking, therefore it can exploit its own share s_j to forge the spoofed signal, i.e., $\boldsymbol{u}'' = \boldsymbol{u}_j$;

SCER: the attacker observes $y(t)$ and tries to estimate $\boldsymbol{c}$ collecting energy from each observed chip by considering a maximum a posteriori criterion (MAP) criterion. Indeed, this attack is effective only if the same signature code is used more than once.

Notice that there is no a priori optimum attack since the optimal strategy is related to the particular scenario or to the resources available to the attacker. For instance, if the attacker has low C/N_0 or, equivalently $T_\text{r} \approx LT_\text{c}$, it will not be possible to lead an effective SCER attack.

In the next sections, we will model in detail each attack by using the parameters ρ and ξ. This will allow us to analytically estimate the performance of the verification in the under-attack condition, by using (2.15) and (2.20).

2.6.1 Open and Flipped Open Code Attack

We start modeling the open and flipped open code attacks, where we have respectively $\boldsymbol{u}'' = \{1\}^L$ and $\boldsymbol{u}'' = \{-1\}^L$. In the first case the fraction of flipped chips (2.5) is $\rho_{\text{open}} = 0$ while the metric in the noiseless case (2.15) is

$$\vartheta_{\text{open}} = \rho_{\text{open}}(1-\rho_{\text{open}}) - \xi_{\text{open}} = -\xi_{\text{open}} == -\frac{d_{\text{H}}(\boldsymbol{u}'', \boldsymbol{u}_i)}{L} = -\frac{k'}{L} = -\gamma \;, \quad (2.22)$$

hence the code used by the attacker differs from the one used by the receiver by just k' chips. Conversely, for the flipped open code, we have $\rho_{\text{flip}} = 1$ and

$$\vartheta_{\text{flip}} = -\xi_{\text{flip}} = -\left(1 - \frac{k'}{L}\right) = -(1-\gamma) \;. \quad (2.23)$$

This means that the attacker always guesses wrong exactly $L - k'$ chips. The aim of the attacker is to maximize ϑ: we conclude that the open is better than the flipped open code attack if the signature is shorter than half the size of the code, i.e., if $k' < L/2$ or, equivalently, $\gamma < 1/2$.

2.6.2 Independent Code Attack

The attacker randomly generates $\mathbf{u}''$, hoping it to be close as possible to $\boldsymbol{u}_i$, i.e., to achieve a low value of $d_{\text{H}}(\boldsymbol{u}'', \boldsymbol{u}_i)$. Let us call X the r.v. modeling the number of chips correctly guessed by the attacker. The distribution of X is then

$$P(X = \ell) = \frac{\binom{k'}{\ell}\binom{L-k'}{k'-\ell}}{\binom{L}{k'}} \;. \quad (2.24)$$

Interestingly, the attacker is not interested in guessing all the k indexes picked by the system, but just the k' indexes contained in the share s_i used by the receiver, therefore $\rho_{\text{ind}} = k'/L = \gamma$. Indeed, flipping more chips than necessary, lead to a lower success probability. The metric under attack is then

$$\vartheta_{\text{ind}} = -\gamma^2 - \gamma + 2\frac{X}{L} \quad (2.25)$$

2.6.3 Internal code Attack

This attack can only target the receiver in the simultaneous authentication mode. The idea is that the malicious user j exploits share s_j to generate the counterfeit signal. Indeed, the higher the side information rates R_i and R_j are, the higher is the probability that share s_j and s_i, used instead by the legitimate user, are similar.

This attack can be modeled similarly to the independent code attack: defined as Y the r.v. modeling the number of chips common to both transmitter and receiver, neglecting the obfuscation, it yields

$$P(Y = \ell) = \frac{\binom{k'}{\ell}\binom{k-k'}{k'-\ell}}{\binom{k}{k'}}. \quad (2.26)$$

thus, it holds

$$\vartheta_{\text{int}} = -\gamma^2 - \gamma + 2\frac{Y}{L} \quad (2.27)$$

Notice that (2.25) and (2.27) are identical but the distributions of X and Y are different.

The threat of the internal code attack is mitigated by using the obfuscation strategy. The idea is that, by using the obfuscation, we are adding some degree of uncorrelatedness to each share, thus, with high probability, the k'' addition (wrong) chips given to the attacker will be different from the k'' chips obtained instead by the legitimate receiver. In Section 2.7.2 we will show how the obfuscation impacts the performance of the internal code attack.

2.6.4 SCER Attack

We consider a scenario where the same signature is used for N consecutive periods, i.e., with signature renewal period $T_\mathrm{r} > NLT_\mathrm{c}$, with $N \in \mathbb{N}_0$. We assume the attacker to be synchronized with the code and able to perform acquisition, tracking, and data demodulation.

Thus, the attacker has at disposal N periods of code to correctly estimate $\boldsymbol{c}$ from the received signal. We now compute the probability that the polarity of the ℓth chip is not correctly estimated by the attacker, $P_\mathrm{SCER}(N)$.

Considering an attacker processing the signal in baseband, with data symbol d publicly known, the results of the correlation between the noisy signal and open code after N periods is, for the ℓth chip,

$$r_\ell(N) = \sum_{n=1}^{N} c_\ell c_{0,\ell} + \sum_{n=1}^{N} \eta_n c_{0,\ell} = N c_\ell c_{\ell,\mathrm{o}} + \sum_{n=1}^{N} \eta_n c_{0,\ell}, \tag{2.28}$$

which is Gaussian r.v. with $r_\ell(N) \sim \mathcal{N}(N c_\ell c_{\ell,0}, N\sigma^2)$.

Next, after collecting energy from N consecutive periods, the ℓth chip is estimated as

$$u''_\ell = \begin{cases} -1 & \text{if } r_\ell(N) < 0 \\ 1 & \text{if } r_\ell(N) \geq 0 \end{cases}. \tag{2.29}$$

Hence, the probability that the attacker guesses the chip wrong, is

$$P_\mathrm{SCER}(N) = P\big(r_\ell(N) < 0 \,|\, c_\ell = c_{0,\ell}\big) = P\big(r_\ell(N) > 0 \,|\, c_\ell \neq c_{0,\ell}\big)\ . \tag{2.30}$$

Now, we can model the SCER attack as we have done for the previous attacks: calling, for compactness, $\xi = P_\mathrm{SCER}(N) = p$, we get

$$E[\rho_\mathrm{SCER}] = \rho_\mathrm{legit}(1-2p) + p \tag{2.31}$$

$$E[\xi_\mathrm{SCER}] = \rho_\mathrm{legit}(1-2p) - \gamma(1-2p) + p \tag{2.32}$$

$$\begin{aligned} E[\vartheta_\mathrm{SCER}] &= E[\rho_\mathrm{SCER}](1 - E[\rho_\mathrm{SCER}]) - E[\xi_\mathrm{SCER}] = \\ &= -\rho^2_\mathrm{legit}(1-2p)^2 - p^2 - 2\rho_\mathrm{legit}(1-2p)p + \gamma(1-2p) \end{aligned} \tag{2.33}$$

Notice that as $P_\mathrm{SCER}(N) \to 0$, achievable when both $N, T_\mathrm{r} \gg 0$, the attacker is able to perfectly estimate the code: the counterfeit signal achieves in fact the same performance as the legit signal.

Calling $\delta = \sqrt{N}/\sigma$ it holds that

$$P_\mathrm{SCER}(N) = \frac{1}{2}\mathrm{erfc}\left(\frac{\sqrt{N}}{\sqrt{2}\sigma}\right) = \frac{1}{2}\mathrm{erfc}\left(\frac{\delta}{\sqrt{2}}\right)\ . \tag{2.34}$$

For a fixed σ, the attacker can estimate how many periods of code, N, it has to observe in order to obtain the desired $P_{\mathrm{SCER}}(N)$, high enough to overcome the ϑ_{th} picked by the receiver. Conversely, if $T_{\mathrm{th}} < NLT_{\mathrm{c}}$ the attacker will not have time to forge a signal with a code close enough $\boldsymbol{c}'$.

In the next Section, we will show the correctness of this statistical model, comparing the analytical results to the one obtained from our signal simulator.

2.7 Numerical Results

In this Section, we present the results for both statistical models, described in the previous sections, and numerical simulations.

Regarding the numerical simulations, we implemented the authentication protocol on both our signal simulator and our software receiver, running 15 000 simulations for each considered scenario. Our simulator has already been used to test the other GNSS attack and defenses, e.g., in [15, 45]

We chose to run the simulations for the Galileo GNSS E1 signals and, to ease the implementation, we considered $L = L = 4092$ chips, i.e., the length of a PRN sequence.

We fixed the sampling frequency to $f_{\mathrm{s}} = 4\,\mathrm{MHz}$ and assumed ideal acquisition and tracking, thus focusing on the performance of the verification check. We consider each SV signal separately, i.e., neglecting the loss due to the cross-correlation between different SV signals. A summary of chosen protocol parameters is reported in Table 2.1, for a renewal period $T_{\mathrm{r}} = LT_{\mathrm{c}} = 4\,\mathrm{ms}$, i.e., generating a new signature after each use. We remark that by increasing T_{r}, we can relax the constraint on the rates R and R_i, which would decrease linearly with N, the number of PRNs with the same signature; on the other hand, in such conditions also the SCER attack is expected to be more effective. This trade-off will be discussed in Section 2.7.2.

We focused on the scenarios with $C/N_0 = 40\,\mathrm{dBHz}$, and $C/N_0 = 30\,\mathrm{dBHz}$: a reasonable working condition for a GNSS receiver is within $35 - 45\,\mathrm{dBHz}$ [46], therefore the considered values represent a typical and a harsh condition scenario (e.g., urban canyon).

First, we verify the match between statistical models and data collected from the signal generator. Next, we evaluated the performance of all the attacks modeled in Section 2.6, for different scenarios and parameters, reporting both receiver operating characteristic (ROC) curves and detection rates.

TABLE 2.1: Parameters used for the numerical results with $T_{\mathrm{r}} = LT_{\mathrm{c}} = 4\,\mathrm{ms}$ and $L = 4092\,\mathrm{bit}$.

Mode	k_i	R_i
Delayed Authentication $k = 512\,\mathrm{bit}$, $R \approx 554.98\,\mathrm{kbit/s}$	$k_{i,1} = 128\,\mathrm{bit}$	$R_{i,1} \approx 204.19\,\mathrm{kbit/s}$
	$k_{i,2} = 256\,\mathrm{bit}$	$R_{i,2} \approx 343.97\,\mathrm{kbit/s}$
	$k_{i,3} = 384\,\mathrm{bit}$	$R_{i,3} \approx 458.11\,\mathrm{kbit/s}$
	$k_{i,4} = 512\,\mathrm{bit}$	$R_{i,4} \approx 554.98\,\mathrm{kbit/s}$
Simultaneous Authentication $k = 2046\,\mathrm{bit}$, $R \approx 1021.4\,\mathrm{kbit/s}$	$k_{i,5} = 1279\,\mathrm{bit}$	$R_{i,5} \approx 915.16\,\mathrm{kbit/s}$
	$k_{i,6} = 1535\,\mathrm{bit}$	$R_{i,6} \approx 974.91\,\mathrm{kbit/s}$
	$k_{i,7} = 1785\,\mathrm{bit}$	$R_{i,7} \approx 1009.4\,\mathrm{kbit/s}$
	$k_{i,8} = 2046\,\mathrm{bit}$	$R_{i,8} \approx 1021.4\,\mathrm{kbit/s}$

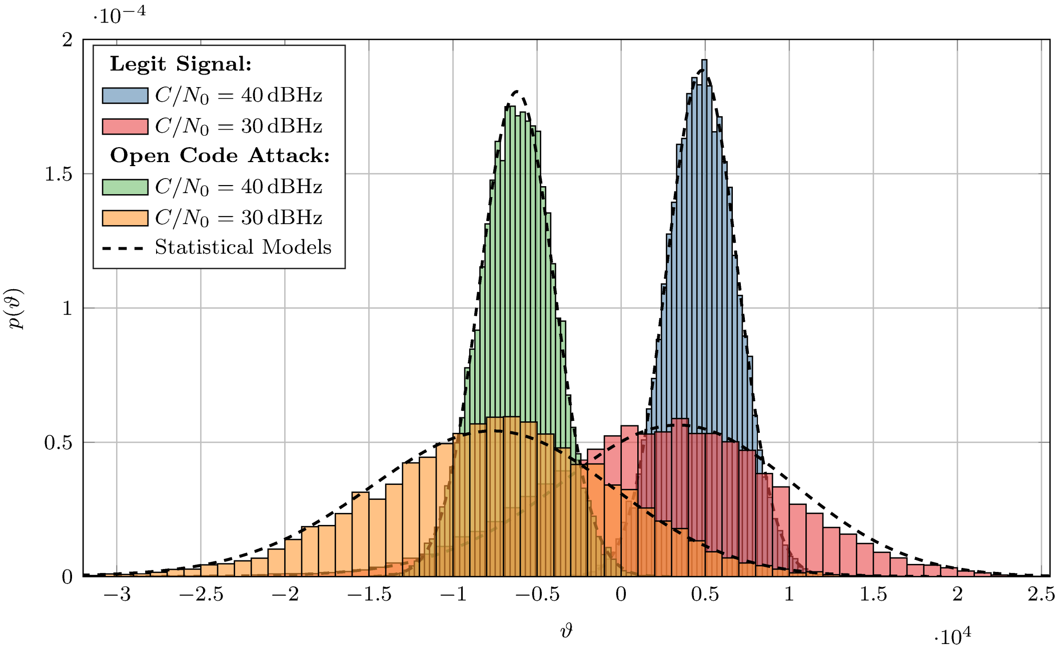


Figure 2.5: Comparison between statistical model (black dashed lines) and simulation results, for delayed authentication scenario with $k_i = 384\,$bit and $C/N_0 = 30$ and $40\,$dBHz.

2.7.1 Comparison between statistical model and simulations

Our aim is to validate the statistical model of the GLRT of (2.20): to do so, we compare the statistical model to the simulation results, obtained using our signal simulator and software receiver. We considered the case where $k = 512\,$bit and $k_i = 384\,$bit, for the delayed authentication mode. We compare the metric in a legitimate vs an under attack scenario, where the attacker uses an open code attack: as anticipated in the previous section, the first is modeled plugging in (2.20) $\rho = k/L$, $\gamma = k_i/L$ and $\xi = \rho - \gamma$ while for the second we have to pick $\rho = 0$, $\gamma = k_i/L$ and $\xi = -\gamma$.

Figure 2.5 compares the results obtained by both the statistical model and the simulation results: the match is clear in both the considered C/N_0 conditions. Moreover, for the $C/N_0 = 40\,$dBHz, we observe that it is significantly easier to distinguish between the two distributions, while for $C/N_0 = 30\,$dBHz they partially overlap. Additionally, as anticipated earlier, even if ϑ is a generalized χ^2 r.v., it is easy to notice that the Gaussian term inside the χ^2 is the dominant one: this confirms that employing a moving a MA filter will improve the results of the verification procedure.

Finally, this shows that, given the triplet (ρ, γ, ξ), it is possible to generate any metrics distribution (and any attack), without the need for a sophisticated signal generator. On the other hand, such models may be also used by an advanced receiver to adapt or tune the protocol parameters, e.g., asking for a higher rate R_i, changing the threshold ϑ_{th} or even for attack detection.

2.7.2 Attack performances

In this Section we evaluate each attack, considering the statistical models reported in Section 2.6, identified by the triplets (ρ, γ, ξ). We exclude the SCER attack: this we will be treated later in a dedicated Section.

First, we report on Table 2.2 the normalized metric results for both delayed and simultaneous authentication modes for the noiseless scenario. Notice that for the independent and the internal code attack the result are not deterministic: for these results, we reported mean and standard deviation (between parenthesis). We remark that the obfuscation strategy was not employed. Indeed, the best attacks are those that get the highest ϑ values, getting as close as possible to the legitimate cases. As expected, by increasing the share size k_i, the metric value in the legitimate case increases as well; conversely, for that attacks, ϑ decreases, except for the internal attack: indeed increasing side information rate means more information for the attacker to exploit.

In the delayed authentication mode, the independent code attack outperforms the open code attack. Instead, in the simultaneous authentication mode, the best is by far the internal attack: this justify the use of the obfuscation strategy, that in this first batch of results has not been employed, where we trade average correlation power in the legitimate case for security against the most threatening class of attacks.

TABLE 2.2: Metric distributions obtained for the simultaneous and the delayed mode considering different attacks and side information rates. Between parenthesis the metric's standard deviation. No obfuscation strategy is in place.

	Delayed Authentication Mode			
	$k_{i,1}$	$k_{i,2}$	$k_{i,3}$	$k_{i,4}$
Legit Signal	0.02	0.05	0.08	0.11
Open Code Att.	-0.03	-0.06	-0.10	-0.13
Indep. Code Att.	-0.02 ($2.5\,10^{-3}$)	-0.05 ($3.5\,10^{-3}$)	-0.08 ($4.3\,10^{-3}$)	-0.11 ($4.8\,10^{-3}$)
	Simultaneous Authentication Mode			
	$k_{i,5}$	$k_{i,6}$	$k_{i,7}$	$k_{i,8}$
Legit Signal	0.063	0.125	0.187	0.25
Open Code Att.	-0.31	-0.38	- 0.44	-0.5
Indep. Code Att.	-0.098 (0.01)	-0.141 (0.01)	-0.1911 (0.01)	-0.25 (0.01)
Internal Code Att.	-0.02 ($7\,10^{-3}$)	0.05 ($6\,10^{-3}$)	0.14 ($3\,10^{-3}$)	0.25

Figure 2.6 shows the achievable p_{e} in the delayed authentication mode as a function of the MA filter duration, T_{w}, for different C/N_0 values and side information rates R_i. For instance, to achieve $p_{\mathrm{e}} \geq 2 \cdot 10^{-3}$ at $C/N_0 = 30\,\mathrm{dBHz}$ with $T_{\mathrm{w}} < 20\,\mathrm{ms}$, the receiver has to have side information rate $R_i \geq R_{i,3} = 458.11\,\mathrm{kbit/s}$.

Figures 2.7 and 2.8 highlight the effectiveness of obfuscation strategy in the simultaneous authentication mode, comparing the results with $k'' = 0$ and $k'' = 384\,\mathrm{bit}$: while the internal attack is still the most effective, the obfuscation strategy drastically decreases the successful probability the attack. For instance considering the $C/N_0 = 30\,\mathrm{dBHz}$ scenario, without the obfuscation strategy it is not possible to get $p_{\mathrm{e}} < 10^{-2}$ in less than 0.5 s, with any side information rate; conversely, using the obfuscation, we get $p_{\mathrm{e}} < 10^{-2}$ with $T_{\mathrm{w}} < 0.3\,\mathrm{s}$ and $R_i' > 1\,\mathrm{Mbit/s}$.

Figure 2.9 reports the detection rate R_{d} in the delayed authentication mode as a function of the side information rate R_{i} for different MA windows sizes, T_{w}, and $C/N_0 = 30$ or $40\,\mathrm{dBHz}$. As expected, increasing either R_{i} or R_{i} increases the detection rates: for instance, for the same side information rate, at $C/N_0 = 30\,\mathrm{dBHz}$ increasing the window size to $T_{\mathrm{w}} = 60\,\mathrm{ms}$ we outperform the scenario with $C/N_0 = 40\,\mathrm{dBHz}$ and $T_{\mathrm{w}} = 4\,\mathrm{ms}$.

Figure 2.10 and 2.11 compares the detection rates achievable in the simultaneous authentication mode achievable in harsh conditions ($C/N_0 = 30\,\mathrm{dBHz}$) against respectively the independent and the internal attacks, with ($k'' = 384\,\mathrm{bit}$) and without the obfuscation strategy $k'' = 0$, as a function of the side information rate R_{i} for different

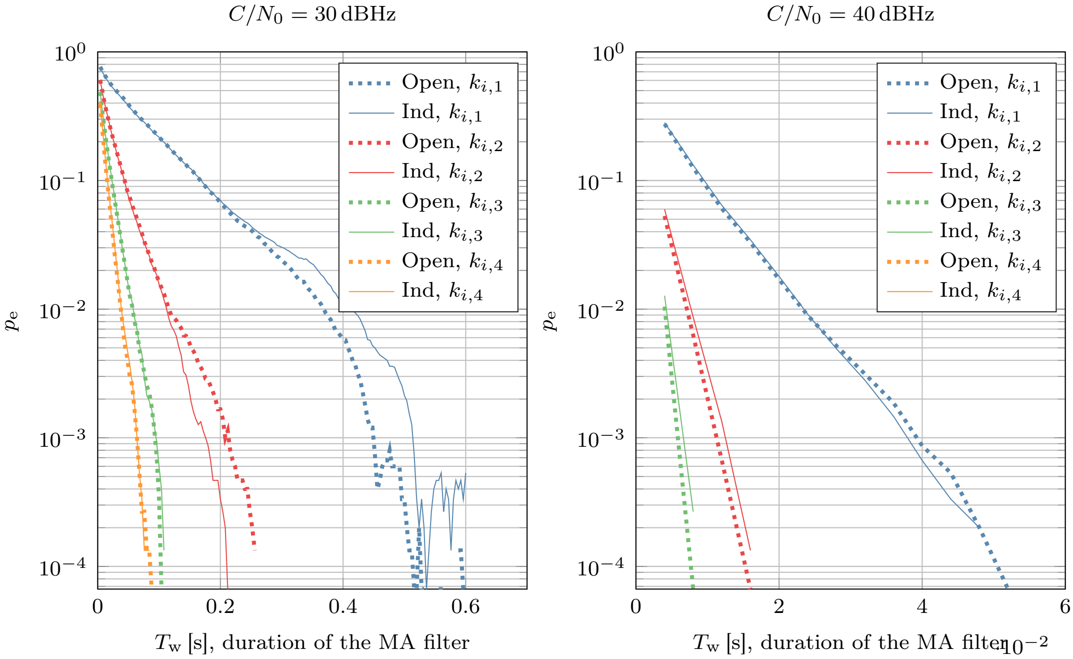


FIGURE 2.6: p_e obtained in the delayed authentication mode as function of the T_w time window for different side information rates, considering the open and the independent code attacks. On the left $C/N_0 = 30\,\mathrm{dBHz}$; on the right $C/N_0 = 40\,\mathrm{dBHz}$. Notice that, for $k_i = k_{i,4}$ at $C/N_0 = 40\,\mathrm{dBHz}$ we get always $p_e = 0$.

MA windows sizes, T_w. As expected, we have an opposite trend with respect to the side information rate: for the independent code attack increasing R_i improves the detection rate since the attacker has more chips to guess; conversely, in the internal code attack, by increasing R_i we are gifting more information about the legitimate receiver's share to the attacker. Opposite trends are observed also when the obfuscation strategy is employed: for the independent attack, R_d slightly degrades, while in the internal code attack the detection rate almost doubles. Indeed, since the first degradation is much lower than the improvement of the second, it is convenient to include the obfuscation strategy in the protocol. These considerations are confirmed also by Figure 2.12, where $C/N_0 = 40\,\mathrm{dBHz}$.

SCER attack We focus now on the SCER attack. As pointed out before, here we consider a signature renewal period $T_r \gg LT_c$. Indeed, if $T_r = LT_c$ the SCER cannot be successful.

Figure 2.13 compares the results obtained for the SCER attack considering either the statistical model and the simulator results: there is a clear match between statistical model and simulated results, thus we can consider the statistical model to be validated. Both figures exhibit a linear relation between T_w and the σ^2 of the attacker.

Figure 2.14 reports the $\vartheta_{\mathrm{SCER}}$ values obtained by using SCER attack: when the correct estimation probability is low (i.e low C/N_0 and/or low T_r) the SCER attack is the least effective attack among the considered ones (see Tab. 2.2). Conversely, in high C/N_0 conditions, the SCER attack is more threatening than the internal attack. Moreover, these plots highlight the trade-off that between rate and security: picking an higher T_r allows the receiver to perform the authentication with at a lower rate R_i but it increases the chances for the attacker to perform a successful SCER attack.

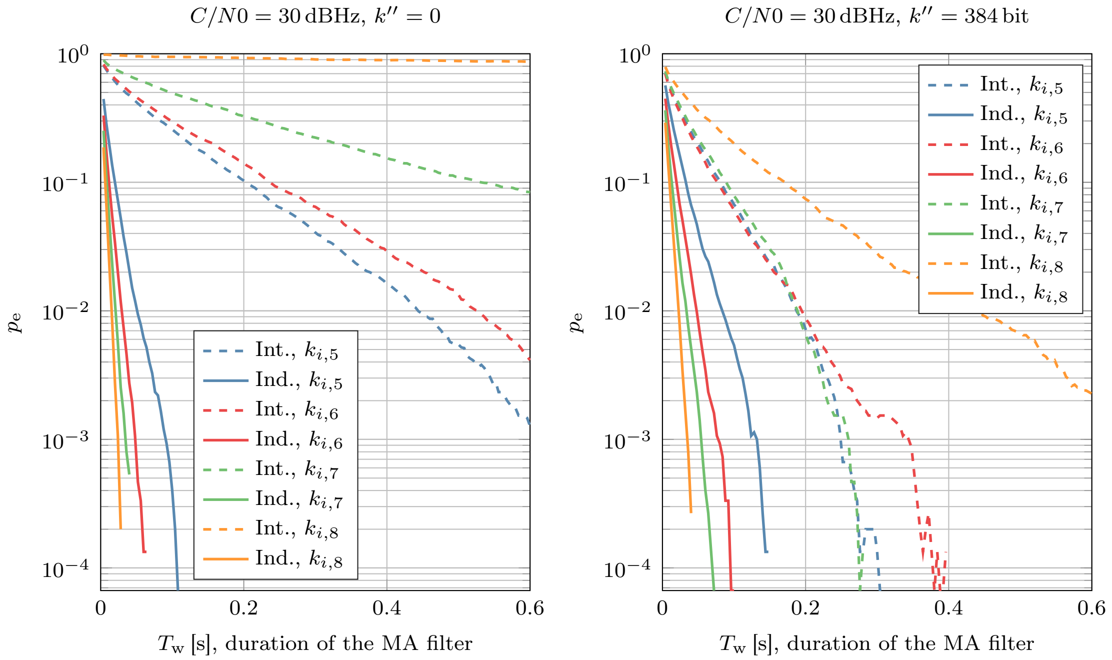


FIGURE 2.7: p_e in the Simultaneous Authentication with obfuscation as function of MA filter length T_w, $C/N_0 = 30\,\mathrm{dBHz}$.

TABLE 2.3: Summary of the attacks and their effectiveness: we circled the ones that proved to be most effective in the considered scenario.

	Delayed Mode		**Simultaneous Mode**	
Attack	Low C/N_0 $T_\mathrm{r} \approx T_\mathrm{s}$	High C/N_0, $T_\mathrm{r} \gg T_\mathrm{s}$	Low C/N_0 $T_\mathrm{r} \approx T_\mathrm{s}$	High C/N_0, $T_\mathrm{r} \gg T_\mathrm{s}$
Open Code Attack	✓	✓	✓	✓
Flip. Open Code Att.	✓	✓	✓	✓
Indep. Open Code Att.	(✓)	✓	✓	✓
Internal Attack	✗	✗	(✓)	(✓)
SCER	✗	(✓)	✗	(✓)

We summarize the obtained results on Table 2.3 where we report which attack can be chosen on each scenario, highlighting the most effective ones: notice that in the simultaneous authentication mode the most threatening attack may be either the internal or the SCER attack, depending on both C/N_0, T_r and side information rate at the receiver.

2.8 Conclusions

In this chapter, we have proposed an authentication protocol for the signal-level authentication in GNSS. More in detail, we have described the current state of art mechanisms for signal-level authentication, such as ACAS and CHIMERA. Next, we have presented a robust authentication verification solution based on the GLRT. We have considered several attacks and statistically characterized their success probability; among these, we have also considered the security code estimation and replay attack and the internal code attack, where the attacker computes the counterfeit signal by using its own (legitimate) signed code. Concerning the internal code attack, we have

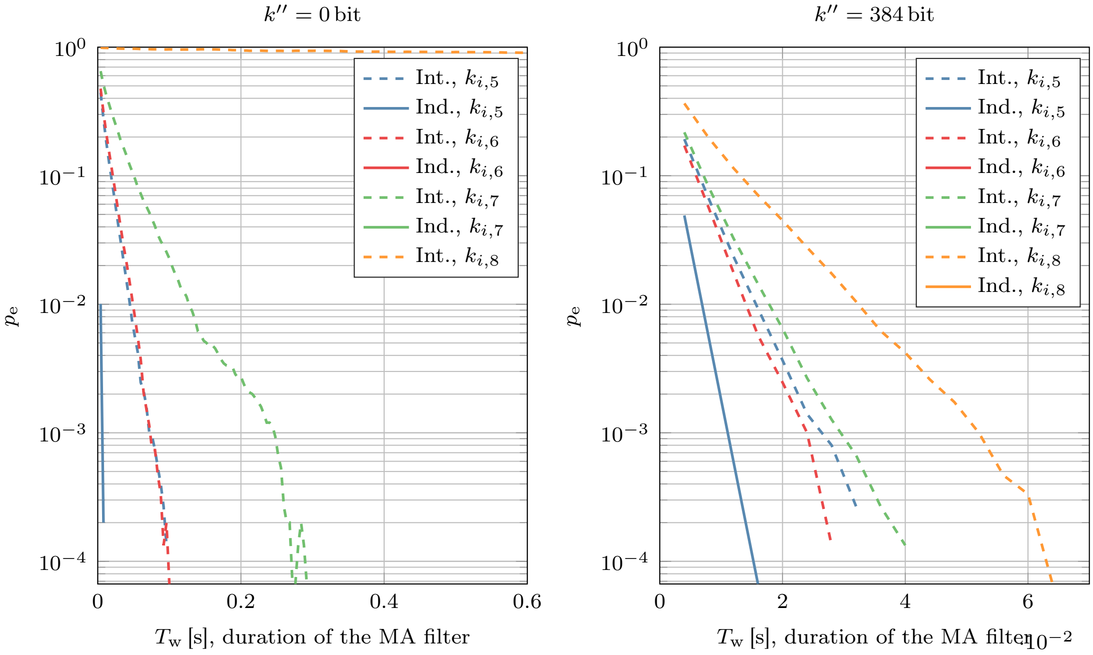


Figure 2.8: p_e in the Simultaneous authentication with obfuscation as function of MA filter length T_w, $C/N_0 = 40$ dBHz.

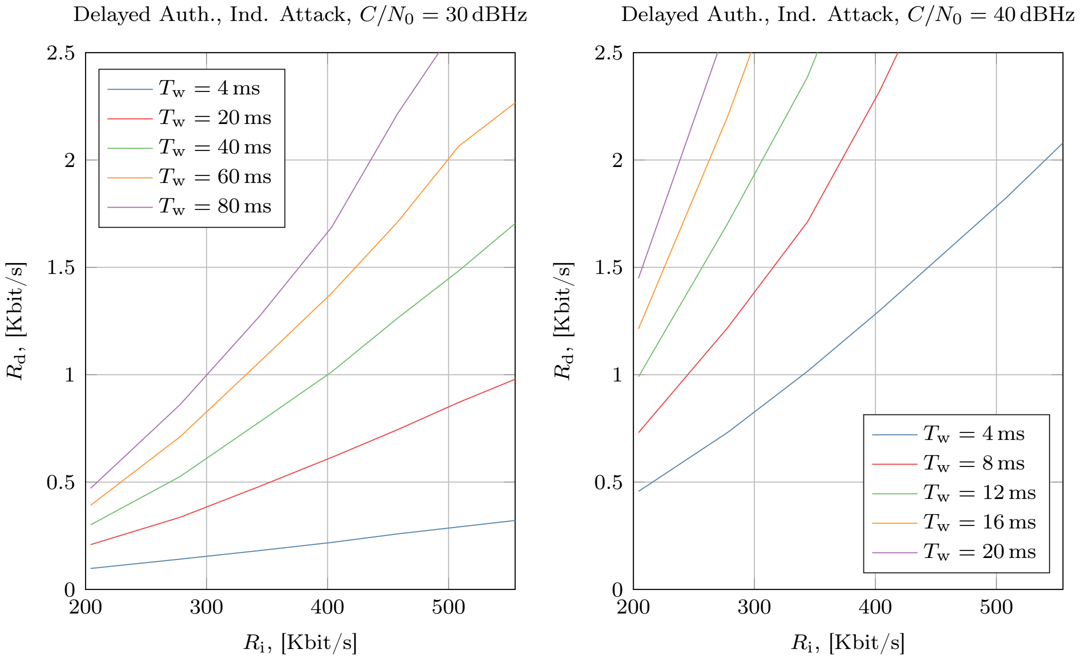


Figure 2.9: Detection Rates achievable in the Delayed Auth. Mode considering several length of the moving average filter, T_w.

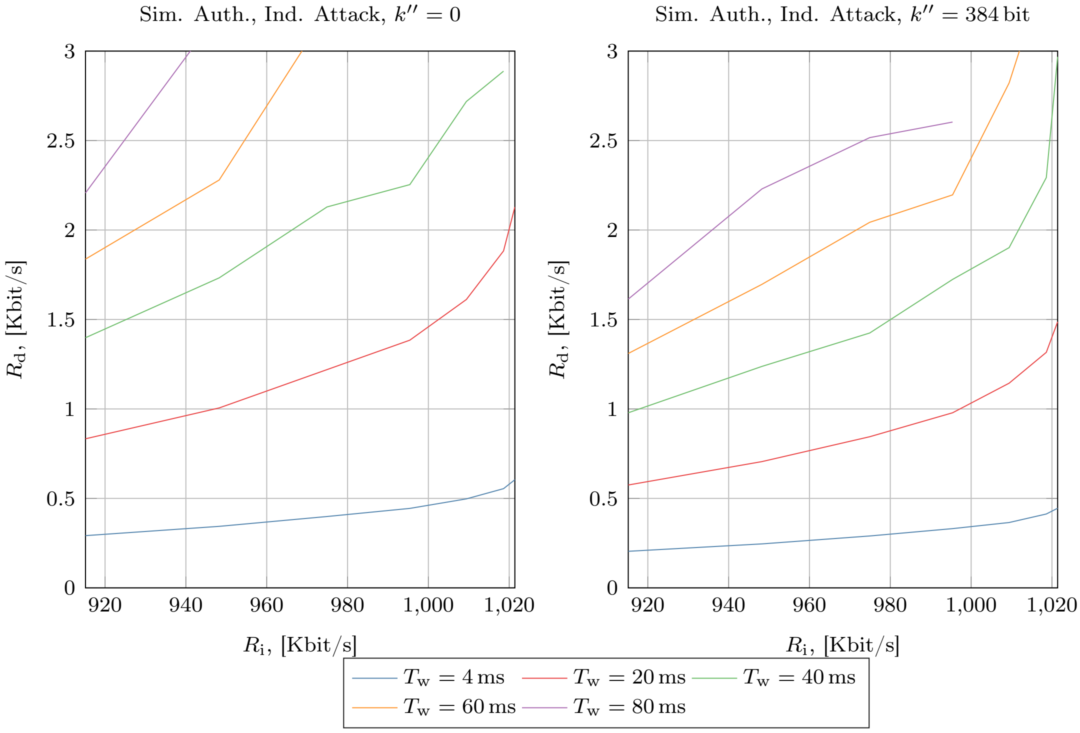


FIGURE 2.10: Detection Rates achievable in the Simultaneous Auth. Mode considering several length of the moving average filter, T_w, at $C/N_0 = 30$ dBHz for the independent attack.

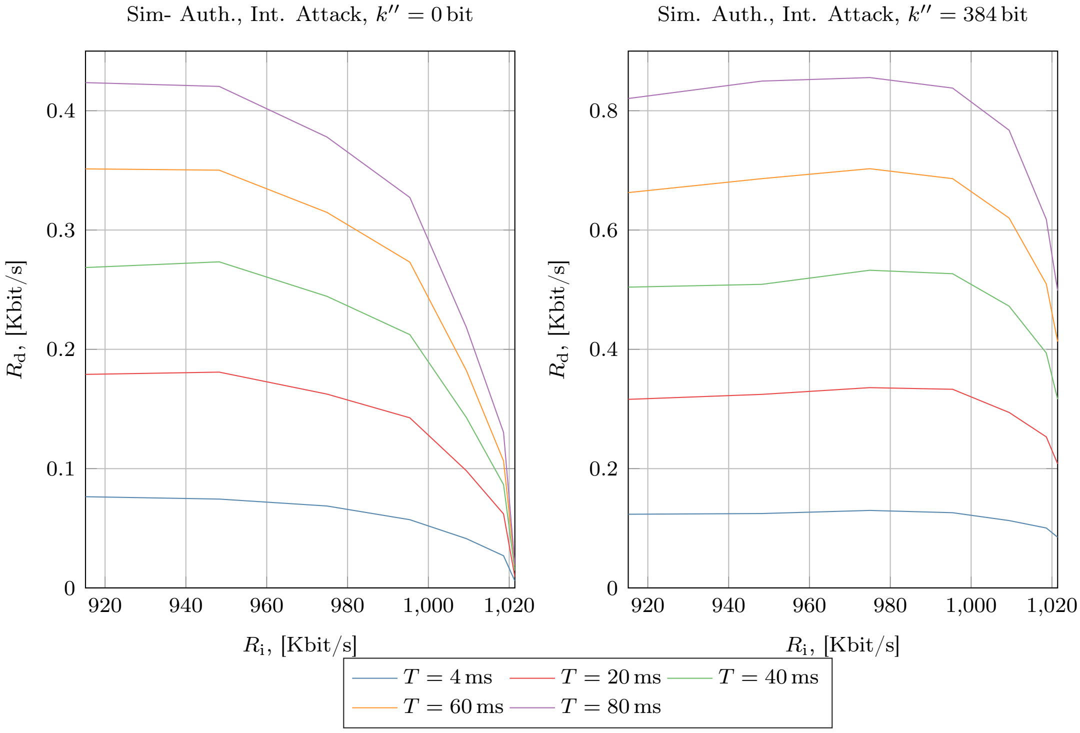


FIGURE 2.11: Detection Rates achievable in the Simultaneous Auth. Mode considering several length of the moving average filter, T_w, at $C/N_0 = 30$ dBHz for the internal attack.

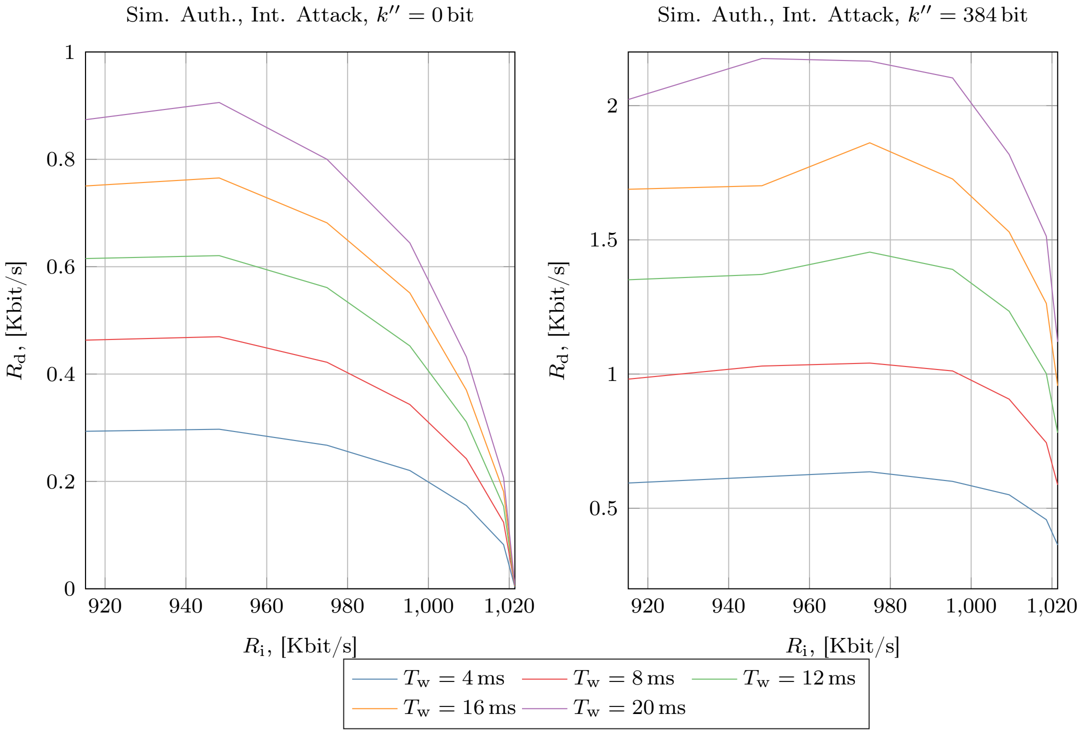


FIGURE 2.12: Detection Rates achievable in the Simultaneous Auth. Mode considering several length of the moving average filter, T_w, at $C/N_0 = 40$ dBHz for the internal attack.

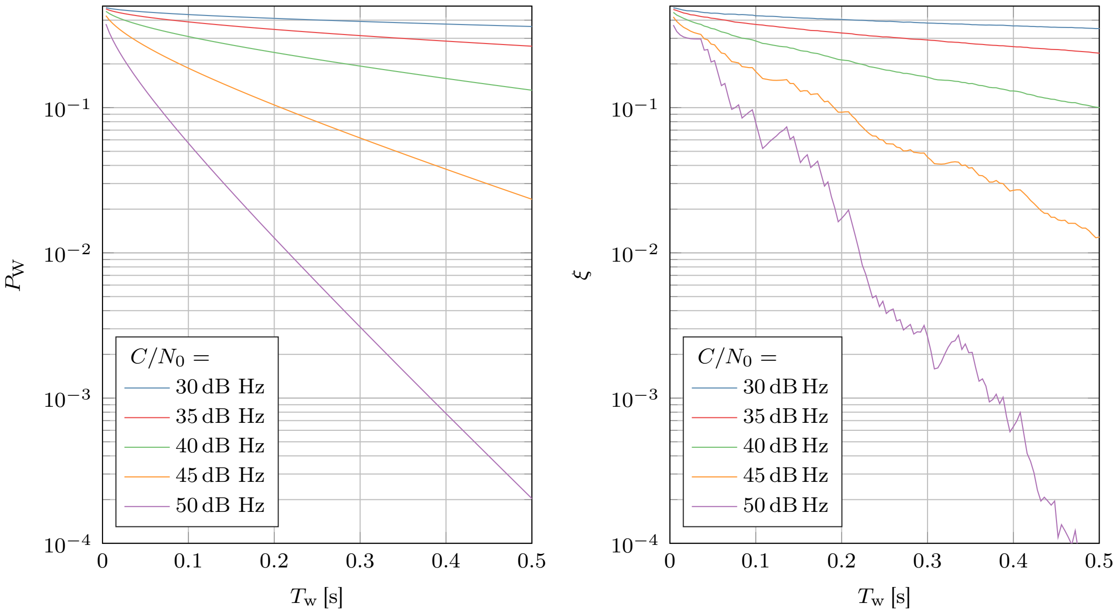


FIGURE 2.13: Probability that the SCER attack is not able to correctly estimate a chip: the statistical distribution of Eq. (2.30) (left) and the results from the signal simulator (right).

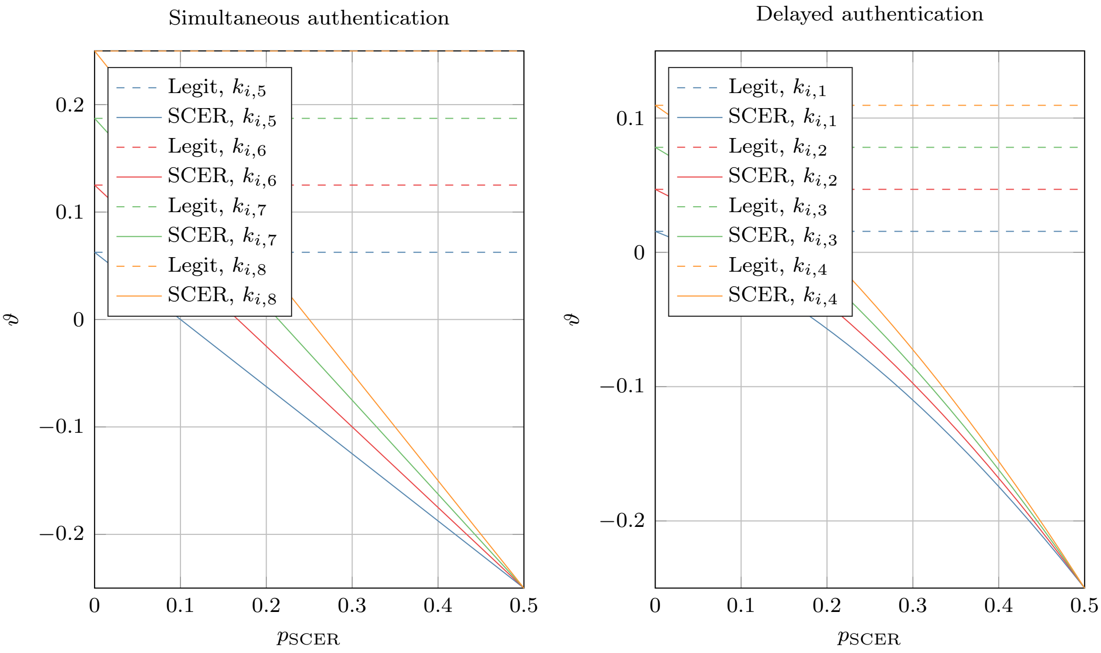


Figure 2.14: ϑ values achievable using the SCER attack as a function of p_{SCER}.

proposed an obfuscation strategy that considerably decreases the success probability of the attacker. Finally, we validated our analytical results by comparing them to the simulation ones and evaluated the performance of our verification algorithm by considering several scenarios and protocol parameters.

Chapter 3

Strategies for Multi-frequency Multi-constellation PVT assurance

3.1 Introduction

Spreading code authentication techniques like CAS and CHIMERA are designed to protect signals in a single bandwidth and a single constellation. On the other hand, a PVT solution can be considered as authentic only if it is computed by authenticated measurements. However, working with a single frequency receiver may lead to issues in terms of accuracy or availability. Moreover, the ranges are just periodically authenticated therefore, within each period, we would not be able to compute any secure PVT.

In this chapter we address the problem of PVT assurance: we propose a strategy where we offer a trade-off between security, accuracy and availability that allows the receiver to compute a *trusted* PVT. A trusted PVT is not authenticated since it is derived using both authenticated and non authenticated signals; still, we will show that, thank to the proposed consistency checks, we are also bounding the attacker capabilities.

More in detail, first, for the epochs when we have available the authenticated ranges, we discuss a step-wise approach in which the receiver exploits the protected signals as trusted anchors to enlarge the set of trusted signals through a series of consistency checks; the trusted signals will be later included in the PVT computation. Secondly, we consider the instants where no authenticated ranges are available and propose another consistency check which verifies the consistency of the new measurements comparing them to the previous ones, by exploiting the knowledge of the receiver dynamic. Finally, we show how these checks effectively limits the attacker capabilities. Part of the results discussed on this chapter has been presented in [20].

The rest of the chapter is organized as follows: in Section 3.2, we outline the system model and present the assumptions on the legitimate and the attacker receiver; in Section 3.3, we discuss the actual consistency check; in Section 3.4 we introduce the idea of security-aware protection levels, considering the performance of a potential attacker; Section 3.5 describes the experimental dataset and shows the numerical results; Section 3.6 outlines the inertial measurement unit (IMU)-based check used when no authenticated measurement is available; Section 3.7 draws the conclusions and describe the future work.

3.2 System Model and Assumptions

We consider a scenario where the legitimate receiver is capable of acquiring and tracking signals for multiple bands and constellations. The receiver has access to a SCA or

SCE service, such as CAS or CHIMERA (see Chapter 2). In this Chapter, we assume that authenticated signals cannot be tampered, therefore any PVT solution computed by using only authenticated signals is indeed authenticated, with MD $p_{\mathrm{MD}} = 0$ and consider scenarios where the attacks on protected signals as out of the scope of this work.

We suppose that, once every period of duration T_{Auth}, the receiver collects a set of authenticated ranging measurements, that will be used to compute the authenticated PVT: the discussion about the actual security of this authentication mechanism is out of the scope of this work, thus we simply assume that the attacker cannot tamper these measurements, with $p_{\mathrm{MD}} = 0$.

Still, a general discussion about the security of these techniques is already reported in [44]. Hence, we call *protected signals* the signals that are protected by the authentication services, e.g., Galileo CAS for signals in band E6C. We assume that the navigation messages are received from a trusted source (e.g., by using OSNMA): thus, satellite position, satellite clock correction, inter-frequency biases and atmospheric delay corrections computed by the receiver are authentic; to perform the checks described in Section 3.6, that evaluate the consistency of the signals over time, the receiver has to be equipped with an IMU sensor; no further assumption is done on the legitimate receiver hardware.

For the attacker we make the following general assumptions.

- The PVT of the legitimate receiver is known by the attacker (in advance).
- The thresholds used for all the consistency checks are public information.
- The attacker cannot tamper with the IMU measurements, the navigation messages and the authenticated signals.
- The noise of the channel between attacker and legitimate receiver is negligible.

We will also consider two cases: in the first, the attacker knows in advance the actual pseudoranges measured by the receiver; in the latter, we consider a more realistic case where the attacker computes only a (noisy) estimate of the (legitimate) pseudoranges.

More in detail, at each epoch t, the receiver collects a set of pseudoranges $\mathcal{R}(t) = \{R_s^{(f)}(t)\}$ where $R_s^{(f)}(t)$ is the range measured at time t from satellite $s \in \mathcal{S}$ from the signal with carrier frequency f; however, at time t_{auth}, thanks to the authentication service, the receiver obtained a set of authenticated ranges $\mathcal{R}_{\mathrm{auth}}(t)$, i.e., $\mathcal{R}_{\mathrm{auth}}(t) \subset \mathcal{R}(t)$.

3.3 Multi-Constellation Multi-Frequency PVT Assurance

In this Section we describe each consistency check used to enlarge the set of trusted signals. Being this a step-wise approach, the first checks will have lower thresholds and are expected to be more robust with respect to the latter ones.

3.3.1 Consistency Checks with Authenticated Ranges

For the instants where a set of authenticated measurements are available, the multi-frequency multi-constellation PVT assurance is proposed via a step-wise approach, where, starting from a PVT based on authenticated data and protected signals, more ranges are added at each step to the PVT computation. The steps are:

1. computing a trusted PVT using only protected signals;

2. adding unprotected signals transmitted by the same satellite and on the same carrier frequency as protected signals;
3. adding unprotected signals from the same satellite but on different carrier frequencies than protected signals;
4. adding signals from different systems.

Each step is described in detail on the next Sections.

3.3.2 Trusted PVT Using Only Protected Signals

The first step is to compute a trusted PVT using only protected signals. To compute this PVT solution, the receiver relies on authenticated navigation message and verified ranging measurements. For instance, in ACAS they used the Galileo I/NAV messages, authenticated by OSNMA, and ranging measurements encrypted by CAS.

3.3.3 Signals on the Same Carrier frequency

The receiver verifies the consistency between protected and open signals transmitted on the same carrier frequency. We consider two cases

Sgnals with the same modulation where both signals are transmitted with the same phase offset. Notice that, in this case, also the multipath envelope is the same, simplifying even more the cross-check. For instance E6B and E6C of Galileo E6 are both BPSK(5), and both are transmitted with the same phase offset.

Signals with different modulations , thus the cross check shall take into account also the different multipath envelopes, leading to a bigger difference between the ranging signals. An example can be GPS L1 C/A vs GPS L1C, i.e., BPSK(1) vs TMBOC(6,1).

It is worth noting that CAS will rely on a fully encrypted signal, while CHIMERA on time multiplexed watermarks. More in detail, neglecting the impact of the BOC(6,1) components, the processing of the open and the encrypted part of the L1C signal can be seen as a special case of signals with the same modulation, where the two signals share the same carrier and have a fixed phase and power relations. To achieve this, the receiver should be able to distinguish the open and encrypted components; hence, the signal should be reprocessed after the key has been disclosed. This is not the case for Galileo E6B, which can always be processed independently of E6C.

When signals with different modulations are considered, the specific tracking errors and the multipath envelopes of the ranging signals considered shall be taken into account in the acceptable pseudorange difference among the signals. For instance, the difference between two code delays with the same modulation can be modeled as

$$\tau_1 - \tau_2 \sim \mathcal{N}\left(0, \sigma^2_{\mathrm{DLL},1} + \sigma^2_{\mathrm{DLL},2}\right) \, , \tag{3.1}$$

where $\sigma_{\mathrm{DLL},i} = d_i/6$ and d_i is the correlator spacing of the ith signal [2, Chapter 5]. We call the chip duration of the ith signal, $T_{\mathrm{c},i}$, and c, the speed of light. Thus, we bound the pseudorange difference by using

$$|R_1 - R_2| \leq K_{\mathrm{B}}\sigma_{\mathrm{B}} \triangleq \gamma_{\mathrm{B}} \quad \text{with } \sigma_{\mathrm{B}} = c\sqrt{T^2_{\mathrm{c},1}\sigma^2_{\mathrm{DLL},1} + T^2_{\mathrm{c},1}\sigma^2_{\mathrm{DLL},1}} \, , \tag{3.2}$$

where K_{B} defines the *confidence level.* For instance, $K_{\mathrm{B}} = 3$ implies a 99.73% confidence interval or $p_{\mathrm{FA}} = 0.27\%$.

Beside the estimated code delay check, the receiver could check also the

Consistency of the Doppler frequency, estimated on the different signals;

Relative phase angle, i.e., the phase of the local replica used for the carrier wipe-off, since both signals are supposedly transmitted in phase;

Estimated C/N_0 or relative power among the signals. Since the transmitted power of the different signals is generally known, the receiver can check whether the received signal respects this relationship. The sensitivity of this check is heavily dependent on the C/N_0 estimator. It should also be noted that GPS has the capability to dynamically change the transmitted power of the signals [47], making this check more complex.

3.3.4 Signals from the Same Satellite and Different Frequency

In general, the error contributions are non-Gaussian: hence the model assumes an overbounding Gaussian distribution where

$$R_s^{(f_1)} - R_s^{(f_2)} \sim \mathcal{N}\left(\mathrm{IFB}^{(s)}, \sigma_{\mathrm{c}}^2\right) \quad \text{with } \sigma_{\mathrm{c}}^2 = \sigma_{\mathrm{iono}}^2 \left(1 - \frac{f_1^2}{f_2^2}\right) + \sigma_{\mathrm{mp},f_1}^2 + \sigma_{\mathrm{mp},f_2}^2 + \sigma_n^2 \ , \tag{3.3}$$

where the variance σ_{c}^2 includes the sum of the variances of the contributing errors, assuming they are independent random variables. More in detail, σ_{iono}^2 is the variance of the remaining ionospheric error after applying the ionospheric delay correction derived from the navigation message; $\sigma_{\mathrm{mp},f_i}^2$ is the variance of the multipath error for the signal with carrier frequency f_i, which is considered a priori different for each received signal [3]; finally, σ_n^2 is the variance of the remaining receiver noise errors (sampling, filtering, quantization) which, for the sake of simplicity, is modeled as the same for each signal. We recall that the tropospheric error is not dispersive thus it affects both measurements equally and therefore it is removed from the difference. Indeed, the receiver has at disposal the authenticated messages, hence it computes the atmospheric delay corrections $\hat{D}_{\mathrm{iono}}$ and $\hat{D}_{\mathrm{tropo}}$ for both bands (Appendix B), and the satellite clock corrections. The contribution to the mean is given by the inter-frequency bias, $\mathrm{IFB}^{(s)}$, that includes the biases for both the satellite and the receiver. Still, we assume the $\mathrm{IFB}^{(s)}$ to be known: actually, the inter-frequency bias between Galileo E1 and E5 is transmitted in I/NAV, while the one for E1 and E6 is disclosed by the CAS servers along with the RECS (see Section 2.2.2). Finally, we assume any additional (minor) error, e.g., the residual after the satellite clock correction, to be captured by other errors. Notice that, if the receiver has access to some form of trusted corrections, e.g., through satellite-based augmentation systems (SBAS), these can be used to reduce the error budget.

Based on these assumptions, the consistency check for signals transmitted by same satellite but on different frequency is

$$|R_s^{(f_1)} - R_s^{(f_2)}| \leq \mathrm{IFB}_{\mathrm{rx}}^{(s)} + K_{\mathrm{C}}\sigma_{\mathrm{C}} \triangleq \gamma_{\mathrm{C}} \ , \tag{3.4}$$

where K_{C} is the confidence level for check C.

3.3.5 Signals for Different GNSSs

The last check concern signals from different satellites, authenticated and unauthenticated. Signals from constellations which do not provide ranging protection mechanisms can be included in the PVT after validating the pseudorange errors from each signal. We assume the inter-system clock bias (ISCB) to be transmitted broadcast and authenticated. For instance, the GPS to Galileo time offset (GGTO) is transmitted by Galileo I/NAV and authenticated by OSNMA. Alternatively, it can be retrieved via some terrestrial assistance network, or computed in the receiver. If the ISCB is deemed legitimate, the receiver can compute the predicted pseudorange measurements given the receiver estimated position, the satellite positions, and the orbital parameters.

More in detail, for each satellite s, we compute the pseudorange estimate, $\hat{R}_s^{(f)}$, summing up

- the geometric range $\hat{r}_s$, computed as distance between the receiver position, obtained by using the trusted signals, and satellite s position, retrieved from the authenticated ephemerides;
- the atmospheric delays[1] and the satellite clock corrections obtained from the (authenticated) navigation messages;
- the receiver clock bias derived from the PVT and the ISCB.

The estimate is then computed to evaluate the estimation error $\Delta R_s^{(f)}$.

Assuming that the measurement residuals are modeled by a zero-mean Gaussian distribution with an a-priori variance estimated in the user equivalent ranging error (UERE), σ^2_{UERE}, and considering a confidence level K_{D}, the following consistency check can be performed (note that the frequency index has been omitted for simplicity):

$$|\Delta R_{s_i} - \Delta R_{s_j}| \leq K_{\mathrm{D}}\sigma_{\mathrm{D}} \triangleq \gamma_{\mathrm{D}}, \tag{3.5}$$

where $\Delta R^{(s_i)}$ is the measured minus predicted pseudorange for satellite s_i, where one of them is authenticated, and

$$\sigma^2_{\mathrm{D}} = \sigma^2_{\mathrm{UERE},j} + \sigma^2_{\mathrm{UERE},k} + \sigma^2_{\mathrm{ISCB},j,k}, \tag{3.6}$$

$$\sigma^2_{\mathrm{UERE},s} = \sigma^2_{\mathrm{SISE},s} + \sigma^2_{\mathrm{iono}} + \sigma^2_{\mathrm{tropo}} + \sigma^2_{\mathrm{mp},s} + \sigma^2_{n}, \tag{3.7}$$

with $\sigma^2_{\mathrm{SISE},s}$ including both the orbit and clock error of satellite s.

3.4 Security Aware Protection Levels

In this Section we investigate the use of the horizontal protection level (HPL) and the vertical protection level (VPL) to a) evaluate the reliability of the signal in the different settings, e.g., PVT computed by using only the authenticated satellites or using all the available signals, and b) to introduce the concept of *security aware protection levels*, where the user is provided of a bound that takes into account both integrity and security, with the idea to further develop it in future work. HPL and VPL model a confidence region where the receiver can safely (i.e., with high probability) assume to be in; these are defined as

$$\mathrm{HPL} = k_{\mathrm{h}} d_{\mathrm{h}} \ , \quad \mathrm{VPL} = k_{\mathrm{v}} d_{\mathrm{v}} \tag{3.8}$$

[1] As pointed out in Appendix B, the ionospheric delay corrections in different bands are related, e.g., see (B.2) for E1 and E6.

where k_{h} and k_{v} are constants [48,49], d_{h}^2 and d_{h}^2 represent "the variance of the model error distribution that upperbounds the true error distribution", respectively along the horizontal plane and the vertical axis [48]. Considering, for instance the vertical axis, it yields

$$d_{\mathrm{v}}^2 = \sum_{i=1}^{N} s_{\mathrm{v},i} \sigma_i^2 \ , \tag{3.9}$$

here σ_i^2 is the variance related to the ith residual pseudorange error[2] and $s_{\mathrm{v},i}$ is a (vertical) geometric factor. More detail about the derivation of this values can be found in [48].

A possible way to compute the security aware protection level would be to include the thresholds used in the checks in the protection level computation, as they effectively represent the worst-case scenario for the error accepted. For the sake of simplicity, assuming that these thresholds are independent from the residuals computed for the integrity protection levels, we may write the security-aware protection levels as

$$\widehat{\mathrm{HPL}} = \mathrm{HPL} + f_{\mathrm{H}}(\boldsymbol{\gamma}) \ , \tag{3.10}$$

$$\widehat{\mathrm{VPL}} = \mathrm{VPL} + f_{\mathrm{V}}(\boldsymbol{\gamma}) \ , \tag{3.11}$$

where we considered the set of thresholds $\boldsymbol{\gamma} = \{\gamma_s^{(f)}\}$, with each entry of γ being a security threshold associated to each pseudorange used in the final PVT; hence, we pick $\gamma = 0$ for the authenticated ranges, $\gamma = \gamma_{\mathrm{B}}$ if the pseudorange was included in the pseudorange using the check (3.2) and so on. $f_{\mathrm{H}}(\cdot)$ and $f_{\mathrm{V}}(\cdot)$ are geometric transformations projecting the security constraint from the pseudorange domain to the position domain, either on the horizontal plane or the vertical axis. While the complete derivation of functions $f_{\mathrm{H}}(\cdot)$ and $f_{\mathrm{V}}(\cdot)$ is left for future work, in the next Section we will consider an alternative strategy to upperbound the influence of a (possible) attacker in the computed PVT.

3.4.1 Position uncertainty from the chosen thresholds

Considering the set of thresholds $\boldsymbol{\gamma}$, our aim is to compute the maximum displacement, $\|\Delta \boldsymbol{x}\|_{\max}$, that the attacker can induce to the legitimate receiver without raising an alert.

The actual received authenticated pseudoranges can be modeled as

$$R_{\mathrm{Auth}}^{(s)} = \tilde{R}_{\mathrm{Auth}}^{(s)} + \omega \ , \tag{3.12}$$

where $\tilde{R}_{\mathrm{Auth}}^{(s)} \in \mathcal{R}_{\mathrm{auth}}$ is the noiseless authenticated range measurement and $\omega \sim \mathcal{N}(0, \sigma_{\mathrm{UERE}}^2)$ is the noise component. Next, we consider two scenarios: first we consider a more strict scenario where the attacker knows the actual authenticated ranges measured by the receiver; secondly, we will consider a more realistic scenario where this assumption does not hold, thus the receiver only knows the statistics of ω, but not the actual realization.

We remark that the former case is hard to be met in practice. Still, it corresponds to a *self spoofing* scenario: for instance, if a GNSS-based device is used for regulating the user's position or velocity, the attacker could be a malicious user that wants to violate the rules without being detected by the regulating device. Indeed, in these cases, the victim receiver and the attacker receive the same signals. While, for simplicity,

[2] If no augmentation system is used $\sigma_i^2 = \sigma_{\mathrm{UERE}}^2$

we will refer to this as the self-spoofing scenario, it may be possible for a powerful attacker to meet this condition even in a different context.

Self-spoofing scenario The attacker transmits signals such that the receiver will compute the pseudorange related to satellite s (we drop the frequency f for ease of reading) as

$$R_{\mathrm{E}}^{(s)} = R_{\mathrm{Auth}}^{(s)} + \Delta R_{\mathrm{E}}^{(s)} = \tilde{R}_{\mathrm{Auth}}^{(s)} + \omega + \Delta R_{\mathrm{E}}^{(s)} \,. \tag{3.13}$$

We remark that this requires the attacker to know the exact noise realization ω, not just its statistics.

Hence, in general, any of the presented consistency checks will surely pass (i.e., $p_{\mathrm{MD}} = 0$) if

$$|R_{\mathrm{E}}^{(s)} - R_{\mathrm{Auth}}^{(s)}| = |\Delta R_{\mathrm{E}}^{(s)}| \leq \gamma_s \,. \tag{3.14}$$

Thus, during the PVT, after the linearization procedure [2, Chapter 2], the set pseudorange biases $\Delta\boldsymbol{R} = \{\Delta R_s\}$ is related to the error in the earth-centered earth-fixed (ECEF) position domain by

$$\boldsymbol{H}\Delta\boldsymbol{x} = \Delta\boldsymbol{R} \,, \tag{3.15}$$

where $\boldsymbol{H}$ is the projection matrix

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{e}_1 & 1 \\ \boldsymbol{e}_2 & 1 \\ \cdots & \cdots \\ \boldsymbol{e}_N & 1 \end{bmatrix} . \tag{3.16}$$

Each component $\boldsymbol{e}_i$ is a unitary vector pointing from ith satellite to the receiver antenna.

Since, in general, $\boldsymbol{H}$ has $N \geq 4$ rows, in a least squares (LS) fashion, we resort the pseudo-inverse and write $\Delta\boldsymbol{x}$ as a function of the ΔR,

$$\Delta\boldsymbol{x}(\Delta\boldsymbol{R}) = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\Delta\boldsymbol{R} \,. \tag{3.17}$$

The previous equation relates range biases, $\Delta\boldsymbol{R}$ to the PVT solution bias, $\Delta\boldsymbol{x}$. Thus, we exploit this relation to bound the impact of a potential attack: in particular solve the following optimization

$$\arg\max_{\Delta\boldsymbol{R}} \|\Delta\boldsymbol{x}(\Delta\boldsymbol{R})\| = \arg\max_{\Delta\boldsymbol{R}} \sqrt{((\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\Delta\boldsymbol{R})^{\mathrm{T}}\left((\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\Delta\boldsymbol{R}\right)} \,, \tag{3.18}$$

$$\text{with } |\Delta_{\mathrm{E}} R^{(s)}| \leq \gamma_s \quad \forall s \in \mathcal{S} \,. \tag{3.19}$$

Notice that (3.18) can be written as the symmetric quadratic form

$$\arg\max_{\Delta\boldsymbol{R}} \|\Delta\boldsymbol{x}(\Delta\boldsymbol{R})\| = \arg\max_{\Delta\boldsymbol{R}} \sqrt{\Delta\boldsymbol{R}^{\mathrm{T}}\boldsymbol{H}\left((\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\right)^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\Delta\boldsymbol{R}} \tag{3.20}$$

$$= \arg\max_{\Delta\boldsymbol{R}} \sqrt{\Delta\boldsymbol{R}^{\mathrm{T}}\left(Q^{\mathrm{T}}Q\right)\Delta\boldsymbol{R}} \,, \tag{3.21}$$

thus (3.18) is concave, and the problem is solvable.

Noisy pseudoranges scenario Here, we consider the scenario where the attacker does not know the actual authenticated range, but only its statistical distribution, i.e., the attacker does not know the exact realization of ω in (3.12): under this condition, (3.14) becomes

$$\left|R_{\mathrm{E}}^{(s)} - \tilde{R}_{\mathrm{Auth}}^{(s)}\right| = \left|\Delta R_{\mathrm{E}}^{(s)} + \omega\right| \leq \gamma_s \,. \tag{3.22}$$

The noise is assumed to be $\omega \sim \mathcal{N}(0, \sigma_{\mathrm{UERE}}^2)$, the MD probability is now

$$P_{\mathrm{MD}}(\Delta R_{\mathrm{E}}^{(s)}) = \operatorname{erfc}\left(\frac{-\gamma_s - \Delta R_{\mathrm{E}}^{(s)}}{\sigma_{\mathrm{UERE}}}\right) - \operatorname{erfc}\left(\frac{\gamma_s - \Delta R_{\mathrm{E}}^{(s)}}{\sigma_{\mathrm{UERE}}}\right), \tag{3.23}$$

where erfc$(\cdot)$ is the *complementary error function*. The attacker can still minimize (3.18) but, now, it has a constraint based on the success probability of the attack itself. Indeed, as a rule of thumb, with respect to the self spoofing scenario, the attacker will be more conservative, inducing small position shifts.

3.5 Numerical Results

This Section collects the numerical results: first, we will introduce the experimental dataset, used to validate the consistency checks; secondly, we evaluate the performance of the checks; in the last part, we evaluate the attacker performances.

3.5.1 Experimental Dataset

The proposed checks were tested by using an experimental dataset: these measurements were collected by a high grade mass market receiver, mounted on a van travelling around Rotterdam (see Figure 3.1). In total, our experimental dataset is composed of approximately 5 hours of observables, collected with a frequency of 1 Hz. Our receiver was a multi-frequency multi-constellation receiver that collected measurement from:

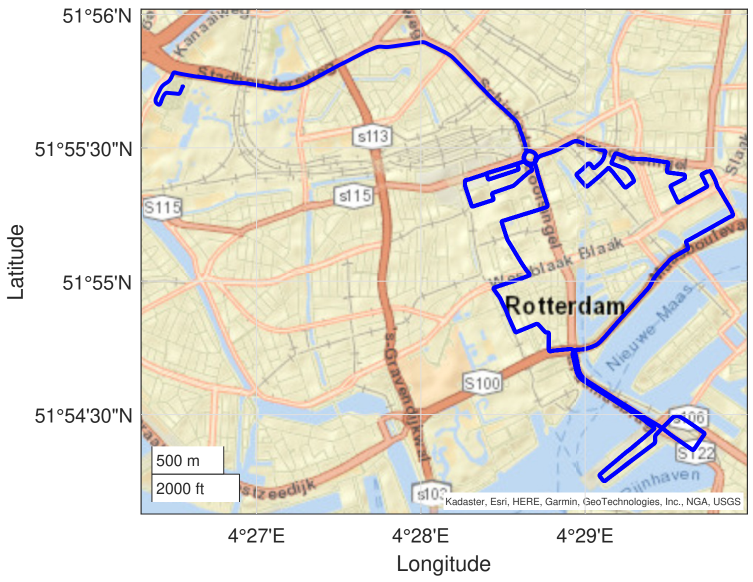


FIGURE 3.1: Route taken for the data collection: we mounted a multi-frequency multi-constellation receiver on a van travelling through Rotterdam. We collected approximately 5 h hours of measurements.

- Galileo: E1BC, E5a, E5b;
- GPS: L1 C/A, L1 P(Y), L2 P(Y), L5;

- GLONASS: L1 C/A, L2 C/A;

- BeiDou: B1, B2.

We remark that we do not have any measurement in the E6 band: thus, in this example, without loss of generality, we will assume E1BC signals to be protected, i.e., authenticated by a service such as CAS. All the post-process operations, such as the computation of the protection level, is done by using the Septentrio software.

Before analyzing the performance of the checks, we report the Stanford diagrams for the vertical direction, plotting the VPL against the actual vertical error level (VEL), measured by using the ground truth, computed in three different scenarios:

1. Only Galileo E1BC, i.e., by using only the originally authenticated signals (Figure 3.2, top left);

2. GPS L1, Galileo E1BC and E5 (Figure 3.2, top right);

3. GPS, Galileo, GLONASS and Beidou (Figure 3.2, bottom).

The protection levels presented where computed by the Septentrio software itself. Note that the vertical alert limit (VAL) has been arbitrarily set to 100 m only for presentation purposes. As expected, we notice that moving from scenario 1. to 3., the obtained PVT becomes both more available and more accurate: the percentage of epochs in nominal conditions, i.e. VPL $\leq$ VAL and VPL $\geq$ VEL, grows from 71.98% to 97.83%;, while in total, we go from 9722 to 10529 epochs with an available PVT.

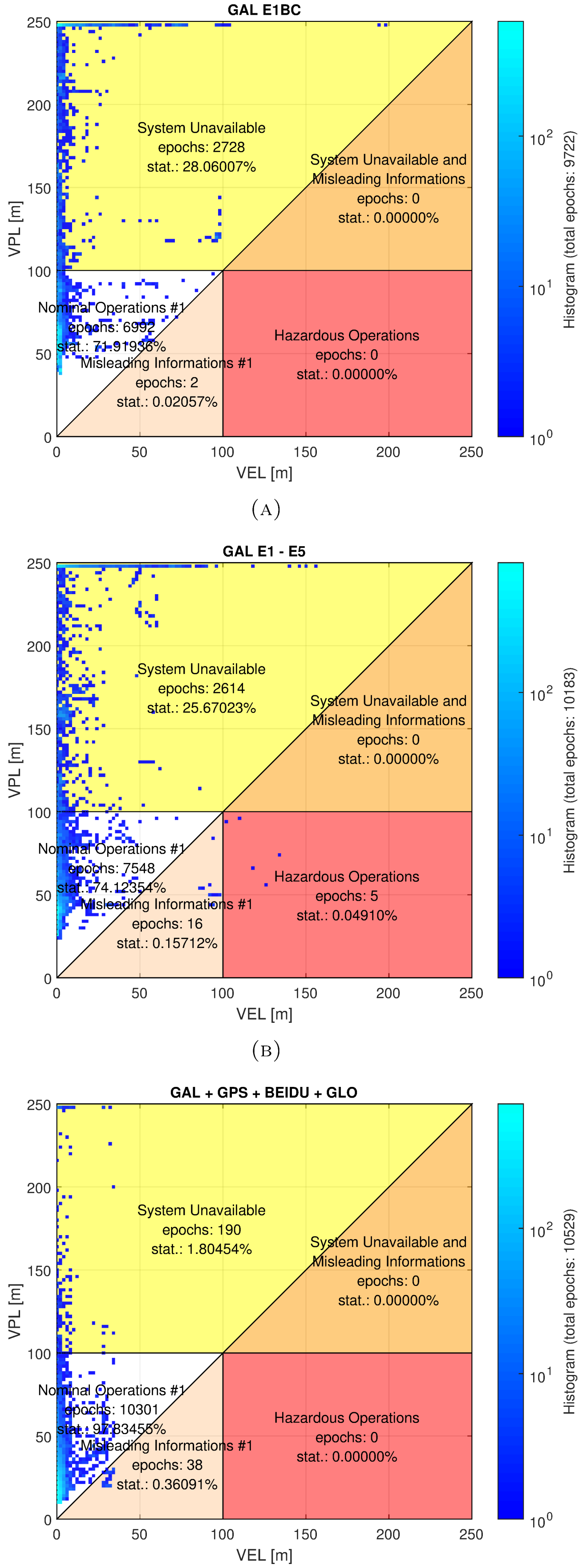
3.5.2 Performance of the Checks

We evaluated some of the proposed checks by using the dataset described in 3.5.1: in particular we verify that the statistical models are indeed able to predict the legitimate ranges.

We remark that the integrity per se, even without spoofing, is an unresolved challenge for such environments due to local effects, e.g., in non-line-of-sight (NLOS) conditions, so we will consider separately the case for open-sky/rural environments, and open sky plus urban. It should also be noted that these results, as well as the theoretical models are rather preliminary, and a more thorough modeling and experimental characterization would be required to derive accurate conclusions.

As previously pointed out, in the dataset we do not have measurement from E6C, thus, in this example, we assume that all Galileo E1BC measurements have been successfully validated, therefore E1BC signals will be used as trusted anchors for the other checks. We will focus on checks described in Section 3.3.4 and 3.3.5.

Table 3.1 summarizes the standard deviations used to compute the thresholds γ_{C} and γ_{D}, with confidence level $K_{\mathrm{C}} = K_{\mathrm{D}} = 3$. The estimation of the residual error due to the ionospheric delay in the E1BC and the L1 band is derived in [4], using the NeQuick ionospheric delay correction model. As anticipated in Section 3.3.4, if the authenticated correction would be available, it would be possible to exploit a more precise estimation of the delays such as the ones reported in [2, Chapter 7], therefore reducing the error residuals. This is however out of the scope of this work. Concerning the multipath, starting from the open sky experimental measurements of NLOS, we consider a multipath error by picking $\sigma_{\mathrm{MP,E1C}} = 3\,\mathrm{m}$ and $\sigma_{\mathrm{MP,E5}} = \sigma_{\mathrm{MP,L1CA}} = 0.3\,\mathrm{m}$. These values should be adjusted for urban environment in future works.



(A)

(B)

(C)

Figure 3.2: Stanford diagram obtained by using only Galileo E1BC (A), Galileo E1 and E5 (B) and all the received signals (C), with VAL = 100 m.

TABLE 3.1: Standard deviation used to model the statistical distribution error residuals. Parameters from [2–6]

$\sigma_{\text{iono,E1}}$	3.5 m	$\sigma_{\text{MP,E1}}$	3 m
σ_{tropo}	0.2 m	$\sigma_{\text{MP,E5}}$	0.3 m
σ_{ISCB}	1.5 m	$\sigma_{\text{UERE,L1}}$	7.1 m
σ_{n}	0.1	$\sigma_{\text{UERE,E1}}$	7.1 m

3.5.2.1 Signals from the Same Satellite and Different Frequency

We want to exploit check (3.4) to authenticate the signals in E5 using signals in E1 as trusted anchors. Considering confidence level $K_{\text{C}} = 3$, plugging the parameters of Table 3.1 we get $\gamma_{\text{C}} \approx 11.37\,\text{m}$. We report the results of this check in Table 3.2: in particular for each observed Galileo satellite, we report the number of epochs in which the satellite was available, the number of false alarms, i.e., how many times the satellite did not pass the consistency check, the (empirical) standard deviation of the pseudoranges differences σ_{C} and the false alarm probability p_{FA}, computed as ratio of failures vs number of epochs. From the latter, we can see that excluding satellite E15, we always get the false alarm probability is comparable with the expected 3-sigma confidence level. As will be highlighted in the next Section, we get that these model matches better the rural than the urban scenario: most of the false alarms are in fact related in this latter case. This is exactly the case of the satellite E15, that is present only in the urban scenario part of the experiment.

TABLE 3.2: Results for the check C between Galileo E1BC and E5 signals for $K_{\text{C}} = 3$ and $\gamma_{\text{C}} \approx 11.37\,\text{m}$ in terms of number of false alarms, total number of epochs in view, measured standard deviation and P_{FA}.

SVID	E02	E03	E07	E08	E11	E12	E15	E19	E21	E25	E27	E30	E36
False Alarms	7	0	52	25	0	0	126	0	0	0	11	0	4
Epochs	16624	5073	10706	8511	3954	2556	3808	684	563	7238	6534	18223	10257
$\hat{\sigma}_{\text{D}}$ [m]	1.31	0.97	2.96	3.08	0.90	0.68	5.86	0.91	0.59	0.93	4.17	0.82	1.12
P_{FA}[%]	0.04	0	0.48	0.29	0	0	3.30	0	0	0	0.17	0	0.04

3.5.2.2 Signals from different GNSSs

By using check C we are able to compute a trusted PVT solution using all the received Galileo signals. We want now to check the consistency of the GPS L1 C/A signals by exploiting check D. By using the values reported in Table 3.1 and confidence level K_{D}, for check D we compute $\gamma_{\text{D}} = 30.04\,\text{m}$.

Results have been reported in Tables 3.3 and 3.4 in terms of number of epochs, number of false alarms, standard deviation and false alarm probability P_{FA}. In particular, Table 3.3 shows the results only for the open sky scenario, i.e., for the measurements collected before entering to Rotterdam. In the rural scenario, the results are in line with the expected performance. On the other hand, in challenging environment such as the Rotterdam city center, the presence of NLOS and other non-modeled impairments degrade the performance, highlighting the need for a tuning the model parameters according to the environment.

Table 3.3: Results for the check D with $\gamma_{\mathrm{D}} = 30.04\,\mathrm{m}$, between the Galileo E02 and GPS L1 signals in terms of number of false alarms, total number of epochs in view, measured standard deviation and P_{FA} in open sky scenario.

SVID	G10	G12	G15	G17	G24	G29	G39	G31
False Alarms	0	0	0	0	27	0	0	0
Epochs	3178	4820	4850	4992	4887	4881	4806	4731
$\hat{\sigma}_{\mathrm{C}}$ [m]	1.56	1.53	1.03	1.16	4.16	1.63	1.09	2.37
P_{FA}[%]	0	0	0	0	0.55	0	0	0

Table 3.4: Results for the check D with $\gamma_{\mathrm{D}} = 30.04\,\mathrm{m}$, between the Galileo E02 and GPS L1 signals in terms of number of false alarms, total number of epochs in view, measured standard deviation and P_{FA} in both open sky and urban scenario.

SVID	G10	G12	G15	G17	G24	G29	G39	G31
False Alarms	118	3	3	25	112	49	326	0
Epochs	4806	15812	5385	6699	7619	9583	12782	5282
$\hat{\sigma}_{\mathrm{D}}$ [m]	19.37	2.33	1.33	4.06	13.64	5.08	12.21	2.47
P_{FA}[%]	2.46	0.02	0.06	0.37	1.47	0.51	2.55	0

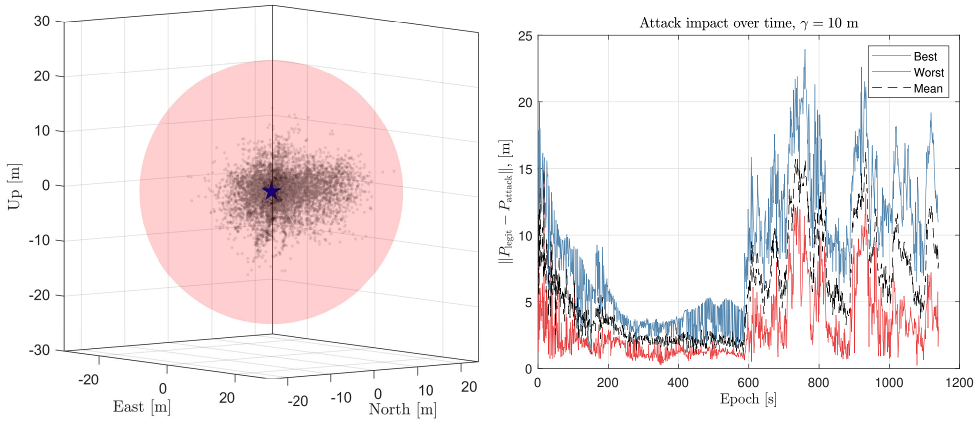
3.5.3 Under Attack Scenario

We start the analysis of the attacker behavior considering a simpler case: given the model (3.13), we run a Montecarlo simulation using the experimental dataset, with the aim of empirically quantify the attacker's freedom to manipulate the position computed with the proposed check. In particular we draw each pseudoranges deviation as $R_{\mathrm{E}}^{(s)} \sim \mathcal{U}([-\gamma_s, \gamma_s])$.

For simplicity, we considered only Galileo and GPS signal, in all the available bandwidths, fixing $\gamma_s = \gamma = 10\,\mathrm{m}$ for all the non-authenticated satellites. We considered the first 20 min of the experimental measurements and iterated the Montecarlo simulation for 30 times, thus including 36000 epochs.

Figure 3.3 reports the difference between the computed position and the ground truth in the East, North, Up (ENU) reference frame. The difference is a combination of the error from the original dataset and the one induced by the attacker. Figure 3.3a shows that the position errors can be contained in a sphere of radius 23.54 m: this maximum induced error is reached at the instant when we also have the highest error (with respect to the ground truth) in the legitimate case, which is around 15 m; this low accuracy is associated to the second part of the considered experimental dataset (Figure 3.3b). Moreover, it is worth noting that the considered attacks do not lead to a PVT more than the 43% of epochs. This first investigation shows that a) the consistency checks proposed effectively limit the ability of the attacker to arbitrarily manipulate the position of the victim receiver and b) the victim receiver visibility influences the attacker capability. Indeed, a receiver in low visibility conditions is more vulnerable.

Attacker for Self Spoofing Scenario We now consider the self spoofing scenario where the attacker knows the actual ranges measured by the receivers. Thus, it is



(A)

(B)

FIGURE 3.3: Position error induced by the random attack, obtained considering 30 Montecarlo simulations for the first 1200 epochs of the experimental data: in the left scatter plot of the induced position errors; on the right best (blue), worst (red) and mean (dashed) induced position error.

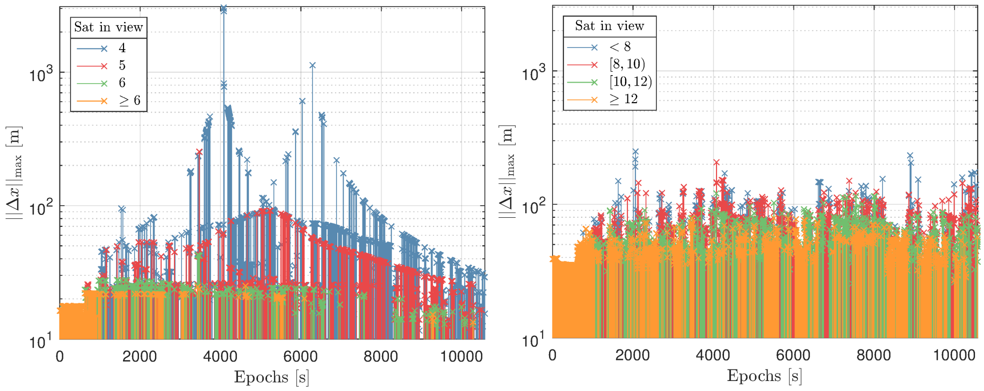
able to generate signals following the rule descried by (3.14). Figure 3.4 reports the results considering the whole experimental dataset for the scenarios where 1) only Galileo signals (E1, E5) are available, therefore with $\gamma_s^{(f)} = 11.37\,\text{m}$ (Figure 3.4a), and 2) by using both GPS and Galileo, i.e., with $\gamma_s^{(f)} = 30.04\,\text{m}$ (Figure 3.4b). Figure 3.5 reports the cumulative distribution functions (CDFs) of the maximum positioning error induced by the attacker $\|\Delta\boldsymbol{x}\|_{\max}$ for the same two scenarios. From these figures, we conclude that, as initially pointed out in the previous paragraph,

- we are indeed bounding the attacker capabilities, therefore even if we are including non authenticated measurements the attacker cannot freely change the receiver PVT;
- the maximum undetected position shift $\|\Delta\boldsymbol{x}\|_{\max}$ greatly decrease with the number of satellite in views.

These results also demonstrate that the defense strategy should take into account the receiver visibility conditions: for instance, in our scenario, when the receiver had in view 6 or more satellites, it was not necessary to include the GPS satellites in the PVT computation; conversely, for the epoch where only few Galileo satellites were in view, the non protected GPS signals should be included. Still, as highlighted in both [20] and the previous section, the check from Galileo to GPS (check 4) has still some margins of improvement: indeed, improving the performance of this check will make more convenient to include signals from different constellations.

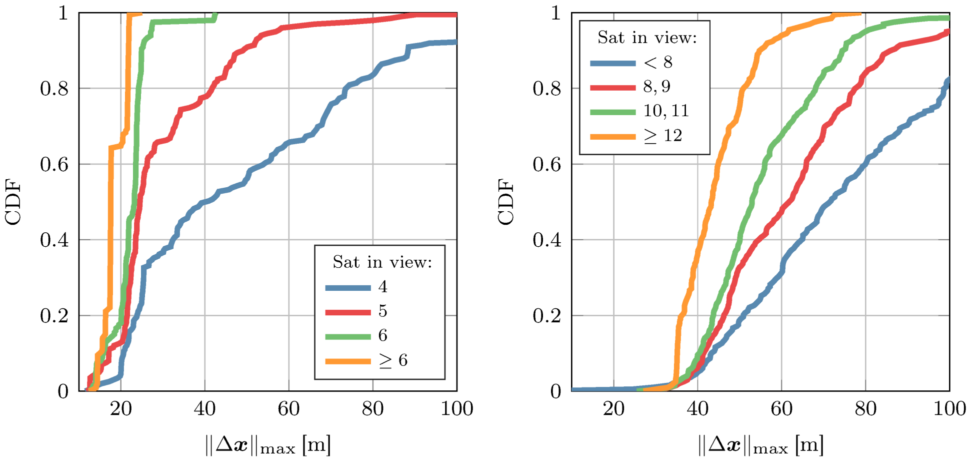
Noisy pseudoranges scenario We now consider the more general scenario where attacker and victim receiver are far from each other, thus, the attacker does not know the actual receiver but just its statistics.

Figure 3.6 shows the MD probability as a function of the chosen range alteration, $p_{\text{MD}}(\Delta R)$, considering $\sigma_{\text{UERE}} = 7.1\,\text{m}$ for various thresholds: slightly increasing the value of $\Delta R_{\text{E}}^{(s)}$ leads to an high impact on the success probability of the attacker. For instance, considering the threshold set for the Galileo plus GPS scenario, picking exactly $\Delta R_{\text{E}}^{(s)} = \gamma_s$ cause the MD to be already $p_{\text{MD}} = 0.5$. Hence, to have a reasonable



(A) (B)

FIGURE 3.4: Max position deviation $\|\Delta \boldsymbol{x}\|_{\max}$ induced by the attacker as a function of the number of satellites in view, considering only Galileo (left) and Galileo + GPS (right).



(A) (B)

FIGURE 3.5: CDF of $\|\Delta \boldsymbol{x}\|_{\max}$ induced by the attacker as a function of the number of satellites in view considering only Galileo (left) and Galileo + GPS (right).

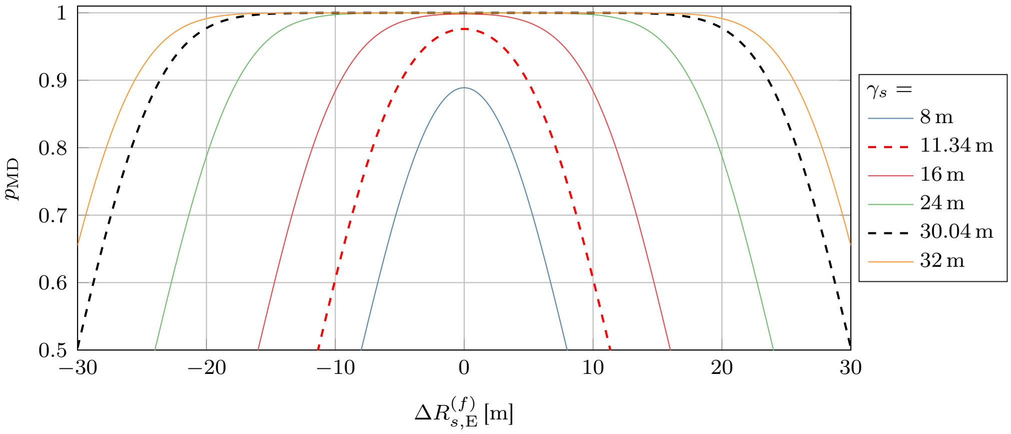


FIGURE 3.6: Miss detection probability p_{MD} computed by the attacker as a function of the induced shift in the range, $\Delta R^{(f)}_{s,\mathrm{E}}$ for various threshold values.

success probability, the attacker has to alter the pseudoranges by piking a stricter constraint, $|\Delta R^{(f)}_s| \ll \gamma^{(f)}_s$; hence, we successfully limited the attacker capabilities.

3.6 Consistency Check for Time $t \geq t_{\mathrm{Auth}}$ for Receivers with Known Dynamic

In this Section we consider the instants $t > t_{\mathrm{Auth}}$ i.e., those instants after the disclosure of the authenticated measurements and before the disclosure of the next. Indeed, if we have no current authenticated measurements we have no anchor to be used to perform the checks described in Section 3.3.1. Formally, we consider those instants t, such that $t = t_{\mathrm{Auth}} + \Delta t$ and assume to have

- a set of previously authenticated (or trusted[3]) pseudoranges, $\mathcal{R}_{\mathrm{Auth}}(t - \Delta t)$;
- a new measurement (eventually a set) $\hat{R}^{(f)}_s(t)$;
- no currently authenticated measurement, i.e.,

$$\mathcal{R}_{\mathrm{Auth}}(t) = \emptyset \, . \tag{3.24}$$

Our aim is then to find a consistency check that exploits the authenticated pseudoranges $R^{(f)}_{\mathrm{Auth},s}(t - \Delta t) \in \mathcal{R}_{\mathrm{Auth}}(t - \Delta t)$ to decide if measurement $R^{(f)}_s(t)$ can be considered as trusted or not. In general, we can write such check as,

$$\left| R^{(f)}_{\mathrm{Auth},s}(t - \Delta t) - R^{(f)}_s(t) \right| \leq \gamma^{(f)}_s(t) \; , \tag{3.25}$$

where $\gamma^{(f)}_s(t)$ is the threshold for this check. Notice that, as discussed in Section 3.4.1, $\gamma^{(f)}_s(\Delta t)$ indirectly increases the legitimate receiver position uncertainty.

For small value of Δt, it yields

$$\frac{1}{\Delta t}\left(R^{(f)}_{\mathrm{Auth},s}(t - \Delta t) - R^{(f)}_s(t) \right) \approx \frac{\partial}{\partial t} R^{(f)}_s(t) \triangleq \dot{R}^{(f)}_s(t), \tag{3.26}$$

[3] eventually built using the previously-described checks

which is usually called *range rate.*

Next, from (B.1) we can decompose the range rate as

$$\dot{R}_s^{(f)}(t) = \dot{r}(t) + c\left(\frac{\partial}{\partial t} dT_{\mathrm{rx}}(t) - \frac{\partial}{\partial t} dT_s(t)\right) + \varepsilon', \tag{3.27}$$

where $\dot{r}(t)$ is the derivative of the geometric range and the second term is the difference between the receiver and satellite clock drifts. The term ε' models the residual errors: in particular it also captures the error due to the fact that atmospheric delays and multipath are only approximately constant over time. The derivative of the geometric range is

$$\dot{r}(t) = (\boldsymbol{v}_s(t) - \boldsymbol{v}_{\mathrm{rx}}(t))^T \frac{\boldsymbol{P}_s(t) - \boldsymbol{P}_{\mathrm{rx}}(t)}{\|\boldsymbol{P}_s(t) - \boldsymbol{P}_{\mathrm{rx}}(t)\|} = (\boldsymbol{v}_s(t) - \boldsymbol{v}_{\mathrm{rx}}(t))^T \boldsymbol{e}_s(t) = v_{\mathrm{LOS}}(t) \tag{3.28}$$

where $\boldsymbol{e}_s(t)$ is the unitary vector pointing from satellite s to the receiver (antenna) at time t while $\boldsymbol{v}_s(t)$ and $\boldsymbol{v}_{\mathrm{rx}}(t)$ are respectively the satellite and the receiver velocity in ECEF coordinates. Finally, $v_{\mathrm{LOS}}(t)$ is the velocity projected along line-of-sight (LOS) direction.

Thus, for the legitimate case it holds

$$\dot{R}_s^{(f)}(t) = \boldsymbol{v}_s^{\mathrm{T}}(t)\boldsymbol{e}_s(t) - \boldsymbol{v}_{\mathrm{rx}}^{\mathrm{T}}(t)\boldsymbol{e}_s(t) + c\left(\frac{\partial}{\partial t} dT_{\mathrm{rx}}(t) - \frac{\partial}{\partial t} dT_s(t)\right) + \varepsilon'. \tag{3.29}$$

Notice that satellite position, velocity and clock drift are derived from the I/NAV message, which are authenticated by OSNMA. The only unknown term is the receiver velocity $\boldsymbol{v}_{\mathrm{rx}}(t)$ that can be derived from the IMU measurements.

By plugging the results in (3.25) we get

$$\begin{aligned} \left|R_{\mathrm{auth},s}^{(f)}(t-\Delta t) - R_s^{(f)}(t)\right| &= \left|R_{\mathrm{auth},s}^{(f)}(t) - \left(R_{\mathrm{auth},s}^{(f)}(t-\Delta t) + \Delta t\, \dot{R}_s^{(f)}(t)\right)\right| = \\ &= \left|\Delta t\, \dot{R}_s^{(f)}(t)\right| < \gamma_s^{(f)}(t). \end{aligned} \tag{3.30}$$

As for the previous check, threshold $\gamma_s^{(f)}(t)$ can be decided to match the desired FA or MD probability: in particular, it has to take into account the statistic of the term ε' and the tolerance on the velocity estimation of the IMU, $\boldsymbol{v}_{\mathrm{rx}}(t)$.

Finally, knowing the value of $\gamma_s^{(f)}$ by using again the consideration of Section 3.4.1, it is possible to update also $\|\Delta\boldsymbol{x}\|_{\max}$, predicting the capabilities of the attacker. It is up to the receiver to decide if it is convenient to keep an old measurement, which has an high associated threshold $\gamma_s^{(f)}(t)$, or to drop it, reducing the set of available measurements but limiting the attacker capabilities. We left the testing of this strategy for future works.

3.7 Conclusions

In this chapter we have proposed a series of consistency checks that allows to extend the set of trusted signals, i.e., those signals can be used to compute a trusted PVT. First, we use the authenticated measurements as anchors to enlarge the set of trusted measurements; secondly, when we have no authentic measurements available, we exploit an IMU sensor to verify the consistency of the new measurements with respect to the previous ones. Moreover, by exploiting the analytical relationship between the thresholds used in the consistency check and the position accuracy, the receiver can evaluate the trade-off between PVT accuracy and the maximum position shift induced by the attacker.

In future works we will investigate the statistical model of the pseudoranges for the urban scenario, obtaining a more robust check for the multi-constellation case and use experimental data to test the performance of the check based on the pseudorange prediction.

Chapter 4

Authenticated Timing Protocol Based on Galileo ACAS

Along with positioning, timing and synchronization are both key services provided by GNSSs. GNSSs allow users to obtain timing with nanosecond-level accuracy [37]. Moreover, this allows also users at different location far from each other to be synchronized: the user just have to be synchronized to the GNSS reference time, e.g., coordinated universal time (UTC) or the Galileo system time (GST). Hence several sectors, from financial institutions that use GNSS to timestamp transactions to Industry 4.0 or IoT applications, rely on GNSS for these operations.

The main standards for the dissemination of time and frequency over digital networks are the network time protocol (NTP) and the precision time protocol (PTP): the NTP achieves an accuracy within tens of milliseconds over the Internet, while it can be less than 1 ms in local area networks (LANs) with ideal network conditions [50]; the PTP provides better accuracy, from hundreds of ns to µs [51]. Hence, to achieve ns-level accuracy we can first synchronize a local server using GNSS; next, we disseminate the time corrections over the LAN by using either NTP or PTP.

The previously mentioned services require the timing service to be accurate, reliable, and trustworthy: on the other hand GNSS open signals are vulnerable to spoofing, in particular to time spoofing or time-push attacks. On the other hand, a service relying only on authenticated GNSS signals provides an authenticated timing service: indeed, the authentication provides trustfulness as they incorporate specific features that cannot be predicted or falsified by malicious attackers, since authentication-enabled receiver can interpret these characteristics to distinguish authentic signals from forgeries. As mentioned in Chapter 2, the authentication can take place at two complementary levels: at the data level, i.e., on navigation messages, and at the ranging level, on pseudoranges between the satellite and receiver.

Still, working at the ranging level means to secure the pseudorange measurements: the time taken by the signal to reach the receiver, which is estimated using the pseudoranges themselves, thus becoming fully reliable. SCE techniques are the most reliable option to limit access to GNSS signals, as they render the spreading code unpredictable: some of the SCE-type solutions have been described in Chapter 2.

On the other hand, the ACAS protocol described in [42], still rely also on non-authenticated measurement to compute the PVT: in this chapter we introduce a secure timing protocol that relies solely on SCE authentication features and on authenticated messages: in particular we will focus on ACAS and OSNMA. We build upon ACAS authenticated features a clock model that is both robust, i.e., able to compute reliable time corrections, and secure, since it can detect signal tampering. Our approach comprised two consecutive steps: first, the receiver processes the E6C measurements to estimate the receiver clock bias and drift; secondly, it combines the obtained measurements to estimate the current clock bias by either using a Kalman filter, or fitting a linear or quadratic LS model. Moreover, we propose a security enhancement strategy for timing attack detection, by checking the consistency of each new measurement with the one predicted by the model. We look at two approaches for this task: clock monitoring and innovation testing. We model a time-push attack to validate the performance of the proposed security checks. Moreover, we evaluate the proposed protocol on both simulated and experimental data collected with a professional GNSS receiver in nominal conditions and under-attack scenarios. Results of this research activity have been described in [15].

The rest of the chapter is organized as follows. The scenario for our analysis is described in Section 4.1. The proposed protocol is provided in Section 4.2, while the attack and its detection are described is Section 4.3. Simulation and experimental results are discussed in Section 4.4. Lastly, we draw the conclusion in Section 4.5.

4.1 System Model

We consider a scenario where a master clock is responsible for the synchronization of a network, composed of several devices or sensors connected via LAN. We assumed that this network was isolated; therefore, no attacker could influence the time dissemination process. The master clock is connected to a GNSS receiver, and we assume that the received signals are transmitted by satellites mostly in LOS, and that the effects of the multipath are minimal: this may be achieved, for instance, by placing the antenna on the roof of a building with clear view of the sky.

The antenna position is fixed and known. Without loss of generality, we focused on the case of a single-antenna receiver: multiple antennas may still be employed to either enhance the performance or security of the scheme by, e.g., checking the angle of arrival of a GNSS signal [52, 53]. A representation of the considered scenario is depicted in Figure 4.1.

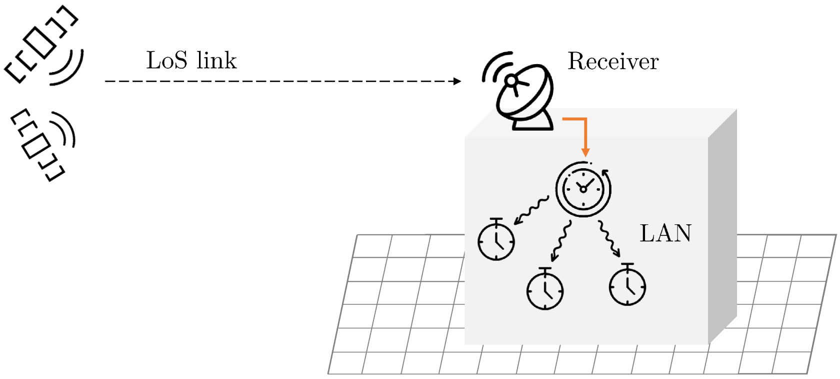


Figure 4.1: Pictorial representation of the considered scenario.

We considered a multi-frequency receiver enabled to acquire and track Galileo signals (at least) in bands E1 and E6. Moreover, the receiver exploits both Galileo OSNMA and ACAS: as described in Chapter 2, once the RECS files are published in the server and the TESLA key is received, the receiver decrypts the RECSs by using the corresponding key to obtain a local replica of the ECSs. Next, for the subset of Galileo satellites in view $\mathcal{S} \subseteq \{1, 2, \ldots, N_{\mathrm{GAL}}\}$, it correlates the local replica with the prerecorded Galileo E6C signal samples and, from the correlation peaks, it computes code delay $u_i^{(s_i)}$ and Doppler frequency $f_{\mathrm{D},i}^{(s_i)}$, measured by the receiver on signal on band E6, transmitted by satellite $s_i \in \mathcal{S}$ and received at time t_i.

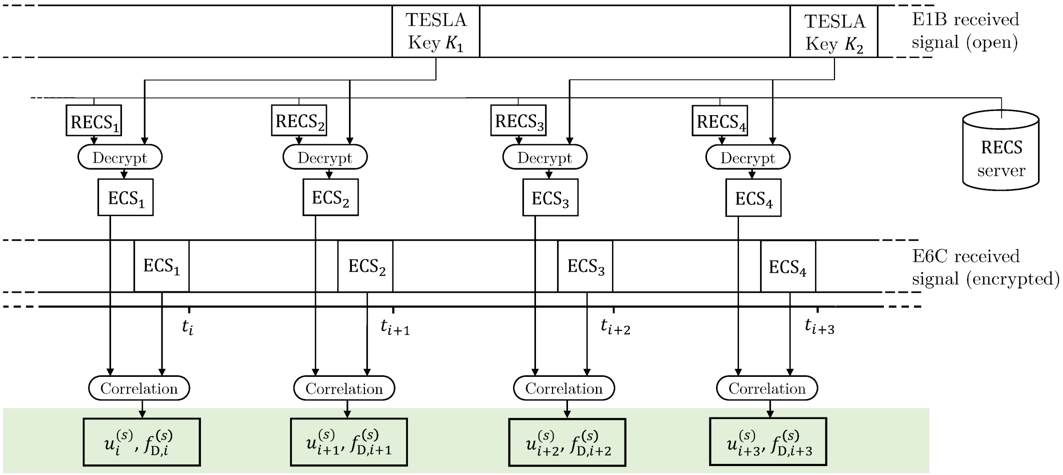

TESLA Key K_1
TESLA Key K_2
E1B received signal (open)
RECS$_1$
RECS$_2$
RECS$_3$
RECS$_4$
RECS server
Decrypt
ECS$_1$
ECS$_2$
ECS$_3$
ECS$_4$
E6C received signal (encrypted)
t_i
t_{i+1}
t_{i+2}
t_{i+3}
Correlation
$u_i^{(s)}, f_{D,i}^{(s)}$
$u_{i+1}^{(s)}, f_{D,i+1}^{(s)}$
$u_{i+2}^{(s)}, f_{D,i+2}^{(s)}$
$u_{i+3}^{(s)}, f_{D,i+3}^{(s)}$

FIGURE 4.2: Summary of ACAS operations at the receiver side for signal transmitted by satellite s.

After collecting M observables, the aim is to estimate the current master clock bias by using, at time t_0, the M measurements collected from satellites in $\mathcal{S}$ respectively at times $t_1, \ldots, t_M$, with $t_{i+1} \geq t_i$, with $t_0 > t_M$. We remark that, unlike the PVT computation, the proposed protocol does not require four satellites in view.

4.2 Proposed Approach

In this section we describe the proposed protocol: we remark that the protocol relies solely on the observables authenticated by ACAS and the message, authenticated instead by OSNMA. The proposed strategy is composed of three steps: *preprocessing phase*, *current-state estimation*, and *security checks*. We now briefly introduce each phase; in the next section we will discuss in detail each block.

We start however formally introducing the concept of measurement in this context: an ACAS observation is a 4-ple $\boldsymbol{O}_i = \{t_i, s_i, u_i, f_{\mathrm{D},i}\}$, where t_i is the observation time, $s_i \in \mathcal{S}$ is the satellite ID, u_i is the observed code delay, and $f_{\mathrm{D},i}$ is the observed frequency offset (Doppler shift). We define the set of observables $\mathcal{O}$ as

$$\mathcal{O} = \{\boldsymbol{O}_i : \ i = 1, \ldots, M\} = \{(t_i, s_i, u_i, f_{\mathrm{D},i}) : \ i = 1, \ldots, M\}, \tag{4.1}$$

with $|\mathcal{O}| = M$, where all the measurements are obtained from the E6 signals.

First, the preprocessing phase, from observation in $\mathcal{O}$, we derived $\hat{T}_{\mathrm{b},i}$, estimated the clock bias at time t_i on the basis of observation $\boldsymbol{O}_i$, and $\hat{T}_{\mathrm{d},i}$, and estimated the clock drift at time t_i on the basis of observation $\boldsymbol{O}_i$. So, the output of the preprocessing phase is the set

$$\mathcal{T} = \{(\hat{T}_{\mathrm{b},i}, \hat{T}_{\mathrm{d},i}) : \ i = 1, \ldots, M\}, \tag{4.2}$$

which had the same cardinality as $\mathcal{O}$. Each measurement in $\mathcal{T}$, indexed by $i = 1, \ldots, M$, may be acquired by a different satellite.

Next, the current-state estimation phase follows, where measurements in $\mathcal{T}$ are used to compute the master clock correction, at time t_0, $\hat{T}_{\mathrm{b},0}$. Figure 4.3 summarizes the two phases.

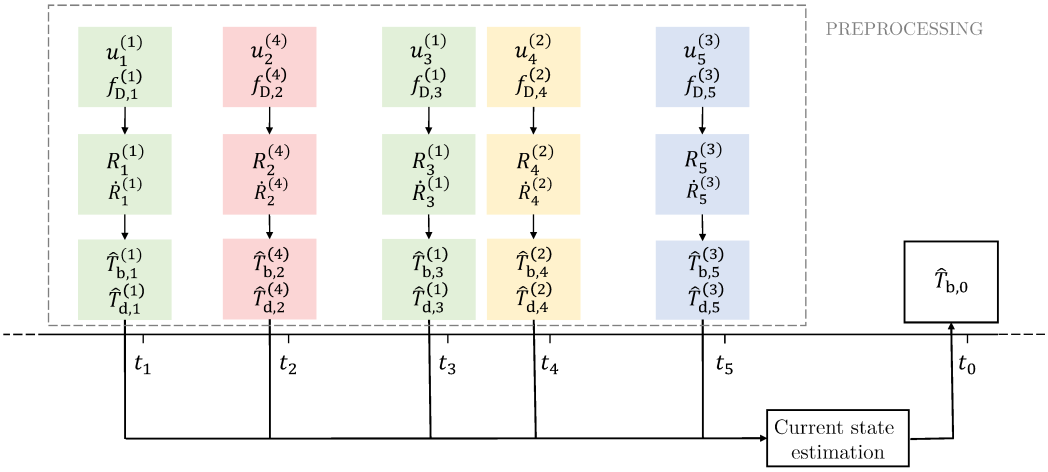

PREPROCESSING
$u_1^{(1)}$ $f_{D,1}^{(1)}$
$u_2^{(4)}$ $f_{D,2}^{(4)}$
$u_3^{(1)}$ $f_{D,3}^{(1)}$
$u_4^{(2)}$ $f_{D,4}^{(2)}$
$u_5^{(3)}$ $f_{D,5}^{(3)}$
$R_1^{(1)}$ $\dot{R}_1^{(1)}$
$R_2^{(4)}$ $\dot{R}_2^{(4)}$
$R_3^{(1)}$ $\dot{R}_3^{(1)}$
$R_4^{(2)}$ $\dot{R}_4^{(2)}$
$R_5^{(3)}$ $\dot{R}_5^{(3)}$
$\hat{T}_{b,1}^{(1)}$ $\hat{T}_{d,1}^{(1)}$
$\hat{T}_{b,2}^{(4)}$ $\hat{T}_{d,2}^{(4)}$
$\hat{T}_{b,3}^{(1)}$ $\hat{T}_{d,3}^{(1)}$
$\hat{T}_{b,4}^{(2)}$ $\hat{T}_{d,4}^{(2)}$
$\hat{T}_{b,5}^{(3)}$ $\hat{T}_{d,5}^{(3)}$
$\hat{T}_{b,0}$
t_1 t_2 t_3 t_4 t_5 t_0
Current state estimation

FIGURE 4.3: Schematic representation of preprocessing and current-state estimation phases.

The last phase concerns *security checks*, where we detect anomalous estimates $\hat{T}_{\mathrm{b},0}$ of the clock bias: for this task we considered *clock monitoring* and *innovation test* as candidate solutions for this task.

4.2.1 Preprocessing

From each code delay measurement $u_i \in \mathcal{O}$, we compute pseudorange R_i at time t_i associated with satellite s_i, as described in [54].

Considering the psedorange decomposition (see Appendix B), and neglecting the dependence on the carrier frequency, we notice that:

- the geometric range r_i can be computed a priori since both receiver and satellites' positions are known;
- the clock bias $T_{\mathrm{b,sat}}^{(s_i)}(t)$ can be retrieved from the authenticated OSNMA message;
- the ionospheric delay (estimation) of band E6 can be computed from the one of band E1, retrieved from the navigation message.

We remark that the troposphere is a non dispersive medium, thus the corrections for the troposheric delay of band E1, $\hat{D}_{\mathrm{tropo,E1}}^{(s)}(t)$, and E6, $\hat{D}_{\mathrm{tropo,E6}}^{(s)}(t)$ were identical for all $s \in \mathcal{S}$.

Calling $\hat{D}_i^{(s_i)}$ the sum of troposheric and ionospheric delay correction in band E6 for satellite s_i, the receiver clock bias estimation at time t_i is

$$\hat{T}_{\mathrm{b},i} \triangleq \frac{1}{c}\Big(R_i - r_i - \hat{D}(s_i)_i\Big) + T_{\mathrm{b,sat}}^{(s_i)}(t_i) = T_{\mathrm{b},i} + \xi_{\mathrm{b},i}\ , \tag{4.3}$$

where $T_{\mathrm{b},i}$ is the real receiver clock bias at time t_i, and $\xi_{\mathrm{b},i}$ is the clock bias estimation error taking into account the error residuals due to the nonperfect atmospheric delays estimation and the additional noise component η_i.

Next, we compute the pseudorange rate $\dot{R}_i$ at time t_i as

$$\dot{R}_i = -\lambda f_{\mathrm{D},i}, \tag{4.4}$$

where $f_{\mathrm{D},i}$ belongs to the authenticated observables set $\mathcal{O}$ and λ is the wavelength of E6. As done in (3.27), the pseudorange rate can then be decomposed as

$$\dot{R}_i = \dot{r}_i + c\left(T_{\mathrm{d},i} - T^{(s_i)}_{\mathrm{d,sat}}(t_i)\right) + \gamma_i + \dot{\eta}_i \ , \tag{4.5}$$

where, here we also consider the term

$$\gamma_i = \gamma^{(s_i)}(t_i)\,, \qquad \gamma^{(s)}(t) \triangleq \frac{\partial}{\partial t}\left[D^{(s)}_{\mathrm{iono}}(t) + D^{(s)}_{\mathrm{tropo}}(t)\right] \tag{4.6}$$

which is a term modeling both the time derivatives of the atmospheric delays. Moreover, as computed in (3.28), the geometric range derivative $\dot{r}^{(s)}(t)$ is given by

$$\dot{r}^{(s)}(t) = (\boldsymbol{v}^{(s)}_{\mathrm{sat}}(t) - \boldsymbol{v}_{\mathrm{rx}}(t))^T \boldsymbol{e}^{(s)}(t) = v^{(s)}_{\mathrm{LOS}}(t) \ . \tag{4.7}$$

Moreover, in our scenario the position of the GNSS receiver static therefore $\boldsymbol{v}_{\mathrm{rx}}(t) = 0 \ \forall t$, thus, term $\dot{r}_i$ appearing in (4.5) is obtained as

$$\dot{r}_i = \dot{r}^{(s_i)}(t_i) = v^{(s_i)}_{\mathrm{LOS}}(t_i) \ . \tag{4.8}$$

Analogously to (4.3), we compute

$$\hat{T}_{\mathrm{d},i} \triangleq \frac{1}{c}\left(\dot{R}_i - v^{(s_i)}_{\mathrm{LOS}}(t_i)\right) + T^{(s_i)}_{\mathrm{d,sat}}(t_i) = T_{\mathrm{d},i} + \xi_{\mathrm{d},i}, \tag{4.9}$$

where $T_{\mathrm{d},i}$ is the real receiver clock drift at time t_i and $\xi_{\mathrm{d},i}$ is the clock drift estimation error. Repeating this procedure for $i = 1, \ldots, M$, we obtain the set $\mathcal{T}$.

It is possible to statistically model both $\xi_{\mathrm{b},i}$ and $\xi_{\mathrm{d},i}$. A partial model for the first term is provided in [20, 42, 55]; however, the second-order descriptions of $\xi_{\mathrm{b},i}$ and $\xi_{\mathrm{d},i}$ are sufficient for the analysis done in this work.

4.2.2 Current-State Estimation

In the previous section, we showed how to derive measurements in $\mathcal{T}$ starting from the authenticated observables in $\mathcal{O}$. These estimates are exploited to compute the actual receiver clock bias that is used to synchronize the master clock to the reference time frame. The design of a specific algorithm for this task is justified, since the clock bias and drift estimations are relative to time $t_i, i = 1, \ldots, M$; therefore, we need a model that exploits the past measurements to compute the current one. Moreover, past measurements are affected by noise, modeled by $\xi_{\mathrm{b},i}$ and $\xi_{\mathrm{d},i}$. We analyzed three different approaches to this task: a LS quadratic model, a LS linear model, and a Kalman filter.

4.2.2.1 LS-Quadratic and Linear Model

The first two solutions leverage the idea that clock bias increases (or decreases) over time following a parabola, where the quadratic term, with coefficient *drift rate*, is expected to have a low impact. For instance, considering the *time of ephemeris* t_{oe}, the Galileo satellite clock bias is computed as follows [39]

$$T^{(s)}_{\mathrm{b,sat}}(t) = a^{(s)}_0 + a^{(s)}_1 (t - t_{\mathrm{oe}}) + a^{(s)}_2 (t - t_{\mathrm{oe}})^2, \tag{4.10}$$

where $a_0^{(s)}$, $a_1^{(s)}$, and $a_2^{(s)}$ represent the satellite clock bias, clock drift, and clock drift rate measured at time t_{oe}, respectively. Typically, the drift rate is transmitted to as $a_2^{(s)} = 0$, leading to a de facto linear model. Thus, we consider both a quadratic and a linear model.

Analogously to (4.10), calling $\tau_i = t_0 - t_i$ the time difference between the current time at which we want to compute the clock bias estimation and the time associated to the measurements, we write

$$\hat{T}_{\mathrm{b},i} = a_0 + a_1\tau_i + a_2\tau_i^2 + \varepsilon_{\mathrm{b},i}, \tag{4.11}$$

$$\hat{T}_{\mathrm{d},i} = a_1 + 2a_2\tau_i + \varepsilon_{\mathrm{d},i}, \tag{4.12}$$

where a_0, a_1 and a_2 are now the parameters modeling the receiver clock behavior, $\hat{T}_{\mathrm{b},i}$ and $\hat{T}_{\mathrm{d},i}$ are the measurements in $\mathcal{T}$ computed in the preprocessing phase, $\varepsilon_{\mathrm{b},i}$ and $\varepsilon_{\mathrm{d},i}$ are the estimation errors related to the i-th measurement. Equivalently to (4.11) and (4.12), in matrix form, we have

$$\begin{pmatrix} \hat{T}_{\mathrm{b},i} \\ \hat{T}_{\mathrm{d},i} \end{pmatrix} = \begin{pmatrix} 1 & \tau_i & \tau_i^2 \\ 0 & 1 & 2\tau_i \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_{\mathrm{b},i} \\ \varepsilon_{\mathrm{d},i} \end{pmatrix} = \begin{pmatrix} \boldsymbol{E}_{\mathrm{b},i} \\ \boldsymbol{E}_{\mathrm{d},i} \end{pmatrix} \boldsymbol{a} + \boldsymbol{\varepsilon_i} \ , \tag{4.13}$$

where $\boldsymbol{a} = [a_0 \ a_1 \ a_2]^T$ is the vector of parameters we aim to estimate. Next, considering all the measurements in $\mathcal{T}$, we stack the matrices, obtaining

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_{\mathrm{b}} \\ \boldsymbol{y}_{\mathrm{d}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{E}_{\mathrm{b}} \\ \boldsymbol{E}_{\mathrm{d}} \end{pmatrix} \boldsymbol{a} + \begin{pmatrix} \boldsymbol{\varepsilon}_{\mathrm{b}} \\ \boldsymbol{\varepsilon}_{\mathrm{d}} \end{pmatrix} = \boldsymbol{E}\boldsymbol{a} + \boldsymbol{\varepsilon} \ , \tag{4.14}$$

where $\boldsymbol{y}_{\mathrm{b}}$ and $\boldsymbol{y}_{\mathrm{d}}$ are the columns vectors collecting the M bias and drift measurements, respectively, in $\mathcal{T}$, $\boldsymbol{E}_{\mathrm{b}} = [\boldsymbol{E}_{\mathrm{b},1}^T, \dots, \boldsymbol{E}_{\mathrm{b},M}^T]^T$ and $\boldsymbol{E}_{\mathrm{d}} = [\boldsymbol{E}_{\mathrm{d},1}^T, \dots, \boldsymbol{E}_{\mathrm{d},M}^T]^T$ contain the time difference terms associated to each measurement in $\boldsymbol{y}_{\mathrm{b}}$ and $\boldsymbol{y}_{\mathrm{d}}$, respectively, and $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_M]^T$. To minimize the mean squared error (MSE), we performed the estimation by using the pseudoinverse

$$\hat{\boldsymbol{a}} = (\boldsymbol{E}^T\boldsymbol{E})^{-1}\boldsymbol{E}^T\boldsymbol{y} \ , \tag{4.15}$$

and we obtained the estimations of clock bias and drift at time t_0 as

$$\hat{T}_{\mathrm{b},0} = \hat{a}_0, \tag{4.16}$$

$$\hat{T}_{\mathrm{d},0} = \hat{a}_1 \ . \tag{4.17}$$

An analogous derivation can be performed starting from a linear model, replacing (4.13) with

$$\begin{pmatrix} \hat{T}_{\mathrm{b},i} \\ \hat{T}_{\mathrm{d},i} \end{pmatrix} = \begin{pmatrix} 1 & \tau_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_{\mathrm{b},i} \\ \varepsilon_{\mathrm{d},i} \end{pmatrix} . \tag{4.18}$$

4.2.2.2 Kalman Filter

In this section, we investigate the use of a Kalman filter to estimate the bias. In particular, every time a new estimate $\{\hat{T}_{\mathrm{b},i}, \hat{T}_{\mathrm{d},i}\}$ is available, we update the model and perform a new prediction; moreover, even if no new measurement is available, it is still possible to exploit the previously trained model to estimate the current clock correction. While we give a brief introduction to the Kalman filter in Appendix C, a more detailed description can be found in [56].

We consider respectively as true state and measurement input at time t_i the vectors $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$, composed as

$$\boldsymbol{x}_i = \begin{pmatrix} T_{\mathrm{b},i} \\ T_{\mathrm{d},i} \\ \dot{T}_{\mathrm{d},i} \end{pmatrix}, \quad \boldsymbol{z_i} = \begin{pmatrix} \hat{T}_{\mathrm{b},i} \\ \hat{T}_{\mathrm{d},i} \end{pmatrix}, \tag{4.19}$$

where $\dot{T}_{\mathrm{d},i}$ represents the *clock drift rate.* Then, we chose as state-transition matrix and the observation matrix

$$\boldsymbol{F}_i = \begin{pmatrix} 1 & t_i - t_{i-1} & (t_i - t_{i-1})^2 \\ 0 & 1 & 2(t_i - t_{i-1}) \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{H}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{4.20}$$

By exploiting the Kalman filter for every measure in $\mathcal{T}$, we obtain the M-th estimation $\boldsymbol{x}_M$. Then, using the Kalman state-transition equation (see (C.1)), we compute the a posteriori estimation at time t_0 as

$$\hat{\boldsymbol{x}}_{0|M} = \boldsymbol{F}_0 \hat{\boldsymbol{x}}_{M|M} , \tag{4.21}$$

where

$$\boldsymbol{F}_0 = \begin{pmatrix} 1 & t_0 - t_M & (t_0 - t_M)^2 \\ 0 & 1 & 2(t_0 - t_M) \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.22}$$

Lastly, $\hat{T}_{\mathrm{b}}$ is obtained as the first element of vector $\hat{\boldsymbol{x}}_{0|M}$.

4.3 Timing Attack and Detection

As described in Section 4.1, the position of the GNSS receiver is fixed and publicly known. This fact alone allows the legitimate receiver to perform two (näive) checks. The first involves a consistency check on the received signal, such that, if the receiver PVT computation yields a result much different from the expected one, e.g., a position much far from the expected one or a significant velocity, an alarm is raised. On the other hand, the satellites' position is known thus the receiver can reject any signal coming from satellites that should not be in view. Hence, the attacker knows that (1) all the attacks causing a relevant change in the victim's computed position or velocity are detected, and (2) signals transmitted by satellites that should not be in view by a legitimate receiver are neglected.

For these reasons, we consider an attacker performing a *time-push* attack: this is a *meaconing* attack where the receiver records the signals and re-transmits them with additional delays, adding an equal bias in all pseudoranges, which results in an error in the time calculation of the PVT solution. Notice that the computed position instead does not change, as will be also proven in Section 4.4. Moreover, this attack may indeed target ACAS, where the signal cannot be tracked since the receiver operates in snapshot mode: this grants the attacker a time window to record the signal and perform a time-push attack. However, sudden changes in the estimated clock bias may alert the receiver: thus, the attacker performs a time push in a smoothly progressive manner, gradually increasing the delay. Still, to be effective, the attacker must be (relatively) close to the victim's antenna to have the same satellites in view with the same delays. A possible countermeasure would be to render the area around the receiver inaccessible by, for instance, installing surveillance cameras and/or surrounding the building with a fence. Still, we considered a worst-case scenario where the attacker managed to approach close enough to the receiver antenna and to isolate the legitimate receiver, therefore ensuring that only fake signals are received to perform the time-push attack.

To detect the presence of false measurements among the obtained corrections, we considered *clock-monitoring* and *innovation-testing* methods. Formally, we frame this problem as hypothesis testing: considering null-hypothesis $\mathcal{H}_0$ as the nominal condition where the signals are transmitted by the legitimate transmitter, the receiver observes a test statistic, β, and decides whether β is compatible with $\mathcal{H}_0$ or not.

4.3.1 Clock Monitoring

As discussed in Section 4.2.2, the receiver clock bias is typically assumed to have either linear or quadratic behavior over time: we can then analyze the clock bias corrections over time and if anomalous discontinuities are detected we raise an alarm. This is the idea behind clock-monitoring techniques. Given the clock model $\hat{\boldsymbol{a}}'$ estimated through either (4.13) or (4.18) at time $t_i - \delta$, i.e., the previous epoch, it is possible to compute a prediction $\{\widetilde{T}_{\mathrm{b},i}, \widetilde{T}_{\mathrm{d},i}\}$ of the measurements at time t_i, as

$$\begin{pmatrix} \widetilde{T}_{\mathrm{b},i} \\ \widetilde{T}_{\mathrm{d},i} \end{pmatrix} = \begin{pmatrix} 1 & \delta & \delta^2 \\ 0 & 1 & 2\delta \end{pmatrix} \hat{\boldsymbol{a}}'. \tag{4.23}$$

Hence, for bias and drift, we adopted as the test statistic the quantities

$$\beta_{\mathrm{b},i} \triangleq \widetilde{T}_{\mathrm{b},i} - \hat{T}_{\mathrm{b},i}, \tag{4.24}$$

$$\beta_{\mathrm{d},i} \triangleq \widetilde{T}_{\mathrm{d},i} - \hat{T}_{\mathrm{d},i}, \tag{4.25}$$

and test

$$\hat{\mathcal{H}}_i = \begin{cases} \mathcal{H}_0 & \text{if} \quad |\beta_{\mathrm{b},i}| < \lambda_{\mathrm{b}} \text{ and } |\beta_{\mathrm{d},i}| < \lambda_{\mathrm{d}}, \\ \mathcal{H}_1 & \text{otherwise}, \end{cases} . \tag{4.26}$$

where thresholds λ_{b} and λ_{d} are chosen a priori by the user as a predefined FA probability. When a specific attack model is available, it may be possible to instead set the thresholds on the MD probability. More in detail, considering, for instance, drift threshold λ_{d}, it may be worth taking into account the actual clock specifications, thus evaluating a bound of the clock drift in nominal conditions [13].

If the distribution of the tests statistics $\beta_{\mathrm{b},i}$ and $\beta_{\mathrm{d},i}$ were known, it would be possible to replace (4.26) with two GLRTs; however, the statistical characterization of such quantities is out of the scope of this work and is left to future works. Lastly, while we show the effectiveness of the clock monitoring only in relation to the LS models, such techniques may also be employed with the Kalman filter.

4.3.2 Innovation Testing

This security check has been proposed in the literature in [57, 58]. While using the Kalman filter, during the update step, each prediction is corrected by the so-called innovation term $\boldsymbol{y}_i$, (for the computation see (C.3c)) that, in steady-state conditions, has mean and covariance

$$E[\boldsymbol{y}_i] = 0 \tag{4.27}$$

$$\mathrm{COV}(\boldsymbol{y}_i) = \boldsymbol{B}_i. \tag{4.28}$$

We can then use the normalized innovation as a test statistic, computed as

$$\beta_{\mathrm{K},i} = \boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{B}_i \boldsymbol{y}_i. \tag{4.29}$$

In nominal conditions, $\beta_{\mathrm{K},i}$ is assumed to be a chi-squared random variable [58], with as many degrees of freedom as the size of the measurement $\boldsymbol{z}_i$, i.e., $\beta_{\mathrm{K},i} \sim \chi^2$. Hence, to assess the authenticity of the measurement, we can use the GLRT test against a uniform distribution[1]

$$\hat{\mathcal{H}}_i = \begin{cases} \mathcal{H}_0 & \text{if} \quad p(\beta_{\mathrm{K},i}|\mathcal{H}_0) \geq \lambda_{\mathrm{K}} \\ \mathcal{H}_1 & \text{otherwise} \end{cases} , \tag{4.30}$$

where λ_{K} is chosen by the user to match a predefined FA probability.

4.4 Results and Discussion

In this section, first, we validate the proposed approach; next, we show that the time-push attack described in Section 4.3 is successful even if a legitimate receiver knows its actual position, highlighting the need for additional security checks.

We collected experimental data to build the set of authenticated observables $\mathcal{O}$ serving as input for the preprocessing phase. The detection capabilities of the methods proposed in Sections 4.3.1 and 4.3.2 were tested against a simulated time-push attack.

4.4.1 Validation Using Experimental Data

To validate the proposed approach described in Section 4.2, we performed experimental tests collecting signals from an open-sky environment with a Septentrio PolarRx5 receiver connected to a A42 Hemisphere antenna. The Septentrio PolaRx5 is equipped with a voltage-controlled and temperature-controlled crystal oscillator (VCTCXO). The experimental setup is depicted in Figure 4.4.

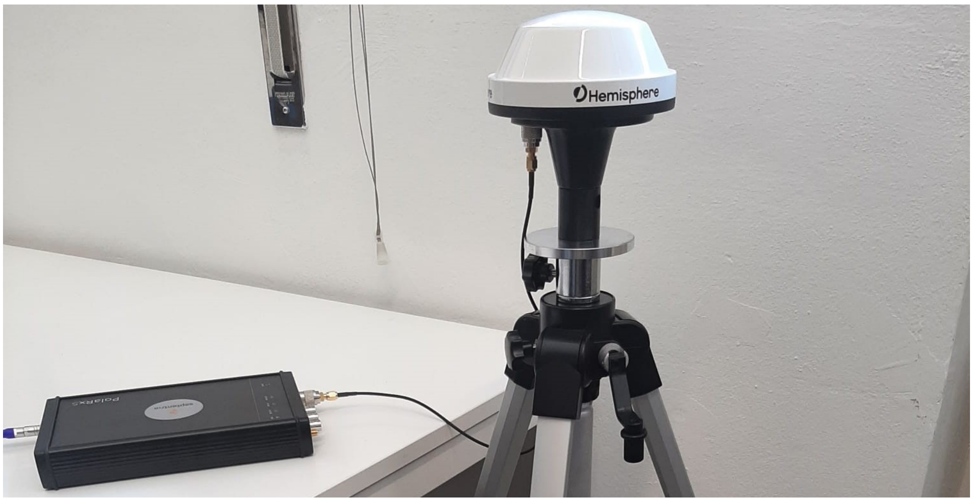


FIGURE 4.4: Setup used for the experimental dataset collection: Septentrio PolarRx5 receiver connected to an A42 Hemisphere antenna.

[1] Since we have no information about the attacker statistics we assume a uniform distribution

The output of the receiver was logged using the Septentrio binary format (SBF) standard and postprocessed after the experiments, obtaining a dataset of measurements from different constellations and frequency bands, summarized in Table 4.1.

TABLE 4.1: Constellations and central frequencies of the measurements collected in the experimental dataset.

	Central Frequency, f_c [MHz]								
	1176.45	**1207.14**	**1227.60**	**1245.5**	**1278.75**	**1268.52**	**1561.098**	**1575.42**	**1601.5**
Galileo	E5a	E5b			E6			E1BC	
GPS	L5		L2 C/A L2 P(Y)					L1 C/A L1 P(Y)	
Beidou	B2a	B2l				B3l	B1l	B1C	
GLONASS				L2 C/A					L1 C/A

We remark that only two Galileo satellite were visible during the whole experiment: thus only measurements collected from this two satellites where to compute the corrections.

We used as ground truth to later evaluate the goodness of our estimates $\hat{T}_\text{b}$, the clock bias measurements calculated from the PVT solution computed by the receiver using the whole set of measurements available in the dataset: on average, the PVT was computed by the receiver using the signal coming from 16 satellites.

Since only E6C ranging measurements were assumed to be authenticated, we set the receiver to use the Klobuchar ionospheric correction model, which is the one typically used for GNSS receivers, estimating the ionospheric delay as from the correction of E1 (see (B.2)). We remark that, in general, more sophisticated models, such as Galileo NeQuick [4] and IRI-P 2017 [59] can be employed. Still, for the sake of simplicity, we show that even the simpler Klobuchar model is enough to obtain satisfactory results, proving the robustness of our method. Next, we extracted set $\mathcal{O}$ from our dataset considering only the measurements from E6C.

Figure 4.5 shows the master clock bias estimation error as the difference between the ground truth and the clock estimations, $\Delta\hat{T}_\text{b}$, obtained using the LS quadratic, LS linear estimation methods and the Kalman filter. The LS methods described in Section 4.2.2.1 were used to compute one clock bias estimation $\hat{T}_\text{b}$ every 2 s by using the 4 most recent available measurements, so that $M = 4$. The Kalman filter computed one new estimate $\hat{T}_\text{b}$ every second. All the tested methods proven to be effective, achieving an error limited to less than 50 ns, obtaining precise timing with fewer than four satellites in view.

4.4.2 Numerical Results and Attack Detection

To simulate the attacks, we used our signal generator and software receiver developed for the MORE GOSSIP project, funded by the ESA, which was used also in [45]. We simulated the Galileo E6 baseband signal: of course, notice that the carrier frequency still influences the Doppler frequency. Data (E6B) and pilot (E6C) components where generated as in Galileo specifications [39], modulated with a BPSK(5), i.e., with code frequency $f_\text{code} = 5.115\,\text{MHz}$. We considered an additional linear (deterministic) clock drift of 0.5 parts per million (ppm).

We modeled a noiseless scenario with RECS duration equal to the PRN code length on E6, i.e., 5115 chips. Concerning CAS, we assumed that one new RECS would be disclosed every second. We generated 5 channels, i.e., 5 signals from five different satellites with 16 bit quantization. The sampling frequency was set to $f_s = 2f_\text{code} = 10.23\,\text{MHz}$, and each simulation scenario lasted for 100 s. On the receiver

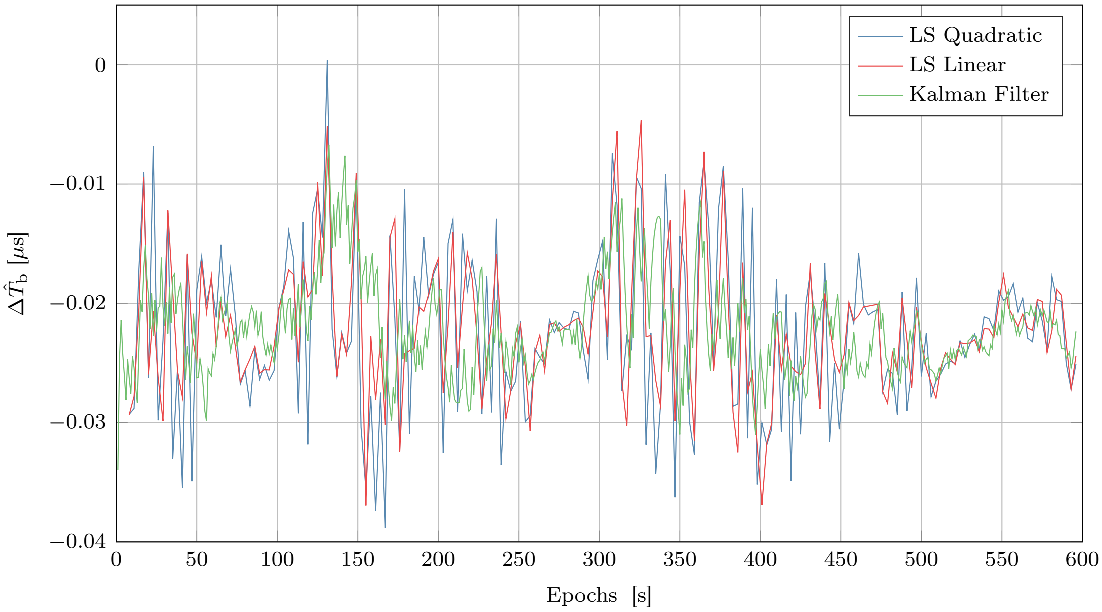


FIGURE 4.5: Difference between the ground truth and the clock estimations, $\Delta\hat{T}_\mathrm{b}$, obtained by using the LS quadratic, LS linear and the Kalman filter on the experimental data.

side, the acquisition was performed by using the same sampling frequency, and the Doppler bin size was set to 75 Hz. The receiver collected measurements $\{\hat{T}_{\mathrm{b},i}, \hat{T}_{\mathrm{d},i}\}$ with a frequency of 1 Hz; as indicated before, since we assumed that the one RECS was made public every 60 s, we used only one of the measurements of the satellite in view per acquisition round as input for the model.

4.4.2.1 Nominal Scenario

We start by considering legitimate dataset $\mathcal{H}_0$. Only one RECS is disclosed at every epoch; thus, only one signal every epoch can be used to update the state.

Figure 4.6 shows the results obtained for the current-state estimation phase described in Section 4.2.2. We show $\Delta\hat{T}_\mathrm{b}$, i.e., the difference between ground truth and clock estimations obtained by using the LS quadratic, LS linear, and the Kalman filter: all the methods were effective, achieving maximal deviation lower than 200 ns and a zero mean even using only one (new) measurement per epoch (i.e., per minute). Thus, all the methods could be employed for this task.

4.4.2.2 Attack Scenario

In this section, we evaluate under-attack scenarios, such as the ones described in Section 4.3: first, we show the impact of a time-push attack, proving that such attacks cannot be detected just by the check on the receiver position; in the second, we discuss the performance of the clock-monitoring and innovation-check methods, showing the different behaviors of the test statistics β_b, β_d, and β_K in the legitimate and under-attack scenarios, i.e., $\mathcal{H}_0$ and $\mathcal{H}_1$.

As indicated in Section 4.3, a sudden spike in the estimated clock bias may alert the receiver; thus, the attacker introduces the delays in a ramp-like fashion. We modeled a scenario where the attacker managed to isolate the victim receiver and acquired only the forged E6 signals.

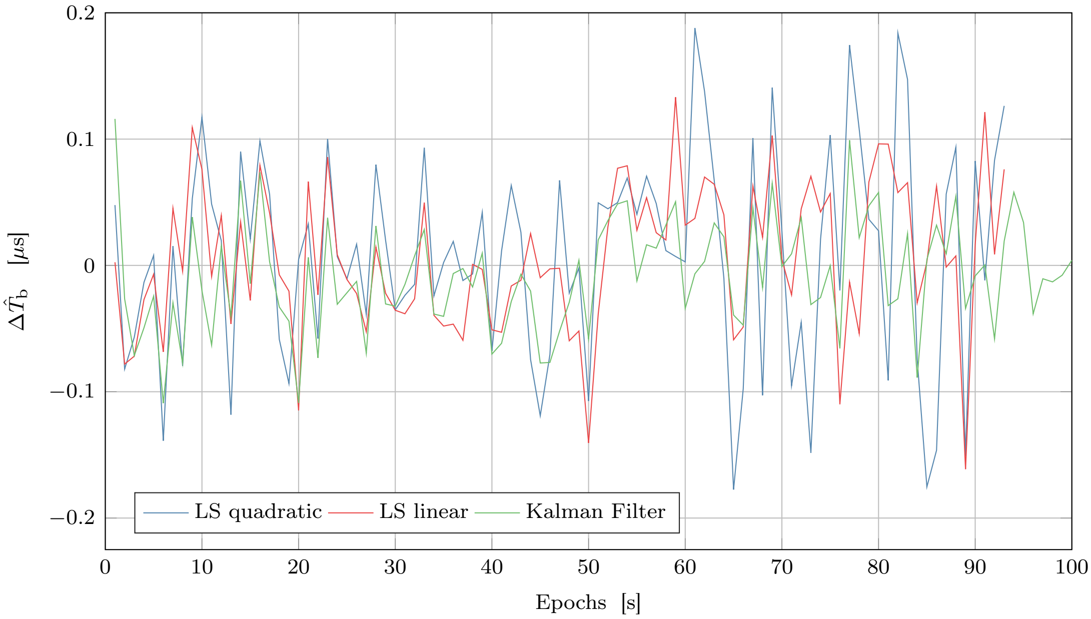


FIGURE 4.6: Difference between the ground truth and the clock estimations, $\Delta\hat{T}_{\mathrm{b}}$, obtained by using the LS quadratic, LS linear and the Kalman filter on the simulated data.

Figure 4.7 reports the results: while the positioning error statistic was indeed indistinguishable in $\mathcal{H}_0$ and $\mathcal{H}_1$, the impact on the clock bias is clear. This confirms that we cannot trust the timing obtained on a PVT that passes by the naive position check. Hence, we suggest dedicated algorithm and strategies specifically designed for secure timing.

Next, we validate the security checks described in Section 4.3 considering a legitimate scenario and three attack scenarios. Each attack lasted 20 s with a constant drift of $1, 2$ and 3 ppm, and achieved a final delay of 20, 40, and 60 μs, respectively. Each attack started at a different time.

Figure 4.8 shows the test statistic obtained via clock monitoring in nominal conditions and an under-attack scenario: both β_{b} and β_{d} presented spikes associated to the start and end of the attack, which had a magnitude much greater than the standard deviation of the same test statistic in the nominal conditions. This test was, thus, indeed effective in detecting time-push attacks since it is easy for the user to set a threshold to distinguish legitimate from under-attack scenarios. Moreover, performing more tests, it could be possible for the user to fine-tune the threshold by observing the ROC curves.

Figure 4.9 shows the test statistic β_K used for the innovation testing and described in Section 4.3.2. After a first transitory phase, a jump is presented when the attacker starts (and ends) the time-push attack. Therefore, this technique is successful at detecting time-push attacks.

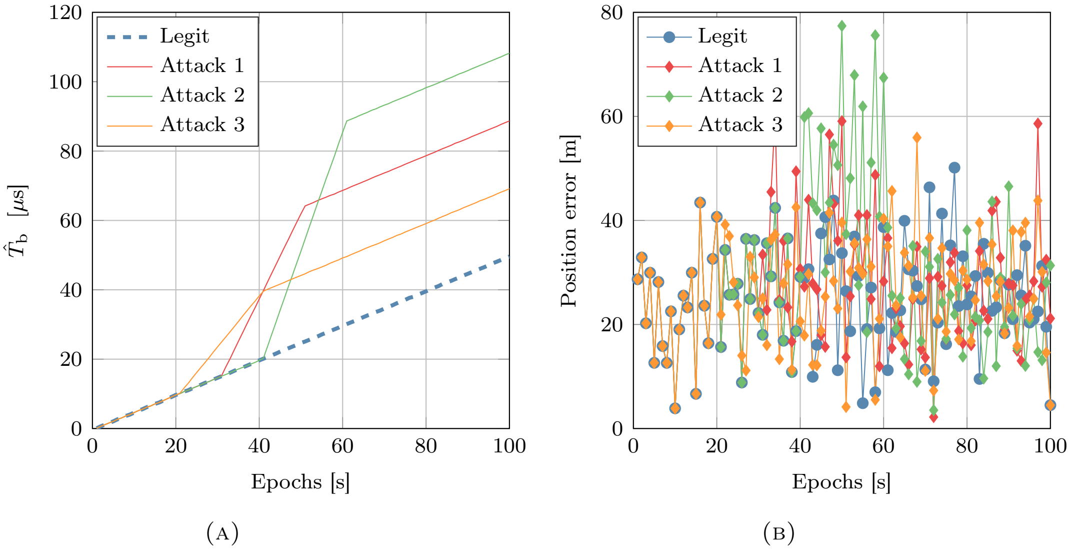
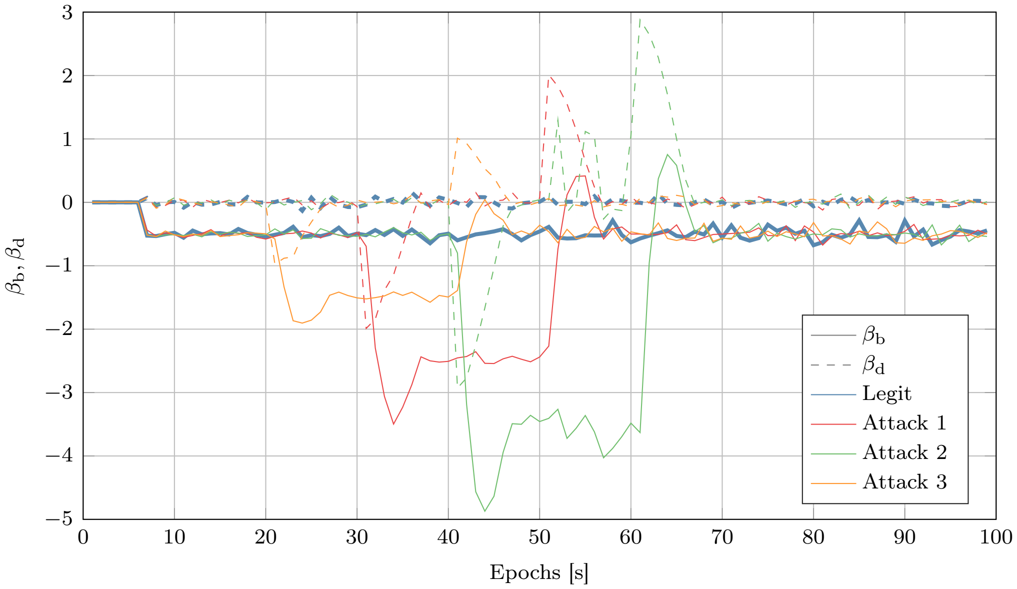


FIGURE 4.7: Comparison of legitimate and under-attack scenarios for (A) clock bias and (B) positioning error obtained using the simulated dataset.

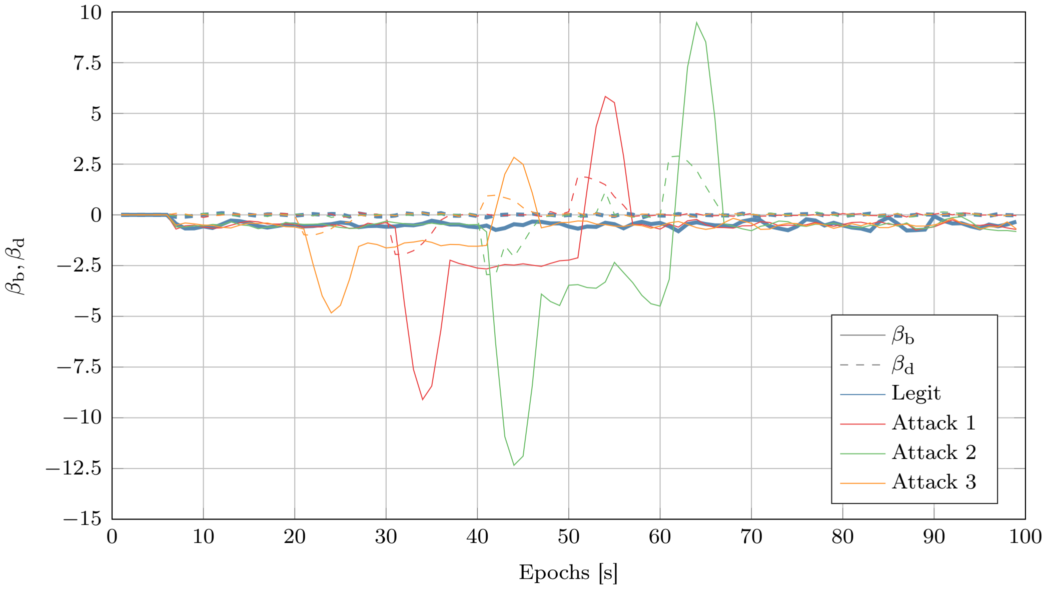
4.5 Conclusions

In this work, we have presented a secure timing protocol that may be used, for instance, by Industry 4.0 applications to synchronize multiple IoT devices within a facility. We have considered a scenario where the master clock was securely connected to a GNSS receiver, and all the devices or sensors aimed to be synchronized. The protocol is based upon the new Galileo ACAS and relies only on authenticated measurements to obtain the clock correction.

The procedure is composed by three blocks: first, exploiting the fact that the facility position is known, the receiver processes the E6C measurements to obtain an estimation of the receiver clock bias and drift; second, the receiver merges the previously obtained measurements to compute the current clock bias estimation by fitting either a linear or a quadratic least-squares model, or by using a Kalman filter. Lastly, we also envision the employment of a security evaluation phase where, the consistency of each new measurement is tested against the one obtained by using the previously estimated prediction model. For this task, we have proposed two methods: clock monitoring and innovation test. We have validated the proposed procedure by using an experimental dataset, collected with a Septentrio PolaRx5 receiver, and simulated data considering both legitimate and under-attack conditions. The obtained numerical and experimental results have shown that our protocol was both able to compute a reliable timing correction and to reject time-push attacks.



(A)



(B)

FIGURE 4.8: Test statistics, β_b (continuous lines), and β_d (dashed lines) used by clock monitoring: comparison of legitimate (thick blue) and under-attack scenarios for the (A) linear and (B) quadratic LS models.

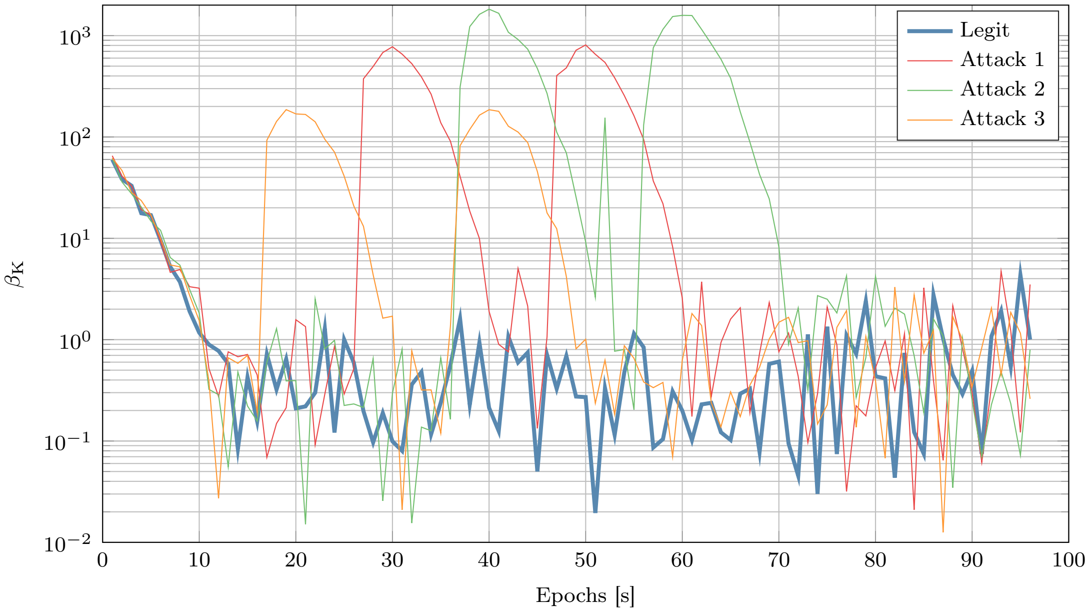


FIGURE 4.9: Test statistic β_K used by the innovation testing: comparison between legitimate (blue) and under-attack scenarios.

Chapter 5

Scheduling Strategies for GNSS Packet Broadcasting

5.1 Introduction

GNSSs are continuously evolving to provide improved and/or new services. Beyond the primary objective of supporting navigation and timing, e.g., by transmitting ephemeris and almanacs, studies have been conducted to support new services, such as authentication data, SAR, and short messaging [1, 60, 61]: these solutions require the broadcast of further data messages. Many of these data transmissions are particularly useful in remote areas, where no other communication network is available: in this case, hybrid satellite-Internet solutions, such as those provided by the high accuracy service (HAS), cannot be adopted. In this context, almanacs may be used to aid the acquisition phase: however, considering for instance Galileo, the typical solution is to resort to the *carousel scheduling strategy* [62], where each different packet is scheduled sequentially in time. With the carousel strategy, only the almanacs for two SVs are retrieved within each sub-frame (i.e., 30 s) [39].

In general, GNSS have data rates in the order of a few hundred bit/s, e.g., 50 bit/s for GPS L1 C/A [2, Chapter 4] and 125 bit/s for Galileo E1B [39]. Hence, latency may become significant, especially for long messages.

Still, at a packet level the diversity provided by multiple satellites can be exploited. Indeed, by splitting the message into several packets and transmitting them via different satellites and specific spreading code, ground receivers can collect the packets and obtain the entire message. A first study on various alternatives for message splitting in GNSS data broadcasting was presented in [1]: in particular, it has been highlighted that the *scheduling* of sub-message transmission among satellites plays an important role in the resulting rate at the GNSS receiver. However, no specific scheduling solutions were proposed there.

Although a message splitting approach was suggested in [63,64], for Galileo OSNMA, the subset of satellites distributing the OSNMA data changes over time: in such a way a potential attacker cannot know in advance which satellite is distributing the OSNMA data [65]. In this context, the use of random fountain codes has been proposed to increase the reliability of the packet dissemination [66].

Message allocation strategies are proposed for Galileo HAS [67] in [68, 69] taking into account the satellite positions but not the actual receivers' visibility. Moreover, in these works the authors focus on dissemination of 10 messages while we consider a more general scheduling problem.

In this scenario, a key issue is the scheduling of packets on the satellites which can be deterministic. In fact, the positions of the SVs at any instant are predictable positions and the set of satellites in view from any point on the surface of the earth can be easily determined. By assigning a specific packet to be broadcast by each satellite, we can determine where it will be reliably received. Yet, only a few studies are available on this topic, in the literature.

This chapter collects the results of two works. In the first [70] we address the problem of scheduling packet transmissions in a constellation of satellites. Two distinct channel models are considered: error-free and erasure channel. For the error-free channel, given the constellation configuration and the number of packets, we aim at scheduling them in order to maximize the area over the earth surface where the entire message can be decoded (i.e., all packets are received). We obtain a binary linear programming problem, that can be optimally solved, yet with a substantial computational effort. Thus, we also propose a suboptimal solution, based only on the relative distance among the constellation satellites. In the erasure channel scenario we consider two alternative, dual scheduling objectives: either the minimization of the maximum packet error rate (PER) experienced by receivers over the earth surface, or the maximization of the area in the earth surface, where all packets are received with PER below a given threshold. Note that a device receiving the same packet by multiple satellites in view may reduce the PER on that packet. The scheduling task for both objectives are framed as integer linear programming problems, and suboptimal, still less computationally demanding solutions are also developed. Numerical results compare the proposed techniques with the random scheduling approach of [1], in terms of either the coverage area fraction (for error-free reception) or the statistics of PER (for the scenario with errors).

One of the results of [70] is that we showed that no scheduling strategy could achieve full coverage and reliable reception for 4 packets at the same time, using the Galileo GNSS. Thus, the second work [71] focuses on the *latency* minimization, i.e., the minimization of the time by which a message is received by all the devices on the ground. We propose a scheduling of packets over both the satellites and the time, to overcome the limitations of existing solutions. We denote as *round* the time used for the transmission of a single packet, and consider that the transmission of the entire message spans several rounds. The scheduling of packets on the satellites among multiple rounds can be performed with two alternative objectives: either a) the minimization of the maximum latency among all receivers, or b) the maximization of the average received packets per round. In the first case, we consider latency as the key metric and aim at minimizing it for the receiver in the worst conditions. With the latter objective instead we aim at maximizing the data rate. A third objective is a variation of b), where we also aim at maximizing the coverage at each round. Numerical results show the validity of our solutions and the improvement with respect to the solution in [70], obtaining reduced average and maximum latency.

The problem of latency minimization for broadcasting services has been studied in different contexts, e.g., WSN [72] and cognitive radio networks (CRN) [73]. Still, We remark that our scenario has significant peculiarities, with respect to the typical broadcast wireless scenarios, namely: a) no feedback channel is available from receiver to transmitter; b) the bipartite graph modeling the network is not complete, since only a subset of satellites is in view of each receiver; c) the position of transmitters and receivers is periodically changing over time, hence the scheduling solution will be time-variant and periodic. Thus, we must develop solutions specifically targeted to the GNSS context.

The Chapter is organized as follows. Section 5.2 describes the system model. Section 5.3 discusses the strategies for single-round message scheduling. Numerical results for the single-round scheduling are reported in Section 5.4. Section 5.5 presents the solutions for multi-round scheduling; numerical results are then shown in Section 5.6. Section 5.7 draws the conclusions.

5.2 System Model

We consider a scenario where a set $S = \{1, \dots, M\}$ of M GNSS satellites aims at transmitting a message p to ground receivers. Since the transmission rate is low and the message p is typically sent periodically, to reduce the latency, we split p into K packets of equal size $\{p_1, \dots, p_K\}$, so that different satellites may transmit distinct packets at the same time. We do not consider any form of packet coding therefore all the packets are needed to reconstruct p.

In particular we assume the ground receivers to be distributed over a region A, described by latitude and longitude coordinates $(\varphi, \lambda) = \boldsymbol{x}$.

We indicate as $S_{k,n} \subset S$ the set of satellites transmitting packet k at round n. Satellite $s \in S$ is assumed to be in view for a ground receiver at position $\boldsymbol{x} \in A$ if its elevation angle $\alpha_s(\boldsymbol{x}) > \alpha_{\min}$, with $\alpha_{\min}$ a suitable threshold. Formally, we introduce the *satellite visibility maps* as

$$v_{s,n}(\boldsymbol{x}) \triangleq \begin{cases} 1 & \text{if } \alpha_{s,n}(\boldsymbol{x}) \geq \alpha_{\min}, \\ 0 & \text{if } \alpha_{s,n}(\boldsymbol{x}) < \alpha_{\min}, \end{cases} \tag{5.1}$$

indicating that satellite s is visible by a receiver in $\boldsymbol{x}$ during round n if $v_{s,n}(\boldsymbol{x}) = 1$. We remark that, indeed, by picking a large $\alpha_{\min}$, we are potentially discarding useful satellites. However signals coming from low elevation satellites are subject to atmospheric and multipath distortion, or possibly even blocked by nearby obstacles [2, Chapter 7], hence these signals can typically be discarded a priori without degrading the performance of the receiver.

Time is divided into rounds, each of duration T in which one packet is transmitted. The transmission of the entire message lasts multiple rounds. Each packet is in general transmitted by multiple satellites over multiple rounds to reach all ground receives. We consider that the relative positions of satellites and receivers change over the rounds: however, since the round duration, T, is much smaller than the orbital period of the satellites, we assume satellite positions to be static within each round.

If a ground receiver collects all the K packets from the satellites in view, it can obtain the entire message. We assume that correct decoding of all K sub messages is necessary to reconstruct the overall message, thus neglecting the possibility of employing erasure coding across packets, for the sake of a more compact formulation. Yet our discussion could be adapted to that case with little effort.

We consider both the case of ideal error-free transmission, and the transmission over a more realistic error-prone channel. For the latter scenario, we assume that the receiver can verify the correctness of the received packets using cyclic redundancy check (CRC) and, for simplicity, we will neglect the possibility of undetected errors. Thus, we model the channel between satellite i and the ground receiver as a packet erasure channel [74, Chapter 8], with PER $r_s(\boldsymbol{x})$. We also assume independent erasures from each satellites, so that if a device receives multiple copies of the same packet from different satellites, it can leverage such diversity to reduce the PER for the corresponding packet to the product of the PERs for all satellite transmitting the same packet.

The two scenarios are summarized as follows:

- an ideal scenario, where error-free decoding is assumed for all packets coming from satellites in view, and

- an erasure channel scenario where the PER is a smoothly decreasing function of the satellite elevation angle, namely $r_s(\boldsymbol{x}) = f(\alpha_s(\boldsymbol{x}))$, and the PER for packet p_k is given by $\prod_{s \in S_k} r_s(\boldsymbol{x})$.

We make the following assumptions: each GNSS receiver

1. has a buffer with size at least equal to the length of p, to store all the received packets;
2. can decode all the packets sent by all satellites in view at the same time without interference degradation; this is typically achieved by transmitting messages in a code division multiple access (CDMA) fashion and using a different code for each satellite.

Summarizing, considering round n, we want to find a packet scheduling protocol that will be composed by the following steps

1. satellite $s \in S$ obtains packet $\pi_{s,n} \in 1, \ldots, K$ from the ground station; the choice of $\pi_{s,n}$ is the subject of the scheduling discussed in the next section;
2. satellite s transmits packet $\pi_{s,n}$;
3. each receiver in a position $\boldsymbol{x}$ such that $v_{s,n}(\boldsymbol{x}) = 1$ decodes the packet;
4. if the receiver has already obtained packet $\pi_{s,n}$ during previous rounds or from another satellite, it discards the packet, otherwise it stores the packet in its buffer;
5. once the receiver has collected all the K packets, it reconstructs message p and waits for a new message.

The transmission scheduling at round n is defined by $\mathcal{S}_n = (S_{1,n}, \ldots, S_{K,n})$, where $S_{k,n}$ is the set of satellites transmitting packet k at round n. The transmission scheduling from round 1 to n will be then $\boldsymbol{\mathcal{P}}_n = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$.

5.3 Single-Round Scheduling Strategies

In the first part of this chapter, we are considering single round scheduling strategies. Thus, to ease the notation we will drop the round index n.

We now introduce the scheduling strategies for the two scenarios. We will consider both optimal scheduling, by integer linear programming (ILP) and heuristic scheduling solutions. Each solution is computed starting from the satellite positions hence it is time-dependent: however it does not need to be computed in real time. It can be computed offline and stored, e.g., in a lookup table indexed by the corresponding instant within an orbital period, and applied periodically.

5.3.1 Optimal Scheduling for the Ideal Scenario

In the ideal scenario, where the message transmission is error-free for all satellites in view, we aim at maximizing the *coverage*, i.e., the fraction of the region A where K packets can be received.

Then, for each satellite subset $S_k \subset S$, we compute the binary *subset visibility map* as the logical OR of the satellite visibility maps for all satellites in S_k

$$u_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) = \bigvee_{s \in S_k} v_s(\boldsymbol{x}) = \min \left\{ 1, \sum_{s \in S_k} v_s(\boldsymbol{x}) \right\}. \tag{5.2}$$

The *coverage* can then be written as

$$\gamma(\boldsymbol{\mathcal{P}}) = \frac{1}{|A|} \int_A \prod_{k=1}^{K} u_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x})\, dA, \tag{5.3}$$

where the product represents the logical AND operation among visibility maps for various packets, and for the surface integral we use the customary definition

$$\int_A g(\boldsymbol{x})\, dA = \iint g(\boldsymbol{x}) R^2 \cos\varphi\, d\varphi\, d\lambda.$$

Our optimization problem can then be written as follows

$$\max_{\boldsymbol{\mathcal{P}} \in \boldsymbol{\mathcal{P}}^\star} \gamma(\boldsymbol{\mathcal{P}}), \tag{5.4}$$

where $\boldsymbol{\mathcal{P}}^\star$ is the family of $\mathcal{S}$ partitions. Note that the optimal solution can be obtained by enumerating all the partitions in $\boldsymbol{\mathcal{P}}^\star$ and selecting the one that maximizes $\gamma(\cdot)$. This algorithms however has exponential complexity, since the cardinality of the $\boldsymbol{\mathcal{P}}^\star$ is

$$|\boldsymbol{\mathcal{P}}^\star| = \left\{ {M \atop K} \right\} \geq \frac{1}{2}(K^2 + K + 2)K^{M-K-1} - 1 \tag{5.5}$$

where $\left\{ {n \atop k} \right\}$ is the Stirling number of second kind.

Therefore, we have investigated a more efficient solution, based on ILP.

In order to formulate the problem in the framework of linear programming, we introduce the indicator variables

$$y_{s,k} = \begin{cases} 1 & \text{for } s \in S_k \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Moreover, we replace the continuous space A by a finite subset $\Omega \subset A$ of sampling locations, and correspondingly partition A into a tessellation $\mathcal{A} = \{A(\boldsymbol{x})\}$ indexed by points $\boldsymbol{x}$ in Ω, essentially treating all the receivers in $A(\boldsymbol{x})$ as a single receiver placed at position $(\boldsymbol{x})$. We define the normalized size of each subregion $A(\boldsymbol{x})$ as

$$a(\boldsymbol{x}) = \frac{|A(\boldsymbol{x})|}{|A|}\ , \tag{5.7}$$

and

$$A(\boldsymbol{x}) \cap A(\boldsymbol{x'}) = \emptyset, \qquad \text{if } \boldsymbol{x} \neq \boldsymbol{x'}\ . \tag{5.8}$$

Finally, let us introduce the variables $\bar{u}(\boldsymbol{\mathcal{P}}, \boldsymbol{x})$ defined as

$$\bar{u}(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) = \prod_{1 \leq k \leq K} u_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}), \tag{5.9}$$

Then, (5.4) can be written as follows

Problem 1 (Binary Linear Programming formulation)**.** *Given the coefficients*

$$v_s(\boldsymbol{x}) \in \{0, 1\} \quad \boldsymbol{x} \in \Omega, s \in S \tag{5.10a}$$

$$a(\boldsymbol{x}) \in [0, 1] \quad \boldsymbol{x} \in \Omega, \tag{5.10b}$$

maximize over the variables

$$y_{s,k} \in \{0,1\} \quad k \in \{1,\dots,K\}, s \in S, \tag{5.10c}$$

$$u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}) \in \{0,1\} \quad k \in \{1,\dots,K\}, \boldsymbol{x} \in \Omega, \tag{5.10d}$$

$$\bar{u}(\boldsymbol{\mathcal{P}},\boldsymbol{x}) \in \{0,1\} \quad \boldsymbol{x} \in \Omega, \tag{5.10e}$$

the objective function

$$\max \sum_{\boldsymbol{x}\in\Omega} a(\boldsymbol{x})\bar{u}(\boldsymbol{\mathcal{P}},\boldsymbol{x}), \tag{5.10f}$$

under the constraints (for all $k \in \{1,\dots,K\}$, $\boldsymbol{x} \in \Omega$, $s \in S$),

$$\sum_k y_{s,k} = 1, \tag{5.10g}$$

$$\sum_s y_{s,k} v_s(\boldsymbol{x}) \le M u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}), \tag{5.10h}$$

$$\sum_s y_{s,k} v_s(\boldsymbol{x}) \ge u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}), \tag{5.10i}$$

$$\sum_k u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}) \ge K\bar{u}(\boldsymbol{\mathcal{P}},\boldsymbol{x}). \tag{5.10j}$$

The solution to this problem can be obtained, for instance, via the Matlab function `intlinprog`, where the complexity is set by the high number of constraints in (5.10h) – (5.10j): in particular, by considering a sampling with $|\Omega| = N_S$ points we have in total $(2K+1)N_S + M$ constraints. Since the complexity required to solve the binary linear programming problem grows exponentially with the number of constraints, we notice that even this approach becomes unfeasible for dense Ω. Thus we now propose two suboptimal, but faster, heuristic solutions.

5.3.2 Heuristic Solutions for the Ideal Scenario

The first heuristic algorithm, max inter-group distance (MID), stems from the observation that satellites far from each other will seldom be in view at the same time, hence it is reasonable to assign them to the same subset S_k. Vice versa, satellites close to each other will be both in view from many positions on the ground, hence they should send different packets.

Having defined the distance matrix $\boldsymbol{D} = [d_{i,j}]$, where entry $d_{i,j}$ represents the Euclidean distance between satellites i and j, and the sum of distances for an arbitrary satellite subset $X \subset S$ as

$$s_{\mathrm{d}}(X) = \sum_{i,j\in X} d_{i,j}. \tag{5.11}$$

we iteratively build the S_k making up the chosen partition. We start by filling the set S_1 with the $\ell = \lceil M/K \rceil$ satellites in S that maximize $s_{\mathrm{d}}(S_1)$, then remove them from S. Then, we repeat the operation starting instead from the set $S' = S \setminus S_1$, filling the set S_2. We repeat this operation until we obtain all the K subsets and all the satellites have been included, hence $S' = \emptyset$. The resulting procedure is shown in Algorithm 1.

For the second algorithm, denoted min inter-group weight (MIW), we define weight $w_{i,j}$ for satellites i and j as the fraction of the area A that is covered at the same time by both satellites, i.e.,

$$w_{i,j} = \sum_{(\boldsymbol{x})\in\Omega} v_i(\boldsymbol{x})v_j(\boldsymbol{x})a(\boldsymbol{x}). \tag{5.12}$$

Hence, similarly to the previous algorithm, we fill S_1 as the subset of ℓ satellites from S, that minimize the weight function

$$s_{\mathrm{w}}(X) = \sum_{i,j \in X} w_{i,j}. \tag{5.13}$$

Then we repeat these operations, filling S_2 with satellites from $S' = S \setminus S_1$, and so on, until we fill all the K sets, thus obtaining the partition $\mathcal{S} = \{S_1, \dots, S_K\}$. The algorithm is thus analogous to the MID algorithm, but it takes as input the weight matrix $\boldsymbol{W}$ instead of the distance matrix $\boldsymbol{D}$, and chooses S_k minimizing (5.13) instead of maximizing (5.11).

Notice that the repeated computation of ℓ for each k is needed since groups may not contain the same number of satellites, ranging between $\lfloor M/K \rfloor$ and $\lceil M/K \rceil$. Regarding the computational cost of these approaches, the largest cost comes from the computation of all the possible combinations of ℓ satellites from the set S': if, for simplicity, we assume that all the subsets contain exactly $\ell = M/K$ satellites we have to compute

$$\sum_{k=0}^{K-1} \binom{M - k\ell}{\ell} \tag{5.14}$$

possible combinations. which represents an upper bound on the total number of checks that need to be done in order to find the best partition. Notice that the bound is exponential in M only, and is independent of N_Ω, which is only used in computing $\boldsymbol{W}$ for the setup of the MIW algorithm: in the GNSS scenarios we will always have to deal with constellations having (relatively) small values of M, therefore the heuristic approaches will be indeed much faster than the ILP.

5.3.3 Optimal Scheduling for the Erasure Channel Scenario

In the erasure channel scenario, message detection is prone to errors, and the error probability is different for each satellite-receiver link. Therefore, we take errors into account for the definition of the scheduling metric. First note that the visibility condition is included in the definition of PER $r_s(\boldsymbol{x})$, since when satellite s is not in view from position $\boldsymbol{x}$ we can set $r_s(\boldsymbol{x}) = 1$. Assuming that page error events for distinct satellites are independent, the probability that all copies of packet p_k, are

Algorithm 1 max inter-group distance (MID)

```
Input: $\boldsymbol{D}, K, S$
Output: $\mathcal{P}$
  $S' := S$
  for $k = 1, \dots, K$ do
    $\ell := \lceil |S'|/(K - k + 1) \rceil$
    find all combinations of $\ell$ satellites from $S'$
    for each combination $X$ do
      compute $s_{\mathrm{d}}(X)$
    end for
    $S_k := \arg\max_X s_{\mathrm{d}}(X)$
    $S' := S' \setminus S_k$
  end for
  return $\boldsymbol{\mathcal{P}} := \{S_1, S_2, ..., S_k\}$
```

received with errors by a receiver in position $\boldsymbol{x} \in A$ is

$$q_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) = \prod_{s \in S_k} r_s(\boldsymbol{x}). \tag{5.15}$$

Now we consider two alternative objectives when performing scheduling: a) the minimization of the maximum PER, or b) the maximization of coverage, with a given PER upper bound. In details:

Problem 2 (**Min-max PER (MP) criterion**). *We aim at minimizing the maximum PER, i.e.,*

$$\arg\min_{\boldsymbol{\mathcal{P}} \in \boldsymbol{\mathcal{P}}^\star} \gamma'(\boldsymbol{\mathcal{P}}), \tag{5.16a}$$

where

$$\gamma'(\boldsymbol{\mathcal{P}}) = \max_{\boldsymbol{x} \in \Omega} q_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}). \tag{5.16b}$$

Problem 3 (**Bounded PER (BP) criterion**). *For a given threshold PER $q_{\max}$ let us define a new map as*

$$u''_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) = \begin{cases} 1 & q_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) \leq q_{\max}, \\ 0 & q_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) > q_{\max}. \end{cases} \tag{5.17a}$$

With the bounded PER criterion we aim at maximizing the area where PER $\leq q_{\max}$ i.e.,

$$\arg\min_{\boldsymbol{\mathcal{P}} \in \boldsymbol{\mathcal{P}}^\star} \gamma''(\boldsymbol{\mathcal{P}}), \tag{5.17b}$$

where

$$\gamma''(\boldsymbol{\mathcal{P}}) = \sum_{\boldsymbol{x} \in \Omega} \min_k u''_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) a(\boldsymbol{x}). \tag{5.17c}$$

We proceed to write problems 2 and 3 as mixed-integer linear programming (MILP) and ILP problems, respectively. Let us define

$$\begin{aligned} \tilde{r}_s(\boldsymbol{x}) &= \log_{10} r_s(\boldsymbol{x}), && (5.18a) \\ \tilde{q}_{\max} &= \log_{10} q_{\max}. && (5.18b) \end{aligned}$$

Then, problem 2 can be formalized as follows:

Problem 4 (**Min-max PER (MP), MILP formulation**). *Given the coefficients*

$$\tilde{r}_s(\boldsymbol{x}) \in \mathbb{R}^- \quad s \in S, \boldsymbol{x} \in \Omega \tag{5.19a}$$

minimize $\tilde{q}_{\max}$ over the variables

$$\begin{aligned} & y_{s,k} \in \{0, 1\} \quad s \in S, k \in \{1, \ldots, K\} && (5.19b) \\ & \tilde{q}_{\max} \in \mathbb{R}^- && (5.19c) \end{aligned}$$

under the constraints

$$\sum_k y_{s,k} = 1 \quad s \in S \tag{5.19d}$$

$$\sum_s y_{s,k} \tilde{r}_s(\boldsymbol{x}) \leq \tilde{q}_{\max} \quad k \in \{1, \ldots, K\}, \boldsymbol{x} \in \Omega \tag{5.19e}$$

For problem 3, we observe that it is similar to Problem 1, by replacing $v_s(\boldsymbol{x})$ with $\tilde{r}_s(\boldsymbol{x})$ and constraint (5.47d)

Problem 5 (Maximum coverage with BP, ILP formulation). *Given the coefficients*

$$\tilde{r}_s(\boldsymbol{x}) \in \mathbb{R}^- \quad \boldsymbol{x} \in \Omega, s \in S, \tag{5.20a}$$
$$a(\boldsymbol{x}) \in \mathbb{R} \quad \boldsymbol{x} \in \Omega, \tag{5.20b}$$

maximize over the variables

$$y_{s,k} \in \{0,1\} \quad k \in \{1,\dots,K\}, s \in S, \tag{5.20c}$$
$$u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}) \in \{0,1\} \quad k \in \{1,\dots,K\}, \boldsymbol{x} \in \Omega, \tag{5.20d}$$
$$\bar{u}(\boldsymbol{\mathcal{P}},\boldsymbol{x}) \in \{0,1\} \quad \boldsymbol{x} \in \Omega, \tag{5.20e}$$

the objective function

$$\hat{f}_{\mathrm{c}} = \max \sum_{\boldsymbol{x}} a(\boldsymbol{x})\bar{u}(\boldsymbol{\mathcal{P}},\boldsymbol{x}), \tag{5.20f}$$

under the constraints $\forall k \in \{1,\dots,K\}$, $\boldsymbol{x} \in \Omega$, $s \in S$

$$\sum_k y_{s,k} = 1, \tag{5.20g}$$
$$\sum_s y_{s,k}\tilde{r}_s(\boldsymbol{x}) \le \tilde{q}_{\max} u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}), \tag{5.20h}$$
$$u_k(\boldsymbol{\mathcal{P}},\boldsymbol{x}) \ge \bar{u}(\boldsymbol{\mathcal{P}},\boldsymbol{x}). \tag{5.20i}$$

5.3.4 Heuristic Scheduling for the Erasure Channel Scenario

The heuristic solutions from Section 5.3.2 can be applied to the erasure channel scenario, as well. In particular, since the MID algorithm operates only on the intersatellite distances, it can be employed here with no modifications. As regards the MIW, it needs instead to be modified, by replacing the satellite visibility maps v_s with a version depending on the PER threshold, that is

$$v_s(\boldsymbol{x}) = \begin{cases} 1 & \text{if } r_s(\boldsymbol{x}) \le q_{\max}, \\ 0 & \text{if } r_s(\boldsymbol{x}) < q_{\max} \end{cases}, \tag{5.21}$$

or equivalently, by replacing $\alpha_{\min}$ in (5.1) with the value that satisfies $f(\alpha) = q_{\max}$ in the relationship between PER and elevation angle. By correspondingly adapting the weights in (5.12) we can use the adapted weight matrix $\boldsymbol{W}$ as the input to Algorithm 1 and obtain the solution in the erasure channel scenario.

5.4 Numerical Results for Single Round Scheduling

In order to show the effectiveness of the proposed solutions we consider A as the whole Earth surface and take Ω to be a regular Cartesian sampling of the (λ, φ) plane with N_φ and N_λ points in each direction, respectively. For $\boldsymbol{x} = (\varphi, \lambda)$ the corresponding tile is $[\varphi - \pi/(2N_\varphi), \varphi + \pi/(2N_\varphi)] \times A(\boldsymbol{x}) \in [\lambda - \pi/N_\lambda, \lambda + \pi/N_\lambda]$ with (normalized) area

$$a(\boldsymbol{x}) = a(\varphi, \lambda) = \frac{1}{N_\lambda}\cos(\varphi)\sin\left(\frac{\pi}{2N_\lambda}\right), \tag{5.22}$$

and R_{E} is the Earth radius. For all numerical results described in the following, we employed $N_\lambda = 48$ and $N_\varphi = 24$.

We considered the Galileo GNSS constellation S_0, composed of 24 medium Earth orbit (MEO) satellites distributed over 3 orbital planes, from which we consider the set $S \subset S_0$ of actually transmitting satellites at a given time as those connected to one of the 5 uplink stations [75] [1]. In order to measure the performance of our algorithm we consider for comparison the random scheduling (RS) [1], where the partition $\mathcal{P}$ is uniformly randomly selected among all the possible partitions. The masking elevation angle is $\alpha_{\min} = 10°$ and we consider as interval $1 \leq K \leq N_s/2$.

In implementing the ILP for the error free case we have precomputed the areas covered by each possible combination of satellites in view, thus reducing the actual computation load for solving the problem. Figure 5.1 shows the coverage (5.3) for the ideal scenario using RS, MID, MIW and the ILP approaches.

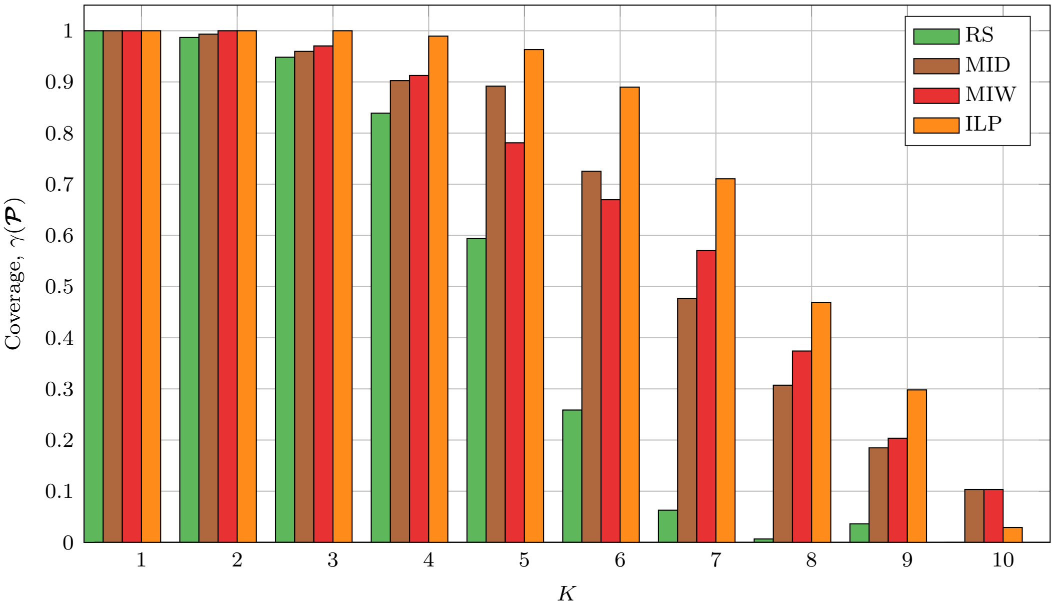


FIGURE 5.1: Coverage $\gamma(\boldsymbol{\mathcal{P}})$ as a function of the number of scheduled packets, K, obtained using the developed scheduling strategies in the ideal scenario: the RS (green), MID in (brown), MIW (red) and the ILP (orange).

In modeling the erasure channel scenario and the PER distribution $r_s(\boldsymbol{x})$ we have taken as a reference for the relationship $f(\alpha)$ between PER and elevation angle, the measurements reported both in [76] and [77]: in particular we have considered the measurement taken in the sub-urban scenario for a Galileo mass market receiver. Moreover we have selected as upper bound for the erasure probability the value $q_{\max} = 0.1$. Figure 5.2 shows the results obtained for the metric $\gamma'(\mathcal{C})$ as defined in Prob. 2, using the RS [1], the heuristic algorithmsMID and MIW and the MILP solutions both in the MP (optimal for this case) and BP formulation. On the other hand, Figure 5.3 shows instead the results obtained for the metric $\gamma''(\mathcal{C})$ as defined in Prob. 3 for the same techniques (in this case ILP BP would be optimal).

Although in both the scenarios the ILP/MILP solution does not achieve the optimal results within the set time limit of 12 h, for large values of K (i.e., for $K \geq 6$ in the ideal scenario and $K \geq 5$ for the erasure channel scenario), we still notice that both the ILP and the MILP achieve the best coverage; on the other hand the heuristic algorithms obtain a lower coverage but their execution requires a much lower computational cost:

TABLE 5.1: CPU time spent using the presented dissemination algorithms in function of K. Last two columns are the solutions for the erasure channel scenario.

K	MID	MIW	ILP	MP	BP
2	4.255 s	3.440 s	2.534 s	5.392 s	0.793 s
3	2.034 s	1.265 s	2.868 s	12.499 s	8.050 s
4	1.086 s	0.323 s	3.297 s	62.308 s	> 12h
5	0.864 s	0.148 s	4.076 s	350.642 s	> 12h
6	0.849 s	0.139 s	8.995 s	21.842 min	> 12h
7	0.755 s	0.057 s	> 12h	28.417 min	> 12h
8	0.776 s	0.083 s	> 12h	> 12h	> 12h
9	0.759 s	0.057 s	> 12h	> 12h	> 12h
10	0.704 s	0.055 s	> 12h	> 12h	> 12h

we report on Table 5.1 the time spent by ILP, MILP and the heuristic algorithms; notice that we only reported the results for the heuristic algorithms in the ideal scenario since there is no difference between the algorithms in terms of computational cost in the two considered scenarios.

However it is also important to notice that the solutions with lower values of K, i.e. $2 \leq K \leq 6$, are more feasible in practice since by using larger sets S_k we build a more fault tolerant configuration and in these conditions, the heuristic approaches achieve a coverage which is only slightly lower than the one obtained by the ILP/MILP, MP and BP, hence their application is still relevant.

In order to summarize the results for the Erasure Channel scenario, we define the probability $\hat{q}(\boldsymbol{\mathcal{P}}, \boldsymbol{x})$ that at least one of the k packets is lost by the receiver in position (φ, λ) as

$$\hat{q}(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) = 1 - \prod_{k=1}^{K} \Big(1 - q_k(\boldsymbol{\mathcal{P}}, \boldsymbol{x})\Big) \tag{5.23}$$

and evaluate the CDF for $\hat{q}(\boldsymbol{x})$, denoted by $F(q)$, that represents the fraction of Earth surface where $\hat{q}(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) \leq q$. We run 100 simulations covering a period of 25 h, with $K = 3$ and $q_{\max} = 0.04$ and we report the CDF of q in Figure 5.4: all the proposed algorithms outperform the RS of [1], while for low values of q the MILP and heuristics achieve similar performances, for high values of q the MILP algorithms achieve the best results; in particular, as expected, the BP algorithm outperform MP for $q \approx q_{\max}$, while MP outperforms BP for $q \geq q_{\max}$.

5.5 Multi-round Message Scheduling in GNSS Packet Broadcasting

In this Section we discuss the problem of multi-round packet broadcasting: as shown in the first part of the Chapter, even in the ideal scenario it is not possible to deliver within a single round more than 4 packets to all the receivers. This means that, to deliver long messages, we have to resort to multi round message scheduling, i.e., scheduling strategies that last for more than one round. In the rest of the Chapter we investigate solutions for the multi-round packet broadcasting problem.

We introduce a new set of metrics that will be used to design the scheduling algorithms, namely a) the average number of different received packets, b) the maximum latency, and c) the average latency.

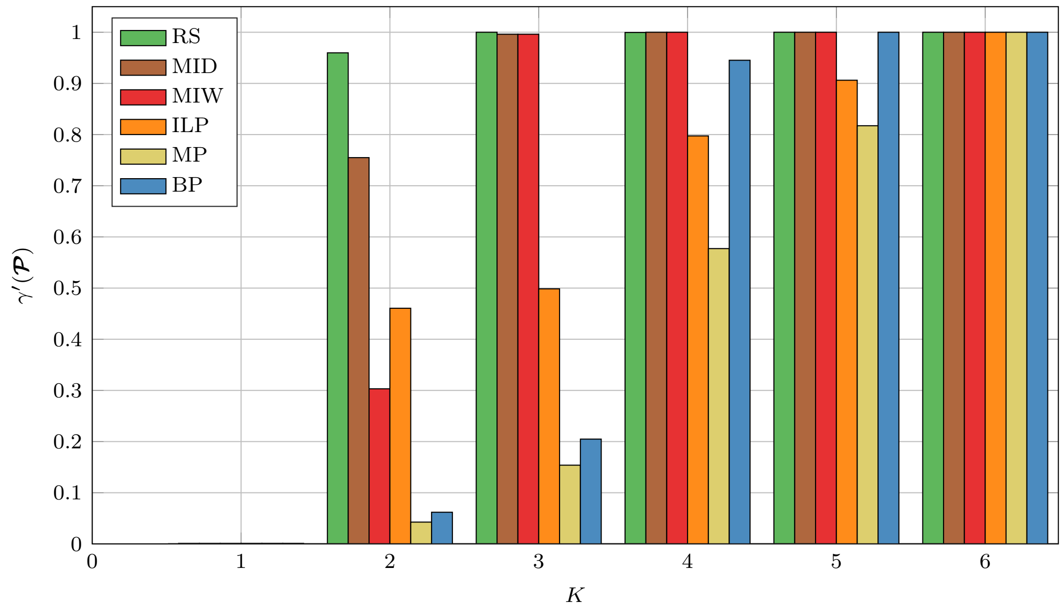


FIGURE 5.2: Values of the metric $\gamma'(\mathcal{P})$ described in Prob. 2, as a function of the number of scheduled packets, K, obtained by the random scheduling [1], the heuristic algorithms MID and MIW, the ILP solution for the BP problem, and the (optimal) MILP solution for the MP problem. The value for $K = 1$, common to all methods is $\gamma(1) \simeq 10^{-3}$.

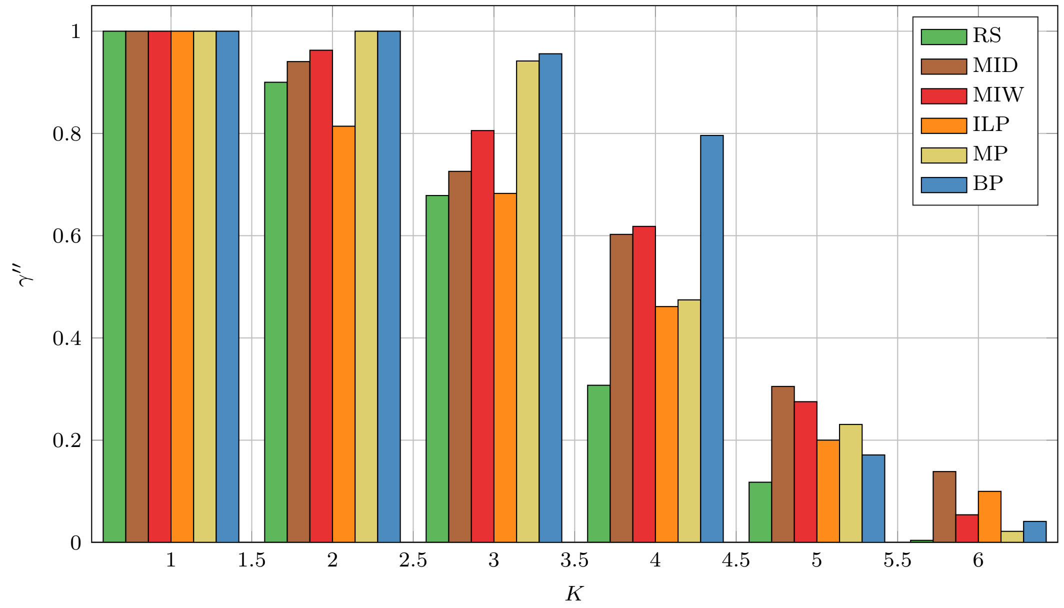


FIGURE 5.3: Values of the metric $\gamma''(\mathcal{C})$ described in Prob. 3, as a function of the number of scheduled packets, K, obtained by the random scheduling [1], the heuristic algorithms MID and MIW, the (optimal) ILP solution for the BP problem and the MILP solution for MP problem.The value for $K = 1$, common to all methods is $\gamma(1) = 1$.

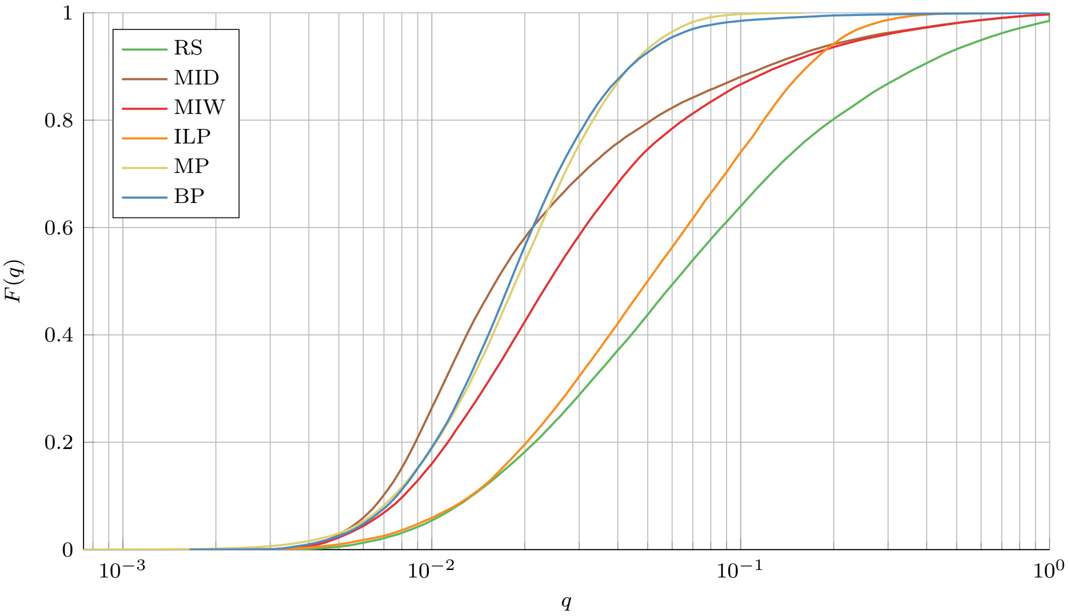


FIGURE 5.4: CDF of the probability of correctly receiving all the $K = 3$ packets for the Erasure Channel scenario.

Average Number of Received Packets The first metric is related to the number of different received packets in a given time and in a given area, thus it merges coverage and throughput performance. By introducing the *availability of packet k at position $\boldsymbol{x}$ by round n*

$$u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) \triangleq \begin{cases} 1 & \text{if } \sum_{m=1}^{n} \sum_{s \in S_{k,m}} v_{s,m}(\boldsymbol{x}) > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{5.24}$$

we denote the *total number of different received packets at position $\boldsymbol{x}$ up to round n*, as

$$\eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) \triangleq \sum_{k=1}^{K} u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}). \tag{5.25}$$

This provides an indication on the data rate in a given position. Lastly, the *average number of different received packets in A up to round n* is

$$\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n) \triangleq \frac{1}{|A|} \int_A \eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) d\boldsymbol{x}. \tag{5.26}$$

This metric is related to the coverage up to the current frame. Note however that a value of $\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n)$ does not ensure that all points in the area have received the same number of packets.

Maximum Latency The second metric is referred to the latency, i.e., the number of rounds necessary for a receiver to obtain all the K packets and reconstruct p. First, we define the latency for a receiver in position $\boldsymbol{x}$ as the following time (in rounds)

$$\tau(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) \triangleq \min \Big\{ n : \eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) = K \Big\}, \tag{5.27}$$

where $\boldsymbol{\mathcal{P}} = (\mathcal{S}_1, \mathcal{S}_2, \ldots)$ is the scheduling sequence at all rounds ($n = 1$ to ∞). Indeed, in (5.27) we consider the first round wherein all K packets have been received, thus we exploit the whole sequence of scheduled transmissions $\boldsymbol{\mathcal{P}}$. The *maximum latency* is defined as the maximum latency among all receivers in the area A, i.e.,

$$\tau_{\max}(\boldsymbol{\mathcal{P}}) \triangleq \max_{\boldsymbol{x} \in A} \tau(\boldsymbol{\mathcal{P}}, \boldsymbol{x}). \tag{5.28}$$

Note that the latency is a relevant metric for several applications, in particular in the area of automation, including self-driving cars or unmanned aerial vehicles. Minimizing the maximum latency means to keep under control the latency in the worst case scenario, which is a suitable criterion for time-critical applications. Still, considering the maximum latency may be significantly penalizing for the average number of different received packets. Therefore we also consider another metric related to the latency.

Average Latency The third metric is the *average latency* for all receivers in area A, defined as

$$\bar{\tau}(\boldsymbol{\mathcal{P}}) \triangleq \frac{1}{|A|} \int_A \tau(\boldsymbol{\mathcal{P}}, \boldsymbol{x}) d\boldsymbol{x}. \tag{5.29}$$

On one hand, keeping the average latency under control does not guarantee good performance in the worst case, but only on average. On the other hand, this milder metric may allow better performance in terms of average received packets, as it will be shown in the following.

5.5.1 Performance bounds

In this Section we obtain bounds on the average number of different received packets, coverage and latency. We will also exploit these results to prove the optimality of one of the proposed scheduling algorithms and develop the solution described in Section 5.5.2.3.

The first bound relates the average number of different received packets to the coverage of the area and the total number of different received packet, i.e., the coverage and the throughput, as already mentioned when introducing the metric. About the coverage, from the availability of packet k at position $\boldsymbol{x}$ by round n, $u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x})$ we obtain the *availability of the entire message at position* $\boldsymbol{x}$ *by round* n as

$$\hat{u}_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) \triangleq \begin{cases} 1 & \text{if } \prod_{k=1}^{K} u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{5.30}$$

This denotes the receivers that at round n, were able to actually receive all the K packets. Next, we can formally introduce the *(fractional) coverage of* A *by round* n

$$\bar{u}_n(\boldsymbol{\mathcal{P}}_n) \triangleq \frac{1}{|A|} \int_A \hat{u}_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) d\boldsymbol{x}. \tag{5.31}$$

All the above expressions always counted just the different packets: in the next we will consider instead the total number of received packets. About the average number of received packets, we define the *total number of received packets (possibly with repetitions) at position* $\boldsymbol{x}$ *up to round* n as

$$C_n(\boldsymbol{x}) \triangleq \sum_{s \in S} \sum_{m=1}^{n} v_{s,m}(\boldsymbol{x}), \tag{5.32}$$

and its average over A, i.e., number of received packets per area, as

$$\bar{C}_n = \frac{1}{|A|}\int_A C_n(\boldsymbol{x})d\boldsymbol{x}. \tag{5.33}$$

We can now formulate the following proposition given upper and lower bounds to the average number of received packets.

Proposition 3. *The average number of different received packets is bounded as*

$$K\bar{u}_n(\boldsymbol{\mathcal{P}}_n) \leq \bar{\eta}_n(\boldsymbol{\mathcal{P}}_n) \leq \min(K, \bar{C}(n)). \tag{5.34}$$

Proof. We start proving the left hand side: given the sequence of partitions $\boldsymbol{\mathcal{P}}_n$ such that $\bar{u}_n(\boldsymbol{\mathcal{P}}_n) = \beta$, at least a fraction β of the receivers obtained all the K packets at the end of the nth round, therefore the average number of delivered packets is at least $K\,\beta$, i.e., $\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n) \geq K\bar{u}_n(\boldsymbol{\mathcal{P}}_n)$.

For the right hand side we can write

$$C_n(\boldsymbol{x}) = \sum_{s\in S}\sum_{m=1}^{n} v_{s,m}(\boldsymbol{\mathcal{P}}_m, \boldsymbol{x}) \geq \eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}), \tag{5.35}$$

then by averaging both sides of (5.35) over A and considering that $\eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) \leq K$, we obtain (5.34). □

From this bound we observe that a solution $\boldsymbol{\mathcal{P}}_n$ achieving full coverage also maximizes the average number of different received packets, with $\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n) = K$; on the other hand among the sequences $\boldsymbol{\mathcal{P}}_n$ that achieve the same partial coverage, some may obtain a higher average number of different received packets. This result will be used for the development of the scheduling algorithm in Section 5.5.2.3.

5.5.1.1 Maximum Diversity Scenario

We also derive bounds on the maximum and average latency. In this case, we focus on a particular scenario, that we denote as maximum diversity scenario, characterized by the fact that each receiver obtains a different packet from each satellite in view. This is clearly a very favorable condition, not always met in practice, since in each round the receiver obtains the maximum number of packets, for a given set of satellites in view. Thus, we will obtain bounds on the performance associated to a best case scenario. In formulas, the maximum diversity scenario can be alternatively described by indicating that the message is received once K satellites have been in view, and thus $C_n(\boldsymbol{x}) = K$ packets have been received. In terms of the availability of the entire message at position $\boldsymbol{x}$ by round n we have

$$\hat{u}_n^*(\boldsymbol{x}) = \begin{cases} 1 & \text{if } C_n(\boldsymbol{x}) \geq K \\ 0 & \text{if } C_n(\boldsymbol{x}) < K \end{cases}, \tag{5.36}$$

or alternatively, the total number of different received packets at position $\boldsymbol{x}$ up to round n is

$$\eta_n^*(\boldsymbol{x}) = \min(K, C_n(\boldsymbol{x}))\;. \tag{5.37}$$

Note that in both $\hat{u}_n^*(\boldsymbol{x})$ and $\eta_n^*(\boldsymbol{x})$ we omitted the indication of the scheduling $\boldsymbol{\mathcal{P}}_n$, as we are considering an ideal scenario which may not be feasible, i.e., for which no scheduling allows to achieve such performance.

However, this ideal scenario will provide bounds on the latency metrics. In particular, by averaging over the area, we obtain upper bounds on the average number of different received packets $\bar{\eta}_n^*$ and the coverage $\bar{u}_n^*$. These values are upper bounds for the performance achieved in any scenario, i.e., we always have

$$\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n) \leq \bar{\eta}_n^*, \quad \bar{u}_n(\boldsymbol{\mathcal{P}}_n) \leq \bar{u}_n^* \ . \tag{5.38}$$

The latency in the maximum diversity scenario for the receiver in position $\boldsymbol{x}$ is

$$\tau^*(\boldsymbol{x}) \triangleq T \min \left\{ n : \hat{u}_n^*(\boldsymbol{x}) = 1 \right\} , \tag{5.39}$$

from which we obtain correspondingly $\tau_{\max}^*$, $\bar{\tau}^*$. These values are bounds for the performance achieved with any scheduling solution, i.e., we always have

$$\tau_{\max}(\boldsymbol{\mathcal{P}}) \geq \tau_{\max}^*, \quad \bar{\tau}(\boldsymbol{\mathcal{P}}) \geq \bar{\tau}^*, \quad \forall \boldsymbol{\mathcal{P}} \ . \tag{5.40}$$

5.5.2 Scheduling Solutions For Multi-round Scheduling

In this Section we propose scheduling algorithms for the considered multi-round transmission problem. We first aim at minimizing the maximum latency $\tau_{\max}(\boldsymbol{\mathcal{P}})$ and propose a solution that is optimal under a suitable condition on $C_n(\boldsymbol{x})$, the total number of received packets up to round n. Then, we observe that the min-max latency algorithm neither minimizes the average latency nor maximizes the throughput. Still, in order to optimize these two metrics we should jointly schedule transmission at all rounds, as from (5.29) and (5.49). This leads to an extremely complex solution. In summary, we propose three heuristic approaches: a) minimizing the maximum latency, b) maximizing the average received packets per round, and c) maximizing the coverage as the primary objective and the average received packets per round as a secondary objective.We remark that, in general, a solution that minimizes the maximum latency does not minimize the average latency, and viceversa. Thus, these will be represented by two separate objective and metrics.

5.5.2.1 Minimization of the Maximum Latency (MIN-MAX)

We now consider the problem of finding a scheduling $\boldsymbol{\mathcal{P}}$ that minimizes the maximum latency among all the receivers, i.e.,

$$\min_{\boldsymbol{\mathcal{P}}} \tau_{\max}(\boldsymbol{\mathcal{P}}). \tag{5.41}$$

First, we observe that the problem (5.41) is equivalent to the cascade of several problems, one for each round as shown by the following proposition and can be solved accordingly.

Proposition 4. *Let* $C_{\min,n} \triangleq \min_{\boldsymbol{x} \in A} C_n(\boldsymbol{x})$, *if*

$$C_{\min,n} = \sum_{m=1}^{n} \min_{\boldsymbol{x} \in A} \sum_{s \in S} v_{s,m}(\boldsymbol{x}). \tag{5.42}$$

the min-max latency problem is equivalent to maximizing the number of packets that can be transmitted in a single round with full coverage.

We report the proof in the Appendix.

Algorithm 2 Min Max Latency Solution

```
Input: $K, A$
Output: $\boldsymbol{\mathcal{P}}$
  $n \leftarrow 0$, $K' \leftarrow K$
  while $K' > 0$ do
    $n \leftarrow n + 1$
    $v_{s,n}(\boldsymbol{x}) \leftarrow$ comp_visibility_maps$(n, A)$
    $K_n \leftarrow 1; \gamma \leftarrow 1; \boldsymbol{\mathcal{P}}_n \leftarrow \{S\}$
    while $K_n < K'$ and $\gamma = 1$ do
      $\{\boldsymbol{\mathcal{P}}'_n, \gamma\} =$ solve_max_coverage$(A, K_n, v_{s,n}(\boldsymbol{x}))$
      if $\gamma = 1$ then
        $\boldsymbol{\mathcal{P}}_n = \boldsymbol{\mathcal{P}}'_n$
        $K_n \leftarrow K_n + 1$
      end if
    end while
    $K' = K' - K_n$
  end while
```

Condition (5.42) essentially states the equivalence between the minima of the sum, i.e., $C_{\min,n}$, and the sum of the minima. This condition can easily be met in this context since if the round duration T is small enough, the visibility map $v_{s,m}(\boldsymbol{x})$ does not exhibit dramatic changes between rounds, hence, with high probability, $\min_{\boldsymbol{x}\in A}\sum_{s\in S} v_{s,m}(\boldsymbol{x})$ stays constant for a few rounds. This allows requirement (5.42) to be met. We will confirm the optimality of this approach in Section 6.5.4, showing that $\min_{\boldsymbol{x}\in A}\sum_{s\in S} v_{s,m}(\boldsymbol{x}) = 4$ for most rounds.

Hence, we propose the following algorithm. Let K_n be the number of different packets transmitted at round n and not transmitted in previous rounds, so that $\sum_n K_n = K$. Then, K_n is chosen as the maximum number of packets that can be transmitted in a single round with full coverage. To this end, we iteratively increase K_n starting from $K = 1$ and resort to the binary integer linear programming (BILP) algorithm of [70], discussed in Section 5.3.1, to compute the corresponding coverage $\gamma(\cdot)$, until we reach full coverage. The resulting solution is reported in Algorithm 2, where $K' = K - \sum_n K_n$ represents the number of packets left to be transmitted.

In the algorithm we exploit the following functions:

`comp_visibility_maps` : computes the satellite positions at round n and outputs the visibility maps, $v_{s,n}(\boldsymbol{x})$;

`solve_max_coverage` : implements the (single round) coverage optimization (Section 5.3.1), which we have shown to be effective for small K values.

Algorithm 2 follows a divide and conquer approach. If the algorithm is iterated for n rounds, the overall computational cost is n times the cost of the single round scheduling solution, which from [70] is exponential in the size $|A|$ and the number of packets to be transmitted, $K_n \leq C_{\min,n}$. However, since $C_{\min,n}$ is typically small ($C_{\min,n} \approx 4$), in practice we expect the algorithm to converge quickly.

We now observe that the min-max latency algorithm neither minimizes the average latency nor maximizes the throughput. Still, in order to optimize these two metrics we should jointly schedule transmissions at all rounds, as from (5.29) and (5.49), yielding a high-complexity solution. Then, in the following we consider two other suboptimal approaches.

5.5.2.2 Maximization of the Average Number of Different Received Packets (MARP)

We now consider the problem of maximizing the average number of different received packets $\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n)$ at each round n, i.e., given the sequence of partitions up to round $n-1$, $\boldsymbol{\mathcal{P}}_{n-1}$, we aim at solving the optimization problem

$$\max_{\boldsymbol{\mathcal{S}}_n} \bar{\eta}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}) \quad \forall n. \tag{5.43}$$

Note that $\boldsymbol{\mathcal{P}}_{n-1} = (\boldsymbol{\mathcal{S}}_1, \ldots, \boldsymbol{\mathcal{S}}_{n-1})$.

As done for the single round scheduling solution, we convert the integrals over the area (for the average number of different received packets) into sums over a discrete set of points $\Omega \subset A$; thus, in this framework, the integrals over A of the performance metrics (e.g., in Section 5.5.1) need to be considered as weighted sums.

We remark that, ideally, all the points of the same tile have the same satellites in view, i.e., it should hold $v_{s,n}(\boldsymbol{x}) = v_{s,n}(\boldsymbol{x})' \; \forall s \in S$, $\forall x' \in \mathcal{A}$ for each tile $A(\boldsymbol{x})$. However, this would require to recompute sampling Ω and the tessellation at each round. In order to avoid this issue, we keep Ω fixed.

Mimicking (5.6), we introduce the indicator variables

$$y_{s,k,n} = \begin{cases} 1 & \text{if } s \in S_{k,n}, \\ 0 & \text{if } s \notin S_{k,n}, \end{cases} \tag{5.44}$$

that converts the search of the set $\boldsymbol{\mathcal{S}}_n$ into the choice of the variables $y_{s,k,n}$ and, problem (5.43) can be written as the following BILP problem

Problem 6. *At round n, given the coefficients (for $\boldsymbol{x} \in \Omega$)*

$$v_{s,n}(\boldsymbol{x}) \in \{0,1\} \quad s \in S \tag{5.45a}$$

$$u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}) \in \{0,1\} \quad k \in \{1, \ldots, K\}, \tag{5.45b}$$

$$a(\boldsymbol{x}) \in [0,1] \tag{5.45c}$$

maximize over the variables (for $k \in \{1, \ldots, K\}$)

$$y_{s,k,n} \in \{0,1\} \quad s \in S, \tag{5.45d}$$

the objective function

$$\bar{\eta}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}) = \sum_{k,\boldsymbol{x}} a(\boldsymbol{x}) u_{k,n}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}), \tag{5.45e}$$

under the following constraints $\forall k \in \{1, \ldots, K\}$, $\boldsymbol{x} \in \Omega$, $s \in S$.

$$\sum_k y_{s,k,n} = 1, \tag{5.45f}$$

$$\sum_s y_{s,k,n} v_{s,n}(\boldsymbol{x}) + u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}) \geq u_{k,n}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}). \tag{5.45g}$$

Notice that (5.45e) is exactly equation (5.26) rewritten for the fixed sampling set Ω. As for the single round case, this BILP problem can be solved through well-known approaches, e.g., the Branch and Bound technique [74, Chapter 9].

The solution to problem (5.43) is reported in Algorithm 3, where `marp_maximization` solves (6) and `round_evaluation` is a function that takes as input the set of received

Algorithm 3 Maximization of the Average Number of Different Received Packets (MARP)

```
Input: $K, \Omega$
Output: $\boldsymbol{\mathcal{P}}$
  $n \leftarrow 1$, $\bar{u}_0 \leftarrow 0$
  $u_{k,0}(\boldsymbol{\mathcal{P}}_0, \boldsymbol{x}) \leftarrow 0, \quad \forall k \in \{1, \dots, K\}, \boldsymbol{x} \in \Omega$
  while $\bar{u}_{n-1} < 1$ do
    $v_{s,n}(\boldsymbol{x}) \leftarrow$ comp_visibility_maps$(n, \Omega)$
    $\{\boldsymbol{\mathcal{P}}_n, \bar{\eta}_n\} \leftarrow$ marp_maximization$(\Omega, K, v_{s,n}(\boldsymbol{x}) u_{k,n-1}(\boldsymbol{x}))$
    $\{\bar{u}_n, u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x})\} \leftarrow$ round_evaluation$(\Omega, \boldsymbol{\mathcal{P}}, v_n(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}))$
    $n \leftarrow n + 1$
  end while
```

packets at the end of the previous round, $u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x})$, and outputs the coverage and the set of received packets at the end of round n, $u_{k,n}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x})$, by using (5.24)-(5.31).

With this approach we iterate at each round the solution of problem (5.43), thus the overall computational cost is linearly dependent to the solution of (5.43) itself. The solution of the BILP problem using Branch and Bound has a cost which is exponential in the number of constraints. More in detail, called $N_\mathrm{s} = |\Omega|$ the number of sampling points of the surface A, we have $M + KN_\mathrm{s}$ constraints. Thus the computational complexity depends on both the chosen sampling and the number of packets to be transmitted.

5.5.2.3 Maximization of the Average Number of Different Received Packets among Maximum Coverage Solutions (MARP-MC)

We now observe there may be several scheduling achieving maximum coverage with different numbers of distinct transmitted packets. Thus, we propose a scheduling algorithm that at each round n, given the set of received packet for each receiver up to the previous round, $u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x})$, first maximizes the coverage $\bar{u}_n(\boldsymbol{\mathcal{P}}_n)$ and then maximizes the average number of different received packets $\bar{\eta}_n(\boldsymbol{\mathcal{P}}_n)$, among all the scheduling solutions that achieve unitary coverage. In formulas, at round n, considering the sequence of partitions up to round $n-1$ as fixed, we solve the problem

$$\max_{\boldsymbol{\mathcal{S}}_n \in \mathcal{M}} \bar{\eta}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}), \text{with} \tag{5.46a}$$

$$\mathcal{M} = \{\boldsymbol{\mathcal{S}}_n^* = \arg\max_{\boldsymbol{\mathcal{S}}_n} \bar{u}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\})\}. \tag{5.46b}$$

Problem (5.46) is the cascade of two maximization problems. First, following the approach of the previous section, we formulate both optimizations as BILP problems and then solve them iteratively.

According to the introduced notation, the inner maximization problem in (5.46) can be written as

Problem 7 (Single round coverage maximization)**.** *At round n given the coefficients for $\boldsymbol{x} \in \Omega$*

$$v_{s,n}(\boldsymbol{x}) \in \{0,1\} \quad s \in S, \tag{5.47a}$$
$$u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}) \in \{0,1\} \quad k \in \{1,\dots,K\}, \tag{5.47b}$$
$$a(\boldsymbol{x}) \in [0,1] \tag{5.47c}$$

for $k \in \{1,\dots,K\}$ maximize over the variables

$$y_{s,k,n} \in \{0,1\} \quad i \in S, \tag{5.47d}$$

the coverage objective function

$$\bar{u}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}) = \sum_{\boldsymbol{x}\in A} a(\boldsymbol{x})\hat{u}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}), \tag{5.47e}$$

under the constraints (for all $k \in \{1,\dots,K\}$, $\boldsymbol{x} \in \Omega$, $s \in S$),

$$\sum_k y_{s,k,n} = 1, \tag{5.47f}$$
$$\sum_s y_{s,k,n} v_{s,n}(\boldsymbol{x}) + u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}) \geq u_{k,n}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}), \tag{5.47g}$$
$$\sum_k u_{k,n}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}) \geq K\hat{u}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}). \tag{5.47h}$$

This is actually a variation to Problem 1, where we focus only on the receivers that are missing some packets.

The outer maximization problem of (5.46) can be written as

Problem 8 (Single round throughput maximization with constrained coverage)**.** *At round n given the coefficients (for $\boldsymbol{x} \in \Omega$)*

$$v_{s,n}(\boldsymbol{x}) \in \{0,1\} \quad s \in S, \tag{5.48a}$$
$$u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}) \in \{0,1\} \quad k \in \{1,\dots,K\}, \tag{5.48b}$$
$$a(\boldsymbol{x}) \in [0,1], \tag{5.48c}$$
$$\bar{u}_n \in [0,1], \tag{5.48d}$$

maximize over the variables

$$y_{s,k,n} \in \{0,1\} \quad k \in \{1,\dots,K\}, s \in S, \tag{5.48e}$$

the objective function

$$\bar{\eta}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}) = \sum_{\boldsymbol{x}\in\Omega} a(\boldsymbol{x}) \sum_k u_{k,n}(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}), \tag{5.48f}$$

under the constraints (5.47f),(5.47g),(5.47h) *and*

$$\sum_{\boldsymbol{x}\in\Omega} a(\boldsymbol{x})\hat{u}_n(\{\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{\mathcal{S}}_n\}, \boldsymbol{x}) \geq \bar{u}_n \tag{5.48g}$$

The solution of Problems 7 and 8 can be obtained through well-known approaches similarly to the BILP problems in the previous sections.

Algorithm 4 Two-step approach

```
Input: $K, \Omega$
Output: $\boldsymbol{\mathcal{P}}$
  $n \leftarrow 0$, $\bar{u}_0 \leftarrow 0$
  $u_{k,0}(\boldsymbol{\mathcal{P}}_0, \boldsymbol{x}) \leftarrow 0, \quad \forall k \in \{1, \dots, K\}, \boldsymbol{x} \in \Omega$
  while $\bar{u}_n < 1$ do
    $n \leftarrow n + 1$
    $v_{s,n}(\boldsymbol{x}) \leftarrow$ comp_visibility_maps$(n, \Omega)$
    $\eta_n^* \leftarrow$ compute_marp_uBound$(v_{s,n}(\boldsymbol{x}), \Omega)$
    if is_feasible($K, v_{s,n}(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x})$) then
      $\boldsymbol{\mathcal{P}}_n \leftarrow$ solve_max_coverage$(\Omega, K, v_{s,n}(\boldsymbol{x}), u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}))$
      $\{\bar{u}_n, \bar{\eta}_n, u_{k,n}(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x})\} \leftarrow$ eval_state$(\Omega, \boldsymbol{\mathcal{P}}_{n-1}, v_n(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}))$
      if $\bar{\eta}_n \leq \eta_n^*$ AND $\bar{u}_n(\boldsymbol{\mathcal{P}}_n) \leq 1$ then
        $\boldsymbol{\mathcal{P}}_n \leftarrow$ marp_cc_maximization$(\Omega, K, v_n(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{\mathcal{P}}_{n-1}, \boldsymbol{x}), \bar{u}_n)$
        $\{\bar{u}_n, \eta_n, u_{k,n}\} \leftarrow$ round_evaluation$(\Omega, \boldsymbol{\mathcal{P}}_{n-1}, v_n(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{x}))$
      end if
    else
      $\boldsymbol{\mathcal{P}}_n \leftarrow$ marp_cc_maximization$(\Omega, K, v_n(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{x}), 0)$
      $\{\bar{u}_n, \bar{\eta}_n, u_{k,n}\} \leftarrow$ round_evaluation$(\Omega, \boldsymbol{\mathcal{P}}_{n-1}, v_n(\boldsymbol{x}), u_{k,n-1}(\boldsymbol{x}))$
    end if
  end while
```

The proposed algorithm to solve (5.46) is reported in Algorithm 4, using the following functions:

`compute_marp_uBound` computes the upperbound η_n^*, by using (5.38);

`solve_max_coverage` and `marp_cc_maximization` are respectively the optimization of the BILP Problems 7 and 8;

`is_feasible` computes the upperbound of Equation (5.38) verifying if it is possible to achieve a solution with non-null coverage (i.e., $\bar{u}_n^* > 0$). Note that if $C_n(\boldsymbol{x}) < K$ $\forall \boldsymbol{x} \in A$, i.e., all the receivers obtained less than K packets, (5.36) yields $\bar{u}_n^* = 0$, thus all the scheduling solutions will achieve zero coverage. So we can skip the coverage maximization step, saving computational power in the early rounds, where we noticed that the computations typically take longer.

The cost of this solution is similar to the MARP algorithm: in particular in the worst case scenario, we have two optimizations per round, each of them involving the solution of a NP problem. As for the MARP, the cost exponentially increases with the number of constraints: for the maximum coverage we have $M + N_s(K+1)$ with $N_s = |\Omega|$, while for the maximum coverage we get $M + N_s(K+1) + 1$ constraints.

5.5.3 Computational Complexity of the Proposed Solutions

Table 5.2 summarizes the complexity of the proposed scheduling solutions. All the problems are formulated by using ILP, therefore these are NP-complete problem with a exponential computational cost, depending on the high number of involved variables and constraints [78]. First, the MIN-MAX requires the solutions of n problems, to be solved sequentially; each problem deals with the scheduling of $K_{\min} \leq K$ messages, thus it has $M + K_{\min}N_{\text{s}}$ constraints. The MARP combines n problems, each with $M + KN$s constraints. The MARP-MC requires the solution of two problems per round, the first with $M + KN_\mathrm{s}$ constraints and the second with $M + (K+1)N_\mathrm{s} + 1$ constraints. Hence since the cost of the second is (exponentially) higher than the first, the computational cost is $\mathcal{O}\left(n \exp(M + (K+1)N_\mathrm{s} + 1)\right)$. Finally, we consider the optimal solution that computes the scheduling by considering all n the rounds altogether. For instance considering Algorithm 3, we can modify it to consider multiple rounds all at once. Hence the variables (and constraints) in the MARP solution are now multiplied by n, i.e., $\exp(nM + nKN_\mathrm{s})$ constraints. Indeed, for non trivial values of K or N_s, such solution is not computationally feasible.

Table 5.2: Complexity of the solutions for the message scheduling problem, considering n rounds, M satellites, K messages, and an area sampling factor N_s.

Solution	**Complexity**
MIN-MAX	$\mathcal{O}\left(n \exp(M + K_{\min} N_\mathrm{s})\right)$
MARP	$\mathcal{O}\left(n \exp(M + KN_\mathrm{s})\right)$
MARP-MC	$\mathcal{O}\left(n \exp(M + (K+1)N_\mathrm{s} + 1)\right)$
Optimal Solution	$\mathcal{O}\left(\exp(nM + nKN_\mathrm{s})\right)$

5.6 Results for Multi-round Scheduling

We consider the same setting to one of Section 5.4: from the Galileo GNSS constellation, S_0, with 24 MEO satellites, we did not consider the whole set S_0 but only $S \subset S_0$ with $|S| = 20$ distributed over 3 orbital planes, i.e., the subset of satellites that, at any given time, are connected to one of the 5 uplink stations [1, 75]. For Galileo, the uplink stations (ULSs) are located in Svalbard (78.2°N, 15.4°E), Kourou (5.2°N, 52.7°W), Papeete (17.5°S, 149.4°W), Sainte-Marie, Réunion (20.9°S, 55.5°E) and Nouméa (21.9°S, 166.0°E). Next, we built the S set, containing the 20 SVs closest to at least one ULSs.

Again the visibility map (5.1) is built considering a masking elevation angle $\alpha_{\min} = 10°$.

We let A be the whole Earth surface, and the sample set Ω is obtained by uniformly sampling the latitude and longitude axis respectively with $N_\lambda = 24$ and $N_\varphi = 48$ respectively. We use the same tessellation for all the rounds, described by (5.22).

Figure 5.5 illustrates the resulting tessellation and the dissemination procedure.

The orbital period of a Galileo satellite is $T_\mathrm{orb} = 14\,\mathrm{h}\,4\,\mathrm{min}$. The satellites are distributed in 3 orbital planes and equally spaced. For the numerical simulation we consider a fast message transmission that starts every 10 min and spans a time interval of $\approx T_\mathrm{orb}$. Indeed, a finer time resolution does not yield significant changes in the visibility map (5.1). The duration of a round is $T = 30\,\mathrm{s}$, equal to the duration of a Galileo sub-frame [39] or a GPS frame [79]. Of course, a much finer time granularity may be needed in more dynamic contexts, such as when scheduling the transmission for a low Earth orbit (LEO) satellite system.

In the following, we will report the performance results of the MIN-MAX, the MARP, and the MARP-MC scheduling algorithm. Additionally, as means of comparison, we will consider both the random scheduling (RS) and the (pure) carousel strategy, where all the SVs transmit the same packet sequentially. Indeed, this strategy yields $\bar{\tau} = \tau_{\max} = \tau(\boldsymbol{x}) = K$ rounds, $\forall \boldsymbol{x} \in \Omega$.

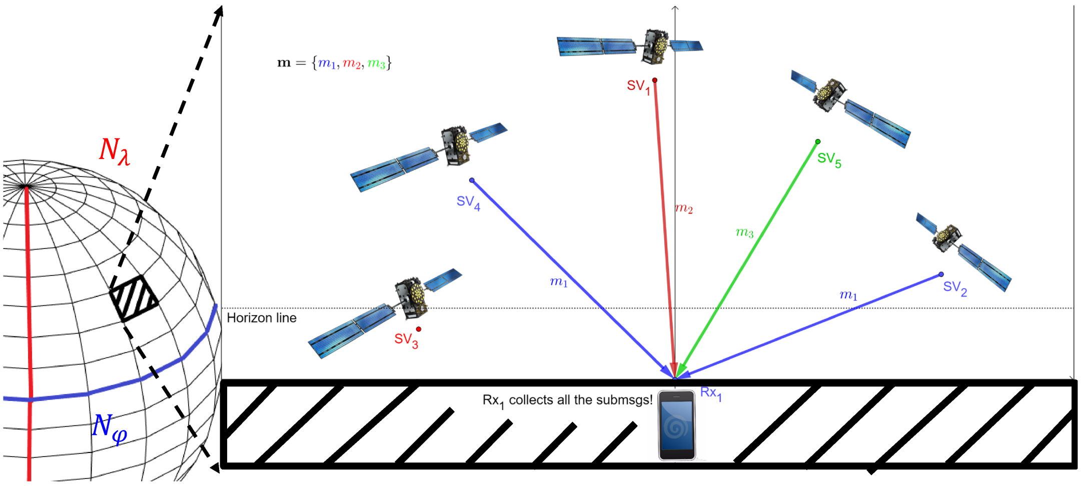


FIGURE 5.5: Pictorial representation with the resulting tessellation and the dissemination process for $K = 3$. All the satellites but SV$_3$ are considered to be in view. The receiver is able to retrieve the message since it has in view satellites transmitting K different messages.

Finally we will also test the performances of the multi-round scheduling algorithms in a more realistic scenario where each channel is modeled as an erasure channel.

5.6.1 Maximum Latency

Figure 5.6 shows the maximum latency $\max_t \tau_{\max}$, maximized over a time period T_{orb}, as a function of the number of packets per message, K, for the MIN-MAX, MARP, MARP-MC, and the random and carousel algorithms. We also show the (maximum of) the maximum latency achieved by bound (5.40). The MIN-MAX scheduling achieves the best performance for all values of K, while other solutions in general exhibit a higher average maximum latency. Moreover, the MIN-MAX scheduling always reaches the bound $\max_t[\tau^*_{\max}]$, computed from the maximum diversity scenario, which suggests the optimality of this approach in the worst-case scenario. As expected the worst performance is achieved by the carousel strategy, followed by the random scheduling solution.

5.6.2 Average Received Packets per Round

Figure 5.7 shows the average received packets per rounds $E(\bar{\eta}_n)$ as a function of the round index n, for the various scheduling algorithms, focusing on $K = 10$ and 20 packets, as average and extreme values among the considered values of K. Also in this case, the average is taken with respect to the satellite positions in the T_{orb} interval. We also report the bound on the average received packets per round $\bar{\eta}^*_n$ obtained in Section 5.5.1.

Clearly the MARP approach, which has been explicitly developed to maximize this metric, achieves the best results. Still, the MARP-MC that uses η as a secondary metric, achieves just a slightly lower value of $\bar{\eta}_n$. Here, the MIN-MAX scheduling is outperformed by all the others, including the random scheduling: this is because, by using the MIN-MAX and excluding low values of K, we do not transmit all K packets in one round, but just a subset per round (typically 4 packets). Unlikely the other solutions typically broadcast more packets per round. Moreover, these plots show that, in both cases, most of the packets are actually delivered in the first rounds; during the latter rounds, the effort is to deliver to the receivers that are still missing some of

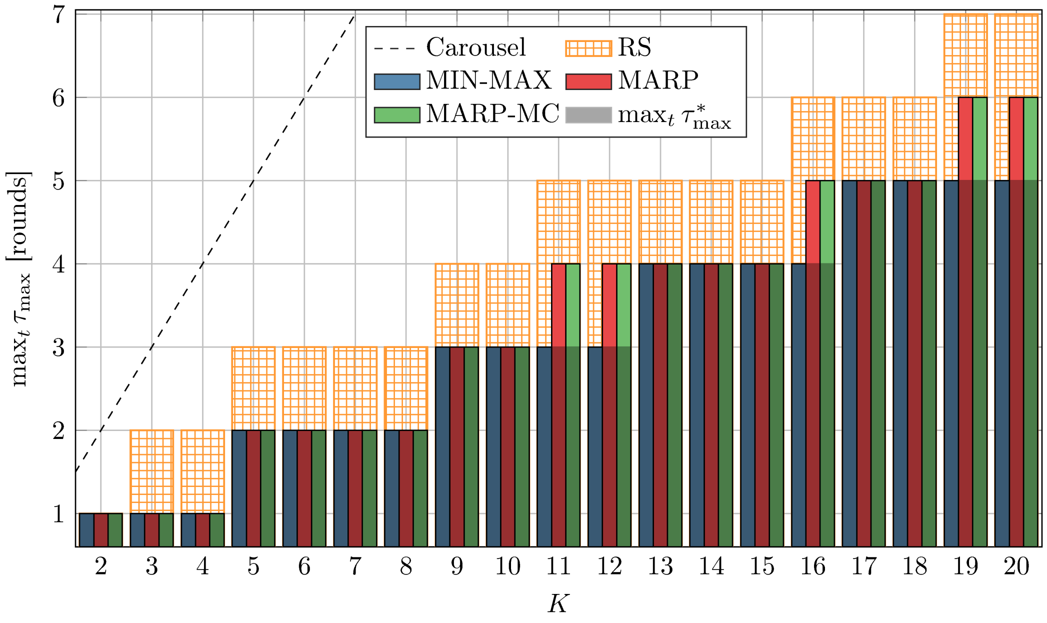


FIGURE 5.6: Maximum latency $\max_t \tau_{\max}$ for the carousel, RS, MIN-MAX, MARP, and MARP-MC scheduling algorithms, as a function of the number of scheduled packets, K. The average bound on the maximum latency obtained from (5.40) is also shown.

packets, typically the receivers that have been in low visibility conditions. We did not report the performance of the carousel strategy: this would achieve $E(\bar{\eta}) = n$, thus obtaining the worst performances among the considered algorithms.

5.6.3 Average Latency

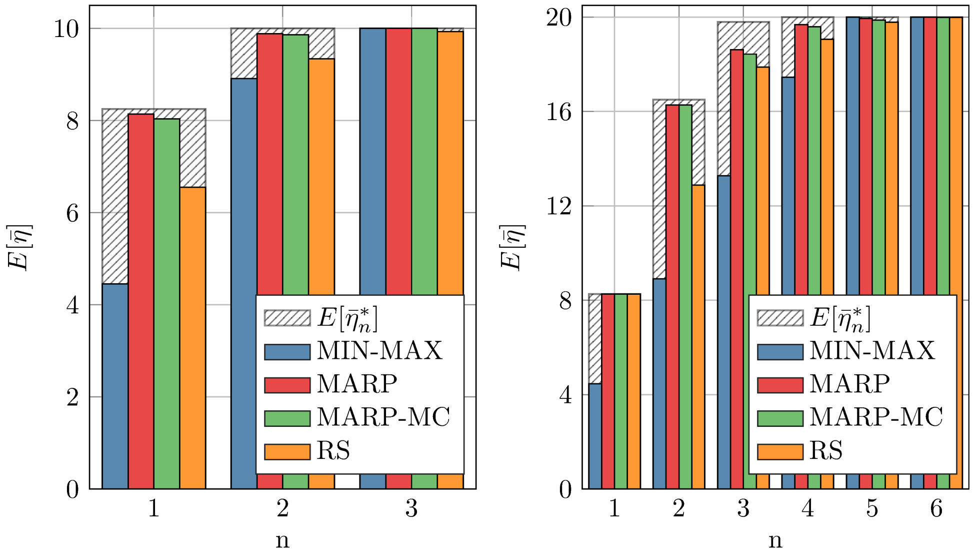
Figure 5.8 shows the average latency $E[\bar{\tau}(\boldsymbol{\mathcal{P}}]$ as a function of K, where, as described above, the average is taken with respect to the satellite positions, for the various scheduling approaches. We also report the average of the lower bound on the average latency $E[\bar{\tau}^*(\boldsymbol{\mathcal{P}})]$, as from (5.40). The behavior of the average latency for the MIN-MAX scheduling is due to the fact at each round only 4 (or 5) packets are transmitted (see Figure 5.6). For $K > 12$ all the scheduling strategies diverge from $E[\bar{\tau}^*(\boldsymbol{\mathcal{P}})]$: still, we remark that the maximum diversity scenario, described in Section 5.5.1.1 is a non-realistic condition that clearly cannot be met in practice. In this case the MARP, and MARP-MC scheduling outperforms both the MIN-MAX and the random scheduling, achieving a lower average latency. In particular, the MARP-MC scheduling achieves the lowest average latency as maximizes the number of receivers that get all the packets at each round. The (pure) carousel strategy achieves by far the worst performance, with $E[\bar{\tau}] = K$.

5.6.4 Average Throughput

In order to summarize the results we now introduce the throughput (in packets per round) as

$$\omega(\boldsymbol{\mathcal{P}}) \triangleq \frac{1}{|A|} \int_A \frac{K}{\tau(\boldsymbol{\mathcal{P}}, \boldsymbol{x})} d\boldsymbol{x}. \tag{5.49}$$

Figure 5.9 shows the average throughput $E[\omega(\boldsymbol{\mathcal{P}})]$ (5.49) as a function of K, where the average is taken with respect to the positions of the satellites over their



(A) $K = 10$

(B) $K = 20$

FIGURE 5.7: Average received packets per round $E(\bar{\eta}_n)$ vs the round index n the MIN-MAX, MARP, and MARP-MC and RS algorithms and average bound on the maximum average received packets (5.37), for $K = 10, 20$.

periods as described above, for various scheduling approaches. First, we note that the average throughput grows linearly for all the scheduling techniques up to $K = 4$ since full coverage is always achieved. Interestingly, the highest throughput is achieved by the MARP-MC scheduling $K = 7$ packets at a time, achieving a throughput $\omega \approx 6.05$ packets per round. For comparison the random scheduling solution only delivers (approximately) 3.62 packets per round.

For $K > 7$ the performance drop: indeed, by increasing K, the receivers need more rounds to obtain all the packets. A considerable fraction of the receivers has in view less than 7 SVs, thus, by having $K > 7$ we are forcing these receivers to wait for additional rounds. A similar behavior is shown also by the MIN-MAX algorithm: since we find that highest K that allows all the receivers to obtain all the packet is (typically) $K = 4$ packets, when using $K = 5$, the first round is devoted to deliver the first 4 packets; next, we use an additional round just to transmit the last packet. This also shows that, in general, increasing K does not necessarily increase the actual throughput.

5.6.5 Performance in realistic scenario

To provide realistic results, we test the proposed algorithms in a condition where we have a non-null PER. We model each channel between an SV s and a receiver in position $\boldsymbol{x}$ as an erasure channel, with a PER $q(\vartheta_s(\boldsymbol{x}))$, where $\vartheta_s(\boldsymbol{x})$ is the satellite elevation angle with respect to the receiver position at round n. The relationship between PER and elevation angle has been derived from [76]. Considering the scheduling solution $\boldsymbol{\mathcal{P}}$, the probability of having successfully received packet k within round n is

$$p_{k,n}(\boldsymbol{x}) = 1 - \prod_{m=1}^{n} \prod_{s \in S_{n,k}} q\left(\vartheta_s\left(\boldsymbol{x}\right)\right) \ . \tag{5.50}$$

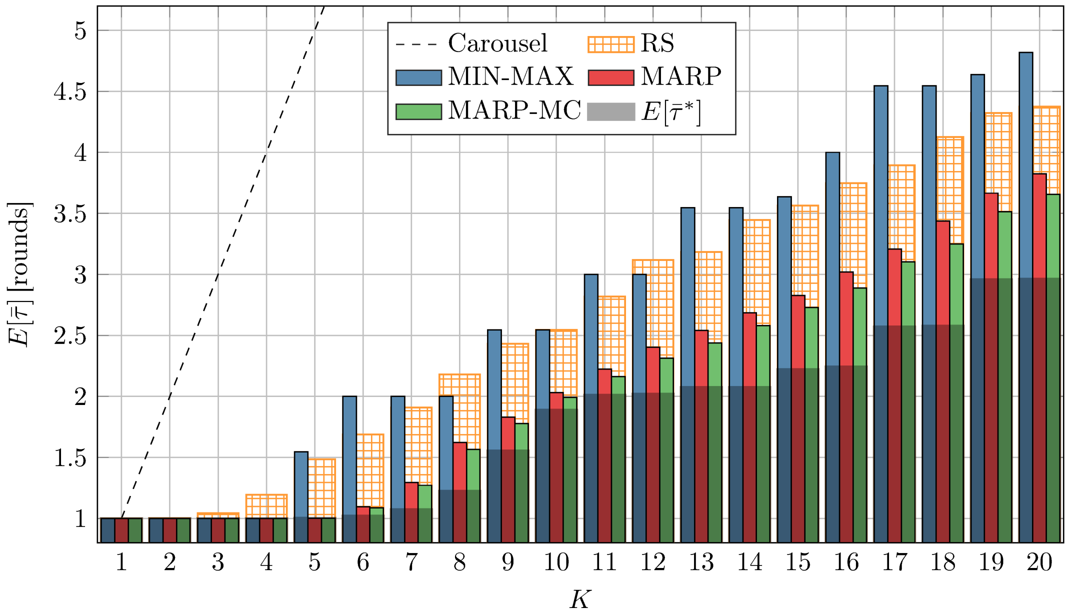


FIGURE 5.8: Average latency $E[\bar{\tau}(\boldsymbol{\mathcal{P}})]$ vs the number of scheduled packets K, where the average is taken with respect to the positions of the satellites over their periods, for various scheduling approaches, and the average of the bound on the average latency $E[\bar{\tau}^*(\boldsymbol{\mathcal{P}})]$.

Thus, the probability that the receiver in position $\boldsymbol{x}$ reconstruct the message within round n is the probability of having received all the packets,

$$\tilde{p}_n(\boldsymbol{x}) = \prod_{k=1}^{K} p_{k,n}(\boldsymbol{x}) \ . \tag{5.51}$$

Finally, we compute the fraction of receivers on A with probability at least $1 - \varepsilon$ as

$$a_n^\star = \sum_{\boldsymbol{x} \in \Omega} \mathbb{1}\left(\tilde{p}_n(\boldsymbol{x}) - (1 - \varepsilon)\right) a(\boldsymbol{x}) \ , \tag{5.52}$$

where $\mathbb{1}$ is the unit function, that is $\mathbb{1}(\cdot) = 1$ if and only if the argument is non negative. In particular, we pick $\varepsilon = 1 \cdot 10^{-2}$.

Figure 5.10 shows the period average of the area (5.52) versus the round index n, choosing $K = 5$ and $K = 15$ as example of low and high values of K, respectively. We considered the MIN-MAX, MARP, MARP-MC and the random scheduling solutions. The same considerations apply to both plots: immediately after the first rounds, where the random and the MIN-MAX achieve the best results, the MARP-MC achieve the best results. This is particularly evident for the $K = 15$ case, where at the 6th round, the MARP-MC obtains $E[a_n^\star] \approx 0.99$ while random scheduling achieves only 0.16. We did not report the performance of the carousel strategy: this yield $E[a_n^\star] = \mathbb{1}(K)$, since the last packet is not transmitted until the Kth round. For both $K = 5$ and $K = 15$, the carousel is outperformed by both MARP and MARP-MC.

5.7 Conclusions

In this paper we have have proposed a general model and solutions for the message scheduling problem in GNSS. We have analyzed two related scenarios, single-round

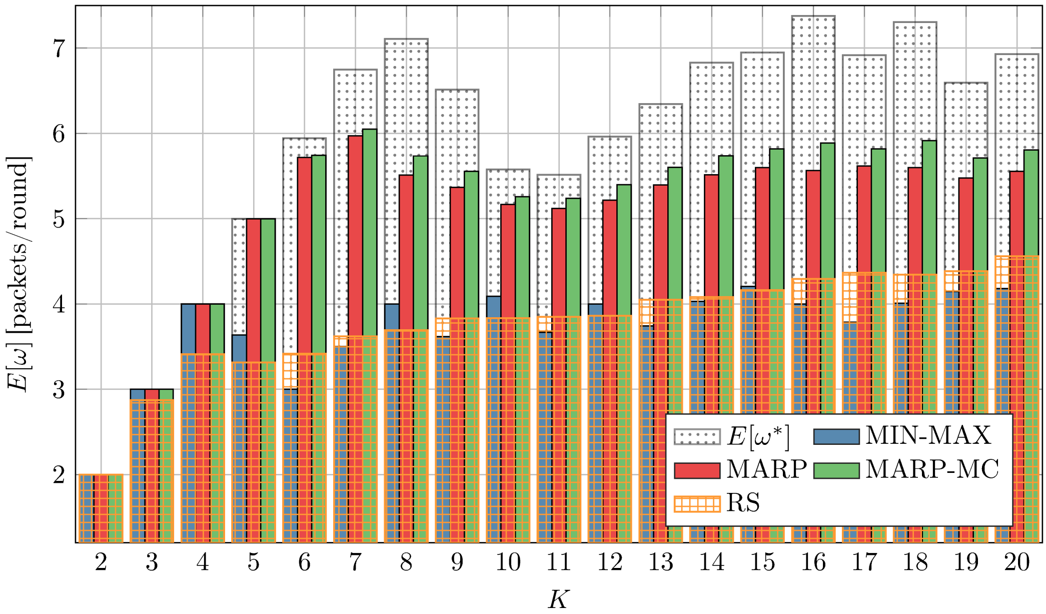
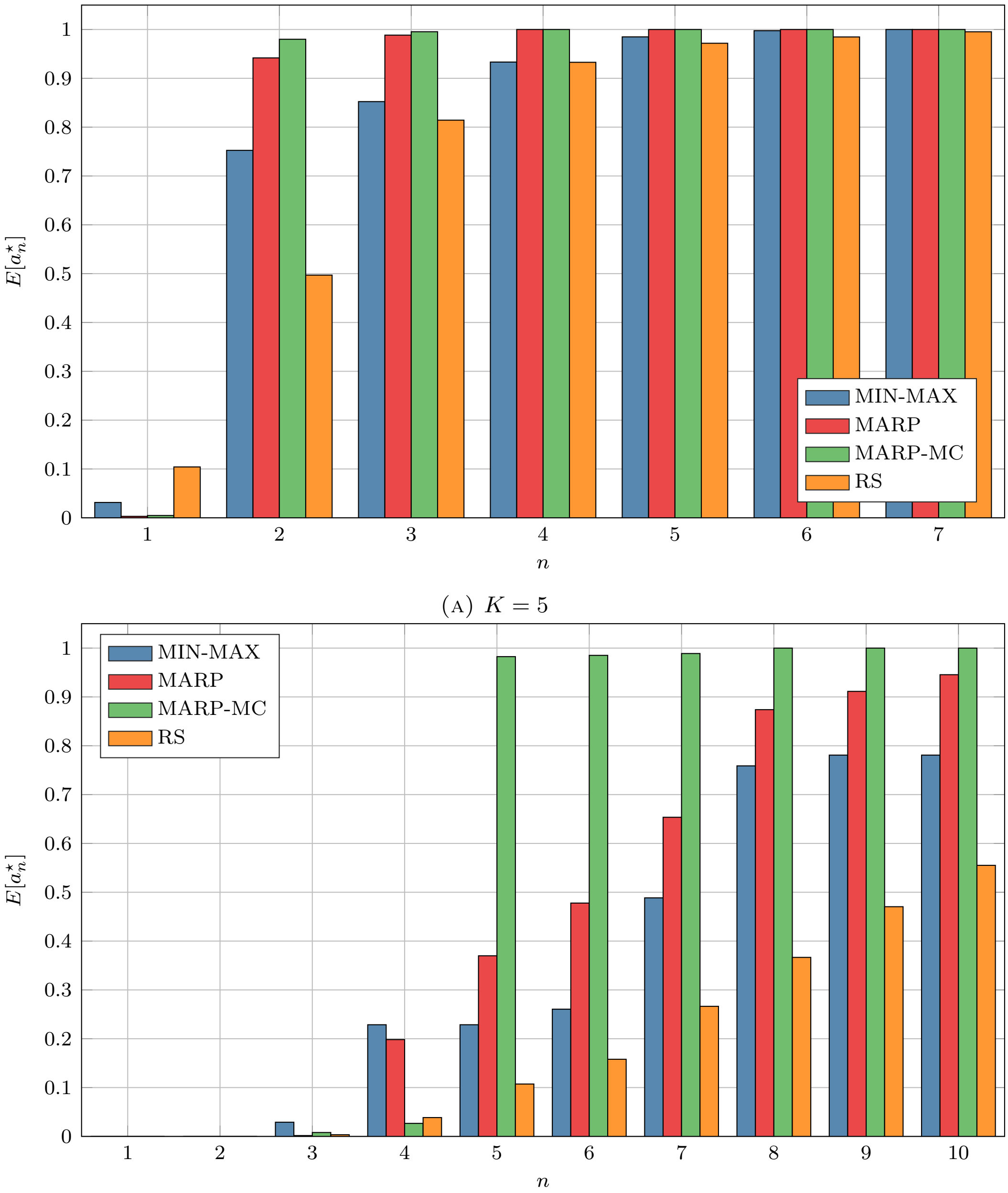


FIGURE 5.9: Average throughput $E[\omega(\boldsymbol{\mathcal{P}})]$ (5.49) vs the number of scheduled packets K, where the average is taken with respect to the positions of the satellites over their periods, for various scheduling approaches.

and multi-round message scheduling.

For the single-round scheduling we have considered the transmission over both error-free and erasure channels, deriving both optimal solutions, obtained by using ILP, and heuristic solutions. Numerical results have shown that the heuristic solution are suboptimal but still close to the ILP-based solutions. For the multi-round scheduling, we have introduced suitable metrics of the problem, and we analytically derived general lower and upper performance bounds. Next, we have proposed one solution for the maximum latency minimization that, under conditions that are easily met in practice, achieves optimality at a reduced computational cost. We then have proposed heuristic solutions for the average latency minimization, the MARP and the MARP-MC. Finally, we have showed in the numerical results Section that min-max latency algorithm achieves the optimal maximum latency, while the other algorithms achieve better performances in terms of average latency and are close to reach the optimal average latency.



(A) $K = 5$

(B) $K = 15$

Figure 5.10: Average received packets per round $E[a_n^\star]$ vs the round index n the MIN-MAX, MARP, and MARP-MC and RS algorithms, for $K = 5$ and $K = 15$.

Chapter 6

Authentication for Underwater Acoustic Networks

6.1 Introduction

UWANs are becoming a feasible option for many oceanic activities that require telemetry, communications, coordination among static and mobile devices, or the periodic monitoring of a given area. Authentication is a key security functionality in UWANs as it is in terrestrial networks [80, 81]. Through authentication mechanisms, underwater network nodes can autonomously decide whether a received message has been sent by a legitimate network member, or rather by an attacker trying to impersonate a legitimate node. However, due to the harsh propagation environment, only low data rates can be achieved: although complex signal processing algorithms are typically employed at the receiver to cope with long channel impulse responses, significant Doppler spread, as well as the interference coming from other acoustic sources [82].

With the broadening of the applications that UWANs can support and with the appearance of the first underwater communication standard JANUS [83], however, greater security concerns are starting to appear. Key types of attacks that affects UWAN vary from simple signal jamming to impersonation attacks, from attacks to routing protocols to attempts of breaking pre-agreed cryptographic keys used for data exchanges among the nodes [84, 85].

In this Chapter we present the research done for ML-based physical layer authentication for UWANs, which was published in [17–19]. We will investigate both the two-class and the one-class classification scenarios: in the first, we assume that the users have training data about both Alice and Eve channels; on the second only data from Alice is available. Even if more challenging, the latter can be considered to be more realistic since it is hard to have a priori data about the attacker. Still, considering for instance semi-static contexts, where Alice is at a fixed position and its dataset can be well characterized, Eve dataset can be simulated, e.g., by using a ray tracing simulator such as the Bellhop simulator [86, 87]. In detail a worst-case scenario will be considered, i.e., assuming that Eve is located close to Alice.

In the first part we consider the semi-static scenario, where the nodes are either static or drifting thanks to the sea current; in the latter part, we focus on a PLS authentication technique for underwater acoustic communications specifically designed to account for moving transmitters.

All these works rely on the use of ML for the robust identification of an impersonating attacker in an UWAN. However, we focus on scenarios where, due to the limited computational and communication capabilities of this sensors, the UWAN nodes are not allowed to share the whole channel observation. Thus, our methods include a local and a cooperative step. First, each node perform a pre-processing of the data collected from the channel: in [17], each sensor trains a neural network (NN) which outputs a real number representing a soft decision on the authenticity of the received packets; in [18] we considered instead a more general but complex scenario, where each sensor shares a vector instead of a number; in [19] each sensor is tracking the distance and the (relative) velocity of the transmitter by exploiting a Kalman filter, thus, it shares with the other the Kalman innovation. Finally, in the second phase, we fuse the NN outputs from all trusted nodes to finally decide on the authenticity of the transmission.

The rest of the chapter is organized as follows: Section 6.2 we the state of art; Section 6.3 describes the system model for the whole chapter; Section 6.4 presents the work for UWAC authentication, employing ML for both one and two-class authentication; Section 6.5 extends the previous results but investigates the solutions that exploits the bottleneck on the shared information; Section 6.6 a PLS technique specifically designed to account for the time variability of the channel is presented; finally, Section 6.7 draws the conclusion.

6.2 Related State of the Art

Several signaling and networking protocols have been proposed in the literature for UWAN [80,88,89],

In [90] the author propose a secure protocol suite for UWANs: as a part of this suite, the authors advocate the use of message authentication codes [91] to preserve message integrity, at the expense of increased message length. The survey in [92] proposes game theory as a mean to foster cooperation among network nodes, by motivating them to improve the effectiveness of end-to-end authentication schemes, which are seen as a key functionality of future UWANs [93]. With the aim to reduce the complexity of underwater authentication, Yuan *et al.* employ matrices of known structure as part of the encryption process, so as to reduce their memory occupancy and the computational cost of the authentication process [94]. The proposed scheme achieves up to four orders of magnitude less complexity than the standard RSA-based authentication.

With a similar purpose, Al Guqhaiman *et al.* propose a multi-factor scheme based on zero-knowledge proofs via message authentication codes [95]. Specifically, the codes depend not only on pre-shared information, but also on communication-related features such as the MAC address of the node, direction of arrival information, as well as the hop count of the sender. Receiving a packet for which this data does not match any of the features of the receiver's neighbors causes the receiver to label the packet as malicious, and to send an alert to its own neighborhood.

On the other hand, the overhead imposed by security protocols operating at the higher layers of the protocol stack may become problematic: Souza *et al.* explored the communication and computation energy toll that terrestrial network authentication primitives based on cryptography may take if directly applied to underwater network nodes for end-to-end authentication [88].

Zhang *et al.*'s approach [96] revolves around classical authentication schemes based on message exchanges, and mandates the use of lightweight primitives such as chaotic maps and hash functions. While being slightly lighter than competing schemes from the literature from a computational point of view, the proposed scheme requires less storage to work. Along the same line, in [97] the attacker is able to impersonate multiple network nodes at once (also known as a Sybil attack). Here, the legitimate nodes attempt to identify its malicious behavior via its node id and the data stored in the cluster head, which feeds a hierarchical fuzzy system-based trust management model

In this chapter we proposed instead PLS based authentication protocols. The general differences with respect to cryptographic approaches have been highlighted in Chapter 1. Still, we remark that the two approaches are not mutually exclusive since it may be possible to use crypto-based mechanism on top of physical layer schemes. To this end, a recent trend in the underwater security field explores the fundamental characteristics of the UWAC to secure underwater communications.

Among the first examples of underwater physical layer security, Kulhandjian *et al.* exploit jamming to disturb unwanted receptions at an eavesdropper, while still allowing communications between a pair of legitimate transceivers [98].

In [99] the authors considered a scenario where there is a UWAN composed by trusted nodes that cooperate to distinguish the channel footprint of the legitimate and impersonating nodes, by computing one belief value each, based the statistics of their channel features; then each local value are fused by a sink node to make a decision. While we will consider the same set of features, considering also the large complexity and variety of scenarios for UWAN [82], the models obtained by estimation may be mismatched thus affecting the outcome of the process. Conversely, the protocols discussed in this chapter are able to successfully distinguish between legitimate and impersonating attackers without the complexity of the scheme proposed in [99].

Related to this, in [100, 101] the authors investigated the use of a different set of channel features with the aim of evaluating which features remain coherent over time while becoming uncorrelated already over short distances. These features would then be used for secret key generation. A UWAC based secret key generation protocol is also reported in [102].

In [103], the authors propose to authenticate nodes based on a single feature, the maximum time-reversal resonating strength, which measures how well a received channel impulse response matches those of previous transmissions, stored in a pre-collected database. The authentication mechanism is then based on a Neyman-Pearson likelihood ratio test (LRT). The authentication scheme of [104] provides a single receiver that exploits reinforcement learning to choose the authentication parameter without being aware of the network and spoofing model. Still, a single receiver may not provide an accurate authentication process. Therefore, solutions based on multiple devices have been explored.

Considering an underwater LOS environment with negligible multipath, Khalid *et al.* propose that the receiver keep a database of angles of arrival for legitimate transmissions from a given node [105]. In this way, the receiver can detect an attacker by comparing the angle of arrival of its transmissions against the distribution of previously collected angles of arrival. However, the work does not consider the case of a more powerful attacker that can craft transmitted signals to change the estimated angle of arrival at the receiver.

Aman *et al.* evaluate the capacity of underwater channels under impersonation attacks [106], assuming that the legitimate receiver uses distance as a feature to discriminate between a legitimate and an impersonating transmitter. After modeling the dynamics of the communications as a Markov chain, the authors employ numerical optimization to find the optimum transmission rate for the legitimate transmitter and show that a small neural network reproduces the optimization process well.

6.3 System Model

We consider an UWAN composed of a legitimate transmitter, namely Alice, a set of N trusted sensors $\mathcal{S} = \{S_n, n = 1, \dots, N\}$ and a node Bob. Bob and the network of

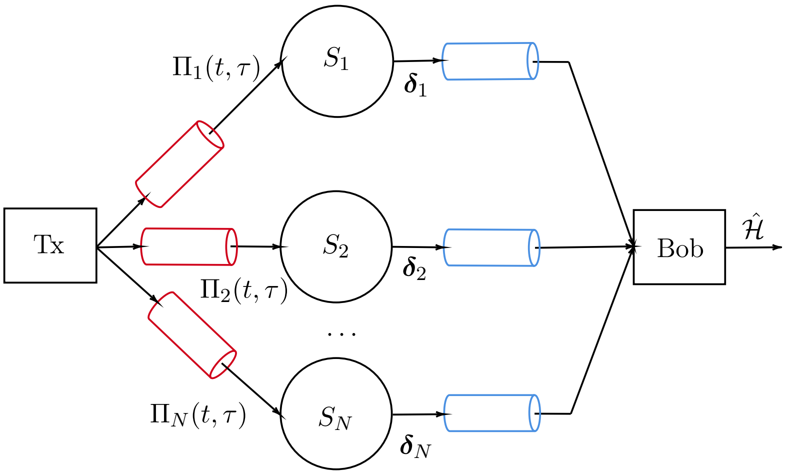


Figure 6.1: Signal model of the proposed UWAN authentication scheme: a transmitter, Tx (i.e., Alice or Eve), open channels (in red), sensors, authenticated channels (blue), and Bob.

sensors cooperate to decide whether the packet was transmitted by Alice or by an attacker, namely Eve. In turn, Eve transmits packets to the sensors in an attempt to impersonate Alice, i.e., aiming at having Bob accepting her packets as coming from Alice.

We assume that all nodes are loosely synchronized [72, 107, 108]), and that each packet has a unique sequential identification number (ID). This allows different sensors to perform a distributed cooperative check by observing the same broadcast packet. The communications occur over UWACs: the transmissions from the sensors to Bob are performed over authenticated channels, i.e., Eve cannot transmit signals over these channels. We also assume that the sensors and Bob employ proper error detection and correction protocols, e.g., CRC and automatic repeat request (ARQ), such that no communication error occurs in the data reception. Moreover, Eve does not modify the transmit signal to specifically break authentication (more sophisticated attacks are left for future study).

The exact location of the different receiver nodes is unknown to both the trusted receivers and Eve.

We assume that all trusted receivers are connected to a sink node via a limited-rate, authenticated, and integrity-protected channel, over which they can share their observations. Then, the sink makes the final decision on the authenticity of the received packets. We make a first decision at each node to avoid transmitting each single observation to the sink. This reduces the communication overhead. We model each point-to-point UWAC as a tapped delay line, having power-delay profile $H'_n(t,\tau)$ at time t.

The attacker Eve is a single malicious node. However, our scheme can be straightforwardly extended for multiple attackers. We also make no assumption on the contents of the packets, i.e., we assume that the packets sent by Alice and Eve are indistinguishable at the data level.

6.3.1 Features for Authentication on UWACs

To assess the authenticity of the received packet, we rely this set of channel statistics. Let $z_{j,n}(t)$ be the measured value of the jth feature with $j = 1, \ldots, N_\mathrm{f}$ measured at time t by node B_n. To extract the features, we zero out low-power arrivals in the power-delay profile, i.e.,

$$\Pi_n(t,\tau) = \begin{cases} 0 & |\Pi'_n(t,\tau)| < T_h, \\ \Pi'_n(t,\tau) & |\Pi'_n(t,\tau)| \geq T_h, \end{cases} \tag{6.1}$$

where we choose T_h to obtain a desired FA probability when discriminating between (true) taps and noise contribution, as detailed in [109]. Call $\mathcal{H}_n(t)$ the set of delays of all channel arrivals that remain after thresholding.

We consider the following features:

1–Number of channel taps. The estimated number of relevant taps revealing the spread of the acoustic channel:

$$z_{1,n}(t) = |\mathcal{H}_n(t)| \, . \tag{6.2}$$

2–Average tap power. The average power of the relevant taps, which reflects how diverse and sparse the channel is:

$$z_{2,n}(t) = \frac{1}{|\mathcal{H}_n(t)|} \sum_{\tau \in \mathcal{H}_n(t)} |\Pi_n(t,\tau)| \, . \tag{6.3}$$

3–Relative root mean square (RMS) delay. This feature reflects the delay spread of the channel. Let $\tau_0 = \min\{\tau : \tau \in \mathcal{H}_n(t)\}$ be the delay of the first tap, then the relative RMS delay is

$$z_{3,n}(t) = \left(\frac{1}{|\mathcal{H}_n(t)| - 1} \sum_{\tau \in \mathcal{H}_n(t), \tau \neq \tau_0} (\tau - \tau_0)^2 \right)^{1/2} . \tag{6.4}$$

4–Smoothed received power. This feature accounts for the overall attenuation in the channel. To track the variation of power over time, let $q_{n,t}$ be the power of a symbol received by node n at time instance t. Given a user-defined parameter $0 \leq \zeta \leq 1$, we recursively compute the smoothed received power as

$$z_{4,n}(t) = \zeta \, q_{n,t} + (1 - \zeta) \, z_{4,n}(t') \, , \tag{6.5}$$

where $z_{4,n}(t')$ is the smoothed received power of the previous symbol received at time t'. In our case we will pick $\zeta = 0.5$.

We choose these features, because their statistics are stable over time and depend strongly on the transmitter's location [99]. We use the estimated statistics for authentication purposes: therefore, by comparing the channel's features, sensor S_n can distinguish between packets arriving from sources located at different locations.

We will use these features for the protocols described in Sections 6.4 and 6.5; in 6.6, we will consider instead an ad-hoc feature, used for tracking the transmitter movement relative to the receiver.

6.4 Authentication of UWACs via ML

In this Section we will describe the first protocol we proposed for authentication in UWAN. In particular we will focus on the case where the local output is a (soft) scalar value, $\delta_n \in \mathbb{R}$. In general there is no reliable statistical model of the UWAC, therefore we cannot use any solution that relies on the probability density function (PDF) of the features. Thus, we consider data driven approaches, turning to NNs to distinguish between a legitimate transmission and a fake one. We consider two alternative scenarios, depending on the data available to train the NN. In the first, we consider two-class classification, where each sensor s_n has observations available from both Alice and Eve; instead, in the second scenario, we consider one-class classification, where each sensor has observations only from Alice, as would be the case if the statistics of Eve's channel features are unknown. In this latter case we resort to autoencoders (AEs).

As described in Section 6.1 we will consider separately the local and the cooperative steps.

We test our scheme both on simulated channels and on data from a sea experiment carried out in the eastern Mediterranean sea near Hadera, Israel. Our results confirm that our proposed scheme successfully distinguishes between authentic and impersonating transmissions. In particular, in our simulations we successfully discriminate between the legitimate transmitter and the attacker even when they are located close to each other, albeit at different depths.

Results for the sea experiment support the same conclusion in a realistic environment. Here, we show that a few hundreds meters between the legitimate transmitter and the attacker are sufficient to tell the two nodes apart, even when relying on a single trusted receiver.

6.4.1 Local Authentication Strategies

Neyman-Pearson (NP) Test We start our analysis considering the Neyman-Pearson (NP) test, based on the LRT. In particular it is proven that 1) the NP test is optimal [110], i.e. it minimizes the p_{MD} for a fixed p_{FA}. Let $p(\boldsymbol{z}|\mathcal{H}_i)$ be the PDF of observation $\boldsymbol{z}$ given that $\varphi \in \mathcal{H}_i$. We compute the (log) LRT as

$$\mathrm{LRT}(\boldsymbol{z}_n) = \log \frac{p(\boldsymbol{z}_n|\mathcal{H}_0)}{p(\boldsymbol{z}_n|\mathcal{H}_1)}, \tag{6.6}$$

and compare it to a threshold λ to obtain the NP test

$$f\left(\mathrm{LRT}(\boldsymbol{z}_n), \lambda\right) = \begin{cases} 1 & \text{if } \log p(\boldsymbol{z}_n|\mathcal{H}_0) - \log p(\boldsymbol{z}_n|\mathcal{H}_1) > \lambda, \\ -1 & \text{if } \log p(\boldsymbol{z}_n|\mathcal{H}_0) - \log p(\boldsymbol{z}_n|\mathcal{H}_1) \leq \lambda \end{cases}, \tag{6.7}$$

where λ is chosen a priori depending on a target P_{FA} value. Notice that by increasing λ we reduce P_{FA} and increase P_{MD}; vice-versa, decreasing λ reduces P_{MD} while increasing P_{FA}. Thus, sensor S_n will share the local output $\delta_n = \mathcal{M}(\boldsymbol{z}_n)$. However, as stated in the previous Sections, we do not have a general statistical model for the UWAC, hence we cannot analytically derive $p(\boldsymbol{x}|\mathcal{H}_i)$ as needed for the NP test. However, to compare the performance of our solutions, we will infer $p(\boldsymbol{x}|\mathcal{H}_i)$ by estimating it directly from our data set.

We remark, that while the NP testing is indeed optimal for the local authentication, we have no guarantee of the optimality of the concatenation of NP tests, i.e., we do not know if by performing NP tests both on the sensors and on the sink node allow to achieves the overall optimum performance.

Local NN-Based Authentication - Local Decision We consider now a NN-based approach, that we call local decision (LD) strategy.

We model the NN as a function $f_{\mathrm{NN},n} : \mathbb{R}^4 \longrightarrow \mathbb{R}$. Still it is actually composed of Q stages, typically called *layers*: layer 0 is the *input layer*, the last stage the *output layer* and the remaining layers *hidden layers*. The first layer input is mapped to a vector $\boldsymbol{y}^{(0)}$, whereas the output of the output layer is $y^{(Q)}$. We represent the output of the kth neuron of the qth layer as $y_k^{(q+1)} = \psi^{(q)}\big(\boldsymbol{w}_k^{(q)}\boldsymbol{y}^{(q)} + b_k^{(q)}\big)$, where $\psi^{(q)}(\cdot)$ is the neuron *activation function*, $\boldsymbol{y}^{(q)}$ is the output of the previous layer q, $b_k^{(q)}$ is a bias value and $\boldsymbol{w}_n^{(q)}$ is a vector of weights. We consider only feedforward NNs [111] with no loops between the layers. While the activation functions are decided a priori, weights $\boldsymbol{w}_k^{(q)}$ are optimized during the algorithm's learning phase. Given the labeled training dataset, with labels $T(\boldsymbol{z}_n) = 1$ if $\boldsymbol{z}_n \in \mathcal{H}_0$ and $T(\boldsymbol{z}_n) = -1$ otherwise, during the *learning phase*, function $f_{\mathrm{NN},n}$, i.e. the weights and the biases of the NN, are optimized to minimize a training loss. Typical training loss for classification task are the cross entropy and the MSE (with respect to the original training tag). Finally during the *testing phase* we evaulate the performance of the trained function. A more detailed explanation of NN design can be found in [111].

Notice that a sufficiently complex NN trained with a sufficiently big training dataset, asymptotically converges to the NP test [112]. In other terms, under these hypotheses and fixed the FA, using the test function (6.7) on the NN output is the same of applying the same test function on the LRT, in terms of MD. Thus we choose as value to be transmitted to Bob the output of the NN itself, i.e., $\delta_n = f_{\mathrm{nn,n}}(\boldsymbol{z}_n)$.

Local AE-based Authentication An AE is an unsupervised NN trained to replicate its input at the output [111, 113]. An AE is composed of an *encoder*, a hidden layer with $M < N$ nodes, and a *decoder*. The task of the encoder $f_{\mathrm{enc},n}(\boldsymbol{z}_n)$, composed of Q_e layers, is to project the input vector $\boldsymbol{z}_n$ into a lower dimensional space of size M. The task of the decoder, $f_{\mathrm{dec},n}(\cdot)$ is to retrieve the original input vector from the encoded word $\hat{\boldsymbol{z}}_n = f_{\mathrm{enc},n}(\boldsymbol{z}_n)$. Notice that the reconstruction process is not perfect, and depends on the size of the training dataset and of the hidden layer. While a larger hidden layer eases the reconstruction of the input, a smaller hidden layer enables a more accurate characterization of the training set. In this sense a smaller values of M forces the AE to learn the useful statistical properties of the training dataset.

For this reason we use the AE for one-class classification: since the AE is trained only by using legitimate data, the NN is supposed to reconstruct with low reconstruction error only the input from Alice, yielding a significant reconstruction error when the input is a set of features from a Eve's transmission. Thus the reconstruction error itself can be used as test statistic for one class classification [112, 114].

Formally, the reconstruction error is the MSE

$$\Gamma_n(\boldsymbol{z}_n) = \|\boldsymbol{z}_n - f_{\mathrm{dec},n}(f_{\mathrm{enc},n}(\boldsymbol{z}_n))\|^2 . \tag{6.8}$$

For one class classification we consider as transmitted value $\delta_n = \Gamma_n(\boldsymbol{z}_n)$.

6.4.2 Cooperative Authentication Strategies

We consider the two-step authentication protocol of Figure 6.1, where each node S_n runs a single-node authentication protocol and transmits to the sink node either value δ_n. As done for the local step, we distinguish one-class from two-class classification scenario.

Two-class Scenario In the two class classification scenario Bob trains NN. Clearly, it is not feasible to use the LRT since in general, the PDF $p(\boldsymbol{\delta})$ is not known. A possible alternative would be to estimate the PDF from the training dataset. But this is either equivalent (or even worse is the training dataset is not sufficient) to using the NN for classification. Thus, we do not consider the case where Bob uses a LRT.

One-class Scenario For the one-classification scenario, we consider instead two solutions: the first is to employ an autoencoder (AE) as done in the local step; the alternative is to fuse by using the linear combination (LC) approach, i.e., by linearly combining each local output as

$$g([\delta_1, \ldots, \delta_N]) = \frac{1}{N} \sum_n \alpha_n \delta_n \,. \tag{6.9}$$

In [99] the weights α_n are decided by taking into account both the distances between each sensor and (estimated) distance between transmitter and the sensor. However we are considering an harsher scenario, where Bob has no reliable information about the sensors position: thus we will consider $\alpha_n = 1 \; \forall n = 1, \ldots, N$.

In both scenarios the output will be used as input for the test function 6.7, tuning the threshold to match the predefined FA. The result of this test is then broadcast by Bob.

6.4.3 Simulation Results

We consider the channel model described in [87]: in particular we model our UWAN composed

- 3 sensors, namely S_1, S_2, and S_3, placed at different depths;
- Alice, located at a depth of 20 m;
- Eve, located close to Alice at depth of 480 m.

Figure 6.2 shows an example of the simulated scenario: for simplicity we did not include the Bob in the figure, assuming that any node may act as him.

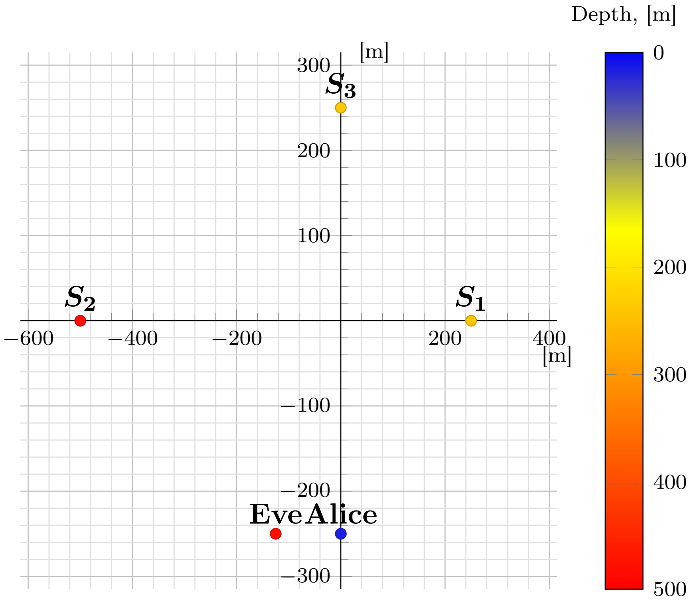


FIGURE 6.2: Example of the simulated scenario.

To generate the data set, we simulate the communication in the UWAC using the Bellhop simulator and the Acoustic Toolbox [86, 87]. In particular to simulate the channel variability, for each node, we pick 500 different position uniformly at random within a sphere of radius 10 m centered on each node's nominal location. For each pair of nodes, this yields 500 × 500 transmitter-receiver pairs. We ran Bellhop for each pair and obtained a realization of the UWAC. We considered a rough sea surface and modeled the sea bottom with hills of sinusoidal shape, with diameter of 200 m and maximum height of 10 m. To model the sound speed profiles (SSP), i.e., the sound speed as function of the depth, we considered the dataset available at [115].

6.4.3.1 Results For the Local Authentication

We start considering the local authentication at each node. In particular as stated in Section 6.4.1, we use the NP test as mean of comparison. However, we do not have the PDFs $p(\boldsymbol{z}_n|\mathcal{H}_i)$ needed to compute the LRT of (6.6). Instead, we estimate them from the channel realizations via kernel density estimation (KDE) [116]. Implicitly, as in [99], we also assume that the features are independent, thus we will compute the $p(\boldsymbol{z}_n|\mathcal{H}_i)$ as the product of each feature (estimated) PDF.

The NN used for the two class classification task is designed as following:

- 4 nodes on the input layer with the rectified linear unit (ReLu) as activation function,
- 2 hidden layers composed of 3 nodes each with the ReLu activation function,
- one node on the output layer with the sigmoid activation function.

The AE used for one-class classification is composed of

- 4 nodes for the input layer, i.e., the encoder, with the ReLu,
- a single hidden layer with $M = 3$ nodes, with the ReLu activation function,
- 4 nodes for the output layer, i.e., the decoder with a linear activation function.

In particular we picked $M = 3$ instead of $M = 1$ or 2 because it achieves the best performance in terms of reconstruction error. For both NN and AE we use 60% of the generated dataset for training, 15% for validation, and 25% for testing.

In our context metrics such as the accuracy, typically used to evaluate the performance of NN trained for classification task, are not particularly relevant; we focus instead on ROC curves, i.e., the value of FA and MD probabilities, for different values of the threshold λ.

Figure 6.3 shows the obtained ROC curve: we observe that, even if Alice and Eve are close to each other, both NN and AE achieve good results. We also observe that the NN outperforms both the AE and the NP test. Indeed, differently from the AE, the NN is trained using also data from Eve UWAC realizations. Moreover, we use estimated PDFs in the NP test and their mismatch with respect to the features' true statistics negatively affects the test performance. Of course if we had at disposal the true PDF $p(\boldsymbol{z}_n|\mathcal{H}_i)$, the results would improve the results of the NP test: in particular we would expect NP and (properly trained) NN to perform the same [112].

6.4.3.2 Cooperative UWAC Authentication

In this section, we report results using the cooperative authentication strategies described in Section 6.4.2.

Figure 6.4 shows the ROC at the sink node for the two-class classification scenario, i.e., considering that all the nodes have access to both Alice and Eve realizations for training, and use a NN for the single node authentication check. Fusion at the Bob is performed with either

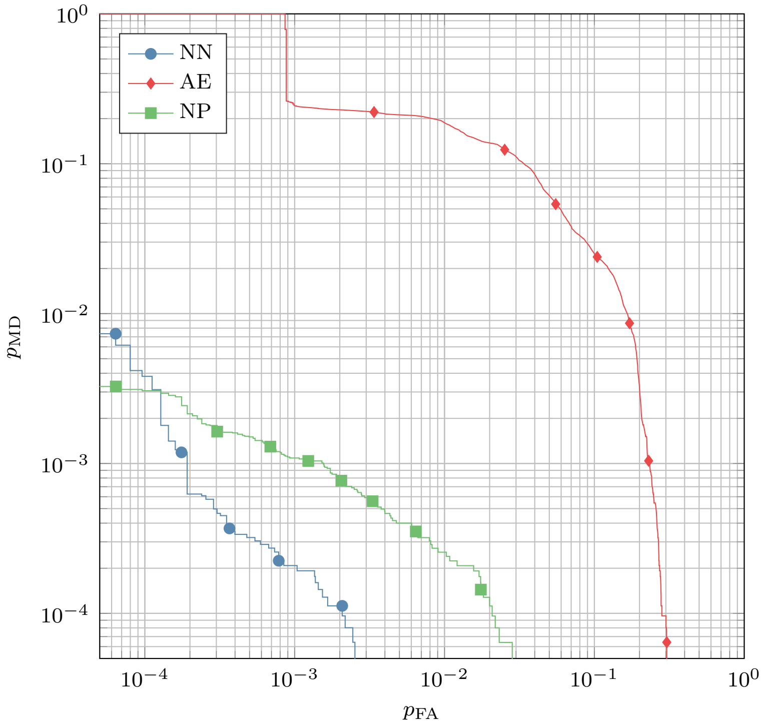


FIGURE 6.3: ROC curves for the simulated scenario obtained using NN, AE, and NP test.

1. a global NN with a design analogous to the local NN, but having 3 nodes on input layer and a single 2-node hidden layer, or

2. the LC strategy of (6.9), with no knowledge about the sensors positions, i.e., $\alpha_n = 1$, $\forall n = 1, \ldots, N$.

The results are compared to the local authentication check, i.e., when each node use the test function (6.7) on its own δ_n.

Both cooperative checks outperform the local authentication: in particular, the NN makes the data set separable, i.e., there exists a value of λ that provides $p_{\mathrm{FA}} = p_{\mathrm{MD}} = 0$, thus no line is reported in the log-scale ROC figure. Note also that the results of the faster and simpler LC, that is still able to coherently fuse that local data, achieving $p_{\mathrm{FA}} = p_{\mathrm{MD}} \approx 10^{-4}$.

Figure 6.5 shows the ROC obtained for the one-class classification scenario, i.e., by using only observations from Alice for training. In this setting each sensors use the AE, while Bob fuse the data by using either

1. a second AE with the same design as the first one but one less node on both input and hidden layer, or

2. the LC strategy of (6.9), with no knowledge about the sensors positions.

Performance are compared to the local authentication check performed by each node S_n by using the test function (6.7) on the reconstruction error $\Gamma(\boldsymbol{z}_n)$. Interestingly the first strategy seems to be only partially effective, i.e., only sensors S_1 and S_3 would have a better performance, with respect to the local check. Conversely the simpler LC method is shown to be effective, i.e., it provides lower probabilities of MD and FA than those of node in the best position, taking advantage of the less reliable nodes.

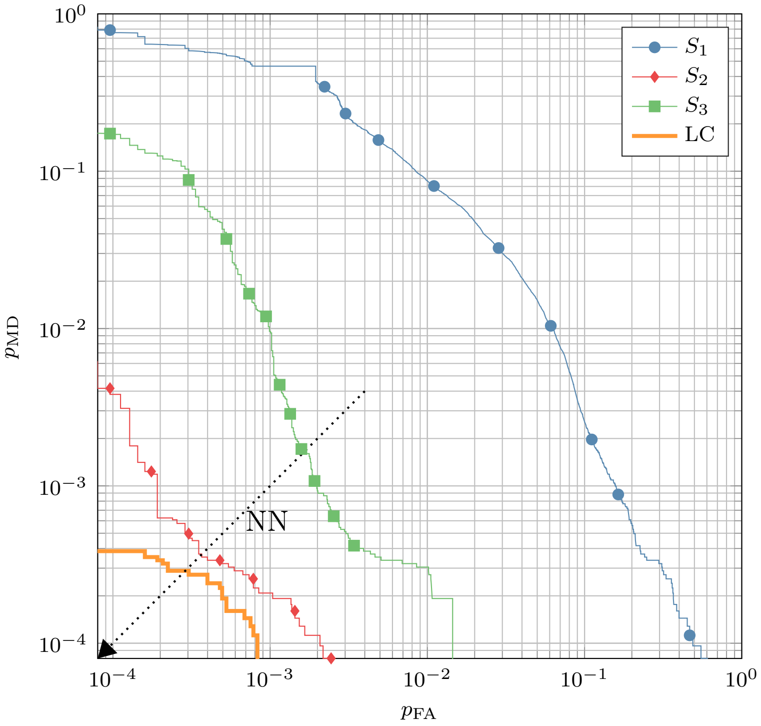


FIGURE 6.4: ROC curves in the simulated scenario, for the two-class setting case. Local authentication checks versus LC and NN-based cooperative checks.

6.4.4 Experimental Results

To demonstrate the performance of our authentication protocol in a realistic environment, we repeated the training and evaluation process of the NN also for a data set obtained from a sea experiment. The experiment was conducted near the Hadera coal pier in Israel in May 2017, with the setup shown in Figure 6.6. In details, we used

- two projectors, Tx1 and Tx2, acting as Alice and Eve, respectively. Tx1 was deployed from the pier, while Tx2 was deployed from a boat. The distance between Tx1 and Tx2 was roughly 1 km;
- three receivers, Rx1, Rx2, and Rx3. Rx1 and Rx3 were deployed from two floating buoys, while Rx2 was deployed from a boat. The distance between each receiver was approximately 500 m.

Tx1 and Tx2 mounted EvoLogics software-defined S2CR 7/17 modems, and transmitted packets composed of 100 chirp symbols of duration 10 ms in the 7-17 kHz band. The source level was roughly 175 dB re 1 μPa@1m. The receivers used a Cetacean CR1 hydrophone and continuously recorded the raw acoustic data. The data set collects the measurements acquired by Rx2. To process the experiment data, we used the same NN and AE design used for the simulated UWAC. We remark that the dataset was limited thus we cannot consider this as a proper performance evaluation. Still this can can be considered as a validation that gives proper fundation to the results obtained from the Bellhop simulator.

Using the NN, considering the output of Bob NN, $f_{\mathrm{nn,Bob}}(\boldsymbol{\delta})$, for the experiments we always have

$$\begin{cases} f_{\mathrm{nn,Bob}}(\boldsymbol{\delta}) \geq 0.9 & \text{if } \varphi \in \mathcal{H}_0, \\ f_{\mathrm{nn,Bob}}(\boldsymbol{\delta}) < 0.9 & \text{if } \varphi \in \mathcal{H}_1 \end{cases} .$$

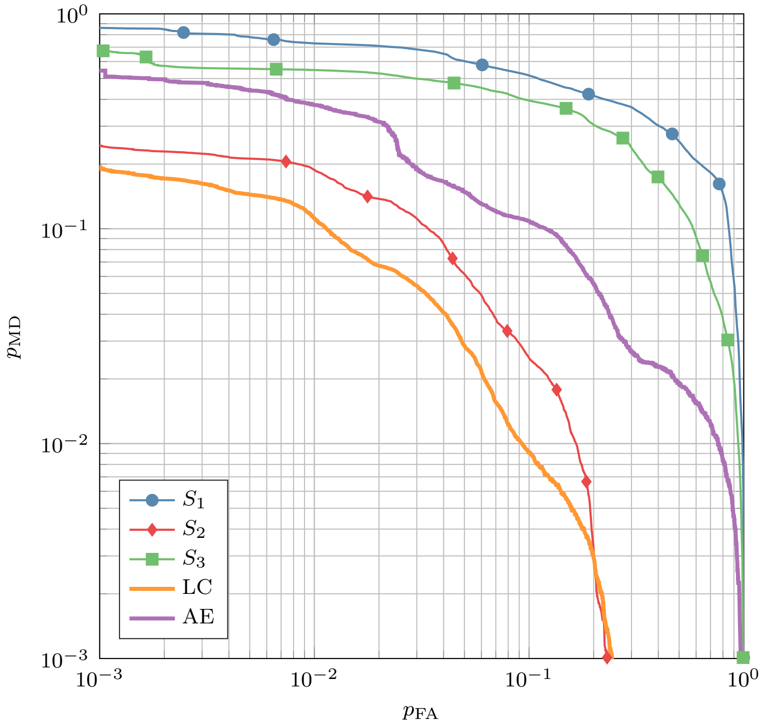


FIGURE 6.5: ROC curves in the simulated scenario, for the one-class setting case. Local authentication checks versus LC and AE-based cooperative checks.

Since the two distributions are separable by a threshold, it is possible to find values of λ (e.g., $\lambda = 0.9$) such that $p_{\text{FA}} = p_{\text{MD}} = 0$.

Equivalently, using instead the AE we get at Bob

$$\begin{cases} \Gamma(\boldsymbol{\delta}) < 0.1 & \text{if } \varphi \in \mathcal{H}_0, \\ \Gamma(\boldsymbol{\delta}) \geq 0.1 & \text{if } \varphi \in \mathcal{H}_1. \end{cases}$$

hence also these distributions are separable after the AE so the Bob is able to perfectly distinguish between Alice and Eve packets.

6.5 ML Distributed Authentication of UWAN Nodes with Limited Shared Information

Considering a bottleneck on the channel between each sensors and Bob, the aim of this Section is to investigate and develop solutions that take full advantage of the bottleneck, sharing only the useful information to Bob. The solution discussed in [17] and presented in the previous Section, considers the case where each node S_n only reports a single soft value per packet: in such case it is reasonable to share the local (soft) decision. We consider now a more broad scenario where each node shares instead a vector. Indeed, passing more values (still compressed with respect to all the observed features) it expected to yield a more accurate final decision. Although the purpose of the system is to take a binary decision (whether a packet is authentic or not) we show the importance of having a rich set of compressed features, still taking into account the transmission rate limits among the sensors and the central nodes. Clearly, the best result would be obtained by globally designing each local function, taking advantage of

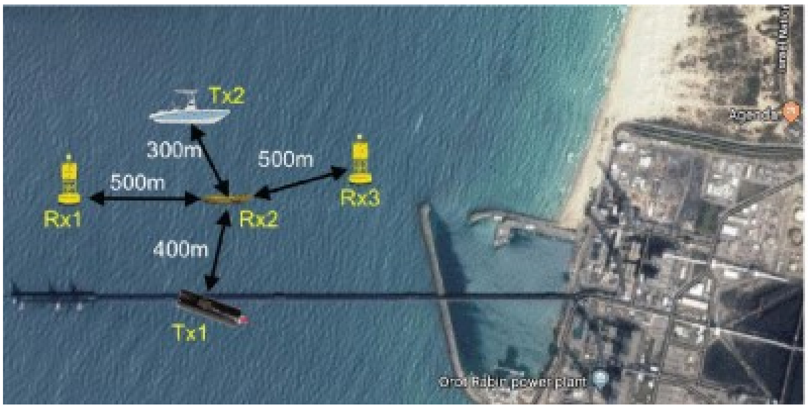


FIGURE 6.6: Setup of the sea experiment in Hadera, Israel.

the local correlation among each node: for instance, we could reduce the rate of nodes placed close to each other in favor of dislocated ones. We consider instead the case where each node has no information about the others and cannot jointly design the local function. In the next Sections, we analyze several possible strategies for localized training comparing them with the global training where all NNs are trained together.

6.5.1 Authentication Protocol

Formally, taking as a reference the scheme in Figure 6.1, we consider a setting where $\boldsymbol{\delta}_n \in \mathbb{R}^M$ with $M < K$. We propose the following authentication protocol, where at time t, upon the reception of a packet by the network, each sensor S_n with $n = 1, \dots, N$,

1. estimates the power delay profile $\{\Pi_n(t, \tau)\}$, i.e., the power of tap with delay τ, of the channel over which the packet was received,
2. extracts the feature vector $\boldsymbol{z}_n \in \mathbb{R}^K$ from $\{\Pi_n(t, \tau)\}$,
3. processes $\boldsymbol{z}_n$ to obtain the (compressed) vector $\boldsymbol{\delta}_n = f_n(\boldsymbol{z}_n) \in \mathbb{R}^M$ and $M < K$.

Next, Bob

4. collects the N local outputs $\boldsymbol{\delta}_n$, $n = 1, \dots, N$, computes $g([\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_N])$, and
5. verifies the authenticity of the packet through the NP test of (6.7).

Again the value of the threshold λ used in the NP test is chosen to match the target FA probability, p_{FA}, hence, indicating with $\mathcal{H} = 1$ ($\mathcal{H} = 0$) the case wherein Alice (Eve) is transmitting, the FA probability is $p_{\text{FA}} = \mathbb{P}[\hat{\mathcal{H}} = 0 | \mathcal{H} = 1]$.

Local functions $f_n(\cdot)$, $n = 1, \dots, N$, and $g(\cdot)$ must be designed to obtain a robust and secure authentication process. Again, the distribution of the input feature vector is unknown, we resort to ML techniques to design them, considering NNs with multiple layers [111].

For a detailed description of the used channel features, see Section 6.3.1. We consider only two-class authentication, assuming that the nodes have at disposal both Alice and Eve channel's observations, leaving the one-class authentication scenario for future works.

As mentioned before we will consider both a) local training and b) global training. With local training, each NN implementing function $f_n(\cdot)$ is trained separately. Afterwards, function $g(\cdot)$ is also trained; this does not require communication between sensors and Bob during training. In the global training scenario, instead, all NNs are trained together as a single large NN including functions $f_n(\cdot)$, $n = 1, \ldots, N$, and $g(\cdot)$; thus, the sensors must communicate with Bob during training.

6.5.2 Local Training

In local training, each function $f_n(\cdot)$, $n = 1, \ldots, N$, is trained locally at its sensor. We propose three options for the output that these NNs report to Bob.

Autoencoder (AE) solution The first solution is to use an autoencoder. As described in 6.4.1, these NN are composed by two sub-nets: the first NN operates as an encoder, $f_{\text{enc}}(\cdot)$ and is in cascade with a second NN, operating as decoder, $f_{\text{dec}}(\cdot)$. The training is performed by minimizing the reconstruction error (6.8).

While previously we used an AE for one-class classification, now we use it as compressor: the trained function f_{enc} acts in fact as a lossy source coder on the input feature vector $\boldsymbol{z}_n$. Note that this solution is not tailored to the hypothesis testing problem, but it aims at providing the best representation of the observed features to Bob (in terms of MSE): in fact, the training of the AE is unsupervised, i.e., we do not exploit the true label of the feature.

Local decision (LD) solution The second solution is the one already described [17] and detailed in Section 6.4.1, where $M = 1$ and the local NN is trained to provide the best authentication test at the local level. In this case, the loss function was the cross entropy with respect to the true label. Although in this case the compression is targeted to the hypothesis testing problem, the information passed to Bob may not allow for an effective exploitation of the correlation among the features observed at the various sensors.

Combined LD and AE (CLDAE) solution To reap the benefits of both local approaches, we propose here a modified version of the AE. In particular, we split the encoder NN $f_n(\boldsymbol{z}_n)$ into two NNs implementing functions $f_{1,n}(\boldsymbol{z}_n)$ and $f_{2,n}(\boldsymbol{z}_n)$, both having the feature vector $\boldsymbol{z}_n$ as input. The first, $f_{1,n}(\boldsymbol{z}_n)$, has a single output and implements the local decision as in Section 6.4.1. The second, $f_{2,n}(\boldsymbol{z}_n)$, has $M - 1$ outputs. The M outputs of both $f_{1,n}(\boldsymbol{z}_n)$ and $f_{2,n}(\boldsymbol{z}_n)$ are seen as the output of the encoder part of an AE providing the reconstructed vector $\tilde{\boldsymbol{z}}_n$. Thus, for training we add in cascade a third NN $f_{3,n}(\boldsymbol{\delta}_n)$ operating as decoder and train both $f_n(\boldsymbol{z}_n)$ and $f_{3,n}(\boldsymbol{\delta}_n)$ to minimize the MSE with respect to the input feature vector, i.e., $\mathbb{E}[||\tilde{\boldsymbol{z}}_n - \boldsymbol{z}_n||^2]$. Figure 6.7 shows an example of design for this solution, for the case $K = 4$ and $M = 3$.

For all three solutions, function $g(\boldsymbol{\delta}_n)$ is implemented as a NN, whose loss function for training is cross entropy with respect to the true label.

Finally, with global training, $f_n(\boldsymbol{x}_n)$, $n = 1, \ldots, N$, and $g(\cdot)$ are jointly trained by using as loss function the MSE with respect to the true label, i.e., $\mathbb{E}[||z - \mathcal{H}||^2]$.

6.5.3 Experiment and Data Augmentation

To test the performance of the proposed authentication technique, we performed a sea experiment in January 2022 in Eilat, Israel. This area is characterized by a complex bathymetry and is thus a good environment to test our method, that relies on source separation based on channel features. For network communications, we used 7 Nanomodem-v3 from Newcastle University, UK, commercialized by Succorfish. These low-cost cylindrical modems measure 4 cm× 6 cm, operate in the 24 kHz to 32 kHz band, and have a source power level of 168 dB. They can be used to transmit packets

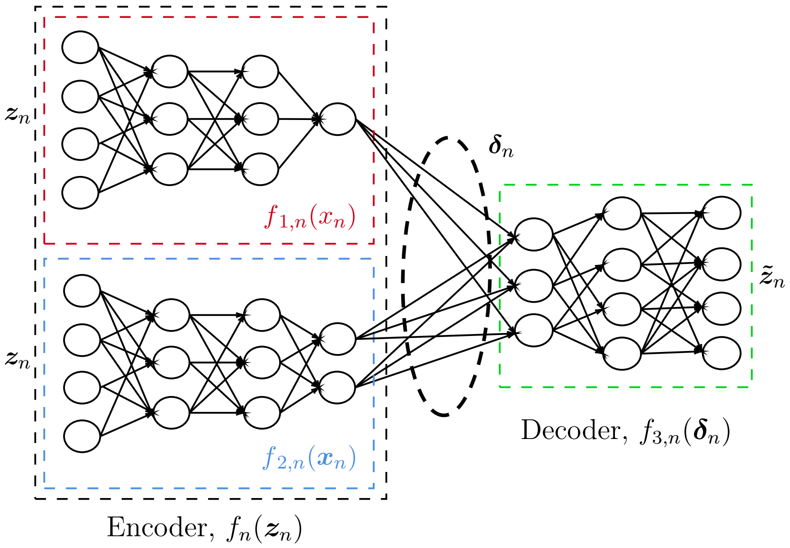

$\boldsymbol{z}_n$
$\boldsymbol{z}_n$
$f_{1,n}(x_n)$
$f_{2,n}(\boldsymbol{x}_n)$
$\boldsymbol{\delta}_n$
$\tilde{\boldsymbol{z}}_n$
Decoder, $f_{3,n}(\boldsymbol{\delta}_n)$
Encoder, $f_n(\boldsymbol{z}_n)$

FIGURE 6.7: Example of architecture for the modified AE with input size $K = 4$ and local output size $M = 3$.

of up to 64 B, either broadcast or unicast, using 640-bps 16-ary orthogonal modulation signals. To obtain the channels' impulse responses, we used Raspberry Pi boards to start polling sequences, in which a trusted node transmits a channel-request and in response receives a message from which the magnitude of the channel's taps is obtained with a resolution of 10 µs.

As shown in Figure 6.8, three floaters were chosen as trusted nodes, and communicated with four submerged drifters. The floaters were anchored roughly 150 m apart along a north-south line at a water depth of 40 m. Each floater integrated a Nanomodem deployed at a depth of 5 m. The four drifters were initially deployed at roughly 250 m from the floaters and formed a north-south line with roughly 50 m spacing. The drifters also integrated one Nanomodem each, and were initially deployed at a place where the water column depth is 85 m. From their deployment point, they drifted at roughly 0.25 knots towards north-west, pushed by the water current. While drifting, they maintained a constant depth of 25 m depth. The three trusted nodes initiated polling cycles using a time-division multiple access (TDMA) protocol having a 1 min cycle. Here, each floater has a 20 s time window, during which it interrogates each of the four drifters to obtain the current channel impulse response.

We obtained a dataset including 30 min of measurements, which were insufficient to train the NNs. Therefore, we artificially generated additional features according to the statistics of the collected data: we fitted each data series $\{z_{n,k}\}$ using a Gaussian KDE model, estimating the PDFs $p_{z_{n,k}}(z)$. Moreover, we add correlation to the estimated features at different sensors to test the importance of a richer ($M > 1$) information transfer from the sensors to Bob.

To generate correlated features, we adopt the following procedure based on the inverse transform sampling method [117]:

1. we generate a $N \times K$ matrix of zero-mean correlated Gaussian variables $\tilde{\boldsymbol{v}}$, with unitary covariances $\text{COV}(v_{n,k}, v_{n',k'}) = 1$, if $n = n', k = k'$, $\text{COV}(v_{n,k}, v_{n',k'}) = \rho$, if $n \neq n', k = k'$, and zero otherwise, where $\rho \in [0, 1]$ is a parameter to control the covariance among sensors;

2. we compute $u_{n,k} = F_x^N(\tilde{v}_{n,k})$, where $F_x^N(\tilde{v}_{n,k})$ is the CDF of a normal distribution, and derive $z_{n,k} = F_{z_{n,k}}^{-1}(u_{n,k})$ via numerical methods.

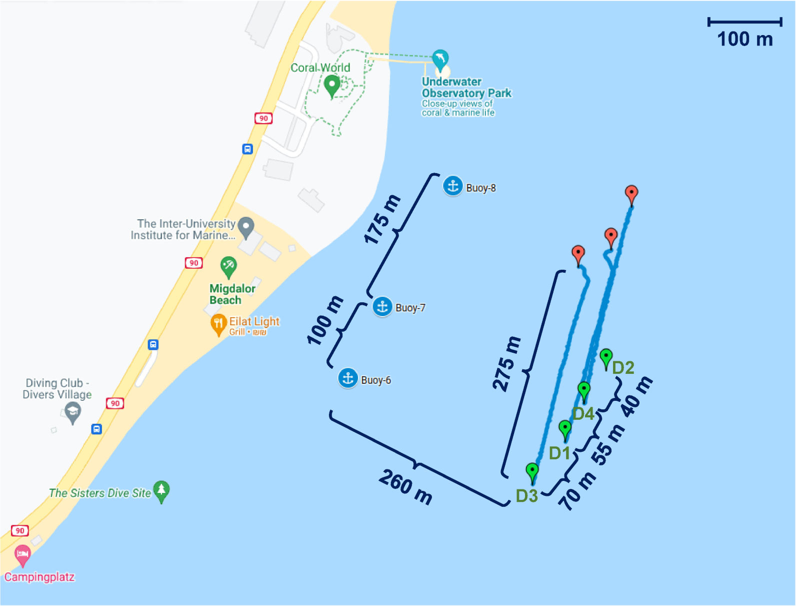

100 m
Coral World
Underwater Observatory Park
Close-up views of coral & marine life
Buoy-8
Buoy-7
Buoy-6
175 m
100 m
275 m
260 m
The Inter-University Institute for Marine...
Migdalor Beach
Eilat Light
Diving Club - Divers Village
The Sisters Dive Site
Campingplatz
D1
D2
D3
D4
40 m
55 m
70 m

FIGURE 6.8: Scheme of the deployment for the sea experiment in Eilat, Israel. The drifters' trajectories are shown as blue lines, where the red circles mark their position at the end of the experiment. The floaters appear as blue circles.

We consider only the measurements from transmitters 1 and 3: the first is considered to be Alice and the second Eve. Each generated dataset contains 10^5 measurements per feature. We generated a dataset for each considered value of ρ and we used 60% of data set for training, 15% for validation, and 25% for testing.

6.5.4 Performance Results

We considered a scenario with $N = 3$ sensors, each computing the $K = 4$ features described in Section 6.3.1. The channel models and the generation of the dataset used for training and performance evaluation have been described in the previous Section. The neurons of the NNs use the ReLu as activation function, unless otherwise specified.

Autoencoder Solution For the encoder AE solution, we considered an input layer with 4 neurons followed by 2 layers with 3 neurons each. The hidden layer has $M = 1$, 2, or 3 neurons. The structure of the decoder mirrors the encoder. The output layer has a linear activation function.

Local decision Solution The LD solution is used both standalone for the case $M = 1$ and as $f_{1,n}(\cdot)$ for the combined LD and AE (CLDAE) solution. We used 4 layers, with 4, 3, 2, and 1 neurons, respectively. The output neuron uses the sigmoid activation function.

Combined LD and AE Solution For the CLDAE solution, function $f_{2,n}(\cdot)$ is implemented as a NN with 3 layers, having 4 neurons in the first layer, 3 in the second, and $M - 1$ in the third. The decoder is implemented as in the AE solution.

In all the three solutions, the NN implementing $g(\cdot)$ was designed with MN neurons in the first layer, N in the second, and a single neuron in the output layer. The output neuron uses the sigmoid activation function.

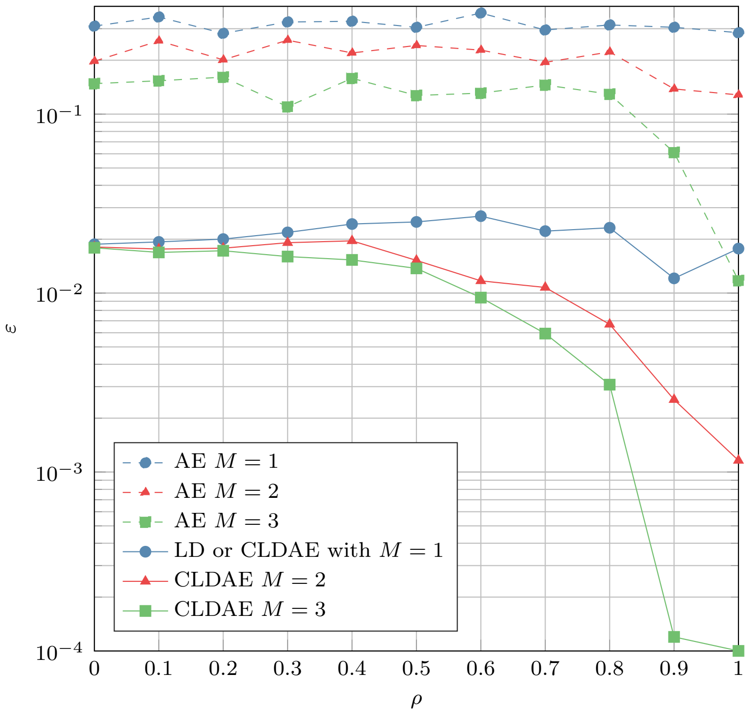


FIGURE 6.9: Error rate ε for the proposed protocol, for $M = 1, 2$ and 3, as a function of ρ considering the local training scenario.

We characterize the performance of our solutions in terms of the error rate, $\varepsilon = \mathbb{P}[\hat{\mathcal{H}} \neq \mathcal{H}]$, obtained for an attack probability of 0.5, i.e., $\varepsilon = 0.5p_{\text{FA}} + 0.5p_{\text{MD}}$, where the testing threshold λ has been optimized to minimize ε.

Figure 6.9 shows the error probability of the local solutions in the considered scenario. As M increases, more information on the observations is provided to Bob, who can also exploit the correlation among them, thus reducing ε. When comparing the various solutions, we note both LD and CLDAE outperform the AE solution, since they are targeted towards the hypothesis testing problem. Moreover, we clearly see that providing more information than only soft local decisions (as would be the case for $M = 1$) decreases the error probability, since Bob can exploit the correlation of the observations at the sensors. Lastly, we observe that the error probability decreases as the correlation increases for all solutions. This is due to the fact that statistics at different sensors are different, thus having multiple highly correlated measurements makes it possible to reduce the decision uncertainty.

Global training For global training, we describe the NNs with notation $a_1 - a_2 - \ldots - a_{Q_L} || b_1 - \ldots - b_{Q_G}$, where a_p indicates the number of neurons of layer p in each local NN, while b_q indicates the number of neurons in layer q of Bob's NN. The considered configurations for $M = 1$, 2, and 3, are reported in the legend of Figure 6.10. Note that in all configurations the total number of neurons is 34, and the output neurons use the sigmoid activation function.

Figure 6.10 shows the error rate ε as a function of the correlation coefficient ρ for different NN configurations. Comparing Figure 6.9 and Figure 6.10, we notice that, even if the latter achieves the best performance, CLDAE with local training achieves almost the same results, even though it operates under more restrictive assumptions: considering, for instance, $\rho = 0.7$ and $M = 2$, by using the AE we would always get $\varepsilon > 10^{-1}$, while with the CLDAE we achieve $\varepsilon \approx 1.52 \cdot 10^{-2}$, close to the value

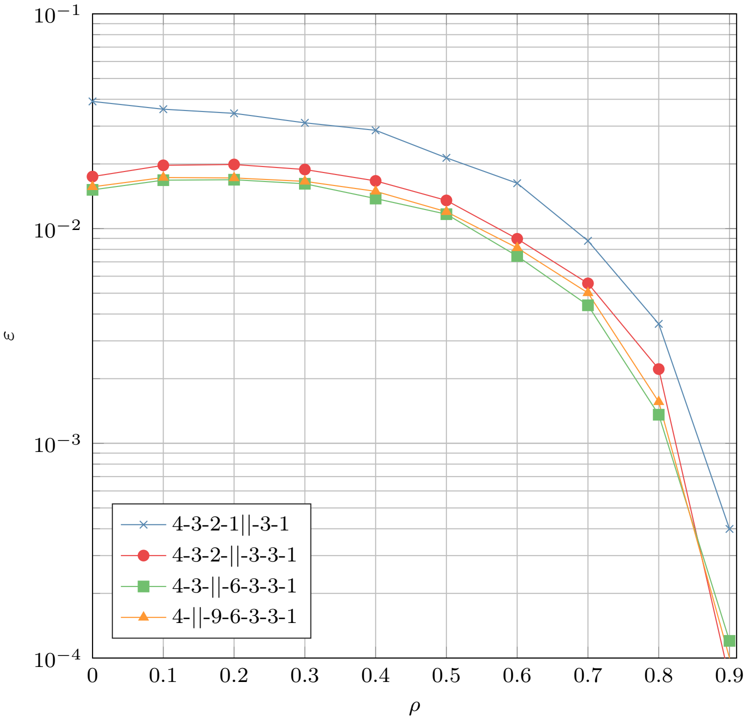


FIGURE 6.10: Error rate ε for the proposed protocol, for $M = 1, 2$ and 3, as a function of ρ considering the global training scenario.

$\varepsilon \approx 1.16 \cdot 10^{-2}$ of the global scenario. Lastly, for $\rho = 1$, we achieve a very low error rate $\varepsilon < 10^{-4}$, not shown in the figure.

6.6 Physical Layer Authentication in UWAN with Mobile Devices

In this section, we propose a PLS authentication technique for underwater acoustic communications, specifically designed to account for the time variability of the channel: for instance this technique may be used to authenticate transmissions coming from an autonomous underwater vehicles (AUVs). Indeed, in UWANs, the variation of the channels due to the continuous device movement (eventually caused also by sea waves and currents) may be significant. As a result, the assumption that the channel does not change across multiple transmissions becomes unrealistic.

To this end, we consider the power-weighed average delay of the channel taps as the authentication parameter, because it is well related to the distance between the transmitter and the receiver. When such delay is measured from the same source to different cooperating receivers, it provides a robust signature of the transmitter. To take mobility into account, we apply a Kalman filter on the average delay, and track delay variations by assuming a simple linear evolution model with slowly varying velocity. The Kalman filter itself will estimate the instantaneous velocity to best track the delay variations. The authentication check is the obtained by comparing the innovation of the Kalman filter with a given threshold. Such innovation indicates the discrepancy between the Kalman-predicted value and the current observation: indeed, an irregular behavior of the observed delay may indicate a possible attack. Then, each innovation is transmitted to Bob, which uses them to verify the authenticity of the packet. Here we focus instead on one-class classification, assuming that Bob has to compute the classifier by having only observation from Alice channel. As classifier we will consider the LC approach, an AE and one-class support vector machine (OC-SVM).

6.6.1 Proposed authentication approach

We consider the system model described in Section 6.3 with some variations, specifically targeted to better model this scenario. As mentioned before Alice is a mobile device, e.g., a drifter or an AUV, that transmits information to Bob periodically. Differently from the previous sections, we consider the set of N sensors, $\mathcal{S} = \{S_1, \cdots, S_N\}$ to have been closely deployed that, as before, will be exploited by Bob to authenticate the received packet.

We remark that, since the decision is based on time information, we assume that Alice is synchronized with Bob: in future work we will consider the case where round-trip delays are estimated via (vulnerable) message exchanges.

In our proposed algorithm, upon receiving a packet at time t:

1. each sensor S_n estimates the power-delay profile $\{\Pi_n(t,\tau)\}$, i.e., $\Pi_n(t,\tau)$ is the power of tap with delay τ, and processes it, extracting a feature $\hat{z}_n(t)$;

2. each sensor S_n then exploits a previously trained model to predict Alice's feature $\tilde{z}_n(t)$; next, it compares the prediction $\tilde{z}_n(t)$ to the measured feature $\hat{z}_n(t)$, computing a model correction term $\beta_n \in \mathbb{R}$;

3. Bob receives the corrections from all his sensors; to improve the performance of scheme, Bob can collect K observations per receiver, concatenated into a vector

$$\boldsymbol{\beta} = [\beta_{1,1}, \cdots, \beta_{1,K}, \beta_{2,1}, \cdots, \beta_{2,K}, \beta_{N,1} \cdots, \beta_{N,K}] \ . \tag{6.10}$$

4. Finally, Bob computes the decision variable $\gamma = g(\boldsymbol{\beta})$, and tests the authenticity of the packet as

$$\hat{\mathcal{H}} = \begin{cases} 0, & \text{if } \gamma < \lambda \quad \text{(packet from Alice)}, \\ 1, & \text{if } \gamma \geq \lambda \quad \text{(packet from Eve)}. \end{cases} \tag{6.11}$$

Bob sets the threshold λ to achieve a target FA probability. In more detail, call $\mathcal{H} = 0$ ($\mathcal{H} = 1$) the case where Alice (Eve) is actually transmitting: the FA probability is $p_{\text{FA}} = \mathbb{P}[\hat{\mathcal{H}} = 1 | \mathcal{H} = 0]$. In the next sections, we describe how each step is implemented, discussing several strategies to design the function $g(\cdot)$.

6.6.1.1 Feature Extraction

We consider the *power-weighed average delay* as the authentication feature. Given the thresholded power delay profile (6.1) $\Pi'_n(t,\tau)$ and assuming that all devices are synchronized, we compute the average delay as

$$z_n(t) = \frac{1}{\bar{\Pi}_n(t)} \sum_{\tau \in \mathcal{H}_n(t)} \tau \Pi'_n(t,\tau) \ , \tag{6.12}$$

where

$$\bar{\Pi}_n(t) = \sum_{\tau \in \mathcal{H}_n(t)} \Pi'_n(t,\tau) \ . \tag{6.13}$$

We remark that it is possible to extend the feature set without loss of generality using the candidate features discussed in [17, 118]. Still a proper state transition and measurement matrices have to introduced. Eventually, it is also possible to resort to the extended Kalman filter (EKF).

6.6.1.2 Prediction Strategy

We now describe how we can use the Kalman to track the average delay feature and understand how much innovation a new transmission brings. To simplify the notation, we drop both the time reference and the sensor index n. Thus, we consider that each sensor collects a sequence of delay measurements $\{\hat{z}_i\}$, where measure $\hat{z}_i$ is associated with the power delay profile observed at time t_i. The task of the Kalman filter is to track the evolution of the distance d_i between S_n and Alice, and the projected velocity v_i, i.e., the velocity component along the direction of the LOS between S_n and Alice. Thus, the true state at step i is $\boldsymbol{x}_i = [d_i, v_i]^{\mathrm{T}}$.

We consider the Kalman Filter model described in Appendix C. Thus, initial conditions aside, we have to choose both the state transition matrix $\boldsymbol{F}_i$ and the measurement matrix $\boldsymbol{H}_i$.

Considering the previously described scenario, we relate subsequent observations of the distance and of the (projected) velocity using a *local linear movement model* with random evolution, i.e., defining the state transition matrix

$$\boldsymbol{F}_i = \begin{pmatrix} 1 & \Delta t_i \\ 0 & 1 \end{pmatrix} , \tag{6.14}$$

where $\Delta t_i = t_i - t_{i-1}$. About the observations, we consider as measurement matrix

$$\boldsymbol{H}_i = \begin{pmatrix} 1/\nu & 0 \end{pmatrix} , \tag{6.15}$$

where ν is the sound speed in water. We remark that, in general, the receiver does not know the actual sound speed in water, also because taking a local sound speed measurement requires the deployment of possibly bulky equipment. Still, we can use an approximated value ν, with the understanding that the term r_i in (C.2) also incorporates sound speed approximation errors.

We also remark that the Kalman filter assumes a Gaussian statistic for both $\boldsymbol{w}_i$ and r_i: while this hypothesis is not always true in our scenario, we still can consider it as an approximation. However, when the hypotheses are met and (C.1) and (C.2) perfectly model the reality, the Kalman filter is proven the be an optimal predictor [119].

The considered feature yields the linear relations (C.1) and (C.2): in general, by choosing a different set of features, these relations may not be linear anymore; in this latter cases it becomes necessary to resort to the EKF.

6.6.1.3 Authenticity Verification

Here, we propose several possible forms of the classification function $g(\cdot)$, which Bob uses to verify the authenticity of a packet (see Scheme in Figure 6.1). We focus on one-class classification solutions, i.e., $g(\cdot)$ can be designed and trained only by using observations from transmissions by Alice. To the best of the authors' knowledge, there is no optimal test for one-class classification, except in specific contexts [120]. Thus, we investigate three classification functions: a) a function based on the LC of the inputs; b) a classifier using an AE NN; and c) a classifier based on a OC-SVM.

Linear combination (LC) The first classifier involves the linear combination of the entries of vector $\boldsymbol{\beta}$, that was considered also in [17] and presented in Section 6.4.2.

In particular, considering that Bob collects K innovations for each sensor in $\mathcal{S}$, it combines them as

$$g_{\mathrm{LC}}(\boldsymbol{\beta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_{n,k} \beta_{n,k} \ . \tag{6.16}$$

As pointed out before, several strategies may be used to estimate the weights, e.g., taking into account the relative distance between each pair of sensors and the (estimated) distance between each sensor S_n and Alice [99]. Here, we consider a worst case analysis where Bob equally weighs each term of $\boldsymbol{\beta}$, i.e., $\alpha_{n,k} = 1$, $\forall k = 1, \ldots, K$ and $\forall n = 1, \ldots, N$.

Autoencoder (AE) As described in the previous Section, AEs are unsupervised feed-forward neural networks whose task is to replicate the input to the output. Between the first part of the AE and the second, we typically have a bottleneck: thus, to generate a vector as close to the original as possible, the AE is forced to learn the useful statistical properties of the input dataset. Typically, the loss function is the MSE between original and reconstructed input.

The AEs can be used as one-class classifier: if trained by using only samples β_ℓ computed after Alice transmissions, the AE will reconstruct with low MSE only the input whose statistic is compatible to the statistical distribution of the input itself [112, 114]. Therefore, the classifier will be $g_{\mathrm{AE}}(\boldsymbol{\beta}_\ell) = \Gamma(\boldsymbol{\beta}_\ell)$ where $\Gamma(\cdot)$ is the MSE function (6.8).

One-class support vector machine (OC-SVM) The idea behind an OC-SVM is to find the boundary that best encloses the training dataset samples. Next, during the testing phase we will consider as legitimate only the samples falling within the boundary, described by the support vector machines (SVM) model. In particular, considering a training dataset of size L, the testing function will be

$$g_{\mathrm{SVM}}(\boldsymbol{\beta}) = \boldsymbol{\alpha}^{\mathrm{T}} \varphi(\boldsymbol{\beta}) + b \ , \tag{6.17}$$

and $\boldsymbol{\alpha}$ and b are respectively weights and bias of the trained OC-SVM classifier and $\varphi(\cdot)$ is a suitable feature transformation function.

To train the classifier we consider the least squares support vector machines (LS-SVM) approach described in [121], where the loss function to be minimized is

$$\begin{gathered} \min_{\boldsymbol{\alpha},b} \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\alpha} + b + C \frac{1}{2} \sum_{\ell=1}^{L} e_\ell^2 \ , \\ \text{with } e_\ell = -\boldsymbol{\alpha}^T \varphi(\boldsymbol{\beta}_\ell) - b \quad \ell = 1, \ldots, L \ , \end{gathered} \tag{6.18}$$

where C is a hyper-parameter that has to be tuned depending on the training dataset itself [122].

6.6.2 Attacker Model for Mobility Scenario

With respect to the previous sections, we consider here a stronger attacker. In detail, we assume that Eve knows all details and parameters of the authentication algorithm, and that it is also synchronized with Alice and Bob. Moreover, Eve can precode its transmissions to reproduce any desired channel impulse response at any of Bob's sensors, including even crafting a different channel response for each sensor. Moreover, Eve does not know the exact location of Alice, but can localize it, e.g., using the well-known approaches of [123, 124], or matched field processing techniques [125, 126]. We remark that the above capabilities imply perfect knowledge of the environment (e.g., the surface and bottom profile, as well as the sound speed profile in the network area), and require channel estimation, precoding, the availability of multiple transceivers, and considerable processing power (including computing multiple ray tracing outputs within a negligible amount of time). Thus, the above model is quite generous towards Eve.

Finally, Eve's estimate of Alice's 3D location vector is then

$$\hat{\boldsymbol{P}}_{\text{Alice}} = \boldsymbol{P}_{\text{Alice}} + \boldsymbol{\varepsilon} \ , \tag{6.19}$$

where $\boldsymbol{P}_{\text{Alice}}$ is the true location of Alice and $\boldsymbol{\varepsilon}$ models the localization error.

6.6.3 Numerical Results

We evaluate our approach by simulating underwater acoustic communication channels via Bellhop [86]. We consider an operational region of about 6×6 km, located in the San Diego bay area and with a depth between 200 and 500 m. The bottom-left corner of the area is located at coordinates $(32°52'34.5''\text{N}, 117°24'12.8''\text{W})$.

At the start of each run, we deploy both Eve and Bob at random within the area. While drifting Alice is forced to stay within the boundary. Bob incorporates four sensors arranged as a tetrahedral pyramid having both a base radius and a height of 5 m.

Alice moves across the area according to a correlated Gauss-Markov mobility model, and sends an acoustic signal once every $\Delta t = 1$ s. Specifically, Alice starts at a random location, $\boldsymbol{P}_{\text{A},0}$, with an initial velocity vector of magnitude $v_0 = \|\boldsymbol{v}_{\text{A},0}\|$ and direction drawn uniformly at random in an interval of 45° around due north. Once every Δt, Alice's location $\boldsymbol{P}_{\text{A},i}$ and velocity $\boldsymbol{v}_{\text{A},i}$ are updated from step t_{i-1} to t_i, as

$$\boldsymbol{P}_{\text{A},i} = \boldsymbol{P}_{\text{A},i-1} + \boldsymbol{v}_{\text{A},i}\Delta t, \tag{6.20}$$

$$\boldsymbol{v}_{\text{A},i} = \rho\, \boldsymbol{v}_{\text{A},i-1} + \boldsymbol{\chi}\, \sqrt{1-\alpha^2}\,, \tag{6.21}$$

where $\rho = 1 - 2 \cdot 10^{-3}$ is the trajectory correlation factor, and $\boldsymbol{\chi}$ is a Gaussian noise vector having (fixed) independent components of standard deviation $[2, 2, 1]$ m/s along the east, north and depth directions, respectively. These choices lead to correlated trajectories, which reproduce the uncertainty of drifting due to currents and eddies. At the same time, the lower variance along the depth dimension signifies a more accurate depth-keeping capability of Alice.

For every signal Alice transmits, we run Bellhop to compute the channel impulse response perceived at each of Bob's sensors as well as at Eve. Moreover, we reproduce different levels of randomness in Eve's estimate of Alice's location by displacing Alice uniformly at random within a radius of either $50, 100, 200$, or 400 m over the azimuthal plane, and within a depth of ±20 m from Alice's actual location. For each random displacement, we recompute the channel impulse response at each of Bob's sensors. Eve's uncertainty on Alice's location then translates into an uncertainty on the channel that Eve should reproduce in order to successfully impersonate Alice.

Figure 6.11 shows the bathymetry map of the area (whose depth is roughly between −250 m and −600 m), the locations of Bob and Eve, and Alice's trajectory for one instance of our simulations. The full Monte-Carlo simulation set includes several realizations of the above scenario, with different locations and trajectories, as well as different movement speeds for Alice, i.e., where the initial velocity magnitude

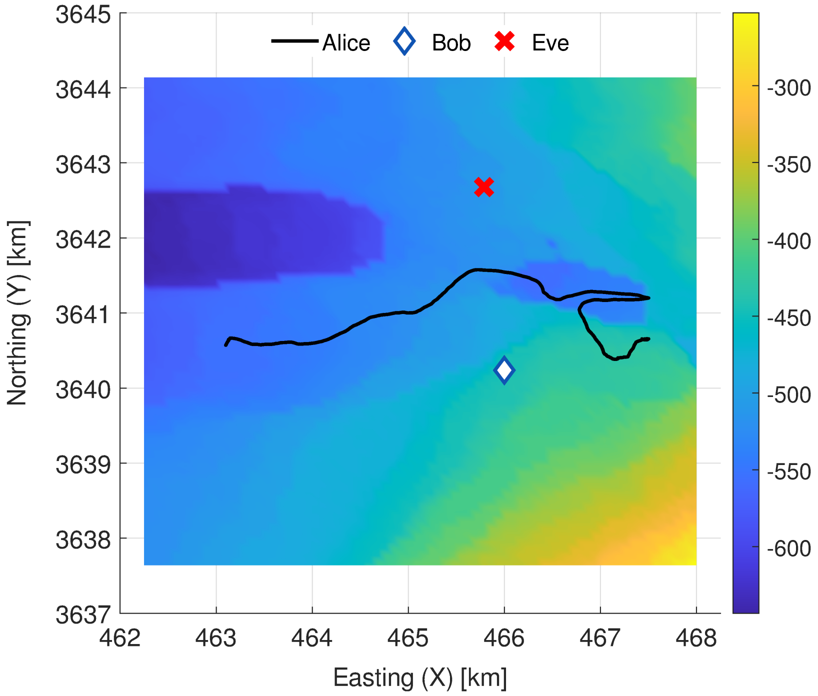


FIGURE 6.11: Example of simulation scenario showing the location of Eve and Bob, and a sample trajectory for Alice. The background colors convey the local depth.

is $v_0 = 0.5, 1$, and $1.5\,\text{m/s}$. In total, we generated 20 simulations for each initial velocity magnitude v_0; each simulation lasts 12 000 s, corresponding to a total of 12 000 power-delay profiles collected per simulation.

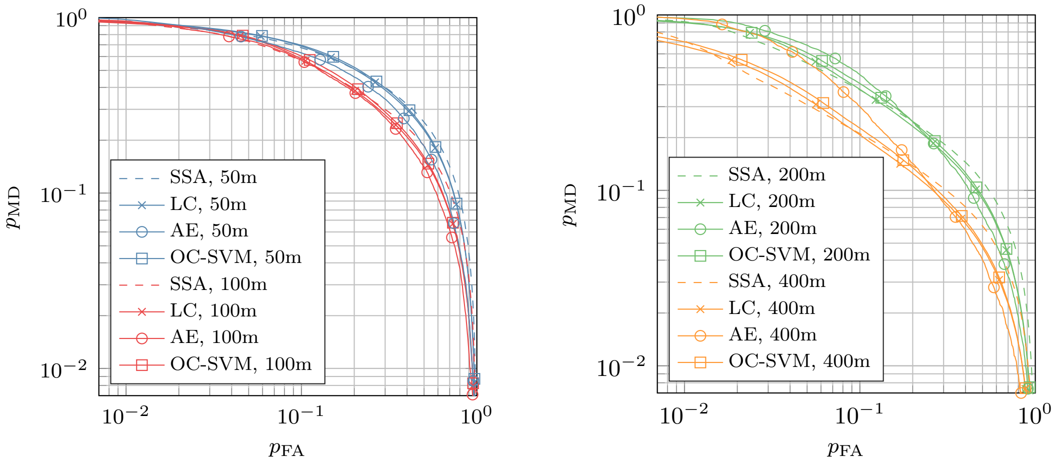
We assume that, for each simulation, there is an initial training period when each sensor S_n receives data only from Alice. We remark that this training dataset has to be used to train both the (local) Kalman filter and the chosen function $g(\cdot)$. In particular, we considered a scenario where each sensor collects 600 (legitimate) feature vectors: thus, we used the first 200 power delay profiles, to train the Kalman filter; next, we input the latter 400 measurements to the Kalman filter, and extract the innovations that will be collected by Bob; this allows us to build the training dataset $\{\beta_\ell\}$ with $L = 400$ observations that will be used to train the actual $g(\cdot)$ functions. Notice that the training dataset size does not depend on K, i.e, the number of innovations that Bob collects from each sensor before making an authentication decision: thus, by increasing K we increase the size of each collected vector $\boldsymbol{\beta}$, but also shorten the training dataset. The remaining part of the legitimate power-delay profiles is used to compute the output in the legitimate case. Once both the Kalman filter and the $g(\cdot)$ function have been trained, we give as input to each sensor up to K subsequent impulse responses associated to transmissions from Eve.

For the Kalman filter, we considered a worst-case scenario where each sensor has no information about Alice. Therefore we always set the initial state to $\boldsymbol{x}_0 = [0, 0]^{\mathrm{T}}$. Indeed, having an initial (even partial) guess on Alice's distance and velocity would allow the Kalman filter to converge with a shorter training dataset, and leave more data to train the $g(\cdot)$ function.

As pointed out in Section 6.3, we assume that each sensor has no information about the others, thus for the LC approach (6.16) we set $\alpha_{n,k} = 1$. For the AE, following the results of [127], we designed both the encoder and the decoder to have one layer each, containing KN neurons. The size of the hidden layer is 2, since it provides the best classifier among the tested configurations. All the neurons have a linear activation function. The training lasted for 5 epochs. Finally, for the OC-SVM, we used a linear kernel function since it achieved better results than both the radial basis function and the polynomial kernels.

6.6.4 Performance Results

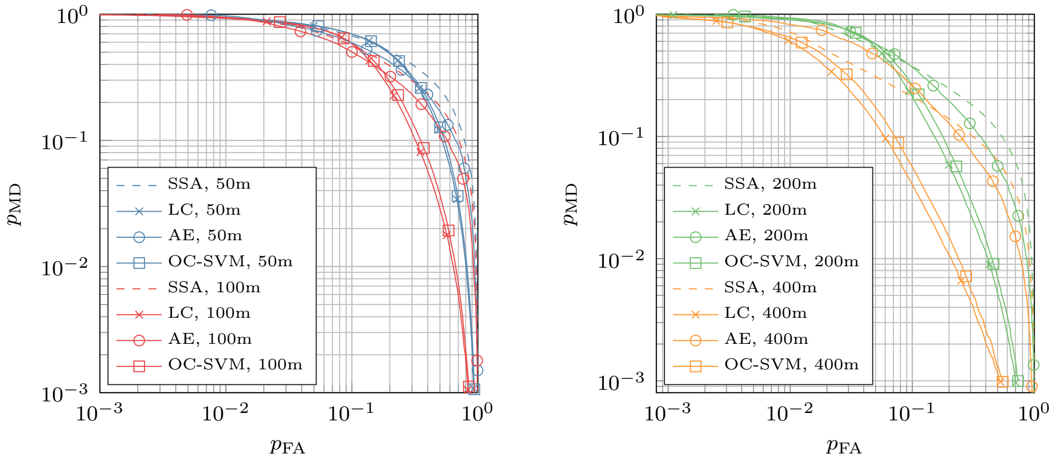
To evaluate the performance of our scheme, we consider the ROC curves, plotting the FA and MD probabilities for different threshold values λ. For comparison purposes we consider also a single-sensor authentication (SSA) classifier, where Bob decides only based on the observation from its topmost sensor. Figs. 6.12a and 6.12b show the results for an Alice initial velocity $v_0 = 1\,\mathrm{m/s}$ and $K = 1$, using the LC, AE, OC-SVM and SSA classifiers, for different attacker estimation accuracies (50 m and 100 m in Figure 6.12a, 200 m and 400 m in Figure 6.12b). As expected, if Eve can estimate the location of Alice more accurately, it has higher chances of successfully impersonating it: in other words, if we fix a given p_{FA}, p_{MD} becomes increasingly higher when the estimate of Alice's location becomes increasingly accurate. Instead, by comparing the different implementations of the function $g(\cdot)$, we notice that all the approaches outperform the SSA, meaning that all proposed schemes can successfully merge local information from different sensors. Moreover, in critical scenarios where Eve's estimate of Alice's location is most accurate, there exist negligible performance differences among the approaches. Conversely, for higher position estimation errors, the AE achieves the worst performance, while the LC and the $OC-SVM$ methods are almost equivalent, with a slight edge for the LC. This may hint to the fact that the components of the vector $\boldsymbol{\beta}$ are (at least close to be) statistically independent.



(A) (B)

FIGURE 6.12: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 50 or 100 m (A) and 200 or 400 m(B) over the azimuthal plane. $K = 1$. AEs: circle; LC: cross; OC-SVM: square.

Figs. 6.13a and 6.13b show the results for the same settings and algorithms, but Bob now collects $K = 3$ observation from each sensor into vector $\boldsymbol{\beta}$. Increasing K improves the classification performance. For example, by directly comparing Figure 6.13b and Figure 6.12b, we observe that setting the threshold λ to achieve a progressively lower p_{MD} leads to a much slower increase in p_{FA} (e.g., $p_{\mathrm{FA}} = 0.06$ for $p_{\mathrm{MD}} = 0.1$ if $K = 3$, against $p_{\text{FA}} = 0.25$ for $K = 1$). We still observe that the AE classifier achieves the worst performance, whereas the LC and the OC-SVM are practically equivalent.



(A)

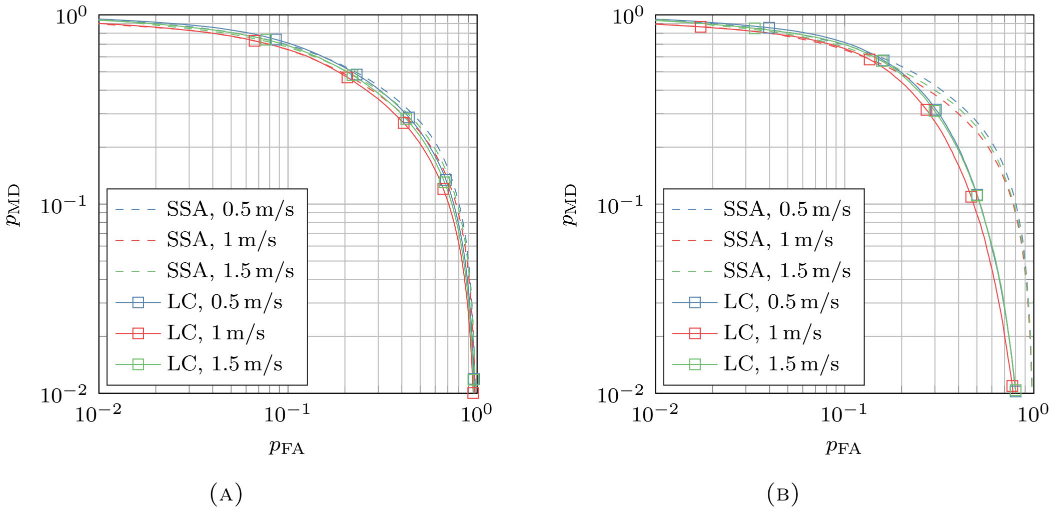
(B)

FIGURE 6.13: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 50 or 100 m (A) and 200 or 400 m(B) over the azimuthal plane. $K = 3$. AEs: circle; LC: cross; OC-SVM: square.

Finally, we investigate how a different average movement speed for Alice affects the classification performance. Besides $v_0 = 1\,\text{m/s}$ as in the previous results, we now considers also $v_0 = 0.5$ and $v_0 = 1.5\,\text{m/s}$. The corresponding results are shown in Figs. 6.14b and 6.14a, respectively for $K = 1$ and $K = 3$. Because the LC classifier exhibits the best tradeoff between complexity and classification performance, we consider only LC in these result, and assume that Eve's accuracy in estimating Alice's location is 400 m. For $K = 1$, we observe no significant difference between the performance of LC for different speeds. However, increasing K (besides leading to better performance for the LC against the SSA classifier) leads to an interesting outcome: the best results are obtained for $v_0 = 1\,\text{m/s}$. This suggests that there are two competing ways in which speed affects the power-weighed average delay metric we use for authentication. On the one hand, a lower speed implies that channels are mostly coherent, so that Eve has good chances to impersonate Alice successfully, even if its location estimate is not too accurate. On the other hand, for $v_0 = 1.5\,\text{m/s}$, the average delay metric tends to change more abruptly, and the innovations computed by the Kalman filter becomes higher with each new legitimate transmission. This also translates into an advantage for Eve, as it decreases the margin the classifier needs to tell apart legitimate and impersonating transmissions.

6.7 Conclusions

In this chapter we have tackled the problem of authentication for UWACs, where a user, Bob takes advantage a of an UWAN to distinguish if a packet was transmitter by a legitimate user, Alice or by an attacker impersonating Alice, Eve. We have investigated several techniques for different settings: first, we have considered the general problem of authentication considering both one-class and two class authentication, i.e., considering if the users at their disposal only Alice observation or a complete dataset containing both Alice and Eve channel observations. Moreover, we have proposed solutions



(A)

(B)

FIGURE 6.14: Results for $K = 1$ (A) and $K = 3$ (B), comparing the single check (dashed) to the described authentication verification functions, with attacker estimation accuracy 400 m and different Alice velocities magnitudes $v_0 = 0.5$, 1, and 1.5 m/s.

for both local and cooperative authentication. Secondly, we have focused on the communication between sensors and Bob: we have presented a solution that aims a filling the bottleneck on the channel between each sensor and Bob that takes also advantage of the correlation among each observation. Finally, we have discussed a PLS authentication technique that targets time varying channels, aiming at authenticating moving users. Simulation results have shown that it is possible to distinguish between Alice and Eve even when Eve has an estimate (and can partially replicate) the legitimate channel.

Chapter 7

Conclusions

The aim of this Thesis was to tackle the challenge of securing wireless communications. To do so, we considered multiple contexts by using diverse tools.

While wireless communication plays in fact an essential role in many fields, by their nature, these are also vulnerable to many threats, since transmitter and receiver have typically limited or no control over the transmission medium. Potentially malicious users are indeed enticed to lead attacks against these transmissions, especially when used for critical infrastructures: jamming and spoofing are examples of attacks that may target wireless communications.

Although often effective, cryptography-based techniques are not suited in many contexts: for instance, some of these solutions may not be used in WSN due to their computational cost, where we have constraints on the energy consumption. Conversely, in GNSS we not only need to authenticate the data, which may be retrieved from alternative sources, but also the signal itself: indeed, cryptographic mechanisms only protect the signal indirectly. Physical layer techniques can provide security exactly in these contexts: for instance, since the security lays on the nature of channel itself, PLS-based mechanisms are typically less computationally expensive than the cryptography-based counterpart. The effort of this Thesis was then to exploit PLS to develop new mechanisms to secure GNSS and UWACs.

In Chapter 2 a novel network aided ranging authentication technique has been proposed. After the analysis of the state of the art security mechanisms, we have described the GLRT-based verification. Next, we have derived an analytical model of the metrics in the under attack scenarios: among others, we have considered the SCER attack and the internal code attack, where the attacker exploits it own -legitimate-code to compute the counterfeit signal. To counteract this last attack, an obfuscation strategy has been proposed. Results have shown both the correctness of our statistical models and the effectiveness of our security protocol against all the considered attacks.

Chapter 3 has described a PVT assurance technique that allows the receiver to enlarge the set of trusted ranging measurements, usable to compute a trusted PVT, by using the authenticated measurements as anchors. By using these techniques, we allow the receiver to freely choose the balance between security and PVT accuracy. Numerical and experimental results have proved the effectiveness of our approach, confirming that it is possible to bound the attacker capabilities even when using non-authenticated measurements.

In Chapter 4 has analyzed a secure timing mechanism targeted for WSN network within a facility. The proposed mechanism exploits the CAS and OSNMA authenticated features to obtain a robust and secure timing source. Moreover, an additional security layer is added to protocol, used to detect those attacks that are not covered by CAS and OSNMA. The performance have been evaluated by using both simulated and experimental results.

GNSSs are used also to broadcast messages, such as for almanacs or for authentication protocols. Since the receiver typically has in view at least 4 satellites, it may be possible to split the message into packets, assign each packet to a satellite, and exploit the receiver diversity to increase the rate. However, this requires designated scheduling solutions, where the system has to assign each packet to each satellite. Thus, Chapter 5 has proposed novel solutions for both the single-round scheduling, where we want most of the receivers on Earth to receive all the packets in one single transmission round, and multi-round scheduling, where we consider multiple consecutive broadcast transmissions. Results have shown that the proposed techniques improve the state of art. In particular, the multi-round scheduling solutions have shown to be close to reach the optimal bounds in terms of both min-max and average latency.

Chapter 6 has tackled the problem of authentication for UWANs, where the sensors belonging to a network cooperate to distinguish legitimate from attacker transmissions. Several aspects have been investigated: the availability of the observations related to attacker's transmissions, distinguishing between one and two-class classification; the presence of a bottleneck that limits the cooperating capabilities of each sensor; the movement of the users, that leads to authentication strategies targeted to time-variant channels. Simulation and experimental results obtained from sea experiments have shown the effectiveness of all the proposed methods.

Besides the ones already mentioned on each Chapter, for GNSS, possible future work will focus on the design of strategies allowing the full interoperability between SCA/SCE solutions, guaranteeing a PVT solution that is both secure, accurate and available all the time. For UWAC, we plan to integrate all the proposed solutions in a broader model that provides authentication capabilities regardless of the considered scenario.

Far from being an extensive overview of all physical layer security aspects related to wireless communications, the effort of this Thesis has focused on tackling the challenge of securing GNSSs and UWACs using a diverse set of tools. This work hopes to be the first step to a broader investigation, which will improve the security and the resilience of the current wireless communication systems.

Appendix A

Analysis and derivations for Network Aided

A.1 Correlation Loss

Upon the reception of a signal $y(t)$, where a fraction ξ of the PRN is unknown to the receiver. The signal is divided in blocks: each block $\boldsymbol{y}$ is a vector composed by the samples associated with a whole PRN. Then, assuming that the receiver is still able to estimate carrier phase and Doppler frequency, we can write

$$\boldsymbol{y} = d\boldsymbol{c} + \boldsymbol{\eta} \tag{A.1}$$

where $\boldsymbol{\eta}$ is the random vector representing the AWGN added by the channel; each sample is then $\eta_\ell \sim \mathcal{N}(0, \sigma^2)$. We consider the process to be successful if the receiver correctly estimates the data symbol d. To do so, the receiver correlates the received signal with its locally generated PRN, $\boldsymbol{c}'$, obtaining

$$\hat{d} = \sum_{\ell=1}^{L} y_\ell c'_\ell = \sum_{\ell=1}^{L} d_n c_\ell c'_\ell + \sum_{\ell=1}^{L} \eta c'_\ell \,. \tag{A.2}$$

Indeed, if $c'_\ell = c_\ell$, the first product is equal to d, while if $c'_\ell \neq c_\ell$, the product is $-d$, so

$$\hat{d} = dL(1-2\xi) + \sum_{\ell=1}^{L} \eta c'_\ell \,, \tag{A.3}$$

therefore the estimated data itself is a Gaussian r.v. with $\hat{d} \sim \mathcal{N}(dL(1-2\xi), L\sigma^2)$.

The receiver will then apply MAP criterion to estimate the received symbol $\hat{d}$: in this case this is equivalent to a sign function. Hence, the error probability is

$$P_\mathrm{e} = P(\hat{d} \neq d) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\mu}{\sqrt{2}\sigma}\right) = -\frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{L}{2}}\frac{(1-2\xi)}{\sigma}\right). \tag{A.4}$$

Calling P'_e the error probability achieved when the receiver knows the whole spreading code (i.e., $\xi = 0$) with noise variance σ', it holds

$$\frac{P_\mathrm{e}}{P'_\mathrm{e}} = (1-2\xi)\frac{\sigma'}{\sigma} \,. \tag{A.5}$$

Fixing $P_\mathrm{e} = P'_\mathrm{e}$, and passing in dB we obtain the relation

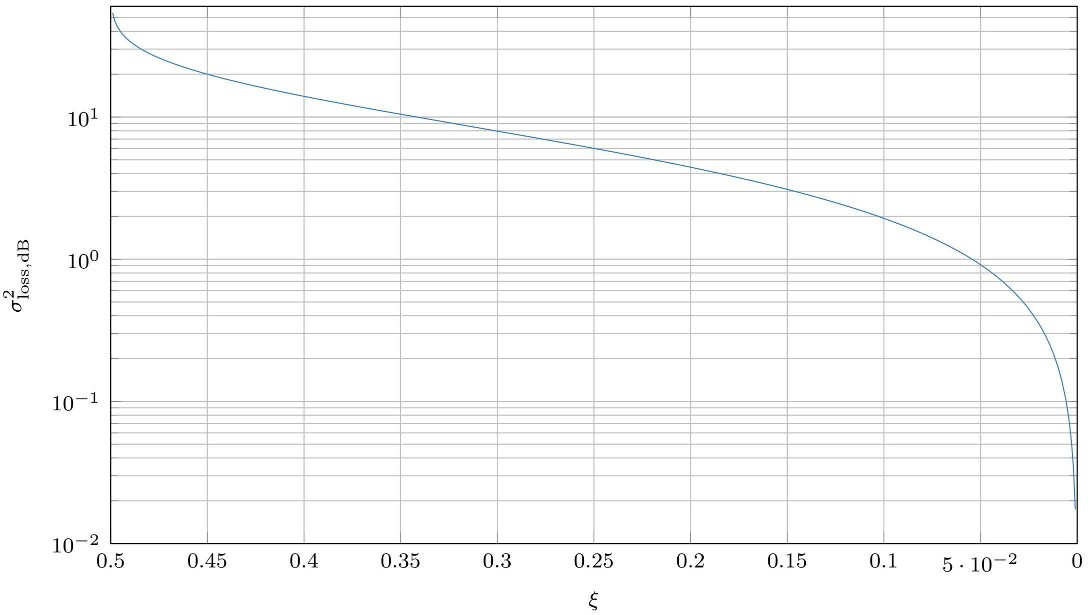


FIGURE A.1: Relationship between correlation loss and ξ for a fixed error probability.

A.2 Proof of Proposition 2

In this Section we report the proof of Proposition 2.

Proof. We start by writing explicitly the GLRT as

$$\begin{aligned}\min_{\alpha} \|\boldsymbol{y}-\alpha\boldsymbol{x}_0\|^2 - \left\|\boldsymbol{y}-\boldsymbol{x}_i'\right\|^2 = \|\boldsymbol{y}\|^2 + \alpha^2\|\boldsymbol{x}_0\|^2 - 2\alpha\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_0 - \|\boldsymbol{y}\|^2 - \left\|\boldsymbol{x}_i'\right\|^2 + 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_i' = \\ = \alpha^2 L - 2\alpha^2 L - L + 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_i' \,,\end{aligned} \tag{A.6}$$

where the $L2$ norm is minimized by picking $\alpha = \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{o}}}{L}$. Next, we write

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_i' = \sum_{\ell}(d\hat{c}_\ell + \eta)dc_\ell' = \sum_{\ell} d^2\hat{c}_\ell c_\ell' + \sum_{\ell} dc_\ell'\boldsymbol{\eta} = L - 2\xi L + \sigma\sqrt{L}\eta_1 \tag{A.7}$$

with $\eta_1 \sim \mathcal{N}(0,1)$. Thus, the GLRT becomes,

$$\min_{\alpha}\ \|y-\alpha x_0\|^2 - \left\|y-x_i'\right\|^2 = L - \alpha^2 L - 4\xi L + 2\sigma\sqrt{L}\eta_1 \tag{A.8}$$

Next, we focus on the second term,

$$\begin{aligned}\alpha\sqrt{L} &= \frac{1}{\sqrt{L}}\sum_{\ell}(d\hat{c}_\ell + \boldsymbol{\eta})(dc_{\ell,\mathrm{o}}) = \frac{1}{\sqrt{L}}\sum_{\ell} d^2\hat{c}_\ell c_{\ell,\mathrm{o}} + \frac{1}{\sqrt{L}}\sum_{\ell} dc_{\ell,\mathrm{o}}\eta \\ &= \sqrt{L}(1-2\rho) + \frac{1}{\sqrt{L}}\sum_{\ell} dc_{\ell,\mathrm{o}}\eta = \sigma\left(\frac{\sqrt{L}}{\sigma}(1-2\rho) + \eta_2\right)\end{aligned} \tag{A.9}$$

Back to the GLRT, we have

$$\begin{aligned}
&\min_{\alpha} \; \|\boldsymbol{y}-\alpha\boldsymbol{x}_0\|^2 - \left\|\boldsymbol{y}-\boldsymbol{x}'_i\right\|^2 = \\
&= L - 4\xi L - \sigma^2\left(\frac{\sqrt{L}}{\sigma}(1-2\rho)+\eta_2\right)^2 + 2\sigma\sqrt{L}\eta_1 = \\
&= L - 4\xi L - L(1-2\rho)^2 - \sigma^2\eta_2^2 - 2\sqrt{L}\sigma(1-2\rho)\eta_2 + 2\sigma\sqrt{L}\eta_1 = \\
&= 4L\rho(1-\rho) - 4\xi L - \sigma^2\eta_2^2 - 2\sigma\sqrt{L}\left(\eta_1-(1-2\rho)\eta_2\right) \; .
\end{aligned} \tag{A.10}$$

Focusing on the last terms, it yields

$$\sigma\sqrt{L}\eta_1 - (1-2\rho)\sigma\sqrt{L}\eta_2 = \sum_\ell dc'_\ell\eta - (1-2\rho)\sum_\ell dc_{0,\ell}\eta = \sum_\ell d(c'_\ell-(1-2\rho)c_{0,\ell})\eta \; , \tag{A.11}$$

which is a linear combination of independent normal r.v.s, therefore it is a normal r.v. with variance

$$\begin{aligned}
\sigma_3^2 &= \sum_\ell d^2\left(c'_\ell-(1-2\rho)c_{0,\ell}\right)^2\sigma^2 = \sigma^2\sum_\ell\left(c'^2_\ell + (1-2\rho)^2c^2_{0,\ell} - 2(1-2\rho)c'_\ell c_{0,\ell}\right) \\
&= 2\sigma^2\sum_\ell(1-2\rho+2\rho^2-(1-2\rho)c'_\ell c_{0,\ell}) \; .
\end{aligned} \tag{A.12}$$

Recalled that $\gamma = \frac{k'}{L}$, we break the series considering separately the cases $c'_\ell \neq c_{0,\ell}$ and $c'_\ell = c_{0,\ell}$, as

$$\begin{aligned}
\sigma_3^2 &= 2\sigma^2\gamma L\left(1-2\rho+2\rho^2+(1-2\rho)\right) + 2\sigma^2(1-\gamma)L(1-2\rho+2\rho^2-(1-2\rho)) = \\
&= 2\sigma^2L\left(2\gamma(1-\rho)^2+2(1-\gamma)\rho^2\right) = 4\sigma^2L(\gamma-2\rho\gamma+\rho^2) \; .
\end{aligned} \tag{A.13}$$

Finally, introducing $\eta_3 \sim \mathcal{N}(0,1)$ we obtain

$$\min_{\alpha} \; \|y-\alpha x_0\|^2 - \left\|y-x'_i\right\|^2 = 4L\rho(1-\rho) - 4\xi L - \sigma^2\eta_2^2 + 4\sigma\sqrt{L}\sqrt{\gamma-2\rho\gamma+\rho^2}\eta_3 \lessgtr \vartheta \; . \tag{A.14}$$

□

Appendix B

Psedoranges derivation for PVT

The pseudorange $R_s^{(f)}(t)$ measured from satellite s with carrier frequency f received at time t can be decomposed in

$$R_s^{(f)}(t) = r_s(t) + c\big(dT_{\mathrm{rx}}(t) - dT_s(t)\big) + T_s(t) + I_s^{(f)}(t) + \mathrm{MP}(t) + \eta(t), \tag{B.1}$$

where $r_s(t)$ is the geometric range, i.e., the distance between transmitter and receiver $r_s(t) = \|\boldsymbol{P}_s(t) - \boldsymbol{P}_{\mathrm{rx}}(t)\|$, $dT_{\mathrm{rx}}(t)$ is the receiver clock bias, $dT_s(t)$ is the satellite clock bias, $T_s(t)$ is the troposheric delay, $I_s^{(f)}(t)$ is the ionospheric delay, $\mathrm{MP}(t)$ is the delay due to multipath and $\eta(t)$ models the errors due to signal processing errors.

We remark that the troposphere is a non-dispersive medium, thus the corrections for the tropospheric delay of band E1, $\hat{D}^{(s)}_{\mathrm{tropo,E1}}(t)$, and E6, $\hat{D}^{(s)}_{\mathrm{tropo,E6}}(t)$ were identical for all $s \in \mathcal{S}$. On the other hand, if the ionosphere is instead a dispersive medium, given the correction for E1, the correction for E6 is [2]

$$\hat{D}^{(s)}_{\mathrm{iono,E6}}(t) = \hat{D}^{(s)}_{\mathrm{iono,E1}}(t)\frac{f^2_{\mathrm{E1}}}{f^2_{\mathrm{E6}}}, \tag{B.2}$$

for all $s \in \mathcal{S}$ and for every time instant t. Correction $\hat{D}^{(s)}_{\mathrm{iono,E1}}(t)$ must be obtained through a proper ionospheric correction model such as the Klobuchar model [128], or more precise models, such as Galileo NeQuick [4] or the IRI-P 2017 [59]. Only the measurements from E6 were actually authenticated; therefore, we could not exploit the measurements from another band (e.g., E1 or E5) to remove the ionospheric delay contribution, as it is typically performed in multifrequency GNSS receivers; instead, we had to use the model computed by using the parameters in the authenticated navigation message.

Appendix C

Kalman Filter

In this appendix we give give a brief description of the Kalman filter used in Chapters 4 and 6.

Differently from the general model, we consider the case where there is no control input.

The procedure was divided into two phases, *prediction* and *model update.* First we introduce the *true state* $\boldsymbol{x}_i$ which is typically unknown. The evolution of the true states is determined by the *state transition matrix* at time t_i, $\boldsymbol{F}_i$, such that

$$\boldsymbol{x}_i = \boldsymbol{F}_i \boldsymbol{x}_{i-1} + \boldsymbol{w}_i \ , \tag{C.1}$$

where $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q_i})$ represents the process noise, assumed to have a Gaussian statistic.

About the observations, we define the *measurement vector* $\boldsymbol{z}_i$ and the measurement matrix $\mathcal{H}$, relating the true state to the observations such that

$$\boldsymbol{x}_i = \boldsymbol{H}_i \boldsymbol{z}_i + \boldsymbol{r}_i \ , \tag{C.2}$$

where $\boldsymbol{r}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R_i})$ models the observation noise.

The Kalman filter performs two operations: *prediction* and *model update.* During the prediction step, it computes the *a priori* state estimate $\tilde{\boldsymbol{x}}_{i|i-1}$ and its covariance matrix $\boldsymbol{P}_{i|i-1}$, respectively, as

$$\tilde{\boldsymbol{x}}_{i|i-1} = \boldsymbol{F}_i \tilde{\boldsymbol{x}}_{i-1|i-1} \tag{C.3a}$$

$$\boldsymbol{P}_{i|i-1} = \boldsymbol{F}_i \boldsymbol{P}_{i-1|i-1} \boldsymbol{F}_i^{\mathrm{T}}. \tag{C.3b}$$

During the update step, the filter computes

$$\boldsymbol{y}_i = \hat{\boldsymbol{z}}_i - \boldsymbol{H}_i \tilde{\boldsymbol{x}}_{i|i-1} \tag{C.3c}$$

$$\boldsymbol{C}_i = \boldsymbol{H}_i \boldsymbol{P}_{i|i-1} \boldsymbol{H}_i^{\mathrm{T}} + \boldsymbol{R}_i \tag{C.3d}$$

$$\boldsymbol{G}_i = \boldsymbol{P}_{i|i-1} \boldsymbol{H}_i^{\mathrm{T}} \boldsymbol{C}_i^{-1} \tag{C.3e}$$

$$\hat{\boldsymbol{x}}_{i|i} = \hat{\boldsymbol{x}}_{i|i-1} + \boldsymbol{G}_i \boldsymbol{y}_i \tag{C.3f}$$

$$\boldsymbol{P}_{i|i} = (\boldsymbol{I} - \boldsymbol{G}_i \boldsymbol{H}_i) \boldsymbol{P}_{i|i-1} \ , \tag{C.3g}$$

where $\hat{\boldsymbol{z}}_{i|i}$ and $\boldsymbol{P}_{i|i}$ are the *updated a posteriori* state estimate and its covariance, respectively, while $\boldsymbol{G}_i$ is called Kalman gain. The prediction error $\boldsymbol{y}_i$ is called *innovation* of the Kalman filter; together with its covariance $\boldsymbol{C}_i$, the innovation is exploited to compute

$$\beta_i = \boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{C}_i^{-1} \boldsymbol{y}_i \ , \tag{C.4}$$

Appendix D

Proof of Optimality of the Min-Max Latency

To prove Proposition 4 we need the following Proposition.

Proposition 3. *If a scheduling sequence* $\boldsymbol{\mathcal{P}}$ *is such that* $\eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) \geq C_{\min,n}$ $\forall n$ *and* $\forall \boldsymbol{x} \in A$*, then it minimizes the maximum latency, achieving the bound* (5.34)*, i.e.* $\tau_{\max}(\boldsymbol{\mathcal{P}}_n) = \tau^*_{\max}$.

Proof. From Proposition 3 we observe that, even in the maximum diversity scenario, $\hat{u}^*_n = 1$ implies $\bar{C}^*_n(\boldsymbol{x}) = K$ $\forall \boldsymbol{x} \in A$, therefore also $C^*_{\min,n} = \min_{\boldsymbol{x}\in A} C^*_n(\boldsymbol{x}) = K$. Thus, we can write the lower bound to the latency for the receiver in position $\boldsymbol{x}$ as

$$\tau^*_{\max}(\boldsymbol{x}) = T \min\left\{n : C_{\min,n} = K\right\}, \tag{D.1}$$

which means that even in the maximum diversity scenario it is possible to deliver K packets to a receiver only if it had at least K satellites in view. Let n^* be the minimum number of rounds that satisfies condition (D.1) for all the receivers on A. Exploiting the hypothesis, after exactly n^* rounds by using scheduling $\boldsymbol{\mathcal{P}}$ that satisfies the hypothesis, we will have delivered at least $C_{\min,n^*} = K$ packets to all the receivers. Thus $\boldsymbol{\mathcal{P}}$ is indeed the scheduling that achieves $\tau_{\max}(\boldsymbol{\mathcal{P}}_n) = \tau^*_{\max}$. □

Now we can recall Proposition 4 and prove it.

Proposition 2. *If* (5.42) *holds, the proposed min-max latency algorithm achieves the optimal latency, i.e.,* $\tau_{\max}(\boldsymbol{\mathcal{P}}_n) = \tau^*_{\max}$.

Proof. The proposed Algorithm 2 delivers K_m packets at round m, where K_m is the largest number of packets that can be delivered in round m with full coverage, i.e, $K_m = \min_{\boldsymbol{x}\in A} \sum_{s\in S} v_{s,m}(\boldsymbol{x})$. Hence, after n rounds, from (5.42) we will deliver

$$\eta_n(\boldsymbol{\mathcal{P}}_n, \boldsymbol{x}) = \sum_{m=1}^{n} K_m = \sum_{m=1}^{n} \min_{\boldsymbol{x}\in A} \sum_{s\in S} v_{s,m}(\boldsymbol{x}) = C_{\min,n}. \tag{D.2}$$

However this also satisfies the requirements of Proposition 3, thus the proposed min-max latency algorithm achieves optimality. □

Bibliography

[1] I. Fernández-Hernández, J. D. Calle Calle, S. Cancela, O. Pozzobon, C. Sarto, and J. Simón, "Packet transmission through navigation satellites: A preliminary analysis using Monte Carlo simulations," in *Proc. of the European Navigation Conference (ENC)*, May 2017, pp. 298–304.

[2] C. Hegarty and E. Kaplan, *Understanding GPS Principles and Applications, Second Edition.* Artech, 2005.

[3] P. Silva, H. Lopes, and J. Silva, "On the evaluation of galileo E5 AltBOC signals for GNSS-INS integration," in *Proc. of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, 09 2012, pp. 1641 – 1650.

[4] EUSPA, "Ionospheric correction algorithm for Galileo single frequency users," https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_Ionospheric_Model.pdf, last access: June 2022.

[5] I. Sesia, G. Signorile, T. T. Thai, P. Defraigne, and P. Tavella, "GNSS-to-GNSS time offsets: study on the broadcast of a common reference time," *GPS Solutions*, vol. 61, no. 25, Apr. 2021.

[6] A. Simsky, D. Mertens, J.-M. Sleewaegen, H. Martin, and C. Massimo, "Experimental results for the multipath performance of Galileo signals transmitted by GIOVE-A satellite," *International Journal of Navigation and Observation*, vol. 2008, Jan. 2008.

[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, p. 120–126, feb 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[8] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[9] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.

[10] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering.* Cambridge University Press, 01 2011.

[11] P. A. van Walree, "Propagation and scattering effects in underwater acoustic communication channels," *IEEE J. Ocean. Eng.*, vol. 38, no. 4, pp. 614–631, Oct. 2013.

[12] D. Grimmett and R. Plate, "Temporal and Doppler coherence limits for the underwater acoustic channel during the LCAS'15 high duty cycle sonar experiment," in *Proc. of OCEANS 2016 MTS/IEEE Monterey.* IEEE, 2016, pp. 1–9.

[13] F. Ardizzon, N. Laurenti, C. Sarto, and G. Gamba, “It’s Galileo time: options for crystal oscillators in OSNMA-enabled receivers,” *GPS World*, vol. 33, no. 1, pp. 16–19, Jan. 2022.

[14] S. Miljanovic, F. Ardizzon, L. Crosara, N. Laurenti, L. Canzian, E. Lovisotto, N. Montini, O. Pozzobon, and R. Ioannides, “Experimental testing and impact analysis of jamming and spoofing attacks on professional GNSS receivers,” 06 2022, international Conference on Localization and GNSS (ICL-GNSS), WIP track.

[15] F. Ardizzon, L. Crosara, N. Laurenti, S. Tomasin, and N. Montini, “Authenticated timing protocol based on Galileo ACAS,” *Sensors*, vol. 22, no. 16, Aug. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6298

[16] R. Diamant, S. Tomasin, F. Ardizzon, D. Eccher, and P. Casari, “Secret key generation from route propagation delays for underwater acoustic networks,” *IEEE Transactions on Information Forensics and Security*, UNDER REWIEV.

[17] L. Bragagnolo, F. Ardizzon, N. Laurenti, P. Casari, R. Diamant, and S. Tomasin, “Authentication of underwater acoustic transmissions via machine learning techniques,” in *Proc. of the International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS).* IEEE, 2021, pp. 255–260.

[18] F. Ardizzon, S. Tomasin, R. Diamant, and P. Casari, “Machine learning-based distributed authentication of UWAN nodes with limited shared information,” in *Proc. the 6th Underwater Communications and Networking Conference (UCOMMS).* Lerici, Italy: IEEE, 2022.

[19] P. Casari, F. Ardizzon, and S. Tomasin, “Physical layer authentication in underwater acoustic networks with mobile devices,” in *Proc. of the 15th International Conference on Underwater Networks (WUWNET22).* ACM, 2022.

[20] F. Ardizzon, G. Caparra, I. Fernandez-Hernandez, and C. O’Driscoll, “A blueprint for multi-frequency and multi-constellation PVT assurance,” in *Proc. of the 10th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Apr. 2022.

[21] U. Maurer, “Authentication theory and hypothesis testing,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1350–1356, June 2000.

[22] P. Baracca, N. Laurenti, and S. Tomasin, “Physical layer authentication over MIMO fading wiretap channels,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2564–2573, July 2012.

[23] A. Ferrante, N. Laurenti, C. Masiero, M. Pavon, and S. Tomasin, “On the error region for channel estimation-based physical layer authentication over Rayleigh fading,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 941–952, Jan. 2015.

[24] E. Jorswieck, S. Tomasin, and A. Sezgin, “Broadcasting into the uncertainty: Authentication and confidentiality by physical-layer processing,” *Proc. IEEE*, vol. 103, no. 10, pp. 1702–1724, Oct. 2015.

[25] A. Rustamov, N. Gogoi, A. Minetto, and F. Dovis, “Assessment of the vulnerability to spoofing attacks of GNSS receivers integrated in consumer devices,” in

Proc. of the International Conference on Localization and GNSS (ICL-GNSS), 2020, pp. 1–6.

[26] G. Caparra, S. Ceccato, N. Laurenti, and J. Cramer, "Feasibility and limitations of self-spoofing attacks on GNSS signals with message authentication," in *Proc. of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 09 2017, pp. 3968–3984.

[27] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach.* Birkhäuser Verlag, 2007.

[28] L. Scott, "Anti-spoofing authenticated signal architectures for civil navigation systems," in *Proc. of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, 9 2003, pp. 1543–1552.

[29] J. M. Anderson, K. L. Carroll, N. DeVilbiss, J. T. Gillis, J. C. Hinks, B. W. O'Hanlon, J. J. Rushanan, L. Scott, and R. A. Yazdi, "Chips-message robust authentication (CHIMERA) for GPS civilian signals," in *Proc. of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 9 2017, pp. 2388–2416.

[30] O. Pozzobon, "Keeping the spoofs out. Signal authentication services for future GNSS," *Inside GNSS*, vol. 6, no. 3, 2010.

[31] L. Scott, "On the achievable equivalent security of GNSS ranging code encryption," in *Proc. of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, 2013, pp. 2880–2892.

[32] B. Motella, D. Margaría, and M. Paonni, "SNAP: An authentication concept for the Galileo open service," in *Proc. of the IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018, pp. 967–977.

[33] L. Meng, L. Yang, W. Yang, and L. Zhang, "A survey of GNSS spoofing and anti-spoofing technology," *Remote Sensing*, vol. 14, no. 19, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/19/4826

[34] E. Commission, "Galileo navigation message authentication specification for signal-in-space testing – v1.1," http://www.kormanyablak.org/it_security/2021-07-04/GALILEO_OSNMA_TESLA.pdf, 10 2018, [Online; accessed Sept-2022].

[35] I. F. Hernández, T. Ashur, V. Rijmen, C. Sarto, S. Cancela, and D. Calle, "Toward an operational navigation message authentication service: Proposal and justification of additional OSNMA protocol features," in *Proc. of the European Navigation Conference (ENC)*, 2019, pp. 1–6.

[36] A. Perrig and J. D. Tygar, *TESLA Broadcast Authentication.* Boston, MA: Springer US, 2003, pp. 29–53.

[37] I. Fernandez-Hernandez, T. Walter, A. Neish, and C. O'Driscoll, "Independent time synchronization for resilient GNSS receivers," in *Proc. of the International Technical Meeting of The Institute of Navigation (ION)*, 01 2020, pp. 964–978.

[38] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simon, I. Rodríguez, and J. D. Calle, "A navigation message authentication proposal for the Galileo open service," *NAVIGATION: Journal of the Institute of Navigation*, vol. 63, no. 1, pp. 85–102, Mar. 2016.

[39] "Galileo Signal-in-Space Interface Control Document," https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf, [Online;Last accessed Sept-2022].

[40] M. G. Kuhn, "An asymmetric security mechanism for navigation signals," in *Information Hiding.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 239–252.

[41] I. Fernandez-Hernandez, S. Cancela, R. Terris-Gallego, G. Seco-Granados, J. A. López-Salcedo, C. O'Driscoll, J. Winkel, A. d. Chiara, C. Sarto, V. Rijmen, D. Blonski, and J. de Blas, "Semi-assisted signal authentication based on galileo acas," 2022. [Online]. Available: https://arxiv.org/abs/2204.14026

[42] R. Terris-Gallego, I. Fernandez-Hernandez, J. A. López-Salcedo, and G. Seco-Granados, "Guidelines for Galileo Assisted Commercial Authentication Service Implementation," in *Proc. of the International Conference on Localization and GNSS (ICL-GNSS)*, 2022, pp. 01–07.

[43] G. Caparra and J. T. Curran, "On the achievable equivalent security of GNSS ranging code encryption," in *Proc. of the IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018, pp. 956–966.

[44] N. Laurenti and A. Poltronieri, "Optimal compromise among security, availability and resources in the design of sequences for GNSS Spreading Code Authentication," in *Proc. of the International Conference on Localization and GNSS (ICL-GNSS)*, 2020, pp. 1–6.

[45] S. Ceccato, F. Formaggio, G. Caparra, N. Laurenti, and S. Tomasin, "Exploiting side-information for resilient GNSS positioning in mobile phones," in *Proc. of the IEEE/ION Position Location and Navigation Symposium (PLANS)*, 04 2018.

[46] A. Joseph, "GNSS solutions: Measuring GNSS signal strength," *Inside GNSS*, 2010.

[47] P. Steigenberger, "The new flex power mode: From GPS IIR-M and IIF satellites with extended coverage area," *INSIDE GNSS*, May 2020.

[48] B. Roturier, E. Chatre, and J. Ventura-Traveset, "The SBAS integrity concept standardised by ICAO-application to EGNOS," http://www.egnos-pro.esa.int/Publications/GNSS%202001/SBAS_integrity.pdf, Dec. 2006, [Online; accessed Sept-2023].

[49] "ICAO Annex 10," radionavigation Aids.

[50] D. L. Mills, *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space, Second Edition.* CRC press, 2016.

[51] S. T. Watt, S. Achanta, H. Abubakari, E. Sagen, Z. Korkmaz, and H. Ahmed, "Understanding and applying precision time protocol," in *Proc. of the Saudi Arabia Smart Grid (SASG)*, 2015, pp. 1–7.

[52] Q. Yang, Y. Zhang, C. Tang, and J. Lian, "A combined antijamming and antispoofing algorithm for GPS arrays," *International Journal of Antennas and Propagation*, Apr. 2019.

[53] M. Meurer, A. Konovaltsev, M. Appel, and M. Cuntz, "Direction-of-arrival assisted sequential spoofing detection and mitigation," in *Proc. of the International Technical Meeting of The Institute of Navigation (ION)*, Feb. 2016.

[54] F. Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS.* Artech, 2009.

[55] T. Walter, J. Blanch, L. DeGroot, L. Norman, and M. Joerger, "Ionospheric rates of change," in *Proc. of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, Sept. 2018, pp. 4158–4170.

[56] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall, 1997.

[57] A. Broumandan and G. Lachapelle, "Spoofing detection using GNSS/INS/odometer coupling for vehicular navigation," *Sensors*, vol. 18, no. 5, Apr. 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/5/1305

[58] Y. Liu, S. Li, F. Qiangwen, and Z. Liu, "Impact assessment of GNSS spoofing attacks on INS/GNSS integrated navigation system," *Sensors*, vol. 18, p. 1433, May 2018.

[59] U. Sezen, T. Gulyaeva, and F. Arikan, "Online computation of international reference ionosphere extended to plasmasphere (iri-plas) model for space weather," *Geodesy and Geodynamics*, vol. 9, no. 5, pp. 347–357, Sept. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S167498471730191X

[60] European Agency for Space Programme (EUSPA), "SAR Galileo service definition document," online; Last accessed Sept 2022. [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo-SAR-SDD.pdf

[61] Y. Wei, H. Liu, J. Ma, Y. Zhao, H. Lu, and G. He, "Global voice chat over short message service of Beidou navigation system," in *Proc. of the 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, pp. 1994–1997.

[62] A. Tarable, R. Andreotti, M. Luise, F. Zanier, and S. Cioni, "Link-layer coding for GNSS navigation messages," *Navigation*, vol. 65, Sept. 2018.

[63] D. Calle, S. Cancela, E. Carbonell, I. Rodríguez, G. Tobias, and I. Fernandez-Hernandez, "First experimentation results with the full Galileo CS demonstrator," in *Proc. of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, Sept. 2016.

[64] I. Fernandez-Hernandez, V. Rijmen, G. Seco-Granados, J. Simón, I. Rodríguez, and J. Calle, "A navigation message authentication proposal for the Galileo open service," *Navigation - Journal of The Institute of Navigation*, vol. 63, Mar. 2016.

[65] E. Union, "Galileo open service navigation message authentication (OSNMA) - user ICD for the test phase." [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf

[66] I. Fernandez-Hernandez, D. Calle, S. Cancela, A. Fernández, R. Martínez, G. Seco-Granados, and P. Walker, "Fountain codes for GNSS," in *Proc. of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, 2017, pp. 1496 – 1507.

[67] European Agency for Space Programme (EUSPA), "Galileo high accuracy service (HAS) - info note," online; Last accessed September 2022. [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_Info_Note.pdf

[68] T. Senni, I. Fernández-Hernández, and S. Cancela, "Message-to-satellite allocation strategies for long GNSS messages and application to the Galileo High Accuracy Service," in *Proc. of the International Conference on Localization and GNSS (ICL-GNSS)*, 2021, pp. 1–7.

[69] T. Senni, I. Fernandez-Hernandez, and S. Cancela, "Evaluation of the Hungarian algorithm for optimal transmission of the Galileo HAS message from multiple satellites," in *Proc. of the International Conference on Localization and GNSS (ICL-GNSS)*, 2022, pp. 1–5.

[70] F. Ardizzon, N. Laurenti, and S. Tomasin, "Sub-messages scheduling in GNSS packet broadcasting by message splitting," in *Proc. of the International Conference on Localization and GNSS (ICL-GNSS)*, 2020, pp. 1–6.

[71] F. Ardizzon, N. Laurenti, and S. Tomasin, "Multi-round message scheduling in GNSS sporadic packet broadcasting," *IEEE Transaction on Aerospace Engineering*, UNDER REVIEW.

[72] Z. Chen, H. Chen, and W. Xu, "Simplified time synchronization for underwater acoustic sensor networks with high propagation latency," in *Proc. of OCEANS 2014 - Taipei*. Taipei, Taiwan: IEEE, 2014, pp. 1–5.

[73] S. Ji, R. Beyah, and Z. Cai, "Minimum-latency broadcast scheduling for cognitive radio networks," in *Proc. of the International Conference on Sensing, Communications and Networking (SECON)*. IEEE, 2013, pp. 389–397.

[74] R. Hayder and L. Dmitri, *Multimedia Over IP And Wireless Networks*, M. Schaar and P. Chou, Eds. Burlington: Academic Press, Jan. 2007.

[75] C. Sarto, O. Pozzobon, S. Fantinato, S. Montagner, I. Fernandez-Hernandez, J. Simon, J. Calle, S. Díaz, P. Walker, D. Burkey, G. Seco-Granados, and E. Göhler, "Implementation and testing of OSNMA for Galileo," in *Proc. of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, 09 2017.

[76] P. Crosta and G. Pirazzi, "A simplified convolutional decoder for galileo os: performance evaluation with a galileo mass-market receiver in live scenario," in *Proc. of the 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Dec. 2016, pp. 1–8.

[77] F. Salgueiro, M. Luise, F. Zanier, and P. Crosta, "Pilot-aided gnss data demodulation performance in realistic channels and urban live tests," in *Proc. of the 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Dec. 2016, pp. 1–8.

[78] C. H. Papadimitriou, "On the complexity of integer programming," *J. ACM*, vol. 28, no. 4, p. 765–768, Oct. 1981. [Online]. Available: https://doi.org/10.1145/322276.322287

[79] "GPS interface specification IS-GPS-200, revision N - August 2022," Last access 1-10-2022. [Online]. Available: https://www.gps.gov/technical/icwg/IS-GPS-200N.pdf

[80] S. Jiang, "On securing underwater acoustic networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 729–752, Aug. 2019.

[81] C. Lal, R. Petroccia, K. Pelekanakis, M. Conti, and J. Alves, "Toward the development of secure underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 1075–1087, Oct. 2017.

[82] M. Stojanovic and J. Preisig, "Underwater acoustic communication: Propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, Jan. 2009.

[83] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The JANUS underwater communications standard," in *Proc. the Underwater Communications and Networking Conference (UCOMMS).* Sestri Levante, Italy: IEEE, Sept. 2014.

[84] C. Lal, R. Petroccia, M. Conti, and J. Alves, "Secure underwater acoustic networks: Current and future research directions," in *Proc. of the Underwater Communications and Networking Conference (UCOMMS).* Lerici, Italy: IEEE, Aug. 2016.

[85] G. Yang, L. Dai, G. Si, S. Wang, and S. Wang, "Challenges and security issues in underwater wireless sensor networks," *Procedia Computer Science*, vol. 147, pp. 210–216, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919302443

[86] M. Porter *et al.*, "Bellhop gaussian beam/finite element beam code," http://oalib.hlsresearch.com/Rays/index.html. Last accessed: June. 2021.

[87] N. Morozs, W. Gorma, B. T. Henson, L. Shen, P. D. Mitchell, and Y. V. Zakharov, "Channel modeling for underwater acoustic network simulation," *IEEE Access*, vol. 8, pp. 136 151–136 175, July 2020.

[88] E. Souza, H. C. Wong, I. Cunha, L. F. M. Vieira, and L. B. Oliveira, "End-to-end authentication in under-water sensor networks," in *Proc. of the 27th Symposium on Computers and Communications (ISCC 2013).* IEEE, 2013, pp. 299–304.

[89] F. Campagnaro, D. Tronchin, A. Signori, R. Petroccia, K. Pelekanakis, P. Paglierani, J. Alves, and M. Zorzi, "Replay-attack countermeasures for underwater acoustic networks," in *Proc. of OCEANS 2020 MTS/IEEE*, Singapore, 2020, pp. 1–9.

[90] G. Dini and A. Lo Duca, "A secure communication suite for underwater acoustic sensor networks," *MDPI Sensors*, vol. 12, pp. 15 133–15 158, Oct. 2012. [Online]. Available: http://dx.doi.org/10.3390/s121115133

[91] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography.* Boca Raton, FL: CRC Press, 1996.

[92] D. Muhammed, M. H. Anisi, M. Zareei, and A. Khan, "Game theory-based cooperation for underwater acoustic sensor networks: Taxonomy, review, research challenges and directions," *Sensors*, vol. 18, no. 2, Feb. 2018.

[93] M. Sharif-Yazd, M. R. Khosrav, and M. K. Moghimi, "A survey on underwater acoustic sensor networks: Perspectives on protocol design for signaling, MAC and routing," *Journal of Computer and Communications*, vol. 5, no. 5, pp. 12–23, Feb. 2017.

[94] C. Yuan, W. Chen, Y. Zhu, D. Li, and J. Tan, "A low computational complexity authentication scheme in underwater wireless sensor network," in *Proc. of the 11th International Conference on Mobile Ad-hoc and Sensor Networks.* Shenzhen, China: IEEE, 2015, pp. 116–123.

[95] A. Al Guqhaiman, O. Akanbi, A. Aljaedi, and C. E. Chow, "Lightweight multi-factor authentication for underwater wireless sensor networks," in *Proc. of the International Conference on Computational Science and Computational Intelligence (CSCI).* Las Vegas, NV, USA: IEEE, 2020, pp. 188–194.

[96] S. Zhang, X. Du, and X. Liu, "A secure remote mutual authentication scheme based on chaotic map for underwater acoustic networks," *IEEE Access*, vol. 8, pp. 48 285–48 298, Mar. 2020.

[97] A. A. Islam and K. A. Taher, "A novel authentication mechanism for securing underwater wireless sensors from Sybil attack," in *Proc. of the 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT).* Mirpur, Dhaka: IEEE, 2021, pp. 1–6.

[98] H. Kulhandjian, T. Melodia, and D. Koutsonikolas, "Securing underwater acoustic communications through analog network coding," in *Proc. IEEE SECON*, Singapore, Jun. 2014.

[99] R. Diamant, P. Casari, and S. Tomasin, "Cooperative authentication in underwater acoustic sensor networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 954–968, Feb. 2019.

[100] K. Pelekanakis, S. Yıldırım, G. Sklivanitis, R. Petroccia, J. Alves, and D. Pados, "Physical layer security against an informed eavesdropper in underwater acoustic channels: Feature extraction and quantization," in *Proc. the Underwater Communications and Networking Conference (UCOMMS).* IEEE, 2021, pp. 1–5.

[101] K. Pelekanakis, C. M. G. Gussen, R. Petroccia, and J. Alves, "Robust channel parameters for crypto key generation in underwater acoustic systems," in *Proc. of OCEANS 2019 MTS/IEEE - Seattle.* Seattle, WA, USA: IEEE, Oct. 2019, pp. 1–7.

[102] Y. Liu, J. Jing, and J. Yang, "Secure underwater acoustic communication based on a robust key generation scheme," in *Proc. of 9th International Conference on Signal Processing*, 2008, pp. 1838–1841.

[103] R. Zhao, M. Khalid, O. A. Dobre, and X. Wang, "Physical layer node authentication in underwater acoustic sensor networks using time-reversal," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3796–3809, Jan. 2022.

[104] L. Xiao, G. Sheng, X. Wan, W. Su, and P. Cheng, “Learning-based PHY-layer authentication for underwater sensor networks,” *IEEE Communications Letters*, vol. 23, no. 1, pp. 60–63, Oct. 2019.

[105] M. Khalid, R. Zhao, and N. Ahmed, “Physical layer authentication in line-of-sight underwater acoustic sensor networks,” in *Proc. MTS/IEEE OCEANS*. Singapore: IEEE, 2020, pp. 1–5.

[106] W. Aman, Z. Haider, S. W. H. Shah, M. M. Ur Rahman, and O. A. Dobre, “On the effective capacity of an underwater acoustic channel under impersonation attack,” in *Proc. IEEE ICC*. Dublin, Ireland: IEEE, 2020.

[107] A. Vermeij and A. Munafò, “A robust, opportunistic clock synchronization algorithm for ad hoc underwater acoustic networks,” *IEEE J. Ocean. Eng.*, vol. 40, no. 4, pp. 841–852, Oct. 2015.

[108] R. Diamant and L. Lampe, “Underwater localization with time-synchronization and propagation speed uncertainties,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1257–1269, July 2013.

[109] R. Diamant, “Closed form analysis of the normalized matched filter with a test case for detection of underwater acoustic signals,” *IEEE Access*, vol. 4, pp. 8225–8235, Nov. 2016.

[110] J. Neyman and E. S. Pearson, “On the problem of the most efficient tests of statistical hypotheses,” *Philosophical Transactions of the Royal Society of London Series A*, vol. 231, pp. 289–337, Jan. 1933.

[111] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[112] A. Brighente, F. Formaggio, G. M. Di Nunzio, and S. Tomasin, “Machine learning for in-region location verification in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2490–2502, Aug. 2019.

[113] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.

[114] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos,” *Pattern Recognition Letters*, vol. 105, no. C, pp. 13–22, Apr. 2018.

[115] B. Dushaw, “Worldwide Sound Speed, Temperature, Salinity, and Buoyancy from the NOAA World Ocean Atlas,” http://staff.washington.edu/dushaw/WOA/. Last accessed: June. 2021.

[116] G. Roussas, *Nonparametric Functional Estimatoin and Related topics*. Springer, 1991, NATO ASI Series.

[117] L. Devroye, *Non-Uniform Random Variate Generation*. Springer New York, 1986. [Online]. Available: https://doi.org/10.1007%2F978-1-4613-8643-8

[118] G. Sklivanitis, K. Pelekanakis, S. Yıldırım, R. Petroccia, J. Alves, and D. Pados, “Physical layer security against an informed eavesdropper in underwater acoustic channels: Reconciliation and privacy amplification,” in *Proc. of the Underwater Communications and Networking Conference (UCOMMS)*. IEEE, 2021, pp. 1–5.

[119] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Englewood Cliffs, NJ: Pearson Educationll, 1993.

[120] O. Zeitouni, J. Ziv, and N. Merhav, "When is the generalized likelihood ratio test optimal?" *IEEE Transaction on Information Theory*, vol. 38, no. 5, pp. 1597–1602, Sept. 1992.

[121] Y.-S. Choi, "Least squares one-class support vector machine," *Pattern Recogn. Lett.*, vol. 30, no. 13, p. 1236–1240, Oct. 2009.

[122] X. Guo, J. Yang, C. Wu, C. Wang, and Y. Liang, "A novel LS-SVMs hyper-parameter selection based on particle swarm optimization," *Neurocomputing*, vol. 71, no. 16, pp. 3211–3215, Oct. 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231208002932

[123] E. Dubrovinskaya, V. Kebkal, O. Kebkal, K. Kebkal, and P. Casari, "Underwater localization via wideband direction-of-arrival estimation using acoustic arrays of arbitrary shape," *Sensors*, vol. 20, no. 14, pp. 1–20, July 2020.

[124] I. Ullah, J. Chen, X. Su, C. Esposito, and C. Choi, "Localization and detection of targets in underwater wireless sensor using distance and angle based algorithms," *IEEE Access*, vol. 7, pp. 45 693–45 704, Apr. 2019.

[125] E. Dubrovinskaya, P. Casari, and R. Diamant, "Bathymetry-aided underwater acoustic localization using a single passive receiver," *Journal of Acoustic Society of America*, vol. 146, no. 6, pp. 4774–4789, Dec. 2019.

[126] Z.-H. Michalopoulou, P. Gerstoft, and D. Caviedes-Nozal, "Matched field source localization with Gaussian processes," *JASA Express Letters*, vol. 1, no. 6, June 2021.

[127] H. Steck and D. G. Garcia, "On the regularization of autoencoders," 2021. [Online]. Available: https://arxiv.org/abs/2110.11402

[128] J. A. Klobuchar, "Ionospheric time-delay algorithm for single-frequency GPS users," *Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 3, pp. 325–331, May 1987.