Binary Tardos Codes and Zero-bit Watermarking

Teddy Furon

▶ **To cite this version:**

Teddy Furon. Binary Tardos Codes and Zero-bit Watermarking. Signal and Image Processing. Université de Rennes 1, 2018. tel-01932766

HAL Id: tel-01932766
https://inria.hal.science/tel-01932766

Submitted on 27 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ANNÉE 2018

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

HABILITATION À DIRIGER DES RECHERCHES

UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

Mention : Traitement du signal - Télécommunications

présentée par

Teddy Furon

préparée à l'unité de recherche IRISA (UMR 6074)
Institut de Recherche en Informatique et Systèmes Aléatoires

Intitulé de la thèse

Binary Tardos Codes and Zero-Bit Watermarking

Thèse soutenue à Rennes le 29 octobre 2018

devant le jury composé de :

Marc CHAUMONT
Maitre de Conférence HDR
Université de Nîmes / rapporteur

Stefan KATZENBEISSER
Professeur
Université Technique de Darmstadt / rapporteur

Mauro BARNI
Professeur - Université de Sienne / examinateur

Patrick BAS
Directeur de Recherche - CNRS / examinateur

David GROSS-AMBLARD
Professeur
Université de Rennes 1 / président du jury

Boris ŠKORIĆ
Professeur assistant
Université Technique d'Eindhoven / examinateur

Sviatoslav VOLOSHYNOVSKIY
Professeur - Université de Genève / examinateur

"Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte."
Blaise Pascal, Les Provinciales, 4 décembre 1656.

often translated as
"I would have written a shorter letter, but I did not have the time."

Contents

Chapter 1

Introduction

This '*Habilitation à Diriger des Recherches*' is divided into two parts. The first part is an overview of traitor tracing with binary Tardos codes. The seminal paper of Gábor Tardos dates back to 2003 [163]. After fifteen years of research, we have now a (almost) complete view of this field and it has recently become a technology deployed in real world products. My purpose here is to summarize these research results into a *pedagogical* presentation of this subject. Yet, I could not help inserting few new contributions (which are indeed natural extensions of previous ideas):

- Rates of conditioned decoders which show that caught colluders denounce their accomplices (see Fig. 4.4),
- A procedure to find the minimum number of colluders to cancel the achievable rate of a single decoder. This shows that key distributions are not equally secure (see Fig. 4.5).
- The estimation of the 'number' of collusion strategies which are more harmful than the simple interleaving attack (see Fig. 4.6 and 4.7). This shows how fast the interleaving attack becomes the worst attack as the number of colluders increases.
- A new model of the watermarking layer yielding symbol erasures and double symbol detections (see Fig. 4.8 and 4.9). This shows that mixing content blocks at the signal processing level is not always a good idea for the collusion.
- A better experimental assessment of the performance of the E-M decoder (see Fig. 5.4). This was poorly done previously in [36].
- A benchmark of the state-of-the-art single decoders including adaptive decoders (see Fig. 6.4 and 6.5). It clearly shows that the adaptive iterated single offers the best trade-off between complexity and performances at the accusation side.

The second part of the Habilitation deals with zero-bit digital watermarking. This domain was born more than twenty years ago, and it has always been industry-driven. Its theoretical foundation is still incomplete, especially for its zero-bit version. This second part is my humble contribution towards this goal. It is a very technical investigation targeting an audience of experts more than a pedagogical overview. I did my best to slowly introduce the concepts (Chap. 7) and to present the study by gradual steps: Zero-bit watermarking without side-information(Chap. 8), with side-information but without noise (Chap. 9), with side-information and noise (Chap. 10). The starting point of this track of research is the key work of P. Comesaña , N. Merhav, and M. Barni [99]. They study how the error probabilities vanish to zero in a theoretical setup (Gaussian distribution of the host and signal, asymptotically long vectors, noiseless case) for the

dual hypercone scheme. This part of the Habilitation generalizes this work by considering a noisy setup and several other schemes (including a new scheme). At last, it tries to make the connection with another piece of theory, which I developed earlier [23].

I have been working on other research fields than traitor tracing and zero-bit watermarking. As a prologue, this introductory chapter presents the topics related to multimedia security from a personal and historical point of view. It aims to bring a 'logical articulation' in the maze of my works. A list of publications by topics concludes this chapter.

Personal views

The open issues after my Ph.d. thesis

I did a so-called in french 'thèse CIFRE' *i.e.* a thesis in the industry, working for Thomson (now named Technicolor) in their security lab. My thesis deals with digital watermarking for the application of DVD copy protection [75]. It proposes a new concept: asymmetric watermarking [113]. This wording was a wrong idea. I should have called it "probabilistic watermarking" as it looks like more a probabilistic encryption scheme than a public key cryptosystem. A youthful mistake. In brief, the embedding of a digital watermark depends on a random variable, whose value is not needed at the detection side. The main result is a rise of the security level.

Two observations fuelled my research work just after the Ph.D. thesis:

1. The robustness is lower than the one of the baseline (at these times, the additive spread spectrum scheme) in expectation over the a random variable. However, for a specific value of the a random variable depending on the host content, the robustness is indeed much better than the baseline.
2. "More security" was an argument not understood at that time, mainly because people confused security with robustness.

The concept of side-information zero-bit watermarking stems from the first observation: the watermark signal should depend on the host signal. Of course, people working on multi-bit watermarking knew this point since the rediscovery of M. Costa's paper [100] by B. Chen and G. Wornell [95]. But there was no and there is still a partial knowledge on how side-information can improve zero-bit watermarking scheme. I also remember Neri Merhav, invited speaker at the Information Hiding workshop 2005 Barcelona, stating that he could not see a solution to this ill-posed problem. Part II presents an overview together with some new contributions defending my early statement: the community is still missing a complete view of zero-bit watermarking.

As for security, my aim after the Ph.d. thesis was to put this feature on the table as a new concept for watermarking. I have appreciated a lot working with my first postdoc, François Cayre, and new colleague Caroline Fontaine. Our program was based on three pillars:

- Evangelize that robustness and security are different concepts [1, 12]
- Establish a theory of watermarking security [11],
- Propose real case studies with practical attacks on well known watermarking techniques [10, 18, 19],

Several years after, I could not help revisiting once again watermarking security from a totally different perspective with my good friend Patrick Bas [4].

I have already published too much on this topic. I was not feeling like summarizing again my contributions in this Habilitation, especially because some overviews were published recently [6, 14].

Zero-bit watermarking

People usually thinks that zero-bit watermarking is a niche as it has no real-life application. Of course, I strongly disagree. Yet, from a theoretical point of view, applying the idea of side-informed encoder from M. Costa to zero-bit watermarking was challenging. Indeed, I have miserably failed because I was not knowledgeable enough in information theory [26]. Being stubborn, I am giving it another try in Part II, but with the will of not betraying my nature: I am a signal processing researcher who knows the requirements of digital watermarking and the gap with the assumptions of theoretical papers. I have been more successful with my own theory of side-informed zero-bit watermarking which is not based on information theory but on statistics. Part II makes the connections between these two theories.

Broken Arrows, the still image watermarking technique co-designed with Patrick Bas and used in the challenge BOWS2 (Break Our Watermarking Scheme, 2^{nd} edition) is the practical side of this track of research.

The square root of Gabor Tardos

During the year 2007, I had the feeling that I could no longer improve my understanding of security and zero-bit watermarking. I was not inclined towards steganography / steganalysis, the hot topic of my community at that time, for some ethical reasons: I still believe that the steganographer is the winner of this game, and that the steganographer is usually a terrorist. On the other hand, fingerprinting, a.k.a. traitor tracing, was not my cup of tea as it relied on code theory (*i.e.* discrete mathematics)... until Gábor Tardos proposed a probabilistic construction achieving minimal asymptotic code length [163, 164]. Moreover, his scoring function totally bewildered me: The observations being statistically independent, the score function obviously is a sum ... but a sum of square roots, whereas I would have expected it to be a sum of logarithms like in usual Neyman-Pearson (a.k.a. Maximum Likelihood) score functions.

The probabilistic construction of Tardos codes and this square root triggered my curiosity in traitor tracing. With Ph.d. students, Fuchun Xie and Ana Charpentier, and postdocs, Luis Perez-Freire and Peter Meerwald, we first tried to understand G. Tardos' choice [34], and then propose some improvements referenced in part I. Group testing is very similar and even easier than traitor tracing so the application of some of my ideas to this application was straightforward.

My expertise on traitor tracing raised the interest of my former company, Technicolor. I worked back in its security lab for a year and a half. Later on, expertise, algorithms, and patent have also been transferred to `b-com` technical institute and the startup company Lamark, which I co-funded.

Security and information retrieval

Back at Inria, I moved in another research team, Texmex now called Linkmedia, working on multimedia information retrieval. I have few papers in nearest neighbors search and image similarity search thanks to the collaboration with my new colleagues expert in the field, Hervé Jégou, Laurent Amsaleg, Ewa Kijak, and Giorgos Tolias. One recent idea was the application of group testing to speed up nearest neighbor search. Both can be summarized as finding needles in a

haystack, so there ought to be a link to be investigated. This was the task of the Ahmet Iscen, who recently earned the Best Ph.D. thesis award in computer science of University of Rennes I.

My first contribution in this field is the introduction of security in information retrieval, and especially CBIR (Content Based Image Retrieval). This technology became a key tool in copyright management as it allows computers to identify multimedia contents, hence to enforce digital content rights. Whereas its robustness to conventional editing processes was well established, the concept of security was absent. In other words, history was repeating from digital watermarking to CBIR. With my dear colleagues Ewa Kijak and Laurent Amsaleg whom I converted to the importance of security, and our Ph.d. student Thanh-Toan Do, we have played the role of the attacker designing specific manipulations to make a piece of content un-identified or mis-identified. This task has also been carried on for face recognition thanks to a collaboration with the GREYC lab of University of Caen.

Another investigation was the possibility to query a large database with some levels of security and privacy. It amounts to perform a nearest neighbour search where both the query and the database vectors are obfuscated. With Benjamin Mathon (Postdoc during the Secular project), we noticed that this task was investigated by two communities inventing systems either highly scalable, or highly secure (using partially of fully homomorphic cryptography). A kind of no man land in between these two worlds stems from the lack of communication between these two communities. Our aim was to bridge this gap with the idea that signal processing is able to promote some security level with a reasonable complexity hampering less the scalability. This work had very few impact because our system is 'half secure' and 'half scalable', therefore not interesting to any research community. I still believe that it makes sense in practice.

More recently, I seized the opportunity of a collaboration with Alcatel Bell Labs (now Nokia Bell Labs) to turn to differential privacy for recommendation systems. Together with my Ph.d. student Raghavendran Balu, we design a system coping with large scale setups. I also applied what I learn from traitor tracing to differential privacy. This concept assesses that an attacker observing the outputs of the system (*i.e.* recommendations) cannot decide whether a particular rating was used for training the recommendation system. The idea was to replace one particular rating by a subset of ratings: Differential privacy limits the efficiency of a single decoder but traitor tracing / group testing learned me that joint decoding is more powerful. This means that by considering subset of ratings, the attacker is theoretically more powerful than what differential privacy foresees. The problem with joint decoding is its complexity which we tackled by using a MCMC (Monte Carlo Markov Chain) algorithm. This work earned the Best Student Paper Award at the ESORICS conference.

Rare Events

I also have the chance of knowing friends who are researchers in different fields fostering pleasant collaborations. This 'rare event' gave me a lot of fun. Together we solve one crucial problem in digital watermarking, traitor tracing, and group testing: the estimation of weak probabilities, especially false positive probabilities. The common point of these fields is test hypothesis: we compute a score function from the observations and compare it to a threshold. This threshold sets the trade-off between false positive and false negative probabilities. However, these are not well-balanced: the false positive probability is usually set to an extremely small value in the list of requirements. The expressions of the score function and the distribution of the observations are not simple which prevents us from deriving the distribution of the score. Of course, one can simplify (usually assuming a Gaussian distribution dut to the Central Limit Theorem) but this leads to wrong threshold values because weak probabilities implies a precise evaluation of the tail of

the score distribution. With Arnaud Guyader, Frédéric Cérou [48, 46], we design an Importance Splitting algorithm which gives accurate probability estimations and confidence intervals. The estimation of a probability P requires a number of trials in the order of $O(\log 1/P)$ which is much smaller than $O(1/P)$ for a crude Monte Carlo estimator.

This 'rare event' algorithm had serious impacts on the design of score functions (see Chap. 6). In traitor tracing for instance, the score function proposed by G. Tardos in his seminal paper [163, 164] (and modified later on by B. Skoric *et al.* [157]) had the advantage of being so simple that the author could 'easily' derive an upper bound of the false positive probability. Nevertheless, this property was constraining the design of the score function. Our algorithm estimates the correct value of the threshold (*i.e.* making the probability of false positive below the required level) for any score function. This frees its design and opens the door to more powerful decoders [32]. In this last paper, the algorithm was the keystone of benchmarking because it provides a fair way to compare the robustness of decoders against collusion attacks.

The power of this algorithm is so amazing that I tend to see any problem as a probability estimation issue. With Patrick Bas, we redefined the security of digital watermarking: a scheme is deemed secure if the attacker finds a 'suitable' key with extremely small probability [15, 2, 3, 4]. Rare events are everywhere.

Conclusions

A first conclusion is my encline to security which I tried to import to several signal / image processing applications: watermarking, traitor tracing, content based image retrieval, nearest neighbours search, recommendation system, and now adversarial machine learning. I can not see any personal root for this tropism. However, I have my way of envisaging security.

First, in these image / signal processing scenarios, security is a soft criterion. Contrary to cryptography, the question is not whether an attacker can or can not lead an attack. We usually do not prevent an attack ; so that the real question is how good this attack performs. I think that this is intimately bounded to the continuous nature of signals, in opposition to discrete mathematics at the heart of cryptography.

Second and consequently, signal processing can provide some security. There are many 'theorems' which assess that a given signal processing task (*i.e.* detection, estimation) is feasible if and only if some conditions are met. These theoretical results usually help exhibiting practical solutions in traditional applications (*i.e.* digital communication, direction of arrival, speech processing *etc*). My favorite trick is to use them the other way around: the attacker is unable to perform this task because the system is designed in such a way that these conditions are not met. However, such theorems are often asymptotical assessments (*i.e.* describing a phase transition), and in practice, the quantities at stake are finite. This is where tools like rare event analysis is useful: to measure quantities asymptotically vanishing, but in practice bounded away from zero.

Administrative facts

Transfer to the industry

I list here some transfers of knowledge to industrial partners about digital watermarking and traitor tracing

- Consultant for MovieLabs (common research lab of the six major motion picture studios),

- Consultant for IRT b-com,
- Co-funder of the startup Lamark,
- Sabbatical leave at Technicolor security lab (Sept. 2008 - Jan. 2010).

Fundings

I list here the projects where I have been involved

- 2 european projects: IST BUSMAN (digital watermarking), CHIST-ERA ID_IOT (identification and authentication)
- 4 national projects: Fabriano (watermarking security), Nebbiano (rare event - Project leader), Secular (security and information retrieval - Project leader), IDFraud (classification of scanned identity documents),

Supervising

- 7 Interns: Julie Josse, Sandrine Lesquin, Jonathan Delhumeau, Guillaume Stehlin, Philippe Roudot, Nicolas Basset, Bui Van Thach
- 1 Engineer: Mathieu Desoubeaux (video watermarking and traitor tracing)
- 5 Ph.D. students (co-supervising): Fuchun Xie (2007-2010: watermarking), Ana Charpentier (2008-2011: traitor tracing), Thanh-Toan Do (2009-2012: security of CBIR), Raghavendran Balu (2013-2016: system recommendation and privacy), Ahmet Iscen (2014-2017: image similarity search), Hanwei Zhang (2017-2020?: adversarial machine learning), Marzieh Gheisari (2018-2021?: authentication of PUF).
- 6 Postdoc researchers: François Cayre (2005 - watermarking security), Luis Perez-Freire (2009 - traitor tracing), Peter Meerwald (2012 - traitor tracing), Benjamin Mathon (2014 - nearest neighbor search security), Li Weng (2015 - nearest neighbor seach security), Ronan Sicre (2016 - classification of scanned identity documents)

Networking

- Co-chair of IH workshop (2007),
- Co-organizer of the international BOWS challenge version 2 (2008),
- Co-organizer of the national workshop “Privacy, similarity search and biometry” (2015)
- Member of the IEEE Technical committee on Information Forensics and Security
- Former associate editor of EURASIP Journal on Information Security, IET Journal on Information Security, Elsevier Digital Signal Processing journal, IEEE Transactions on Information Forensics and Security
- Reviewer for too many conferences and journals
- Jury member/reviewer of Ph.d. defenses: Antonino Simone (TU Eindhoven), Thijs Laarhoven (TU Eindhoven), Benjamin Mathon (Univ. Grenoble), Wei Fan (Univ. Grenoble), Mathieu Desoubeaux (Univ. Montpellier), Luis Perez-Freire (Univ. of Vigo), Pedro Comesana (Univ. of Vigo).

Teaching

Except some non regular short talks in engineer schools (Supelec, Telecom Brest, Ecole Normale Supérieure de Kerlann), I am teaching very few regular courses:

- Introduction to probability (40h), AgroCampus Rennes
- Rare events (20h), Insa Rennes
- Nearest neighbours search in high dimensional space (20h), ENS Rennes (with L. Amsaleg)

Publications

Here is the list of my publications (excluding ph.D. works) sorted by research topics. I thank all my co-authors.

Watermarking Security

[A1] M. Barni, F. Bartolini, and T. Furon. A general framework for robust watermarking security. *Signal Processing*, 83(10):2069–2084, October 2003. Special issue on Security of Data Hiding Technologies, invited paper.

[A2] P. Bas and T. Furon. Are 128 Bits Long Keys Possible in Watermarking? In B. Decker and D. W. Chadwick, editors, *13th International Conference on Communications and Multimedia Security (CMS)*, volume LNCS-7394 of *Communications and Multimedia Security*, pages 191–191, Canterbury, United Kingdom, Sept. 2012. Springer. Part 3: Extended Abstracts.

[A3] P. Bas and T. Furon. Key Length Estimation of Zero-Bit Watermarking Schemes. In *EUSIPCO - 20th European Signal Processing Conference*, page TBA, Romania, Aug. 2012.

[A4] P. Bas and T. Furon. A New Measure of Watermarking Security: The Effective Key Length. *IEEE Transactions on Information Forensics and Security*, 8(8):1306 – 1317, July 2013.

[A5] P. Bas, T. Furon, and F. Cayre. Practical Key Length of Watermarking Systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, Kyoto, Japan, Mar. 2012. IEEE.

[A6] P. Bas, T. FURON, F. Cayre, G. Doërr, and B. Mathon. *Watermarking Security*. Springer Briefs. springer, Jan. 2016.

[A7] F. Cayre, C. Fontaine, and T. Furon. Watermarking attack: Security of wss techniques. In I. Cox, T. Kalker, and H.-K. Lee, editors, *Proc. of Int. Workshop on Digital Watermarking*, volume 3304 of *Lecture Notes in Computer Science*, pages 171–183, Seoul, Corea, oct 2004. IWDW'04, Springer-Verlag. Best Paper Award.

[A8] F. Cayre, C. Fontaine, and T. Furon. A theoretical study of watermarking security. In *ISIT 2005*, pages 1868–1872. IEEE, 2005.

[A9] F. Cayre, C. Fontaine, and T. Furon. Watermarking security part I: Theory. In E. J. Delp and P. W. Wong, editors, *Proc. SPIE-IS&T Electronic Imaging, SPIE*, volume 5681, pages 746–757, San Jose, CA, USA, jan 2005. Security, Steganography, and Watermarking of Multimedia Contents VII.

[A10] F. Cayre, C. Fontaine, and T. Furon. Watermarking security part II: Practice. In E. J. Delp and P. W. Wong, editors, *Proc. of SPIE-IS&T Electronic Imaging, SPIE*, volume 5681, pages 758–768, San Jose, CA, USA, jan 2005. Security, Steganography, and Watermarking of Multimedia Contents VII.

[A11] F. Cayre, C. Fontaine, and T. Furon. Watermarking security: Theory and practice. *IEEE Trans. Signal Processing*, 53(10):3976 – 3987, oct 2005.

[A12] I. Cox, G. Doerr, and T. Furon. Watermarking is not cryptography. In Springer-Verlag, editor, *Proc. Int. Work. on Digital Watermarking, inivited talk*, L.N.C.S., Jeju island, Korea, nov 2006.

[A13] T. Furon. A survey of watermarking security. In M. Barni, editor, *Proc. of Int. Work. on Digital Watermarking*, volume 3710 of *Lecture Notes on Computer Science*, pages 201–215, Sienna, Italy, sep 2005. Springer-Verlag.

[A14] T. Furon. Watermarking security. In *Information hiding*, Information security and privacy series. Artech House, 2016.

[A15] T. Furon and P. Bas. A New Measure of Watermarking Security Applied on DC-DM QIM. In *IH - Information Hiding*, page TBA, Berkeley, United States, May 2012.

[A16] T. Furon, F. Cayre, and C. Fontaine. Watermarking Security. In N. Cvejic and T. Seppanen, editors, *Digital Audio Watermarking Techniques and Technologies*. Information Science Reference, 2008.

[A17] G. L. Guelvouit, T. Furon, and F. Cayre. The good, the bad, and the ugly: three different approaches to break their watermarking system. In E. Delp and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, page 650517. SPIE, jan 2007.

[A18] L. Pérez-Freire, F. Pérez-González, and T. Furon. On achievable security levels for lattice data hiding in the known message attack scenario. In *Proc. ACM Multimedia and Security*, pages 68–79, Geneva, Switzeland, sep 2006.

[A19] L. Pérez-Freire, F. Pérez-González, T. Furon, and P. Comesańa. Security of lattice-based data hiding against the Known Message Attack. *IEEE Trans. on Information Forensics and Security*, 1((4)):421–439, dec 2006.

[A20] F. Xie, T. Furon, and C. Fontaine. Better security levels for Broken Arrows. In S. I. T, editor, *Proc. of SPIE Electronic Imaging on Media Forensics and Security XII*, San Jose, CA, USA, 2010.

[A21] F. Xie, T. Furon, and C. Fontaine. Towards Robust and Secure Watermarking. In *ACM Multimedia and Security*, Roma, Italy, Sept. 2010.

Zero-bit watermarking

[B22] T. Furon. Hermite polynomials as provably good functions to watermark white gaussian hosts. In *Proc. ACM Multimedia and Security*, Geneva, Switzeland, sep 2006. ACM.

[B23] T. Furon. A constructive and unifying framework for zero-bit watermarking. *IEEE Trans. Information Forensics and Security*, 2(2):149–163, jun 2007.

[B24] T. Furon. About zero bit watermarking error exponents. In *ICASSP2017 - IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, United States, Mar. 2017. IEEE.

[B25] T. Furon and P. Bas. Broken arrows. *EURASIP Jounal on Information Security*, 2008(ID 597040):doi:10.1155/2008/597040, 2008.

[B26] T. Furon, J. Josse, and S. L. Squin. Some theoretical aspects of watermarking detection. In *Proc. Security, steganography and watermarking of multimedia content*, San Jose, CA, USA, jan 2006.

Traitor Tracing

[C27] A. Charpentier, C. Fontaine, and T. Furon. Decodage EM du code de Tardos pour le fingerprinting. In 22^{eme} *Colloque en Traitment du Signal et des Images (GRETSI)*, Dijon, France, septembre 2009.

[C28] A. Charpentier, C. Fontaine, and T. Furon. EM decoding of Tardos fingerprinting codes. *Traitement du Signal*, 27(2):127 – 147, 2010.

[C29] A. Charpentier, C. Fontaine, T. Furon, and I. Cox. An Asymmetric Fingerprinting Scheme based on Tardos Codes. In T. Filler, T. Pevný, S. Craver, and A. Ker, editors, *IH'11 - 13th International Conference Information Hiding*, volume 6958 of *LNCS - Lecture Notes in Computer Science*, Prague, Czech Republic, May 2011. Springer-Verlag.

[C30] A. Charpentier, F. Xie, C. Fontaine, and T. Furon. Expectation Maximisation decoding of Tardos probabilistic fingerprinting code. In N. Memon, editor, *Security, steganography and watermarking of multimedia contents*, San Jose, CA, USA, jan 2009. SPIE Electronic Imaging.

[C31] T. Furon. Le traçage de traîtres. In *Proc. of SSTIC, Symposium sur la Sécurité des Technologies de l'Information et des Communications*, Rennes, France, June 2009.

[C32] T. Furon and M. Desoubeaux. Tardos codes for real. In *IEEE Workshop on Information Forensics and Security*, page 7, Atlanta, United States, Dec. 2014. Yan Lindsay Sun and Vicky H. Zhao, IEEE.

[C33] T. Furon and G. Doërr. Tracing pirated content on the internet: Unwinding Ariadne's thread. *IEEE Security and Privacy Magazine*, 2010.

[C34] T. Furon, A. Guyader, and F. Cérou. On the design and optimisation of Tardos probabilistic fingerprinting codes. In *Proc. of the 10th Information Hiding Workshop*, LNCS, Santa Barbara, Cal, USA, may 2008.

[C35] T. Furon, A. Guyader, and F. Cérou. Decoding Fingerprinting Using the Markov Chain Monte Carlo Method. In *WIFS - IEEE Workshop on Information Forensics and Security*, Tenerife, Spain, Dec. 2012. IEEE.

[C36] T. Furon and L. Pérez-Freire. EM decoding of Tardos traitor tracing codes. In *Proc. of ACM Multimedia Security*, Princeton, NJ, USA, September 2009.

[C37] T. Furon and L. Pérez-Freire. Worst case attacks against binary probabilistic traitor tracing codes. In *Proceedings of First IEEE International Workshop on Information Forensics and Security*, pages 46–50, London, UK, December 2009. WIFS'09.

[C38] T. Furon, L. Pérez-Freire, A. Guyader, and F. Cérou. Estimating the minimal length of Tardos code. In *accepted to Information Hiding*, Darmstadt, Germany, jun 2009.

[C39] P. Meerwald and T. Furon. Group testing meets traitor tracing. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, volume accepted, 2011.

[C40] P. Meerwald and T. Furon. Iterative single Tardos decoder with controlled probability of false positive. In *International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011. IEEE.

[C41] P. Meerwald and T. Furon. Towards Joint Tardos Decoding: The 'Don Quixote' Algorithm. In S. C. A. K. T. Filler, T. Pevny, editor, *Information Hiding*, pages 28–42, Prague, Czech Republic, May 2011. Springer Berlin Heidelberg.

[C42] P. Meerwald and T. Furon. Towards practical joint decoding of binary Tardos fingerprinting codes. *Information Forensics and Security, IEEE Transactions on*, PP(99):1, 2012.

[C43] L. Pérez-Freire and T. Furon. Blind decoder for binary probabilistic traitor tracing codes. In *Proceedings of First IEEE International Workshop on Information Forensics and Security*, pages 56–60, London, UK, December 2009. IEEE WIFS'09.

[C44] F. Xie, C. Fontaine, and T. Furon. Un schéma complet de traçage de documents multimedia reposant sur des versions améliorées des codes de Tardos et de la technique de tatouage. In 22^{eme} *Colloque en Traitment du Signal et des Images (GRETSI)*, Dijon, France, septembre 2009.

[C45] F. Xie, T. Furon, and C. Fontaine. On-off keying modulation and Tardos fingerprinting. In *Proc. ACM Multimedia and Security*, Oxford, UK, September 2008. ACM.

Rare event

[D46] F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, pages 1–14, 2011. 10.1007/s11222-011-9231-6.

[D47] F. Cérou, T. Furon, and A. Guyader. Experimental assessment of the reliability for watermarking and fingerprinting schemes. *EURASIP Journal on Information Security*, (ID 414962):12 pages, 2008.

[D48] F. Cérou, P. D. Moral, T. Furon, and A. Guyader. Rare event simulation for a static distribution. In *Proc. 7th Int. Work. on Rare Event Simulation*, Rennes, France, September 2008.

[D49] T. Furon, C. Jégourel, A. Guyader, and F. Cérou. Estimating the probability of false alarm for a zero-bit watermarking technique. In *Proc. of IEEE Int. conf. on Digital Signal Processing*, Santorini, Greece, July 2009.

Content Based Image Retrieval Security

[E50] T.-T. Do, L. Amsaleg, E. Kijak, and T. Furon. Security-Oriented Picture-In-Picture Visual Modifications. In *ICMR - ACM International Conference on Multimedia Retrieval*, Hong-Kong, China, June 2012.

[E51] T.-T. Do, E. Kijak, L. Amsaleg, and T. Furon. Enlarging hacker's toolbox: deluding image recognition by attacking keypoint orientations. In *ICASSP - 37th International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012. IEEE.

[E52] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg. Challenging the security of content based image retrieval systems. In *Proc. of Int. Work. on Multimedia Signal Processing (IEEE MMSP)*. IEEE, 2010.

[E53] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg. Deluding image recognition in sift-based cbir systems. In *Proc. of ACM Workshop on Multimedia in Forensics, Security and Intelligence*, Firenze, Italy, october 2010. ACM.

[E54] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg. Understanding the security and robustness of sift. In *Proc. of ACM multimedia conference*, October 2010.

Approximate Nearest Neighbour Search

[F55] L. Amsaleg, O. Chelly, T. Furon, S. Girard, M. E. Houle, K.-I. Kawarabayashi, and M. Nett. Estimating Local Intrinsic Dimensionality. In *21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'15*, pages 29–38, Sidney, Australia, Aug. 2015. ACM, ACM.

[F56] R. Balu, T. Furon, and L. Amsaleg. Sketching techniques for very large matrix factorization. In *ECIR 2016 - 38th European Conference on Information Retrieval* , Proceedings of the European conference on Information Retrieval, Padoue, Italy, 2016.

[F57] R. Balu, T. Furon, and H. Jégou. Beyond "project and sign" for cosine estimation with binary codes. In *ICASPP - International Conference on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014. IEEE.

[F58] T. Furon, H. Jégou, L. Amsaleg, and B. Mathon. Fast and secure similarity search in high dimensional space. In *IEEE International Workshop on Information Forensics and Security*, Guangzhou, China, 2013.

[F59] A. Iscen, L. Amsaleg, and T. Furon. Scaling Group Testing Similarity Search. In *ACM International Conference on Multimedia Retrieval 2016* , New York, United States, June 2016.

[F60] A. Iscen, T. Furon, V. Gripon, M. RABBAT, and H. Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Transactions on Big Data*, 2017.

[F61] A. Iscen, M. RABBAT, and T. Furon. Efficient Large-Scale Similarity Search Using Matrix Factorization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, United States, June 2016.

[F62] H. Jégou, T. Furon, and J.-J. Fuchs. Anti-sparse coding for approximate nearest neighbor search. In *ICASSP - 37th International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012.

[F63] B. Mathon, T. Furon, L. Amsaleg, and J. Bringer. Recherche approximative de plus proches voisins efficace et sûre. In *GRETSI*, page ID238, Brest, France, Sept. 2013.

[F64] B. Mathon, T. Furon, L. Amsaleg, and J. Bringer. Secure and Efficient Approximate Nearest Neighbors Search. In A. Uhl, editor, *1st ACM Workshop on Information Hiding and Multimedia Security*, IH & MMSec '13, pages 175–180, Montpellier, France, June 2013.

[F65] M. Shi, T. Furon, and H. Jégou. A Group Testing Framework for Similarity Search in High-dimensional Spaces. In *ACM Multimedia*, Orlando, United States, Nov. 2014.

Computer Vision

[G66] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, United States, July 2017.

[G67] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Panorama to panorama matching for location recognition. In *ACM International Conference on Multimedia Retrieval (ICMR) 2017*, Bucharest, Romania, June 2017.

[G68] J. Krapac, F. Perronnin, T. Furon, and H. Jégou. Instance classification with prototype selection. In *ICMR - ACM International Conference on Multimedia Retrieval*, Glasgow, United Kingdom, Apr. 2014.

[G69] G. Tolias, A. Bursuc, T. Furon, and H. Jégou. Rotation and translation covariant match kernels for image retrieval. *Computer Vision and Image Understanding*, page 15, June 2015.

[G70] G. Tolias, T. Furon, and H. Jégou. Orientation covariant aggregation of local descriptors with embeddings. In *European Conference on Computer Vision*, Zurich, Switzerland, Sept. 2014.

Privacy and Differential Privacy

[H71] R. Balu and T. Furon. Differentially Private Matrix Factorization using Sketching Techniques. In *ACM WORKSHOP ON INFORMATION HIDING AND MULTIMEDIA SECURITY*, Vigo, Spain, June 2016.

[H72] R. Balu, T. Furon, and S. Gambs. Challenging differential privacy: the case of non-interactive mechanisms. In *European Symposium on Research in Computer Security*, volume 8657 of *LNCS*, Wroclaw, Poland, Sept. 2014. Springer-Verlag. Best Student Paper Award.

[H73] B. Bhattarai, A. Mignon, F. Jurie, and T. Furon. Puzzling Face Verification Algorithms for Privacy Protection. In Y. L. Sun and V. H. Zhao, editors, *IEEE Workshop on Information Forensics and Security*, Atlanta, United States, Dec. 2014. IEEE.

[H74] L. Weng, L. Amsaleg, and T. Furon. Privacy-Preserving Outsourced Media Search. *IEEE Transactions on Knowledge and Data Engineering*, 28(10), July 2016.

Miscellanea: Laws and DRM systems

[I75] J. Andreaux, A. Durand, T. Furon, and E. Diehl. Copy protection system for digital home networks. *IEEE Signal Processing Magazine*, 21(2):100–108, March 2004. Special Issue on Digital Right Management.

[I76] T. Furon. Les Mesures Techniques de Protection... autrement dit les DRM. In *Colloque PRIAM*, Grenoble, France, Nov. 2008.

[I77] T. Furon. *LES TECHNOLOGIES DE L'INFORMATION AU SERVICE DES DROITS : OPPORTUNITÉS, DÉFIS, LIMITES*, volume 32 of *Cahiers du Centre de Recherches Informatique et Droit*, chapter Les mesures techniques de protection. Bruylant, 2010.

[I78] T. Maillard and T. Furon. Towards digital rights and exemptions management systems. *Computer Law and Security Report*, 20(4):281–287, July 2004.

[I79] F. Pérez-González, T. Furon, M. Barni, P.Comesaña, L. Pérez-Freire, J. R. Troncoso, and M. Gauvin. Fair compensation for acts of copying. Response to the second Call for Comments launched by Internal Market and Services Directorate General, University of Vigo, 2008.

Part I

Traitor tracing with binary Tardos codes

Chapter 2

General overview

This chapter presents binary Tardos codes, which are the most well known probabilistic codes in traitor tracing. After detailing the application, the chapter introduces the main components of the problem: the code construction with the key generation, the collusion strategy, and the accusation based on a scoring function and thresholding mechanism. Each of these components will be the subject of the following chapters.

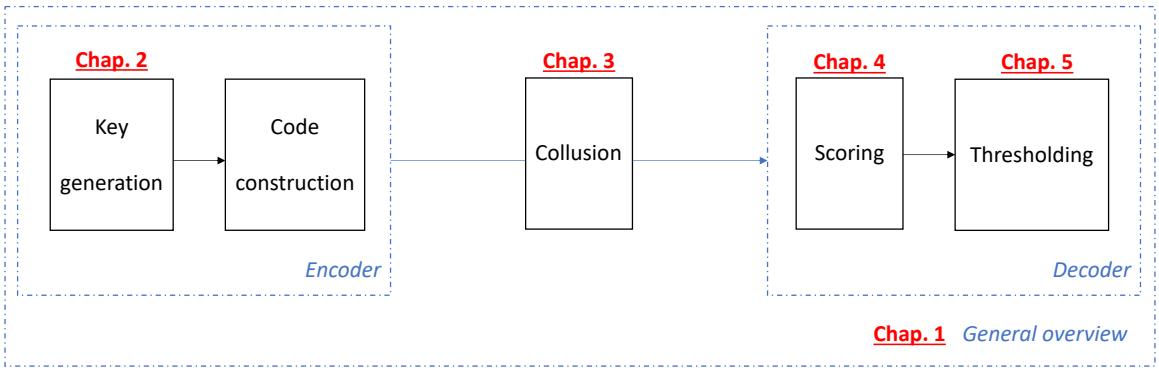


Figure 2.1: General overview of the Part I.

2.1 The application framework

A valuable document is distributed to a countable set $\mathcal{U}$ of n users, each labeled by an integer: $\mathcal{U} = [n]$ with $[n] := \{1, \ldots, n\}$. Users are not trusted and some of them might leak this document. Nothing can prevent this but a dissuasive means. Each user receives a personal copy of the content and, if a traitor leaks the document, this 'illegal' copy reveals his/her identity, exposing the traitor to severe proceedings. Traitor tracing studies the case where there are several traitors secretly cooperating to deceive the identification. This group of dishonest users is called a collusion.

A codebook of n unique m-symbol long codewords (a.k.a. identifiers or fingerprints) is first generated. The symbol belongs to an alphabet, which is the binary alphabet $\{0, 1\}$ in this thesis. Codeword $\mathbf{x}_j$ is associated to user j who receives a personal copy of the content because $\mathbf{x}_j$ has been embedded by a watermarking technique. We denote by $\mathcal{C} \subset \mathcal{U}$ the set of c dishonest users. They merge their personal copies to forge one pirated content. The watermarking decoder

extracts the pirated sequence $\mathbf{y}$ from this illegal content. The goal of the traitor tracing decoder is to identify one, some or all colluders from the pirated sequence $\mathbf{y}$.

2.1.1 The work of the cryptographers

Cryptographers were the first to work on this application.

They propose a model of the collusion attack. They model the content as a string of symbols where some of them can be changed without spoiling the regular use of the content. Only the code designer knows the location of these modifiable symbols, which he uses to insert the codeword. After receiving the content, the colluders can disclose symbols of their codewords in *detectable positions* by comparing their personal versions. This concept, known as the *marking assumption*, was invented by Boneh and Shaw [91].

The second step is to constrain what the colluders are able to do in these detectable positions. Denote by $\mathcal{C} = \{j_1, \dots, j_c\}$ the labels of the colluders. The *narrow-sense version* (or restricted digit model) imposes that $y_i \in \{x_{j_1,i}, \cdots, x_{j_c,i}\}$, $\forall i \in [m]$. In words: when the colluders' symbols are identical, the position is not detected and this common symbol value is decoded at this index in the pirated copy. Otherwise, they pick one of their symbols following a given collusion strategy. Another possibility is the *wide-sense version* (or arbitrary digit model): in detectable positions, colluders paste whatever symbol of the alphabet [83]. Narrow-sense and wide-sense are equivalent for binary codes studied in this document, therefore we omit the precision. A third variant, the *unreadable digit model*, allows the collusion to produce a small number of erasures in detectable positions [91]. A more rare collusion model, the *general digit model* a.k.a. *weak marking assumption*, allows few erasures [144] or random symbols [111] even in undetectable positions.

A second contribution of the cryptographic community is the establishment of desirable properties for traitor tracing codes [161]: frameproof, secure frameproof, identifiable parents property (IPP), and traceability. These properties describe categories of codes which are more and more powerful. For instance, IPP codes enable the identification of at least one colluder without framing any innocent. Traceability ensures the same guarantee with an efficient decoding algorithm. Yet, there is no binary code enforcing these properties for $c > 2$ [160, Lemma 1.6]. The alternative is to let the decoder make some errors provided these remain provably lower than some required levels [91].

2.1.2 Multimedia content

In a multimedia application, the content is modeled as a series of consecutive blocks (few seconds of audio, or a video scene) sequentially hiding the symbols of the codeword (see Fig. 2.2). The concept of detectable position *a priori* does not apply here: anyone knows that each block hides a symbol. However, watermarking is a secret keyed primitive. Without this secret key, we assume that it is impossible to neither modify nor erase the hidden symbol. One possible attack consists in swapping colluders' blocks, which enforces the marking assumption in the narrow-sense model: Sequentially creating a pirated content by copying and pasting one of their blocks amounts to forge a pirated sequence $\mathbf{y}$ compliant with the marking assumption in the narrow-sense model:

$$\forall i \in [m],\ y_i \in \{x_{j_1,i}, \cdots, x_{j_c,i}\}. \tag{2.1}$$

Sect. 2.4.3 introduces other models of collusion.

Fig. 2.2 shows a common solution in the industry (so-called 'DNA watermarking') because time consuming watermarking is done offline. For a binary code, the server stores two watermarked

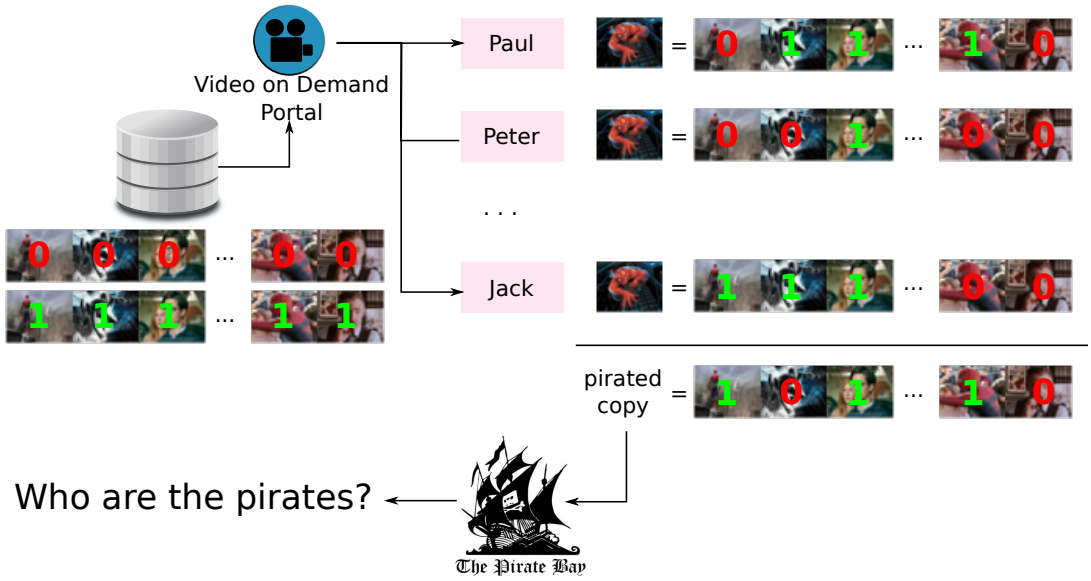


Figure 2.2: Sequentially embedding codeword into a movie. A thumbnail image pictures a video block, which is watermarked in two different ways to encode symbol '1' or '0'. The colluders sequentially recompose a pirated movie by selecting one of their blocks.

versions of a content block: one embedding symbol '1', the other embedding symbol '0'. Online distribution of a content is fast as it just has to sequentially pick the blocks according to the user's codeword [149]. The alternative approach is to watermark the content at the client side according to his/her codeword [123].

2.1.3 Probabilities of errors

The requirement of utmost importance is to avoid framing innocent users. The probability $\mathbb{P}_{\mathsf{FP}}$ of accusing at least one innocent user must be provably lower than a small level η_S. Fulfilling this requirement is called the soundness of the scheme.

The second error is to miss the identification of colluders. Theoretical works consider one of the three objectives below:

- 'Catch One': The goal is to identify at least one of the c colluders.
- 'Catch All': The goal is to identify all the colluders. This objective assumes that the colluders share evenly the risk of being caught. It is an illusion to identify all colluders when some of them have participated very little in the creation of the forged copy.
- 'Catch a colluder': The goal is to identify a given colluder.

The completeness of the code amounts to prove that the probability $\mathbb{P}_{\mathsf{FN}}$ of failing to achieve the objective is lower than a given level η_C. The two first objectives require to study the c events of accusing a colluder jointly. This is difficult because these events are not independent. The third objective is a relaxation of these problems which studies one event alone.

Note that in real life, there is no concrete or only a vague requirement about these objective. Level η_C should be low enough such that traitor tracing becomes dissuasive for the colluders. The

probability to miss colluders (under a given objective) provides a means to benchmark traitor tracing schemes in practice under the constraint that they all meet the requirement on η_S. Another benchmark measurement is the average number of identified colluders. This would correspond to a 'Catch Many' objective, whose goal is to identify as many colluders as possible (under the requirement on η_S).

2.2 The collusion strategy

The word *collusion* defines a group of people secretly conspiring to deceive other forces. The *collusion strategy* or *collusion attack* defines the process that colluders employ to craft forged content.

The descendance set $\mathsf{desc}(\mathbf{X}_\mathcal{C})$ is the set of all the pirated sequences $\mathbf{y}$ that collusion $\mathcal{C}$ can forge from their codewords $\mathbf{X}_\mathcal{C} := \{\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_c}\}$. The marking assumption (2.1) constrains this set, but its size remains huge: $|\mathsf{desc}(\mathbf{X}_\mathcal{C})| = 2^{m_d}$ for a binary code where $m_d < m$ is the number of detectable positions.

Another constraint is that the colluders certainly share the risk (unless there are traitors among the traitors [167]). A pirated sequence $\mathbf{y}$ is defined by the assignation sequence $\mathbf{a} = (a_1, \dots, a_m) \in \mathcal{C}^m$ which sequentially indicates which colluder is copying and pasting his/her block: $y_i = x_{a_i,i}$, $\forall i \in [m]$. Sharing the risk implies that $|\{i|a_i = j\}| \approx mc^{-1}, \forall j \in \mathcal{C}$. Another possibility is that the pirated sequence is not way closer to the codeword of one particular colluder: $d_H(\mathbf{y}, \mathbf{x}_j) \sim cst$, $\forall j \in \mathcal{C}$, with $d_H(.,.)$ the Hamming distance. If this is not the case, the accusation should more easily trace the closest colluder codeword. Indeed, unfair collusion process helps more in a 'Catch One' objective than it deludes the accusation.

2.2.1 Assumptions

My work on Tardos codes pushed a statistical model of the collusion strategy [34]. It is essentially based on four main assumptions.

- **Memoryless**: Since in a Tardos code, the symbols of the codewords are statistically independent, it seems relevant that the pirated sequence $\mathbf{y}$ also shares this property. Therefore, the value of y_i only depends on $\{x_{j_1,i}, \cdots, x_{j_c,i}\}$. These collusion strategies are called memoryless collusion channels in [136, Def. 2.4].

- **Invariance to permutation**: The colluders select the value of the symbol y_i depending on their symbols, but not on their order. There is no ordering within the colluders. The collusion channel is *invariant to permutation* of $\{x_{j_1,i}, \cdots, x_{j_c,i}\}$. Therefore, the input of the collusion process is indeed the type[1] of their symbols. In the binary case, this type is fully defined by the following sufficient statistic: The number σ_i of symbols '1', $\sigma_i := \sum_{j\in\mathcal{C}} x_{j,i}$ (the number of '0's being $c - \sigma_i$). These collusion strategies are called permutation-invariant collusion channel in [136, Def. 2.5].

- **Stationarity**: This assumes that the collusion strategy is independent of the index i in the sequence. Therefore, a collusion strategy is described forgetting the index i in the sequel.

- **Randomness**: The collusion process may not be deterministic, but *random*.

[1] Empirical distribution, ie. the histogram.

2.2.2 The model

These four assumptions yield that the attack of a collusion of size c is fully described by the following parameter vector: $\boldsymbol{\theta}_c := (\theta_{c,0}, \dots, \theta_{c,c})^\top \in [0,1]^{(c+1)}$, with

$$\theta_{c,\sigma} := \mathbb{P}\left(Y = 1 | \Sigma = \sigma\right). \tag{2.2}$$

This reads as the probability that the colluders put a symbol '1' in the pirated sequence when they have σ '1' over c symbols in their copies. The marking assumption enforces that $\theta_{c,0} = 0$ and $\theta_{c,c} = 1$. The authors of [81] also speak about 'eligible channel'. These collusion strategies are fair in the sense that the colluders share evenly the risks. To summarize, the colluders have $c+1$ biased coins. When they have σ symbols '1', they flip the σ-th coin to decide the value of the symbol in the pirated sequence. This defines the set $\boldsymbol{\Theta}_c$ of collusion strategies of size c as:

$$\boldsymbol{\Theta}_c := \{\boldsymbol{\theta}_c \in [0,1]^{(c+1)} | \theta_{c,0} = 1 - \theta_{c,c} = 0\}. \tag{2.3}$$

Here is a list of typical collusion strategies:

- **Interleaving Attack** (a.k.a. uniform attack): The colluders randomly draw who copy-pastes his/her block in the forged content, *i.e.* they randomly draw the assignation sequence: $A_i \overset{\text{i.i.d.}}{\sim} \mathcal{U}_{\{j_1,\dots,j_c\}}$. Therefore,

$$\theta_{c,\sigma} = \sigma/c, \quad \forall \sigma \in \{0, \dots, c\}. \tag{2.4}$$

- **Majority vote**: The colluders put the majority symbol in their hands:

$$\theta_{c,\sigma} = \begin{cases} 1, & \sigma > c/2 \\ 1/2, & \sigma = c/2 \quad (\text{if } c \text{ is even}) \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

- **Minority vote**: The colluders put the minority symbol in their hands:

$$\theta_{c,\sigma} = \begin{cases} 1, & 0 < \sigma < c/2, \text{ or } \sigma = c \\ 1/2, & \sigma = c/2 \quad (\text{if } c \text{ is even}) \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

- **Coin flip**: The colluders flip an unbiased coin to choose the symbol

$$\boldsymbol{\theta}_c = (0, 1/2, \dots, 1/2, 1)^\top. \tag{2.7}$$

- **All ones**: The colluders put a '1' wherever they can, *i.e.* $\sigma > 0$:

$$\boldsymbol{\theta}_c = (0, 1, \dots, 1)^\top. \tag{2.8}$$

- **All zeros**: The colluders put a '0' wherever they can, *i.e.* $\sigma < c$:

$$\boldsymbol{\theta}_c = (0, 0, \dots, 0, 1)^\top. \tag{2.9}$$

These are examples but indeed $\boldsymbol{\Theta}_c$ contains an infinite number of collusion strategies. Chapter 4 investigates this collusion model in more details.

2.3 Tardos codes

In this chapter, Tardos codes [163, 164] are seen as a broad family of codes having the same code construction. In other words, any modification about the parameters of the code construction or about the way to identify colluders compared to the seminal work of G. Tardos [163, 164] is not a reason for a change of name: These schemes all pertain to the family of Tardos codes.

2.3.1 Code construction

The code construction is composed of the three following steps:

- A random variable $P \in (0,1)$ is defined. It can be a discrete random variable with a set of possible outcomes $\mathcal{P} = \{\omega_i\}_{i=1}^K \subset (0,1)$ and associated probability mass function $\{f_i := \mathbb{P}(P = \omega_i)\}_{i=1}^K$ or an absolutely continuous random variable with probability density function $f(\cdot) : (0,1) \to \mathbb{R}^+$.
- The encoder first randomly draws m independent and identically distributed variables $P_1, \dots, P_m$ according to the above distribution. These occurences are stored in vector $\mathbf{p} := (p_1, \dots, p_m)^\top$ which is named the secret sequence.
- The encoder then randomly generates the codebook by independently drawing nm Bernoulli random variables s.t. $X_{j,i} \sim \mathcal{B}(p_i)$ for any user j. The codebook is denoted by a binary $m \times n$ matrix $\mathbf{X}$ and the codeword of user j by $\mathbf{x}_j = (x_{j,1}, \dots, x_{j,m})^\top$.

The distribution of P is public whereas vector $\mathbf{p}$ *is a secret* shared with the accusation algorithm. The codebook is composed of n private codewords in the sense that a user knows at most his/her codeword unless he/she is part of a collusion. By forming a collusion, the colluders may share the knowledge of their c codewords.

2.3.2 The distribution of the symbols in the pirated sequence

Chapter 4 connects the probabilistic code construction of a Tardos code with the model of collusion of Sect. 2.2.2. For the moment, we just introduce probabilities concerning the binary r.v. (random variable) $\{Y_i\}_{i=1}^m$: Define $\Pi(p_i) := \mathbb{P}(Y_i = 1|p_i)$ (dependence on $\boldsymbol{\theta}_c$ usually omitted unless there is an ambiguity). For a given index i, Y_i is a Bernoulli r.v.: $Y_i \sim \mathcal{B}(\Pi(p_i))$. Chapter 4 gives the expression of $\Pi(\cdot)$ in (4.6).

Thanks to the marking assumption, we have

$$\lim_{p\to 0} \Pi(p) = \theta_{c,0} = 0 \quad \text{and} \quad \lim_{p\to 1} \Pi(p) = \theta_{c,c} = 1. \tag{2.10}$$

2.3.3 Single linear decoders

Chapter 5 details several accusation algorithms. For the moment, this section considers a single linear decoder. To decide whether user j is a colluder, a score s_j is computed based on his codeword $\mathbf{x}_j$, the pirated sequence $\mathbf{y}$, and the code secret $\mathbf{p}$ as follows:

$$s_j = \sum_{i=1}^{m} U(x_{j,i}, y_i, p_i), \tag{2.11}$$

where $U(\cdot)\colon \{0,1\}\times\{0,1\}\times(0,1)\to\mathbb{R}$ is called the score function. This user is accused if $s_j > \tau$, where τ is the threshold. In other words, this accusation procedure tests two hypothesis per user: $\mathcal{H}_0$, user j is innocent *vs.* $\mathcal{H}_1$, user j is guilty. Denote by $\mathbb{P}_{\mathsf{fp}}$ and $\mathbb{P}_{\mathsf{fn}}$ the probabilities of false positive (accusing an innocent) and false negative (missing a colluder) of this test *per user*. The random variable modelling the score of an innocent (of a colluder) is denoted by S_{inn} (resp. S_{col}). Hence $\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(S_{\mathsf{inn}} > \tau)$ and $\mathbb{P}_{\mathsf{fn}} = \mathbb{P}(S_{\mathsf{col}} < \tau)$.

Soundness The soundness of the accusation procedure requires that probability $\mathbb{P}_{\mathsf{FP}}$ of accusing an innocent over the n tests is lower than η_S. Since the codewords of the innocent users are mutually independent (knowing $\mathbf{p}$) and independent of the pirated sequence, the outputs of these tests are independent as well. The requirement amounts to:

$$\mathbb{P}_{\mathsf{FP}} = \mathbb{P}\left(\cup_{j\notin\mathcal{C}}\{S_j > \tau\}\right) = 1 - \mathbb{P}\left(\cap_{j\notin\mathcal{C}}\{S_j < \tau\}\right) = 1 - (1-\mathbb{P}_{\mathsf{fp}})^{n-c} < \eta_S, \tag{2.12}$$

where $n-c$ is the number of innocent users. By proving that $\mathbb{P}_{\mathsf{fp}} < \eta_S/n$, requirement (2.12) is met. Soundness is challenging because we are facing a rare event: A setup commonly found in literature is $\eta_S \approx 10^{-3}$ and $n \approx 10^6$ making η_S/n in the order of 10^{-9}.

Completeness The completeness jointly considers the c scores of the colluders. This is complicated because colluders scores are not independent knowing $(\mathbf{y}, \mathbf{p})$. Note that the collusion strategy envisaged in this work ensures that the colluders share evenly the risk of being identified. In other words, $\mathbb{P}(S_j < \tau) = \mathbb{P}_{\mathsf{fn}}$, $\forall j \in \mathcal{C}$. Fig. 2.3 shows the case of $c = 2$ colluders scores.

The goal now is to link $\mathbb{P}_{\mathsf{fn}}$ to $\mathbb{P}_{\mathsf{FN}}$ for each objective.

- 'Catch One': Failing to accuse at least one colluder means that none of the colluders was identified:
$$\mathbb{P}_{\mathsf{FN}} = \mathbb{P}\left(\cap_{j\in\mathcal{C}}\{S_j < \tau\}\right) \leq \min_{j\in\mathcal{C}} \mathbb{P}(S_j < \tau) = \mathbb{P}_{\mathsf{fn}}. \tag{2.13}$$
This inequality shows that the requirement on the probability of failing to achieve the 'Catch One' objective is met if $\mathbb{P}_{\mathsf{fn}} < \eta_C$.

- 'Catch All': Failing to accuse all colluders means that at least one colluder was not identified:
$$\mathbb{P}_{\mathsf{FN}} = \mathbb{P}\left(\cup_{j\in\mathcal{C}}\{S_j < \tau\}\right) \leq \sum_{j\in\mathcal{C}} \mathbb{P}(S_j < \tau) = c\mathbb{P}_{\mathsf{fn}} \tag{2.14}$$
This inequality shows that the requirement on the probability of failing to achieve the 'Catch All' objective is met if $\mathbb{P}_{\mathsf{fn}} < \eta_C/c$.

- 'Catch a colluder': The probability of missing a given colluder is just $\mathbb{P}_{\mathsf{FN}} = \mathbb{P}_{\mathsf{fn}}$.

For a given $\mathbb{P}_{\mathsf{fn}}$, let us rank the objectives by their probability of failure in decreasing order: 'Catch All', 'Catch a colluder', 'Catch One'. This shows that 'Catch All' is the most difficult objective.

2.3.4 A typical article on Tardos codes

A typical article on Tardos codes starts with the following set of requirements:

- There are c colluders amongst n users,

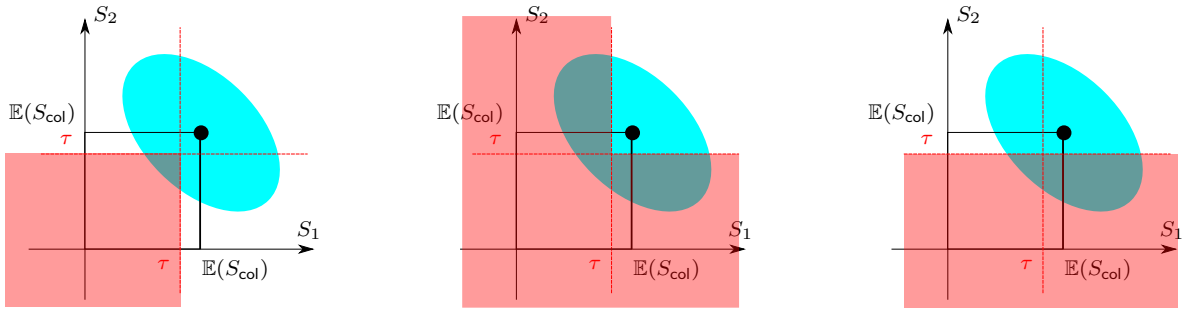


Figure 2.3: The colluders are users 1 and 2. The blue region region is the support of the density function of the random score vector (S_1, S_2) centered on the point $(\mathbb{E}(S_{\mathsf{col}}), \mathbb{E}(S_{\mathsf{col}}))$. The red region depicts the event of a failure for the three objectives: 'Catch One' (left), 'Catch All' (middle), and 'Catch a colluder' (right).

- Soundness: The probability of accusing an innocent, $\mathbb{P}_{\mathsf{FP}}$ must be lower than η_S,
- Completeness: The probability of failing to accuse colluders $\mathbb{P}_{\mathsf{FN}}$ (under one the three objectives) must be lower than η_C.

The main result of a typical article is the minimum code length $\underline{m}(n, c, \eta_S, \eta_C)$: if the code designer can not hide more than $\underline{m}(n, c, \eta_S, \eta_C)$ binary symbols in a content, then traitor tracing is vain as there is no guarantee that the requirements are fulfilled. *It is not recommended to play this game because the code designer is not sure to win.* The side-products of a typical article on 'Tardos code' are the parameters that lead to the derivation of $\underline{m}(n, c, \eta_S, \eta_C)$. For a single decoder as presented in Sect. 2.3.3, these parameters are $(P, U(\cdot), \tau)$.

2.3.5 The example of the Tardos-Škorić scheme

To illustrate this approach, this section details the choices made by G. Tardos [163, 164] and improved by B. Škorić *et al.* [154]. The proofs of soundness and completeness are as less technical as possible for the sake of simplicity. Assume that $\eta_C = 1/2$ for the 'Catch One' or 'Catch a colluder' objective so that it is sufficient to impose $\mathbb{P}_{\mathsf{fn}} < 1/2$.

The key idea is to design a score function such that the statistics of S_{inn} do not depend on the collusion strategy. The reason is that user scores are compared to a *universal* threshold τ. Universal here means that the threshold value is fixed independently of the collusion strategy, and of $(\mathbf{y}, \mathbf{p})$. The probability $\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(S_{\mathsf{inn}} > \tau)$ must be guaranteed whatever the size and the strategy of the collusion. In terms of statistics, the size c of the collusion and its strategy are deemed as *nuisance parameters* because they are unknown at the decoding side. Ideally, we aim at computing scores which are *pivotal quantities*, *i.e.* whose distribution does not depend on the nuisance parameters. In practice, only the first and second moments will be invariant.

For an innocent user, imposing that $\forall p \in (0,1)$, $\forall y \in \{0,1\}$, $\mathbb{E}(U(X_{\mathsf{inn}}, y, p)) = 0$ and $\mathbb{V}(U(X_{\mathsf{inn}}, y, p)) = 1$, determines the scoring function [34]:

$$U(1,1,p) = \sqrt{\frac{1-p}{p}}, \qquad U(0,0,p) = \sqrt{\frac{p}{1-p}},$$
$$U(0,1,p) = -\sqrt{\frac{p}{1-p}}, \qquad U(1,0,p) = -\sqrt{\frac{1-p}{p}}. \tag{2.15}$$

This score function has the merit of enforcing $\mathbb{E}(S_{\mathsf{inn}}) = 0$ and $\mathbb{V}(S_{\mathsf{inn}}) = m$ whatever the values of $\mathbf{p}$ (hence, the distribution of P) and $\mathbf{y}$ (hence, the collusion strategy).

The Bernstein inequality gives an upper bound of $\mathbb{P}_{\mathsf{fp}}$ [32], which is exponentially decreasing with τ. This is necessary to rediscover the code length of [163, 164]. Random variables $U(X_{j,i}, y_i, p_i)$ are independent (thanks to the code construction and the assumption on the collusion strategy), with zero expectation and unit variance, and $\forall i \in [m]$, $|U(X_{j,i}, y_i, p_i)| < M$ with $M = \max(\{p_i^{-1/2}\} \cup \{(1-p_i)^{-1/2}\})$. Then, $\forall \tau > 0$,

$$\mathbb{P}_{\mathsf{fp}} \leq e^{-\frac{3\tau^2}{6m+2M\tau}}. \tag{2.16}$$

This bound holds for a given secret sequence because M depends on $\mathbf{p}$. It might be a great tool to set the threshold in practice, but it fails delivering a universal result. A solution is to introduce a cutoff parameter $0 < t < 1/2$ which is used to clip the score function[2]. The clipped score function is given by:

$$\bar{U}(x, y, p) = \begin{cases} U(x, y, p) & \text{if } t < p < 1 - t \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

The first consequence of this clipping is that we can now set M to $1/\sqrt{t}$. Eq. (2.16) is now an upper bound valid for any $(\mathbf{p}, \mathbf{y})$.

As for the score of a colluder, by using the model of the collusion strategy of Sect. 2.2.2, Chapter 4 shows that:

$$\mathbb{E}(\bar{U}(X_{\mathsf{col}}, Y, p)) = \begin{cases} \frac{2}{c}\sqrt{p(1-p)}\Pi'(p), & \text{if } t < p < 1 - t \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

where $\Pi'(\cdot)$ is the derivative of $\Pi(\cdot)$ (4.6) w.r.t. p. The choice made in [163] for the absolutely continuous random variable P with density

$$f^{(T)}(p) = \frac{1}{\pi\sqrt{p(1-p)}}, \quad \forall 0 < p < 1, \tag{2.19}$$

makes

$$\mathbb{E}\left(S_{\mathsf{col}}\right) = m\mathbb{E}\left(U(X_{\mathsf{col}}, Y, P)\right) = \frac{2m}{\pi c}\int_t^{1-t} \Pi'(P)dp = \frac{2m}{\pi c}\left(\Pi(1-t) - \Pi(t)\right). \tag{2.20}$$

For any collusion strategy following the model of Sect.2.2.2, $\mathbb{E}\left(S_{\mathsf{col}}\right) \to 2m/\pi c$ as $t \to 0$. Indeed, we have $2(1-t)^c - 1 < \Pi(1-t) - \Pi(t) \leq 1$.

Score S_{col} is a sum of m independent r.v. with finite variance. The law of S_{col} tends to a Gaussian distribution as $m \to \infty$ by the Central Limit Theorem. We prefer to state that mean and median are asymptotically equal: $\mathbb{P}(S_{\mathsf{col}} < \mathbb{E}(S_{\mathsf{col}})) \approx 1/2$ (A more precise statement of completeness resorts to a bound of the probability that S_{col} deviates from its expectation, like the Chebychev inequality for instance). This makes $\mathbb{P}_{\mathsf{fn}}$ less than $1/2$ if τ is lower than $\mathbb{E}(S_{\mathsf{col}})$.

All these arguments packed together end up with a constraint on the code length m. With a probability above $1/2$ (assuming this is dissuasive enough), a given colluder gets caught (whatever the collusion strategy) if $2m/\pi c(2(1-t)^c - 1) = \tau$. At the same time the probability of accusing

[2] G. Tardos and his followers use the cutoff to clip the density f of P at the encoding, but I believe this way eases the derivation.

at least one innocent is lower than η_S if $e^{-\frac{3\tau^2}{6m+2M\tau}} \leq \eta_S/n$. These two requirements constrain the code length:

$$m \geq \frac{\pi^2 c^2}{2} \log\left(\frac{n}{\eta_S}\right) G(t,c), \tag{2.21}$$

with

$$G(t,c) := \frac{1}{(2(1-t)^c-1)^2}\left(1+\frac{2}{3\pi}\frac{2(1-t)^c-1}{c\sqrt{t}}\right). \tag{2.22}$$

Making t dependent on c allows to bound the quantity $G(t,c)$. For instance, define $t_c := 1 - ((h+1)/2)^{1/c}$ (s.t. $2(1-t_c)^c - 1 = h$ with $0 < h < 1$) to obtain $G(t_c, c) \leq 2$ for $h = 0.9$. In the end, the two requirements are met if $m > \underline{m}(n, c, \eta_S, 1/2)$ with:

$$\underline{m}(n, c, \eta_S, 1/2) = \pi^2 c^2 \log\left(\frac{n}{\eta_S}\right). \tag{2.23}$$

This is however an asymptotical result. In other words, $m = \Omega(c^2 \log n/\eta_S)$. Note that this ultra-simple rationale finds a constant which is only twice bigger than the best result $\pi^2/2$ (asymptotically valid when $c \to \infty$ [154]). Also, the strength of the collusion strategy is gauged by its ability to decrease $\mathbb{E}(S_{\mathsf{col}})$ proportional to the quantity $\Pi(1-t_c) - \Pi(t_c)$. Indeed, for a fixed collusion size, the minority attack is the worse w.r.t. to this criteria. The same conclusion holds in paper [154]. As a last comment, t_c is small: $t_c < 0.01$, $\forall c \geq 6$.

2.3.6 Pitfalls

There are three pitfalls in the above demonstration.

Too restrictive This pedagogical study is simple but only works for $\eta_C = 1/2$ and asymptotically as $n \to \infty$. A more useful proof needs an upper bound of $\mathbb{P}_{\mathsf{fn}}$. This usually requires the expression of $\mathbb{V}(S_{\mathsf{col}})$, which is upper bounded by m because $\mathbb{E}(\bar{U}(X_{\mathsf{col}}, Y, p)) = 1$, $\forall p \in [t, 1-t]$. Even a loose inequality such as the Chebyshev bound gives a relevant result: Suppose that $\mathbb{P}_{\mathsf{fn}}$ is required to be lower than ϵ, then for any n:

$$\underline{m}(n, c, \eta_S, \epsilon) = \pi^2 c^2 \left(\log\frac{n}{\eta_S} + \frac{1}{3\epsilon} + O\left(\sqrt{\frac{1}{\epsilon}\log\frac{n}{\eta_S}}\right)\right). \tag{2.24}$$

The pressure is on the soundness part where the tightest bound has to used, whereas completeness is a less stringent constraint.

Ill-posed problem This drawback is common to any theoretical work on traitor tracing which starts from the set of requirements listed in Sect. 2.3.4. The collusion size c is part of the requirements and so that the design depends on c. This especially concerns the parameters (m, t, τ). One way to circumvent this pitfall is to suppose that $c \leq c_{\max}$. This makes the scheme oblivious to c. Soundness is always achieved, but completeness is proven under the assumption that the true collusion size is below or equal to $c_{\max}$.

Involved score function This is a drawback related to the score function $U(\cdot)$ (2.15). The design of the score function is too constrained: It provides distinguishability (the score of a colluder is statistically higher than the score of an innocent) and independence w.r.t. to the collusion strategy (the distributions of S_{inn} and S_{col} are almost fixed, at least up to their first two moments (approximately for S_{col})). The second point is key to find an universal threshold thanks to (2.16).

Uncoupling distinguishability and independence to the nuisance parameters opens the door to more discriminative score functions (see Chap. 5). Yet, the setting of the threshold becomes a crucial problem that Chap. 6 is solving.

2.4 The followers of G. Tardos

Many papers have been published about Tardos codes. To make an overview of this literature, this section first summarizes known results on the code length in the very same setup as Sect. 2.3.5, then it lists other figures of merit and extensions.

2.4.1 Code length for binary Tardos codes

The first idea appearing in the literature is that soundness and completeness was originally proven for a code length $m = 100c^2 \log n/\eta_S$ [163]. That constant 100 is 'strange', but understandable: a code length scaling as $c^2 \log(n)$ was the Holy Graal at that time. G. Tardos never claimed to come up with the sharpest estimation.

The most well-known improvement comes from B. Škorić *et al.* [154] who 'symmetrized' the initial score function of G. Tardos, reducing the code-length by a factor of two. Nowadays, this score function (2.15) is the baseline.

The following works keep the same definition of P and $U(\cdot)$ as in [154, 88, 127] but develop a finer analysis to get the smallest constant. The state of the art w.r.t. to this constant $\kappa := m/c^2 \log n/\eta_S$ is the following, assuming that completeness is not an issue:

- Asymptotically as $c \to \infty$, $\kappa \to \pi^2/2 \approx 4.93$ [154],
- For a large collusion size, $\kappa = \pi^2/2 + O(c^{-1/3})$ [127],
- For $\forall c > 2$, $\kappa = 10.89$.

2.4.2 Figures of merit

The code length The ultimate figure of merit is the code length m. The authors of an article about Tardos code made a better job if they prove that, by their clever choice on P and $U(\cdot)$, and / or by their skill in bounding probabilities of errors, the code length necessary to fulfill the requirements (c, n, η_S, η_C) is lower than the state-of-the-art. This figure of merit is used in [163, 164, 88, 127].

Signal to Noise power Ratio An alternative states that finding the correct value of τ is a minor issue. One assumes that there is a work-around to set the threshold (maybe specific for a given $\mathbf{y}$ and $\mathbf{p}$). Replacing τ by $\mathbb{E}(S_{\mathsf{col}})$, we look for this condition: $\mathbb{P}(S_{\mathsf{inn}} > \mathbb{E}(S_{\mathsf{col}})) = \eta_S/n$, which merely says that an innocent user should not have a score as high as the typical score of a colluder. A coarse approximation is to pretend that, for a single linear decoder, S_{inn} is Gaussian distributed because it is the sum of a large number of independent random variables of finite variance. Then $m \approx \rho^{-1} \left(\Phi^{-1}(1 - \eta_S/n)\right)^2$, where ρ is similar to a Signal to Noise power Ratio per code index at the output of the score function:

$$\rho := \frac{(\mathbb{E}(U(X_{\mathsf{col}}, Y, P)) - \mathbb{E}(U(X_{\mathsf{inn}}, Y, P)))^2}{\mathbb{V}(U(X_{\mathsf{inn}}, Y, P))}. \tag{2.25}$$

This also corresponds to the double of the Kulback-Leibler divergence between $\mathcal{N}(\mathbb{E}(S_{\mathsf{col}}); \mathbb{V}(S_{\mathsf{col}}))$ and $\mathcal{N}(\mathbb{E}(S_{\mathsf{inn}}); \mathbb{V}(S_{\mathsf{inn}}))$. This figure of merit is used in [157, 154, 34]. One particular attention looks for the worst case attack defined as the one minimizing ρ. This indicates the required code length to fight any collusion attack including the worst case. Note that this Central Limit Theorem approximation is here to motivate ρ as a figure of merit (the higher, the better) but it cannot be used for a proof of soundness.

Achievable rates A third figure of merit is the achievable rate of single decoders $R^{(S)}(P; \boldsymbol{\theta}_c)$ defined as the mutual information $I(Y; X_{\mathsf{col}}|P, \boldsymbol{\theta}_c)$ between the symbols of the pirate sequence and of the codeword of a colluder knowing the secret sequence. Chapter 4 gives more details on this information theoretical quantity defined in (4.19). Considering that the decoder is testing two hypothesis per user (user j is innocent or guilty), this quantity is the supremum over all single decoders of the error exponent E_{fp} of the probability of accusing a given innocent (for a fixed false negative probability η_C):

$$E_{\mathsf{fp}} := -\lim_{m\to\infty} \frac{1}{m} \log \mathbb{P}_{\mathsf{fp}}. \tag{2.26}$$

If not null, $\mathbb{P}_{\mathsf{fp}}$ may vanish as $m \to \infty$ as fast as $e^{-mR^{(S)}(P;\boldsymbol{\theta}_c)}$ (indeed this is an upper bound thanks to the Chernoff bound - See App. 15 - of the best single decoder) so that $\mathbb{P}_{\mathsf{fp}} \leq \eta_S/n$ is enforced if

$$m \geq \frac{1}{R^{(S)}(P; \boldsymbol{\theta}_c)} \log \frac{n}{\eta_S}. \tag{2.27}$$

This makes the quantity $R^{(S)}(P; \boldsymbol{\theta}_c)$ a figure of merit (the higher the better). Note that this is a theoretical result: the best score function achieving this error exponent is given by the Neyman-Pearson lemma (see Sect. 5.2.1). Yet, its score function is the log likelihood ratio whose implementation (5.4) needs the expressions of $\mathbb{P}(Y = y, X_{\mathsf{col}} = x|p, \boldsymbol{\theta}_c)$ and $\mathbb{P}(Y = y|p, \boldsymbol{\theta}_c)$. This is a big issue because these expressions depend on the collusion strategy $\boldsymbol{\theta}_c$ a priori unknown to the decoder. On the other hand, any single decoder has an error exponent lower than $R^{(S)}(P; \boldsymbol{\theta}_c)$. Therefore a collusion strategy lowering this quantity 'hurts' any single decoder.

This figure of merit has been generalized to joint decoders (see Sect. 4.3.2), which compute a score per group of ℓ users. Indeed, when $\ell = c$, this leads to the concept of capacity (see Sect. 4.5).

2.4.3 Extensions

The literature since G. Tardos seminal article [163] proposed several extensions grouped in two families:

Binary codes

The idea is to keep Tardos binary code construction while the pirated sequence $\mathbf{y}$ is no longer binary. This is relevant for multimedia applications where the symbols are embedded in content thanks to a watermarking scheme. The communication channel provided by watermarking is not perfect especially if the colluders degrade the quality of the forged content (with a lossy video compression, for instance) after the collusion process. Erasures (denoted by symbol $\times$) or decoding errors (decoding a '1' whereas '0' was hidden in the content block) may occur violating the marking assumption.

The collusion strategy itself might be more complex than copy-pasting content blocks. The colluders may also merge blocks '1' and '0' by some signal processing [145]. The most obvious merging being the averaging of the pixel values for image or video blocks. This may result in

decoding a symbol '1' or '0', an erasure $\times$ (like in the *unreadable digit model* [91, 117] or the *general digit model* [144]), or a double detection (denoted by d): the watermark decoder is able to detect a merge of blocks hiding symbols '1' and '0'. In other words, the pirated sequence is no longer binary: $\mathbf{y} \in \{0, 1, \times, \mathsf{d}\}^m$. This extension is discussed in more details in Sect. 4.6.

Another extension is to work with a watermarking decoder which outputs soft informations [125] or [42, Sect. V.B], like the likelihoods $(l_{1,i}, l_{1,0})$ that the i-th block contains the symbols '1' and '0'. This thesis does not consider this extension.

From the theoretical point of view, the difficulty is to extend the marking assumption: We must limit the power of the colluders by constraining the family of collusion strategies in order to have a chance to identify colluders. For instance, if $\theta_{c,\sigma}(\times) = 1$, $\forall \sigma \in \{0, \dots, c\}$, then the pirated sequence is just a series of erasures pulling down any traitor tracing mechanism. From the practical point of view, the difficulty is to design score functions taking into account these new symbols, $\times$ and d, [43, Sect. 4] or soft outputs [42, Sect. V.B].

q-ary codes

Another extension proposes code constructions giving codewords defined on larger alphabet: $\mathbf{x}_j \in \{0, 1, \dots, q-1\}^m$. The marking assumption remains the same: when the colluders share the same symbol, this symbol is decoded in the pirated sequence (some articles foresee a limited number of violations as in the *general digit model*). The difficulty is to model the collusion strategy in the detectable positions. Two examples are the *wide-sense version* (a.k.a. arbitrary digit model) [83] and the *narrow-sense version* of the marking assumption.

The generalization of the model $\boldsymbol{\theta}_c$ to q-ary codes was never envisaged as its size is too big: $\boldsymbol{\theta}_c$ stores the probabilities $\mathbb{P}(y|\mathbf{t})$ for $y \in \{0, 1, \dots, q-1\}$ and $\mathbf{t}$ the type (a.k.a. empirical distribution, histogram, or tallies) of the colluders' symbols. It happens that the number of types of c-long sequences over an alphabet of size q equals $\binom{c+q-1}{q-1}$, *i.e.* $O(c^{q-1})$. More tractable models of the collusion strategy are indeed used [89, 90, 158, 117].

2.5 Conclusion

This chapter introduces Tardos codes in its original version. This study has been simplified for the sake of pedagogy. The following chapters present advances with respect to the choice of the random variable P, the collusion attack and the accusation procedure.

Chapter 3

Key generation

This chapter looks at improvements on the code construction since the seminal work of G. Tardos. The first section considers the very same score function so that the only degree of freedom is the distribution of random variable P. These optimized distributions from [34] or [139] yields an advantage only for a small collusion size c. Most importantly, they provide a rationale of the original choice of G. Tardos (which was not motivated in [163, 164]). The second section enlarges the study to figures of merit related to information theory.

3.1 Better distributions of P

This section keeps the same framework as in the previous chapter: the code construction, the single decoder, and the same figure of merit as in the previous chapter (see Sect. 2.3.5). The score function (2.15) enforces that $\mathbb{E}(S_{\mathsf{inn}}) = 0$ and $\mathbb{V}(S_{\mathsf{inn}}) = m$ whatever the distribution of P, the collusion size, and the collusion strategy. According to Sect. 2.3.5, the goal is now to look for a r.v. P which makes $\mathbb{E}(S_{\mathsf{col}})$ independent of the collusion strategy. The only modification here concerns the definition of the random variable P.

The expectation of a colluder score has the following expression:

$$\mathbb{E}(S_{\mathsf{col}}) = m\mathbb{E}\left(\sum_{(y,x)\in\{0,1\}} \bar{U}(x,y,P)\mathbb{P}(Y=y|X_{\mathsf{col}}=x,P)\mathbb{P}(X_{\mathsf{col}}=x|P)\right). \tag{3.1}$$

The codeword of a colluder has the same distribution as any codeword: $\mathbb{P}(X_{\mathsf{col}} = x|P) = P^x(1-P)^{1-x}$. Using the score function (2.15) and neglecting the cutoff parameter t, we obtain $\mathbb{E}(S_{\mathsf{col}}) \approx 2m\mathbb{E}\left(\sqrt{P(1-P)}(\mathbb{P}(Y=1|X_{\mathsf{col}}=1,P) - \mathbb{P}(Y=1|X_{\mathsf{col}}=0,P))\right)$. For the collusion model explained in Sect. 2.2.2, the difference of these conditional probabilities appears to equal $c^{-1}\Pi'(P)$ (see Sect. 4.2.4), the derivative of the polynomial $\Pi(P) = \mathbb{P}(Y=1|P)$. This leads to:

$$\begin{aligned}\mathbb{E}(S_{\mathsf{col}}) &\approx \frac{2m}{c}\mathbb{E}_P\left(\sqrt{P(1-P)}\Pi'(P)\right) &(3.2)\\ &\approx 2m\mathbb{E}_P\left(\sqrt{P(1-P)}P^{c-1}\right) - \frac{2m}{c}\sum_{\sigma=1}^{c-1}\theta_{c,\sigma}\binom{\sigma}{c}I_{c,\sigma}, &(3.3)\end{aligned}$$

where $I_{c,\sigma} := \mathbb{E}_P(\sqrt{P(1-P)}H_{c,\sigma}(P))$ and $H_{\sigma,c}(p) := p^{\sigma-1}(1-p)^{c-\sigma-1}(cp-\sigma)$. This shows that $\mathbb{E}(S_{\mathsf{col}})$ gets approximately independent of $\boldsymbol{\theta}_c$ if the expectations $\{I_{c,\sigma}\}_{\sigma=1}^{c-1}$ cancel out.

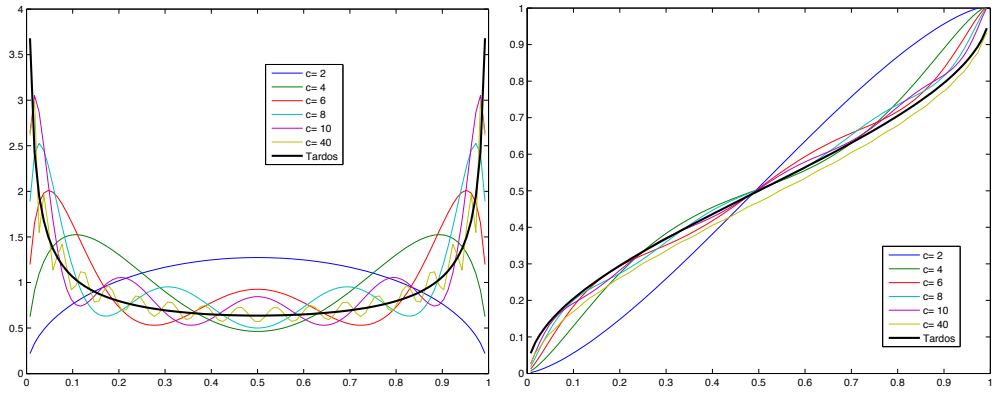


Figure 3.1: (left) Plots of the distribution $f_c^\star$ for $c \in \{2, 4, 6, 8, 10, 20\}$ and the Tardos distribution $f^{(T)}$ (2.19); (right) Plots of the corresponding cumulative density functions.

The choice of G. Tardos (2.19) implements this idea, but we can find other distributions when the size of the collusion c is known at the code construction. This means that the random variable P depends on c. In practice, the code designer bets on a maximum collusion size $c_{\max}$.

3.1.1 Absolutely continuous random variable P

Paper [34] applies this approach and looks for an absolutely continuous random variable P. In this case, expectation $I_{c,\sigma}$ is an integral over $(0,1)$ which is seen as a scalar product between polynomial $H_{\sigma,c}(\cdot)$ and $p \mapsto \sqrt{p(1-p)}f(p)$, with f the density of P. It happens that the family of polynomials $\{H_{\sigma,c}(\cdot)\}_{\sigma=1}^{c-1}$ spans the subspace $\mathcal{P}_c$ of polynomials of degree equal or less than $(c-1)$ and whose integral over $(0,1)$ is null. This tells that expectations $\{I_{c,\sigma}\}$ are null for a density f s.t. $p \mapsto \sqrt{p(1-p)}f(p)$ is orthogonal to the subspace $\mathcal{P}_c$. A simple way is to decompose this function as a series of shifted Legendre polynomials. Since these $(c-1)$ first polynomials form a basis of $\mathcal{P}_c$, the first elements of the series should be zero: $\mathbb{E}(S_{\mathsf{col}})$ is now approximately independent of $\theta_{c,\sigma}$, $\forall 0 < \sigma < c-1$. At last, the remaining terms of the series are chosen in order to maximize $\mathbb{E}(S_{\mathsf{col}})$ [34]. In the end, the best density is

$$f_c^\star(p) = \left(\pi\left(1 - 2^{-4\lfloor c/2\rfloor}\binom{2\lfloor c/2\rfloor}{\lfloor c/2\rfloor}^2\right)\right)^{-1} \frac{1 - P_{2\lfloor c/2\rfloor}^{\mathcal{SL}}(p)}{\sqrt{p(1-p)}}, \tag{3.4}$$

where $P_k^{\mathcal{SL}}(\cdot)$ is the shifted Legendre polynomial of order k. The corresponding expectation of a colluder's score is given in Table 3.1. The improvement compared to Tardos' choice quickly vanishes as c increases. This is not surprising as $f_c^\star(p)$ converges to $f^{(T)}(p)$ (2.19) for any $p \in (0,1)$, as shown in Fig 3.1.

3.1.2 Discrete random variable P

K. Nuida *et al.* followed the same road, but looking for a *discrete random variable* P maximizing $\mathbb{E}(S_{\mathsf{col}})$ with the constraint of canceling the above-mentioned expectations $\{I_{\sigma,c}\}_{\sigma=1}^{c-1}$ (what they call c-indistinguishability in [139, Def. 1]). A nice relationship is established with quadrature systems [139, Th. 1]. A natural choice is the quadrature of the Gauss-Legendre system based on the Legendre polynomials (see [137, Sec. 3.5(v)]). To fight against c colluders, the nodes are the roots $\{\xi_k\}$ of the $\lceil c/2\rceil$-th Legendre polynomial that we need to shift in between $[0,1]$ and ψ_k are

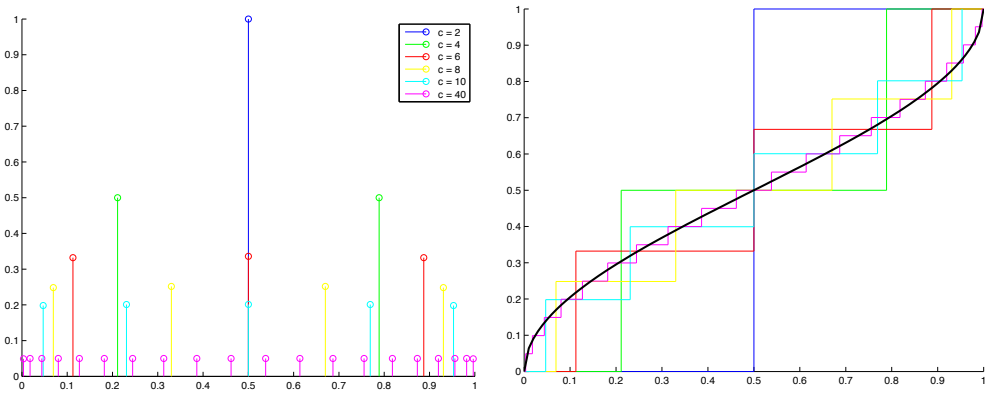


Figure 3.2: (left) Plots of the discrete distribution $f_c^\star$ for $c \in \{2, 4, 6, 8, 40\}$ recommended by K. Nuida *et al.* [139]; (right) Plots of the corresponding cumulative density functions.

the weights of the quadrature modified according to [139, Th. 1]:

$$f_c^\star : \{(\omega_k, f_k)\} = \{(\xi_k + 1)/2, C^{-1}\psi_k(1 - \xi_k^2)^{-1/2}\}. \tag{3.5}$$

Values of $\{(\xi_k, \psi_k)\}$ for some quadrature degrees are listed in [137, Sec. 3.5(v)]. Constant C is defined such that $\sum_k f_k = 1$. Fig. 3.2 illustrates Nuida's discrete distributions. The convergence of the probability distribution as c increases to the Tardos' distribution (whose c.d.f. is the shifted arcsine function) is not surprising:

$$\mathbb{P}(P < x) = C \sum_{\omega_k < x} \frac{\psi_k}{\sqrt{\omega_k(1-\omega_k)}} = \frac{\int_0^x \frac{1}{\sqrt{p(1-p)}} dp - R_c}{\pi - R'_c} \overset{c\to\infty}{\to} \frac{1}{\pi} \arcsin(2x-1) + \frac{1}{2}. \tag{3.6}$$

where R_c and R'_c are the residual errors of the quadrature which tends to zero as c increases.

The expectation of a colluder's score is proportional to m/c. Tab. 3.1 compares the constant for the optimal distribution in the continuous and discrete r.v. setups. The choice of K. Nuida *et al.* produces better improvements.

Most importantly, these works show that when i) the single decoder is based on the score function (2.15) (which fixes the first two moment of S_{inn}) ii) $\mathbb{E}(S_{\mathsf{col}})$ is approximately invariant to the collusion strategy $\boldsymbol{\theta}_c$ for *any* c (neglecting the effect to the cutoff parameter), then the distribution $f^{(T)}$ defined in (2.19) (a.k.a. Tardos distribution or arcsine distribution) is optimal as it maximizes the Signal to Noise power ratio at the output of the score function.

$m^{-1}c\mathbb{E}(S_{\mathsf{col}})$ / c	2	4	6	8	10
Continuous (Sect. 3.1.1)	0.85	0.74	0.71	0.69	0.68
Discrete (Sect. 3.1.2)	1.0	0.82	0.76	0.73	0.71

Table 3.1: Best expectations when the size of the collusion is known (cut-off parameter t being neglected). G. Tardos' solution yields $m^{-1}c\mathbb{E}(S_{\mathsf{col}}) = 2/\pi = 0.637$.

3.1.3 Approximated optimal discrete distributions

The discrete distributions (3.5) have quite an involved definition. T. Laarhoven [128] proposes the following approximation:

$$\hat{f}_c^\star : \{\omega_k, f_k\}_{k=1}^{\lceil c/2 \rceil} = \left\{ \lceil c/2 \rceil^{-1}, \sin^2\left(\frac{4k-1}{8\lceil c/2 \rceil + 4}\pi\right) \right\}_{k=1}^{\lceil c/2 \rceil}. \tag{3.7}$$

This approximation produces $\mathbb{E}(S_{\mathsf{col}})$ slightly lower than the optimal distribution of K. Nuida. Yet, this expectation is no longer invariant to the collusion strategy, unless c is large. This distribution also converges to Tardos distribution as $c \to \infty$.

3.2 Recent views on the optimal distributions

The criterion of optimality of the distribution of the secret bias P has long been the maximization of the Signal to Noise power ratio $(\mathbb{E}(S_{\mathsf{col}}) - \mathbb{E}(S_{\mathsf{inn}}))^2/\mathbb{V}(S_{\mathsf{inn}})$. The original work of G. Tardos and the improvements of the previous section pertain to this trend with the constraints on the first two moments of S_{inn} to ease the definition of a universal threshold τ. This section lists some other criteria found in the literature.

3.2.1 Jeffrey's prior

The probability density function (2.19) is known as the arcsine distribution or the beta distribution Beta($1/2, 1/2$). It is also the Jeffreys prior, which is the 'least informative *a priori* distribution' for Bernoulli's trial [98]. The section makes the connection with traitor tracing[1], which is indeed not so insightful.

Let us consider the collusion attack as a prediction. At the i-th index of code, the collusion first elaborates a statistical model of their c symbols denoted here $\mathbf{x} = (x_{j_1,i}, \ldots, x_{j_c,i})$. Denote by $\tilde{\pi}_c(\mathbf{x})$ their statistical model of $\mathbf{x}$ and by $\pi_c(\mathbf{x}; p_i) = p_i^{\sigma_i}(1-p_i)^{c-\sigma_i}$ the true model (with $\sigma_i = \sum_{k=1}^c x_{j_k,i}$). Then, the colluders use their model to predict another symbol which they copy in the pirated sequence, as if it were the symbol of a $(c+1)$-th user. In other words, their attack consists in generating a codeword as typical as possible. It amounts to set $y_i = 1$ with probability $\theta_{c,\sigma} = \tilde{\pi}_{c+1}((\mathbf{x}||1))/\tilde{\pi}_c(\mathbf{x})$, where $(\mathbf{x}||1)$ is the appending of the symbol '1' to the binary word $\mathbf{x}$.

The power of this attack is related to the accuracy of their statistical model $\tilde{\pi}_c(\cdot)$. This accuracy is often measured by the Kullback-Leibler divergence w.r.t. the true distribution and averaged over the a priori distribution of P: $\mathbb{E}_P(D_{KL}(\tilde{\pi}_c||\pi_c))$. This is the so-called Bayes risk [98]. The collusion looks for the model $\tilde{\pi}_c^\star$ minimizing this risk while the code designer looks for the random variable P maximizing this risk. This defines the following maxmin game:

$$\max_P \min_{\tilde{\pi}_c} \mathbb{E}_P(D_{KL}(\tilde{\pi}_c||\pi_c)). \tag{3.8}$$

The minimization is easily solved by $\tilde{\pi}_c(\mathbf{x}) = \mathbb{E}_P(\pi_c(\mathbf{x}; P))$. The pay-off is then equal to the mutual information $I(P; \mathbf{X})$. The 'least informative prior' (a.k.a. Jeffreys prior) for Bernoulli trial maximizes this quantity making the collusion's prediction task harder. The maximization is reached by a discrete random variable P whose distribution depends on c and is difficult to compute [134, 3.1.1. - Examples]. But asymptotically as $c \to \infty$, its law converges to a absolutely continuous random variable whose density is the Tardos distribution (2.19) [98].

[1] which I mentioned in my tutorial given at conference IEEE WIFS 2009.

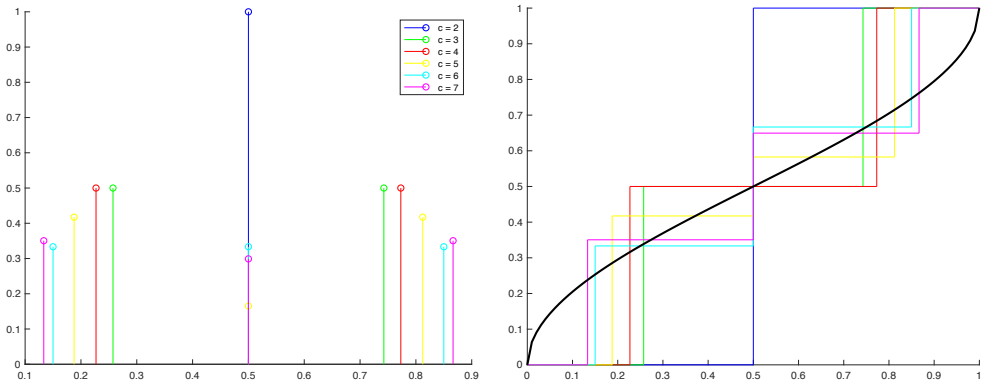


Figure 3.3: (left) Plots of the discrete distribution $f_c^\star$ for $c \in \{2,3,4,5,6,7\}$ recommended by E. Amiri *et al.* [81] and Y.-W. Huang *et al.* [118]; (right) Plots of the corresponding cumulative density functions.

This relation with Jeffreys prior seems appealing: the code designer selects the density of P as the 'least informative prior' such that the collusion has difficulty estimating the secret sequence $\mathbf{p}$ and their prediction of a new symbol is inaccurate. Yet, this connection is not sound because this gives birth to the following collusion strategy:

$$\theta_{c,\sigma} = \frac{\tilde{\pi}_{c+1}((\mathbf{x}||1))}{\tilde{\pi}_c(\mathbf{x})} = \frac{\mathbb{E}_P(p^{\sigma+1}(1-p)^{c-\sigma})}{\mathbb{E}_P(p^\sigma(1-p)^{c-\sigma})}, \forall \sigma \in \{0,\dots,c\} \tag{3.9}$$

where $\theta_{c,0} > 0$ and $\theta_{c,c} < 1$. Therefore, it violates the marking assumption. With the choice of density $f^{(T)}$, the violation 'vanishes' as $c \to \infty$: $\theta_{c,0} = 1 - \theta_{c,c} = 1/2(c+1)$.

3.2.2 Achievable rates and capacities

A sounder figure of merit is based on the achievable rate of a joint decoder:

$$R^{(J)}(P;\boldsymbol{\theta}_c) := c^{-1} I(Y; X_{\mathcal{C}}|P). \tag{3.10}$$

This concept is properly introduced in Sect. 4.3. This rate is indeed the maximum error exponent of the probability of false positive (see Sect. 4.3.2). It depends on the collusion strategy $\boldsymbol{\theta}_c$ and on the random variable P. It defines a new maxmin game between the code designer (selecting P to maximise $R^{(J)}(P;\boldsymbol{\theta}_c)$) and the collusion (selecting $\boldsymbol{\theta}_c$ to minimize $R^{(J)}(P;\boldsymbol{\theta}_c)$). The good news is that this game admits a saddle-point solution for a given c [119, Th. 3].

$$P_c^\star = \arg\max \min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_c} R^{(J)}(P;\boldsymbol{\theta}). \tag{3.11}$$

Denote by $P_c^\star$ the random variable of the saddle point: it is a discrete r.v. and its distribution depends on c [81]. Choosing $P_c^\star$ at the code construction ensures that the actual rate is larger or equal to the rate at the saddle point for any collusion (provided that the true collusion size is less or equal to c). The achievable rate at the saddle point is also called the fingerprinting capacity (see Sect. 4.5).

However, the computation of the law of this optimal $P_c^\star$ requires numerical simulation (see Fig. 3.3). Also, the difference between the minimum rates provided by $P_c^\star$ and by the r.v. of density $f^{(T)}$ (2.19) quickly vanishes as c increases [119].

3.3 Conclusion

The conclusion of the this chapter is that 'all roads lead to Rome'. This chapter looks for alternatives of the random variable P proposed by G. Tardos from different perspectives. There exist better distributions but they share the same drawbacks:

- They depend on c. At the code construction, the code designer has to bet on a collusion size. At the moment, what happens if the true collusion size is bigger is not very clear.
- Their advantage vanishes as the collusion size increases. Indeed, these random variables converge in some sense to Tardos initial choice.

These are arguments for using density $f^{(T)}$ (2.19) if it is hard to bet on a collusion size at the code generation: when there is no reason why c would be small, there is not much gain and possibly a high risk underestimating c.

Chapter 4

The collusion strategies

Considering collusion strategies amounts to figure oneself in the colluders' shoes. The natural question is what strategy is the most harmful against the decoder. This is not easy because the colluders a priori do not know how the accusation will proceed. Before presenting the criterion gauging the strength of a collusion strategy and then the worst case attacks, this chapter presents classes of collusions strategies and some of their properties.

4.1 Classes of collusion strategies

Chapter 2 models the collusion strategy as a probabilistic process in Sect. 2.2.2. As a reminder, the collusion strategy $\boldsymbol{\theta}_c$ is a vector of $(c+1)$ components which are the probabilities that the collusion puts the symbol '1' in the pirated sequence when they have σ '1' over c symbol:

$$\theta_{c,\sigma} = \mathbb{P}(Y = 1|\sigma), \quad \forall \sigma \in \{0, 1, \ldots, c\}. \tag{4.1}$$

The set of size c collusion strategies compliant with the marking assumption is denoted $\boldsymbol{\Theta}_c$ (2.3). This section splits the set $\boldsymbol{\Theta}_c$ into classes depending on the skills or knowledge at the disposal of the colluders.

The interleaving attack The interleaving attack is very special because the collusion can lead this attack without any information about their codewords. They randomly draw an assignation sequence $\mathbf{A} \in \mathcal{C}^m$, where each component $A_i \overset{\text{i.i.d.}}{\sim} \mathcal{U}_{\mathcal{C}}$ and $\mathcal{C} = \{j_1, \ldots, j_c\}$ is the set of user indices of the colluders. This component indicates which colluder puts his/her block in the pirated sequence at index i: $Y_i = x_{A_i,i}, \forall i \in \{1, \ldots, m\}$. When the colluders have σ '1' over c symbols, the probability that the assignation points to one of the colluders having a '1' is σ/c:

$$\theta_{c,\sigma} = \frac{\sigma}{c}, \forall \sigma \in \{0, \ldots, c\}. \tag{4.2}$$

Symbol-symmetric collusion strategies For this class of attack, the colluders do not know their codewords, yet they analyze the content they received block by block. When the symbols are embedded into blocks of multimedia content thanks to a secret-keyed digital watermarking techniques, the colluders can not know the symbols embedded in their copies without the watermarking secret key. However, two colluders can tell whether they share the same block of content at index i. For bigger collusions, the colluders know that they share κ blocks of the one kind and $c - \kappa$ blocks of the other kind. In other words, they may have $\sigma = \kappa$ or $\sigma = c - \kappa$ symbols '1'

over c. This ambiguity constraints the model of the collusion attack by imposing the following condition:

$$\theta_{c,\sigma} = 1 - \theta_{c,c-\sigma}, \forall \sigma \in \{0, \ldots, c\}. \tag{4.3}$$

We denote this subset of symbol-symmetric attacks by $\tilde{\boldsymbol{\Theta}}_c \subset \boldsymbol{\Theta}_c$. Majority vote, minority vote, coin flip as well as the interleaving attack (see Sect. 2.2.2) belong to $\tilde{\boldsymbol{\Theta}}_c$.

Non symbol-symmetric collusion strategies The colluders must know not only which symbols they have exactly at index i but also which blocks hide symbol '1' or '0' to perform attacks which do not fulfil the above symmetry. Yet, for a given index, they are not allowed to embed symbol they don't have in the pirated copy due to the marking assumption. 'All ones' and 'All zeros' (see Sect. 2.2.2) are examples of such collusion strategies. In multimedia applications where the symbol are embedded in content block by a watermarking scheme, this is a priori impossible. B. Mathon *et al.* mitigate this fact by assuming partial information leakages about the symbols [133].

4.2 Probabilities

It is difficult to present further results without the expression and properties of some probabilities. The expressions in the sequel suppose a given collusion strategy $\boldsymbol{\theta}_c$ of size c. For sake of simplicity, we omit this conditioning in the notations (unless necessary). In the same way, simplified notations such as $\mathbb{P}(a|b)$ (instead of $\mathbb{P}(A = a|B = b)$) are used whenever not confusing.

4.2.1 Model of Σ_i

Consider the r.v. Σ_i encoding the number of symbol '1' the collusion has at block index i: $\Sigma_i = \sum_{j \in \mathcal{C}} X_{j,i}$. There are two statistical models.

The binomial model

The symbols of the colluders have been generated by the code construction independently at random: $X_{j,i} \sim \mathcal{B}(p_i)$, $\forall j \in \mathcal{C}$ and Σ_i has a binomial distribution $\Sigma_i \sim \mathcal{B}(c, p_i)$.

$$\mathbb{P}(\Sigma_i = \sigma | p_i, c) = \binom{c}{\sigma} p_i^{\sigma} (1 - p_i)^{c-\sigma}, \quad \sigma \in \{0, \ldots, c\}. \tag{4.4}$$

This is the distribution of Σ_i knowing p_i. This chapter uses this distribution as it is more tractable for calculation.

The hypergeometric model

When the codebook has been generated, symbols $\{X_{j,i}\}_{j \in \mathcal{C}}$ are random because we don't know the user indices of the colluders. At index i, denote by $n_{1,i}$ the number of codewords having a '1' ($n_{1,i} = |\{j \in \mathcal{U} | x_{j,i} = 1\}|$) out of n. Conditioned on this information, we then have:

$$\mathbb{P}(\Sigma_i = \sigma | n_{1,i}, n, c) = \frac{\binom{n_{1,i}}{\sigma}\binom{n-n_{1,i}}{c-\sigma}}{\binom{n}{c}}, \quad \sigma \in \{l, \ldots, u\}, \tag{4.5}$$

with $l = \max(0, c - n + n_{1,i})$ and $u = \min(n_{1,i}, c)$. In other words, Σ_i is distributed as a hypergeometric distribution, like the number of successes out of c draws in a population of n without

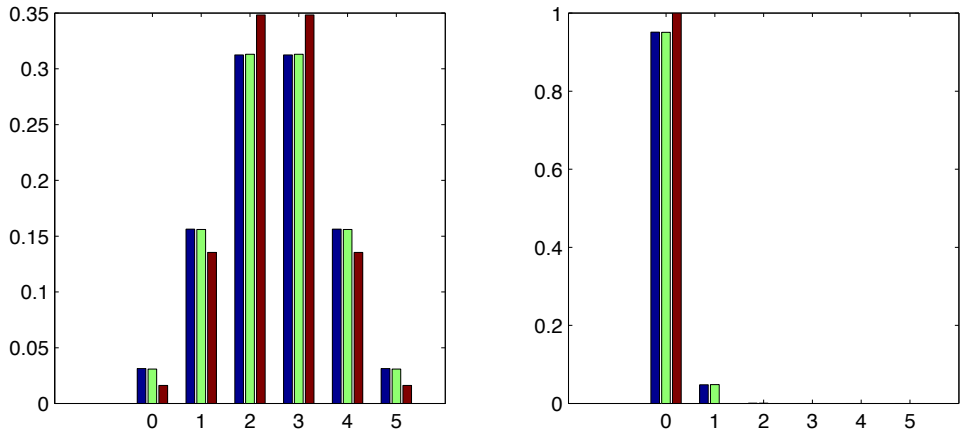


Figure 4.1: $\mathbb{P}(\Sigma = \sigma)$ for $c = 5$: Hypergeometric model with $(n, n_1) = (1\,000, 500)$ (left) and $(1\,000, 10)$ (right); Binomial model with $p = 0.5$ (left) and $p = 0.01$ (right); Hypergeometric model with $(n, n_1) = (20, 10)$ (left) and $(20, 0)$ (right).

replacement. There exist algorithms computing exact or approximate value of this probability distribution [129] circumventing the difficulty of computing $\binom{n}{k}$ for large n. This distribution of Σ_i knowing the codebook is useful at the accusation side and more precise than the binomial model [153]. Yet, the difference between the two distributions vanishes for large n as illustrated in Fig. 4.1.

4.2.2 Laws of the symbols of the pirated sequence

The above probabilities in conjunction with the model of collusion allow to express probabilities concerning the binary r.v. Y conditioned on p. Define $\Pi(p) = \mathbb{P}(Y = 1|p)$ and $\Pi_x(p) = \mathbb{P}(Y = 1|x, p)$ the probability that the collusion puts symbol '1' knowing that one colluder has the symbol $x \in \{0, 1\}$. We have for the binomial model:

$$\Pi(p) = \sum_{\sigma=0}^{c} \theta_{c,\sigma} \mathbb{P}(\Sigma = \sigma|p, c) = \sum_{\sigma=0}^{c} \theta_{c,\sigma} \binom{c}{\sigma} p^{\sigma}(1-p)^{(c-\sigma)}, \tag{4.6}$$

$$\Pi_x(p) = \sum_{\sigma=x}^{c-1+x} \theta_{c,\sigma} \mathbb{P}(\Sigma = \sigma - x|p, c-1) = \sum_{\sigma=x}^{c-1+x} \theta_{c,\sigma} \binom{c-1}{\sigma-x} p^{\sigma-x}(1-p)^{(c-1-\sigma+x)}. \tag{4.7}$$

Injecting (4.4) in (4.6) with the appropriate value of $\boldsymbol{\theta}_c$ yields the expressions for some of the collusion strategies mentioned in Sect 2.2.2, which are plotted in Fig. 4.2:

- Interleaving attack: $\Pi(p) = p$,
- Coin flip attack: $\Pi(p) = (1 - (1-p)^c + p^c)/2$,
- All-1 attack: $\Pi(p) = 1 - (1-p)^c$,
- All-0 attack: $\Pi(p) = p^c$.

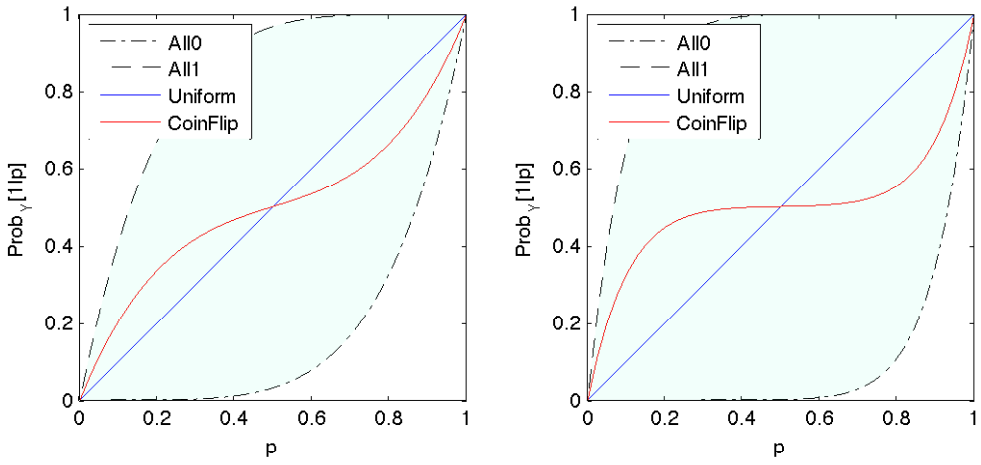


Figure 4.2: $\Pi(p) = \mathbb{P}(Y = 1|p)$ for $c = 5$ (left) and $c = 10$ (right). This function lies into the light-blue area delimitated by the All-0 and All-1 strategies.

4.2.3 Simple properties about $\Pi(p) = \mathbb{P}(Y = 1|p)$

Here is a list of trivial properties about $\Pi(\cdot)$ *for a collusion of size* c:

1. $\Pi(p)$ is a polynomial of degree at most c, with $\Pi(0) = \theta_{c,0} = 0$ and $\Pi(1) = \theta_{c,c} = 1$.
2. Two collusion strategies yield two different distributions:
$$\boldsymbol{\theta}_c \neq \boldsymbol{\theta}'_c \rightarrow \exists p \in (0,1) \,|\, \Pi(p;\boldsymbol{\theta}_c) \neq \Pi(p;\boldsymbol{\theta}'_c). \tag{4.8}$$
3. The collusion produces an attack 'in between' the All-0 and All-1 strategies:
$$\forall p \in [0,1],\ p^c \leq \Pi(p) \leq 1 - (1-p)^c. \tag{4.9}$$
4. The interleaving attack has the remarkable property of having a distribution $\Pi(p)$ independent of the size of the collusion: $\forall p \in [0,1],\ \Pi(p) = p$.
5. For symbol-symmetric attacks, the distribution is 'odd' around point $(1/2, 1/2)$: $\forall p \in [0,1],\ \Pi(1-p) = 1 - \Pi(p)$.

The second property leads to identifiability as soon as we know the collusion size. Identifiability means that we could learn the value of $\boldsymbol{\theta}_c$ if we observe an infinity of occurences of the pair (Y, P) provided that the set $\mathcal{P}$ of possible outcomes of the r.v. P is large enough. Function $p \mapsto \Pi(p;\boldsymbol{\theta}_c) - \Pi(p;\boldsymbol{\theta}'_c)$ is a polynomial of degree at most c. Therefore, it has at most c roots. Two roots are $p = 0$ and $p = 1$. This polynomial cannot cancel for each of the possible outcomes of P if $|\mathcal{P}| \geq c - 1$.

However, the fourth property proves that identifiability doesn't hold whenever we ignore the collusion size: it is not possible to distinguish two interleaving attacks of different sizes.

More generally: Suppose a collusion of size c produces an attack $\boldsymbol{\theta}_c$ yielding the distribution $\Pi(p;\boldsymbol{\theta}_c)$. A collusion of size $c+1$ can produce an attack $\boldsymbol{\theta}_{(c+1)}$ yielding the same distribution. It amounts to impose:

$$\theta_{(c+1),\sigma} = \frac{\sigma}{c+1}\theta_{c,\sigma-1} + \frac{c+1-\sigma}{c+1}\theta_{c,\sigma}, \quad \forall \sigma \in \{0,\ldots,c+1\}. \tag{4.10}$$

We call this mechanism 'Leave one colluder out': At a given i, one random colluder leaves the collusion, and the c remaining colluders lead the attack $\boldsymbol{\theta}_c$. This mechanism implies that a collusion of size $c' > c$ can create an attack with the same distribution as any collusion strategy of size c.

4.2.4 Simple properties about $\Pi_x(p) = \mathbb{P}(Y = 1|x, p)$

Interesting properties concerning the conditional probability $\Pi_x(p)$ of (4.7) are also worth mentioning:

1. For a given p, $\Pi(p)$ is at the barycenter of $\Pi_1(p)$ and $\Pi_0(p)$ with weights p and $1-p$ respectively:
$$\Pi(p) = p\Pi_1(p) + (1-p)\Pi_0(p). \tag{4.11}$$
2. $\Pi_0(0) = 0$, $\Pi_1(0) = \theta_{c,1}$, $\Pi_0(1) = \theta_{c,c-1}$, $\Pi_1(1) = 1$,
3. $\forall p \in [0,1]$, $\Pi_1(p) = \Pi(p) + c^{-1}(1-p)\Pi'(p)$.
4. $\forall p \in [0,1]$, $\Pi_0(p) = \Pi(p) - c^{-1}p\Pi'(p)$.
5. When the collusion strategy is symbol-symmetric:
$$\Pi_1(p) = 1 - \Pi_0(1-p), \forall p \in [0,1]. \tag{4.12}$$

The combination of properties 3 and 4 gives $\Pi_1(p) - \Pi_0(p) = c^{-1}\Pi'(p)$, which was used to establish (2.20) and (3.2).

Here is a comparison with digital communication. Discrete memoryless channels are modeled by conditioned probabilities $\mathbb{P}(Y = y|X = x)$ (the probability of observing output $Y = y$ knowing the input is $X = x$). The very specificity of the collusion channel here is that its characteristic also depends on the distribution of the source: $\mathbb{P}(Y = y|X = x, \mathbb{P}(X = 1) = p)$. Another point is that the conditional probability $\Pi_1(p)$ (the probability of observing the output $Y = 1$ knowing the input is $X = 1$) is bigger or lower than the marginal $\Pi(p)$ (the probability of observing the output $Y = 1$) depending on the sign of $\Pi'(p)$. Usually in digital communication, a binary channel is either bit-flipping or non bit-flipping (in the sense that the conditional probability is lower - resp. bigger - than the marginal). Here, as p varies and so the sign of $\Pi'(p)$, the nature of the channel may change. This is illustrated in Fig. 4.3:right with the Minority vote attack for $c = 10$: The channel is bit-flipping for $p \in [0.25, 0.75]$ (approximately) and non bit-flipping otherwise. Fig. 4.3:left also shows an important behaviour of the Interleaving attack. As c increases, both conditional probabilities $\Pi_1(p)$ and $\Pi_0(p)$ converges to $\Pi(p)$, $\forall p \in [0,1]$. Yet, this doesn't hold in general (see Fig. 4.3:right).

Conditional probabilities can be extended for cases when $k \leq c$ colluders symbols are known, and among them $\rho \leq k$ are symbol '1':

$$\mathbb{P}(Y = 1|\rho, k, p) = \sum_{\sigma=\rho}^{c-k+\rho} \theta_{c,\sigma} \binom{c-k}{\sigma-\rho} p^{(\sigma-\rho)}(1-p)^{(c-k-\sigma+\rho)}. \tag{4.13}$$

This expression is useful when k colluders are already identified, and the accusation looks for the remaining accomplices.

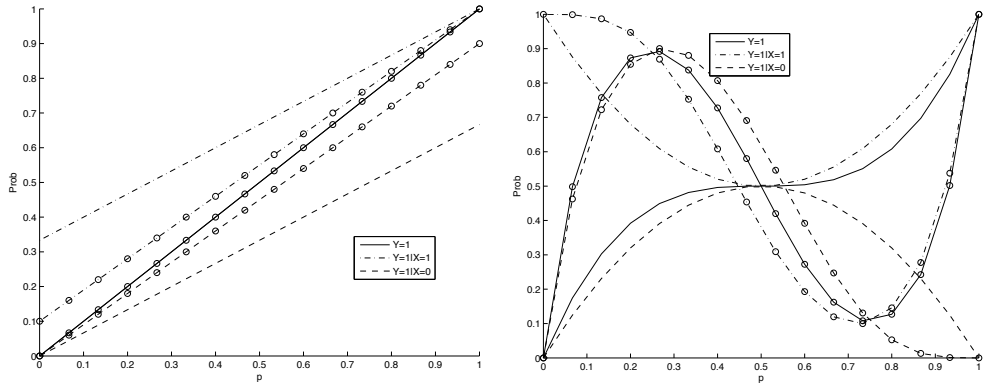


Figure 4.3: Two collusion sizes: $c = 3$ (-) and $c = 10$ (o). (left) the Interleaving attack, (right) the Minority vote attack.

4.3 Distinguishability

This section aims at measuring how harmful is a collusion strategy by the induced distinguishability between the innocents and colluders' codewords. This gives birth to the concept of achievable rate. It is first explained for single decoders and then for joint decoders.

4.3.1 The case of a single decoder

A single decoder has been presented so far as a sequence of hypothesis tests deciding whether user j is innocent or guilty. It is natural to measure how the codeword $\mathbf{X}_{\mathsf{inn}}$ of an innocent user is statistically distinguishable from the codeword $\mathbf{X}_{\mathsf{col}}$ of a colluder. The Kullback-Leibler divergence is one way to measure how different are the distributions of $\mathbf{X}_{\mathsf{col}}$ and $\mathbf{X}_{\mathsf{inn}}$. If a collusion strategy succeeds to make this difference small, then the hypothesis tests will not be reliable, and there is no hope to identify colluders while not accusing any innocent.

The symbols of $\mathbf{X}_{\mathsf{col}}$ and $\mathbf{X}_{\mathsf{inn}}$ being independent for a given observation $(\mathbf{y}, \mathbf{p})$ (thanks to the code construction and the memoryless assumption about the collusion strategy), the Kullback-Leibler divergence equals:

$$D(\mathbf{X}_{\mathsf{col}}; \mathbf{X}_{\mathsf{inn}}|\mathbf{y}, \mathbf{p}) = \sum_{i=1}^{m} \sum_{x\in\{0,1\}} \mathbb{P}(X_{\mathsf{col}} = x|y_i, p_i) \log \frac{\mathbb{P}(X_{\mathsf{col}} = x|y_i, p_i)}{\mathbb{P}(X_{\mathsf{inn}} = x|y_i, p_i)} \tag{4.14}$$

$$= \sum_{i=1}^{m} \sum_{x\in\{0,1\}} \frac{\mathbb{P}(Y = y_i|x, p_i)}{\mathbb{P}(Y = y_i|p_i)} \mathbb{P}(X_{\mathsf{col}} = x|p) \log \frac{\mathbb{P}(Y = y|x, p_i)}{\mathbb{P}(Y = y_i|p_i)}. \tag{4.15}$$

The second equation uses the Bayes rule, the fact that X_{inn} is independent of y_i, and that $\mathbb{P}(X_{\mathsf{col}} = x|p) = \mathbb{P}(X_{\mathsf{inn}} = x|p)$. The collusion doesn't know the secret sequence $\mathbf{p}$, but only the law of P. Moreover, we aim at measuring the strength of a collusion strategy $\boldsymbol{\theta}_c$ which generates sequence $\mathbf{Y}$. This suggests to measure distinguishability as the expectation over $\mathbf{Y}$ and P of the Kullback-Leibler divergence

$$D(\mathbf{X}_{\mathsf{col}}; \mathbf{X}_{\mathsf{inn}}|\boldsymbol{\theta}_c, \mathbf{P}) = \mathbb{E}_{\mathbf{P}} \left(\mathbb{E}_{\mathbf{Y}} (D(\mathbf{X}_{\mathsf{col}}; \mathbf{X}_{\mathsf{inn}}|\mathbf{Y} = \mathbf{y}, \mathbf{P} = \mathbf{p})) \right) \tag{4.16}$$

$$= m\mathbb{E}_P \left(D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P = p) \right), \quad \text{with}$$

$$D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, p) = \sum_{x\in\{0,1\}} p^x(1-p)^{1-x} \left(\Pi_x(p) \log \frac{\Pi_x(p)}{\Pi(p)} + (1 - \Pi_x(p)) \log \frac{1 - \Pi_x(p)}{1 - \Pi(p)} \right).$$

This quantity depends on the collusion strategy because functions $\Pi(\cdot)$ (4.6), $\Pi_0(\cdot)$, and $\Pi_1(\cdot)$ (4.7) depend on $\boldsymbol{\theta}_c$.

Asymptotic setup In an asymptotical setup where $m \to \infty$, the probabilities of error $\mathbb{P}_{\mathsf{fp}}$ and $\mathbb{P}_{\mathsf{fn}}$ for accusing a given user j may converge to zero exponentially. The error exponents are defined as:

$$E_{\mathsf{fp}} := \lim_{m\to\infty} -\frac{1}{m}\log \mathbb{P}_{\mathsf{fp}}, \quad E_{\mathsf{fn}} := \lim_{m\to\infty} -\frac{1}{m}\log \mathbb{P}_{\mathsf{fn}}. \tag{4.17}$$

Appendix 13 shows that there is a tradeoff between the two error exponents. The requirement of utmost importance is not to accuse any innocent. Therefore, informally, the accusation prefers to operate at a large E_{fp} and small E_{fn}.

Achievable rate We now consider a set of n users and an accusation objective (see Sect. 2.3.3) defining the global error probabilities $\mathbb{P}_{\mathsf{FN}}$ and $\mathbb{P}_{\mathsf{FP}}$. The rate R of the Tardos code is defined as

$$R := \frac{\log n}{m}. \tag{4.18}$$

Assume that this quantity is fixed so that, asymptotically as $m \to \infty$, n scales as e^{mR}. If the code can sustain a rate $R = \log 2$, then it can manage a set of $n = 2^m$ users with codewords composed of m binary symbols. This happens when there is a unique 'colluder': according to the marking assumption, the pirated sequence $\mathbf{y}$ exactly corresponds to the codeword of this dishonest user. A perfect accusation is possible provided that each user has a unique codeword. This requires $m = {}^{\log n}/_{\log 2}$ binary symbols (or more precisely $m = \lceil {}^{\log n}/_{\log 2} \rceil$). Yet, as soon as there are several colluders, sequence $\mathbf{y}$ will correspond to an innocent user codeword. This shows that a rate as big as $\log 2$ is not possible. If $R < \log 2$, it means that the code is a small (and random) subset of the 2^m binary sequences. How to choose R?

A rate R is *achievable* if it leads to error probabilities $\mathbb{P}_{\mathsf{FN}}$ and $\mathbb{P}_{\mathsf{FP}}$ exponentially vanishing as $m \to \infty$ while n is increasing as fast as e^{mR}. Appendix 13 shows that the supremum of the achievable rates indeed equals the distinguishability $D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$.

Nevertheless, this argument only holds for a particular accusation procedure, the so-called single decoder: It computes a score per user as a function of $(\mathbf{x}, \mathbf{y}, \mathbf{p})$ and compares this score to a threshold to accuse a user. First, there might be other accusation procedures yielding exponentially vanishing error probabilities at rate bigger than $D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$. Second, a rate as big as $D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$ is achievable only when the score function is the likelihood ratio [102, Th. 12.7.1]. But its expression depends on $\boldsymbol{\theta}_c$ unknown in practice.

Relationship to information theory The distinguishability $D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P = p)$ appearing in (4.16) equals the mutual information $I(Y; X_{\mathsf{col}}|\boldsymbol{\theta}_c, P = p)$. This measures in nats how much information a symbol Y of the pirated sequence crafted by the collusion strategy $\boldsymbol{\theta}_c$ reveals about the symbol of a colluder when $P = p$. Taking the expectation over P defines the supremum of the achievable rates of any single decoder (against collusion strategy $\boldsymbol{\theta}_c$ and for a given definition of P) denoted by:

$$R^{(S)}(P, \boldsymbol{\theta}_c) := I(Y; X_{\mathsf{col}}|\boldsymbol{\theta}_c, P), \tag{4.19}$$

where $I(Y; X_{\mathsf{col}}|\boldsymbol{\theta}_c, P)$ reads as $\mathbb{E}_P(I(Y; X_{\mathsf{col}}|\boldsymbol{\theta}_c, P = p))$.

4.3.2 Joint decoders

A joint decoder is a generalization of a single decoder, which analyzes a group of $\ell > 1$ users jointly (we assume that $\ell \leq c$). Indeed, $\ell + 1$ hypotheses are competing about the composition of a given group. Denote by $\mathcal{H}_j$ the hypothesis that there are j colluders in this group, with $j \in \{0, 1, \cdots, \ell\}$. There are $\ell(\ell + 1)$ types of error like estimating the number of colluders by $\hat{\jmath} \neq j$ whereas $\mathcal{H}_j$ is true. The probabilities of these errors may exponentially converge to zero, but with different exponents. Moreover, the number of groups under hypothesis $\mathcal{H}_j$ depends on j. This makes the study of the achievable rate of a joint decoder more complex than for a single decoder. Sect. 13.1.2 in Appendix 13 shows that the supremum of the achievable rates of a joint decoder with group size ℓ equals the distinguishability $\ell^{-1} D(X_{\mathsf{col}}^{(\ell)}; X_{\mathsf{inn}}^{(\ell)} | \boldsymbol{\theta}_c, P)$ which is also the mutual information $\ell^{-1} I(Y; X_{\mathsf{col}}^{(\ell)} | \boldsymbol{\theta}_c, P)$.

This last quantity is increasing with ℓ. As a consequence, joint decoding ($\ell > 1$) performs better than single decoding ($\ell = 1$) in the sense that it provides higher achievable rates. Stated differently, for a fixed rate R, joint decoding provides higher error exponents. Moreover, the best joint decoder of this kind is the one dealing with groups of size $\ell = c$, whose supremum of achievable rates is denoted by [136, Sec. 5]:

$$R^{(J)}(P, \boldsymbol{\theta}_c) := c^{-1} \mathbb{E}_P \left(I(Y; X_{\mathcal{C}} | P = p, \boldsymbol{\theta}_c) \right) \quad \text{with} \tag{4.20}$$

$$\begin{aligned} I(Y; X_{\mathcal{C}} | P = p, \boldsymbol{\theta}_c) &= \sum_{\sigma=0}^{c} \mathbb{P}\left(\Sigma = \sigma | p\right) \theta_{c,\sigma} \log \left(\frac{\theta_{c,\sigma}}{\Pi(p; \boldsymbol{\theta}_c)} \right) \qquad (4.21) \\ &+ \sum_{\sigma=0}^{c} \mathbb{P}\left(\Sigma = \sigma | p\right) \left(1 - \theta_{c,\sigma}\right) \log \left(\frac{1 - \theta_{c,\sigma}}{1 - \Pi(p; \boldsymbol{\theta}_c)} \right), \\ &= h_b \left(\sum_{\sigma=0}^{c} \theta_{c,\sigma} \mathbb{P}\left(\Sigma = \sigma | p\right) \right) - \sum_{\sigma=0}^{c} h_b(\theta_{c,\sigma}) \mathbb{P}\left(\Sigma = \sigma | P = p\right). \qquad (4.22) \end{aligned}$$

This is a theoretical result not implementable in practice for the following reasons:

- The collusion size c is unknown at the decoding stage.
- This decoder has to compute log-likelihoods that depend on $\boldsymbol{\theta}_c$, also not known.
- The complexity of the decoder is in $O((c+1).n^c)$. The fact that the condition related to error probability $\mathbb{P}_e(c|0)$ is the limiting factor for the achievable rate suggests that the decoder could be only based on the score $\log \mathbb{P}(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_c} | \mathcal{H}_c, \mathbf{y}, \mathbf{p}) - \log \mathbb{P}(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_c} | \mathcal{H}_0, \mathbf{y}, \mathbf{p})$. Yet, there are still as many scores to be computed as groups of size c. Since $\binom{n}{c} = O(n^c)$, such joint decoder is not tractable in practice for large n.

4.3.3 Informed Decoders

This section evidences that caught colluders relentlessly inform on the remaining accomplices. It generalizes the family of decoders with conditioning on some colluder codewords. Let $R(\ell, k, P, \boldsymbol{\theta}_c)$ denote the rate of a joint decoder computing a score per group of ℓ users and informed by the disclosure of the identity of k colluders, assuming that

$$1 \leq \ell \leq c \quad \text{and} \quad 0 \leq k \leq c - \ell. \tag{4.23}$$

In other words,

$$R(\ell, k, P, \boldsymbol{\theta}_c) := \frac{1}{\ell} I(Y; X_{\mathsf{col}}^{(\ell)} | X_{\mathsf{col}}^{(k)}, \boldsymbol{\theta}_c, P). \tag{4.24}$$

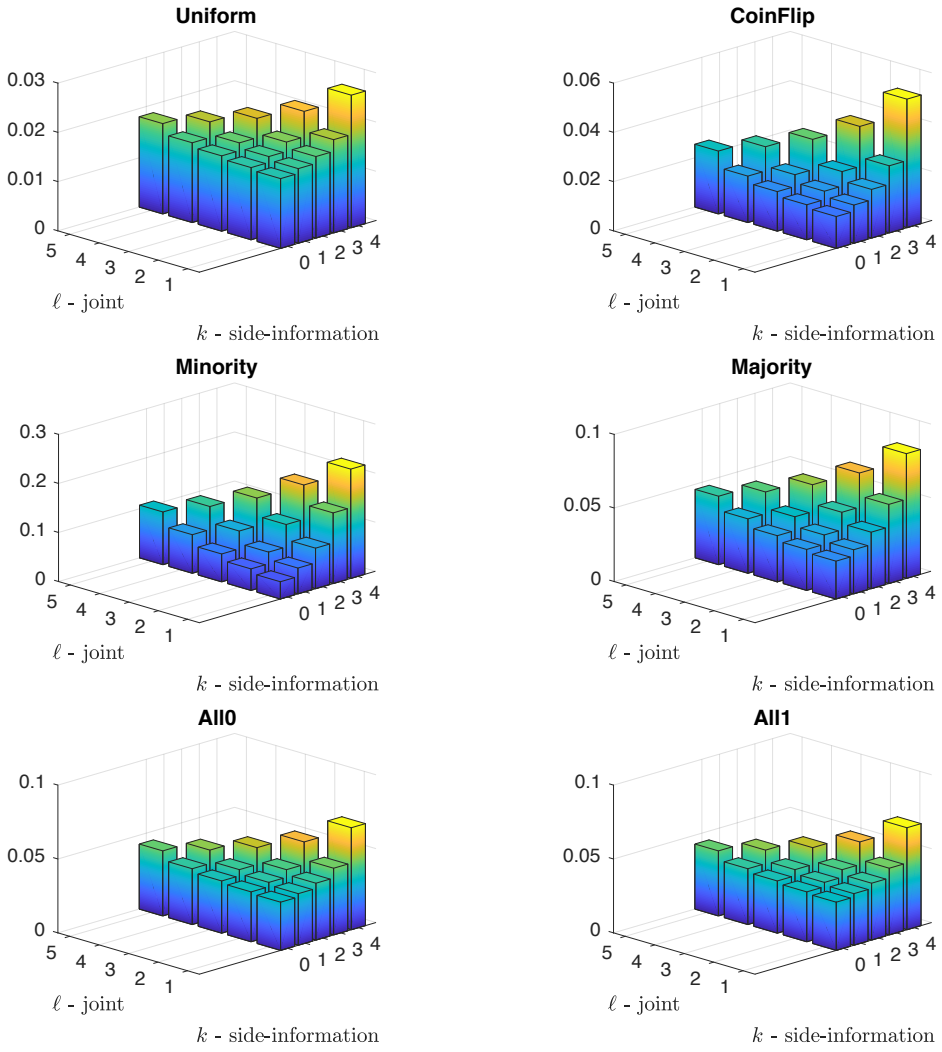


Figure 4.4: Rates $R(\ell, k, P, \boldsymbol{\theta}_c)$ of informed decoders for common collusion strategies (listed in Sect. 2.2.2), with $c = 5$ and $f^{(T)}$as the density of P.

Previous sections deal with $R(1, 0, P, \boldsymbol{\theta}_c) = R^{(S)}(P, \boldsymbol{\theta}_c)$ and $R(c, 0, P, \boldsymbol{\theta}_c) = R^{(J)}(P, \boldsymbol{\theta}_c)$. For $k > 0$, the expression of the informed achievable rate needs the conditional probability (4.13).

Caught colluders relentlessly inform on the remaining accomplices because $R(\ell, k, P, \boldsymbol{\theta}_c)$ is increasing with k as shown in Sect. 13.1.3 of Appendix 13:

$$R(\ell, k, P, \boldsymbol{\theta}_c) \leq R(\ell + 1, k, P, \boldsymbol{\theta}_c) \leq R(\ell, k + 1, P, \theta_c). \tag{4.25}$$

Therefore, the bigger k the easier it is to catch the remaining colluders. Fig. 4.4 shows the rates of informed decoder for $\ell \in \{1, \ldots, c\}$ and $k \in \{0, \ldots, c - \ell\}$.

Suppose that $0 \leq k < c$ colluders are already identified and that the accusation proceeds to a joint decoding over groups of size $\ell < c-k$ with a 'Catch One' target. Two outputs are possible:

- If the actual rate R is lower than $R(\ell, k, P, \boldsymbol{\theta}_c)$, this decoding succeeds (in an asymptotical setup). Therefore, one extra colluder is identified, and the accusation can now proceed to

another decoding with $k+1$ identified colluders. Eq. (4.25) shows that this new will again succeed. And all remaining colluders will be caught one by one.

- If $R > R(\ell, k, P, \boldsymbol{\theta}_c)$, this decoding is expected to fail, outputting an empty set. However, the game is not over yet. The accusation can still proceed to a joint decoding over groups of size $\ell+1$. This new decoding handles higher rate (Eq. (4.25)), but at a bigger complexity. If it succeeds in identifying a new colluder, Eq. (4.25) also shows that the accusation can safely go back to a joint decoding over groups of size ℓ with this new side-information.

These arguments are theoretical: the achievable rate $R(\ell, k, P, \boldsymbol{\theta}_c)$ is the expectation of the conditioned mutual information over $X^{(k)}$. It means that there might be some cases $X^{(k)}_{\mathsf{col}} = x^{(k)}_{\mathsf{col}}$ where the inequalities (4.25) does not hold. Yet, this analysis drives the architecture of iterative decoder detailed in Chap. 5.

4.4 Worst case attacks

The worst case attack is usually defined as the collusion strategy which minimizes a given figure of merit under a constraint. The figure of merit is usually the supremum of the achievable rates (4.19) or (4.20) when the colluders do not know the exact accusation procedure. This figure of merit measures the performance of the best decoder of a kind (*i.e.* , single or joint) against the collusion strategy $\boldsymbol{\theta}_c$. Therefore, an attack lowering this upper bound is likely to 'hurt' any other decoder of the same kind. As for the constraint, the worst case attack belongs to a set $\mathcal{F}$, which is either $\boldsymbol{\Theta}_c$ (*i.e.* fulfilling the marking assumption) or $\tilde{\boldsymbol{\Theta}}_c$ (with constraint (4.3)). The assumption here is that the collusion does not know the secret sequence $\mathbf{p}$, but random variable P (*i.e.* its distribution) is public. This rationale has two options depending on the complexity of the decoder.

4.4.1 Single decoder

In the first option, the accusation procedure uses a single decoder. The worst case attack is defined by:

$$\breve{\boldsymbol{\theta}}_c^{(S)} := \arg\min_{\boldsymbol{\theta}_c \in \mathcal{F}} R^{(S)}(P, \boldsymbol{\theta}_c). \tag{4.26}$$

Note that this is the worst case attack for a given random variable P. Here are some results about $\breve{\boldsymbol{\theta}}_c^{(S)}$ and $R^{(S)}(P, \breve{\boldsymbol{\theta}}_c^{(S)})$:

1. If P has a law symmetric around $1/2$ (*i.e.* $\mathbb{P}(P \leq p) = \mathbb{P}(P \geq 1-p)$, $\forall p \in [0,1]$), then $\breve{\boldsymbol{\theta}}_c^{(S)} \in \tilde{\boldsymbol{\Theta}}_c$.
2. A collusion of size $c < \infty$ cancels $R^{(S)}(P, \boldsymbol{\theta}_c)$ if $\Pi_0(P) - \Pi_1(P) = 0$. Therefore, this cannot happen if
   - If P is an absolutely continuous random variable,
   - If P is a discrete random variable with $|\mathcal{P}| > c-1$ (because $\Pi_0(\cdot) - \Pi_1(\cdot)$ is a polynomial of degree at most $c-1$),
   - If P is a discrete random variable with $\mathcal{P} \not\subset [\eta_c, 1-\eta_c]$ with $\eta_c \in (1/c, 2/c)$ the smallest real root of $p \mapsto (1-p)^{c-2}(1-cp) + p^{c-1}$ [37, Prop. 3].
3. The minimizer is unique whenever identifiability is granted among the collusion strategies of size c (see Sect. 4.2.2).

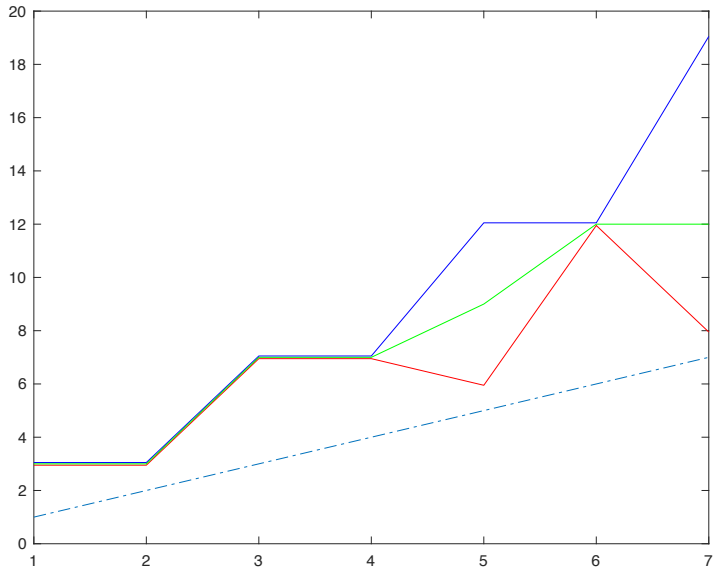


Figure 4.5: The minimum number of colluders c to cancel $R^{(S)}(P^\star_{c_e}, \boldsymbol{\theta}_c)$ when $P^\star_{c_e}$ is a discrete random variable provably good for fighting a collusion of size c_e. Provably good in the sense of K. Nuida (3.5), T. Laarhoven (3.7), or E. Amiri and Y.-W. Huang *et al.* (3.11)

The first proposition implies that the colluders do not have to know their symbols to perform the worst case attack when P is symmetric.

The second proposition outlines a potential danger with a discrete random variable P: A large enough collusion always succeeds to cancel the mutual information. This implies that, whatever the length m of the code, the single decoder can no longer identify colluders. For instance, $P = 1/2$ (*i.e.* a constant) is the best choice when fighting against a collusion of size 2 (in the sense that it maximizes the achievable rate). Yet, this is a catastrophic choice if $c > 2$. For instance, with $c = 3$ and a minority vote (*i.e.* $\boldsymbol{\theta}_{3,\min} = (0, 1, 0, 1)$) then $R^{(S)}(1/2, \boldsymbol{\theta}_{3,\min}) = 0$. The POCS algorithm [87] (Projection Onto Convex Sets) is able to find whether there exists a collusion strategy of size c canceling the rate for a given P discrete random variable P. It alternates between projecting θ_c onto the hypercube $\boldsymbol{\Theta}_c$ and projecting onto the affine subspace $\{\boldsymbol{\theta}_c \in \mathbb{R}^{c+1} | \Pi_0(P) - \Pi_1(P) = 0 \text{ and } \theta_{0,c} = 1 - \theta_{c,c} = 0\}$. Fig. 4.5 shows the minimum size of the collusion cancelling the rate of the single decoder for the discrete distributions seen in Chapter 3. Obviously, some distributions are more risky than others.

In general, there is no closed-form solution of this minimization problem. A numerical solver has difficulty finding the worst case attack for large collusion size even if the optimization problem is convex. Not only the dimension of the problem is big ($c - 1$ parameters to find the worst case attack in $\boldsymbol{\Theta}_c$ or $\lfloor c-1/2 \rfloor$ parameters in $\tilde{\boldsymbol{\Theta}}_c$), but also approximation of binomial coefficient $\binom{c}{k}$ necessary to compute $\Pi(\cdot)$ are less accurate at large c.

The interleaving attack $\boldsymbol{\theta}_{c,\mathsf{int}} = (0, 1/c, \ldots, c-1/c, 1)$ is not the worst case attack. It yields the following achievable rate:

$$R^{(S)}(P, \boldsymbol{\theta}_{c,\mathsf{int}}) = \mathbb{E}(h_b(P)) - \mathbb{E}(P h_b(P + (1-p)/c) + (1-P) h_b(P - p/c)) \quad (4.27)$$

$$= \frac{1}{2c^2} + o\left(\frac{1}{c^2}\right). \quad (4.28)$$

with $h_b(p) := -p\log(p) - (1-p)\log(1-p)$ (the entropy in nats of the Bernoulli distribution $\mathcal{B}(p)$).

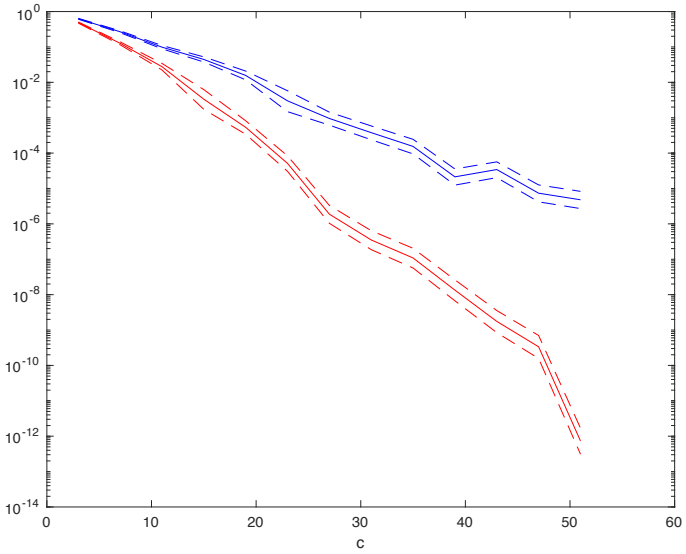


Figure 4.6: Estimation of the probability that a random collusion strategy uniformly distributed over $\boldsymbol{\Theta}_c$ (resp. over $\tilde{\boldsymbol{\Theta}}_c$) is worse than the interleaving attack $\boldsymbol{\theta}_{c,\mathsf{int}}$ for a single decoder and $c \in \{3, \ldots, 51\}$. The bounds of the confidence interval at 95% are plotted in dashed lines. This simulation uses the approximated optimal discrete distributions of T. Laarhoven (3.7) (with $|\mathcal{P}| = 200$) as an approximation of $f^{(T)}$ to avoid numerical integration. Volume $|\mathcal{W}_c^{(S)}|$ is indeed the probability that $\boldsymbol{\theta}_c \in \mathcal{W}_c^{(S)}$ when $\boldsymbol{\theta}_c \sim \mathcal{U}_{\boldsymbol{\Theta}_c}$ because $|\boldsymbol{\Theta}_c| = 1$. A rare event simulation estimates this probability (see Chapter. 6).

Yet, when P is distributed according to the density $f^{(T)}$, the interleaving attack becomes one of the worst attacks:

- Its achievable rate $R^{(S)}(P, \boldsymbol{\theta}_{c,\mathsf{int}})$ converges to $R^{(S)}(P, \breve{\boldsymbol{\theta}}_c^{(S)})$ as $c \to \infty$ [119], as illustrated in Table 4.1.
- Define $\mathcal{W}_c^{(S)} \subset \boldsymbol{\Theta}_c$ as the set of collusion strategies having a lower achievable rate than $R^{(S)}(P, \boldsymbol{\theta}_{c,\mathsf{int}})$. Define similarly $\tilde{\mathcal{W}}_c^{(S)} \subset \tilde{\boldsymbol{\Theta}}_c$ for symbol-symmetric collusion strategies. Fig. 4.6 experimentally shows that the volume $|\mathcal{W}_c^{(S)}|$ (resp. $|\tilde{\mathcal{W}}_c^{(S)}|$) converges exponentially to zero as $c \to \infty$.

4.4.2 Joint decoder

In the second option, the accusation procedure uses a joint decoder, which computes a score per group of c users only based on their codewords. The figure of merit is then the achievable rate of a joint decoder $R^{(J)}(P, \boldsymbol{\theta}_c) = c^{-1}.I(Y; X_{\mathcal{C}}|P, \boldsymbol{\theta}_c)$, the mutual information between Y, a symbol of the pirated sequence, and $X_{\mathcal{C}}$, the symbols of the colluders' code sequences. The worst case attack against a joint decoder is then defined as

$$\breve{\boldsymbol{\theta}}_c^{(J)} := \arg\min_{\boldsymbol{\theta}_c \in \mathcal{F}} R^{(J)}(P, \boldsymbol{\theta}_c). \tag{4.29}$$

Note that this is the worst case attack for a given random variable P.

Table 4.1: Worst case attacks against a single decoder when $P \sim f^{(T)}$. Their achievable rates are compared to the achievable rate against the interleaving collusion strategy of the same size (in nats).

c	$\breve{\boldsymbol{\theta}}_c^{(S)}$	$R^{(S)}(P, \breve{\boldsymbol{\theta}}_c^{(S)})$	$R^{(S)}(P, \boldsymbol{\theta}_{c,\mathsf{int}})$
2	(0, 0.5, 1)	8.15×10^{-2}	8.15×10^{-2}
3	(0, 0.651, 0.349, 1)	3.26×10^{-2}	3.73×10^{-2}
4	(0, 0.487,0.5, 0.513, 1)	1.91×10^{-2}	2.16×10^{-2}
5	(0, 0.593,0.00,1.00, 0.407, 1)	1.19×10^{-2}	1.41×10^{-2}
6	(0, 0.503,0.173,0.5,0.819, 0.497, 1)	8.54×10^{-3}	1.00×10^{-2}
7	(0, 0.490,0.00,0.896,0.104, 1.00, 0.510, 1)	6.29×10^{-3}	7.49×10^{-3}
8	(0, 0.470,0.00,0.688,0.5,0.312, 1.00, 0.530, 1)	4.88×10^{-3}	5.82×10^{-3}
9	(0, 0.439,0.00,0.690,0.248,0.752,0.310, 1.00, 0.561, 1)	3.91×10^{-3}	4.66×10^{-3}
10	(0, 0.415,0.00,0.642,0.282,0.5,0.718,0.358, 1.00, 0.585, 1)	3.21×10^{-3}	3.80×10^{-3}

In general, there is no closed-form solution of this minimization problem. Yet, the Blahut-Arimoto algorithm iteratively finds the solution [37, Sect. 3.1]. This worst case attack is also symbol-symmetric when P has a law symmetric around $1/2$ (*i.e.* $\mathbb{P}(P \leq p) = \mathbb{P}(P \geq 1-p)$, $\forall p \in [0,1]$) [37, Prop. 1].

Again, the interleaving attack $\boldsymbol{\theta}_{c,\mathsf{int}} = (0, 1/c, \ldots, {c-1}/c, 1)$ is not the worst case attack for a joint decoder. Yet, when P is distributed according to the density $f^{(T)}$, the interleaving attack becomes one of the worst attacks for large c:

- Its achievable rate $R^{(J)}(P, \boldsymbol{\theta}_{c,\mathsf{int}})$ converges to $R^{(J)}(P, \breve{\boldsymbol{\theta}}_c^{(J)})$ as $c \to \infty$ [119], as illustrated in Table 4.2. Indeed

$$R^{(J)}(P, \boldsymbol{\theta}_{c,\mathsf{int}}) = \frac{1}{c}\left(2\log(2) - 1 - \frac{1}{\pi}\sum_{\sigma=1}^{c-1} \frac{\Gamma(\sigma+1/2)\Gamma(c-\sigma+1/2)}{\Gamma(\sigma+1)\Gamma(c-\sigma+1)} h_b(\sigma/c)\right). \tag{4.30}$$

- Define $\mathcal{W}_c^{(J)} \subset \boldsymbol{\Theta}_c$ as the set of collusion strategies having a lower achievable rate than $R^{(J)}(P, \boldsymbol{\theta}_{c,\mathsf{int}})$. Define similarly $\tilde{\mathcal{W}}_c^{(J)} \subset \tilde{\boldsymbol{\Theta}}_c$ for symbol-symmetric collusion strategies. Fig. 4.7 shows that the volumes $|\mathcal{W}_c^{(J)}|$ and $|\tilde{\mathcal{W}}_c^{(J)}|$ converge to zero as $c \to \infty$ even more rapidly than for a single decoder.

4.5 Capacities

The above sections introduce the concept of supremum of achievable rates, but it was restricted so far to specific decoders computing score per user (single) or per group (joint). These decoders are theoretically sound because their scoring functions are based on log-likelihoods, but one may think of totally different accusation procedures.

4.5.1 The converse proof

The goal of the converse proof is to show that whatever the accusation procedure, a code operating at a rate higher than these achievable rates is bound to fail.

Table 4.2: Worst case attacks against a joint decoder when $P \sim f^{(T)}$. Their achievable rates are compared to the achievable rate against the interleaving collusion strategy of the same size (in nats).

c	$\breve{\boldsymbol{\theta}}_c^{(J)}$	$R^{(J)}(P, \breve{\boldsymbol{\theta}}_c^{(J)})$	$R^{(J)}(P, \boldsymbol{\theta}_{c,\mathsf{int}})$
2	(0, 0.5, 1)	1.065×10^{-1}	1.065×10^{-1}
3	(0, 0.34, 0.66, 1)	4.919×10^{-2}	4.920×10^{-2}
4	(0, 0.259,0.5, 0.741, 1)	2.826×10^{-2}	2.827×10^{-2}
5	(0, 0.209,0,403, 0.597, 0.791,1)	1.834×10^{-2}	1.834×10^{-2}
6	(0, 0.176,0.338,0.50,0.662, 0.824, 1)	1.286×10^{-2}	1.287×10^{-2}
7	(0, 0.151,0.291, 0.431, 0.569, 0.709, 0.849, 1)	9.515×10^{-3}	9.524×10^{-3}
8	(0, 0.133,0.256,0.378,0.50,0.622, 0.744, 0.867, 1)	7.327×10^{-3}	7.335×10^{-3}
9	(0, 0.119,0.229,0.338,0.446,0.554,0.662, 0.771, 0.881, 1)	5.816×10^{-3}	5.825×10^{-3}
10	(0, 0.107,0.206,0.305,0.403,0.50,0.597, 0.695, 0.794, 0.893, 1)	4.729×10^{-3}	4.739×10^{-3}

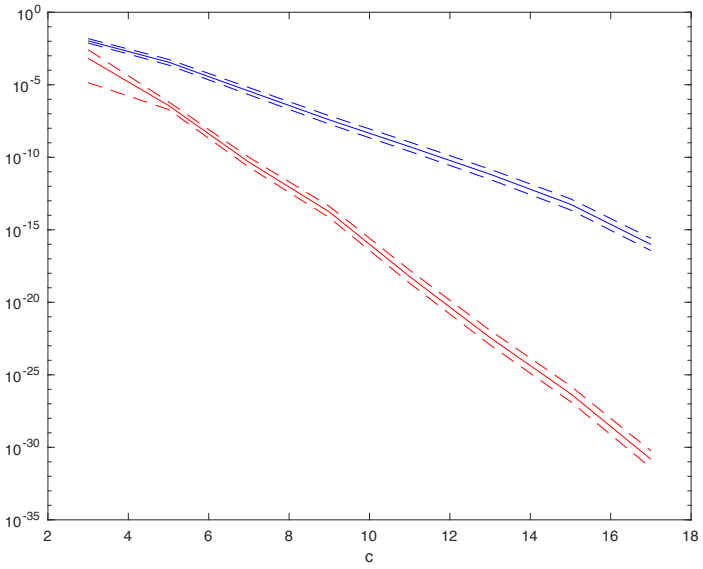


Figure 4.7: Estimation of the probability that a random collusion strategy uniformly distributed over $\boldsymbol{\Theta}_c$ (resp. over $\tilde{\boldsymbol{\Theta}}_c$) is worse than the interleaving attack $\boldsymbol{\theta}_{c,\mathsf{int}}$ for a joint decoder and $c \in \{3, \ldots, 17\}$. The bounds of the confidence interval at 95% are plotted in dashed lines. This simulation uses the approximated optimal discrete distributions of T. Laarhoven (3.7) (with $|\mathcal{P}| = 200$) as an approximation of $f^{(T)}$ to avoid numerical integration.

For instance, we consider a joint decoder of size $\ell = c$ under the 'Catch all' objective. There are $\binom{n}{c}$ groups of size c over n users. The collusion $\mathcal{C}$ is one of these groups, selected at random (uniform distribution). Its entropy thus equals $H(\mathcal{C}) = \log \binom{n}{c}$. The accusation procedure outputs a group $\hat{\mathcal{C}}$, and we denote $\mathbb{P}_e = \mathbb{P}(\hat{\mathcal{C}} \neq \mathcal{C})$. Then

$$\begin{aligned} H(\mathcal{C}) &= H(\mathcal{C}|\mathbf{P}) = H(\mathcal{C}|\mathbf{Y}, \boldsymbol{\theta}_c, \mathbf{P}) + I(\mathbf{Y}; \mathcal{C}|\boldsymbol{\theta}_c, \mathbf{P}) && (4.31)\\ &\leq H(\mathcal{C}|\mathbf{Y}, \mathbf{P}) + I(\mathbf{Y}; \mathbf{X}_{\mathsf{col}}^{(c)}|\boldsymbol{\theta}_c, \mathbf{P}) && (4.32)\\ &\leq h_B(\mathbb{P}_e) + \mathbb{P}_e \log \binom{n}{c} + mI(Y; X_{\mathsf{col}}^{(c)}|\boldsymbol{\theta}_c, P). && (4.33) \end{aligned}$$

The first inequality is due to the data processing inequality [102, Th. 2.8.1] as $\mathcal{C} - \mathbf{X}_{\mathsf{col}}^{(c)} - \mathbf{Y}$ forms a Markov chain, and the second inequality is due to Fano's theorem [102, Th. 2.11.1]. Dividing the last inequality by mcR and taking it to the limit as $m \to \infty$ while enforcing $n = e^{mR}$ constraints $\mathbb{P}_e$ as follows:

$$1 - \frac{1}{Rc} I(Y; X_{\mathsf{col}}^{(c)}|\boldsymbol{\theta}_c, P) \leq \lim_{m\to\infty} \mathbb{P}_e. \tag{4.34}$$

This shows that having a rate R bigger than $c^{-1}I(Y; X_{\mathsf{col}}^{(c)}|\boldsymbol{\theta}_c, P)$ prevents $\mathbb{P}_e$ from converging to 0 while m increases.

4.5.2 Game theory

The constraint above holds for any collusion strategy including the worst case, so that indeed, the rate is upper bounded by $R^{(J)}(P, \breve{\boldsymbol{\theta}}_c^{(J)})$. However, random variable P was so far fixed. The code designer uses it as a defence means to maximize the latter quantity. In the end, the capacity of the traitor tracing scheme with a joint decoder and under the 'Catch all' objective is defined via a game in between the code designer and the collusion:

$$C_c^{(J),\mathsf{all}} = \max_P \min_{\boldsymbol{\theta}_c \in \boldsymbol{\Theta}_c} R^{(J)}(P, \boldsymbol{\theta}_c). \tag{4.35}$$

This is a max-min game because the code designer first selects P, which is public, and then the collusion selects $\boldsymbol{\theta}_c$. The error probabilities $\mathbb{P}_{\mathsf{FN}}$ and $\mathbb{P}_{\mathsf{FP}}$ vanish to zero as $m \to \infty$ for any collusion strategy if and only if the rate R is lower than the capacity.

However, the capacity relies on a joint decoder dealing with groups of size $\ell = c$, whose complexity is prohibitive in practice. It is interesting to introduce the concept of capacity of a traitor tracing scheme with a less complex accusation procedure. Unfortunately, the converse proof for the other cases (other objectives and other decoders) is more involved [136]. The capacity of a single decoder for instance equals [136, Th. 4.1]:

$$C_c^{(S),\mathsf{one}} = \max_P \min_{\boldsymbol{\theta}_c \in \boldsymbol{\Theta}_c} R^{(S)}(P, \boldsymbol{\theta}_c). \tag{4.36}$$

4.5.3 Known results

Here is a summary of some known results. Capacities are measured in nats.

1. The 'Catch-all' objective is harder than the 'Catch-one' [136, Lemma 3.2]:

$$\begin{aligned} C_c^{(J),\mathsf{all}} &\leq C_c^{(J),\mathsf{one}}, && (4.37)\\ C_c^{(S),\mathsf{all}} &\leq C_c^{(S),\mathsf{one}}. && (4.38) \end{aligned}$$

Indeed, the 'Catch-all' objective might be impossible. For instance, identifying a colluder who participated to the forgery just for only one symbol index is an illusion even if $m \to \infty$. The capacity under this objective is thus 0, unless the set of collusion strategies is restricted. The collusion strategies considered in this report are invariant to permutation (a.k.a. user-symmetric, see Sect. 2.2.1) so that all colluders evenly contribute to the forgery. In that particular case, the above inequalities are indeed equalities, and the super-scripts all and one can be dropped. This is a very positive result.

2. A single decoder is less powerful than a joint decoder [136, Th. 4.1]:

$$C_c^{(S)} \leq C_c^{(J)}. \tag{4.39}$$

Yet, the single decoder is less complex than a joint decoder.

3. The solution of these games (joint or single) is a saddle point where P has a law symmetric around $1/2$ (*i.e.* $\mathbb{P}(P < p) = 1 - \mathbb{P}(P < 1-p)$) and $\boldsymbol{\theta}_c$ is symbol symmetric (see Sect. 4.1) [119, Th. 4].

4. The capacities are bounded by [119, Cor. 2, Cor. 3]

$$\frac{2}{\pi^2 c^2} \leq \quad C_c^{(S)} \quad \leq \frac{1}{2c^2} + O(c^{-4}), \tag{4.40}$$

$$\frac{2}{\pi^2 c^2} \leq \quad C_c^{(J)} \quad \leq \frac{1}{c^2}. \tag{4.41}$$

5. If P is an absolutely continuous random variable whose density satisfies $\int_0^1 (f(p)p(1-p))^{-1} dp < \infty$, then asymptotically, as $c \to \infty$ [119, Cor. 7]:

$$C_c^{(J)} \approx \frac{1}{2c^2}, \tag{4.42}$$

and the arguments of this asymptotical saddle point are the Tardos density $f^{(T)}$ and the interleaving attack $\boldsymbol{\theta}_{c,\mathsf{int}}$.

6. If P is not restricted as above, then, asymptotically, as $c \to \infty$:

$$\frac{1}{2c^2} \leq C_c^{(J)} \leq \frac{0.58}{c^2}. \tag{4.43}$$

7. Asymptotically as $c \to \infty$ [140, Prop. 17]

$$C_c^{(S)} \approx \frac{1}{2c^2}. \tag{4.44}$$

This is a very positive result as it shows that single decoding becomes as powerful as joint decoding for large collusion size and under the condition in 5.

4.6 Application to multimedia content fingerprinting

This section investigates the achievable rates and capacities when dealing with multimedia contents. The main difficulty is to model the watermarking layer and its related collusion strategies. They are mostly two types of content distortion :

- The colluders distort a content block by some regular content editing process, such as a lossy source compression. This degrades the performance of the watermarking decoder.

- The colluders merge two content block versions (one embedding symbol '1', the other symbol '0') when $0 < \sigma < c$. This merge is on the samples on the content block. In a first example, the forged block is a weighted average, sample-wise, of the block '1' with weight $w \in [0, 1]$ and the block '0' with weight $1 - w$. In a second example, the forged block is composed of a fraction $w \in [0, 1]$ of the samples of block '1' and a fraction $1 - w$ of the samples of the block '0'.

Merging and distortion can happen sequentially. They give rise to an extension of the marking assumption. While the codewords of the users remain binary, the pirated sequence may be composed of more than two symbols:

- Erasures: The watermarking layer may be unable to decode any binary symbol in a content block due to the distortion or the merging operations. An erasure is denoted by the symbol $\times$.
- Double detection: When two blocks are merged, the watermark decoder may be able to detect it as a merge, outputting both symbols '1' and '0'. A double detection is denoted by symbol d.

The collusion strategy is then modelled as a $4\times(c+1)$ matrix $\boldsymbol{\psi}_c$, where $\psi_{c,\sigma}(y) = \mathbb{P}(Y = y|\sigma)$ for $y \in \{0, 1, \mathsf{d}, \times\}$ and $\sigma \in \{0, \ldots, c\}$, s.t. $\sum_{y\in\{0,1,\mathsf{d},\times\}} \psi_{c,\sigma}(y) = 1$, $\forall\sigma \in \{0, \ldots, c\}$.

We assume that the colluders cannot tell which symbol is embedded in a block because the watermarking layer is a secret keyed primitive. The collusion is thus symbol symmetric:

$$\psi_{c,\sigma}(1) = \psi_{c,c-\sigma}(0). \tag{4.45}$$

This also implies the following rule for the probabilities of erasure and double detection:

$$\psi_{c,\sigma}(\times) = \psi_{c,c-\sigma}(\times), \tag{4.46}$$
$$\psi_{c,\sigma}(\mathsf{d}) = \psi_{c,c-\sigma}(\mathsf{d}). \tag{4.47}$$

Several models have been proposed in the literature encompassing mostly erasures [111] and more rarely double detections [155]. They restrict the power of the collusion by setting upper bounds on the probabilities $\{\psi_{c,\sigma}(\times)\}$ and $\{\psi_{c,\sigma}(\mathsf{d})\}$ [155, Th. 1]. The collusion is then free to pick one strategy complying to these constraints. We refuse these proposals. Our approach considers in more details the watermarking layer. It sets a strong relation between probabilities $\{\psi_{c,\sigma}(0), \psi_{c,\sigma}(1), \psi_{c,\sigma}(\times), \psi_{c,\sigma}(\mathsf{d})\}$. The set of collusion strategies is then much lower. The sequel proposes three families of collusion strategies depending on the watermarking layer.

4.6.1 Copy and Distort

These collusion strategies proceed in two stages:

1. The colluders create the forgery by sequentially copying-pasting one of their blocks. This is the way a collusion process was devised so far. There is no double detection as the blocks on the pirated copy always contain only one symbol.
2. This pirated copy is then distorted by some multimedia processing such as a lossy source coding, a low pass-filtering, etc. This second step produces erasures but no double detection.

We assume that the erasures happen at random in the pirated copy. Denote $\mathbb{P}(\times) := \zeta$ the probability that the last step produces an erasure. Then, $\psi_{c,\sigma}(\times) = \zeta, \forall \sigma \in \{0, \ldots, c\}$. This shows that erasures are independent from the colluders symbol: $\mathbb{P}(\times | X_{\mathsf{col}} = x) = \zeta$. Indeed the collusion strategy follows for $\forall \sigma \in \{0, \ldots, c\}$:

$$\begin{aligned} \psi_{c,\sigma}(0) &= (1-\zeta)(1-\theta_{c,\sigma}), & (4.48)\\ \psi_{c,\sigma}(1) &= (1-\zeta)\theta_{c,\sigma}, & (4.49)\\ \psi_{c,\sigma}(\times) &= \zeta & (4.50)\\ \psi_{c,\sigma}(\mathsf{d}) &= 0, & (4.51) \end{aligned}$$

where $\boldsymbol{\theta}_c$ is a binary collusion strategy compliant with the marking assumption. Then, it can be shown that for joint or single decoder:

$$R(P, \boldsymbol{\psi}_c) = (1-\zeta) R(P, \boldsymbol{\theta}_c). \quad (4.52)$$

An interpretation is that the erasures leak no clue to identify the collusion. They are useless and they can be removed from the pirated sequence at the decoding side. This leaves on expectation $m' = m(1-\zeta)$ symbols. This symbol dropping artificially increases the rate of the tracing scheme going from $R = {}^{\log n}/_{m}$ to $R' = {}^{\log n}/_{m'} = {}^{R}/_{1-\zeta}$. Equation 4.52 holds for any collusion strategy $\boldsymbol{\theta}_c$, therefore the capacity of this family is $C_c(\zeta) := (1-\zeta)C_c$. Since the erasures are ignored at the decoding side, the probabilities of accusation errors are asymptotically vanishing if $R' < C_c$ (the capacity of a single or a joint decoder) which in turn constrains $R < (1-\zeta)C_c$.

4.6.2 Merge and Distort

The collusion strategies of this second family also proceed in two stages:

1. The colluders first merge their content blocks sample-wise. This can be done as soon as the collusion has the two different blocks, *i.e.* when $0 < \sigma < c$. The collusion strategy is modelled by the vector $\mathbf{w}_c = (w_{c,0}, \ldots, w_{c,c})^\top$, where $w_{c,\sigma}$ is the merging weight used when the collusion has σ blocks '1' over c. Remember that $w_{c,\sigma} = 0$ (resp. $w_{c,\sigma} = 1$) means that the colluders actually don't mix but only copy content block '0' (resp. '1'). Obviously, $w_{c,0} = 1 - w_{c,c} = 0$. A merging weight $0 < w < 1$ may give birth to double detections or erasures.

2. This pirated copy is later on distorted by some multimedia processing such as a lossy source coding, a low pass-filtering, etc. This second step may introduce more erasures.

For a given $\sigma \in \{0, \ldots, c\}$, the four probabilities $\{\psi_{c,\sigma}(0), \psi_{c,\sigma}(1), \psi_{c,\sigma}(\times), \psi_{c,\sigma}(\mathsf{d})\}$ depend on the merging parameter $w_{c,\sigma}$. For compliance with the 'Copy and Distort' family, we impose that:

$$\begin{aligned} (\psi_{c,\sigma}(0), \psi_{c,\sigma}(1), \psi_{c,\sigma}(\times), \psi_{c,\sigma}(\mathsf{d})) &= (1-\zeta, 0, \zeta, 0) \text{ if } w_{c,\sigma} = 0, & (4.53)\\ (\psi_{c,\sigma}(0), \psi_{c,\sigma}(1), \psi_{c,\sigma}(\times), \psi_{c,\sigma}(\mathsf{d})) &= (0, 1-\zeta, \zeta, 0) \text{ if } w_{c,\sigma} = 1. & (4.54) \end{aligned}$$

Since $w_{c,0} = 1 - w_{c,c} = 0$, there is no double detection when the collusion sees only content blocks '0' (or only content blocks '1').

The dependence to the merging parameter is based on the watermarking layer. Appendix 14 proposes two models.

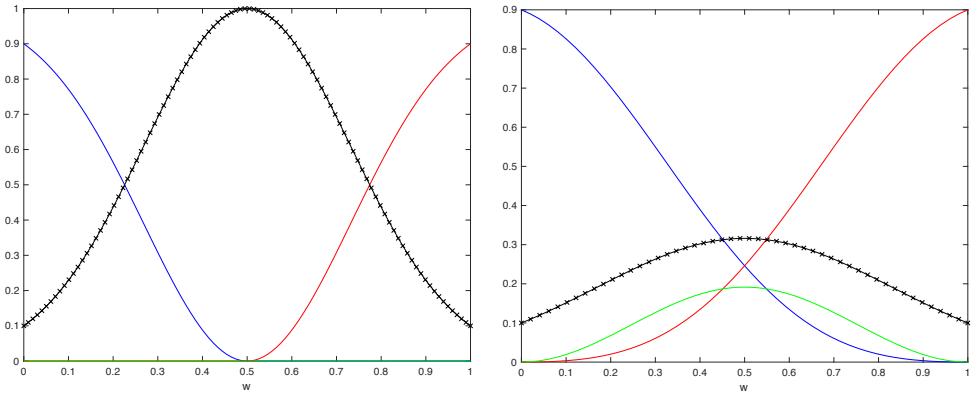


Figure 4.8: The probabilities $\{\psi_{c,\sigma}(y)\}$ for $y \in \{0, 1, \times, \mathsf{d}\}$ and $0 < \sigma < c$ as functions of the merging parameter $w \in [0, 1]$ for antipodal (left) and on-off keying (right) modulations.

Antipodal modulation

The first models holds for the antipodal modulation watermarking scheme. Binary symbols are embedded in a content block with a unique watermarking secret key k. The watermark decoding outputs a binary symbol or an erasure with probabilities:

$$\begin{aligned}
\mathbb{P}(Y = 0|w) &= 1 - \zeta^{|1-2w|_+^2}, && (4.55)\\
\mathbb{P}(Y = 1|w) &= 1 - \zeta^{|2w-1|_+^2}, && (4.56)\\
\mathbb{P}(Y = \times|w) &= \zeta^{(2w-1)^2}, && (4.57)\\
\mathbb{P}(Y = \mathsf{d}|w) &= 0, && (4.58)
\end{aligned}$$

where $|a|_+ = a$ if $a > 0$ and 0 otherwise. Fig. 4.8:left plots these probabilities as a function of $w \in [0, 1]$.

On-off Keying modulation

The other model is for the on-off keying modulation watermarking scheme, where symbol $X \in \{0, 1\}$ is embedded in a content block with a watermarking secret key k_X. In other words, the watermark detections are independent, outputting a binary symbol, an erasure or a double detection with probabilities:

$$\begin{aligned}
\mathbb{P}(Y = 0|w) &= \left(1 - \zeta^{(1-w)^2}\right)\zeta^{w^2}, && (4.59)\\
\mathbb{P}(Y = 1|w) &= \left(1 - \zeta^{w^2}\right)\zeta^{(1-w)^2}, && (4.60)\\
\mathbb{P}(Y = \times|w) &= \zeta^{w^2+(1-w)^2}, && (4.61)\\
\mathbb{P}(Y = \mathsf{d}|w) &= \left(1 - \zeta^{(1-w)^2}\right)\left(1 - \zeta^{w^2}\right). && (4.62)
\end{aligned}$$

Fig. 4.8:right plots these probabilities as a function of $w \in [0, 1]$.

These relations and the strategy for setting merging weights $\mathbf{w}_c$ fix the collusion strategy $\boldsymbol{\psi}_c$. Note that there is no violation of the marking assumption, $\psi_{c,0}(1) = \psi_{c,c}(0) = 0$, because $w_{c,0} = 1 - w_{c,c} = 0$.

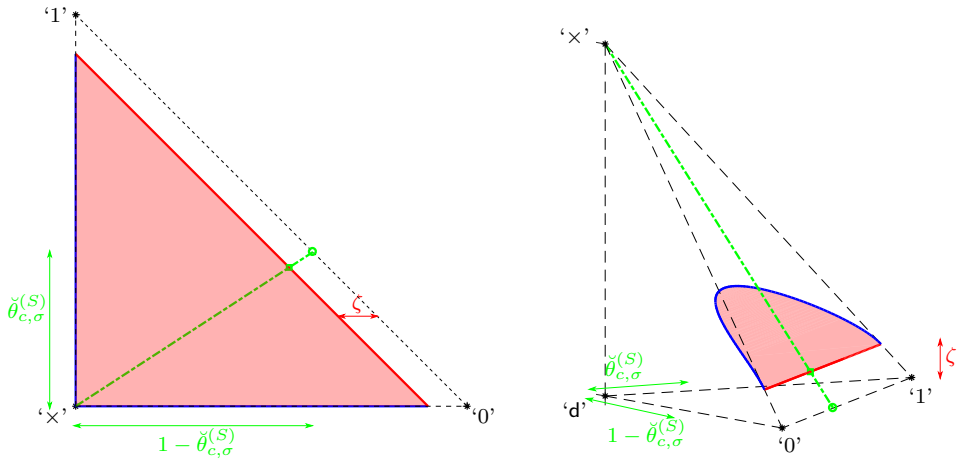


Figure 4.9: The locus of the point $\{\psi_{c,\sigma}(y)\}_{y\in\{0,1,\times,\mathsf{d}\}}$ with $0<\sigma<1$ for the 'Copy and Distort', 'Merge and Distort', and the 'Hybrid' strategies families. For the antipodal modulation (left), the points are drawn on the 2-simplex, for the on-off keying modulation (right), the points are drawn on the 3-simplex. The worst case attack is drawn as a green circle when $\zeta=0$, and a green square when $\zeta=0.1$.

4.6.3 Hybrid strategies

The last family of collusion strategies are the mix of the 'Copy and Distort' (Sect. 4.6.1) and the 'Merge and Distort' families (Sect. 4.6.2). When having σ '1' over c, the colluders flips a coin with bias $\mu_{c,\sigma}$. In one case, they use a 'Copy and Distort' strategy $\boldsymbol{\psi}_c^{(CD)}$, in the other case the use a 'Merge and Distort' strategy $\boldsymbol{\psi}_c^{(MD)}$. In the end, the collusion strategy follows: $\forall y\in\{0,1,\times,\mathsf{d}\}$

$$\psi_{c,\sigma}(y)=\mu_{c,\sigma}\psi^{(CD)}_{c,\sigma}(y)+(1-\mu_{c,\sigma})\psi^{(MD)}_{c,\sigma}(y). \tag{4.63}$$

For a given σ, $0<\sigma<c$, and $\zeta\geq 0$, $\psi_{c,\sigma}(y)$ is parametrized by $\mu_{c,\sigma}$, $\psi^{(MD)}_{c,\sigma}(1)$, and $w_{c,\sigma}$.

Note that for the antipodal modulation and $0<\sigma<c$, a clever choice of these parameters allows to put $\{\psi_{c,\sigma}(y)\}$ anywhere on the probability simplex provided that $\psi_{c,\sigma}(\mathsf{d})=0$ and $\psi_{c,\sigma}(\times)\geq\zeta$ (see Fig. 4.9:left). This is in strong contradiction with the Unreadable Digit Model which upper bounds $\psi_{c,\sigma}(\times)$.

4.6.4 Comparison

Figure 4.10 shows the achievable rates for the different collusion strategy families. The minimisation of the rate over the 'Hybrid' collusion strategies family (*i.e.* the area in light red in Fig. 4.9) always gives a strategy belonging to the 'Copy and Distort' family. Strategies of the 'Merge and Distort' family always produce a higher rate. This is especially true for the joint decoder.

Production of erasures is a double edged sword. The achievable rate decreases with parameter ζ as shown in Fig. 4.10 for the three families. But, equalling $\psi_{c,\sigma}(\times)=\zeta$ for all $\sigma\in\{1,\ldots,c-1\}$ as in the 'Copy and Distort' family is the best option for the collusion. This strategy is not possible in the 'Merge and Distort' family

Production of double detections is also a double edged sword. The 'Merge and Distort' family contains more harmful collusion strategies with the antipodal modulation. Yet, the worst case attack still belongs to the 'Copy and Distort' family, which does not produce any double detection.

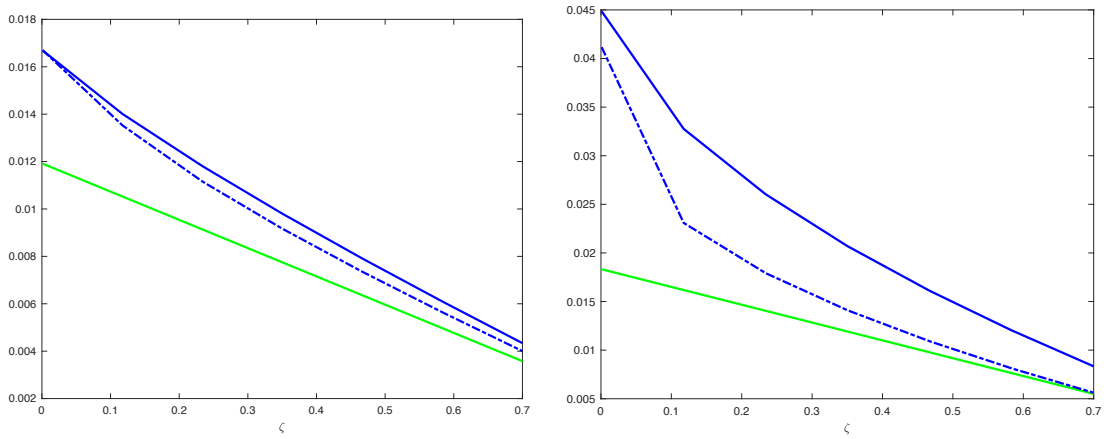


Figure 4.10: Achievable rates as a function of $\zeta \in [0, 0.7]$ for the 'Copy and Distort' (4.52) and 'Merge and Distort' (antipodal modulation in plain line, on-off keying modulation in dashed line) collusion strategy families ($c = 5$). Left: Single decoder $R^{(S)}(\check{\boldsymbol{\psi}}_c, f^{(T)})$, Right: Joint decoder $R^{(J)}(\check{\boldsymbol{\psi}}_c, f^{(T)})$.

The interpretation is that erasures and double detections may reveal a lot information to the decoder. Consider for instance the following probability (when properly defined):

$$\mathbb{P}(0 < \sigma < c | Y = y, P = p) = \frac{\sum_{\sigma=1}^{c-1} \psi_{c,\sigma}(y) \mathbb{P}(\sigma | P = p)}{\sum_{\sigma=0}^{c} \psi_{c,\sigma}(y) \mathbb{P}(\sigma | P = p)}. \tag{4.64}$$

For $y = \mathsf{d}$, $\mathbb{P}(0 < \sigma < c | Y = \mathsf{d}, P = p) = 1$, $\forall 0 < p < 1$, because $\psi_{c,0}(\mathsf{d}) = \psi_{c,c}(\mathsf{d}) = 0$. A double detection clearly reveals that the collusion has both symbols '1' and '0' at that block index. To avoid this, the collusion has to restrict $w_{c,\sigma} \in \{0, 1\}$ so that $\psi_{c,\sigma}(\mathsf{d}) = 0$ for all $\sigma \in \{0, \ldots, c\}$. On the contrary, for $y = \times$, $\mathbb{P}(0 < \sigma < c | Y = \times, P = p) = \mathbb{P}(0 < \sigma < c | P = p)$ if and only if $\psi_{c,\sigma}(\times) = \zeta$, $\forall 0 \leq \sigma \leq c$. This symbol $\times$ then does not reveal any information about σ. Again, in the 'Merge and Distort' family, this only happens by constraining $w_{c,\sigma} \in \{0, 1\}$. With this respect, the 'Copy and Distort' family allows much more freedom while enforcing constant $\psi_{c,\sigma}(\times) = \zeta$ and $\psi_{c,\sigma}(\mathsf{d}) = 0$, $\forall 0 \leq \sigma \leq c$.

4.7 Conclusion

The last section shows that the 'Copy and Distort' family produces attacks worst than the 'Merge and Distort' family. 'Copy and Distort' sums up as picking a collusion strategy $\boldsymbol{\theta}_c$ and applying a distortion introducing erasures with probability ζ. This justifies the fact that most of the content of this chapter is devoted to 'binary' collusion strategies $\boldsymbol{\theta}_c$. Distorting more increases ζ while damaging the quality of the forged content. Yet, the chapter assumes that the colluders have no fine control on ζ. That parameter was assumed to be constant over the blocks. A more challenging setup would allow the colluders to distort more some blocks than others, for instance those where $\sigma \in \{0, 1\}$ in order to 'equalize' the values $\{\psi_{c,\sigma}(\times)\}$.

Chapter 5

The decoders

This chapter focuses on the decoding part of the accusation processes, and especially on the computation of a score per user (or per group of users). The main feature of the score function is its discriminability, *i.e.* its power to statistically make a colluder score significantly different than an innocent score. The issue of thresholding the scores to decide who is guilty without accusing any innocent user (or almost) is postponed to the next chapter.

5.1 Introduction

There are indeed many accusation processes sharing different characteristics:

- **Single *vs.* Joint decoding.** A single decoder computes a score per user from the secret key $\mathbf{p}$, the pirated sequence $\mathbf{y}$, and the codeword of the user. A joint decoder computes a score for a group of users (pair, triplet, ...) from $\mathbf{p}$, $\mathbf{y}$, and the codewords of these users.

- **Fixed *vs.* adaptive decoders.** All decoders are oblivious to the collusion size and strategy. However, some first try to infer information about the collusion parameters by analyzing $\mathbf{p}$ and $\mathbf{y}$. These informations then adapt the score function in order to match the collusion strategy. On contrary, fixed decoders never change their scoring functions.

- **Linear *vs.* non-linear.** A linear score function computes a sum over the m indices. For a single decoder:

$$s(\mathbf{x}_j, \mathbf{y}, \mathbf{p}) = \sum_{i=1}^{m} U(x_{j,i}, y_i, p_i). \tag{5.1}$$

  This is for instance the well-known Tardos-Škorić score function (see Sect. 2.3.5). Examples of non-linear decoder are the generalized linear decoder (see Sect. 5.2.3), which aggregates several linear decoders, and the Maximum empirical Mutual Information decoder [136, Sect. 4.3].

- **Deterministic *vs.* probabilistic decoding.** A probabilistic decoder runs a stochastic simulation to identify colluders. This is for instance the Markov Chain Monte Carlo algorithm of Sect. 5.4.3.

One can also compose iterative decoders based on single and/or joint decoding, and collusion strategy estimation. Typical iteration procedures are the following:

- The decoder computes scores to infer the guiltiness of the users. This allows a more accurate estimation of the collusion strategy (for an adaptive decoder), which in turn allows a better score function. This is typically an Expectation-Maximization algorithm (see Sect. 5.4.1).
- The decoder computes scores and accuses a user because his score was above the threshold. His/her codeword is used as a side-information to compute a better score function. This might be sufficient to accuse another colluder (see Sect. 4.3.3).
- The decoder computes joint decoding scores for group of size ℓ. It selects a subset of users having the biggest scores, and it computes joint decoding scores for the size $(\ell + 1)$ within this subset. This is typically a list-decoding approach which is iteratively refined by more powerful but more complex joint decoders (see Sect. 5.4.4).

In practice, these iterative decoders are very powerful. However, there is no way to theoretically prove their efficiency, *i.e.* how much they reduce the length of the codes, what is their worst case attacks...

5.2 Single decoders

A single decoder is an accusation process computing a score per user based on a single codeword. This section presents several implementations proposed in the literature. The MAP (Maximum A Posteriori) based on the LLR (Log-Likelihood Ratio) is the optimum score function but it depends on the collusion size and strategy. Obviously these parameters are unknown in practice, but its design motivates alternative score functions. They are classified into two categories: the 'fixed' and 'adaptive' decoders. The performances of these single decoders are compared in Fig. 5.2.

5.2.1 The optimum single decoder: MAP

In essence, a single decoder sequentially checks the guiltiness of users via a hypotheses test. For a given user j, the decoder has two hypotheses: $\mathcal{H}_0$ - j is innocent; $\mathcal{H}_1$ - j is a colluder. According to the Neyman-Pearson lemma [102, Th. 12.7.1], under the constraint that the probability of wrongly accusing user j is below $\mathbb{P}_{\mathsf{fp}}$, the best test is based on the (log-)likelihood ratio (LLR) of the observation of $\mathbf{x}_j$ conditioned on $(\mathbf{y}, \mathbf{p})$.

- $\mathcal{H}_0$. This codeword has been created by the code generator knowing $\mathbf{p}$ and the pirated sequence has been forged independently from $\mathbf{x}_j$:

$$\mathbb{P}\left(\mathbf{X} = \mathbf{x}_j | \mathcal{H}_0\right) = \mathbb{P}\left(\mathbf{X} = \mathbf{x}_j | \mathbf{p}\right). \tag{5.2}$$

- $\mathcal{H}_1$. This codeword has been created by the code generator knowing $\mathbf{p}$ and the pirated sequence has been forged from $\mathbf{x}_j$ (among others). Thanks to the Bayes' rule:

$$\mathbb{P}\left(\mathbf{X} = \mathbf{x}_j | \mathcal{H}_1\right) = \mathbb{P}\left(\mathbf{X} = \mathbf{x}_j | \mathbf{y}, \mathbf{p}, \boldsymbol{\theta}_c\right) = \frac{\mathbb{P}\left(\mathbf{Y} = \mathbf{y} | \mathbf{x}_j, \mathbf{p}, \boldsymbol{\theta}_c\right) \mathbb{P}\left(\mathbf{X} = \mathbf{x}_j | \mathbf{p}\right)}{\mathbb{P}\left(\mathbf{Y} = \mathbf{y} | \mathbf{p}, \boldsymbol{\theta}_c\right)}. \tag{5.3}$$

Thanks to the mutual independence of the symbols within sequences (due to the codeword generation and the model of the collusion attack), the log of the likelihood ratio gives a linear score (5.1) based on the function:

$$U^\star(x, y, p | \boldsymbol{\theta}_c) := \log \frac{\mathbb{P}\left(X = x | \mathcal{H}_1\right)}{\mathbb{P}\left(X = x | \mathcal{H}_0\right)} = \log \frac{\mathbb{P}\left(Y = y | x, p, \boldsymbol{\theta}_c\right)}{\mathbb{P}\left(Y = y | p, \boldsymbol{\theta}_c\right)}. \tag{5.4}$$

This decoding rule is often named as the Neyman-Pearson decoder, the Maximum Likelihood decoder, or the Maximum A Posteriori (MAP) decoder[1] like in digital communications. Its complexity is in $O(mn)$, as for the Tardos-Škorić decoder. To apply the expressions of (4.6) and (4.7), we however need the value of the real collusion parameter $\boldsymbol{\theta}_c$. This decoder is out of reach in practice.

It is interesting to see how much we could gain in performances with the MAP compared to the Tardos-Škorić single decoder presented in Sect. 2.3.5. Fig. 5.1 illustrates the gap in performance for two different attacks with $c = 5$ colluders: the less harmful is the minority vote $\boldsymbol{\theta}_5^{(\min)} = (0, 1, 1, 0, 0, 1)^T$, and the most aggressive $\breve{\boldsymbol{\theta}}_5^{(S)} = (0, 0.593, 0, 1, 0.407, 1)^T$. This collusion strategy maximizes (resp. minimizes) $I(Y; X|P, \boldsymbol{\theta}_5)$ for $P \sim f^{(T)}$. In this simulation, scores are computed and compared to a threshold τ, and

$$\mathbb{P}_{\mathsf{fp}}(\tau) = \mathbb{P}(s(\mathbf{X}_{\mathsf{inn}}, \mathbf{Y}, \mathbf{p}) > \tau), \tag{5.5}$$

$$\mathbb{P}_{\mathsf{fn}}(\tau) = \mathbb{P}(s(\mathbf{X}_{\mathsf{col}}, \mathbf{Y}, \mathbf{p}) \leq \tau), \tag{5.6}$$

are error probabilities *per user* for a given $\mathbf{p}$. These probabilities are estimated thanks to a Rare Event technique explained in Chapter 6.

Fig. 5.1:top shows how $\mathbb{P}_{\mathsf{fp}}(\tau)$ and $\mathbb{P}_{\mathsf{fn}}(\tau)$ evolve with parameter τ. The Tardos-Škorić decoder has the remarkable property that these probability functions are almost not affected by the collusion strategy, as justified in Chapter 2. On the contrary, it is challenging to find a threshold τ for the MAP decoder, since the distribution of the score varies a lot with the collusion strategy. Fig. 5.1:bottom shows the Detection Error Tradeoff (DET), *i.e.* the probability of false negative as a function of the probability of false positive. When the attack is very aggressive, the gap is small. When the attack is not aggressive, the MAP decoder outperforms by several orders of magnitude the Tardos-Škorić decoder. This is the price to be paid for having performance 'invariant' to the collusion strategy, which is the main rationale underlying the design of the Tardos-Škorić decoder (see Chapter 2).

5.2.2 Fixed linear single decoders

This section presents three fixed linear single decoders besides the most well-known decoder of this class, the Tardos-Škorić decoder, which was already covered in Sect. 2.3.5. The complexity of a fixed single decoder is in $O(mn)$.

Be prepared for the worst

The article [42] makes the connection between traitor tracing and communication through a discrete memoryless compound channel. In brief, a compound channel is a set $\boldsymbol{\Theta}$ of channels (discrete set or a continuum of channels). The encoder emits a codeword which passes through one of the channels of the set $\boldsymbol{\Theta}$. The decoder must recover the codeword without knowing the actual communication channel. Yet, the decoder knows the set $\boldsymbol{\Theta}$ of channels. E. Abbe and L. Zheng have shown in [80] that, under a certain condition (*i.e.* the set $\boldsymbol{\Theta}$ is one-sided [80, Def. 3]), there exists a linear universal decoder. Universal means in information theory that this decoder (together with the optimal encoder) achieves the compound channel capacity. Indeed, this linear decoder is simple: it is the MAP decoder matching the worst channel of the set $\boldsymbol{\Theta}$. This worst channel is defined as the minimizer of the mutual information between the transmitted and

[1] MAP and ML are the same decoder because the a priori probability of being a colluder is uniform over the set of users.

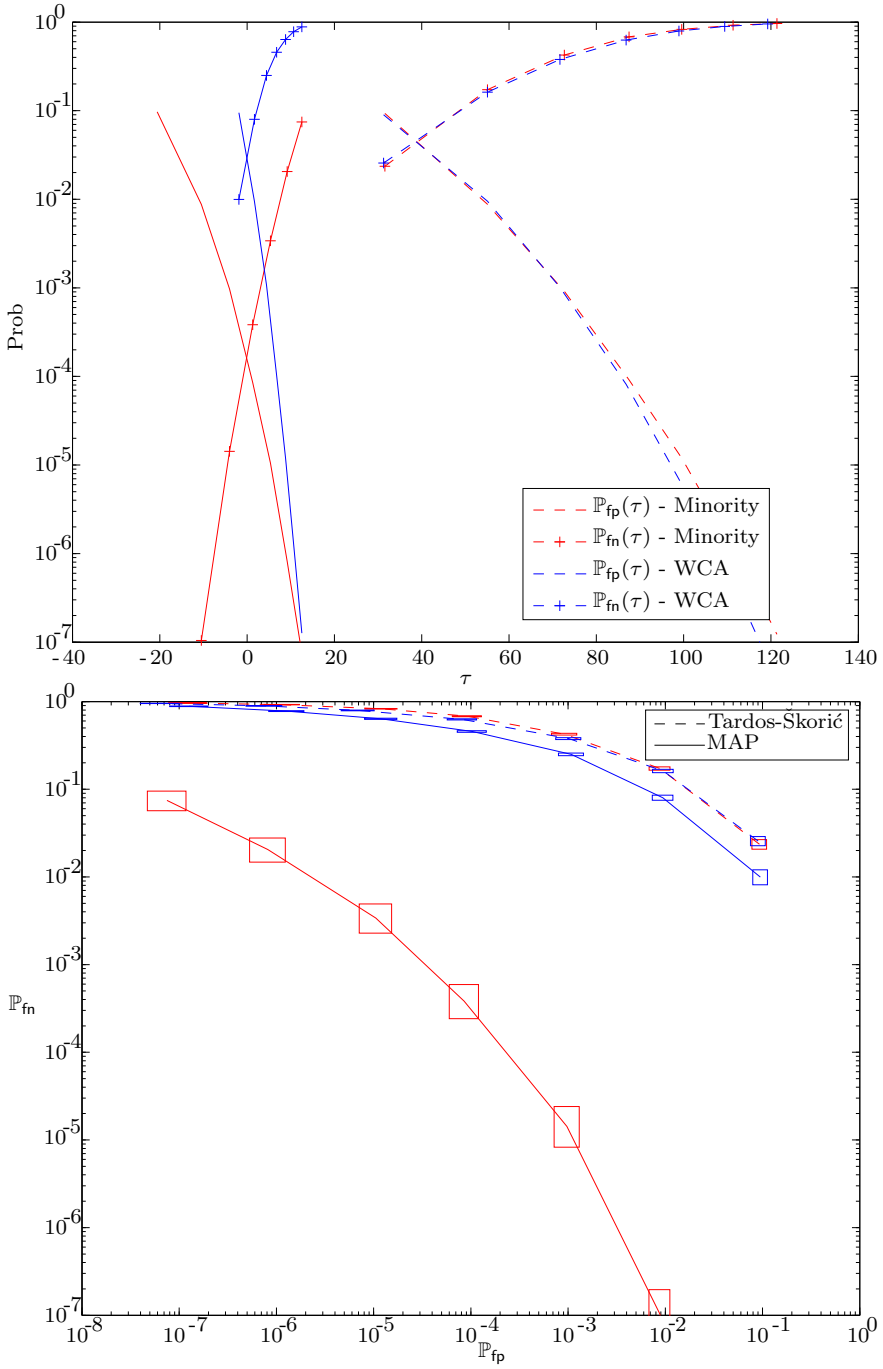


Figure 5.1: Top: $\mathbb{P}_{\mathsf{fp}}(\tau)$ and $\mathbb{P}_{\mathsf{fn}}(\tau)$) for two collusion attacks with $c = 5$: The minority vote and the worst case attack against a single decoder: Tardos-Škorić decoder (dashed), optimum MAP decoder (plain). Bottom: DET plot: $\mathbb{P}_{\mathsf{fn}}$ as a function of $\mathbb{P}_{\mathsf{fp}}$. Boxes represent confidence interval at 95%.

received symbols. If the actual communication channel is the worst channel, then the achievable rate is the compound channel capacity (*i.e.* the capacity of this worst channel) because the decoder matches this channel. If the actual communication channel is not the worst channel, the achievable rate is lower or equal to the *capacity of this channel* but, $\mathbf{\Theta}$ being one-sided, the achievable rate is indeed greater or equal to the *capacity of the compound channel*.

When the worst case happens, this decoder is optimal. Otherwise, it is guaranteed to perform reasonably well in the sense that it performs better or equally than when facing the worst case.

The application to Tardos codes is made in [42]. The set of collusion strategy $\mathbf{\Theta}_c$ is one-sided for any c when P is an absolutely continuous random variable ($I(Y; X_{\mathsf{col}}|P, \boldsymbol{\theta})$ has a global minimum and $\mathbf{\Theta}_c$ is convex [80, Lemma. 4]). If the decoder were knowing c but not $\boldsymbol{\theta}_c$, it would use the linear MAP decoder tuned for $\breve{\boldsymbol{\theta}}_c^{(S)}$ as defined in (4.26).

Suppose now that the decoder knows that the collusion size is not bigger than $c_{\max}$. This makes sense in practice: given the length m and the number n of users, the decoder limits itself to chase at most $c_{\max}$ colluders. If $c \leq c_{\max}$, the decoder aims at identifying colluders. If not, the decoder cannot pretend to meet this goal as it is hopeless to trace a bigger collusion. The finite union $\cup_{c=1}^{c_{\max}} \mathbf{\Theta}_c$ being also one-sided, the MAP decoder tuned on $\breve{\boldsymbol{\theta}}_{c_{\max}}^{(S)}$, which is the worst collusion strategy over this finite union, is theoretically sound:

$$U(x, y, p) = \log \frac{\mathbb{P}\left(Y = y | x, p, \breve{\boldsymbol{\theta}}_{c_{\max}}^{(S)}\right)}{\mathbb{P}\left(Y = y | p, \breve{\boldsymbol{\theta}}_{c_{\max}}^{(S)}\right)}. \tag{5.7}$$

Laarhoven score function

The worst collusion strategy $\breve{\boldsymbol{\theta}}_{c_{\max}}^{(S)}$ has to be computed with a numerical minimizer. The interleaving attack is not a priori the worst attack but it gets closer for large $c_{\max}$ (see Sect. 4.4). The decoder from T. Laarhoven [126, Eq. (2)] is indeed based on the MAP score function tuned on the interleaving attack $\boldsymbol{\theta}_{c_{\max}} = (0, {}^1\!/_{c_{\max}}, \ldots, {}^{(c_{\max}-1)}\!/_{c_{\max}}, 1)$. This simplifies to

$$U(x, y, p) = \begin{cases} \log\left(1 + \frac{1}{c_{\max}} \left(\frac{1-p}{p}\right)^{(2y-1)}\right) & \text{if } x = y, \\ \log(1 - {}^1\!/_{c_{\max}}) & \text{if } x \neq y. \end{cases} \tag{5.8}$$

This is optimal in the sense that it is capacity-achieving for large $c_{\max}$ [126, Prop. 3]. Moreover, there is no longer need of a cut-off parameter [126, Th. 4]. A conjecture is that this last property indeed holds for any decoder based on a Log-Likelihood Ratio (5.4).

Oosterwijk score function

The score function from Oosterwijk *et al.* [141, Eq. (43)] can be seen as the first order approximation of Laarhoven score function when $c_{\max}$ is large:

$$U(x, y, p) = \begin{cases} \left(\frac{1-p}{p}\right)^{2y-1} & \text{if } x = y, \\ -1 & \text{if } x \neq y. \end{cases} \tag{5.9}$$

Like the Tardos-Škorić linear decoder, a cutoff $t > 0$ is needed to bound its amplitude. This is optimal in the sense that it yields a saddle-point of the figure of merit $\mathbb{E}(\sum_{j \in \mathcal{C}} S_j)/\sqrt{\mathbb{V}(S_{j \notin \mathcal{C}})}$ [141, Th. 2], and that it is capacity-achieving for large c [141, Prop. 17].

5.2.3 Fixed non linear single decoders

This section gives examples of non linear single decoders.

Be prepared for the worst cases

In [80], the definition of the compound channel is also extended to a finite set of compound channels. This eases the application to traitor tracing: When the decoder bets that the collusion size is not bigger than $c_{\max}$, $\cup_{c=1}^{c_{\max}} \boldsymbol{\Theta}_c$ is a finite set of compound channels.

The authors of [80] also introduce the generalized linear decoder: It computes the linear scores for each one-sided compound channel and then aggregates these metrics into a final score by a max pooling:

$$s(\mathbf{x}, \mathbf{y}, \mathbf{p}) = \max(s^{(1)}(\mathbf{x}, \mathbf{y}, \mathbf{p}) \dots, s^{(c_{\max})}(\mathbf{x}, \mathbf{y}, \mathbf{p})), \tag{5.10}$$

where $s^{(k)}(\cdot)$ is the linear score based on $U^\star(x, y, p|\breve{\boldsymbol{\theta}}_k^{(S)})$ in (5.4) with $\breve{\boldsymbol{\theta}}_k^{(S)}$ being the worst case attack for a collusion of size k. A crucial point is that only scores in the form of Log-Likelihood Ratio (5.4) should be aggregated. This decoder proposed in [42, Sect. III.B.1] is capacity achieving as an application to [80, Th. 1] provided $c \leq c_{\max}$. The complexity is in $O(mnc_{\max})$.

Be prepared for the expected cases

M. Desoubeaux and C. Herzet take the point of view of Bayesian statistics. Knowing c, the decoder assumes the least informative prior, *i.e.* Jeffrey prior (see Sect. 3.2.1), on the components of $\boldsymbol{\theta}_c$ (except for $\theta_{c,0}$ and $\theta_{c,c}$ which are imposed by the marking assumption). The expectation of the collusion strategy w.r.t. this prior gives the 'Coin flip' attack, $\boldsymbol{\theta}_c = (0, 1/2, \dots, 1/2, 1)$, since Jeffrey prior has a symmetric distribution w.r.t. $1/2$.

Not knowing c, the decoder bets that there are at most $c_{\max}$ colluders and that the prior on the collusion size is uniform over $\{1, \dots, c_{\max}\}$. This is again the least informative prior about the collusion size. The Bayesian approach gives:

$$s(\mathbf{x}, \mathbf{y}, \mathbf{p}) = \log \left(\sum_{k=1}^{c_{\max}} k. \exp \left(s_j^{(k)}(\mathbf{y}, \mathbf{x}, \mathbf{p}) \right) \right), \tag{5.11}$$

where $s_j^{(k)}(\cdot)$ is the linear score function based on the LLR (5.4) and tuned for the 'Coin flip' attack of size k. This decoder is optimal in the sense of a Bayesian decoder with the least informative prior on the collusion attack [109]. Its complexity is in $O(mnc_{\max})$.

Maximum Mutual Information

The last example of a non-linear single decoder comes from information theory [136, Sect. 4.3] or [159, Sect. V]. The score of user j is the empirical mutual information between $\mathbf{y}$ and his/her codeword $\mathbf{x}_j$, knowing $\mathbf{p}$. When P is not a discrete random variable, it needs to be discretized into bins $\{B_k\}_{k=1}^K$, which form a partition of $[0, 1]$, in order to compute the empirical joint probabilities $\{\hat{\mathbb{P}}(Y, X|p \in B_k)\}_{k=1}^K$. Fig. 5.2 illustrates this scheme with a uniform partition of $K = 10$ bins. It is a good choice against the majority attack (top) but it performs worse than the Tardos-Škorić decoder against $\breve{\boldsymbol{\theta}}_5^{(S)}$ (bottom). This shows that the partition of $[0, 1]$ is a real issue. This decoder is recommended for a discrete random variable P. Its complexity is in $O(Kmn)$, but its non linearity makes its implementation slower than the generalized linear decoder.

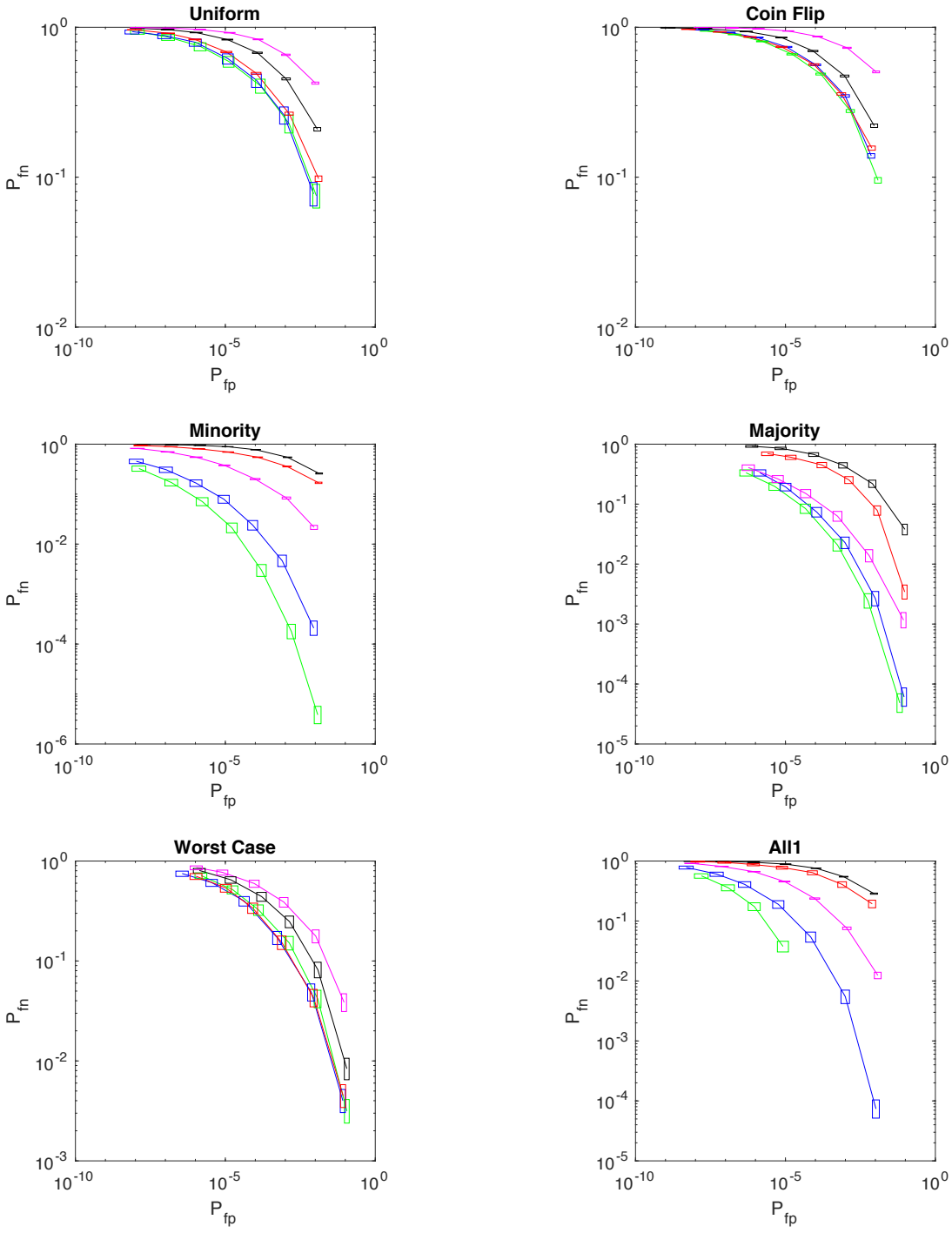


Figure 5.2: DET plot for several fixed single decoders; $m = 512$, $c = 5$, $c_{\max} = 10$. Tardos-Škorić as defined in Sect. 2.3.5 in **black**. MAP is the optimal single decoder (5.4). MAP tuned on $\tilde{\boldsymbol{\theta}}_{c_{\max}}$ (see Sect. 5.2.4). MMI is the Maximum Mutual Information decoder where $[0, 1]$ is partitioned into 10 uniform bins. Generalized linear decoder tuned on worst case attacks (5.10) up to $c_{\max} = 10$.

5.2.4 Adaptive decoders

Adaptive decoders proceed in two steps. They first analyse the received sequence $\mathbf{y}$ knowing the secret $\mathbf{p}$ to deduce some knowledge about the collusion strategy. Then, they compute score thanks to a scoring function adapted to what it learned. This approach is sometimes called 'Learn and Match'.

Estimation of the collusion strategy knowing c

If the decoder were knowing c but not $\boldsymbol{\theta}_c$, it could first estimate the collusion strategy $\boldsymbol{\theta}_c$ from the observation $(\mathbf{y}, \mathbf{p})$. This is possible thanks to the identifiability property (see Sect. 4.2.3 item 2). The Maximum Likelihood estimator is given by:

$$\hat{\boldsymbol{\theta}}_c^{(\mathsf{MLE})} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_c} \sum_{i=1}^{m} y_i \log(\Pi(p_i;\boldsymbol{\theta})) + (1-y_i)\log(1-\Pi(p_i;\boldsymbol{\theta})), \tag{5.12}$$

with $\Pi(\cdot)$ defined by (4.6). The estimate is given by a numerical solver in practice.

Another possibility is to resort to the Expectation-Maximisation algorithm as the law of Y_i given p_i is a mixture of Bernoulli distributions. Let $\Sigma_i = \sum_{j\in\mathcal{C}} X_{j,i}$, *i.e.* the number of symbol '1' the colluders have at index i, be a latent variable. The E-M algorithm starts from a random guess of the collusion strategy, $\boldsymbol{\theta}_c^{(0)} \in \Theta_c$, and then iterates the following two steps:

E-step At iteration t, this step evaluates the law of $\Sigma_i \in \{0,\dots,c\}$ for a fixed collusion model $\boldsymbol{\theta}_c^{(t-1)}$ thanks to the Bayes rule:

$$\begin{aligned} T_{i,\sigma}^{(t)} &:= \mathbb{P}(\Sigma_i = \sigma | y_i, p_i, \boldsymbol{\theta}_c^{(t-1)}) = \mathbb{P}(Y_i = y_i | \Sigma_i = \sigma, p_i, \boldsymbol{\theta}_c^{(t-1)}) \frac{\mathbb{P}(\Sigma_i = \sigma | p_i)}{\mathbb{P}(Y_i = y_i | p_i, \boldsymbol{\theta}_c^{(t-1)})} \\ &= (\theta_{c,\sigma}^{(t-1)})^{y_i}(1-\theta_{c,\sigma}^{(t-1)})^{1-y_i} \frac{\mathbb{P}(\Sigma_i = \sigma | p_i)}{\sum_{\sigma'=0}^{c} (\theta_{c,\sigma'}^{(t-1)})^{y_i}(1-\theta_{c,\sigma'}^{(t-1)})^{1-y_i}\mathbb{P}(\Sigma_i = \sigma' | p_i)}, \end{aligned} \tag{5.13}$$

where $\mathbb{P}(\Sigma_i = \sigma | p_i)$ is given in Sect. 4.2.1 (here, we use the binomial model, but the hypergeometrical is possible as well).

M-step At iteration t, this step outputs a new estimation $\boldsymbol{\theta}_c^{(t)}$ knowing the probabilities $\{T_{i,\sigma}^{(t)}\}$ of the latent variables. This is done by maximizing function $Q(\boldsymbol{\theta}_c)$ defined as the expectation of the log likelihood:

$$Q(\boldsymbol{\theta}_c) = \sum_{i=1}^{m}\sum_{\sigma=0}^{c} T_{i,\sigma}^{(t)} \log(\mathbb{P}(Y_i = y_i | \Sigma_i = \sigma, \boldsymbol{\theta}_c)). \tag{5.14}$$

The $(c-1)$ parameters $\theta_{c,1},\dots,\theta_{c,c-1}$ can be processed independently while $\theta_{c,0}^{(t)} = 1-\theta_{c,c}^{(t)} = 0$ if the marking assumption holds. The second derivative being negative, the maximizer cancels the gradient given by:

$$\frac{\partial Q(\boldsymbol{\theta}_c)}{\partial \theta_{c,\sigma}} = \sum_{i=1}^{m} T_{i,\sigma}^{(t)} \left(\frac{y_i}{\theta_{c,\sigma}} - \frac{1-y_i}{1-\theta_{c,\sigma}} \right), \tag{5.15}$$

so that the maximizer equals

$$\theta_{c,\sigma}^{(t)} = \frac{\sum_{i=1}^{m} y_i T_{i,\sigma}^{(t)}}{\sum_{i=1}^{m} T_{i,\sigma}^{(t)}}. \tag{5.16}$$

We can also take into account symbol-symmetric collusion strategy as follows:

$$\theta_{c,\sigma}^{(t)} = \frac{\sum_{i=1}^{m} y_i T_{i,\sigma}^{(t)} + (1-y_i) T_{i,c-\sigma}^{(t)}}{\sum_{i=1}^{m} T_{i,\sigma}^{(t)} + T_{i,c-\sigma}^{(t)}}. \tag{5.17}$$

The algorithm iterates these two steps until the maximum of function Q no longer increases (convergence to a local maximum) or during a fixed number of iterations to keep the complexity under control. The E-M algorithm can also be generalized to handle collusion strategies producing erasures and double detections (see Sect. 4.6) or soft outputs [35, Sect. V].

Estimation of the collusion strategy not knowing c

If the collusion strategy were correctly estimated, the decoder could use the LLR (5.4) tuned on $\hat{\boldsymbol{\theta}}_c$. Nevertheless, the collusion size is not known at the decoding side, and this prevents identifiability of both c and $\boldsymbol{\theta}_c$ (see Sect. 4.2.3).

This is where the concept of compound channel turns the situation around. Denote by $\mathcal{E}(\boldsymbol{\theta}_c) = \{\boldsymbol{\theta} | \Pi(p;\boldsymbol{\theta}) = \Pi(p;\boldsymbol{\theta}_c), \forall p \in (0,1)\}$. Sect. 4.2.3 states that this set is not restricted to the singleton $\{\boldsymbol{\theta}_c\}$: For any $c' > c$, $\mathcal{E}(\boldsymbol{\theta}_c) \cap \boldsymbol{\Theta}_{c'} = \{\tilde{\boldsymbol{\theta}}_{c'}\}$. Assuming at the decoding side that $c < c_{\max}$ is equivalent to assuming that $\mathcal{E}(\boldsymbol{\theta}_c) \cap (\cup_{c=1}^{c_{\max}} \boldsymbol{\Theta}_c)$ is not empty. It turns out that this set is one-sided [42, Appendix] and that its worst collusion attack is $\tilde{\boldsymbol{\theta}}_{c_{\max}}$. This theoretically justifies the following adaptive single decoder parametrised by $c_{\max}$: i) estimate $\hat{\boldsymbol{\theta}}_{c_{\max}}$, ii) use the LLR (5.4) tuned on this estimate. A generalized linear decoder also stems from this analysis: i) for some $c' \in \{2, \ldots, c_{\max}\}$, estimate $\hat{\boldsymbol{\theta}}_{c'}$, ii) for each c', compute scores using the LLR tuned on $\hat{\boldsymbol{\theta}}_{c'}$, iii) aggregate these scores per user with max pooling as in (5.10).

5.3 Joint decoders

A joint decoder computes a score per group of users, where the size of group is $\ell > 1$. Its complexity is in $O(n^\ell)$ since there are $\binom{n}{\ell} = O(n^\ell)$ groups. This is the reason why its achievable rate is $\ell^{-1} I(Y; (X_{\mathsf{col},1}, \ldots, X_{\mathsf{col},\ell}) | P, \boldsymbol{\theta}_c)$. Sect. 13.1.2 shows that this rate is greater or equal to the achievable rate of a single decoder, the ultimate case being when $\ell = c$. This makes joint decoding theoretically appealing, but its complexity is not tractable whenever n is large.

Since there is no ordering of the users within a group $g \subset \{1, \ldots, n\}$, a sufficient statistic is the sum $\boldsymbol{\rho}_g \in \{0, \ldots, \ell\}^m$ of their codewords: $\rho_{g,i} = \sum_{j \in g} x_{j,i}$, $\forall i \in \{1, \ldots, m\}$. For $\ell \leq c$, The MAP joint decoder is based on the LLR:

$$s_g = \sum_{i=1}^{m} U(y_i, \rho_{g,i}, p_i | \boldsymbol{\theta}_c), \tag{5.18}$$

$$U(y, \rho, p | \boldsymbol{\theta}_c) = \log \frac{\mathbb{P}(Y = y | \rho, \ell, p, \boldsymbol{\theta}_c)}{\mathbb{P}(Y = y | p, \boldsymbol{\theta}_c)}, \tag{5.19}$$

with $\mathbb{P}(Y = y | \rho, \ell, p, \boldsymbol{\theta}_c)$ defined in (4.13).

The ideas developed for the single linear decoder also apply to joint decoding. The MAP joint decoder can be tuned on:

- $\breve{\boldsymbol{\theta}}_{c_{\max}}^{(J,\ell)}$, *i.e.* the collusion strategy of size $c_{\max}$ minimizing $I(Y; (X_{\mathsf{col},1}, \ldots, X_{\mathsf{col},\ell}) | P, \boldsymbol{\theta}_{c_{\max}})$ (note that it is a priori different from $\breve{\boldsymbol{\theta}}_{c_{\max}}^{(S)}$).

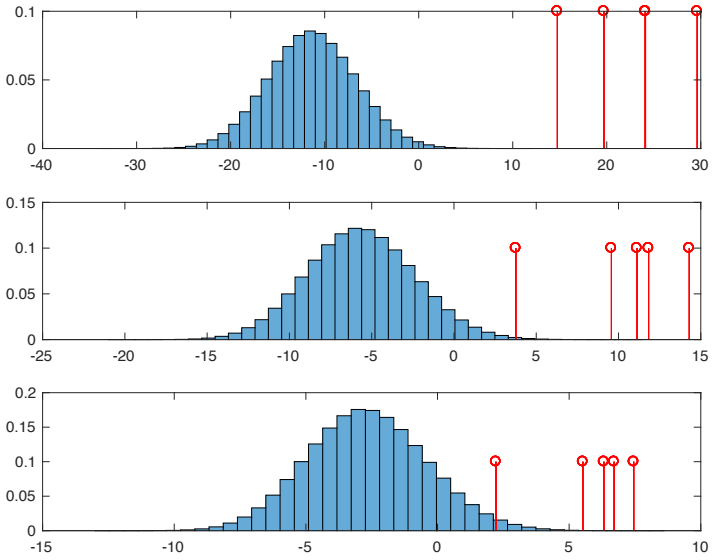


Figure 5.3: Empirical distribution of S_{inn} computed over a single run with $n = 2.10^6$. The collusion strategy is a majority vote with $c = 5$. The decoder is adaptive based on the LLR tuned on $\hat{\boldsymbol{\theta}}_{c_{\max}}$ with $c_{\max} = 8$. The score of the c colluders for this run are represented in red. The length of the code decreases from top to bottom: $m \in \{1024, 512, 256\}$.

- $\boldsymbol{\theta}_{c_{\max}} = (0, 1/c_{\max}, \ldots, (c_{\max}-1)/c_{\max}, 1)$ as an approximation of $\breve{\boldsymbol{\theta}}^{(J,\ell)}_{c_{\max}}$ for large $c_{\max}$.
- or the estimation $\hat{\boldsymbol{\theta}}_{c_{\max}}$.

5.4 Iterative decoders

An iterative decoder is an accusation procedure composed of different components such as collusion attack estimators, and single or joint decoders. It also relies on thresholding, *i.e.* comparison of score to a threshold in order to accuse some users (or not), which is the focus of Chapter 6.

Figure 5.3 motivates the flow of this section by considering codes of decreasing length. The top figure depicts the ideal situation where the code length m is long enough w.r.t. the collusion size and strategy and the number n of users. The c colluders have the biggest scores, and the decoder can find a threshold distinguishing scores of colluders from those of the innocents. Indeed the scheme is operating at a rate $\log(n)/m \approx 0.014$ nats smaller than the achievable rate $I(Y; X|P, \boldsymbol{\theta}_c) \approx 0.025$ nats. In this first situation, a single decoder would do the job.

The second situation is worse. The code length is not long enough. The rate now equals 0.028 nats which is bigger than the achievable rate. Two colluders have the biggest scores, followed by some innocent users. With the required $\mathbb{P}_{\mathsf{FP}}$, only the first colluder gets caught.

The third situation is the worst. The code length is so short that an innocent user has the biggest scores and no colluder is identified.

The decoders presented so far aim at high discriminability separating the scores of innocents and colluders. Yet, whatever their discriminability power, when the code length becomes too short, some colluders or all of them may delude the accusation procedure. The next sections deal with the second situation while Sect. 5.4.4 tackles the third case.

5.4.1 Expectation-Maximization single decoder

This section shows a possible implementation of a iterative single decoder that tries to approximate the optimal single decoder LLR (5.4) by estimating the collusion strategy and the collusion by the Expectation-Maximization algorithm [108] running on $(\mathbf{X}, \mathbf{y}, \mathbf{p})$. This is different from the estimation $\hat{\boldsymbol{\theta}}_c^{\mathsf{EM}}$ described in Sect. 5.2.4 which only analyzes $(\mathbf{y}, \mathbf{p})$. It summarizes the work done in publications [27, 28, 30, 36] and to some extend in [146]. The key idea is to perform an iterative learn and match strategy:

1. Given an estimated collusion strategy, use its matched decoder to suspect some colluders,
2. Given some suspected colluders, their sequences and the pirated sequence, build a more accurate estimate of the collusion stratrgy.

In other words, as we better estimate the collusion strategy, we better accuse dishonest users (i.e. scores are more discriminative), and in turn, we better estimate what they have been doing as a strategy.

Conditioned on $(\mathbf{y}, \mathbf{p})$, the decoder gets a mixture of observations $\{\mathbf{x}_j\}$ which belong to two families 'innocent' or 'colluder'. Therefore, decoding boils down to estimating the hidden state, a.k.a. 'latent variable' in E.-M. literature, of each observation. It is represented by the sequence $\mathbf{z}$ where $z_j = 1$ if user j is guilty (*i.e.* hypothesis $\mathcal{H}_1$) and 0 otherwise for hypothesis $\mathcal{H}_0$. Our decoding problem is thus very similar to a mixture modelling which is a typical application of the E.-M. algorithm. It proceeds by iterating the two following steps.

E-step

The goal of the E-step is to have a better guess about the identities of the colluders. At iteration $k+1$, its inputs are the code $\mathbf{X}$ and the sequences $(\mathbf{y}, \mathbf{p})$. It also benefits from the estimate $c^{(k)}$ of the collusion size and an estimate $\boldsymbol{\theta}^{(k)}$ of the collusion channel. It simply computes the probability $\pi_j(\boldsymbol{\theta}^{(k)})$ that $z_j = 1$ (*i.e.* user j is guilty) according to this estimated collusion channel:

$$\begin{aligned} \pi_j(\boldsymbol{\theta}^{(k)}) &= \mathbb{P}(\mathcal{H}_1|\mathbf{x}_j, \boldsymbol{\theta}^{(k)}) \\ &= \frac{\mathbb{P}(\mathbf{X}_{\mathsf{col}} = \mathbf{x}_j|\boldsymbol{\theta}^{(k)})\mathbb{P}(\mathcal{H}_1|\boldsymbol{\theta}^{(k)})}{\mathbb{P}(\mathbf{X}_{\mathsf{col}} = \mathbf{x}_j|\boldsymbol{\theta}^{(k)})\mathbb{P}(\mathcal{H}_1|\boldsymbol{\theta}^{(k)}) + \mathbb{P}(\mathbf{X}_{\mathsf{inn}} = \mathbf{x}_j)\mathbb{P}(\mathcal{H}_0|\boldsymbol{\theta}^{(k)})}, \end{aligned} \tag{5.20}$$

where the application of the Bayes rule leads to the second line. Since there are $c^{(k)}$ colluders out of n users, we have $\mathbb{P}(\mathcal{H}_1|\boldsymbol{\theta}^{(k)}) = {}^{c^{(k)}}/_n$ and $\mathbb{P}(\mathcal{H}_0|\boldsymbol{\theta}^{(k)}) = 1 - {}^{c^{(k)}}/_n$. With Eq (5.2) & (5.3):

$$\pi_j(\boldsymbol{\theta}^{(k)}) = \frac{c^{(k)}}{c^{(k)} + (n - c^{(k)}) \exp\left(-s(\mathbf{x}_j, \mathbf{y}, \mathbf{p}|\boldsymbol{\theta}^{(k)})\right)}, \tag{5.21}$$

with $s(\cdot|\boldsymbol{\theta}^{(k)})$ the LLR score function (5.4) matching $\boldsymbol{\theta}^{(k)}$.

M-step

The goal of the M-step is to have a better guess about the collusion strategy. At iteration $k+1$, its inputs are the code $\mathbf{X}$, sequence $\mathbf{y}$, the previous estimation of the collusion channel $\boldsymbol{\theta}^{(k)}$ and the estimation of the identities of the colluders thanks to probabilities $\{\pi_j(\boldsymbol{\theta}^{(k)})\}_{j=1}^n$ evaluated in the E-step. According to the classical E.-M. formulation, the estimate of the collusion strategy

is refined by maximizing in $\boldsymbol{\theta}$ the log likelihood conditioned on the distribution of $\mathbf{Z}$ under the current estimate $\boldsymbol{\theta}^{(k)}$:

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) = \sum_{\mathbf{z}} \mathbb{P}(\mathbf{z}|\mathbf{X},\mathbf{y},\mathbf{p},\boldsymbol{\theta}^{(k)}) \log P(\mathbf{z},\mathbf{X}|\mathbf{y},\mathbf{p},\boldsymbol{\theta}). \tag{5.22}$$

The sequence $\mathbf{z}$ represents the hidden state of the full system. It is composed of n binary variables indicating which users are colluders. Eq. (5.22) requires that we consider the 2^n possible hidden state combinations, which is not tractable in practice for a large number of users. Some simplifications are needed:

$$\mathbb{P}(\mathbf{z},\mathbf{X}|\mathbf{y},\mathbf{p},\boldsymbol{\theta}) \approx \prod_{j=1}^{n} \mathbb{P}(z_j,\mathbf{x}_j|\mathbf{y},\mathbf{p},\boldsymbol{\theta}), \tag{5.23}$$

$$\mathbb{P}(\mathbf{z}|\mathbf{X},\mathbf{y},\mathbf{p},\boldsymbol{\theta}^{(k)}) \approx \prod_{j=1}^{n} \mathbb{P}(z_j|\mathbf{x}_j,\mathbf{y},\mathbf{p},\boldsymbol{\theta}^{(k)}). \tag{5.24}$$

These are approximations because the state of the colluders depend on each other, but this is a relaxation needed for having affordable complexity. This leads to the following approximation for a given c:

$$\begin{aligned}\tilde{Q}_c(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) &= n\log\frac{n-c}{n} + \left(\sum_{j=1}^{n}\pi_j(\boldsymbol{\theta}^{(k)})\right)\log\frac{c}{n} + \sum_{j=1}^{n}\log\mathbb{P}(\mathbf{X}_{\mathsf{inn}}=\mathbf{x}_j|\mathbf{p}) \\ &+ \sum_{j=1}^{n}\pi_j(\boldsymbol{\theta}^{(k)})\log\frac{\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x}_j,\mathbf{p},\boldsymbol{\theta})}{\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{p},\boldsymbol{\theta})}.\end{aligned} \tag{5.25}$$

Maximizing $\tilde{Q}_c(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)})$ amounts to maximize the very last term of (5.25). Unfortunately there is no closed form expression for the solution of the M-step, so it must be sought by numerical means. A typical optimization runs as follows:

1. For a parameter c starting from 2 to $c_{\max}$, the function $\tilde{Q}_c(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ is maximized over $\boldsymbol{\Theta}_c$.
2. Then, the $(c_{\max}-1)$ partial maxima $\tilde{Q}_c(\boldsymbol{\theta}_c^{\star}|\boldsymbol{\theta}^{(k)})$ are compared in order to isolate the global maximum, and the parameter $\boldsymbol{\theta}^{(k+1)}$ is updated accordingly:

$$c^{(k+1)} = \arg\max_{c\in\{2,\dots,c_{\max}\}} \tilde{Q}_c(\boldsymbol{\theta}_c^{\star}|\boldsymbol{\theta}^{(k)}), \tag{5.26}$$

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{\star}_{c^{(k+1)}}. \tag{5.27}$$

Initialization and termination of the E.-M. algorithm

E.-M. converges to a local maximum of the likelihood function, and the initialization step is crucial. At the beginning, the accusation process has no idea about the values of the hidden states, the number of colluders and the collusion strategy. We set $c^{(0)} = c_{\max}$ which is a pessimistic scenario since the decoder is prepared for the worst case. Then $\boldsymbol{\theta}^{(0)}$ is the maximizer of the likelihood of $\mathbf{y}$ knowing $\mathbf{p}$, ignoring the code $\mathbf{X}$ as given in (5.12).

The E and M steps are iterated until some termination criterion, usually when $\tilde{Q}(\boldsymbol{\theta}^{(k+1)},\boldsymbol{\theta}^{(k)})$ is no longer improving or when a maximum number of iteration $k_{\max}$ has been reached. Let k_f be the last iteration, the final decision of the decoder is made by computing the LLR (5.4) with $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k_f)}$.

A probability $\pi_j(\boldsymbol{\theta}^{(k_f)}) > 1/2$ would indicate that user is more likely a colluder than an innocent. This is wrong because $\boldsymbol{\theta}^{(k_f)}$ might not be equal to the true collusion strategy. However, (5.21) is just a monotonic mapping of (5.4) from $[0, \infty]$ to the interval $[0, 1]$. Hence, thresholding of (5.4) is equivalent of thresholding of (5.21). The setting of this threshold is postponed to Chap. 6.

Experimental results

The experimental setup is the following: $n = 1,000$ users , $m = 300$ bits, and $c = 5$ colluders. The iterative decoder is assuming that the maximum number of colluders is $c_{\max} = 10$. The maximum number of iterations for the E.-M. decoder is set to $k_{\max} = 5$. Two collusion attacks are considered: Minority vote and the Worst Case Attack against a single decoder.

We first show that this algorithm yields discriminative scores. The D.E.T. plot is displayed in Fig. 5.4:right. For a threshold τ ranging from 0 to 1, the probability of false positive per user $\mathbb{P}(\pi_{\mathsf{inn}}(\boldsymbol{\theta}^{(k_f)}) > \tau)$ and the probability of false negative per colluder $\mathbb{P}(\pi_{\mathsf{col}}(\boldsymbol{\theta}^{(k_f)}) \leq \tau)$ are estimated with a Monte Carlo simulation with $N = 1000$ independent experiments. Each experiment generates a secret sequence $\mathbf{p}$ and a code $\mathbf{X}$ of n sequences, c of these collude, and the E.-M. decoding proceeds with the received pirated sequence. For both attacks, the scores are as discriminative as the scores of the optimal LLR. However, if the colluders are smart enough to lead the worst attack, then the E.-M. algorithm does not gain much compared to the Tardos-Škorić scores.

Fig. 5.4 (left) shows the percentage of correct guesses about the collusion size. The proposed iterative decoder does not correctly estimate the collusion strategy but this does not imply a bad decoding (see Sect. 5.2.4). Yet, accusing the $c^{(k_f)}$ biggest final scores would be a disaster as $c^{(k_f)}$ is absolutely unreliable and most of time bigger than the real c.

To conclude, the main drawback of this E.-M. decoder resides in its complexity. The M-step computes $O(mn)$ operations, the evaluation of function $\tilde{Q}$ needs $O(mc)$ operations, and the E-step must perform a maximization of $c_{\max} - 1$ of these functions. This is the reason why our experiments do not tackle long codes and many users. Each run lasts around 10 minutes on a regular CPU. Indeed, $m = 300$ is quite short to fight against a collusion of size $c = 5$, but we did it in purpose to have 'big' probabilities of errors, measurable with a Monte Carlo simulation. This experimental setup would not be relevant in practice where a huge number of users is involved. A trick would be to first build a subset of suspects, *i.e.* a thousand of users with the highest initial scores $s_j(\boldsymbol{\theta}^{(0)})$.

5.4.2 Side-informed single decoder

The iterative single decoder presented in [40] is a simplified version of the E-M decoding algorithm explained above. At iteration k+1, the soft output $\pi_j(\boldsymbol{\theta}^{(k)})$ is replaced by a binary decision whether user j is a colluder. This assumes that the accusation procedure has a thresholding mechanism to identify the colluders with the biggest scores under a controlled probability of false positive (see Chapter 6). Identifying colluders not only may make the score function of the next iteration more discriminative, but also helps the estimation of the collusion strategy: the codewords of the identified colluders play the role of a side information conditioning the likelihood in (5.12).

Its typical use is the second situation of Fig. 5.3 where some colluders have been identified after the first iteration. The fact that these colluders may not be able to create the pirated sequence under the marking assumption (*e.g.* there are indices in the code where they all have symbols '1' while symbol '0' is in the pirated sequence) proves the existence of remaining accomplices to be chased. The 'catch one' target is reached, but not the 'catch all' or 'catch many' (see Sect. 2.1.3).

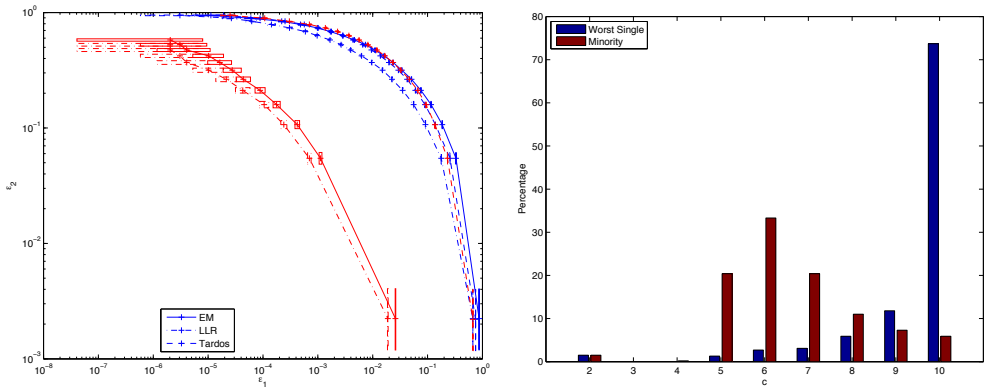


Figure 5.4: E.-M. decoding for two collusion strategies: minority vote and the worst case attack against a single decoder, with $c = 5$, $n = 1000$ and $m = 300$. (left) D.E.T.: probability of false negative per colluder vs. probability of false positive per user (boxes represent confidence interval at 95%.) for three scorings (E.M., optimal LLR and Tardos / Škorić) ; (right) histogram of $c^{(k_f)}$.

An iterative side-informed single decoder can use the codewords of the already accused users as a side-information in order to chase the remaining colluders. This side-information generates a new score function at the next iteration which may be more discriminative. Again, the decoder uses its thresholding mechanism to proceed to new accusations. The codewords of these newly accused colluders enrich the side-information. This iterative decoder runs until no new accusation is made.

Encompassing side information is easily done for LLR-based decoders. Suppose that κ users $\mathcal{A} = \{j_1, \ldots, j_\kappa\}$ were already deemed guilty and that the LLR scoring function is tuned for a collusion strategy $\boldsymbol{\theta}_{c_{\max}}$ of size $c_{\max} \geq \kappa$. Denote by $\boldsymbol{\rho} \in \{0, \ldots, \kappa\}^m$ the sum of their codewords: $\rho_i = \sum_{k=1}^{\kappa} x_{j_k,i}, \forall i \in \{1, \ldots, m\}$. The side-informed LLR is as follows:

$$s_{j|\mathcal{A}} = s(\mathbf{x}_j, \mathbf{y}, \mathbf{p}|\boldsymbol{\rho}, \kappa) = \sum_{i=1}^{m} U(x_{j,i}, y_i, p_i|\rho_i, \kappa, \boldsymbol{\theta}_{c_{\max}}) \tag{5.28}$$

$$U(x, y, p|\rho, \kappa, \boldsymbol{\theta}_{c_{\max}}) = \log \frac{\mathbb{P}\left(Y = y|\rho + x, \kappa + 1, p, \boldsymbol{\theta}_{c_{\max}}\right)}{\mathbb{P}\left(Y = y|\rho, \kappa, p, \boldsymbol{\theta}_{c_{\max}}\right)}. \tag{5.29}$$

with $\mathbb{P}\left(Y = y|\rho, \kappa, p, \boldsymbol{\theta}\right)$ defined in (4.13). Figure 5.5 shows how the scores of the colluders and the innocents get more separated as the size of the side-information increases. This approach is easily extended to generalized linear decoders of Sect. 5.2.3.

Taking into account side information also makes the estimation of the collusion strategy more accurate in case we use an adaptive decoder. The Maximum Likelihood estimator is modified into:

$$\hat{\boldsymbol{\theta}}_{c_{\max}}^{(\mathsf{MLE})} = \arg \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_{c_{\max}}} \sum_{i=1}^{m} \log \mathbb{P}\left(Y = y_i|\rho_i, \kappa, p_i, \boldsymbol{\theta}\right). \tag{5.30}$$

Figure 5.6 is similar to Fig. 5.5 except that the Maximum Likelihood estimator is recomputed anytime a new colluder is accused. The scores of the colluders and innocents get even more separated along with the iterations. This approach is easily extended to generalized linear decoders of Sect. 5.2.3.

The theoretical study of Sect. 13.1.3 shows that the achievable rates keeps on increasing while identifying more colluders. Therefore, the hardest step is the identification of a first colluder,

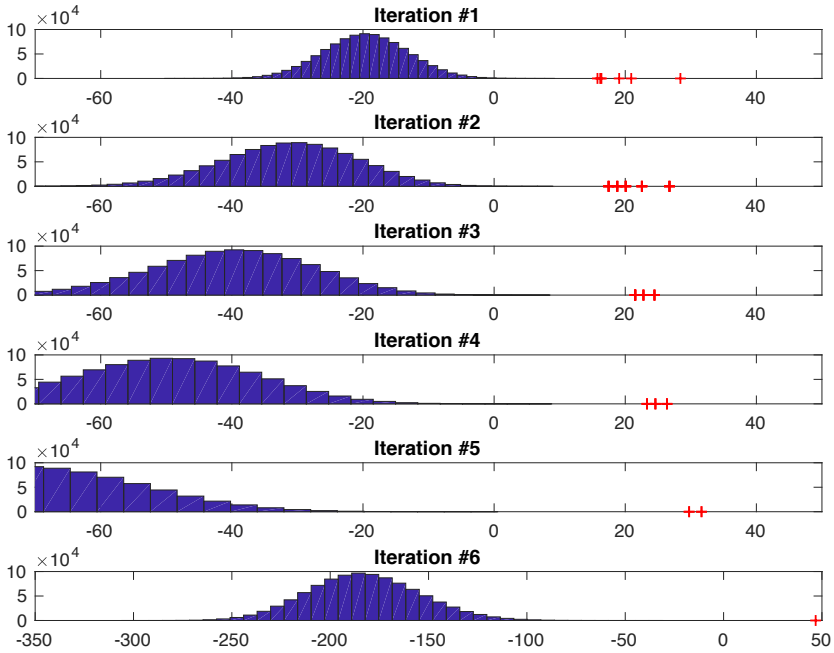


Figure 5.5: An iterative single decoder with $c = 6$, $m = 512$, $n = 10^6$ with antipodal modulation. The colluders' scores are shown with '+', while the histogram represents the score of innocent users. The estimation of the collusion strategy for $c_{\max} = 10$ is done once for all at the beginning of the decoding (*i.e.* without side information). These plots hold for a given $(\mathbf{y}, \mathbf{p})$.

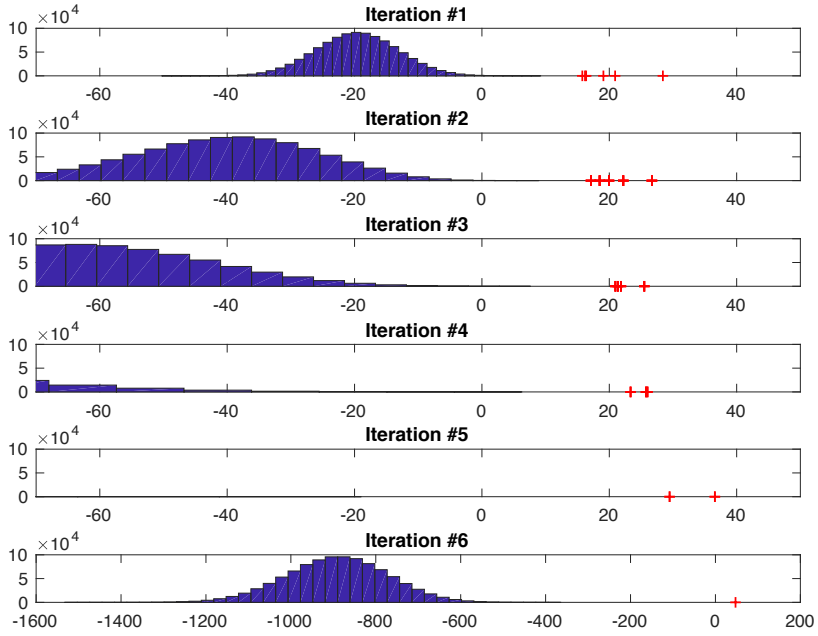


Figure 5.6: An iterative single decoder with $c = 6$, $m = 512$, $n = 10^6$ with antipodal modulation. The colluders' scores are shown with '+', while the histogram represents the score of innocent users. The estimation of the collusion strategy for $c_{\max} = 10$ is refined any time a new colluder is accused. These plots hold for a given $(\mathbf{y}, \mathbf{p})$.

which initialises this iterative decoder. This implements in practice the idea that the capacity of a single decoder is the same under the target 'catch all' or 'catch one' (4.38) provided that the colluders share the risk.

5.4.3 Markov Chain Monte Carlo joint decoder

As far as I know, the Markov Chain Monte Carlo decoder [35] is the only 'tractable' implementation of a joint decoding. It is a probabilistic decoder because it is based on a Monte Carlo simulation.

Denote by Z_j a random variable coding whether user j is a colluder. A priori, $\mathbb{P}(Z_j = 1) = c/n$. Once sequences $(\mathbf{y}, \mathbf{p})$ are known, the goal is to compute the a posteriori probability $\mathbb{P}(Z_j = 1|\mathbf{y}, \mathbf{p})$. This is done by leveraging the power of a joint decoder and to consider $\mathbb{P}(Z_j = 1|\mathbf{y}, \mathbf{p})$ as the marginal probability:

$$\mathbb{P}(Z_j = 1|\mathbf{y}, \mathbf{p}) = \sum_{g: j \in g} \mathbb{P}(g = \mathcal{C}|\mathbf{y}, \mathbf{p}), \tag{5.31}$$

where $\mathbb{P}(g = \mathcal{C}|\mathbf{y}, \mathbf{p})$ is the probability that g, a group of users, is the actual collusion. This sum is not tractable as there are too many groups. This sum is replaced by a Monte Carlo simulation. K groups $\{g_k\}_{k=1}^K$ are randomly drawn and the marginal is simply estimated as the empirical frequency (see Fig. 5.7):

$$\hat{\mathbb{P}}(Z_j = 1|\mathbf{y}, \mathbf{p}) = K^{-1}|\{g_k | j \in g_k\}|. \tag{5.32}$$

This works provided that the groups are drawn according to the distribution $\mathbb{P}(g|\mathbf{y}, \mathbf{p})$:

$$\mathbb{P}(g|\mathbf{y}, \mathbf{p}) \propto \mathbb{P}(\mathbf{y}|g, \mathbf{p})\mathbb{P}(g). \tag{5.33}$$

The a priori probability of a group is modelled as follows. The size of the group $|g|$ is uniformly distributed over $\{2, \dots, c_{\max}\}$. Given the size, any group is equally likely:

$$\mathbb{P}(g) = \mathbb{P}(g||g|)\mathbb{P}(|g|) = \binom{n}{|g|}^{-1} c_{\max}^{-1}. \tag{5.34}$$

The probability $\mathbb{P}(\mathbf{y}|g, \mathbf{p})$ is given by Eq. (4.13) for collusion strategy $\hat{\boldsymbol{\theta}}_{c_{\max}}$ estimated from $(\mathbf{y}, \mathbf{p})$.

The last difficulty is to generate groups distributed as $\mathbb{P}(g|\mathbf{y}, \mathbf{p})$. The Markov Chain plays this role. It is implemented as a Gibbs sampler. It starts with a random group $g^{(0)}$, and makes a random walk from a group to a neighbouring group according to a transition probability distribution:

$$\mathbb{P}(g^{(t+1)} = g|g^{(t)}) = \frac{\mathbb{P}(\mathbf{y}|g, \mathbf{p})\mathbb{P}(g)}{\sum_{g' \in \mathcal{N}(g^{(t)})} \mathbb{P}(\mathbf{y}|g', \mathbf{p})\mathbb{P}(g')}, \tag{5.35}$$

where $\mathcal{N}(g)$ denotes the set of neighbouring groups of g (*e.g.* groups which are different from g by a single user). By doing so, the stationary distribution of the Markov Chain is $\mathbb{P}(g|\mathbf{y}, \mathbf{p})$, which means that after a burning period T, the states $\{g^{(t)}\}_{t>T}$ are distributed as $\mathbb{P}(g|\mathbf{y}, \mathbf{p})$. The Monte Carlo simulation starts after the burning period and samples K groups from which the marginal are estimated.

Paper [35] uses a Gibbs sampler with random scan replacing a single user in $g^{(t)}$. The size of $\mathcal{N}(g^{(t)})$ is then n, and the complexity is in $O(n(T + K))$. This work also conjectures a burning period proportional to n, which makes the complexity quadratic in n. This is more costly than a single decoder, but at the same, more tractable than the complexity of a joint decoder in $O(n^c)$.

There are two drawbacks: The performances depend on the estimation of the collusion strategy and collapse if $c_{\max}$ is much bigger than c. Moreover, it is difficult (if not impossible) to set a threshold guaranteeing a given probability of false alarm.

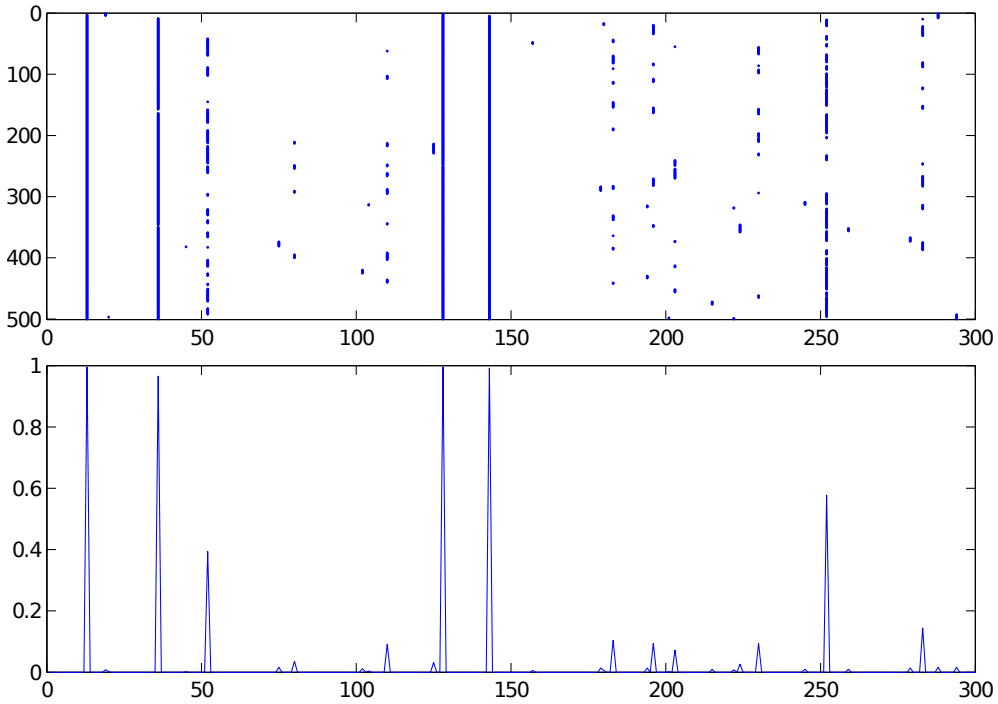


Figure 5.7: Illustration of the MCMC method for $K = 500$ and $n = 300$: [Up] The Markov chain: the binary matrix $K \times n$ indicating which users belong to the group $g^{(t)}$ for $1 \leq t \leq K$, [Down] The Monte Carlo estimation: the empirical marginal probabilities, *i.e.* the mean of the columns of the above binary matrix. Four users are clearly identified as colluders.

5.4.4 Iterative joint decoder

The last iterative decoder resorts to joint decoding to tackle situations illustrated in Fig. 5.3 (bottom): No score is above the threshold so that no new user is accused. This may happen at any round of side-informed single decoder, and this stops the accusation procedure.

Fig. 4.4 shows that if conditioning on a new colluder's codewords is not possible, another way to increase the achievable rate is to proceed to a joint decoding. This task is made tractable by selecting a small subset of suspected users. It takes advantage of the previous iterations to infer which users are more suspicious than others.

The overall architecture is sketched in Fig. 5.8 and detailed in Algorithm 1. It has been published in [41, 42] under the name 'Don Quixote'. Some subroutines are discussed below.

From group to user

Once the scores for groups of size t have been computed in line 13, an easy way to fall back on user scores is to record the maximum scores per user:

$$s_j = \max_{g : j \in g} s_g, \tag{5.36}$$

where j is the index of a user and $g = \{j_i, \ldots, j_t\}$ is a group of t user indices.

Algorithm 1 Iterative Joint Tardos Decoder.

Require: $\mathbf{y}$, $\mathbf{X}$, $\mathbf{p}$, $c_{\max}$, $t_{\max} \leq c_{\max}$, $\mathbb{P}_{\mathsf{fp}}$, n

$\mathcal{U} \leftarrow \{1, \ldots, n\}$, $\mathcal{U}_{SI} \leftarrow \emptyset$
repeat
  $t \leftarrow 1$
  $\hat{\boldsymbol{\theta}}_{c_{\max}} \leftarrow \texttt{estimate}(\mathbf{y}, \mathbf{p}, \mathcal{U}_{SI}, c_{\max})$
  $s(\cdot) \leftarrow \texttt{score\_function}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\max}}, \mathcal{U}_{SI}, \mathrm{t})$
  $\mathbf{s} \leftarrow \texttt{scores}(\mathcal{U} \setminus \mathcal{U}_{SI}, \mathbf{X}, s(\cdot))$
  $\tau^{+} \leftarrow \texttt{threshold}(s(\cdot), \mathbb{P}_{\mathsf{fp}}, n, t)$
  $\mathcal{A} \leftarrow \{j \in \mathcal{U} \setminus \mathcal{U}_{SI} | s_j > \tau^{+}\}$
  while $\mathcal{A} = \emptyset$ AND $t < \min(t_{\max}, c_{\max} - |\mathcal{U}_{SI}|)$ **do**
    $t \leftarrow t + 1$
    $\mathcal{U}^{(t)} \leftarrow \texttt{top}(\mathbf{s}, \mathcal{U} \setminus \mathcal{U}_{SI}, n^{(t)})$
    $s(\cdot) \leftarrow \texttt{score\_function}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\max}}, \mathcal{U}_{SI}, \mathrm{t})$
    $\mathbf{s} \leftarrow \texttt{scores}(\binom{\mathcal{U}^{(t)}}{t}, \mathbf{X}, s(\cdot))$
    $\tau^{+} \leftarrow \texttt{threshold}(s(\cdot), \mathbb{P}_{\mathsf{fp}}, n^{(t)}, t)$
    $g^{\diamond} \leftarrow \underset{g \subset \mathcal{U}^{(t)}, |g| = t}{\arg\max} \; s_g$
    if $s_{g^{\diamond}} > \tau^{+}$ **then**
      for all $j \in g^{\diamond}$ AND **while** $\mathcal{A} = \emptyset$ **do**
        $s(\cdot) \leftarrow \texttt{score\_function}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\max}}, \mathcal{U}_{SI} \cup (g^{\diamond} \setminus \{j\}), 1)$
        $\tau^{+\prime} \leftarrow \texttt{threshold}(s(\cdot), \mathbb{P}_{\mathsf{fp}}, n, 1)$
        if $\texttt{score}(\{j\}, \mathbf{X}, s(\cdot)) > \tau^{+\prime}$ **then** $\mathcal{A} \leftarrow \{j\}$
      end for
    end if
  end while
  $\mathcal{U}_{SI} \leftarrow \mathcal{U}_{SI} \cup \mathcal{A}$
until $\mathcal{A} = \emptyset$ OR $|\mathcal{U}_{SI}| \geq c_{\max}$
return $\mathcal{U}_{SI}$

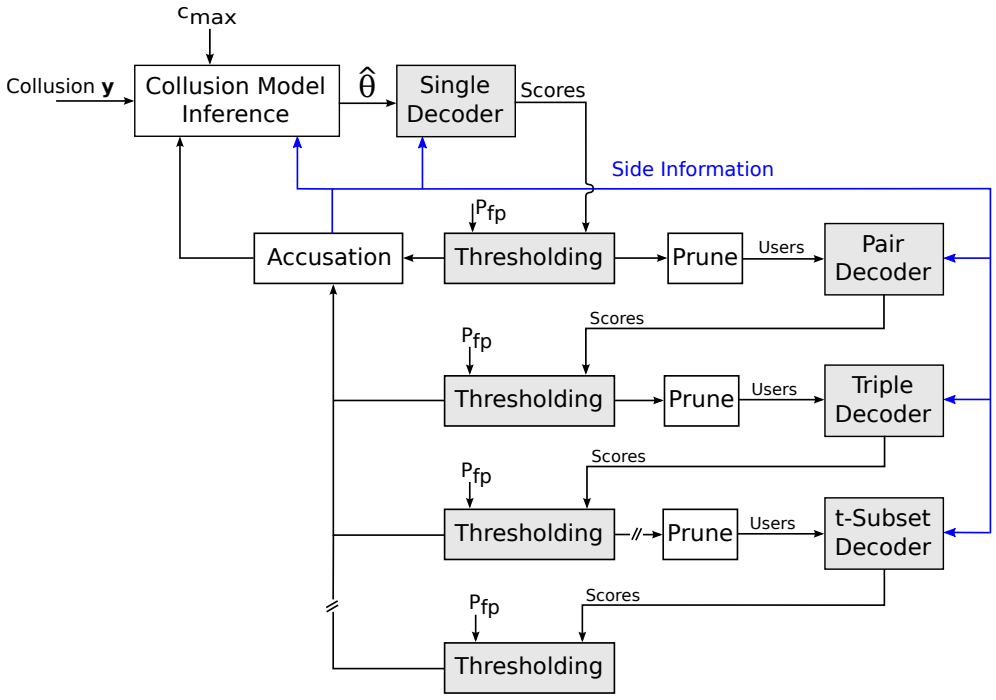


Figure 5.8: Overview of the iterative joint decoder.

Pruning the list of users

Computing group scores is not tractable when n is large. For this reason, the users least likely to be guilty are gradually filtered out.

Suppose that, in the `WHILE` loop (line 9 and below), iteration $t-1$ (working with groups of size $t-1$) finds no new accused user. No new codeword is included as side-information, and the next iteration t will consider groups of size t. It starts by limiting the number of users to $n^{(t)}$ in line 11. If $t-1=1$, the last computed scores are already user scores. Otherwise, (5.36) translates group scores into user scores. These are ranked and the first $n^{(t)}$ users of the list are selected as potential guilty users. The number $n^{(t)}$ depends on t with the motivation that the complexity of any iteration should be roughly the same. An iteration working with single decoding has a complexity in $O(nm)$; an iteration working with joint decoding has a complexity $O((n^{(t)})^t m)$. Therefore, we set $n^{(t)} \approx n^{1/t}$.

Accusation of a new user

The score of a group compares the likelihood that all the users in the group are colluders to the likelihood that all of them are innocents (see Sect. 5.3). Chap. 6 explains how to find a suitable threshold as the $(1-\mathbb{P}_{\mathsf{fp}})$-quantile of the distribution of the score of a group composed of innocent users only. Therefore, a score bigger than this threshold almost surely indicates that this group contains at least one colluder.

The way to identify the most likely guilty user within a group g has been determined in [136, Sect. 5.3]. It is implemented in line 16 and below. It computes a single decoding score for each user of the group with all the other users considered as guilty. This means that their codewords

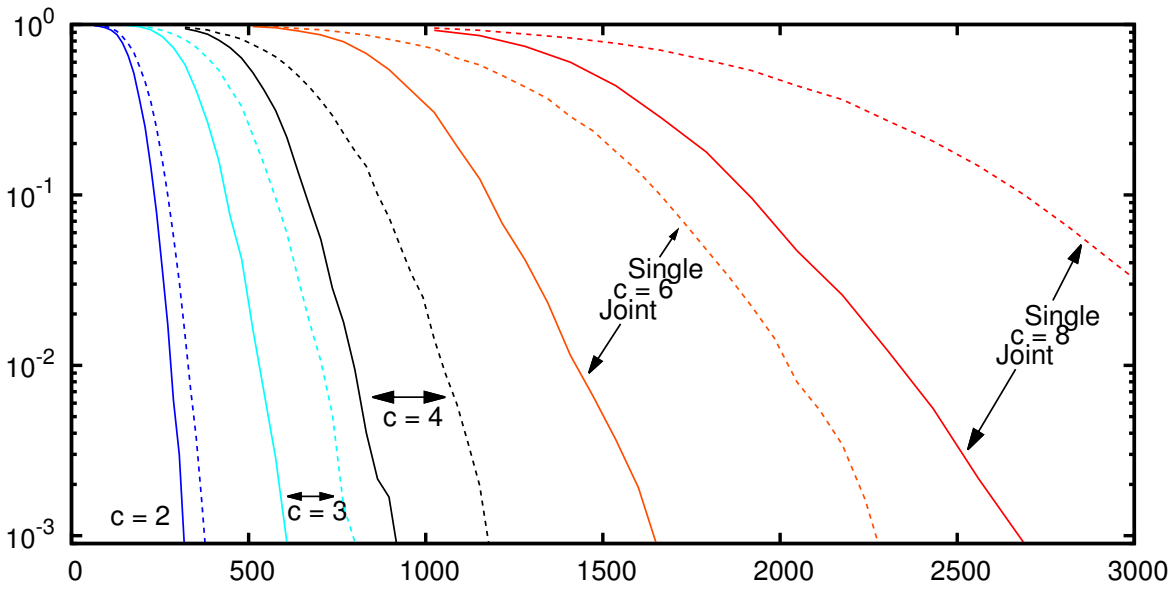


Figure 5.9: Code length vs. $\mathbb{P}_e = \mathbb{P}_{\mathsf{FP}} + \mathbb{P}_{\mathsf{FN}}$ for $n = 10^6$ users and c colluders performing *worst-case* attack against a single decoder; $c_{\max} = 8$. This experiment generated 10^{10} codewords.

are integrated in the side-information. As defined in (5.28):

$$s_j = s_{j|g\setminus\{j\}}, \tag{5.37}$$

This refinement is obviously costly in complexity. It will be applied only to the group having the biggest score and only if it is above the threshold.

In the end, user j is accused if $s_{j|g\setminus\{j\}}$ is above a suitable threshold (line 19). In that way, any accused user is deemed guilty based on a single decoding score with some side-information (line 8 or 20). This stops the `WHILE` loop and the algorithm carries on setting t back to 1, *i.e.* a single decoding with this new piece of side-information.

Experimental results

A first experimental setup considers a 'Catch One' objective with $n = 10^6$ users and $c \in \{2, 3, 4, 6, 8\}$ colluders performing the worst-case attack against a single decoder (4.26). Fig. 5.9 plots the empirical probability of error $\mathbb{P}_e = \mathbb{P}_{\mathsf{FP}} + \mathbb{P}_{\mathsf{FN}}$ obtained by running 10^4 experiments for each setting versus the code length m. The false-positive error is controlled by thresholding at a global false positive probability $\mathbb{P}_{\mathsf{FP}} = 10^{-3}$ (see Chap. 6), which is confirmed experimentally.

For a given probability of error, the iterative joint decoder succeeds in reducing the required code length over the side-informed single decoder of Sect. 5.4.2, especially for larger collusions. When the code length is long enough, the iterative joint decoder indeed never resorts to a joint decoding: single scoring functions are enough discriminative to identify at least one colluder. When the code gets shorter, single decoding begins to fail finding a first colluder. Note that the transition is all the more so abrupt as the collusion size is small. This is when joint decoding with its larger achievable rate saves the accusation process.

A second experimental setup considers a 'Catch-many' objective. Figure 5.10 shows the average number of identified colluders by different decoding approaches. The experimental setup considers $n = 10^6$ users, code length $m = 2048$, and several collusion attacks carried out by two to

eight colluders. The global probability of a false positive error is fixed to $\mathbb{P}_{\mathsf{FP}} = 10^{-3}$. As expected, the MAP single decoder knowing $\boldsymbol{\theta}_c$ provides the best decoding performance amongst the single decoders. The symmetric Tardos decoder performs poorly but evenly against all attacks; the generalized linear decoder tuned on collusion strategy estimations (see Sect. sec:AdaptiveDecoders) improves the results only slightly.

The iterative joint decoders consistently achieve to identify most colluders – with a dramatic margin in case the traitors choose the worst-case attack against a single decoder. This attack bothers the very first step of our decoder which is a single decoding. Yet, as soon as some side information is available or a joint decoding is used, this is no longer the worst case attack. Finding the worst case attack against the iterative joint decoder is indeed difficult. For large c, a good guess is the interleaving attack which is asymptotically the worst case against joint and single decoding (see Sect. 4.4).

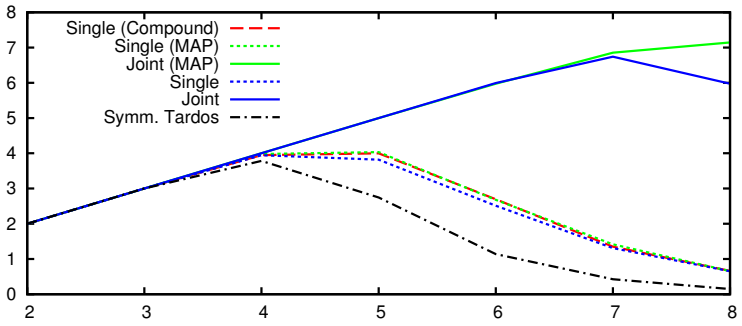
The single decoder based on estimation $\hat{\boldsymbol{\theta}}_{c_{\max}}$ and the true MAP are different when c is lower than $c_{\max}$. However, this is not a great concern in practice for a fixed m: for small c, the code is long enough to face the collusion even if the score is less discriminative than the ideal MAP; for big c the score of our decoder gets closer to the ideal MAP. We also observe that the estimation of the collusion strategy is inaccurate as the performance gap between the iterative joint decoders using $\boldsymbol{\theta}_c$ and $\hat{\boldsymbol{\theta}}_{c_{\max}}$ reveals for larger c.

Many more experiments are commented in articles [41] and [42], including runtimes and statistics demonstrating the power of joint decoding.

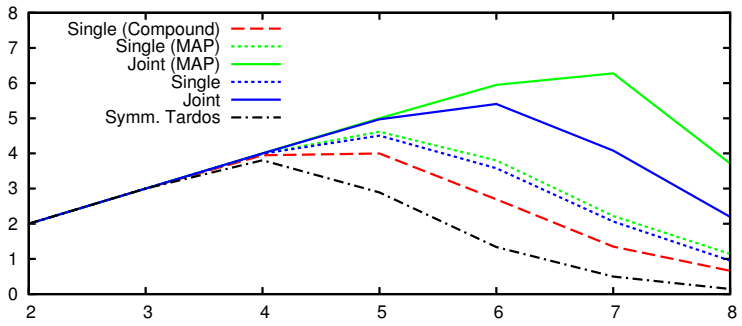
5.5 Conclusion

The chapter offers a panel of accusation procedures going from fixed linear single decoders, generalised linear decoders to iterative schemes employing joint decoding. This plurality of decoders produces a trade-off between complexity and discriminability. This goes from the E-M algorithm of Sect. 5.4.1 to the simple fixed linear single decoders of Sect. 5.2.2. An extensive investigation of this trade-off is still missing in the literature.

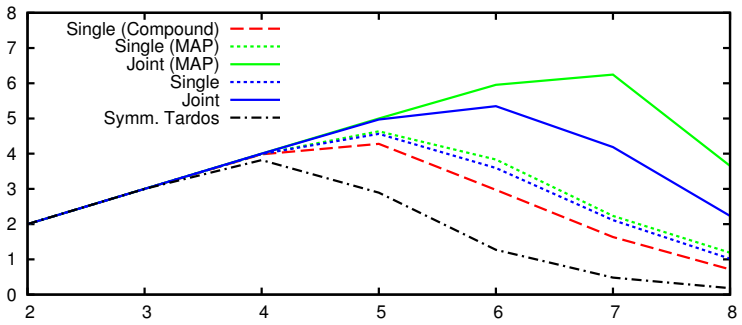
Nevertheless, the main feature of an accusation procedure is its soundness. The MCMC algorithm of Sect. 5.4.3 offer no guarantee that some innocent user will not be accused. Moreover, finding empirically the conditions when this algorithm reliably identifies the colluders is difficult as it is not easily simulatable. A simulation must consider the whole codebook $\mathbf{X}$ at each run. All the other procedures presented in this chapter accuse a user by comparing a score with respect to a threshold. The next chapter shows how to estimate the probability of framing an innocent as a function of this threshold. This probability depends of the discriminability of the score function: If the estimation of the collusion strategy gets completely spoiled, the score function is not discriminative, and the procedure of the next chapter ends up with a rather high threshold preventing the accusation of any innocent user.



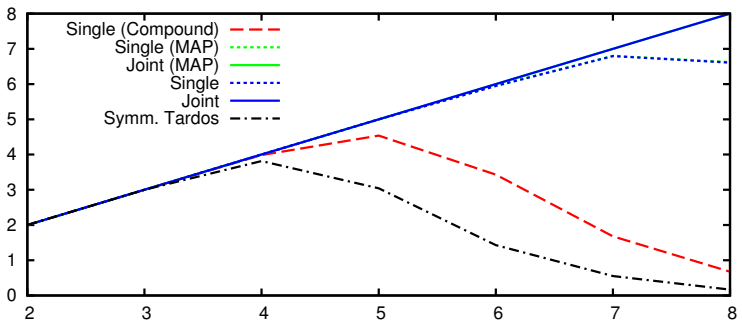
(a) Worst-Case Attack against Single Decoding



(b) Worst-Case Attack against Joint Decoding



(c) Interleaving Attack



(d) Majority Attack

Figure 5.10: Decoder comparison with the 'Catch many' objective: $n = 10^6$, $m = 2048$, $\mathbb{P}_{\mathsf{FP}} = 10^{-3}$, $c_{\max} = 8$.

Chapter 6

Thresholding and rare event simulation

The key idea of the previous chapter is to decouple two issues about the score functions: discriminability and thresholding. Discriminability has been improved compared to the original Tardos-Škorić score function (2.15) by betting that there will be a means to find a threshold enforcing a targeted probability of false positive. This chapter presents this mechanism for the score function $s(\cdot)$ of a single decoder. It can be extended to joint and side-informed decoders

In brief, the score $s(\mathbf{x}, \mathbf{y}, \mathbf{p})$ is a function of the pirated sequence $\mathbf{y}$, the codeword $\mathbf{x}$ of the user and the secret bias vector $\mathbf{p}$. The false positive (per user) probability $\mathbb{P}_{\mathsf{fp}}$ is the probability that $s(\mathbf{x}, \mathbf{y}, \mathbf{p})$ is bigger than the threshold whereas the user is not a colluder. The problem is difficult because the decoder is oblivious to the collusion size and strategy.

6.1 From theory to practice

6.1.1 From universal thresholds to adaptive thresholds

As stated above, the problem is not well posed. Evaluating a probability implies that 'something' is deemed as random. In theoretical papers about Tardos codes, that 'something' is indeed 'everything'. Tardos codes are probabilistic, the distributions of the random vectors follow a Markov chain, $\mathbf{P} - \mathbf{X}_{\mathcal{C}} - \mathbf{Y}$, where $\mathbf{X}_{\mathcal{C}}$ the codewords of the colluders. Their goal consists in finding a threshold τ s.t.

$$\mathbb{P}(s(\mathbf{X}_{\mathsf{inn}}, \mathbf{Y}, \mathbf{P}) > \tau) < \mathbb{P}_{\mathsf{fp}}, \tag{6.1}$$

where $\mathbf{X}_{\mathsf{inn}}$ is a random binary vector modelling the codeword of an innocent user. Its distribution knowing $\mathbf{P}$ is $X_{\mathsf{inn},i} \overset{\text{i.i.d.}}{\sim} \mathcal{B}(p_i)$. The difficulty lies in the statistical model of $\mathbf{Y}$ because we don't know the collusion process, which parameterizes the transition $\mathbf{X}_{\mathcal{C}} - \mathbf{Y}$. Even if we knew it, the score function might be so complicated that the distribution of the scores of an innocent user is impossible to derive. One resorts to upper bounds whose tightness is most of time not mentioned, or approximations (like the sempiternal Gaussian distribution thanks to the Central Limit Theorem) whose tails are not accurate enough to estimate weak probabilities. Finer approximations are difficult to establish [152].

The practitioner has a different problem than a theorist: A Tardos code of length m has been deployed, its secret key is $\mathbf{p}$, and a pirated sequence $\mathbf{y}$ has been extracted from a forged copy. These sequences are observations, not random objects. The practitioner is not looking for

an universal threshold τ, but for an adaptative threshold $\tau(\mathbf{y}, \mathbf{p})$ working for these observations only.

$$\mathbb{P}(s(\mathbf{X}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}) > \tau(\mathbf{y}, \mathbf{p})) < \mathbb{P}_{\mathsf{fp}}. \tag{6.2}$$

This problem is simpler because the collusion strategy is no longer a nuisance parameter: $\mathbf{Y}$ has replaced by the observation $\mathbf{y}$. This opens the door to tighter upper bounds or to numerical estimations [32].

6.1.2 A shift in paradigm

A theoretical paper about Tardos codes takes the point of view of the code designer willing to deploy a traitor tracing solution. The required levels (η_S, η_C) on the global error probabilities $(\mathbb{P}_{\mathsf{FP}}, \mathbb{P}_{\mathsf{FN}})$ induces levels (ϵ_S, ϵ_C) on the error probabilities $(\mathbb{P}_{\mathsf{fp}}, \mathbb{P}_{\mathsf{fn}})$ per user (depending on the accusation objective). Assuming at most c colluders among n users, a typical theoretical paper exhibits a density f, a score function $s(\cdot)$, and a threshold τ s.t.:

$$\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(S_{\mathsf{inn}} > \tau) \quad < \quad \epsilon_S, \quad \text{[Proof of soundness]} \tag{6.3}$$
$$\mathbb{P}_{\mathsf{fn}} = \mathbb{P}(S_{\mathsf{col}} < z) \quad < \quad \epsilon_C, \quad \text{[Proof of completeness]} \tag{6.4}$$

whatever the collusion attack and provided that the code length m is bigger than $\underline{m}(n, c, \eta_S, \eta_C)$.

In words, a theoretical paper is a white box taking as inputs the requirements (n, c, η_S, η_C) and giving as output the necessary code length. This means that its authors are advising the code designer not to deploy this traitor tracing solution if the condition on the code length cannot be met.

In this chapter, the *operational mode* denotes the point of view of the practitioner at the decoding side. A system *has already been deployed*: A content has been distributed to n users. The watermarking technique in use has been able to embed m bits in the content. Therefore, m is *not* an output, *i.e.* a parameter depending on other parameters, but an input in the operational mode.

The decoder might not know the distribution f or how suitable this distribution is w.r.t. to collusion attacks. It only knows $\mathbf{p}$ and the codebook $\mathbf{X}$. The decoder has a priori no clue on the collusion size c and the collusion strategy. It only observes its result $\mathbf{y}$. However, it must not accuse a given innocent user or with a probability at most ϵ_S. This means that it should automatically 'notice' and give up when the conditions are not suitable for reliably accusing colluders in the observed context. To summarize, the inputs of the problem in the operational mode are $(n, m, \mathbf{p}, \mathbf{y}, \epsilon_S)$.

In the operational mode, there is thus no randomness. However, since we don't know which codewords have contributed to the forgery $\mathbf{y}$, we consider that the codeword of an innocent is a random variable $\mathbf{X}_{\mathsf{inn}}$ distributed according to the secret bias vector $\mathbf{p}$: $X_{\mathsf{inn},i} \overset{\text{i.i.d.}}{\sim} \mathcal{B}(p_i)$, $\forall i \in \{1, \cdots, m\}$. The score of this innocent is random and equals $S_{\mathsf{inn}} = s(\mathbf{X}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p})$. The index inn tells that $\mathbf{X}_{\mathsf{inn}}$ and S_{inn} are not related to a given user in particular, but to an innocent in general.

This shift of paradigm also allows to map the score s_j of user j onto a probability of being innocent knowing the observations $\mathbf{y}$ and $\mathbf{p}$: $s_j \mapsto \Pi(s_j) = \mathbb{P}(S_{\mathsf{inn}} > s_j | \mathbf{y}, \mathbf{p})$.

6.2 The 'rare event' probability estimator

6.2.1 The scope of the algorithm

Problem (6.2) can be solved in a simple way with a Monte-Carlo simulation. It consists in randomly drawing N i.i.d. new codewords $\{\tilde{\mathbf{x}}_j\}_{j=1}^N$ (according to the distribution given by the secret $\mathbf{p}$). Sequence $\mathbf{y}$ was forged before, therefore these are codewords of innocent 'users'. We would like to test if z is a proper value for $\tau(\mathbf{y},\mathbf{p})$. The Monte Carlo gives the estimation $\hat{\mathbb{P}}(S_{\mathsf{inn}} > z) = N^{-1}|\{j|s(\tilde{\mathbf{x}}_j,\mathbf{y},\mathbf{p}) > z\}|$. This is simple but not efficient as $N = O(1/\mathbb{P}(S_{\mathsf{inn}} > z))$ for a given estimation accuracy. It becomes hardly tractable when the level ϵ_S on the probability of false positive is lower than 10^{-9}.

This section presents a 'rare event' simulation estimating small probabilities (or big quantiles) more efficiently. The scope of the algorithm is indeed much larger than traitor tracing. The general problem is to estimate the probability $P = \mathbb{P}(s(\mathbf{X}) > \tau)$. The algorithm is an adaptive version of Importance Splitting, a.k.a. Multilevel Splitting. Let us denote the distribution of $\mathbf{X}$ by $f_{\mathbf{X}}$ and its definition set $\mathcal{X}$. Our algorithm needs three routines:

- a (pseudo) random generator of independent samples distributed as $f_{\mathbf{X}}$,
- the score function $s(\cdot) : \mathcal{X} \to \mathbb{R}$,
- a random replicator $r(\cdot) : \mathcal{X} \to \mathcal{X}$ invariant to the specific distribution $f_{\mathbf{X}}$. This means that i) $r(\mathbf{x})$ is random and ii) the output $r(\mathbf{X})$ is distributed as $f_{\mathbf{X}}$ if the input $\mathbf{X}$ is distributed as $f_{\mathbf{X}}$.

6.2.2 Adaptive Importance Splitting with fixed effort

The idea of Importance Splitting is to consider a sequence of nested events $\mathcal{A}_N \subset \mathcal{A}_{N-1} \ldots \subset \mathcal{A}_1 \subset \mathcal{A}_0$. In our case, we defined them as $\mathcal{A}_j = \{\mathbf{x} \in \mathcal{X} | s(\mathbf{x}) > \tau_j\}$ with $-\infty = \tau_0 < \tau_1 \ldots < \tau_{N-1} < \tau_N = \tau$. In other words, we would like to estimate $P = \mathbb{P}(\mathcal{A}_N)$ which can be decomposed into:

$$P = \mathbb{P}(\mathcal{A}_N) = \mathbb{P}(\mathcal{A}_N|\mathcal{A}_{N-1})\mathbb{P}(\mathcal{A}_{N-1}|\mathcal{A}_{N-2}) \ldots \mathbb{P}(\mathcal{A}_1). \tag{6.5}$$

The estimation of P thanks to a numerical simulation is difficult because $\mathcal{A}_N$ is a rare event whose probability is small. The equation above breaks it into N easier problems because the conditional probabilities are much larger. Indeed, the algorithm estimates each conditional probability with a simple crude Monte Carlo simulation: Over n_j independent samples $\{\mathbf{X}_i^{(j)}\}_{i=1}^{n_j} \subset \mathcal{A}_j$, we count the number k_{j+1} of samples which also belong to $\mathcal{A}_{j+1}$ and $\hat{\mathbb{P}}(\mathcal{A}_{j+1}|\mathcal{A}_j) = {}^{k_{j+1}}/_{n_j}$. In practice, at each iteration, $n_j = n$, a parameter of the algorithm.

Our algorithm is adaptive because the subsets $\{\mathcal{A}_j\}_j$ are defined by the intermediate thresholds $\{\tau_j\}_{j=1}^{N-1}$ adaptively: at the j-th iteration, we fix $k_{j+1} = k$, a parameter of the algorithm lower than n, by setting τ_{j+1} as the $(k+1)$-th biggest scores observed in $\{s(\mathbf{X}_i^{(j)})\}_{i=1}^n$. In that way, exactly k samples have a score larger than τ_{j+1}.

If this intermediate threshold is larger than the target τ, the algorithm stops and we need to count the number k_N of scores larger than τ (which is bigger or equal than k). Note that these intermediate thresholds are indeed random variables and so is N, the total number of iterations.

In the end, the estimation of $P = \mathbb{P}(\mathcal{A}_N)$ is given by

$$\hat{P} = \prod_{j=1}^{N} \hat{\mathbb{P}}(\mathcal{A}_j|\mathcal{A}_{j-1}) = \rho^{N-1}.\frac{k_N}{n}, \tag{6.6}$$

with $\rho = k/n$.

The main difficulty is the generation of the random samples $\{\mathbf{X}_i^{(j)}\}_{i=1}^n \subset \mathcal{A}_j$. This set is indeed composed of the k samples of the previous iteration which belong to $\mathcal{A}_j$ plus $n-k$ 'fresh' new samples. A 'fresh' sample is generated as follows: we randomly pick a sample $\mathbf{Z}$ uniformly in $\mathcal{A}_j$ (among the k samples we already have), and we apply T iterations of the following routine:

$$\text{If } \mathbf{Y} = r(\mathbf{Z}) \in \mathcal{A}_j, \text{then } \mathbf{Z} \leftarrow \mathbf{Y}. \tag{6.7}$$

The random replicator proposes a random vector $\mathbf{Y}$, which is accepted (*i.e.* it replaces $\mathbf{Z}$) if $\mathbf{Y} \in \mathcal{A}_j$. Over T iterations, by constantly monitoring that $\mathbf{Z} \in \mathcal{A}_j$ we render the replicator invariant to $f_{\mathbf{X}|\mathcal{A}_j}$, *i.e.* the distribution $f_{\mathbf{X}}$ conditioned on $\mathcal{A}_j$. Moreover, as $T \to \infty$, the 'fresh' output sample becomes statistically independent of the initial sample $\mathbf{Z} \in \mathcal{A}_j$. We repeat this process $n-k$ times. In the end, we have n samples i.i.d. distributed as $f_{\mathbf{X}|\mathcal{A}_j}$ (k samples from the previous iteration and $n-k$ 'fresh' samples) which we use to estimate the next conditional probability $\hat{\mathbb{P}}(\mathcal{A}_{j+1}|\mathcal{A}_j)$.

6.2.3 Properties

If we assuming that the 'fresh' sample is always different than the input from which it has been derived (*i.e.* at least one of the T applications of the replicator was accepted as a new sample in $\mathcal{A}_j$), then the estimator is unbiased: $\mathbb{E}(\hat{P}) = P$.

In practice, T is a finite iteration number, therefore the samples are *a priori* not independent. We suppose that T is big enough to provide independence. This is the only approximation made in the proof of a Central Limit Theorem [48, 46]:

$$\sqrt{n}(\hat{P} - P) \xrightarrow[n\to\infty]{\text{law}} \mathcal{N}\left(0, P^2\left((N-1)\frac{1-\rho}{\rho} + \frac{n-k_N}{k_N}\right)\right). \tag{6.8}$$

We now measure the cost C of this algorithm by the number of calls to the routine computing the score function: Since $N \approx \log P / \log \rho$, we have

$$C = n + nT(N-1)(1-\rho) \approx nT\frac{1-\rho}{\log 1/\rho}\log 1/P. \tag{6.9}$$

The key feature is that the cost is proportional to $\log 1/P$ whereas the cost of the crude Monte Carlo is proportional to $1/P$. A standard measurement in the 'rare event' literature is the cost weighted relative variance[1] encompassing both the cost and the accuracy of the estimator:

$$C.\frac{\mathbb{V}(\hat{P})}{P^2} \approx (\log P)^2.\frac{T(1-\rho)^2}{(\log\rho)^2\rho}. \tag{6.10}$$

Usually, we set $\rho > 1/2$ and $T = 20$. With this setup, our algorithm has a lower cost weighted relative variance than the one of the crude Monte Carlo simulation (*i.e.* $(1-P)/P$) if $P \lesssim 10^{-3}$. The algorithm is thus dedicated to the estimation of small probabilities. Fig. 6.1 shows one estimation for a problem where the expression of the true probability is know. The algorithm succeeds to estimate a probability in the order of 10^{-11} with a good accuracy with just $850,000$ calls to the score function. A crude Monte Carlo simulation would have required more than 10^{12} calls.

[1] Some prefer to benchmark estimators with the *computational efficiency* which is indeed the inverse of the cost weighted relative variance.

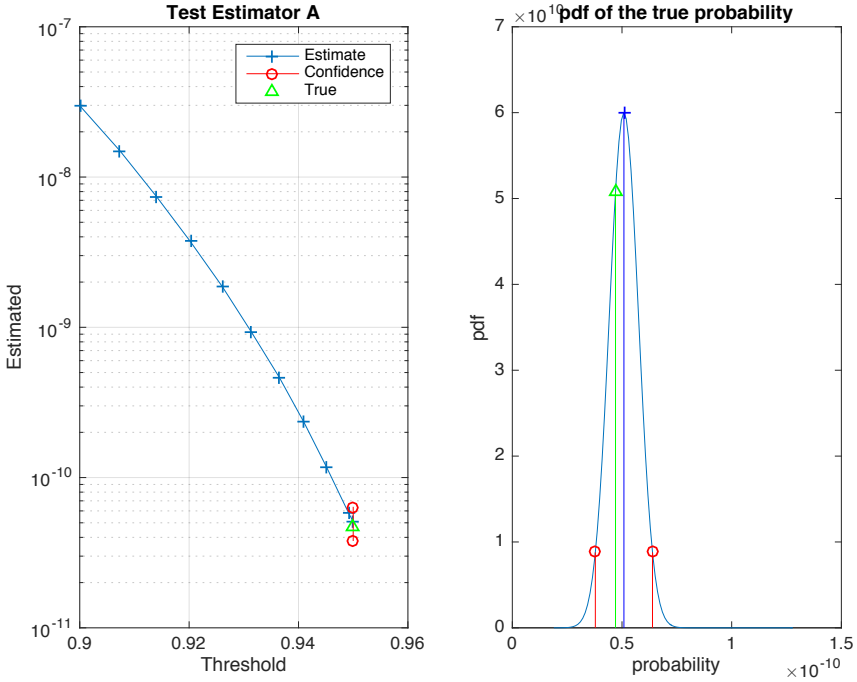


Figure 6.1: Example of one simulation. Estimation problem: $s(\mathbf{X}) = \mathbf{X}^\top \mathbf{u}/\|\mathbf{X}\|$, with $\|\mathbf{u}\| = 1$ and $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{20})$, $\tau = 0.95$. The true probability is $P = 4.7 * 10^{-11}$. Setup: $k = 1000$, $n = 2000$, $T = 10$. Results: $\hat{P} = 5.1 * 10^{-11}$, Confidence interval $[3.8, 6.4] * 10^{-11}$, $N = 35$, $C = 842,000$.

6.2.4 Improvements

This algorithm has been improved by A. Guyader, N. Hengartner and E. Matzner-Løber [116]. They noticed that (6.10) is indeed a decreasing function of ρ, therefore the algorithm is more efficient when setting ρ to its minimum value, $1 - 1/n$ for $k = n - 1$. This means that from one iteration to another, they keep all the samples except the 'last' one whose score is the lowest. They need to 'refresh' only this 'last' sample. The algorithm makes many more iterations as the conditional probabilities equal $1 - 1/n$: The algorithm makes tiny steps towards $\mathcal{A}_N$. This is *a priori* just a special case of our algorithm, but it has one huge advantage: The statistical properties of the estimator are proven for a finite n.

6.2.5 Byproducts

As a last word, both algorithms have the following byproducts:

- At the end of the simulation, we have examples of 'rare events'. This may help understanding what provoques such an event.
- The final output is not only a single estimation, but also a mapping $\{(\tau_j, \rho^j)\}_{j=1}^{N-1}$ (see Fig. 6.1 for our algorithm and Fig. 6.2 for [116]). This is quite useful for drawing Receiver Operating Characteristic in hypotheses testing (we need one simulation per hypothesis). In the same way, the algorithm gives confidence intervals and the probability density function of the true probability knowing the estimate (see Fig. 6.1 for our algorithm and Fig. 6.2 for [116]).

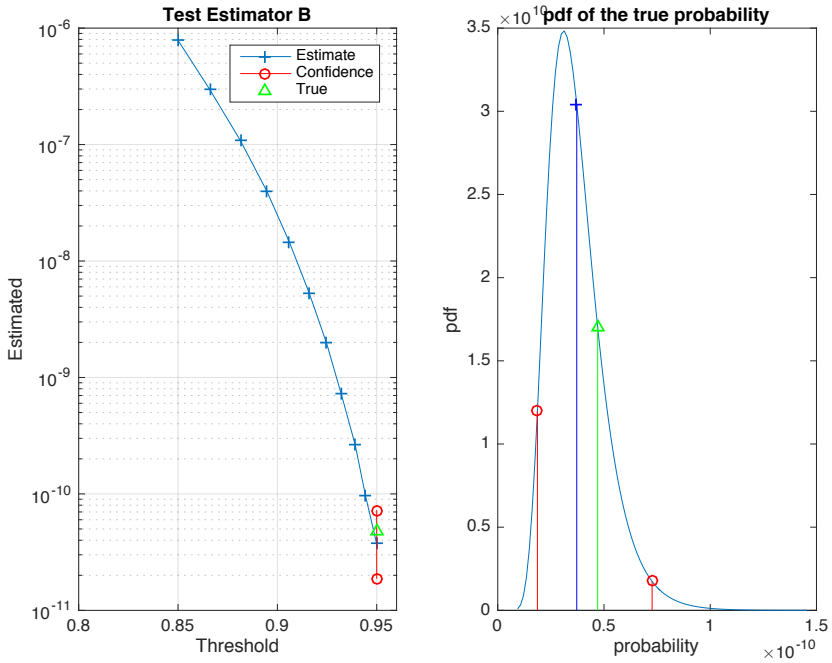


Figure 6.2: Example of one simulation. Estimation problem: $s(\mathbf{X}) = \mathbf{X}^\top \mathbf{u}/\|\mathbf{X}\|$, with $\|\mathbf{u}\| = 1$ and $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{20})$, $\tau = 0.95$. The true probability is $P = 4.7 * 10^{-11}$. Setup: $n = 200$, $T = 30$. Results: $\hat{P} = 3.7 * 10^{-11}$, Confidence interval $[1.9, 7.3] * 10^{-11}$, $N = 4,792$, $C = 138,000$.

- There is a version of the algorithm for estimating extreme quantile, *i.e.* estimate the value τ s.t. $\mathbb{P}(s(\mathbf{X}) > \tau)$ equals a given probability P [116].

6.3 Application to Tardos codes

To apply these algorithms to the decoding of Tardos codes, we need to specify the generator, the score, and the random replicator:

- $\mathbf{X}$ is a codeword (new and thus from an innocent user). Its law is:

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \Pi_{i=1}^{m} p_i^{x_i} (1 - p_i)^{(1-x_i)}. \tag{6.11}$$

  In practice, we flip m independent Bernoulli r.v. according to the secret sequence $\mathbf{p}$.

- The score is the score function of the single decoder $s(\mathbf{X}, \mathbf{y}, \mathbf{p})$.

- The random replicator copies the input codeword $\mathbf{X}$ in $\mathbf{Y}$. It randomly draws an index $I \sim \mathcal{U}_{\{1,\dots,m\}}$, and it re-generates the selected symbol: $Y_I \sim \mathcal{B}(p_I)$. This can be done several times before outputting $\mathbf{Y}$.

  Here are the three main advantages of these algorithms in traitor tracing.

6.3.1 Accurate estimation of probabilities

This algorithm replaces upper bounds on the probability of false positive (whose tightness is difficult to assess in practice) by an accurate estimation. As an example, we have generated a

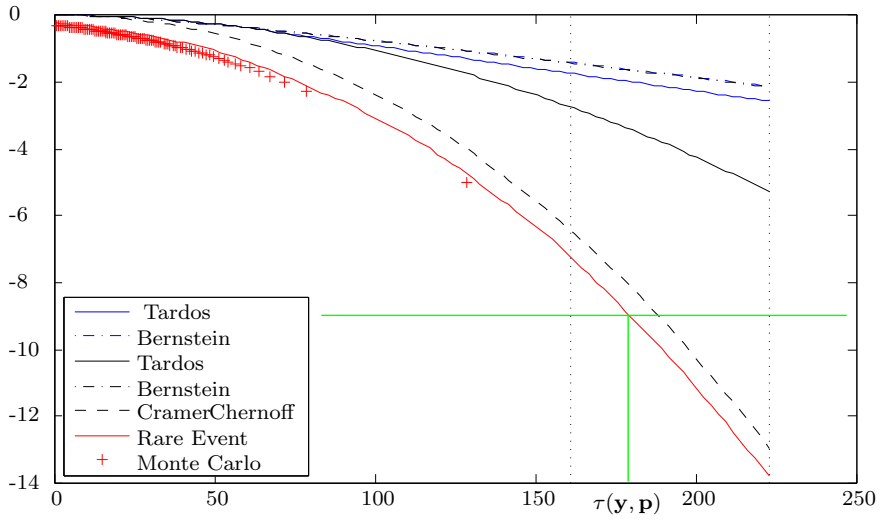


Figure 6.3: Bounds for the theoretical setup and the **operational** mode, and estimations of $\log_{10}(\mathbb{P}(S_{\mathsf{inn}} > \tau))$ for accusation score function (2.15). The minimum and the maximum of the $c = 6$ colluder scores are shown with dotted vertical lines. The rare event simulation recommends the threshold $\tau(\mathbf{y}, \mathbf{p})$ to meet the requirement on $\log_{10} \mathbb{P}_{\mathsf{fp}} \leq -9$.

sequence $\mathbf{p}$ of length $m = 1,024$ with the density $f^{(T)}$ (2.19), and $c = 6$ codewords colluding via the interleaving attack. Figure 6.3 shows some upper bounds and the estimations of $\mathbb{P}(S_{\mathsf{inn}} > \tau)$ for the score function (2.15) (with cut-off parameter $t = 3.3 * 10^{-4}$).

Suppose that we require $\mathbb{P}_{\mathsf{fp}} \leq \epsilon_S = 10^{-9}$. Figure 6.3 shows that m is not large enough to fulfill this requirement according to the theoretical upper bounds (Tardos or Bernstein's inequalities - blue curves) used in the proof of soundness in the literature. $\epsilon_S = 10^{-9}$ implies a threshold τ way too big to be on the figure. In this example, no colluders get caught because the maximum of the colluders scores is much smaller than an universal threshold τ recommended by these bounds.

Yet, the conclusion is very different in the operational mode. The Cramer-Chernoff bound or the rare event estimator give a customized threshold $\tau(\mathbf{y}, \mathbf{p})$ which is much smaller. It lies in between the minimum and the maximum of the colluder scores. Therefore, at least one colluder is caught in Fig. 6.3.

6.3.2 Mapping of the scores

Figure 6.3 also offers a map of a score s onto a probability $\pi = \Pi(s)$: The bounds or estimations yield a probability $\pi = \Pi(s) = \mathbb{P}(S_{\mathsf{inn}} > s)$, which reads as the probability that the score of an innocent is higher than s. In Fig. 6.3, the probability that an innocent has a score as big as the maximum of the colluders' scores is around 10^{-2} according to Tardos' bound. Nobody would take the risk of accusing him if the probability of being wrong is only 10^{-2}. Yet, the rare event estimation yields a probability around 10^{-14}, which raises much suspicion.

This stresses the fact that we need tight bounds or accurate estimations especially for the biggest scores. Figures 6.3 illustrates that the bounds of the theoretical setup fail achieving this goal. As for the estimators, the computational inefficacy of the Monte-Carlo approach prevents accurate mapping for too big scores.

A tight mapping is very helpful to decide whether users with the biggest scores are to be accused because it is more meaningful than raw scores or their comparisons to a threshold.

However, the probability π has to be related to the number of users n: The bigger n, the more likely at least one innocent user has a small π. Indeed if $c \ll n$, the probability that at least one innocent user has a mapping as low as π (equivalently a score as big as s) is $\eta = 1 - (1-\pi)^n$. In Fig. 6.3, the biggest colluders' score yields $\eta \approx n\pi = 10^{-8}$ if $n = 10^6$, which clears any doubt about his guiltiness.

6.3.3 Benchmark of single decoders

The scores obtained by two score functions might be totally different in nature. The mapping of a score onto a probability allows a fair comparison.

Here is an example of a benchmark of single decoders. The code is constructed with the usual 'Tardos' distribution $f^{(T)}$ (2.19). We consider two setups with short and long codes:

setup $\mathcal{A}$: $m = 256$, $c = 3$, $t = 5.5 * 10^{-4}$ and $c_{\max} = 6$,

setup $\mathcal{B}$: $m = 1\,024$, $c = 6$, $t = 3.3 * 10^{-4}$, and $c_{\max} = 10$.

Parameter $c_{\max}$ is needed for decoders (5.8), (5.10), and (5.11). In a nutshell, these latter decoders bet that there is no use in looking for more than $c_{\max}$ colluders for such a code length.

These decoders are facing the following collusion attacks: Coin flip, all-1, all-0, interleaving, WCA, majority, and minority (see their definition in Sect. 2.2.2). WCA is theoretically the worst case attack defined as the minimizer $\breve{\boldsymbol{\theta}}_c^{(S)}$ of the averaged mutual information per sample $\mathbb{E}_P(I(Y;X|P))$ for $P \sim f^{(T)}$ (See Sect. 4.4).

Our experimental protocol is composed of $N_r = 200$ runs. A run starts by generating $\mathbf{p}$ and c colluder codewords $\{\mathbf{x}_j\}_{j=1}^c$, which then forge $\mathbf{y}$ according to a given attack. The single decoder computes the scores $\{s_j\}_{j=1}^c$ for this particular $\mathbf{y}$. The minimum $s_{\min}$ and maximum $s_{\max}$ of these c scores are then translated into probabilities $\Pi(s_{\min})$ and $\Pi(s_{\max})$ respectively thanks to the rare event simulation.

Figures 6.4 and 6.5 show some statistics (median, 5% and 95% quantiles) about $\Pi(s_{\min})$ and $\Pi(s_{\max})$ over $N_r = 200$ simulation runs. The best decoder is the one providing the smallest probabilities, which will trigger accusation. The benchmark encompasses the following single decoders:

(T) Tardos decoder (2.15),

(O) Oosterwijk decoder (5.9),

(L) Laarhoven decoder (5.8),

(D) Desoubeaux generalized linear decoder (5.11),

(wca) MAP decoder tuned on the worst case attack (5.7),

(g.wca) generalized linear decoder aggregating MAP decoders tuned on worst case attacks (5.10),

(mle) MAP decoder tuned on the Maximum Likelihood Estimator $\hat{\boldsymbol{\theta}}_{c_{\max}}^{(\mathsf{MLE})}$ (5.12),

(em) MAP decoder tuned on the Expectation Maximisation Estimation $\hat{\boldsymbol{\theta}}_{c_{\max}}^{(\mathsf{EM})}$ (5.16),

(g.mle) generalized linear decoder aggregating MAP decoders tuned on Maximum Likelihood Estimations $\{\hat{\boldsymbol{\theta}}_c^{(\mathsf{MLE})}\}_{c=2}^{c_{\max}}$,

(g.em) generalized linear decoder aggregating MAP decoders tuned on Expectation Maximisation Estimations $\{\hat{\boldsymbol{\theta}}_c^{(\mathsf{EM})}\}_{c=2}^{c_{\max}}$

This benchmark leads to the following conclusions:

- When the collusion strategy is strong (*e.g.* WCA, Uniform, or Coin Flip), all the single decoders perform more or less the same, *i.e.* as good as (or as bad as) the Tardos decoder.
- Among the *fixed* single decoders, there is a slight advantage in using a generalized linear decoder like (D) (5.11) or (g.wca) (5.10). Note however that their complexity is $\approx (c_{\max}-1)$ times bigger than a linear decoder.
- The scores are always mapped onto lower probabilities than their counter-part for the worst case attack. This shows that these decoders are robust. This is the practical advantage of decoders having theoretical achievable rates (for a given collusion strategy) always larger or equal to the fingerprinting capacity (achievable rate under the worst case attack –Sect. 4.5).
- When the collusion strategy is not so harmful (*e.g.* All1, All0, Minority for $c = 6$), adaptive decoders perform much better than fixed decoders.
- Among the *adaptive* decoders, the generalized linear decoders (*i.e.* (g.mle) and (g.em)) bring a more pronounced advantage (than their fixed versions (D) and (g.wca)), especially more constant over the collusion strategy. Again, the price to be paid is a bigger complexity.
- The type of estimation of $\hat{\boldsymbol{\theta}}$, MLE or EM, does not make a difference. The complexity of the E-M algorithm seems to be more practical, or under control (with a fixed number of iterations).

6.4 Conclusion

This algorithm has been applied to Tardos codes for single and joint decoding [47, 42], but also to the evaluation of the probability of false alarm in watermarking [49], and the evaluation of the security level of a watermarking primitive [15, 4].

This algorithm has also some limitations. The algorithm estimates probabilities of the form $\mathbb{P}(h(\mathbf{Z}) > \tau)$. It has been applied to Tardos code by setting $\mathbf{Z}$ as a codeword $\mathbf{X}_{\mathsf{inn}}$ of an innocent user and the score $h(\mathbf{Z})$ as the single decoder score function $s(\mathbf{X}, \mathbf{y}, \mathbf{p})$. This fully covers the needs in the operational mode.

In the theoretical mode, we can imagine estimating the probability $\mathbb{P}(s(\mathbf{Y}, \mathbf{X}_{\mathsf{inn}}, \mathbf{P}) > \tau)$: It means that sample $\mathbf{Z}$ is now a random element composed of $c + 3$ vectors:

- a bias sequence $\mathbf{P}$,
- the codeword of an innocent user $\mathbf{X}_{\mathsf{inn}}$,
- the codewords of c colluders,
- the pirated sequence $\mathbf{Y}$ made under a given collusion strategy $\boldsymbol{\theta}_c$.

while $h(\mathbf{Z})$ becomes $s(\mathbf{X}_{\mathsf{inn}}, \mathbf{Y}, \mathbf{P})$. I have tested this for q-ary Tardos codes with the idea of comparing with the theoretical derivations of A. Simone and B. Škorić . I miserably failed and, worse than that, I still do not fully know why.

I have also tried to evaluate the error exponent characteristic of some zero-bit watermarking schemes (see Part II). The idea is to evaluate probabilities of false positive $\{\hat{P}_i\}_i$ for some signal lengths $\{n_i\}_i$. Then, a least square regression estimates the error exponent as the slope of the points $\{n_i, -\log \hat{P}_i\}$. This was not very successful. We need long lengths $\{n_i\}$ to evaluate the limit $\lim_{n\to\infty} -\log P_n/n$ by just the ratio $-\log P_n/n$. But, long signals mean slower simulation and very very small probabilities to be estimated.

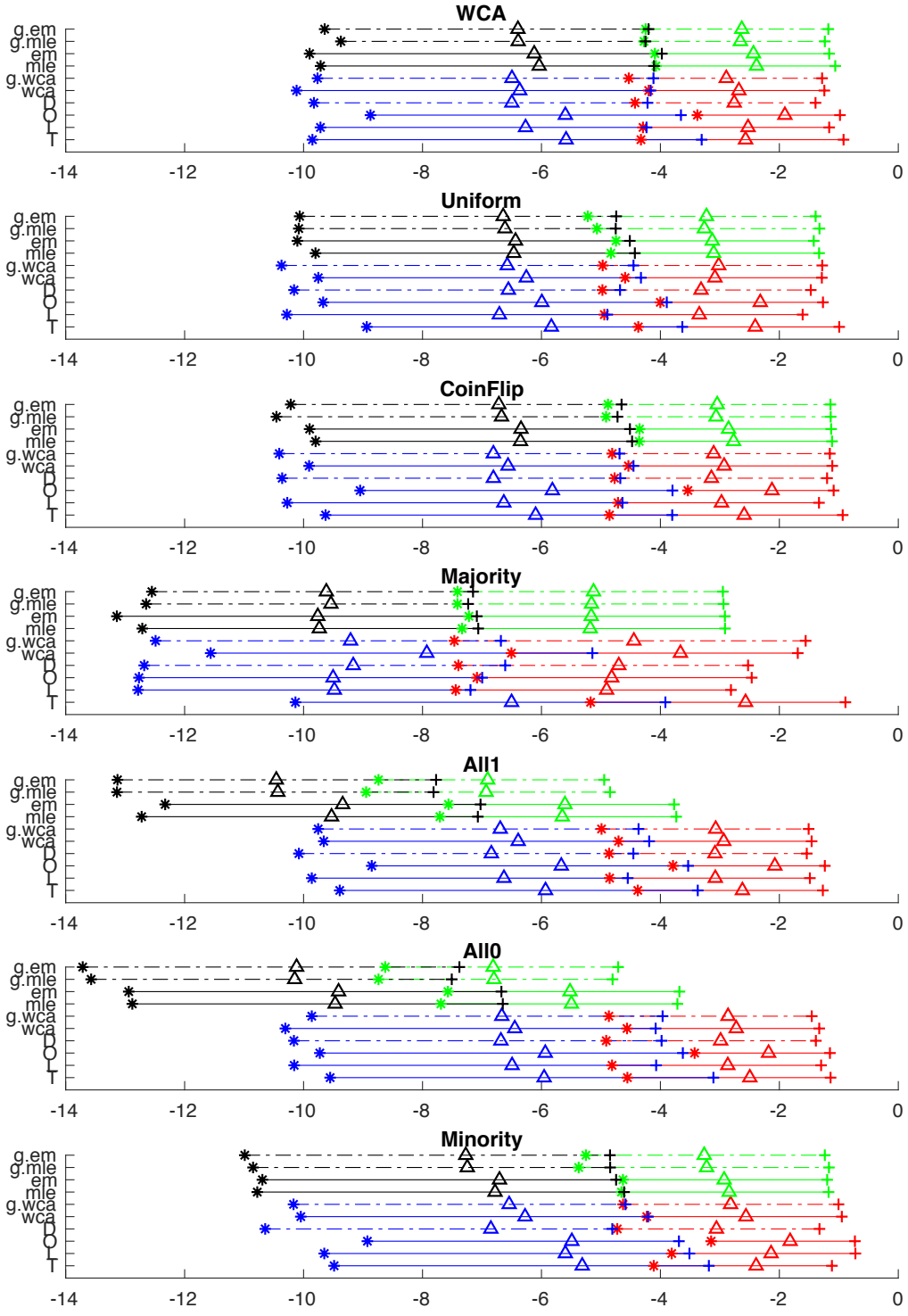


Figure 6.4: Benchmark of single decoders under setup $\mathcal{A}$. Statistics of $\log_{10}(\Pi(s_{\min}))$ (blue for the fixed decoders –Sect. 5.2.2, black for the adaptive decoders –Sect. 5.2.4) and $\log_{10}(\Pi(s_{\max}))$ (red for the fixed decoders –Sect. 5.2.2, green for the adaptive decoders –Sect. 5.2.4): Median ($\triangle$), 5% ($*$) and 95% ($+$) quantiles. The dashed lines show the generalized linear decoders.

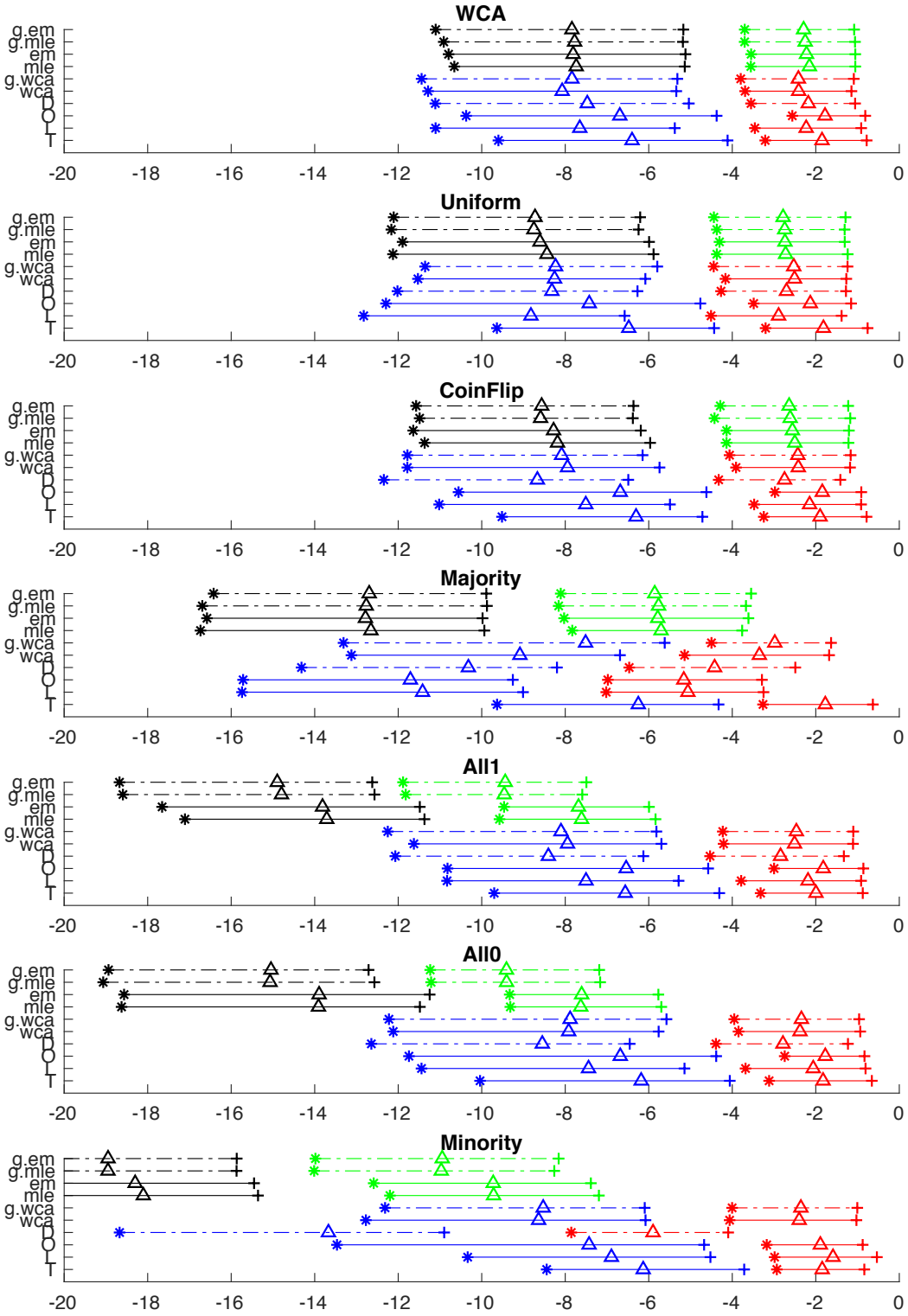


Figure 6.5: Benchmark of single decoders under setup $\mathcal{B}$. Statistics of $\log_{10}(\Pi(s_{\min}))$ (blue for the fixed decoders –Sect. 5.2.2, black for the adaptive decoders –Sect. 5.2.4) and $\log_{10}(\Pi(s_{\max}))$ (red for the fixed decoders –Sect. 5.2.2, green for the adaptive decoders –Sect. 5.2.4): Median ($\triangle$), 5% ($*$) and 95% ($+$) quantiles. The dashed lines show the generalized linear decoders.

Part II

Zero-bit watermarking

Chapter 7

Error exponents of zero-bit watermarking

This part of the habilitation investigates the robustness of zero-bit watermarking schemes from a theoretical point of view based on the exponents of error detection probabilities. This first chapter introduces zero-bit watermarking and the setup of the study. This framework is theoretical but it should encompass the specificities of zero-bit watermarking detailed in Sect. 7.4. Even if these specificities are coarsely sketched in the model, they bring limitations that help us deriving a piece of theory whose relevance in practice is more solid.

For the reader expert in the field, the fundamental quest underlying this part is to answer the questions of M. Costa (see Sect. 7.5): Is it possible to achieve performances which do not depend on the host thanks to the use of side-information at the embedding side? In that case, how do they compare to the performances of a non-blind detector which knows the host signal? What information about the problem does such a scheme needs to achieve this Holy Grail?

7.1 Zero-bit watermarking

Zero-bit watermarking is different from multi-bit watermarking. While people usually knows what watermarking means, some get confused between the *detection* and the *decoding* of a watermark. In multi-bit watermarking, a first algorithm, so-called embedder, hides a message (possibly encoded in several bits) into a piece of content. A second algorithm analyses a piece of content and proceeds to a decoding (see Fig. 7.1). The decoding outputs the hidden message or the decision that the piece of content under scrutiny is indeed not watermarked.

In *zero-bit* watermarking, one is solely interested in distinguishing watermarked from non watermarked content. Therefore, the embedding does not hide any message, but just a mark. There is no modulation of a signal by the message to be transmitted since there is no message.

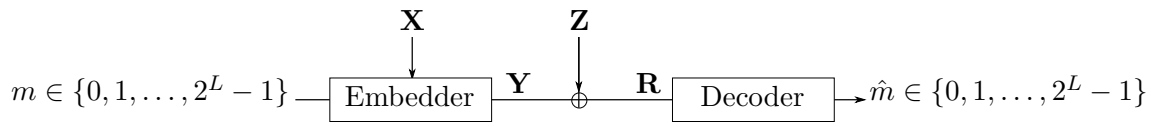


Figure 7.1: Multi-bit watermarking. The message to be embedded m is encoded within L bits. In this scenario, contents are always watermarked so that the problem is the decoding of the hidden message.

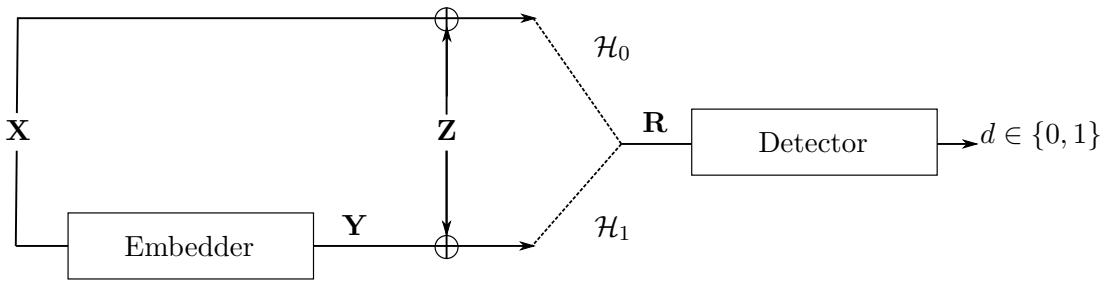


Figure 7.2: Zero-bit watermarking. The embedder hides a mark into the content. The detector checks for the presence of this mark.

Hence, the term *zero-bit* watermarking. In the same way, the second algorithm does not perform a decoding, but a detection of the presence or the absence of the mark (see Fig. 7.2).

There has been some confusion with the terminology 'one-bit watermarking': A 'one-bit' watermarking scheme is when one detects and then decodes a message of a single bit. Hence, there are three cases: the content under scrutiny is not watermarked, is watermarked with a '1', or is watermarked with a '0'. In some applications, one is sure that the received content is always watermarked. Therefore, there are only two hypotheses to be tested as in zero-bit watermarking. This similarity brings confusion. In 'one-bit' watermarking, the received signal has been modified under both hypotheses by the embedder to hide either the '0' or the '1' symbol. On contrary, in zero-bit watermarking, the content has not been modified under one hypothesis ('Non watermarked'). It is given by Nature.

7.2 Notations

A feature vector in $\mathbb{R}^n$ is extracted from a piece of multimedia content. Vectors $\mathbf{x}$ and $\mathbf{r}$ denote respectively the extracted features from an original content, so-called the host, and from the content received by the detector. The embedder transforms $\mathbf{x}$ into $\mathbf{y}$ by adding a watermark $\mathbf{w}$: $\mathbf{y} = \mathbf{x} + \mathbf{w}(\mathbf{x})$. This vector depends on the host (for a side-informed watermarking scheme) and on a secret key (not indicated to keep notations simple).

We consider a power constraint watermark problem where the energy of the watermark per sample is limited. The literature usually considers two definitions:

- Strict embedding constraint:
$$\frac{1}{n}\|\mathbf{w}(\mathbf{x})\|^2 \leq P, \forall \mathbf{x} \in \mathbb{R}^n, \tag{7.1}$$
- Embedding constraint on expectation:
$$\frac{1}{n}\mathbb{E}[\|\mathbf{w}(\mathbf{X})\|^2] \leq P. \tag{7.2}$$

The Euclidean norm of vector $\mathbf{x} \in \mathbb{R}^n$ is denoted by $\|\mathbf{x}\|$, and $\mathbb{E}[A]$ is the expectation of random variable A.

The model of an attack is the addition of a noise vector $\mathbf{z}$, and the received vector extracted from the content under scrutiny is $\mathbf{r} = \mathbf{y} + \mathbf{z}$.

At the detection side, two hypotheses are competing. Under hypothesis $\mathcal{H}_0$, the received vector has not been watermarked. Under hypothesis $\mathcal{H}_1$, the received vector has been water-

marked. The decision of the detector is denoted by d: $d = 1$ if the received content is deemed watermarked, $d = 0$ otherwise. There are two types of errors:

Under $\mathcal{H}_0$: $d = 1$ whereas the received vector has not been watermarked. This is a *false positive* whose probability is denoted by $\mathbb{P}_{\mathsf{fp}} := \mathbb{P}[d = 1|\mathcal{H}_0]$.

Under $\mathcal{H}_1$: $d = 0$ whereas the received vector has been watermarked. This is a *false negative* whose probability is denoted by $\mathbb{P}_{\mathsf{fn}} := \mathbb{P}[d = 0|\mathcal{H}_1]$.

To take a decision, we assume that the detector first computes a score from received vector $\mathbf{r}$: $s = s(\mathbf{r})$ with $s(\cdot) : \mathbb{R}^n \to \mathbb{R}$. Then, it compares this score to a threshold τ: $d = 1$ if $s \geq \tau$ and 0 otherwise. This defines the region $\mathcal{W} \subset \mathbb{R}^n$ of the vectors deemed as watermarked:

$$\mathcal{W} = \{\mathbf{x} \in \mathbb{R}^n | s(\mathbf{x}) \geq \tau\}. \tag{7.3}$$

7.3 Asymptotical and Gaussian setup

The theoretical setup models the host and noise vector by random vectors $\mathbf{X}$ and $\mathbf{Z}$ distributed as white Gaussian random vectors: $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_n, \sigma_X^2 \mathbf{I}_n)$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}_n, \sigma_Z^2 \mathbf{I}_n)$. Vector $\mathbf{0}_n$ denotes the vector of n zero components, and $\mathbf{I}_n$ the identity matrix of size n. This stems into the following statistical model of the received vector $\mathbf{R}$:

$\mathcal{H}_0$: $\mathbf{R} = \mathbf{X} + \mathbf{Z}$,

$\mathcal{H}_1$: $\mathbf{R} = \mathbf{X} + \mathbf{w}(\mathbf{X}) + \mathbf{Z}$.

Hypothesis $\mathcal{H}_0$ deserves some explanations. The addition of $\mathbf{Z}$ models an attack whose goal would be to produce a false negative error. There is a priori no reason why an attacker would add noise on a content which is not watermarked. From a practical point of view, pieces of content are always edited or post-processed. From a theoretical point of view, not including any noise under $\mathcal{H}_0$ may yield a detector checking the presence of noise rather than spotting the watermark. To avoid this, the two hypotheses must get closer (the distance between the distributions under $\mathcal{H}_0$ and $\mathcal{H}_1$ vanishes) as the power P of the watermark goes to zero.

The aim of this study is to characterize how fast the error probabilities vanish to zero as n increases. To this aim, the error exponents are defined as follows:

$$E_{\mathsf{fp}} := \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}_{\mathsf{fp}}, \quad E_{\mathsf{fn}} := \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}_{\mathsf{fn}}. \tag{7.4}$$

A strictly positive exponent indicates that the related error probability vanishes exponentially fast to zero as n increases. A null exponent means that the probability does not converge to zero or decreases to zero but not exponentially.

It is expected to have a trade-off between the false positives and false negatives: Both error exponents cannot be big at the same time. By carefully crafting series of thresholds $\{\tau_n\}$ for increasing n, it is possible to investigate this trade-off, *i.e.* to find the characteristic $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$ where $F(\cdot)$ is a priori a decreasing function from $[0, +\infty)$ to $[0, +\infty)$.

In the sequel, we focus on two particular points of the characteristic: the left and the right endpoints. The graph of the function $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$ starts on the left by the point $(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = (0, E_{\mathsf{fn}}^L)$ where $E_{\mathsf{fn}}^L := \lim_{E_{\mathsf{fp}}\to 0^+} F(E_{\mathsf{fp}})$. It is possible to achieve higher false negative rate but then, for sure, $E_{\mathsf{fp}} = 0$ (for instance, $\mathbb{P}_{\mathsf{fp}}$ does not converge to 0 for such a high false negative rate). E_{fn}^L

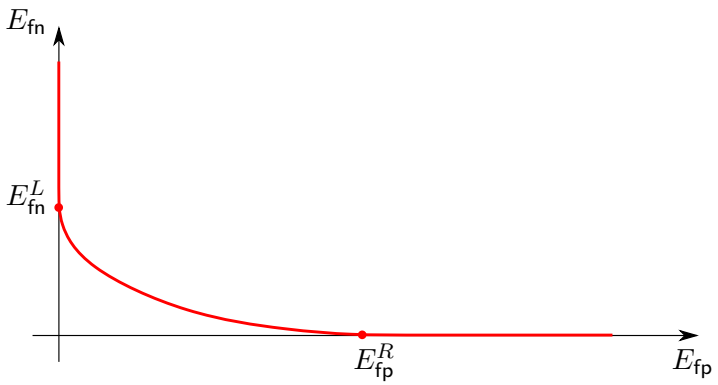


Figure 7.3: The error exponent characteristic $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$. E^R_{fp} (E^L_{fn}) is defined by the point where the characteristic hits the x-axis (y-axis respectively).

is thus the minimum false negative rate for which $E_{\mathsf{fp}} = 0$. On the other hand, the interesting part of graph $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$ ends on the right by the point $(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = (E^R_{\mathsf{fp}}, 0)$. Again, larger false positive rate are achievable but then, for sure, $E_{\mathsf{fn}} = 0$. E^R_{fp} can be seen as the minimum false positive rate for which $E_{\mathsf{fn}} = 0$.

7.4 Specificities of watermarking

The asymptotical and Gaussian setup is an artificial sandbox: In practice, n is large but not infinite, and signals are not stationary, white and Gaussian distributed. These assumptions are here to simplify the analysis. Nevertheless, this setup should not be too disconnected from reality and we should pay attention to specificities of some watermarking applications:

- **Variance σ_X^2 is not fixed**. Content to be watermarked have a huge diversity at least when we consider multimedia contents such as images, audio and video clips. The theoretical model is also wrong when pretending that, from one content to another, features vectors share the same statistical model. To make a small step towards the real world, we may assume that the detector does not know σ_X^2 because this variance is not fixed from one piece of content to another.

- **The embedding power P is not fixed**. For the same reason of diversity, pieces of multimedia content have different masking properties, which impact the non-perceptibility of the watermark signal. In academic papers dealing with image or video watermarking, the embedding constraint is often stated as a targeted PSNR in between the host and the watermarked content. This fixes P. In audio watermarking, the target is given by a SNR, which makes P varying. In the same way, real world applications aiming at pristine quality of watermarked content use complex human perceptual model analyzing the masking properties (both in frequency and in time/space domains). It is then less wrong to consider that P varies from one content to another. Power P is positive, certainly small compared to σ_X^2, and above all, unknown to the detector. Running the same human perceptual model at the detection to obtain an estimation of P is hazardous, especially under mean attacks.

- **The embedding constraint on expectation** (7.2) raises the question of what is random. There are a priori two sources of randomness. First, watermarking is usually operated by a secret key. This chapter does *not* deal with mechanism making the embedding and detection private. The secret key does not appear in the notations for the sake of simplicity.

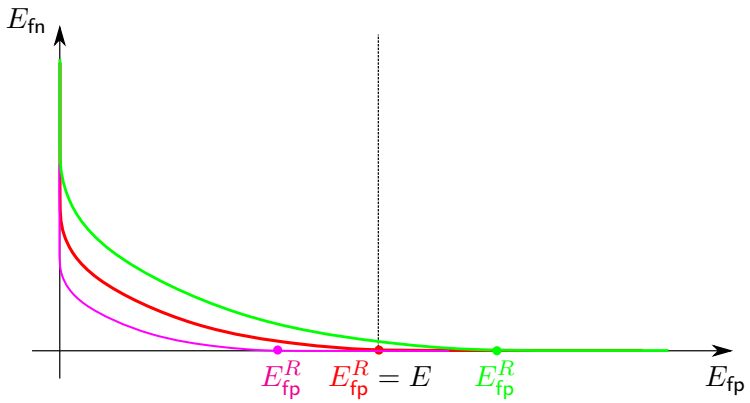


Figure 7.4: The watermark detector operates at $E_{\mathsf{fp}} = E$. Three characteristics are depicted as we consider three noise powers. The watermark is deemed robust against the 'green attack' ($E_{\mathsf{fp}}^R > E$) but not against the 'magenta attack' ($E_{\mathsf{fp}}^R < E$). The 'red attack' is the limiting case ($E_{\mathsf{fp}}^R = E$) whose noise power defines $\bar{\sigma_Z}^2$.

The question is whether the secret key plays the role of a source of randomness. Drawing randomly a secret key at each embedding call is not possible in many applications because there is no auxiliary channel between the embedder and the detector to 'synchronize' the secret key. For instance, the watermark detector inside Blu-Ray disc players has a fixed secret key which has been drawn once for all by 'Hollywood movie industry'. Second, the host signal $\mathbf{X}$ is another source of randomness when the embedding is side-informed: the watermark signal $\mathbf{W} = \mathbf{w}(\mathbf{X})$ depends on $\mathbf{X}$ and the embedding constraint is an expectation of $\|\mathbf{W}\|^2$ over $\mathbf{X}$.

- **Variance σ_Z^2 is not fixed**. There is also a wide diversity of attacks. Modelling an attack by the addition of a white Gaussian noise is pure theory. But, the biggest misconception may be that the embedder and the detector know the noise power. In practice, the goal is to be as robust as possible: $\mathbb{P}_{\mathsf{fn}}$ should smoothly degrades as σ_Z^2 gets stronger.

- **False positives matter more than false negatives**. In many applications, a false positive means accusing someone innocent (*i.e.* in copyright protection) or stopping the playback of a content whereas the user has the right to do so (*i.e.* in copy protection). Probability $\mathbb{P}_{\mathsf{fp}}$ is required to be very small, which, in our theoretical setup, means E_{fp} set to a high value. On the other hand, watermarking is usually a dissuasive means: the probability to catch the attacker or to prevent the illegal use of a pirated content should be strictly positive; but dissuasion doesn't need $\mathbb{P}_{\mathsf{fn}}$ to be exponentially vanishing.

- **Requirement on the false positive probability**. Whereas $\mathbb{P}_{\mathsf{fn}}$ cannot be under control since the power of the attack is a priori unknown, requirements usually set a level for the probability of false positive detection $\mathbb{P}_{\mathsf{fp}}$. What matters in practice is to tune the watermarking scheme to meet this requirement. This translates in our theoretical setup as the possibility to set values for a series of thresholds $\{\tau_n\}$ such that, asymptotically, a given error exponent E_{fp} is achieved.

The penultimate point outlines the fact that the right hand side of the characteristic (large E_{fp} but small E_{fn}) is the most interesting part. This stresses the importance of the quantity E_{fp}^R and its dependence on $(P, \sigma_X^2, \sigma_Z^2)$.

The last point suggests the following evaluation criterion: The watermark designer decides to operate at $E_{\mathsf{fp}} = E$ ($E > 0$). He/she needs $n \approx -\log(\mathbb{P}_{\mathsf{fp}})/E$ samples to meet the requirement

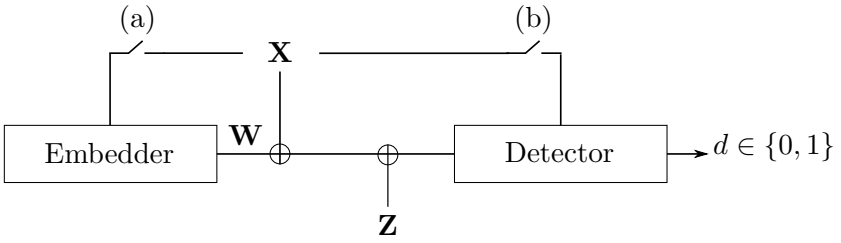


Figure 7.5: Case 1: Non side informed embedder and blind detector (switches (a) and (b) are open). Case 2: Side informed embedder and blind detector (switch (a) closed, (b) open). Case 3: Non blind detector (switch (b) closed, (a) open or closed).

on the false positive probability. The design of the watermark detector must enforce this error exponent. If $E_{\mathsf{fp}}^R > E$ for a given attack and embedding power, then the operator is sure that $E_{\mathsf{fn}} = F(E) > 0$. The watermark is deemed robust. If $E_{\mathsf{fp}}^R < E$, then $F(E) = 0$ and the watermark may not be dissuasive enough. Fig. 7.4 illustrates the differences. The game is not to maximize E_{fn} for the given set of parameters $(E_{\mathsf{fp}}, P, \sigma_X^2, \sigma_Z^2)$. It makes more sense to maximize the range of σ_Z^2 for which $E_{\mathsf{fn}} > 0$ at a required $E_{\mathsf{fp}} = E$.

Proposition 7.4.1 *A meaningful definition of robustness in our context is the following: For a given setup $(E, P, \sigma_X^2, \sigma_Z^2)$, a watermarking scheme is deemed robust if $E_{\mathsf{fp}}^R \geq E$. The maximum noise power for which this inequality holds is denoted by $\bar{\sigma_Z}^2$.*

To conclude, the next chapters analyze the error exponent characteristic of several watermarking schemes. We shall pay attention to which parameters are needed on the embedding and detection sides to achieve these theoretical performances.

7.5 The question of M. Costa

M. Costa considers three cases in the context of digital communication with side-information [101]. In 1983, digital watermarking did not exist. These cases, depicted in Fig. 7.5, are translated to zero-bit watermarking terminology as follows:

1. **Neither the embedder, nor the detector knows the host signal X**. The embedder is not side-informed and the detector is blind. Switches (a) and (b) are open in Fig. 7.5. The embedder emits a signal $\mathbf{W}$ independent of $\mathbf{X}$, then the channel adds first $\mathbf{X}$ and then $\mathbf{Z}$. These sources of noise being Gaussian and independent, $\mathbf{N} = \mathbf{W} + \mathbf{Z}$ is distributed as $\mathcal{N}(\mathbf{0}_n, N\mathbf{I}_n)$ with $N = \sigma_X^2 + \sigma_Z^2$. The performances depend on P and N.

2. **The embedder knows X, but not the detector**. The embedder is side-informed and the detector is blind. Switch (a) is closed but switch (b) is open in Fig. 7.5. Many watermarking applications follow this case.

3. **The embedder and the detector knows X**. Both switches are closed in Fig. 7.5. The detector is not blind and it removes $\mathbf{X}$ from the received signal $\mathbf{R}$. That way, the embedder may not use $\mathbf{X}$ because $\mathbf{W}$ only suffers from one source of noise, $\mathbf{N} = \mathbf{Z}$. The performances depend on P and $N = \sigma_Z^2$.

From cases 1 to 3, we keep on taking into account more information (more switches are closed). Therefore, the performances of case 2 (under our best efforts) should lie in between the perfor-

mances of cases 1 and 3. In other words, cases 1 and 3 play the role of the lower and upper bounds.

In article [101], Costa considers communication with side-information at the emitter (*i.e.* a *decoding* problem). He measures performances by the channel capacity, and shows that:

$$C_1 = \frac{1}{2}\log\left(1+\frac{P}{\sigma_X^2+\sigma_Z^2}\right) \leq C_2 \leq C_3 = \frac{1}{2}\log\left(1+\frac{P}{\sigma_Z^2}\right). \tag{7.5}$$

Then he exhibits a scheme under case 2 whose achievable rate does not depend on σ_X^2 and moreover matches C_3. This proves that, thanks to the side-information at the emitter, $C_2 = C_3$.

In our *detection* problem, we may measure the performances by the error exponent characteristic and the same rationale translates as

$$F_1(E_{\mathsf{fp}}) \leq F_2(E_{\mathsf{fp}}) \leq F_3(E_{\mathsf{fp}}). \tag{7.6}$$

The fundamental question in zero-bit watermarking is whether there exists a scheme such that $F_2(\cdot) = F_3(\cdot)$. Indeed, we are more interested if such equality happens for high E_{fp}. In the same way, the characteristic $F(\cdot)$ can be replaced by more relevant measurements like E_{fp}^R or $\bar{\sigma_Z}^2$ in inequality (7.6) and the underlying fundamental question.

Chapter 8 investigates schemes under case 1. Their performances translate to case 3 replacing $N = \sigma_X^2 + \sigma_Z^2$ by $N = \sigma_Z^2$. Chapter 9 investigates zero-bit watermarking under case 2 in the noiseless scenario: $\sigma_Z = 0$, while Chapter 10 assumes $\sigma_Z > 0$.

7.6 Conclusion

As stated so far, this fundamental question is ill-posed w.r.t. to the specificities of watermarking listed in Sect. 7.4. We need to be careful about the working assumptions. This especially holds for the knowledge the embedder and the knowledge of the detector about parameters $(P, \sigma_X^2, \sigma_Z^2)$. Two schemes can only be compared if they work under the same assumptions.

In the following chapters, these assumptions vary from one scheme to another, but they are always clearly stated. It is obvious that the schemes where the embedder is oblivious to σ_Z^2 and the detector is oblivious to $(P, \sigma_X^2, \sigma_Z^2)$ are more practical in real-life applications.

Chapter 8

One unique source of noise

Before investigating zero-bit side-informed watermarking, this section elaborates on a simpler problem defined as:

$\mathcal{H}_0$: $\mathbf{R} = \mathbf{N}$,

$\mathcal{H}_1$: $\mathbf{R} = \mathbf{w} + \mathbf{N}$,

with $\mathbf{N} \sim \mathcal{N}(\mathbf{0}_n, N\mathbf{I}_n)$. This models two cases introduced in the previous chapter:

- Case 1: zero-bit watermarking *without* side-information at the embedding side and blind detection. Since the host and the noise sources are independent, $N = \sigma_X^2 + \sigma_Z^2$. The host is not a side information but a source of noise and the watermark signal cannot depend on $\mathbf{X}$. It is a constant vector of squared norm $\|\mathbf{w}\|^2 = nP$ shared by the embedder and the detector.
- Case 3: zero-bit watermarking with a non-blind detection. The detector removes $\mathbf{X}$ and the embedder is not obliged to take it into account. In that case, $N = \sigma_Z^2$.

8.1 Optimal Neyman-Pearson detector

Appendix 15 explains how to derive the error exponents with a probabilistic point of view using the moment generating function. It applies this method to Spread Spectrum in App. 15.1. It shows that by considering the Neyman-Pearson test (the score function is the likelihood ratio $s(\mathbf{R}) = {}^{p(\mathbf{R}|\mathcal{H}_1)}/_{p(\mathbf{R}|\mathcal{H}_0)}$), and the Chernoff's bound for both $\mathbb{P}_{\mathsf{fn}}$ and $\mathbb{P}_{\mathsf{fp}}$ which gets tighter as n increases, the characteristic for this simple detection problem is given by:

$$E_{\mathsf{fn}} = \left(\left| \sqrt{\frac{P}{2N}} - \sqrt{E_{\mathsf{fp}}} \right|_+ \right)^2, \tag{8.1}$$

with $|a|_+ := a$ if $a > 0$, and 0 otherwise. This gives birth to left and right endpoints:

$$E_{\mathsf{fn}}^L = E_{\mathsf{fp}}^R = \frac{P}{2N}. \tag{8.2}$$

When operating at $E_{\mathsf{fp}} = E$, the watermark is robust (in the sense that $E_{\mathsf{fp}}^R \geq E$) when $N \leq {}^N/_{2P}$, which translates to

$$\text{Case 1: } \bar{\sigma_Z}^2 = \left| \frac{P}{2E} - \sigma_X^2 \right|_+, \quad \text{Case 3: } \bar{\sigma_Z}^2 = \frac{P}{2E}. \tag{8.3}$$

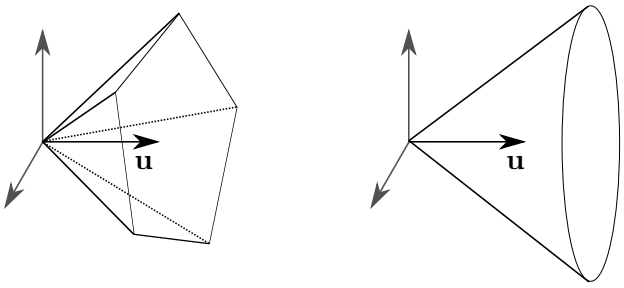


Figure 8.1: According to Shannon, the circular cone (right) is optimum in the sense that it maximizes $Q(\Omega)$ for a given solid angle Ω.

The bigger E, the less robust the watermark is. Roughly speaking, for a required $\mathbb{P}_{\mathsf{fp}}$, operating at $E_{\mathsf{fp}} = E$ implies that the dimension of the vectors is about $n \approx -\log \mathbb{P}_{\mathsf{fp}}/E$. The bigger E, the shorter the vectors are. As a consequence, for instance under case 3, $\bar{\sigma_Z}^2 \approx n^P/2|\log \mathbb{P}_{\mathsf{fp}}|$. We rediscover here the well-known rule of thumb of digital watermarking: the more spread (*i.e.* large n), the more robust the watermark is (large $\bar{\sigma_Z}^2$).

Obliviousness to parameters (P, N) prevents computing $p(\mathbf{R}|\mathcal{H}_0)$ and $p(\mathbf{R}|\mathcal{H}_1)$. But, by applying a suitable increasing function, the likelihood ratio indeed boils down to the simple sufficient statistic $s(\mathbf{R}) = \mathbf{R}^\top \mathbf{u}$ where $\mathbf{u} := \mathbf{w}/\|\mathbf{w}\|$. The detection region is thus a half-space delimited by the hyper-plane $\mathbf{R}^\top \mathbf{u} = \tau_n$. This simple detector achieves the characteristic function (8.1).

The main problem is that the threshold $\tau_n = \sqrt{N}\Phi^{-1}(1 - \mathbb{P}_{\mathsf{fp}})$ to meet a prescribed probability of false positive depends on N usually unknown in practice. In the same way, for a targeted $E_{\mathsf{fp}} = E$, threshold τ_n must scale as $\sqrt{2NEn}$. The obliviousness to N prevents the use of the optimal Neyman-Pearson detector. Chapter 11 deals with other schemes sharing the same drawback.

8.2 Hypercone detector

The only way to become invariant to N is by designing a score function which is independent of the norm of $\mathbf{R}$. In other words, the detector is now based on the assumption that $\mathbf{R}$ has an isotropic distribution. This draws a detection region which is a linear cone with its apex at the origin $\mathbf{0}_n$ and surrounding vector $\mathbf{w} = \sqrt{nP}\mathbf{u}$ with $\|\mathbf{u}\| = 1$ (see Fig 8.1). Denote by Ω its solid angle (*i.e.* the area of the intersection of this linear cone with the unit hypersphere $\mathcal{S}_n$ in this n-dimensional space) and by Ω_0 the solid angle of $\mathcal{S}_n$. The distribution under $\mathcal{H}_0$ is isotropic so that:

$$\mathbb{P}_{\mathsf{fp}} = \frac{\Omega}{\Omega_0} \leq 1. \tag{8.4}$$

Denote by $Q(\Omega)$ the probability that the point $\mathbf{w}$ is being carried outside the cone due to the noise $\mathbf{N}$. In other words, $\mathbb{P}_{\mathsf{fn}} = Q(\Omega)$ (we omit to write the dependance on N and P). Shannon showed that for a given solid angle Ω, the linear cone minimizing $Q(\Omega)$ is indeed the right *circular* hypercone $\mathcal{C}$ whose axis is carried by $\mathbf{u}$ [151, Sect. III] (see Fig 8.1:right):

$$\mathcal{C} = \left\{ \mathbf{r} \in \mathbb{R}^n : \frac{\mathbf{r}^\top \mathbf{u}}{\|\mathbf{r}\|} > \cos\theta \right\}. \tag{8.5}$$

This justifies a long tradition in the history of digital watermarking. Since the seminal paper of I. Cox *et al.* [104], normalized correlation $\mathbf{r}^\top \mathbf{u}/\|\mathbf{r}\|$ has been used in a vast majority of papers until side-information schemes were introduced [105, 96]. The argument of the seminal paper [104] was purely image processing oriented: normalizing the correlation is a way to be

robust to contrast enhancement. Then some signal processing arguments defended this option [86, Chap. 6, p. 237]: Decompose $\mathbf{R}$ as $(\mathbf{R}^\top\mathbf{u})\mathbf{u} + \mathbf{R}^\perp$ where $\mathbf{R}^\perp$ is the Euclidean projection of $\mathbf{R}$ onto the subspace orthogonal to $\mathbf{u}$. Under hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, this projection has the same distribution $\mathcal{N}(\mathbf{0}_{n-1}, N\mathbf{I}_{n-1})$. Variance N is then estimated by $\|\mathbf{R}^\perp\|^2/_{n-1}$ and used for comparing $\mathbf{R}^\top\mathbf{u}$ to the threshold τ_n of the previous Sect. 8.1. This indeed amounts to compare the ratio $\mathbf{R}^\top\mathbf{u}/\|\mathbf{R}^\perp\|$ to a threshold, say $1/\tan(\theta)$, or equivalently, to compare $s(\mathbf{R}) = \mathbf{R}^\top\mathbf{u}/\|\mathbf{R}\|$ to $\cos(\theta)$ like in the definition of the circular hypercone (8.5).

The half-angle θ sets the false positive probability (8.4) by the following closed form:

$$\mathbb{P}_{\mathsf{fp}} = \frac{1}{2}\left(1 - I_{\cos^2\theta}(1/2; (n-1)/2)\right), \tag{8.6}$$

where $I_x(a;b)$ is the incomplete regularized beta function. As promised, fixing the threshold $\tau_n = \cos\theta$ by inverting the equation above doesn't require any parameter of the model except the dimension n of the space. As $n \to +\infty$, $\mathbb{P}_{\mathsf{fp}}$ has the following asymptotic expression [151, Eq. (27)]:

$$\mathbb{P}_{\mathsf{fp}} = \frac{\Gamma(n/2+1)(\sin\theta)^{n-1}}{n\Gamma((n+1)/2)\sqrt{\pi}\cos\theta}\left(1 + O(1/n)\right), \tag{8.7}$$

so that the error exponent is simply:

$$E_{\mathsf{fp}} = -\log\sin\theta. \tag{8.8}$$

The probability of false negative $Q(\Omega)$ is related to the non-central F-distribution. Shannon gives the following asymptotic expression [151, Eq. (51)]:

$$\mathbb{P}_{\mathsf{fn}} \approx \frac{1}{\sqrt{n\pi}}\frac{1}{\sqrt{1+G^2}\sin\theta}\frac{\left(G\sin(\theta)e^{-A^2/2 + AG/2\cos\theta}\right)^n}{AG\sin^2\theta - \cos\theta}, \tag{8.9}$$

with

$$A = \sqrt{\frac{P}{N}}, \quad \tan(\theta_0) = A^{-1}, \quad \text{and } G = \frac{1}{2}\left(A\cos\theta + \sqrt{A^2\cos^2\theta + 4}\right). \tag{8.10}$$

The error exponent is thus:

$$E_{\mathsf{fn}} = \begin{cases} 0, & \text{if } 0 \le \theta \le \theta_0 \\ \frac{A^2}{2} - \frac{A}{2}G\cos\theta - \log(G\sin\theta) & \text{if } \theta_0 < \theta \le \pi/2. \end{cases} \tag{8.11}$$

The appendix 16.1.1 gives a different proof of this result than that of [151] based on the Laplace method. It gives the reader insights because this method is our main tool to prove new results in the sequel.

Equations (8.8) and (8.11) give a parameterization of the characteristic function. As θ decreases from $\pi/2$ to θ_0, it starts from the left endpoint $(0, E^L_{\mathsf{fn}})$ down to the right endpoint $(E^R_{\mathsf{fp}}, 0)$ with

$$E^L_{\mathsf{fn}} = \frac{P}{2N}, \tag{8.12}$$

$$E^R_{\mathsf{fp}} = \frac{1}{2}\log\left(1 + \frac{P}{N}\right). \tag{8.13}$$

Operating at $E_{\mathsf{fp}} = E$ amounts to fix the angle θ thanks to (8.8). This provides robustness up to the noise powers:

$$\text{Case 1: } \bar{\sigma_Z}^2 = \left|\frac{P}{e^{2E}-1} - \sigma_X^2\right|_+, \quad \text{Case 3: } \bar{\sigma_Z}^2 = \frac{P}{e^{2E}-1}. \tag{8.14}$$

These figures of merit are lower than their counter-parts (8.3), although the differences vanish as $E \to 0$. In this sense, the hypercone detector is almost as robust as the optimal Neyman-Pearson detector at very low E_{fp}. This in turn implies working with longer vectors.

Note that E_{fp}^R is the information capacity C in nats of the Gaussian channel under a power constraint P. This is not a surprise. The half angle θ_0 defines the thinest cone for which the noise pushes the transmitted signal $\mathbf{w}$ outside with an exponentially vanishing probability. This probability represents the decoding error probability in a communication scenario. From the definition of the error exponent and (8.4), we see that E_{fp}^R equals, in logarithmic scale and per dimension, the number of hypercones with half angle θ_0 needed to fill the full hypersphere $\mathcal{S}_n$, hence the maximum number of messages (in logarithmic scale and per dimension) which can be reliably (*i.e.* with exponentially vanishing error probability) transmitted over this channel.

In this scheme, the embedder is oblivious to N and the detector is oblivious to N and P. This does not impact the left endpoint of the characteristic (compared to (8.2)), but it lowers the right endpoint which is of utmost importance in watermarking. This is especially true for big Signal to Noise power Ratio P/N, which may happen in Case 3. Yet, by operating at a very low false positive error exponent, this scheme is almost as robust as the Neyman-Pearson detector.

These results are summarized in Fig. 8.3, 8.4, and 8.5 at the end of the chapter.

8.3 Detection thanks to a communication scheme

The fact that E_{fp}^R equals the information capacity mitigates the introduction of the previous chapter (see Sect. 7.1), which made a clear cut between detection and decoding. Indeed, a communication scheme can be turned into a zero-bit watermarking scheme. From a set of M messages which can be reliably transmitted through a channel and a block code of length n, we select at random one reference message, say m_0. The embedding amounts to emit this message while the detector sets $d = 1$ if the decoded message $\hat{m} = m_0$, and $d = 0$ otherwise.

Under $\mathcal{H}_0$, the decoder receives only noise and outputs a random message. Assuming *equiprobability* of the decoded messages, $\mathbb{P}_{\mathsf{fp}} = 1/M$. The rate of communication is defined as $R := \log(M)/n$. This corresponds in our context to E_{fp}.

Under $\mathcal{H}_1$, the message m_0 is transmitted and $\mathbb{P}_{\mathsf{fn}} = \mathbb{P}(\hat{m} \neq m_0)$. The reliability function $E(R)$ is defined by Shannon as the exponential decay rate of the probability of decoding error $\mathbb{P}(\hat{m} \neq m_0)$ [151, Eq. (2)], which in our case corresponds to E_{fn}. When we derive a detection scheme from a communication code, the characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$ equals Shannon's reliability function $(R, E(R))$.

The seminal work of Shannon [150] shows that $E(R) = 0$ for $R \geq C := 1/2 \log(1 + A^2)$ the capacity of the AWGN channel, for any communication scheme. Gallager [115] gives a lower bound of the reliability function of the optimal communication code:

$$E(R) \geq \begin{cases} \frac{A^2}{4}\left(1 - \sqrt{1 - e^{-2R}}\right), & 0 \leq R \leq R_1 \\ \frac{1}{2}\left(1 + \frac{A^2}{2} - \sqrt{1 + \frac{A^4}{4}}\right) + \frac{1}{2}\log\left(\frac{1}{2}\left(\sqrt{1 + \frac{A^4}{4}} + 1\right)\right) - R, & R_1 \leq R \leq R_2 \\ \frac{A^2}{4}\left(1 + e^{-2R} - (1 - e^{-2R})\sqrt{1 + \frac{4}{A^2(1 - e^{-2R})}}\right) & \\ + \frac{1}{2}\log\left(1 - \frac{A^2}{2}(1 - e^{-2R})\left(\sqrt{1 + \frac{4}{A^2(1 - e^{-2R})}} - 1\right)\right) + R, & R_2 \leq R \leq C \end{cases} \tag{8.15}$$

with R_1 and R_2 the following critical rates:

$$R_1 = \frac{1}{2}\log\left(\frac{1}{2}+\frac{1}{2}\sqrt{1+\frac{A^2}{4}}\right), \tag{8.16}$$

$$R_2 = \frac{1}{2}\log\left(\frac{1}{2}+\frac{A^2}{4}+\frac{1}{2}\sqrt{1+\frac{A^4}{4}}\right). \tag{8.17}$$

This is 'just' a lower bound of the reliability function of the optimal code, but achieving such a characteristic is not trivial.

8.3.1 Example: Random codes over the hypersphere

Shannon [151] proposes to randomly draw $M = e^{nR}$ vectors on the sphere $\mathcal{S}_n$ of unity radius. This codebook is shared with the decoder. The encoder emits the vector related to the message to be transmitted scaled by $\sqrt{nP}$. The decoder outputs the message whose codeword is the closest to the received signal (in Euclidean distance). This is the Maximum Likelihood decoder when the noise $\mathbf{N}$ is Gaussian distributed. Since the codewords share the same module $\sqrt{nP}$, the closest codeword is also the more correlated with the received signal. This means that the decoder is oblivious to P and N, while the emitter is oblivious to N.

We have the following properties:

- Eq. (8.11) is an upper bound of $E(R)$ for any instance of a code following this construction and for $0 \leq R \leq C$. Equality would imply that we are able to place M points on the sphere $\mathcal{S}_n$ such that each of their decoding regions (in the sense of Euclidean nearest neighbors) is the 'ideal' circular cone of solid angle Ω_0/M.
- Eq. (8.11) is also a lower bound of $E(R)$ for the best code given by such construction and for $R_2 < R < C$. It is the exponent of the average error probability over the code ensemble, so that the best code has an $E(R)$ bigger than this exponent.

Therefore, for $R_2 < R < C$, Eq. (8.11) is the expression of $E(R)$ for the best code. Indeed, after some rewriting, this equals the third expression of (8.15). See [151] for instance for bounds when $0 \leq R < R_2$.

This communication scheme used as a zero-bit watermarking scheme and the hypercone detector scheme of Sect. 8.2 have indeed the same characteristic in the range $E_{\mathsf{fp}} \in [R_2, C]$. There is a superb geometrical interpretation of this surprising fact in paper [162] based on Gallager's bounding technique. A crude handwaving justification is that when $R < R_2$, the Maximum Likelihood decoding region related to message m_0 is shaped with fewer hyperplanes (due to the codewords surrounding $\mathbf{v}_{m_0}$ - see Fig 8.1) and gets significantly different than the circular hypercone sharing the same solid angle. The characteristic of communication scheme is significantly lower than the one of the hypercone detector as R gets lower than R_2.

The communication scheme has moreover a much higher complexity than the hypercone detector scheme due to its exhaustive decoding. Yet, it will show some interest later on when considering side-informed embedding.

8.4 Voronoï modulation

To reduce the complexity, we now consider a structured codebook like the '*Voronoï modulation*' [166, Sect. 9.1]. This scheme is well known in multi-bit watermarking especially from the

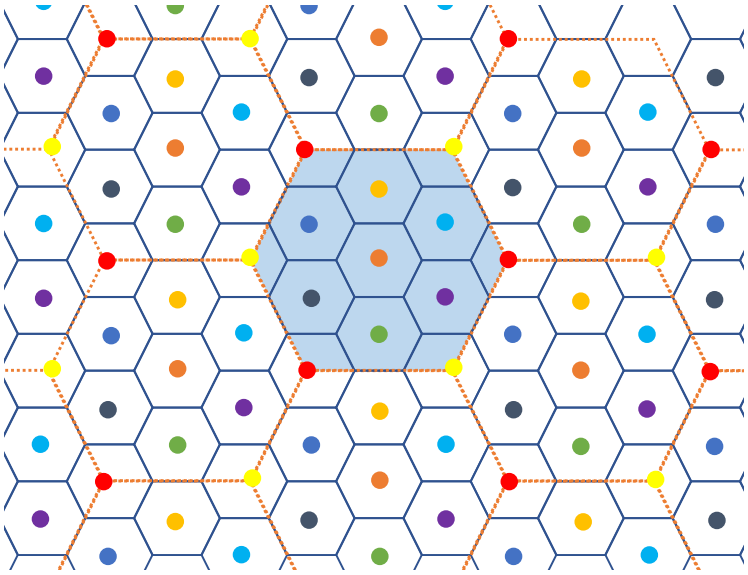


Figure 8.2: The coarse lattice Λ_2 is the set of orange points, with Voronoï cells in dashed orange line, the fine lattice Λ_1 is the set of all the colored points (Voronoï cells in dark blue line). Λ_1 is the union of 9 shifted version of Λ_2 (one per color). Embedding (8.18) ensures that $\mathbf{W}$ lies into a Voronoï central cell of Λ_2 in light blue.

theoretical point of view [96, 135]. Its implementation in real technique is more challenging [94].

It needs two nested lattices in $\mathbb{R}^n$. The fine Lattice Λ_1 is used for coding the message: Some of its points play the role of the codewords. Lattice Λ_1 is a good channel code in the sense that its points are dense (high rate) and robust to noise (this property is also known as Poltyrev good lattice). The coarse lattice $\Lambda_2 \subset \Lambda_1$ shapes the transmitted signal.It is a good quantizer in the sense that the volume of its Voronoï cell $\mathcal{V}(\Lambda_2)$ is big for a constrained second moment. A stronger requirement is that the maximum amplitude of the transmitted signal approaches its average (this property is also known as Rogers good lattice).

The scheme

The points $\{\mathbf{v}_m\}_{m=1}^M$ of Λ_1 inside the Voronoï cell $\mathcal{V}(\Lambda_2)$ compose the codebook, each associated to one message. They are called coset representatives in the sense that $\Lambda_1 = \bigcup_{m=1}^M (\mathbf{v}_m + \Lambda_2)$ (see Fig. 8.2). We have $M = e^{nR}$ messages with $R = {}^1\!/_n \log {}^{|\mathcal{V}(\Lambda_2)|}\!/_{|\mathcal{V}(\Lambda_1)|}$ (where $|\mathcal{A}|$ is the volume of region $\mathcal{A}$). For transmitting message m, the encoder emits the following signal:

$$\mathbf{W} = (\mathbf{v}_m + \mathbf{U}) \mod \Lambda_2, \tag{8.18}$$

where $\mathbf{U} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$ is the dither signal and $\mathbf{x} \mod \Lambda := \mathbf{x} - Q_\Lambda(\mathbf{x})$ with $Q_\Lambda(\mathbf{x})$ the Euclidean quantizer of $\mathbf{x}$ onto Λ, $Q_\Lambda(\mathbf{x}) := \arg\min_{\mathbf{v}\in\Lambda} \|\mathbf{x} - \mathbf{v}\|$. The goal of the dither is to ensure that $\mathbf{W} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$ for any message m [166, Prop. 9.1.1], whence the shaping role of Λ_2. This implies that $\mathbb{E}[\|\mathbf{W}\|^2] = n\sigma^2(\Lambda_2)$, the second moment of the coarse lattice:

$$\sigma^2(\Lambda_2) := \frac{1}{n|\mathcal{V}(\Lambda_2)|} \int_{\mathcal{V}(\Lambda_2)} \|\mathbf{x}\|^2 d\mathbf{x}. \tag{8.19}$$

The watermark signal has also a bounded amplitude: $\forall \mathbf{X} \in \mathbb{R}^n$, $\forall \mathbf{U} \in \mathcal{V}(\Lambda_2)$, $\forall m \in \{1, \ldots, M\}$:

$$\|\mathbf{W}\|^2 \leq r_{\mathsf{cov}}^2(\Lambda_2), \tag{8.20}$$

where $r_{\mathsf{cov}}(\Lambda_2)$ is the covering radius of lattice Λ_2: $r_{\mathsf{cov}}(\Lambda_2) := \max_{\mathbf{x}\in\mathcal{V}(\Lambda_2)} \|\mathbf{x}\|$.

As for the decoding, there are many options (see [166, Sect. 9.1.2]) and we restrict ourselves to the lattice decoder with linear estimation [166, Sect. 9.1.3]. When receiving $\mathbf{R}$, this decoder first estimates the transmitted signal $\mathbf{W}$ by $\hat{\mathbf{W}} = \alpha\mathbf{R}$. We can gauge the quality of this estimation by the mean square error $\mathsf{MSE}(\alpha) := n^{-1}\mathbb{E}[\|\hat{\mathbf{W}} - \mathbf{W}\|^2]$. Note that $\hat{\mathbf{W}} - \mathbf{W} = \alpha\mathbf{N} + (\alpha-1)\mathbf{W}$ so that, $\mathbf{N}$ and $\mathbf{W}$ being independent, $\mathsf{MSE}(\alpha) = \alpha^2 N + (\alpha-1)^2 P$. This quantity is minimized for $\alpha = \alpha_{\mathsf{MMSE}}$ with

$$\alpha_{\mathsf{MMSE}} := {}^{P}\!/_{(P+N)}, \tag{8.21}$$

achieving $\mathsf{MSE}(\alpha_{\mathsf{MMSE}}) = {}^{PN}\!/_{(P+N)}$. This is the so-called Wiener filtering.

Then, the decoder computes the decision vector $\tilde{\mathbf{Y}} = (\hat{\mathbf{W}} - \mathbf{U}) \mod \Lambda_2$ and finally outputs:

$$\mathbf{v}_{\hat{m}} = Q_{\Lambda_1}(\tilde{\mathbf{Y}}) \mod \Lambda_2. \tag{8.22}$$

It happens that $\tilde{\mathbf{Y}} = (\mathbf{v}_m + \mathbf{N}_{\mathsf{eq}}) \mod \Lambda_2$ with $\mathbf{N}_{\mathsf{eq}} = (\alpha\mathbf{N} + (\alpha-1)\mathbf{W}) \mod \Lambda_2$, where $\mathbf{W} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$ is independent of $\mathbf{v}_m$ (thanks to the dither) and of $\mathbf{N}$. This relation between $\mathbf{v}_m$ and $\tilde{\mathbf{Y}}$ is called the *equivalent modulo* Λ *channel* [166, Sect. 9.5].

The reliability function

The expression of the reliability function is known under some conditions [131]. First, we need asymptotically good pairs of nested lattices: As $n \to +\infty$, the quantity $G(\Lambda_2)\mu(\Lambda_1, P_e) \to 1^+$, where $G(\Lambda_2)$ is the normalized second moment [166, Def. 3.2.2] of the coarse lattice and $\mu(\Lambda_1, P_e)$ the normalized volume to noise ratio of the fine lattice [166, Def. 3.3.3]. Theorem [166, Th. 8.5.1] shows such good pairs exist. Then, if $\alpha = \alpha_{\mathsf{MMSE}}$, this scheme is capacity achieving [166, Th. 9.6.2]. This translates in zero-bit watermarking as $E_{\mathsf{fp}}^R = {}^1\!/_2 \log(1 + {}^P\!/_N)$.

There is an even better result due to [131]. For such good pairs of nested lattices, for Λ_2 being Rogers good (asymptotically $n \to \infty$, ${}^{r_{\mathsf{cov}}(\Lambda_2)}\!/_{r_{\mathsf{eff}}(\Lambda_2)} \to 1$, and for a *fixed* parameter α, the reliability function is given by [166, Th. 13.8.3]: $\forall R, 0 \le R \le {}^1\!/_2 \log({}^P\!/_{\mathrm{MSE}(\alpha)})$

$$E(R) = E_{\Lambda_1/\Lambda_2}\left(\frac{Pe^{-2R}}{N\alpha^2}, \frac{(1-\alpha)^2 P}{\alpha^2 N}\right), \text{ with} \tag{8.23}$$

$$E_{\Lambda_1/\Lambda_2}(x,y) = \frac{1}{2}\left(x + y - \sqrt{1+4xy} + \log\frac{1+\sqrt{1+4xy}}{2x}\right). \tag{8.24}$$

For rates $R > {}^1\!/_2 \log({}^P\!/_{\mathsf{MSE}(\alpha)})$, $E(R) = 0$. Thus, this setting is a priori not capacity achieving (unless P and N are s.t. $\alpha_{\mathsf{MMSE}} = \alpha$).

The error exponent is maximized for high rate R (*i.e.* $R_2 \le R \le C$) by setting $\alpha = \alpha_{\mathsf{opt}}$:

$$\alpha_{\mathsf{opt}} := -\frac{\gamma}{2} + \sqrt{\frac{\gamma^2}{4} + \gamma} \quad \text{with } \gamma := A(1 - e^{-2R}). \tag{8.25}$$

In this case,

$$E(R) = \frac{1}{2}\left(\frac{Pe^{-2R}}{N} - \alpha_{\mathsf{opt}} + \log((1-\alpha_{\mathsf{opt}})) + 2R\right). \tag{8.26}$$

Note that α_{opt} is not equal to α_{MMSE}. Parameter α_{opt} is the right choice to lighten the distribution tail of the mixture $\alpha\mathbf{N}+(1-\alpha)\mathbf{W}$ where $\mathbf{N}$ is a Gaussian noise and $\mathbf{W}$ is uniformly distributed over $\mathcal{V}(\Lambda_2)$. By lightening the tail distribution, we concentrate this mixture inside $\mathcal{V}(\Lambda_1)$ to minimize the decoding error probability. On the other hand, α_{MMSE} is the optimal value to minimize the variance of the mixture, which is not the same objective. Yet, as $R \to C$, $\alpha_{\mathsf{opt}} \to \alpha_{\mathsf{MMSE}}$. With

a handwaving argumentation: when R is close to C, there are so many neighboring codewords around $\mathbf{0}_n$, each delimiting $\mathcal{V}(\Lambda_1)$ by an hyperplane, that this Voronoï cell asymptotically becomes an hyperball. On the other hand, asymptotically as $n \to +\infty$, the uniform distribution over $\mathcal{V}(\Lambda_2)$ tends to a Gaussian distribution. The mixture then becomes Gaussian distributed and concentration in a hyperball boils down to variance reduction. Last comment: this communication scheme with $\alpha = \alpha_{\mathsf{opt}}$ is optimal from the reliability point of the view: Eq. (8.26) indeed equals the third line of (8.15).

Difficulties in practice

There are several difficulties for applying this scheme to watermarking in practice:

- **The necessity of the dither**. The decoder cannot work without $\mathbf{U}$, and the randomness of $\mathbf{U}$ is key to meet the embedding constraint on expectation (7.2). This will be an issue in some applications (see Sect. 7.4). This limitation is relaxed under the strict embedding constraint (7.1) because $\|\mathbf{W}\|^2 \leq r^2_{\mathsf{cov}}(\Lambda)$ for any fixed $\mathbf{U}$. The role of $\mathbf{U}$ is also to equalize the error probability over all codewords. Without dither, some codewords are more prone to decoding error especially at low Signal to Noise Ratio [166, Sect. 9.9].

- **Not oblivious to** P. The shaping lattice Λ_2 ensures that $\mathbb{E}[\|\mathbf{W}\|^2] = n\sigma^2(\Lambda_2)$ and $\|\mathbf{W}\|^2 \leq r^2_{\mathsf{cov}}(\Lambda_2)$ which must be lower than nP (depending on the embedding constraints – both of them leading to the same error exponent). Sect. 7.4 outlines that in some watermarking applications, P is varying from one host to another. Lattice Λ_2 must then be scaled appropriately and so is Λ_1 since they are nested. This is a problem because the decoder needs both lattices.

- **Not oblivious to** N. To achieve the best characteristic function, the embedder and the decoder needs P and N to compute α_{opt} (8.25). Even if we only consider its right endpoint, the scheme is optimal when the embedder and the decoder know P and N to compute α_{MMSE} (8.21). Obliviousness imposes to arbitrarily fix α. Then,

$$E^R_{\mathsf{fp}} = \frac{1}{2}\log\left(\frac{P}{\alpha^2 N + (1-\alpha)^2 P}\right) \leq \frac{1}{2}\log\left(1 + \frac{P}{N}\right) \leq \frac{P}{2N}. \tag{8.27}$$

The last comment raises the issue of setting parameter α. Suppose that we wish to operate at false positive error exponent $E_{\mathsf{fp}} = E > 0$. We would like $E^{(R)}_{\mathsf{fp}}$ to be greater or equal to E on a large range of noise power. Here, the inequality

$$E^{(R)}_{\mathsf{fp}} = \left|\frac{1}{2}\log\left(\frac{P}{\alpha^2 N + (1-\alpha)^2 P}\right)\right|_+ \geq E \tag{8.28}$$

implies that the noise power ratio is weak enough:

$$N \leq P\left|\frac{e^{-2E} - (1-\alpha)^2}{\alpha^2}\right|_+. \tag{8.29}$$

This upper limit is maximized for $\alpha^\star = 1 - e^{-2E}$, which provides robustness (in the sense that $E_{\mathsf{fn}} > 0$ at $E_{\mathsf{fp}} = E$) whenever $N < \bar{N}$ with

$$\bar{N} := \frac{P}{e^{2E} - 1}. \tag{8.30}$$

This is a decreasing function of E. There is no surprise: a big E is an ambitious target for E_{fp} whose benefit is a small n to fulfill the requirement on $\mathbb{P}_{\mathsf{fp}}$, but it yields less robustness. This is yet another illustration of the well known rule of thumb: spread the watermark to gain robustness.

Yet, what is indeed a big surprise is that this critical noise power value is indeed optimal. Knowing N and P enables to set α to α_{opt} or α_{MMSE} which in turn makes E_{fp}^R equalling the channel capacity C. This quantity is bigger than E for $N < {}^{P}/_{(e^{2E}-1)}$.

Aiming at an operating point at $E_{\mathsf{fp}} = E$ not only provides a rationale for setting α, but also shows that maximum robustness can be achieved with an embedder oblivious to N and a decoder oblivious to P and N.

8.5 Conclusion

We draw the following conclusions about zero-bit watermarking with one unique source of noise illustrated in Figures 8.3, 8.4, and 8.5.

When N is known at the detection side, the best scheme is the Optimal Neyman-Pearson detector of Sect. 8.1. The optimal performances are:

$$\begin{aligned} E_{\mathsf{fn}} &= \left(\left| \sqrt{\frac{P}{2N}} - \sqrt{E_{\mathsf{fp}}} \right|_+ \right)^2, \\ E_{\mathsf{fn}}^L &= E_{\mathsf{fp}}^R = \frac{P}{2N}, \quad \bar{N} = \frac{P}{2E} \end{aligned}$$

Communication schemes perform worse: Detection and communication are not the same problem.

When N is unknown at the detection side, the best scheme is the Hypercone detector of Sect. 8.2. The optimal performances are:

$$\begin{aligned} (E_{\mathsf{fp}}, E_{\mathsf{fn}}) &= \left(-\log \sin\theta, \frac{A^2}{2} - \frac{A}{2} G \cos\theta - \log(G \sin\theta) \right), \\ E_{\mathsf{fn}}^L &= \frac{P}{2N}, \quad E_{\mathsf{fn}}^R = \frac{1}{2} \log\left(1 + \frac{P}{N}\right), \quad \bar{N} = \frac{P}{e^{2E} - 1}. \end{aligned}$$

Detection and communication perform equally well only at high E_{fp}. Yet, a watermarking scheme usually handles long vectors in order to provide robustness. It typically operates at low $E_{\mathsf{fp}} = E$ so that the loss in robustness is not substantial: Since $\bar{N} = {}^{P}/_{2E}(1 - E + O(E^2))$, the obliviousness w.r.t. N gives a loss of ${}^{P}/_{2}$ in terms of $\bar{N}$ as $E \to 0$.

Not knowing N at the detection side yields a lower characteristic. Obliviousness does not prevent the detector from reaching the highest value of E_{fn} on the left endpoint, however it precludes reaching operating points with large E_{fp} on the right hand side.

In the case of zero-bit watermarking without side-information (case 1 in Sect. 7.5), $N = \sigma_X^2 + \sigma_Z^2$ and P is usually small compared to N so that knowing or not knowing N at the detection side is not a big deal. In the case of non-blind detection (case 3 in Sect. 7.5), $N = \sigma_Z^2$, which might be small, and then the difference might be large. Once again, the only remedy seems to operate at low $E_{\mathsf{fp}} = E$ but this implies more complexity as the vectors to handle are longer.

The use of Voronoï modulation for zero-bit watermarking (with a non side-informed embedder) is mostly challenged by the need of the random dither and the fact that the decoder is not oblivious to P as said in Sect. 8.4. The detection needs P and N to reach the characteristic or E_{fp}^R of the oblivious and simpler hypercone detector. However, when only looking at robustness at a given $E_{\mathsf{fp}} = E$, the optimal setting of α does not depend on P and N.

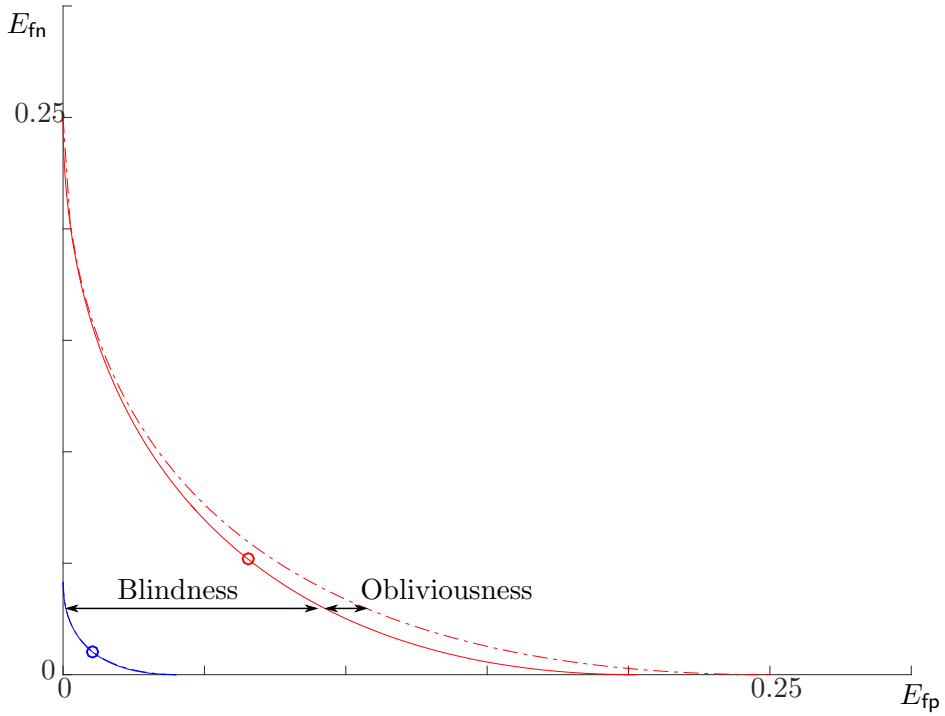


Figure 8.3: Characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$ for $P = 0.1$, $\sigma_X^2 = 1$, and $\sigma_Z^2 = 0.2$. Non side-informed embedder (Case 1, $N = \sigma_X^2 + \sigma_Z^2$). Non blind detector (Case 3, $N = \sigma_Z^2$). Neyman-Pearson optimal detector (8.1) in dashed line. Hypercone detector (8.11) in plain line. The difference is not visible in Case 1. Circles show the critical rates R_2 (8.17). Communication schemes (Sect. 8.3.1 and 8.4) achieve the same performances for $E_{\mathsf{fp}} > R_2$, otherwise they perform worse.

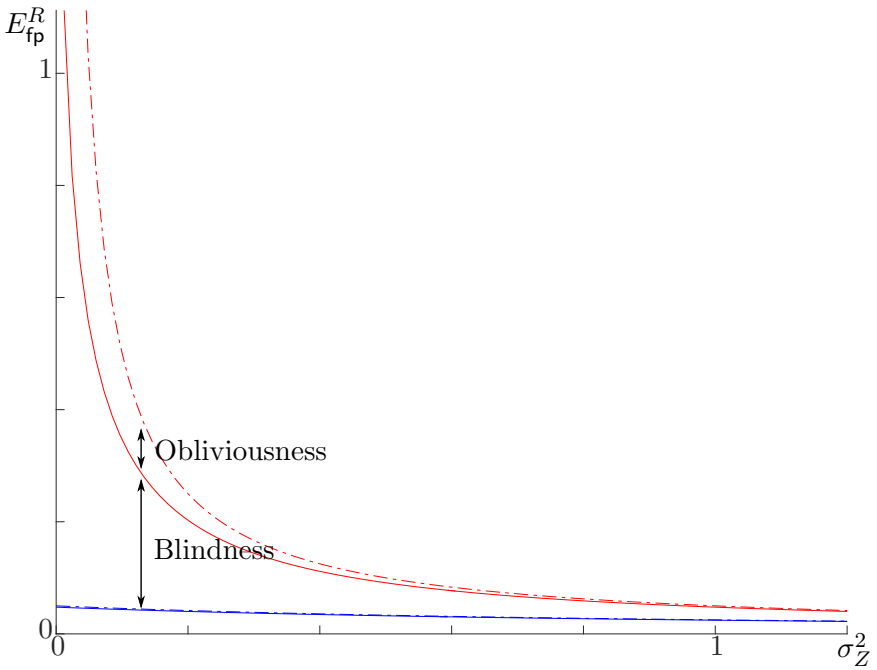


Figure 8.4: E_{fp}^R as a function of σ_Z^2 for $P = 0.1$ and $\sigma_X^2 = 1$. Non side-informed embedder (Case 1, $N = \sigma_X^2 + \sigma_Z^2$). Non blind detector (Case 3, $N = \sigma_Z^2$). Neyman-Pearson optimal detector (8.2) in dashed line. Hypercone detector (8.12) in plain line.

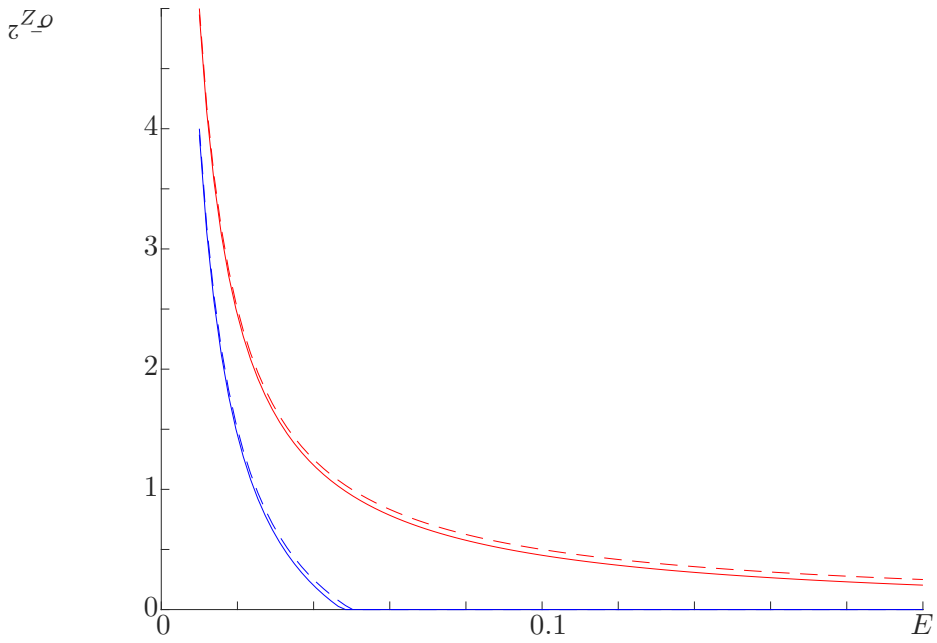


Figure 8.5: $\bar{\sigma_Z}^2$ as a function of E for $P = 0.1$ and $\sigma_X^2 = 1$. Non side-informed embedder (Case 1, $\bar{N} = \sigma_X^2 + \bar{\sigma_Z}^2$). Non blind detector (Case 3, *i.e.* $\bar{N} = \bar{\sigma_Z}^2$). Neyman-Pearson optimal detector (8.3) in dashed line. Hypercone detector (8.14) in plain line.

Chapter 9

One source of side information and no noise

In this section, we elaborate on the model:

- $\mathcal{H}_0$: $\mathbf{R} = \mathbf{X}$
- $\mathcal{H}_1$: $\mathbf{R} = \mathbf{X} + \mathbf{W}$

At first sight, there is no difference with the last chapter: we just have to transpose its results with $\mathbf{X}$ being the single source of noise: $N = \sigma_X^2$ (in this chapter, we still use Signal to Noise power Ratio $A = \sqrt{P/N} = \sqrt{P/\sigma_X^2}$). What is changing if $\mathbf{X}$ is not a source of noise but a side-information? In other words, $\mathbf{X}$ is an information at the disposal of the embedder in order to create a better $\mathbf{W} = \mathbf{w}(\mathbf{X})$ suitable for that $\mathbf{X}$.

This section assumes that there is no extra noise between the embedder and the detector. This is called the *noiseless scenario*. The false negative has then a particular meaning: the receiver fails to detect the presence of the watermark when the embedder has failed to watermark $\mathbf{X}$: it had not enough power P to push $\mathbf{X}$ into the detection region.

This chapter gives the characteristic $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$ of several schemes, and then deduces the figure of merit E_{fp}^R. Fig. 9.4 and 9.5 summarize the results at the end of the chapter.

9.1 Geometrical interpretation

In the sequel and *when the embedding constraint is strict* (7.1), we denote by $\mathcal{E}$ the embedding region, the subset of $\mathbb{R}^n$ where vectors can be watermarked, *i.e.* vectors which are at most $\sqrt{nP}$ away from the detection region $\mathcal{W}$ (7.3):

$$\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^n | \exists \mathbf{r} \in \mathcal{W} : \quad \|\mathbf{r} - \mathbf{x}\| \leq \sqrt{nP}\}. \tag{9.1}$$

Region $\mathcal{E}$ is thus the filtering of $\mathcal{W}$ by a ball of radius $\sqrt{nP}$, a.k.a. the rolling ball technique [147]: By rolling a ball of radius $\sqrt{nP}$ over the boundary of $\mathcal{W}$, the center of that ball draws the boundary of region $\mathcal{E}$. The error probabilities are then related to the measures of these sets by the distribution of $\mathbf{X}$:

$$\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(\mathbf{X} \in \mathcal{W}), \tag{9.2}$$

$$\mathbb{P}_{\mathsf{fn}} = 1 - \mathbb{P}(\mathbf{X} \in \mathcal{E}). \tag{9.3}$$

Under the noiseless scenario, we are thus looking for a region $\mathcal{W}$ s.t. $\mathbb{P}(\mathbf{X} \in \mathcal{W}) = \mathbb{P}_{\mathsf{fp}}$ and which, once filtered by a rolling ball, gives the lowest $\mathbb{P}_{\mathsf{fn}}$, *i.e.* the biggest probability $\mathbb{P}(\mathbf{X} \in \mathcal{E})$. This is an elegant way to conceive side-informed watermarking under the noiseless scenario and the strict embedding constraint because there is no need to specify anything about the embedding side (*i.e.* how $\mathbf{W} = w(\mathbf{X})$ is not specified).

I don't know the solution of this problem, but the Gaussian isoperimetric inequality gives the worse possible region: For any region $\mathcal{W} \subset \mathbb{R}^n$ and $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_n, \sigma_X^2 \mathbf{I}_n)$ s.t. $\mathbb{P}(\mathbf{X} \in \mathcal{W}) = \mathbb{P}_{\mathsf{fp}}$, we have

$$\mathbb{P}(\mathbf{X} \in \mathcal{E}) \geq \Phi\left(\Phi^{-1}(\mathbb{P}_{\mathsf{fp}}) + A\sqrt{n}\right), \tag{9.4}$$

where $\mathcal{E}$ is the filtering of $\mathcal{W}$ by a ball of radius $\sqrt{nP}$ and $A = \sqrt{P/\sigma_X^2}$. Equality happens if and only if $\mathcal{W}$ is a half-space. The Gaussian isoperimetric inequality gives us an upper bound of $\mathbb{P}_{\mathsf{fn}}$, which translates into a lower bound for E_{fn}. This shows that the optimal Neyman-Pearson detector of Sect. 8.1 is indeed the worse choice. In other words, it is another proof that Case 1 gives a lower bound of the performance of a side-informed watermarking scheme (at least in the noiseless scenario).

This chapter investigates other forms of region $\mathcal{E}$ in the asymptotical setup.

9.2 Asymptotically perfect schemes

We start considering a very simple scheme, named ZATT [106, Sect. 3.3], illustrating that a perfect scheme is easily achievable in the noiseless scenario. For a fixed integer $k > 0$, we denote $\mathbf{X}_{(k)} := (X(1), \dots, X(k), 0, \dots, 0)$ and $\mathbf{X}^{(n-k)} = (0, \dots, 0, X(k+1), \dots, X(n))$ so that $\mathbf{X} = \mathbf{X}_{(k)} + \mathbf{X}^{(n-k)}$.

If the embedding constraint is on expectation (7.2), then the embedder sets $\mathbf{W} = -\mathbf{X}_{(k)}$ which is admissible if $k\sigma_X^2 \leq nP$. This is always possible asymptotically as $n \to +\infty$ provided $P > 0$. If the embedding constraint is strict (7.1), the embedder sets $\mathbf{W} = -\mathbf{X}_{(k)}$ whenever $\|\mathbf{X}_{(k)}\|^2 \leq nP$, and $\mathbf{W} = \mathbf{0}_n$ otherwise.

The embedding 'kills' the k first components[1] of $\mathbf{X}$. In other words, it artificially turns the continuous random vector $\mathbf{X}_{(k)}$ into the constant vector $\mathbf{0}_n$. Region $\mathcal{W}$ is reduced to the singleton $\{\mathbf{0}_n\}$. There is no false positive because $\mathbb{P}(\mathbf{X}_{(k)} = \mathbf{0}_n) = 0$ as $\mathbf{X}_{(k)}$ is continuous random vector under $\mathcal{H}_0$.

Under $\mathcal{H}_1$, if the embedding constraint holds on expectation (7.2), asymptotically as $n \to \infty$, watermarking and detection are always successful and there is no false negative. This happens even for a fixed n, provided that it is bigger than $k\sigma_X^2/P$. The scheme is perfect in the sense that $(\mathbb{P}_{\mathsf{fp}}, \mathbb{P}_{\mathsf{fn}}) = (0, 0)$ and $(E_{\mathsf{fp}}, E_{\mathsf{fp}}) = (+\infty, +\infty)$. If the embedding constraint is strict (7.1), then the embedder fails watermarking when $\mathbf{X} \notin \mathcal{E}$, *i.e.* $\|\mathbf{X}_{(k)}\|^2 > nP$. This is the only source of error yielding the error exponent $E_{\mathsf{fn}} = A^2/2$ because $\sigma_X^{-2}\|\mathbf{X}_{(k)}\|^2 \sim \chi_k^2$. The characteristic is then $(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = (+\infty, A^2/2)$.

In this scheme, the detector is oblivious to P and N.

9.3 Voronoï Modulation with side information

ZATT quantizes $\mathbf{X}_{(k)}$ onto a codebook which solely contains the null vector. This is generalized with side-informed approaches of the Voronoï modulation of Sect 8.4. The following subsections

[1] For the sake of security, we pass $\mathbf{X}$ through a pseudo-random rotation matrix.

present this idea with quantization over whole vectors in $\mathbb{R}^n$, but it can also work with part of the vector in $\mathbb{R}^k$ to make the connection with ZATT.

9.3.1 First version: scaled lattice

This first version modifies the embedding (8.18) by setting $\mathbf{v}_m = \mathbf{0}_n$ (this is a simplification without any loss of generality) and by taking into account the side information $\mathbf{X}$ in the following way:

$$\mathbf{W} = (\mathbf{U} - \mathbf{X}) \mod \Lambda_2. \tag{9.5}$$

This last expression can be written as $\mathbf{W} = -(\mathbf{X} \mod \Lambda_2') = Q_{\Lambda_2'}(\mathbf{X}) - \mathbf{X}$ where Λ_2' is the lattice $-\Lambda_2$ (which equals Λ_2) shifted by the translation vector $\mathbf{U}$. This shows that the watermarked vector is simply $\mathbf{Y} = \mathbf{W} + \mathbf{X} = Q_{\Lambda_2'}(\mathbf{X})$. The detector simply consists in verifying that the received vector is exactly a codeword of lattice Λ_2' under $\mathcal{H}_1$:

$$\mathbf{v}_{\hat{m}} = \mathbf{R} \mod \Lambda_2' = (\mathbf{R} - \mathbf{U}) \mod \Lambda_2. \tag{9.6}$$

Under $\mathcal{H}_1$ in the noiseless scenario, we have

$$\begin{aligned} \mathbf{v}_{\hat{m}} &= (\mathbf{X} + (\mathbf{U} - \mathbf{X}) - Q_{\Lambda_2}(\mathbf{U} - \mathbf{X}) - \mathbf{U}) \mod \Lambda_2 & (9.7)\\ &= Q_{\Lambda_2}(\mathbf{U} - \mathbf{X}) \mod \Lambda_2 = \mathbf{0}_n. & (9.8) \end{aligned}$$

Under $\mathcal{H}_0$, $\mathbf{v}_{\hat{m}}$ is a continuous random vector whose probability to be strictly equal to $\mathbf{0}_n$ is null.

The shaping lattice Λ_1 is no longer needed because no message is transmitted as we focus on the single codeword $\mathbf{v}_m = \mathbf{0}_n$. This scheme never produces any false negative if the embedding is always successful. This is granted under the following requirements:

- If the embedding constraint is on expectation (7.2), we have $\mathbb{E}[\|\mathbf{W}\|^2] = n\sigma^2(\Lambda_2)$ so we need $\sigma^2(\Lambda_2) < P$. This holds provided that dither $\mathbf{U}$ is random, distributed as $\mathcal{U}_{\mathcal{V}(\Lambda_2)}$, and independent of $\mathbf{X}$. As discussed in Sect. 7.4 and Sect. 8.4, this raises the issue of the detector knowing $\mathbf{U}$. Yet, contrary to Sect. 8.4, the 'flat host assumption' alleviates this drawback: when $\sigma_X^2 \gg P$, the p.d.f. of the host is smooth w.r.t. the scale of the lattice. In other words, over the Voronoï cell centered on any codeword of Λ_2, this p.d.f. is assumed to be flat so that $\mathbf{X} \mod \Lambda_2 \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$. Then $\mathbb{E}[\|\mathbf{W}\|^2] = n\sigma^2(\Lambda_2)$ even if $\mathbf{U}$ is fixed. The 'flat host assumption' corresponds to the 'high resolution regime' in quantization theory: $\mathbf{W} = Q_{\Lambda_2}(\mathbf{X}) - \mathbf{X}$ is a quantization noise independent of $\mathbf{X}$ and uniformly distributed in such a regime.

- If the embedding constraint is strict (7.1), we have $\|\mathbf{W}\|^2 \leq r_{\mathsf{cov}}^2(\Lambda_2)$, the squared covering radius of lattice Λ_2. In that case, $\mathbf{U}$ can also be fixed.

The infinite codebook Λ_2 gives a perfect scheme under both embedding constraint assumptions. However, the major issue is that Λ_2 is inflated by a scaling factor dependent on P. Therefore, this detector is not oblivious to P.

9.3.2 Second version: fixed lattice

The second version is another adaptation of the Voronoï modulation with side-information where Λ_2 is now fixed. We assume the 'flat host assumption' to get rid off the dither (a dithered version

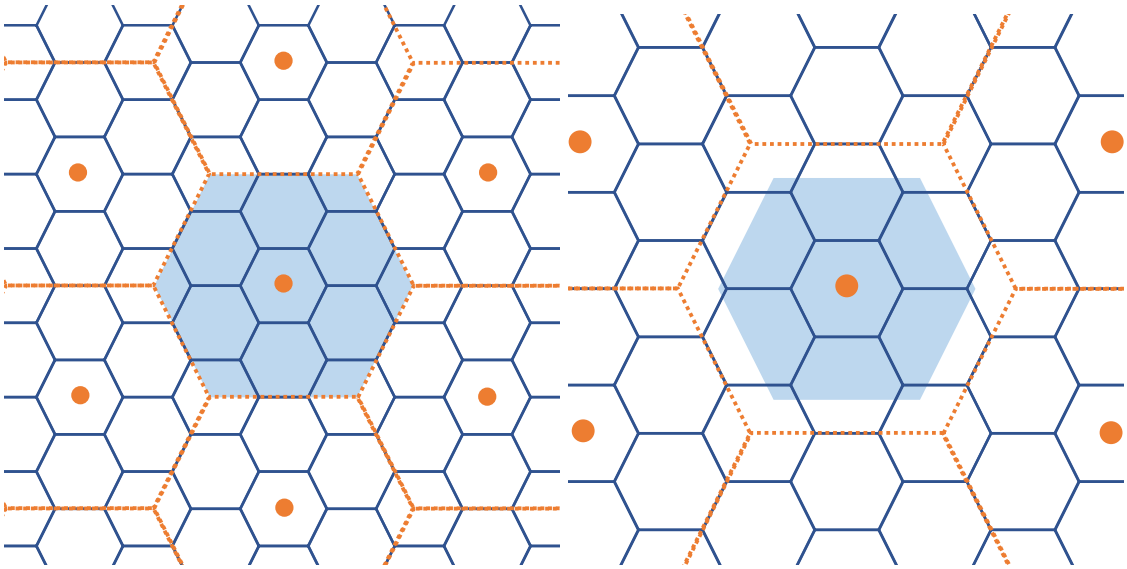


Figure 9.1: Two versions of the Voronoï Modulation. Left: Lattices Λ_1 and Λ_2 have been scaled by a factor ≈ 0.8 to fullfill the embedding distortion constraint. Right: Lattices Λ_1 and Λ_2 are fixed (the same as in Fig. 8.2). $\mathbf{W}$ lies in $\beta\mathcal{V}(\Lambda_2)$ (with $\beta \approx 0.8$ in this example). In both version, the support of $\mathbf{W}$ is the same light blue regions.

straightforwardly replaces Λ_2 by $\Lambda_2' = \mathbf{U} + \Lambda_2$ if the 'flat host assumption' doesn't hold) and we set $\mathbf{v}_m = \mathbf{0}_n$. The embedding (8.18) is changed to:

$$\mathbf{W} = \beta(-\mathbf{X} \mod \Lambda_2), \tag{9.9}$$

with $0 \leq \beta \leq 1$ a scaling factor, which handles varying P, whereas Λ_2 is now a fixed lattice shared with the detector. The embedding constraint (7.2) implies:

$$\beta = \begin{cases} \sqrt{\frac{P}{\sigma^2(\Lambda_2)}} & \text{if } P < \sigma^2(\Lambda_2), \\ 1 & \text{otherwise.} \end{cases} \tag{9.10}$$

The watermarked signal $\mathbf{Y}$ is a point of Λ_2 plus a term called the host self-interference, which is deemed noise from the point of view of the detector:

$$\mathbf{Y} = Q_{\Lambda_2}(\mathbf{X}) + (1-\beta)(\mathbf{X} \mod \Lambda_2). \tag{9.11}$$

The detector set $d = 1$ if $\mathbf{R} \mod \Lambda_2 \in \gamma\mathcal{V}(\Lambda_2)$ with $0 \leq \gamma \leq 1$, and $d = 0$ otherwise. This rule gives simple expressions for false positive assuming that, under $\mathcal{H}_0$, $\mathbf{R} = \mathbf{X}$ is uniformly distributed under $\mathcal{H}_0$ (flat host assumption):

$$\mathbb{P}_{\mathsf{fp}} = \frac{|\gamma\mathcal{V}(\Lambda_2)|}{|\mathcal{V}(\Lambda_2)|} = \gamma^n, \tag{9.12}$$

$$E_{\mathsf{fp}} = -\log\gamma. \tag{9.13}$$

Under $\mathcal{H}_1$, $d = 1$ and $\mathbb{P}_{\mathsf{fn}} = 0$ when $(1-\beta) \leq \gamma$. Otherwise, $\mathbb{P}_{\mathsf{fn}} = 1 - (\gamma/1-\beta)^n$ and $E_{\mathsf{fn}} = 0$. The characteristic is a 'on-off' function: $E_{\mathsf{fn}} = +\infty$ if $E_{\mathsf{fp}} \leq E_{\mathsf{fp}}^R$ and 0 otherwise, with

$$E_{\mathsf{fp}}^R = \begin{cases} -\log\left(1 - \sqrt{\frac{P}{\sigma^2(\Lambda_2)}}\right) & \text{if } P < \sigma^2(\Lambda_2) \\ +\infty & \text{otherwise.} \end{cases} \tag{9.14}$$

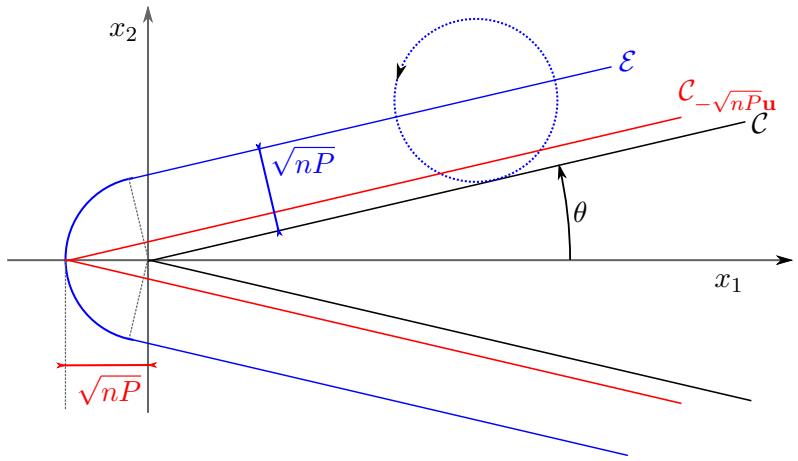


Figure 9.2: The detection region is the hypercone $\mathcal{C}$ (in black) while the embedding region $\mathcal{E}$ (in blue) contains the shifted hypercone $\mathcal{C}_{-\sqrt{nP}\mathbf{u}}$ (in red).

Clearly, Λ_2 is a fine lattice with a small $\sigma^2(\Lambda_2)$ to make this scheme asymptotically perfect on a large range of P. This also enforces the 'flat host assumption'. The same analysis holds with the strict embedding constraint (7.1).

In this second version of the Voronoï modulation, the detector is oblivious to P and σ_X^2.

9.4 Hypercone detector

This section goes back to the scheme presented in Sect. 8.2 where the detection region is a circular hypercone. Without specifying yet the embedding, *i.e.* the way $\mathbf{W}$ depends on $\mathbf{X}$, we elaborate on the achievable error exponents in the noiseless scenario and the strict embedding distortion constraint (7.1).

9.4.1 Single hypercone detection region

The detection region is the circular hypercone $\mathcal{C}$ whose apex is the origin and half-angle θ. This produces $E_{\mathsf{fp}} = -\log \sin\theta$ (see Sect. 8.2). Under $\mathcal{H}_1$, the embedder succeeds in watermarking a content when $\mathbf{X}$ is not too far away from the frontier of this hypercone. The embedding region $\mathcal{E}$ is depicted in two dimensions in Fig. 9.2.

Denote by $\mathcal{C}_{\mathbf{v}}$ the circular hypercone $\mathcal{C}$ translated by the vector $\mathbf{v}$ (whose apex is now the point $\mathbf{v}$). Fig. 9.2 shows that $\mathcal{C}_{-\sqrt{nP}\mathbf{u}} \subset \mathcal{E}$ (see also [147, Prop. 4.1.c]). Thus, $\mathbb{P}(\mathbf{X} \notin \mathcal{E})$ is lower bounded by $\mathbb{P}(\mathbf{X} \notin \mathcal{C}_{-\sqrt{nP}\mathbf{u}}) = \mathbb{P}((\mathbf{X} + \sqrt{nP}\mathbf{u}) \notin \mathcal{C})$, whose expression is given by Shannon's formula (8.9). This confirms that E_{fn} is bigger or equal to the error exponent (8.11) of the hypercone detector without side-information (see Sect. 8.2).

Appendix 16.1.2 derives the following expression thanks to Laplace method:

$$E_{\mathsf{fn}} = \begin{cases} 0, & \text{if } A := \sqrt{P/\sigma_X^2} \leq \cos\theta \\ S(\tilde{r}^{\star 2}) + 1/2 \left(\frac{\tilde{r}^\star}{\tan\theta} - \frac{A}{\sin\theta} \right)^2 & \text{otherwise} \end{cases} \tag{9.15}$$

with

$$S(x) := 1/2\,(x - 1 - \log(x))\,,\ \forall x \in \mathbb{R}^{+\star} \tag{9.16}$$

$$\tilde{r}^{\star} := 1/2\left(A\cos\theta + \sqrt{A^2\cos^2\theta + 4\sin^2\theta}\right). \tag{9.17}$$

On the left endpoint, *i.e.* as $\theta \to \pi/2$, there is no improvement compared to Sect 8.2: $E_{\mathsf{fn}}^L = A^2/2$. The main difference comes on the right endpoint:

$$E_{\mathsf{fp}}^R = \begin{cases} +\infty & \text{if } A \geq 1 \\ -1/2\log(1 - A^2) & \text{otherwise.} \end{cases} \tag{9.18}$$

In the case $A > 1$ (which is not relevant in practice), E_{fn} is strictly positive for any E_{fp}. Indeed, $\lim_{E_{\mathsf{fp}}\to\infty} E_{\mathsf{fn}} = S(A^2) \leq A^2/2$. For $0 < A \leq 1$, the whole characteristic is even better than the one of the optimal Neyman-Pearson detector without side-information of Sect. 8.1, especially on the right endpoint as $-1/2\log(1 - A^2) \geq A^2/2$. This shows that, in the noiseless scenario, side information gains back the loss that was due to being oblivious to parameters (P, N) between Sect. 8.1 and Sect. 8.2.

Yet, this characteristic is far from the perfection or almost perfection of the schemes seen in Sect. 9.2.

9.4.2 Dual hypercone detection region

A simple way to improve the performances is to consider a dual hypercone detection region, *i.e.* the union of two single hypercones around directions $\mathbf{u}$ and $-\mathbf{u}$. This was invented in 1999 [105, Eq.(5)], and theoretically justified only seven years after in [143].

The probability of false positive is just the double of (8.6), providing the same error exponent: $E_{\mathsf{fp}} = -\log\sin\theta$. The false negative under the noiseless scenario is different and detailed in Appendix 16.2.1:

$$E_{\mathsf{fn}} = \begin{cases} 0, & \text{if } A \leq \cos\theta \\ S\left(\frac{A^2}{\cos^2\theta}\right) & \text{otherwise} \end{cases} \tag{9.19}$$

The improvements compared to the single hypercone are mitigated as illustrated by the following properties.

- **Left endpoint**. $E_{\mathsf{fn}}^L = +\infty$: As the half angle of the dual cone opens up, the embedding region tends to the full space, and watermarking is feasible almost surely. This is a huge improvement compared to the single hypercone.
- **Right endpoint**. As for E_{fp}^R:

$$E_{\mathsf{fp}}^R = \begin{cases} +\infty & \text{if } A \geq 1, \\ -\frac{1}{2}\log\left(1 - A^2\right) & \text{otherwise.} \end{cases} \tag{9.20}$$

  This right endpoint value was already achieved by the single hypercone.

The dual hypercone is good for boosting the characteristic on the left hand side, but it does not help much for the right hand side. The main point is that this improvement comes for free from the complexity point of view.

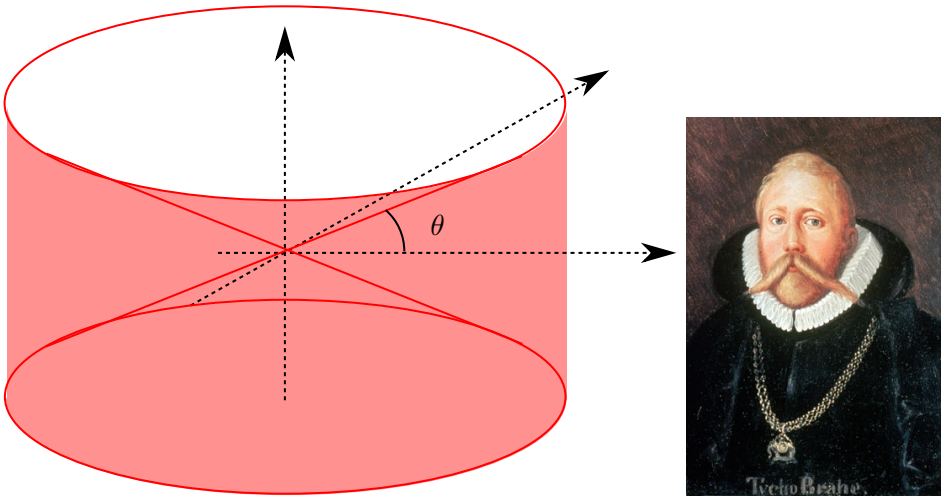


Figure 9.3: The detection region of the k-dimensional ruff with $k = 2$ and $n = 3$. Tycho Brahe wearing a ruff collar.

9.5 Extension of the dual cone: k dimensional Ruff

This section introduces an extension of the dual cone called the k dimensional ruff.

The scheme

Remember the following notation:

$$\mathbf{X}_{(k)} := (X(1), \dots, X(k), 0, \dots, 0) \quad \text{and} \quad \mathbf{X}^{(n-k)} := (0, \dots, 0, X(k+1), \dots, X(n)),$$

so that $\mathbf{X} = \mathbf{X}_{(k)} + \mathbf{X}^{(n-k)}$. Vectors $\mathbf{X}_{(k)}$ and $\mathbf{X}^{(n-k)}$ are the projections of $\mathbf{X}$ onto two orthogonal complimentary subspaces of dimension k and $n-k$. The dual cone scheme is a special case with $k = 1$ of the following detection: compare the projection energy ratio $\|\mathbf{X}_{(k)}\|/\|\mathbf{X}\|$ to the threshold $\cos(\theta)$. The detection region looks like a 'ruff' collar when $k = 2$ and $n = 3$ (see Fig. 9.3).

False positive error exponent

The probability of false positive is given by

$$\mathbb{P}_{\mathsf{fp}} = 1 - I_{\cos^2\theta}\left(\frac{k}{2}, \frac{n-k}{2}\right). \tag{9.21}$$

For a fixed $k > 1$, this probability has the same error exponent as for $k = 1$. Yet, if $k = \rho n$ with $0 \leq \rho < 1$ (or more rigorously if $\lim_{n\to+\infty} k/n = \rho$), the exponent is modified as follows (see Appendix 16.3.1):

$$E_{\mathsf{fp}} = \begin{cases} 0 & \text{if } \cos(\theta) \leq \sqrt{\rho} \\ \frac{1}{2}\left(\rho \log\left(\frac{\rho}{\cos^2(\theta)}\right) + (1-\rho)\log\left(\frac{1-\rho}{\sin^2(\theta)}\right)\right) & \text{otherwise} \end{cases} \tag{9.22}$$

This last expression equals $1/2 D(\mathcal{B}(\rho)||\mathcal{B}(\cos^2(\theta)))$, the Kullback Leibler divergence between two Bernoulli distributions. Note that:

- $E_{\mathsf{fp}} \leq -\log \sin \theta$. Equality holds for $\rho = 0$, the case of a fixed k like the dual hypercone. For a given half angle θ, the ruff detection region as a much wider solid angle than the dual hypercone.
- There is a symmetry $(\rho, \theta) \leftrightarrow (1-\rho, \pi/2 - \theta)$. The complementary of a ruff of dimension $k = \rho n$ and half angle θ is a ruff of dimension $n - k = (1-\rho)n$ and half angle $\pi/2 - \theta$.

False negative error exponent

As for E_{fn} in the noiseless scenario (see Appendix 16.3.1):

$$E_{\mathsf{fn}} = \begin{cases} 0 & \text{if } A \leq \cos(\alpha + \theta) \\ \min_{\tilde{\mathcal{D}}} \ \rho S\left(\frac{\tilde{r}_1^2}{\rho \sigma_X^2}\right) + (1-\rho) S\left(\frac{\tilde{r}_n^2}{(1-\rho)\sigma_X^2}\right) & \\ \quad \text{with } \tilde{\mathcal{D}} = \left\{(\tilde{r}_1, \tilde{r}_n) \in \mathbb{R}^+ \times \mathbb{R}^+ | \tilde{r}_n \geq \tilde{r}_1 \tan\theta + \frac{\sqrt{P}}{\cos\theta}\right\} & \text{otherwise} \end{cases} \tag{9.23}$$

where angle α ($0 \leq \alpha \leq \pi/2$) is defined s.t. $\cos\alpha := \sqrt{1-\rho}$.

Note that $\lim_{\rho \to 0} \rho S(\tilde{r}_1^2/\rho\sigma_X^2) = \tilde{r}_1^2/2\sigma_X^2$ and $\lim_{\rho \to 0} \alpha = 0$, so that (9.23) converges to (9.19) as $\rho \to 0$. The k-dimensional ruff is a generalization of the dual hypercone.

The solution of the minimization problem has a complicated expression. Yet, simple upper bounds exist:

$$E_{\mathsf{fn}} \leq \min\left((1-\rho) S\left(\frac{(\sqrt{\rho}\sin\theta + A)^2}{(1-\rho)\cos^2\theta}\right), \rho S\left(\frac{(\sqrt{1-\rho}\cos\theta - A)^2}{\rho \sin^2\theta}\right)\right). \tag{9.24}$$

This upper bound is tight in the following cases: i) as $\rho \to 0$, it converges to (9.19), the false negative error rate of the dual hypercone scheme ; ii) as A goes down to $\cos(\alpha + \theta)$, the upper bound vanishes to 0, and so does E_{fn} as already stated by (9.23). Nevertheless, the upper bound is not tight as $\theta \to 0$, *i.e.* as $E_{\mathsf{fp}} \to +\infty$. In that case:

$$\lim_{E_{\mathsf{fp}} \to +\infty} E_{\mathsf{fn}} = \begin{cases} 0 & \text{if } A \leq \cos\alpha \\ (1-\rho) S\left(\frac{A^2}{1-\rho}\right) & \text{otherwise} \end{cases} \tag{9.25}$$

Right endpoint

The computation of E_{fp}^R comes from comment ii) above:

- If $A \geq \cos\alpha$, $E_{\mathsf{fp}}^R = +\infty$,
- Otherwise, $E_{\mathsf{fn}} = 0$ as θ increases from 0 until it meets the angle $\psi - \alpha$, with ψ defined s.t. $\cos\psi := A$. The angle of the ruff is open enough to enable watermarking. E_{fp}^R is given by (9.22) with $\theta = \psi - \alpha$.

The following properties concerning the right endpoint are interesting in order to compare ruff ($\rho > 0$) and dual hypercone ($\rho = 0$):

- For a fixed ρ, E_{fp}^R is an increasing function of A. As for all previous schemes, more watermarking power means higher E_{fp}^R. This holds in the range $A \in [0, \cos\alpha)$. After that value, $E_{\mathsf{fp}}^R = +\infty$. In other words, for $A > \sqrt{1-\rho}$, $E_{\mathsf{fn}} > 0$ for any half angle of the ruff. Note that for the dual hypercone scheme, this 'perfection' is only achieved when $A > 1$.

- For a fixed ρ, the limit of E_{fn} as $E_{\mathsf{fp}} \to +\infty$ (9.25) is greater or equal to the same limit value for the dual hypercone. Equality holds only when this limit is 0, *i.e.* when $A \leq \cos\alpha$.
- For a fixed A, E_{fp}^R equals the right endpoint of the dual hypercone (9.20) for $\rho = 0$. Then, E_{fp}^R increases with $\rho \in [0, 1 - A^2)$. This means that the ruff detector with $\rho > 0$ performs always better than a dual hypercone in terms of E_{fp}^R. If $\rho > 1 - A^2$, then $E_{\mathsf{fp}}^R = +\infty$.
- For a fixed A, the limit (9.25) increases with ρ. It goes up to $A^2/2$ when $\rho = 1$, the false negative exponent of ZATT (see Sect. 9.2).

Operating point

Optimizing parameters (ρ, θ) to maximize E_{fn} for a given A while E_{fp} is set to a targeted level E would be a mistake: We would like to maintain the detector oblivious to A. The designer first fixes parameters (ρ, θ) operating at $E_{\mathsf{fp}} = E$, and then E_{fn} varies with A. The last properties above-mentioned tells that a good choice is to set $\rho \lesssim 1$. By doing so, E_{fp}^R is larger than the operating level E for a large range of A.

Note that, for a given E, $\rho \lesssim 1$ implies that $\theta \gtrsim 0$. The ruff is high dimensional but almost flat. A vector is deemed watermarked if its last $(n-k) = n(1-\rho)$ components are almost zeros. We find back the ZATT scheme detailed in Sect. 9.2 (except that $n-k$ and k have been exchanged).

There are however prices to be paid:

- E_{fn} is much smaller when $\rho > 0$. For instance, E_{fn}^L is no longer infinite as it used to be with the hypercone. The upper bound (9.24) shows that $E_{\mathsf{fn}}^L \leq A^2/\rho$ for $0 < \rho < 1$ and $\lim_{\rho \to 1} E_{\mathsf{fn}}^L = A^2/2$ as for the ZATT scheme of Sect. 9.2.
- The complexity of the scheme: we have to compute the energy of two projections. As I have presented it so far, the complexity of the ruff's scheme is $O(n)$, like for the single and dual hypercone detectors. Yet, for security reason, the two projections should be secret keyed: vectors first go through an orthogonal secret matrix and then the norms $\|\mathbf{X}_{(k)}\|$ and $\|\mathbf{X}\|$ are computed. The complexity of this orthogonal transform scales as $O(n \log n)$ for fast implementation. Comparisons with the dual hypercone, which was proven optimal under the assumption of low complexity [143, 99], are unfair with this respect.

9.6 Conclusion of the noiseless scenario

In the Case 3 of Sect. 7.5, the schemes of Chapter 8 yield perfect detection when $\sigma_Z^2 \to 0$. The question of Costa in Sect. 7.5 in the noiseless scenario asks whether there exists a side-informed scheme (Case 2) asymptotically reaching perfection $(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = (+\infty, +\infty)$ of Case 3. Figures 9.4 and 9.5 summarize the results of this chapter.

When the detector is not oblivious, the answer to the question of Costa is positive. The first side-informed version of the Voronoï modulation detailed in Sect. 9.3.1 has perfect performances, the embedding constraint being strict or on expectation. The 'flat assumption' solves the issue of transmitting the dither at the detection side.

When the detector is oblivious, the answer to the question of Costa is mostly positive. ZATT (see Sect. 9.2) provides perfect performances with the embedding constraint on

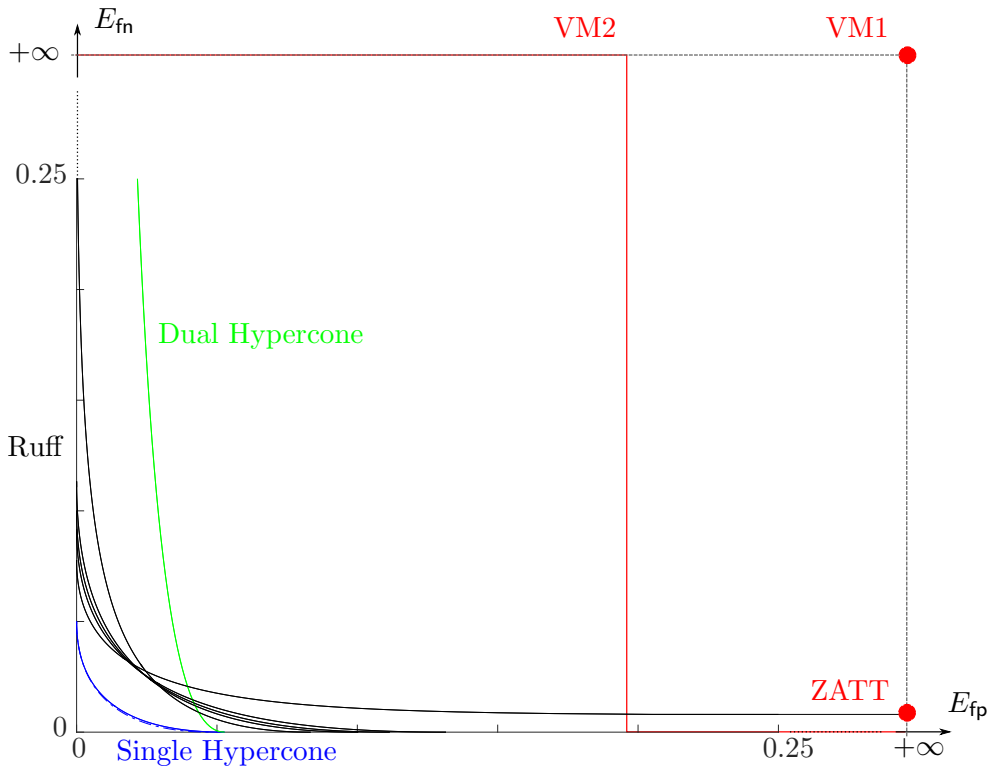


Figure 9.4: Characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$ for $P = 0.1$, $\sigma_X^2 = 1$, and $\sigma_Z^2 = 0$. Non side-informed embedder (Case 1, *i.e.* $N = \sigma_X^2$) in dashed *blue*. Schemes providing some perfection: Voronoï modulation with scaled lattice (version 1–VM1) and with fixed lattice (version 2–VM2), and ZATT. The dual hypercone has a greater E_{fn} than the single hypercone. Yet, they achieve the same E_{fp}^R. In black, the ρn-dimensional ruff for $\rho \in \{0.05, 0.275, 0.5, 0.725, 0.95\}$. For this last value, $E_{\mathsf{fp}}^R = +\infty$.

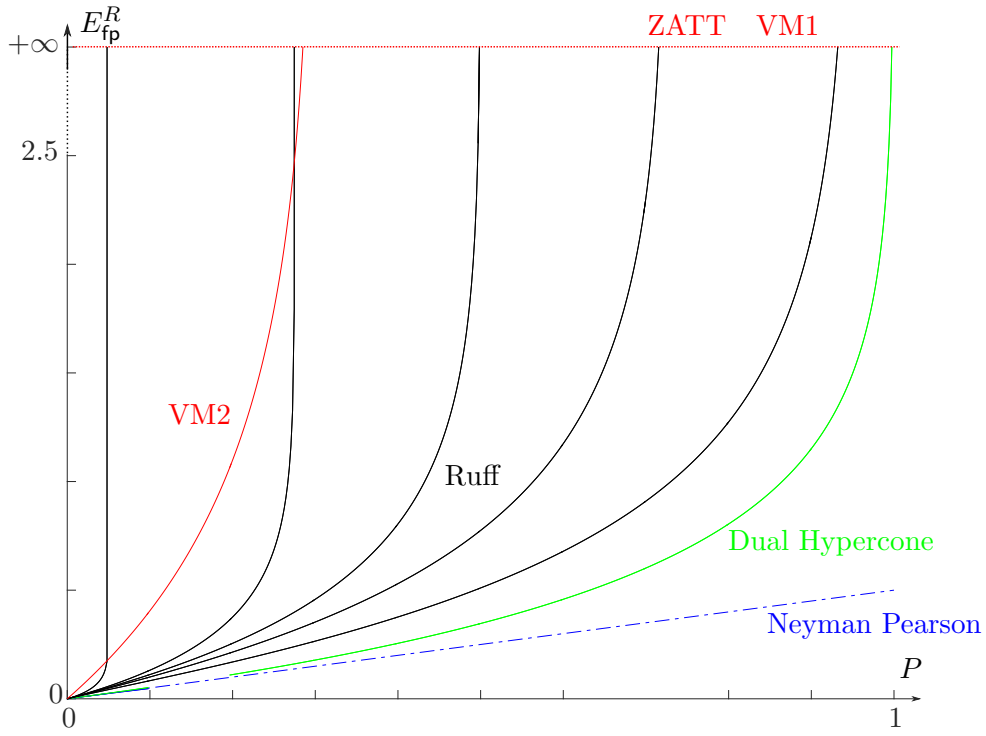


Figure 9.5: E_{fp}^R as a function of P for $\sigma_X^2 = 1$ and $\sigma_Z^2 = 0$. Schemes providing some perfection: Voronoï modulation with scaled lattice (version 1–VM1) and with fixed lattice (version 2–VM2), and ZATT. The dual hypercone has the same E_{fp}^R than the single hypercone (not shown). They outperform the Neyman Pearson (Not side-informed but not oblivious) (*i.e.* Case 1 (8.1) with $N = \sigma_X^2$). In black, the ρn-dimensional ruff for $\rho \in \{0.05, 0.275, 0.5, 0.725, 0.95\}$.

expectation. It does this even non asymptotically. With the strict embedding constraint, the scheme is 'less perfect': $(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = (+\infty, {}^{A^2}/_2)$. Similar conclusion holds for the second version of the Voronoï modulation detailed in Sect. 9.3.2.

Yet, these schemes are very artificial. They are designed for the noiseless scenario. ZATT for instance cannot handle any attack noise. This is the reason why we also presented side informed versions of the hypercone detector, which the next chapter reveals to be more robust.

Side information makes the oblivious single hypercone works better than the non-oblivious non-side-informed Neyman-Pearson detector. The dual hypercone gives an extra boost on E_{fn}. However, it does not improve E_{fp}^R. The ruff generalizes the dual hypercone. Its main feature is to trade E_{fn} for E_{fp}^R thanks to parameter ρ: Exponent E_{fn} is smaller but remains strictly positive over a larger range of E_{fp}.

Chapter 10

One source of side information and one source of noise

Here is now the most interesting setup considering a side-informed embedder and a source of noise.

- $\mathcal{H}_0$: $\mathbf{R} = \mathbf{X} + \mathbf{Z}$
- $\mathcal{H}_1$: $\mathbf{R} = \mathbf{X} + \mathbf{w}(\mathbf{X}) + \mathbf{Z}$

We shall however restrict our study to two schemes:

- Oblivious schemes: We have seen that the most promising scheme is the k-dimensional ruff detection, a generalization of the hypercone detection scheme. In this section, $k = n\rho$, knowing that the ruff encompasses the hypercone by setting $\rho = 0$. The hypercone detector has been investigated in [99] but only in the 'high SNR regime' (indeed, asymptotically for $\sigma_Z^2 \to 0$, so in the noiseless setup seen in Chapter 9). The study is somewhat more complex when $\rho > 0$, and we shall go back to the hypercone times to times to gain some insights. This also provides new results for the most well-know scheme in the literature, *i.e.* the hypercone detection. The case $\rho = 1$, which is a model of the ZATT scheme, cannot be used in this setup (ZATT needs knowing N to set the radius of the detection ball region around the single codeword $\mathbf{0}_k$).
- Non-oblivious schemes: We focus on the Voronoï modulation with side information based on the articles [130, 131, 162].

10.1 k-dimensional ruff detection

Section 9.5 has already revealed the expression of E_{fp} (9.22). This section assumes that $\cos\theta > \sin\alpha$ so that $E_{\mathsf{fp}} > 0$. The appendix 16.3.2 shows that

$$E_{\mathsf{fn}} = \min_{\tilde{\mathcal{D}}} \frac{\tilde{z}_1^2 + \tilde{z}_n^2}{2\sigma_Z^2} + \rho S\left(\frac{\tilde{x}_1^2}{\rho\sigma_X^2}\right) + \rho S\left(\frac{\tilde{q}_1^2}{\rho\sigma_Z^2}\right) + (1-\rho)S\left(\frac{\tilde{x}_n^2}{(1-\rho)\sigma_X^2}\right) + (1-\rho)S\left(\frac{\tilde{q}_n^2}{(1-\rho)\sigma_Z^2}\right) \tag{10.1}$$

with $\tilde{\mathcal{D}} \subset \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}^+$ defined by:

$$\tilde{\mathcal{D}} := \{(\tilde{x}_1, \tilde{x}_n, \tilde{z}_1, \tilde{z}_n, \tilde{q}_1, \tilde{q}_n) | \left((\tilde{x}_1 + \tilde{w}_1 + \tilde{z}_1)^2 + \tilde{q}_1^2\right) \tan^2\theta < (\tilde{x}_n + \tilde{w}_n + \tilde{z}_n)^2 + \tilde{q}_n^2\} \tag{10.2}$$

This encompasses the hypercone detector by the fact that $\lim_{\rho\to 0} \rho S(x/\rho) = x/2$, $\forall x \geq 0$.

10.1.1 When does $E_{\mathsf{fn}} = 0$?

Equation (10.1) is a complex expression of the characteristic $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$. As motivated in Chapter 7, the right end-point is indeed the figure of merit of interest. This section studies this end of the characteristic by seeing when $E_{\mathsf{fn}} = 0$.

Equations (10.1) and (10.2) tell that $E_{\mathsf{fn}} = 0$ if and only if:

$$\left(\sqrt{\rho\sigma_X^2}, \sqrt{(1-\rho)\sigma_X^2}, 0, 0, \sqrt{\rho\sigma_Z^2}, \sqrt{(1-\rho)\sigma_Z^2}\right) \in \tilde{\mathcal{D}} \tag{10.3}$$

because $S(x) = 0$ if and only if $x = 1$. This can be rewritten as

$$H(\tilde{w}_1, \tilde{w}_n) < \sigma_Z^2, \quad \text{with} \tag{10.4}$$

$$H(\tilde{w}_1, \tilde{w}_n) := \frac{(\sigma_X \sin\alpha + \tilde{w}_1)^2 \sin^2\theta - (\sigma_X \cos\alpha + \tilde{w}_n)^2 \cos^2\theta}{\cos^2\theta - \sin^2\alpha}. \tag{10.5}$$

For the hypercone detector where $\sin\alpha = \sqrt{\rho} = 0$, a statistical interpretation is the following. Asymptotically, the performance of the scheme is governed by the way the typical realization of host signal is watermarked. This typical host is orthogonal to the axis of the hypercone ($\tilde{x}_1 = 0$) and has a norm $\sqrt{n}\sigma_X$ ($\tilde{x}_1^2 + \tilde{x}_n^2 = \sigma_X^2$). Moreover, the typical realization of the noise has a norm norm $\sqrt{n}\sigma_Z$ ($\tilde{z}_1^2 + \tilde{z}_n^2 + \tilde{q}_1^2 + \tilde{q}_n^2 = \sigma_Z^2$), is orthogonal to the the axis of the hypercone ($\tilde{z}_1 = \tilde{q}_1 = 0$), and is orthogonal to the host ($\tilde{z}_n = 0$). Exponent E_{fn} is null if this typical noise drives the watermarked signal outside the hypercone. The intersection of the hypercone with the plane $\tilde{q}_n = \sigma_Z$ gives the hyperbola $\tilde{y}_1^2 \tan^2\theta - \tilde{y}_n^2 = \sigma_Z^2$ as depicted in Fig. 10.1.

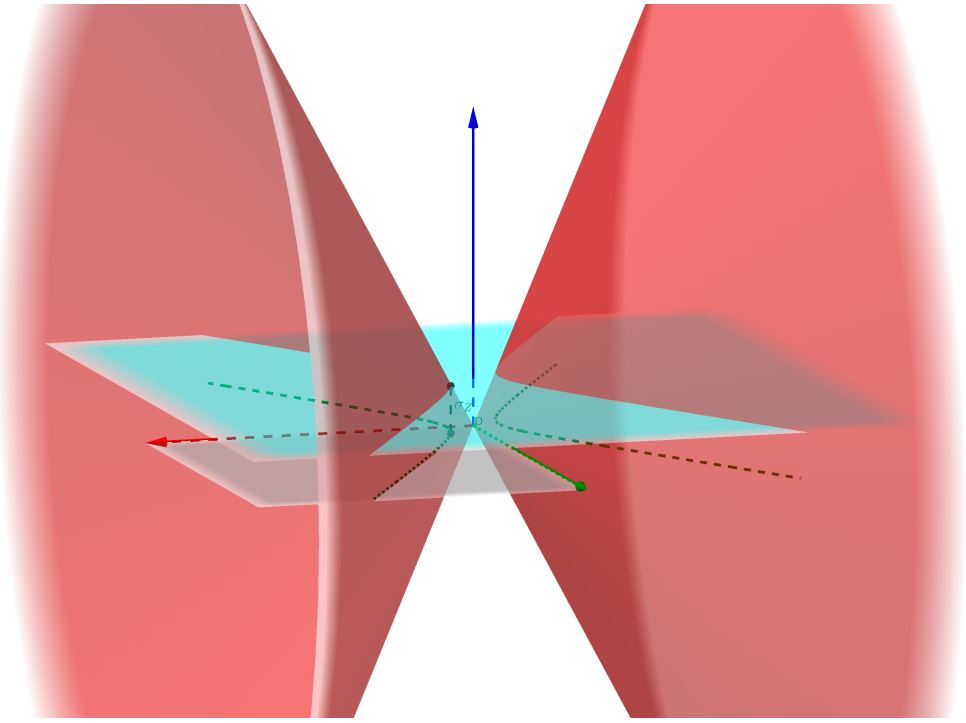


Figure 10.1: Graphical representation in 3D space $(\tilde{y}_1, \tilde{y}_n, \tilde{q}_n)$ for the hypercone detector. The detection region is the inside of the red hypercone. Equation (10.2) describes the hyperbola $\tilde{y}_1^2 \tan^2\theta - \tilde{y}_n^2 = \sigma_Z^2$ (dashed green line) which is the intersection of the hypercone by the plane $\tilde{q}_n = \sigma_Z$ in cyan projected onto the plane $\tilde{q}_n = 0$.

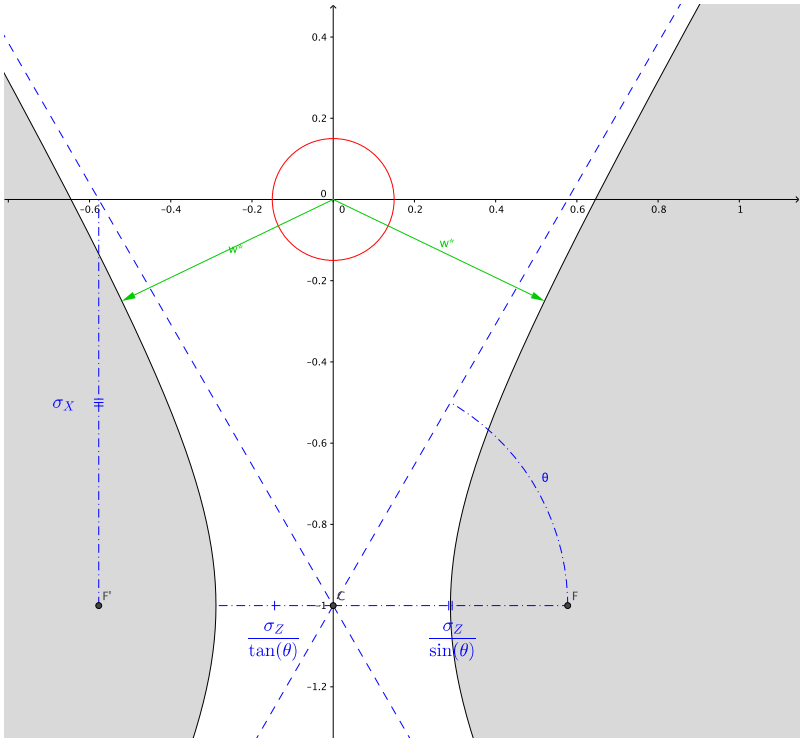


Figure 10.2: Conditions for $E_{\mathsf{fn}} \geq 0$ in the plane $(\tilde{w}_1, \tilde{w}_2)$: P is too small and $E_{\mathsf{fn}} = 0$. Setup: $\alpha = 0$ (hypercone detector), $\sigma_X = 1$, $\sigma_Z = 0.5$, and $\theta = \pi/3$. The distortion constraint defines the red circle of radius $\sqrt{P}$ centered on the origin $(0, 0)$; $E_{\mathsf{fn}} \geq 0$ defines the gray area outside the hyperbola of focuses F and F' $((\pm\sigma_Z/\sin\theta, -\sigma_X))$ and center $\mathsf{C} = (0, -\sigma_X)$. Its asymptotes are the dashed blue lines.

10.1.2 Provably good embeddings

We adopt now the point of view of the embedder. Our goal is to avoid such a null error exponent by carefully designing a watermark embedding $(\tilde{w}_1, \tilde{w}_n)$. In a 2D plane mapping point $(\tilde{w}_1, \tilde{w}_n)$, the embedding constraint $\tilde{w}_1^2 + \tilde{w}_n^2 \leq P$ defines a ball of radius $\sqrt{P}$ centered on $(0, 0)$ whereas equation (10.4) defines a region delimited by an hyperbola (equality in (10.4)) of center $(-\sigma_X \sin\alpha, -\sigma_X \cos\alpha)$. As $\sigma_Z \to 0$, the high-SNR regime tends to the noiseless scenario, and the hyperbola 'shrinks' towards its asymptotes: $\sigma_X \cos\alpha + \tilde{w}_n = \pm(\sigma_X \sin\alpha + \tilde{w}_1) \tan\theta$. Figures 10.2, 10.3, and 10.4 shows the situation for $\alpha = 0$ (hypercone detector).

When P is too small, the entire ball is contained 'inside' the hyperbola (*i.e.* in between the two branches of the hyperbola as depicted in Fig. 10.2): Whatever the embedding, the false negative error exponent is zero. If P is big enough, the ball intersects the hyperbola and there are some embedding strategies which provide non zero error exponent (Fig. 10.4). We are interested in the limit case when the ball has a kissing point with the hyperbola (Fig. 10.3).

Hypercone detector ($\rho = 0$).

When $\rho = 0$ so that $\alpha = 0$, the hyperbola is symmetric w.r.t. the axis $\{\tilde{w}_1 = 0\}$ and there are two kissing points (one on the left hand side, the other on the right hand side of the hyperbola

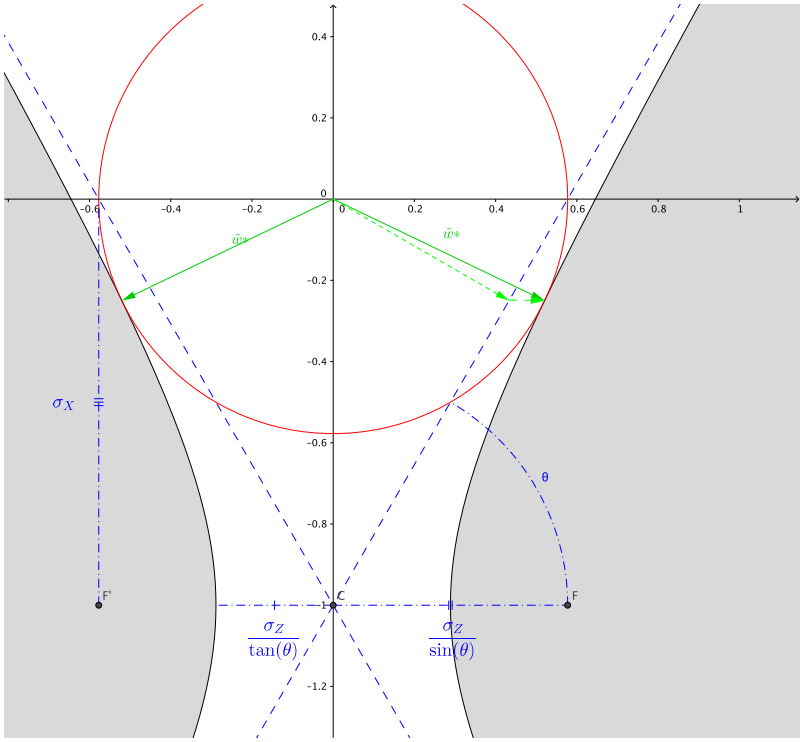


Figure 10.3: Conditions for $E_{\mathsf{fn}} \geq 0$ in the plane $(\tilde{w}_1, \tilde{w}_2)$: $P = \sigma_X^2 \cos^2\theta + \sigma_Z^2 \tan^{-2}\theta$ and there are two kissing points given by the optimal watermark vector $\tilde{\mathbf{w}}^\star$ in green. Setup: $\alpha = 0$ (hypercone detector), $\sigma_X = 1$, $\sigma_Z = 0.5$, and $\theta = \pi/3$. The distortion constraint defines the red circle of radius $\sqrt{P}$ centered on the origin $(0, 0)$; $E_{\mathsf{fn}} \geq 0$ defines the gray area outside the hyperbola of focuses F and F' $((\pm\sigma_Z/\sin\theta, -\sigma_X))$ and center $\mathsf{C} = (0, -\sigma_X)$. Its asymptotes are the dashed blue lines.

–Fig. 10.3). The system of equations provided by (10.4) (with equality) and $\tilde{w}_1^2 + \tilde{w}_n^2 = P$ implies that:

$$-(1+\tan^2\theta).\tilde{w}_n^2 - 2\sigma_X.\tilde{w}_n + (P\tan^2\theta - \sigma_X^2 - \sigma_Z^2) = 0. \tag{10.6}$$

This polynomial of degree two has a unique solution if and only if $P = P_0 + \sigma_Z^2 \tan^{-2}\theta$ with $P_0 := \sigma_X^2 \cos^2\theta$. Consider the three following cases:

- if $P \leq P_0$ (as in Fig. 10.2), then $E_{\mathsf{fn}} = 0$ for any σ_Z, including $\sigma_Z = 0$. We rediscover result (9.19) from the noiseless scenario.
- if $P_0 < P \leq P_0 + \sigma_Z^2 \tan^{-2}\theta$, then $E_{\mathsf{fn}} = 0$ for that particular noise level σ_Z, but it might be strictly positive for a less harmful attack.
- if $P_0 + \sigma_Z^2 \tan^{-2}\theta < P$ (as in Fig. 10.4), then $E_{\mathsf{fn}} > 0$ for this noise level and, in this sense, the watermark is robust to that attack.

When we have exact equality $P = P_0 + \sigma_Z^2 \tan^{-2}\theta$ (as in Fig. 10.3), the two kissing points are given by:

$$(\tilde{w}_1^\star, \tilde{w}_n^\star) := \left(\pm\sqrt{P - \sigma_X^2\cos^4\theta}, -\sigma_X\cos^2\theta\right). \tag{10.7}$$

In practice, the sign of $\tilde{w}_1^\star$ agrees with the sign of x_1, *i.e.* the sign of $\mathbf{x}^\top\mathbf{u}$. The surprise is that this embedding is the one coined by Comesaña *et al.* [99] as optimum in the noiseless scenario (they use the term 'high SNR regime', but indeed their study only covers the noiseless scenario), whereas this assumption is not needed here.

We call this embedding optimal with the following meaning: it is *not* the embedding that maximizes E_{fn} for a given σ_Z^2. This would certainly be a function of σ_Z^2. On the contrary, (10.7) is oblivious to σ_Z^2. It is the embedding that makes $E_{\mathsf{fn}} > 0$ over the biggest noise power range. In other words, it maximizes $E_{\mathsf{fp}}^{(R)}$.

A nice interpretation follows: if $P = P_0 + \delta P$ with $\delta P > 0$, then $\tilde{w}_1^{\star 2} = P_0\sin^2\theta + \delta P$ and $\tilde{w}_n^{\star 2} = P_0\cos^2\theta$. In words, the watermark signal first reaches the asymptotes of the hyperbola in order to guarantee $E_{\mathsf{fn}} > 0$ in the noiseless scenario. The 'shortest path' is to project $(0,0)$ on the asymptote by going along direction $(\sin\theta, -\cos\theta)$. This consumes the embedding power P_0. If it remains some extra embedding power $\delta P > 0$, the watermark signal carries on pushing the host signal *only* along the direction of the axis of the hypercone. This is depicted in Fig. 10.3.

Ruff detector ($\rho > 0$).

The story is quite different for the ruff detector because the axis of the hyperbola no longer contains the center of the embedding constraint ball. There is no closed-form expression to project a point (here the origin $(0,0)$) on a hyperbola and thus to derive its distance.

However, we can find the optimal embedding strategy under the noiseless scenario. When $\sigma_Z = 0$, the hyperbola converges to its asymptotes. The kissing point between the embedding ball and the hyperbola is the projection of the center of the ball onto the asymptote line. This amounts to looking for the unique root of a polynomial of degree 2 which exists if $P = P_0$ with $P_0 := \sigma_X^2\cos^2(\alpha+\theta)$. This rediscovers a result from the noiseless scenario (see (9.23)). The novelty is that we now get the optimal embedding:

$$(\tilde{w}_1^\circ, \tilde{w}_n^\circ) = \sqrt{P_0}(\sin\theta, -\cos\theta). \tag{10.8}$$

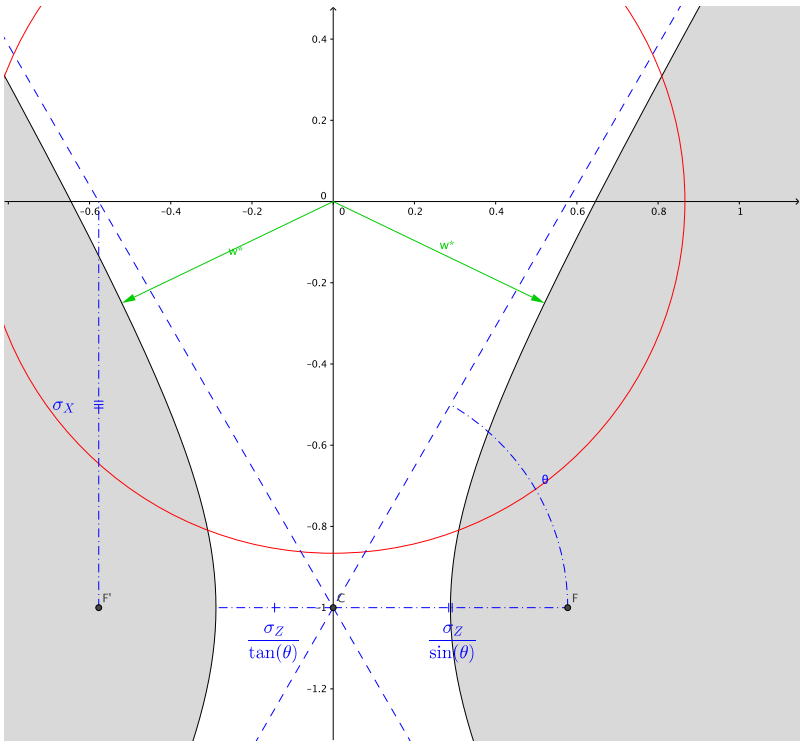


Figure 10.4: Conditions for $E_{\mathsf{fn}} \geq 0$ in the plane $(\tilde{w}_1, \tilde{w}_2)$: P is big enough so that there exist embeddings $(\tilde{w}_1, \tilde{w}_2)$ on the red circle inside the gray area. Setup: $\alpha = 0$ (hypercone detector), $\sigma_X = 1$, $\sigma_Z = 0.5$, and $\theta = \pi/3$. The distortion constraint defines the red circle of radius $\sqrt{P}$ centered on the origin $(0, 0)$; $E_{\mathsf{fn}} \geq 0$ defines the gray area outside the hyperbola of focuses F and F' $((\pm\sigma_Z/\sin\theta, -\sigma_X))$ and center $\mathsf{C} = (0, -\sigma_X)$. Its asymptotes are the dashed blue lines.

In the noisy scenario, the optimal embedding maximizes $H(\tilde{w}_1, \tilde{w}_n)$ so that σ_Z^2 must be big to cancel E_{fn} (see (10.4)). As already said, there is no closed-form expression but a numerical solver easily finds the solution. The embedding is prototyped as $(\tilde{w}_1, \tilde{w}_n) = \sqrt{P}(\cos\beta, \sin\beta)$. Obviously, $\beta \in [-\pi/2, 0]$ because the watermark pushes the host vector towards the inside of the ruff.

$$\beta^\star := \arg \max_{\beta \in [-\pi/2, 0]} H(\sqrt{P}\cos\beta, \sqrt{P}\sin\beta), \tag{10.9}$$

$$(\tilde{w}_1^\star, \tilde{w}_n^\star) := \sqrt{P}(\cos\beta^\star, \sin\beta^\star). \tag{10.10}$$

If $H(\tilde{w}_1^\star, \tilde{w}_n^\star) < 0$, it means that P is not big enough to push the host vector inside the detection region. In other words, watermarking is not possible.

In order to avoid the call to a numerical solver, we propose the following suboptimal embeddings inspired by the optimal embedding of the hypercone detector. Suppose $P = P_0 + \delta P$ with $\delta P > 0$ and write $(\tilde{w}_1, \tilde{w}_n) = (\tilde{w}_1^\circ + \epsilon_1, \tilde{w}_n^\circ + \epsilon_n)$. This embedding strategy is in a way 'conservative': it first ensures $E_{\mathsf{fn}} > 0$ in the noiseless scenario by going to $(\tilde{w}_1^\circ, \tilde{w}_2^\circ)$ (defined in (10.8)), and then looks for the best direction (ϵ_1, ϵ_n) starting from there. It happens that $E_{\mathsf{fn}} = 0$ if

$$\sigma_Z^2 \left(1 - \frac{\sin^2\alpha}{\cos^2\theta}\right) > \left(\epsilon_1^2 + 2(\sigma_X \sin\alpha + \tilde{w}_1^\circ)\epsilon_1\right)\tan^2\theta - (\epsilon_n^2 + 2(\sigma_X\cos\alpha + \tilde{w}_n^\circ)\epsilon_n). \tag{10.11}$$

Knowing that $(\sigma_X \cos\alpha + \tilde{w}_n^\circ) > 0$, we clearly see that ϵ_n should not be positive.

A first idea is to spend the extra embedding power δP only on ϵ_1. This makes the right hand side of the above inequality positive whatever the value of δP. This generalizes the embedding (10.7):

$$(\tilde{w}_1, \tilde{w}_n) = \left(\sqrt{P - \sigma_X^2\cos^2(\alpha+\theta)\cos^2(\theta)}, -\sigma_X\cos(\alpha+\theta)\cos(\theta)\right). \tag{10.12}$$

A second idea assumes that δP is small and increases the right hand side of the above inequality by going along the gradient of this expression. This gives $(\epsilon_1, \epsilon_n) \propto (\sin\theta, -\cos\theta)$ so that $(\tilde{w}_1^\circ + \epsilon_1, \tilde{w}_n^\circ + \epsilon_n) = \sqrt{P}(\sin\theta, -\cos\theta)$. The *Broken Arrows* scheme uses this idea [25] (although it works with the hypercone detector). This embedding indeed maximizes the distance between $(\tilde{x}_1 + \tilde{w}_1, \tilde{x}_n + \tilde{w}_n)$ and the boundary of the detection region. This distance represents the minimal norm of the noise vectors pushing the watermarked vector outside the detection region. In other words, this is the best embedding strategy against the worst case attack.

Note that the worst case attack is not included in our model of attacks. The worst case attack is defined from geometrical arguments with the assumption that the attacker knows how to drive the watermarked vector outside the detection region. A watermarking scheme usually depends on a secret key defining the detection region. The worst case attack is only possible when this secret key has been compromised. On the contrary, the attacks so far in this study are based on a statistical model: they are the most likely noise vectors knowing that they succeed to push the watermarked vector outside the detection region. As a fundamental difference, the worst case attack is only based on the distance from the detection boundary, whereas the likely successful attacks are based on not only this distance but also on the surface of the ensemble of points located on the boundary at this same distance.

As a summary, the embedding $(\tilde{w}_1^\star, \tilde{w}_n^\star)$ (*i.e.* (10.7) for the hypercone detector, (10.10) for the ruff detector) may not maximize E_{fn}. It is optimum from the robustness point of view, in the sense that it maximizes the noise power needed to cancel E_{fn}.

What matters in practice

These are theoretical developments with limited practicality. The previous embeddings have been derived when $(x_1, x_n) = \sqrt{n}\sigma_X(\sqrt{\rho}, \sqrt{1-\rho})$. Asymptotically as $n \to +\infty$, typical host vectors are such that their energy is $n\sigma_X^2$ exactly split into the subspaces proportionally to their dimensions. The way these typical vectors are watermarked governs E_{fn}. However, it gives no clue on how to watermark host vectors $(x_1, x_2) \neq \sqrt{n}\sigma_X(\sqrt{\rho}, \sqrt{1-\rho})$.

To do so, the whole analysis has to be carried over replacing the typical value of $(\tilde{x}_1, \tilde{x}_n)$ (*i.e.* $\sigma_X(\sqrt{\rho}, \sqrt{1-\rho})$) by its true value $n^{-1/2}(\|\mathbf{x}_{(k)}\|, \|\mathbf{x}^{(n-k)}\|)$. Variables $\tilde{x}_1$ and $\tilde{x}_n$ disappear from (10.1) since they are no longer random when watermarking this particular host. The following algorithm is the embedding providing $E_{\mathsf{fn}} > 0$ over a large range of noise power, that E_{fn} being for this particular host vector.

1. Set $(\tilde{x}_1, \tilde{x}_n) = n^{-1/2}(\|\mathbf{x}_{(k)}\|, \|\mathbf{x}^{(n-k)}\|)$
2. Compute $Q_0 = \tilde{x}_n \cos\theta - \tilde{x}_1 \sin\theta$.
3. If $Q_0 > 0$, then the host is outside the ruff (as expected). Set $P_0 = Q_0^2$.
   - If $P < P_0$: Watermarking is not possible because P is too small to reach the detector region. Abort.
   - If $P \geq P_0$: Watermarking is possible. Numerically solve the following optimization problem:
   $$\beta^\star = \arg \max_{-\pi/2 \leq \beta \leq 0} \left(\tilde{x}_1 + \sqrt{P}\cos\beta\right)^2 \tan^2\theta - \left(\tilde{x}_n + \sqrt{P}\sin\beta\right)^2. \tag{10.13}$$
   and set $(w_1, w_2) = \sqrt{nP}(\cos\beta^\star, \sin\beta^\star)$.
   Or use a suboptimal embedding like:
   $$(w_1, w_2) = \sqrt{n}\left(\sqrt{P - P_0\cos^2\theta}, -\sqrt{P_0}\cos\theta\right). \tag{10.14}$$
4. If $Q_0 < 0$, then the host signal is already inside the detection region (this happens with probability $\mathbb{P}_{\mathsf{fp}}$). Numerically solve the optimization problem (10.13) or use a suboptimal embedding like:
   $$(w_1, w_2) = (\sqrt{nP}, 0). \tag{10.15}$$
5. The watermark vector to be added to $\mathbf{x}$ is given by
   $$\mathbf{w}(\mathbf{x}) = w_1 \frac{\mathbf{x}_{(k)}}{\|\mathbf{x}_{(k)}\|} + w_n \frac{\mathbf{x}^{(n-k)}}{\|\mathbf{x}^{(n-k)}\|} \tag{10.16}$$

Fig. 10.5 shows that there is little difference between the suboptimal embedding and numerical solution of (10.13).

10.1.3 Right endpoint E_{fp}^R

This section studies the right endpoint of the characteristic. For a given setup $(P, \sigma_X, \sigma_Z, \alpha)$ and a given embedding, we now look at inequality (10.4) as a constraint on the half angle θ: starting from its upper limit $\pi/2 - \alpha$, we decrease θ until (10.4) is true producing $E_{\mathsf{fn}} = 0$. The ruff becomes so narrow that watermarking is no longer robust. This critical half angle is then translated into the error exponent E_{fp}^R thanks to (9.22). Again, we consider the hypercone detector as an interesting special case as it leads to a closed-form expression. For the ruff detector scheme, we have to resort to a numerical solver.

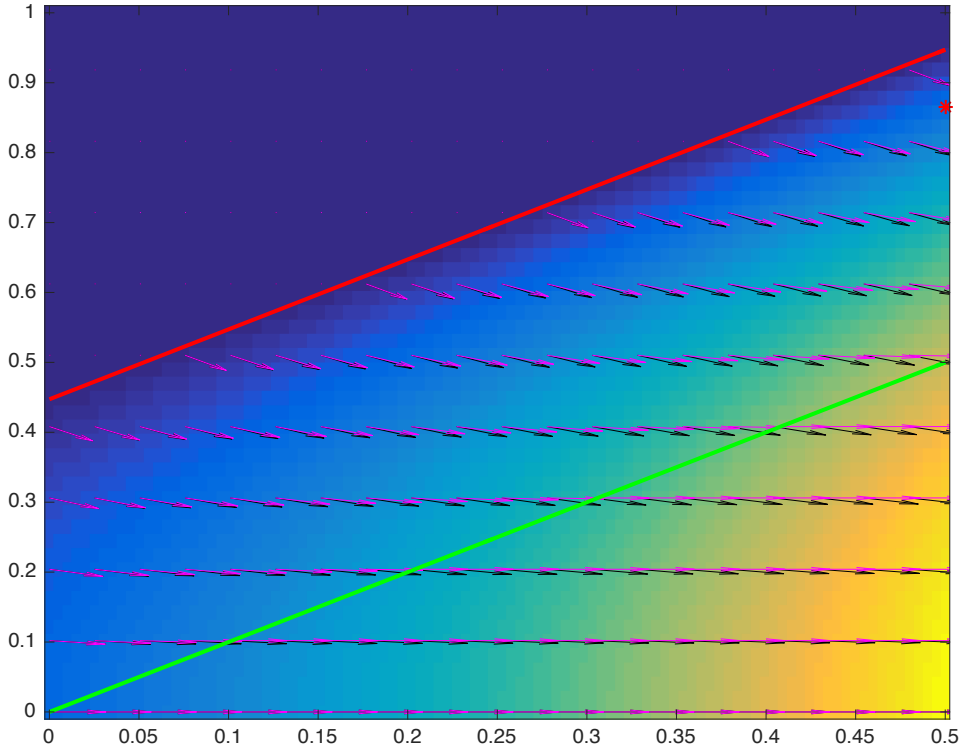


Figure 10.5: Embeddings in practice in the plane $(\tilde{x}_1, \tilde{x}_n)$. The embedding for the vector $(\tilde{x}_1, \tilde{x}_n)$ is shown as a vector $(\tilde{w}_1, \tilde{w}_n)$ starting at point $(\tilde{x}_1, \tilde{x}_n)$. The watermark vectors have all been scaled down to make the figure more visible. There is little difference in between the numerical solution of (10.13) in black and the suboptimal embedding (10.14) in magenta. The green line is the intersection between the boundary of the ruff and the plane $(\tilde{x}_1, \tilde{x}_n)$. The red line is the intersection between the boundary region of the embedding region (the rolling ball smoothing of the detection region) and the plane $(\tilde{x}_1, \tilde{x}_n)$. The colormap reflects the maximum value of σ_Z^2 for which $E_{\mathsf{fn}} > 0$, for the particular host realization $(\tilde{x}_1, \tilde{x}_n)$.

Hypercone detector ($\rho = 0$)**.**

Whenever the optimal embedding (10.7) is possible (*i.e.* if $P > P_0$), inequality (10.4) implies that error exponent E_{fn} is not null if:

$$\sigma_X^2 \sin^4\theta + (P + \sigma_Z^2 - \sigma_X^2)\sin^2\theta - \sigma_Z^2 \geq 0. \tag{10.17}$$

A special case is $\sigma_Z^2 = 0$ and $P \geq \sigma_X^2$: the above inequality always holds which means that E_{fn} is not null for any the angle of the hypercone $\theta \in (0, \pi/2]$, and thus for any value of E_{fp}. This was shown in (9.20) and [143, 99].

Yet in the noisy scenario, (10.17) is a polynomial of degree two w.r.t. $\xi := \sin^2\theta$ which has two roots ξ_- and ξ_+ s.t. $\xi_- < 0 < \xi_+ < 1$:

$$\xi_+ := \frac{\sqrt{(P + \sigma_Z^2 - \sigma_X^2)^2 + 4\sigma_X^2\sigma_Z^2} - (P + \sigma_Z^2 - \sigma_X^2)}{2\sigma_X^2} (> 0) \tag{10.18}$$

$$= 1 - \frac{(P + \sigma_Z^2 + \sigma_X^2) - \sqrt{(P + \sigma_Z^2 + \sigma_X^2)^2 - 4P\sigma_X^2}}{2\sigma_X^2} (< 1) \tag{10.19}$$

This polynomial takes positive values outside the interval $[\xi_-, \xi_+]$. This means that $E_{\mathsf{fn}} > 0$ if $\xi_+ < \sin^2\theta \leq 1$. This translates into the following right endpoint:

$$E_{\mathsf{fp}}^R = -\log\sqrt{\xi_+} = -\frac{1}{2}\log\frac{\sqrt{(P + \sigma_Z^2 - \sigma_X^2)^2 + 4\sigma_X^2\sigma_Z^2} - (P + \sigma_Z^2 - \sigma_X^2)}{2\sigma_X^2}. \tag{10.20}$$

Again, as $\sigma_Z^2 \to 0$ (*i.e.* in the noiseless setup), this E_{fp}^R tends to $+\infty$ if $P \geq \sigma_X^2$, or to $-1/2\log(1 - P/\sigma_X^2)$ if $P < \sigma_X^2$. Expression (10.20) is thus compliant with the results of [99, Sect. V-C].

Ruff detector ($\rho > 0$)**.**

We follow the same line. For a given ρ and embedding strategy, define the function $J(\theta) := H(\tilde{w}_1, \tilde{w}_n)$ over the interval $[0, \pi/2 - \alpha]$ (where $H(\cdot,\cdot)$ is defined by (10.5), and the embedding is chosen as (10.10)). Function $J(\cdot)$ is a continuous function s.t. $J(0) \leq 0$ and $\lim_{\theta\to\pi/2-\alpha} J(\theta) = +\infty$. Starting from the half-angle 0, θ is increased until $J(\theta) = \sigma_Z^2$. That critical value of θ is injected in (9.22) to compute E_{fp}^R.

Fig. 10.6 shows E_{fp}^R as a function of ρ and σ_Z^2. The best value of ρ maximizing E_{fp}^R for a given σ_Z^2 goes from 1 in the noiseless scenario down to almost 0 in the noisy scenario when σ_Z^2 is strong. The ruff detector can perform better than the hypercone scheme if ρ is properly tuned. This is especially true when σ_Z^2 is small. However, the setting of ρ is risky and a wrong choice can lead to an E_{fp}^R even lower than the lower bound as depicted in Fig. 10.7. This last figure also shows that even in the best setting, the ruff detector is well below the upper bound in terms of E_{fp}^R.

10.1.4 Robustness at operating point $E_{\mathsf{fp}} = E$

The designer operates at $E_{\mathsf{fp}} = E > 0$. In the hypercone detector, this fixes the half angle to $\theta = \theta(E)$ with $\theta(E) := \arcsin(e^{-E})$ due to (8.8). In the ruff detector, E_{fp} as defined in (9.22) is a decreasing continuous function of θ with $\lim_{\theta\to 0} E_{\mathsf{fp}} = +\infty$ and $\lim_{\theta\to\pi/2-\alpha} E_{\mathsf{fp}} = 0$. Therefore,

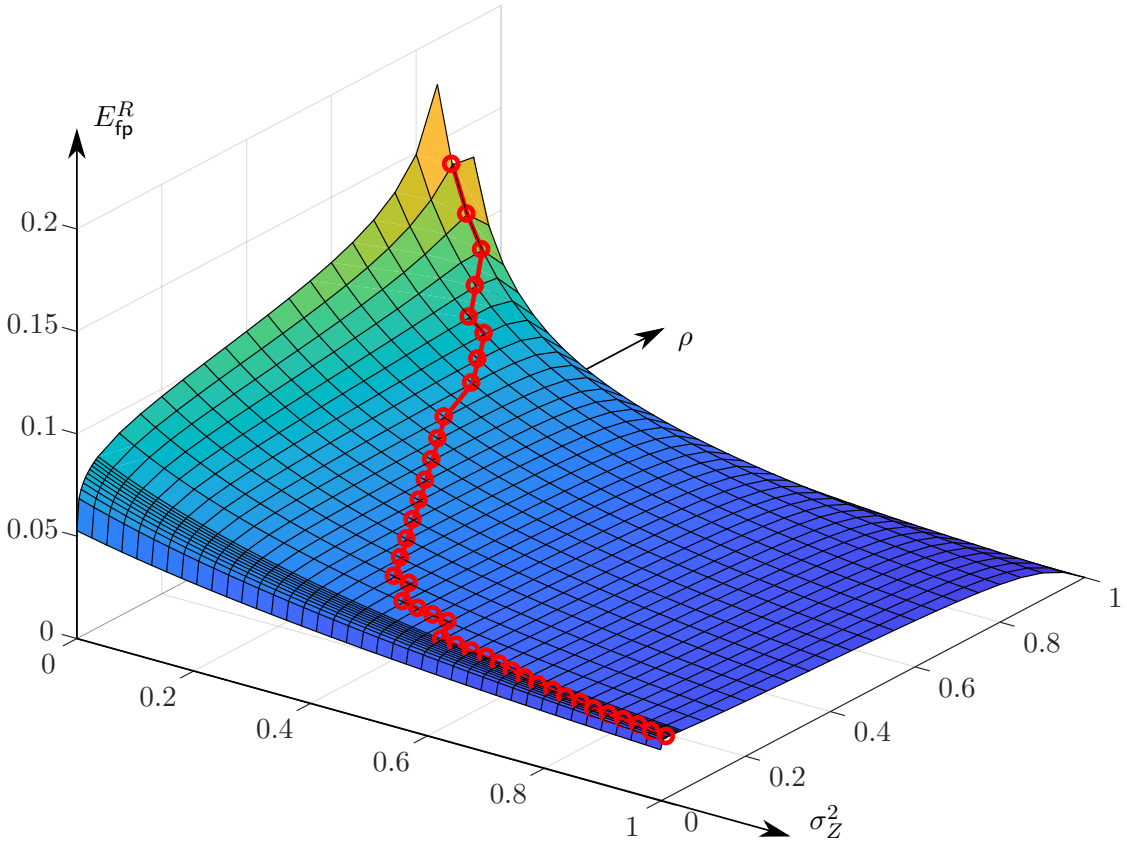


Figure 10.6: E^R_{fp} of the ruff detection as a function of σ^2_Z and ρ ($\sigma^2_X = 1$ and $P = 0.1$). The red line shows the maximum of this quantity over ρ for a fixed σ^2_Z.

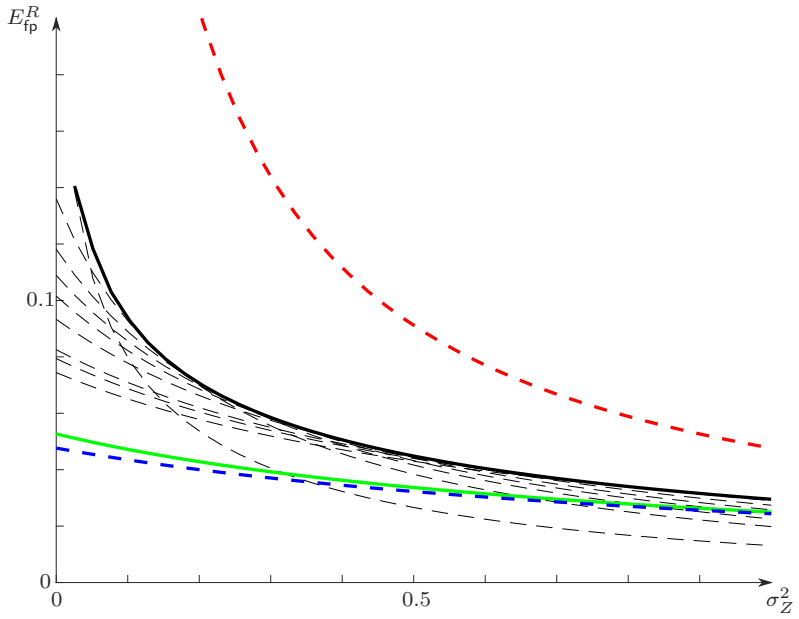


Figure 10.7: E_{fp}^R of the ruff detection as a function of σ_Z^2 ($\sigma_X^2 = 1$ and $P = 0.1$). The dotted lines shows this function for some values of ρ. The black line is E_{fp}^R for the best values of ρ knowing σ_Z^2, *i.e.* the enveloppe of the family of curves parametrized by ρ. E_{fp}^R for the hypercone detection (*i.e.* $\rho = 0$). The upper and lower bounds corresponding to the hypercone detector in Case 3 (*i.e.* non blind detector with $N = \sigma_Z^2$) and in Case 1 (*i.e.* non side-informed embeddder and blind detector with $N = \sigma_Z^2 + \sigma_X^2$) respectively (see (8.14)).

for $E > 0$, there exists $\theta(E) \in (0, \pi/2 - \alpha)$ s.t. $E_{\mathsf{fp}} = E$. This value is easily estimated numerically with a binary search.

The watermark is deemed robust if $E_{\mathsf{fn}} > 0$ for a given setting $(P, \sigma_X^2, \sigma_Z^2, E)$, which implies that $E_{\mathsf{fp}}^R > E$. Defining $\bar{\sigma_Z}^2$ as the noise power s.t. $E_{\mathsf{fp}}^R = E$, the watermark is then robust for $\sigma_Z^2 \in [0, \bar{\sigma_Z}^2)$.

Hypercone detector

The expression (10.20) shows that E_{fp}^R is a decreasing function starting at $-1/2 \log(1 - P/\sigma_X^2)$ for $\sigma_Z^2 = 0$ (unless $P \geq \sigma_X^2$, then $\lim_{\sigma_Z^2 \to 0} E_{\mathsf{fp}}^R = +\infty$), down to 0 as $\sigma_Z^2 \to +\infty$. In the practical cases where $P < \sigma_X^2$, $\bar{\sigma_Z}^2 = 0$ if $E > -1/2 \log(1 - P/\sigma_X^2)$. Otherwise, reversing (10.20) leads to:

$$\bar{\sigma_Z}^2 = \left| \frac{P}{e^{2E} - 1} - \sigma_X^2 e^{-2E} \right|_+ = \left| \left(\frac{P}{e^{2E} - 1} - \sigma_X^2 \right) + (1 - e^{-2E}) \sigma_X^2 \right|_+ \tag{10.21}$$

The last equality shows that $\bar{\sigma_Z}^2$ is bigger or equal to (8.14), *i.e.* the maximum robustness of the hypercone detector under case 1. Yet, this advantage vanishes when $E \to 0$ as depicted in Fig. 10.9.

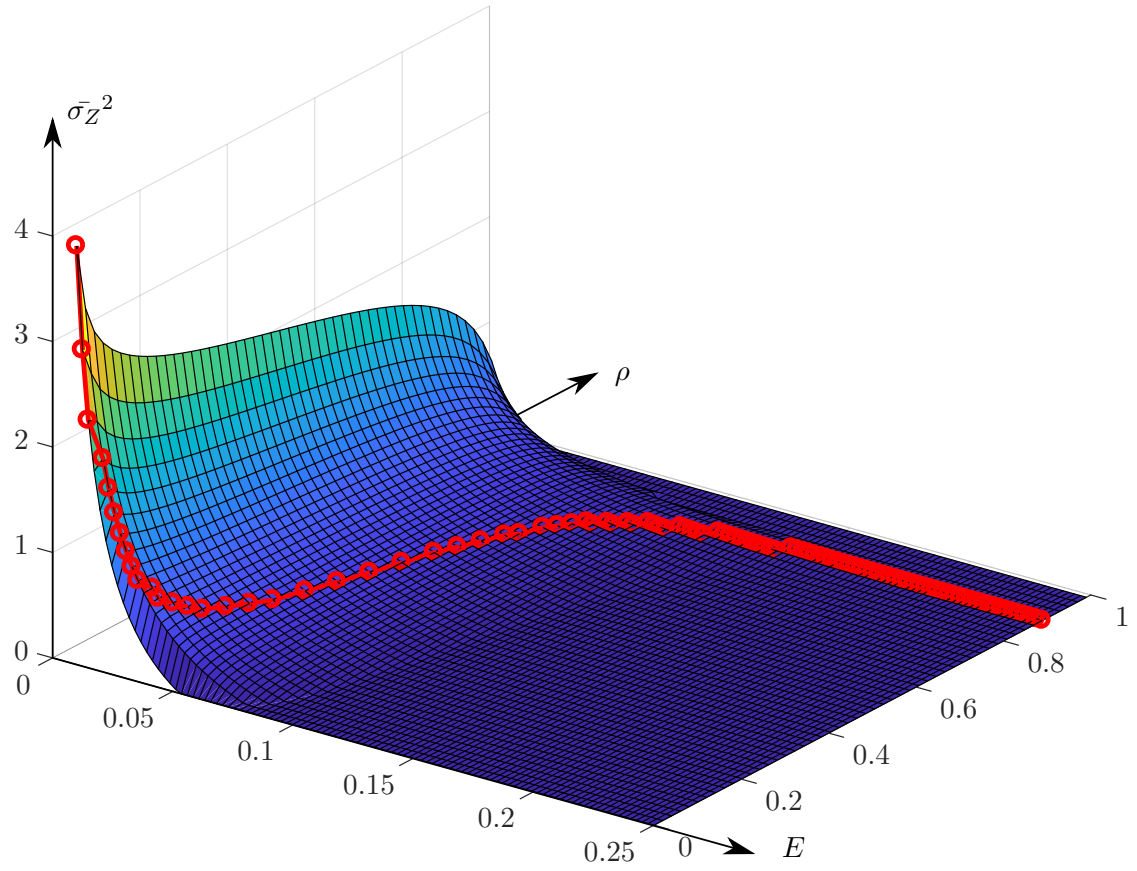


Figure 10.8: $\bar{\sigma_Z}^2$ for the ruff detection as a function of E and ρ ($\sigma_X^2 = 1$ and $P = 0.1$). The red line shows the maximum of this quantity over ρ for a fixed E. Embedding (10.10) was used.

Ruff detector

For a given ρ, operating at $E_{\mathsf{fp}} = E$ fixes the half angle $\theta = \theta(E)$ (found numerically). Reversing (10.4) leads to

$$\bar{\sigma_Z}^2 = \frac{1}{\cos^2\theta(E) - \sin^2\alpha} \left|(\sigma_X \sin\alpha + \tilde{w}_1)^2 \sin^2\theta(E) - (\sigma_X \cos\alpha + \tilde{w}_n)^2 \cos^2\theta(E)\right|_+ \qquad (10.22)$$

where the watermark embedding is encoded by the expressions of $(\tilde{w}_1, \tilde{w}_n)$ (for instance (10.10)).

Figures 10.8 and 10.9 show that the setting of parameter ρ is an issue. In Fig. 10.8, the best value for ρ converges to 0 when E tends to 0, and to a value closer to 1 for large E. Note that $P = 0.1$ in this figure. A similar behavior is observed for different value of P. Indeed, the smaller P is, the steeper is the variation of the best ρ. If the designer is looking for extreme robustness, the best choice is to operate at a very small E with $\rho = 0$. The price to be paid is the complexity as the vectors need to be very long. Fig. 10.8 shows that $\bar{\sigma_Z}^2$ can be as large as 3, *i.e.* 30 times bigger than P in this example. For a less extreme scenario, a good choice is to set ρ to a small value.

Fig. 10.9 shows that a proper tuning of ρ increases a lot $\bar{\sigma_Z}^2$ compared to the hypercone detector. However the difference vanishes for extremely small E. The setting of ρ is also risky in the sense that a too large ρ may perform worse (in term of $\bar{\sigma_Z}^2$) than the lower bound, *i.e.* the hypercone detector in case 1.

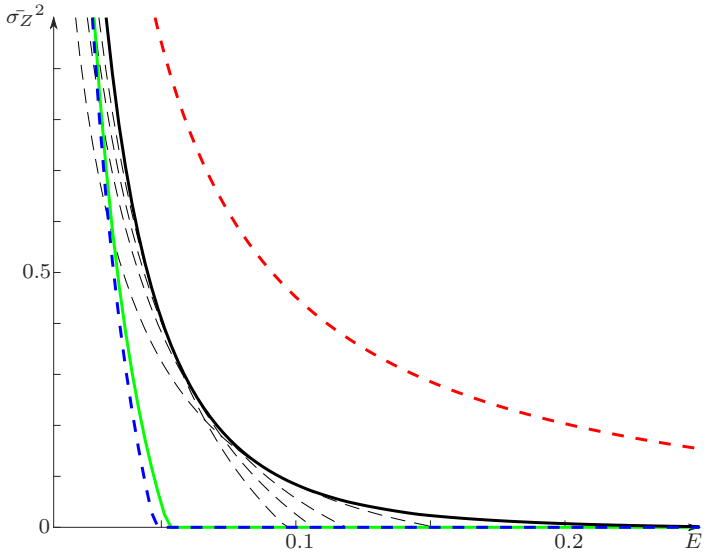


Figure 10.9: $\bar{\sigma_Z}^2$ for the ruff detection as a function of E ($\sigma_X^2 = 1$ and $P = 0.1$). The dashed lines shows this function for some values of ρ. The black line displays $\bar{\sigma_Z}^2$ for the best value of ρ, *i.e.* the enveloppe of the family of curves parametrized by ρ. The green line shows $\bar{\sigma_Z}^2$ for the hypercone detection (*i.e.* $\rho = 0$). The upper and lower bounds corresponding to the hypercone detector in case 3 (*i.e.* non blind detector with $N = \sigma_Z^2$) in red and in case 1 (*i.e.* non side-informed embeddder and blind detector with $N = \sigma_Z^2 + \sigma_X^2$) in blue respectively (8.14).

10.1.5 False negative error exponent E_{fn}

This chapter has been focusing so far on the right-side of the characteristic by investigating when E_{fn} vanishes. This section now investigates the case where $E_{\mathsf{fn}} > 0$.

Left endpoint E_{fn}^L

The characteristic reaches the left endpoint for $E_{\mathsf{fp}} = 0$, *i.e.* when the half angle θ goes up to $\pi/2 - \alpha$. In that case, the general optimization problem (10.1) becomes easy to be solved. We can set $\tilde{q}_1 = \sigma_Z\sqrt{\rho}$ and $\tilde{q}_n = \sigma_Z\sqrt{1-\rho}$ (this cancels their $S(\cdot)$ functions in (10.1)) for free because they disappear from the inequality (10.2). By the same token, nothing prevents us from setting $(\tilde{x}_1, \tilde{x}_n) = \sigma_X(\sqrt{\rho}, \sqrt{1-\rho})$ as they also disappear from the inequality. In the end, only $(\tilde{z}_1, \tilde{z}_n)$ remain and exponent E_{fn}^L is now defined via this simple optimization problem:

$$E_{\mathsf{fn}}^L = \min_{(\tilde{z}_1+\tilde{w}_1)\leq(\tilde{z}_n+\tilde{w}_n)\tan\alpha} \frac{\tilde{z}_1^2 + \tilde{z}_n^2}{2\sigma_Z^2}. \tag{10.23}$$

Its solution is

$$E_{\mathsf{fn}}^L = \frac{(\tilde{w}_1 - \tilde{w}_n \tan\alpha)^2 \cos^4\alpha}{2\sigma_Z^2}. \tag{10.24}$$

The embedding maximizing E_{fn}^L is $(\tilde{w}_1, \tilde{w}_n) = (\sqrt{P}, 0)$, leading to

$$E_{\mathsf{fn}}^L = \frac{P}{2\sigma_Z^2}\cos^2\alpha. \tag{10.25}$$

Side-information succeeds to make E_{fn}^L independent of σ_X^2. But, that E_{fn}^L is lower than $P/2\sigma_Z^2$ which is the upper bound given by the left endpoint (8.12) in Case 3 (the detector knows the

host). Therefore, we cannot really say that side information cancels the presence host, unless $\cos\alpha = 1$ (hypercone detector). We clearly see that the role of ρ is to trade E_{fn}^L for E_{fp}^R.

Upper bounds $\bar{E}_{\mathsf{fn}}$

The strategy to find upper bounds is to cast the minimization problem (10.1) onto subsets of $\tilde{\mathcal{D}}$. Here are some results for the following subsets.

First, by setting $\tilde{x}_n = \tilde{x}_1 \tan\theta$ and $\tilde{q}_n = \tilde{q}_1 \tan\theta$, the constraint on the other variables is simply $(\tilde{z}_1 + \tilde{w}_1)\tan\theta \leq \tilde{z}_n + \tilde{w}_n$. This gives

$$\bar{E}_{\mathrm{fn}} = \frac{Q^2 \cos^2\theta}{2\sigma_Z^2} + 2S\left(\frac{\cos^2\theta}{\sin^2\alpha}\right)\sin^2\alpha + 2S\left(\frac{\sin^2\theta}{\cos^2\alpha}\right)\cos^2\alpha, \tag{10.26}$$

with $Q := \tilde{w}_1 \tan\theta - \tilde{w}_n$. This upper bound is tight at the left hand side of the characteristic because $\bar{E}_{\mathrm{fn}}^L = E_{\mathsf{fn}}^L$ at least for the embedding rule (10.12).

Another subset fixes

$$\tilde{x}_1 = \sqrt{\rho}\sigma_X, \tilde{x}_n = \sqrt{1-\rho}\sigma_X, \tilde{q}_1 = \sqrt{\rho}\sigma_Z, \tilde{q}_n = \sqrt{1-\rho}\sigma_Z, \tag{10.27}$$

which implies that $(\tilde{z}_1, \tilde{z}_2)$ satisfies:

$$G(\tilde{z}_1, \tilde{z}_n) \leq \sigma_Z^2\left(1 - \frac{\sin^2\alpha}{\cos^2\theta}\right) \quad \text{with} \tag{10.28}$$

$$G(\tilde{z}_1, \tilde{z}_2) := (\sigma_X \sin\alpha + \tilde{w}_1 + \tilde{z}_1)^2 \tan^2\theta - (\sigma_X \cos\alpha + \tilde{w}_n + \tilde{z}_n)^2. \tag{10.29}$$

This last equation defines the region of $\mathbb{R}^2$ mapping $(\tilde{z}_1, \tilde{z}_n)$ in between the two branches of the hyperbola defined by (10.28) with equality. The upper bound $\bar{E}'_{\mathrm{fn}}$ turns out to have a simple expression:

$$\bar{E}'_{\mathrm{fn}} = \min \frac{\tilde{z}_1^2 + \tilde{z}_n^2}{2\sigma_Z^2}. \tag{10.30}$$

We rediscover the three cases detailed in Sect. 10.1.2 and depicted in Fig. 10.10 in the plane $(\tilde{z}_1, \tilde{z}_n)$:

- If $G(0,0) < 0$, then $(0,0)$ is between in the asymptotes of the hyperbola, and therefore in between the branches of the hyperbola for any σ_Z^2. $\bar{E}_{\mathrm{fn}} = 0$ and E_{fn} as well even in the noiseless scenario.
- If $0 \leq G(0,0) \leq \sigma_Z^2(1 - {}^{\sin^2\alpha}/_{\cos^2\theta})$, then $(0,0)$ lies in between the right branches of the hyperbola and its asymptotes. $\bar{E}_{\mathrm{fn}} = E_{\mathsf{fn}} = 0$ for that particular σ_Z^2.
- If $G(0,0) > \sigma_Z^2(1 - {}^{\sin^2\alpha}/_{\cos^2\theta})$, then $(0,0)$ is located on the right of the right hyperbola. $\bar{E}_{\mathrm{fn}} > 0$ and E_{fn} might be strictly positive.

Indeed, the last statement can be more strongly asserted. The minimizer of the problem in the subset $\tilde{\mathcal{D}}'$ yields a strictly positive $\bar{E}_{\mathrm{fn}}$. This minimum might be the true minimum over $\tilde{\mathcal{D}}$. If it is not, then $(\tilde{x}_1, \tilde{x}_n, \tilde{q}_1, \tilde{q}_n) \neq (\sigma_X\sqrt{\rho}, \sigma_X\sqrt{1-\rho}, \sigma_Z\sqrt{\rho}, \sigma_Z\sqrt{1-\rho})$ and E_{fn} is strictly positive due to the property of $S(\cdot)$.

As depicted in Fig. 10.10, the goal is to find the minimum radius for which the ball touches the hyperbola. It amounts to project the origin $(0,0)$ onto the hyperbola. Again, there is no close

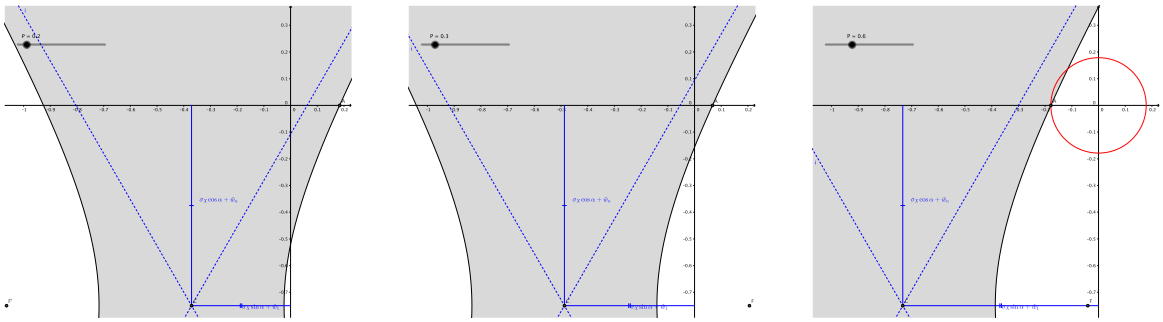


Figure 10.10: Conditions for $\bar{E}_{\text{fn}} \geq 0$ in the plane $(\tilde{z}_1, \tilde{z}_n)$. Setup: $\sigma_X = 1$, $\sigma_Z = 0.6$, and $\theta = \pi/3$. $\tilde{\mathcal{D}}$ defines the gray area. The hyperbola is shifted to the left when P increases. Left: $P = 0.2$, the origin $(0,0)$ is inside $\tilde{\mathcal{D}}$ and $\bar{E}_{\text{fn}} = 0$ for any value of σ_Z because the origin lies in between the asymptotes. Middle: $P = 0.3$, $(0,0) \in \tilde{\mathcal{D}}$ and $\bar{E}_{\text{fn}} = 0$ for this particular value of σ_Z. Since the origin lies in between the asymptote and the hyperbola, it gets out of $\tilde{\mathcal{D}}$ for small value of σ_Z. Right: $P = 0.6$, the origin is outside $\tilde{\mathcal{D}}$ and $\bar{E}_{\text{fn}} > 0$. The red circle of center $(0,0)$ and passing through A has points which belongs to $\tilde{\mathcal{D}}$. Consequently, its radius gives birth to an upper bound of $\bar{E}_{\text{fn}}$.

form for projecting a point on an hyperbola. We approximate this by the point $\mathsf{A} = (\bar{z}_1^\star, 0)$ on the hyperbola (see Fig. 10.10). Hence, $\bar{E}'_{\text{fn}} \approx \tilde{z}_1^{\star 2}/2\sigma_Z$ with:

$$\tilde{z}_1^\star = \tan^{-1}\theta \sqrt{(\sigma_X \sin\alpha + \tilde{w}_n)^2 + \sigma_Z^2 \cos^2\alpha} - \tilde{w}_1 - \sigma_X \sin\alpha. \tag{10.31}$$

For the hypercone detector, we obtain $\bar{E}'^L_{\text{fn}} = E^L_{\mathsf{fn}}$. This approximation is tight on the left hand side of the characteristic. $\bar{E}'_{\text{fn}}$ cancels when $\tilde{z}_1^\star = 0$. For the hypercone detector (*i.e.* $\alpha = 0$), we find back the condition $P = \sigma_X^2 \cos^2\theta + \sigma_Z^2 \tan^{-2}\theta$ as previously stated in Sect. 10.1.1. In other words, this approximation is tight on both endpoints for the hypercone detection. For the ruff detector, it is more complicated to evaluate the tightness of the upper bound.

Nevertheless, even for the hypercone detector, this upper bound is not tight for small σ_Z^2, *i.e.* in the high SNR regime: $\lim_{\sigma_Z^2 \to 0} \bar{E}_{\text{fn}} = \infty$ whereas this limit of E_{fn} was given in Chapter 9.

10.2 Voronoï Modulation with side information

Section 8.4 introduces the Voronoï modulation as a communication scheme which may be capacity achieving or best reliability achieving (at least for $R > R_2$) if its scaling factor α is set to the proper value α_{MMSE} or α_{opt} respectively. Section 9.3.1 shows how to take into account side information at the embedding of a Voronoï modulation scheme in the noiseless scenario. This section blends the two results. The main sources of inspiration are [135, 166].

10.2.1 First version: scaled lattice

Codeword $\mathbf{v}_m$ is transmitted thanks to the following embedding:

$$\mathbf{W} = (\mathbf{v}_m - \alpha\mathbf{X} + \mathbf{U}) \mod \Lambda_2. \tag{10.32}$$

Compared to Sect. 9.3.1, the side information $\mathbf{X}$ is now scaled by the factor α (or that factor was set to 1 in the noiseless scenario of Sect. 9.3.1). Thanks to the dither $\mathbf{U} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$, $\mathbf{W} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$ so that the shaping lattice Λ_2 controls the embedding distortion (be it strict or on expectation). The dither also makes $\mathbf{W}$ independent of $\mathbf{v}_m$.

As in Sect. 8.4, upon receiving signal $\mathbf{R}$, the decoder first makes a linear estimation of $\mathbf{W}$:

$$\hat{\mathbf{W}} = \alpha\mathbf{R} = \alpha(\mathbf{X} + \mathbf{W} + \mathbf{Z}) = \mathbf{v}_m + \mathbf{U} + \alpha\mathbf{Z} + (\alpha - 1)\mathbf{W} - Q_{\Lambda_2}(\mathbf{v}_m - \alpha\mathbf{X} + \mathbf{U}). \tag{10.33}$$

By removing the dither and taking this modulo Λ_2, we obtain:

$$\tilde{\mathbf{Y}} = (\hat{\mathbf{W}} - \mathbf{U}) \mod \Lambda_2 = (\mathbf{v}_m + \alpha\mathbf{Z} + (\alpha - 1)\mathbf{W}) \mod \Lambda_2. \tag{10.34}$$

This is exactly the same expression binding $\mathbf{v}_m$ and $\tilde{\mathbf{Y}}$ as in Sect. 8.4, *i.e.* the equivalent modulo-Λ channel. This expression also shows that in the noiseless scenario where $\mathbf{Z} = \mathbf{0}$, $\alpha = 1$ is indeed best choice as done in Sect. 9.3.1.

For a decoding purpose, $\mathbf{v}_m$ is a codeword from a random code where each $M = e^{nR}$ codewords are uniformly distributed over $\mathcal{V}(\Lambda_2)$. This is for instance the codewords of a fine coding lattice Λ_1 inside $\mathcal{V}(\Lambda_2)$, and the final decoding step is to map $\tilde{\mathbf{Y}}$ onto a codeword: $\mathbf{v}_{\hat{m}} = Q_{\Lambda_1}(\tilde{\mathbf{Y}})$. The capacity and the reliability function are given by (8.23) in Sect. 8.4 replacing $N = \sigma_X^2 + \sigma_Z^2$ by $N = \sigma_Z^2$.

For a detection purpose, we choose $\mathbf{v}_m = \mathbf{0}_n$ and the received vector is deemed watermarked if $\|\tilde{\mathbf{Y}}\| < \sqrt{nb}$. The detection region is thus the set of balls centered on the points of Λ_2 [130, Eq. (26)]. Under $\mathcal{H}_0$, $\tilde{\mathbf{Y}} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$ so that (provided $\sqrt{nb}$ is smaller than the packing radius of Λ_2):

$$\mathbb{P}_{\mathsf{fp}} = \frac{|\mathcal{B}(\sqrt{nb})|}{|\mathcal{V}(\Lambda_2)|} = \left(\frac{n}{n+2}\frac{G(\Lambda_2)}{G(\mathcal{S}_n)}\frac{b}{\sigma^2(\Lambda_2)}\right)^{n/2}, \tag{10.35}$$

$$E_{\mathsf{fp}} = \frac{1}{2}\log\frac{\sigma^2(\Lambda_2)}{b}. \tag{10.36}$$

Appendix 16.4 proves the expression of E_{fn}, the most important fact being that both methods (decoding or detection) yield the same characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$. The detection method is less complex as it saves the quantization onto Λ_1. When the embedder and the detector both know σ_Z^2 and P, they can use the optimal value α_{opt} (see (8.25) with $A = \sqrt{P/\sigma_Z^2}$). Fig. 10.11:left compares E_{fp}^R with the upper and lower bounds which are provided by the Neyman-Pearson detector since obliviousness does not hold here.

Again, we face the same limitations as pointed in Sect. 8.4:

- The decoder/detector needs the occurence of $\mathbf{U}$ randomly generated at each embedding. The flat-host assumption is a way to get rid off the dither: the size $\mathcal{V}(\Lambda_2)$ is small compared to the scale of the host so that $\mathbf{W} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$. This assumption also makes $\tilde{\mathbf{Y}} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$ under $\mathcal{H}_0$.
- The decoder/detector knows Λ_2 which inherently means that they are not oblivious to P.
- When the embedder and the decoder/detector are oblivious to σ_Z^2, parameter α cannot be set to α_{MMSE} nor α_{opt}.

The last comment raises the issue of setting parameter α. This is difficult from the viewpoint of E_{fp}^R: For a fixed α, E_{fp}^R gets lower than the lower bound when the noise power is too strong (see Fig. 10.11:right), or, more annoyingly, when the embedding power is too big (see Fig. 10.12:left).

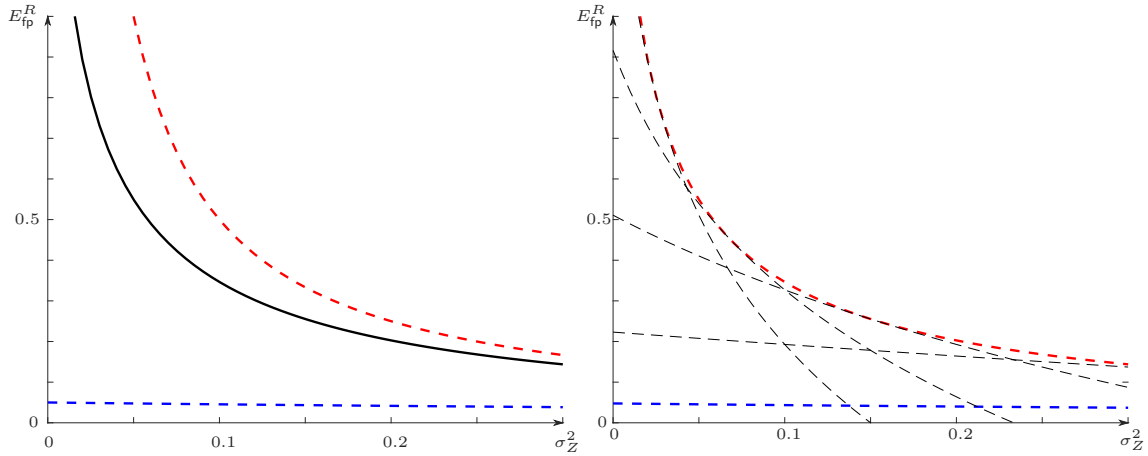


Figure 10.11: E_{fp}^R as a function of σ_Z^2 for $\sigma_X^2 = 1$, $P = 0.1$. Left: The watermarking scheme is not oblivious to N. The Voronoï modulation with $\alpha = \alpha_{\mathsf{opt}}$ (black) is compared to the upper and lower bounds (8.2) corresponding to the Neyman-Pearson detector with $N = \sigma_Z^2$ (red) and $N = \sigma_Z^2 + \sigma_X^2$ (blue) respectively. Right: The watermarking scheme is oblivious to N. The Voronoï modulation (dashed black) works with a fixed α (here $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$). It is compared to the upper and lower bounds corresponding to hypercone detector with $N = \sigma_Z^2$ (red) and $N = \sigma_Z^2 + \sigma_X^2$ (blue) respectively.

Yet, the problem is simpler from the the viewpoint of $\bar{\sigma_Z}^2$: Suppose that we wish to operate at false positive error exponent $E_{\mathsf{fp}} = E > 0$ (for a given level $\mathbb{P}_{\mathsf{fp}}$, this means that $n \approx -\log \mathbb{P}_{\mathsf{fp}}/E$). We would like $E_{\mathsf{fp}}^{(R)}$ to be greater or equal to E on a large range of noise power. Here, the inequality

$$E_{\mathsf{fp}}^{(R)} = \left| \frac{1}{2} \log \left(\frac{P}{\alpha^2 \sigma_Z^2 + (1-\alpha)^2 P} \right) \right|_+ \geq E \tag{10.37}$$

implies that the noise power ratio is weak enough:

$$\sigma_Z^2 \leq P \left| \frac{e^{-2E} - (1-\alpha)^2}{\alpha^2} \right|_+ . \tag{10.38}$$

This upper limit is maximized for $\alpha^\star = 1 - e^{-2E}$, which provides robustness (in the sense that $E_{\mathsf{fn}} > 0$ at $E_{\mathsf{fp}} = E$) whenever $\sigma_Z^2 < \bar{\sigma_Z}^2$:

$$\bar{\sigma_Z}^2 := \frac{P}{e^{2E} - 1}. \tag{10.39}$$

This critical noise power value is indeed optimal. Being oblivious to N, the upper bound of E_{fp}^R is the capacity of the channel in Case 3 (*i.e.* without host interference) $C = {}^1\!/_2 \log(1 + {}^P\!/_{\sigma_Z^2})$. This quantity is bigger than E for $\sigma_Z^2 < {}^P\!/_{(e^{2E}-1)}$ (See Fig. 10.13).

10.2.2 Second version: fixed lattice

This section works on the detection version of the Voronoï modulation with a fixed lattice Λ_2: The detection is now oblivious to P. This tackles applications where P varies from one content to another. Quantization onto this lattice induces a mean squared error $\sigma^2(\Lambda_2)$. The embedding

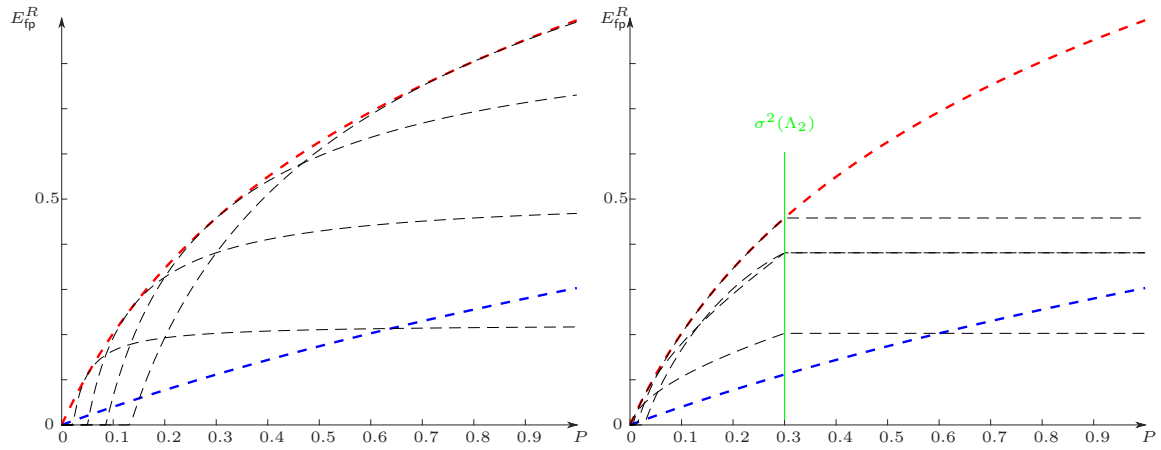


Figure 10.12: E_{fp}^R as a function of P for $\sigma_X^2 = 1$ and $\sigma_Z^2 = 0.2$. Left: The Voronoï modulation (dashed black) works with a scaled lattice Λ_2 (version 1) and a fixed α (here $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$). It is compared to the upper and lower bounds corresponding to hypercone detector with $N = \sigma_Z^2$ (red) and $N = \sigma_Z^2 + \sigma_X^2$ (blue) respectively. Right: The Voronoï modulation (dashed black) works with a fixed lattice Λ_2 (version 2 with $\sigma^2(\Lambda_2) = 0.3$) and a fixed α (here $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$).

scales the quantization error if $P < \sigma^2(\Lambda_2)$:

$$\mathbf{W} = \beta \mathbf{W}_0, \quad \text{with } \mathbf{W}_0 := ((\mathbf{v}_m - \alpha \mathbf{X}) \mod \Lambda_2) \quad \text{and} \tag{10.40}$$

$$\beta = \min\left(\sqrt{\frac{P}{\sigma^2(\Lambda_2)}}, 1\right). \tag{10.41}$$

As in Sect. 9.3.2, the watermark signal has a clipped power: $n^{-1}\mathbb{E}(\|\mathbf{W}\|^2) \leq \sigma^2(\Lambda_2)$. For content to be watermarked with $P > \sigma^2(\Lambda_2)$, the whole embedding distortion budget is not consumed. Therefore, Λ_2 shall not be too fine to make this waste rarely effective. Note also that the dither has been removed and the following holds under the flat host assumption: Λ_2 shall not be too coarse. Since the watermark power is usually small w.r.t. the host's power, there is some hope to find a trade-off between these two conflicting requirements.

For the detection based on balls centered on the centroids of Λ_2, E_{fp} is given by (10.36), whereas under $\mathcal{H}_1$:

$$\tilde{\mathbf{Y}} = \hat{\mathbf{W}} \mod \Lambda_2 = (\mathbf{v}_m + \alpha \mathbf{Z} + (\alpha\beta - 1)\mathbf{W}_0) \mod \Lambda_2. \tag{10.42}$$

Error exponent $E_{\mathsf{fp}}^{(R)}$ is now given by (see Appendix 16.4):

$$E_{\mathsf{fp}}^{(R)} = \frac{1}{2}\left|\log \frac{\sigma^2(\Lambda_2)}{\alpha^2\sigma_Z^2 + (\alpha\beta - 1)^2\sigma^2(\Lambda_2)}\right|_+. \tag{10.43}$$

Fig. 10.12:right shows how E_{fp}^R stays constant even if P increases after $\sigma^2(\Lambda_2)$.

As done previously, for a targeted $E_{\mathsf{fp}} = E$, this watermarking scheme is robust when the noise power is lower than $\bar{\sigma_Z}^2$:

$$\bar{\sigma_Z}^2 = \sigma^2(\Lambda_2)\left|\frac{e^{-2E} - (1 - \alpha\beta)^2}{\alpha^2}\right|_+. \tag{10.44}$$

It would be a mistake to look for α maximizing this upper limit because this optimal setting depends on β which is not known at the detection side: this is the crucial difference with the

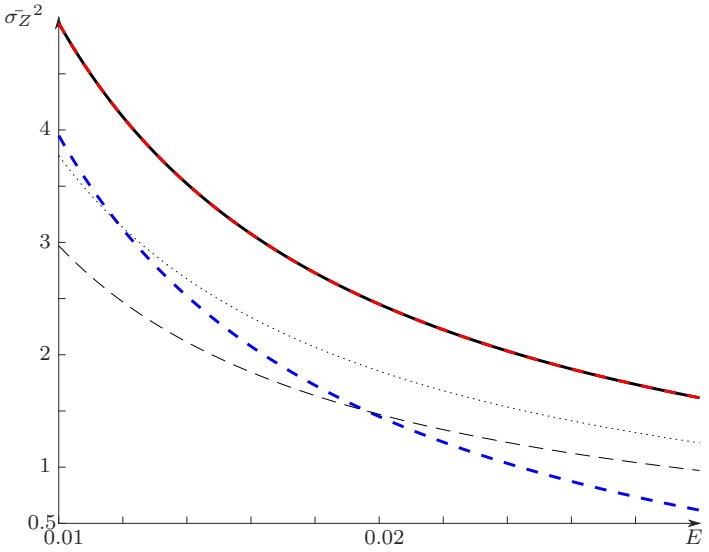


Figure 10.13: $\bar{\sigma_Z}^2$ as a function of E for $\sigma_X^2 = 1$, $P = 0.1$. The upper bound (red) and the lower bound (blue) are given by the hypercone detector under cases 1 and 3 (respectively). The solid black line corresponds to the Voronoï modulation with scaled lattice (version 1) whereas the dashed (resp. dotted) line corresponds to its fixed lattice version with $\sigma^2(\Lambda_2) = 0.06$ (resp. $\sigma^2(\Lambda_2) = 0.22$). Parameter α is set to $1 - e^{-2R}$.

previous section. It is also wrong to believe that this upper limit is an increasing function of $\sigma^2(\Lambda_2)$. It is strictly positive only when $\beta > \alpha^{-1}(1 - e^{-E})$ (if not, the watermark is not robust even in the noiseless scenario). Yet, increasing $\sigma^2(\Lambda_2)$ raises the chance that $\beta = \sqrt{P/\sigma^2(\Lambda_2)}$ is smaller than that critical value.

To state this more clearly, for a given (α, E), the watermark is robust to some extend (*i.e.* the r.h.s. of (10.44) is strictly positive)

- $P \geq \sigma^2(\Lambda_2)$ (*i.e.* $\beta = 1$): if $\alpha \geq 1 - e^{-E}$,
- $P < \sigma^2(\Lambda_2)$ (*i.e.* $\beta < 1$): if $P\alpha^2(1 - e^{-E})^{-2} > \sigma^2(\Lambda_2)$

Fig. 10.13 illustrates that the tuning of $\sigma^2(\Lambda_2)$ is a delicate issue. In this example, $\sigma^2(\Lambda_2)$ is either smaller than P so that the watermarking scheme is not spending the full embedding power budget, or $\sigma^2(\Lambda_2)$ is bigger than P so that $\beta < 1$. Both phenomena provoque a loss in robustness. Indeed, at low E, $\bar{\sigma_Z}^2$ is indeed smaller than the lower bound (*i.e.* the robustness of the hypercone detector without side-information) if E is too big.

A recommendation would be to first set $\alpha = 1 - e^{-2E}$ as in the previous section: This maximizes the robustness at full watermarking power, *i.e.* when $\beta = 1$. Then $\sigma^2(\Lambda_2)$ should be set according to the minimum value of P (or its p-quantile to guarantee an 'outage' probability).

10.3 Conclusion

When σ_Z^2 and P are known and the embedding and detection sides A remarkable fact is that the zero-bit watermarking scheme based on Voronoï modulation with a scaled lattice (Sect 10.2.1) produces a characteristic which is independent of the host power. Moreover, thanks to a clever side-informed embedding, it matches the characteristic without host interference replacing

$N = \sigma_X^2 + \sigma_Z^2$ by $N = \sigma_Z^2$. This amazing discovery is due to Tie Liu, Pierre Moulin and Ralf Koetter [131], strengthening a preliminary result on the capacity by Uri Erez and Ram Zamir [110].

At first sight, this schemes delivers the promise of M. Costa: With the terminology of Sect. 7.5, the performance of Case 2 (side-informed embedder) reaches the upper bound of Case 3 (non-blind detector). Not knowing the host at the detection side does not hurt thanks to a side-informed embedding.

This is indeed a mistake for the following reason: The detector is not oblivious to σ_Z^2. Under the same assumption, the upper bound is indeed given by the optimal Neyman-Pearson detector under Case 3 of Sect. 8.1 (this scheme needs as well σ_Z^2 to set the threshold enforcing a given false positive probability). The Voronoï modulation does not match this upper bound. In terms of the right endpoint: $E_{\mathsf{fp}}^R = 1/2 \log(1 + P/\sigma_Z^2) \leq P/2\sigma_Z^2$. This is especially true at high watermark to noise power ratio P/σ_Z^2 where σ_Z^2 can be small (see Fig. 10.11:left).

Conversely, this non-oblivious Voronoï modulation only matches the oblivious upper bound, which is the hypercone detector under Case 3. Yet, the hypercone detector scheme knows neither σ_Z^2 at the embedding side nor (P, σ_Z^2) at the detection side. In other words, the characteristic of the Voronoï modulation is independent to σ_X^2 not just thanks to an embedding taking into account $\mathbf{X}$, there is also the benefit of extra informations (*i.e.* the values of σ_Z^2 and P) shared on both sides. This makes the comparison between schemes unfair.

However, this statement is mitigated when $\bar{\sigma_Z}^2$ becomes the figure of merit to gauge a watermarking scheme when operating at $E_{\mathsf{fp}} = E$. The Voronoï modulation achieves $\bar{\sigma_Z}^2 = P/(e^{2E}-1) \approx P/2E - P/2$ which is only $P/2$ smaller than the optimal $P/2E$ when $E \to 0$.

When σ_Z^2 is unknown and the embedding and the detection side Obliviousness to σ_Z^2 is more relevant in practice. The drawback is that α is fixed to a value a priori not optimal for the first version with a scaled lattice. Section 10.2.1 gives a rationale on how to fix α. Its right endpoint (10.37) should be compared to the upper bound $1/2 \log(1 + P/\sigma_Z^2)$ as done in Fig. 10.11:right.

For applications where P varies from content to another and where the detection is unaware of these variations, the second version with fixed lattice is more practical. The drawback is that the whole embedding distortion budget may not be used in some cases. Its right endpoint (10.43) should be compared to the same upper bound as done in Fig 10.12.

This is indeed the only setup where the comparison with the k-dimensional ruff is fair. This comparison is summed up in Fig. 10.14. An extreme scenario is depicted on the left where the embedding distortion is very small: $P = \sigma_X^2/100$. Robustness is enforced by operating at very low E in the range $[10^{-3}, 10^{-2}]$. For the hypercone detector, a side-informed embedding does not help compared to its non side-informed embedding (*i.e.* Case 1). The ruff detector gives some improvement if E is not too small (parameter $\rho > 0$ when the black line leaves the green curve). The performance of the Voronoï modulation with a fixed lattice depends on the ratio $P/\sigma^2(\Lambda_2)$. A too small $\sigma^2(\Lambda_2)$ must be avoided. Note that even if $\sigma^2(\Lambda_2) > P$, the robustness becomes smaller than the lower bound (here, for $E < 10^{-3}$ which is not depicted in Fig. 10.14:left).

Another scenario is depicted on the right with a bigger embedding distortion: $P = \sigma_X^2/10$. We consider the range $E \in [10^{-2}, 10^{-1}]$ to match the same robustness level as for the first scenario. The same comments hold, except that the hypercone detector performs slightly better than the lower bound for large E.

The question of Costa Whatever the configuration (obliviousness w.r.t. N and/or P), it seems that side-informed embedding doesn't make zero-bit watermarking performing as well as if

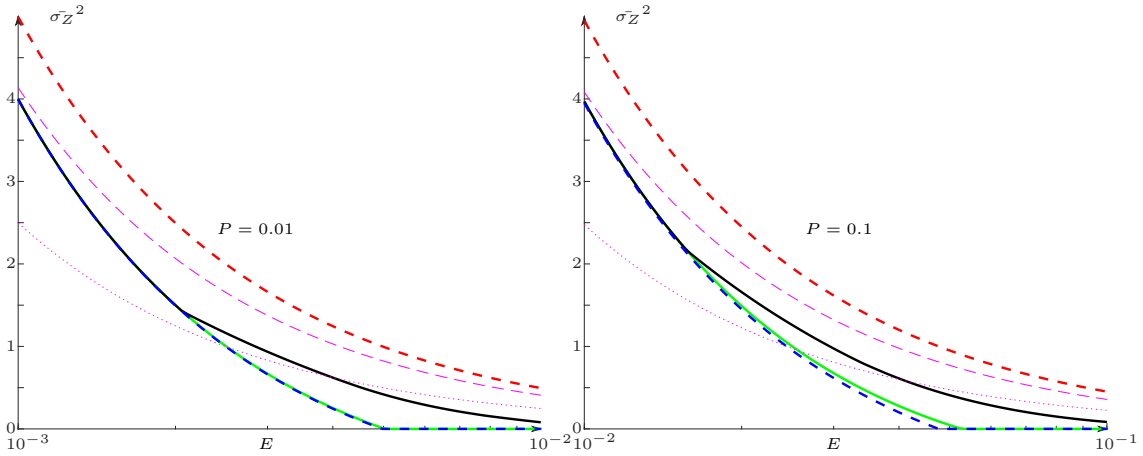


Figure 10.14: $\sigma_{\bar{Z}}{}^2$ as a function of E for $\sigma_X^2 = 1$ for oblivious schemes. Left: $P = 0.01$ and $E \in [10^{-3}, 10^{-2}]$, Right: $P = 0.1$ and $E \in [10^{-2}, 10^{-1}]$. The upper bound and the lower bound are given by the hypercone detector under Cases 1 and 3 (respectively). The green line corresponds to the hypercone detector with side-information (Case 2). The black line corresponds to the ruff detector. The Voronoï modulation with a fixed lattice with $\sigma^2(\Lambda_2) = 2P$ (dashed) and $\sigma^2(\Lambda_2) = P/2$ (dotted). Parameter α is set to $1 - e^{-2R}$.

there were no host interfering (Case 3, see Sect. 7.5). At least for the schemes considered in this habilitation, this is true when performances are gauged by the full characteristic $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$ or by the right endpoint E_{fp}^R.

The only setup where Case 2 performs almost as good as Case 3 is when

- robustness $\sigma_{\bar{Z}}{}^2$ is the figure of merit,
- the scheme is not oblivious with respect to (P, N),
- the detector operates at a very low $E_{\mathsf{fp}} = E$.

Then, with the Voronoï modulation, the loss between Cases 3 and 2 is $P/2$ at low E_{fp}, which becomes small compared to $\sigma_{\bar{Z}}{}^2$.

The Voronoï modulation and the ruff scheme have the same drawback: At $E_{\mathsf{fp}} = E$, the configuration yielding the larger $\sigma_{\bar{Z}}{}^2$ depends on P. This is a problem when P varies and the detector is oblivious to P. In other words, we did not find a rationale for setting $\sigma^2(\Lambda_2)$ for the Voronoï modulation, or for setting ρ for the ruff scheme.

The wording *"It seems that"* in the first sentence means that the above empirical conclusion is drawn from case studies. Indeed, this manuscript only reveals that the best scheme is still not known.

Chapter 11

The bad designs

This chapter lists some 'bad' designs of zero-bit watermarking schemes. They share the same frustrating fact: it is possible to compute their characteristic, but there is no way to control the level on $\mathbb{P}_{\mathsf{fp}}$ (or asymptotically E_{fp}) without extra information at the detection side (like P or N). While previous Chapter 10 focuses on practical schemes where E_{fp} was under control, this chapter deals with other 'forgotten' schemes.

This pitch is indeed a pretext because the first two schemes of Sect. 11.1 do not deliver large E^R_{fp}. The main point of this chapter is indeed Sect. 11.2 making the connection between two pieces of theory: the study of the error exponent characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$ as done in [99, 130] (and in the previous chapters), and the study of the asymptotic efficacy as presented in [23].

11.1 The forgotten schemes

11.1.1 ZATT

Chapter 9 introduces the scheme ZATT in Sect. 9.2. It provides a perfect test (under certain conditions) in the noiseless scenario. It is not difficult to derive its characteristic because the detector receives a vector distributed as a Gaussian under both hypotheses. Appendix 15.4 gives:

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = \frac{P}{2\sigma_Z^2}\left(\frac{\log(1+t\zeta)}{\zeta} - \frac{t}{1+t\zeta}, \frac{1}{\zeta}\log\frac{1+t\zeta}{1+\zeta} + \frac{(1-t)}{1+t\zeta}\right), \tag{11.1}$$

with $\zeta := \sigma_X^2/\sigma_Z^2$. The two end-points of the characteristic are given by:

$$t = 0: \quad E^L_{\mathsf{fn}} = \frac{P}{2\sigma_Z^2}\left(1 - \frac{\log(1+\zeta)}{\zeta}\right), \tag{11.2}$$

$$t = 1: \quad E^R_{\mathsf{fp}} = \frac{P}{2\sigma_Z^2}\left(\frac{\log(1+\zeta)}{\zeta} - \frac{1}{1+\zeta}\right). \tag{11.3}$$

The difference between the upper bound $P/2\sigma_Z^2$ and E^L_{fn} vanishes as $\zeta \to \infty$, *i.e.* $\sigma_Z^2 \to 0$. Unfortunately, it does not hold for E^R_{fp}.

11.1.2 Improved Spread Spectrum

ISS (Improved Spread Spectrum [132]) is a version of spread spectrum where the embedder takes into account the host signal by a linear feedback: $\mathbf{W} = (\alpha - \lambda \mathbf{X}^\top \mathbf{u})\mathbf{u}$ with $0 \le \lambda \le 1$. Host

vector naturally correlated positively (negatively) with the reference vector $\mathbf{u}$ receive less (resp. more) embedding power. This only makes sense under the embedding distortion over expectation constraint, which translates into $\alpha^2 + \lambda^2\sigma_X^2 \leq nP$.

Appendix 15.2 shows that the characteristic is given by the following parametric formulation:

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) \;=\; \frac{P}{2(\sigma_Z^2 + t\sigma_X^2)^2}\left(t^2(\sigma_Z^2+\sigma_X^2), (1-t)^2\sigma_Z^2\right) \tag{11.4}$$

The two end-points of the characteristic are given by:

$$t \;=\; 0: \quad E_{\mathsf{fn}}^L = \frac{P}{2\sigma_Z^2}, \tag{11.5}$$

$$t \;=\; 1: \quad E_{\mathsf{fp}}^R = \frac{P}{2(\sigma_Z^2+\sigma_X^2)} \tag{11.6}$$

This characteristic lies in between the characteristics (15.20) with $N = \sigma_Z^2 + \sigma_X^2$ (lower bound of Case 1) and $N = \sigma_Z^2$ (upper bound of Case 3). On the left endpoint, it converges to the upper bound while on the right endpoint, it converges to the lower bound. Since we are interested in the right endpoint, ISS does not bring any value.

11.2 Link with the asymptotic efficacy of Pitman-Noether

This section makes the connection with the article [23] based on the asymptotic efficacy of Pitman-Noether [138]. At first sight, this tentative is doomed because the concepts of error exponent and of asymptotic efficacy rely on incompatible setups:

- The error exponent is based on the asymptotical study of the error probabilities in a *power* constraint setup: $n^{-1}\mathbb{E}(\|\mathbf{w}(\mathbf{X})\|^2) \leq P$. These probabilities are converging to zero as n goes to infinity. Their study is based on 'large deviation' tools (like the Laplace method).
- The asymptotic efficacy is based on the asymptotical study of the error probabilities in an *energy* constraint setup: $\mathbb{E}(\|\mathbf{w}(\mathbf{X})\|^2) \leq E$. Therefore the watermarking power decreases as n goes to infinity and the probabilities of errors converge to a positive limit [138]. Their study is based on a Taylor series for infinitesimal power.

To make these two approaches closer, we consider that n goes to infinity while P remains fixed but to a very small value. We make use of Taylor series around $\pi := \sqrt{P}$.

Let us consider the following watermarking scheme: At the embedding, $\mathbf{y} = \mathbf{x} + \pi\mathbf{u}(\mathbf{x})$ with $\mathbb{E}(\|\mathbf{u}(\mathbf{X})\|^2) = n$ and $\pi = \sqrt{P}$, and the detector computes the score $s = s(\mathbf{r})$ s.t.:

- Under $\mathcal{H}_0$, the expectation and the variance of the score are $m_{0,n} = 0$ and $s_{0,n}^2 > 0$.
- Under $\mathcal{H}_1$, the expectation and the variance of the score are $m_{\pi,n} > 0$ and $s_{\pi,n}^2 > 0$ with

$$H(\pi) := \lim_{n\to+\infty} \frac{m_{\pi,n}}{\sqrt{n}s_{0,n}} \quad \text{and} \quad G(\pi) := \lim_{n\to+\infty} \frac{s_{\pi,n}^2}{s_{0,n}^2} - 1. \tag{11.7}$$

- Asymptotically as $n \to +\infty$, the law of the score under both hypotheses tends to a Gaussian distribution.

Note that $H(0) = 0$. The additive spread spectrum scheme where $\mathbf{u}(\mathbf{x}) = \mathbf{u}$ complies with these conditions: $H(\pi) = \pi/\sqrt{\sigma_X^2+\sigma_Z^2}$ and $G(\pi) = 0$. In short, $H(\pi)$ is the square root of the Signal to Noise power Ratio per sample at the output of the score function while $G(\pi)$ measures how much the variances of the score deviate from one hypothesis to another.

Then, we compute the characteristic function $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$ based on these asymptotic distributions of the score. Appendix 15.3 shows that it follows the parametric representation:

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = \frac{H(\pi)^2}{2(1+G(\pi)(1-t))^2}(t^2, (1-t)^2(1+G(\pi))), \forall t \in [0,1]. \tag{11.8}$$

The endpoints are given by:

$$E_{\mathsf{fn}}^L = \frac{H(\pi)^2}{2(1+G(\pi))}, \quad E_{\mathsf{fp}}^R = \frac{H(\pi)^2}{2}. \tag{11.9}$$

The idea is to design watermarking schemes where $H(\pi)$ is large even if $G(\pi)$ becomes much bigger than $H(\pi)^2$. We are making a trade-off between the two endpoints, lowering the left endpoint in order to push further the right endpoint.

Instead of studying $H(\pi)^2$, we prefer to deal with its second order approximation:

$$\begin{aligned} H(\pi)^2 &= H(0)^2 + \pi \frac{\partial H(\pi)^2}{\partial \pi}\bigg|_{\pi=0} + \frac{\pi^2}{2}\frac{\partial^2 H(\pi)^2}{\partial^2 \pi}\bigg|_{\pi=0} + o(\pi^2) && (11.10)\\ &= H(0)^2 + \pi(2H(0)H'(0)) + \frac{\pi^2}{2}(2H'(0)^2 + 2H(0)H''(0)) + o(\pi^2) && (11.11)\\ &= \pi^2\eta + o(\pi^2), && (11.12)\end{aligned}$$

with

$$\eta := \lim_{n\to+\infty}\left[\frac{1}{\sqrt{n}\sigma_{0,n}}\frac{\partial m_{\pi,n}}{\partial \pi}\bigg|_{\pi=0}\right]^2. \tag{11.13}$$

Indeed, η is nothing more than the asymptotic efficacy as defined by Pitman and Noether [138, Eq. (3) with $m = 1$ and $\delta = 1/2$]. The link with the error exponents is given by (11.8). Especially, the right endpoint $E_{\mathsf{fp}}^R = \eta P/2 + o(P)$. For the additive spread spectrum scheme, $\eta = (\sigma_X^2+\sigma_Z^2)^{-1}$ and we recover exactly $E_{\mathsf{fp}}^R = P/2(\sigma_X^2+\sigma_Z^2)$.

11.2.1 Use of the efficacy

The efficacy usually serves to motivate a Locally Most Powerful test in the detection of weak signals, an application which is very similar to zero-bit watermarking. Both applications deal with a one-sided hypothesis test: $\mathcal{H}_0 : \pi = 0$ versus $\mathcal{H}_1 : \pi > 0$. Under $\mathcal{H}_1$, the detector only knows that π is positive and small.

The optimal detector is based on the Neyman-Pearson score function $s_{\mathsf{NP}}(\mathbf{r}) = p(\mathbf{r}|\mathcal{H}_1)/p(\mathbf{r}|\mathcal{H}_0)$, which needs parameter π to compute $p(\mathbf{r}|\mathcal{H}_1)$. This optimality (maximizing the power of the test for a given false positive probability) is thus out of reach. Let us change the optimality criterion for the maximization of the efficacy η. The score function maximizing the efficacy η is indeed:

$$s_{\mathsf{LMP}}(\mathbf{r}) = \frac{1}{p(\mathbf{r}|\mathcal{H}_0)}\frac{\partial p(\mathbf{r}|\mathcal{H}_1)}{\partial \pi}\bigg|_{\pi=0}, \tag{11.14}$$

known as the Locally Most Powerful test. In brief, this one-sided detection problem is more difficult when the distance between distributions $p(\mathbf{r}|\mathcal{H}_0)$ and $p(\mathbf{r}|\mathcal{H}_1)$ is close to 0, *i.e.* when

$\pi \approx 0$. As π is unknown at the detection side, it is wise to use the optimal score function for this worst case. The optimal test is the Neyman-Pearson one, and when assuming $\pi \approx 0$ its first order Taylor approximation yields the LMP score function (11.14). Again, as π increases, the LMP test becomes suboptimal (less powerful than the N.-P. test) but this does not matter as the detection problem gets intrinsically easier.

Another advantage is that the LMP score function does not need π: the detector is oblivious to P.

11.2.2 One source of noise

With the setup of Chapter 8, the received vector is $\mathbf{R} = \mathbf{N} + \pi\mathbf{u}$, with $\pi = 0$ under $\mathcal{H}_0$, and $\pi > 0$ under $\mathcal{H}_1$. This means that $p(\mathbf{r}|\mathcal{H}_1) = p(\mathbf{r} - \pi\mathbf{u}|\mathcal{H}_0)$ and

$$s_{\mathsf{LMP}}(\mathbf{r}) = -\frac{1}{p(\mathbf{r}|\mathcal{H}_0)}\mathbf{u}^\top \nabla p(\mathbf{r}|\mathcal{H}_0), \tag{11.15}$$

where ∇ is the gradient operator. Under the Gaussian setup, the result is disappointing: $s_{\mathsf{LMP}}(\mathbf{r}) \propto \mathbf{u}^\top\mathbf{r}$. This score function was already obtained by applying a monotonic function on $s_{\mathsf{NP}}(\mathbf{r})$ (see Sect. 8.1). Yet, this is a particularity of the Gaussian distribution (A statistician would say that there exists a Uniformly Most Powerful test for this distribution family). This detector has been applied to additive spread spectrum with non Gaussian distributions in the early ages of digital watermarking [85, 84, 97, 92].

Last but not least: If the LMP score function provides obliviousness to P, it still needs the variance N in the Gaussian setup to control the probability of false positive (see Sect. 8.1). This is the reason why it is classified as a 'bad' design. In the above-mentioned papers, the LMP score function is computed based on a statistical model $p(\mathbf{r}|\mathcal{H}_0)$ of the samples. The probability of false positive is thus as accurate as the veracity of this model. This is doubtful especially when the model is learned from the received vector $\mathbf{r}$ itself.

11.2.3 One source of side information and no noise

With the setup of Chapter 9, the received vector is $\mathbf{R} = \mathbf{X} + \pi\mathbf{u}(\mathbf{X})$. The work [23] gives the LMP test for such side-informed embedding:

$$s(\mathbf{r}) \propto -\left(\frac{1}{p_{\mathbf{X}}(\mathbf{r})}\mathbf{u}(\mathbf{r})^\top \nabla p_{\mathbf{X}}(\mathbf{r}) + \mathsf{div}(\mathbf{u}(\mathbf{r}))\right), \tag{11.16}$$

where $\mathsf{div}(\cdot)$ is the divergence operator and $p_{\mathbf{X}}$ is the density of the host vector, *i.e.* $p_{\mathbf{X}}(\cdot) = p(\cdot|\mathcal{H}_0)$. This is the score function maximizing the efficacy for a given embedding $\mathbf{u}(\cdot)$.

The surprise is that this rationale can be reversed: What is the embedding function $\mathbf{u}(\cdot)$ maximizing the efficacy for a given score function $s(\cdot)$? The answer is given in [23]:

$$\mathbf{u}(\mathbf{x}) \propto \nabla s(\mathbf{x}). \tag{11.17}$$

Intuitively, when sitting on point $\mathbf{x}$, $\nabla s(\mathbf{x})$ is the direction in the space along which $s(\mathbf{x})$ grows more quickly. Therefore it is natural that the watermark signal pushes the host $\mathbf{x}$ along this direction with the hope that the score gets eventually bigger than the threshold. Note that when applied to $s(\mathbf{x}) = \mathbf{u}^\top\mathbf{x}$, the result is disappointing as the embedding function is a constant vector: $\mathbf{u}(\mathbf{x}) = \mathbf{u}$. We need a nonlinear score function to discover something else than the (too) well-know spread spectrum embedding.

The last step is to combine (11.16) and (11.17) into a single formula coined 'fundamental equation of zero-bit watermarking' in [23]:

$$\eta s(\mathbf{x}) + \frac{1}{p_{\mathbf{X}}(\mathbf{x})} \nabla p_{\mathbf{X}}(\mathbf{x})^\top \nabla s(\mathbf{x}) + \nabla^2 s(\mathbf{x}) = 0, \tag{11.18}$$

where ∇^2 is the Laplacian operator. A score function solution of this equation together with its embedding defined by (11.17) produces an optimal scheme in the sense that neither the embedding nor the score function can be improved and whose efficacy is η.

Many known watermarking schemes are indeed solutions of (11.18), hence the wording 'unifying framework'. For Gaussian distributions, this includes the additive and the proportional spread spectrum schemes and the JANIS watermarking [114] of order $k \in \mathbb{N}^*$:

$$s(\mathbf{r}) = \sqrt{\frac{k}{n}} \sum_{i=1}^{n/k} u_i H_{(11\ldots1)}(\mathbf{r}_{\Pi_i}/\sigma). \tag{11.19}$$

Here, we assume that k divides n, $\{\Pi_1, \ldots, \Pi_{n/k}\}$ is a partition of the set $\{1, \ldots, n\}$ into n/k subsets, each composed of k indices, $\mathbf{r}_{\Pi_i}$ is the restriction of vector $\mathbf{r}$ to the indices of Π_i, $H_{(11\ldots1)}$ is the multivariate Hermite polynomial of order 1 and multiplicity k, *i.e.* $H_{(11\ldots1)}(\mathbf{x}) = \Pi_{i=1}^k x(i)$, and σ is the standard deviation of the components of $\mathbf{r}$. Samples (u_i) play the role of the secret key. For $k = 1$, we are back to the linear correlation score function.

A noticeable exception are the hypercone and ruff schemes introduced in Chapter 9 : they are not solutions of (11.18). Indeed, the most 'similar' solution draws an hyperbola as the detection region (instead of an hypercone or a ruff).

Some new schemes with controllable efficacy (virtually as high as possible) are also given in [23]: For instance, a generalization of the additive and proportional spread spectrum schemes:

$$s(\mathbf{r}) = \sum_{i=1}^{n} u_i H_k(r_i/\sigma), \tag{11.20}$$

where H_k is the univariate Hermite polynomial of order k. Samples (u_i) play the role of the secret key. For $k = 1$, $H_k(r) = r$ and we are back to the linear correlation score function.

The fact that Hermite polynomials (multivariate and univariate) appear in the solution of (11.18) indicates a connection with Gram-Charlier or Edgeworth developments around the Gaussian distribution. In brief, the embedding creates a probability distribution of the watermarked signals, which is as 'far' as possible away from the host distribution (white Gaussian noise) but under a limited distortion budget. The generalization of additive and proportional spread spectrum (11.20) tweaks the probability distribution of the marginals whereas JANIS (11.19) creates correlation between samples. This can be seen as small perturbations on the manifold of distributions as described by J.-F. Cardoso [93]: There are two ways to get away from the white Gaussian distribution: increasing the non-Gaussianity of the marginals or increasing the correlation between components.

In the noiseless setup, schemes (11.19) and (11.20) of order k offer the right endpoint $E_{\mathsf{fp}}^R = kP/2\sigma_X^2 + o(P)$. This parameter k can be virtually very high. This gives us the hope of pushing the right endpoint as close as possible to its upper bound.

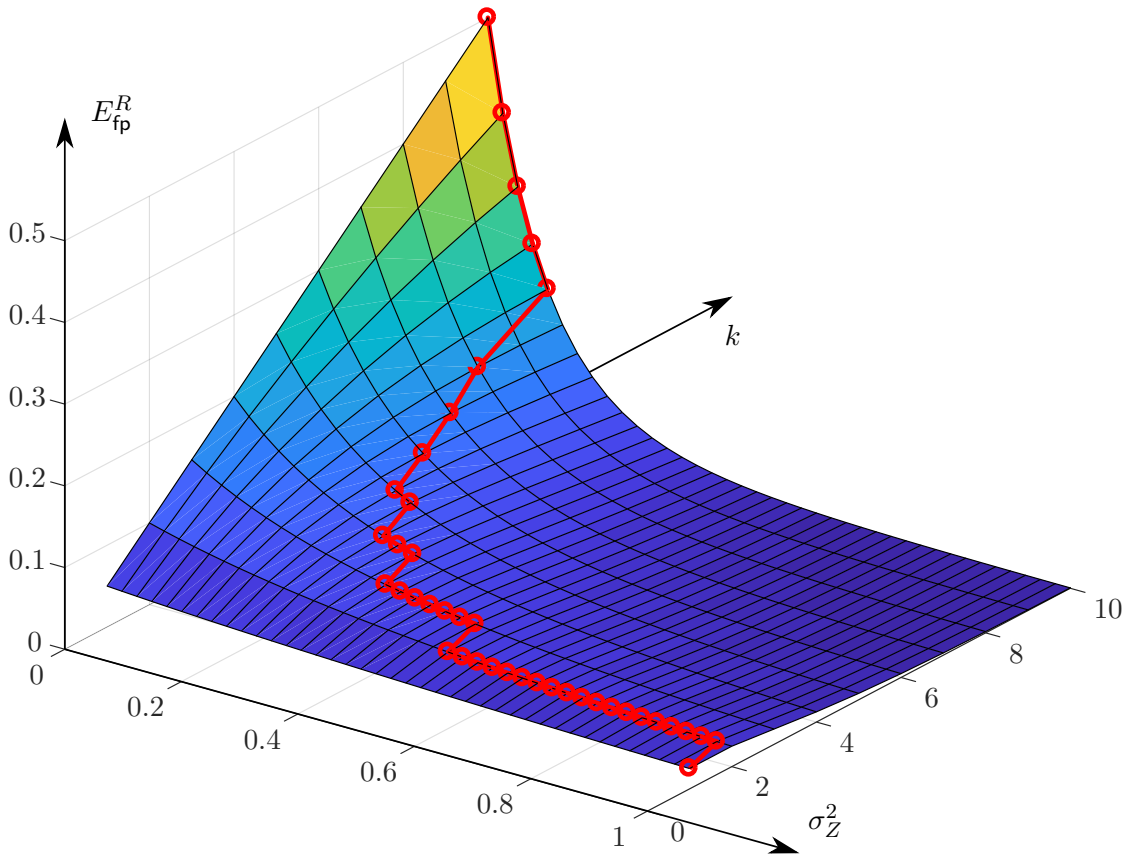


Figure 11.1: E_{fp}^R of the Janis scheme as a function of σ_Z^2 and ρ ($\sigma_X^2 = 1$ and $P = 0.1$). The red line shows the maximum of this quantity over k for a fixed σ_Z^2.

11.2.4 One source of side information and one source of noise

In the noisy setup of chapter 10, the efficacy and thus E_{fp}^R decreases faster as k is bigger [23] (see also Appendix 17):

$$E_{\mathsf{fp}}^R = k\frac{P}{2\sigma_X^2}\left(\frac{\sigma_X^2}{\sigma_X^2+\sigma_Z^2}\right)^k + o(P). \tag{11.21}$$

Fig. 11.1 shows this function. The choice of the best parameter k does not depend on P, but solely on the ratio σ_Z^2/σ_X^2. Indeed, for $\sigma_Z^2 > \sigma_X^2$, the best choice is $k = 1$, *i.e.* the Neyman-Pearson detector under Case 1. Otherwise, k is the best value over the range $\sigma_Z^2/\sigma_X^2 \in [1/k, 1/k-1]$.

Fig. 11.3 gives a comparison with the bounds (Case 1 and Case 3). Since the detector is not oblivious to (σ_X^2, σ_Z^2), these bounds are given by the Neyman-Pearson detector of Sect. 8.1. JANIS and Hermite polynomial embeddings are interesting only over a limited range of E_{fp}. For instance, their right endpoint lies above the lower bound $P/2(\sigma_X^2+\sigma_Z^2)$ only for $\sigma_Z^2 \leq \sigma_X^2\left(k^{1/k-1}-1\right)$ whose first values are given in Tab. 11.1.

As for a rationale for selecting the order k, suppose the designer wishes to operate at $E_{\mathsf{fp}} = E$ so that he/she needs $n \approx -\log \mathbb{P}_{\mathsf{fp}}/E$ samples. The watermark is deemed robust when $E_{\mathsf{fp}}^R > E$. By neglecting the term $o(P)$, this happens when $\sigma_Z^2 < \bar{\sigma_Z}^2$

$$\bar{\sigma_Z}^2 = \sigma_X^2\left|\left(\frac{kP}{2E\sigma_X^2}\right)^{1/k} - 1\right|_+. \tag{11.22}$$

Table 11.1: Maximum value of σ_Z^2 for which JANIS and Hermite polynomial gives a better right endpoint than the lower bound (for $\sigma_X^2 = 1$)

k	2	3	4	5	6	7	8	9	10	11	12
$k^{1/k-1} - 1$	1.0	0.73	0.59	0.50	0.43	0.38	0.35	0.32	0.29	0.27	0.25

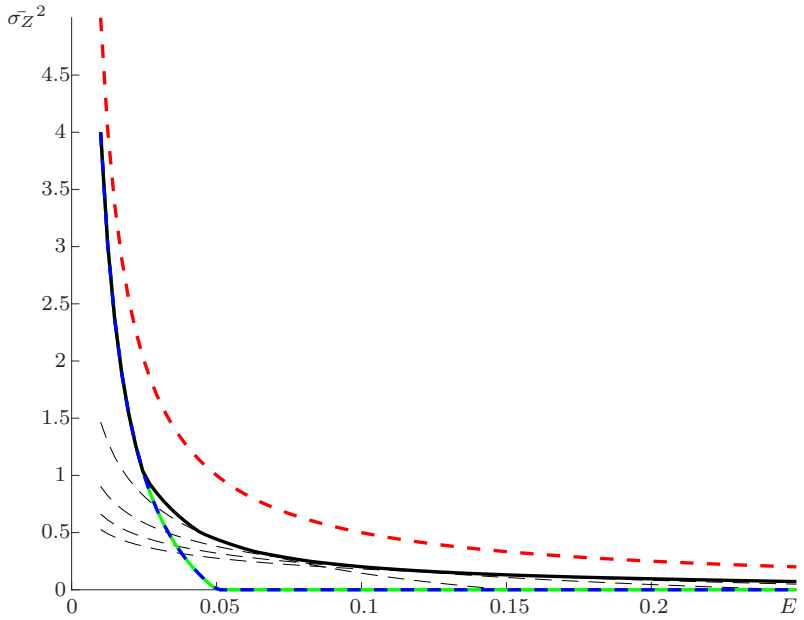


Figure 11.2: $\bar{\sigma_Z}^2$ as a function of E for $\sigma_X^2 = 1$, $P = 0.1$. The upper bound and the lower bound are given by the Neyman-Pearson detector under Cases 1 and 3 (respectively). The **black line** is $\bar{\sigma_Z}^2$ for the best values k knowing E, *i.e.* the enveloppe of the family of curves parametrized by k. The green line is the Neyman-Pearson detection (*i.e.* $k = 1$).

This upper bound is maximized for an integer $k^\star$ (ceil or floor) rounding the value $2eE\sigma_X^2/P$ so that $\bar{\sigma_Z}^2 \approx \sigma_X^2(e^{\frac{P}{2eE\sigma_X^2}} - 1)$. Indeed, $k^\star = 1$ if $E \leq P/4\sigma_X^2$, as shown in Fig. 11.2.

11.2.5 Advantages and drawbacks

The fact that the ratio P/E appears in the expression of $\bar{\sigma_X}^2$ is an advantage. Many watermarking techniques extract n coefficients from content and project them into a secret subspace of dimension n' [25]. In this subspace, the host is more Gaussian distributed and both embedder and detector then deal with shorter vectors. Variances σ_X^2 and σ_Z^2 are the same. However, the embedding constraint becomes $P' = nP/n'$. To keep an asymptotical setup, consider that $n' = \kappa n$, with $0 < \kappa < 1$, so that $n' \to +\infty$ as $n \to +\infty$. This also relates a watermark detection in the full space with a watermark detection in the subspace: Operating at $E_{\mathsf{fp}} = E$ in the full space amounts to operate at $E_{\mathsf{fp}} = E' = \kappa E$ in the subspace. In the end, $E'/P' = E/P$ and $\bar{\sigma_X}^2$ remains the same. In other words, projecting onto a subpsace doesn't bring any suboptimality. This property does not hold for the ruff scheme and the Voronoï modulation.

The fact that the quantity P/σ_X^2 appears in the expression of $\bar{\sigma_X}^2$ is an advantage for some watermarking techniques where the embedding constraint is given as a Watermark to Host power

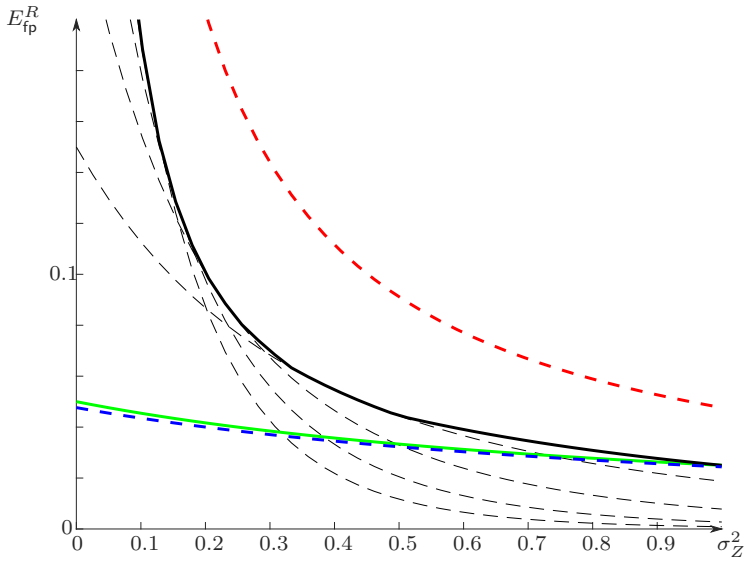


Figure 11.3: E_{fp}^R of the Janis scheme as a function of σ_Z^2 ($\sigma_X^2 = 1$ and $P = 0.1$). The dotted lines shows this function for some values of $k \in \{1, \dots, 10\}$. The **black line** is E_{fp}^R for the best values of k knowing σ_Z^2, *i.e.* the enveloppe of the family of curves parametrized by ρ. The green line shows E_{fp}^R for the Neyman-Pearson detection (*i.e.* $k = 1$). The upper and lower bounds corresponding to the Neyman-Pearson detector in Case 3 (*i.e.* non blind detector with $N = \sigma_Z^2$) and in Case 1 (*i.e.* non side-informed embeddder and blind detector with $N = \sigma_Z^2 + \sigma_X^2$) respectively (see (8.14)).

ratio. This is more or less the case when watermarking audio content. Then, P/σ_X^2 is fixed so that the designer can use the best value for k.

There is however a big drawback: the quantity $G(\pi)$ defined in (11.7) explodes with k even in the noiseless setup (see Appendix 17). Therefore, this scheme sustains strictly positive $E_{\mathsf{fn}} = F(E_{\mathsf{fp}})$ on a large interval $E_{\mathsf{fp}} \in [0, E_{\mathsf{fp}}^R)$ and over a range of noise power of practical interest, but these false negative error exponents are indeed very weak (see (11.9)).

11.3 Conclusion

This chapter bridges two pieces of theory about zero-bit watermarking by calculating the error exponents of efficacy optimum schemes solutions of the fundamental equation of zero-bit watermarking (11.18). One may find many advantages to these schemes. Yet, they are not usable in practice for the sole reason that this piece of theory does not provide a means to set the threshold guaranteeing a given level of false positive probability. The 'perspectives' chapter gives preliminary ideas to solve this problem in Sect. 12.1.

Part III

Perspectives

Chapter 12

Perspectives

12.1 About the use of rare event simulation in zero-bit watermarking

Chapter 6 shows the application of rare event simulation technique to traitor tracing. The estimation of weak probabilities is also critical in digital watermarking. In zero-bit watermarking, the score $s(\mathbf{r}, k)$ is a function of the extracted vector $\mathbf{r}$ from the received content and the secret key k. The false positive probability $\mathbb{P}_{\mathsf{fp}}$ is the probability that $s(\mathbf{r}, k)$ is bigger than the threshold whereas the received content has not been watermarked (*i.e.* $\mathbf{r} = \mathbf{x} + \mathbf{n}$).

12.1.1 A shift of paradigm

As stated above, the problem is not well posed. Evaluating a probability implies that 'something' is deemed random. In the previous chapters, that 'something' is indeed 'everything'. For a given threshold τ, we need to evaluate

$$\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(s(\mathbf{R}, K) > \tau) = \int_{\mathcal{K}} p_K(k) \left(\int_{\mathbb{R}^n} \mathbb{1}_{s(\mathbf{r},k)>\tau}(\mathbf{r}, k).p_{\mathbf{R}}(\mathbf{r}) \partial \mathbf{r} \right) \partial k \tag{12.1}$$

The capital letters denote random elements, function $\mathbb{1}_{\mathsf{A}}(\cdot)$ is the indicator function of event A. The second part of the equation relies on a model of the host signal $\mathbf{R}$ as an absolutely continuous random vector with density $p_{\mathbf{R}}(\cdot)$, and the key K (denoted as well as an absolutely continuous random variable, but it could be a discrete r.v.) has a density $p_K(\cdot)$.

This raises the crucial issue of modelling the huge diversity of the multimedia contents by a single density $p_{\mathbf{R}}(\cdot)$. We have played that game in the previous chapters but it was under the cover of a theoretical study. Yet, this approach is not sound for a practitioner: no feature vector extracted from multimedia content is Gaussian distributed.

In the very early ages of watermarking, researchers were illustrating how good their scheme was with plots of $\{s(\mathbf{r}, k_i)\}_i$. The extracted vector was fixed (usually extracted from the watermarked version of the image 'Lena') and several keys were tested. The reader could observe a peak when the detection key was the same as the embedding key, and lower noisy measurements when the detection key was different. This kind of illustration gradually disappeared from watermarking literature. I suppose two reasons:

- Reviewers complained that testing on a single image was not convincing. In the same way, plotting hundreds of scores has no scientific value for evaluating low probabilities.

- Research in watermarking has always been driven by industrial needs. At that time, big companies were fighting in the battle of video watermarking for DVD copy protection. In this application, the watermark detector is embedded in DVD players and there is a single secret key owned by the movie industry (a.k.a. Hollywood).

This explains why the false positive probability is usually defined nowadays as $\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(s(\mathbf{R}, k) > \tau)$. Note that, if for any $k \in \mathcal{K}$, $\mathbb{P}(s(\mathbf{R}, k) > \tau) < \eta$, then $\mathbb{P}(s(\mathbf{R}, K) > \tau) < \eta$ whatever the distribution of K over its definition set $\mathcal{K}$.

I propose two simple ideas with the bet that their combination makes more sense in practice.

- Let us go back in the early ages of watermarking!
- Why do we need a universal threshold?

In other words, my proposal redefines the probability of false positive as $\mathbb{P}(s(\mathbf{r}, K) > \tau(\mathbf{r}))$. If we were testing $1/\mathbb{P}_{\mathsf{fp}}$ secret keys, one key (on expectation) would give a score larger than the threshold. The threshold is no longer universal in the sense that it may now depend on the received vector $\mathbf{r}$. If, for any host vector $\mathbf{r}$, we are able to find $\tau(\mathbf{r})$ s.t. this $\mathbb{P}_{\mathsf{fp}}$ is lower than the level η, then this level is guaranteed whatever the distribution of $\mathbf{R}$. We have just interchange the integrals over $\mathbf{r}$ and k in (12.1):

$$\mathbb{P}_{\mathsf{fp}} = \mathbb{P}(s(\mathbf{R}, K) > \tau(\mathbf{R})) = \int_{\mathbb{R}^n} p_{\mathbf{R}}(\mathbf{r}) \left(\int_{\mathcal{K}} \mathbb{1}_{s(\mathbf{r},k)>\tau(\mathbf{x})}(\mathbf{r}, k).p_K(k)\partial k \right) \partial \mathbf{r} = \int_{\mathbb{R}^n} p_{\mathbf{R}}(\mathbf{r})\eta \partial \mathbf{r} = \eta. \tag{12.2}$$

The main advantage of this approach is that we no longer need a would-be distribution of $\mathbf{R}$. Instead we need a distribution of K, which is a more realistic and less daunting task. Usually, K is a seed uniformly distributed over $\mathcal{K} = \{0, 1\}^L$. It is used to generate a pseudo-random reference vector $\mathbf{U}$, say distributed as a white Gaussian noise or uniformly over the hypersphere. This statistical model is obviously wrong because we can only generate 2^L vectors. However, this statistical model of $\mathbf{U}$ is closer to reality by a large margin than any statistical model on $\mathbf{R}$ describing feature vector extract from multimedia content.

Another important advantage: It makes the bad designs of Chap. 11 practical! These watermarking schemes were banned because their detector oblivious to the distributions of the host and of the noise could not guarantee the required false positive probability level (or error exponent). The only schemes left were the ones based on solid angle (single or double hypercones, ruffs) under the assumption that the distributions are isotropic, and the ones based on lattice under the assumption of a whitening dither $\mathbf{U}$ or the flat-host assumption. This approach enlarges the toolbox of watermarking schemes for the practitioners.

12.1.2 A question for the theoretician

This approach is not so beneficial for the theoretician. The threshold $\tau(\mathbf{r})$ ensuring a required false positive probability is found by a rare event simulation (see Chap. 6). This prevents any theoretical study. However, there is at least one insightful exception.

Under Cases 1 and 3 of Sect. 7.5, the received signal is $\mathbf{R} = \mathbf{w} + \mathbf{N}$ with $\mathbf{w} = \sqrt{nP}\mathbf{u}$ and $\|\mathbf{u}\| = 1$. The Neyman-Pearson detector is optimal but it is a bad design as it needs the power N to set the threshold accordingly (see Sect. 8.1). The hypercone detector of Sect. 8.2 is in a way the 'oblivious' version of the Neyman-Pearson detector. But, the performances are lower: for instance, the right endpoint E_{fp}^R decreases from $P/2N$ to $1/2 \log(1 + P/N)$.

With this new approach on the false positive probability, the Neyman-Pearson is no longer a bad design and we could dream of its optimality. This is indeed misleading. Its score is $s(\mathbf{r}, \mathbf{u}) = \mathbf{r}^\top \mathbf{u}$. Vector $\mathbf{u}$ plays the role of the secret key whose statistical model is $\mathbf{U} \sim \mathcal{U}_{\mathcal{S}_n}$, where $\mathcal{S}_n$ is the unit hypersphere in $\mathbb{R}^n$. We have:

$$\mathbb{P}(s(\mathbf{r}, \mathbf{U}) > \|\mathbf{r}\| \cos\theta) = 1 - I_{\cos^2\theta}\left(\frac{1}{2}; \frac{n-1}{2}\right). \tag{12.3}$$

Contrary to the first approach, the threshold is no longer universal: It is a threshold only valid for the received $\mathbf{r}$ as $\tau(\mathbf{r}) = \|\mathbf{r}\| \cos\theta$. Yet, comparing $s(\mathbf{r}, \mathbf{u})$ to that $\tau(\mathbf{r})$ amounts to compare $\mathbf{r}^\top \mathbf{u}/\|\mathbf{r}\|$ to $\cos\theta$: We are back to the hypercone detector.

This simple example shows that the shift in the definition of the false positive probability brings two news. The good news is that it makes the schemes of Chapt. 11 practical. We are now able to control the probability of false positive. The bad news is that it questions the expression of their performances, be it measured by its characteristic or the right endpoint E^R_{fp}. Will this shift of paradigm allow for more powerful side-informed zero-bit watermarking?

12.1.3 A question for the practitioner

If the above question cannot be solved, the only way to compare schemes is by numerical simulations. Here is a very preliminary work comparing the ruff detection scheme with the bad design Janis (of order 2).

The simulation is the following: We estimate $\mathbb{P}_{\mathsf{fn}}$ under $\mathcal{H}_1$ for a given attack power N. This probability is not low s.t. a Monte Carlo simulation over hundreds of trials is sufficient. For the ruff detector scheme, the threshold is universal set by (9.22). For the JANIS scheme, for any $\mathbf{r}$, we compute $s(\mathbf{r}, k)$ where k is the embedding key, and then a rare event simulation estimates the probability that $s(\mathbf{r}, K)$ is bigger than that $s(\mathbf{r}, k)$. If the estimation is lower than the required false positive probability, then the detector deems $\mathbf{r}$ as watermarked. In the Janis scheme, the secret key is a permutation. The rare event simulation (*i.e.* the random replicator) modifies permutations by swapping two indices.

Fig. 12.1 shows that Janis outperforms the hypercone and the ruff detectors. These three schemes are theoretically sound, but based on two different pieces of theory. Chap. 11 tries to bridge the gap, but this empirical observation questions this work. Is there a theoretical justification of this empirical observation?

12.2 On the complementarity of content based retrieval and zero-bit watermarking

This Habilitation focuses on the robustness of digital watermarking but this technology has many other issues to be fixed.

Trade-off between payload and robustness. In multi-bit watermarking, the payload is defined as the length of the embedded message. The robustness is the ability of the watermark decoder to retrieve the embedded message despite modifications of watermarked contents. It is well-known that there is a trade-off between the payload and the robustness [103]. The longer the message, the less likely we decode the correct message after severe distortions. As a corollary, the most robust scheme is the zero-bit watermarking.

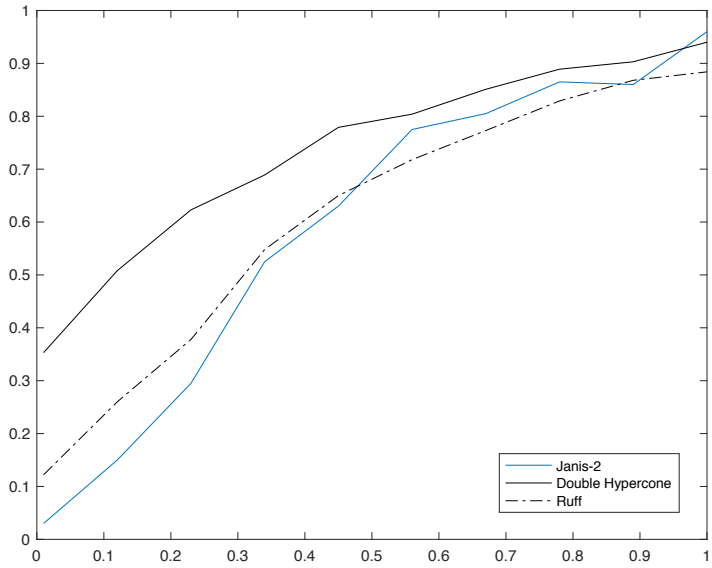


Figure 12.1: $\mathbb{P}_{\mathsf{fn}}$ as a function of σ_Z^2 for $\mathbb{P}_{\mathsf{fp}} \leq \eta = 10^{-6}$, $n = 2048$, $\sigma_X = 1$ and $P = 0.01$. Janis (order 2) vs. the double hypercone (**black line**) and the ruff detector $\rho = 1/128$ (**black dotted line**).

Robustness against geometric attacks. There are watermarking techniques providing very good robustness against compression, filtering, color calibration, or noise (so-called *valumetric* attacks). There are watermarking techniques providing good robustness against *geometric* attacks, which modify the geometry of the image like rotation, resizing, cropping. However, very few watermarking techniques provide robustness against both types of attacks.

Security. A digital watermarking scheme needs a secret key. It prevents reading, modifying or erasing the embedded messages by unauthorized users. However, the security levels of well-known watermarking schemes are low [5, 19, 11]. By processing several contents (in the order of a hundred) watermarked with the same technique and the same secret key but with different messages, an attacker can estimate the secret key.

12.2.1 Content based multimedia document retrieval

This section briefly introduces another technology competitor to watermarking in some applications. Content based multimedia document retrieval (a.k.a. multimedia robust hash or fingerprinting, similarity search) enables computers to recognize multimedia contents. This technology is mature and deployed in services like Shazam, Google Image, Tineye, Pixsy. Let us consider this technology for still images, a.k.a. CBIR (Content Based Image Retrieval).

Like biometry, it proceeds in two steps: the enrollment phase and the recognition phase. The core of this technology extracts from an image a compact representation of its visual content. This representation is discriminative (two dissimilar images have very different representations) while being robust (quasi copies of an image share similar representations). Here are some scientific publications describing such classical representations for still images [121, 142, 107].

In the enrollment phase, the CBIR system receives a collection of multimedia materials together with some metadata. The system extracts the representation of each piece of content.

Then, it performs the indexing task, which organizes this set of representations into a database in order to ease the search. In the recognition phase, the CBIR system receives a content under scrutiny, so-called the query. It computes its representation and looks in the database for the most similar representations.

This technology was originally invented to ease the management of large multimedia content collection providing a search by query example. The CBIR system returns a list of most similar (in some sense) pieces of content. This is the service provided for instance by Google image. Nowadays, this technology is also used as a content recognition tool. The CBIR system recognizes the query as a copy of the image whose representation is the most similar.

This technology provides a great robustness to both geometric and valuemetric attacks. Moreover, some CBIR techniques can resynchronize the query. In other words, the representation carries information about the geometry of the enrolled image. By comparison with the representation of the query, a geometric registration aligns the query with respect of the original image used at the enrollment. This inverts rotation and resizing. Another advantage of this technology is its speed. The recognition step is fast even if the database is composed of millions of compact representations.

The state of the art in Content Based Image Retrieval witnesses the following pitfalls.

Trade-off between false positive and false negative. Based on the similarity between their representations, the system must state whether the query is a copy of the database image ranked first. This is usually solved by comparing the similarity to a threshold. This technology has a poor false positive / false negative rate trade-off. This issue becomes more critical as the database gets larger. As a corollary, the system fails distinguishing two images which are similar: for instance, pictures taken from the same event by two near photographers at the same time.

Multiple sources. A CBIR system fails identifying an image when several versions exist. For instance, suppose that two photo agencies manage the rights of the same picture. The system might identify the image but it is not possible to tell which version it is and therefore to find back the photo agency which should claim the royalties.

12.2.2 A question for the practitioners: Combining the two technologies

The above description shows that these two technologies are indeed complementary. Can we combine them into one system to have the best of both worlds and to mitigate their drawbacks?

Here is a preliminary draft of such system for monitoring the use of copyrighted images.

Images of the collection are watermarked with random keys which are recorded in a database. This provides **perfect security** since a secret key is used only once. The database might contain as well information about each image. The representations of images are computed and indexed. At query time, the CBIR technique returns a list of the most similar images known in the collection together with their secret keys. This is **fast**. For each candidate, CBIR provides geometric resynchronization. This offers **robustness against geometric attacks**. For each candidate, the watermarking detector yields a yes or no answer. It has to cope with valumetric attacks only, and since it is a zero-bit watermarking, it is **extremely robust**. If the detection is positive, the database gives back the information associated to the image. This gives **payloads virtually infinite**. The false positive rate is small and controlled thanks to watermarking (contrary to any traditional CBIR systems).

The combination of the two technologies can be seen as a perfect watermarking system (although the watermark detector has to store the whole database, which is unusual) or as a perfect CBIR tool (but with a low query time due to the multiple watermark detections). Note that the database does not store any image, but a compact representation, a secret key and some metadata per image. If the application allows it, it is also possible to store the original image and to run a non-blind watermarking detector which will be even more robust.

At the query time, the system filters first by the CBIR and then by the watermarking detector. This sequential filtering provides very low false positive rate, but it might be detrimental to the false negative rate. Indeed, the global robustness of the system is the weakest robustness in between CBIR and zero-bit watermarking. This is a question for the practitioners.

12.3 Traitor tracing with Tardos codes

As far as I am concerned, the research efforts about *binary* Tardos codes can stop. Our knowledge on this topic is almost complete. Here are some remaining questions (whose answers will not change the fate of binary Tardos codes):

- There is a little uncertainty about the single capacity (4.43) as revealed in [140].
- For adaptive decoders, the impact of the accuracy of the collusion strategy estimation on the distinguishability is not well investigated, especially for short codes.
- A generalized linear decoder aggregates several scores which are MAP linear scores tuned on the worst collusion strategies of one-sided subsets. Sect. 5.2.3 naturally divided the set of collusion strategies into $c_{\max} - 1$ subsets, one per collusion size (from 2 to $c_{\max}$). But article [80] raises the question of the optimum partition for a general compound channel.
- I do not know any definite real implementation where a watermarking technique yields *soft* outputs fully exploited by the Tardos decoder thanks to a relevant marking assumption. One proposal is the article [124] or in [42, Sect. II.C].

The efforts should now focus on q-ary Tardos codes where codewords are sequences of symbols in alphabet $\{0, 1, \ldots, q-1\}$. Nevertheless, we already know many theoretical results about q-ary Tardos codes (achievable rates, asymptotical capacity, provably good key distributions, provably worst collusion strategies, capacity achieving fixed score functions) [81, 89, 90, 156, 117, 120, 153, 111]. Chapter 6 also applies to q-ary code without any difficulty. The main result is the following promise: Asymptotically as $c \to +\infty$ and under the marking assumption

$$C_c^{(J)} = \frac{(q-1)\log 2}{2c^2 \log q}, \tag{12.4}$$

i.e. an increasing function of q. This means shorter codes provided that a watermarking scheme can embed one q-ary symbol per content block in a robust manner.

The design of adaptive decoders for q-ary codes is a challenge. It aims at learning some inference about the collusion strategy in order to match the score function. Unfortunately, as soon as $q > 2$, the model of the collusion strategy is much more complex (see Sect. 2.4.3) and its estimation becomes a nightmare. There are other models of the collusion strategies (especially the combined digit model [155]). As far as I know, the 'Learn and Match' decoding strategy has not been applied to these models. This is important as Fig. 6.4 and 6.5 show that adaptive decoders make the difference when the collusion strategy is not the worst case. Yet, we might

be disappointed. Larger alphabets means higher capacities and therefore shorter codes. Too few information about the collusion strategy might leak from the pirated sequence. The decoder will not learn much about it preventing its adaptivity. In the end, the 'Learn and Match' decoding strategy might be deceiving. On the other hands, capacity achieving scoring functions might be 'sufficient' since the capacity is larger.

Another track of research aims at easing the integration of Tardos codes (be it binary or q-ary) into real-life system architectures. This encompasses its connection with a watermarking scheme compliant with the constraints of content distribution [122, 45, 148, 149], but also protocols for deploying these codes. In the literature, the code designer and the accusation are always trusted entities. Yet, the accusation can frame any innocent user by tweaking the secret sequence $\mathbf{p}$ at the accusation side [29]. This lack of trust between entities calls for protocols protecting the users [29, 112].

Part IV

Appendices

Chapter 13

Achievable rates of Tardos codes

13.1 Distinguishability

This section aims at measuring how harmful is a collusion strategy by the induced distinguishability between the innocents and colluders' codewords. This gives birth to the concept of achievable rate as explained in Sect. 4.3. It is first explained for single decoders and then for joint decoders.

13.1.1 The case of a single decoder

The text defines the distinguishability as $D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$, which is the expectation over P of Kullback-Leibler divergence between the distributions of X_{inn} and X_{col} for a given collusion strategy $\boldsymbol{\theta}_c$.

Asymptotic setup In an asymptotical setup where $m \to \infty$, the probabilities of error $\mathbb{P}_{\mathsf{fp}}$ and $\mathbb{P}_{\mathsf{fn}}$ for accusing a given user j may converge to zero exponentially. The error exponent are defined as:

$$E_{\mathsf{fp}} := \lim_{m\to\infty} -\frac{1}{m}\log \mathbb{P}_{\mathsf{fp}}, \quad E_{\mathsf{fn}} := \lim_{m\to\infty} -\frac{1}{m}\log \mathbb{P}_{\mathsf{fn}}. \tag{13.1}$$

There is a tradeoff between the two error exponents. The Neyman-Pearson detector based on the likelihood ratio score function achieves the best trade-off. The application of Sanov's theorem shows that, for this detector [102, Eq. (12.196) and Eq. (12.197)]:

$$E_{\mathsf{fn}} = D(X_\lambda; X_{\mathsf{col}}|\boldsymbol{\theta}_c, P) \quad \text{and} \quad E_{\mathsf{fp}} = D(X_\lambda; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P), \tag{13.2}$$

where X_λ is a binary random variable, 'mixture' of X_{inn} and X_{col}. Its law is given by

$$\mathbb{P}(X_\lambda = x|y,p) \propto \mathbb{P}(X_{\mathsf{inn}} = x|y,p)^{1-\lambda}\mathbb{P}(X_{\mathsf{col}} = x|y,p)^{\lambda}, \quad \forall x \in \{0,1\}, \forall \lambda \in [0,1]. \tag{13.3}$$

When $\lambda = 1$, $X_\lambda = X_{\mathsf{col}}$ so that $E_{\mathsf{fn}} = 0$ and $E_{\mathsf{fp}} = D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$. When $\lambda = 0$, $X_\lambda = X_{\mathsf{inn}}$ so that $E_{\mathsf{fn}} = D(X_{\mathsf{inn}}; X_{\mathsf{col}}|\boldsymbol{\theta}_c, P)$ and $E_{\mathsf{fp}} = 0$. For any value $\lambda \in (0,1)$, both exponents are strictly positive if X_{col} and X_{inn} are not identically distributed. Also, E_{fp} is a continuous increasing function, while E_{fn} is a continuous decreasing function of λ (see Fig 13.1).

Achievable rate The text defines the rate $R := {}^{\log n}/_{m}$. Whatever the objective concerning the identification of colluders of Sect. 2.3.3, the probability of false negative is bounded by $\mathbb{P}_{\mathsf{FN}} \leq c\mathbb{P}_{\mathsf{fn}}$. Since c is assumed to be fixed, $\mathbb{P}_{\mathsf{FN}}$ vanishes exponentially as $m \to \infty$ if $\mathbb{P}_{\mathsf{fn}}$ does so (*i.e.* $E_{\mathsf{fn}} > 0$

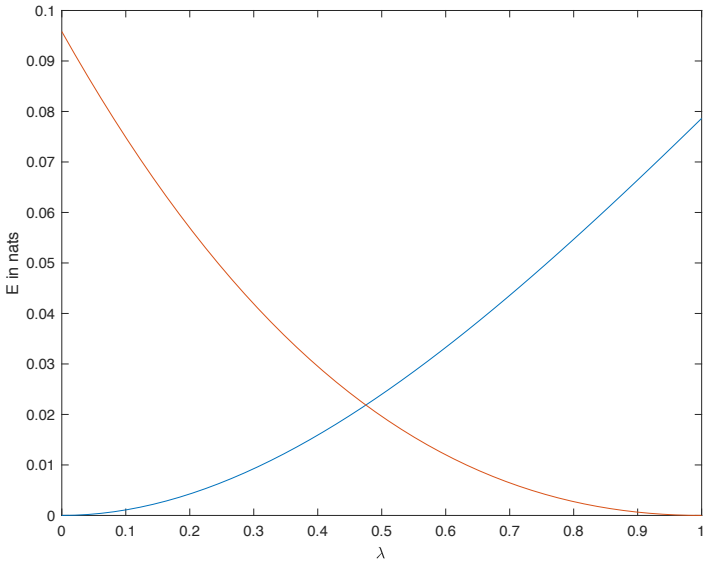


Figure 13.1: Error exponents E_{fp} and E_{fn} as function of $\lambda \in [0,1]$ for an interleaving attack with $c = 5$ and $p = 0.3$.

and $\lambda < 1$). On the other hand, the probability of accusing at least one innocent is bounded by $\mathbb{P}_{\mathsf{FP}} \leq n\mathbb{P}_{\mathsf{fp}}$, which is translated in terms of error exponent as:

$$\lim_{m\to\infty} -\frac{1}{m}\log \mathbb{P}_{\mathsf{FP}} \geq E_{\mathsf{fn}} - R = D(X_\lambda; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P) - R. \tag{13.4}$$

If $R < D(X_{\lambda=1}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$, the accusation procedure can operate at a $\lambda \lesssim 1$ so that both $\mathbb{P}_{\mathsf{FN}}$ and $\mathbb{P}_{\mathsf{FP}}$ vanishes exponentially. This last statement is indeed the definition of an achievable rate. This shows that distinguishability $D(X_{\mathsf{col}}; X_{\mathsf{inn}}|\boldsymbol{\theta}_c, P)$ is the supremum of the achievable rates.

13.1.2 Joint decoders

Denote by $\mathcal{H}_j$ the hypothesis that there are j colluders in a group of ℓ users, with $j \in \{0, 1, \cdots, \ell\}$. For a given group of ℓ codewords $\{\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_\ell}\}$, the joint decoder computes $\ell + 1$ log-likelihoods $\{\log \mathbb{P}(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_\ell}|\mathcal{H}_i, \mathbf{y}, \mathbf{p})\}_{i=0}^{\ell}$, and decides this groups contains $\hat{\jmath}$ colluders when the $\hat{\jmath}$-th log-likelihood is the biggest:

$$\hat{\jmath} = \arg\max_{0\leq i\leq \ell} \log \mathbb{P}(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_\ell}|\mathcal{H}_i, \mathbf{y}, \mathbf{p}). \tag{13.5}$$

This is a classical procedure in decision theory when dealing with multiple hypothesis [165, Part I, Sect. 2.3].

This makes $\ell(\ell+1)$ types of error: the events that the joint decoder decides that there are $\hat{\jmath}$ colluders whereas $\mathcal{H}_j$ is true ($\hat{\jmath} \neq j$). Denote by $\mathbb{P}_e(\hat{\jmath}|j)$ the probability of such an event per group.

These types of error have not the same importance. For instance, suppose that the decoder accuses all the users of a group if it detects that the group is composed of ℓ colluders. Otherwise, the decoder doesn't risk any accusation. Therefore, failing to the 'Catch one' objective relates to the probabilities $\{\mathbb{P}_e(\hat{\jmath}|\ell)\}_{\hat{\jmath}=0}^{\ell-1}$. On the other hand, the false positive relates to the probabilities $\{\mathbb{P}_e(\ell|j)\}_{j=0}^{\ell-1}$. Any other type of error is not important.

The joint decoder analyses $\binom{n}{\ell}$ different groups. Among them, there are $\binom{c}{\ell}$ groups exclusively composed of colluders. The accusation fails to reach the objective if it misses all these groups. The decisions about these groups are not independent due to the overlapping between groups. Yet, the probability of missing all these groups is upper bounded by the probability of missing one group in particular: $\mathbb{P}_{\mathsf{FN}} \leq \sum_{\hat{\jmath}=0}^{\ell-1} \mathbb{P}_e(\hat{\jmath}|\ell)$.

On the other hand, there are $\binom{c}{j}\binom{n-c}{\ell-j}$ groups under hypothesis $\mathcal{H}_j$. Thanks to the union bound, the false positive is bounded by $\mathbb{P}_{\mathsf{FP}} \leq \sum_{j=0}^{\ell-1} \binom{c}{j}\binom{n-c}{\ell-j}\mathbb{P}_e(\ell|j)$.

Asymptotic setup We assume that m and n go to infinity according to the rate of the code: $n = e^{mR}$. Since ℓ is fixed, if all $\{\mathbb{P}_e(\hat{\jmath}|\ell)\}_{\hat{\jmath}=0}^{\ell-1}$ exponentially converge to 0, then $\mathbb{P}_{\mathsf{FN}}$ does so as well. The situation is more complicated for $\mathbb{P}_{\mathsf{FP}}$. We need, $\forall j \in \{0, \dots, \ell-1\}$

$$\lim_{m\to\infty} -\frac{1}{m}\log\left(\binom{c}{j}\binom{n-c}{\ell-j}\mathbb{P}_e(\ell|j)\right) = \left(\lim_{m\to\infty} -\frac{1}{m}\log \mathbb{P}_e(\ell|j)\right) - R.(\ell-j) > 0. \tag{13.6}$$

Like for the single decoder, the application of Sanov's theorem gives an upper bound on the limit in the last equation. It shows that if, $\forall j \in \{0, \dots, \ell-1\}$

$$R < \frac{1}{\ell-j} D(X_{\mathsf{col}}^{(\ell)}; X_{\mathsf{col}}^{(j)} X_{\mathsf{inn}}^{(\ell-j)} | \boldsymbol{\theta}_c, P) \tag{13.7}$$

then, $\mathbb{P}_{\mathsf{FP}}$ and $\mathbb{P}_{\mathsf{FN}}$ vanish exponentially as $m \to \infty$. The right hand side shows the Kullback-Leibler divergence between the distributions of $X_{\mathsf{col}}^{(\ell)}$, a group of ℓ 'colluders' symbols, and of $X_{\mathsf{col}}^{(j)} X_{\mathsf{inn}}^{(\ell-j)}$, a group of ℓ symbols composed of j 'colluders' symbols and $(\ell-j)$ 'innocent' symbols. When j is close to 0, this Kullback-Leibler divergence is expected to be big, but there are more groups of this type, so that it is divided by a bigger amount $(\ell-j)$. In the end, R must be lower than the smallest ratio. A similar expression was found by G. Atia and V. Saligrama for group testing [82, Th. III.1].

Relationship with information theory It happens that $D(X_{\mathsf{col}}^{(\ell)}; X_{\mathsf{col}}^{(j)} X_{\mathsf{inn}}^{(\ell-j)} | \boldsymbol{\theta}_c, P)$ equals the mutual information $I(Y X_{\mathsf{col}}^{(j)}; X_{\mathsf{col}}^{(\ell-j)} | \boldsymbol{\theta}_c, P)$, which equals

$$I(Y X_{\mathsf{col}}^{(j)}; X_{\mathsf{col}}^{(\ell-j)} | \boldsymbol{\theta}_c, P) = H(X_{\mathsf{col}}^{(\ell-j)} | \boldsymbol{\theta}_c, P) - H(X_{\mathsf{col}}^{(\ell-j)} | Y X_{\mathsf{col}}^{(j)}, \boldsymbol{\theta}_c, P). \tag{13.8}$$

Once again, the conditioning of a quantity on P is defined as the expectation over P of that quantity conditioned on $P = p$.

Since the codewords are generated independently, we have

$$H(X_{\mathsf{col}}^{(\ell-j)} | \boldsymbol{\theta}_c, P) = H(X_{\mathsf{col}}^{(\ell-j)} | P) = (\ell-j) H(X_{\mathsf{col}} | P). \tag{13.9}$$

On the other hand, P. Moulin showed the following inequalities thanks to the invariance to permutation of the symbols in $X_{\mathsf{col}}^{(\ell)}$ [136, App. A]: If $0 \leq j \leq i < \ell$, then

$$\frac{1}{\ell-j} H(X_{\mathsf{col}}^{(\ell-j)} | Y X_{\mathsf{col}}^{(j)}, \boldsymbol{\theta}_c, P) \geq \frac{1}{\ell-i} H(X_{\mathsf{col}}^{(\ell-i)} | Y X_{\mathsf{col}}^{(i)}, \boldsymbol{\theta}_c, P), \tag{13.10}$$

$$\frac{1}{j} H(X_{\mathsf{col}}^{(j)} | Y, \boldsymbol{\theta}_c, P) \geq \frac{1}{i} H(X_{\mathsf{col}}^{(i)} | Y, \boldsymbol{\theta}_c, P). \tag{13.11}$$

Equations (13.9) and (13.10) lead to [136, Sec. 3.1]: If $0 \leq j \leq i \leq \ell$, then

$$\frac{1}{\ell-j} I(Y X_{\mathsf{col}}^{(j)}; X_{\mathsf{col}}^{(\ell-j)} | \boldsymbol{\theta}_c, P) \leq \frac{1}{\ell-i} I(Y X_{\mathsf{col}}^{(i)}; X_{\mathsf{col}}^{(\ell-i)} | \boldsymbol{\theta}_c, P). \tag{13.12}$$

As a consequence, the most stringent of the conditions (13.7) is indeed when $j = 0$: If $R < \ell^{-1} D(X_{\mathsf{col}}^{(\ell)}; X_{\mathsf{inn}}^{(\ell)} | \boldsymbol{\theta}_c, P) = \ell^{-1} I(Y; X_{\mathsf{col}}^{(\ell)} | \boldsymbol{\theta}_c, P)$, then R automatically satisfies the ℓ conditions (13.7), and $\mathbb{P}_{\mathsf{FN}}$ and $\mathbb{P}_{\mathsf{FP}}$ exponentially converge towards 0. In other words, $\ell^{-1} I(Y; X_{\mathsf{col}}^{(\ell)} | \boldsymbol{\theta}_c, P)$ is the supremum of the achievable rates for any joint decoder dealing with groups of size ℓ.

Inequality (13.11) leads to the following property [136, Eq. (3.4)]: If $1 \leq k \leq \ell \leq c$, then

$$\begin{aligned}
\frac{1}{k} I(Y; X_{\mathsf{col}}^{(k)} | \boldsymbol{\theta}_c, P) &= \frac{1}{k} H(X_{\mathsf{col}}^{(k)} | \boldsymbol{\theta}_c, P) - \frac{1}{k} H(X_{\mathsf{col}}^{(k)} | Y, \boldsymbol{\theta}_c, P) \\
&= \frac{1}{\ell} H(X_{\mathsf{col}}^{(\ell)} | \boldsymbol{\theta}_c, P) - \frac{1}{k} H(X_{\mathsf{col}}^{(k)} | Y, \boldsymbol{\theta}_c, P) \\
&\leq \frac{1}{\ell} H(X_{\mathsf{col}}^{(\ell)} | \boldsymbol{\theta}_c, P) - \frac{1}{\ell} H(X_{\mathsf{col}}^{(\ell)} | Y, \boldsymbol{\theta}_c, P) \\
&= \frac{1}{\ell} I(Y; X_{\mathsf{col}}^{(\ell)} | \boldsymbol{\theta}_c, P). \qquad (13.13)
\end{aligned}$$

As a consequence, joint decoding ($\ell > 1$) performs better than single decoding ($k = 1$) in the sense that it provides higher achievable rates. Stated differently, for a fixed rate R, joint decoding provides higher error exponents. Moreover, the best joint decoder of this kind is the one dealing with groups of size $\ell = c$, whose supremum of achievable rates is denoted by [136, Sec. 5]:

$$R^{(J)}(P, \boldsymbol{\theta}_c) := c^{-1} \mathbb{E}_P \left(I(Y; X_{\mathcal{C}} | P = p, \boldsymbol{\theta}_c) \right). \qquad (13.14)$$

13.1.3 Informed Decoders

This section generalizes the family of decoders with conditioning on some colluder codewords. Let $R(\ell, k, P, \boldsymbol{\theta}_c)$ denote the rate of a joint decoder computing a score per group of ℓ users and informed by the disclosure of the identity of k colluders, assuming that

$$1 \leq \ell \leq c \quad \text{and} \quad 0 \leq k \leq c - \ell. \qquad (13.15)$$

In other words,

$$R(\ell, k, P, \boldsymbol{\theta}_c) := \frac{1}{\ell} I(Y; X_{\mathsf{col}}^{(\ell)} | X_{\mathsf{col}}^{(k)}, \boldsymbol{\theta}_c, P). \qquad (13.16)$$

We first consider the subsets of colluders $\mathcal{L}$ and $\mathcal{K}$ of size ℓ and k such that $\mathcal{L} \cap \mathcal{K} = \emptyset$ and $\ell + k < c$. Therefore, there exists a colluder, say user j, who does not belong to $\mathcal{L} \cup \mathcal{K}$. The conditioning on the codewords of the caught colluders of $\mathcal{K}$ does not change the inequality (13.13): $R(\ell, k, P, \boldsymbol{\theta}_c) \leq R(\ell + 1, k, P, \boldsymbol{\theta}_c)$. And inequality (13.10) allows us to bound

$$\begin{aligned}
R(\ell + 1, k, P, \boldsymbol{\theta}_c) &= \frac{1}{\ell + 1} H(X_{\mathsf{col}}^{(\ell)} X_j | X_{\mathsf{col}}^{(k)} P, \boldsymbol{\theta}_c) - \frac{1}{\ell + 1} H(X_{\mathsf{col}}^{(\ell)} X_j | Y X_{\mathsf{col}}^{(k)} P, \boldsymbol{\theta}_c) \\
&= \frac{1}{\ell} H(X_{\mathsf{col}}^{(\ell)} | X_{\mathsf{col}}^{(k)}, P, \boldsymbol{\theta}_c) - \frac{1}{\ell + 1} H(X_{\mathsf{col}}^{(\ell)} X_j | Y X_{\mathsf{col}}^{(k)}, P, \boldsymbol{\theta}_c) \\
&\leq \frac{1}{\ell} H(X_{\mathsf{col}}^{(\ell)} | X_{\mathsf{col}}^{(k)} X_j, P, \boldsymbol{\theta}_c) - \frac{1}{\ell} H(X_{\mathsf{col}}^{(\ell)} | Y X_{\mathsf{col}}^{(k)} X_j, P, \boldsymbol{\theta}_c) \\
&= R(\ell, k + 1, P, \theta_c).
\end{aligned}$$

In the end,

$$R(\ell, k, P, \boldsymbol{\theta}_c) \leq R(\ell + 1, k, P, \boldsymbol{\theta}_c) \leq R(\ell, k + 1, P, \theta_c). \qquad (13.17)$$

Fig. 4.4 shows the rates of informed decoder for $\ell \in \{1, \ldots, c\}$ and $k \in \{0, \ldots, c - \ell\}$.

Chapter 14

A marking assumption taking into account the watermarking layer

We denote by $\mathbf{B}$ a feature vector of L components extracted from the original block, and $\mathbf{W}_i$, $i \in \{0,1\}$, the watermark vector that is added. There are thus two watermarked versions of a block: $\mathbf{B}_0 = \mathbf{B} + \mathbf{W}_0$ and $\mathbf{B}_1 = \mathbf{B} + \mathbf{W}_1$. The power of the watermark signal is denoted by P and assumed to be constant over the blocks. The mixing of blocks is modelled as a linear combination in the feature space: $\mathbf{B}_w = (1-w).\mathbf{B}_0 + w.\mathbf{B}_1$, so that:

$$\mathbf{B}_w = \mathbf{B} + (1-w)\mathbf{W}_0 + w\mathbf{W}_1. \tag{14.1}$$

The distortion later on added to the block introduces a noise of power N. We assume that the detection of a watermark signal of power P and under noise power N has a probability of failing in the order of $\exp(-L\frac{P}{2N})$ (*i.e.* L is large enough s.t. the error probability is given by its error exponent, which is here the error exponent of spread spectrum).

14.1 On-off keying modulation

In connection with the second part of this report, this layer is based on a zero-bit watermarking technique. Symbol '0' is embedded in a multimedia content block using the watermarking secret key k_0. Symbol '1' is embedded in a block using the watermarking secret key k_1. At the decoding side, both detectors run in parallel producing two binary outputs $(D_0, D_1) \in \{0,1\}^2$: $D_i = 1$ meaning that the watermark has been detected when testing secret k_i. There are four possible outputs:

- $(D_0, D_1) = (1,0)$: Symbol '0' is decoded, $Y = 0$,
- $(D_0, D_1) = (0,1)$: Symbol '1' is decoded, $Y = 1$,
- $(D_0, D_1) = (1,1)$: Both symbols are decoded. This is a double detection denoted by $Y = \mathsf{d}$,
- $(D_0, D_1) = (0,0)$: No symbol is decoded. This is an erasure denoted by $Y = \times$.

The keys k_0 and k_1 are independent, so are $\mathbf{W}_0$ and $\mathbf{W}_1$, and their detection outputs D_0 and D_1 are independent as well ($D_i = 1$ if $\mathbf{W}_i$ is detected, 0 otherwise). The probabilities of false negative watermark detection are denoted $\mathbb{P}(D_0 = 0|w)$ and $\mathbb{P}(D_1 = 0|w)$. This gives the

following probabilities:

$$\begin{aligned}
\mathbb{P}(Y=0|w) &= (1-\mathbb{P}(D_0=0|w))\mathbb{P}(D_1=0|w), && (14.2)\\
\mathbb{P}(Y=1|w) &= (1-\mathbb{P}(D_1=0|w))\mathbb{P}(D_0=0|w), && (14.3)\\
\mathbb{P}(Y=\mathsf{d}|w) &= (1-\mathbb{P}(D_0=0|w))(1-\mathbb{P}(D_1=0|w)), && (14.4)\\
\mathbb{P}(Y=\times|w) &= \mathbb{P}(D_1=0|w)\mathbb{P}(D_0=0|w). && (14.5)
\end{aligned}$$

By symmetry, $\mathbb{P}(D_0=0|w)=\mathbb{P}(D_1=0|1-w)$, and the above probabilities may be defined solely by the function $w\mapsto\mathbb{P}(D_0=0|w)$. The mixing (14.1) reduces the power of $\mathbf{W}_0$ to $P(1-w)^2$ and the power of $\mathbf{W}_1$ to Pw^2. Under our assumption, $\mathbb{P}(D_0=0|w)\approx\exp(-L\frac{P(1-w)^2}{2N})$. Note that $w\mathbf{W}_1$ is a noise for the detection of $(1-w)\mathbf{W}_0$, whose power Pw^2 is neglected compared to N for the sake of simplicity. Moreover, when $w=1$, the mixed block equals the block $\mathbf{B}_1$, and $\mathbb{P}(D_0=0|1)\approx 1$.

To make the link with the first collusion strategies family 'Copy and Distort' (see Sect. 4.6), we set

$$\mathbb{P}(D_0=0|w=0)=\mathbb{P}(D_1=0|w=1)=\zeta\approx\exp(-L\frac{P}{2N}) \quad (14.6)$$

This is compliant with the idea that when there is no mixing of blocks ($w\in\{0,1\}$), the watermark detector either outputs a binary symbol ($Y=0$ if $w=0$, or $Y=1$ if $w=1$) with probability $1-\zeta$, or outputs an erasure with probability ζ. Finally, $\mathbb{P}(D_0=0|w)\approx\zeta^{(1-w)^2}$ so that:

$$\begin{aligned}
\mathbb{P}(Y=0|w) &\approx \left(1-\zeta^{(1-w)^2}\right)\zeta^{w^2}, && (14.7)\\
\mathbb{P}(Y=1|w) &\approx \left(1-\zeta^{w^2}\right)\zeta^{(1-w)^2}, && (14.8)\\
\mathbb{P}(Y=\mathsf{d}|w) &\approx \left(1-\zeta^{(1-w)^2}\right)\left(1-\zeta^{w^2}\right), && (14.9)\\
\mathbb{P}(Y=\times|w) &\approx \zeta^{w^2+(1-w)^2}. && (14.10)
\end{aligned}$$

In the end, this family of collusion strategies is parametrized by the single scalar ζ.

14.2 Antipodal modulation

With an antipodal modulation, $\mathbf{W}_1=-\mathbf{W}_0$ so that $\mathbf{B}_w=\mathbf{B}+(1-2w)\mathbf{W}_0$, $\forall w\in[0,1]$. The decoding computes the correlation $\mathbf{B}_w^\top\mathbf{W}_0$, and outputs $Y=0$ if this correlation is strongly positive, $Y=1$ if the correlation strongly negative, and $Y=\times$ else. There is no possibility to decode both symbol at the same time.

After the addition of noise of power N, we assume that the watermark decoder succeeds outputting $Y=0$ with probability $\approx 1-\exp(-L\frac{P(1-2w)^2}{2N})=1-\zeta^{1-2w}$ if $(1-2w)>0$, and 0 otherwise. In the end,

$$\begin{aligned}
\mathbb{P}(Y=0|w) &= 1-\zeta^{|1-2w|_+^2}, && (14.11)\\
\mathbb{P}(Y=1|w) &= 1-\zeta^{|2w-1|_+^2}, && (14.12)\\
\mathbb{P}(Y=\times|w) &= \zeta^{(2w-1)^2}, && (14.13)\\
\mathbb{P}(Y=\mathsf{d}|w) &= 0, && (14.14)
\end{aligned}$$

where $|a|_+=a$ if $a>0$ and 0 otherwise.

Chapter 15

Error exponents of the Neyman-Pearson detector

This appendix explains a simple way to derive the error exponents E_{fp} and E_{fn} based on the book [165, I-Sect. 2.7]. The detector receives a random vector $\mathbf{R} \in \mathbb{R}^n$ following distribution $p(\mathbf{r}|\mathcal{H}_0)$ or $p(\mathbf{r}|\mathcal{H}_1)$. We assume the following decision process: the detector first computes a score $s(\mathbf{r}) \in \mathbb{R}$, and then outputs decision $d = 1$ if $s(\mathbf{r}) \geq \tau$, and $d = 0$ otherwise. The probability of false positive equals:

$$\mathbb{P}_{\mathsf{fp}} := \mathbb{P}[s(\mathbf{R}) \geq \tau|\mathcal{H}_0] = \mathbb{E}[\mathbb{1}_{[\tau,+\infty)}(s(\mathbf{R}))|\mathcal{H}_0], \tag{15.1}$$

with $\mathbb{1}_{\mathcal{I}}(x) = 1$ if $x \in \mathcal{I}$, 0 otherwise. The first step uses the Chernoff bound: Because, for any $t \geq 0$, $\mathbb{1}_{[0,+\infty)}(x) \leq e^{tx}\ \forall x \in \mathbb{R}$, the probability of false positive is bounded by:

$$\begin{aligned} \mathbb{P}_{\mathsf{fp}} &\leq \mathbb{E}[e^{t(s(\mathbf{R})-\tau)}|\mathcal{H}_0] = e^{\mu_n(t)-t\tau}, \quad \text{with} & (15.2)\\ \mu_n(t) &:= \log \mathbb{E}[e^{ts(\mathbf{R})}|\mathcal{H}_0]. & (15.3) \end{aligned}$$

Function $\mu_n(\cdot)$ is non positive and convex. The tightest bound is given by the value of t minimizing the exponent: $t^\star = \arg\min_{t>0} \mu_n(t) - t\tau$. Canceling the derivative implies that $\mu_n'(t^\star) = \tau$, so that $\mathbb{P}_{\mathsf{fp}} \leq e^{\mu_n(t^\star)-t^\star\mu_n'(t^\star)}$. The same work applied to the false negative case gives the bound: $\mathbb{P}_{\mathsf{fn}} \leq e^{\mu_n(t^\star)+(1-t^\star)\mu_n'(t^\star)}$ with $t^\star \leq 1$.

The second step aims at showing that the Chernoff bound becomes tighter as $n \to \infty$. For a given score function, denote by $p_S(s|\mathcal{H}_0)$ the distribution of $S = s(\mathbf{R})$ under hypothesis $\mathcal{H}_0$. For a given value of t, we introduce the random variable Z_t whose distribution is given by $p_{Z_t}(z) := p_S(z|\mathcal{H}_0)e^{(tz-\mu_n(t))}$. Its integral sums up to one thanks to the definition of $\mu_n(\cdot)$. Moreover, its expectation and variance have simple expressions [165, I-Eq. (452-453)]:

$$\begin{aligned} \mathbb{E}[Z_t] &= \mu_n'(t), & (15.4)\\ \mathbb{V}[Z_t] &= \mu_n''(t). & (15.5) \end{aligned}$$

This new random variable allows us to rewrite the probability of false positive as the Chernoff bound times a multiplicative constant:

$$\mathbb{P}_{\mathsf{fp}} = e^{\mu_n(t)-t\mu_n'(t)} \int_\tau^{+\infty} e^{t(\mu_n'(t)-z)} p_{Z_t}(z)\partial z. \tag{15.6}$$

This holds for any $t > 0$ and in particular for the optimum $t^\star$ for which $\tau = \mu_n'(t^\star)$. We rewrite the last expression considering the centered reduced version of $Z_{t^\star}$, $\tilde{Z}_{t^\star} := (Z_{t^\star} - \mathbb{E}[Z_{t^\star}])/\sqrt{\mathbb{V}[Z_{t^\star}]}$:

$$\begin{aligned} \mathbb{P}_{\mathsf{fp}} &= e^{\mu_n(t^\star)-t^\star\mu_n'(t^\star)} K_n & (15.7)\\ K_n &= \int_0^{+\infty} e^{-t^\star\sqrt{\mu_n''(t^\star)}\tilde{z}} p_{\tilde{Z}_{t^\star}}(\tilde{z})\partial\tilde{z} & (15.8) \end{aligned}$$

If we can show that, for the particular score function used in the detector, $Z_{t^\star}$ is indeed Gaussian distributed, then $\tilde{Z}_{t^\star} \sim \mathcal{N}(0,1)$. This is often the case under the Gaussian setup ($p(\mathbf{r}|\mathcal{H}_0)$ and $p(\mathbf{r}|\mathcal{H}_1)$ are Gaussian distributions) or at least asymptotically when the components of $\mathbf{R}$ are i.i.d. and the score function is a sum over these n r.v. thanks to the Central Limit Theorem. Under the assumption of the Gaussianity of $Z_{t^\star}$, then the multiplicative constant simplifies to:

$$K_n = e^{\frac{(t^\star)^2}{2}\mu_n''(t^\star)} Q\left(t^\star \sqrt{\mu_n''(t^\star)}\right), \tag{15.9}$$

with $Q(x) := 1 - \Phi(x)$, $\Phi(\cdot)$ being the cumulative density function of $\mathcal{N}(0,1)$. A last assumption considers that $\lim_{n\to+\infty} t\sqrt{\mu_n''(t)} = +\infty$ for any $t \in (0,1]$. Knowing that ${x\phi(x)}/{(1+x^2)} \leq Q(x) \leq {\phi(x)}/{x}$, with $\phi(x) := e^{-x^2/2}/\sqrt{2\pi}$, it comes that:

$$\lim_{n\to+\infty} \frac{1}{n} \log(K_n) = 0. \tag{15.10}$$

In other words, the multiplicative constant of the Chernoff bound does not yield any extra rate. In the end, the false positive error exponent is given by:

$$E_{\mathsf{fp}} = \lim_{n\to+\infty} -\frac{1}{n}\log(\mathbb{P}_{\mathsf{fp}}) = \lim_{n\to+\infty} \frac{-\mu_n(t^\star) + t^\star \mu_n'(t^\star)}{n}. \tag{15.11}$$

The same rationale applied to the false negative case gives:

$$E_{\mathsf{fn}} = \lim_{n\to+\infty} -\frac{1}{n}\log(\mathbb{P}_{\mathsf{fn}}) = \lim_{n\to+\infty} \frac{-\mu_n(t^\star) - (1-t^\star)\mu_n'(t^\star)}{n}. \tag{15.12}$$

The value of $t^\star \in (0,1)$ has little importance as we have now a parametric definition of the characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$.

The last step looks for the best expression of the function $\mu_n(\cdot)$. It considers an ideal detector which knows distributions $p(\mathbf{r}|\mathcal{H}_0)$ and $p(\mathbf{r}|\mathcal{H}_1)$. The best detector is then the Neyman-Pearson test whose score function is the log likelihood ratio: $s(\mathbf{r}) = \log({p(\mathbf{r}|\mathcal{H}_1)}/{p(\mathbf{r}|\mathcal{H}_0)})$. For this particular test, function $\mu_n(\cdot)$ takes the following expression:

$$\mu_n(t) = \log \int_{\mathbb{R}^n} (p(\mathbf{r}|\mathcal{H}_1))^t (p(\mathbf{r}|\mathcal{H}_0))^{(1-t)} \partial\mathbf{r}, \forall t \in [0,1]. \tag{15.13}$$

Note that the following properties hold for the Neyman-Pearson test:

$$\mu_n(0) = \mu_n(1) = 0, \tag{15.14}$$
$$\mu_n'(0) = -\mathbb{E}[s(\mathbf{R})|\mathcal{H}_0] = -D_{KL}[p(\mathbf{R}|\mathcal{H}_0)||p(\mathbf{R}|\mathcal{H}_1)], \tag{15.15}$$
$$\mu_n'(1) = -\mathbb{E}[s(\mathbf{R})|\mathcal{H}_1] = D_{KL}[p(\mathbf{R}|\mathcal{H}_1)||p(\mathbf{R}|\mathcal{H}_0)], \tag{15.16}$$

where $D_{KL}[f||g]$ denotes the Kullback-Leibler distance between distributions f and g. This means that the error exponent characteristic has the following endpoints:

$$t = 0: \quad E_{\mathsf{fn}}^L = \lim_{n\to\infty} n^{-1} D_{KL}[p(\mathbf{R}|\mathcal{H}_0)||p(\mathbf{R}|\mathcal{H}_1)], \tag{15.17}$$
$$t = 1: \quad E_{\mathsf{fp}}^R = \lim_{n\to\infty} n^{-1} D_{KL}[p(\mathbf{R}|\mathcal{H}_1)||p(\mathbf{R}|\mathcal{H}_0)]. \tag{15.18}$$

15.1 Spread Spectrum

As an application, let $\mathbf{R} \sim \mathcal{N}(\mathbf{0}_n, N\mathbf{I}_n)$ under $\mathcal{H}_0$ and $\mathbf{R} = \mathcal{N}(\mathbf{w}, N\mathbf{I}_n)$ under $\mathcal{H}_1$ with $\mathbf{w} = \sqrt{nP}\mathbf{e}_1$ and $\mathbf{e}_1$ the first canonical basis vector. Calculations lead to:

$$\mu_n(t) = \frac{nP}{2N} t(t-1). \tag{15.19}$$

As a consequence, $t\sqrt{\mu_n''(t)} = t\sqrt{nP/N} \overset{n\to\infty}{\longrightarrow} \infty$ if $t > 0$. Moreover, $Z_t \sim \mathcal{N}(-nP(1-2t)/2N, nP/N)$. This implies that the Chernoff bound is tighter as $n \to \infty$. This ends up with the following error exponents:

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = \frac{P}{2N}\left(t^2, (1-t)^2\right), \quad \forall t \in (0,1) \tag{15.20}$$

which can be rewritten as $E_{\mathsf{fn}} = \left(\left|\sqrt{P/2N} - \sqrt{E_{\mathsf{fp}}}\right|_+\right)^2$. In the text, $N = \sigma_X^2 + \sigma_Z^2$ when the embedder and the detector do not know the host (case 1), and $N = \sigma_Z^2$ when both of them know the host (case 3).

15.2 Improved Spread Spectrum

The embedding is inspired by Malvar and Florencio paper [132]: $\mathbf{w}(\mathbf{x}) = (\alpha - \lambda \mathbf{x}^\top \mathbf{u})\mathbf{u}$ with $0 \leq \lambda \leq 1$. Without loss of generality, we assume that $\mathbf{u} = \mathbf{e}_1$, the first canonical vector. This means that a single component of $\mathbf{R}$ has different distributions under $\mathcal{H}_0$ and $\mathcal{H}_1$. Parameter α is set to satisfy the distortion constraint: $\alpha^2 + \lambda^2\sigma_X^2 \leq nP$. Asymptotically as $n \to \infty$, this constraint no longer restricts the value of λ and we can set λ to one, which operates a perfect host rejection on this component. We have $R(1) \sim \mathcal{N}(0, \sigma_X^2 + \sigma_Z^2)$ under $\mathcal{H}_0$ and $R(1) \sim \mathcal{N}(\sqrt{nP - \sigma_X^2}, \sigma_Z^2)$ under $\mathcal{H}_1$. Calculations lead to:

$$\lim_{n\to\infty} n^{-1}\mu_n(t) = \frac{P}{2}\frac{t(t-1)}{\sigma_Z^2 + t\sigma_X^2}. \tag{15.21}$$

This ends up with the following error exponents:

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) \;=\; \frac{P}{2(\sigma_Z^2 + t\sigma_X^2)^2}\left(t^2(\sigma_Z^2 + \sigma_X^2), (1-t)^2\sigma_Z^2\right) \tag{15.22}$$

The two end-points of the characteristic are given by:

$$t \;=\; 0: \quad E_{\mathsf{fn}}^L = \frac{P}{2\sigma_Z^2}, \tag{15.23}$$

$$t \;=\; 1: \quad E_{\mathsf{fp}}^R = \frac{P}{2(\sigma_Z^2 + \sigma_X^2)} \tag{15.24}$$

This characteristic lies in between the characteristics (15.20) with $N = \sigma_Z^2 + \sigma_X^2$ (lower bound of case 1) and $N = \sigma_Z^2$ (upper bound of case 3). On the left endpoint, it converges to the upper bound while on the right endpoint, it converges to the lower bound. Since we are interested in the right endpoint, ISS does not bring any value.

15.3 Output of a score function

Suppose a score function $s(\cdot) : \mathbb{R}^n \to \mathbb{R}$ gives a score S_n with the following distributions: $\mathcal{H}_0$: $S_n \sim \mathcal{N}(0, s_{0,n}^2)$, and $\mathcal{H}_1$: $S_n \sim \mathcal{N}(m_{1,n}, s_{1,n}^2)$. We assume that

$$\lim_{n\to\infty} \frac{m_1}{\sqrt{n}s_{0,n}} = H, \quad \text{and} \quad \lim_{n\to\infty} \frac{s_{1,n}^2}{s_{0,n}^2} = 1 + G. \tag{15.25}$$

Then, we have

$$\lim_{n\to\infty} n^{-1}\mu_n(t) = \lim_{n\to\infty} -\frac{m_{1,n}^2}{2ns_{0,n}^2}\frac{s_{1,n}^2}{s_{0,n}^2}t\left(1-t\frac{s_{t,n}^2}{s_{1,n}^2}\right), \tag{15.26}$$

with

$$s_{t,n}^2 = s_{1,n}^2\frac{s_{0,n}^2}{s_{1,n}^2+t(s_{0,n}^2-s_{1,n}^2)}. \tag{15.27}$$

This makes

$$\lim_{n\to\infty} n^{-1}\mu_n(t) = \lim_{n\to\infty} -\frac{H^2(1+G)^2}{2}\frac{t(1-t)}{1+G(1-t)}. \tag{15.28}$$

Using (15.11) and (15.12), we obtain a parametric definition of the characterstic

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = \frac{H^2}{2(1+G(1-t))^2}\left(t^2, (1-t)^2(1+G)\right), \quad \forall t\in[0,1]. \tag{15.29}$$

15.4 ZATT

In the ZATT scheme, $n-k$ components of vector $\mathbf{X}$ are not modified by the embedding whereas k components are cancelled. Denote $\mathbf{R}_k$ the part of the vector concerned by watermarking. We have $\mathbf{R}_k \sim \mathcal{N}(\mathbf{0}_k, (\sigma_X^2+\sigma_Z^2)\mathbf{I}_n)$ under $\mathcal{H}_0$ and $\mathbf{R}_k \sim \mathcal{N}(\mathbf{0}_k, \sigma_Z^2\mathbf{I}_n)$ under $\mathcal{H}_1$. Calculations lead to:

$$\mu_n(t) = \frac{k}{2}\left(t\log(1+\rho) - \log\left(1-\rho t\right)\right), \tag{15.30}$$

with $\rho := {}^{\sigma_X^2}\!/_{\sigma_Z^2}$. The distortion constraint imposes $k = {}^{nP}\!/_{\sigma_X^2}$. As a consequence, $t\sqrt{\mu_n''(t)} = t\sqrt{{}^{nP}\!/_{\sigma_X^2}\rho/(1+t\rho)} \overset{n\to\infty}{\longrightarrow} \infty$ if $t>0$. This ends up with the following error exponents:

$$(E_{\mathsf{fp}}, E_{\mathsf{fn}}) = \frac{P}{2\sigma_Z^2}\left(\frac{\log(1+t\rho)}{\rho} - \frac{t}{1+t\rho}, \frac{1}{\rho}\log\frac{1+t\rho}{1+\rho} + \frac{(1-t)}{1+t\rho}\right). \tag{15.31}$$

For any $t\in(0,1)$, the point of the characteristic $(E_{\mathsf{fp}}, E_{\mathsf{fn}})$ tends to $(+\infty, \log t - 1 + {}^1\!/_t)$ when $\rho\to+\infty$, *i.e.* $\sigma_Z^2\to 0$. Then, E_{fn} can be set as big as possible by driving t close to 0. This is not surprising: Deciding $d=1$ if $r(1)^2=0$ and $d=0$ otherwise gives a perfect test (*i.e.* $\mathbb{P}_{\mathsf{fp}} = \mathbb{P}_{\mathsf{fn}} = 0$) in the noiseless setup. This even holds when n is finite provided that $n \geq \sigma_X^2/P$. This scheme has little interest in practice, but it stresses the fact that, under the noiseless setup, it is very easy to design a watermarking scheme achieving perfect performances.

The two end-points of the characteristic are given by:

$$t = 0: \quad E_{\mathsf{fn}}^L = \frac{P}{2\sigma_Z^2}\left(1-\frac{\log(1+\rho)}{\rho}\right), \tag{15.32}$$

$$t = 1: \quad E_{\mathsf{fp}}^R = \frac{P}{2\sigma_Z^2}\left(\frac{\log(1+\rho)}{\rho} - \frac{1}{1+\rho}\right). \tag{15.33}$$

ZATT fails reaching the upper bounds ${}^P\!/_{2\sigma_Z^2}$ on both endpoints, except in the 'high SNR regime', *i.e.* $\rho\to+\infty$, and for the left endpoint only.

Chapter 16

Computation of error exponents with Laplace method

We summarize here a method applied below to many cases in order to compute error exponent of probabilities of false negative or positive. All these problems share the following characteristic: They involved several independent random variables, say $R_1, \cdots, R_\ell$, which are either Gaussian distributed or scaled chi distributed. R_i follows the scaled chi distribution $\sigma - \chi_k$ with k degrees of freedom if ${}^{R_i}/_{\sigma} \sim \chi_k$. In other words, its density is:

$$f_{R_i}(r_i) = \frac{2}{\sigma 2^{\frac{k}{2}} \Gamma(\frac{k}{2})} \left(\frac{r_i}{\sigma}\right)^{k-1} e^{-\frac{r_i^2}{\sigma^2}}. \tag{16.1}$$

This should not be confused with the chi-squared distribution χ_k^2.

The probability to be estimated equals the integral of the product of these densities over a domain $\mathcal{D}_n$, which depends on the dimension n of the space. The change of variables: $r_i = \sqrt{n}\tilde{r}_i$ (for all the involved random variables) modifies $\mathcal{D}_n$ into a domain $\tilde{\mathcal{D}}$ which no longer depends on n. In this way:

$$\mathbb{P} = \int_{\tilde{\mathcal{D}}} K_n g(\tilde{r}_1, \cdots, \tilde{r}_\ell) e^{-nh(\tilde{r}_1, \cdots, \tilde{r}_\ell)} \partial\tilde{r}_1 \cdots \partial\tilde{r}_\ell. \tag{16.2}$$

On one hand, we compute the 'exponent' of the multiplicative constant: $\kappa = \lim_{n\to+\infty} -{}^1/_n \log K_n$. On the other hand, the Laplace method states that, as $n \to +\infty$, the integral is dominated by the value $e^{-nh^\star}$ where $h^\star$ is the minimum of function $h(\cdot)$ over $\tilde{\mathcal{D}}$ provided that i) function $g(\cdot)$ takes a finite and non null value at this minimizer, ii) the second derivative of $h(\cdots)$ doesn't cancel at this minimizer. Then, the error exponent of the probability is given by:

$$E = \min_{\tilde{\mathcal{D}}} h(\tilde{r}_1, \cdots, \tilde{r}_\ell) + \kappa. \tag{16.3}$$

16.1 Single hypercone

16.1.1 Without side Information: Shannon's expression

This appendix derives the false negative error exponent when there is one source of noise $\mathbf{N} \sim \mathcal{N}(\mathbf{0}_n, N\mathbf{I}_n)$ added to the transmitted signal $\mathbf{W} = \sqrt{nP}\mathbf{u}$ (without loss of generality, $\mathbf{u} = \mathbf{e}_1$, the first vector of the canonical basis of $\mathbb{R}^n$), and the detection region is the single circular hypercone $\mathcal{C}$ of apex $\mathbf{0}_n$, axis $\mathbf{u}$, and half angle θ. We introduce $R = \sqrt{\sum_{i=2}^n N_i^2} \sim \sqrt{N} - \chi_{n-1}$ and the first

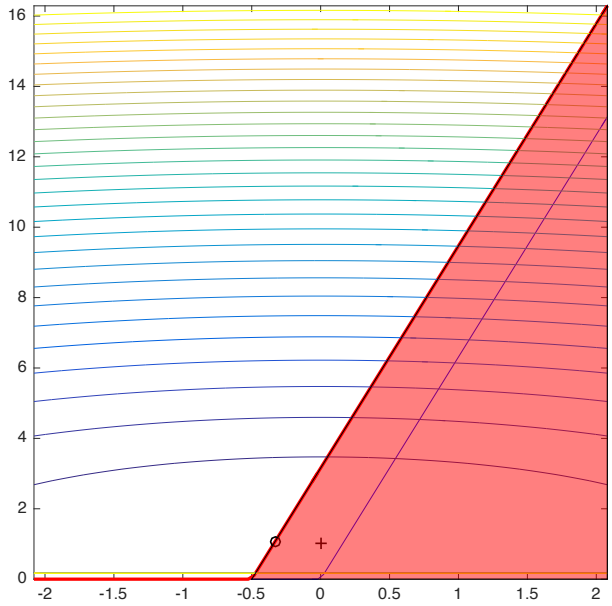


Figure 16.1: Laplace method for the single hypercone with no side information. The red area is the embedding region in the plane $(\tilde{n}_1, \tilde{r})$. E_{fn} is related the minimum of the potential function over the domain $\tilde{\mathcal{D}}$ (the complement of the red region). The levels set of the potential function are depicted in colors. Here, global minimum (black $+$) is not in $\tilde{\mathcal{D}}$. The local minimum (black **o**) lies on the boundary.

component of $\mathbf{N}$ $N_1 \sim \mathcal{N}(0, N)$ so that:

$$\begin{aligned} \mathbb{P}_{\mathsf{fn}} &= \mathbb{P}((\mathbf{N}+\mathbf{W}) \notin \mathcal{C}) = \mathbb{P}((R, N_1) \in \mathcal{D}), & (16.4)\\ \mathcal{D}_n &= \{(r, n_1) \in \mathbb{R}^+ \times \mathbb{R} | r > (n_1 + \sqrt{nP}) \tan\theta\}. & (16.5) \end{aligned}$$

Applying the method above mentioned gives:

$$\begin{aligned} E_{\mathsf{fn}} &= \min_{\tilde{\mathcal{D}}} \left(\frac{\tilde{n}_1^2}{2N} + S\left(\frac{\tilde{r}^2}{N}\right)\right) & (16.6)\\ \tilde{\mathcal{D}} &= \{(\tilde{r}, \tilde{n}_1) \in \mathbb{R}^+ \times \mathbb{R} : \tilde{r} > (\tilde{n}_1 + \sqrt{P}) \tan\theta\}. & (16.7) \end{aligned}$$

with $S(x) := 1/2(x - 1 - \log(x))$.

Function $S(\cdot)$ has a unique minimum at $S(1) = 0$. Therefore, $E_{\mathsf{fn}} = 0$ when $(\tilde{r}, \tilde{n}_1) = (\sqrt{N}, 0)$ is an admissible solution, *i.e.* it belongs to $\tilde{\mathcal{D}}$. This is true if and only if $\tan\theta \leq \tan\theta_0$ with $\tan\theta_0 = A^{-1}$ as denoted by Shannon.

By mapping back this minimum to the random vector $\mathbf{N}$, we have the following interpretation: Knowing $\mathbf{u}$ is fixed, a typical realization of $\mathbf{N}$ has a norm concentrating around $\sqrt{nN}$ while being orthogonal to $\mathbf{u}$, as $n \to \infty$. The hypercone is so thin or P, whence A, is so small that this realization pushes the transmitted signal outside the hypercone with a high probability.

If $\tan\theta > \tan\theta_0$, this global minimum is no longer admissible. It also implies that the gradient of the objective function never cancels over $\tilde{\mathcal{D}}$ since this global minimum is unique. Therefore, the minimum $(\tilde{r}^\star, \tilde{n}_1^\star)$ is on the boundary of $\tilde{\mathcal{D}}$. This cannot be on the boundary $\{\tilde{r} = 0\}$ because the objective function takes infinite values there. This shows that only the other boundary remains, which is the half line $\{\tilde{r} = (\tilde{n}_1 + \sqrt{P}) \tan\theta, \tilde{n}_1 > -A\}$. We are now looking for

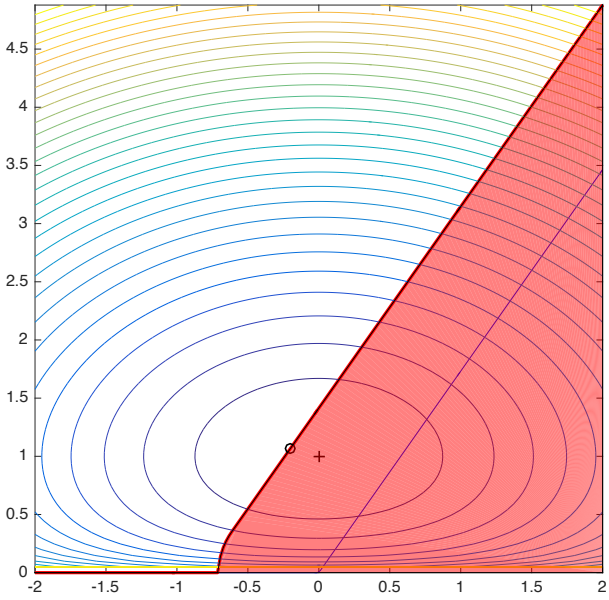


Figure 16.2: Laplace method for the single hypercone with side information in the noiseless scenario. The red area is the embedding region in the plane $(\tilde{n}_1, \tilde{r})$. E_{fn} is related the minimum of the potential function over the domain $\tilde{\mathcal{D}}$ (the complement of the red region). The levels set of the potential function are depicted in colors. Here, the global minimum (black $+$) is not in $\tilde{\mathcal{D}}$. The local minimum (black **o**) lies on the boundary.

the minimum of a single variate function that we find by cancelling its derivative. It happens for

$$\tilde{r}^\star = \frac{\sqrt{N}}{2}\left(A\cos\theta + \sqrt{A^2\cos^2\theta + 4}\right)\sin\theta > 0. \tag{16.8}$$

With Shannon's notations, we recognize that $\tilde{r}^\star = \sqrt{N}.G\sin\theta$. In the end

$$E_{\mathsf{fn}} = S(G^2\sin^2\theta) + \frac{(G\sin\theta\tan^{-1}\theta - A)^2}{2}, \tag{16.9}$$

which equals the result of Shannon (8.11).

The Laplace method tells that $\mathbb{P}_{\mathsf{fn}}$ is dominated by a typical realization, which (perhaps counter-intuitively) is neither orthogonal to $\mathbf{u}$, nor the shortest norm realization driving the transmitted signal outside the hypercone (see Fig. 16.1).

16.1.2 With side information: noiseless scenario

In the noiseless scenario, the objective function remains the same but the domain is now described by the rolling ball technique over the hypercone. Domain $\tilde{\mathcal{D}}$ is now a 'smoothed' hypercone defined by [147, Example 4.1]:

$$\tilde{r} \geq \begin{cases} 0 & \text{if } \tilde{n}_1 \leq -\sqrt{P} \\ \sqrt{P - \tilde{n}_1^2} & \text{if } -\sqrt{P} \leq \tilde{n}_1 \leq -\sqrt{P}\sin\theta \\ \tilde{n}_1\tan\theta + \frac{\sqrt{P}}{\cos\theta} & \text{if } \tilde{n}_1 \geq -\sqrt{P}\sin\theta \end{cases} \tag{16.10}$$

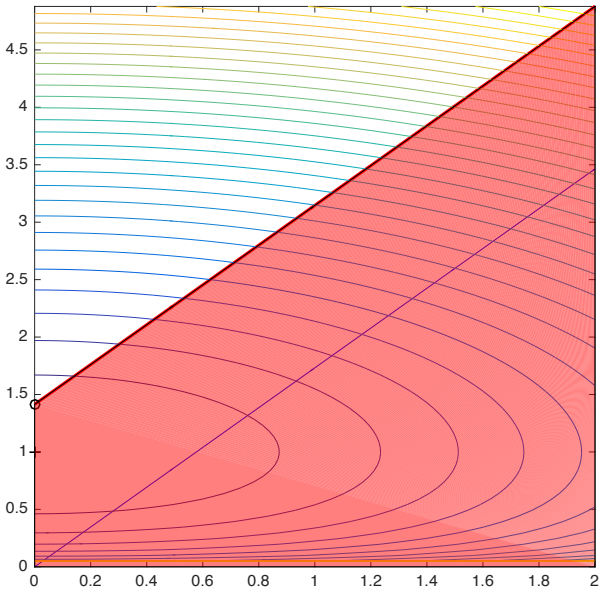


Figure 16.3: Laplace method for the dual hypercone with side information in the noiseless scenario. The red area is the embedding region in the plane $(\tilde{n}_1, \tilde{r})$. E_{fn} is related the minimum of the potential function over the domain $\tilde{\mathcal{D}}$ (the complement of the red region). The levels set of the potential function are depicted in colors. Here, the global minimum (black $+$) is not in $\tilde{\mathcal{D}}$. The local minimum (black **o**) lies on the boundary.

The same rationale as above shows that $E_{\mathsf{fn}} = 0$ if and only if $A \leq \cos\theta$. Otherwise, the minimum in $\tilde{\mathcal{D}}$ is on its boundary (see Fig. 16.2). A study of the three cases in the above equation shows that

$$\tilde{r}^\star = \frac{\sigma_X}{2}\left(A\cos\theta + \sqrt{A^2\cos^2\theta + 4\sin^2\theta}\right) \tag{16.11}$$

$$\tilde{n}_1^\star = \frac{\tilde{r}^\star}{\tan\theta} - \frac{\sqrt{P}}{\sin\theta}. \tag{16.12}$$

In the end, $E_{\mathsf{fn}} = S(\tilde{r}^{\star 2}/\sigma_X^2) + \tilde{n}_1^{\star 2}/2\sigma_X^2$.

16.2 Dual hypercone

16.2.1 Noiseless scenario

In the noiseless scenario, watermarking is successful in embedding region $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^n : \sqrt{\sum_{i=2}^n x_i^2} \leq |x_1|\tan\theta + \sqrt{nP}/\cos\theta\}$. Random variable $R_n = \sqrt{\sum_{i=2}^n X_i^2}$ is distributed as $\sigma_X - \chi_{n-1}$ while $R_1 = |X_1| \sim \sigma_X - \chi_1$. The method yields the definition:

$$E_{\mathsf{fn}} = \min_{\tilde{\mathcal{D}}} \frac{\tilde{r}_1^2}{2\sigma_X^2} + S\left(\frac{\tilde{r}_n^2}{\sigma_X^2}\right), \text{ with} \tag{16.13}$$

$$\tilde{\mathcal{D}} = \{(\tilde{r}_n, \tilde{r}_1) \in \mathbb{R}^+ \times \mathbb{R}^+ | \tilde{r}_n \geq \tilde{r}_1\tan\theta + \sqrt{P}/\cos\theta\}. \tag{16.14}$$

The solution is the following:

$$E_{\mathsf{fn}} = \begin{cases} 0, & \text{if } A \leq \cos\theta \quad (\tilde{r}_1^\star = 0, \tilde{r}_{n-1}^\star = \sigma_X) \\ S\left(\frac{A^2}{\cos^2\theta}\right) & \text{otherwise} \quad (\tilde{r}_1^\star = 0, \tilde{r}_{n-1}^\star = \sqrt{P}/\cos\theta) \end{cases} \tag{16.15}$$

The second case comes from the minimization of the functional over the boundary $\tilde{r}_n = \tilde{r}_1 \tan\theta + \sqrt{P}/\cos\theta$. This yields a univariate function in $\tilde{r}_1$ whose derivative is a polynomial of order 2 with positive coefficients. This polynomial always takes positive value for $\tilde{r}_1 \geq 0$. This shows that the minimum happens for the smallest value of $\tilde{r}_1$, *i.e.* $\tilde{r}_1^\star = 0$ (see Fig. 16.3).

Probability $\mathbb{P}_{\mathsf{fn}}$ is dominated by the probability that $\mathbf{X}$ lies around the closest point to the origin in $\bar{\mathcal{E}}$. If this minimum distance $\sqrt{nP}/\cos\theta$ is lower than the typical module of $\mathbf{X}$, *i.e.* $\sqrt{n}\sigma_X$, then watermarking fails almost surely as $n \to \infty$ so that $E_{\mathsf{fn}} = 0$.

16.3 k dimensional ruff

16.3.1 Noiseless scenario

Hypothesis $\mathcal{H}_0$

A false positive occurs when $\sqrt{\sum_{i=1}^k X_i^2}\tan\theta \geq \sqrt{\sum_{i=k+1}^n X_i^2}$. We introduce $R_1 = \sqrt{\sum_{i=1}^k X_i^2}$ and $R_n = \sqrt{\sum_{i=k+1}^n X_i^2}$, which are two random variables following a scaled chi distribution with degree of freedom k and $n-k$ respectively. We again follow the same line to end up with:

$$E_{\mathsf{fp}} = \min_{\tilde{\mathcal{D}}} \rho S\left(\frac{\tilde{r}_1^2}{\rho\sigma_X^2}\right) + (1-\rho)S\left(\frac{\tilde{r}_n^2}{(1-\rho)\sigma_X^2}\right) \tag{16.16}$$

$$\tilde{\mathcal{D}} = \{(\tilde{r}_1, \tilde{r}_n) \in \mathbb{R}^+ \times \mathbb{R}^+ | \tilde{r}_1 \tan\theta \geq \tilde{r}_n\}. \tag{16.17}$$

Error exponent $E_{\mathsf{fp}} = 0$ if and only if $(\tilde{r}_1, \tilde{r}_n) = \sigma_X(\sqrt{\rho}, \sqrt{1-\rho})$ is admissible, which translates to $\cos\theta \leq \sqrt{\rho}$. Otherwise, the minimum lies on the boundary $\{\tilde{r}_1 \tan\theta = \tilde{r}_n\}$. Then, we minimize an univariate function by canceling its unique derivative. This gives $\tilde{r}_1^\star = \sigma_X \cos\theta$, $\tilde{r}_n^\star = \sigma_X \sin\theta$. In the end,

$$E_{\mathsf{fp}} = \begin{cases} 0 & \text{if } \cos\theta \leq \sqrt{\rho} \\ \frac{1}{2}\left(\rho \log\frac{\rho}{\cos^2\theta} + (1-\rho)\log\frac{1-\rho}{\sin^2\theta}\right) & \text{otherwise.} \end{cases} \tag{16.18}$$

The last expression is half the Kullback Leibler distance between the two Bernoulli distributions $\mathcal{B}(\rho)$ and $\mathcal{B}(\cos^2\theta)$. Denote the angle α s.t. $\cos(\alpha) = \sqrt{1-\rho}$, then $E_{\mathsf{fp}} > 0$ for $0 \leq \theta < \pi/2 - \alpha$.

Hypothesis $\mathcal{H}_1$

The boundary of embedding region $\mathcal{E}$ is defined by the rolling ball technique. This translates in the following minimization problem:

$$E_{\mathsf{fn}} = \min_{\tilde{\mathcal{D}}} \rho S\left(\frac{\tilde{r}_1^2}{\rho\sigma_X^2}\right) + (1-\rho)S\left(\frac{\tilde{r}_n^2}{(1-\rho)\sigma_X^2}\right) \tag{16.19}$$

$$\tilde{\mathcal{D}} = \left\{(\tilde{r}_1, \tilde{r}_n) \in \mathbb{R}^+ \times \mathbb{R}^+ | \tilde{r}_n \geq \tilde{r}_1 \tan\theta + \frac{\sqrt{P}}{\cos\theta}\right\}. \tag{16.20}$$

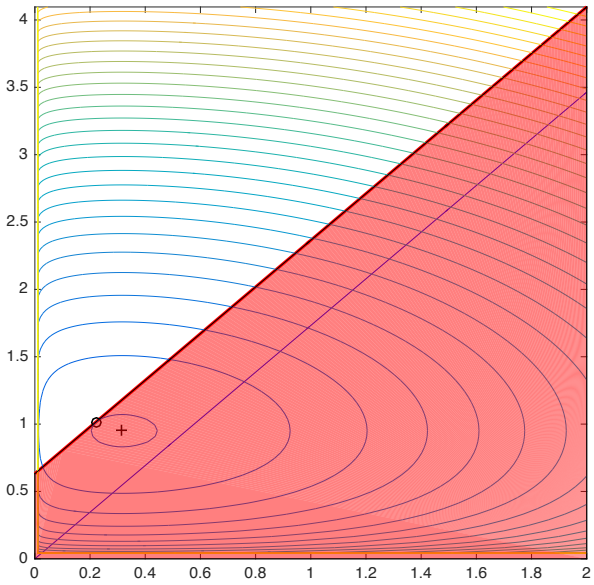


Figure 16.4: Laplace method for the ruff with side information in the noiseless scenario. The red area is the embedding region in the plane $(\tilde{n}_1, \tilde{r})$. E_{fn} is related the minimum of the potential function over the domain $\tilde{\mathcal{D}}$ (the complement of the red region). The levels set of the potential function are depicted in colored lines. Here, the global minimum (black $+$) is not in $\tilde{\mathcal{D}}$. The local minimum (black $\mathbf{o}$) lies on the boundary.

Error exponent $E_{\mathsf{fn}} = 0$ if and only if $(\tilde{r}_1, \tilde{r}_n) = \sigma_X(\sqrt{\rho}, \sqrt{1-\rho})$ is admissible, which translates to $A \leq \sqrt{1-\rho}\cos\theta - \sqrt{\rho}\sin\theta = \cos(\theta+\alpha)$. Otherwise, the minimum is on the boundary (see Fig. 16.4). The derivative of the univariate function cancels at the roots of a polynomial of order 3. These expressions are complicated and bring no insight. A numerical solver has no difficulty in estimating E_{fn}.

16.3.2 Noisy scenario

This section assumes that $\cos\theta > \sin(\alpha)$ so that $E_{\mathsf{fp}} > 0$. Error exponent E_{fn} under a noisy setup is derived following the Laplace method again. First we consider a new basis of $\mathbb{R}^n$ with $\mathbf{e}_1 = \mathbf{X}^{(k)}/\|\mathbf{X}^{(k)}\|$, $\mathbf{e}_n = \mathbf{X}^{(n-k)}/\|\mathbf{X}^{(n-k)}\|$, and the remaining basis vectors are such that $(\mathbf{e}_1, \dots, \mathbf{e}_k)$ form a basis of the first subspace, and $(\mathbf{e}_{k+1}, \dots, \mathbf{e}_n)$ form a basis of the second subspace.

Denote $X_1 := \mathbf{X}^\top \mathbf{e}_1 = \|\mathbf{X}^{(k)}\|$ and $X_n := \mathbf{X}^\top \mathbf{e}_n = \|\mathbf{X}^{(n-k)}\|$. With probability $1 - \mathbb{P}_{\mathsf{fp}}$, the host signal is not in the ruff: $X_1 \leq X_n \tan(\theta)$, with X_1 and X_n following the scaled chi distributions $\sigma_X - \chi_k$ and $\sigma_X - \chi_{n-k}$ respectively. Denote $Y_1 := \mathbf{Y}^\top \mathbf{e}_1$ and $Y_n := \mathbf{Y}^\top \mathbf{e}_n$ the components of the watermarked signal in this basis. The watermark signal $\mathbf{W}$ is here to push $\mathbf{X}$ inside the detection region. Since a successful embedding only depends on the values of Y_1 and Y_n, $\mathbf{W}$ should solely modify X_1 and X_n: $\mathbf{W} = W_1\mathbf{e}_1 + W_n\mathbf{e}_n$, with $W_1^2 + W_n^2 \leq nP$. Any other embedding strategy wastes embedding energy in space directions not useful for detecting ints presence.

The noise vector is also projected on this new basis with $Z_1 = \mathbf{Z}^\top \mathbf{e}_1 \sim \mathcal{N}(0, \sigma_Z^2)$, $Z_n = \mathbf{Z}^\top \mathbf{e}_n \sim \mathcal{N}(0, \sigma_Z^2)$. We also introduce two random variables $Q_1 = \sqrt{\sum_{i=2}^{k} Z_i^2}$, and $Q_n = \sqrt{\sum_{i=k+1}^{n-1} Z_i^2}$, which follow the scaled chi distributions $\sigma_Z - \chi_{k-1}$ and $\sigma_Z - \chi_{n-k-1}$ respectively.

With this formulation, the probability of false negative finds the following expression:

$$\mathbb{P}_{\mathsf{fn}} = \int \cdots \int_{\mathcal{D}_n} f_{X_1}(x_1) f_{X_n}(x_n) f_{Z_1}(z_1) f_{Z_n}(z_n) f_{Q_1}(q_1) f_{Q_n}(q_n) \partial x_1 \partial x_n \partial z_1 \partial z_n \partial q_1 \partial q_n, \quad (16.21)$$

with $\mathcal{D}_n \subset \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}^+$ defined by:

$$\mathcal{D}_n := \{(x_1, x_n, z_1, z_n, q_1, q_n) | \left((x_1 + w_1 + z_1)^2 + q_1^2\right) \tan^2 \theta < (x_n + w_n + z_n)^2 + q_n^2\} \quad (16.22)$$

These notations omit the fact that w_1 and w_n are indeed functions of (x_1, x_n).

The next step is a change of variables with $x_1 = \sqrt{n}\tilde{x}_1$, ..., $q_n = \sqrt{n}\tilde{q}_n$. Domain $\tilde{\mathcal{D}}$ has the same definition as $\mathcal{D}_n$ but with variables $\tilde{x}_1$, ..., $\tilde{q}_n$. This comes from the fact that the detection region is a linear hypercone: if $\mathbf{x} \in \mathbb{R}^n$ belongs to the detection region, so is $\gamma\mathbf{x}$ with $\gamma \in \mathbb{R}^{\star+}$. This translates into $\mathcal{D}$ being also (positively) linear. As for the embedding constraint, we now have $\tilde{w}_1^2 + \tilde{w}_n^2 \leq P$.

The Laplace method assesses that

$$E_{\mathsf{fn}} = \min_{\tilde{\mathcal{D}}} \frac{\tilde{z}_1^2 + \tilde{z}_n^2}{2\sigma_Z^2} + \rho S\left(\frac{\tilde{x}_1^2}{\rho\sigma_X^2}\right) + \rho S\left(\frac{\tilde{q}_1^2}{\rho\sigma_Z^2}\right) + (1-\rho) S\left(\frac{\tilde{x}_n^2}{(1-\rho)\sigma_X^2}\right) + (1-\rho) S\left(\frac{\tilde{q}_n^2}{(1-\rho)\sigma_Z^2}\right) \quad (16.23)$$

16.4 Voronoï modulation

We calculate the error exponent E of the probability $\mathbb{P}(\|\alpha\mathbf{N}+\gamma\mathbf{W}\|^2 > nb)$ where $\mathbf{N} \sim \mathcal{N}(\mathbf{0}_n, N\mathbf{I}_n)$ and $\mathbf{W} \sim \mathcal{U}_{\mathcal{V}(\Lambda_2)}$. It is shown in [110][162, Prop. 3] that E is also the error exponent of probability $\mathbb{P}(\|\alpha\mathbf{N} + \sqrt{na}\mathbf{W}'\|^2 > nb)$ where $\mathbf{W}' \sim \mathcal{U}_{\mathcal{S}_n}$, *i.e.* uniformly distributed over the hypersphere of radius 1, and $a = \gamma^2\sigma(\Lambda_2)$.

This last probability remains unchanged for any occurence of $\mathbf{W}'$ (thanks to rotational invariance), so without loss of generality, we assume $\mathbf{W}' = \mathbf{e}_1$. With the notation $\tilde{r}_1 = n^{-1}\alpha n_1$ and $\tilde{r}_n = n^{-1}\alpha\sqrt{\sum_{i=2}^n n_i^2}$ and assuming $b > a$:

$$E = \min_{\tilde{\mathcal{D}}} \frac{\tilde{r}_1^2}{2\alpha^2 N} + S\left(\frac{\tilde{r}_n^2}{\alpha^2 N}\right) \quad (16.24)$$

$$\tilde{\mathcal{D}} = \{(\tilde{r}_n, \tilde{r}_1) \in \mathbb{R}^+ \times \mathbb{R} | \tilde{r}_n^2 \geq b - (\tilde{r}_1 + \sqrt{a})^2\} \quad (16.25)$$

We have $E = 0$ if the global minimum $(\alpha\sqrt{N}, 0) \in \tilde{\mathcal{D}}$. This is possible if $\alpha^2 N + a \geq b$. Otherwise, the local minimum lies on the boundary of $\tilde{\mathcal{D}}$. It cannot be for $\tilde{r}_n = 0$ as $S(\cdot)$ takes infinity value there. It means the local minimum lies on the circle (see Fig. 16.5): $\tilde{r}_n^2 = b - (\tilde{r}_1 + \sqrt{a})^2$ with $\tilde{r}_1 \in [-\sqrt{b} - \sqrt{a}, \sqrt{b} - \sqrt{a}]$. Indeed, for a given $\tilde{r}_n \in [0, \sqrt{b}]$, $\tilde{r}_1 = \sqrt{b - \tilde{r}_n^2} - \sqrt{a}$ (the other solution yields a bigger exponent). We now have a univariate function to be minimized. The solution gives $\tilde{r}_n^\star = \alpha N\sqrt{(\sqrt{1+4xy}-1)/2y}$ with $x = b/\alpha^2 N$ and $y = a/\alpha^2 N$, so that

$$E = \frac{1}{2}\left(x + y - \sqrt{1+4xy} - \log\frac{\sqrt{1+4xy}-1}{2y}\right) \quad (16.26)$$

$$= \frac{1}{2}\left(x + y - \sqrt{1+4xy} + \log\frac{\sqrt{1+4xy}+1}{2x}\right). \quad (16.27)$$

This rationale justifies the reliability function for the Voronoï modulation scheme with a fixed α. The lattice Λ_2 being Rogers-good, its Voronoï cell becomes a ball as $n \to \infty$ whose radius scales as $\sqrt{n\sigma(\Lambda_2)}$. This means that $a = (1-\alpha)^2 P$.

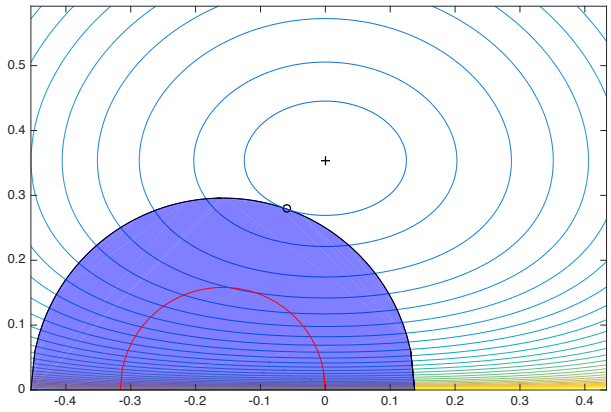


Figure 16.5: Laplace method for the Voronoï modulation in the plane $(\tilde{r}_1, \tilde{r}_n)$. E_{fn} is given by the minimum of the potential function over the domain $\tilde{\mathcal{D}}$ (outside the blue disk). The levels set of the potential function are depicted in colors. Here, the global minimum (black $+$) is not in $\tilde{\mathcal{D}}$. The local minimum (black $\mathbf{o}$) lies on the boundary. The red circle of radius $\sqrt{a}$ is the support of the random vector $\sqrt{a}\mathbf{W}'$.

With an Euclidean decoder, a decoding error occurs when $\alpha\mathbf{N}+(\alpha-1)\mathbf{W} \notin \mathcal{V}(\Lambda_1)$. Knowing that $|\mathcal{V}(\Lambda_1)|/|\mathcal{V}(\Lambda_2)| = e^{-nR}$, the efficient radius of Λ_1 is e^{-R} times smaller than the one of Λ_2. This justifies that we set $b = Pe^{-2R}$ in this case to find back the reliability function $E_{\Lambda_1/\Lambda_2}(R)$ given by (8.23) in Sect. 8.4.

For the detection case where there is no decoding on Λ_1 (See Sect. 10.2.1), the probability corresponds to a false negative probability and E is indeed the false negative error exponent E_{fn}, knowing that $b = Pe^{-2E_{\mathsf{fp}}}$ (See (10.36)).

For a fixed lattice Λ_2 as in Sect. 10.2.2, $\gamma = (\alpha\beta - 1)$ where β scales the watermark signal to ensure $\|\mathbf{W}\|^2 = nP$.

Chapter 17

Expressions for JANIS with order k

The expression of the detector (11.19) ensures that, under $\mathcal{H}_0$, $m_{0,n} = 0$ and $s^2_{0,n} = 1$. This is because the components the received vector are independent with zero expectation and unit variance once normalized.

The expectation and the variance under $\mathcal{H}_1$ are much more cumbersome to derive. Under the noiseless setup, we have:

$$m_{\pi,n} = \sqrt{\frac{n}{k}} A \quad \text{with} \quad A = \sum_{\ell=0}^{k} \binom{k}{\ell} M_{\ell+1}^{k-\ell} M_{\ell-1}^{\ell} \left(\frac{\pi}{\sigma_X}\right)^{\ell} \tag{17.1}$$

$$s^2_{\pi,n} = \sum_{\ell=0}^{k} \binom{k}{\ell} 2^{\ell} \sum_{i=0}^{k-\ell} \binom{k-\ell}{i} \left(\frac{\pi}{\sigma_X}\right)^{2(k-i)-\ell} M^{i}_{2(k-i)-\ell+2} M^{k-\ell-i}_{2(k-i)-\ell-2} - A^2, \tag{17.2}$$

where M_ℓ is the moment of order ℓ of the standard Gaussian law (*i.e.* $M_\ell = (\ell-1)!!$ if ℓ is even and 0 if ℓ is odd). Up to the first order, for $k \geq 2$, we have $m_{\pi,n} = {}^{\pi\sqrt{nk}}/_{\sigma_X} + o(\pi)$ and $s^2_{\pi,n} = 1 + ({}^{\pi}/_{\sigma_X})^2 k((2k+1)3^{k-1} - k) + o(\pi^2)$. This gives $H(\pi) = {}^{\pi\sqrt{k}}/_{\sigma_X} + o(\pi)$ and an efficacy $\eta = {}^{k}/_{\sigma_X^2}$. On the other hand, $G(\pi) = s^2_{\pi,n} - 1$ grows exponentially fast as k increases.

In the noisy setup, things get even more complicated. The expectation is just scaled down by a factor $\gamma = {}^{\sigma_X}/_{\sqrt{\sigma_X^2+\sigma_Z^2}}$ (this is due to the normalization by σ in (11.19) which here equals $\sqrt{\sigma_X^2 + \sigma_Z^2}$):

$$m_{\pi,n} = \gamma^k \sqrt{\frac{n}{k}} A \tag{17.3}$$

$$s^2_{\pi,n} = \gamma^{2k} \left(\sum_{a+b+c+d=k} \binom{k}{a,b,c,d} 2^c \left(\frac{\pi}{\sigma_X}\right)^{2b+c} M^{a}_{2b+c+2} M^{b}_{2b+c-2} \left(\frac{\sigma_Z}{\sigma_X}\right)^{2d} - E^2 \right) \tag{17.4}$$

Bibliography

[80] E. Abbe and L. Zheng, "Linear universal decoding for compound channels," *IEEE Trans. on Inf. Theory*, vol. 56, no. 12, pp. 5999–6013, december 2010.

[81] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in *Proc. of 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009)*, jan 2009.

[82] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1880–1901, March 2012.

[83] A. Barg, G. R. Blakley, and G. A. Kabatiansky, "Digital fingerprinting codes: problem statements, constructions, identification of traitors," *IEEE Trans. Inform Theory*, vol. 49, no. 4, pp. 852–865, apr 2003.

[84] M. Barni, F. Bartolini, A. de Rosa, and A. Piva, "Optimum decoding and detection of multiplicative watermarks," *IEEE Trans. Signal Processing*, vol. 51, no. 4, pp. 1118–1123, apr 2003.

[85] M. Barni, F. Bartolini, A. D. Rosa, and A. Piva, "A new decoder for the optimum recovery of non-additive watermarks," *IEEE Trans. on Image Processing*, vol. 5, pp. 755–66, 2001.

[86] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, 1st ed., ser. Signal Processing and Communications. Marcel Dekker, 2004.

[87] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Review*, vol. 38, no. 3, pp. 367–426, 1996. [Online]. Available: https://doi.org/10.1137/S0036144593251710

[88] O. Blayer and T. Tassa, "Improved versions of Tardos' fingerprinting scheme," *Des. Codes Cryptography*, vol. 48, no. 1, pp. 79–103, 2008.

[89] D. Boesten and B. Škorić, *Asymptotic Fingerprinting Capacity for Non-binary Alphabets*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–13. [Online]. Available: https://doi.org/10.1007/978-3-642-24178-9_1

[90] ——, *Asymptotic Fingerprinting Capacity in the Combined Digit Model*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 255–268. [Online]. Available: https://doi.org/10.1007/978-3-642-36373-3_17

[91] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1897–1905, September 1998.

[92] A. Briassouli and M. Strinzis, "Locally optimum nonlinearities for DCT watermarking detection," *IEEE Trans. Image Processing*, vol. 13, no. 12, pp. 1604–16 017, dec 2004.

[93] J.-F. Cardoso, "Dependence, correlation and gaussianity in independent component analysis," *J. Mach. Learn. Res.*, vol. 4, no. 7-8, pp. 1177–1203, Oct. 2004. [Online]. Available: http://dx.doi.org/10.1162/jmlr.2003.4.7-8.1177

[94] M. Chaumont and D. Goudia, "TCQ practical evaluation in the hyper-cube watermarking framework," in *2011 IEEE International Conference on Multimedia and Expo*, July 2011, pp. 1–6.

[95] B. Chen and G. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory*, vol. 4è, pp. 1423–1443, May 2001.

[96] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *Information Theory, IEEE Transactions on*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[97] Q. Cheng and T. Huang, "Robust optimum detection of transform domain multiplicative watermarks," *IEEE Trans. Sig. Processing*, vol. 51, no. 4, pp. 906–924, apr 2003.

[98] B. Clarke and A. Barron, "Jeffreys' prior is asymptotically least favorable under entropy risk," *Journal of Stat. Planning and Inference*, vol. 41, pp. 37–60, 1994.

[99] P. Comesana, N. Merhav, and M. Barni, "Asymptotically optimum universal watermark embedding and detection in the high-snr regime," *Information Theory, IEEE Transactions on*, vol. 56, no. 6, pp. 2804–2815, June 2010.

[100] M. Costa, "Writing on dirty paper," *IEEE Trans. on Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[101] ——, "Writing on dirty paper (corresp.)," *Information Theory, IEEE Transactions on*, vol. 29, no. 3, pp. 439–441, May 1983.

[102] T. Cover and J. Thomas, *Elements of information theory*, ser. Wiley series in telecommunications. Wiley, 1991, no. ISBN-0-471-06259-6.

[103] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking*, 2nd ed. Morgan Kaufmann Publisher, 2008.

[104] I. J. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *Image Processing, IEEE Transactions on*, vol. 6, no. 12, pp. 1673–1687, Dec 1997.

[105] I. J. Cox, M. Miller, and A. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, Jul 1999.

[106] J. Delhumeau, T. Furon, G. Silvestre, and N. Hurley, "Improved Polynomial Detectors for Side-Informed Watermarking," in *Security and Watermarking of Multimedia Contents IV*, P. W. Wong and E. Delp, Eds. San José, CA, USA, United States: SPIE, Jan. 2002, pp. 311–321. [Online]. Available: https://hal.inria.fr/inria-00080827

[107] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *ACM Int. Conference on MultiMedia*, Oct. 2013.

[108] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. pp. 1–38, 1977. [Online]. Available: http://www.jstor.org/stable/2984875

[109] M. Desoubeaux, C. Herzet, W. Puech, and G. Le Guelvouit, "Enhanced Blind Decoding of Tardos Codes with New Map-Based Functions," in *IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, Pula, Italie, Oct. 2013. [Online]. Available: http://hal.inria.fr/hal-00907670

[110] U. Erez and R. Zamir, "Achieving 1/2 log (1+snr) on the awgn channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293–2314, Oct 2004.

[111] G. Fodor, P. Schelkens, and A. Dooms, "Fingerprinting codes under the weak marking assumption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1495–1508, June 2018.

[112] C. Fontaine, S. Gambs, J. Lolive, and C. Onete, "Private asymmetric fingerprinting : a protocol with optimal traitor tracing using Tardos codes," in *Third International Conference on Cryptology and Information Security in Latin America (Latincrypt'14)*, vol. 8895 - LNCS (Lecture Notes in Computer Science), Florianopolis, Brazil, Sep. 2014. [Online]. Available: https://hal.inria.fr/hal-01090053

[113] T. Furon and P. Duhamel, "An asymmetric watermarking method," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 981–995, April 2003, special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery.

[114] T. Furon, G. Silvestre, and N. Hurley, "JANIS: Just Another N-order side-Informed Scheme," in *Proc. of Int. Conf. on Image Processing ICIP'02*, vol. 2, Rochester, NY, USA, September 2002, pp. 153–156.

[115] R. Gallager, "A simple derivation of the coding theorem and some applications," *IEEE Transactions on Information Theory*, vol. 11, no. 1, pp. 3–18, Jan 1965.

[116] A. Guyader, N. Hengartner, and E. Matzner-Løber, "Simulation and estimation of extreme quantiles and extreme probabilities," *Applied Mathematics & Optimization*, vol. 64, no. 2, pp. 171–196, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00245-011-9135-z

[117] Y. W. Huang and P. Moulin, "On the fingerprinting capacity games for arbitrary alphabets and their asymptotics," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 9, pp. 1477–1490, Sept 2014.

[118] Y.-W. Huang and P. Moulin, "Saddle-point solution of the fingerprinting capacity game under the marking assumption," in *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 4*, ser. ISIT'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2256–2260. [Online]. Available: http://portal.acm.org/citation.cfm?id=1700967.1700985

[119] ——, "On the saddle-point solution and the large-coalition asymptotics of fingerprinting games," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 160–175, 2012.

[120] S. Ibrahimi, B. Škorić, and J.-J. Oosterwijk, "Riding the saddle point: asymptotics of the capacity-achieving simple decoder for bias-based traitor tracing," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 12, Aug 2014. [Online]. Available: https://doi.org/10.1186/s13635-014-0012-6

[121] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. Journal on Computer Vision*, vol. 87, no. 3, pp. 316–336, Feb. 2010.

[122] S. Katzenbeisser, B. Skoric, M. Celik, and A.-R. Sadeghi, "Combining Tardos fingerprinting codes and fingercasting," in *Proc. of 9th Information Hiding*, ser. Lecture Notes in Computer Science, S. Verlag, Ed., vol. 4567, 2007.

[123] S. Katzenbeisser, B. Škorić, M. U. Celik, and A.-R. Sadeghi, "Combining tardos fingerprinting codes and fingercasting," in *Information Hiding*, T. Furon, F. Cayre, G. Doërr, and P. Bas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 294–310.

[124] M. Kuribayashi, "Simplified map detector for binary fingerprinting code embedded by spread spectrum watermarking scheme," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 610–623, April 2014.

[125] ——, *Tardos's Fingerprinting Code over AWGN Channel*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 103–117. [Online]. Available: https://doi.org/10.1007/978-3-642-16435-4_9

[126] T. Laarhoven, "Capacities and capacity-achieving decoders for various fingerprinting games," in *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, June 2014.

[127] T. Laarhoven and B. de Weger, "Optimal symmetric tardos traitor tracing schemes," *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 83–103, Apr 2014. [Online]. Available: https://doi.org/10.1007/s10623-012-9718-y

[128] T. Laarhoven and B. d. Weger, "Discrete distributions in the tardos scheme, revisited," in *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec '13. New York, NY, USA: ACM, 2013, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/2482513.2482533

[129] J. G. Liao and O. Rosen, "Fast and stable algorithms for computing and sampling from the noncentral hypergeometric distribution," *The American Statistician*, vol. 55, no. 4, pp. 366–369, nov. 2001.

[130] T. Liu and P. Moulin, "Error exponents for one-bit watermarking," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 3, April 2003, pp. III–65–8 vol.3.

[131] T. Liu, P. Moulin, and R. Koetter, "On error exponents of modulo lattice additive noise channels," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 454–471, Feb 2006.

[132] H. Malvar and D. Florencio, "Improved spread spectrum: A new modulation technique for robust watermarking," *Trans. Sig. Proc.*, vol. 51, no. 4, pp. 898–905, Apr. 2003. [Online]. Available: http://dx.doi.org/10.1109/TSP.2003.809385

[133] B. Mathon, P. Bas, F. Cayre, and B. Macq, "Impacts of Watermarking Security on Tardos-based Fingerprinting," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 1038 – 1050, Apr. 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00813362

[134] N. Merhav and M. Feder, "Universal prediction," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2124–2147, October 1998.

[135] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec 2005.

[136] P. Moulin, "Universal fingerprinting: capacity and random-coding exponents," vol. arXiv:0801.3837, January 2008, preprint available at `http://arxiv.org/abs/0801.3837`.

[137] National Institute of Standards and Technology. Digital library of mathematical functions. [Online]. Available: http://dlmf.nist.gov/

[138] G. E. Noether, "On a theorem of pitman," *Ann. Math. Statist.*, vol. 26, no. 1, pp. 64–68, 03 1955. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177728593

[139] K. Nuida, M. Hagiwara, H. Watanabe, and H. Imai, "Optimization of memory usage in tardos' fingerprinting codes," January 2008, arXiv:cs/0610036v3.

[140] J. J. Oosterwijk, B. Škorić, and J. Doumen, "A capacity-achieving simple decoder for bias-based traitor tracing schemes," *IEEE Transactions on Information Theory*, vol. 61, no. 7, pp. 3882–3900, July 2015.

[141] J.-J. Oosterwijk, B. Škorić, and J. Doumen, "A capacity-achieving simple decoder for bias-based traitor tracing schemes," Cryptology ePrint Archive, Report 2013/389, 2013, http://eprint.iacr.org/.

[142] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Int. Conference on Computer Vision and Pattern Recognition*, Jun. 2010.

[143] E. Sabbag and N. Merhav, "Optimal watermark embedding and detection strategies under limited detection resources," in *Information Theory, 2006 IEEE International Symposium on*, July 2006, pp. 173–177.

[144] R. Safavi-Naini and Y. Wang, "Collusion-secure q-ary fingerprinting for perceptual content," in *Proc. Security and Privacy in Digital Rights Management, SPDRM'01*, ser. Lecture Notes in Computer Science, Springer-Verlag, Ed., vol. 2320, 2001, pp. 57–75.

[145] H. G. Schaathun, "Attacks on kuribayashi's fingerprinting scheme," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 607–609, April 2014.

[146] M. Schäfer, S. Mair, W. Berchtold, and M. Steinebach, "Universal threshold calculation for fingerprinting decoders using mixture models," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec '15. New York, NY, USA: ACM, 2015, pp. 109–114. [Online]. Available: http://doi.acm.org/10.1145/2756601.2756611

[147] A. Seeger, "Smoothing a nondifferentiable convex function: the technique of the rolling ball," *Revista de Matematicas Aplicadas*, vol. 18, 1997.

[148] Z. Shahid, M. Chaumont, and W. Puech, "Spread spectrum-based watermarking for Tardos code-based fingerprinting for H.264/AVC video," in *2010 IEEE International Conference on Image Processing*, Sept 2010, pp. 2105–2108.

[149] ——, "H.264/AVC video watermarking for active fingerprinting based on Tardos code," *Signal, Image and Video Processing*, vol. 7, no. 4, pp. 679–694, Mar. 2013. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-00807061

[150] C. E. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, vol. 27, 1948.

[151] ——, "Probability of error for optimal codes in a gaussian channel," *Bell System Tech. J.*, vol. 38, pp. 611–656, 1959.

[152] A. Simone and B. Škorić, "False positive probabilities in q-ary tardos codes: comparison of attacks," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 519–542, Jun 2015. [Online]. Available: https://doi.org/10.1007/s10623-014-9937-5

[153] B. Škorić, "Tally-based simple decoders for traitor tracing and group testing," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1221–1233, June 2015.

[154] B. Skoric, S. Katzenbeisser, and M. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Designs, Codes and Cryptography*, vol. 46, no. 2, pp. 137–166, February 2008.

[155] B. Skoric, S. Katzenbeisser, H. G. Schaathun, and M. U. Celik, "Tardos fingerprinting codes in the combined digit model," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 906–919, Sept 2011.

[156] B. Škorić, J. J. Oosterwijk, and J. Doumen, "The holey grail a special score function for non-binary traitor tracing," in *2013 IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov 2013, pp. 180–185.

[157] B. Skoric, T. Vladimirova, M. Celik, and J. Talstra, "Tardos fingerprinting is better than we thought," *IEEE Trans. on Inf. Theory*, vol. 54, no. 8, August 2008, arXiv:cs/0607131v1.

[158] B. Škorić and J.-J. Oosterwijk, "Binary and q-ary tardos codes, revisited," *Designs, Codes and Cryptography*, vol. 74, no. 1, Jan 2015.

[159] A. Somekh-Baruch and N. Merhav, "On the capacity game of private fingerprinting systems under collusion attacks," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 884–899, 2005.

[160] J. R. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1042–1049, mar 2001.

[161] D. R. Stinson and R. Wei, "Combinatorial properties and construction of traceability schemes and frameproof codes," *SIAM Journal on Discrete Mathematics*, vol. 11, pp. 41–53, 1998.

[162] C. H. Swannack, U. Erez, and G. W. Wornell, "Geometric relationships between gaussian and modulo-lattice error exponents," *CoRR*, vol. abs/1308.1609, 2013. [Online]. Available: http://arxiv.org/abs/1308.1609

[163] G. Tardos, "Optimal probabilistic fingerprint codes," in *Proc. of the 35th annual ACM symposium on theory of computing*. San Diego, CA, USA: ACM, 2003, pp. 116–125. [Online]. Available: http://www.renyi.hu/~tardos/publications.html

[164] ——, "Optimal probabilistic fingerprint codes," *Journal of the ACM*, vol. 55, no. 2, May 2008.

[165] H. V. Trees, *Detection, Estimation, and Modulation Theory, Part III*. John Wiley & Sons, 2001, no. 0-471-09517-6.

[166] R. Zamir, B. Nazer, Y. Kochman, and I. Bistritz, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory*. Cambridge University Press, 2014.

[167] H. V. Zhao and K. J. R. Liu, "Traitor-within-traitor behavior forensics: Strategy and risk minimization," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 4, pp. 440–456, December 2006.